RM0432
参考手册

STM32L4Rxxx 和 STM32L4Sxxx
基于 Arm® 内核的 32 位高级 MCU

前言

本参考手册面向应用开发人员，提供有关使用 STM32L4Rxxx 和 STM32L4Sxxx 微控制器存储器与外设的完整信息。

STM32L4Rxxx 和 STM32L4Sxxx 构成微控制器系列，各产品具有不同的存储器大小、封装和外设。

有关订购信息以及器件的机械与电气特性，请参见相应的数据手册。

有关 Arm® Cortex®-M4 内核的信息，请参见 Cortex®-M4 *技术参考手册*。

相关文档

- Cortex®-M4 技术参考手册，可从 http://infocenter.arm.com 下载
- STM32L4S5xx、STM32L4S7xx 和 STM32L4S9xx 数据手册
- STM32L4R5xx、STM32L4R7xx 和 STM32L4R9xx 数据手册
- STM32F3、STM32F4、STM32L4 和 STM32L4+ 系列 Cortex®-M4 (PM0214)

目录

1 文档约定 72
1.1 寄存器相关缩写词列表 72
1.2 词汇表 73
1.3 外设可用性 73

2 系统和存储器概述 74
2.1 系统架构 74
2.1.1 I 总线 75
2.1.2 D 总线 75
2.1.3 S 总线 75
2.1.4 DMA 总线 76
2.1.5 DMA2D 总线 76
2.1.6 LCD-TFT 控制器 DMA 总线 76
2.1.7 SDMMC 控制器 DMA 总线 76
2.1.8 GFXMMU 总线 76
2.1.9 总线矩阵 76
2.2 存储器组织结构 77
2.2.1 简介 77
2.2.2 存储器映射和寄存器边界地址 78
2.3 位段 84
2.4 嵌入式 SRAM 85
2.4.1 SRAM2 奇偶校验 85
2.4.2 SRAM2 写保护 86
2.4.3 SRAM2 读保护 88
2.4.4 SRAM2 擦除 88
2.5 Flash 概述 88
2.6 自举配置 88
2.6.1 自举配置 88

3 嵌入式 Flash (FLASH) 91
3.1 简介 91
3.2 Flash 主要特性 91
3.3 Flash 功能说明 92

3.3.1 Flash 构成 . . . 92
3.3.2 误码校正 (ECC) . . . 94
3.3.3 读访问延迟 . . . 95
3.3.4 自适应实时存储器加速器 (ART Accelerator™) . . . 96
3.3.5 Flash 编程/擦除操作 . . . 98
3.3.6 Flash 主存储器擦除顺序 . . . 99
3.3.7 Flash 主存储器编程顺序 . . . 100
3.3.8 同步读写 (RWW) 仅适用于双存储区模式 (DBANK=1) . . . 103
3.4 FLASH 选项字节 . . . 104
3.4.1 选项字节说明 . . . 104
3.4.2 选项字节编程 . . . 112
3.5 Flash 保护 . . . 114
3.5.1 读保护 (RDP) . . . 114
3.5.2 专有代码读保护 (PCROP) . . . 117
3.5.3 写保护 (WRP) . . . 119
3.6 Flash 中断 . . . 120
3.7 FLASH 寄存器 . . . 121
3.7.1 Flash 访问控制寄存器 (FLASH_ACR) . . . 121
3.7.2 Flash 掉电密钥寄存器 (FLASH_PDKEYR) . . . 122
3.7.3 Flash 密钥寄存器 (FLASH_KEYR) . . . 123
3.7.4 Flash 选项密钥寄存器 (FLASH_OPTKEYR) . . . 123
3.7.5 Flash 状态寄存器 (FLASH_SR) . . . 124
3.7.6 Flash 控制寄存器 (FLASH_CR) . . . 126
3.7.7 Flash ECC 寄存器 (FLASH_ECCR) . . . 128
3.7.8 Flash 选项寄存器 (FLASH_OPTR) . . . 130
3.7.9 Flash PCROP1 起始地址寄存器 (FLASH_PCROP1SR) . . . 132
3.7.10 Flash PCROP1 结束地址寄存器 (FLASH_PCROP1ER) . . . 132
3.7.11 Flash WRP1 区域 A 地址寄存器 (FLASH_WRP1AR) . . . 133
3.7.12 Flash WRP1 区域 B 地址寄存器 (FLASH_WRP1BR) . . . 134
3.7.13 Flash PCROP2 起始地址寄存器 (FLASH_PCROP2SR) . . . 134
3.7.14 Flash PCROP2 结束地址寄存器 (FLASH_PCROP2ER) . . . 135
3.7.15 Flash WRP2 区域 A 地址寄存器 (FLASH_WRP2AR) . . . 135
3.7.16 Flash WRP2 区域 B 地址寄存器 (FLASH_WRP2BR) . . . 136
3.7.17 Flash 配置寄存器 (FLASH_CFGR) . . . 137
3.7.18 FLASH 寄存器映射 . . . 138

4 防火墙 (FW) 140
4.1 简介 140
4.2 防火墙主要特性 140
4.3 防火墙功能说明 141
4.3.1 防火墙 AMBA 总线监听 141
4.3.2 功能要求 141
4.3.3 防火墙段 142
4.3.4 段访问方式及其属性 143
4.3.5 防火墙初始化 144
4.3.6 防火墙状态 144
4.4 防火墙寄存器 146
4.4.1 代码段起始地址寄存器 (FW_CSSA) 146
4.4.2 代码段长度寄存器 (FW_CSL) 147
4.4.3 非易失性数据段起始地址寄存器 (FW_NVDSSA) 147
4.4.4 非易失性数据段长度寄存器 (FW_NVDSL) 148
4.4.5 易失性数据段起始地址寄存器 (FW_VDSSA) 148
4.4.6 易失性数据段长度寄存器 (FW_VDSL) 149
4.4.7 配置寄存器 (FW_CR) 150
4.4.8 防火墙寄存器映射 151

5 电源控制 (PWR) 152
5.1 电源 152
5.1.1 独立模拟外设电源 154
5.1.2 独立 I/O 电源轨 155
5.1.3 独立的 USB 收发器电源 155
5.1.4 独立 DSI 供电 155
5.1.5 电池备份域 156
5.1.6 稳压器 157
5.1.7 动态电压调节管理 157
5.2 电源监控器 159
5.2.1 上电复位上电复位 (POR)/掉电复位 (PDR)/欠压复位 (BOR) 159
5.2.2 可编程电压检测器 (PVD) 159
5.2.3 外设电压监测 (PVM) 160
5.3 低功耗模式 161
5.3.1 运行模式 168
5.3.2 低功耗运行模式（LP 运行） 168

5.3.3 低功耗模式 . . . 169
5.3.4 睡眠模式 . . . 170
5.3.5 低功耗睡眠模式（LP 睡眠） . . . 170
5.3.6 停止 0 模式 . . . 171
5.3.7 停止 1 模式 . . . 173
5.3.8 停止 2 模式 . . . 174
5.3.9 待机模式 . . . 177
5.3.10 关断模式 . . . 179
5.3.11 从低功耗模式自动唤醒 . . . 180
5.4 PWR 寄存器 . . . 181
5.4.1 电源控制寄存器 1 (PWR_CR1) . . . 181
5.4.2 电源控制寄存器 2 (PWR_CR2) . . . 182
5.4.3 电源控制寄存器 3 (PWR_CR3) . . . 183
5.4.4 电源控制寄存器 4 (PWR_CR4) . . . 184
5.4.5 电源状态寄存器 1 (PWR_SR1) . . . 185
5.4.6 电源状态寄存器 2 (PWR_SR2) . . . 186
5.4.7 电源状态清零寄存器 (PWR_SCR) . . . 188
5.4.8 电源端口 A 上拉控制寄存器 (PWR_PUCRA) . . . 189
5.4.9 电源端口 A 下拉控制寄存器 (PWR_PDCRA) . . . 190
5.4.10 电源端口 B 上拉控制寄存器 (PWR_PUCRB) . . . 190
5.4.11 电源端口 B 下拉控制寄存器 (PWR_PDCRB) . . . 191
5.4.12 电源端口 C 上拉控制寄存器 (PWR_PUCRC) . . . 191
5.4.13 电源端口 C 下拉控制寄存器 (PWR_PDCRC) . . . 192
5.4.14 电源端口 D 上拉控制寄存器 (PWR_PUCRD) . . . 192
5.4.15 电源端口 D 下拉控制寄存器 (PWR_PDCRD) . . . 193
5.4.16 电源端口 E 上拉控制寄存器 (PWR_PUCRE) . . . 193
5.4.17 电源端口 E 下拉控制寄存器 (PWR_PDCRE) . . . 194
5.4.18 电源端口 F 上拉控制寄存器 (PWR_PUCRF) . . . 194
5.4.19 电源端口 F 下拉控制寄存器 (PWR_PDCRF) . . . 195
5.4.20 电源端口 G 上拉控制寄存器 (PWR_PUCRG) . . . 195
5.4.21 电源端口 G 下拉控制寄存器 (PWR_PDCRG) . . . 196
5.4.22 电源端口 H 上拉控制寄存器 (PWR_PUCRH) . . . 196
5.4.23 电源端口 H 下拉控制寄存器 (PWR_PDCRH) . . . 197
5.4.24 电源端口 I 上拉控制寄存器 (PWR_PUCRI) . . . 197
5.4.25 电源端口 I 下拉控制寄存器 (PWR_PDCRI) . . . 198
5.4.26 PWR 控制寄存器 (PWR_CR5) . . . 198
5.4.27 PWR 寄存器映射和复位值表 . . . 199

6 复位和时钟控制 (RCC) 201
6.1 复位 201
6.1.1 电源复位 201
6.1.2 系统复位 201
6.1.3 备份域复位 202
6.2 时钟 203
6.2.1 HSE 时钟 206
6.2.2 HSI16 时钟 208
6.2.3 MSI 时钟 208
6.2.4 HSI48 时钟 209
6.2.5 PLL 209
6.2.6 LSE 时钟 210
6.2.7 LSI 时钟 210
6.2.8 系统时钟 (SYSCLK) 选择 211
6.2.9 时钟源频率与电压调节 211
6.2.10 时钟安全系统 (CSS) 212
6.2.11 LSE 上的时钟安全系统 212
6.2.12 ADC 时钟 212
6.2.13 RTC 时钟 213
6.2.14 定时器时钟 213
6.2.15 看门狗时钟 213
6.2.16 时钟输出功能 214
6.2.17 基于 TIM15/TIM16/TIM17 的内部/外部时钟测量 214
6.2.18 外设时钟使能寄存器（RCC_AHBxENR、RCC_APBxENRy） 216
6.3 低功耗模式 217
6.4 RCC 寄存器 218
6.4.1 时钟控制寄存器 (RCC_CR) 218
6.4.2 内部时钟源校准寄存器 (RCC_ICSCR) 221
6.4.3 时钟配置寄存器 (RCC_CFGR) 221
6.4.4 PLL 配置寄存器 (RCC_PLLCFGR) 224
6.4.5 PLLSAI1 配置寄存器 (RCC_PLLSAI1CFGR) 227
6.4.6 PLLSAI2 配置寄存器 (RCC_PLLSAI2CFGR) 230
6.4.7 时钟中断使能寄存器 (RCC_CIER) 232
6.4.8 时钟中断标志寄存器 (RCC_CIFR) 234
6.4.9 时钟中断清零寄存器 (RCC_CICR) 235
6.4.10 AHB1 外设复位寄存器 (RCC_AHB1RSTR) 237

6.4.11 AHB2 外设复位寄存器 (RCC_AHB2RSTR) 238
6.4.12 AHB3 外设复位寄存器 (RCC_AHB3RSTR) 240
6.4.13 APB1 外设复位寄存器 1 (RCC_APB1RSTR1) 241
6.4.14 APB1 外设复位寄存器 2 (RCC_APB1RSTR2) 243
6.4.15 APB2 外设复位寄存器 (RCC_APB2RSTR) 244
6.4.16 AHB1 外设时钟使能寄存器 (RCC_AHB1ENR) 245
6.4.17 AHB2 外设时钟使能寄存器 (RCC_AHB2ENR) 247
6.4.18 AHB3 外设时钟使能寄存器 (RCC_AHB3ENR) 249
6.4.19 APB1 外设时钟使能寄存器 1 (RCC_APB1ENR1) 250
6.4.20 APB1 外设时钟使能寄存器 2 (RCC_APB1ENR2) 252
6.4.21 APB2 外设时钟使能寄存器 (RCC_APB2ENR) 253
6.4.22 睡眠和停止模式下的 AHB1 外设时钟使能寄存器 (RCC_AHB1SMENR) 255
6.4.23 睡眠和停止模式下的 AHB2 外设时钟使能寄存器 (RCC_AHB2SMENR) 256
6.4.24 睡眠和停止模式下的 AHB3 外设时钟使能寄存器 (RCC_AHB3SMENR) 259
6.4.25 睡眠和停止模式下的 APB1 外设时钟使能寄存器 1 (RCC_APB1SMENR1) 260
6.4.26 睡眠和停止模式下的 APB1 外设时钟使能寄存器 2 (RCC_APB1SMENR2) 264
6.4.27 睡眠和停止模式下的 APB2 外设时钟使能寄存器 (RCC_APB2SMENR) 265
6.4.28 外设专用时钟配置寄存器 (RCC_CCIPR) 267
6.4.29 备份域控制寄存器 (RCC_BDCR) 269
6.4.30 控制/状态寄存器 (RCC_CSR) 271
6.4.31 时钟恢复 RC 寄存器 (RCC_CRRCR) 273
6.4.32 外设独立时钟配置寄存器 (RCC_CCIPR2) 274
6.4.33 RCC 寄存器映射 276

7 时钟恢复系统 (CRS) 281

7.1 简介 281
7.2 CRS 主要特性 281
7.3 CRS 功能说明 282
7.3.1 CRS 框图 282
7.3.2 同步输入 282
7.3.3 频率误差测量 283
7.3.4 频率误差评估和自动微调 284
7.3.5 CRS 初始化和配置 284

7.4 CRS 低功耗模式 285
7.5 CRS 中断 285
7.6 CRS 寄存器 286
7.6.1 CRS 控制寄存器 (CRS_CR) 286
7.6.2 CRS 配置寄存器 (CRS_CFGR) 287
7.6.3 CRS 中断和状态寄存器 (CRS_ISR) 288
7.6.4 CRS 中断标志清零寄存器 (CRS_ICR) 290
7.6.5 CRS 寄存器映射 291

8 通用 I/O (GPIO) 292
8.1 简介 292
8.2 GPIO 主要特性 292
8.3 GPIO 功能描述 292
8.3.1 通用 I/O (GPIO) 295
8.3.2 I/O 引脚复用功能复用器和映射 295
8.3.3 I/O 端口控制寄存器 296
8.3.4 I/O 端口数据寄存器 296
8.3.5 I/O 数据位操作 296
8.3.6 GPIO 锁定机制 296
8.3.7 I/O 复用功能输入/输出 297
8.3.8 外部中断线/唤醒线 297
8.3.9 输入配置 297
8.3.10 输出配置 298
8.3.11 复用功能配置 298
8.3.12 模拟配置 299
8.3.13 将 HSE 或 LSE 振荡器引脚用作 GPIO 300
8.3.14 在 RTC 电源域中使用 GPIO 引脚 300
8.3.15 将 PH3 用作 GPIO 300
8.4 GPIO寄存器 300
8.4.1 GPIO 端口模式寄存器 (GPIOx_MODER) (x = A..I) 301
8.4.2 GPIO 端口输出类型寄存器 (GPIOx_OTYPER) (x = A..I) 301
8.4.3 GPIO 端口输出速度寄存器 (GPIOx_OSPEEDR) (x = A..I) 302
8.4.4 GPIO 端口上拉/下拉寄存器 (GPIOx_PUPDR) (x = A..I) 302
8.4.5 GPIO 端口输入数据寄存器 (GPIOx_IDR) (x = A..I) 303
8.4.6 GPIO 端口输出数据寄存器 (GPIOx_ODR) (x = A..I) 303
8.4.7 GPIO 端口位置 1/复位寄存器 (GPIOx_BSRR) (x = A..I) 304

8.4.8 GPIO 端口配置锁定寄存器 (GPIOx_LCKR) (x = A..I) 304
8.4.9 GPIO 复用功能低位寄存器 (GPIOx_AFRL) (x = A..I) 305
8.4.10 GPIO 复用功能高位寄存器 (GPIOx_AFRH) (x = A..I) 306
8.4.11 GPIO 端口位复位寄存器 (GPIOx_BRR) (x = A..I) 306
8.4.12 GPIO 寄存器映射 307

9 系统配置控制器 (SYSCFG) 309
9.1 SYSCFG 主要特性 309
9.2 SYSCFG 寄存器 309
9.2.1 SYSCFG 存储器重映射寄存器 (SYSCFG_MEMRMP) 309
9.2.2 SYSCFG 配置寄存器 1 (SYSCFG_CFGR1) 310
9.2.3 SYSCFG 外部中断配置寄存器 1 (SYSCFG_EXTICR1) 312
9.2.4 SYSCFG 外部中断配置寄存器 2 (SYSCFG_EXTICR2) 314
9.2.5 SYSCFG 外部中断配置寄存器 3 (SYSCFG_EXTICR3) 315
9.2.6 SYSCFG 外部中断配置寄存器 4 (SYSCFG_EXTICR4) 317
9.2.7 SYSCFG SRAM2 控制和状态寄存器 (SYSCFG_SCSR) 318
9.2.8 SYSCFG 配置寄存器 2 (SYSCFG_CFGR2) 319
9.2.9 SYSCFG SRAM2 写保护寄存器 (SYSCFG_SWPR) 320
9.2.10 SYSCFG SRAM2 密钥寄存器 (SYSCFG_SKR) 320
9.2.11 SYSCFG SRAM2 写保护寄存器 2 (SYSCFG_SWPR2) 321
9.2.12 SYSCFG 寄存器映射 321

10 外设互连矩阵 323
10.1 简介 323
10.2 连接汇总 323
10.3 互连详细信息 324
10.3.1 从定时器 (TIM1/TIM2/TIM3/TIM4/TIM5/TIM8/TIM15/TIM16/TIM17) 到定时器 (TIM1/TIM2/TIM3/TIM4/TIM5/TIM8/TIM15) 324
10.3.2 从定时器 (TIM1/TIM2/TIM3/TIM4/TIM6/TIM8/TIM15) 和 EXTI 到 ADC (ADC1) 325
10.3.3 从 ADC 到定时器 (TIM1/TIM8) 325
10.3.4 从定时器 (TIM2/TIM4/TIM5/TIM6/TIM7/TIM8) 和 EXTI 到 DAC (DAC1/DAC2) 326
10.3.5 从定时器 (TIM1/TIM3/TIM4/TIM6/TIM7/TIM8/TIM16/LPTIM1/LPTIM2) 和 EXTI 到 DFSDM1 326
10.3.6 从 DFSDM1 到定时器 (TIM1/TIM8/TIM15/TIM16/TIM17) 327
10.3.7 从 HSE、LSE、LSI、MSI、MCO、RTC 到定时器 (TIM2/TIM15/TIM16/TIM17) 327

10.3.8 从 RTC、COMP1、COMP2 到低功耗定时器 (LPTIM1/LPTIM2) 328
10.3.9 从定时器 (TIM1/TIM2/TIM3/TIM8/TIM15) 到比较器 (COMP1/COMP2) . . 328
10.3.10 从 USB 到定时器 (TIM2) 328
10.3.11 从内部模拟源到 ADC 和 OPAMP (OPAMP1/OPAM2) 329
10.3.12 从比较器 (COMP1/COMP2) 到定时器 (TIM1/TIM2/TIM3/TIM8/TIM15/TIM16/TIM17) 329
10.3.13 从系统错误到定时器 (TIM1/TIM8/TIM15/TIM16/TIM17) 330
10.3.14 从定时器 (TIM16/TIM17) 到 IRTIM 330
10.3.15 从 ADC (ADC1) 到 DFSDM 330

11 直接存储器访问控制器 (DMA) 331

11.1 简介 331
11.2 DMA 主要特性 331
11.3 DMA 实现 332
11.3.1 DMA1 和 DMA2 332
11.3.2 DMA 请求映射 332
11.4 DMA 功能说明 332
11.4.1 DMA 框图 332
11.4.2 DMA 引脚和内部信号 334
11.4.3 DMA 传输 334
11.4.4 DMA 仲裁 335
11.4.5 DMA 通道 335
11.4.6 DMA 数据宽度、对齐和字节序 338
11.4.7 DMA 错误管理 339
11.5 DMA 中断 339
11.6 DMA 寄存器 340
11.6.1 DMA 中断状态寄存器 (DMA_ISR) 340
11.6.2 DMA 中断标志清零寄存器 (DMA_IFCR) 342
11.6.3 DMA 通道 x 配置寄存器 (DMA_CCRx) 343
11.6.4 DMA 通道 x 待传输数据数量寄存器 (DMA_CNDTRx) 346
11.6.5 DMA 通道 x 外设地址寄存器 (DMA_CPARx) 347
11.6.6 DMA 通道 x 存储器地址寄存器 (DMA_CMARx) 347
11.6.7 DMA 寄存器映射和复位值 348

12 DMA 请求复用器 (DMAMUX) 350
12.1 简介 350
12.2 DMAMUX 主要特性 350
12.3 DMAMUX 实现 351
12.3.1 DMAMUX 实例化 351
12.3.2 DMAMUX 映射 351
12.4 DMAMUX 功能说明 354
12.4.1 DMAMUX 框图 354
12.4.2 DMAMUX 信号 355
12.4.3 DMAMUX 通道 355
12.4.4 DMAMUX 请求线复用器 355
12.4.5 DMAMUX 请求发生器 358
12.5 DMAMUX 中断 359
12.6 DMAMUX 寄存器 359
12.6.1 DMAMUX 请求线复用器通道 x 配置寄存器 (DMAMUX_CxCR) 360
12.6.2 DMAMUX 请求线复用器中断通道状态寄存器 (DMAMUX_CSR) 361
12.6.3 DMAMUX 请求线复用器中断清除标志寄存器 (DMAMUX_CFR) 361
12.6.4 DMAMUX 请求发生器通道 x 配置寄存器 (DMAMUX_RGxCR) 362
12.6.5 DMAMUX 请求发生器中断状态寄存器 (DMAMUX_RGSR) 363
12.6.6 DMAMUX 请求发生器中断清除标志寄存器 (DMAMUX_RGCFR) 363
12.6.7 DMAMUX寄存器映射 364

13 Chrom-ART Accelerator™ 控制器 (DMA2D) 366
13.1 DMA2D 简介 366
13.2 DMA2D 的主要特性 366
13.3 DMA2D 功能说明 367
13.3.1 概述 367
13.3.2 DMA2D 控制 368
13.3.3 DMA2D 前景层 FIFO 和背景层 FIFO 368
13.3.4 DMA2D 前景层和背景层像素格式转换器 (PFC) 368
13.3.5 DMA2D 前景层 FIFO 和背景层 CLUT 接口 370
13.3.6 DMA2D 混合器 371
13.3.7 DMA2D 输出 PFC 372
13.3.8 DMA2D 输出 FIFO 372
13.3.9 DMA2D 输出 FIFO 字节重新排序 373
13.3.10 DMA2D AHB 主设备端口定时器 374

13.3.11 DMA2D 事务 375
13.3.12 DMA2D 配置 375
13.3.13 DMA2D 传输控制（启动、挂起、中止和完成） 379
13.3.14 水印 379
13.3.15 错误管理 379
13.3.16 AHB 死区 379
13.4 DMA2D 中断 380
13.5 DMA2D 寄存器 381
13.5.1 DMA2D 控制寄存器 (DMA2D_CR) 381
13.5.2 DMA2D 中断状态寄存器 (DMA2D_ISR) 383
13.5.3 DMA2D 中断标志清零寄存器 (DMA2D_IFCR) 384
13.5.4 DMA2D 前景层存储器地址寄存器 (DMA2D_FGMAR) 385
13.5.5 DMA2D 前景层偏移寄存器 (DMA2D_FGOR) 385
13.5.6 DMA2D 背景层存储器地址寄存器 (DMA2D_BGMAR) 386
13.5.7 DMA2D 背景层偏移寄存器 (DMA2D_BGOR) 386
13.5.8 DMA2D 前景层 PFC 控制寄存器 (DMA2D_FGPFCCR) 387
13.5.9 DMA2D 前景层颜色寄存器 (DMA2D_FGCOLR) 389
13.5.10 DMA2D 背景层 PFC 控制寄存器 (DMA2D_BGPFCCR) 390
13.5.11 DMA2D 背景层颜色寄存器 (DMA2D_BGCOLR) 392
13.5.12 DMA2D 前景层 CLUT 存储器地址寄存器 (DMA2D_FGCMAR) 392
13.5.13 DMA2D 背景层 CLUT 存储器地址寄存器 (DMA2D_BGCMAR) 393
13.5.14 DMA2D 输出 PFC 控制寄存器 (DMA2D_OPFCCR) 393
13.5.15 DMA2D 输出颜色寄存器 (DMA2D_OCOLR) 394
13.5.16 DMA2D 输出存储器地址寄存器 (DMA2D_OMAR) 395
13.5.17 DMA2D 输出偏移寄存器 (DMA2D_OOR) 396
13.5.18 DMA2D 行数寄存器 (DMA2D_NLR) 396
13.5.19 DMA2D 行水印寄存器 (DMA2D_LWR) 397
13.5.20 DMA2D AHB 主设备定时器配置寄存器 (DMA2D_AMTCR) 397
13.5.21 DMA2D 寄存器映射 398

14 Chrom-GRC™ (GFXMMU) 400
14.1 简介 400
14.2 Chrom-GRC™ 主要特性 400
14.3 Chrom-GRC™ 功能和架构说明 400
14.3.1 虚拟存储器 401
14.3.2 MMU 架构 403

14.4 图形 MMU 中断 406
14.5 图形 MMU 寄存器 407
14.5.1 图形 MMU 配置寄存器 (GFXMMU_CR) 407
14.5.2 图形 MMU 状态寄存器 (GFXMMU_SR) 408
14.5.3 图形 MMU 标志清零寄存器 (GFXMMU_FCR) 408
14.5.4 图形 MMU 默认值寄存器 (GFXMMU_DVR) 409
14.5.5 图形 MMU 缓冲区 0 配置寄存器 (GFXMMU_B0CR) 409
14.5.6 图形 MMU 缓冲区 1 配置寄存器 (GFXMMU_B1CR) 410
14.5.7 图形 MMU 缓冲区 2 配置寄存器 (GFXMMU_B2CR) 410
14.5.8 图形 MMU 缓冲区 3 配置寄存器 (GFXMMU_B3CR) 411
14.5.9 图形 MMU LUT 条目 x 低位 (GFXMMU_LUTxL) 411
14.5.10 图形 MMU LUT 条目 x 高位 (GFXMMU_LUTxH) 412
14.5.11 图形 MMU 寄存器映射 413

15 嵌套向量中断控制器 (NVIC) 414
15.1 NVIC 主要特性 414
15.2 SysTick 校准值寄存器 414
15.3 中断和异常向量 415

16 扩展中断和事件控制器 (EXTI) 419
16.1 简介 419
16.2 EXTI 主要特性 419
16.3 EXTI 功能说明 419
16.3.1 EXTI 框图 420
16.3.2 唤醒事件管理 420
16.3.3 外设异步中断 421
16.3.4 硬件中断选择 421
16.3.5 硬件事件选择 421
16.3.6 软件中断/事件选择 421
16.4 EXTI 中断/事件线映射 421
16.5 EXTI 寄存器 424
16.5.1 中断屏蔽寄存器 1 (EXTI_IMR1) 424
16.5.2 事件屏蔽寄存器 1 (EXTI_EMR1) 424
16.5.3 上升沿触发选择寄存器 1 (EXTI_RTSR1) 425
16.5.4 下降沿触发选择寄存器 1 (EXTI_FTSR1) 425
16.5.5 软件中断事件寄存器 1 (EXTI_SWIER1) 426

16.5.6 挂起寄存器 1 (EXTI_PR1) 427
16.5.7 中断屏蔽寄存器 2 (EXTI_IMR2) 427
16.5.8 事件屏蔽寄存器 2 (EXTI_EMR2) 428
16.5.9 上升沿触发选择寄存器 2 (EXTI_RTSR2) 428
16.5.10 下降沿触发选择寄存器 2 (EXTI_FTSR2) 429
16.5.11 软件中断事件寄存器 2 (EXTI_SWIER2) 429
16.5.12 挂起寄存器 2 (EXTI_PR2) 430
16.5.13 EXTI 寄存器映射 431

17 循环冗余校验计算单元 (CRC) 432
17.1 简介 432
17.2 CRC 主要特性 432
17.3 CRC 功能说明 432
17.3.1 CRC 框图 432
17.3.2 CRC 内部信号 433
17.3.3 CRC 操作 433
17.4 CRC 寄存器 434
17.4.1 数据寄存器 (CRC_DR) 434
17.4.2 独立数据寄存器 (CRC_IDR) 435
17.4.3 控制寄存器 (CRC_CR) 435
17.4.4 CRC 初始值 (CRC_INIT) 436
17.4.5 CRC 多项式 (CRC_POL) 436
17.4.6 CRC 寄存器映射 437

18 灵活的静态存储控制器 (FSMC) 438
18.1 FSMC 主要特性 438
18.2 FMC 框图 439
18.3 AHB 接口 439
18.3.1 支持的存储器和事务 440
18.4 外部器件地址映射 441
18.4.1 NOR/PSRAM 地址映射 442
18.4.2 NAND Flash 地址映射 442
18.5 NOR Flash/PSRAM 控制器 443
18.5.1 外部存储器接口信号 445
18.5.2 支持的存储器和事务 447
18.5.3 通用时序规则 448

18.5.4 NOR Flash/PSRAM 控制器异步事务 448
18.5.5 同步事务 466
18.5.6 NOR/PSRAM 控制器寄存器 473
18.6 NAND Flash 控制器 480
18.6.1 外部存储器接口信号 480
18.6.2 NAND Flash 支持的存储器和事务 481
18.6.3 NAND Flash 的时序图 482
18.6.4 NAND Flash 操作 483
18.6.5 NAND Flash 预等待功能 484
18.6.6 NAND Flash 中的纠错码 (ECC) 计算 485
18.6.7 NAND Flash 控制器寄存器 485
18.7 FSMC 寄存器映射 491

19 八线 SPI 接口 (OCTOSPI) 493
19.1 OCTOSPI 主要特性 493
19.2 OCTOSPI 功能说明 494
19.2.1 OCTOSPI 常规命令模式 495
19.2.2 OCTOSPI 常规命令模式信号接口 498
19.2.3 HyperBus™ 模式 501
19.2.4 常规命令模式与 HyperBus™ 的通用功能 505
19.3 OCTOSPI 工作模式 507
19.3.1 OCTOSPI 间接模式 507
19.3.2 OCTOSPI 状态标志轮询模式 509
19.3.3 OCTOSPI 内存映射模式 509
19.4 OCTOSPI 配置 510
19.4.1 OCTOSPI 系统配置 510
19.4.2 OCTOSPI 器件配置 510
19.4.3 OCTOSPI 常规命令模式配置 511
19.4.4 OCTOSPI HyperBus™ 模式配置 513
19.4.5 OCTOSPI 错误管理 514
19.4.6 OCTOSPI 的繁忙位和中止功能 515
19.4.7 OCTOSPI 重新配置或禁用 515
19.4.8 nCS 行为 515
19.5 OCTOSPI 中断 517

19.6 OCTOSPI 寄存器 518
19.6.1 OCTOSPI 控制寄存器 (OCTOSPI_CR) 518
19.6.2 OCTOSPI 器件配置寄存器 1 (OCTOSPI_DCR1) 520
19.6.3 OCTOSPI 器件配置寄存器 2 (OCTOSPI_DCR2) 522
19.6.4 OCTOSPI 器件配置寄存器 3 (OCTOSPI_DCR3) 523
19.6.5 OCTOSPI 器件配置寄存器 4 (OCTOSPI_DCR4) 523
19.6.6 OCTOSPI 状态寄存器 (OCTOSPI_SR) 524
19.6.7 OCTOSPI 标志清零寄存器 (OCTOSPI_FCR) 525
19.6.8 OCTOSPI 数据长度寄存器 (OCTOSPI_DLR) 526
19.6.9 OCTOSPI 地址寄存器 (OCTOSPI_AR) 527
19.6.10 OCTOSPI 数据寄存器 (OCTOSPI_DR) 527
19.6.11 OCTOSPI 轮询状态屏蔽寄存器 (OCTOSPI _PSMKR) 528
19.6.12 OCTOSPI 轮询状态匹配寄存器 (OCTOSPI _PSMAR) 528
19.6.13 OCTOSPI 轮询间隔寄存器 (OCTOSPI _PIR) 529
19.6.14 OCTOSPI 通信配置寄存器 (OCTOSPI_CCR) 529
19.6.15 OCTOSPI 时序配置寄存器 (OCTOSPI_TCR) 532
19.6.16 OCTOSPI 指令寄存器 (OCTOSPI_IR) 532
19.6.17 OCTOSPI 交替字节寄存器 (OCTOSPI_ABR) 533
19.6.18 OCTOSPI 低功耗超时寄存器 (OCTOSPI_LPTR) 533
19.6.19 OCTOSPI 回卷通信配置寄存器 (OCTOSPI_WPCCR) 534
19.6.20 OCTOSPI 回卷时序配置寄存器 (OCTOSPI_WPTCR) 536
19.6.21 OCTOSPI 回卷指令寄存器 (OCTOSPI_WPIR) 537
19.6.22 OCTOSPI 回卷交替字节寄存器 (OCTOSPI_WPABR) 537
19.6.23 OCTOSPI 写入通信配置寄存器 (OCTOSPI_WCCR) 538
19.6.24 OCTOSPI 写入时序配置寄存器 (OCTOSPI_WTCR) 540
19.6.25 OCTOSPI 写入指令寄存器 (OCTOSPI_WIR) 541
19.6.26 OCTOSPI 写入交替字节寄存器 (OCTOSPI_WABR) 541
19.6.27 OCTOSPI HyperBus™ 延迟配置寄存器 (OCTOSPI_HLCR) 542
19.6.28 OCTOSPI 寄存器映射 543

20 OctoSPI IO 管理器 (OCTOSPIM) 547

20.1 简介 547
20.2 OctoSPI IO 管理器的主要特性 547
20.3 OctoSPI IO 管理器功能说明 547
20.3.1 概述 547
20.3.2 OctoSPI IO 管理器矩阵 548

20.4 OctoSPI IO 管理器寄存器 549
20.4.1 OctoSPI IO 管理器端口 n 配置寄存器 (OCTOSPIM_PnCR) 549
20.4.2 OctoSPI IO 管理器寄存器映射 551

21 模数转换器 (ADC) 552
21.1 简介 552
21.2 ADC 主要特性 552
21.3 ADC 实现 553
21.4 ADC 功能说明 553
21.4.1 ADC 框图 553
21.4.2 ADC 引脚和内部信号 555
21.4.3 时钟 556
21.4.4 ADC1 连接功能 558
21.4.5 从设备 AHB 接口 559
21.4.6 ADC 深度掉电模式 (DEEPPWD) 和 ADC 稳压器 (ADVREGEN) 559
21.4.7 单端通道和差分输入通道 559
21.4.8 校准（ADCAL、ADCALDIF、ADCx_CALFACT） 560
21.4.9 ADC 开关控制（ADEN、ADDIS、ADRDY） 562
21.4.10 写入 ADC 控制位时的限制 563
21.4.11 通道选择（SQRx、JSQRx） 564
21.4.12 可独立设置各通道采样时间（SMPR1、SMPR2） 564
21.4.13 单次转换模式 (CONT=0) 565
21.4.14 连续转换模式 (CONT=1) 566
21.4.15 开始转换（ADSTART、JADSTART） 567
21.4.16 ADC 时序 568
21.4.17 停止正在进行的转换（ADSTP、JADSTP） 568
21.4.18 外部触发转换和触发极性（EXTSEL、EXTEN、JEXTSEL、JEXTEN） 570
21.4.19 注入通道管理 572
21.4.20 不连续模式（DISCEN、DISCNUM、JDISCEN） 573
21.4.21 注入转换的上下文队列 574
21.4.22 可编程分辨率 (RES)——快速转换模式 582
21.4.23 转换结束、采样阶段结束（EOC、JEOC、EOSMP） 583
21.4.24 转换序列结束（EOS、JEOS） 583
21.4.25 时序图示例（单次模式/连续模式，硬件/软件触发） 584
21.4.26 数据管理 586
21.4.27 使用 DFSDM 管理转换 591

21.4.28 动态低功耗特性 591
21.4.29 模拟窗口看门狗（AWD1EN、JAWD1EN、AWD1SGL、AWD1CH、AWD2CH、AWD3CH、AWD_HTx、AWD_LTx、AWDx） 596
21.4.30 过采样器 600
21.4.31 温度传感器 605
21.4.32 VBAT 电源监测 606
21.4.33 监测内部参考电压 607
21.5 ADC 中断 609
21.6 ADC 寄存器 610
21.6.1 ADC 中断和状态寄存器 (ADC_ISR) 610
21.6.2 ADC 中断使能寄存器 (ADC_IER) 612
21.6.3 ADC 控制寄存器 (ADC_CR) 614
21.6.4 ADC 配置寄存器 (ADC_CFGR) 617
21.6.5 ADC 配置寄存器 2 (ADC_CFGR2) 621
21.6.6 ADC 采样时间寄存器 1 (ADC_SMPR1) 623
21.6.7 ADC 采样时间寄存器 2 (ADC_SMPR2) 623
21.6.8 ADC 看门狗阈值寄存器 1 (ADC_TR1) 624
21.6.9 ADC 看门狗阈值寄存器 2 (ADC_TR2) 625
21.6.10 ADC 看门狗阈值寄存器 3 (ADC_TR3) 625
21.6.11 ADC 常规序列寄存器 1 (ADC_SQR1) 626
21.6.12 ADC 常规序列寄存器 2 (ADC_SQR2) 627
21.6.13 ADC 常规序列寄存器 3 (ADC_SQR3) 628
21.6.14 ADC 常规序列寄存器 4 (ADC_SQR4) 629
21.6.15 ADC 常规数据寄存器 (ADC_DR) 630
21.6.16 ADC 注入序列寄存器 (ADC_JSQR) 630
21.6.17 ADC 偏移 y 寄存器 (ADC_OFRy) 631
21.6.18 ADC 注入通道 y 数据寄存器 (ADC_JDRy) 632
21.6.19 ADC 模拟看门狗 2 配置寄存器 (ADC_AWD2CR) 633
21.6.20 ADC 模拟看门狗 3 配置寄存器 (ADC_AWD3CR) 633
21.6.21 ADC 差分模式选择寄存器 (ADC_DIFSEL) 634
21.6.22 ADC 校准系数 (ADC_CALFACT) 634
21.7 ADC 通用寄存器 635
21.7.1 ADC 通用状态寄存器 (ADC_CSR) 635
21.7.2 ADC 通用控制寄存器 (ADC_CCR) 636
21.7.3 ADC 寄存器映射 637

22 数模转换器 (DAC) 641
22.1 简介 641
22.2 DAC 主要特性 641
22.3 DAC 实现 642
22.4 DAC 功能说明 643
22.4.1 DAC框图 643
22.4.2 DAC 通道使能 644
22.4.3 DAC 数据格式 644
22.4.4 DAC 转换 646
22.4.5 DAC 输出电压 646
22.4.6 DAC 触发选择 647
22.4.7 DMA 请求 648
22.4.8 生成噪声 648
22.4.9 生成三角波 649
22.4.10 DAC 通道模式 650
22.4.11 DAC 通道缓冲器校准 653
22.4.12 DAC 双通道转换（如果可用） 654
22.5 DAC 低功耗模式 658
22.6 DAC 中断 658
22.7 DAC 寄存器 658
22.7.1 DAC 控制寄存器 (DAC_CR) 658
22.7.2 DAC 软件触发寄存器 (DAC_SWTRGR) 662
22.7.3 DAC 1 通道 12 位右对齐数据保持寄存器 (DAC_DHR12R1) 662
22.7.4 DAC 1 通道 12 位左对齐数据保持寄存器 (DAC_DHR12L1) 663
22.7.5 DAC 1 通道 8 位右对齐数据保持寄存器 (DAC_DHR8R1) 663
22.7.6 DAC 2 通道 12 位右对齐数据保持寄存器 (DAC_DHR12R2) 664
22.7.7 DAC 2 通道 12 位左对齐数据保持寄存器 (DAC_DHR12L2) 664
22.7.8 DAC 2 通道 8 位右对齐数据保持寄存器 (DAC_DHR8R2) 665
22.7.9 双 DAC 12 位右对齐数据保持寄存器 (DAC_DHR12RD) 665
22.7.10 双 DAC 12 位左对齐数据保持寄存器 (DAC_DHR12LD) 666
22.7.11 双 DAC 8 位右对齐数据保持寄存器 (DAC_DHR8RD) 666
22.7.12 DAC 1 通道数据输出寄存器 (DAC_DOR1) 667
22.7.13 DAC 2 通道数据输出寄存器 (DAC_DOR2) 667
22.7.14 DAC 状态寄存器 (DAC_SR) 668
22.7.15 DAC 校准控制寄存器 (DAC_CCR) 669
22.7.16 DAC 模式控制寄存器 (DAC_MCR) 669

22.7.17 DAC 通道 1 采样和保持采样时间寄存器 (DAC_SHSR1) 671
22.7.18 DAC 通道 2 采样和保持采样时间寄存器 (DAC_SHSR2) 671
22.7.19 DAC 采样和保持时间寄存器 (DAC_SHHR) 672
22.7.20 DAC 采样和保持刷新时间寄存器 (DAC_SHRR) 673
22.7.21 DAC 寄存器映射 674

23 数字摄像头接口 (DCMI) 676
23.1 DCMI 简介 676
23.2 DCMI 主要特性 676
23.3 DCMI 时钟 676
23.4 DCMI 功能概述 676
23.4.1 DCMI 框图 677
23.4.2 DMA 接口 677
23.4.3 DCMI 物理接口 678
23.4.4 同步 680
23.4.5 捕获模式 682
23.4.6 裁剪功能 683
23.4.7 JPEG 格式 684
23.4.8 FIFO 684
23.5 数据格式说明 684
23.5.1 数据格式 684
23.5.2 单色格式 685
23.5.3 RGB 格式 685
23.5.4 YCbCr 格式 686
23.5.5 YCbCr 格式——仅含 Y 分量 686
23.5.6 半分辨率图像提取 686
23.6 DCMI 中断 687
23.7 DCMI 寄存器说明 687
23.7.1 DCMI 控制寄存器 (DCMI_CR) 687
23.7.2 DCMI 状态寄存器 (DCMI_SR) 690
23.7.3 DCMI 原始中断状态寄存器 (DCMI_RIS) 691
23.7.4 DCMI 中断使能寄存器 (DCMI_IER) 692
23.7.5 DCMI 屏蔽中断状态寄存器 (DCMI_MIS) 693
23.7.6 DCMI 中断清零寄存器 (DCMI_ICR) 694
23.7.7 DCMI 内嵌同步码寄存器 (DCMI_ESCR) 695
23.7.8 DCMI 内嵌码同步取消屏蔽寄存器 (DCMI_ESUR) 696

23.7.9 DCMI 裁剪窗口起点 (DCMI_CWSTRT) 697
23.7.10 DCMI 裁剪窗口大小 (DCMI_CWSIZE) 697
23.7.11 DCMI 数据寄存器 (DCMI_DR) 698
23.7.12 DCMI 寄存器映射 699

24 电压参考缓冲器 (VREFBUF) 700
24.1 简介 700
24.2 VREFBUF 功能说明 700
24.3 VREFBUF 寄存器 701
24.3.1 VREFBUF 控制和状态寄存器 (VREFBUF_CSR) 701
24.3.2 VREFBUF 校准控制寄存器 (VREFBUF_CCR) 702
24.3.3 VREFBUF 寄存器映射 702

25 比较器 (COMP) 703
25.1 简介 703
25.2 COMP 主要特性 703
25.3 COMP 功能说明 704
25.3.1 COMP 框图 704
25.3.2 COMP 引脚和内部信号 704
25.3.3 COMP 复位和时钟 705
25.3.4 比较器锁定机制 706
25.3.5 窗口比较器 706
25.3.6 迟滞 706
25.3.7 比较器输出消隐功能 707
25.3.8 COMP 功耗和速度模式 708
25.4 COMP 低功耗模式 708
25.5 COMP 中断 708
25.6 COMP 寄存器 709
25.6.1 比较器 1 控制和状态寄存器 (COMP1_CSR) 709
25.6.2 比较器 2 控制和状态寄存器 (COMP2_CSR) 711
25.6.3 COMP 寄存器映射 713

26 运算放大器 (OPAMP) 714
26.1 简介 714
26.2 OPAMP 主要特性 714

26.3 OPAMP 功能说明 714
26.3.1 OPAMP 复位和时钟 714
26.3.2 初始配置 715
26.3.3 信号走线 715
26.3.4 OPAMP 模式 716
26.3.5 校准 719
26.4 OPAMP 低功耗模式 721
26.5 OPAMP 寄存器 722
26.5.1 OPAMP1 控制/状态寄存器 (OPAMP1_CSR) 722
26.5.2 正常模式下的 OPAMP1 偏移微调寄存器 (OPAMP1_OTR) 723
26.5.3 低功耗模式下的 OPAMP1 偏移微调寄存器 (OPAMP1_LPOTR) 724
26.5.4 OPAMP2 控制/状态寄存器 (OPAMP2_CRS) 724
26.5.5 正常模式下的 OPAMP2 偏移微调寄存器 (OPAMP2_OTR) 726
26.5.6 低功耗模式下的 OPAMP2 偏移微调寄存器 (OPAMP2_LPOTR) 726
26.5.7 OPAMP 寄存器映射 727

27 数字滤波器，用于ΣΔ调制器 (DFSDM) 728
27.1 简介 728
27.2 DFSDM主要特性 729
27.3 DFSDM 实现 730
27.4 DFSDM 功能说明 731
27.4.1 DFSDM框图 731
27.4.2 DFSDM 引脚和内部信号 732
27.4.3 DFSDM 复位和时钟 733
27.4.4 串行通道收发器 734
27.4.5 配置输入串行接口 743
27.4.6 并行数据输入 743
27.4.7 通道选择 745
27.4.8 数字滤波器配置 746
27.4.9 积分器单元 747
27.4.10 模拟看门狗 747
27.4.11 短路检测器 749
27.4.12 极值检测器 750
27.4.13 数据单元模块 750
27.4.14 有符号数据格式 751
27.4.15 启动转换 751

27.4.16 连续和快速连续模式 752
27.4.17 请求优先级 753
27.4.18 在运行模式下的功耗优化 753
27.5 DFSDM 中断 753
27.6 DFSDM DMA 传输 755
27.7 DFSDM 通道 y 寄存器 (y=0..7) 755
27.7.1 DFSDM 通道 y 配置寄存器 (DFSDM_CHyCFGR1) 755
27.7.2 DFSDM 通道 y 配置寄存器 (DFSDM_CHyCFGR2) 758
27.7.3 DFSDM 通道 y 模拟看门狗和短路检测器寄存器 (DFSDM_CHyAWSCDR) 759
27.7.4 DFSDM 通道 y 看门狗滤波器数据寄存器 (DFSDM_CHyWDATR) . . . 760
27.7.5 DFSDM 通道 y 数据输入寄存器 (DFSDM_CHyDATINR) 760
27.7.6 DFSDM 通道 y 延迟寄存器 (DFSDM_CHyDLYR) 761
27.8 DFSDM 滤波器 x 模块寄存器 (x=0..3) 762
27.8.1 DFSDM 滤波器 x 控制寄存器 1 (DFSDM_FLTxCR1) 762
27.8.2 DFSDM 滤波器 x 控制寄存器 2 (DFSDM_FLTxCR2) 765
27.8.3 DFSDM 滤波器 x 中断和状态寄存器 (DFSDM_FLTxISR) 766
27.8.4 DFSDM 滤波器 x 中断标志清零寄存器 (DFSDM_FLTxICR) 768
27.8.5 DFSDM 滤波器 x 注入通道组选择寄存器 (DFSDM_FLTxJCHGR) . . . 769
27.8.6 DFSDM 滤波器 x 控制寄存器 (DFSDM_FLTxFCR) 769
27.8.7 注入组的 DFSDM 滤波器 x 数据寄存器 (DFSDM_FLTxJDATAR) 770
27.8.8 常规通道的 DFSDM 滤波器 x 数据寄存器 (DFSDM_FLTxRDATAR) 771
27.8.9 DFSDM 滤波器 x 模拟看门狗阈值上限寄存器 (DFSDM_FLTxAWHTR) 772
27.8.10 DFSDM 滤波器 x 模拟看门狗阈值下限寄存器 (DFSDM_FLTxAWLTR) . . 772
27.8.11 DFSDM 滤波器 x 模拟看门狗状态寄存器 (DFSDM_FLTxAWSR) 773
27.8.12 DFSDM 滤波器 x 模拟看门狗清零标志寄存器 (DFSDM_FLTxAWCFR) 774
27.8.13 DFSDM 滤波器 x 极值检测器最大值寄存器 (DFSDM_FLTxEXMAX) . . . 774
27.8.14 DFSDM 滤波器 x 极值检测器最小值寄存器(DFSDM_FLTxEXMIN) . . . 775
27.8.15 DFSDM 滤波器 x 转换定时器寄存器 (DFSDM_FLTxCNVTIMR) 775
27.8.16 DFSDM寄存器映射 776

28 LCD-TFT 显示控制器 (LTDC) 786
28.1 简介 786
28.2 LTDC 主要特性 786

28.3 LTDC 功能说明 787
28.3.1 LTDC 框图 787
28.3.2 LTDC 引脚和外部信号接口 787
28.3.3 LTDC 复位和时钟 788
28.4 LTDC 可编程参数 789
28.4.1 LTDC 全局配置参数 789
28.4.2 层可编程参数 792
28.5 LTDC 中断 796
28.6 LTDC 编程步骤 797
28.7 LTDC 寄存器 798
28.7.1 LTDC 同步大小配置寄存器 (LTDC_SSCR) 798
28.7.2 LTDC 后沿配置寄存器 (LTDC_BPCR) 798
28.7.3 LTDC 有效宽度配置寄存器 (LTDC_AWCR) 799
28.7.4 LTDC 总宽度配置寄存器 (LTDC_TWCR) 800
28.7.5 LTDC 全局控制寄存器 (LTDC_GCR) 800
28.7.6 LTDC 影子重载配置寄存器 (LTDC_SRCR) 802
28.7.7 LTDC 背景色配置寄存器 (LTDC_BCCR) 802
28.7.8 LTDC 中断使能寄存器 (LTDC_IER) 803
28.7.9 LTDC 中断状态寄存器 (LTDC_ISR) 804
28.7.10 LTDC 中断清零寄存器 (LTDC_ICR) 804
28.7.11 LTDC 行中断位置配置寄存器 (LTDC_LIPCR) 805
28.7.12 LTDC 当前位置状态寄存器 (LTDC_CPSR) 806
28.7.13 LTDC 当前显示状态寄存器 (LTDC_CDSR) 806
28.7.14 LTDC 第 x 层控制寄存器 (LTDC_LxCR) 807
28.7.15 LTDC 第 x 层窗口水平位置配置寄存器 (LTDC_LxWHPCR) 808
28.7.16 LTDC 第 x 层窗口垂直位置配置寄存器 (LTDC_LxWVPCR) 809
28.7.17 LTDC 第 x 层色键配置寄存器 (LTDC_LxCKCR) 810
28.7.18 LTDC 第 x 层像素格式配置寄存器 (LTDC_LxPFCR) 810
28.7.19 LTDC 第 x 层常数 alpha 配置寄存器 (LTDC_LxCACR) 811
28.7.20 LTDC 第 x 层默认颜色配置寄存器 (LTDC_LxDCCR) 812
28.7.21 LTDC 第 x 层混合系数配置寄存器 (LTDC_LxBFCR) 813
28.7.22 LTDC 第 x 层颜色帧缓冲区地址寄存器 (LTDC_LxCFBAR) 814
28.7.23 LTDC 第 x 层颜色帧缓冲区长度寄存器 (LTDC_LxCFBLR) 814
28.7.24 LTDC 第 x 层颜色帧缓冲区行数寄存器 (LTDC_LxCFBLNR) 815
28.7.25 LTDC 第 x 层 CLUT 写寄存器 (LTDC_LxCLUTWR) 816
28.7.26 LTDC 寄存器映射 817

29 DSI 主机 (DSIHOST) 820
29.1 简介 820
29.2 标准和参考文献 820
29.3 DSI 主机主要特性 821
29.4 DSI 主机功能说明 822
29.4.1 概述 822
29.4.2 支持的分辨率和帧速率 822
29.4.3 系统级架构 823
29.5 功能说明：LTDC 接口上的视频模式 824
29.5.1 视频发送模式 826
29.5.2 在视频模式下更新 LTDC 接口配置 827
29.6 功能说明：LTDC 接口上的适配命令模式 828
29.7 功能说明：APB 从通用接口 831
29.7.1 使用通用接口进行数据包发送 832
29.8 功能说明：超时计数器 835
29.8.1 争用错误检测超时计数器 835
29.8.2 外设响应超时计数器 836
29.9 功能说明：命令发送 840
29.9.1 视频模式下的命令发送 840
29.9.2 低功耗模式下的命令发送 842
29.9.3 高速模式下的命令发送 846
29.9.4 读取命令发送 846
29.9.5 低功耗模式下的时钟通道 847
29.10 功能说明：虚拟通道 849
29.11 功能说明：视频模式测试图发生器 850
29.11.1 彩条测试卡 850
29.11.2 颜色编码 852
29.11.3 BER 测试模式 852
29.11.4 视频模式测试图发生器分辨率 853
29.12 功能说明：D-PHY 管理 854
29.12.1 D-PHY 配置 854
29.12.2 特殊 D-PHY 操作 856
29.12.3 特殊低功耗 D-PHY 功能 856
29.12.4 DSI PLL 控制 857
29.12.5 稳压器控制 858

29.13 功能说明：中断和错误 858
29.13.1 DSI 封装器中断 858
29.13.2 DSI 主机中断和错误 859
29.14 编程步骤 864
29.14.1 编程步骤概述 864
29.14.2 配置 D-PHY 参数 865
29.14.3 配置 DSI 主机时序 865
29.14.4 配置流控制和 DBI 接口 866
29.14.5 配置 DSI 主机 LTDC 接口 866
29.14.6 配置视频模式 867
29.14.7 配置适配命令模式 869
29.14.8 配置视频模式测试图发生器 870
29.14.9 管理 ULPM 871
29.15 DSI 主机寄存器 873
29.15.1 DSI 主机版本寄存器 (DSI_VR) 873
29.15.2 DSI 主机控制寄存器 (DSI_CR) 873
29.15.3 DSI 主机时钟控制寄存器 (DSI_CCR) 874
29.15.4 DSI 主机 LTDC VCID 寄存器 (DSI_LVCIDR) 874
29.15.5 DSI 主机 LTDC 颜色编码寄存器 (DSI_LCOLCR) 875
29.15.6 DSI 主机 LTDC 极性配置寄存器 (DSI_LPCR) 875
29.15.7 DSI 主机低功耗模式配置寄存器 (DSI_LPMCR) 876
29.15.8 DSI 主机协议配置寄存器 (DSI_PCR) 876
29.15.9 DSI 主机通用 VCID 寄存器 (DSI_GVCIDR) 878
29.15.10 DSI 主机模式配置寄存器 (DSI_MCR) 878
29.15.11 DSI 主机视频模式配置寄存器 (DSI_VMCR) 879
29.15.12 DSI 主机视频数据包配置寄存器 (DSI_VPCR) 880
29.15.13 DSI 主机视频块配置寄存器 (DSI_VCCR) 881
29.15.14 DSI 主机视频空包配置寄存器 (DSI_VNPCR) 881
29.15.15 DSI 主机视频 HSA 配置寄存器 (DSI_VHSACR) 882
29.15.16 DSI 主机视频 HBP 配置寄存器 (DSI_VHBPCR) 882
29.15.17 DSI 主机视频行配置寄存器 (DSI_VLCR) 883
29.15.18 DSI 主机视频 VSA 配置寄存器 (DSI_VVSACR) 883
29.15.19 DSI 主机视频 VBP 配置寄存器 (DSI_VVBPCR) 884
29.15.20 DSI 主机视频 VFP 配置寄存器 (DSI_VVFPCR) 884
29.15.21 DSI 主机视频 VA 配置寄存器 (DSI_VVACR) 885
29.15.22 DSI 主机 LTDC 命令配置寄存器 (DSI_LCCR) 885
29.15.23 DSI 主机命令模式配置寄存器 (DSI_CMCR) 886

29.15.24 DSI 主机通用头配置寄存器 (DSI_GHCR) 888
29.15.25 DSI 主机通用有效负载数据寄存器 (DSI_GPDR) 888
29.15.26 DSI 主机通用数据包状态寄存器 (DSI_GPSR) 889
29.15.27 DSI 主机超时计数器配置寄存器 0 (DSI_TCCR0) 890
29.15.28 DSI 主机超时计数器配置寄存器 1 (DSI_TCCR1) 891
29.15.29 DSI 主机超时计数器配置寄存器 2 (DSI_TCCR2) 891
29.15.30 DSI 主机超时计数器配置寄存器 3 (DSI_TCCR3) 892
29.15.31 DSI 主机超时计数器配置寄存器 4 (DSI_TCCR4) 893
29.15.32 DSI 主机超时计数器配置寄存器 5 (DSI_TCCR5) 893
29.15.33 DSI 主机时钟通道配置寄存器 (DSI_CLCR) 894
29.15.34 DSI 主机时钟通道计时器配置寄存器 (DSI_CLTCR) 894
29.15.35 DSI 主机数据通道计时器配置寄存器 (DSI_DLTCR) 895
29.15.36 DSI 主机 PHY 控制寄存器 (DSI_PCTLR) 895
29.15.37 DSI 主机 PHY 配置寄存器 (DSI_PCONFR) 896
29.15.38 DSI 主机 PHY ULPS 控制寄存器 (DSI_PUCR) 896
29.15.39 DSI 主机 PHY TX 触发配置寄存器 (DSI_PTTCR) 897
29.15.40 DSI 主机 PHY 状态寄存器 (DSI_PSR) 898
29.15.41 DSI 主机中断和状态寄存器 0 (DSI_ISR0) 899
29.15.42 DSI 主机中断和状态寄存器 1 (DSI_ISR1) 900
29.15.43 DSI 主机中断使能寄存器 0 (DSI_IER0) 901
29.15.44 DSI 主机中断使能寄存器 1 (DSI_IER1) 904
29.15.45 DSI 主机强制中断寄存器 0 (DSI_FIR0) 905
29.15.46 DSI 主机强制中断寄存器 1 (DSI_FIR1) 907
29.15.47 DSI 主机视频影子控制寄存器 (DSI_VSCR) 908
29.15.48 DSI 主机 LTDC 当前 VCID 寄存器 (DSI_LCVCIDR) 908
29.15.49 DSI 主机 LTDC 当前颜色编码寄存器 (DSI_LCCCR) 909
29.15.50 DSI 主机低功耗模式当前配置寄存器 (DSI_LPMCCR) 910
29.15.51 DSI 主机视频模式当前配置寄存器 (DSI_VMCCR) 910
29.15.52 DSI 主机视频数据包当前配置寄存器 (DSI_VPCCR) 912
29.15.53 DSI 主机视频块当前配置寄存器 (DSI_VCCCR) 912
29.15.54 DSI 主机视频空包当前配置寄存器 (DSI_VNPCCR) 913
29.15.55 DSI 主机视频 HSA 当前配置寄存器 (DSI_VHSACCR) 913
29.15.56 DSI 主机视频 HBP 当前配置寄存器 (DSI_VHBPCCR) 914
29.15.57 DSI 主机视频行当前配置寄存器 (DSI_VLCCR) 914
29.15.58 DSI 主机视频 VSA 当前配置寄存器 (DSI_VVSACCR) 915
29.15.59 DSI 主机视频 VBP 当前配置寄存器 (DSI_VVBPCCR) 915
29.15.60 DSI 主机视频 VFP 当前配置寄存器 (DSI_VVFPCCR) 916

29.15.61 DSI 主机视频 VA 当前配置寄存器 (DSI_VVACCR) 916
29.16 DSI 封装器寄存器 917
29.16.1 DSI 封装器配置寄存器 (DSI_WCFGR) 917
29.16.2 DSI 封装器控制寄存器 (DSI_WCR) 918
29.16.3 DSI 封装器中断使能寄存器 (DSI_WIER) 919
29.16.4 DSI 封装器中断和状态寄存器 (DSI_WISR) 920
29.16.5 DSI 封装器中断标志清零寄存器 (DSI_WIFCR) 921
29.16.6 DSI 封装器 PHY 配置寄存器 0 (DSI_WPCR0) 922
29.16.7 DSI 封装器 PHY 配置寄存器 1 (DSI_WPCR1) 924
29.16.8 DSI 封装器 PHY 配置寄存器 2 (DSI_WPCR2) 926
29.16.9 DSI 封装器 PHY 配置寄存器 3 (DSI_WPCR4) 927
29.16.10 DSI 封装器 PHY 配置寄存器 4 (DSI_WPCR4) 928
29.16.11 DSI 封装器稳压器和 PLL 控制寄存器 (DSI_WRPCR) 928
29.17 DSI 主机寄存器映射 930

30 触摸感应控制器 (TSC) 935
30.1 简介 935
30.2 TSC 主要特性 935
30.3 TSC 功能说明 936
30.3.1 TSC 框图 936
30.3.2 表面电荷转移采集概述 936
30.3.3 复位和时钟 938
30.3.4 电荷转移采集序列 939
30.3.5 扩频功能 940
30.3.6 最大计数错误 940
30.3.7 采样电容 I/O 和通道 I/O 模式选择 941
30.3.8 采集模式 942
30.3.9 I/O 滞后和模拟开关控制 942
30.4 TSC 低功耗模式 943
30.5 TSC 中断 943
30.6 TSC 寄存器 944
30.6.1 TSC 控制寄存器 (TSC_CR) 944
30.6.2 TSC 中断使能寄存器 (TSC_IER) 946
30.6.3 TSC 中断清零寄存器 (TSC_ICR) 947
30.6.4 TSC 中断状态寄存器 (TSC_ISR) 948
30.6.5 TSC I/O 滞后控制寄存器 (TSC_IOHCR) 948

30.6.6 TSC I/O 模拟开关控制寄存器 (TSC_IOASCR) 949
30.6.7 TSC I/O 采样控制寄存器 (TSC_IOSCR) 949
30.6.8 TSC I/O 通道控制寄存器 (TSC_IOCCR) 950
30.6.9 TSC I/O 组控制状态寄存器 (TSC_IOGCSR) 950
30.6.10 TSC I/O 组 x 计数器寄存器 (TSC_IOGxCR) 951
30.6.11 TSC 寄存器映射 952

31 真随机数发生器 (RNG) 954
31.1 简介 954
31.2 RNG 主要特性 954
31.3 RNG 功能说明 955
31.3.1 RNG 框图 955
31.3.2 RNG 内部信号 955
31.3.3 随机数生成 956
31.3.4 RNG 初始化 958
31.3.5 RNG 操作 958
31.3.6 RNG 时钟 959
31.3.7 错误管理 959
31.4 RNG 低功耗使用 960
31.5 RNG 中断 960
31.6 RNG 处理时间 961
31.7 熵源验证 961
31.7.1 简介 961
31.7.2 验证条件 961
31.7.3 数据采集 961
31.8 RNG 寄存器 962
31.8.1 RNG 控制寄存器 (RNG_CR) 962
31.8.2 RNG 状态寄存器 (RNG_SR) 963
31.8.3 RNG 数据寄存器 (RNG_DR) 964
31.8.4 RNG 寄存器映射 964

32 AES 硬件加速器 (AES) 965
32.1 简介 965
32.2 AES 主要特性 965
32.3 AES 实现 966

32.4 AES 功能描述 966
32.4.1 AES 框图 966
32.4.2 AES 内部信号 966
32.4.3 AES 加密内核 967
32.4.4 执行密码操作的 AES 过程 973
32.4.5 AES 解密密钥准备 976
32.4.6 AES 密文窃取和数据填充 977
32.4.7 AES 任务挂起和恢复 977
32.4.8 AES 基本链接模式（ECB 和 CBC） 978
32.4.9 AES 计数器 (CTR) 模式 983
32.4.10 AES Galois/计数器模式 (GCM) 986
32.4.11 AESGalois 消息认证码 (GMAC) 990
32.4.12 AES CBC-MAC 计数器模式 (CCM) 991
32.4.13 AES 数据寄存器和数据交换 997
32.4.14 AES 密钥寄存器 999
32.4.15 AES 初始化向量寄存器 999
32.4.16 AES DMA 接口 999
32.4.17 AES 错误管理 1002
32.5 AES 中断 1002
32.6 AES 处理延迟 1003
32.7 AES 寄存器 1004
32.7.1 AES 控制寄存器 (AES_CR) 1004
32.7.2 AES 状态寄存器 (AES_SR) 1007
32.7.3 AES 数据输入寄存器 (AES_DINR) 1008
32.7.4 AES 数据输出寄存器 (AES_DOUTR) 1008
32.7.5 AES 密钥寄存器 0 (AES_KEYR0) 1009
32.7.6 AES 密钥寄存器 1 (AES_KEYR1) 1010
32.7.7 AES 密钥寄存器 2 (AES_KEYR2) 1010
32.7.8 AES 密钥寄存器 3 (AES_KEYR3) 1011
32.7.9 AES 初始化向量寄存器 0 (AES_IVR0) 1011
32.7.10 AES 初始化向量寄存器 1 (AES_IVR1) 1012
32.7.11 AES 初始化向量寄存器 2 (AES_IVR2) 1012
32.7.12 AES 初始化向量寄存器 3 (AES_IVR3) 1013
32.7.13 AES 密钥寄存器 4 (AES_KEYR4) 1013
32.7.14 AES 密钥寄存器 5 (AES_KEYR5) 1014
32.7.15 AES 密钥寄存器 6 (AES_KEYR6) 1014
32.7.16 AES 密钥寄存器 7 (AES_KEYR7) 1015

32.7.17 AES 挂起寄存器 (AES_SUSPxR) 1015
32.7.18 AES 寄存器映射 1016

33 散列处理器 (HASH) 1018
33.1 简介 1018
33.2 散列主要特性 1018
33.3 散列功能说明 1019
33.3.1 HASH 框图 1019
33.3.2 HASH 内部信号 1019
33.3.3 关于安全散列算法 1020
33.3.4 消息数据馈送 1020
33.3.5 消息摘要计算 1022
33.3.6 消息填充 1023
33.3.7 HMAC 运算 1025
33.3.8 上下文交换 1026
33.3.9 HASH DMA 接口 1028
33.3.10 HASH 错误管理 1028
33.4 HASH 中断 1028
33.5 HASH 处理时间 1029
33.6 HASH 寄存器 1030
33.6.1 HASH 控制寄存器 (HASH_CR) 1030
33.6.2 HASH 数据输入寄存器 (HASH_DIN) 1032
33.6.3 HASH 启动寄存器 (HASH_STR) 1033
33.6.4 HASH 摘要寄存器 (HASH_HR0..7) 1034
33.6.5 HASH 中断使能寄存器 (HASH_IMR) 1036
33.6.6 HASH 状态寄存器 (HASH_SR) 1037
33.6.7 HASH 上下文交换寄存器 (HASH_CSRx) 1038
33.6.8 HASH 寄存器映射 1039

34 高级控制定时器 (TIM1/TIM8) 1040
34.1 TIM1/TIM8 简介 1040
34.2 TIM1/TIM8 主要特性 1040
34.3 TIM1/TIM8 功能描述 1042
34.3.1 时基单元 1042
34.3.2 计数器模式 1044
34.3.3 重复计数器 1055

34.3.4 外部触发输入 1057
34.3.5 时钟选择 1058
34.3.6 捕获/比较通道 1062
34.3.7 输入捕获模式 1065
34.3.8 PWM 输入模式 1066
34.3.9 强制输出模式 1067
34.3.10 输出比较模式 1067
34.3.11 PWM 模式 1068
34.3.12 不对称 PWM 模式 1071
34.3.13 组合 PWM 模式 1072
34.3.14 组合三相 PWM 模式 1073
34.3.15 互补输出和死区插入 1074
34.3.16 使用断路功能 1076
34.3.17 双向断路输入 1081
34.3.18 发生外部事件时清除 OCxREF 信号 1083
34.3.19 生成 6 步 PWM 1084
34.3.20 单脉冲模式 1085
34.3.21 可再触发单脉冲模式 (OPM) 1086
34.3.22 编码器接口模式 1087
34.3.23 UIF 位重映射 1089
34.3.24 定时器输入异或功能 1090
34.3.25 连接霍尔传感器 1090
34.3.26 定时器同步 1092
34.3.27 ADC 同步 1095
34.3.28 DMA 连续传送模式 1096
34.3.29 调试模式 1097
34.4 TIM1/TIM8 寄存器 1097
34.4.1 TIM1/TIM8 控制寄存器 1 (TIMx_CR1) 1097
34.4.2 TIM1/TIM8 控制寄存器 2 (TIMx_CR2) 1099
34.4.3 TIM1/TIM8 从模式控制寄存器 (TIMx_SMCR) 1101
34.4.4 TIM1/TIM8 DMA/中断使能寄存器 (TIMx_DIER) 1103
34.4.5 TIM1/TIM8 状态寄存器 (TIMx_SR) 1105
34.4.6 TIM1/TIM8 事件生成寄存器 (TIMx_EGR) 1107
34.4.7 TIM1/TIM8 捕获/比较模式寄存器 1 (TIMx_CCMR1) 1108
34.4.8 TIM1/TIM8 捕获/比较模式寄存器 2 (TIMx_CCMR2) 1111
34.4.9 TIM1/TIM8 捕获/比较使能寄存器 (TIMx_CCER) 1113
34.4.10 TIM1/TIM8 计数器 (TIMx_CNT) 1117

34.4.11 TIM1/TIM8 预分频器 (TIMx_PSC) 1117
34.4.12 TIM1/TIM8 自动重载寄存器 (TIMx_ARR) 1118
34.4.13 TIM1/TIM8 重复计数器寄存器 (TIMx_RCR) 1118
34.4.14 TIM1/TIM8 捕获/比较寄存器 1 (TIMx_CCR1) 1119
34.4.15 TIM1/TIM8 捕获/比较寄存器 2 (TIMx_CCR2) 1119
34.4.16 TIM1/TIM8 捕获/比较寄存器 3 (TIMx_CCR3) 1120
34.4.17 TIM1/TIM8 捕获/比较寄存器 4 (TIMx_CCR4) 1120
34.4.18 TIM1/TIM8 断路和死区寄存器 (TIMx_BDTR) 1121
34.4.19 TIM1/TIM8 DMA 控制寄存器 (TIMx_DCR) 1124
34.4.20 TIM1/TIM8 全传输 DMA 地址 (TIMx_DMAR) 1125
34.4.21 TIM1 选项寄存器 1 (TIM1_OR1) 1126
34.4.22 TIM8 选项寄存器 1 (TIM8_OR1) 1126
34.4.23 TIM1/TIM8 捕获/比较模式寄存器 3 (TIMx_CCMR3) 1127
34.4.24 TIM1/TIM8 捕获/比较寄存器 5 (TIMx_CCR5) 1128
34.4.25 TIM1/TIM8 捕获/比较寄存器 6 (TIMx_CCR6) 1129
34.4.26 TIM1 选项寄存器 2 (TIM1_OR2) 1129
34.4.27 TIM1 选项寄存器 3 (TIM1_OR3) 1131
34.4.28 TIM8 选项寄存器 2 (TIM8_OR2) 1132
34.4.29 TIM8 选项寄存器 3 (TIM8_OR3) 1133
34.4.30 TIM1 寄存器映射 1135
34.4.31 TIM8 寄存器映射 1138

35 通用定时器 (TIM2/TIM3/TIM4/TIM5) 1141

35.1 TIM2/TIM3/TIM4/TIM5 简介 1141
35.2 TIM2/TIM3/TIM4/TIM5 主要特性 1141
35.3 TIM2/TIM3/TIM4/TIM5 功能描述 1143
35.3.1 时基单元 1143
35.3.2 计数器模式 1145
35.3.3 时钟选择 1155
35.3.4 捕获/比较通道 1159
35.3.5 输入捕获模式 1161
35.3.6 PWM 输入模式 1162
35.3.7 强制输出模式 1163
35.3.8 输出比较模式 1163
35.3.9 PWM 模式 1165
35.3.10 不对称 PWM 模式 1168
35.3.11 组合 PWM 模式 1168

35.3.12 发生外部事件时清除 OCxREF 信号 1169
35.3.13 单脉冲模式 1171
35.3.14 可再触发单脉冲模式 (OPM) 1172
35.3.15 编码器接口模式 1173
35.3.16 UIF 位重映射 1175
35.3.17 定时器输入异或功能 1175
35.3.18 定时器与外部触发同步 1175
35.3.19 定时器同步 1179
35.3.20 DMA 连续传送模式 1183
35.3.21 调试模式 1184
35.4 TIM2/TIM3/TIM4/TIM5 寄存器 1185
35.4.1 TIMx 控制寄存器 1 (TIMx_CR1) 1185
35.4.2 TIMx 控制寄存器 2 (TIMx_CR2) 1187
35.4.3 TIMx 从模式控制寄存器 (TIMx_SMCR) 1188
35.4.4 TIMx DMA/中断使能寄存器 (TIMx_DIER) 1190
35.4.5 TIMx 状态寄存器 (TIMx_SR) 1191
35.4.6 TIMx 事件生成寄存器 (TIMx_EGR) 1193
35.4.7 TIMx 捕获/比较模式寄存器 1 (TIMx_CCMR1) 1194
35.4.8 TIMx 捕获/比较模式寄存器 2 (TIMx_CCMR2) 1197
35.4.9 TIMx 捕获/比较使能寄存器 (TIMx_CCER) 1199
35.4.10 TIMx 计数器 (TIMx_CNT) 1201
35.4.11 TIMx 预分频器 (TIMx_PSC) 1201
35.4.12 TIMx 自动重载寄存器 (TIMx_ARR) 1202
35.4.13 TIMx 捕获/比较寄存器 1 (TIMx_CCR1) 1202
35.4.14 TIMx 捕获/比较寄存器 2 (TIMx_CCR2) 1203
35.4.15 TIMx 捕获/比较寄存器 3 (TIMx_CCR3) 1203
35.4.16 TIMx 捕获/比较寄存器 4 (TIMx_CCR4) 1204
35.4.17 TIMx DMA 控制寄存器 (TIMx_DCR) 1205
35.4.18 TIMx 全传输 DMA 地址 (TIMx_DMAR) 1205
35.4.19 TIM2 选项寄存器 1 (TIM2_OR1) 1206
35.4.20 TIM3 选项寄存器 1 (TIM3_OR1) 1206
35.4.21 TIM2 选项寄存器 2 (TIM2_OR2) 1207
35.4.22 TIM3 选项寄存器 2 (TIM3_OR2) 1207
35.4.23 TIMx 寄存器映射 1208

36 通用定时器 (TIM15/TIM16/TIM17) 1211

36.1 TIM15/TIM16/TIM17 简介 1211

36.2 TIM15 主要特性 1211

36.3 TIM16/TIM17 主要特性 1212

36.4 TIM15/TIM16/TIM17 功能描述 1215

36.4.1 时基单元 1215

36.4.2 计数器模式 1217

36.4.3 重复计数器 1221

36.4.4 时钟选择 1222

36.4.5 捕获/比较通道 1224

36.4.6 输入捕获模式 1227

36.4.7 PWM 输入模式（仅适用于 TIM15） 1228

36.4.8 强制输出模式 1229

36.4.9 输出比较模式 1229

36.4.10 PWM 模式 1231

36.4.11 组合 PWM 模式（仅适用于 TIM15） 1232

36.4.12 互补输出和死区插入 1233

36.4.13 使用断路功能 1235

36.4.14 双向断路输入 1239

36.4.15 单脉冲模式 1241

36.4.16 可再触发单脉冲模式 (OPM)（仅限 TIM15） 1242

36.4.17 UIF 位重映射 1243

36.4.18 定时器输入异或功能（仅适用于 TIM15） 1243

36.4.19 外部触发同步（仅适用于 TIM15） 1244

36.4.20 从模式——组合复位 + 触发模式 1246

36.4.21 DMA 连续传送模式 1246

36.4.22 定时器同步 (TIM15) 1247

36.4.23 调试模式 1247

36.5 TIM15 寄存器 1248

36.5.1 TIM15 控制寄存器 1 (TIM15_CR1) 1248

36.5.2 TIM15 控制寄存器 2 (TIM15_CR2) 1249

36.5.3 TIM15 从模式控制寄存器 (TIM15_SMCR) 1251

36.5.4 TIM15 DMA/中断使能寄存器 (TIM15_DIER) 1252

36.5.5 TIM15 状态寄存器 (TIM15_SR) 1254

36.5.6 TIM15 事件产生寄存器 (TIM15_EGR) 1255

36.5.7 TIM15 捕获/比较模式寄存器 1 (TIM15_CCMR1) 1256

36.5.8 TIM15 捕获/比较使能寄存器 (TIM15_CCER) 1260
36.5.9 TIM15 计数器 (TIM15_CNT) 1263
36.5.10 TIM15 预分频器 (TIM15_PSC) 1263
36.5.11 TIM15 自动重载寄存器 (TIM15_ARR) 1263
36.5.12 TIM15 重复计数器寄存器 (TIM15_RCR) 1264
36.5.13 TIM15 捕获/比较寄存器 1 (TIM15_CCR1) 1264
36.5.14 TIM15 捕获/比较寄存器 2 (TIM15_CCR2) 1265
36.5.15 TIM15 断路和死区寄存器 (TIM15_BDTR) 1265
36.5.16 TIM15 DMA 控制寄存器 (TIM15_DCR) 1268
36.5.17 TIM15 全传输 DMA 地址 (TIM15_DMAR) 1268
36.5.18 TIM15 选项寄存器 1 (TIM15_OR1) 1269
36.5.19 TIM15 选项寄存器 2 (TIM15_OR2) 1269
36.5.20 TIM15寄存器映射 1271
36.6 TIM16/TIM17 寄存器 1274
36.6.1 TIM16/TIM17 控制寄存器 1 (TIMx_CR1) 1274
36.6.2 TIM16/TIM17 控制寄存器 2 (TIMx_CR2) 1275
36.6.3 TIM16/TIM17 DMA/中断使能寄存器 (TIMx_DIER) 1276
36.6.4 TIM16/TIM17 状态寄存器 (TIMx_SR) 1277
36.6.5 TIM16/TIM17 事件生成寄存器 (TIMx_EGR) 1278
36.6.6 TIM16/TIM17 捕获/比较模式寄存器 1 (TIMx_CCMR1) 1279
36.6.7 TIM16/TIM17 捕获/比较使能寄存器 (TIMx_CCER) 1281
36.6.8 TIM16/TIM17 计数器 (TIMx_CNT) 1284
36.6.9 TIM16/TIM17 预分频器 (TIMx_PSC) 1284
36.6.10 TIM16/TIM17 自动重载寄存器 (TIMx_ARR) 1284
36.6.11 TIM16/TIM17 重复计数器寄存器 (TIMx_RCR) 1285
36.6.12 TIM16/TIM17 捕获/比较寄存器 1 (TIMx_CCR1) 1285
36.6.13 TIM16/TIM17 断路和死区寄存器 (TIMx_BDTR) 1286
36.6.14 TIM16/TIM17 DMA 控制寄存器 (TIMx_DCR) 1288
36.6.15 TIM16/TIM17 全传输 DMA 地址 (TIMx_DMAR) 1289
36.6.16 TIM16 选项寄存器 1 (TIM16_OR1) 1289
36.6.17 TIM16 选项寄存器 2 (TIM16_OR2) 1290
36.6.18 TIM17 选项寄存器 1 (TIM17_OR1) 1291
36.6.19 TIM17 选项寄存器 2 (TIM17_OR2) 1291
36.6.20 TIM16/TIM17 寄存器映射 1293

37 基本定时器 (TIM6/TIM7) 1295
37.1 TIM6/TIM7 简介 1295
37.2 TIM6/TIM7 主要特性 1295
37.3 TIM6/TIM7 功能说明 1296
37.3.1 时基单元 1296
37.3.2 计数模式 1298
37.3.3 UIF 位重映射 1301
37.3.4 时钟源 1301
37.3.5 调试模式 1302
37.4 TIM6/TIM7 寄存器 1302
37.4.1 TIM6/TIM7 控制寄存器 1 (TIMx_CR1) 1302
37.4.2 TIM6/TIM7 控制寄存器 2 (TIMx_CR2) 1304
37.4.3 TIM6/TIM7 DMA/中断使能寄存器 (TIMx_DIER) 1304
37.4.4 TIM6/TIM7 状态寄存器 (TIMx_SR) 1305
37.4.5 TIM6/TIM7 事件产生寄存器 (TIMx_EGR) 1305
37.4.6 TIM6/TIM7 计数器 (TIMx_CNT) 1305
37.4.7 TIM6/TIM7 预分频器 (TIMx_PSC) 1306
37.4.8 TIM6/TIM7 自动重载寄存器 (TIMx_ARR) 1306
37.4.9 TIM6/TIM7 寄存器映射 1307

38 低功耗定时器 (LPTIM) 1308
38.1 简介 1308
38.2 LPTIM 主要特性 1308
38.3 LPTIM 实现 1308
38.4 LPTIM 功能说明 1309
38.4.1 LPTIM 框图 1309
38.4.2 LPTIM 触发映射 1309
38.4.3 LPTIM 复位和时钟 1310
38.4.4 干扰滤波器 1310
38.4.5 预分频器 1311
38.4.6 触发多路复用器 1312
38.4.7 工作模式 1312
38.4.8 超时功能 1314
38.4.9 生成波形 1314
38.4.10 寄存器更新 1315
38.4.11 计数器模式 1316

38.4.12 定时器使能 1316
38.4.13 定时器计数器复位 1317
38.4.14 编码器模式 1317
38.5 LPTIM 低功耗模式 1319
38.6 LPTIM 中断 1319
38.7 LPTIM 寄存器 1320
38.7.1 LPTIM 中断和状态寄存器 (LPTIM_ISR) 1320
38.7.2 LPTIM 中断清零寄存器 (LPTIM_ICR) 1321
38.7.3 LPTIM 中断使能寄存器 (LPTIM_IER) 1322
38.7.4 LPTIM 配置寄存器 (LPTIM_CFGR) 1323
38.7.5 LPTIM 控制寄存器 (LPTIM_CR) 1325
38.7.6 LPTIM 比较寄存器 (LPTIM_CMP) 1326
38.7.7 LPTIM 自动重载寄存器 (LPTIM_ARR) 1327
38.7.8 LPTIM 计数器寄存器 (LPTIM_CNT) 1327
38.7.9 LPTIM1 选项寄存器 (LPTIM1_OR) 1328
38.7.10 LPTIM2 选项寄存器 (LPTIM2_OR) 1328
38.7.11 LPTIM 寄存器映射 1329

39 红外接口 (IRTIM) 1330

40 独立看门狗 (IWDG) 1331
40.1 前言 1331
40.2 IWDG 主要特性 1331
40.3 IWDG 功能说明 1331
40.3.1 IWDG 框图 1331
40.3.2 窗口选项 1332
40.3.3 硬件看门狗 1333
40.3.4 低功耗冻结 1333
40.3.5 停止和待机模式下的行为 1333
40.3.6 寄存器访问保护 1333
40.3.7 调试模式 1333
40.4 IWDG 寄存器 1334
40.4.1 关键字寄存器 (IWDG_KR) 1334
40.4.2 预分频器寄存器 (IWDG_PR) 1335
40.4.3 重载寄存器 (IWDG_RLR) 1336
40.4.4 状态寄存器 (IWDG_SR) 1337

40.4.5 窗口寄存器 (IWDG_WINR) 1338
40.4.6 IWDG 寄存器映射 1339

41 系统窗口看门狗 (WWDG) 1340
41.1 简介 1340
41.2 WWDG 主要特性 1340
41.3 WWDG 功能说明 1340
41.3.1 使能看门狗 1341
41.3.2 控制递减计数器 1341
41.3.3 看门狗中断高级特性 1341
41.3.4 如何设置看门狗超时 1342
41.3.5 调试模式 1343
41.4 WWDG 寄存器 1343
41.4.1 控制寄存器 (WWDG_CR) 1343
41.4.2 配置寄存器 (WWDG_CFR) 1344
41.4.3 状态寄存器 (WWDG_SR) 1344
41.4.4 WWDG 寄存器映射 1345

42 实时时钟 (RTC) 1346
42.1 简介 1346
42.2 RTC 主要特性 1346
42.3 RTC 功能说明 1347
42.3.1 RTC 框图 1347
42.3.2 RTC 控制的 GPIO 1348
42.3.3 时钟和预分频器 1350
42.3.4 实时时钟和日历 1351
42.3.5 可编程闹钟 1351
42.3.6 周期性自动唤醒 1352
42.3.7 RTC 初始化和配置 1352
42.3.8 读取日历 1354
42.3.9 复位 RTC 1355
42.3.10 RTC 同步 1355
42.3.11 RTC 参考时钟检测 1356
42.3.12 RTC 精密数字校准 1356
42.3.13 时间戳功能 1358
42.3.14 入侵检测 1359

42.3.15 校准时钟输出 1360
42.3.16 闹钟输出 1361
42.4 RTC 低功耗模式 1361
42.5 RTC 中断 1362
42.6 RTC 寄存器 1363
42.6.1 RTC 时间寄存器 (RTC_TR) 1363
42.6.2 RTC 日期寄存器 (RTC_DR) 1364
42.6.3 RTC 控制寄存器 (RTC_CR) 1365
42.6.4 RTC 初始化和状态寄存器 (RTC_ISR) 1368
42.6.5 RTC 预分频器寄存器 (RTC_PRER) 1371
42.6.6 RTC 唤醒定时器寄存器 (RTC_WUTR) 1372
42.6.7 RTC 闹钟 A 寄存器 (RTC_ALRMAR) 1373
42.6.8 RTC 闹钟 B 寄存器 (RTC_ALRMBR) 1374
42.6.9 RTC 写保护寄存器 (RTC_WPR) 1375
42.6.10 RTC 亚秒寄存器 (RTC_SSR) 1375
42.6.11 RTC 平移控制寄存器 (RTC_SHIFTR) 1376
42.6.12 RTC 时间戳时间寄存器 (RTC_TSTR) 1377
42.6.13 RTC 时间戳日期寄存器 (RTC_TSDR) 1378
42.6.14 RTC 时间戳亚秒寄存器 (RTC_TSSSR) 1378
42.6.15 RTC 校准寄存器 (RTC_CALR) 1379
42.6.16 RTC 入侵配置寄存器 (RTC_TAMPCR) 1380
42.6.17 RTC 闹钟 A 亚秒寄存器 (RTC_ALRMASSR) 1383
42.6.18 RTC 闹钟 B 亚秒寄存器 (RTC_ALRMBSSR) 1384
42.6.19 RTC 选项寄存器 (RTC_OR) 1385
42.6.20 RTC 备份寄存器 (RTC_BKPxR) 1385
42.6.21 RTC 寄存器映射 1386

43 内部集成电路 (I2C) 接口 1388
43.1 简介 1388
43.2 I2C 主要特性 1388
43.3 I2C 特性实现 1389
43.4 I2C 功能说明 1389
43.4.1 I2C 框图 1390
43.4.2 I2C 时钟要求 1391
43.4.3 模式选择 1391
43.4.4 I2C 初始化 1392

43.4.5 软件复位 1396
43.4.6 数据传输 1397
43.4.7 I2C 从模式 1399
43.4.8 I2C 主模式 1407
43.4.9 I2C_TIMINGR 寄存器配置示例 1419
43.4.10 SMBus 特性 1421
43.4.11 SMBus 初始化 1423
43.4.12 SMBus：I2C_TIMEOUTR 寄存器配置示例 1425
43.4.13 SMBus 从模式 1426
43.4.14 地址匹配时从停止模式唤醒 1433
43.4.15 错误条件 1433
43.4.16 DMA 请求 1435
43.4.17 调试模式 1435
43.5 I2C 低功耗模式 1436
43.6 I2C 中断 1436
43.7 I2C 寄存器 1437
43.7.1 控制寄存器 1 (I2C_CR1) 1437
43.7.2 控制寄存器 2 (I2C_CR2) 1440
43.7.3 设备自身地址 1 寄存器 (I2C_OAR1) 1443
43.7.4 设备自身地址 2 寄存器 (I2C_OAR2) 1444
43.7.5 时序寄存器 (I2C_TIMINGR) 1445
43.7.6 超时寄存器 (I2C_TIMEOUTR) 1446
43.7.7 中断和状态寄存器 (I2C_ISR) 1447
43.7.8 中断清零寄存器 (I2C_ICR) 1449
43.7.9 PEC 寄存器 (I2C_PECR) 1450
43.7.10 接收数据寄存器 (I2C_RXDR) 1451
43.7.11 发送数据寄存器 (I2C_TXDR) 1451
43.7.12 I2C寄存器映射 1452

44 通用同步异步收发器 (USART) 1454
44.1 USART 简介 1454
44.2 USART 主要特性 1454
44.3 USART 扩展特性 1455
44.4 USART 实现 1455

44.5 USART 功能说明 1456
44.5.1 USART 框图 1456
44.5.2 USART 信号 1457
44.5.3 USART 字符说明 1458
44.5.4 USART FIFO 和阈值 1460
44.5.5 USART 发送器 1460
44.5.6 USART 接收器 1464
44.5.7 USART 波特率生成 1472
44.5.8 USART 接收器对时钟偏差的容差 1473
44.5.9 USART 自动波特率检测 1474
44.5.10 USART 多处理器通信 1476
44.5.11 USART Modbus 通信 1477
44.5.12 USART 极性控制 1478
44.5.13 USART LIN（局域互连网络）模式 1479
44.5.14 USART 同步模式 1481
44.5.15 USART 单线半双工通信 1485
44.5.16 USART 接收器超时 1485
44.5.17 USART 智能卡模式 1486
44.5.18 USART IrDA SIR ENDEC 模块 1490
44.5.19 使用 USART 和 DMA 进行连续通信 1492
44.5.20 RS232 硬件流控制和 RS485 驱动器使能 1494
44.5.21 USART 低功耗管理 1497
44.6 USART 中断 1500
44.7 USART 寄存器 1502
44.7.1 USART 控制寄存器 1 (USART_CR1) 1502
44.7.2 USART 控制寄存器 2 (USART_CR2) 1506
44.7.3 USART 控制寄存器 3 (USART_CR3) 1510
44.7.4 USART 波特率寄存器 (USART_BRR) 1514
44.7.5 USART 保护时间和预分频器寄存器 (USART_GTPR) 1514
44.7.6 USART 接收器超时寄存器 (USART_RTOR) 1515
44.7.7 USART 请求寄存器 (USART_RQR) 1516
44.7.8 USART 中断和状态寄存器 (USART_ISR) 1517
44.7.9 USART 中断标志清零寄存器 (USART_ICR) 1523
44.7.10 USART 接收数据寄存器 (USART_RDR) 1525
44.7.11 USART 发送数据寄存器 (USART_TDR) 1525
44.7.12 USART 预分频器寄存器 (USART_PRESC) 1526
44.7.13 USART 寄存器映射 1527

45 低功耗通用异步接收器 (LPUART) 1529
45.1 LPUART 简介 1529
45.2 LPUART 主要特性 1530
45.3 LPUART 功能说明 1531
45.3.1 LPUART 框图 1531
45.3.2 LPUART 信号 1532
45.3.3 LPUART 字符说明 1532
45.3.4 LPUART FIFO 和阈值 1534
45.3.5 LPUART 发送器 1534
45.3.6 LPUART 接收器 1537
45.3.7 LPUART 波特率生成 1540
45.3.8 LPUART 接收器对时钟偏差的容差 1542
45.3.9 LPUART 多处理器通信 1543
45.3.10 LPUART 极性控制 1545
45.3.11 LPUART 单线半双工通信 1546
45.3.12 使用 DMA 和 LPUART 进行连续通信 1546
45.3.13 RS232 硬件流控制和 RS485 驱动器使能 1549
45.3.14 LPUART 低功耗管理 1551
45.4 LPUART 中断 1554
45.5 LPUART寄存器 1556
45.5.1 控制寄存器 1 (LPUART_CR1) 1556
45.5.2 控制寄存器 2 (LPUART_CR2) 1559
45.5.3 控制寄存器 3 (LPUART_CR3) 1560
45.5.4 波特率寄存器 (LPUART_BRR) 1563
45.5.5 请求寄存器 (LPUART_RQR) 1564
45.5.6 中断和状态寄存器 (LPUART_ISR) 1564
45.5.7 中断标志清零寄存器 (LPUART_ICR) 1569
45.5.8 接收数据寄存器 (LPUART_RDR) 1570
45.5.9 发送数据寄存器 (LPUART_TDR) 1570
45.5.10 预分频器寄存器 (LPUART_PRESC) 1571
45.5.11 LPUART 寄存器映射 1572

46 串行外设接口 (SPI) 1574
46.1 简介 1574
46.2 SPI 主要特性 1574
46.3 SPI 实现 1574

46.4 SPI 功能说明 1575
46.4.1 概述 1575
46.4.2 一个主器件和一个从器件之间的通信 1576
46.4.3 标准多从器件通信 1578
46.4.4 多主通信 1579
46.4.5 从器件选择 (NSS) 引脚管理 1580
46.4.6 通信格式 1581
46.4.7 SPI 配置 1583
46.4.8 使能 SPI 的步骤 1584
46.4.9 数据发送和接收过程 1584
46.4.10 SPI 状态标志 1593
46.4.11 SPI 错误标志 1594
46.4.12 NSS 脉冲模式 1595
46.4.13 TI 模式 1595
46.4.14 CRC 计算 1596
46.5 SPI 中断 1598
46.6 SPI 寄存器 1599
46.6.1 SPI 控制寄存器 1 (SPIx_CR1) 1599
46.6.2 SPI 控制寄存器 2 (SPIx_CR2) 1601
46.6.3 SPI 状态寄存器 (SPIx_SR) 1604
46.6.4 SPI 数据寄存器 (SPIx_DR) 1605
46.6.5 SPI CRC 多项式寄存器 (SPIx_CRCPR) 1605
46.6.6 SPI 接收 CRC 寄存器 (SPIx_RXCRCR) 1606
46.6.7 SPI 发送 CRC 寄存器 (SPIx_TXCRCR) 1606
46.6.8 SPI 寄存器映射 1607

47 串行音频接口 (SAI) 1608
47.1 简介 1608
47.2 SAI 主要特性 1609
47.3 SAI 功能说明 1610
47.3.1 SAI 框图 1610
47.3.2 SAI 引脚和内部信号 1611
47.3.3 SAI 的主要模式 1611
47.3.4 SAI 同步模式 1612
47.3.5 音频数据大小 1613
47.3.6 帧同步 1614

47.3.7 Slot 配置 1617
47.3.8 SAI 时钟发生器 1619
47.3.9 内部 FIFO 1621
47.3.10 PDM 接口 1623
47.3.11 AC'97 链路控制器 1630
47.3.12 SPDIF 输出 1632
47.3.13 特性 1634
47.3.14 错误标志 1638
47.3.15 禁止 SAI 1641
47.3.16 SAI DMA 接口 1641
47.4 SAI 中断 1642
47.5 SAI 寄存器 1643
47.5.1 全局配置寄存器 (SAI_GCR) 1643
47.5.2 配置寄存器 1 (SAI_ACR1/SAI_BCR1) 1643
47.5.3 配置寄存器 2 (SAI_ACR2/SAI_BCR2) 1646
47.5.4 帧配置寄存器 (SAI_AFRCR/SAI_BFRCR) 1648
47.5.5 Slot 寄存器 (SAI_ASLOTR/SAI_BSLOTR) 1649
47.5.6 中断屏蔽寄存器 2 (SAI_AIM/SAI_BIM) 1650
47.5.7 状态寄存器 (SAI_ASR/SAI_BSR) 1651
47.5.8 清除标志寄存器 (SAI_ACLRFR/SAI_BCLRFR) 1653
47.5.9 数据寄存器 (SAI_ADR/SAI_BDR) 1654
47.5.10 PDM 控制寄存器 (SAI_PDMCR) 1655
47.5.11 PDM 延迟寄存器 (SAI_PDMDLY) 1656
47.5.12 SAI 寄存器映射 1659

48 安全数字输入/输出多媒体卡接口 (SDMMC) 1661
48.1 SDMMC 主要特性 1661
48.2 SDMMC 总线拓扑 1661
48.3 SDMMC 工作模式 1663
48.4 SDMMC 功能说明 1664
48.4.1 SDMMC 图 1664
48.4.2 SDMMC 引脚和内部信号 1665
48.4.3 概述 1665
48.4.4 SDMMC 适配器 1667
48.4.5 SDMMC AHB 从接口 1686

48.4.6 SDMMC AHB 主接口 1687
48.4.7 AHB 和 SDMMC_CK 时钟的关系 1688
48.5 卡功能说明 1689
48.5.1 SD I/O 模式 1689
48.5.2 CMD12 发送时序 1697
48.5.3 睡眠 (CMD5) 1699
48.5.4 中断模式 (Wait-IRQ) 1700
48.5.5 启动操作 1701
48.5.6 响应 R1b 的处理 1704
48.5.7 复位和卡掉电再上电 1705
48.6 硬件流控制 1706
48.7 超高速 I 相 (UHS-I) 电压切换 1707
48.8 SDMMC 寄存器 1709
48.8.1 SDMMC 电源控制寄存器 (SDMMC_POWER) 1709
48.8.2 SDMMC 时钟控制寄存器 (SDMMC_CLKCR) 1710
48.8.3 SDMMC 参数寄存器 (SDMMC_ARGR) 1712
48.8.4 SDMMC 命令寄存器 (SDMMC_CMDR) 1713
48.8.5 SDMMC 命令响应寄存器 (SDMMC_RESPCMDR) 1714
48.8.6 SDMMC 响应 x 寄存器 (SDMMC_RESPxR) 1715
48.8.7 SDMMC 数据定时器寄存器 (SDMMC_DTIMER) 1715
48.8.8 SDMMC 数据长度寄存器 (SDMMC_DLENR) 1716
48.8.9 SDMMC 数据控制寄存器 (SDMMC_DCTRL) 1717
48.8.10 SDMMC 数据计数器寄存器 (SDMMC_DCNTR) 1719
48.8.11 SDMMC 状态寄存器 (SDMMC_STAR) 1720
48.8.12 SDMMC 中断清零寄存器 (SDMMC_ICR) 1722
48.8.13 SDMMC 屏蔽寄存器 (SDMMC_MASKR) 1725
48.8.14 SDMMC 确认定时器寄存器 (SDMMC_ACKTIMER) 1728
48.8.15 SDMMC 数据 FIFO 寄存器 (SDMMC_FIFOR) 1728
48.8.16 SDMMC DMA 控制寄存器 (SDMMC_IDMACTRLR) 1729
48.8.17 SDMMC IDMA 缓冲区大小寄存器 (SDMMC_IDMABSIZER) 1730
48.8.18 SDMMC IDMA 缓冲区 0 基址寄存器 (SDMMC_IDMABASE0R) 1730
48.8.19 SDMMC IDMA 缓冲区 1 基址寄存器 (SDMMC_IDMABASE1R) 1731
48.8.20 SDMMC 寄存器映射 1732

49 控制器局域网络 (bxCAN) 1735
49.1 简介 1735
49.2 bxCAN 主要特性 1735
49.3 bxCAN 一般说明 1736
49.3.1 CAN 2.0B 主动内核 1736
49.3.2 控制、状态和配置寄存器 1736
49.3.3 发送邮箱 1736
49.3.4 验收过滤器 1737
49.4 bxCAN 工作模式 1738
49.4.1 初始化模式 1738
49.4.2 正常模式 1738
49.4.3 睡眠模式（低功耗） 1739
49.5 测试模式 1740
49.5.1 静默模式 1740
49.5.2 环回模式 1740
49.5.3 环回与静默组合模式 1741
49.6 调试模式下的行为 1741
49.7 bxCAN 功能说明 1742
49.7.1 发送处理 1742
49.7.2 时间触发通信模式 1743
49.7.3 接收处理 1743
49.7.4 标识符筛选 1745
49.7.5 消息存储 1749
49.7.6 错误管理 1750
49.7.7 位时序 1750
49.8 bxCAN 中断 1753
49.9 CAN 寄存器 1754
49.9.1 寄存器访问保护 1754
49.9.2 CAN 控制和状态寄存器 1755
49.9.3 CAN 邮箱寄存器 1764
49.9.4 CAN 过滤器寄存器 1771
49.9.5 bxCAN 寄存器映射 1775

50 USB on-the-go 全速 (OTG_FS) 1779
50.1 简介 1779
50.2 OTG 主要特性 1780

50.2.1 通用特性 1780
50.2.2 主机模式特性 1781
50.2.3 从机模式特性 1781
50.3 OTG 实现 1781
50.4 OTG 功能说明 1782
50.4.1 OTG 框图 1782
50.4.2 USB OTG 引脚和内部信号 1782
50.4.3 OTG 模块 1783
50.4.4 全速 OTG PHY 1783
50.5 OTG 双角色设备 (DRD) 1784
50.5.1 ID 线检测 1784
50.5.2 HNP 双角色设备 1784
50.5.3 SRP 双角色设备 1785
50.6 USB 设备 1785
50.6.1 支持 SRP 功能的 USB 设备 1786
50.6.2 USB 设备状态 1786
50.6.3 USB 设备端点 1787
50.7 USB 主机 1789
50.7.1 支持 SRP 功能的 USB 主机 1789
50.7.2 USB 主机状态 1790
50.7.3 主机通道 1791
50.7.4 主机调度器 1792
50.8 SOF 触发 1793
50.8.1 主机 SOF 1793
50.8.2 设备 SOF 1793
50.9 OTG 低功耗模式 1794
50.10 动态更新 OTG_HFIR 寄存器 1795
50.11 USB 数据 FIFO 1795
50.11.1 设备 FIFO 架构 1796
50.11.2 主机 FIFO 架构 1797
50.11.3 FIFO RAM 分配 1798
50.12 OTG_FS 系统性能 1800
50.13 OTG_FS 中断 1800
50.14 OTG_FS 控制和状态寄存器 1802
50.14.1 CSR 存储器映射 1802

50.15 OTG_FS 寄存器 1806
50.15.1 OTG 控制和状态寄存器 (OTG_GOTGCTL) 1807
50.15.2 OTG 中断寄存器 (OTG_GOTGINT) 1810
50.15.3 OTG AHB 配置寄存器 (OTG_GAHBCFG) 1811
50.15.4 OTG USB 配置寄存器 (OTG_GUSBCFG) 1812
50.15.5 OTG 复位寄存器 (OTG_GRSTCTL) 1814
50.15.6 OTG 模块中断寄存器 (OTG_GINTSTS) 1816
50.15.7 OTG 中断屏蔽寄存器 (OTG_GINTMSK) 1820
50.15.8 OTG 接收状态调试读取/OTG 状态读取和出栈寄存器 (OTG_GRXSTSR/OTG_GRXSTSP) 1823
50.15.9 OTG 接收 FIFO 大小寄存器 (OTG_GRXFSIZ) 1825
50.15.10 OTG 主机非周期性发送 FIFO 大小寄存器 (OTG_HNPTXFSIZ)/ 端点 0 发送 FIFO 大小 (OTG_DIEPTXF0) 1825
50.15.11 OTG 非周期性发送 FIFO/队列状态寄存器 (OTG_HNPTXSTS) 1826
50.15.12 OTG 通用模块配置寄存器 (OTG_GCCFG) 1827
50.15.13 OTG 模块 ID 寄存器 (OTG_CID) 1829
50.15.14 OTG 模块 LPM 配置寄存器 (OTG_GLPMCFG) 1829
50.15.15 OTG 掉电寄存器 (OTG_GPWRDN) 1833
50.15.16 OTG ADP 定时器、控制和状态寄存器 (OTG_GADPCTL) 1833
50.15.17 OTG 主机周期性发送 FIFO 大小寄存器 (OTG_HPTXFSIZ) 1835
50.15.18 OTG 设备 IN 端点发送 FIFO 大小寄存器 (OTG_DIEPTXFx) （x = 1..5，其中 x 为 FIFO 号） 1836
50.15.19 主机模式寄存器 1836
50.15.20 OTG 主机配置寄存器 (OTG_HCFG) 1836
50.15.21 OTG 主机帧间隔寄存器 (OTG_HFIR) 1837
50.15.22 OTG 主机帧编号/帧剩余时间寄存器 (OTG_HFNUM) 1838
50.15.23 OTG_Host 周期性发送 FIFO/队列状态寄存器 (OTG_HPTXSTS) . . . 1839
50.15.24 OTG 主机全体通道中断寄存器 (OTG_HAINT) 1840
50.15.25 OTG 主机全体通道中断屏蔽寄存器 (OTG_HAINTMSK) 1840
50.15.26 OTG 主机端口控制和状态寄存器 (OTG_HPRT) 1841
50.15.27 OTG 主机通道 x 特性寄存器 (OTG_HCCHARx) （x = 0..11，其中 x 表示通道编号） 1843
50.15.28 OTG 主机通道 x 中断寄存器 (OTG_HCINTx) （x = 0..11，其中 x 表示通道编号） 1844
50.15.29 OTG 主机通道 x 中断屏蔽寄存器 (OTG_HCINTMSKx) （x = 0..11，其中 x 表示通道编号） 1845
50.15.30 OTG 主机通道 x 传输大小寄存器 (OTG_HCTSIZx) （x = 0..11，其中 x 表示通道编号） 1846
50.15.31 设备模式寄存器 1847

50.15.32 OTG 设备配置寄存器 (OTG_DCFG) 1847
50.15.33 OTG 设备控制寄存器 (OTG_DCTL) 1848
50.15.34 OTG 设备状态寄存器 (OTG_DSTS) 1850
50.15.35 OTG 设备 IN 端点通用中断屏蔽寄存器 (OTG_DIEPMSK) 1851
50.15.36 OTG 设备 OUT 端点通用中断屏蔽寄存器 (OTG_DOEPMSK) 1852
50.15.37 OTG 设备全体端点中断寄存器 (OTG_DAINT) 1854
50.15.38 OTG 全体端点中断屏蔽寄存器 (OTG_DAINTMSK) 1854
50.15.39 OTG 设备 V_{BUS} 放电时间寄存器 (OTG_DVBUSDIS) 1855
50.15.40 OTG 设备 V_{BUS} 脉冲时间寄存器 (OTG_DVBUSPULSE) 1856
50.15.41 OTG 设备 IN 端点 FIFO 空中断屏蔽寄存器 (OTG_DIEPEMPMSK) . . . 1856
50.15.42 OTG 设备控制 IN 端点 0 控制寄存器 (OTG_DIEPCTL0) 1857
50.15.43 OTG 设备 IN 端点 x 控制寄存器 (OTG_DIEPCTLx)
（x = 1..5，其中 x 表示端点编号） 1858
50.15.44 OTG 设备 IN 端点 x 中断寄存器 (OTG_DIEPINTx)
（x = 0..5，其中 x 表示端点编号） 1860
50.15.45 OTG 设备 IN 端点 0 传输大小寄存器 (OTG_DIEPTSIZ0) 1862
50.15.46 OTG 设备 IN 端点发送 FIFO 状态寄存器 (OTG_DTXFSTSx)
（x = 0..5，其中 x = 端点编号） 1863
50.15.47 OTG 设备 IN 端点 x 传输大小寄存器 (OTG_DIEPTSIZx)
（x = 1..5，其中 x 表示端点编号） 1863
50.15.48 OTG 设备控制 OUT 端点 0 控制寄存器 (OTG_DOEPCTL0) 1864
50.15.49 OTG 设备 OUT 端点 x 中断寄存器 (OTG_DOEPINTx)
（x = 0..5，其中 x 表示端点编号） 1865
50.15.50 OTG 设备 OUT 端点 0 传输大小寄存器 (OTG_DOEPTSIZ0) 1867
50.15.51 OTG 设备 OUT 端点 x 控制寄存器 (OTG_DOEPCTLx)
（x = 1..5，其中 x 表示端点编号） 1868
50.15.52 OTG 设备 OUT 端点 x 传输大小寄存器 (OTG_DOEPTSIZx)
（x = 1..5，其中 x 表示端点编号） 1870
50.15.53 OTG 电源和时钟门控控制寄存器 (OTG_PCGCCTL) 1871
50.15.54 OTG_FS 寄存器映射 1872
50.16 OTG_FS 编程模块 1880
50.16.1 模块初始化 1880
50.16.2 主机初始化 1881
50.16.3 设备初始化 1881
50.16.4 主机编程模型 1882
50.16.5 设备编程模型 1901
50.16.6 最坏情况下的响应时间 1918
50.16.7 OTG 编程模型 1920

51 调试支持 (DBG) 1925

51.1 概述 1925

51.2 Arm® 参考文档 1926

51.3 SWJ 调试端口（串行接口和 JTAG） 1926

51.3.1 JTAG-DP 或 SW-DP 的切换机制 1927

51.4 引脚排列和调试端口引脚 1927

51.4.1 SWJ 调试端口引脚 1928

51.4.2 灵活的 SWJ-DP 引脚分配 1928

51.4.3 JTAG引脚上的内部上拉和下拉 1929

51.4.4 使用串行接口以及释放未使用的调试引脚以用作 GPIO 1929

51.5 STM32L4Rxxx 和 STM32L4Sxxx JTAG TAP 连接 1930

51.6 ID 代码和锁定机制 1930

51.6.1 MCU 器件 ID 代码 1931

51.6.2 边界扫描 TAP 1931

51.6.3 Cortex®-M4 TAP 1931

51.6.4 Cortex®-M4 JEDEC-106 ID 代码 1931

51.7 JTAG 调试端口 1932

51.8 SW 调试端口 1933

51.8.1 SW 协议简介 1933

51.8.2 SW 协议序列 1934

51.8.3 SW-DP 状态机（复位、空闲状态、ID 代码） 1935

51.8.4 DP 和 AP 读/写访问 1935

51.8.5 SW-DP 寄存器 1936

51.8.6 SW-AP 寄存器 1936

51.9 AHB-AP（AHB 访问端口）——对 JTAG-DP 和 SW-DP 同时有效 . . . 1937

51.10 内核调试 1938

51.11 调试主机在系统复位状态下建立连接的功能 1938

51.12 FPB（Flash 补丁断点） 1939

51.13 DWT（数据观察点触发） 1939

51.14 ITM（指令跟踪宏单元） 1940

51.14.1 概述 1940

51.14.2 时间戳数据包、同步和溢出数据包 1940

51.15 ETM（嵌入式跟踪宏单元） 1942

51.15.1 概述 1942

51.15.2 信号协议和数据包类型 1942

51.15.3 主要的 ETM 寄存器 1942
51.15.4 配置示例 1943
51.16 MCU 调试组件 (DBGMCU) 1943
51.16.1 对低功耗模式的调试支持 1943
51.16.2 对定时器、RTC、看门狗、bxCAN 和 I^2C 的调试支持 1944
51.16.3 调试 MCU 配置寄存器 (DBGMCU_CR) 1944
51.16.4 MCU APB1 调试冻结寄存器 1 (DBGMCU_APB1FZR1) 1945
51.16.5 调试 MCU APB1 冻结寄存器 2 (DBGMCU_APB1FZR2) 1947
51.16.6 调试 MCU APB2 冻结寄存器 (DBGMCU_APB2FZR) 1948
51.17 TPIU（跟踪端口接口单元） 1949
51.17.1 简介 1949
51.17.2 TRACE 引脚分配 1949
51.17.3 TPUI 格式化器 1951
51.17.4 TPUI 帧同步数据包 1952
51.17.5 发送同步帧数据包 1952
51.17.6 同步模式 1952
51.17.7 异步模式 1953
51.17.8 STM32L4Rxxx 和 STM32L4Sxxx 内的 TRACECLKIN 连接 STM32L4Rxxx 和 STM32L4Sxxx 1953
51.17.9 TPIU 寄存器 1954
51.17.10 配置示例 1955
51.18 DBG 寄存器映射 1956

52 器件电子签名 1957
52.1 唯一器件 ID 寄存器（96 位） 1957
52.2 Flash 大小数据寄存器 1958
52.3 封装数据寄存器 1959

53 版本历史 1960

表格索引

表 1. STM32L4Rxxx 和 STM32L4Sxxx 存储器映射和外设寄存器边界地址 79
表 2. SRAM2 组织构成 86
表 3. 自举模式 88
表 4. 存储器映射与自举模式/物理重映射 90
表 5. Flash 模块——2 MB 双存储区结构（64 位读宽度） 93
表 6. Flash 模块——2 MB 单存储区结构（128 位读宽度） 93
表 7. CPU 时钟 (HCLK) 频率对应的等待状态数 95
表 8. 选项字节格式 104
表 9. 选项字节构成 105
表 10. Flash 读保护状态 114
表 11. 访问状态 vs 保护级别和执行模式 116
表 12. PCROP 保护 118
表 13. WRP 保护 120
表 14. Flash 中断请求 120
表 15. Flash 接口——寄存器映射和复位值 138
表 16. 不同防火墙状态下的段访问 143
表 17. 段粒度和区域范围 143
表 18. 防火墙寄存器映射与复位值 151
表 19. 范围 1 升压模式配置 157
表 20. PVM 功能 160
表 21. 低功耗模式汇总 164
表 22. 功能取决于工作模式 165
表 23. 低功耗运行 169
表 24. 睡眠 170
表 25. 低功耗睡眠 171
表 26. 停止 0 模式 173
表 27. 停止 1 模式 174
表 28. 停止 2 模式 176
表 29. 待机模式 178
表 30. 关断模式 180
表 31. PWR 寄存器映射和复位值 199
表 32. 时钟源频率 211
表 33. RCC 寄存器映射和复位值 276
表 34. 低功耗模式对 CRS 的作用 285
表 35. 中断控制位 285
表 36. CRS 寄存器映射和复位值 291
表 37. 端口位配置表 294
表 38. GPIO 寄存器映射和复位值 307
表 39. BOOSTEN 和 ANASWVDD 置 1/复位 312
表 40. SYSCFG 寄存器映射和复位值 321
表 41. STM32L4Rxxx 和 STM32L4Sxxx 外设互连矩阵 323
表 42. DMA1 和 DMA2 实现 332
表 43. DMA 内部输入/输出信号 334
表 44. 可编程的数据宽度和字节序（PINC = MINC = 1 时） 338
表 45. DMA 中断请求 339
表 46. DMA 寄存器映射和复位值 348
表 47. DMAMUX 实例化 351

表 48. DMAMUX：复用器输入到资源的分配 . . . 351
表 49. DMAMUX：触发输入到资源的分配 . . . 352
表 50. DMAMUX：同步输入到资源的分配 . . . 353
表 51. DMAMUX 信号 . . . 355
表 52. DMAMUX 中断 . . . 359
表 53. DMAMUX 寄存器映射和复位值 . . . 364
表 54. 输入时支持的颜色模式 . . . 368
表 55. 存储器中的数据顺序 . . . 369
表 56. Alpha 模式配置 . . . 370
表 57. 支持的 CLUT 颜色模式 . . . 371
表 58. 系统存储器中的 CLUT 数据顺序 . . . 371
表 59. 输出时支持的颜色模式 . . . 372
表 60. 存储器中的数据顺序 . . . 372
表 61. 存储器中的标准数据顺序 . . . 373
表 62. 输出 FIFO 字节重新排序步骤 . . . 374
表 63. DMA2D 中断请求 . . . 380
表 64. DMA2D 寄存器映射和复位值 . . . 398
表 65. 图形 MMU 中断请求 . . . 406
表 66. 图形 MMU 寄存器映射和复位值 . . . 413
表 67. STM32L4Rxxx 和 STM32L4Sxxx 向量表 . . . 415
表 68. EXTI 线连接 . . . 423
表 69. 扩展中断/事件控制器寄存器映射和复位值 . . . 431
表 70. CRC 内部输入/输出信号 . . . 433
表 71. CRC 寄存器映射和复位值 . . . 437
表 72. NOR/PSRAM 存储区域选择 . . . 442
表 73. NOR/PSRAM 外部存储器地址 . . . 442
表 74. NAND 存储映射和时序寄存器 . . . 442
表 75. NAND 存储区域选择 . . . 443
表 76. NOR/PSRAM 的可编程访问参数 . . . 444
表 77. 非复用 I/O NOR Flash . . . 445
表 78. 16 位复用 I/O NOR Flash . . . 445
表 79. 非复用 I/O PSRAM/SRAM . . . 446
表 80. 16 位复用 I/O PSRAM . . . 446
表 81. NOR Flash/PSRAM：支持的存储器和事务示例 . . . 447
表 82. FSMC_BCRx 位域 . . . 450
表 83. FSMC_BTRx 位域 . . . 450
表 84. FSMC_BCRx 位域 . . . 452
表 85. FSMC_BTRx 位域 . . . 452
表 86. FSMC_BWTRx 位域 . . . 453
表 87. FSMC_BCRx 位域 . . . 455
表 88. FSMC_BTRx 位域 . . . 455
表 89. FSMC_BWTRx 位域 . . . 456
表 90. FSMC_BCRx 位域 . . . 457
表 91. FSMC_BTRx 位域 . . . 458
表 92. FSMC_BWTRx 位域 . . . 458
表 93. FSMC_BCRx 位域 . . . 460
表 94. FSMC_BTRx 位域 . . . 461
表 95. FSMC_BWTRx 位域 . . . 461
表 96. FSMC_BCRx 位域 . . . 463
表 97. FSMC_BTRx 位域 . . . 464
表 98. FSMC_BCRx 位域 . . . 469
表 99. FSMC_BTRx 位域 . . . 470

表 100. FSMC_BCRx 位域 . . . 471
表 101. FSMC_BTRx 位域 . . . 472
表 102. 可编程的 NAND Flash 访问参数 . . . 480
表 103. 8 位 NAND Flash . . . 480
表 104. 16 位 NAND Flash . . . 481
表 105. 支持的存储器和事务 . . . 481
表 106. ECC 结果相关位 . . . 490
表 107. FSMC 寄存器映射 . . . 491
表 108. 命令/地址阶段说明 . . . 502
表 109. OCTOSPI 中断请求 . . . 517
表 110. OCTOSPI 寄存器映射 . . . 543
表 111. OCTOSPI IO 管理器寄存器映射和复位值 . . . 551
表 112. ADC 内部输入/输出信号 . . . 555
表 113. ADC 输入/输出引脚 . . . 555
表 114. 为常规外部触发配置触发极性 . . . 570
表 115. 为注入外部触发配置触发极性 . . . 570
表 116. ADC1——常规通道的外部触发 . . . 571
表 117. ADC1——注入通道的外部触发 . . . 571
表 118. TSAR 与分辨率的对应关系 . . . 583
表 119. 偏移计算与数据分辨率 . . . 586
表 120. 模拟看门狗通道选择 . . . 596
表 121. 模拟看门狗 1 比较 . . . 597
表 122. 模拟看门狗 2 和 3 比较 . . . 597
表 123. 最大输出结果与 N 和 M 的对应关系（灰色单元格表示截断的部分） . . . 601
表 124. ADC 中断 . . . 609
表 125. ADC 寄存器映射和复位值 . . . 637
表 126. ADC 寄存器映射和复位值（主 ADC 和从 ADC 通用寄存器，偏移 = 0x300） . . . 640
表 127. DAC 实现 . . . 642
表 128. DAC 输入/输出引脚 . . . 644
表 129. DAC 触发选择 . . . 647
表 130. 采样时间和刷新时间 . . . 651
表 131. 通道输出模式汇总 . . . 653
表 132. DAC 低功耗模式的作用 . . . 658
表 133. DAC 中断 . . . 658
表 134. DAC 寄存器映射和复位值 . . . 674
表 135. DCMI 外部信号 . . . 678
表 136. 捕获的数据字节在 32 位字（宽 8 位）中的位置排布 . . . 679
表 137. 捕获的数据字节在 32 位字（宽 10 位）中的位置排布 . . . 679
表 138. 捕获的数据字节在 32 位字（宽 12 位）中的位置排布 . . . 679
表 139. 捕获的数据字节在 32 位字（宽 14 位）中的位置排布 . . . 679
表 140. 单色逐行视频格式的数据存储 . . . 685
表 141. 以 RGB 逐行视频格式存储数据 . . . 685
表 142. YCbCr 逐行视频格式下的数据存储 . . . 686
表 143. YCbCr 逐行视频格式下的数据存储——Y 分量提取模式 . . . 686
表 144. DCMI 中断 . . . 687
表 145. DCMI 寄存器地址映射和复位值 . . . 699
表 146. VREF 缓冲器模式 . . . 700
表 147. VREFBUF 寄存器映射和复位值 . . . 702
表 148. COMP1 正输入分配 . . . 704
表 149. COMP1 负输入分配 . . . 704
表 150. COMP2 正输入分配 . . . 705

表 151. COMP2 负输入分配 . . . 705
表 152. 低功耗模式下的比较器行为 . . . 708
表 153. 中断控制位 . . . 709
表 154. COMP 寄存器映射和复位值 . . . 713
表 155. 运算放大器连接情形 . . . 715
表 156. 工作模式和校准 . . . 720
表 157. 低功耗模式对 OPAMP 的影响 . . . 721
表 158. OPAMP 寄存器映射和复位值 . . . 727
表 159. DFSDM1 实现 . . . 730
表 160. DFSDM 外部引脚 . . . 732
表 161. DFSDM 内部信号 . . . 732
表 162. DFSDM 触发器连接 . . . 732
表 163. DFSDM 断路连接 . . . 733
表 164. 滤波器最大输出分辨率（来自滤波器输出的峰值数据值），基于某些 FOSR 值 . . . 747
表 165. 积分器最大输出分辨率（来自积分器输出的峰值数据值），基于某些 IOSR 值，FOSR = 256 以及 Sinc3 滤波器类型（最大数据） . . . 747
表 166. DFSDM 中断请求 . . . 754
表 167. DFSDM 寄存器映射和复位值 . . . 776
表 168. LTDC 引脚和信号接口 . . . 787
表 169. 各个寄存器的时钟域 . . . 788
表 170. 像素数据映射与颜色格式的关系 . . . 793
表 171. LTDC 中断请求 . . . 796
表 172. LTDC 寄存器映射和复位值 . . . 817
表 173. LTDC 接口中色彩分量的位置 . . . 824
表 174. 每个数据类型的有效负载大小（以像素为单位）的多样性 . . . 825
表 175. 争用检测超时计数器配置 . . . 835
表 176. PRESP_TO 计数器不同类别的事件列表 . . . 836
表 177. PRESP_TO 计数器配置 . . . 839
表 178. 帧请求配置寄存器 . . . 850
表 179. RGB 分量 . . . 852
表 180. 压摆率和延迟调整 . . . 854
表 181. 自定义通道配置 . . . 855
表 182. 自定义时序参数 . . . 855
表 183. DSI 封装器中断请求 . . . 858
表 184. 错误原因和恢复 . . . 860
表 185. DSIHOST 寄存器映射和复位值 . . . 930
表 186. 采集序列汇总 . . . 938
表 187. 扩频偏差与 AHB 时钟频率之间的关系 . . . 940
表 188. I/O 状态（取决于其模式和 IODEF 位值） . . . 941
表 189. 低功耗模式对 TSC 的作用 . . . 943
表 190. 中断控制位 . . . 943
表 191. TSC 寄存器映射和复位值 . . . 952
表 192. RNG 内部输入/输出信号 . . . 955
表 193. RNG 中断请求 . . . 960
表 194. RNG 寄存器映射和复位值 . . . 964
表 195. AES 内部输入/输出信号 . . . 966
表 196. CTR 模式初始化矢量定义 . . . 985
表 197. GCM 最后一个块定义 . . . 986
表 198. GCM 模式 IVI 位域初始化 . . . 987
表 199. 在 CCM 模式下初始化 AES_IVRx 寄存器 . . . 993
表 200. AES_KEYRx 寄存器中的密钥字节序（128 位或 256 位密钥长度） . . . 999

表 201. 用于存储器到 AES 数据传输的 DMA 通道配置 . . . 1000
表 202. 用于 AES 到存储器数据传输的 DMA 通道配置 . . . 1001
表 203. AES 中断请求 . . . 1003
表 204. ECB、CBC 和 CTR 模式下的处理延迟（以时钟周期计） . . . 1003
表 205. GCM 和 CCM 模式下的处理延迟（以时钟周期计） . . . 1003
表 206. AES 寄存器映射和复位值 . . . 1016
表 207. HASH 内部输入/输出信号 . . . 1019
表 208. 散列处理器输出 . . . 1023
表 209. HASH 中断请求 . . . 1028
表 210. 处理时间（时钟周期数） . . . 1029
表 211. HASH 寄存器映射和复位值 . . . 1039
表 212. 定时器输出行为与 BRK/BRK2 输入 . . . 1080
表 213. 断路保护解除条件 . . . 1082
表 214. 计数方向与编码器信号的关系 . . . 1088
表 215. TIMx 内部触发连接 . . . 1103
表 216. 具有断路功能的互补通道 OCx 和 OCxN 的输出控制位 . . . 1116
表 217. TIM1 寄存器映射和复位值 . . . 1135
表 218. TIM8 寄存器映射和复位值 . . . 1138
表 219. 计数方向与编码器信号的关系 . . . 1173
表 220. TIMx 内部触发连接 . . . 1190
表 221. 标准 OCx 通道的输出控制位 . . . 1200
表 222. TIM2/TIM3/TIM4/TIM5 寄存器映射和复位值 . . . 1208
表 223. 断路保护解除条件 . . . 1239
表 224. TIMx 内部触发连接 . . . 1252
表 225. 具有断路功能的互补通道 OCx 和 OCxN 的输出控制位 (TIM15) . . . 1262
表 226. TIM15 寄存器映射和复位值 . . . 1271
表 227. 具有断路功能的互补通道 OCx 和 OCxN 的输出控制位 (TIM16/17) . . . 1283
表 228. TIM16/TIM17 寄存器映射和复位值 . . . 1293
表 229. TIM6/TIM7 寄存器映射和复位值 . . . 1307
表 230. STM32L4Rxxx 和 STM32L4Sxxx LPTIM 特性 . . . 1308
表 231. LPTIM1 外部触发连接 . . . 1309
表 232. LPTIM2 外部触发连接 . . . 1310
表 233. 预分频器的分频比 . . . 1311
表 234. 编码器计数方案 . . . 1318
表 235. 低功耗模式对 LPTIM 的影响 . . . 1319
表 236. 中断事件 . . . 1319
表 237. LPTIM 寄存器映射和复位值 . . . 1329
表 238. IWDG 寄存器映射和复位值 . . . 1339
表 239. WWDG 寄存器映射和复位值 . . . 1345
表 240. RTC 引脚 PC13 配置 . . . 1348
表 241. RTC_OUT 映射 . . . 1349
表 242. 各个模式下的 RTC 功能 . . . 1350
表 243. 低功耗模式对 RTC 的作用 . . . 1361
表 244. 中断控制位 . . . 1362
表 245. RTC 寄存器映射和复位值 . . . 1386
表 246. I2C 特性实现 . . . 1389
表 247. 模拟滤波器与数字滤波器对比 . . . 1393
表 248. I2C-SMBUS 规范数据建立和保持时间 . . . 1395
表 249. I2C 配置 . . . 1399
表 250. I2C-SMBUS 规范时钟时序 . . . 1409

表 251. fI2CCLK = 8 MHz 时的时序设置示例 1419
表 252. fI2CCLK = 16 MHz 时的时序设置示例 1420
表 253. fI2CCLK = 48 MHz 时的时序设置示例 1420
表 254. SMBus 超时规范 1422
表 255. 带 PEC 的 SMBUS 配置 1424
表 256. 不同 I2CCLK 频率下的 TIMEOUTA 设置示例（最大 $t_{TIMEOUT}$ = 25 ms） 1425
表 257. 不同 I2CCLK 频率下的 TIMEOUTB 设置示例 1425
表 258. 不同 I2CCLK 频率下的 TIMEOUTA 设置示例（最大 t_{IDLE} = 50 µs） 1425
表 259. 低功耗模式对 I2C 的影响 1436
表 260. I2C 中断请求 1436
表 261. I2C 寄存器映射和复位值 1452
表 262. USART/LPUART 功能 1455
表 263. 通过采样数据进行噪声检测 1470
表 264. BRR [3:0] = 0000 时的 USART 接收器容差 1474
表 265. BRR[3:0] 不等于 0000 时的 USART 接收器容差 1474
表 266. USART 帧格式 1478
表 267. USART 中断请求 1500
表 268. USART 寄存器映射和复位值 1527
表 269. lpuart_ker_ck_pres = 32,768 KHz 时编程的波特率的误差计算 1541
表 270. 采用 16 倍过采样 (OVER8 = 0) 时，在 fCK = 100 MHz 下 1541
表 271. LPUART 接收器的容差 1542
表 273. LPUART 中断请求 1554
表 274. LPUART 寄存器映射和复位值 1572
表 275. SPI 中断请求 1598
表 276. SPI 寄存器映射和复位值 1607
表 277. SAI 内部输入/输出信号 1611
表 278. SAI 输入/输出引脚 1611
表 279. 外部同步选择 1613
表 280. 时钟发生器编程示例 1621
表 281. TDM 设置 1627
表 282. 允许的 TDM 帧配置 1629
表 283. SOPD 模式 1632
表 284. 奇偶校验位计算 1633
表 285. 音频采样频率与符号率 1634
表 286. SAI 中断源 1642
表 287. SAI 寄存器映射和复位值 1659
表 288. SDMMC 工作模式 SD 和 SDIO 1663
表 289. SDMMC 工作模式 eMMC 1664
表 290. SDMMC 内部输入/输出信号 1665
表 291. SDMMC 引脚 1665
表 292. SDMMC 命令和数据相位选择 1666
表 293. 命令令牌格式 1671
表 294. 带 CRC 的短响应令牌格式 1672
表 295. 不带 CRC 的短响应令牌格式 1672
表 296. 带 CRC 的长响应令牌格式 1672
表 297. 特定命令概述 1673
表 298. 命令路径状态标志 1674
表 299. 命令路径错误处理 1674
表 300. 数据令牌格式 1681
表 301. 数据路径状态标志和清零位 1682

表 302. 数据路径错误处理 . . . 1683
表 303. 数据 FIFO 访问 . . . 1684
表 304. 传输 FIFO 状态标志 . . . 1685
表 305. 接收 FIFO 状态标志 . . . 1686
表 306. AHB 和 SDMMC_CK 时钟频率的关系 . . . 1688
表 307. SDIO 特殊操作控制 . . . 1689
表 308. 4 位模式启动、中断和 CRC 状态信号检测 . . . 1693
表 309. CMD12 用例 . . . 1697
表 310. 响应类型和 SDMMC_RESPxR 寄存器 . . . 1715
表 311. SDMMC 寄存器映射 . . . 1732
表 312. 发送邮箱映射 . . . 1749
表 313. 接收邮箱映射 . . . 1749
表 314. bxCAN 寄存器映射和复位值 . . . 1775
表 315. OTG_FS 支持的速度 . . . 1779
表 316. OTG 实现 . . . 1781
表 317. OTG_FS 输入/输出引脚 . . . 1782
表 318. OTG_FS 输入/输出信号 . . . 1783
表 319. STM32 低功耗模式与 OTG 的兼容性 . . . 1794
表 320. 模块全局控制和状态寄存器 (CSR) . . . 1802
表 321. 主机模式控制和状态寄存器 (CSR) . . . 1803
表 322. 设备模式控制和状态寄存器 . . . 1804
表 323. 数据 FIFO (DFIFO) 访问寄存器地址映射 . . . 1806
表 324. 电源和时钟门控控制和状态寄存器 . . . 1806
表 325. TRDT 值 (FS) . . . 1813
表 326. 软断开的最小时间 . . . 1850
表 327. OTG_FS 寄存器映射和复位值 . . . 1872
表 328. SWJ 调试端口引脚 . . . 1928
表 329. 灵活的 SWJ-DP 引脚分配 . . . 1928
表 330. JTAG 调试端口数据寄存器 . . . 1932
表 331. 32 位调试端口寄存器，通过移位值 A[3:2] 进行寻址 . . . 1933
表 332. 数据包请求（8 位） . . . 1934
表 333. ACK 响应（3 位） . . . 1934
表 334. DATA 传输（33 位） . . . 1934
表 335. SW-DP 寄存器 . . . 1936
表 336. Cortex®-M4 AHB-AP 寄存器 . . . 1937
表 337. 内核调试寄存器 . . . 1938
表 338. 主要的 ITM 寄存器 . . . 1941
表 339. 主要的 ETM 寄存器 . . . 1942
表 340. 异步 TRACE 引脚分配 . . . 1949
表 341. 同步 TRACE 引脚分配 . . . 1950
表 342. 灵活的 TRACE 引脚分配 . . . 1950
表 343. 重要的 TPIU 寄存器 . . . 1954
表 344. DBG 寄存器映射和复位值 . . . 1956
表 345. 文档版本历史 . . . 1960

图片索引

图 1. 系统架构 . . . 75
图 2. 存储器映射 . . . 78
图 3. 连续 16 位指令的执行（64 位读数据宽度） . . . 97
图 4. 更改读保护 (RDP) 级别 . . . 116
图 5. STM32L4Rxxx 和 STM32L4Sxxx 防火墙连接示意图 . . . 141
图 6. 防火墙功能状态 . . . 145
图 7. STM32L4S5xx/R5xx 和 STM32L4S7xx/L4R7xx 电源概述 . . . 153
图 8. STM32L4S9xx/L4R9xx 电源概述 . . . 154
图 9. 欠压复位波形 . . . 159
图 10. PVD 阈值 . . . 160
图 11. 低功耗模式可能的转换 . . . 163
图 12. 复位电路简化框图 . . . 202
图 13. 时钟树 . . . 205
图 14. DSI 时钟树 . . . 206
图 15. HSE/LSE 时钟源 . . . 207
图 16. TIM15 在捕获模式下的频率测量 . . . 214
图 17. TIM16 在捕获模式下的频率测量 . . . 215
图 18. TIM17 在捕获模式下的频率测量 . . . 215
图 19. CRS 框图 . . . 282
图 20. CRS 计数器行为 . . . 283
图 21. I/O 端口位的基本结构 . . . 293
图 22. 5 V 容限 I/O 端口位的基本结构 . . . 293
图 23. 输入浮空/上拉/下拉配置 . . . 297
图 24. 输出配置 . . . 298
图 25. 复用功能配置 . . . 299
图 26. 高阻态模拟配置 . . . 299
图 27. DMA 框图 . . . 333
图 28. DMAMUX 框图 . . . 354
图 29. DMAMUX 请求线复用器通道的同步模式 . . . 356
图 30. DMA 请求线复用器通道的事件生成 . . . 357
图 31. DMA2D 框图 . . . 367
图 32. Intel 8080 16 位模式 (RGB565) . . . 373
图 33. Intel 8080 18/24 位模式 (RGB888) . . . 374
图 34. Chrom-GRC™ 框图 . . . 400
图 35. 虚拟缓冲区 . . . 401
图 36. 虚拟缓冲区和物理缓冲区存储器映射 . . . 402
图 37. MMU 框图 . . . 403
图 38. 块验证/比较器实现 . . . 405
图 39. 可配置中断/事件框图 . . . 420
图 40. 外部中断/事件 GPIO 映射 . . . 422
图 41. CRC 计算单元框图 . . . 432
图 42. FSMC 框图 . . . 439
图 43. FSMC 存储区域 . . . 441
图 44. 模式 1 读取访问波形 . . . 449
图 45. 模式 1 写入访问波形 . . . 449
图 46. 模式 A 读取访问波形 . . . 451
图 47. 模式 A 读取访问波形 . . . 451
图 48. 模式 2 和模式 B 读取访问波形 . . . 453

图 49. 模式 2 读取访问波形 . . . 454
图 50. 模式 B 读取访问波形 . . . 454
图 51. 模式 C 读取访问波形 . . . 456
图 52. 模式 C 读取访问波形 . . . 457
图 53. 模式 D 读取访问波形 . . . 459
图 54. 模式 D 写入访问波形 . . . 460
图 55. 复用读取访问波形 . . . 462
图 56. 复用写入访问波形 . . . 463
图 57. 读取访问波形期间的异步等待 . . . 465
图 58. 写入访问波形期间的异步等待 . . . 466
图 59. 等待配置波形 . . . 468
图 60. 同步复用读取模式波形 - NOR、PSRAM (CRAM) . . . 469
图 61. 同步复用写入模式波形 - PSRAM (CRAM) . . . 471
图 62. NAND Flash 控制器的通用存储器访问波形 . . . 482
图 63. 访问非“CE 无关”NAND-Flash . . . 484
图 64. OCTOSPI 框图（八线通信） . . . 494
图 65. OCTOSPI 框图（四线模式） . . . 494
图 66. OCTOSPI 框图（使能双 Flash） . . . 495
图 67. 八线模式下的 SDR 读命令示例 . . . 496
图 68. 八线模式下使用 DQS 的 DTR 读（Macronix 模式）示例 . . . 498
图 69. 八线 SPI 模式下的 SDR 写命令示例 . . . 500
图 70. 八线模式下的 DTR 写（Macronix 模式）示例 . . . 500
图 71. HyperBus™ 读取操作示例 . . . 502
图 72. 具有初始延迟的 HyperBus™ 读取操作 . . . 503
图 73. 具有初始延迟的 HyperBus™ 写入操作 . . . 503
图 74. 具有额外延迟的 HyperBus™ 读取操作 . . . 504
图 75. 具有额外延迟的 HyperBus™ 写入操作 . . . 504
图 76. 无延迟的 HyperBus™ 写入操作 . . . 505
图 77. 具有越过页延迟的 HyperBus™ 读取操作 . . . 505
图 78. CKMODE = 0 时的 nCS（T = CLK 周期） . . . 515
图 79. SDR 模式下 CKMODE = 1 时的 nCS（T = CLK 周期） . . . 516
图 80. DDTR 模式下 CKMODE = 1 时的 nCS（T = CLK 周期） . . . 516
图 81. CKMODE = 1 且发生中止时的 nCS（T = CLK 周期） . . . 516
图 82. OctoSPI IO 管理器框图 . . . 547
图 83. ADC 框图 . . . 554
图 84. ADC 时钟方案 . . . 557
图 85. ADC1 连接功能 . . . 558
图 86. ADC 校准 . . . 560
图 87. 更新 ADC 校准系数 . . . 561
图 88. 混合单端通道和差分通道 . . . 562
图 89. 使能/禁止 ADC . . . 563
图 90. Bulb 模式时序图 . . . 565
图 91. 模数转换时间 . . . 568
图 92. 停止正在进行的常规转换 . . . 569
图 93. 停止正在进行的常规转换和注入转换 . . . 569
图 94. 注入转换延迟 . . . 573
图 95. JSQR 上下文队列示例（队列更改） . . . 576
图 96. JSQR 上下文队列示例（触发更改） . . . 576
图 97. 转换前发生溢出的 JSQR 上下文队列示例 . . . 577
图 98. 转换期间发生溢出的 JSQR 上下文队列示例 . . . 577
图 99. 队列为空时的 JSQR 上下文队列示例（JQM=0 的情况） . . . 578
图 100. 队列为空时的 JSQR 上下文队列示例（JQM=1 的情况） . . . 579

图 101. 通过将 JADSTP 置 1 的方式清空 JSQR 上下文队列 (JQM=0)。正在进行转换时发生 JADSTP 的情况。 579
图 102. 通过将 JADSTP 置 1 的方式清空 JSQR 上下文队列 (JQM=0)。正在进行转换时发生 JADSTP 并出现新触发的情况。 580
图 103. 通过将 JADSTP 置 1 的方式清空 JSQR 上下文队列 (JQM=0)。在正在进行转换的范围之外发生 JADSTP 的情况。 580
图 104. 通过将 JADSTP 置 1 的方式清空 JSQR 上下文队列 (JQM=1)。 581
图 105. 通过将 ADDIS 置 1 的方式清空 JSQR 上下文队列 (JQM=0)。 581
图 106. 通过将 ADDIS 置 1 的方式清空 JSQR 上下文队列 (JQM=1)。 582
图 107. 单次序列转换，软件触发 584
图 108. 连续序列转换，软件触发 584
图 109. 单次序列转换，硬件触发 585
图 110. 连续序列转换，硬件触发 585
图 111. 右对齐（偏移禁止，无符号值） 587
图 112. 右对齐（偏移使能，有符号值） 587
图 113. 左对齐（偏移禁止，无符号值） 588
图 114. 左对齐（偏移使能，有符号值） 588
图 115. 溢出示例 (OVR) 589
图 116. AUTODLY=1，连续模式下的常规转换，软件触发 592
图 117. AUTODLY=1，被注入转换中断的常规硬件转换（DISCEN=0；JDISCEN=0） 593
图 118. AUTODLY=1，被注入转换中断的常规硬件转换（DISCEN=1，JDISCEN=1） 594
图 119. AUTODLY=1，被注入转换中断的常规连续转换 595
图 120. AUTODLY=1，自动注入模式 (JAUTO=1) 595
图 121. 模拟看门狗的保护区域 596
图 122. ADCy_AWDx_OUT 信号生成（所有常规通道上） 598
图 123. ADCy_AWDx_OUT 信号生成（AWDx 标志未通过软件清零） 599
图 124. ADCy_AWDx_OUT 信号生成（单条常规通道上） 599
图 125. ADCy_AWDx_OUT 信号生成（所有注入通道上） 599
图 126. 20 位到 16 位结果的截断过程 600
图 127. 移 5 位并进行舍入的数值示例 600
图 128. 已触发的常规过采样模式（TROVS 位 = 1） 602
图 129. 常规过采样模式（4x 过采样率） 603
图 130. 同时使用常规和注入过采样模式 604
图 131. 已触发常规过采样支持注入 604
图 132. 在自动注入模式下进行过采样 605
图 133. 温度传感器通道框图 606
图 134. VBAT 通道框图 607
图 135. VREFINT 通道框图 607
图 136. 双通道 DAC 框图 643
图 137. DAC 单通道模式下的数据寄存器 645
图 138. DAC 双通道模式下的数据寄存器 645
图 139. 关闭触发 (TEN = 0) 时的转换时序图 TEN = 0 646
图 140. DAC LFSR 寄存器计算算法 649
图 141. LFSR 产生波形的 DAC 转换（使能软件触发） 649
图 142. 生成 DAC 三角波 650
图 143. 生成三角波波形的 DAC 转换（使能软件触发） 650
图 144. DAC 采样和保持模式阶段图 652
图 145. DCMI 框图 677
图 146. 顶级框图 677
图 147. DCMI 信号波形 678
图 148. 时序图 680
图 149. 快照模式下的帧捕获波形 682

图 150. 连续采集模式下的帧捕获波形 . . . 682
图 151. 裁剪后窗口的坐标和大小 . . . 683
图 152. 数据捕获波形 . . . 683
图 153. 像素光栅扫描顺序 . . . 684
图 154. 比较器框图 . . . 704
图 155. 窗口模式 . . . 706
图 156. 比较器迟滞 . . . 707
图 157. 比较器输出消隐 . . . 707
图 158. 独立模式：外部增益设置模式 . . . 716
图 159. 跟随器配置 . . . 717
图 160. PGA 模式，内部增益设置 (x2/x4/x8/x16)，未使用反相输入 . . . 718
图 161. PGA 模式，内部增益设置 (x2/x4/x8/x16)，使用反相输入进行滤波 . . . 719
图 162. 单个 DFSDM 框图 . . . 731
图 163. 输入通道引脚重定向 . . . 735
图 164. 通道收发器时序图 . . . 737
图 165. SPI 时钟缺失时序图 . . . 738
图 166. 曼彻斯特编码时钟缺失时序图 . . . 739
图 167. 曼彻斯特编码首次转换（曼彻斯特同步） . . . 741
图 168. DFSDM_CHyDATINR 寄存器操作模式和分配 . . . 745
图 169. 示例：Sinc3 滤波器响应 . . . 746
图 170. LTDC 框图 . . . 787
图 171. LCD-TFT 同步时序 . . . 790
图 172. 层窗口可编程参数 . . . 792
图 173. 两层与背景混合 . . . 795
图 174. 中断事件 . . . 796
图 175. DSI 主机框图 . . . 822
图 176. DSI 主机架构 . . . 823
图 177. 使用影子寄存器更新 LTDC 接口配置的流程 . . . 827
图 178. 立即更新步骤 . . . 827
图 179. 帧发送过程中的配置更新 . . . 828
图 180. 适配命令模式的使用流程 . . . 829
图 181. 24 bpp APB 像素到字节构成 . . . 833
图 182. 18 bpp APB 像素到字节构成 . . . 833
图 183. 16 bpp APB 像素到字节构成 . . . 834
图 184. 12 bpp APB 像素到字节构成 . . . 834
图 185. 8 bpp APB 像素到字节构成 . . . 834
图 186. 总线转向后的 PRESP_TO 时序 . . . 837
图 187. 读取请求（HS 或 LP）后的 PRESP_TO 时序 . . . 838
图 188. 写入请求（HS 或 LP）后的 PRESP_TO 时序 . . . 839
图 189. 准备模式为 1 时的影响 . . . 840
图 190. 图像区域内的命令发送周期 . . . 840
图 191. 在帧的最后一行发送命令 . . . 841
图 192. 使用同步脉冲的非突发模式下的 LPSIZE 计算 . . . 842
图 193. 使用同步事件的突发或非突发模式下的 LPSIZE 计算 . . . 842
图 194. 使用同步脉冲的非突发模式下的 VLPSIZE 计算 . . . 844
图 195. 使用同步事件的非突发模式下的 VLPSIZE 计算 . . . 844
图 196. 突发模式下的 VLPSIZE . . . 844
图 197. LPSIZE 和 VLPSIZE 在图像区域中的位置 . . . 846
图 198. HS 中的时钟通道和数据通道 . . . 848
图 199. HS 中的时钟通道和 LP 中的数据通道 . . . 848
图 200. LP 中的时钟通道和数据通道 . . . 848
图 201. 通过通用接口进行命令发送 . . . 849

图 202. 垂直彩条模式 . . . 851
图 203. 水平彩条模式 . . . 851
图 204. RGB888 BER 测试模式 . . . 852
图 205. 垂直模式 (103x15) . . . 853
图 206. 水平模式 (103x15) . . . 853
图 207. PLL 框图 . . . 857
图 208. 错误源 . . . 859
图 209. 视频数据包发送配置流程图 . . . 868
图 210. 发送测试模式的编程顺序 . . . 870
图 211. 帧配置寄存器 . . . 871
图 212. TSC 框图 . . . 936
图 213. 表面电荷转移模拟 I/O 组结构 . . . 937
图 214. 采样电容电压变化 . . . 938
图 215. 电荷转移采集序列 . . . 939
图 216. 扩频变化原理 . . . 940
图 217. RNG 框图 . . . 955
图 218. 熵源模型 . . . 956
图 219. RNG 初始化概述 . . . 958
图 220. AES 框图 . . . 966
图 221. ECB 加密和解密原理 . . . 968
图 222. CBC 加密和解密原理 . . . 969
图 223. CTR 加密和解密原理 . . . 970
图 224. GCM 加密和认证原理 . . . 971
图 225. GMAC 认证原理 . . . 971
图 226. CCM 加密和认证原理 . . . 972
图 227. STM32 加密库 AES 流程图示例 . . . 973
图 228. STM32 加密库 AES 流程图示例（续） . . . 974
图 229. 用于 ECB/CBC 解密的加密密钥生成（模式 2） . . . 977
图 230. 挂起模式管理示例 . . . 978
图 231. ECB 加密 . . . 978
图 232. ECB 解密 . . . 979
图 233. CBC 加密 . . . 979
图 234. CBC 解密 . . . 980
图 235. ECB/CBC 加密（模式 1） . . . 981
图 236. ECB/CBC 解密（模式 3） . . . 981
图 237. CTR 模式下的消息结构 . . . 983
图 238. CTR 加密 . . . 984
图 239. CTR 解密 . . . 984
图 240. GCM 中的消息结构 . . . 986
图 241. GCM 已验证加密 . . . 987
图 242. GMAC 模式下的消息结构 . . . 990
图 243. GMAC 认证模式 . . . 991
图 244. CCM 模式下的消息结构 . . . 992
图 245. CCM 模式认证加密 . . . 993
图 246. 根据数据交换构建 128 位块 . . . 998
图 247. 输入阶段 128 位数据块的 DMA 传输 . . . 1000
图 248. 输出阶段 128 位数据块的 DMA 传输 . . . 1001
图 249. AES 中断信号生成 . . . 1002
图 250. HASH 框图 . . . 1019
图 251. 消息数据交换功能 . . . 1021
图 252. HASH 保存/恢复机制 . . . 1026
图 253. 散列中断映射图 . . . 1028

图 254. 高级控制定时器框图 1041
图 255. 预分频器分频由 1 变为 2 时的计数器时序图. 1043
图 256. 预分频器分频由 1 变为 4 时的计数器时序图. 1043
图 257. 计数器时序图，1 分频内部时钟 1045
图 258. 计数器时序图，2 分频内部时钟 1045
图 259. 计数器时序图，4 分频内部时钟 1046
图 260. 计数器时序图，N 分频内部时钟 1046
图 261. 计数器时序图，ARPE=0 时更新事件（TIMx_ARR 未预装载） 1047
图 262. 计数器时序图，ARPE=1 时更新事件（TIMx_ARR 已预装载） 1047
图 263. 计数器时序图，1 分频内部时钟 1049
图 264. 计数器时序图，2 分频内部时钟 1049
图 265. 计数器时序图，4 分频内部时钟 1050
图 266. 计数器时序图，N 分频内部时钟 1050
图 267. 计数器时序图，未使用重复计数器时更新事件 1051
图 268. 计数器时序图，1 分频内部时钟，TIMx_ARR = 0x6 1052
图 269. 计数器时序图，2 分频内部时钟 1053
图 270. 计数器时序图，4 分频内部时钟，TIMx_ARR=0x36 1053
图 271. 计数器时序图，N 分频内部时钟 1054
图 272. 计数器时序图，ARPE=1 时的更新事件（计数器下溢） 1054
图 273. 计数器时序图，ARPE=1 时的更新事件（计数器上溢） 1055
图 274. 不同模式和 TIMx_RCR 寄存器设置下的更新频率示例 1056
图 275. 外部触发输入模块 1057
图 276. TIM1 ETR 输入电路 1057
图 277. TIM8 ETR 输入电路 1057
图 278. 正常模式下的控制电路，1 分频内部时钟 1058
图 279. TI2 外部时钟连接示例 1059
图 280. 外部时钟模式 1 下的控制电路. 1060
图 281. 外部触发输入模块 1060
图 282. 外部时钟模式 2 下的控制电路. 1061
图 283. 捕获/比较通道（示例：通道 1 输入阶段） 1062
图 284. 捕获/比较通道 1 主电路 1063
图 285. 捕获/比较通道的输出阶段（通道 1、通道 2 和通道 3） 1063
图 286. 捕获/比较通道的输出阶段（通道 4） 1064
图 287. 捕获/比较通道的输出阶段（通道 5 和通道 6） 1064
图 288. PWM 输入模式时序 1066
图 289. 输出比较模式，翻转 OC1 1068
图 290. 边沿对齐模式的 PWM 波形 (ARR=8) 1069
图 291. 中心对齐模式 PWM 波形 (ARR=8) 1070
图 292. 50% 占空比时产生的 2 个相移 PWM 信号 1071
图 293. 通道 1 和通道 3 上的组合 PWM 模式 1072
图 294. 三相组合 PWM 信号（每个周期多个触发脉冲） 1073
图 295. 带死区插入的互补输出 1074
图 296. 延迟时间大于负脉冲宽度的死区波形 1075
图 297. 延迟时间大于正脉冲宽度的死区波形 1075
图 298. 断路和断路 2 电路概述 1077
图 299. 响应 BRK 上的断路事件的不同输出行为 (OSSI = 1) 1079
图 300. BRK 和 BRK2 引脚使能后的 PWM 输出状态 (OSSI=1) 1080
图 301. BRK 使能后的 PWM 输出状态 (OSSI=0) 1081
图 302. 输出重定向（图中未显示 BRK2 请求） 1082
图 303. 清除 TIMx 的 OCxREF 1083
图 304. COM 事件生成 6 步 PWM 的示例 (OSSR=1) 1084
图 305. 单脉冲模式示例. 1085

图 306. 可再触发单脉冲模式 1087
图 307. 编码器接口模式下的计数器工作示例 1088
图 308. TI1FP1 极性反相时的编码器接口模式示例 1089
图 309. 测量 3 个信号上边沿之间的时间间隔 1090
图 310. 霍尔传感器接口的示例 1091
图 311. 复位模式下的控制电路 1092
图 312. 门控模式下的控制电路 1093
图 313. 触发模式下的控制电路 1094
图 314. 外部时钟模式 2 + 触发模式下的控制电路 1095
图 315. 通用定时器框图 1142
图 316. 预分频器分频由 1 变为 2 时的计数器时序图 1144
图 317. 预分频器分频由 1 变为 4 时的计数器时序图 1144
图 318. 计数器时序图，1 分频内部时钟 1145
图 319. 计数器时序图，2 分频内部时钟 1146
图 320. 计数器时序图，4 分频内部时钟 1146
图 321. 计数器时序图，N 分频内部时钟 1147
图 322. 计数器时序图，ARPE=0 时更新事件（TIMx_ARR 未预装载） 1147
图 323. 计数器时序图，ARPE=1 时更新事件（TIMx_ARR 已预装载） 1148
图 324. 计数器时序图，1 分频内部时钟 1149
图 325. 计数器时序图，2 分频内部时钟 1149
图 326. 计数器时序图，4 分频内部时钟 1150
图 327. 计数器时序图，N 分频内部时钟 1150
图 328. 计数器时序图，不使用重复计数器时更新事件 1151
图 329. 计数器时序图，1 分频内部时钟，TIMx_ARR=0x6 1152
图 330. 计数器时序图，2 分频内部时钟 1153
图 331. 计数器时序图，4 分频内部时钟，TIMx_ARR=0x36 1153
图 332. 计数器时序图，N 分频内部时钟 1154
图 333. 计数器时序图，ARPE=1 时的更新事件（计数器下溢） 1154
图 334. 计数器时序图，ARPE=1 时的更新事件（计数器上溢） 1155
图 335. 正常模式下的控制电路，1 分频内部时钟 1156
图 336. TI2 外部时钟连接示例 1156
图 337. 外部时钟模式 1 下的控制电路 1157
图 338. 外部触发输入模块 1158
图 339. 外部时钟模式 2 下的控制电路 1159
图 340. 捕获/比较通道（示例：通道 1 输入阶段） 1160
图 341. 捕获/比较通道 1 主电路 1160
图 342. 捕获/比较通道的输出阶段（通道 1） 1161
图 343. PWM 输入模式时序 1163
图 344. 输出比较模式，翻转 OC1 1164
图 345. 边沿对齐模式的 PWM 波形 (ARR=8) 1166
图 346. 中心对齐模式 PWM 波形 (ARR=8) 1167
图 347. 50% 占空比时产生的 2 个相移 PWM 信号 1168
图 348. 通道 1 和通道 3 上的组合 PWM 模式 1169
图 349. 清除 TIMx 的 OCxREF 1170
图 350. 单脉冲模式示例 1171
图 351. 可再触发单脉冲模式 1172
图 352. 编码器接口模式下的计数器工作示例 1174
图 353. TI1FP1 极性反相时的编码器接口模式示例 1174
图 354. 复位模式下的控制电路 1176
图 355. 门控模式下的控制电路 1177
图 356. 触发模式下的控制电路 1177
图 357. 外部时钟模式 2 + 触发模式下的控制电路 1178

图 358. 主/从定时器示例 . . . 1179
图 359. 使用 TIM3 的 OC1REF 对 TIM2 实施门控控制 . . . 1180
图 360. 使用 TIM3 的使能信号对 TIM2 实施门控控制 . . . 1181
图 361. 使用 TIM3 的更新事件触发 TIM2 . . . 1181
图 362. 使用 TIM3 的使能信号触发 TIM2 . . . 1182
图 363. 使用 TIM3 的 TI1 输入触发 TIM3 和 TIM2 . . . 1183
图 364. TIM15 框图 . . . 1213
图 365. TIM16/TIM17 框图 . . . 1214
图 366. 预分频器分频由 1 变为 2 时的计数器时序图 . . . 1216
图 367. 预分频器分频由 1 变为 4 时的计数器时序图 . . . 1216
图 368. 计数器时序图，1 分频内部时钟 . . . 1218
图 369. 计数器时序图，2 分频内部时钟 . . . 1218
图 370. 计数器时序图，4 分频内部时钟 . . . 1219
图 371. 计数器时序图，N 分频内部时钟 . . . 1219
图 372. 计数器时序图，ARPE=0 时更新事件（TIMx_ARR 未预装载） . . . 1220
图 373. 计数器时序图，ARPE=1 时更新事件（TIMx_ARR 未预装载） . . . 1220
图 374. 不同模式和 TIMx_RCR 寄存器设置下的更新频率示例 . . . 1221
图 375. 正常模式下的控制电路，1 分频内部时钟 . . . 1222
图 376. TI2 外部时钟连接示例 . . . 1223
图 377. 外部时钟模式 1 下的控制电路 . . . 1224
图 378. 捕获/比较通道（示例：通道 1 输入阶段） . . . 1225
图 379. 捕获/比较通道 1 主电路 . . . 1225
图 380. 捕获/比较通道的输出阶段（通道 1） . . . 1226
图 381. 捕获/比较通道的输出阶段（TIM15 的通道 2） . . . 1226
图 382. PWM 输入模式时序 . . . 1228
图 383. 输出比较模式，翻转 OC1 . . . 1230
图 384. 边沿对齐模式的 PWM 波形 (ARR=8) . . . 1231
图 385. 通道 1 和通道 2 上的组合 PWM 模式 . . . 1232
图 386. 带死区插入的互补输出 . . . 1233
图 387. 延迟时间大于负脉冲宽度的死区波形 . . . 1234
图 388. 延迟时间大于正脉冲宽度的死区波形 . . . 1234
图 389. 断路电路概述 . . . 1236
图 390. 输出的断路响应行为 . . . 1238
图 391. 输出重定向 . . . 1240
图 392. 单脉冲模式示例 . . . 1241
图 393. 可再触发单脉冲模式 . . . 1243
图 394. 测量 2 个信号上边沿之间的时间间隔 . . . 1243
图 395. 复位模式下的控制电路 . . . 1244
图 396. 门控模式下的控制电路 . . . 1245
图 397. 触发模式下的控制电路 . . . 1246
图 398. 基本定时器框图 . . . 1295
图 399. 预分频器分频由 1 变为 2 时的计数器时序图 . . . 1297
图 400. 预分频器分频由 1 变为 4 时的计数器时序图 . . . 1297
图 401. 计数器时序图，1 分频内部时钟 . . . 1298
图 402. 计数器时序图，2 分频内部时钟 . . . 1299
图 403. 计数器时序图，4 分频内部时钟 . . . 1299
图 404. 计数器时序图，N 分频内部时钟 . . . 1300
图 405. 计数器时序图，ARPE = 0 时更新事件（TIMx_ARR 未预装载） . . . 1300
图 406. 计数器时序图，ARPE=1 时更新事件（TIMx_ARR 预装载） . . . 1301
图 407. 正常模式下的控制电路，1 分频内部时钟 . . . 1302
图 408. 低功耗定时器框图 . . . 1309
图 409. 干扰滤波器时序图 . . . 1311

图 410. LPTIM 输出波形，单次计数模式配置 1312
图 411. LPTIM 输出波形，单次计数模式配置且激活置 1 一次模式（WAVE 位置 1） 1313
图 412. LPTIM 输出波形、连续计数模式配置 1313
图 413. 生成波形 1315
图 414. 编码器模式计数序列 1318
图 415. IRTIM 与 TIM16 和 TIM17 的内部硬件连接 1330
图 416. 独立看门狗框图 1331
图 417. 看门狗框图 1341
图 418. 窗口看门狗时序图 1342
图 419. RTC 框图 1347
图 420. I2C 框图 1390
图 421. I2C 总线协议 1392
图 422. 建立和保持时序 1393
图 423. I2C 初始化流程图 1396
图 424. 数据接收 1397
图 425. 数据发送 1398
图 426. 从器件初始化流程图 1401
图 427. I2C 从发送器的传输序列流程图 (NOSTRETCH=0) 1402
图 428. I2C 从发送器的传输序列流程图 (NOSTRETCH=1) 1403
图 429. I2C 从发送器的传输总线图 1404
图 430. 从接收器的传输序列流程图 (NOSTRETCH=0) 1405
图 431. 从接收器的传输序列流程图 (NOSTRETCH=1) 1406
图 432. I2C 从接收器的传输总线图 1406
图 433. 主时钟生成 1408
图 434. 主模式初始化流程图 1410
图 435. 10 位地址读访问 (HEAD10R=0) 1410
图 436. 10 位地址读访问 (HEAD10R=1) 1410
图 437. I2C 主发送器的传输序列流程图（N≤255 字节） 1412
图 438. I2C 主发送器的传输序列流程图（N>255 字节） 1413
图 439. I2C 主发送器的传输总线图 1414
图 440. I2C 主接收器的传输序列流程图（N≤255 字节） 1416
图 441. I2C 主接收器的传输序列流程图（N>255 字节） 1417
图 442. I2C 主接收器的传输总线图 1418
图 443. $t_{LOW:SEXT}$ 和 $t_{LOW:MEXT}$ 的超时间隔 1423
图 444. SMBus 从发送器的传输序列流程图（N 字节 + PEC） 1426
图 445. SMBus 从发送器的传输总线图 (SBC=1) 1427
图 446. SMBus 从接收器的传输序列流程图（N 字节 + PEC） 1428
图 447. SMBus 从接收器的总线传输图 (SBC=1) 1429
图 448. SMBus 主发送器的总线传输图 1430
图 449. SMBus 主接收器的总线传输图 1432
图 450. I2C 中断映射图 1437
图 451. USART 框图 1456
图 452. 字长编程 1459
图 453. 可配置的停止位 1461
图 454. 发送时的 TC/TXE 行为 1464
图 455. 16 倍或 8 倍过采样时的起始位检测 1465
图 456. usart_ker_ck 时钟分频器框图 1468
图 457. 16 倍过采样时的数据采样 1470
图 458. 8 倍过采样时的数据采样 1470
图 459. 使用空闲线路检测时的静默模式 1476
图 460. 使用地址标记检测时的静默模式 1477
图 461. LIN 模式下的中断检测（11 位中断长度——LBDL 位置 1） 1480

图 462. LIN 模式下的中断检测与帧错误检测 1481
图 463. USART 同步主发送示例 1482
图 464. USART 在同步主模式下的数据时钟时序图（M 位 = 00） 1482
图 465. USART 在同步主模式下的数据时钟时序图（M 位 =“01”） 1483
图 466. USART 在同步从模式下的数据时钟时序图（M 位 = 00） 1484
图 467. ISO 7816-3 异步协议 1486
图 468. 使用 1.5 个停止位检测奇偶校验错误 1487
图 469. IrDA SIR ENDEC 框图 1491
图 470. IrDA 数据调制 (3/16)——正常模式 1491
图 471. 使用 DMA 进行发送 1493
图 472. 使用 DMA 进行接收 1494
图 473. 2 个 USART 间的硬件流控制 1494
图 474. RS232 RTS 流控制 1495
图 475. RS232 CTS 流控制 1496
图 476. 唤醒事件通过验证（唤醒事件 = 地址匹配，禁止 FIFO） 1498
图 477. 唤醒事件未通过验证（唤醒事件 = 地址匹配，禁止 FIFO） 1499
图 478. LPUART 框图 1531
图 479. LPUART 字长编程 1533
图 480. 可配置的停止位 1535
图 481. 发送时的 TC/TXE 行为 1536
图 482. lpuart_ker_ck 时钟分频器框图 1539
图 483. 使用空闲线路检测时的静默模式 1543
图 484. 使用地址标记检测时的静默模式 1544
图 485. 使用 DMA 进行发送 1547
图 486. 使用 DMA 进行接收 1548
图 487. 2 个 LPUART 间的硬件流控制 1549
图 488. RS232 RTS 流控制 1549
图 489. RS232 CTS 流控制 1550
图 490. 唤醒事件通过验证（唤醒事件 = 地址匹配，禁止 FIFO） 1552
图 491. 唤醒事件未通过验证（唤醒事件 = 地址匹配，禁止 FIFO） 1553
图 492. SPI 框图 1575
图 493. 全双工单个主器件/单个从器件应用 1576
图 494. 半双工单个主器件/单个从器件应用 1577
图 495. 单工单个主器件/单个从器件应用（主器件为只发送模式/从器件为只接收模式） 1578
图 496. 主器件和三个独立的从器件 1579
图 497. 多主应用 1580
图 498. 硬件/软件从器件选择管理 1581
图 499. 数据时钟时序图 1582
图 500. 数据长度不等于 8 位或 16 位时的数据对齐 1583
图 501. 发送和接收 FIFO 中的数据封装 1586
图 502. 主器件全双工通信 1589
图 503. 从器件全双工通信 1590
图 504. 带有 CRC 的主器件全双工通信 1591
图 505. 封装模式下的主器件全双工通信 1592
图 506. Motorola SPI 主模式下的 NSSP 脉冲生成 1595
图 507. TI 模式传输 1596
图 508. SAI 功能框图 1610
图 509. 音频帧 1614
图 510. FS 的作用是 SOF 信号 + 通道识别信号 (FSDEF = TRIS = 1) 1616
图 511. FS 的作用是 SOF 信号 (FSDEF = 0) 1617
图 512. SAI_xSLOTR 中的 FBOFF = 0 时的 Slot 大小配置 1618
图 513. 第一位偏移 1618

图 514. 音频模块时钟发生器概览 1619
图 515. PDM 典型连接和时序 1623
图 516. 详细的 PDM 接口模块框图 1624
图 517. 启动序列 1625
图 518. TDM 中的 SAI_ADR 格式（32 位 Slot 宽度） 1626
图 519. TDM 中的 SAI_ADR 格式（16 位 Slot 宽度） 1626
图 520. TDM 中的 SAI_ADR 格式（8 位 Slot 宽度） 1627
图 521. AC’97 音频帧 1630
图 522. 至少具有 2 个嵌入式 SAI 的器件的典型 AC’97 配置示例（三个外部 AC’97 解码器） 1631
图 523. SPDIF 格式 1632
图 524. SAI_xDR 寄存器定序 1633
图 525. SAI 的音频模块中的数据压扩硬件 1636
图 526. 发送无效 Slot 时 SD 输出线上的三态策略 1637
图 527. 采用 I2S 等协议时输出数据线上的三态策略 1638
图 528. 上溢错误检测 1639
图 529. FIFO 下溢事件 1639
图 530. SDMMC“无响应”和“无数据”操作 1662
图 531. SDMMC（多个）块读取操作 1662
图 532. SDMMC（多个）块写入操作 1662
图 533. SDMMC（连续）流读取操作 1663
图 534. SDMMC（连续）流写入操作 1663
图 535. SDMMC 框图 1664
图 536. SDMMC 命令和数据相位关系 1666
图 537. 控制单元 1667
图 538. 命令/响应路径 1668
图 539. 命令路径状态机 (CPSM) 1669
图 540. 数据路径 1675
图 541. DDR 模式数据包时钟驱动 1676
图 542. DDR 模式 CRC 状态/启动确认时钟驱动 1676
图 543. 数据路径状态机 (DPSM) 1677
图 544. CLKMUX 单元 1686
图 545. 异步中断生成 1690
图 546. 同步中断周期数据读取 1690
图 547. 同步中断周期数据写入 1691
图 548. 异步中断周期数据读取 1692
图 549. 异步中断周期数据写入 1692
图 550. 在 DS、HS、SDR12 和 SDR25 模式下通过 SDMMC_CK 停止时钟 1695
图 551. 在 DDR50 和 SDR50 模式下通过 SDMMC_CK 停止时钟 1695
图 552. SDMMC_CK < 50 MHz 时的读取等待 1696
图 553. SDMMC_CK > 50 MHz 时的读取等待 1696
图 554. CMD12 流时序 1698
图 555. CMD5 睡眠唤醒过程 1700
图 556. 正常启动模式操作 1702
图 557. 备用启动模式操作 1703
图 558. 命令响应 R1b 繁忙信号 1704
图 559. SDMMC 状态控制 1705
图 560. 卡掉电再上电/上电图 1706
图 561. CMD11 信号电压切换序列 1707
图 562. 电压切换收发器的典型应用 1708
图 563. CAN 网络拓扑结构 1736
图 564. Dual-CAN 框图 1737
图 565. bxCAN 工作模式 1739

图 566. 静默模式下的 bxCAN 1740
图 567. 环回模式下的 bxCAN 1740
图 568. 组合模式下的 bxCAN 1741
图 569. 发送邮箱状态 1743
图 570. 接收 FIFO 状态 1744
图 571. 过滤器组尺度配置 - 寄存器构成 1746
图 572. 过滤器编号示例 1747
图 573. 筛选机制 - 示例 1748
图 574. CAN 错误状态图 1749
图 575. 位时序 1751
图 576. CAN 帧 1752
图 577. 事件标志与中断产生 1753
图 578. CAN 邮箱寄存器 1764
图 579. 全速 OTG 模块框图 1782
图 580. OTG_FS A-B 设备连接 1784
图 581. USB_FS 仅做设备的连接 1785
图 582. USB_FS 仅做主机的连接 1789
图 583. SOF 连接（SOF 触发输出与 TIM 和 ITR1 的连接） 1793
图 584. 动态更新 OTG_HFIR (RLDCTRL = 0) 1795
图 585. 设备模式下的 FIFO 地址映射和 AHB FIFO 访问映射 1796
图 586. 主机模式下的 FIFO 地址映射和 AHB FIFO 访问映射 1797
图 587. 中断层级 1801
图 588. 发送 FIFO 写任务 1883
图 589. 接收 FIFO 读任务 1884
图 590. 正常批量/控制 OUT/SETUP 1885
图 591. 批量/控制 IN 事务 1889
图 592. 正常中断 OUT 1891
图 593. 正常中断 IN 1895
图 594. 同步 OUT 事务 1897
图 595. 同步 IN 事务 1900
图 596. 接收 FIFO 数据包读取 1904
图 597. 处理 SETUP 数据包 1906
图 598. 批量 OUT 事务 1911
图 599. TRDT 最大时序情况 1919
图 600. A 设备 SRP 1920
图 601. B 设备 SRP 1921
图 602. A 设备 HNP 1922
图 603. B 设备 HNP 1923
图 604. STM32 MCU 和 Cortex®-M4 级调试支持框图 1925
图 605. SWJ 调试端口 1927
图 606. JTAG TAP 连接 1930
图 607. TPIU 框图 1949

1 文档约定

1.1 寄存器相关缩写词列表

寄存器说明中使用以下缩写词[a]：

读/写 (rw)	软件可以读写该位。
只读 (r)	软件只能读取该位。
只写 (w)	软件只能写入该位。读取该位时将返回复位值。
读取/写入 0 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/写入 1 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/写入清零 (rc_w)	软件可以读取该位，也可以通过写入寄存器将该位清零。写入该位的值并不重要。
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。写入该位对其值无影响。
读取/读取置位 (rs_r)	软件可以读取该位。读取该位时，将自动置 1。写入该位对其值无影响。
读取/置位 (rs)	软件可以读取该位，也可将其置 1。写入 0 对该位的值无影响。
读/仅可写入一次 (rwo)	软件仅可写入一次该位，但可随时读取该位。只能通过复位将该位返回到复位值。
切换 (t)	软件可以通过写入 1 来切换该位。写入 0 无影响。
只读，写触发 (rt_w1)	软件可以读取该位。写入 1 时，将触发事件，但不会影响该位的值。
保留 (Res.)	保留位，必须保持复位值。

a. 这是一个涵盖适用于 STM 微控制器的所有缩写词的详尽列表，其中一些缩写词在当前文档中可能并未使用。

1.2 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：

- **字**：32 位数据。
- **半字**：16 位数据。
- **字节**：8 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。
- **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。
- **APB**：高级外设总线。

1.3 外设可用性

有关各型号产品的外设数量意义可用性信息，请参见相关器件数据手册。

2 系统和存储器概述

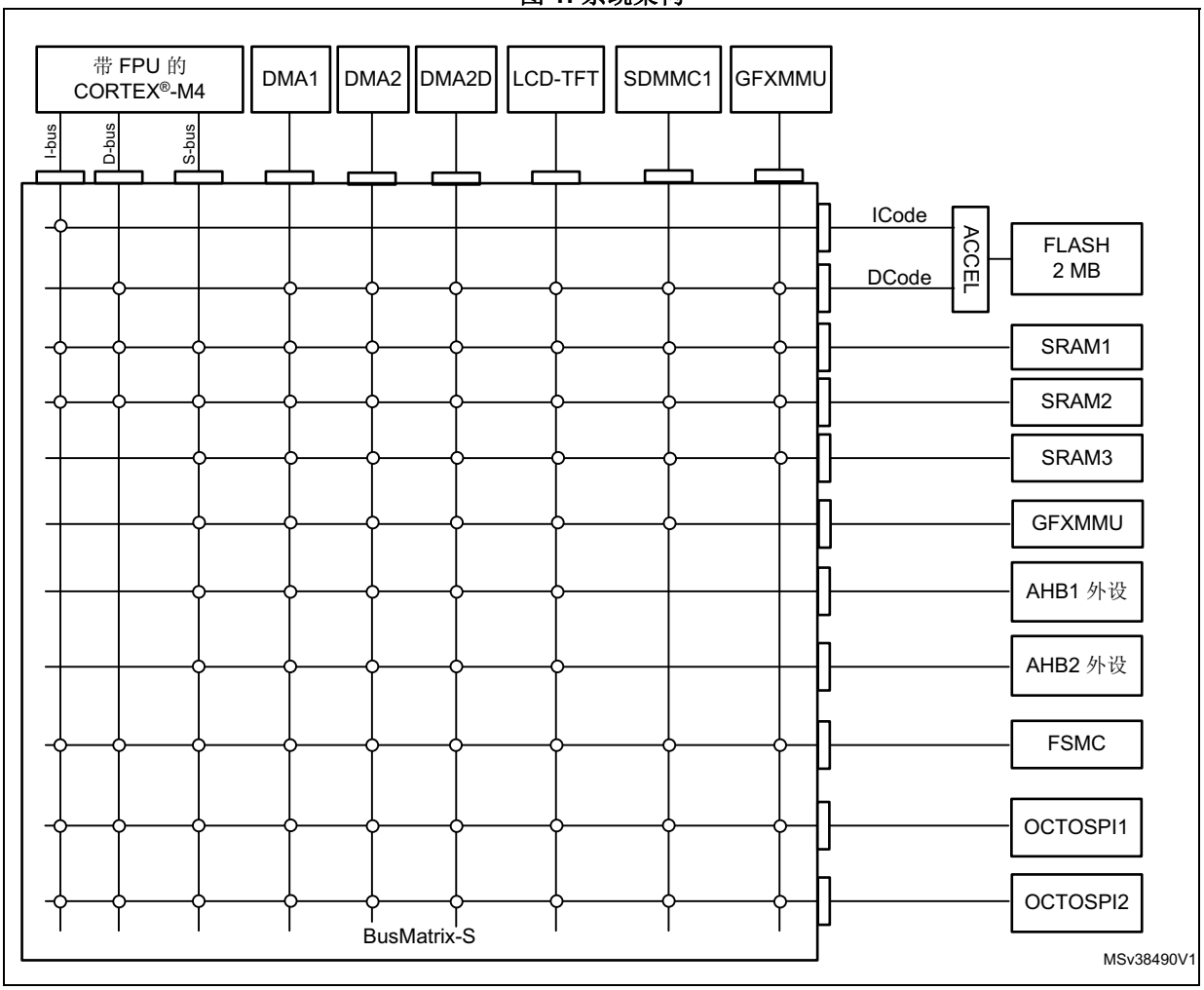
2.1 系统架构

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：

- 最多九条主控总线：
  - Cortex®-M4（带 FPU 内核 I 总线）
  - Cortex®-M4（带 FPU 内核 D 总线）
  - Cortex®-M4（带 FPU 内核 S 总线）
  - DMA1
  - DMA2
  - DMA2D（Chrom-Art 加速器™）存储器总线
  - LCD-TFT 控制器 DMA 总线
  - SDMMC1 总线
  - GFXMMU (Chrom-GRC™) 总线
- 最多十一条被控总线：
  - ICode 总线上的内部 Flash
  - DCode 总线上的内部 Flash
  - 内部 SRAM1 (192 KB)
  - 内部 SRAM2 (64 KB)
  - 内部 SRAM3 (384 KB)
  - GFXMMU (Chrom-GRC™)
  - AHB1 外设（包括 AHB-APB 总线桥和 APB 外设（连接到 APB1 和 APB2））
  - AHB2 外设
  - 灵活存储控制器 (FMC)
  - OCTOSPI1
  - OCTOSPI2

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 1* 所示：

图 1. 系统架构



2.1.1 I 总线

此总线用于将 Cortex®-M4 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、内部 SRAM 或外部存储器（通过外设接口 FMC 或 OCTOSPI）。

2.1.2 D 总线

此总线用于将 Cortex®-M4 内核的数据总线连接到总线矩阵。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的存储器：内部 Flash、内部 SRAM 或外部存储器（通过外设接口 FMC 或 OCTOSPI）。

2.1.3 S 总线

此总线用于将 Cortex®-M4 内核的系统总线连接到总线矩阵。内核使用此总线访问位于外设或 SRAM 区域中的数据。此总线访问的对象是内部 SRAM、包括 APB1 和 APB2 外设在内的 AHB1 外设、AHB2 外设以及外部存储器（通过外设接口 OCTOSPI 或 FMC）。

此总线也可访问 SRAM2，以实现 SRAM1 和 SRAM3 的连续映射。

2.1.4 DMA 总线

此总线用于将 DMA 的 AHB 主接口连接到总线矩阵。此总线访问的对象是 SRAM1、SRAM2 和 SRAM3，包括 APB1 和 APB2 外设在内的 AHB1 外设，AHB2 外设以及外部存储器（通过外设接口 OCTOSPI 或 FMC）。

2.1.5 DMA2D 总线

此总线用于将 DMA2D 的 AHB 主接口连接到总线矩阵。此总线访问的对象是 SRAM1、SRAM2 和 SRAM3 以及外部存储器（通过外设接口 OCTOSPI 或 FMC）。

2.1.6 LCD-TFT 控制器 DMA 总线

此总线用于将 LCD 控制器 DMA 主接口连接到总线矩阵。此总线访问的对象是数据存储器：内部 SRAM（SRAM1、SRAM2、SRAM3）、内部 Flash 或外部存储器（通过外设接口 FMC 或 OCTOSPI）。

2.1.7 SDMMC 控制器 DMA 总线

此总线用于将 SDMMC1 DMA 主接口连接到总线矩阵。仅 SDMMC1 DMA 通过此总线向存储器存取数据。此总线访问的对象是数据存储器：内部 SRAM（SRAM1、SRAM2、SRAM3）、内部 Flash 或外部存储器（通过外设接口 FMC 或 OCTOSPI）。

2.1.8 GFXMMU 总线

此总线用于将 GFXMMU (Chrom-GRC™) AHB 主接口连接到总线矩阵。此总线访问的对象是数据存储器：内部 SRAM（SRAM1、SRAM2、SRAM3）、内部 Flash 或外部存储器（通过外设接口 FMC 或 OCTOSPI）。

2.1.9 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。总线矩阵由最多九条主控总线（CPU AHB、系统总线、DCode 总线、ICode 总线、DMA1、DMA2、DMA2D、SDMMC1、LCD-TFT 和 GFXMMU）和最多十一条被控总线（FLASH、SRAM1、SRAM2、SRAM3、AHB1（包括 APB1 和 APB2）、AHB2、OCTOSTPI1、OCTOSPI2 和 FMC）构成。

AHB/APB 总线桥

借助两个 AHB/APB 总线桥，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。

有关连接到此总线桥的外设的地址映射，请参见*第 78 页的第 2.2.2 节：存储器映射和寄存器边界地址*。

每次芯片复位后，所有外设时钟都被关闭（SRAM1/2 和 Flash 接口除外）。使用外设前，必须在 RCC_AHBxENR 和 RCC_APBxENR 寄存器中使能其时钟。

注： *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。*

2.2 存储器组织结构

2.2.1 简介

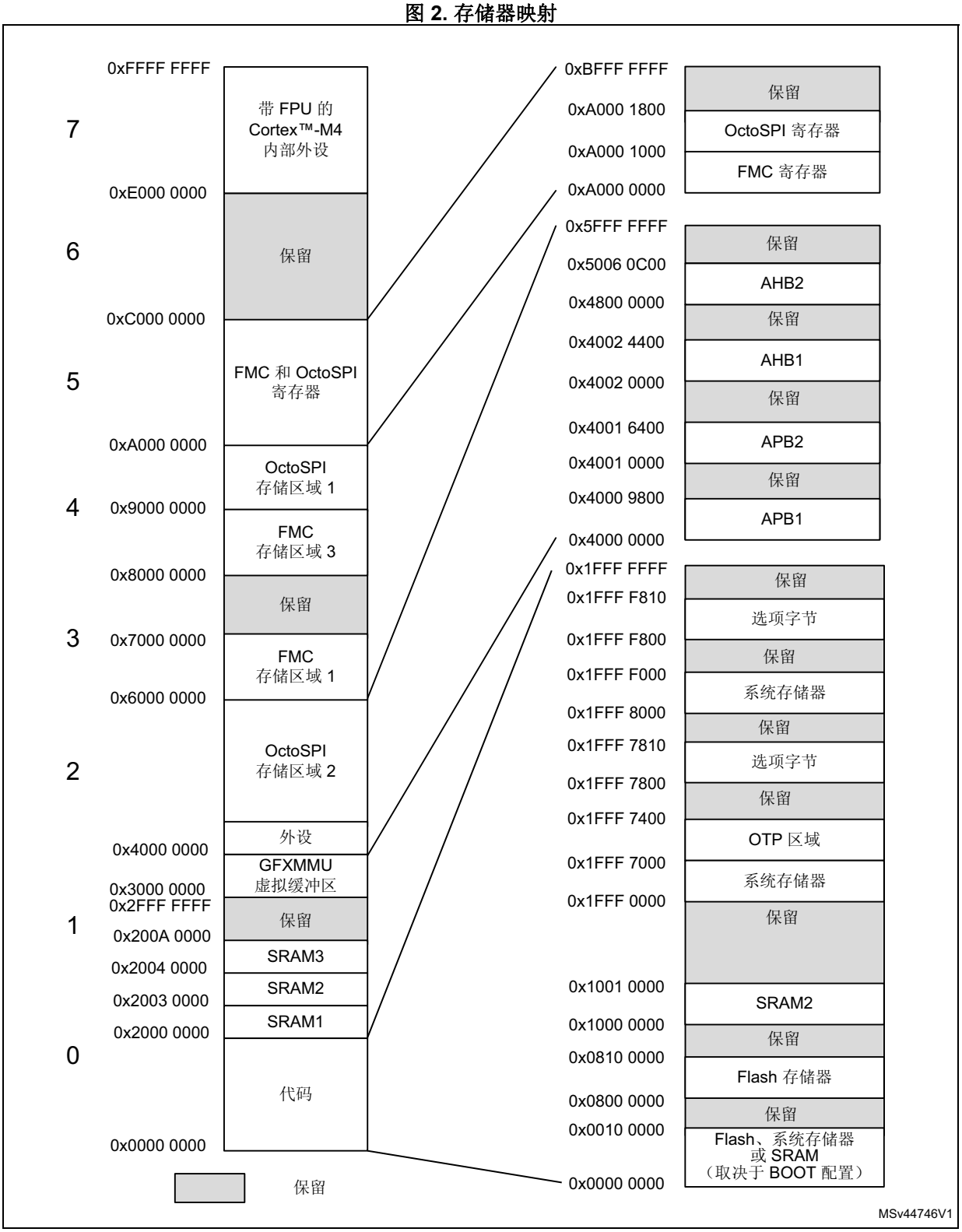
程序存储器、数据存储器、寄存器和 I/O 端口排列在同一个线性（即地址连续）的 4 GB 地址空间内。

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。

可寻址的存储空间分为 8 个主要块，每个块为 512 MB。

2.2.2 存储器映射和寄存器边界地址

图 2. 存储器映射



未分配给片上存储器和外设的所有存储映射区域均视为“保留区”。有关可用存储器和寄存器区域的详细映射，请参见下表。

下表给出了器件中可用外设的边界地址。

表 1. STM32L4Rxxx 和 STM32L4Sxxx 存储器映射和外设寄存器边界地址

总线	边界地址	大小（字节）	外设	外设寄存器映射
-	0xA000 1800 - 0xDFFF FFFF	1 KB	保留	-
	0xA000 1400 - 0xA000 17FF	1 KB	OCTOSPI2 寄存器	*第 19.6.28 节：OCTOSPI 寄存器映射*
	0xA000 1000 - 0xA000 13FF	1 KB	OCTOSPI1 寄存器	*第 19.6.28 节：OCTOSPI 寄存器映射*
	0xA000 0400 - 0xA000 0FFF	1 KB	保留	-
	0xA000 0000 - 0xA000 03FF	1 KB	FSMC 寄存器	*第 18.7 节：FSMC 寄存器映射*

表 1. STM32L4Rxxx 和 STM32L4Sxxx 存储器映射和外设寄存器边界地址（续）

总线	边界地址	大小（字节）	外设	外设寄存器映射
AHB2	0x5006 2000 - 0x5FFF FFFF	~260 MB	保留	-
	0x5006 2400 - 0x5006 27FF	1 KB	SDMMC1	*第 48.8.20 节：SDMMC 寄存器映射*
	0x5006 2000 - 0x5006 23FF	1 KB	保留	-
	0x5006 1C00 - 0x5006 1FFF	1 KB	OCTOSPIM	*第 20.4.2 节：OctoSPI IO 管理器寄存器映射*
	0x5006 0C00 - 0x5006 1BFF	4 KB	保留	-
	0x5006 0800 - 0x5006 0BFF	1 KB	RNG	*第 31.8.4 节：RNG 寄存器映射*
	0x5006 0400 - 0x5006 07FF	1 KB	HASH	*第 33.6.8 节：HASH 寄存器映射*
	0x5006 0000 - 0x5006 03FF	1 KB	AES	*第 32.7.18 节：AES 寄存器映射*
	0x5005 0800 - 0x5005 FFFF	61 KB	保留	-
	0x5005 0400 - 0x5005 07FF	1 KB	保留	-
	0x5005 0000 - 0x5005 03FF	1 KB	DCMI	*第 23.7.12 节：DCMI 寄存器映射*
	0x5004 0400 - 0x5004 FFFF	62 KB	保留	-
	0x5004 0000 - 0x5004 03FF	1 KB	ADC	*第 637 页的第 21.7.3 节：ADC 寄存器映射*
	0x5000 0000 - 0x5003 FFFF	16 KB	OTG_FS	*第 50.15.54 节：OTG_FS 寄存器映射*
	0x4800 2400 - 0x4FFF FFFF	~127 MB	保留	-
	0x4800 2000 - 0x4800 23FF	1 KB	GPIOI	*第 8.4.12 节：GPIO 寄存器映射*
	0x4800 1C00 - 0x4800 1FFF	1 KB	GPIOH	*第 8.4.12 节：GPIO 寄存器映射*
	0x4800 1800 - 0x4800 1BFF	1 KB	GPIOG	*第 8.4.12 节：GPIO 寄存器映射*
	0x4800 1400 - 0x4800 17FF	1 KB	GPIOF	*第 8.4.12 节：GPIO 寄存器映射*
	0x4800 1000 - 0x4800 13FF	1 KB	GPIOE	*第 8.4.12 节：GPIO 寄存器映射*
	0x4800 0C00 - 0x4800 0FFF	1 KB	GPIOD	*第 8.4.12 节：GPIO 寄存器映射*
	0x4800 0800 - 0x4800 0BFF	1 KB	GPIOC	*第 8.4.12 节：GPIO 寄存器映射*
	0x4800 0400 - 0x4800 07FF	1 KB	GPIOB	*第 8.4.12 节：GPIO 寄存器映射*
	0x4800 0000 - 0x4800 03FF	1 KB	GPIOA	*第 8.4.12 节：GPIO 寄存器映射*

表 1. STM32L4Rxxx 和 STM32L4Sxxx 存储器映射和外设寄存器边界地址（续）

总线	边界地址	大小（字节）	外设	外设寄存器映射
AHB1	0x4002 F000 - 0x47FF FFFF	~127 MB	保留	-
	0x4002 C000 - 0x4002 EFFF	1KB	GFXMMU	*第 14.5.11 节：图形 MMU 寄存器映射*
	0x4002 BC00 - 0x4002 BBFF	1 KB	保留	-
	0x4002 B000 - 0x4002 BBFF	3 KB	DMA2D	*第 13.5.21 节：DMA2D 寄存器映射*
	0x4002 4400 - 0x4002 AFFF	26 KB	保留	-
	0x4002 4000 - 0x4002 43FF	1 KB	TSC	*第 30.6.11 节：TSC 寄存器映射*
	0x4002 3400 - 0x4002 3FFF	1 KB	保留	-
	0x4002 3000 - 0x4002 33FF	1 KB	CRC	*第 17.4.6 节：CRC 寄存器映射*
	0x4002 2400 - 0x4002 2FFF	3 KB	保留	-
	0x4002 2000 - 0x4002 23FF	1 KB	FLASH 寄存器	*第 3.7.18 节：FLASH 寄存器映射*
	0x4002 1400 - 0x4002 1FFF	3 KB	保留	-
	0x4002 1000 - 0x4002 13FF	1 KB	RCC	*第 6.4.33 节：RCC 寄存器映射*
	0x4002 0800 - 0x4002 0FFF	2 KB	保留	-
	0x4002 0400 - 0x4002 07FF	1 KB	DMA2	*第 11.6.7 节：DMA 寄存器映射和复位值*
	0x4002 0800 - 0x4002 0BFF	1 KB	DMAMUX1	*第 12.6.7 节：DMAMUX 寄存器映射*
	0x4002 0C00 - 0x4002 0FFF	1 KB	保留	-
	0x4002 0000 - 0x4002 03FF	1 KB	DMA1	*第 11.6.7 节：DMA 寄存器映射和复位值*

表 1. STM32L4Rxxx 和 STM32L4Sxxx 存储器映射和外设寄存器边界地址（续）

总线	边界地址	大小（字节）	外设	外设寄存器映射
APB2	0x4001 7400 - 0x4001 FFFF	33 KB	保留	-
	0x4001 6C00 - 0x4001 73FF	1 KB	DSIHOST	*第 29.17 节：DSI 主机寄存器映射*
	0x4001 6800 - 0x4001 6BFF	1 KB	LCD-TFT	*第 28.7.26 节：LTDC 寄存器映射*
	0x4001 6000 - 0x4001 67FF	2 KB	DFSDM1	*第 27.8.16 节：DFSDM 寄存器映射*
	0x4001 5C00 - 0x4001 5FFF	1 KB	保留	-
	0x4001 5800 - 0x4001 5BFF	1 KB	SAI2	*第 47.5.12 节：SAI 寄存器映射*
	0x4001 5400 - 0x4001 57FF	1 KB	SAI1	*第 47.5.12 节：SAI 寄存器映射*
	0x4001 4C00 - 0x4001 53FF	2 KB	保留	-
	0x4001 4800 - 0x4001 4BFF	1 KB	TIM17	*第 36.6.20 节：TIM16/TIM17 寄存器映射*
	0x4001 4400 - 0x4001 47FF	1 KB	TIM16	*第 36.6.20 节：TIM16/TIM17 寄存器映射*
	0x4001 4000 - 0x4001 43FF	1 KB	TIM15	*第 36.5.20 节：TIM15 寄存器映射*
	0x4001 3C00 - 0x4001 3FFF	1 KB	保留	-
	0x4001 3800 - 0x4001 3BFF	1 KB	USART1	*第 44.7.13 节：USART 寄存器映射*
	0x4001 3400 - 0x4001 37FF	1 KB	TIM8	*第 34.4.31 节：TIM8 寄存器映射*
	0x4001 3000 - 0x4001 33FF	1 KB	SPI1	*第 46.6.8 节：SPI 寄存器映射*
	0x4001 2C00 - 0x4001 2FFF	1 KB	TIM1	*第 34.4.30 节：TIM1 寄存器映射*
	0x4001 2000 - 0x4001 2BFF	3 KB	保留	-
	0x4001 1C00 - 0x4001 1FFF	1 KB	FIREWALL	*第 4.4.8 节：防火墙寄存器映射*
	0x4001 0800 - 0x4001 1BFF	5 KB	保留	-
	0x4001 0400 - 0x4001 07FF	1 KB	EXTI	*第 16.5.13 节：EXTI 寄存器映射*
	0x4001 0200 - 0x4001 03FF	1 KB	COMP	*第 25.6.3 节：COMP 寄存器映射*
	0x4001 0030 - 0x4001 01FF	1 KB	VREFBUF	*第 24.3.3 节：VREFBUF 寄存器映射*
	0x4001 0000 - 0x4001 002F	1 KB	SYSCFG	*第 9.2.12 节：SYSCFG 寄存器映射*

表 1. STM32L4Rxxx 和 STM32L4Sxxx 存储器映射和外设寄存器边界地址（续）

总线	边界地址	大小（字节）	外设	外设寄存器映射
APB1	0x4000 9800 - 0x4000 FFFF	26 KB	保留	-
	0x4000 9400 - 0x4000 97FF	1 KB	LPTIM2	*第 38.7.11 节：LPTIM 寄存器映射*
	0x4000 8C00 - 0x4000 93FF	3 KB	保留	-
	0x4000 8400 - 0x4000 87FF	1 KB	I2C4	*第 43.7.12 节：I2C 寄存器映射*
	0x4000 8000 - 0x4000 83FF	1 KB	LPUART1	*第 45.5.11 节：LPUART 寄存器映射*
	0x4000 7C00 - 0x4000 7FFF	1 KB	LPTIM1	*第 38.7.11 节：LPTIM 寄存器映射*
	0x4000 7800 - 0x4000 7BFF	1 KB	OPAMP	*第 26.5.7 节：OPAMP 寄存器映射*
	0x4000 7400 - 0x4000 77FF	1 KB	DAC1	*第 22.7.21 节：DAC 寄存器映射*
	0x4000 7000 - 0x4000 73FF	1 KB	PWR	*第 5.4.27 节：PWR 寄存器映射和复位值表*
	0x4000 6800 - 0x4000 6FFF	2 KB	保留	-
	0x4000 6400 - 0x4000 67FF	1 KB	CAN1	*第 49.9.5 节：bxCAN 寄存器映射*
	0x4000 6000 - 0x4000 63FF	1 KB	CRS	*第 7.6.5 节：CRS 寄存器映射*
	0x4000 5C00 - 0x4000 5FFF	1 KB	I2C3	*第 43.7.12 节：I2C 寄存器映射*
	0x4000 5800 - 0x4000 5BFF	1 KB	I2C2	*第 43.7.12 节：I2C 寄存器映射*
	0x4000 5400 - 0x4000 57FF	1 KB	I2C1	*第 43.7.12 节：I2C 寄存器映射*
	0x4000 5000 - 0x4000 53FF	1 KB	UART5	*第 44.7.13 节：USART 寄存器映射*
	0x4000 4C00 - 0x4000 4FFF	1 KB	UART4	*第 44.7.13 节：USART 寄存器映射*

表 1. STM32L4Rxxx 和 STM32L4Sxxx 存储器映射和外设寄存器边界地址（续）

总线	边界地址	大小（字节）	外设	外设寄存器映射
APB1	0x4000 4800 - 0x4000 4BFF	1 KB	USART3	*第 44.7.13 节：USART 寄存器映射*
	0x4000 4400 - 0x4000 47FF	1 KB	USART2	*第 44.7.13 节：USART 寄存器映射*
	0x4000 4000 - 0x4000 43FF	1 KB	保留	-
	0x4000 3C00 - 0x4000 3FFF	1 KB	SPI3	*第 46.6.8 节：SPI 寄存器映射*
	0x4000 3800 - 0x4000 3BFF	1 KB	SPI2	*第 46.6.8 节：SPI 寄存器映射*
	0x4000 3400 - 0x4000 37FF	1 KB	保留	-
	0x4000 3000 - 0x4000 33FF	1 KB	IWDG	*第 40.4.6 节：IWDG 寄存器映射*
	0x4000 2C00 - 0x4000 2FFF	1 KB	WWDG	*第 41.4.4 节：WWDG 寄存器映射*
	0x4000 2800 - 0x4000 2BFF	1 KB	RTC	*第 42.6.21 节：RTC 寄存器映射*
	0x4000 1800 - 0x4000 23FF	4 KB	保留	-
	0x4000 1400 - 0x4000 17FF	1 KB	TIM7	*第 37.4.9 节：TIM6/TIM7 寄存器映射*
	0x4000 1000 - 0x4000 13FF	1 KB	TIM6	*第 37.4.9 节：TIM6/TIM7 寄存器映射*
	0x4000 0C00 - 0x4000 0FFF	1 KB	TIM5	*第 35.4.23 节：TIMx 寄存器映射*
	0x4000 0800 - 0x4000 0BFF	1 KB	TIM4	*第 35.4.23 节：TIMx 寄存器映射*
	0x4000 0400 - 0x4000 07FF	1 KB	TIM3	*第 35.4.23 节：TIMx 寄存器映射*
	0x4000 0000 - 0x4000 03FF	1 KB	TIM2	*第 35.4.23 节：TIMx 寄存器映射*

2.3 位段

Cortex®-M4 存储器映射包括两个位段区域。这些区域将存储器别名区域中的每个字映射到存储器位段区域中的相应位。在别名区域写入字时，相当于对位段区域的目标位执行读-修改-写操作。

在 STM324Rxxx 和 STM32L4Sxxx 器件中，外设寄存器和 SRAM 均映射到一个位段区域，这样可实现单个位段的读写操作。这些操作仅适用于 *Cortex*®-M4 访问，对于其他总线主接口（如 DMA）无效。

可通过一个映射公式说明别名区域中的每个字与位段区域中各个位之间的对应关系。映射公式为：

bit_word_addr = *bit_band_base* + (*byte_offset* x 32) + (*bit_number* × 4)

其中：

– *bit_word_addr* 代表别名区域中将映射到目标位的字的地址
– *bit_band_base* 代表别名区域的起始地址
– *byte_offset* 代表目标位所在位段区域中的字节编号
– *bit_number* 代表目标位的位位置 (0-7)

示例

下例说明如何将 SRAM1 地址 0x20000300 处字节的位 2 映射到别名区域：

0x22006008 = 0x22000000 + (0x300*32) + (2*4)

对地址 0x22006008 执行写操作相当于在 SRAM1 地址 0x20000300 处字节的位 2 执行读-修改-写操作。

对地址 0x22006008 执行读操作将返回 SRAM1 地址 0x20000300 处字节的位 2 的值（0x01 表示位置位，0x00 表示位复位）。

有关位段的详细信息，请参见 *Cortex®-M4 编程手册*（请参见*第 1 页的相关文档*）。

2.4 嵌入式 SRAM

STM32L4Rxxx 和 STM32L4Sxxx 器件的 SRAM 高达 640 KB：

- 192 KB SRAM1
- 64 KB SRAM2
- 384 KB SRAM3

这些 SRAM 可按字节、半字（16 位）或全字（32 位）访问。这些存储器可以最大系统时钟频率进行寻址，无等待状态，因此可由 CPU 和 DMA 访问。
如果选择从 SRAM1 自举或选择物理重映射（SYSCFG 控制器中的*第 9.2.1 节：SYSCFG 存储器重映射寄存器 (SYSCFG_MEMRMP)*），则 CPU 可通过系统总线或 ICode/DCode 总线访问 SRAM1。要在 SRAM1 执行期间获得最佳的性能，应选择物理重映射（通过自举引脚及软件配置来选择）。
凭借 ICode 总线访问，无需任何重映射即可从 CCM SRAM 以最佳性能执行。
SRAM2 在地址 0x2004 0000 处有别名，为 SRAM1 和 SRAM3 提供连续地址空间。

2.4.1 SRAM2 奇偶校验

用户可以使用用户选项字节中的选项位 SRAM2_PE 启用 SRAM2 奇偶校验（请参见*第 3.4.1 节：选项字节说明*）。

由于 B 类或 SIL 标准等要求提高存储器稳健性，因此数据总线宽度为 36 位，其中有 4 位用于奇偶校验（每字节 1 位）。

写入 SRAM2 时会计算并存储奇偶校验位。随后读取时会自动对其进行校验。如果某一位失败，则将生成 NMI。该错误也可以链接到 TIM1/TIM8/TIM15/TIM16/TIM17 的断路输入 BRK_IN（通过 *SYSCFG 配置寄存器 2 (SYSCFG_CFGR2)* 中的 SPL 控制位）。*SYSCFG 配置寄存器 2 (SYSCFG_CFGR2)* 中有 SRAM2 奇偶校验错误标志 (SPF)。

注： *当启用 RAM 奇偶校验时，建议在代码开始处使用软件初始化整个 RAM 存储器，以免在读取非初始化位置时出现奇偶校验错误。*

2.4.2 SRAM2 写保护

SRAM2 可获得写保护，页面大小为 1 KB。

表 2. SRAM2 组织构成

页面编号	起始地址	结束地址
第 0 页	0x1000 0000	0x1000 03FF
第 1 页	0x1000 0400	0x1000 07FF
第 2 页	0x1000 0800	0x1000 0BFF
第 3 页	0x1000 0C00	0x1000 0FFF
第 4 页	0x1000 1000	0x1000 13FF
第 5 页	0x1000 1400	0x1000 17FF
第 6 页	0x1000 1800	0x1000 1BFF
第 7 页	0x1000 1C00	0x1000 1FFF
第 8 页	0x1000 2000	0x1000 23FF
第 9 页	0x1000 2400	0x1000 27FF
第 10 页	0x1000 2800	0x1000 2BFF
第 11 页	0x1000 2C00	0x1000 2FFF
第 12 页	0x1000 3000	0x1000 33FF
第 13 页	0x1000 3400	0x1000 37FF
第 14 页	0x1000 3800	0x1000 3BFF
第 15 页	0x1000 3C00	0x1000 3FFF
第 16 页	0x1000 4000	0x1000 43FF
第 17 页	0x1000 4400	0x1000 47FF
第 18 页	0x1000 4800	0x1000 4BFF
第 19 页	0x1000 4C00	0x1000 4FFF
第 20 页	0x1000 5000	0x1000 53FF
第 21 页	0x1000 5400	0x1000 57FF
第 22 页	0x1000 5800	0x1000 5BFF
第 23 页	0x1000 5C00	0x1000 5FFF
第 24 页	0x1000 6000	0x1000 63FF
第 25 页	0x1000 6400	0x1000 67FF
第 26 页	0x1000 6800	0x1000 6BFF
第 27 页	0x1000 6C00	0x1000 6FFF
第 28 页	0x1000 7000	0x1000 73FF
第 29 页	0x1000 7400	0x1000 77FF
第 30 页	0x1000 7800	0x1000 7BFF
第 31 页	0x1000 7C00	0x1000 7FFF

表 2. SRAM2 组织构成（续）

页面编号	起始地址	结束地址
第 32 页	0x1000 8000	0x1000 83FF
第 33 页	0x1000 8400	0x1000 87FF
第 34 页	0x1000 8800	0x1000 8BFF
第 35 页	0x1000 8C00	0x1000 8FFF
第 36 页	0x1000 9000	0x1000 93FF
第 37 页	0x1000 9400	0x1000 97FF
第 38 页	0x1000 9800	0x1000 9BFF
第 39 页	0x1000 9C00	0x1000 9FFF
第 40 页	0x1000 A000	0x1000 A3FF
第 41 页	0x1000 A400	0x1000 A7FF
第 42 页	0x1000 A800	0x1000 ABFF
第 43 页	0x1000 AC00	0x1000 AFFF
第 44 页	0x1000 B000	0x1000 B3FF
第 45 页	0x1000 B400	0x1000 B7FF
第 46 页	0x1000 B800	0x1000 BBFF
第 47 页	0x1000 BC00	0x1000 BFFF
第 48 页	0x1000 C000	0x1000 C3FF
第 49 页	0x1000 C400	0x1000 C7FF
第 50 页	0x1000 C800	0x1000 CBFF
第 51 页	0x1000 CC00	0x1000 CFFF
第 52 页	0x1000 D000	0x1000 D3FF
第 53 页	0x1000 D400	0x1000 D7FF
第 54 页	0x1000 D800	0x1000 DBFF
第 55 页	0x1000 DC00	0x1000 DFFF
第 56 页	0x1000 E000	0x1000 E3FF
第 57 页	0x1000 E400	0x1000 E7FF
第 58 页	0x1000 E800	0x1000 EBFF
第 59 页	0x1000 EC00	0x1000 EFFF
第 60 页	0x1000 F000	0x1000 F3FF
第 61 页	0x1000 F400	0x1000 F7FF
第 62 页	0x1000 F800	0x1000 FBFF
第 63 页	0x1000 FC00	0x1000 FFFF

可在 SYSCFG 模块的 *SYSCFG SRAM2 写保护寄存器 (SYSCFG_SWPR)* 中启用写保护。这是一个采用一次性写“1”操作机制的寄存器，这意味着在某位写入“1”将为 SRAM 的相应页设置写保护，之后只能通过系统复位的方式将该位删除/清零。

2.4.3 SRAM2 读保护

SRAM2 受读保护 (RDP)。更多详细信息，请参见*第 3.5.1 节：读保护 (RDP)*。

2.4.4 SRAM2 擦除

通过使用用户选项字节中的选项位 SRAM2_RST 进行系统复位，可擦除 SRAM2（请参见*第 3.4.1 节：选项字节说明*）。

此外，也可以通过软件将 *SYSCFG SRAM2 控制和状态寄存器 (SYSCFG_SCSR)* 中的 SRAM2ER 位置 1 来请求擦除 SRAM2。

2.5 Flash 概述

Flash 由两个不同的物理区域组成：

- 主 Flash 块。它包含应用程序和用户数据（如有必要）。
- 信息块。它包含三部分：
  - 用于硬件和存储器保护用户配置的选项字节。
  - 包含专有自举程序代码的系统存储器。
  - OTP（一次性可编程）区域。

Flash 接口根据 AHB 协议实施指令访问和数据访问。它将实施可加快 CPU 代码执行速度的指令预取和缓存机制，以及执行通过 Flash 寄存器进行控制的 Flash 操作（编程/擦除）所必需的逻辑。更多详细信息，请参见*第 3 节：嵌入式 Flash (FLASH)*。

2.6 自举配置

2.6.1 自举配置

可以通过 BOOT0 引脚或 FLASH_OPTR 寄存器中的 nBOOT0 位（如果 FLASH_OPTR 寄存器中的 nSWBOOT0 位清零）和 FLASH_OPTR 寄存器中的 nBOOT1 位选择三种不同的自举模式，如下表所示。

表 3. 自举模式

nBOOT1 FLASH_OPTR[23]	nBOOT0 FLASH_OPTR[27]	BOOT0 引脚 PH3	nSWBOOT0 FLASH_OPTR[26]	主 Flash 为空(1)	自举存储器空间别名
X	X	0	1	0	选择主 Flash 作为自举区域
X	X	0	1	1	选择系统存储器作为自举区域
X	1	X	0	X	选择主 Flash 作为自举区域
0	X	1	1	X	选择嵌入式 SRAM1 作为自举区域
0	0	X	0	X	选择嵌入式 SRAM1 作为自举区域

表 3. 自举模式（续）

nBOOT1 FLASH_OPTR[23]	nBOOT0 FLASH_OPTR[27]	BOOT0 引脚 PH3	nSWBOOT0 FLASH_OPTR[26]	主 Flash 为空(1)	自举存储器空间别名
1	X	1	1	X	选择系统存储器作为自举区域
1	0	X	0	X	选择系统存储器作为自举区域

1. 如果未编程第一个 Flash 位置 (0xFFFF FFFF) 且选择了从主 Flash 自举，则会执行 Flash 空检查机制以强制从系统 Flash 自举。

复位后，锁存 BOOT0 引脚（来自引脚或选项位）和 nBOOT1 位的值。用户可以通过设置 nBOOT1 和 BOOT0 引脚来选择需要的自举模式。

当退出待机模式时，还会对 BOOT0 引脚或用户选项位（取决于 FLASH_OPTR 寄存器中的 nSWBOOT0 位值）和 nBOOT1 位进行重新采样。因此，当处于待机模式时，这些引脚必须保持所需的自举模式配置。该启动延迟结束后，CPU 将从地址 0x0000 0000 获取栈顶值，然后从始于 0x0000 0004 的自举存储器开始执行代码。

根据所选的自举模式，主 Flash 存储器、系统存储器或 SRAM1 可如下访问：

- 从主 Flash 存储器自举：主 Flash 存储器在自举存储器空间 (0x0000 0000) 中有别名，但也可从它原来的存储器空间 (0x0800 0000) 访问。换句话说，闪存内容可从地址 0x0000 0000 或 0x0800 0000 开始访问。
- 从系统存储器自举：系统存储器在自举存储器空间 (0x0000 0000) 中有别名，但也可从它原来的存储器空间 (0x1FFF 0000) 访问。
- 从嵌入式 SRAM1 自举：SRAM1 在自举内存空间 (0x0000 0000) 中有别名，但也可从它原来的内存空间 (0x2000 0000) 访问。

PH3/BOOT0 GPIO 配置如下：

- 如果 FLASH_OPTR 寄存器中的选项位 nSWBOOT0 置 1，则在整个复位阶段配置为输入模式，并在复位后自动切换到模拟模式（BOOT0 引脚）。
- 如果 FLASH_OPTR 寄存器中的 nSWBOOT0 位清零，则从复位阶段到选项字节加载完成阶段配置为输入模式（选项位的 BOOT0 值）。之后，即使复位阶段未完成，也会自动切换到模拟模式。

注： *如果器件从 SRAM 自举，在应用程序初始化代码中，需要使用 NVIC 异常及中断向量表和偏移寄存器来重新分配 SRAM 中的向量表。*

从主闪存自举时，应用软件可从存储区 1 或存储区 2 自举。默认情况下，选择从存储区 1 自举。要选择从闪存存储区 2 自举，则设置用户选项字节中的 BFB2 位。当该位置 1 且自举引脚采用从主 Flash 自举的配置时，器件从系统存储器自举，自举程序跳转到执行 Flash 存储区 2 中编程的用户应用程序。更多详细信息，请参见 AN2606。

物理重映射

选择自举引脚模式后，应用程序软件可以将某些存储器设定为从代码空间进行访问（这样，可通过 ICode 总线而非系统总线执行代码）。这样的修改通过在 SYSCFG 控制器中编程 *SYSCFG 存储器重映射寄存器 (SYSCFG_MEMRMP)* 来实现。

因此可重映射以下存储器：

- 主 Flash
- 系统存储器
- 嵌入式 SRAM1 (192 KB)
- FSMC 块 1（NOR/PSRAM 1 和 2）
- OctoSPI（OCTOSPI1 或 OSCTOSPI2）存储器

表 4. 存储器映射与自举模式/物理重映射

地址	主 Flash 中的自举/重映射	嵌入式 SRAM 1 中的自举/重映射	系统中的自举/重映射存储器	FSMC 中的重映射	OCTOSPI 中的重映射
0x2000 0000 - 0x2002 FFFF	SRAM1	SRAM1	SRAM1	SRAM1	SRAM1
0x1FFF 7000 - 0x1FFF FFFF	系统存储器/OTP/选项字节	系统存储器/OTP/选项字节	系统存储器/OTP/选项字节	系统存储器/OTP/选项字节	系统存储器/OTP/选项字节
0x1000 8000 - 0x1FFE FFFF	保留	保留	保留	保留	保留
0x1000 0000 - 0x1000 FFFF	SRAM2	SRAM2	SRAM2	SRAM2	SRAM2
0x0820 0000 - 0x0FFF FFFF	保留	保留	保留	保留	保留
0x0800 0000 - 0x081F FFFF	Flash	Flash	Flash	Flash	Flash
0x0400 0000 - 0x07FF FFFF	保留	保留	保留	FSMC 块 1 NOR/ PSRAM 2 (128 MB) （使用别名）	OCTOSPI 块 (128 MB) （使用别名）
0x0010 0000 - 0x03FF FFFF	保留	保留	保留	FSMC 块 1 NOR/ PSRAM 1 (128 MB) （使用别名）	OCTOSPI 块 (128 MB) （使用别名）
0x0000 0000 - 0x001F FFFF (1) (2)	Flash (2 MB) （使用别名）	SRAM1 (192 KB) （使用别名）	系统存储器 (28 KB) （使用别名）	FSMC 块 1 NOR/ PSRAM 1 (128 MB) （使用别名）	OCTOSPI 块 (128 MB) （使用别名）

1. 在地址 0x0000 0000 重映射 FSMC 时，只能重映射块 1 存储器控制器的前两个区域（块 1 NOR/PSRAM 1 和 NOR/PSRAM 2）。在地址 0x0000 0000 重映射 OCTOSPI 时，只能重映射 128 MB。在重映射模式下，CPU 可以通过 ICode 总线（而不是系统总线）访问外部存储器来提高性能。
2. 即使在自举存储空间中使用别名，相关存储器仍可通过其原始存储空间进行访问。

嵌入式自举程序

嵌入式自举程序位于系统存储器中，由 ST 在生产阶段编程。请参见 AN2606 STM32 微控制器系统存储器自举模式。

3 嵌入式 Flash (FLASH)

3.1 简介

Flash 接口可管理 CPU 通过 AHB ICode 和 DCode 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。

3.2 Flash 主要特性

- 高达 2 MB 的 Flash，采用双存储区架构，支持同步读写功能 (RWW)。
- Flash 读操作，支持以下两种数据宽度模式：
  - 单存储区模式 DBANK=0：128 位的读访问
  - 双存储区模式 DBANK=1：64 位的读访问
- 页擦除、存储区擦除和批量擦除（两个存储区）

Flash 接口特性：

- Flash 读操作
- Flash 编程/擦除操作
- 由选项字节 (RDP) 激活的读保护
- 4 个写保护区域（DBANK=1 时每个存储区 2 个，DBANK=0 时 4 个区域用于完整存储器）
- 2 个专有代码读保护区域（DBANK=1 时每个存储区 1 个，DBANK=0 时 2 个区域用于完整存储器）
- Flash 为空检查
- ICODE 上的预取操作
- 指令缓存：ICode 上的 32 个缓存行，4 x 64 或 2 x 128 位宽 (1 KB RAM)
- 数据缓存：DCode 上的 8 个缓存行，4 x 64 或 2 x 128 位宽（256 字节 RAM）
- 误码校正 ECC：8 位/64 位双字
  - DBANK=1：8 + 64 = 72 位，2 位检测，1 位修正
  - DBANK=0：(8+64) + (8+64) = 144 位，2 位检测，1 位修正
- 选项字节加载器
- 低功耗模式

3.3 Flash 功能说明

3.3.1 Flash 构成

Flash 具有以下主要特性：

- 容量高达 2 MB，支持单存储区模式（读宽度为 128 位）或双存储区模式（读宽度为 64 位）
- 凭借 BFB2 选项位，支持双存储区自举模式（仅限双存储区模式）
- DBANK 位置 1 时为双存储区模式：
  - 2 MB 包含在主存储器的 2 个存储区中
  - 页大小为 4 KB
  - 数据读取宽度为 72 位（64 位 + 8 个 ECC 位）
  - 存储区擦除与批量擦除
- DBANK 位复位时为单存储区模式：
  - 2 MB 包含在主存储器的 1 个存储区中
  - 页大小为 8 KB
  - 数据读取宽度为 144 位（128 位 + 2x8 个 ECC 位）
  - 批量擦除

Flash 结构如下：

- 主存储器块构成取决于双存储区配置位：
  - 使能双存储区时（DBANK 位置 1），Flash 分为 2 个 1 MB 的存储区，每个存储区的构成如下：
    主存储器块包含 256 页，每页 4 KB
    每页 8 行，每行 512 字节
  - 禁止双存储区时（DBANK 位复位），主存储器块由一个 2 MB 的存储区构成，如下所述：
    主存储器块包含 256 页，每页 8 KB
    每页 8 行，每行 1024 字节
- 信息块包括：
  - 系统存储器，器件在系统存储器自举模式下从该存储器启动。此区域留给意法半导体使用，其中包含自举程序，用以通过以下接口之一对 Flash 进行重新编程：USART1、USART2、USART3、USB (DFU)、I2C1、I2C2、I2C3、SPI1、SPI2、SPI3。它由意法半导体在器件制造期间编程并加以保护，以防止误写/误擦除操作。更多详细信息，请参见 *www.st.com* 上提供的 AN2606。
  - 1 KB（128 个双字）OTP（一次性可编程）字节，用于存储用户数据。OTP 区域仅适用于存储区 1。OTP 数据不能擦除，只能执行一次写操作。如果仅一位为 0，则即使值为 0x0000 0000 0000 0000，整个双字也无法再写入。
  - 用于用户配置的选项字节。

存储器由一个主区域和一个信息块构成，如图所示。

表 5. Flash 模块——2 MB 双存储区结构（64 位读宽度）

Flash 区域		Flash 地址	大小（字节）	名称
主存储器	存储区 1	0x0800 0000 - 0x0800 0FFF	4 K	第 0 页
		0x0800 1000 - 0x0800 1FFF	4 K	第 1 页
		0x0800 2000 - 0x0800 2FFF	4 K	第 2 页
		0x0800 3000 - 0x0800 3FFF	4 K	第 3 页
		- - - -	- - - -	- - - -
		0x080F F000 - 0x080F FFFF	4 K	第 255 页
	存储区 2	0x0810 0000 - 0x0810 0FFF	4 K	第 0 页
		0x0810 1000 - 0x0810 1FFF	4 K	第 1 页
		0x0810 2000 - 0x0810 2FFF	4 K	第 2 页
		0x0810 3000 - 0x0810 3FFF	4 K	第 3 页
		- - - -	- - - -	- - - -
		0x081F F000 - 0x081F FFFF	4 K	第 255 页
信息块	存储区 1	0x1FFF 0000 - 0x1FFF 6FFF	28 K	系统存储器
	存储区 2	0x1FFF 8000 - 0x1FFF EFFF	28 K	
	存储区 1	0x1FFF 7000 - 0x1FFF 73FF	1 K	OTP 区域
	存储区 1	0x1FF0 0000 - 0x1FF0 000F	16	选项字节
	存储区 2	0x1FF0 1000 - 0x1FF0 100F	16	

表 6. Flash 模块——2 MB 单存储区结构（128 位读宽度）

Flash 区域	Flash 地址	大小（字节）	名称
主存储器	0x0800 0000 - 0x0800 1FFF	8 K	第 0 页
	0x0800 2000 - 0x0800 3FFF	8 K	第 1 页
	0x0800 4000 - 0x0800 5FFF	8 K	第 2 页
	- - - -	- - - -	- - - -
	0x0801 C000 0x0801 DFFF	8 K	第 244 页
	0x081F E000 - 0x081F FFFF	8 K	第 255 页

表 6. Flash 模块——2 MB 单存储区结构（128 位读宽度）（续）

Flash 区域		Flash 地址	大小（字节）	名称
信息块	存储区 1	0x1FFF 0000 - 0x1FFF 6FFF	28 K	系统存储器
	存储区 2	0x1FFF 8000 - 0x1FFF EFFF	28 K	
	存储区 1	0x1FFF 7000 - 0x1FFF 73FF	1 K	OTP 区域
	存储区 1	0x1FF0 0000 - 0x1FF0 000F	16	选项字节
	存储区 2	0x1FF0 1000 - 0x1FF0 100F	16	

3.3.2 误码校正 (ECC)

双存储区模式（DBANK=1，64 位数据宽度）

Flash 中的数据为 72 位字：每个双字（64 位）增加了 8 位。ECC 机制支持：

- 单个错误检测和修正
- 两个错误检测

当检测到一个错误并对其进行修正时，*Flash ECC 寄存器 (FLASH_ECCR)* 中的标志 ECCC（ECC 校正）置 1。如果 ECCCIE 置 1，则会生成中断。

检测到两个错误时，FLASH_ECCR 寄存器中的标志 ECCD（ECC 检测）置 1。在这种情况下会生成 NMI。

检测到一个 ECC 错误时，出错双字的地址及其相关存储区会保存在 FLASH_ECCR 寄存器的 ADDR_ECC[20:0] 和 BK_ECC 中。ADDR_ECC[2:0] 始终清零。

ECCC 或 ECCD 置 1 时，如果出现新的 ECC 错误，则 ADDR_ECC 和 BK_ECC 不会更新。仅当 ECC 标志清零后，FLASH_ECCR 才会更新。

注： *对于初始数据：0xFF FFFF FFFF FFFF FFFF，可检测并纠正一个错误，但不支持两个错误检测。*

报告 ECC 错误时，如果数据仍存在于当前缓冲区中，则即使 ECCC 和 ECCD 清零，在失效地址处的新读取操作也可能不会生成 ECC 错误。

单存储区模式（DBANK=0，128 位数据宽度）

Flash 中的数据为 144 位字：每个双字增加了 8 位。ECC 机制支持：

- 单个错误检测和修正
- 基于 64 个双字进行两个错误检测

用户必须先检查 SYSF_ECC 位，如果此位置 1，则用户必须参考 DBANK=1 编程模型（因为系统 Flash 始终位于 2 个存储区）。如果此位未置 1，则用户必须参考以下编程模型：

每个双字（位 63:0 和位 127:64）都有 ECC。

如果在 64 个 LSB 位（位 63:0）检测到一个错误并对其进行了纠正，FLASH_ECCR 寄存器中的标志 ECCC（ECC 校正）置 1。

如果在 64 个 MSB 位（位 127:64）检测到一个错误并对其进行了纠正，FLASH_ECCR 寄存器中的标志 ECCC2（ECC2 校正）置 1。

如果 ECCCIE 置 1，则会生成中断。用户必须读取 ECCC 和 ECCC2，确定 128 位数据已修正的部分（63:0、127:64 或二者）。

在 64 个 LSB 位中检测到两个错误时，FLASH_ECCR 寄存器中的标志 ECCD（ECC 检测）置 1。

在 64 个 MSB 位（位 127:64）中检测到两个错误时，FLASH_ECCR 寄存器中的标志 ECCD2（ECC2 检测）置 1。

在这种情况下会生成 NMI。用户必须读取 ECCD 和 ECCD2，确定 128 位数据已检测到错误的部分（63:0、127:64 或二者）。

检测到 ECC 错误时，2 倍双字的失效地址保存到 FLASH_ECCR 的 ADDR_ECC[20:0] 中。ADDR_ECC[20:0] 包含 2 倍双字的地址。

ADDR_ECC[3:0] 始终清零。在此模式下不使用 BK_ECC。

ECCC/ECCC2 或 ECCD/ECCD2 置 1 时，如果发生新的 ECC 错误，则 ADDR_ECC 不会更新。仅当 ECC 标志 (ECCC/ECCC2/ECCD/ECCD2) 清零时，FLASH_ECCR 才会更新。

注： *对于初始数据：0xFF FFFF FFFF FFFF FFFF，可检测并纠正一个错误，但不支持两个错误检测。*
报告 ECC 错误时，如果数据仍存在于当前缓冲区中，则即使 ECCC 和 ECCD 清零，在失效地址处的新读取操作也可能不会生成 ECC 错误。

3.3.3 读访问延迟

为了准确读取 Flash 数据，必须根据 CPU 时钟 (HCLK) 频率和器件 V_{CORE} 内部电压范围在 *Flash 访问控制寄存器 (FLASH_ACR)* 中正确地编程等待状态数 (LATENCY)。请参见*第 5.1.7 节：动态电压调节管理*。*表 7* 所示为等待状态与 CPU 时钟频率之间的对应关系。

表 7. CPU 时钟 (HCLK) 频率对应的等待状态数

等待状态 (WS)（延迟）	HCLK (MHz)	
	V_{CORE} 范围 1	V_{CORE} 范围 2
0 WS（1 个 CPU 周期）	≤20	≤8
1 WS（2 个 CPU 周期）	≤40	≤16
2 WS（3 个 CPU 周期）	≤60	≤26
3 WS（4 个 CPU 周期）	≤80	-
4 WS（5 个 CPU 周期）	≤100	-
5 WS（6 个 CPU 周期）	≤120	-

复位后，CPU 时钟频率为 4 MHz，FLASH_ACR 寄存器中的等待状态 (WS) 为 0。

更改 CPU 频率时，必须应用以下软件顺序来调节访问 Flash 所需的等待状态数：

需要提高 CPU 频率时的操作步骤：

1. 将新的等待状态数编程到 *Flash 访问控制寄存器 (FLASH_ACR)* 中的 LATENCY 位。
2. 通过读取 FLASH_ACR 寄存器，检查新的等待状态数是否设置成功。
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. 将新的等待状态数编程到 *Flash 访问控制寄存器 (FLASH_ACR)* 中的 LATENCY 位。
5. 通过读取 FLASH_ACR 寄存器，检查新的等待状态数是否设置成功。

3.3.4 自适应实时存储器加速器 (ART Accelerator™)

特有的自适应实时 (ART) 存储器加速器对 STM32（符合工业标准，带有 FPU 的 Arm® Cortex®-M4 处理器）进行了优化。该加速器很好地体现了带有 FPU 的 Arm® Cortex®-M4 在 Flash 技术方面的固有性能优势，克服了通常条件下，高速的处理器在运行中需要经常等待 Flash 读取的情况。

为了发挥处理器的全部性能，该加速器将实施指令预取队列和分支缓存，从而提高了 64 位 Flash 的程序执行速度。根据 CoreMark 基准测试，凭借 ART 加速器所获得的性能相当于 Flash 在 CPU 频率高达 120 MHz 时以 0 个等待状态执行程序。

指令预取

Cortex®-M4 通过 ICode 总线获取指令，通过 DCode 总线获取缓冲池（常数/数据）。预取块用于提高 ICode 总线访问的效率。

在单存储区模式（DBANK 选项位复位）下，每个 Flash 读操作可读取 128 位，可以是四行 32 位指令，也可以是八行 16 位指令，具体取决于启动的程序。此 128 位当前指令行保存在当前缓冲区中，对于顺序执行的代码，至少需要四个 CPU 周期来执行前一指令行的读取操作。

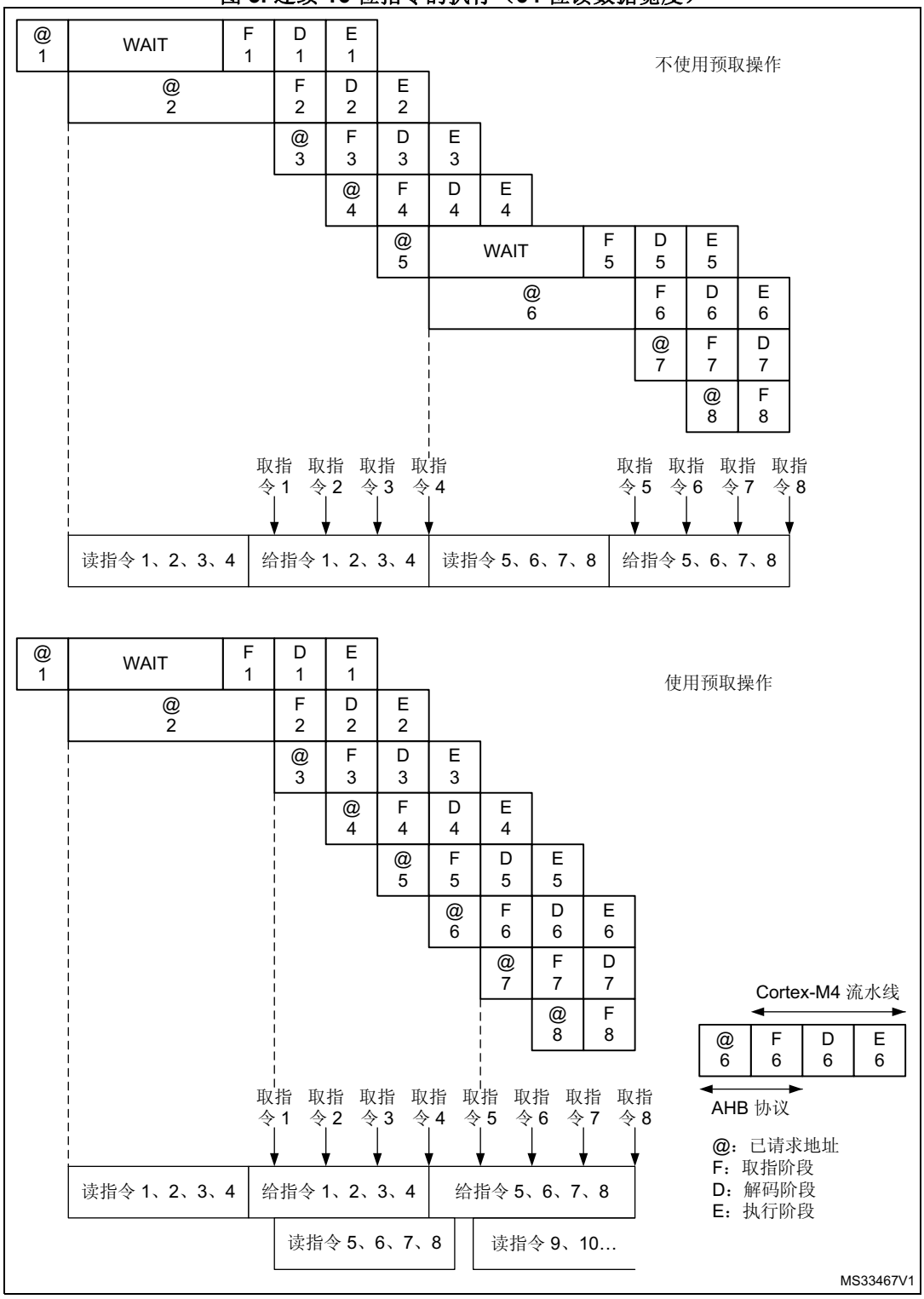
在双存储区模式（DBANK 选项位置 1）下，每个 Flash 读操作可读取 64 位，可以是两行 32 位指令，也可以是四行 16 位指令，具体取决于启动的程序。此 64 位当前指令行保存在当前缓冲区中，对于顺序执行的代码，至少需要两个 CPU 周期来执行前一指令行的读取操作。

在 CPU 请求当前指令行时，可使用 ICode 总线的预取操作读取 Flash 中的下一个连续存放的 128 位指令行。

可将 *Flash 访问控制寄存器 (FLASH_ACR)* 中的 PRFTEN 位置 1，来使能预取功能。当访问 Flash 至少需要一个等待状态时，此功能非常有用。

图 3 所示为需要 3 WS（3 个等待状态）访问 Flash 时连续 16 位指令的执行过程，图中分别介绍了使用和不使用预取操作两种情况。

图 3. 连续 16 位指令的执行（64 位读数据宽度）



处理非顺序执行的代码（有分支）时，指令可能并不存在于当前使用的或预取的指令行中。这种情况下，CPU 等待时间至少等于等待状态数。

如果当前缓冲区中存在循环，则不会执行新的 Flash 访问。

指令缓存存储器 (I-Cache)

为了减少因指令跳转而产生的时间损耗，可将 32 行 4 x 64 位的指令保持在指令缓存存储器的双存储区模式下，或将 32 行 2 x 128 的指令保持在指令缓存存储器的单存储区模式。可将 *Flash 访问控制寄存器 (FLASH_ACR)* 中的指令缓存使能 (ICEN) 位置 1，来使能此特性。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存储器中，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。

系统复位后指令缓存存储器使能。

数据缓存存储器 (D-Cache)

在 CPU 流水线执行阶段，将通过 DCode 总线访问 Flash 中的数据缓冲池。每次 DCode 总线读访问都会获取保存在当前缓冲区中的 64 位或 128 位。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 DCode 进行的访问优先于通过 AHB 指令总线 ICode 进行的访问。

如果频繁使用某些数据缓冲池，可将 *Flash 访问控制寄存器 (FLASH_ACR)* 中的数据缓存使能 (DCEN) 位置 1，来使能数据缓存存储器。此功能的工作原理与指令缓存存储器类似，但保留的数据大小在双存储区模式下被限制为 8 行 4x64 位，在单存储区模式下被限制为 8 行 2x128 位。

系统复位后数据缓存存储器使能。

注： *仅当 CPU（而非 DMA1 和 DMA2）请求数据后 D-Cache 才会激活。*

选项字节块中的数据不可缓存。

3.3.5 Flash 编程/擦除操作

STM32L4Rxxx 和 STM32L4Sxxx 嵌入式 Flash 可采用在线编程或在应用中编程两种方式。

在线编程 (ICP) 方式适用于更新 Flash 的所有内容，更新时使用 JTAG、SWD 协议或自举程序将用户应用程序加载到微控制器。ICP 可实现快速而高效的设计迭代，并且避免了不必要的器件封装处理或插接。

与 ICP 方法相比，**在应用中编程 (IAP)** 可通过微控制器支持的任何通信接口（I/O、USB、CAN、UART、I^2C 和 SPI 等）将编程数据下载到存储器。IAP 允许用户在应用运行时重新编程 Flash。但是，部分应用程序必须事先通过 ICP 方式编程到 Flash。

如果在 Flash 操作期间发生器件复位，无法保证 Flash 中的内容。

只要 CPU 不访问同一 Flash 存储区，正在进行的 Flash 操作就不会阻塞 CPU。代码或数据获取操作以及写入/擦除操作可在不同的存储区同时执行（参见 *第 3.3.8 节：同步读写 (RWW) 仅适用于双存储区模式 (DBANK=1)*）。Flash 擦除和编程操作仅适用于电压调节范围 1。必须将 PWR_CR1 中的 VOS[1:0] 位编程为 01b。

相反，对 Flash 执行编程/擦除操作期间，如果尝试读取同一 Flash 存储区，总线便会停止工作。在完成编程/擦除操作后，会正确执行读操作。

解锁 Flash

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_KEYR 寄存器中写入 KEY2 = 0xCDEF89AB。

如果操作顺序不正确，会锁定 FLASH_CR 寄存器，下次系统复位才会解锁。如果密钥操作顺序不正确，会检测到总线错误并生成硬性故障中断。

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

注： *Flash 状态寄存器 (FLASH_SR) 中的 BSY 位置 1 时，FLASH_CR 寄存器无法写入。BSY 位为 1 时，对该寄存器尝试进行写操作会导致 AHB 总线阻塞，直到 BSY 位清零。*

3.3.6 Flash 主存储器擦除顺序

Flash 擦除操作可在页级别、存储区级别或在整个 Flash（批量擦除）执行。批量擦除不影响信息块（系统 Flash、OTP 和选项字节）。

页擦除

页擦除的具体步骤如下：

1. 检查 *Flash 状态寄存器 (FLASH_SR)* 中的 BSY 位，以确认当前未执行任何 Flash 操作。
2. 检查并清零之前的编程所导致的全部错误编程标志。否则，PGSERR 将置 1。
3. 在双存储区模式下（DBANK 选项位置 1），将 PER 位置 1，并选择要擦除的页 (PNB) 和相关存储区 (BKER)（在 Flash 控制寄存器 (FLASH_CR) 中）。在单存储区模式下（DBANK 选项位置 1），将 PER 位置 1，选择要擦除的页 (PNB)。Flash 控制寄存器 (FLASH_CR) 中的 BKER 位必须保持清零。
4. 将 FLASH_CR 寄存器中的 STRT 位置 1。
5. 等待 FLASH_SR 寄存器中的 BSY 位清零。

注： *STRT 位置 1 时，内部振荡器 HSI16 (16 MHz) 自动使能，STRT 位清零时其会自动禁止，但 HSI16 之前已通过 RCC_CR 寄存器中的 HSION 使能的情况除外。*

如果页擦除的内容是写保护区域（受 WRP 或 PCROP 保护）的一部分，则 WRPERR 置 1，同时页擦除请求中止。

存储区 1、存储区 2 批量擦除（仅适用于双存储区模式 DBANK=1）

存储区批量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. 检查并清零之前的编程所导致的全部错误编程标志。否则，PGSERR 将置 1。
3. 将 *Flash 控制寄存器 (FLASH_CR)* 中的 MER1 位或 MER2 位置 1（具体取决于存储区）。可基于相同的操作选择这两个存储区，此时对应于批量擦除。
4. 将 FLACH_CR 寄存器中的 STRT 位置 1。
5. 等待 *Flash 状态寄存器 (FLASH_SR)* 中的 BSY 位清零。

批量擦除

批量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. 检查并清零之前的编程所导致的全部错误编程标志。否则，PGSERR 将置 1。
3. 将 Flash 控制寄存器 (FLASH_CR) 中的 MER1 和 MER2 位置 1。
4. 将 FLACH_CR 寄存器中的 STRT 位置 1。
5. 等待 Flash 状态寄存器 (FLASH_SR) 中的 BSY 位清零。

注： *STRT 位置 1 时，内部振荡器 HSI16 (16 MHz) 自动使能，STRT 位清零时其会自动禁止，但 HSI16 之前已通过 RCC_CR 寄存器中的 HSION 使能的情况除外。*

DBANK=0 时，如果仅 MERA 或 MERB 位置 1，则 PGSERR 会置 1，不执行任何擦除操作。

如果要擦除的存储区或要擦除的存储区之一包含写保护区域（受 WRP 或 PCROP 保护），则 WRPERR 置 1，批量擦除请求被中止（如果选择了两个存储区，则对于二者均适用）。

3.3.7 Flash 主存储器编程顺序

Flash 一次编程为 72 位（64 位 + 8 位 ECC）。

不允许在之前编程过的地址中编程，尝试执行此操作将使 *Flash 状态寄存器 (FLASH_SR)* 中的 PROGERR 标志置 1，但要写入的输入全为 0 时除外。

只能编程双字（2 x 32 位数据）。

- 尝试写入字节或半字将使 FLASH_SR 寄存器中的 SIZERR 标志置 1。
- 尝试写入与双字地址不匹配的双字时，将使 FLASH_SR 寄存器中的 PGAERR 标志置 1。

标准编程

标准模式下，Flash 编程顺序如下：

1. 检查 *Flash 状态寄存器 (FLASH_SR)* 中的 BSY 位，以确认当前未执行任何 Flash 主存储器操作。
2. 检查并清零之前的编程所导致的全部错误编程标志。否则，PGSERR 将置 1。
3. 将 *Flash 控制寄存器 (FLASH_CR)* 中的 PG 位置 1。
4. 针对所需存储器地址（主存储器块或 OTP 区域内）执行数据写入操作。只能编程双字。
   - 在与双字匹配的地址中写入第一个字
   - 写入第二个字
5. 等待，直到 FLASH_SR 寄存器中的 BSY 位清零。
6. 检查 FLASH_SR 寄存器中的 EOP 标志是否置 1（表示已成功执行编程操作），并用软件将其清零。
7. 如果不再有编程请求，则将 FLASH_SR 寄存器中的 PG 位清零。

注： *Flash 接口接收到适当的序列（双字）后，自动启动编程且 BSY 位置 1。PG 位置 1 时，内部振荡器 HSI16 (16 MHz) 自动使能，PG 位清零时其会自动禁止，但 HSI16 之前已通过 RCC_CR 寄存器中的 HSION 使能的情况除外。*

如果用户需要仅编程一个字，则必须使用擦除值 0xFFFF FFFF 补全双字以启动自动编程。

ECC 由双字计算得出以供编程。

针对一行（DBANK=1 时 64 个双字）或半行（DBANK=0 时 64 个双字）的快速编程

此模式允许编程一行（DBANK=1 时 64 个双字）或半行（DBANK=0 时 64 个双字），无需在编程之前验证 Flash 单元，从而缩短了页编程时间，同时可针对每个双字避免高压的上升和下降时间。快速编程过程中，CPU 时钟频率 (HCLK) 必须至少为 8 MHz。

只有主存储器可编程为快速编程模式。

标准模式下，Flash 主存储器编程顺序如下：

1. 在单存储区模式下 (DBANK=0)，执行批量擦除。否则，PGSERR 将置 1。只有在基于 RAM 或自举程序执行代码时，才能执行快速编程。在双存储区模式下 (DBANK=1)，对存储区执行批量擦除以进行编程。否则，PGSERR 将置 1。
2. 检查 *Flash 状态寄存器 (FLASH_SR)* 中的 BSY 位，以确认当前未执行任何 Flash 主存储器操作。
3. 检查并清零之前的编程所导致的全部错误编程标志。
4. 将 *Flash 控制寄存器 (FLASH_CR)* 中的 FSTPG 位置 1。
5. 写入 64 个双字以编程一行或半行。只能编程双字：
   - 在与双字匹配的地址中写入第一个字。
   - 写入第二个字。
6. 等待，直到 FLASH_SR 寄存器中的 BSY 位清零。
7. 检查 FLASH_SR 寄存器中的 EOP 标志是否置 1（表示已成功执行编程操作），并用软件将其清零。
8. 如果不再有编程请求，则将 FLASH_SR 寄存器中的 FSTPG 位清零。

注： *如果同一存储区中正在执行读操作时，尝试在快速编程模式下对 Flash 进行写操作，则会中止编程，且不会发出任何系统通知（无错误标志置 1）。*

Flash 接口接收到第一个双字后，编程自动启动。对第一个双字施加高压时，BSY 位置 1，最后一个双字已编程或发生错误时，该位清零。FSTPG 位置 1 时，内部振荡器 HSI16 (16 MHz) 自动使能，FSTPG 位清零时其会自动禁止，但 HSI16 之前已通过 RCC_CR 寄存器中的 HSION 使能的情况除外。

64 个双字必须连续写入。对于所有编程，Flash 均保持高压（持续时间约 2 x 25 µs）。两个双字写入请求之间的最长时间为编程时间。如果第二个双字在此编程时间过后传到，则快速编程被中断，MISSERR 置 1。

在 2 次擦除之间，对于一整行，高压持续时间不得超过 8 ms。可通过基于大于等于 8 MHz 的时钟系统连续写入的 64 个双字的序列来保证这一点。设置快速编程后内部超时计数器计数 7 ms，并在超时后会停止编程。此时，FASTERR 位将置 1。

如果发生错误，则高压停止，不会对下一个要编程的双字进行编程。总之，之前所有的双字均已正确编程。

编程错误

可检测到多种错误。若发生错误，Flash 操作（编程或擦除）会中止。

- **PROGERR**：编程错误

  在标准编程过程中：如果要写入的字之前未擦除，则 PROGERR 置 1（要编程的值全为 0 时除外）。

- **SIZERR**：大小编程错误

  在标准编程或快速编程过程中：只能编程双字且只能写入 32 位数据。如果写入一个字节或半字，则 SIZERR 置 1。

- **PGAERR**：编程对齐错误

  如果发生以下事件，PGAERR 位会置 1：

  – 在标准编程过程中：要编程的第一个字与双字地址不匹配，或第二个字不属于同一双字地址。

  – 在快速编程过程中：要编程的数据与之前编程的双字不属于同一行，或者要编程的地址小于或等于之前的地址。

- **PGSERR**：编程顺序错误

  如果发生以下事件，PGSERR 位会置 1：

  – 对于标准编程顺序或快速编程顺序：PG 和 FSTPG 清零时写入数据。

  – 对于标准编程顺序或快速编程顺序：PG 或 FSTPG 置 1 后，MER1、MER2 和 PER 不清零。

  – 对于快速编程顺序：将 FSTPG 位置 1 后执行批量擦除操作。

  – 对于批量擦除顺序：MER1 或 MER2 置 1 后，PG、FSTPG 和 PER 不清零。

  – 对于页擦除顺序：PER 置 1 后，PG、FSTPG、MER1 和 MER2 不清零。

  – 如果由于之前发生编程错误，导致 PROGERR、SIZERR、PGAERR、WRPERR、MISSERR、FASTERR 或 PGSERR 置 1，则 PGSERR 也会置 1。

  – DBANK=0 时，如果仅 MER1 或 MER2 置 1，则 PGSERR 也会置 1（不允许存储区批量擦除）。

- **WRPERR**：写保护错误

  如果发生以下事件，WRPERR 位会置 1：

  – 尝试在写保护区域 (WRP) 或 PCROP 区域执行编程或擦除操作。

  – 尝试在一页或多页受 WRP 或 PCROP 保护时，执行存储区擦除操作。

  – 读保护 (RDP) 设为级别 1 时，已连接调试功能或从 SRAM 或系统 Flash 进行自举。

  – 尝试在读保护 (RDP) 设置为级别 2 时修改选项字节。

- **MISSERR**：快速编程数据丢失错误

  在快速编程过程中：所有数据必须连续写入。如果之前的数据编程已完成，并且要编程的下一个数据尚未写入，则 MISSERR 置 1。

- **FASTERR**：快速编程错误

  在快速编程过程中：如果发生以下事件，FASTERR 会置 1：

  – FSTPG 位置 1 的时间超过 7 ms，这会生成超时检测。

  – 快速编程被 MISSERR、PGAERR、WRPERR 或 SIZERR 中断。

如果在编程或擦除操作期间出现错误，则 FLASH_SR 寄存器中的以下错误标志之一将置 1：

PROGERR、SIZERR、PGAERR、PGSERR、MISSERR（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标志 OPERR 置 1，并且如果 *Flash 状态寄存器 (FLASH_SR)* 中的错误中断使能位 ERRIE 置 1，则将产生一个中断。

注： *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。*

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 访问控制寄存器 (FLASH_ACR)* 中的 DCRST 和 ICRST 位置 1，以刷新缓存。

注： *I/D 缓存只有在被禁止 (I/DCEN = 0) 的情况下才能刷新。*

3.3.8 同步读写 (RWW) 仅适用于双存储区模式 (DBANK=1)

双存储区模式仅在 DBANK 选项位复位时可用，允许执行同步读写操作。该功能允许在不同存储区同时执行读取操作和擦除/编程操作。

注： *不允许执行同步写入操作。例如，不能在不同存储区同时执行擦除操作和编程操作。*

从存储区 1 读取，同时在存储区 2 执行页擦除（或反之）

从存储区 1 执行程序代码时，可在存储区 2 执行页擦除操作（或反之）。具体步骤如下：

1. 检查 *Flash 状态寄存器 (FLASH_SR)* 中的 BSY 位，以确认当前未执行任何 Flash 操作（在存储区 1 或存储区 2 执行擦除/编程操作时，BSY 有效）。
2. 在 *Flash 控制寄存器 (FLASH_CR)* 中，将 PER 位、PSB 置 1 来选择页，将 BKER 置 1 来选择存储区。
3. 将 FLASH_CR 寄存器中的 STRT 位置 1。
4. 等待 BSY 位清零（或使用 EOP 中断）。

从存储区 1 读取，同时在存储区 2 执行批量擦除（或反之）

从存储区 1 执行程序代码时，可在存储区 2 执行批量擦除操作（或反之）。具体步骤如下：

1. 检查 *Flash 状态寄存器 (FLASH_SR)* 中的 BSY 位，以确认当前未执行任何 Flash 操作（在存储区 1 或存储区 2 执行擦除/编程操作时，BSY 有效）。
2. 将 *Flash 控制寄存器 (FLASH_CR)* 中的 MER1 或 MER2 置 1。
3. 将 FLASH_CR 寄存器中的 STRT 位置 1。
4. 等待 BSY 位清零（或使用 EOP 中断）。

从存储区 1 读取，同时在存储区 2 编程（或反之）

从存储区 1 执行程序代码时，可在存储区 2 执行编程操作。（或反之）。具体步骤如下：

1. 检查 *Flash 状态寄存器 (FLASH_SR)* 中的 BSY 位，以确认当前未执行任何 Flash 操作（在存储区 1 或存储区 2 执行擦除/编程操作时，BSY 有效）。
2. 将 *Flash 控制寄存器 (FLASH_CR)* 中的 PG 位置 1。
3. 针对所需地址存储器（主存储器块或 OTP 区域内）执行数据写入操作。
4. 等待 BSY 位清零（或使用 EOP 中断）。

3.4 FLASH 选项字节

3.4.1 选项字节说明

选项字节由最终用户根据具体的应用要求进行配置。以如下配置为例：可在硬件或软件模式下选择看门狗（参见*第 3.4.2 节：选项字节编程*）。

一个双字基于选项字节进行拆分，如下所示：

表 8. 选项字节格式

63-24	23-16	15 -8	7-0	31-24	23-16	15 -8	7-0
补码 选项字节 3	补码 选项字节 2	补码 选项字节 1	补码 选项字节 0	选项 字节 3	选项 字节 2	选项 字节 1	选项 字节 0

表 9 显示了这些字节在信息块中的构成。

选项字节可从*表 9* 列出的存储单元或从选项字节寄存器中读取：

- *Flash 选项寄存器 (FLASH_OPTR)*
- *Flash PCROP1 起始地址寄存器 (FLASH_PCROP1SR)*
- *Flash PCROP1 结束地址寄存器 (FLASH_PCROP1ER)*
- *Flash WRP1 区域 A 地址寄存器 (FLASH_WRP1AR)*
- *Flash WRP1 区域 B 地址寄存器 (FLASH_WRP1BR)*
- *Flash PCROP2 起始地址寄存器 (FLASH_PCROP2SR)*
- *Flash PCROP2 结束地址寄存器 (FLASH_PCROP2ER)*
- *Flash WRP2 区域 A 地址寄存器 (FLASH_WRP2AR)*
- *Flash WRP2 区域 B 地址寄存器 (FLASH_WRP2BR)*

表 9. 选项字节构成

存储区	地址	[63:56]	[55:48]	[47:40]	[39:32]	[31:24]	[23:16]	[15:8]	[7:0]
存储区 1	1FF00000	USER OPT			RDP	USER OPT			RDP
	1FF00008	未使用和 PCROP1_STRT[16]		PCROP1_STRT[15:0]		未使用和 PCROP1_STRT[16]		PCROP1_STRT[15]	
	1FF00010	PCROP_RDP、未使用和 PCROP1_END[16]		PCROP1_END[15:0]		PCROP_RDP、未使用和 PCROP1_END[16]		PCROP1_END[15:0]	
	1FF00018	未使用	WRP1A_END	未使用	WRP1A_STRT	未使用	WRP1A_END	未使用	WRP1A_STRT
	1FF00020	未使用	WRP2A_END	未使用	WRP2A_STRT	未使用	WRP2A_END	未使用	WRP2A_STRT
存储区 2	1FF01000	未使用							
	1FF01008	未使用和 PCROP2_STRT[16]		PCROP2_STRT[15]		未使用和 PCROP2_STRT[16]		PCROP2_STRT[15]	
	1FF01010	未使用和 PCROP2_END[16]		PCROP2_END[15]		未使用和 PCROP2_END[16]		PCROP2_END[15]	
	1FF01018	未使用	WRP1A_END	未使用	WRP1B_STRT	未使用	WRP1A_END	未使用	WRP1B_STRT
	1FF01020	未使用	WRP2A_END	未使用	WRP2B_STRT	未使用	WRP2A_END	未使用	WRP2B_STRT

用户和读保护选项字节

Flash 地址：0x1FF0 0000

ST 生产值：0xFFEF F8AA

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	nBOOT0	nSWBOOT0	SRAM2_RST	SRAM2_PE	nBOOT1	DBANK	DB1M	BFB2	WWDG_SW	IWGD_STDBY	IWDG_STOP	IWDG_SW
				r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	nRST_SHDW	nRST_STDBY	nRST_STOP	Res.	BOR_LEV[2:0]			RDP[7:0]							
	r	r	r		r	r	r	r	r	r	r	r	r	r	r

位 31:28 保留，必须保持复位值。

位 27 **nBOOT0**：nBOOT0 选项位 (nBOOT0 option bit)

0：nBOOT0 = 0

1：nBOOT0 = 1

位 26 **nSWBOOT0**：软件 BOOT0 (Software BOOT0)

0：BOOT0 取自选项位 nBOOT0

1：BOOT0 取自 PH3/BOOT0 引脚

位 25 **SRAM2_RST**：系统复位时进行 SRAM2 擦除 (SRAM2 Erase when system reset)
0：发生系统复位时进行 SRAM2 擦除
1：发生系统复位时不擦除 SRAM2

位 24 **SRAM2_PE**：SRAM2 奇偶校验使能 (SRAM2 parity check enable)
0：使能 SRAM2 奇偶校验
1：禁止 SRAM2 奇偶校验

位 23 **nBOOT1**：自举配置 (Boot configuration)
该位与 BOOT0 引脚搭配使用，用于选择从 Flash 主存储器、SRAM1 或系统存储器自举模式。请参见*第 2.6 节：自举配置*。

位 22 **DBANK**：
0：单存储区模式，128 位数据读取宽度
1：双存储区模式，64 位数据
该位只能在 PCROP1/2 已禁止时写入。

位 21 **DB1M**：1 MB Flash 或 512 KB Flash 器件上的双存储区 (Dual-Bank on 1 MB Flash or 512 KB Flash memory devices)
0：1 MB/512 KB 单存储区 Flash：存储区 1 中采用连续地址
1：1 MB/512 KB 双存储区 Flash，采用连续地址

位 20 **BFB2**：双存储区自举 (Dual-bank boot)
0：禁止双存储区自举
1：使能双存储区自举

位 19 **WWDG_SW**：窗口看门狗选择 (Window watchdog selection)
0：硬件窗口看门狗
1：软件窗口看门狗

位 18 **IWDG_STDBY**：在待机模式下冻结独立看门狗计数器 (Independent watchdog counter freeze in Standby mode)
0：在待机模式下冻结独立看门狗计数器
1：在待机模式下运行独立看门狗计数器

位 17 **IWDG_STOP**：在停止模式下冻结独立看门狗计数器 (Independent watchdog counter freeze in stop mode)
0：在停止模式下冻结独立看门狗计数器
1：在停止模式下运行独立看门狗计数器

位 16 **IDWG_SW**：独立看门狗选择 (Independent watchdog selection)
0：硬件独立看门狗
1：软件独立看门狗

位 15 保留，必须保持复位值。

位 14 **nRST_SHDW**：
0：进入关断模式时产生复位
1：进入关断模式时不产生复位

位 13 **nRST_STDBY**
0：进入待机模式时产生复位
1：进入待机模式时不产生复位

位 12 **nRST_STOP**

0：进入停机模式时产生复位
1：进入停机模式时不产生复位

位 11 保留，必须保持复位值。

位 10:8 **BOR_LEV**：BOR 复位级别 (BOR reset Level)

这些位包含激活/释放复位信号所需达到的 VDD 供电电压阈值。
000：BOR 0 级。复位阈值电压约 1.7 V
001：BOR 1 级。复位阈值电压约 2.0 V
010：BOR 2 级。复位阈值电压约 2.2 V
011：BOR 3 级。复位阈值电压约 2.5 V
100：BOR 4 级。复位阈值电压约 2.8 V

位 7:0 **RDP**：读保护级别 (Read protection level)

0xAA：级别 0，未激活读保护
0xCC：级别 2，激活芯片读保护
其他：级别 1，激活存储器读保护

PCROP1 起始地址选项字节

Flash 地址：0x1FF0 0008

ST 生产值：0xFFFF FFFF

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.	PCROP1_STRT[16:0]
															rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PCROP1_STRT[16:0]															
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 16:0 **PCROP1_STRT[16:0]**：PCROP 区域起始偏移 (PCROP area start offset)

DBANK=1
PCROP1_STRT 包含存储区 1 的 PCROP 区域的第一个双字。
DBANK=0
PCROP1_STRT 包含完整存储器的 PCROP 区域的第一个（2 x 双字）。

PCROP1 结束地址选项字节

Flash 地址：0x1FF0 0010

ST 生产值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PCROP_RDP	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PCROP1_END
rw															rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PCROP1_END[16:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31 **PCROP_RDP**：RDP 级别下降时，PCROP 区域保持 (PCROP area preserved when RDP level decreased)

该位只置 1。执行由 RDP 从级别 1 更改到级别 0 导致的完全批量擦除后，该位复位。

0：RDP 级别从级别 1 降低到级别 0 时，不擦除 PCROP 区域。

1：RDP 级别从级别 1 降低到级别 0 时，擦除 PCROP 区域（完全批量擦除）。

位 30:17 保留，必须保持复位值。

位 16:0 **PCROP1_END**：存储区 1 PCROP 区域结束偏移 (Bank 1 PCROP area end offset)

DBANK=1

PCROP1_END 包含存储区 1 PCROP 区域的最后一个双字。

DBANK=0

PCROP1_END 包含完整存储器的 PCROP 区域的最后一个（2 x 双字）。

WRP 区域 A 地址选项字节

Flash 地址：0x1FF0 0018

ST 生产值：0x0000 00FF

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WRP1A_END[7:0]							
								rw	rw	rw	rw	rw	rw	rw	rw

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

位 31:24 保留，必须保持复位值。

位 23:16 **WRP1A_END[7:0]**：WRP 第一个区域“A”结束偏移 (WRP first area “A” end offset)

DBANK=1

WRP1A_END 包含存储区 1 中 WRP 第一个区域的最后一页。

DBANK=0

WRP1A_END 包含完整存储器 WRP 第一个区域的最后一页。

位 15:8 保留，必须保持复位值。

位 7:0 **WRP1A_STRT[7:0]**：WRP 第一个区域“A”起始偏移 (WRP first area “A” start offset)
DBANK=1
WRP1A_STRT 包含存储区 1 中 WRP 第一个区域的第一页。
DBANK=0
WRP1A_STRT 包含完整存储器 WRP 第一个区域的第一页。

WRP1 区域 B 地址选项字节

Flash 地址：0x1FF0 0020

ST 生产值：0x0000 00FF

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WRP1B_END[7:0]							
								rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WRP1B_STRT[7:0]							
								rw	rw	rw	rw	rw	rw	rw	rw

位 31:24 保留，必须保持复位值。

位 23:16 **WRP1B_END[7:0]**：WRP 第二个区域“B”结束偏移 (WRP second area “B” end offset)
DBANK=1
WRP1B_END 包含存储区 1 中 WRP 第二个区域的最后一页。
DBANK=0
WRP1B_END 包含完整存储器 WPR 第二个区域的最后一页。

位 15:8 保留，必须保持复位值。

位 7:0 **WRP1B_STRT[7:0]**：WRP 第二个区域“B”起始偏移 (WRP second area “B” start offset)
DBANK=1
WRP1B_STRT 包含存储区 1 中 WRP 第二个区域的最后一页。
DBANK=0
WRP1B_STRT 包含完整存储器 WPR 第二个区域的最后一页。

PCROP2 起始地址选项字节

Flash 地址：0x1FF0 1008

ST 生产值：0xFFFF FFFF

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.	PCROP2_STRT[16:0]
															rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PCROP2_STRT[16:0]															
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 16:0 **PCROP2_STRT[16:0]**：PCROP 区域起始偏移 (PCROP area start offset)

DBANK=1

PCROP2_STRT 包含存储区 2 的 PCROP 区域的第一个双字。

DBANK=0

PCROP2_STRT 包含完整存储器的 PCROP 区域的第一个双字。

PCROP2 结束地址选项字节

Flash 地址：0x1FF0 1010

ST 生产值：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.	PCROP2_END[16:0]
															rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PCROP2_END[16:0]															
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 16:0 **PCROP2_END[16:0]**：PCROP 区域结束偏移 (PCROP area end offset)

DBANK=1

PCROP2_END 包含存储区 2 的 PCROP 区域的最后一个双字。

DBANK=0

PCROP2_END 包含完整存储器的 PCROP 区域的最后一个（2 x 双字）。

WRP2 区域 A 地址选项字节

Flash 地址：0x1FF0 1018

ST 生产值：0x0000 00FF

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WRP2A_END[23:16]							
								rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WRP2A_STRT[7:0]							
								rw	rw	rw	rw	rw	rw	rw	rw

位 31:24 保留，必须保持复位值。

位 23:16 **WRP2A_END**：WRP 第一个区域“A”结束偏移 (WRP first area “A” end offset)
DBANK=1
WRP2A_END 包含存储区 2 中 WRP 第一个区域的最后一页。
DBANK=0
WRP2A_END 包含完整存储器 WRP 第三个区域的最后一页。

位 15:8 保留，必须保持复位值。

位 7:0 **WRP2A_STRT**：WRP 第一个区域“A”起始偏移 (WRP first area “A” start offset)
DBANK=1
WRP2A_STRT 包含存储区 2 中 WRP 第一个区域的第一页。
DBANK=0
WRP2A_STRT 包含完整存储器 WRP 第三个区域的第一页。

WRP 区域 B 地址选项字节

Flash 地址：0x1FF0 1020

ST 生产值：0x0000 00FF

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WRP2B_END[23:16]							
								rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WRP2B_STRT[7:0]							
								rw	rw	rw	rw	rw	rw	rw	rw

位 31:24 保留，必须保持复位值。

位 23:16 **WRP2B_END**：WRP 第二个区域“B”结束偏移 (WRP second area “B” end offset)
DBANK=1
WRP2B_END 包含存储区 2 中 WRP 第二个区域的最后一页。
DBANK=0
WRP2B_END 包含完整存储器 WRP 第四个区域的最后一页。

位 15:8 保留，必须保持复位值。

位 7:0 **WRP2B_STRT**：WRP 第二个区域“B”起始偏移 (WRP second area “B” start offset)
DBANK=1
WRP2B_STRT 包含存储区 2 中 WRP 第二个区域的第一页。
DBANK=0
WRP2B_STRT 包含完整存储器 WRP 第二个区域的第一页。

3.4.2 选项字节编程

复位后，*Flash 控制寄存器 (FLASH_CR)* 中的选项相关位受到写保护。要针对选项字节页执行任何操作，*Flash 控制寄存器 (FLASH_CR)* 中的选项锁定位 OPTLOCK 必须清零。此寄存器的解锁顺序如下：

1. 使用 LOCK 清零序列解锁 FLASH_CR（参见*解锁 Flash*）。
2. 在 *Flash 选项密钥寄存器 (FLASH_OPTKEYR)* 中写入 OPTKEY1 = 0x08192A3B。
3. 在 FLASH_OPTKEYR 寄存器中写入 OPTKEY2 = 0x4C5D6E7F。

通过软件将 OPTLOCK 位置 1 后，可防止用户选项发生意外的擦除/编程操作。

注： *如果 LOCK 由软件置 1，则 OPTLOCK 也自动置 1。*

修改用户选项

选项字节与主存储器用户地址的编程方式不同。无法单独修改存储区 1 或存储区 2 的用户选项。首先修改存储区 1 的用户选项。

要修改用户选项值，请执行以下步骤：

1. 检查 *Flash 状态寄存器 (FLASH_SR)* 中的 BSY 位，以确认当前未执行任何 Flash 操作。
2. 按照上述清零序列将 OPTLOCK 选项锁定位清零。
3. 在以下选项寄存器中写入所需的选项值：*Flash 选项寄存器 (FLASH_OPTR)*、*Flash PCROP1 起始地址寄存器 (FLASH_PCROP1SR)*、*Flash PCROP1 结束地址寄存器 (FLASH_PCROP1ER)*、*Flash WRP1 区域 A 地址寄存器 (FLASH_WRP1AR)*、*Flash WRP1 区域 B 地址寄存器 (FLASH_WRP1BR)*、*Flash PCROP2 起始地址寄存器 (FLASH_PCROP2SR)*、*Flash PCROP2 结束地址寄存器 (FLASH_PCROP2ER)*、*Flash WRP2 区域 A 地址寄存器 (FLASH_WRP2AR)*、*Flash WRP2 区域 B 地址寄存器 (FLASH_WRP2BR)*。
4. 将 *Flash 控制寄存器 (FLASH_CR)* 中的选项启动位 OPTSTRT 置 1。
5. 等待 BSY 位清零。

注： *先擦除两个用户选项字节页（存储区 1 和存储区 2），然后以 Flash 选项寄存器中包含的值对所有选项字节进行编程，从而可自动修改一个选项值。*

选项字节加载

BSY 位清零后，所有新选项都将更新到 Flash，但不应用到系统。对其进行加载时将影响系统。在以下两种情况下执行选项字节加载 (OBL)：

- *Flash 控制寄存器 (FLASH_CR)* 中的 OBL_LAUNCH 位置 1 时。
- 电源复位后（BOR 复位或退出待机/关断模式）。

选项字节加载器读取选项块并将数据存储到内部选项寄存器。这些内部寄存器配置系统，无法由软件读取。将 OBL_LAUNCH 置 1 会产生复位，因此在系统复位下执行选项字节加载。

每个选项位在同一双字中也有各自的补码。选项加载期间，验证选项位及其补码可检查是否已正确进行加载。

选项字节加载期间，选项通过具有 ECC 的双字读取。如果字与其补码匹配，则将选项字/字节复制到选项寄存器。

如果字与其补码不匹配，则 OPTVERR 状态位置 1。不匹配值被强制放入选项寄存器：

- 对于 USR OPT 选项，不匹配值为所有选项均为“1”，为“000”的 BOR_LEV（最低阈值）除外
- 对于 WRP 选项，不匹配值为默认值“无保护”
- 对于 RDP 选项，不匹配值为默认值“级别 1”
- 对于 PCROP，不匹配值为“所有存储器均受保护”

在系统复位（上升沿）时，将内部选项寄存器复制到可由软件读写的选项寄存器中（FLASH_OPTR、FLASH_PCROP1/2SR、FLASH_PCROP1/2ER、FLASH_WRP1/2AR、FLASH_WRP1/2BR）。上述寄存器也用于修改选项。如果这些寄存器未由用户修改，则会反映系统的选项状态。请参见*修改用户选项*一节以获得更多信息。

激活双存储区模式（从 DBANK=0 切换到 DBANK=1）

从一种 Flash 模式切换到另一种 Flash 模式（例如从单存储区切换到双存储区）时，建议通过 SRAM 执行代码或使用自举程序。为避免在更改存储器构成时，从 Flash 读取损坏的数据，重新编程前应避免（CPU 或 DMA）访问 Flash。

- 如果指令/数据缓存和/或预取已使能，则将其禁止（复位 FLASH_ACR 寄存器中的 PRFTEN 和 ICEN/DCEN 位）。
- 通过将 FLASH_ACR 寄存器中的 DCRST/ICRST 位置 1，刷新指令和数据缓存。
- 将 DBANK 选项位置 1，清除全部 WRP 写保护（按照用户选项修改和选项字节加载器过程）。
  - 基于 DBANK=0 完成 OBL 后，执行批量擦除。
  - 基于 DBANK=0 存储器映射，在 64 位模式下开始执行新的代码编程。
  - 必要时，基于 DBANK=0 方案设置新的 WRP/PCROP。
  - 如有需要，将 PRFTEN 和 ICEN/DCEN 置 1。

新软件已准备好使用存储区配置来运行。

取消激活双存储区模式（从 DBANK=1 切换到 DBANK=0）

从一种 Flash 模式切换到另一种 Flash 模式（例如从单存储区切换到双存储区）时，建议通过 SRAM 执行代码或使用自举程序。为避免在更改存储器构成时，从 Flash 读取损坏的数据，重新编程前应避免（CPU 或 DMA）访问 Flash。

- 如果指令/数据缓存和/或预取已使能，则将其禁止（复位 FLASH_ACR 寄存器中的 PRFTEN 和 ICEN/DCEN 位）。
- 通过将 FLASH_ACR 寄存器中的 DCRST/ICRST 位置 1，刷新指令和数据缓存。
- 将 DBANK 选项位清零，清除全部 WRP 写保护（按照用户选项修改和选项字节加载器过程）。
  - 基于 DBANK=0 完成 OBL 后，执行批量擦除。
  - 基于 DBANK=0 存储器映射，在 128 位模式下开始执行新的代码编程。
  - 必要时，基于 DBANK=0 方案设置新的 WRP/PCROP。如有需要，将 PRFTEN 和 ICEN/DCEN 置 1。

新软件已准备好使用存储区配置来运行。

3.5 Flash 保护

可对 Flash 主存储器进行保护，使其具有读保护 (RDP)，以防其遭受外部访问。也可对 Flash 页进行保护，以防因程序指针错乱而发生意外的写操作。写保护 (WRP) 粒度为一页 (2 KB)。还可防止 Flash 遭受第三方的读取和写入 (PCROP)。PCROP 粒度为双字（64 位）。

3.5.1 读保护 (RDP)

通过将 RDP 选项字节置 1，然后应用系统复位来重载新的 RDP 选项字节，可激活读保护。读保护可对 Flash 主存储器、选项字节、备份寄存器（RTC 中的 RTC_BKPxR）和 SRAM2 进行保护。

注： *如果在调试器仍通过 JTAG/SWD 连接时设置读保护，则应用 POR（上电复位），而非系统复位。*

有三种读保护级别：无保护（级别 0）到最大保护或禁止调试（级别 2）。

RDP 选项字节及其补码包含成对值时，Flash 受保护，如*表 10* 所示。

表 10. Flash 读保护状态

RDP 字节值	RDP 补码值	读保护级别
0xAA	0x55	级别 0
除 0xAA 或 0xCC 外的任意值	除 0x55 和 0x33 外的任意值（不一定是补码值）	级别 1（默认值）
0xCC	0x33	级别 2

无论保护级别如何，系统存储器区域均可进行读访问。对于编程/擦除操作，则不可访问。

级别 0：无保护

可对 Flash 主存储器区域执行读取、编程和擦除操作。也可对选项字节、SRAM2 和备份寄存器进行所有操作。

级别 1：读保护

RDP 选项字节被擦除时，此为默认保护级别。RDP 值是除 0xAA 和 0xCC 外的任意值时，或者即使补码不正确，也会定义此级别。

- **用户模式：**在用户模式（**自举 Flash**）下执行的代码可访问 Flash 主存储器、选项字节、SRAM2 和备份寄存器，可对其进行所有操作。
- **调试、自举 RAM 和自举程序模式：**在调试模式下或从自举 RAM 或自举程序运行代码时，Flash 主存储器、备份寄存器（RTC 中的 RTC_BKPxR）和 SRAM2 均不可访问。在上述模式下，对 Flash 进行读访问或写访问会生成总线错误和硬性故障中断。

注意： 如果配置了级别 1 且未定义 PCROP 区域，则必须将 PCROP_RDP 位置 1（RDP 级别从级别 1 降到级别 0 时，进行完全批量擦除）。如果配置了级别 1 且定义了 PCROP 区域，则在需要通过 RDP（而非 PCROP）保护用户代码时，不得将用户代码置于包含 PCROP 区域的页。

级别 2：禁止调试

在此级别下，可保证保护级别 1。此外，Cortex®-M4 调试端口、从 RAM 自举（自举 RAM 模式）和从系统存储器自举（自举程序模式）不再可用。在用户执行模式（自举 FLASH 模式）下，允许对 Flash 主存储器执行所有操作。相反，只能对选项字节执行读取操作。

无法对选项字节进行编程或擦除。因此，无法删除级别 2：此级别不可更改。尝试修改选项字节时，Flash_SR 寄存器中的保护错误标志 WRPERR 置 1，并且可能会生成中断。

注： *在复位时调试功能也被禁止。*

意法半导体无法对设为保护级别 2 的器件做失效分析。

更改读保护级别

通过将 RDP 字节的值更改为除 0xCC 外的任意值，可轻松从级别 0 更改为级别 1。通过在 RDP 字节中编程 0xCC 值，可直接从级别 0 或级别 1 更改为级别 2。更改为级别 2 后，无法再更改读保护级别。

将 RDP 编程为值 0xAA 来从级别 1 更改为级别 0 时，如果 *Flash PCROP1 结束地址寄存器 (FLASH_PCROP1ER)* 中的 PCROP_RDP 置 1，则会对 Flash 主存储器执行批量擦除。也会擦除备份寄存器（RTC 中的 RTC_BKPxR）和 SRAM2。除 PCROP 保护外的用户选项设置为之前从 FLASH_OPTR 和 FLASH_WRPxyR（x=1、2 且 y =A 或 B）复制的值。PCROP 被禁止。OTP 区域不受批量擦除操作的影响，同样保持不变。

如果 FLASH_PCROP1ER 中的 PCROP_RDP 位被清零，则完全批量擦除由部分批量擦除所代替，该部分页面擦除是对 PCROP 激活的存储区内进行连续页面擦除（PCROP 保护的页面除外）。这样做是为了保持 PCROP 代码。对于两个存储区，如果 PCROP 均激活，则会通过页擦除将这两个存储区擦除。只有两个存储区均被擦除后，才会使用之前的选项值对选项重新编程。这同样适用于 FLASH_PCROPxSR 和 FLASH_PCROPxER 寄存器（x = 1、2）。

注： *只有在已激活级别 1 并请求级别 0 时，才会执行完全批量擦除或部分批量擦除。当提高保护级别 (0->1、1->2、0->2) 时，不会执行批量擦除。*

为验证保护级别更改，必须通过 Flash 控制寄存器中的 OBL_LAUNCH 位重载选项字节。

图 4. 更改读保护 (RDP) 级别

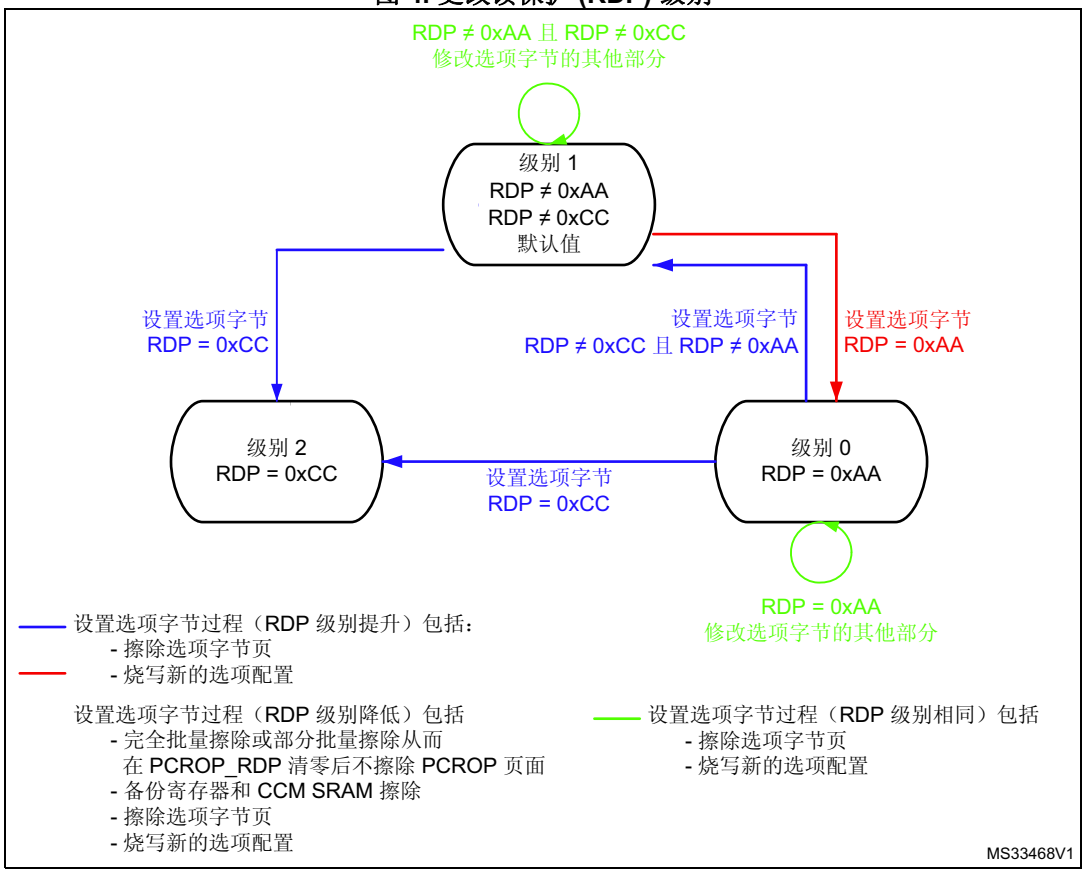


表 11. 访问状态 vs 保护级别和执行模式

区域	保护级别	用户执行（从 Flash 自举）			调试/从 RAM 自举/从加载程序自举(1)		
		读	写	擦除	读	写	擦除
Flash 主存储器	1	有	有	有	无	无	无(3)
	2	有	有	有	N/A	N/A	N/A
系统存储器 (2)	1	有	无	无	有	无	无
	2	有	无	无	N/A	N/A	N/A
选项字节	1	有	有(3)	有	有	有(3)	有
	2	有	无	无	N/A	N/A	N/A

表 11. 访问状态 vs 保护级别和执行模式（续）

区域	保护级别	用户执行（从 Flash 自举）			调试/从 RAM 自举/从加载程序自举(1)		
		读	写	擦除	读	写	擦除
OTP	1	有	有(4)	N/A	无	无	N/A
	2	有	有(4)	N/A	N/A	N/A	N/A
备份寄存器	1	有	有	N/A	无	无	无(5)
	2	有	有	N/A	N/A	N/A	N/A
SRAM2	1	有	有	N/A	无	无	无(6)
	2	有	有	N/A	N/A	N/A	N/A

1. 级别 2 激活后，将禁止调试端口、从 RAM 自举和从系统存储器自举。
2. 无论保护级别（0、1 或 2）和执行模式如何，都只能对系统存储器进行读访问。
3. 在禁止所有级别保护 (0xAA) 的情况下编程 RDP 选项字节时，会擦除 Flash 主存储器。
4. OTP 只能写入一次。
5. 当 RDP 从级别 1 更改为级别 0 时擦除备份寄存器。
6. 当 RDP 从级别 1 更改为级别 0 时擦除 SRAM2。

3.5.2 专有代码读保护 (PCROP)

可防止 Flash 遭受第三方的读取和写入。受保护区域为仅执行区域：只能由 STM32 CPU 访问（作为指令代码），所有其他访问（DMA、调试和 CPU 数据读取、写入和擦除）被严格禁止。可在双存储区模式下为每个存储区指定一个 PCROP 区域，或为完整存储器指定两个 PCROP 区域，具体取决于 DBANK 模式。另有一个选项位 (PCROP_RDP) 可用来选择在 RDP 保护从级别 1 变为级别 0 时 PCROP 区域是否被擦除（参见*更改读保护级别*）。

每个 PCROP 区域由与物理 Flash 存储区基址相关的起始页偏移和结束页偏移定义。上述偏移在以下 PCROP 地址寄存器中定义：*Flash PCROP1 起始地址寄存器 (FLASH_PCROP1SR)*、*Flash PCROP1 结束地址寄存器 (FLASH_PCROP1ER)*、*Flash PCROP2 起始地址寄存器 (FLASH_PCROP2SR)*、*Flash PCROP2 结束地址寄存器 (FLASH_PCROP2ER)*。

在单存储区模式下 (DBANK=0)：

- PCROPx（x = 1、2）区域定义为从地址（基址 + [PCROPx_STRT x 16]（包含））到地址（基址 + [(PCROPx_END+1) x 16]（不包含））。PCROP 区域最小为两个 2 x 双字（256 位）。

在双存储区模式下 (DBANK=1)：

- PCROPx（x = 1、2）区域定义为从地址（存储区“x”基址 + [PCROPx_STRT x 0x8]（包含））到地址（存储区“x”基址 + [(PCROPx_END+1) x 0x8]（不包含））。PCROP 区域最小为两个双字（128 位）。

例如，通过从地址 0x0806 2F80（包含）到地址 0x0807 0004（包含）的 PCROP 进行保护：

- 如果在存储区 1 中完成 Flash 自举，则必须使用以下内容对 FLASH_PCROP1SR 和 FLASH_PCROP1ER 寄存器进行编程：
  - PCROP1_STRT = 0xC5F0。
  - PCROP1_END = 0xE000。
- 如果将两个存储区交换，则必须对存储区 2 应用保护，并且必须使用以下内容对 FLASH_PCROP2SR 和 FLASH_PCROP2ER 寄存器进行编程：
  - PCROP2_STRT = 0xC5F0。
  - PCROP2_END = 0xE000。

通过 D 总线对 PCROP 保护区域执行读访问时，将触发 RDERR 标志错误。

任何 PCROP 保护地址同时也受写保护，对其中任一地址执行任何写访问都将触发 WRPERR。

任何 PCROP 区域也受擦除保护。因此，无法对此区域的页进行擦除（包括含有该区域起始地址和结束地址的页）。此外，如果某区域受 PCROP 保护，则无法执行软件批量擦除操作。

对于上述示例，出于按页擦除的原因，进行页擦除时从 0x62 到 0x70 的所有页都受保护。（无法擦除从 0x0806 2000 到 0x080 70FFF 的所有地址）。

只有当 RDP 从级别 1 切换到级别 0 时，PCROP 才会无效。如果用户选项修改尝试清除 PCROP 或缩小 PCROP 区域，则会启动选项编程，但 PCROP 区域保持不变。相反，可能会增大 PCROP 区域。

选项位 PCROP_RDP 清零，且 RDP 从级别 1 切换到级别 0 时，为保持 PCROP 区域，完全批量擦除由部分批量擦除所代替（参见*更改读保护级别*）。在这种情况下，也不会擦除 PCROP1/2_STRT 和 PCROP1/2_END。

注： *建议在使用 PCROP_RDP 时将 PCROP 区域与页面大小匹配，或留出 PCROP 区域起始或结束页的剩余部分。*

表 12. PCROP 保护(1)

PCROPx 寄存器值（x = 1、2）	PCROP 保护区
PCROPx_offset_strt > PCROPx_offset_end	无 PCROP 区域。
PCROPx_offset_strt < PCROPx_offset_end	PCROPx_offset_strt 和 PCROPx_offset_end 之间的区域受保护。 可： – 用较低值写入 PCROPx_offset_strt。 – 用较高值写入 PCROPx_offset_end。

1. DBANK=1 时，PCROP 区域最小为 2 个双字：PCROPx_offset_strt 和 PCROPx_offset_end。
   DBANK=0 时，PCROP 区域最小为两个（2 x 双字）：PCROPx_offset_strt 和 PCROPx_offset_end。
   DBANK=1 时，用户负责确保 PCROP 区域不发生重叠。

3.5.3 写保护 (WRP)

可对 Flash 中的用户区域实施保护，以防其遭受意外的写操作。

根据 DBANK 选项位配置，可指定：

- 在单存储区模式 (DBANK=0) 下：可在每个存储区内定义四个写保护 (WRP) 区域，页面大小为 8 KB。
- 在双存储区模式 (DBANK=1) 下：可在每个存储区内定义两个写保护 (WRP) 区域，页面大小为 4 KB。

每个区域由与物理 Flash 存储区基址相关的起始页偏移和结束页偏移定义。上述偏移在以下 WRP 地址寄存器中定义：*Flash WRP1 区域 A 地址寄存器 (FLASH_WRP1AR)*、*Flash WRP1 区域 B 地址寄存器 (FLASH_WRP1BR)*、*Flash WRP2 区域 A 地址寄存器 (FLASH_WRP2AR)*、*Flash WRP2 区域 B 地址寄存器 (FLASH_WRP2BR)*。

双存储区模式 (DBANK=1)

存储区“x”WRP“y”区域（x = 1、2 且 y = A、B）定义为从地址（存储区“x”基址 + [WRPxy_STRT x 0x1000]（包含））到地址（存储区“x”基址 + [(WRPxy_END+1) x 0x1000]（不包含））。

单存储区模式 (DBANK=0)

WRPx“y”区域（x = 1、2 且 y = A、B）定义为从地址（基址 + [WRPy_STRT x 0x2000]（包含））到地址（基址 + [(WRPy_END+1) x 0x2000]（不包含））。

例如，通过从地址 0x0806 2800（包含）到地址 0x0807 07FF（包含）的 WRP 进行保护：

- 如果在存储区 1 中完成 Flash 自举，则必须使用以下内容对 FLASH_WRP1AR 寄存器进行编程：
  - WRP1A_STRT = 0x62。
  - WRP1A_END = 0x70。

  也可使用 FLASH_WRP1BR 中的 WRP1B_STRT 和 WRP1B_END（存储区 1 中的区域“B”）。
- 如果将两个存储区交换，则必须对存储区 2 应用保护，并且必须使用以下内容对 FLASH_WRP2AR 寄存器进行编程：
  - WRP2A_STRT = 0x62。
  - WRP2A_END = 0x70。
  - WRP2A_STRT = 0xC5。
  - WRP2A_END = 0xE0。

  也可使用 FLASH_WRP2BR 中的 WRP2B_STRT 和 WRP2B_END（存储区 2 中的区域“B”）。

WRP 有效时，无法对其执行擦除或编程操作。因此，如果某区域受写保护，则无法执行软件批量擦除操作。

如果尝试对 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 (WRPERR) 将置 1。对以下区域进行写访问时，此标志也会置 1：

- Flash 中永远不能执行写操作的 OTP 区域部分（例如 ICP）
- PCROP 区域。

注： *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 或系统 Flash 执行自举代码，则即使未激活 WRP，也无法对 Flash 执行编程或擦除操作。*

注： *为验证 WRP 选项，必须通过 Flash 控制寄存器中的 OBL_LAUNCH 位重载选项字节。*

注： *DBANK=0 时，用户负责确保 WRP 区域不发生重叠。*

表 13. WRP 保护

WRP 寄存器值 （x=1/2，y= A/B）	WRP 保护区
WRPxy_STRT = WRPxy_END	页 WRPxy 受保护。
WRPxy_STRT >
WRPxy_END	无 WRP 区域。
WRPxy_STRT <
WRPxy_END	– 从 WRPxy_STRT 到 WRPxy_END 的页受保护。

3.6 Flash 中断

表 14. Flash 中断请求

中断事件	事件标志	事件标志/中断 清除方法	中断使能控制位
操作结束	EOP(1)	写入 EOP=1	EOPIE
操作错误	OPERR(2)	写入 OPERR=1	ERRIE
读取错误	RDERR	写入 RDERR=1	RDERRIE
ECC 校正	ECCC	写入 ECCC=1	ECCCIE

1. 仅当 EOPIE 置 1 后，EOP 才会置 1。
2. 仅当 ERRIE 置 1 后，OPERR 才会置 1。

3.7 FLASH 寄存器

3.7.1 Flash 访问控制寄存器 (FLASH_ACR)

Flash access control register

偏移地址：0x00

复位值：0x0000 0600

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
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.	SLEEP_PD	RUN_PD	DCRST	ICRST	DCEN	ICEN	PRFTEN	Res.	Res.	Res.	Res.	LATENCY [3:0			
	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 **SLEEP_PD**：睡眠或低功耗睡眠模式下 Flash 处于掉电模式 (Flash Power-down mode during Sleep or Low-power sleep mode)

该位确定当器件处于睡眠或低功耗睡眠模式时，Flash 是处于掉电模式还是空闲模式。

0：在睡眠和低功耗睡眠模式下，Flash 处于空闲模式

1：在睡眠和低功耗睡眠模式下，Flash 处于掉电模式

注意： 执行编程或擦除操作时，Flash 不得处于掉电模式。

位 13 **RUN_PD**：运行或低功耗运行模式下 Flash 处于掉电模式 (Flash Power-down mode during Run or Low-power run mode)

该位受 FLASH_PDKEYR 写保护。

该位确定当器件处于运行或低功耗运行模式时，Flash 是处于掉电模式还是空闲模式。只有从 RAM 执行代码时，Flash 才可处于掉电模式。RUN_PD 置 1 时，不得访问 Flash。

0：Flash 处于空闲模式

1：Flash 处于掉电模式

注意： 执行编程或擦除操作时，Flash 不得处于掉电模式。

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

只有在禁止数据缓存后才能写入该位。

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

只有在禁止指令缓存后才能写入该位。

位 10 **DCEN**：数据缓存使能 (Data cache enable)

0：禁止数据缓存

1：使能数据缓存

位 9 **ICEN**：指令缓存使能 (Instruction cache enable)
0：禁止指令缓存
1：使能指令缓存

位 8 **PRFTEN**：预取使能 (Prefetch enable)
0：禁止预取
1：使能预取

位 7:4 保留，必须保持复位值。

位 3:0 **LATENCY[3:0]**：延迟 (Latency)
这些位表示 SYSCLK（系统时钟）周期与 Flash 访问时间之比。
0000：0 个等待状态
0001：1 个等待状态
0010：2 个等待状态
0011：3 个等待状态
0100：4 个等待状态
...1111：15 个等待状态

3.7.2 Flash 掉电密钥寄存器 (FLASH_PDKEYR)

Flash Power-down key register

偏移地址：0x04

复位值：0x0000 0000

访问：无等待状态，按字访问

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

位 31:0 **PDKEYR**：Flash 运行模式下掉电密钥 (Power-down in Run mode Flash key)
要将 FLASH_ACR 中的 RUN_PD 位解锁，必须顺序写入以下值：
PDKEY1：0x0415 2637
PDKEY2：0xFAFB FCFD

3.7.3 Flash 密钥寄存器 (FLASH_KEYR)

Flash key register

偏移地址：0x08

复位值：0x0000 0000

访问：无等待状态，按字访问

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

位 31:0 **KEYR**：Flash 密钥 (Flash key)

要将 FLASH_CR 寄存器解锁以允许执行 Flash 编程/擦除操作，必须顺序写入以下值：

KEY1：0x4567 0123

KEY2：0xCDEF 89AB

3.7.4 Flash 选项密钥寄存器 (FLASH_OPTKEYR)

Flash option key register

偏移地址：0x0C

复位值：0x0000 0000

访问：无等待状态，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 FLACH_OPTR 寄存器解锁以允许执行选项字节编程/擦除操作，必须顺序写入以下值：

KEY1：0x0819 2A3B

KEY2：0x4C5D 6E7F

3.7.5 Flash 状态寄存器 (FLASH_SR)

Flash status 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
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.		BSY
															r

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OPTV ERR	RD ERR	Res.	Res.	Res.	Res.	FAST ERR	MISS ERR	PGS ERR	SIZ ERR	PGA ERR	WRP ERR	PROG ERR	Res.	OP ERR	EOP
rc_w1	rc_w1					rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1		rc_w1	rc_w1

位 31:18 保留，必须保持复位值。

位 17 **PEMPTY**：程序为空 (Program EMPTY)

如果 Flash 未编程且用户要从主 Flash 自举，则在上电复位时或执行 OBL_LAUNCH 命令后此位由硬件置 1。如果 Flash 已编程且用户要从主 Flash 自举，则在上电复位时或执行 OBL_LAUNCH 命令后此位由硬件清零。该位还可由软件置 1 和清零。

1：该位值正在翻转

0：无影响

如果 Flash 编程（已选择在主 Flash 中自举）后，由软件实现 OBL_LAUNCH，则可将该位置 1，来将“程序为空”位清零。最终会强制在主 Flash 中自举，无需断开调试器连接。

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复位。

位 15 **OPTVERR**：选项有效性错误 (Option validity error)

读取的选项可能不是用户配置的选项时，该位由硬件置 1。如果选项尚未正确加载，则在每次系统复位后，OPTVERR 将再次置 1。

写入 1 即可将该位清零。

位 14 **RDERR**：PCROP 读取错误 (PCROP read error)

通过 D 总线读取的地址属于 Flash 的读保护区域（PCROP 保护）时，此位由硬件置 1。如果 FLASH_CR 中的 RDERRIE 置 1，将生成中断。

写入 1 即可将该位清零。

位 13:10 保留，必须保持复位值。

位 9 **FASTERR**：快速编程错误 (Fast programming error)

因错误（对齐、大小、写保护或数据丢失）导致快速编程顺序（由 FSTPG 激活）中断时，此位由硬件置 1。同时，相应状态位（PGAERR、SIZERR、WRPERR 或 MISSERR）也会置 1。

写入 1 即可将该位清零。

位 8 **MISERR**：快速编程数据丢失错误 (Fast programming data miss error)

在快速编程模式下，必须将 32 个双字连续发送到 Flash，并且必须在当前数据完全编程之前将新数据发送到 Flash 逻辑控制。如果新数据未及时出现，则 MISSERR 由硬件置 1。

写入 1 即可将该位清零。

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

如果代码在 PG 或 FSTPG 先前未置 1 的情况下对 Flash 执行写访问，则该位由硬件置 1。由于之前的编程错误而导致 PROGERR、SIZERR、PGAERR、WRPERR、MISSERR 或 FASTERR 置 1 时，该位也由硬件置 1。

DBANK=0 的情况下，尝试执行存储区擦除时，该位也置 1。

写入 1 即可将该位清零。

位 6 **SIZERR**：大小错误 (Size error)

在编程或快速编程顺序中，访问大小为字节或半字时，该位由硬件置 1。只允许双字编程（即按字访问）。

写入 1 即可将该位清零。

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

在标准编程期间要编程的数据无法包含在同一 64 位 Flash 行中，或快速编程时存在页面更改，该位由硬件置 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 中受写保护（受 WRP、PCROP 或 RDP 等级 1 保护）的区域，则该位由硬件置 1。

写入 1 即可将该位清零。

位 3 **PROGERR**：编程错误 (Programming error)

编程前，要编程的双字地址包含不同于“0xFFFF FFFF”的值时，该位由硬件置 1（要写入的数据为“0x0000 0000”时除外）。

写入 1 即可将该位清零。

位 2 保留，必须保持复位值。

位 1 **OPERR**：操作错误 (Operation error)

当 Flash 操作（编程/擦除）失败时，该位由硬件置 1。

只有在使能错误中断 (ERRIE = 1) 后，该位才会置 1。

写入“1”即可将该位清零。

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，该位由硬件置 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。

3.7.6 Flash 控制寄存器 (FLASH_CR)

Flash control register

偏移地址：0x14

复位值：0xC000 0000

访问：当前未执行任何 Flash 操作时无等待状态，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	OPT LOCK	Res.	Res.	OBL_ LAUNCH	RD ERRIE	ERR IE	EOP IE	Res.	Res.	Res.	Res.	Res.	FSTPG	OPT STRT	STRT
rs	rs			rc_w1	rw	rw	rw						rw	rs	rs
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MER2	Res.	Res.	Res.	BKER	PNB[7:0]								MER1	PER	PG
rw				rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31 **LOCK**：FLASH_CR 锁定 (FLASH_CR Lock)

该位只置 1。置 1 后，FLASH_CR 寄存器被锁定。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，直到下一次系统复位。

位 30 **OPTLOCK**：选项锁定 (Options Lock)

该位只置 1。置 1 后，FLASH_CR 寄存器中所有与用户选项相关的位以及选项页都被锁定。当检测到解锁序列时，由硬件将该位清零。对 OPTLOCK 位执行解锁序列之前，必须将 LOCK 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 29:28 保留，必须保持复位值。

位 27 **OBL_LAUNCH**：强制选项字节加载 (Force the option byte loading)

此位置 1 时，会强制进行选项字节重载。只有在选项字节加载完成时此位才会清零。如果 OPTLOCK 置 1，则此位无法写入。

0：选项字节加载完成

1：已请求选项字节加载

位 26 **RDERRIE**：PCROP 读取错误中断使能 (PCROP read error interrupt enable)

当 FLASH_SR 中的 RDERR 位置 1 后，可通过该位使能中断产生功能。

0：禁止 PCROP 读取错误中断

1：使能 PCROP 读取错误中断

位 25 **ERRIE**：错误中断使能 (Error interrupt enable)

当 FLASH_SR 中的 OPERR 位置 1 后，可通过该位使能中断产生功能。

0：禁止 OPERR 错误中断

1：使能 OPERR 错误中断

位 24 **EOPIE**：操作结束中断使能 (End of operation interrupt enable)

当 FLASH_SR 中的 EOP 位置 1 后，可通过该位使能中断产生功能。

0：禁止 EOP 中断

1：使能 EOP 中断

位 23:19 保留，必须保持复位值

位 18 **FSTPG**：快速编程 (Fast programming)
0：禁止快速编程
1：使能快速编程

位 17 **OPTSTRT**：启动选项修改 (Options modification start)
该位置 1 后可触发选项操作。
该位只能通过软件置 1，并在 FLASH_SR 中的 BSY 位清零后随之清零。

位 16 **START**：启动 (Start)
该位置 1 后可触发擦除操作。如果 MER1、MER2 和 PER 位均复位并且 STRT 位置 1，可能会发生无法预知的行为而不会生成任何错误标志。应禁止此情况发生。
该位只能通过软件置 1，并在 FLASH_SR 中的 BSY 位清零后随之清零。

位 15 **MER2**：存储区 2 批量擦除 (Bank 2 Mass erase)
该位置 1 时，会触发存储区 2 批量擦除（所有存储区 2 用户页）。

位 14:12 保留，必须保持复位值。

位 11 **BKER**：存储区擦除 (Bank erase)
DBANK=1
0：已选择存储区 1 进行页擦除
1：已选择存储区 2 进行页擦除
DBANK=0
保留，必须保持清零

位 10:3 **PNB[7:0]**：页编号选择 (Page number selection)
这些位用于选择要擦除的页：
00000000：第 0 页
00000001：第 1 页
...
11111111： 第 255 页

位 2 **MER1**：存储区 1 批量擦除 (Bank 1 Mass erase)
该位置 1 时，会触发存储区 1 批量擦除（所有存储区 1 用户页）。

位 1 **PER**：页擦除 (Page erase)
0：禁止页擦除
1：使能页擦除

位 0 **PG**：编程 (Programming)
0：禁止 Flash 编程
1：使能 Flash 编程

3.7.7 Flash ECC 寄存器 (FLASH_ECCR)

Flash ECC register

偏移地址：0x18

复位值：0x0000 0000

访问：当前未执行任何 Flash 操作时无等待状态，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
ECCD	ECCC	ECCD2	ECCD2	Res.	Res.	Res.	ECCCIE	Res.	SYSF_ECC	BK_ECC	ADDR_ECC[20:16]				
rc_w1	rc_w1	rc_w1	rc_w1				rw		r	r	r	r	r	r	r

ADDR_ECC[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31 **ECCD**：ECC 检测 (ECC detection)

DBANK=0

检测到两个 ECC 错误时，该位由硬件置 1（仅限 ECCC/ECCC2/ECCD/ECCD2 之前已清零的情况）。该位置 1 时，会生成 NMI。

写入 1 即可将该位清零。

DBANK=1

在 64 位 LSB（位 63:0）检测到两个 ECC 错误时，该位由硬件置 1（仅限 ECCC/ECCC2/ECCD/ECCD2 之前已清零的情况）。该位置 1 时，会生成 NMI。

写入 1 即可将该位清零。

位 30 **ECCC**：ECC 校正 (ECC correction)

检测到并校正一个 ECC 错误时，该位由硬件置 1（仅限 ECCC/ECCC2/ECCD/ECCD2 之前已清零的情况）。如果 ECCIE 置 1，则会生成中断。

写入 1 即可将该位清零。

位 29 **ECCD2**：ECC2 检测 (ECC2 detection)

DBANK=0

在 64 位 MSB（位 127:64）上检测到两个 ECC 错误时，该位由硬件置 1。仅限 ECCC/ECCC2/ECCD/ECCD2 之前已清零的情况下，该位才置 1。该位置 1 时，会生成 NMI。

写入 1 即可将该位清零。

DBANK=1

保留，必须保持复位值。

位 28 **ECCC2**：ECC 校正 (ECC correction)

DBANK=0

在 64 位 MSB（位 127:64）上检测到一个 ECC 错误并对其进行校正时，该位由硬件置 1。仅限 ECCC/ECCC2/ECCD/ECCD2 之前已清零的情况下，该位才置 1。如果 ECCIE 置 1，则会生成中断。

写入 1 即可将该位清零。

DBANK=1

保留，必须保持复位值。

位 27:25 保留，必须保持复位值。

位 24 **ECCIE**：ECC 校正中断使能 (ECC correction interrupt enable)

0：禁止 ECCC 中断

1：使能 ECCC 中断

DBANK=0

当 FLASH_ECCR 寄存器中的 ECCC 或 ECCC2 位置 1 后，可通过该位使能中断产生功能。

DBANK=1

当 FLASH_ECCR 寄存器中的 ECCC 位置 1 后，可通过该位使能中断产生功能。

位 23 保留，必须保持复位值。

位 22 **SYSF_ECC**：系统 Flash ECC 失效 (System Flash ECC fail)

该位指示位于系统 Flash 中的 ECC 错误校正或双重 ECC 错误检测。

位 21 **BK_ECC**：ECC 失效存储区 (ECC fail bank)

DBANK=1

该位指示 ECC 错误校正或双重 ECC 错误检测涉及的存储区。

0：存储区 1

1：存储区 2

DBANK=0

如果 SYSF_ECC 为 1，指示 ECC 错误涉及的存储区。

如果 SYSF_ECC 为 0，则保留，必须保持清零。

位 19 **BK_ECC**：ECC 失效存储区 (ECC fail bank)

该位指示 ECC 错误校正或双重 ECC 错误检测涉及的存储区。

0：存储区 1

1：存储区 2

位 20:0 **ADDR_ECC**：ECC 失效地址 (ECC fail address)

DBANK=0

该位指示 Flash 中涉及 ECC 错误校正或双重 ECC 错误检测的地址。

DBANK=1

该位指示存储区中涉及 ECC 错误校正或双重 ECC 错误检测的地址。

3.7.8 Flash 选项寄存器 (FLASH_OPTR)

Flash option register

偏移地址：0x20

复位值：0xFFEF F8AA。OBL 时，将 Flash 中的值加载到寄存器位。

访问：当前未执行任何 Flash 操作时无等待状态；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
Res.	Res.	Res.	Res.	nBOOT0	nSWBOOT0	SRAM2_RST	SRAM2_PE	nBOOT1	DBANK	DB1M	BFB2	WWDG_SW	IWGD_STDBY	IWDG_StOP	IWDG_SW
				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.	nRST_SHDW	nRST_STDBY	nRST_STOP	Res.	BOR_LEV[2:0]			RDP[7:0]							
	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 **nBOOT0**：nBOOT0 选项位 (nBOOT0 option bit)

0：nBOOT0 = 0

1：nBOOT0 = 1

位 26 **nSWBOOT0**：软件 BOOT0 (Software BOOT0)

0：BOOT0 取自选项位 nBOOT0

1：BOOT0 取自 PH3/BOOT0 引脚

位 25 **SRAM2_RST**：系统复位时进行 SRAM2 擦除 (SRAM2 Erase when system reset)

0：发生系统复位时进行 SRAM2 擦除

1：发生系统复位时不擦除 SRAM2

位 24 **SRAM2_PE**：SRAM2 奇偶校验使能 (SRAM2 parity check enable)

0：使能 SRAM2 奇偶校验

1：禁止 SRAM2 奇偶校验

位 23 **nBOOT1**：自举配置 (Boot configuration)

该位与 BOOT0 引脚搭配使用，用于选择从 Flash 主存储器、SRAM1 或系统存储器自举模式。请参见*第 2.6 节：自举配置*。

位 22 **DBANK**：

0：单存储区模式，128 位数据读取宽度

1：双存储区模式，64 位数据

只有在禁止 PCROPA/B 时才能写入此位。

位 21 **DB1M**：1 MB Flash 或 512 KB Flash 器件上的双存储区 (Dual-Bank on 1 MB Flash or 512 KB Flash memory devices)

0：1 MB 或 512 KB 单存储区 Flash：存储区 1 中采用连续地址。

1：1 MB/512 KB 双存储区 Flash，采用连续地址。

DB1M 置 1 后，请求的地址超过 1 MB 或 512 KB 时，将生成硬性故障。

位 20 **BFB2**：双存储区自举 (Dual-bank boot)

0：禁止双存储区自举

1：使能双存储区自举

位 19 **WWDG_SW**：窗口看门狗选择 (Window watchdog selection)
0：硬件窗口看门狗
1：软件窗口看门狗

位 18 **IWDG_STDBY**：在待机模式下冻结独立看门狗计数器 (Independent watchdog counter freeze in Standby mode)
0：在待机模式下冻结独立看门狗计数器
1：在待机模式下运行独立看门狗计数器

位 17 **IWDG_STOP**：在停止模式下冻结独立看门狗计数器 (Independent watchdog counter freeze in stop mode)
0：在停止模式下冻结独立看门狗计数器
1：在停止模式下运行独立看门狗计数器

位 16 **IDWG_SW**：独立看门狗选择 (Independent watchdog selection)
0：硬件独立看门狗
1：软件独立看门狗

位 15 保留，必须保持清零

位 14 **nRST_SHDW**
0：进入关断模式时产生复位
1：进入关断模式时不产生复位

位 13 **nRST_STDBY**
0：进入待机模式时产生复位
1：进入待机模式时不产生复位

位 12 **nRST_STOP**
0：进入停机模式时产生复位
1：进入停机模式时不产生复位

位 11 保留，必须保持清零

位 10:8 **BOR_LEV**：BOR 复位级别 (BOR reset Level)
这些位包含激活/释放复位信号所需达到的 VDD 供电电压阈值。
000：BOR 0 级。复位阈值电压约 1.7 V
001：BOR 1 级。复位阈值电压约 2.0 V
010：BOR 2 级。复位阈值电压约 2.2 V
011：BOR 3 级。复位阈值电压约 2.5 V
100：BOR 4 级。复位阈值电压约 2.8 V

位 7:0 **RDP**：读保护级别 (Read protection level)
0xAA：级别 0，未激活读保护
0xCC：级别 2，激活芯片读保护
其他：级别 1，激活存储器读保护

注：　请注意级别 1 下的 PCROP_RDP 配置。更多详细信息，请参见级别 1：读保护一节。

3.7.9 Flash PCROP1 起始地址寄存器 (FLASH_PCROP1SR)

Flash PCROP1 Start address register

偏移地址：0x24

复位值：0xFFFF FFFF。OBL 时，将 Flash 中的值加载到寄存器位。

访问：当前未执行任何 Flash 操作时无等待状态；按字访问。

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.	PCROP1_STRT
															rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PCROP1_STRT[16:0]															
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

位 16:0 **PCROP1_STRT**：PCROP 区域起始偏移 (PCROP area start offset)

DBANK=1

PCROP1_STRT 包含存储区 1 的 PCROP 区域的第一个双字。

DBANK=0

PCROP1_STRT 包含完整存储器的 PCROP 区域的第一个（2 x 双字）。

3.7.10 Flash PCROP1 结束地址寄存器 (FLASH_PCROP1ER)

Flash PCROP1 End address register

偏移地址：0x28

复位值：0xXFFX XXXX。OBL 时，将 Flash 中的值加载到寄存器位。

访问：当前未执行任何 Flash 操作时无等待状态；按字、半字访问。可通过字节访问的形式访问 PCROP_RDP 位。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PCROP_RDP	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PCROP1_END
rs															rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PCROP1_END[16:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31 **PCROP_RDP**：RDP 级别下降时，PCROP 区域保持 (PCROP area preserved when RDP level decreased)

该位只置 1。执行由 RDP 从级别 1 更改到级别 0 导致的完全批量擦除后，该位复位。

0：RDP 级别从级别 1 降低到级别 0 时，不擦除 PCROP 区域。

1：RDP 级别从级别 1 降低到级别 0 时，擦除 PCROP 区域（完全批量擦除）。

位 30:17 保留，必须保持清零

位 16:0 **PCROP1_END**：存储区 1 PCROP 区域结束偏移 (Bank 1 PCROP area end offset)

DBANK=1

PCROP1_END 包含存储区 1 PCROP 区域的最后一个双字。

DBANK=0

PCROP1_END 包含完整存储器的 PCROP 区域的最后一个（2 x 双字）。

3.7.11 Flash WRP1 区域 A 地址寄存器 (FLASH_WRP1AR)

Flash WRP1 area A address register

偏移地址：0x2C

复位值：0xFFXX FFXX。OBL 时，将 Flash 中的值加载到寄存器位。

访问：当前未执行任何 Flash 操作时无等待状态；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
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WRP1A_END[7:0]							
								rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WRP1A_STRT[7:0]							
								rw	rw	rw	rw	rw	rw	rw	rw

位 31:24 保留，必须保持清零

位 23:16 **WRP1A_END**：WRP 第一个区域“A”结束偏移 (WRP first area “A” end offset)

DBANK=1

WRP1A_END 包含存储区 1 中 WRP 第一个区域的最后一页。

DBANK=0

WRP1A_END 包含完整存储器 WRP 第一个区域的最后一页。

位 15:8 保留，必须保持清零

位 7:0 **WRP1A_STRT**：WRP 第一个区域“A”起始偏移 (WRP first area “A” start offset)

DBANK=1

WRP1A_STRT 包含存储区 1 中 WRP 第一个区域的第一页。

DBANK=0

WRP1A_STRT 包含完整存储器 WRP 第一个区域的第一页。

3.7.12 Flash WRP1 区域 B 地址寄存器 (FLASH_WRP1BR)

Flash WRP1 area B address register

偏移地址：0x30

复位值：0xFFXX FFXX。OBL 时，将 Flash 中的值加载到寄存器位。

访问：当前未执行任何 Flash 操作时无等待状态；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
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WRP1B_END[7:0]							
								rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WRP1B_STRT[7:0]							
								rw	rw	rw	rw	rw	rw	rw	rw

位 31:24 保留，必须保持清零

位 23:16 **WRP1B_END**：WRP 第二个区域“B”结束偏移 (WRP second area “B” end offset)
DBANK=1
WRP1B_END 包含存储区 1 中 WRP 第二个区域的最后一页。
DBANK=0
WRP1B_END 包含完整存储器 WPR 第二个区域的最后一页。

位 15:8 保留，必须保持清零

位 7:0 **WRP1B_STRT**：WRP 第二个区域“B”起始偏移 (WRP second area “B” start offset)
DBANK=1
WRP1B_STRT 包含存储区 1 中 WRP 第二个区域的最后一页。
DBANK=0
WRP1B_STRT 包含完整存储器 WPR 第二个区域的最后一页。

3.7.13 Flash PCROP2 起始地址寄存器 (FLASH_PCROP2SR)

Flash PCROP2 Start address register

偏移地址：0x44

复位值：0xFFFF FFFF

访问：当前未执行任何 Flash 操作时无等待状态；按字访问。

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.	PCROP2_STRT
															rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PCROP2_STRT[16:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:17 保留，必须保持清零

位 16:0 **PCROP2_STRT**：PCROP 区域起始偏移 (PCROP area start offset)
DBANK=1
PCROP2_STRT 包含存储区 2 的 PCROP 区域的第一个双字。
DBANK=0
PCROP2_STRT 包含完整存储器的 PCROP 区域的第一个双字。

3.7.14 Flash PCROP2 结束地址寄存器 (FLASH_PCROP2ER)

Flash PCROP2 End address register

偏移地址：0x48

复位值：0xXFFX XXXX

访问：当前未执行任何 Flash 操作时无等待状态；按字访问

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.	PCROP2_END
															rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PCROP2_END[16:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:17 保留，必须保持清零

位 23:16 **PCROP2_END**：PCROP 区域结束偏移 (PCROP area end offset)
DBANK=1
PCROP2_END 包含存储区 2 的 PCROP 区域的最后一个双字。
DBANK=0
PCROP2_END 包含完整存储器的 PCROP 区域的最后一个（2 x 双字）。

3.7.15 Flash WRP2 区域 A 地址寄存器 (FLASH_WRP2AR)

Flash WRP2 area A address register

偏移地址：0x4C

复位值：0xFFXX FFXX

访问：当前未执行任何 Flash 操作时无等待状态；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
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WRP2A_END[7:0]							
								rw	rw	rw	rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WRP2A_STRT[7:0]							
								rw	rw	rw	rw	rw	rw	rw	rw

位 31:24 保留，必须保持清零

位 23:16 **WRP2A_END**：WRP 第一个区域“A”结束偏移 (WRP first area “A” end offset)
DBANK=1
WRP2A_END 包含存储区 2 中 WRP 第一个区域的最后一页。
DBANK=0
WRP2A_END 包含完整存储器 WRP 第三个区域的最后一页。

位 15:8 保留，必须保持清零

位 7:0 **WRP2A_STRT**：WRP 第一个区域“A”起始偏移 (WRP first area “A” start offset)
DBANK=1
WRP2A_STRT 包含存储区 2 中 WRP 第一个区域的第一页。
DBANK=0
WRP2A_STRT 包含完整存储器 WRP 第三个区域的第一页。

3.7.16 Flash WRP2 区域 B 地址寄存器 (FLASH_WRP2BR)

Flash WRP2 area B address register

偏移地址：0x50

复位值：0xFFXX FFXX

访问：当前未执行任何 Flash 操作时无等待状态；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
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WRP2B_END[7:0]							
								rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WRP2B_STRT[7:0]							
								rw	rw	rw	rw	rw	rw	rw	rw

位 31:24 保留，必须保持清零

位 23:16 **WRP2B_END**：WRP 第二个区域“B”结束偏移 (WRP second area “B” end offset)
DBANK=1
WRP2B_END 包含存储区 2 中 WRP 第二个区域的最后一页。
DBANK=0
WRP2B_END 包含完整存储器 WRP 第四个区域的最后一页。

位 15:8 保留，必须保持清零

位 23:16 **WRP2B_STRT**：WRP 第二个区域“B”起始偏移 (WRP second area “B” start offset)
DBANK=1
WRP2B_STRT 包含存储区 2 中 WRP 第二个区域的第一页。
DBANK=0
WRP2B_STRT 包含完整存储器 WRP 第二个区域的第一页。

3.7.17 Flash 配置寄存器 (FLASH_CFGR)

Flash configuration register

偏移地址：0x130

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.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LVEN
															rw

位 31:1 保留，必须保持清零

位 0 **LVEN**：低压使能 (Low voltage enable)

此位由软件置 1 和清零。该位必须仅用于外部 SMPS。

0：禁止 Flash 低压。

1：使能 Flash 低压。将该位置 1 前，建议：

- 将电压调节范围切换到范围 2。
- 确保最低 VDD12 为 1.08 V。
- 外部 SMPS 开启时，可将 LVEN 位置 1。

3.7.18 FLASH 寄存器映射

表 15. Flash 接口——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	FLASH_ACR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SLEEP_PD	RUN_PD	DCRST	ICRST	DCEN	ICEN	PRFTEN	Res.	Res.	Res.	Res.	LATENCY [3:0]			
	Reset value																		0	0	0	0	1	1	0					0	0	0	0
0x04	FLASH_ PDKEYR	PDKEYR[31:0]																															
	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_KEYR	KEYR[31:0]																															
	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_OPT KEYR	OPTKEYR[31:0]																															
	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	FLASH_SR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PEMPTY	BSY	OPTVERR	RDERR	Res.	Res.	Res.	Res.	FASTERR	MISERR	PGSERR	SIZERR	PGAERR	WRPERR	PROGERR	Res.	OPERR	EOP
	Reset value															X	0	0	0					0	0	0	0	0	0	0		0	0
0x14	FLASH_CR	LOCK	OPTLOCK	Res.	Res.	OBL_LAUNCH	RDERRIE	ERRIE	EOPIE	Res.	Res.	Res.	Res.	Res.	FSTPG	OPTSTRT	STRT	MER2	Res.	Res.	Res.	BKER	PNB[7:0]								MER1	PER	PG
	Reset value	1	1			0	0	0	0						0	0	0	0				0	0	0	0	0	0	0	0	0	0	0	0
0x18	FLASH_ECCR	ECCD	ECCC	ECCD2	ECCC2	Res.	Res.	Res.	ECCCIE	Res.	SYSF_ECC	BK_ECC	ADDR_ECC[20:0]																				
	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
0x20	FLASH_OPTR	Res.	Res.	Res.	Res.	nBOOT0	nSWBOOT0	SRAM2_RST	SRAM2_PE	nBOOT1	DBANK	DB1M	BFB2	WWDG_SW	IWDG_STBY	IWDG_STOP	IWDG_SW	Res.	nRST_SHDW	nRST_STDBY	nRST_STOP	Res.	BOR_ LEV[2:0]			RDP[7:0]							
	Reset value					X	X	X	X	X	X	X	X	X	X	X	X		X	X	X		X	X	X	X	X	X	X	X	X	X	X
0x24	FLASH_ PCROP1SR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PCROP1_STRT[16:0]																
	Reset value																1	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
0x28	FLASH_ PCROP1ER	PCROP_RDP	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PCROP1_END[16:0]																
	Reset value	x															0	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
0x2C	FLASH_ WRP1AR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WRP1A_END[7:0]								Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WRP1A_STRT[7:0]							
	Reset value									X	X	X	X	X	X	X	X									X	X	X	X	X	X	X	X

表 15. Flash 接口——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
0x30	FLASH_ WRP1BR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WRP1B_END[7:0]								Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WRP1B_STRT[7:0]							
	Reset value									X	X	X	X	X	X	X	X									X	X	X	X	X	X	X	X
0x44	FLASH_ PCROP2SR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PCROP2_STRT[16:0]																
	Reset value																1	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
0x48	FLASH_ PCROP2ER	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PCROP2_END[16:0]																
	Reset value																0	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
0x4C	FLASH_ WRP2AR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WRP2A_END[7:0]								Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WRP2A_STRT[7:0]							
	Reset value									X	X	X	X	X	X	X	X									X	X	X	X	X	X	X	X
0x50	FLASH_ WRP2BR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WRP2B_END[7:0]								Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WRP2B_STRT[7:0]							
	Reset value									X	X	X	X	X	X	X	X									X	X	X	X	X	X	X	X
0x130	FLASH_CFGR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LVEN
	Reset value																																0

有关寄存器边界地址的信息，请参见*第 78 页的第 2.2.2 节*。

4 防火墙 (FW)

4.1 简介

防火墙用于保护非易失性存储器上的特殊代码或数据，和/或用于隔离 SRAM 1 上的易失性数据与其余在保护区域外执行的代码。

4.2 防火墙主要特性

- 防火墙保护的代码（代码段）可能位于以下区域：
  - Flash 映射
  - SRAM 1 存储器，在防火墙配置阶段须将此存储器声明为可执行保护区
- 被保护数据可位于以下区域：
  - Flash（非易失性数据段）
  - SRAM 1 存储器（易失性数据段）

防火墙打开后，软件可以访问上述保护区。通过使用一种基于“调用门”的机制可打开或关闭防火墙（请参见*打开防火墙*）。

在使能防火墙之前，必须先配置各数据段的起始地址及相应长度（请参见*第 4.3.5 节：防火墙初始化*）。

若防火墙已使能，则任何对这些保护段的非法访问均会产生复位，从而立即终止非法访问。

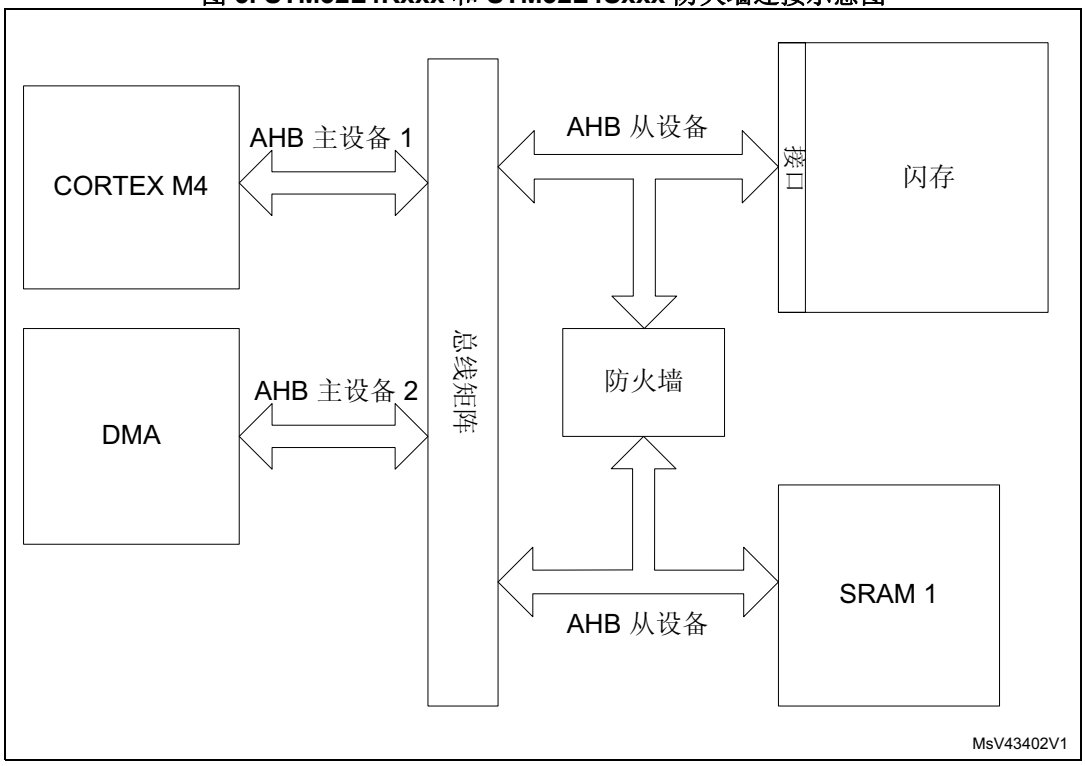
无论防火墙是否开启，任何对保护段的 DMA 访问均不允许。此类访问将视为非法访问并产生复位。

4.3 防火墙功能说明

4.3.1 防火墙 AMBA 总线监听

防火墙外设始终会监听存储器（易失性和非易失性）所连接的 AMBA 总线状态。整体架构图如*图 5* 所示。

图 5. STM32L4Rxxx 和 STM32L4Sxxx 防火墙连接示意图



4.3.2 功能要求

为保证防火墙所保护的应用程序代码和数据获得最高级安全等级，需要考虑以下若干事项，以防止产生不必要的防火墙报警（即产生复位）。

调试注意事项

在调试模式中，即使防火墙已开启，调试器仍可访问保护段。为此，在使用防火墙的同时，还需将读保护级别设置为 2 级。

若需要调试，则可按照如下方式进行操作：

- 在最终用户代码开发阶段期间，可开发伪代码，伪代码的 API 必须与被保护代码相同。由于被保护代码最终将应用在实际产品上，因此伪代码在功能实现上和时序要求（必要的话）上应与被保护代码保持一致。
- 在开发阶段，若签订了 NDA 协议，则被保护代码可交给用户端使用，相应软件可以在 0 保护等级下开发。如果产品的 ROP 等级为 2 级，为方便更新代码，需要在用户最终代码的写保护段中嵌入 IAP。

写保护

为提供最高安全等级，须遵守如下说明：

- 必须对使能防火墙的代码部分进行写保护。该激活代码应位于防火墙保护的代码段外。
- 还必须对防火墙保护的代码段进行写保护。
- 必须对包含复位向量的页进行写保护。

中断管理

防火墙保护的代码在执行期间不应接收任何中断请求。这取决于用户在执行被保护代码前是否禁止了所有中断源。如果不遵守此要求，则在被保护代码执行期间发现中断请求（防火墙开启的情况下）时，一旦执行中断子程序，防火墙就会立即关闭。在程序返回到被保护代区域时，由于未应用“调用门”序列，将生成防火墙报警，从而产生复位信号。

有关 Flash 中的中断向量以及第一个用户页的信息如下：

- 如果第一个用户页（包含复位向量）受防火墙保护，则应在被保护段以外区域对此 NVIC 向量重新编程。
- 如果第一个用户页未受防火墙保护，则可将此中断向量放在此位置。

防火墙不产生任何中断。

4.3.3 防火墙段

防火墙用于保护以下三种不同段区域：

代码段

该段位于 Flash 存储器中。其中应包含在执行时需由防火墙保护的代码。在访问此段时，须使用“调用门”序列打开防火墙。如果通过系统配置寄存器的 FWDIS 位使能了防火墙，而打开防火墙的“调用门”入口序列是错误的（请参见*打开防火墙*），将产生系统复位。使能防火墙前，必须先配置段长度和基地址（请参见*第 4.3.5 节：防火墙初始化*）。

非易失性数据段

该段用于存储防护墙保护的代码段所使用的非易失性数据。在*第 4.3.4 节：段访问方式及其属性*中对该段的访问方式进行了说明。在访问该区域的数据前，必须先打开防火墙。非易失性数据段应位于 Flash 存储器中。使能防火墙前，必须先配置段长度和基地址（请参见*第 4.3.5 节：防火墙初始化*）。

易失性数据段

被保护代码所使用的易失性数据必须存储在 SRAM 1 存储器中。在*第 4.3.4 节：段访问方式及其属性*中对该段的访问方式进行了说明。通过配置易失性数据段，可决定在访问该段之前是否必须开启防火墙。使能防火墙前，必须先配置段长度和基地址以及段选项（请参见*第 4.3.5 节：防火墙初始化*）。

此外，通过设置防火墙寄存器中的两个位，可将易失性数据段定义为可执行段（用于执行代码）或共享段，即位 VDE 用于定义易失性数据段是否为可执行段，位 VDS 用于定义易失性数据段是否为共享段。有关详细信息，请参见*表 16*。

4.3.4 段访问方式及其属性

禁止对被保护段进行任何形式的 DMA 访问，无论防火墙处于何种状态，这种访问都会引起系统复位。

不同防火墙状态下的段访问

上述三种段的访问属性各不相同，相关信息请参见*表 16*。

表 16. 不同防火墙状态下的段访问

段	防火墙开启 允许的访问	防火墙关闭 允许的访问	防火墙禁用 允许的访问
代码段	读取和执行	禁止任何访问。 对段进行任何访问（“调用门”条目除外）均会引起系统复位	允许各种访问操作（取决于代码所在 Flash 页的保护特性）
非易失性数据段	读取和写入	禁止任何访问	允许各种访问操作（取决于代码所在 Flash 页的保护特性）
易失性数据段	读取和写入 防火墙配置寄存器的 VDE = 1 且 VDS = 0 时，可进行执行访问	防火墙配置寄存器的 VDS = 0 且 VDE = 0 时，禁止任何访问 VDS = 1 时，无论 VDE 位的值为何，可进行读取/写入/执行访问 若 VDE = 1 且 VDS = 0，但先使用“调用门”条目打开防火墙，则可进行执行访问。	允许各种访问操作

易失性数据段的情况与其他两段的情况略有差异。易失性数据段可以是：

- 共享段（寄存器中 VDS 位）

  这意味着该段内的区域以及数据可在被保护代码和用户代码（在非保护区内执行）之间进行共享。访问共享段不受防火墙限制。

  在这种情况下，VDS 位的优先级高于 VDE 位，VDE 位的值被忽略。这表示易失性数据段可直接执行位于该段的代码，无需在执行前开启防火墙。

- 执行段

  当 FW_CR 寄存器中的 VDS = 0 时，将考虑 VDE 位。若 VDS = 1，请参见上述有关易失性数据共享段的描述。若 VDS = 0 且 VDE = 1，则易失性数据段为可执行段。为避免防火墙引起系统复位，应对易失性数据段应用“调用门”序列打开防火墙，将此序列作为代码执行的入口点。

段属性

系统为各个段分配了特定寄存器来定义防火墙所保护段的大小：CSL 为代码段长度寄存器，NVDSL 为非易失性数据段长度寄存器，VDSL 为易失性数据段长度寄存器。有关上述段的粒度和区域范围信息，请参见*表 17*。

表 17. 段粒度和区域范围

段	粒度	区域范围
代码段	256 字节	2048 KB 到 256 字节
非易失性数据段	256 字节	2048 KB 到 256 字节
易失性数据段	64 字节	192 KB 到 64 字节

4.3.5 防火墙初始化

防火墙初始化应在执行用户代码之前进行（请参见*写保护*）。

在初始化阶段，需要设置防火墙所保护段的地址和长度。由于防火墙使能位的值只能写入一次，因此在使能防火墙前完成此设置。于是，防火墙使能后，在下一次系统复位前将无法禁止。

防火墙使能后，将不能访问段地址和长度。所有写操作均视为无效操作。

若段长度定义为 0，则该段将不受防火墙保护。这时，对段基地址进行任何操作都不会使防火墙产生复位。

默认情况下，在复位后会禁止防火墙（SYSCFG 寄存器中 FWDIS 位置位）。必须清除此位方可使能防火墙功能。

防火墙初始化过程如下所示：

1. 配置 RCC 以使能防火墙模块的时钟
2. 配置 RCC 以使能系统配置寄存器的时钟
3. 设置各段的基地址和长度（CSSA、CSL、NVDSSA、NVDSL、VDSSA 和 VDSL 寄存器）
4. 设置防火墙配置寄存器（FW_CR 寄存器）
5. 通过清除系统配置寄存器的 FWDIS 位使能防火墙

防火墙配置寄存器（FW_CR 寄存器）是唯一可以采用动态方式进行管理的寄存器（即使防火墙已使能）：

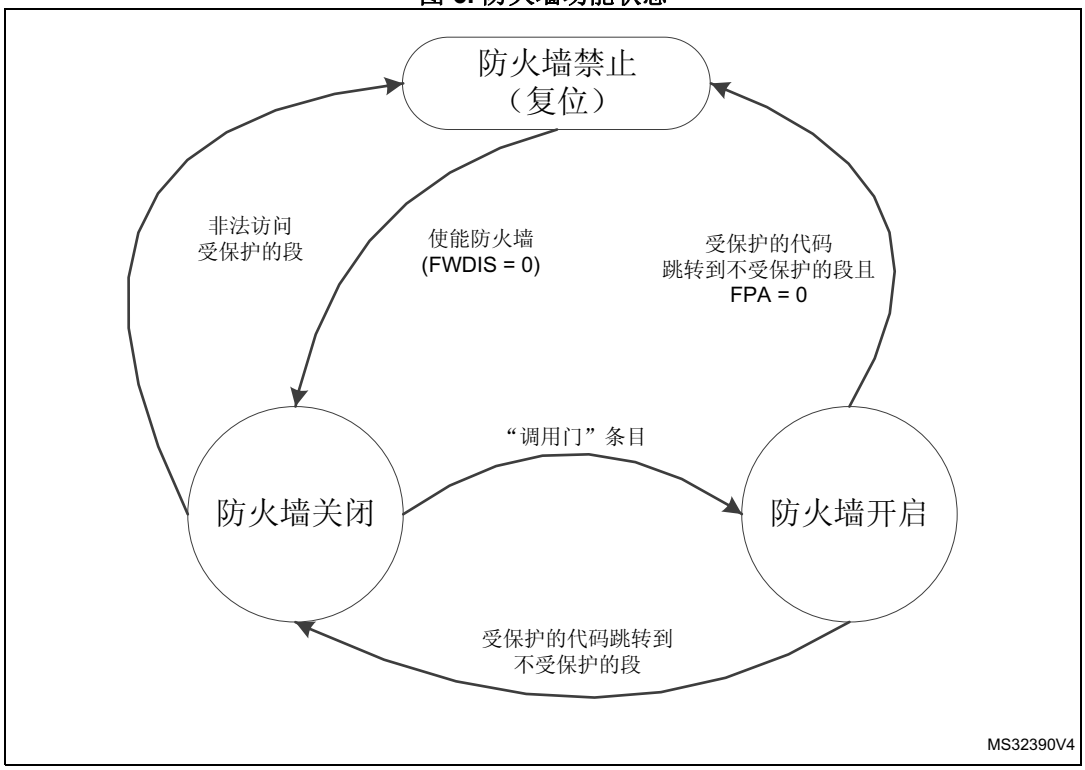
- 若未定义非易失性数据段（NVDSL 寄存器为 0），则无论防火墙是否打开，均可访问该寄存器。
- 若已定义非易失性数据段（NVDSL 寄存器为 1），则只有在防火墙打开的情况下，方可访问该寄存器。

4.3.6 防火墙状态

防火墙的状态分为三种，如*图 6* 所示：

- 禁止：复位后 FWDIS 位默认会置位。防火墙未激活。
- 关闭：防火墙限制对三类段（代码段、非易失性数据段和易失性数据段）的访问。
- 打开：防火墙允许对被保护段进行访问，如*第 4.3.4 节：段访问方式及其属性*所述。

图 6. 防火墙功能状态



打开防火墙

防火墙使能后，立即关闭。这表示对被保护段的大部分访问方式被禁止（请参见*第 4.3.4 节：段访问方式及其属性*）。为打开防火墙以与被保护段进行交互，必须采用以下所述的“调用门”序列。

“调用门”序列

“调用门”序列由 3 个字组成，它占据代码段和声明为执行段 (VDE = 1) 但非共享段 (VDS = 0) 的易失性数据段之基地址的头三个 32 位地址空间。

– 第一个字：空 32 位字，此字不对外开放，目的是防止预取缓冲器访问此字从而导致“调用门”机制开启。

– 第二个和第三个字：这两个特定的 32 位字即为“调用门”，始终处于开放状态。

要打开防火墙，当前执行的代码必须跳转至“调用门”的第二个字，并从该点执行代码。任何中间指令取操作均不得中断第二个字和第三个字的执行过程，否则防火墙不视为打开状态，而是回到关闭状态。若在接收到中间指令取操作后执行第三个字，则将产生系统复位。

防火墙打开后，可按照*第 4.3.4 节：段访问方式及其属性*中的描述访问被保护段。

关闭防火墙

防火墙在使能后会立即关闭（清除系统配置寄存器的 FWDIS 位）。

要关闭防火墙，受保护代码必须：

- 将正确的“防火墙预武装标志”值写入 FW_CR 寄存器中。
- 跳转至防火墙段以外的任何可执行位置。

如果未设置“防火墙预武装标志”位，则在被保护段跳转至非保护段时将产生复位。该控制位属于附加保护方法，可防止任何非法访问在私人信息未清除的情况下关闭防火墙，详情请见下文说明。

出于安全性考虑，建议在执行使用防火墙功能的应用程序后，清理 CPU 寄存器和硬件单元中的私人信息。

4.4 防火墙寄存器

4.4.1 代码段起始地址寄存器 (FW_CSSA)

Code segment start address

偏移地址：0x00

复位值：0x0000 0000

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

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

位 31:24 保留，必须保持复位值。

位 23:8 **ADD[23:8]：**代码段起始地址 (Code segment start address)

起始地址的 LSB 位（位 7:0）为保留位，被强制清零以满足 256 字节粒度的要求。

注： 这些位只能在使能防火墙之前写入。请参见第 4.3.5 节：防火墙初始化。

位 7:0 保留，必须保持复位值。

4.4.2 代码段长度寄存器 (FW_CSL)

Code segment length

偏移地址：0x04

复位值：0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LENG15:8]								Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
rw															

位 31:22 保留，必须保持复位值。

位 21:8 **LENG[21:8]**：代码段长度 (Code segment length)

LENG[21:8] 设置代码段的大小，以字节表示，且与 256 字节成倍数关系。

段区域定义为从 {ADD[23:8]，0x00} 到 {ADD[23:8]+LENG[21:8], 0x00} - 0x01

注： 如果 LENG[21:8] = 0，则使能防火墙后，将视该段为未定义段，因而不受防火墙保护。

这些位只能在使能防火墙之前写入。请参见第 4.3.5 节：防火墙初始化。

位 7:0 保留，必须保持复位值。

4.4.3 非易失性数据段起始地址寄存器 (FW_NVDSSA)

Non-volatile data segment start address

偏移地址：0x08

复位值：0x0000 0000

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

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

位 31:24 保留，必须保持复位值。

位 23:8 **ADD[23:8]**：非易失性数据段起始地址 (Non-volatile data segment start address)

起始地址的 LSB 位（位 7:0）为保留位，被强制清零以满足 256 字节粒度的要求。

注： 这些位只能在使能防火墙之前写入。请参见第 4.3.5 节：防火墙初始化。

位 7:0 保留，必须保持复位值。

4.4.4 非易失性数据段长度寄存器 (FW_NVDSL)

Non-volatile data segment length

偏移地址：0x0C

复位值：0x0000 0000

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

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

位 31:22 保留，必须保持复位值。

位 21:8 **LENG[21:8]**：非易失性数据段长度 (Non-volatile data segment length)

LENG[21:8] 设置非易失性数据段的大小，以字节表示，且与 256 字节成倍数关系。

段区域定义为从 {ADD[23:8]，0x00} 到 {ADD[23:8]+LENG[21:8], 0x00} - 0x01

注： 如果 LENG[21:8] = 0，则使能防火墙后，将视该段为未定义段，因而不受防火墙保护。

这些位只能在使能防火墙之前写入。请参见第 4.3.5 节：防火墙初始化。

位 7:0 保留，必须保持复位值。

4.4.5 易失性数据段起始地址寄存器 (FW_VDSSA)

Volatile data segment start address

偏移地址：0x10

复位值：0x0000 0000

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

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

位 31:18 保留，必须保持复位值。

位 17 **ADD[17]**：易失性数据段起始地址 (Volatile data segment start address)

位 16:6 **ADD[16:6]**：易失性数据段起始地址 (Volatile data segment start address)

起始地址的 LSB 位（位 5:0）为保留位，被强制清零以满足 64 字节粒度的要求。

注： 这些位只能在使能防火墙之前写入。请参见第 4.3.5 节：防火墙初始化。

位 5:0 保留，必须保持复位值。

4.4.6 易失性数据段长度寄存器 (FW_VDSL)

Volatile data segment length

偏移地址：0x14

复位值：0x0000 0000

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

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

位 31:18 保留，必须保持复位值。

位 17 **LENG[17]**：易失性数据段长度 (volatile data segment length)

位 16:6 **LENG[16:6]**：易失性数据段长度 (volatile data segment length)

LENG[16:6] 设置易失性数据段的大小，以字节表示，且与 64 字节成倍数关系。

段区域定义为从 {ADD[16:6]，0x00} 到 {ADD[16:6]+LENG[16:6], 0x00} - 0x01

注： 如果 LENG[17:6] = 0，则使能防火墙后，将视该段为未定义段，因而不受防火墙保护。这些位只能在使能防火墙之前写入。请参见第 4.3.5 节：防火墙初始化。

位 5:0 保留，必须保持复位值。

4.4.7 配置寄存器 (FW_CR)

Configuration 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
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.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	VDE	VDS	FPA
													rw	rw	rw

位 31:3 保留，必须保持复位值。

位 2 **VDE**：易失性数据可执行 (Volatile data execution)

0：若 VDS = 0，则无法执行易失性数据段

1：无论 VDS 位为何值，易失性数据段均声明为可执行段

若 VDS = 1，则该位无意义。此时无论 VDE 位为何值，均可执行易失性数据段。

若 VDS = 1，则无论防火墙是否开启，均可执行代码。

若 VDS = 0，则仅当防火墙打开，或者在防火墙关闭情况下应用了“调用门”序列，才可执行代码。

请参见*不同防火墙状态下的段访问*。

位 1 **VDS**：易失性数据可共享 (Volatile data shared)

0：易失性数据段不对外共享，在防火墙关闭时，不会遭受来自非保护可执行代码的攻击。在这种情况下访问该段，防火墙将产生系统复位。

1：可与非保护应用程序代码共享易失性数据段。无论防火墙是否开启，均可访问该段。

请参见*不同防火墙状态下的段访问*。

位 0 **FPA**：防火墙预武装 (Firewall prearm)

0：如果防火墙已打开，则在被保护段以外执行任何代码都将产生系统复位。

1：在被保护段以外执行任何代码都将关闭防火墙。

请参见*关闭防火墙*。

该寄存器受保护的方式与易失性数据段相同（请参见*第 4.3.5 节：防火墙初始化*）。

4.4.8 防火墙寄存器映射

下表提供了防火墙寄存器映射和复位值。

表 18. 防火墙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
0x0	FW_CSSA	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ADD																Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	Reset Value									0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0								
0x4	FW_CSL	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LENG														Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	Reset Value											0	0	0	0	0	0	0	0	0	0	0	0	0	0								
0x8	FW_NVDSSA	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ADD																Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	Reset Value									0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0								
0xC	FW_NVDSL	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LENG														Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	Reset Value											0	0	0	0	0	0	0	0	0	0	0	0	0	0								
0x10	FW_VDSSA	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ADD												Res.	Res.	Res.	Res.	Res.	Res.
	Reset Value															0	0	0	0	0	0	0	0	0	0	0	0						
0x14	FW_VDSL	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LENG												Res.	Res.	Res.	Res.	Res.	Res.
	Reset Value															0	0	0	0	0	0	0	0	0	0	0	0						
0x18		Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res	Res	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	Reset Value																																
0x1C		Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res	Res	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	Reset Value																																
0x20	FW_CR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res	Res	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	VDE	VDS	FPA
	Reset Value																														0	0	0

有关寄存器边界地址的信息，请参见*第 78 页的第 2.2.2 节*。

5 电源控制 (PWR)

5.1 电源

STM32L4Rxxx 和 STM32L4Sxxx 器件的工作电压 (V_{DD}) 要求介于 1.71 V 到 3.6 V 之间。通过以下独立电源域为多个外设供电：V_{DDA}、V_{DDIO2}、V_{DDUSB}、V_{DDDSI}。这些电源必须使用 V_{DD} 引脚上的有效工作供电提供。

- V_{DD} = 1.71 V至3.6 V

  V_{DD} 是为I/O、内部稳压器和系统模拟信号（如复位、电源管理和内部时钟）供电的外部电源。通过 VDD 引脚从外部提供。

- V_{DDA} = 1.62 V (ADC/COMP)/1.8 V (DAC/OPAMP)/2.4 V (VREFBUF) 至3.6 V

  V_{DDA} 是为 A/D 转换器、D/A 转换器、电压参考缓冲器、运算放大器和比较器供电的外部模拟电源。V_{DDA} 电压电平独立于 V_{DD} 电压。当不使用这些外设时，最好将 V_{DDA} 连接到 V_{DD}。

- V_{DD12} = 1.05 至 1.32 V

  V_{DD12} 是外部电源，在连接到外部 SMPS 时会旁路内部稳压器。它通过 V_{DD12} 引脚由外部供电，仅在具有外部 SMPS 电源选项的封装上可用。V_{DD12} 不需要任何外部去耦电容，也无法支持任何外部负载。

- V_{DDUSB} = 3.0 V 到 3.6 V

  V_{DDUSB} 为外部独立电源，为 USB 收发器供电。V_{DDUSB} 电压电平独立于 V_{DD} 电压。当不使用 USB 时，最好将 V_{DDUSB} 连接到 V_{DD}。

  V_{DDUSB} 电源可能不以专用引脚的形式存在，而是内部绑定到 V_{DD}。对于此类器件，在使用 USB 时，V_{DD} 必须遵循 V_{DDUSB} 供电范围。

- V_{DDIO2} = 1.08 V 到 3.6 V

  V_{DDIO2} 是为 14 个 I/O（Port G[15:2]）供电的外部电源。V_{DDIO2} 电压级别与 V_{DD} 电压无关，不使用 PG[15:2] 时，最好连接到 V_{DD}。

- V_{DDDSI} 是独立 DSI 电源，专用于 DSI 调节器和 MIPI DPHY。此电源必须连接到全局 V_{DD}。
- V_{CAPDSI} 引脚是 DSI 调节器 (1.2 V) 的输出，此调节器必须外部连接到 $V_{DD12DSI}$。
- $V_{DD12DSI}$ 引脚用于为 MIPI D-PHY、时钟和数据通道引脚供电。$V_{DD12DSI}$ 引脚必须连接一个 2.2 uF 外部电容。
- V_{BAT} = 1.55 V 到 3.6 V

  当 V_{DD} 掉电时，V_{BAT} 作为 RTC、32 kHz 外部时钟振荡器和备份寄存器的电源（通过电源开关供电）。对于不带专用引脚的小型封装，VBAT 内部绑定到 VDD。

- V_{REF-} 和 V_{REF+}

  V_{REF+} 为 ADC 和 DAC 的输入参考电压。使能时，它还是内部电压参考缓冲器的输出。

  当 V_{DDA} < 2 V时，V_{REF+} 必须等于 V_{DDA}。

  当 V_{DDA} ≥ 2 V 时，V_{REF+} 必须在 2 V 和 V_{DDA} 之间。

  当 ADC 和 DAC 不活动时，V_{REF+} 可接地。

  内部电压参考缓冲器支持两种输出电压值，可利用 VREFBUF_CSR 寄存器中的 VRS 位进行配置：

  – V_{REF+} 大约为 2.048 V。这要求 V_{DDA} 大于等于 2.4 V。

  – V_{REF+} 大约为 2.5 V。这要求 V_{DDA} 大于等于 2.8 V。

  VREF- 和 VREF+ 引脚并非在所有封装上可用。当在封装上不可用时，它们分别绑定到 VSSA 和 VDDA。

  当 VREF+ 与 VDDA 在封装中互相绑定时，内部电压参考缓冲器不可用且必须禁用（关于封装引脚分配说明，请参见相关器件数据手册）。

  V_{REF-} 必须始终等于 V_{SSA}。

使用一个嵌入式线性稳压器来为内部数字电源 V_{CORE} 供电。V_{CORE} 是数字外设和存储器的电源。

图 7. STM32L4S5xx/R5xx 和 STM32L4S7xx/L4R7xx 电源概述

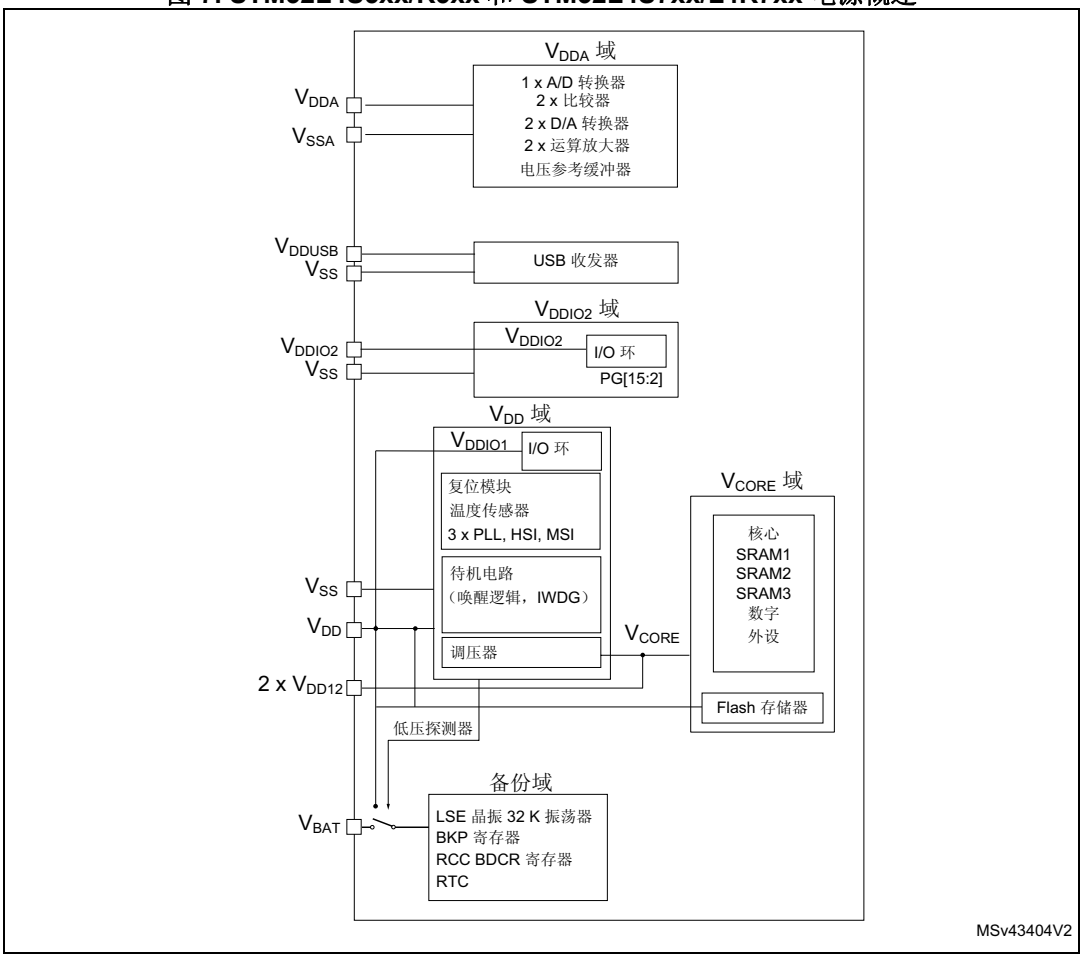
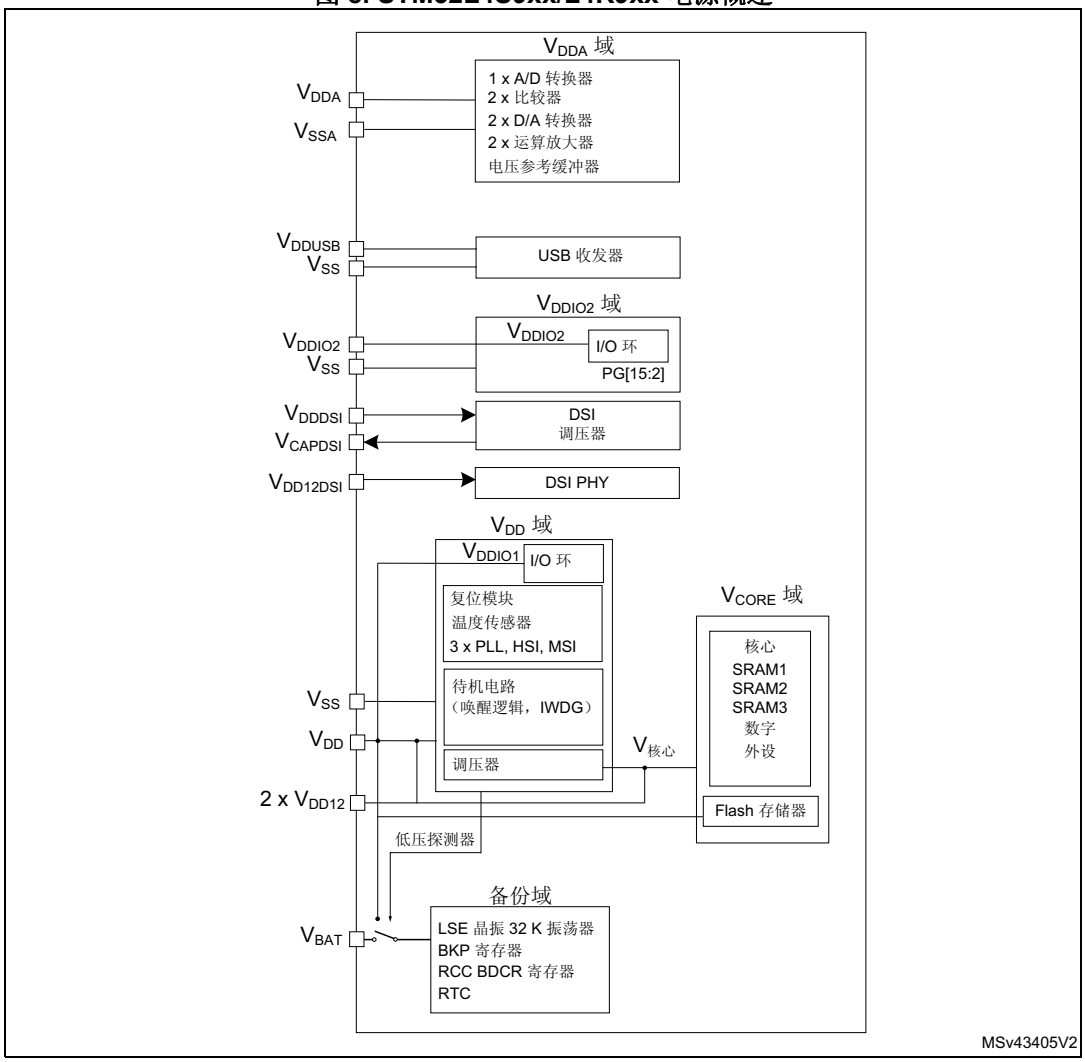


图 8. STM32L4S9xx/L4R9xx 电源概述



5.1.1 独立模拟外设电源

为了提高 ADC 和 DAC 转换精度、扩展供电的灵活性，模拟外设配有独立电源，可以单独滤波并屏蔽 PCB 上的噪声。

- 模拟外设电压源从单独的 V_{DDA} 引脚输入
- V_{SSA} 引脚提供了独立的电源接地连接

V_{DDA} 供电电压可与 V_{DD} 不同。在使能任一由 V_{DDA} 供电的模拟外设（A/D 转换器、D/A 转换器、比较器、运算放大器、电压参考缓冲器）之前，必须检查是否存在 V_{DDA}。

V_{DDA} 电源可由外设电压监测功能监测，并与两个阈值（对于 PVM3，阈值为 1.65 V，或对于 PVM4，为 2.2 V）进行比较，有关详细信息，请参见*第 5.2.3 节：外设电压监测 (PVM)*。

当使用单供电时，可通过外部滤波电路将 V_{DDA} 外部连接至 V_{DD}，以获得无噪声的 V_{DDA} 参考电压。

ADC 和 DAC 参考电压

为确保低电压输入和输出上实现更高精度，用户可将 V_{REF+} 连接至一个低于 V_{DDA} 的独立参考电压。对于模拟输入 (ADC) 或输出 (DAC) 信号，V_{REF+} 为最高电压，以满量程值表示。

V_{REF+} 可由外部参考或内部缓冲的电压参考 (VREFBUF) 来提供。

通过将*第 24.3.1 节：VREFBUF 控制和状态寄存器 (VREFBUF_CSR)* 中的 ENVR 位置 1 来使能内部电压参考。当 VRS 位置位时，电压参考设为 2.5 V，当 VRS 位清零时，电压参考设为 2.048 V。内部电压参考还可通过 V_{REF+} 引脚为外部元件供电。有关更多信息，请参见器件数据手册或*第 24 节：电压参考缓冲器 (VREFBUF)*。

5.1.2 独立 I/O 电源轨

来自 Port G（PG[15:2]）的一些 I/O 由单独的电源轨供电。此轨的电源范围为 1.08 至 3.6 V，可通过 V_{DDIO2} 引脚外部提供。V_{DDIO2} 电压电平与 V_{DD} 或 V_{DDA} 完全独立。V_{DDIO2} 引脚仅在一些封装中可用。关于 I/O 列表，请参见相关器件数据手册中的引脚分配图表。

复位后，由 V_{DDIO2} 供电的 I/O 是逻辑隔离且电隔离的，因而不可用。当 V_{DDIO2} 电源存在时，从 PG[15:2] 使用任一 I/O 前，须通过将 PWR_CR2 寄存器中的 IOSV 位置 1，解除此隔离。

V_{DDIO2} 电源可由外设电压监测功能 (PVM2) 监测，并与内部参考电压（3/4 V_{REFINT}，0.9V 左右）进行比较，有关更多详细信息，请参见*第 5.2.3 节：外设电压监测 (PVM)*。

5.1.3 独立的 USB 收发器电源

USB 收发器通过一个单独的电源引脚 V_{DDUSB} 供电。V_{DDUSB} 范围从 3.0 V 到 3.6 V，完全独立于 V_{DD} 或 V_{DDA}。

复位后，由 V_{DDUSB} 供电的 USB 功能是逻辑隔离且电隔离的，因而不可用。当 V_{DDUSB} 电源存在时，使用 USB OTG 外设前，须通过将 PWR_CR2 寄存器中的 USV 位置 1，解除此隔离。

V_{DDUSB} 电源可由外设电压监测功能 (PVM1) 监测，并与内部参考电压（V_{REFINT}，1.2 V 左右）进行比较，有关更多详细信息，请参见*第 5.2.3 节：外设电压监测 (PVM)*。

5.1.4 独立 DSI 供电

DSI（显示串行接口）子系统使用多个彼此独立的电源引脚：

- VDDDSI 是独立 DSI 电源，专用于 DSI 调节器和 MIPI D-PHY。该电源必须连接到全局 VDD。
- VCAPDSI 引脚是 DSI 调节器 (1.2V) 的输出，该调节器必须外部连接到 VDD12DSI。
- VDD12DSI 引脚用于为 MIPI D-PHY、时钟和数据通道引脚供电。VDD12DSI 引脚必须连接一个 2.2 uF 外部电容。
- VSSDSI 引脚是一个独立的电源接地，用于 DSI 子系统。

如果完全没有使用 DSI 功能，则：

- VDDDSI 引脚必须连接到全局 VDD。
- VCAPDSI 引脚必须外部连接到 VDD12DSI，但是不再需要连接外部电容。如果 VDD12DSI 在该封装上不可用，则 VCAPDSI 保持浮空。
- VSSDSI 引脚必须接地。

注： *VDDDSI 和 VDD12DSI 引脚并非在所有封装上可用。当不可用时，它们分别绑定到 VDD 和 VCAPDSI。*

5.1.5 电池备份域

为了在 V_{DD} 掉电时，还能保留备份寄存器的内容，且让 RTC 继续工作，可将 VBAT 引脚连接到由电池或者其他电源提供的可选备用电源上。

VBAT 引脚为 RTC 单元、LSE 振荡器和 PC13 到 PC15 I/O 供电，允许 RTC 在主电源关闭时也可工作。V_{BAT} 电源的开关由复位模块中内置的掉电复位电路进行控制。

警告： **在 $t_{RSTTEMPO}$（V_{DD} 启动后的一段延迟）期间或检测到 PDR 后，V_{BAT} 与 V_{DD} 之间的电源开关仍连接到 V_{BAT}。**
在启动阶段，如果 V_{DD} 的建立时间小于 $t_{RSTTEMPO}$（有关 $t_{RSTTEMPO}$ 的值，请参见数据手册）且 $V_{DD} > V_{BAT} + 0.6$ V，会有电流经由 V_{DD} 和电源开关 (V_{BAT}) 之间连接的内部二极管注入 V_{BAT} 引脚。
如果连接到 VBAT 引脚的电源/电池无法承受此注入电流，则强烈建议在该电源与 VBAT 引脚之间连接一个低压降二极管。

如果应用中没有使用任何外部电池，建议将该 V_{BAT} 外部连接到带有 100 nF 外部去耦电容的 V_{DD} 上。

通过 V_{DD} 对备份域供电时（模拟开关连接到 V_{DD}），以下引脚可用：

- PC13、PC14 和 PC15，可用作 GPIO 引脚
- PC13、PC14 和 PC15，可由 RTC 或 LSE 配置（请参见*第 1347 页的第 42.3 节：RTC 功能说明*）
- PA0/RTC_TAMP2 和 PE6/RTC_TAMP3 可通过 RTC 配置为 tamper 引脚

注： *由于模拟开关仅能传递有限的电流 (3 mA)，因此使用输出模式的 GPIO PC13 到 PC15 是受限的：速率必须限制在 2 Mhz，最大负载为 30 pF，且这些 I/O 不能作为电流源使用（如，驱动 LED）。*

通过 V_{BAT} 对备份域供电时（由于不存在 V_{DD}，内部电源开关连接到 V_{BAT}），可实现以下功能：

- PC13、PC14 和 PC15 仅受 RTC 或 LSE 控制（请参见*第 42.3 节：RTC 功能说明*）
- PA0/RTC_TAMP2 和 PE6/RTC_TAMP3 可通过 RTC 配置为 tamper 引脚

备份域访问

系统复位后，备份域（RTC 寄存器和备份寄存器）将受到保护，以防止意外的写访问。要使能对备份域的访问，请按以下步骤进行操作：

1. 将*第 6.4.19 节：APB1 外设时钟使能寄存器 1 (RCC_APB1ENR1)* 中的 PWREN 置 1，使能电源接口时钟。
2. 将*电源控制寄存器 1 (PWR_CR1)* 中的 DBP 位置 1，使能对备份域的访问。
3. 在*备份域控制寄存器 (RCC_BDCR)* 中选择 RTC 时钟源。
4. 通过将*备份域控制寄存器 (RCC_BDCR)* 中的 RTCEN [15] 位置 1，使能 RTC 时钟。

VBAT 电池充电

当 VDD 存在时，可通过内部电阻为 VBAT 上的外部电池充电。

VBAT 充电可通过一个 5 kΩ 的电阻或 1.5 kΩ 的电阻来实现，取决于 PWR_CR4 寄存器中 VBRS 位的值。

可将 PWR_CR4 寄存器中的 VBE 位置 1 来使能电池充电。在 VBAT 模式下自动禁用。

5.1.6 稳压器

两个嵌入式线性稳压器为待机电路和备份域以外的所有数字电路供电。根据系统的最大工作频率，主稳压器输出电压 (V_{CORE}) 可由软件编程为两种不同的电源范围（范围 1 和范围 2），以优化功耗（请参见*第 6.2.9 节：时钟源频率与电压调节*和*第 3.3.3 节：读访问延迟*）。

稳压器在复位后始终处于使能状态。根据应用模式，V_{CORE} 电源由主稳压器 (MR) 或低功耗稳压器 (LPR) 提供。

- 在运行模式、睡眠和停止 0 模式中，两个稳压器使能，主稳压器 (MR) 为 V_{CORE} 域（内核、存储器和数字外设）提供全功率。
- 在低功耗运行模式和低功耗睡眠模式中，主稳压器关闭，低功耗稳压器 (LPR) 为 V_{CORE} 域提供低功率，保持寄存器、SRAM1、SRAM2 和 SRAM3 的内容。
- 在停止 1 和停止 2 模式中，主稳压器关闭，低功耗稳压器 (LPR) 为 V_{CORE} 域提供低功率，保持寄存器、SRAM1、SRAM2 和 SRAM3 的内容。
- 在待机模式且 SRAM2 内容保持时（PWR_CR3 寄存器中的 RRS 位置 1），主稳压器 (MR) 关闭，低功耗稳压器 (LPR) 仅为 SRAM2 供电。内核、数字外设（待机电路和备份域除外）和 SRAM1 掉电。
- 在待机模式中，两个稳压器均掉电。除待机电路和备份域外，寄存器、SRAM1、SRAM2 和 SRAM3 的内容都将丢失。
- 在关断模式中，两个稳压器均掉电。离开关断模式时，产生一个上电复位。因此，除备份域外，寄存器、SRAM1、SRAM2 和 SRAM3 的内容都将丢失。

5.1.7 动态电压调节管理

动态电压调节是一种电源管理技术，包括根据应用性能和功耗要求，增大或减小用于数字外设的电压 (V_{CORE})。

动态电压调节增大 V_{CORE} 称作过压。它可提高器件性能。

动态电压调节减小 V_{CORE} 称作欠压。它用来节省功率，尤其是用于笔记本电脑和其他移动设备，其有限能源来自电池的场合。

- 范围 1：高性能范围。

在范围 1 中，根据 PWR_CR5 寄存器中 R1MODE 位的设置，主稳压器在两种模式下工作：

- 主稳压器范围 1 正常模式：提供 1.2 V 典型输出电压。当系统时钟频率高达 80 MHz 时使用。读访问的 Flash 访问时间最小，可进行写入和擦除操作。
- 主稳压器范围 1 升压模式：提供 1.28 V 典型输出电压。当系统时钟频率高达 120 MHz 时使用。读访问的 Flash 访问时间最小，可进行写入和擦除操作。为优化功率损耗，建议在系统时钟频率高于 80 MHz 时选择范围 1 升压模式。请参见*表 19*。

表 19. 范围 1 升压模式配置

系统频率	26 MHz < SYSCLK ≤ 80 MHz	80 MHz < SYSCLK ≤ 120 MHz
R1MODE 位配置	1	0

- 范围 2：低功耗范围。

主稳压器提供了 1.0 V 的典型输出电压。系统时钟频率可达 26 MHz。相比于范围 1，其读访问的 Flash 访问时间增加；不可进行写入和擦除操作。

通过 PWR_CR1 寄存器中的 VOS 位选择电压调节。

从范围 1（正常/升压）进入范围 2 的顺序为：

1. 如果要从范围 1 升压模式切换到范围 2，则在切换到较低系统频率并至少持续 1 µs 前，必须使用 AHB 预分频器将系统时钟 2 分频，然后配置 AHB 预分频器：
2. 降低系统频率至低于 26 MHz。
3. 根据范围 2 中新的频率目标来调整等待状态的数量（FLASH_ACR 中的 LATENCY 位）。
4. 将 PWR_CR1 寄存器中的 VOS 位编程为“10”。

从范围 2 进入范围 1 的顺序为：

1. 将 PWR_CR1 寄存器中的 VOS 位编程为“01”。
2. 等待，直至 PWR_SR2 寄存器中的 VOSF 标志清零。
3. 根据范围 1 中新的频率目标来调整等待状态的数量（FLASH_ACR 中的 LATENCY 位）。
4. 按照以下步骤提升系统频率：

– 如果系统频率为 26 MHz < SYSCLK ≤ 80 MHz：
  - 通过将 PWR_CR5 寄存器中的 R1MODE 位置 1，选择范围 1 正常模式。
  - 配置并切换到 PLL，以获得新系统频率。

– 如果系统频率为 SYSCLK > 80 MHz：
  - 在切换到更高的系统频率之前，必须使用 AHB 预分频器将系统时钟 2 分频。
  - 通过将 PWR_CR5 寄存器中的 R1MODE 位清零，选择范围 1 升压模式。
  - 配置并切换到 PLL，以获得新系统频率。
  - 等待至少 1 µs，然后配置 AHB 预分频器，获得所需 HCLK 时钟频率。

从范围 1 正常模式切换到范围 1 升压模式的顺序为：

1. 在切换到更高的系统频率之前，必须使用 AHB 预分频器将系统时钟 2 分频。
2. 将 PWR_CR5 寄存器中的 R1MODE 位清零。
3. 根据范围 1 升压模式中的新频率目标调整等待状态数。
4. 配置并切换到新系统频率。
5. 等待至少 1 µs，然后配置 AHB 预分频器，获得所需 HCLK 时钟频率。

从范围 1 升压模式切换到范围 1 正常模式的顺序为：

1. 将 PWR_CR5 寄存器中的 R1MODE 位置 1。
2. 根据范围 1 默认模式中的新频率目标调整等待状态数。
3. 配置并切换到新系统频率。

5.2 电源监控器

5.2.1 上电复位上电复位 (POR)/掉电复位 (PDR)/欠压复位 (BOR)

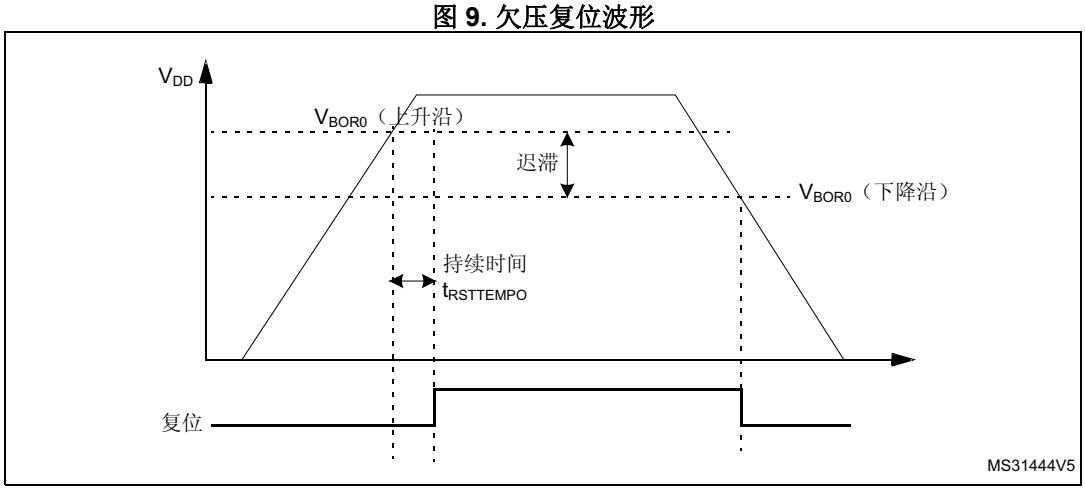
该器件具有一个集成的上电复位 (POR)/掉电复位 (PDR)，与欠压复位 (BOR) 电路耦合。除关断模式外，BOR在所有功耗模式下均激活，且不可禁用。

通过选项字节，可对 5 个 BOR 阈值进行选择。

上电期间，BOR 将使器件保持复位状态，直到电源电压 V_{DD} 达到指定的 V_{BORx} 阈值。当 V_{DD} 降至所选阈值以下时，将使器件复位。当 V_{DD} 高于 V_{BORx} 上限时，释放器件复位，系统可以启动。

有关欠压复位阈值的相关详细信息，请参见数据手册的电气特性部分。

图 9. 欠压复位波形



1. 复位持续时间 $t_{RSTTEMPO}$ 仅针对 BOR 最低阈值 (V_{BOR0}) 存在。

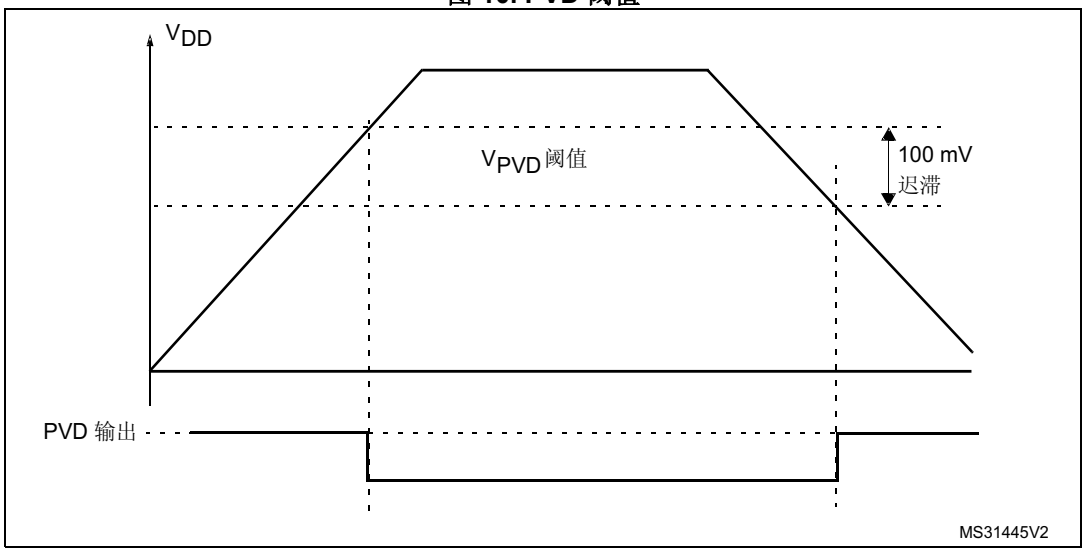
5.2.2 可编程电压检测器 (PVD)

可以使用 PVD 监视 V_{DD} 电源，方法是将其与 *电源控制寄存器 2 (PWR_CR2)* 中的 PLS[2:0] 位所选的阈值进行比较。

通过设置 PVDE 位来使能 PVD。

电源状态寄存器 2 (PWR_SR2) 中提供了 PVDO 标志位，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。根据 PVD 输出行为配置 EXTI 线 16 上升沿/下降沿有效性，即，如果 EXTI 线 16 配置为上升沿有效，则会在 VDD 下降到低于 PVD 阈值时生成中断。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 10. PVD 阈值



5.2.3 外设电压监测 (PVM)

默认情况下仅监测 V_{DD}，因为它是唯一与所有系统功能相关的电源。其他电源（V_{DDA}、V_{DDIO2} 和 V_{DDUSB}）可以独立于 V_{DD}，并使用四个外设电源监测 (PVM) 功能进行监测。

四个 PVMx（x=1、2、3 和 4）中的每一个都是比较器，用于比较固定阈值 V_{PVMx} 和所选电源。PVMOx 标志用于表示独立电源是高于还是低于 PVMx 阈值：当电源电压高于 PVMx 阈值时，PVMOx 标志清零，当电源低于 PVMx 阈值时，PVMOx 标志置 1。

每个 PVM 输出都连接到 EXTI 线，如果通过 EXTI 寄存器使能，则可以产生中断。当独立电源降至 PVMx 阈值以下以及/或者升至 PVMx 阈值以上时，会产生 PVMx 输出中断，具体取决于 EXTI 线上升/下降沿的配置。

在停止 0、停止 1 和停止 2 模式下，每个 PVM 都可以保持有效，并且可以从停止模式唤醒 PVM 中断。

表 20. PVM 功能

PVM	电源	PVM 阈值	EXTI 线
PVM1	V_{DDUSB}	V_{PVM1}（1.2 V 左右）	35
PVM2	V_{DDIO2}	V_{PVM2}（0.9 V 左右）	36
PVM3	V_{DDA}	V_{PVM3}（1.65 V 左右）	37
PVM4	V_{DDA}	V_{PVM4}（2.2 V 左右）	38

默认情况下认为独立电源（V_{DDA}、V_{DDIO2} 和 V_{DDUSB}）不存在，并应用逻辑隔离和电气隔离以忽略由这些专用电源供电的外设发出的任何信息。

- 如果这些电源外部短接到 V_{DD}，应用应假定它们可以在不使能任何外设电压监测的情况下使用。
- 如果这些电源独立于 V_{DD}，则可以使能外设电压监测 (PVM)，确认是否存在电源。

使用 USB OTG 外设前，必须按以下顺序进行操作：

1. 如果 V_{DDUSB} 独立于 V_{DD}：
   a) 通过将 *电源控制寄存器 2 (PWR_CR2)* 中的 PVME1 位置 1，使能 PVM1。
   b) 等待 PVM1 唤醒时间。
   c) 等待 *电源状态寄存器 2 (PWR_SR2)* 中的 PVMO1 位清零。
   d) 可选：禁止 PVM1 以节约功耗。
2. 将 *电源控制寄存器 2 (PWR_CR2)* 中的 USV 位置 1，以移除 V_{DDUSB} 电源隔离。

从 PG[15:2] 使用任一 I/O 前，必须按以下顺序进行操作：

1. 如果 V_{DDIO2} 独立于 V_{DD}：
   a) 通过将 *电源控制寄存器 2 (PWR_CR2)* 中的 PVME2 位置 1，使能 PVM2。
   b) 等待 PVM2 唤醒时间。
   c) 等待 *电源状态寄存器 2 (PWR_SR2)* 中的 PVMO2 位清零。
   d) 可选：禁止 PVM2 以节约功耗。
2. 将 *电源控制寄存器 2 (PWR_CR2)* 中的 IOSV 位置 1，以移除 V_{DDIO2} 电源隔离。

使用以下任一模拟外设前，必须按照以下顺序进行操作：模数转换器、数模转换器、比较器、运算放大器和电压参考缓冲器：

1. 如果 V_{DDA} 独立于 V_{DD}：
   a) 通过将 *电源控制寄存器 2 (PWR_CR2)* 中的 PVME3（或 PVME4）位置 1，使能 PVM3（或 PVM4）。
   b) 等待 PVM3（或 PVM4）唤醒时间。
   c) 等待 *电源状态寄存器 2 (PWR_SR2)* 中的 PVME3（或 PVME4）位清零。
   d) 可选：禁止 PVM3（或 PVM4）以节约功耗。
2. 使能模拟外设，此操作自动移除 V_{DDA} 隔离。

5.3 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。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。

器件有七种低功耗模式：

- 睡眠模式：CPU 时钟关闭，包括 Cortex®-M4 内核外设（例如 NVIC、SysTick 等）在内的所有外设都可以运行，并在发生中断或事件时唤醒 CPU。请参见 *第 5.3.4 节：睡眠模式*。
- 低功耗运行模式：当系统时钟频率减少到 2 MHz 以下时实现此模式。从 SRAM 或 Flash 执行代码。稳压器处于低功耗模式以最大程度降低稳压器的工作电流。请参见 *第 5.3.2 节：低功耗运行模式（LP 运行）*。

- 低功耗睡眠模式：从低功耗运行模式进入此模式。Cortex®-M4 已关闭。请参见*第 5.3.5 节：低功耗睡眠模式（LP 睡眠）*。
- 停止 0、停止 1 和停止 2 模式：SRAM1、SRAM2、SRAM3 和所有寄存器的内容保留。V_{CORE} 域中的所有时钟都停止，并禁止 PLL、MSI、HSI16 和 HSE。LSI 和 LSE 可以保持运行。

  RTC 可以保持有效（带 RTC 的停止模式，不带 RTC 的停止模式）。

  一些具有唤醒功能的外设可以在停止模式期间使能 HSI16 RC，以检测它们的唤醒条件。

  在停止 2 模式下，大多数 V_{CORE} 域处于低漏电模式。为了进一步减少停止 2 期间的电流损耗，可以关闭 SRAM3：
  - 在停止 2 模式下，当 PWR_CR1 寄存器中 RRSTP 位清零（默认设置）时，SRAM3 内容丢失。

  在停止 2 模式下，当 PWR_CR1 寄存器中的 RRSTP 位置 1 时，SRAM3 内容保留。

  停止 1 提供最大数量的有效外设和唤醒源，相比停止 2，其唤醒时间较短，但功耗高很多。在停止 0 模式下，主稳压器保持开启，可实现最快的唤醒时间，但功耗高很多。有效外设和唤醒源与停止 1 模式下相同。

  当从停止 0、停止 1 或停止 2 模式退出时，系统时钟为最高 48 MHz 的 MSI 或 HSI16，具体取决于软件配置。

  请参见*第 5.3.6 节：停止 0 模式*和*第 5.3.8 节：停止 2 模式*。
- 待机模式：V_{CORE} 域断电。但是，可以保留 SRAM2 内容：
  - 在待机模式下，当 PWR_CR3 寄存器中的 RRS 位置 1 时，SRAM2 内容保留。在这种情况下，SRAM2 由低功耗稳压器供电。
  - 在待机模式下，当 PWR_CR3 寄存器中的 RRS 位清零时。在这种情况下，主稳压器和低功耗稳压器关闭。

  V_{CORE} 域中的所有时钟都停止，并禁止 PLL、MSI、HSI16 和 HSE。LSI 和 LSE 可以保持运行。

  RTC 可以保持有效（带 RTC 的待机模式，不带 RTC 的待机模式）。

  当退出待机模式时，系统时钟为 1 MHz 到最高 8 MHz 的 MSI。

  请参见*第 5.3.9 节：待机模式*。
- 关断模式：V_{CORE} 域断电。V_{CORE} 域中的所有时钟都停止，并禁止 PLL、MSI、HSI16、LSI 和 HSE。LSE 可以保持运行。当退出关断模式时，系统时钟为 4 MHz 的 MSI。在该模式下，会禁止电源电压监测，并且电源电压下降时不能保证产品特性。请参见*第 5.3.10 节：关断模式*。

此外，可通过下列方法之一降低运行模式下的功耗：

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

图 11. 低功耗模式可能的转换

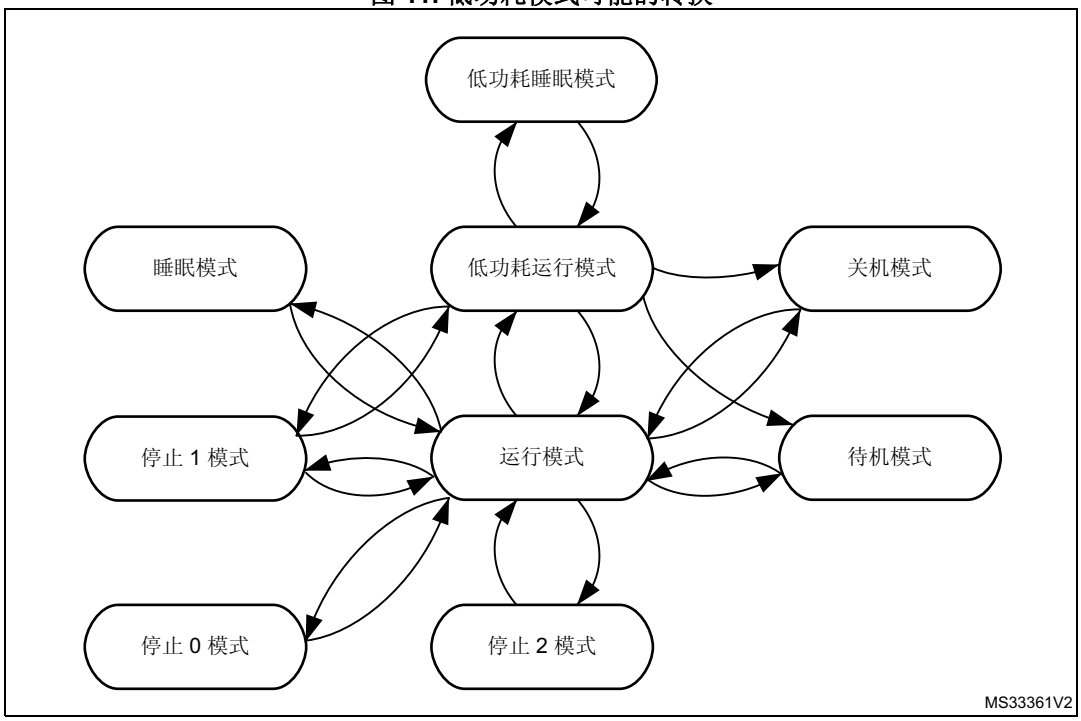


表 21. 低功耗模式汇总

模式名称	条目	唤醒源(1)	唤醒系统时钟	对时钟的影响	稳压器	
					MR	LPR
睡眠（立即睡眠或退出时睡眠）	WFI 或从 ISR 返回	任意中断	与进入睡眠模式之前相同	CPU 时钟关闭 对其他时钟或模拟时钟源无影响	开启	开启
	WFE	唤醒事件				
低功耗运行	将 LPR 位置 1	将 LPR 位清零	与低功耗运行时钟相同	无	关闭	开启
低功耗睡眠	将 LPR 位置 1 + WFI 或从 ISR 返回	任意中断	与进入低功耗睡眠模式之前相同	CPU 时钟关闭 对其他时钟或模拟时钟源无影响	关闭	开启
	将 LPR 位置 1 + WFE	唤醒事件			关闭	开启
停止 0	LPMS=“000”+ SLEEPDEEP 位 + WFI 或从 ISR 或 WFE 返回	任意 EXTI 线（在 EXTI 寄存器中配置） 特定外设事件	当 RCC_CFGR 中 STOPWUCK=1 时为 HSI16 当 STOPWUCK=0 时，在进入停止模式前为带频率的 MSI	除 LSI 和 LSE 之外的所有时钟关闭	开启	开启
停止 1	LPMS=“001”+ SLEEPDEEP 位 + WFI 或从 ISR 或 WFE 返回				关闭	
停止 2	LPMS=“010”+ SLEEPDEEP 位 + WFI 或从 ISR 或 WFE 返回					
带 SRAM2 的待机	LPMS=“011”+ 将 RRS 位置 1 + SLEEPDEEP 位 + WFI 或从 ISR 或 WFE 返回	WKUP 引脚边沿、RTC 事件、NRST 引脚的外部复位、IWDG 复位	1 MHz 到最高 8 MHz 的 MSI			
待机	LPMS=“011”+ 清零 RRS 位 + SLEEPDEEP 位 + WFI 或从 ISR 或 WFE 返回	WKUP 引脚边沿、RTC 事件、NRST 引脚的外部复位、IWDG 复位			关闭	关闭
关机	LPMS=“1--”+ SLEEPDEEP 位 + WFI 或从 ISR 或 WFE 返回	WKUP 引脚边沿、RTC 事件、NRST 引脚的外部复位	MSI 4 MHz	除 LSE 之外的所有时钟关闭	关闭	关闭

1. 请参见*表 22：功能取决于工作模式*。

表 22. 功能取决于工作模式(1)

外设	运行	睡眠	低功耗运行	低功耗睡眠	停止 0/1		停止 2		待机		关机		VBAT
					-	唤醒能力	-	唤醒能力	-	唤醒能力	-	唤醒能力	
CPU	是	-	是	-	-	-	-	-	-	-	-	-	-
Flash (2 MB)	O(2)	O(2)	O(2)	O(2)	-	-	-	-	-	-	-	-	-
SRAM1 (192 KB)	是	是(3)	是	是(3)	是	-	是	-	-	-	-	-	-
SRAM2 (64 KB)	是	是(3)	是	是(3)	是	-	是	-	O(4)	-	-	-	-
SRAM3 (384 KB)	是	是(3)	是	是(3)	是	-	是	-	-	-	-	-	-
FSMC	O	O	O	O	-	-	-	-	-	-	-	-	-
OCTOSPI	O	O	O	O	-	-	-	-	-	-	-	-	-
备份寄存器	是	是	是	是	是	-	是	-	是	-	是	-	是
欠压复位 (BOR)	是	是	是	是	是	是	是	是	是	是	-	-	-
可编程电压检测器 (PVD)	O	O	O	O	O	O	O	O	-	-	-	-	-
外设电压监测（PVMx；x=1、2、3、4）	O	O	O	O	O	O	O	O	-	-	-	-	-
DMA	O	O	O	O	-	-	-	-	-	-	-	-	-
DMA2D	O	O	O	O	-	-	-	-	-	-	-	-	-
振荡器 HSI16	O	O	O	O	(5)	-	(5)	-	-	-	-	-	-
振荡器 HSI48	O	O	-	-	-	-	-	-	-	-	-	-	-
高速外部 (HSE)	O	O	O	O	-	-	-	-	-	-	-	-	-
低速内部 (LSI)	O	O	O	O	O	-	O	-	O	-	-	-	-
低速外部 (LSE)	O	O	O	O	O	-	O	-	O	-	O	-	O
多速内部 (MSI)	O	O	O	O	-	-	-	-	-	-	-	-	-
时钟安全系统 (CSS)	O	O	O	O	-	-	-	-	-	-	-	-	-
LSE 上的时钟安全系统	O	O	O	O	O	O	O	O	O	O	-	-	-
RTC/自动唤醒	O	O	O	O	O	O	O	O	O	O	O	O	O
RTC 入侵引脚的数量	3	3	3	3	3	O	3	O	3	O	3	O	3
相机接口	O	O	O	O	-	-	-	-	-	-	-	-	-

表 22. 功能取决于工作模式(1)（续）

外设	运行	睡眠	低功耗运行	低功耗睡眠	停止 0/1		停止 2		待机		关机		VBAT
					-	唤醒能力	-	唤醒能力	-	唤醒能力	-	唤醒能力	
LCD-TFT	O	O	-	-	-	-	-	-	-	-	-	-	-
GFXMMU	是	是(3)	是	是(3)	是	-	是	-	-	-	-	-	-
DSIHOST	O	O	-	-	-	-	-	-	-	-	-	-	-
USB OTG FS	O(8)	O(8)	-	-	-	O	-	-	-	-	-	-	-
USARTx （x=1、2、3、4、5）	O	O	O	O	O (6)	O (6)	-	-	-	-	-	-	-
低功耗 UART (LPUART1)	O	O	O	O	O (6)	O (6)	O(6)	O (6)	-	-	-	-	-
I2Cx (x=1,2,4)	O	O	O	O	O (7)	O (7)	-	-	-	-	-	-	-
I2C3	O	O	O	O	O (7)	O (7)	O (7)	O (7)	-	-	-	-	-
SPIx (x=1,2,3)	O	O	O	O	-	-	-	-	-	-	-	-	-
CAN1	O	O	O	O	-	-	-	-	-	-	-	-	-
SDMMC1	O	O	O	O	-	-	-	-	-	-	-	-	-
SAIx (x=1,2)	O	O	O	O	-	-	-	-	-	-	-	-	-
DFSDM1	O	O	O	O	-	-	-	-	-	-	-	-	-
ADC1	O	O	O	O	-	-	-	-	-	-	-	-	-
DACx (x=1,2)	O	O	O	O	O	-	-	-	-	-	-	-	-
VREFBUF	O	O	O	O	O	-	-	-	-	-	-	-	-
OPAMPx (x=1,2)	O	O	O	O	O	-	-	-	-	-	-	-	-
COMPx (x=1,2)	O	O	O	O	O	O	O	O	-	-	-	-	-
温度传感器	O	O	O	O	-	-	-	-	-	-	-	-	-
定时器 (TIMx)	O	O	O	O	-	-	-	-	-	-	-	-	-
低功耗定时器 1 (LPTIM1)	O	O	O	O	O	O	O	O	-	-	-	-	-
低功耗定时器 2 (LPTIM2)	O	O	O	O	O	O	-	-	-	-	-	-	-
独立看门狗 (IWDG)	O	O	O	O	O	O	O	O	O	O	-	-	-
窗口看门狗 (WWDG)	O	O	O	O	-	-	-	-	-	-	-	-	-
SysTick 定时器	O	O	O	O	-	-	-	-	-	-	-	-	-

表 22. 功能取决于工作模式(1)（续）

外设	运行	睡眠	低功耗运行	低功耗睡眠	停止 0/1		停止 2		待机		关机		VBAT
					-	唤醒能力	-	唤醒能力	-	唤醒能力	-	唤醒能力	
触摸感应控制器 (TSC)	O	O	O	O	-	-	-	-	-	-	-	-	-
随机数发生器 (RNG)	O(8)	O(8)	-	-	-	-	-	-	-	-	-	-	-
AES 硬件加速器	O	O	O	O	-	-	-	-	-	-	-	-	-
HASH 硬件加速器	O	O	O	O	-	-	-	-	-	-	-	-	-
CRC 计算单元	O	O	O	O	-	-	-	-	-	-	-	-	-
GPIO	O	O	O	O	O	O	O	O	(9)	5 个引脚 (10)	(11)	5 个引脚 (10)	-

1. 图例：Y = 支持（使能）。O = 可选（默认情况下禁止。可通过软件使能）。- = 不提供。
2. Flash 可配置为断电模式。默认情况下，它不是断电模式。
3. SRAM 时钟可门控打开或关闭。
4. 当 PWR_CR3 寄存器中的 RRS 位置 1 时，SRAM2 内容保留。
5. 一些能够从停止模式唤醒的外设可请求将 HSI16 使能。在这种情况下，HSI16 由外设唤醒，并且仅响应请求它的外设。当外设不再需要 HSI16 时，HSI 将自动关闭。
6. UART 和 LPUART 接收是停止模式下的功能，并能在开始、地址匹配或接收到帧事件时产生一个唤醒中断。
7. I2C 地址检测是停止模式下的功能，并能在地址匹配时产生一个唤醒中断。
8. 电压调节仅限范围 1。
9. 待机模式下，I/O 可配置为内部上拉、下拉或浮空。
10. 具有从待机/关断唤醒能力的 I/O 为：PA0、PC13、PE6、PA2、PC5。
11. I/O 在关断模式中可配置为内部上拉、下拉或浮空，但是当退出关断模式时该配置会丢失。

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 0、停止 1、停止 2、停止模式或待机模式，调试连接将中断。这是因为 Cortex®-M4 内核时钟停止。

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。有关详细信息，请参见*第 51.16.1 节：对低功耗模式的调试支持*。

5.3.1 运行模式

降低系统时钟速度

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

有关详细信息，请参见*第 6.4.3 节：时钟配置寄存器 (RCC_CFGR)*。

外设时钟门控

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

要进一步降低睡眠模式的功耗，可在执行 WFI 或 WFE 指令之前禁止外设时钟。

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

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

5.3.2 低功耗运行模式（LP 运行）

当系统处于停止模式时，要进一步降低功耗，可以在低功耗模式下配置稳压器。在此模式下，系统频率不应超过 2 MHz。

有关稳压器和外设工作条件的详细信息，请参见产品数据手册。

低功耗运行模式下的 I/O 状态

在低功耗运行模式下，所有 I/O 引脚的状态与运行模式下相同。

进入低功耗运行模式

要进入低功耗运行模式，请执行下列操作：

1. 可选：跳转到 SRAM，并通过将 *Flash 访问控制寄存器 (FLASH_ACR)* 中的 RUN_PD 位置 1 关断 Flash。
2. 将系统时钟频率降到 2 MHz 以下。
3. 将 PWR_CR1 寄存器中的 LPR 位置 1，强制寄存器进入低功耗模式。

有关如何进入低功耗运行模式，请参见*表 23：低功耗运行*。

退出低功耗运行模式

要退出低功耗运行模式，请执行下列操作：

1. 将 PWR_CR1 寄存器中的 LPR 位清零，强制寄存器进入主模式。
2. 等待 PWR_SR2 寄存器中的 REGLPF 位清零。
3. 增加系统时钟频率。

有关如何退出低功耗运行模式，请参见*表 23：低功耗运行*。

表 23. 低功耗运行

低功耗运行模式	说明
进入模式	将系统时钟频率降到 2 MHz 以下 LPR = 1
退出模式	LPR = 0 等待 REGLPF = 0 增加系统时钟频率
唤醒延迟	稳压器从低功耗模式唤醒的时间

5.3.3 低功耗模式

进入低功耗模式

MCU 通过执行 WFI（等待中断）或 WFE（等待事件）指令进入低功耗模式，也可通过在从 ISR 返回时将 Cortex®-M4 系统控制寄存器中的 SLEEPONEXIT 位置 1 进入低功耗模式。
在没有中断或事件挂起时，才可通过 WFI 或 WFE 进入低功耗模式。

退出低功耗模式

MCU 根据进入睡眠和停止模式的方式退出这两种低功耗模式：

- 如果使用 WFI 指令或从 ISR 返回进入低功耗模式，通过 NVIC 确认的任何外设中断都能唤醒器件。
- 如果使用 WFE 指令进入低功耗模式，MCU 将在有事件发生时立即退出低功耗模式。唤醒事件可通过以下方式产生：
  - NVIC IRQ 中断。

    - 当 Cortex®-M4 系统控制寄存器中的 SEVONPEND = 0 时。通过在外设控制寄存器和 NVIC 中使能中断。当 MCU 从 WFE 恢复时，必须清零外设中断挂起位和 NVIC 外设 IRQ 通道挂起位（在 NVIC 中断清零挂起寄存器中）。

    只有当 NVIC 中断的优先级足够高时，才能唤醒和中断 MCU。

    - 当 Cortex®-M4 系统控制寄存器中的 SEVONPEND = 1 时。

    通过在外设控制寄存器或 NVIC 中使能中断。当 MCU 从 WFE 恢复时，必须清零外设中断挂起位和使能 NVIC 外设 IRQ 通道挂起位时（在 NVIC 中断清零挂起寄存器中）。

    所有 NVIC 中断将唤醒 MCU，即使是禁止的亦是如此。只有当已使能 NVIC 中断的优先级足够高时，才能唤醒和中断 MCU。
  - 事件

    将一个 EXTI 线配置为事件模式。当 CPU 从 WFE 恢复时，因为对应事件线的挂起位没有被置 1，不必清零 EXTI 外设的中断挂起位或 NVIC IRQ 通道挂起位。

    可能需要清零相应外设中的中断标志。

MCU 通过外部复位（NRST 引脚）、IWDG 复位、已使能 WKUPx 引脚之一上的上升沿或者触发 RTC 事件来退出待机或关断低功耗模式（请参见*图 419：RTC 框图*）。
从待机或关断模式唤醒后，程序将按照复位（启动引脚采样、选项字节加载和复位向量已获取等）后的方式重新执行。

5.3.4 睡眠模式

睡眠模式下的 I/O 状态

在睡眠模式下，所有 I/O 引脚的状态与运行模式下相同。

进入睡眠模式

当 Cortex®-M4 系统控制寄存器的 SLEEPDEEP 位清零时，将根据*进入低功耗模式*一节进入睡眠模式。

有关如何进入停止模式的详细信息，请参见*表 24：睡眠*。

退出睡眠模式

按照以下章节退出睡眠模式：退出低功耗模式。

有关如何退出睡眠模式的更多详细信息，请参见*表 24：睡眠*。

表 24. 睡眠

立即睡眠模式	说明
进入模式	WFI（等待中断）或 WFE（等待事件），且： – SLEEPDEEP = 0 – 没有中断（对于 WFI）或事件（对于 WFE）挂起 请参见 Cortex®-M4 系统控制寄存器。
	从 ISR 返回，且： – SLEEPDEEP = 0 及 – SLEEPONEXIT = 1 – 没有中断挂起 请参见 Cortex®-M4 系统控制寄存器。
退出模式	如果使用 WFI 或从 ISR 返回进入 中断：请参见*表 67：STM32L4Rxxx 和 STM32L4Sxxx 向量表* 如果使用 WFE 进入且 SEVONPEND = 0： 唤醒事件：请参见*第 16.3.2 节：唤醒事件管理* 如果使用 WFE 进入且 SEVONPEND = 1： 中断事件（当在 NVIC 中禁止时，请参见*表 67：STM32L4Rxxx 和 STM32L4Sxxx 向量表*）或唤醒事件（请参见*第 16.3.2 节：唤醒事件管理*）
唤醒延迟	无

5.3.5 低功耗睡眠模式（LP 睡眠）

有关稳压器和外设工作条件的详细信息，请参见产品数据手册。

低功耗睡眠模式下的 I/O 状态

在低功耗睡眠模式下，所有 I/O 引脚的状态与运行模式下相同。

进入低功耗睡眠模式

当 Cortex®-M4 系统控制寄存器的 SLEEPDEEP 位清零时，将根据*进入低功耗模式*一节从低功耗运行模式进入低功耗睡眠模式。

有关如何进入低功耗睡眠模式的详细信息，请参见*表 25：低功耗睡眠*。

退出低功耗睡眠模式

根据*退出低功耗模式*一节退出低功耗睡眠模式。当通过发出中断或事件来退出低功耗睡眠模式时，MCU 处于低功耗运行模式。

有关如何退出低功耗睡眠模式的详细信息，请参见*表 25：低功耗睡眠*。

表 25. 低功耗睡眠

低功耗立即睡眠模式	说明
进入模式	从低功耗运行模式进入低功耗睡眠模式。 WFI（等待中断）或 WFE（等待事件），且： – SLEEPDEEP = 0 – 没有中断（对于 WFI）或事件（对于 WFE）挂起 请参见 Cortex®-M4 系统控制寄存器。
	从低功耗运行模式进入低功耗睡眠模式。 从 ISR 返回，且： – SLEEPDEEP = 0 及 – SLEEPONEXIT = 1 – 没有中断挂起 请参见 Cortex®-M4 系统控制寄存器。
退出模式	如果使用 WFI 或从 ISR 返回进入 中断：请参见*表 67：STM32L4Rxxx 和 STM32L4Sxxx 向量表* 如果使用 WFE 进入且 SEVONPEND = 0： 唤醒事件：请参见*第 16.3.2 节：唤醒事件管理* 如果使用 WFE 进入且 SEVONPEND = 1： 中断（即使在 NVIC 中禁止时）请参见*表 67：STM32L4Rxxx 和 STM32L4Sxxx 向量表* 唤醒事件：请参见*第 16.3.2 节：唤醒事件管理* 当退出低功耗睡眠模式时，MCU 会处于低功耗运行模式。
唤醒延迟	无

5.3.6 停止 0 模式

停止 0 模式基于 Cortex®-M4 深度睡眠模式与外设时钟门控。稳压器配置为主稳压器模式。在停止 0 模式下，V_{CORE} 域中的所有时钟都会停止，PLL、MSI、HSI16 和 HSE 振荡器也被禁止。一些带有唤醒功能（I2Cx（x=1、2、3）、U(S)ARTx（x=1、2...5）和 LPUART）的外设可以开启 HSI16 以接收帧，如果该帧不是唤醒帧，也可以在接收到帧后关闭 HSI16。在这种情况下，HSI16 时钟仅传播到请求该时钟的外设中。

SRAM1、SRAM2 和寄存器内容将保留。

停止 0 模式中的 BOR 始终可用。当使用的阈值高于 V_{BOR0} 时，功耗会增加。

停止 0 模式下的 I/O 状态

在停止 0 模式下，所有 I/O 引脚的状态与运行模式下相同。

进入停止 0 模式

当Cortex®-M4 系统控制寄存器的 SLEEPDEEP 位置 1 时，将根据*进入低功耗模式*一节进入停止 0 模式。

有关如何进入停止 0 模式的详细信息，请参见*表 26：停止 0 模式*。

如果正在执行 Flash 编程，停止 0 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 0 模式的进入则延迟到 APB 访问结束后执行。

在停止 0 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 40.3 节：IWDG 功能说明*。
- 实时时钟 (RTC)：通过*备份域控制寄存器 (RCC_BDCR)* 中的 RTCEN 位进行配置。
- 内部 RC 振荡器 (LSI)：通过*控制/状态寄存器 (RCC_CSR)* 中的 LSION 位进行配置。
- 外部 32.768 kHz 振荡器 (LSE)：通过*备份域控制寄存器 (RCC_BDCR)* 中的 LSEON 位进行配置。

在停止 0 模式中可以使用多个外设，当这些外设已使能并且由 LSI 或 LSE 提供时钟，或者请求 HSI16 时钟 LPTIM1、LPTIM2、I2Cx（x=1、2、3、4）U(S)ARTx（x=1,2...5）、LPUART 时，会增加功耗。

在停止 0 模式下，可以使用 DACx（x=1、2）、OPAMP、比较器、PVMx（x=1、2、3、4）和 PVD。如果不需要上述功能，可以通过软件禁止功能，以节省其功耗。

在停止 0 模式期间，ADCx（x=1、2、3）、温度传感器和 VREFBUF 缓冲区会产生功耗，除非在进入该模式前将它们禁用。

退出停止 0 模式

根据*进入低功耗模式*一节退出停止 0 模式。

有关如何退出停止 0 模式的详细信息，请参见*表 26：停止 0 模式*。

通过发出中断或唤醒事件退出停止 0 模式时，如果*时钟配置寄存器 (RCC_CFGR)* 中的 STOPWUCK 位置 1，则选择 HSI16 RC 振荡器作为系统时钟。如果 STOPWUCK 位清零，则选择 MSI 振荡器作为系统时钟。当选择 HSI16 作为唤醒系统时钟时，唤醒时间更短。选择 MSI 能够以最高 48 MHz 的更高频率进行唤醒操作。

当稳压器在低功耗模式下工作时，将器件从使用 HSI16 的停止 0 模式唤醒将需要额外的启动延时。在停止 0 模式下一直开启内部稳压器虽然可以缩短启动时间，但功耗却增大。

当退出停止 0 模式时，MCU 处于运行模式（范围 1 或范围 2，具体取决于 PWR_CR1 中的 VOS 位），如果 PWR_CR1 寄存器中的 LPR 位置 1，MCU 会处于低功耗运行模式。

表 26. 停止 0 模式

停止 0 模式	说明
进入模式	WFI（等待中断）或 WFE（等待事件），且： – 将 Cortex®-M4 系统控制寄存器中的 SLEEPDEEP 位置 1 – 没有中断（对于 WFI）或事件（对于 WFE）挂起 – PWR_CR1 中的 LPMS = “000”
	从 ISR 返回，且： – 将 Cortex®-M4 系统控制寄存器中的 SLEEPDEEP 位置 1 – SLEEPONEXIT = 1 – 没有中断挂起 – PWR_CR1 中的 LPMS = “000”
	注： 要进入停止 0 模式，必须清零所有 EXTI 线挂起位（挂起寄存器 1 (EXTI_PR1) 中）和生成唤醒中断的外设标志。否则将忽略进入停止 0 模式这一过程，继续执行程序。
退出模式	如果使用 WFI 或从 ISR 返回进入 任何配置为中断模式的 EXTI 线（必须在 NVIC 中使能对应的 EXTI 中断向量）。中断源可以是外部中断或具有唤醒功能的外设。请参见*表 67：STM32L4Rxxx 和 STM32L4Sxxx 向量表*。 如果使用 WFE 进入且 SEVONPEND = 0： 任何配置为事件模式的 EXTI 线。请参见*第 16.3.2 节：唤醒事件管理*。 如果使用 WFE 进入且 SEVONPEND = 1： 任何配置为中断模式的 EXTI 线（即使在 NVIC 中禁止对应的 EXTI 中断向量）。中断源可以是外部中断或具有唤醒功能的外设。请参见*表 67：STM32L4Rxxx 和 STM32L4Sxxx 向量表*。 唤醒事件：请参见*第 16.3.2 节：唤醒事件管理*。
唤醒延迟	最长唤醒时间介于：MSI 或 HSI16 唤醒时间和 Flash 从停止 0 模式的唤醒时间。

5.3.7 停止 1 模式

在停止 1 模式下，主稳压器关闭，仅低功率稳压器开启，除此之外与停止 0 模式相同。可以从运行模式和低功耗运行模式进入停止 1 模式。

有关如何进入和退出停止 1 模式的详细信息，请参见*表 27：停止 1 模式*。

表 27. 停止 1 模式

停止 1 模式	说明
进入模式	WFI（等待中断）或 WFE（等待事件），且： – 将 Cortex®-M4 系统控制寄存器中的 SLEEPDEEP 位置 1 – 没有中断（对于 WFI）或事件（对于 WFE）挂起 – PWR_CR1 中的 LPMS = “001”
	从 ISR 返回，且： – 将 Cortex®-M4 系统控制寄存器中的 SLEEPDEEP 位置 1 – SLEEPONEXIT = 1 – 没有中断挂起 – PWR_CR1 中的 LPMS = “001”
	注： 要进入停止 1 模式，则必须清零所有 EXTI 线挂起位（挂起寄存器 1 (EXTI_PR1) 中）和生成唤醒中断的外设标志。否则将忽略进入停止 1 模式这一过程，继续执行程序。
退出模式	如果使用 WFI 或从 ISR 返回进入 任何配置为中断模式的 EXTI 线（必须在 NVIC 中使能对应的 EXTI 中断向量）。中断源可以是外部中断或具有唤醒功能的外设。请参见*表 67：STM32L4Rxxx 和 STM32L4Sxxx 向量表*。 如果使用 WFE 进入且 SEVONPEND = 0： 任何配置为事件模式的 EXTI 线。请参见*第 16.3.2 节：唤醒事件管理*。 如果使用 WFE 进入且 SEVONPEND = 1： 任何配置为中断模式的 EXTI 线（即使在 NVIC 中禁止对应的 EXTI 中断向量）。中断源可以是外部中断或具有唤醒功能的外设。请参见*表 67：STM32L4Rxxx 和 STM32L4Sxxx 向量表*。 唤醒事件：请参见*第 16.3.2 节：唤醒事件管理*。
唤醒延迟	最长唤醒时间介于：MSI 或 HSI16 唤醒时间和稳压器从低功耗模式唤醒时间 + Flash 从停止 1 模式的唤醒时间。

5.3.8 停止 2 模式

停止 2 模式基于 Cortex®-M4 深度睡眠模式与外设时钟门控。在停止 2 模式下，V_{CORE} 域中的所有时钟都会停止，PLL、MSI、HSI16 和 HSE 振荡器也被禁止。一些带有唤醒电容（I2C3 和 LPUART）的外设可以开启 HSI16 以获取帧，如果该帧不是唤醒帧，也可以在接收到帧后关闭 HSI16。在这种情况下，HSI16 时钟仅传播到请求该时钟的外设中。

SRAM1、SRAM2、SRAM3 和寄存器内容将保留。根据 PWR_CR1 寄存器中的 RRSTP 位配置决定保留或丢失 SRAM3 内容。默认情况下，在复位后，RRSTP 位复位，因此停止 2 期间会丢失 SRAM3 的内容。

停止 2 模式中 BOR 始终可用。当使用的阈值高于 V_{BOR0} 时，功耗会增加。

注： 停止 2 模式期间，会强制比较器、LPUART 输出和 LPTIM1 输出变为低速 (OSPEEDy=00)。

停止 2 模式下的 I/O 状态

在停止 2 模式下，所有 I/O 引脚的状态与运行模式下相同。

进入停止 2 模式

当Cortex®-M4 系统控制寄存器的 SLEEPDEEP 位置 1 时，将根据*进入低功耗模式*一节进入停止 2 模式。

有关如何进入停止 2 模式的详细信息，请参见*表 28：停止 2 模式*。

仅可以从运行模式进入停止 2 模式。无法从低功耗运行模式进入停止 2 模式。

如果正在执行 Flash 编程，停止 2 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 2 模式的进入则延迟到 APB 访问结束后执行。

在停止 2 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 40 节：独立看门狗 (IWDG)* 中的*第 40.3 节：IWDG 功能说明*。
- 实时时钟 (RTC)：通过*备份域控制寄存器 (RCC_BDCR)* 中的 RTCEN 位进行配置。
- 内部 RC 振荡器 (LSI)：通过*控制/状态寄存器 (RCC_CSR)* 中的 LSION 位进行配置。
- 外部 32.768 kHz 振荡器 (LSE)：通过*备份域控制寄存器 (RCC_BDCR)* 中的 LSEON 位进行配置。

在停止 2 模式中可以使用多个外设，当这些外设已使能并且由 LSI 或 LSE 提供时钟，或者外设请求 HSI16 时钟 LPTIM1、I2C3、LPUART 时，会增加功耗。

在停止 2 模式下，可以使用比较器、PVMx（x=1、2、3、4）和 PVD。如果不需要上述功能，可以通过软件禁止功能，以节省其功耗。

在停止 2 模式期间 ADCx、OPAMPx、DACx、温度传感器和 VREFBUF 缓冲区会产生功耗，除非在进入该模式前将它们禁用。

必须通过清零外设自身的使能位来禁止无法在停止 2 模式下使能的所有外设，或者通过将*AHB1 外设复位寄存器 (RCC_AHB1RSTR)*、*AHB2 外设复位寄存器 (RCC_AHB2RSTR)*、*AHB3 外设复位寄存器 (RCC_AHB3RSTR)*、*APB1 外设复位寄存器 1 (RCC_APB1RSTR1)*、*APB1 外设复位寄存器 2 (RCC_APB1RSTR2)*、*APB2 外设复位寄存器 (RCC_APB2RSTR)* 中的相应位置 1 使其处于复位状态。

退出停止 2 模式

根据*退出低功耗模式*一节退出停止 2 模式。

有关如何退出停止 2 模式的详细信息，请参见*表 28：停止 2 模式*。

通过发出中断或唤醒事件退出停止 2 模式时，如果*时钟配置寄存器 (RCC_CFGR)* 中的 STOPWUCK 位置 1，则选择 HSI16 RC 振荡器作为系统时钟。如果 STOPWUCK 位清零，则选择 MSI 振荡器作为系统时钟。当选择 HSI16 作为唤醒系统时钟时，唤醒时间更短。选择 MSI 能够以最高 48 MHz 的更高频率进行唤醒操作。

退出停止 2 模式后，MCU 处于运行模式（范围 1 或范围 2，具体取决于 PWR_CR1 中的 VOS 位）。

表 28. 停止 2 模式

停止 2 模式	说明
进入模式	WFI（等待中断）或 WFE（等待事件），且： – 将 Cortex®-M4 系统控制寄存器中的 SLEEPDEEP 位置 1 – 没有中断（对于 WFI）或事件（对于 WFE）挂起 – PWR_CR1 中的 LPMS = “010”
	从 ISR 返回，且： – 将 Cortex®-M4 系统控制寄存器中的 SLEEPDEEP 位置 1 – SLEEPONEXIT = 1 – 没有中断挂起 – PWR_CR1 中的 LPMS = “010”
	注：　要进入停止 2 模式，则必须清零所有 EXTI 线挂起位（挂起寄存器 1 (EXTI_PR1) 中）和生成唤醒中断的外设标志。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 或从 ISR 返回进入： 任何配置为中断模式的 EXTI 线（必须在 NVIC 中使能对应的 EXTI 中断向量）。中断源可以是外部中断或具有唤醒功能的外设。请参见*表 67：STM32L4Rxxx 和 STM32L4Sxxx 向量表*。 如果使用 WFE 进入且 SEVONPEND = 0： 任何配置为事件模式的 EXTI 线。请参见*第 16.3.2 节：唤醒事件管理*。 如果使用 WFE 进入且 SEVONPEND = 1： 任何配置为中断模式的 EXTI 线（即使在 NVIC 中禁止对应的 EXTI 中断向量）。中断源可以是外部中断或具有唤醒功能的外设。请参见*表 67：STM32L4Rxxx 和 STM32L4Sxxx 向量表*。 任何配置为事件模式的 EXTI 线。请参见*第 16.3.2 节：唤醒事件管理*。
唤醒延迟	最长唤醒时间介于：MSI 或 HSI16 唤醒时间和稳压器从低功耗模式唤醒时间 + Flash 从停止 2 模式的唤醒时间。

5.3.9 待机模式

待机模式下可以使用 BOR 达到最低功耗。待机模式基于 Cortex®-M4 深度睡眠模式，其中稳压器被禁止（保留 SRAM2 内容时除外）。PLL、HSI16、MSI 和 HSE 振荡器也会关闭。

除备份域和待机电路中的寄存器外，SRAM1 和寄存器内容都将丢失（请参见*图 7*）。当 PWR_CR3 寄存器中的 RRS 位置 1 时，SRAM2 内容保留。在这种情况下，低功耗稳压器开启并仅为 SRAM2 供电。

在待机模式下 BOR 始终可用。当使用的阈值高于 V_{BOR0} 时，功耗会增加。

待机模式下的 I/O 状态

在待机模式下，IO 默认处于浮空状态。如果 PWR_CR3 寄存器的 APC 位已置 1，则 I/O 可配置为上拉（请参见 PWR_PUCRx 寄存器（x=A、B、C、D、E、F、G、H））者下拉（请参见 PWR_PDCRx 寄存器（x=A、B、C、D、E、F、G、H）），或者在 PWR_PUCRx 或 PWR_PDCRx 寄存器均未置 1 的情况下保持模拟状态。同一 IO 的 PWR_PUCRx 和 PWR_PDCRx 都置 1 的情况下，下拉配置的优先级高于上拉配置。

一些 I/O（请参见*第 8.3.1 节：通用 I/O (GPIO)*）用于 JTAG/SW 调试，并且在待机模式期间只能通过将 PWR_PUCRx 或 PWR_PDCRx 寄存器中的相应位设置为“1”来将其配置为相应的复位上拉或下拉状态，或者将该位设置为“0”将其配置为浮空状态。

待机模式下可以使用 PC13 的 RTC 输出功能。也可以为 LSE 使用 PC14 和 PC15 功能。5 个唤醒引脚（WKUPx，x=1、2...5）和 3 个 RTC 入侵引脚可用。

进入待机模式

当 Cortex®-M4 系统控制寄存器的 SLEEPDEEP 位置 1 时，将根据*进入低功耗模式*一节进入待机模式。

有关如何进入待机模式的详细信息，请参见*表 29：待机模式*。

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 40 节：独立看门狗 (IWDG)*中的*第 40.3 节：IWDG 功能说明*。
- 实时时钟 (RTC)：通过备份域控制寄存器 (RCC_BDCR) 中的 RTCEN 位进行配置。
- 内部 RC 振荡器 (LSI)：通过控制/状态寄存器 (RCC_CSR) 中的 LSION 位进行配置。
- 外部 32.768 kHz 振荡器 (LSE)：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

根据*进入低功耗模式*一节退出待机模式。*电源控制寄存器 3 (PWR_CR3)* 中的 SBF 状态标志指示 MCU 已处于待机模式。从待机模式唤醒后，除*电源控制寄存器 3 (PWR_CR3)* 外，所有寄存器都将复位。

有关如何退出待机模式的详细信息，请参见*表 29：待机模式*。

退出待机模式时，待机期间通过 PWR_PUCRx 或 PWR_PDCRx 寄存器配置为上拉或下拉的 I/O 在刚刚退出待机模式时仍然保持此配置，直到 PWR_CR3 寄存器的 APC 位清零。一旦 APC 位清零，I/O 便会配置为复位值或根据 GPIOx_PUPDR 寄存器配置为上拉/下拉状态。但是，PWR_PUCRx 或 PWR_PDCRx 寄存器的内容不会丢失，并且在随后进入待机模式时可以重新使用。

一些 I/O（请参见*第 8.3.1 节：通用 I/O (GPIO)*）用于 JTAG/SW 调试且已在复位后激活内部上拉或下拉，因此在退出待机模式时会配置为此复位值。

对于在复位后预定义上拉或下拉的 IO（一些 JTAG/SW IO），或者具有从待机模式退出后完成 GPIOx_PUPDR 编程的 Io，如果这些编程与待机期间 PWR_PUCRx 或 PWR_PDCRx 编程的值不同，则会同时应用下拉和上拉，直到 APC 位清零，此时会释放 PWR_PUCRx 或 PWR_PDCRx 编程值。

表 29. 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®-M4 系统控制寄存器中的 SLEEPDEEP 位置 1 – 没有中断（对于 WFI）或事件（对于 WFE）挂起 – PWR_CR1 中的 LPMS = “011” – 清零功率状态寄存器 1 (PWR_SR1) 中的 WUFx 位
	从 ISR 返回，且： – 将 Cortex®-M4 系统控制寄存器中的 SLEEPDEEP 位置 1 – SLEEPONEXIT = 1 – 没有中断挂起 – PWR_CR1 中的 LPMS = “011”并且 – 清零功率状态寄存器 1 (PWR_SR1) 中的 WUFx 位 – 将与所选唤醒源（RTC 闹钟 A、RTC 闹钟 B、RTC 唤醒、入侵或时间戳标志）对应的 RTC 标志清零
退出模式	WKUPx 引脚边沿、RTC 事件、NRST 引脚的外部复位、IWDG 复位、BOR 复位
唤醒延迟	复位阶段

5.3.10 关断模式

关断模式下可达到最低功耗。待机模式基于深度睡眠模式，其中稳压器被禁止。因此 V_{CORE} 域断电。PLL、HSI16、MSI、LSI 和 HSE 振荡器也会关闭。

除备份域中的寄存器之外，SRAM1、SRAM2、SRAM3 和寄存器的内容都将丢失。在关断模式中 BOR 不可用。在此模式中电源电压监测不可用，因此不支持备份域的开启。

关断模式下的 I/O 状态

在关断模式下，IO 默认为浮空状态。如果 PWR_CR3 寄存器的 APC 位置 1，I/O 可用配置为上拉（请参见 PWR_PUCRx 寄存器（x=A、B、C、D、E、F、G、H）），或者下拉（请参见 PWR_PDCRx 寄存器（x=A、B、C、D、E、F、G、H）），或者在没有 PWR_PUCRx 或 PWR_PDCRx 寄存器置 1 的情况下保持模拟状态。如果同一 IO 的 PWR_PUCRx 和 PWR_PDCRx 都置 1，则下拉配置优先级最高，要高于上拉配置。当由于上电复位而退出关断模式时，此配置会丢失。

一些 I/O（请参见*第 8.3.1 节：通用 I/O (GPIO)*）用于 JTAG/SW 调试，并且在待机模式期间只能通过将 PWR_PUCRx 或 PWR_PDCRx 寄存器中的相应位设置为“1”来将其配置为相应的复位上拉或下拉状态，或者将该位设置为“0”将其配置为浮空状态。

关断模式下可以使用 PC13 的 RTC 输出功能。也可以为 LSE 使用 PC14 和 PC15 功能。5 个唤醒引脚（WKUPx，x=1、2...5）和 3 个 RTC 入侵引脚可用。

进入关断模式

当 Cortex®-M4 系统控制寄存器的 SLEEPDEEP 位置 1 时，将根据*进入低功耗模式*进入关断模式。

有关如何进入关断模式的详细信息，请参见*表 30：关断模式*。

在关断模式下，可以通过对各控制位进行编程来选择以下功能：

- 实时时钟 (RTC)：通过备份域控制寄存器 (RCC_BDCR) 中的 RTCEN 位进行配置。注意事项：如果 VDD 断电，则 RTC 内容丢失。
- 外部 32.768 kHz 振荡器 (LSE)：通过备份域控制寄存器 (RCC_BDCR) 中的 LSEON 位进行配置

退出关断模式

根据*退出低功耗模式*一节退出关断模式。从关断模式退出时会发生上电复位。从关断模式唤醒后会复位所有寄存器（除了备份域中的寄存器）。

有关如何退出关断模式的详细信息，请参见*表 30：关断模式*。

退出关断模式时，关断期间通过 PWR_PUCRx 或 PWR_PDCRx 寄存器配置为上拉或下拉的 I/O 将失去其配置，并将被配置为浮空状态或各自的上拉下拉复位值（对于*第 8.3.1 节：通用 I/O (GPIO)* 中列出的一些 I/O）。

表 30. 关断模式

关断模式	说明
进入模式	WFI（等待中断）或 WFE（等待事件），且： – 将 Cortex®-M4 系统控制寄存器中的 SLEEPDEEP 位置 1 – 没有中断（对于 WFI）或事件（对于 WFE）挂起 – PWR_CR1 中的 LPMS = “1XX” – 清零功率状态寄存器 1 (PWR_SR1) 中的 WUFx 位
	从 ISR 返回，且： – 将 Cortex®-M4 系统控制寄存器中的 SLEEPDEEP 位置 1 – SLEEPONEXT = 1 – 没有中断挂起 – PWR_CR1 中的 LPMS = “1XX”并且 – 清零功率状态寄存器 1 (PWR_SR1) 中的 WUFx 位 – 将与所选唤醒源（RTC 闹钟 A、RTC 闹钟 B、RTC 唤醒、入侵或时间戳标志）对应的 RTC 标志清零
退出模式	WKUPx 引脚边沿、RTC 事件、NRST 引脚的外部复位
唤醒延迟	复位阶段

5.3.11 从低功耗模式自动唤醒

RTC 用于在没有外部中断的情况下从低功耗模式唤醒 MCU（自动唤醒模式）。RTC 提供了可编程时基，便于定期从停止（0、1 或 2）或待机模式唤醒器件。为此，通过对 *备份域控制寄存器 (RCC_BDCR)* 中的 RTCSEL[1:0] 位进行编程，可以选择三个复用 RTC 时钟源中的其中两个：

- 低功耗 32.768 kHz 外部晶振 (LSE OSC)
  该时钟源能够以极低的功耗提供精确时基。
- 低功耗内部 RC 振荡器 (LSI)
  此时钟源的优势在于可以节省 32.768 kHz 晶振的成本。此内部 RC 振荡器非常省电。

要通过 RTC 闹钟事件从停止模式唤醒，必须：

- 将 EXTI 线 18 配置为上升沿有效
- 配置 RTC 以生成 RTC 闹钟

要从待机模式唤醒，无需配置 EXTI 线 18。

要通过 RTC 唤醒事件从停止模式唤醒，必须：

- 将 EXTI 线 20 配置上升沿有效
- 配置 RTC 以生成 RTC 闹钟

要从待机模式唤醒，则不需要配置 EXTI 线 20。

5.4 PWR 寄存器

外设寄存器可支持半字（16 位）或字（32 位）访问。

5.4.1 电源控制寄存器 1 (PWR_CR1)

Power control register 1

偏移地址：0x00

复位值：0x0000 0200。从待机模式唤醒后，此寄存器会复位。

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.	LPR	Res.	Res.	Res.	VOS[1:0]		DBP	Res.	Res.	Res.	RRSTP	Res.	LPMS[2:0]		
	rw				rw	rw	rw				rw		rw	rw	rw

位 31:15 保留，必须保持复位值。

位 14 **LPR**：低功耗运行 (Low-power run)
当此位置 1 时，稳压器会从主模式 (MR) 切换为低功耗模式 (LPR)。
注： 当 LPR 位置 1 时无法进入停止 2 模式。而是会进入停止 1 模式。

位 13:11 保留，必须保持复位值。

位 10:9 **VOS**：电压调节范围选择 (Voltage scaling range selection)
00：无法写入（由硬件禁止）
01：范围 1
10：范围 2
11：无法写入（由硬件禁止）

位 8 **DBP**：禁止备份域写保护 (Disable backup domain write protection)
在复位状态下，RTC 和备份寄存器均受到保护，以防止寄生写访问。必须将此位置 1 才能使能对这些寄存器的写访问。
0：禁止对 RTC 和备份寄存器的访问
1：使能对 RTC 和备份寄存器的访问

位 7:5 保留，必须保持复位值。

位 4 **RRSTP**：停止 2 模式下 SRAM3 内容保留 (SRAM3 retention in Stop 2 mode)
0：停止 2 模式下 SRAM3 断电（SRAM3 内容丢失）
1：停止 2 模式下 SRAM3 供电（SRAM3 内容保留）

位 3 保留，必须保持复位值。

位 2:0 **LPMS[2:0]**：低功耗模式选择 (Low-power mode selection)
当 CPU 进入深睡眠模式时，使用这些位选择进入的低功耗模式。
000：停止 0 模式
001：停止 1 模式
010：停止 2 模式
011：待机模式
1xx：关断模式
注： 如果 LPR 位置 1，则无法选择停止 2 模式，因此会进入停止 1 模式而不是停止 2 模式。
在待机模式下，SRAM2 内容可以保留或丢失，具体取决于 PWR_CR3 中的 RRS 位。

5.4.2 电源控制寄存器 2 (PWR_CR2)

Power control register 2

偏移地址：0x04

复位值：0x0000 0000。退出待机模式时，寄存器会复位。

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.	Res.	Res.	Res.	USV	IOSV	Res.	PVME4	PVME3	PVME2	PVME1	PLS[2:0]			PVDE
					rw	rw		rw	rw	rw	rw	rw	rw	rw	rw

位 31:11 保留，必须保持复位值。

位 10 **USV**：V_{DDUSB} USB 电源有效 (V_{DDUSB} USB supply valid)
此位用于验证 V_{DDUSB} 电源是否已应用电气和逻辑隔离。
必须将将此位置 1 才能使用 USB OTG_FS 外设。如果应用中并非始终存在 V_{DDUSB}，则可以使用 PVM 确定电源是否就绪。
0：V_{DDUSB} 不存在。应用逻辑和电气隔离以忽略此电源。
1：V_{DDUSB} 有效。

位 9 **IOSV**：V_{DDIO2} 独立 I/O 电源有效 (V_{DDIO2} Independent I/Os supply valid)
此位用于验证 V_{DDIO2} 电源是否已应用电气和逻辑隔离。
必须将将此位置 1 才能使用 PG[15:2]。如果应用中并非始终存在 V_{DDIO2}，则可以使用 PVM 确定此电源是否就绪。
0：V_{DDIO2} 不存在。应用逻辑和电气隔离以忽略此电源。
1：V_{DDIO2} 有效。

位 8 保留，必须保持复位值。

位 7 **PVME4**：外设电压监测 4 使能：V_{DDA} 与 2.2V (Peripheral voltage monitoring 4 enable: V_{DDA} vs. 2.2V)
0：PVM4（以 2.2V 阈值监测 V_{DDA}）禁止。
1：PVM4（以 2.2V 阈值监测 V_{DDA}）使能。

位 6 **PVME3**：外设电压监测 3 使能：V_{DDA} 与 1.62V (Peripheral voltage monitoring 3 enable: V_{DDA} vs. 1.62V)

0：PVM3（以 1.62V 阈值监测 V_{DDA}）禁止。
1：PVM3（以 1.62V 阈值监测 V_{DDA}）使能。

位 5 **PVME2**：外设电压监测 2 使能：V_{DDA} 与 0.9V (Peripheral voltage monitoring 2 enable: V_{DDA} vs. 0.9V)

0：PVM2（以 0.9V 阈值监测 V_{DDIO2}）禁止。
1：PVM2（以 0.9V 阈值监测 V_{DDIO2}）使能。

位 4 **PVME1**：外设电压监测 1 使能：V_{DDUSB} 与 1.2V (Peripheral voltage monitoring 1 enable: V_{DDUSB} vs. 1.2V)

0：PVM1（以 1.2V 阈值监测 V_{DDUSB}）禁止。
1：PVM1（以 1.2V 阈值监测 V_{DDUSB}）使能。

位 3:1 **PLS[2:0]**：电源电压检测器电平选择 (Power voltage detector level selection)

这些位用于选择电源电压检测器检测的电压阈值：
000：约 2.0 V 的 V_{PVD0}
001：约 2.2 V 的 V_{PVD1}
010：约 2.4 V 的 V_{PVD2}
011：约 2.5 V 的 V_{PVD3}
100：约 2.6 V 的 V_{PVD4}
101：约 2.8 V 的 V_{PVD5}
110：约 2.9 V 的 V_{PVD6}
111：外部输入模拟电压 PVD_IN（内部与 VREFINT 进行比较）

注：　当 SYSCFG_CBR 寄存器中的 PVDL（PVD 锁定）位置 1 时，这些位受到写保护。这些位仅由系统复位进行复位。

位 0 **PVDE**：电源电压检测器使能 (Power voltage detector enable)

0：禁止电源电压检测器。
1：使能电源电压检测器

注：　当 SYSCFG_CBR 寄存器中的 PVDL（PVD 锁定）位置 1 时，该位受到写保护。此位仅由系统复位进行复位。

5.4.3 电源控制寄存器 3 (PWR_CR3)

Power control register 3

偏移地址：0x08

复位值：0x0000 8000。该寄存器在退出待机模式时不会复位，而是根据 RCC_APB1RSTR1 寄存器中的 PWRRST 位执行复位。

访问：与标准 APB 访问（3 个写周期和 2 个读周期）相比，访问该寄存器需要额外的 APB 周期。

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
EIWUL	Res.	Res.	DSIPDEN	Res.	APC	Res.	RRS	Res.	Res.	Res.	EWUP5	EWUP4	EWUP3	EWUP2	EWUP1
rw			rw		rw		rw				rw	rw	rw	rw	rw

位 31:16 保留，必须保持复位值。

位 15 **EIWUL**：内部唤醒线使能 (Enable internal wakeup line)
0：禁止内部唤醒线。
1：使能内部唤醒线。

位 14:13 保留，必须保持复位值。

位 12 **DSIPDEN**：使能 DSI 引脚上的下拉激活 (Enable Pull-down activation on DSI pins)
1：使能 DSI 引脚上的下拉。
0：禁止 DSI 引脚上的下拉。

位 11 保留，必须保持复位值。

位 10 **APC**：应用上拉和下拉配置 (Apply pull-up and pull-down configuration)
当该位置 1，应用 PWR_PUCRx 和 PWR_PDCRx 寄存器中定义的 I/O 上拉和下拉配置。当该位清零时，不会将 PWR_PUCRx 和 PWR_PDCRx 寄存器中的设置应用于 I/O，而是 I/O 在待机期间会处于浮空模式，或在运行模式期间根据 GPIO 控制器 GPIOx_PUPDR 寄存器进行配置。

位 9 保留，必须保持复位值。

位 8 **RRS**：待机模式下 SRAM2 内容保留 (SRAM2 retention in Standby mode)
0：待机模式下 SRAM2 断电（SRAM2 内容丢失）。
1：待机模式下使用低功耗稳压器接通 SRAM2（SRAM2 内容保留）。

位 7:5 保留，必须保持复位值。

位 4 **EWUP5**：使能唤醒引脚 WKUP5 (Enable Wakeup pin WKUP5)
该位置 1 会使能外部唤醒引脚 WKUP5，并在出现上升沿或下降沿时从待机或关断事件中触发唤醒。通过 PWR_CR4 寄存器中的 WP5 位配置有效边沿。

位 3 **EWUP4**：使能唤醒引脚 WKUP4 (Enable Wakeup pin WKUP4)
该位置 1 会使能外部唤醒引脚 WKUP4，并在出现上升沿或下降沿时从待机或关断事件中触发唤醒。通过 PWR_CR4 寄存器中的 WP4 位配置有效边沿。

位 2 **EWUP3**：使能唤醒引脚 WKUP3 (Enable Wakeup pin WKUP3)
该位置 1 会使能外部唤醒引脚 WKUP3，并在出现上升沿或下降沿时从待机或关断事件中触发唤醒。通过 PWR_CR4 寄存器中的 WP3 位配置有效边沿。

位 1 **EWUP2**：使能唤醒引脚 WKUP2 (Enable Wakeup pin WKUP2)
该位置 1 会使能外部唤醒引脚 WKUP2，并在出现上升沿或下降沿时从待机或关断事件中触发唤醒。通过 PWR_CR4 寄存器中的 WP2 位配置有效边沿。

位 0 **EWUP1**：使能唤醒引脚 WKUP1 (Enable Wakeup pin WKUP1)
该位置 1 会使能外部唤醒引脚 WKUP1，并在出现上升沿或下降沿时从待机或关断事件中触发唤醒。通过 PWR_CR4 寄存器中的 WP1 位配置有效边沿。

5.4.4 电源控制寄存器 4 (PWR_CR4)

Power control register 4

偏移地址：0x0C

复位值：0x0000 0000。该寄存器在退出待机模式时不会复位，而是根据 RCC_APB1RSTR1 寄存器中的 PWRRST 位执行复位。

访问：与标准 APB 访问（3 个写周期和 2 个读周期）相比，访问该寄存器需要额外的 APB 周期。

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.	Res.	Res.	Res.	Res.	VBRS	VBE	Res.	Res.	Res.	WP5	WP4	WP3	WP2	WP1
						rw	rw				rw	rw	rw	rw	rw

位 31:10 保留，必须保持复位值。

位 9 **VBRS**：V_{BAT} 电池充电电阻选择 (V_{BAT} battery charging resistor selection)
0：通过 5 kΩ 电阻为 V_{BAT} 充电
1：通过 1.5 kΩ 电阻为 V_{BAT} 充电

位 8 **VBE**：V_{BAT} 电池充电使能 (V_{BAT} battery charging enable)
0：禁止 V_{BAT} 电池充电
1：使能 V_{BAT} 电池充电

位 7:5 保留，必须保持复位值。

位 4 **WP5**：唤醒引脚 WKUP5 极性 (Wakeup pin WKUP5 polarity)
该位用于定义在外部唤醒引脚 WKUP5 上进行事件检测所用的极性
0：在高电平（上升沿）检测
1：在低电平（下降沿）检测

位 3 **WP4**：唤醒引脚 WKUP4 极性 (Wakeup pin WKUP4 polarity)
该位用于定义在外部唤醒引脚 WKUP4 上进行事件检测所用的极性
0：在高电平（上升沿）检测
1：在低电平（下降沿）检测

位 2 **WP3**：唤醒引脚 WKUP3 极性 (Wakeup pin WKUP3 polarity)
该位用于定义在外部唤醒引脚 WKUP3 上进行事件检测所用的极性
0：在高电平（上升沿）检测
1：在低电平（下降沿）检测

位 1 **WP2**：唤醒引脚 WKUP2 极性 (Wakeup pin WKUP2 polarity)
该位用于定义在外部唤醒引脚 WKUP2 上进行事件检测所用的极性
0：在高电平（上升沿）检测
1：在低电平（下降沿）检测

位 0 **WP1**：唤醒引脚 WKUP1 极性 (Wakeup pin WKUP1 polarity)
该位用于定义在外部唤醒引脚 WKUP1 上进行事件检测所用的极性
0：在高电平（上升沿）检测
1：在低电平（下降沿）检测

5.4.5 电源状态寄存器 1 (PWR_SR1)

Power status register 1

偏移地址：0x10

复位值：0x0000 0000。该寄存器在退出待机模式时不会复位，而是根据 RCC_APB1RSTR1 寄存器中的 PWRRST 位执行复位。

与标准 APB 读操作相比，读取此寄存器需要 2 个额外的 APB 周期。

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
WUFI	Res.	Res.	Res.	Res.	Res.	Res.	SBF	Res.	Res.	Res.	WUF5	WUF4	WUF3	WUF2	WUF1
r							r				r	r	r	r	r

位 31:16 保留，必须保持复位值。

位 15 **WUFI**：内部唤醒标志 (Wakeup flag internal)
在内部唤醒线上检测到唤醒事件时，该位置 1。清除所有内部唤醒源时，该位清零。

位 14:9 保留，必须保持复位值。

位 8 **SBF**：待机标志 (Standby flag)
当器件进入待机模式时，该位由硬件置 1，通过将 PWR_SCR 寄存器的 CSBF 位置 1 或通过上电复位，该位清零。该位不通过系统复位清零。
0：器件未进入待机模式
1：器件进入待机模式

位 7:5 保留，必须保持复位值。

位 4 **WUF5**：唤醒标志 5 (Wakeup flag 5)
当在 WKUP5 唤醒引脚上检测到唤醒事件时该位置 1。通过向 PWR_SCR 寄存器中的 CWUF5 写入 1 进行清零。

位 3 **WUF4**：唤醒标志 4 (Wakeup flag 4)
当在 WKUP4 唤醒引脚上检测到唤醒事件时该位置 1。通过向 PWR_SCR 寄存器中的 CWUF4 写入 1 进行清零。

位 2 **WUF3**：唤醒标志 3 (Wakeup flag 3)
当在 WKUP3 唤醒引脚上检测到唤醒事件时该位置 1。通过向 PWR_SCR 寄存器中的 CWUF3 写入 1 进行清零。

位 1 **WUF2**：唤醒标志 2 (Wakeup flag 2)
当在 WKUP2 唤醒引脚上检测到唤醒事件时该位置 1。通过向 PWR_SCR 寄存器中的 CWUF2 写入 1 进行清零。

位 0 **WUF1**：唤醒标志 1 (Wakeup flag 1)
当在 WKUP1 唤醒引脚上检测到唤醒事件时该位置 1。通过向 PWR_SCR 寄存器中的 CWUF1 写入 1 进行清零。

5.4.6 电源状态寄存器 2 (PWR_SR2)

Power status register 2

偏移地址：0x14

复位值：0x0000 0000。退出待机/关断模式后，寄存器会部分复位。

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
PVMO4	PVMO3	PVMO2	PVMO1	PVDO	VOSF	REGLPF	REGLPS	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
r	r	r	r	r	r	r	r								

位 31:16 保留，必须保持复位值。

位 15 **PVMO4**：外设电压监测输出：V_{DDA} 与 2.2 V (Peripheral voltage monitoring output: V_{DDA} vs. 2.2 V)

0：V_{DDA} 电压高于 PVM4 阈值（约 2.2 V）。

1：V_{DDA} 电压低于 PVM4 阈值（约 2.2 V）。

注：　禁止 PVM4 时 (PVME4 = 0) 会清零 PVMO4。使能 PVM4 后经过 PVM4 唤醒时间，PVM4 输出开始有效。

位 14 **PVMO3**：外设电压监测输出：V_{DDA} 与 1.62 V (Peripheral voltage monitoring output: V_{DDA} vs. 1.62 V)

0：V_{DDA} 电压高于 PVM3 阈值（约 1.62 V）。

1：V_{DDA} 电压低于 PVM3 阈值（约 1.62 V）。

注：　禁止 PVM3 时 (PVME3 = 0) 会清零 PVMO3。使能 PVM3 后经过 PVM3 唤醒时间，PVM3 输出开始有效。

位 13 **PVMO2**：外设电压监测输出：V_{DDIO2} 与 0.9 V (Peripheral voltage monitoring output: V_{DDIO2} vs. 0.9 V)

0：V_{DDIO2} 电压高于 PVM2 阈值（约 0.9 V）。

1：V_{DDIO2} 电压低于 PVM2 阈值（约 0.9 V）。

注：　禁止 PVM2 时 (PVME2 = 0) 会清零 PVMO2。使能 PVM2 后经过 PVM2 唤醒时间，PVM2 输出开始有效。

位 12 **PVMO1**：外设电压监测输出：V_{DDUSB} 与 1.2 V (Peripheral voltage monitoring output: V_{DDUSB} vs. 1.2 V)

0：V_{DDUSB} 电压高于 PVM1 阈值（约 1.2 V）。

1：V_{DDUSB} 电压低于 PVM1 阈值（约 1.2 V）。

注：　禁止 PVM1 时 (PVME1 = 0) 会清零 PVMO1。使能 PVM1 后经过 PVM1 唤醒时间，PVM1 输出开始有效。

位 11 **PVDO**：电源电压检测器输出 (Power voltage detector output)

0：V_{DD} 高于所选 PVD 阈值

1：V_{DD} 低于所选 PVD 阈值

位 10 **VOSF**：电压调节标志 (Voltage scaling flag)

电压调节变更后，内部稳压器准备好需要一定的延时。VOSF 用于指示稳压器已达到 PWR_CR1 寄存器中 VOS 位所定义的电压电平。

0：稳压器在所选电压范围内准备就绪

1：稳压器输出电压更改为所需电平

位 9 **REGLPF**：低功耗稳压器标志 (Low-power regulator flag)

MCU 处于低功耗运行模式时，此位由硬件置 1。当 MCU 退出低功耗运行模式时，该位保持为 1，直到稳压器在主模式下准备就绪。在提高产品频率之前必须对该位进行轮询。

当稳压器准备就绪后，此位由硬件清零。

0：稳压器在主模式 (MR) 下准备就绪

1：稳压器处于低功耗模式 (LPR)

位 8 **REGLPS**：低功耗稳压器启动 (Low-power regulator started)

此位用于提供上电、复位或待机/关断后低功耗稳压器是否准备就绪的信息。如果在 REGLPS 位仍然清零的情况下进入待机模式，则会增加从待机模式唤醒的时间。

0：低功耗稳压器未准备就绪

1：低功耗稳压器已准备就绪

位 7:0 保留，必须保持复位值。

5.4.7 电源状态清零寄存器 (PWR_SCR)

Power status clear register

偏移地址：0x18

复位值：0x0000 0000。

与标准的 APB 写操作相比，写入此寄存器需要 3 个额外的 APB 周期。

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.	Res.	Res.	Res.	Res.	Res.	CSBF	Res.	Res.	Res.	CWUF5	CWUF4	CWUF3	CWUF2	CWUF1
							w				w	w	w	w	w

位 31:9 保留，必须保持复位值。

位 8 **CSBF**：将待机标志清零 (Clear standby flag)

将此位置 1 会清零 PWR_SR1 寄存器中的 SBF 标志。

位 7:5 保留，必须保持复位值。

位 4 **CWUF5**：唤醒标志 5 清零 (Clear wakeup flag 5)

将此位置 1 会清零 PWR_SR1 寄存器中的 WUF5 标志。

位 3 **CWUF4**：唤醒标志 4 清零 (Clear wakeup flag 4)

将此位置 1 会清零 PWR_SR1 寄存器中的 WUF4 标志。

位 2 **CWUF3**：唤醒标志 3 清零 (Clear wakeup flag 3)

将此位置 1 会清零 PWR_SR1 寄存器中的 WUF3 标志。

位 1 **CWUF2**：唤醒标志 2 清零 (Clear wakeup flag 2)

将此位置 1 会清零 PWR_SR1 寄存器中的 WUF2 标志。

位 0 **CWUF1**：唤醒标志 1 清零 (Clear wakeup flag 1)

将此位置 1 会清零 PWR_SR1 寄存器中的 WUF1 标志。

5.4.8 电源端口 A 上拉控制寄存器 (PWR_PUCRA)

Power Port A pull-up control register

偏移地址：0x20。

复位值：0x0000 0000。该寄存器在退出待机模式时不会复位，而是根据 RCC_APB1RSTR1 寄存器中的 PWRRST 位执行复位。

访问：与标准 APB 访问（3 个写周期和 2 个读周期）相比，访问该寄存器需要额外的 APB 周期。

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
PU15	Res.	PU13	PU12	PU11	PU10	PU9	PU8	PU7	PU6	PU5	PU4	PU3	PU2	PU1	PU0
rw		rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:16 保留，必须保持复位值。

位 15 **PU15**：端口 A 上拉位 15 (Port A pull-up bit 15)
在 PWR_CR3 寄存器中的 APC 位置 1 时将此位置 1，会激活 PA[15] 的上拉。
如果相应的 PD15 位也置 1，则会激活具有最高优先级的下拉，而不是激活上拉。

位 14 保留，必须保持复位值。

位 13:0 **PUy**：端口 A 上拉位 y (Port A pull-up bit y) (y=0..13)
在 PWR_CR3 寄存器中的 APC 位置 1 时将此位置 1，会激活 PA[y] 的上拉。如果相应的 PDy 位也置 1，则不激活上拉。

5.4.9 电源端口 A 下拉控制寄存器 (PWR_PDCRA)

Power Port A pull-down control register

偏移地址：0x24。

复位值：0x0000 0000。该寄存器在退出待机模式时不会复位，而是根据 RCC_APB1RSTR1 寄存器中的 PWRRST 位执行复位。

访问：与标准 APB 访问（3 个写周期和 2 个读周期）相比，访问该寄存器需要额外的 APB 周期。

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.	PD14	Res.	PD12	PD11	PD10	PD9	PD8	PD7	PD6	PD5	PD4	PD3	PD2	PD1	PD0
	rw		rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:15 保留，必须保持复位值。

位 14 **PD14**：端口 A 下拉位 14 (Port A pull-down bit 14)
在 PWR_CR3 寄存器中的 APC 位置 1 时将此位置 1，会激活 PA[14] 的下拉。

位 13 保留，必须保持复位值。

位 12:0 **PDy**：端口 A 下拉位 y (Port A pull-down bit y) (y=0..12)
在 PWR_CR3 寄存器中的 APC 位置 1 时将此位置 1，会激活 PA[y] 的下拉。

5.4.10 电源端口 B 上拉控制寄存器 (PWR_PUCRB)

Power Port B pull-up control register

偏移地址：0x28。

复位值：0x0000 0000。该寄存器在退出待机模式时不会复位，而是根据 RCC_APB1RSTR1 寄存器中的 PWRRST 位执行复位。

访问：与标准 APB 访问（3 个写周期和 2 个读周期）相比，访问该寄存器需要额外的 APB 周期。

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
PU15	PU14	PU13	PU12	PU11	PU10	PU9	PU8	PU7	PU6	PU5	PU4	PU3	PU2	PU1	PU0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:16 保留，必须保持复位值。

位 15:0 **PUy**：端口 B 上拉位 y (y=0..15)
在 PWR_CR3 寄存器中的 APC 位置 1 时将此位置 1，会激活 PB[y] 的上拉。如果相应的 PDy 位也置 1，则不激活上拉。

5.4.11 电源端口 B 下拉控制寄存器 (PWR_PDCRB)

Power Port B pull-down control register

偏移地址：0x2C。

复位值：0x0000 0000。该寄存器在退出待机模式时不会复位，而是根据 RCC_APB1RSTR1 寄存器中的 PWRRST 位执行复位。

访问：与标准 APB 访问（3 个写周期和 2 个读周期）相比，访问该寄存器需要额外的 APB 周期。

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
PD15	PD14	PD13	PD12	PD11	PD10	PD9	PD8	PD7	PD6	PD5	Res.	PD3	PD2	PD1	PD0
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 15:5 **PDy**：端口 B 下拉位 y (Port B pull-down bit y) (y=5..15)
在 PWR_CR3 寄存器中的 APC 位置 1 时将此位置 1，会激活 PB[y] 的下拉。

位 4 保留，必须保持复位值。

位 3:0 **PDy**：端口 B 下拉位 y (Port B pull-down bit y) (y=0..3)
在 PWR_CR3 寄存器中的 APC 位置 1 时将此位置 1，会激活 PB[y] 的下拉。

5.4.12 电源端口 C 上拉控制寄存器 (PWR_PUCRC)

Power Port C pull-up control register

偏移地址：0x30。

复位值：0x0000 0000。该寄存器在退出待机模式时不会复位，而是根据 RCC_APB1RSTR1 寄存器中的 PWRRST 位执行复位。

访问：与标准 APB 访问（3 个写周期和 2 个读周期）相比，访问该寄存器需要额外的 APB 周期。

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
PU15	PU14	PU13	PU12	PU11	PU10	PU9	PU8	PU7	PU6	PU5	PU4	PU3	PU2	PU1	PU0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:16 保留，必须保持复位值。

位 15:0 **PUy**：端口 C 上拉位 y (Port C pull-up bit y) (y=0..15)
在 PWR_CR3 寄存器中的 APC 位置 1 时将此位置 1，会激活 PC[y] 的上拉。如果相应的 PDy 位也置 1，则不激活上拉。

5.4.13 电源端口 C 下拉控制寄存器 (PWR_PDCRC)

Power Port C pull-down control register

偏移地址：0x34。

复位值：0x0000 0000。该寄存器在退出待机模式时不会复位，而是根据 RCC_APB1RSTR1 寄存器中的 PWRRST 位执行复位。

访问：与标准 APB 访问（3 个写周期和 2 个读周期）相比，访问该寄存器需要额外的 APB 周期。

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
PD15	PD14	PD13	PD12	PD11	PD10	PD9	PD8	PD7	PD6	PD5	PD4	PD3	PD2	PD1	PD0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:16 保留，必须保持复位值。

位 15:0 **PDy**：端口 C 下拉位 y (Port C pull-down bit y) (y=0..15)
在 PWR_CR3 寄存器中的 APC 位置 1 时将此位置 1，会激活 PC[y] 的下拉。

5.4.14 电源端口 D 上拉控制寄存器 (PWR_PUCRD)

Power Port D pull-up control register

偏移地址：0x38。

复位值：0x0000 0000。该寄存器在退出待机模式时不会复位，而是根据 RCC_APB1RSTR1 寄存器中的 PWRRST 位执行复位。

访问：与标准 APB 访问（3 个写周期和 2 个读周期）相比，访问该寄存器需要额外的 APB 周期。

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
PU15	PU14	PU13	PU12	PU11	PU10	PU9	PU8	PU7	PU6	PU5	PU4	PU3	PU2	PU1	PU0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:16 保留，必须保持复位值。

位 15:0 **PUy**：端口 D 上拉位 y (Port D pull-up bit y) (y=0..15)
在 PWR_CR3 寄存器中的 APC 位置 1 时将此位置 1，会激活 PD[y] 的上拉。如果相应的 PDy 位也置 1，则不激活上拉。

5.4.15 电源端口 D 下拉控制寄存器 (PWR_PDCRD)

Power Port D pull-down control register

偏移地址：0x3C。

复位值：0x0000 0000。该寄存器在退出待机模式时不会复位，而是根据 RCC_APB1RSTR1 寄存器中的 PWRRST 位执行复位。

访问：与标准 APB 访问（3 个写周期和 2 个读周期）相比，访问该寄存器需要额外的 APB 周期。

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
PD15	PD14	PD13	PD12	PD11	PD10	PD9	PD8	PD7	PD6	PD5	PD4	PD3	PD2	PD1	PD0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:16 保留，必须保持复位值。

位 15:0 **PDy**：端口 D 下拉位 y (Port D pull-down bit y) (y=0..15)
在 PWR_CR3 寄存器中的 APC 位置 1 时将此位置 1，会激活 PD[y] 的下拉。

5.4.16 电源端口 E 上拉控制寄存器 (PWR_PUCRE)

Power Port E pull-up control register

偏移地址：0x20。

复位值：0x0000 0000。该寄存器在退出待机模式时不会复位，而是根据 RCC_APB1RSTR1 寄存器中的 PWRRST 位执行复位。

访问：与标准 APB 访问（3 个写周期和 2 个读周期）相比，访问该寄存器需要额外的 APB 周期。

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
PU15	PU14	PU13	PU12	PU11	PU10	PU9	PU8	PU7	PU6	PU5	PU4	PU3	PU2	PU1	PU0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:16 保留，必须保持复位值。

位 15:0 **PUy**：端口 E 上拉位 y (Port E pull-up bit y) (y=0..15)
在 PWR_CR3 寄存器中的 APC 位置 1 时将此位置 1，会激活 PE[y] 的上拉。如果相应的 PDy 位也置 1，则不激活上拉。

5.4.17 电源端口 E 下拉控制寄存器 (PWR_PDCRE)

Power Port E pull-down control register

偏移地址：0x44。

复位值：0x0000 0000。该寄存器在退出待机模式时不会复位，而是根据 RCC_APB1RSTR1 寄存器中的 PWRRST 位执行复位。

访问：与标准 APB 访问（3 个写周期和 2 个读周期）相比，访问该寄存器需要额外的 APB 周期。

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
PD15	PD14	PD13	PD12	PD11	PD10	PD9	PD8	PD7	PD6	PD5	PD4	PD3	PD2	PD1	PD0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:16 保留，必须保持复位值。

位 15:0 **PDy**：端口 E 下拉位 y (Port E pull-down bit y) (y=0..15)
在 PWR_CR3 寄存器中的 APC 位置 1 时将此位置 1，会激活 PE[y] 的下拉。

5.4.18 电源端口 F 上拉控制寄存器 (PWR_PUCRF)

Power Port F pull-up control register

偏移地址：0x48。

复位值：0x0000 0000。该寄存器在退出待机模式时不会复位，而是根据 RCC_APB1RSTR1 寄存器中的 PWRRST 位执行复位。

访问：与标准 APB 访问（3 个写周期和 2 个读周期）相比，访问该寄存器需要额外的 APB 周期。

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
PU15	PU14	PU13	PU12	PU11	PU10	PU9	PU8	PU7	PU6	PU5	PU4	PU3	PU2	PU1	PU0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:16 保留，必须保持复位值。

位 15:0 **PUy**：端口 F 上拉位 y (Port F pull-up bit y) (y=0..15)
在 PWR_CR3 寄存器中的 APC 位置 1 时将此位置 1，会激活 PF[y] 的上拉。如果相应的 PDy 位也置 1，则不激活上拉。

5.4.19 电源端口 F 下拉控制寄存器 (PWR_PDCRF)

Power Port F pull-down control register

偏移地址：0x4C。

复位值：0x0000 0000。该寄存器在退出待机模式时不会复位，而是根据 RCC_APB1RSTR1 寄存器中的 PWRRST 位执行复位。

访问：与标准 APB 访问（3 个写周期和 2 个读周期）相比，访问该寄存器需要额外的 APB 周期。

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
PD15	PD14	PD13	PD12	PD11	PD10	PD9	PD8	PD7	PD6	PD5	PD4	PD3	PD2	PD1	PD0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:16 保留，必须保持复位值。

位 15:0 **PDy**：端口 F 下拉位 y (Port F pull-down bit y) (y=0..15)
在 PWR_CR3 寄存器中的 APC 位置 1 时将此位置 1，会激活 PF[y] 的下拉。

5.4.20 电源端口 G 上拉控制寄存器 (PWR_PUCRG)

Power Port G pull-up control register

偏移地址：0x50。

复位值：0x0000 0000。该寄存器在退出待机模式时不会复位，而是根据 RCC_APB1RSTR1 寄存器中的 PWRRST 位执行复位。

访问：与标准 APB 访问（3 个写周期和 2 个读周期）相比，访问该寄存器需要额外的 APB 周期。

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
PU15	PU14	PU13	PU12	PU11	PU10	PU9	PU8	PU7	PU6	PU5	PU4	PU3	PU2	PU1	PU0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:16 保留，必须保持复位值。

位 15:0 **PUy**：端口 G 上拉位 y (Port G pull-up bit y) (y=0..15)
在 PWR_CR3 寄存器中的 APC 位置 1 时将此位置 1，会激活 PG[y] 的上拉。如果相应的 PDy 位也置 1，则不激活上拉。

5.4.21 电源端口 G 下拉控制寄存器 (PWR_PDCRG)

Power Port G pull-down control register

偏移地址：0x54。

复位值：0x0000 0000。该寄存器在退出待机模式时不会复位，而是根据 RCC_APB1RSTR1 寄存器中的 PWRRST 位执行复位。

访问：与标准 APB 访问（3 个写周期和 2 个读周期）相比，访问该寄存器需要额外的 APB 周期。

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
PD15	PD14	PD13	PD12	PD11	PD10	PD9	PD8	PD7	PD6	PD5	PD4	PD3	PD2	PD1	PD0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:16 保留，必须保持复位值。

位 15:0 **PDy**：端口 G 下拉位 y (Port G pull-down bit y) (y=0..15)
在 PWR_CR3 寄存器中的 APC 位置 1 时将此位置 1，会激活 PG[y] 的下拉。

5.4.22 电源端口 H 上拉控制寄存器 (PWR_PUCRH)

Power Port H pull-up control register

偏移地址：0x58。

复位值：0x0000 0000。该寄存器在退出待机模式时不会复位，而是根据 RCC_APB1RSTR1 寄存器中的 PWRRST 位执行复位。

访问：与标准 APB 访问（3 个写周期和 2 个读周期）相比，访问该寄存器需要额外的 APB 周期。

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
PU15	PU14	PU13	PU12	PU11	PU10	PU9	PU8	PU7	PU6	PU5	PU4	PU3	PU2	PU1	PU0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:16 保留，必须保持复位值。

位 15:0 **PUy**：端口 H 上拉位 y (Port H pull-up bit y) (y=0..15)
在 PWR_CR3 寄存器中的 APC 位置 1 时将此位置 1，会激活 PH[y] 的上拉。
如果相应的 PDy 位也置 1，则不激活上拉。

5.4.23 电源端口 H 下拉控制寄存器 (PWR_PDCRH)

Power Port H pull-down control register

偏移地址：0x5C。

复位值：0x0000 0000。该寄存器在退出待机模式时不会复位，而是根据 RCC_APB1RSTR1 寄存器中的 PWRRST 位执行复位。

访问：与标准 APB 访问（3 个写周期和 2 个读周期）相比，访问该寄存器需要额外的 APB 周期。

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
PD15	PD14	PD13	PD12	PD11	PD10	PD9	PD8	PD7	PD6	PD5	PD4	PD3	PD2	PD1	PD0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:16 保留，必须保持复位值。

位 15:0 **PDy**：端口 H 下拉位 y (Port H pull-down bit y) (y=0..15)
在 PWR_CR3 寄存器中的 APC 位置 1 时将此位置 1，会激活 PH[y] 的下拉。

5.4.24 电源端口 I 上拉控制寄存器 (PWR_PUCRI)

Power Port I pull-up control register

偏移地址：0x60。

复位值：0x0000 0000。该寄存器在退出待机模式时不会复位，而是根据 RCC_APB1RSTR1 寄存器中的 PWRRST 位执行复位。

访问：与标准 APB 访问（3 个写周期和 2 个读周期）相比，访问该寄存器需要额外的 APB 周期。

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.	Res.	Res.	PU11	PU10	PU9	PU8	PU7	PU6	PU5	PU4	PU3	PU2	PU1	PU0
				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 11:0 **PUy**：端口 I 上拉位 y (Port I pull-up bit y) (y=0..11)
在 PWR_CR3 寄存器中的 APC 位置 1 时将此位置 1，会激活 PI[y] 的上拉。如果相应的 PDy 位也置 1，则不激活上拉。

5.4.25 电源端口 I 下拉控制寄存器 (PWR_PDCRI)

Power Port I pull-down control register

偏移地址：0x64。

复位值：0x0000 0000。该寄存器在退出待机模式时不会复位，而是根据 RCC_APB1RSTR1 寄存器中的 PWRRST 位执行复位。

访问：与标准 APB 访问（3 个写周期和 2 个读周期）相比，访问该寄存器需要额外的 APB 周期。

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.	Res.	Res.	PD11	PD10	PD9	PD8	PD7	PD6	PD5	PD4	PD3	PD2	PD1	PD0
				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 11:0 **PDy**：端口 I 下拉位 y (Port I pull-down bit y) (y=0..11)
在 PWR_CR3 寄存器中的 APC 位置 1 时将此位置 1，会激活 PI[y] 的下拉。

5.4.26 PWR 控制寄存器 (PWR_CR5)

PWR control register

偏移地址：0x80。

复位值：0x0000 0100。从待机模式唤醒后，此寄存器不会复位。

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.	Res.	Res.	Res.	Res.	Res.	R1MODE	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
							rw								

位 31:9 保留，必须保持复位值。

位 8 **R1MODE**：主稳压器范围 1 模式 (Main regulator range 1 mode)
该位仅对主稳压器范围 1 有效，对范围 2 无影响。当系统频率大于 80 MHz 时，建议复位此位。请参见*表 19：范围 1 升压模式配置*。
0：主稳压器处于范围 1 升压模式。
1：主稳压器处于范围 1 正常模式。

位 7:0 保留，必须保持复位值。

5.4.27 PWR 寄存器映射和复位值表

表 31. 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
0x000	PWR_CR1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LPR	Res.	Res.	Res.	VOS [1:0]		DBP	Res.	Res.	Res.	RRSTP	Res.	LPMS [2:0]		
	Reset value																		0				0	1	0				0		0	0	0
0x004	PWR_CR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	USV	IOSV	Res.	PVME4	PVME3	PVME2	PVME1	PLS [2:0]			PVDE
	Reset value																						0	0		0	0	0	0	0	0	0	0
0x008	PWR_CR3	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	EIWUL	Res.	Res.	DSIPDEN	Res.	APC	Res.	RRS	Res.	Res.	Res.	EWUP5	EWUP4	EWUP3	EWUP2	EWUP1
	Reset value																	1			1		0		0				0	0	0	0	0
0x00C	PWR_CR4	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	VBRS	VBE	Res.	Res.	Res.	WP5	WP4	WP3	WP2	WP1
	Reset value																							0	0				0	0	0	0	0
0x010	PWR_SR1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WUFI	Res.	Res.	Res.	Res.	Res.	Res.	SBF	Res.	Res.	Res.	WUF5	WUF4	WUF3	WUF2	WUF1
	Reset value																	0							0				0	0	0	0	0
0x014	PWR_SR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PVMO4	PVMO3	PVMO2	PVMO1	PVDO	VOSF	REGLPF	REGLPS	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	Reset value																	0	0	0	0	0	0	0	0								
0x018	PWR_SCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CSBF	Res.	Res.	Res.	CWUF5	CWUF4	CWUF3	CWUF2	CWUF1
	Reset value																								0				0	0	0	0	0
0x020	PWR_PUCRA	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PU15	Res.	PU13	PU12	PU11	PU10	PU9	PU8	PU7	PU6	PU5	PU4	PU3	PU2	PU1	PU0
	Reset value																	0		0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x024	PWR_PDCRA	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PD14	Res.	PD12	PD11	PD10	PD9	PD8	PD7	PD6	PD5	PD4	PD3	PD2	PD1	PD0
	Reset value																		0		0	0	0	0	0	0	0	0	0	0	0	0	0
0x028	PWR_PUCRB	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PU15	PU14	PU13	PU12	PU11	PU10	PU9	PU8	PU7	PU6	PU5	PU4	PU3	PU2	PU1	PU0
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x02C	PWR_PDCRB	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PD15	PD14	PD13	PD12	PD11	PD10	PD9	PD8	PD7	PD6	PD5	Res.	PD3	PD2	PD1	PD0
	Reset value																	0	0	0	0	0	0	0	0	0	0	0		0	0	0	0
0x030	PWR_PUCRC	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PU15	PU14	PU13	PU12	PU11	PU10	PU9	PU8	PU7	PU6	PU5	PU4	PU3	PU2	PU1	PU0
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x034	PWR_PDCRC	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PD15	PD14	PD13	PD12	PD11	PD10	PD9	PD8	PD7	PD6	PD5	PD4	PD3	PD2	PD1	PD0
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x038	PWR_PUCRD	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PU15	PU14	PU13	PU12	PU11	PU10	PU9	PU8	PU7	PU6	PU5	PU4	PU3	PU2	PU1	PU0
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x03C	PWR_PDCRD	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PD15	PD14	PD13	PD12	PD11	PD10	PD9	PD8	PD7	PD6	PD5	PD4	PD3	PD2	PD1	PD0
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

表 31. PWR 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
0x040	PWR_PUCRE	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PU15	PU14	PU13	PU12	PU11	PU10	PU9	PU8	PU7	PU6	PU5	PU4	PU3	PU2	PU1	PU0
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x044	PWR_PDCRE	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PD15	PD14	PD13	PD12	PD11	PD10	PD9	PD8	PD7	PD6	PD5	PD4	PD3	PD2	PD1	PD0
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x048	PWR_PUCRF	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PU15	PU14	PU13	PU12	PU11	PU10	PU9	PU8	PU7	PU6	PU5	PU4	PU3	PU2	PU1	PU0
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x04C	PWR_PDCRF	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PD15	PD14	PD13	PD12	PD11	PD10	PD9	PD8	PD7	PD6	PD5	PD4	PD3	PD2	PD1	PD0
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x050	PWR_PUCRG	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PU15	PU14	PU13	PU12	PU11	PU10	PU9	PU8	PU7	PU6	PU5	PU4	PU3	PU2	PU1	PU0
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x054	PWR_PDCRG	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PD15	PD14	PD13	PD12	PD11	PD10	PD9	PD8	PD7	PD6	PD5	PD4	PD3	PD2	PD1	PD0
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x058	PWR_PUCRH	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PU15	PU14	PU13	PU12	PU11	PU10	PU9	PU8	PU7	PU6	PU5	PU4	PU3	PD2	PU1	PU0
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x05C	PWR_PDCRH	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PD15	PD14	PD13	PD12	PD11	PD10	PD9	PD8	PD7	PD6	PD5	PD4	PD3	PD2	PD1	PD0
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x060	PWR_PUCRI	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PU11	PU10	PU9	PU8	PU7	PU6	PU5	PU4	PU3	PU2	PU1	PU0
	Reset value																					0	0	0	0	0	0	0	0	0	0	0	0
0x064	PWR_PDCRI	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PD11	PD10	PD9	PD8	PD7	PD6	PD5	PD4	PD3	PD2	PD1	PD0
	Reset value																					0	0	0	0	0	0	0	0	0	0	0	0
0x080	PWR_CR5	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	R1MODE	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	Reset value																								1								

有关寄存器边界地址的信息，请参见*第 2.2.2 节：存储器映射和寄存器边界地址*。

6 复位和时钟控制 (RCC)

6.1 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。

6.1.1 电源复位

只要发生以下事件之一，就会产生电源复位：

1. 欠压复位 (BOR)。
2. 退出待机模式时。
3. 退出关断模式时。

欠压复位，包括上电或掉电复位 (POR/PDR)，将所有寄存器设置为其复位值，备份域除外。

退出待机模式时，V_{CORE} 域的所有寄存器都设置为其复位值。V_{CORE} 域外的寄存器（RTC，WKUP，IWDG，以及待机/关断模式控制）不受影响。

退出关断模式时，会产生欠压复位，将除备份域中的寄存器以外的所有寄存器全部复位。

6.1.2 系统复位

除了时钟控制/状态寄存器 (RCC_CSR) 中的复位标志和备份域中的寄存器外，系统复位会将其他全部寄存器都复位为复位值。

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. 窗口看门狗事件（WWDG 复位）
3. 独立看门狗事件（IWDG 复位）
4. 防火墙事件（FIREWALL 复位）
5. 软件复位（SW 复位）（请参见*软件复位*）
6. 低功耗模式安全复位（请参见*低功耗模式安全复位*）
7. 选项字节加载器复位（请参见*选项字节加载器复位*）
8. 欠压复位

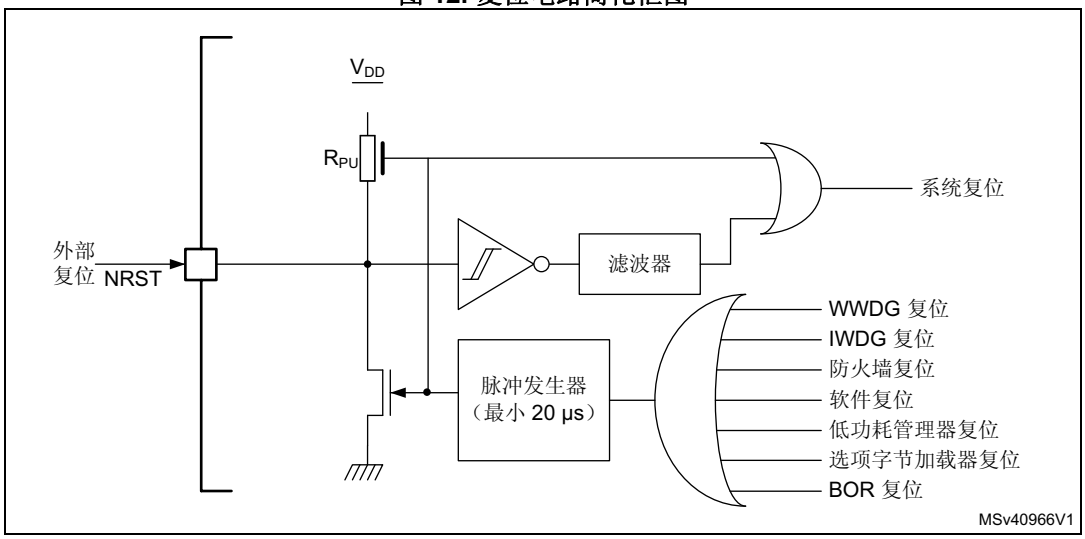
可通过查看控制/状态寄存器 (RCC_CSR) 中的复位标志确定复位源（请参见*第 6.4.30 节：控制/状态寄存器 (RCC_CSR)*）。

这些源均作用于 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向 NRST 引脚上输出一个低电平脉冲。脉冲发生器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。

内部复位情况下，内部上拉 R_{PU} 失效，通过上拉电阻节约功耗。

图 12. 复位电路简化框图



软件复位

Cortex®-M4 应用中断和复位控制寄存器中的 SYSRESETREQ 位必须设置为强制器件上的软件复位（请参见 *STM32F3、STM32F4、STM32L4 和 STM32L4 + 系列 Cortex®-M4* (PM0214)）。

低功耗模式安全复位

为了防止关键应用错误地进入低功耗模式，提供了两种低功耗模式安全复位。如果在选项字节中使能，则在下列情况下会产生这种复位：

1. 进入待机模式：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：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。
3. 进入关断模式：此复位的使能方式是清零用户选项字节中的 nRST_SHDW 位。使能后，只要成功执行进入关断模式序列，器件就将复位，而非进入关断模式。

有关用户选项字节的详细信息，请参见*第 3.4.1 节：选项字节说明*。

选项字节加载器复位

当 FLASH_CR 寄存器中的 OBL_LAUNCH 位（位 27）置 1 时，将产生选项字节加载器复位。此位用来通过软件启动选项字节加载。

6.1.3 备份域复位

备份域具有两个特殊复位。

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*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。

备份域复位仅影响 LSE 振荡器、RTC、备份寄存器和 RCC 备份域控制寄存器。

6.2 时钟

可以使用四种不同的时钟源来驱动系统时钟 (SYSCLK)：

- HSI16（高速内部）16 MHz RC 振荡器时钟
- MSI（多速内部）RC 振荡器时钟
- HSE 振荡器时钟，4 至 48 MHz
- PLL 时钟

从复位中启动后，MSI 用作系统时钟源，配置为 4 MHz。

器件具有以下附加时钟源：

- 32 kHz 低速内部 RC (LSI RC)，该 RC 用于驱动独立看门狗，也可选择提供给 RTC 用于停止/待机模式下的自动唤醒。
- 32.768 kHz 低速外部晶振（LSE 晶振），用于驱动实时时钟 (RTCCLK)。
- RC 48 MHz 内部时钟源 (HSI48)，可用于驱动 USB FS、SDMMC 和 RNG。

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。

可通过多个预分频器配置 AHB 频率、APB1 和 APB2 域。AHB、APB1 和 APB2 域的最大频率为 120 MHz。

所有外设时钟均由其总线时钟（HCLK、PCLK1 或 PCLK2）提供，但以下时钟除外：

- 48 MHz 时钟，用于 USB OTG FS、SDMMC 和 RNG。该时钟由以下四个时钟源之一提供（由软件选择）：
  - 主 PLL VCO (PLL48M1CLK)
  - PLLSAI1 VCO (PLL48M2CLK)
  - MSI 时钟
  - HSI48 内部振荡器

  当 MSI 时钟利用 LSE 自动调整时，它可用于 USB OTG FS 器件。

  如果 HSI48 48 MHz 时钟可用，则可将其耦合到时钟恢复系统，从而为 USB OTG FS（无晶振型解决方案）提供足够的时钟连接。
- 由以下任一时钟源提供的 ADC 时钟（由软件选择）：
  - 系统时钟 (SYSCLK)
  - PLLSAI1 VCO (PLLADC1CLK)
- 由以下四个时钟源之一提供的 U(S)ART 时钟（由软件选择）：
  - 系统时钟 (SYSCLK)
  - HSI16 时钟
  - LSE 时钟
  - APB1 或 APB2 时钟（是 PCLK1 还是 PCLK2 取决于 U(S)ART 映射哪个 APB）

  仅当时钟为 HSI16 或 LSE 时，支持从停止模式唤醒。
- 由以下三个时钟源之一提供的 I^2C 时钟（由软件选择）：
  - 系统时钟 (SYSCLK)
  - HSI16 时钟
  - APB1 时钟 (PCLK1)

  仅当时钟为 HSI16 时，支持从停止模式唤醒。

- 由以下任一时钟源提供的 SAI1 和 SAI2 时钟（由软件选择）：
  - 外部时钟，对于 SAI1，它映射到 SAI1_EXTCLK，而对于 SAI2，它映射到 SAI2_EXTCLK
  - PLLSAI1 VCO (PLLSAI1CLK)
  - PLLSAI2 VCO (PLLSAI2CLK)
  - 主 PLL VCO (PLLSAI3CLK)
  - HSI16 时钟
- 由以下任一时钟源提供的 DFSDM 音频时钟（由软件选择）：
  - SAI1 时钟
  - HSI 时钟
  - MSI 时钟
- LTDC 时钟。LTDC 时钟由特定的 PLL (PLLSAI2) 生成。
- DSI 时钟。要生成 DSI 时钟，必须使用高速外部晶振 (HSE)。
  - DSI lanebyteclk 时钟：在 DSI-PHY 关闭的情况下，从 DSI-PHY 或 PLLSAI2 的特定输出（频率低于 62.5 MHz）输出的 DSI 通道字节时钟（8 分频的高速时钟）。
  - DSI 主机 rxclkesc 时钟。DSI RX 逸出模式时钟从 DSI-PHY 输出（即使未向 DSI-PHY 提供时钟，也会从 DP0/DN0 生成）。
- 由以下任一时钟源提供的 OctoSPI 内核时钟（由软件选择）：
  - 系统时钟
  - PLL48M1CLK
  - MSI 时钟
- 由以下五个时钟源之一提供的低功耗定时器 (LPTIMx) 时钟（由软件选择）：
  - LSI 时钟
  - LSE 时钟
  - HSI16 时钟
  - APB1 时钟 (PCLK1)
  - 映射到 LPTIMx_IN1 上的外部时钟

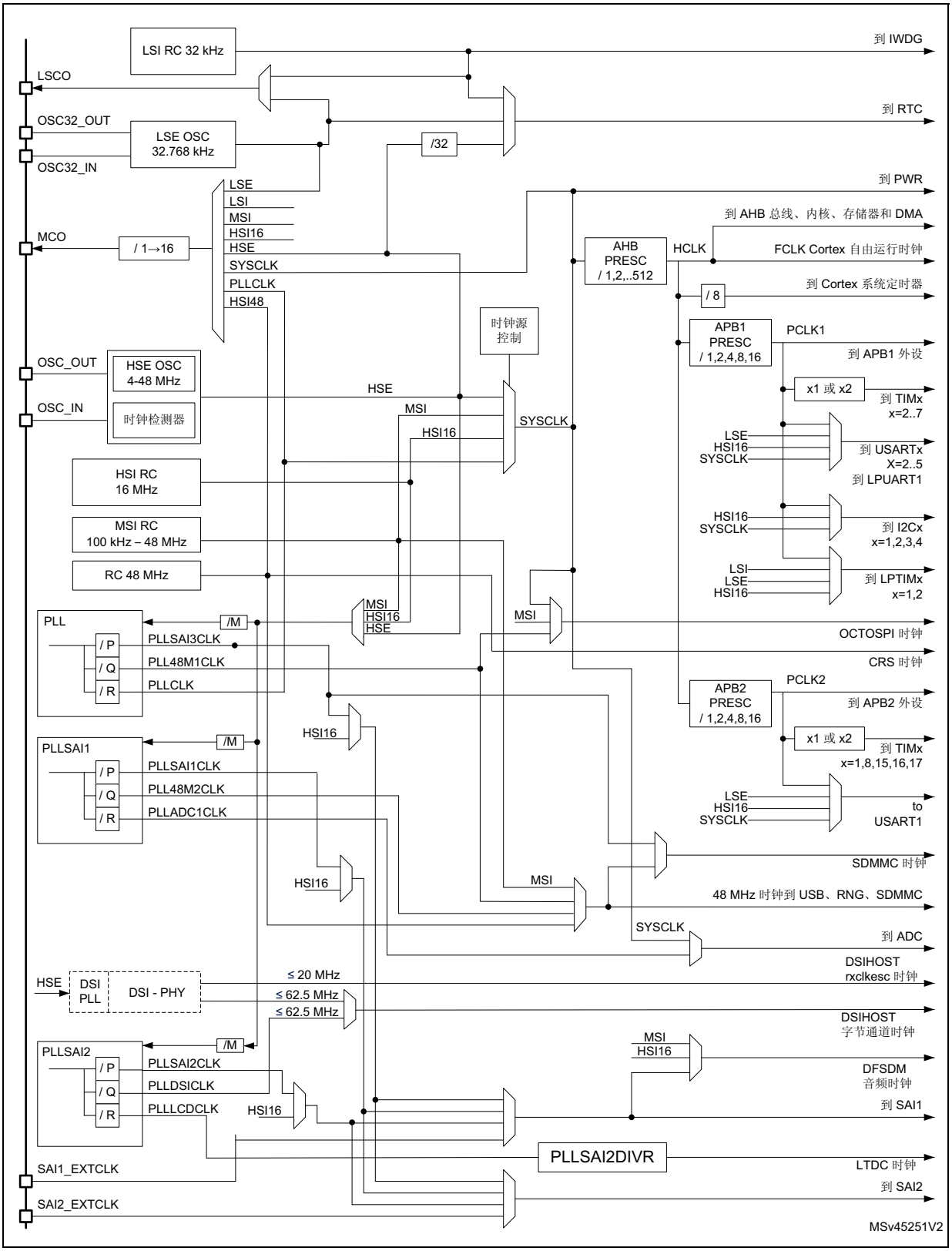
  仅当时钟为 LSI 或 LSE，或处于外部时钟模式时，支持停止模式的功能（包括唤醒）。
- 由以下三个时钟源之一提供的 RTC 时钟（由软件选择）：
  - LSE 时钟
  - LSI 时钟
  - HSE 时钟 32 分频

  仅当时钟为 LSI 或 LSE 时，支持停止模式的功能（包括唤醒）。
- 始终为 LSI 时钟的 IWDG 时钟。

RCC 向 Cortex® 系统定时器 (SysTick) 外部时钟馈送 8 分频的 AHB 时钟 (HCLK)。SysTick 可使用此时钟作为时钟源，也可直接使用 Cortex® 时钟 (HCLK) 作为时钟源，具体可在 SysTick 控制和状态寄存器中配置。

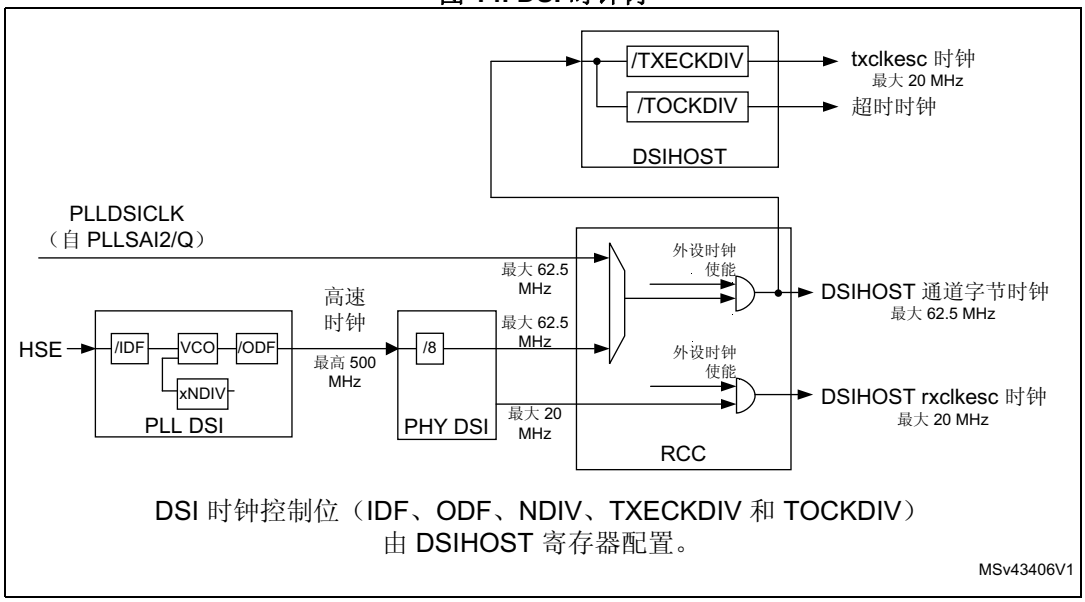
FCLK 充当 Cortex®-M4 的自由运行时钟。有关详细信息，请参见 *STM32F3*、*STM32F4*、*STM32L4 和 STM32L4+ 系列 Cortex®-M4* 编程手册 (PM0214)。

图 13. 时钟树


LSI RC 32 kHz
到 IWDG
LSCO
OSC32_OUT
LSE OSC
32.768 kHz
OSC32_IN
/32
到 RTC
LSE
LSI
MSI
HSI16
HSE
SYSCLK
PLLCLK
HSI48
到 PWR
到 AHB 总线、内核、存储器和 DMA
MCO
/ 1→16
AHB
PRESC
/ 1,2,..512
HCLK
FCLK Cortex 自由运行时钟
/ 8
到 Cortex 系统定时器
时钟源
控制
APB1
PRESC
/ 1,2,4,8,16
PCLK1
到 APB1 外设
OSC_OUT
HSE OSC
4-48 MHz
OSC_IN
时钟检测器
HSE
MSI
HSI16
SYSCLK
x1 或 x2
到 TIMx
x=2..7
LSE
HSI16
SYSCLK
到 USARTx
X=2..5
到 LPUART1
HSI RC
16 MHz
MSI RC
100 kHz – 48 MHz
RC 48 MHz
HSI16
SYSCLK
到 I2Cx
x=1,2,3,4
LSI
LSE
HSI16
到 LPTIMx
x=1,2
PLL
/M
/ P
/ Q
/ R
PLLSAI3CLK
PLL48M1CLK
PLLCLK
MSI
HSI16
HSE
MSI
OCTOSPI 时钟
CRS 时钟
APB2
PRESC
/ 1,2,4,8,16
PCLK2
到 APB2 外设
HSI16
PLLSAI1
/M
/ P
/ Q
/ R
PLLSAI1CLK
PLL48M2CLK
PLLADC1CLK
x1 或 x2
到 TIMx
x=1,8,15,16,17
LSE
HSI16
SYSCLK
to
USART1
SDMMC 时钟
HSI16
MSI
48 MHz 时钟到 USB、RNG、SDMMC
SYSCLK
到 ADC
DSIHOST
rxclkesc 时钟
HSE
DSI
PLL
DSI - PHY
≤ 20 MHz
≤ 62.5 MHz
≤ 62.5 MHz
DSIHOST
字节通道时钟
PLLSAI2
/M
/ P
/ Q
/ R
PLLSAI2CLK
PLLDSICLK
PLLLCDCLK
HSI16
MSI
HSI16
DFSDM
音频时钟
到 SAI1
PLLSAI2DIVR
LTDC 时钟
SAI1_EXTCLK
到 SAI2
SAI2_EXTCLK
MSv45251V2

1. 有关内部和外部时钟源特性的所有详细信息，请参见器件数据手册的“电气特性”部分。
2. ADC 时钟可由 ADC 总线接口的 AHB 时钟除以一个可编程的因数（1、2 或 4）来提供。当可编程因数为“1”时，AHB 预分频器必须等于“1”。

图 14. DSI 时钟树



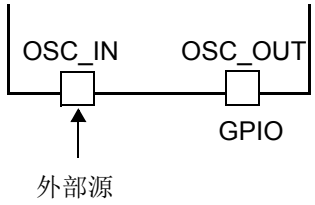
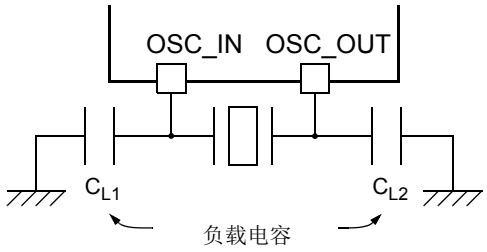
6.2.1 HSE 时钟

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 用户外部时钟

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 15. HSE/LSE 时钟源

时钟源	硬件配置
外部时钟	
晶振/陶瓷谐振器	

外部晶振/陶瓷谐振器（HSE 晶振）

4 到 48 MHz 外部振荡器的优点是精度非常高。

相关的硬件配置如*图 15* 所示。有关详细信息，请参见*数据手册*的电气特性部分。

时钟控制寄存器 (RCC_CR) 中的 HSERDY 标志指示 HSE 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*时钟中断使能寄存器 (RCC_CIER)* 中使能中断，则可产生中断。

HSE 晶振可通过*时钟控制寄存器 (RCC_CR)* 中的 HSEON 位打开或关闭。

外部源（HSE 旁路）

在此模式下，必须提供外部时钟源。最高频率不超过 48 MHz。通过将*时钟控制寄存器 (RCC_CR)* 的 HSEBYP 和 HSEON 位置 1 可选择此模式。必须使用占空比为 ~40-60 % 的外部时钟信号（方波、正弦波或三角波）来驱动 OSC_IN 引脚，具体取决于频率（参考数据手册），同时 OSC_OUT 引脚可用作 GPIO。请参见*图 15*。

6.2.2 HSI16 时钟

HSI16 时钟信号是从 16 MHz 内部 RC 振荡器生成的。

HSI16 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 晶振快，但即使校准后，其精度也不及外部晶振或陶瓷谐振器。

从停止模式（停止 0、停止 1 或停止 2）唤醒后，可选择 HSI16 时钟作为系统时钟。请参见*第 6.3 节：低功耗模式*。MSI RC 还可作为备份时钟源（辅助时钟）使用，以防 HSE 晶振发生故障。请参见*第 6.2.10 节：时钟安全系统 (CSS)*。

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 ST 会对每个器件进行出厂校准，达到 T_A= 25°C 时 1 % 的精度。

复位后，工厂校准值将加载到*内部时钟源校准寄存器 (RCC_ICSCR)* 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*内部时钟源校准寄存器 (RCC_ICSCR)* 中的 HSITRIM[6:0] 对 HSI16 频率进行微调。

有关如何测量 HSI16 频率变化的更多详细信息，请参见*第 6.2.17 节：基于 TIM15/TIM16/TIM17 的内部/外部时钟测量*。

时钟控制寄存器 (RCC_CR) 中的 HSIRDY 标志指示 HSI16 RC 是否稳定。在启动时，硬件将该位置 1 后，HSI16 才可以使用。

HSI16 RC 可通过*时钟控制寄存器 (RCC_CR)* 中的 HSION 位打开或关闭。

HSI16 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。请参见*第 212 页的第 6.2.10 节：时钟安全系统 (CSS)*。

6.2.3 MSI 时钟

MSI 时钟信号是从内部 RC 振荡器生成的。频率范围可通过软件使用*时钟控制寄存器 (RCC_CR)* 中的 MSIRANGE[3:0] 位进行选择。有 12 个频率范围可用：100 kHz、200 kHz、400 kHz、800 kHz、1 MHz、2 MHz、4 MHz（默认值）、8 MHz、16 MHz、24 MHz、32 Mhz 和 48 MHz。

在从复位重启、从待机、关断低功耗模式后唤醒，MSI 时钟被用作系统时钟。从复位重启后，MSI 频率被置位其默认值 4 MHz。请参见*第 6.3 节：低功耗模式*。

从停止模式（停止 0、停止 1 或停止 2）唤醒后，可选择 MSI 时钟作为系统时钟。请参见*第 6.3 节：低功耗模式*。MSI RC 还可作为备份时钟源（辅助时钟）使用，以防 HSE 晶振发生故障。请参见*第 6.2.10 节：时钟安全系统 (CSS)*。

MSI RC 振荡器的优势在于可提供一个低成本（无外部元件）低功耗的时钟源。此外，当和 LSE 一起用于 PLL 模式时，它可提供一个非常精确的时钟源，该时钟源可用于 USB OTG FS 器件，并且向主 PLL 反馈，使系统以最大速率运行。

时钟控制寄存器 (RCC_CR) 中的 MSIRDY 标志指示 MSI RC 是否稳定。在启动时，硬件将此位置 1 后，MSI RC 输出时钟才可以使用。MSI RC 可通过*时钟控制寄存器 (RCC_CR)* 中的 MSION 位打开或关闭。

利用 LSE 进行硬件自动校准（PLL 模式）

当应用中存在 32.768 kHz 的外部振荡器时，可通过设置*时钟控制寄存器 (RCC_CR)* 中的 MSIPLLEN 位将 MSI 配置为 PLL 模式。当配置为 PLL 模式，MSI 利用 LSE 自动校准。该模式可用于所有 MSI 频率范围。48 MHz 时，处于 PLL 模式的 MSI 可用于 USB OTG FS 器件，不需要外部高速晶振。

软件校准

由于生产工艺不同，不同芯片的 MSI RC 振荡器频率也不同，因此 ST 会对每个器件进行出厂校准，使器件在 25 °C 室温 (TA) 时达到 1 % 的精度。复位后，出厂校准值载入*内部时钟源校准寄存器 (RCC_ICSCR)* 的 MSICAL[7:0] 位。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_ICSCR 寄存器中的 MSITRIM[7:0] 位对 MSI 频率进行微调。有关如何测量 MSI 频率变化的更多详细信息，请参见*第 6.2.17 节：基于 TIM15/TIM16/TIM17 的内部/外部时钟测量*。

6.2.4 HSI48 时钟

HSI48 时钟信号由内部 48 MHz RC 振荡器生成，可直接用于 USB、随机数发生器 (RNG) 以及 SDMMC。

内部 48 MHz RC 振荡器主要用于通过特殊的时钟恢复系统 (CRS) 电路向 USB 外设提供高精度时钟。CRS 可使用 USB SOF 信号、LSE 或外部信号自动而快速地实时调整振荡器频率。系统进入停止或待机模式后，CRS 会立即被禁止。未使用 CRS 时，HSI48 RC 振荡器以其默认频率运行，具体默认频率值取决于生产工艺。

有关如何配置和使用 CRS 外设的更多详细信息，请参见*第 7 节：时钟恢复系统 (CRS)*。

时钟恢复 RC 寄存器 (RCC_CRRCR) 中的 HSI48RDY 标志指示 HSI48 RC 振荡器是否稳定。在启动时，硬件将此位置 1 后，HSI48 RC 振荡器才可以使用。

HSI48 可通过时钟恢复 RC 寄存器 (RCC_CRRCR) 中的 HSI48ON 位打开或关闭。

6.2.5 PLL

器件嵌入了 3 个 PLL：PLL，PLLSAI1，PLLSAI2。每个 PLL 提供了多达 3 个独立输出。内部 PLL 可用来倍频 HSI16、HSE 或 MSI 输出时钟频率。PLL 输入频率必须介于 4 至 16 MHz 之间。所选时钟源被可编程因数 PLLM（因数为 1 至 8）分频，提供一个在所需输入范围内的时钟频率。请参见*图 13：时钟树*和*PLL 配置寄存器 (RCC_PLLCFGR)*。

PLL 配置（对输入时钟和倍频因数进行选择）必须在使能 PLL 之前完成。当 PLL 使能后，这些参数就不能再更改。

要修改 PLL 配置，请按照以下步骤操作：

1. 通过将*时钟控制寄存器 (RCC_CR)* 中的 PLLON 位设置为 0 禁止 PLL。
2. 等待，直至 PLLRDY 清零。PLL 现已完全停止。
3. 根据需要更改参数。
4. 通过将 PLLON 位置 1 再次使能 PLL。
5. 通过配置 *PLL 配置寄存器 (RCC_PLLCFGR)* 中的 PLLPEN、PLLQEN、PLLREN，使能所需要的 PLL 输出。

如果已在*时钟中断使能寄存器 (RCC_CIER)* 中使能中断，则 PLL 就绪时便可产生中断。

更改 PLLSAI1 或 PLLSAI2 配置可应用同样的步骤：

1. 通过将*时钟控制寄存器 (RCC_CR)* 中的 PLLSAI1ON/PLLSAI2ON 位设置为 0 禁止 PLLSAI1/PLLSAI2。
2. 等待，直至 PLLSAI1RDY/PLLSAI2RDY 清零。此时 PLLSAI1/PLLSAI2 彻底停止。
3. 根据需要更改参数。
4. 通过将 PLLSAI1ON/PLLSAI2ON 置为 1，再次使能 PLLSAI1/PLLSAI2。
5. 通过配置 *PLLSAI1 配置寄存器 (RCC_PLLSAI1CFGR)* 和 *PLLSAI2 配置寄存器 (RCC_PLLSAI2CFGR)* 中的 PLLSAI1PEN/PLLSAI2PEN、PLLSAI1QEN/PLLSAI2QEN、PLLSAI1REN/PLLSAI2REN，使能所需要的 PLL 输出。

PLL 输出频率不可超过 120 MHz。

每个 PLL 输出时钟的使能位（PLLPEN、PLLQEN、PLLREN、PLLSAI1PEN、PLLSAI1QEN、PLLSAI1REN、PLLSAI2PEN 和 PLLSAI2REN）可随时修改，而不必停止相应的 PLL。如果 PLLCLK 用作系统时钟，则 PLLREN 不可被清零。

6.2.6 LSE 时钟

LSE 晶振是 32.768 kHz 低速外部晶振或陶瓷谐振器，可作为实时时钟 (RTC) 的时钟源来提供时钟/日历或其他定时功能，具有功耗低且精度高的优点。

LSE 晶振通过 *备份域控制寄存器 (RCC_BDCR)* 中的 LSEON 位打开和关闭。使用 *备份域控制寄存器 (RCC_BDCR)* 中的 LSEDRV[1:0] 位，可在运行时更改晶振驱动强度，以实现稳健性、短启动时间和低功耗之间的最佳平衡。当 LSE 为 ON 时，LSE 驱动可降低为更低的驱动能力 (LSEDRV=00)。但是，当选择了 LSEDRV，如果 LSEON=1 则不能提高驱动能力。

备份域控制寄存器 (RCC_BDCR) 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*时钟中断使能寄存器 (RCC_CIER)* 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*睡眠和停止模式下的 AHB1 外设时钟使能寄存器 (RCC_AHB1SMENR)* 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 (GPIO)。请参见*图 15*。

6.2.7 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) RTC 使用。时钟频率为 32 kHz。有关详细信息，请参见数据手册的电气特性部分。

LSI RC 可通过 *控制/状态寄存器 (RCC_CSR)* 中的 LSION 位打开或关闭。

控制/状态寄存器 (RCC_CSR) 中的 LSIRDY 标志指示 LSI 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*时钟中断使能寄存器 (RCC_CIER)* 中使能中断，则可产生中断。

6.2.8 系统时钟 (SYSCLK) 选择

可以使用四种不同的时钟源来驱动系统时钟 (SYSCLK)：

- MSI 振荡器
- HSI16 振荡器
- HSE 振荡器
- PLL

系统时钟最大频率为 120 MHz。系统复位后，选择 MSI 振荡器（为 4 MHz）作为系统时钟。在直接使用 HSI 或者通过 PLL 使用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（时钟在启动延迟或 PLL 锁相后稳定时），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。*内部时钟源校准寄存器 (RCC_ICSCR)* 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。

若要在低速系统时钟与高速系统时钟之间进行切换，建议使用一个中速时钟的过渡状态，并至少持续 1us。

时钟源切换条件：

- 从 HSE、HSI 或 MSI 切换到 PLL（AHB 频率 (HCLK) 高于 80 MHz）
- 从 PLL（HCLK 高于 80 MHz）切换到 HSE、HSI 或 MSI

过渡状态：

- 设置 AHB 预分频器 HPRE[3:0] 位，将系统时钟 2 分频
- 将系统时钟切换为 PLL
- 等待至少 1us，然后将 AHB 预分频器位重新配置为所需 HCLK 频率

6.2.9 时钟源频率与电压调节

下表给出了不同产品电压范围下不同的时钟源频率。

表 32. 时钟源频率

产品电压范围	时钟频率			
	MSI	HSI16	HSE	PLL/PLLSAI1/PLLSAI2
范围 1 升压模式	48 MHz	16 MHz	48 MHz	120 MHz
范围 1 正常模式	48 MHz	16 MHz	48 MHz	80 MHz
范围 2	24 MHz 范围	16 MHz	26 MHz	26 MHz

6.2.10 时钟安全系统 (CSS)

时钟安全系统可通过软件激活。激活后，时钟监测器将在 HSE 振荡器启动延迟后使能，并在此振荡器停止时被关闭。

如果 HSE 时钟发生故障，HSE 振荡器将自动禁止，一个时钟故障事件将发送到高级控制定时器（TIM1/TIM8 和 TIM15/16/17）的断路输入，并且同时还将生成一个中断来向软件通知此故障（时钟安全系统中断，CSSI），以使 MCU 能够执行救援操作。CSSI 与 Cortex®-M4 NMI（不可屏蔽中断）异常向量相链接。

注： *当 CSS 使能后，如果 HSE 时钟发生故障，则会产生 CSS 中断，并会自动产生 NMI。NMI 将无限期执行，除非将 CSS 中断挂起位清零。因此，在 NMI ISR 中，用户必须将时钟中断清零寄存器 (RCC_CICR) 中的 CSSC 位置 1，以清除 CSS 中断。*

如果直接或间接使用 HSE 振荡器作为系统时钟（间接是指它用作 PLL 输入时钟，PLL 时钟用作系统时钟），检测到故障时会导致系统时钟切换到 MSI 或 HSI16 振荡器（取决于*时钟配置寄存器 (RCC_CFGR)* 中的 STOPWUCK 配置），并禁止 HSE 振荡器。当故障发生时，若 HSE 时钟（分频或不分频）为正在用作系统时钟的 PLL 时钟输入，则也会禁用该 PLL。

6.2.11 LSE 上的时钟安全系统

可由软件通过写入*备份域控制寄存器 (RCC_BDCR)* 中的 LSECSSON 位，来激活 LSE 上的时钟安全系统。仅可通过硬件复位、RTC 软件复位、或在检测到 LSE 故障后禁止该位。LSE 和 LSI 使能（LSEON 和 LSION 使能）并就绪（LSERDY 和 LSIRDY 由硬件置 1）后，以及通过 RTCSEL 选择 RTC 时钟后，必须写入 LSECSSON。

LSE 上的 CSS 适用于除 VBAT 外的所有模式，以及除上电复位外的所有系统复位。如果在外部 32 kHz 振荡器上检测到故障，则不会再向 RTC 提供 LSE 时钟，但不会对寄存器进行硬件操作。如果 MSI 处于 PLL 模式，则禁止该模式。

在待机模式下，会产生唤醒。在其他模式下，可发送中断来唤醒软件（请参见*时钟中断使能寄存器 (RCC_CIER)*、*时钟中断标志寄存器 (RCC_CIFR)* 和*时钟中断清零寄存器 (RCC_CICR)*）。

软件随后必须禁止 LSECSSON 位并停止出现故障的 32 kHz 振荡器（禁止 LSEON），并更改 RTC 时钟源（无时钟、LSI 或 HSE，通过 RTCSEL），或者采取任何必要措施来确保应用的安全。

LSE 振荡器的频率必须高于 30 kHz，以免发生 CSS 误检。

6.2.12 ADC 时钟

ADC 时钟由系统时钟或 PLLSAI1 输出提供。此时钟可达 120 MHz，并可通过配置 ADC1_CCR 寄存器使用以下预分频器值进行分频：1、2、4、6、8、10、12、16、32、64、128 或 256。它与 AHB 时钟异步。或者，ADC 时钟可由 ADC 总线接口的 AHB 时钟除以一个可编程的因数（1、2 或 4）来提供。该可编程的因数使用 ADC123_CCR 中的 CKMODE 位域进行配置。

如果可编程因数为“1”，必须将 AHB 预分频器设置为“1”。

6.2.13 RTC 时钟

RTCCLK 时钟源可以是 HSE/32、LSE 或 LSI 时钟。选择方式是编程*备份域控制寄存器 (RCC_BDCR)* 中的 RTCSEL[1:0] 位。所做的选择只能通过复位备份域的方式修改。配置系统时，必须始终使 PCLK 频率大于或等于 RTCCLK 频率，以便 RTC 正常工作。

LSE 时钟位于备份域中，而 HSE 和 LSI 时钟则不是。因此：

- 如果选择 LSE 作为 RTC 时钟：
  - 只要 V_{BAT} 电源保持工作，即使 V_{DD} 电源关闭，RTC 仍可继续工作。
- 如果选择 LSI 作为 RTC 时钟：
  - 在 V_{DD} 电源掉电时，RTC 的状态将不能保证。
- 如果使用按预分频器进行分频的 HSE 时钟作为 RTC 时钟：
  - 如果 V_{DD} 电源掉电或者内部调压器关闭（切断 V_{CORE} 域的供电），则 RTC 的状态将不能保证。

如果 RTC 时钟为 LSE 或 LSI，则 RTC 在系统复位后仍可获得时钟并保持正常工作。

6.2.14 定时器时钟

定时器时钟频率由硬件自动定义为固定值。分为两种情况：

1. 如果 APB 预分频器等于 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)。

6.2.15 看门狗时钟

如果独立看门狗 (IWDG) 已通过硬件选项或软件访问的方式启动，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。

6.2.16 时钟输出功能

- MCO

  微控制器时钟输出 (MCO) 功能允许将时钟输出到外部 MCO 引脚上。可选择以下八种时钟信号之一作为 MCO 时钟。

  – LSI
  – LSE
  – SYSCLK
  – HSI16
  – HSI48
  – HSE
  – PLLCLK
  – MSI

  此选择由*时钟配置寄存器 (RCC_CFGR)* 中的 MCOSEL[3:0] 位控制。所选时钟可以通过*时钟配置寄存器 (RCC_CFGR)* 的 MCOPRE[2:0] 位域进行分频。

- LSCO

  另一个输出 (LSCO) 允许将低速时钟输出到外部 LSCO 引脚上：

  – LSI
  – LSE

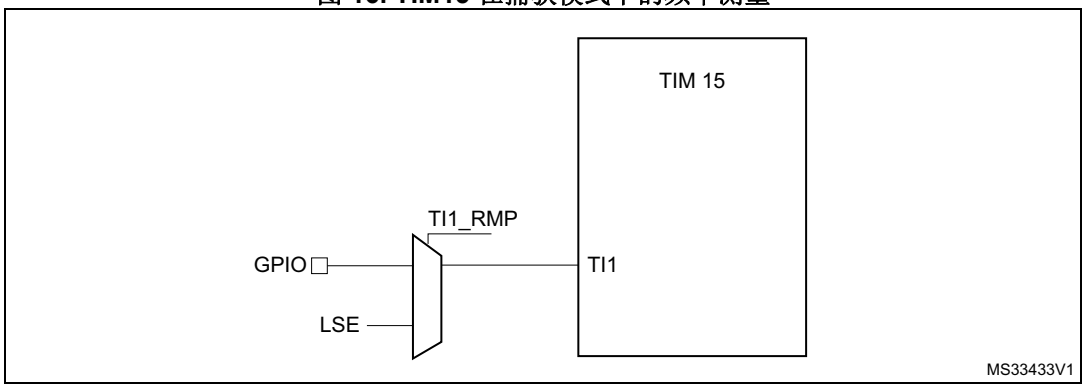
  在停止模式（停止 0、停止 1 和停止 2）和待机模式下，此输出依旧可用。此选择由*备份域控制寄存器 (RCC_BDCR)* 中的 LSCOSEL 控制，并通过 LSCOEN 使能。

MCO 时钟输出需要在 MCO 引脚上选择的相应的复用功能，LSCO 引脚应保持默认的 POR 状态。

6.2.17 基于 TIM15/TIM16/TIM17 的内部/外部时钟测量

所有板上时钟源的频率都可通过对 TIM15、TIM16 或 TIM17 通道 1 的输入捕获进行间接测量，如*图 16*、*图 17* 和*图 18* 所示。

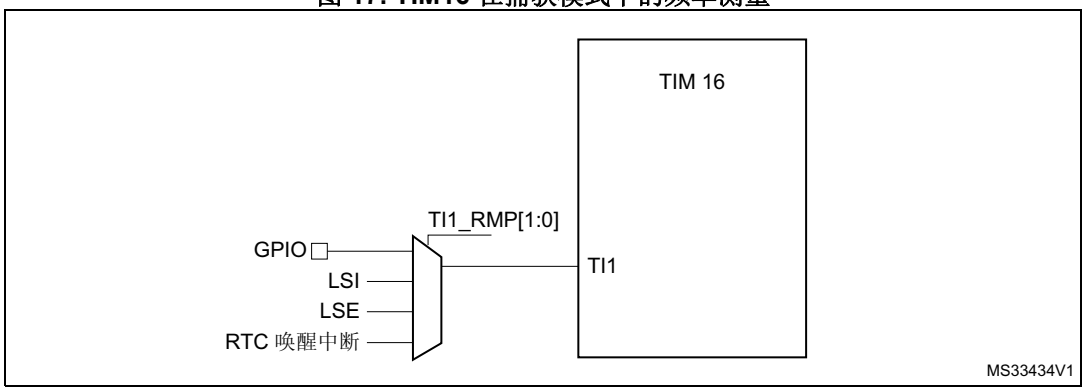
图 16. TIM15 在捕获模式下的频率测量



定时器 15 的输入捕获通道可以是 MCU 的 GPIO 线或内部时钟。此选择通过 TIM15_OR 寄存器的 TI1_RMP 位执行。可能的情况如下：

- TIM15 通道 1 连接到 GPIO。请参见器件数据手册中的复用功能映射。
- TIM15 通道 1 连接到 LSE。

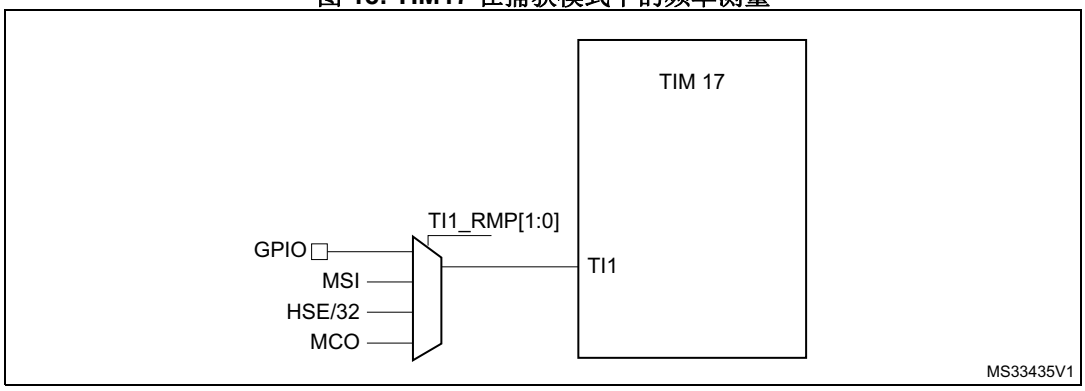
图 17. TIM16 在捕获模式下的频率测量



定时器 16 的输入捕获通道可以是 MCU 的 GPIO 线或内部时钟。此选择通过 TIM16_OR 寄存器的 TI1_RMP[1:0] 位执行。可能的情况如下：

- TIM16 通道 1 连接到 GPIO。请参见器件数据手册中的复用功能映射。
- TIM16 通道 1 连接到 LSI 时钟。
- TIM16 通道 1 连接到 LSE 时钟。
- TIM16 通道 1 连接到 RTC 唤醒中断信号。在这种情况下，应使能 RTC 中断。

图 18. TIM17 在捕获模式下的频率测量



定时器 17 的输入捕获通道可以是 MCU 的 GPIO 线或内部时钟。此选择通过 TIM17_OR 寄存器的 TI1_RMP[1:0] 位执行。可能的情况如下：

- TIM17 通道 1 连接到 GPIO。请参见器件数据手册中的复用功能映射。
- TIM17 通道 1 连接到 MSI 时钟。
- TIM17 通道 1 连接到 HSE/32 时钟。
- TIM17 通道 1 连接到微控制器时钟输出 (MCO)，此选择由时钟配置寄存器 (RCC_CFGR) 的 MCOSEL[3:0] 位控制。

HSI16 和 MSI 校准

对于 TIM15 和 TIM16，将 LSE 连接到通道 1 输入捕获的主要目的是为了能够精确测量 HSI16 和 MSI 系统时钟（为此，应将 HSI16 或 MSI 用作系统时钟源）。借助 LSE 信号连续边沿之间的 HSI16（或 MSI）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与生产、工艺、温度和/或电压相关的频率偏差。

MSI 和 HSI16 振荡器均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，HSI16/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。

如果 LSE 不可用，为了尽可能达到最精确的校准，HSE/32 将是更好的选择。

但是，如果 MSI 时钟频率较低（通常低于 1 MHz），测量精度会不够好。在这种情况下，建议：

- 累加连续多次捕获的结果。
- 使用定时器的输入捕获预分频值（最多每 8 个周期进行 1 次捕获）。
- 使用 RTC 唤醒中断信号（如果 RTC 的时钟由 LSE 提供）作为通道 1 输入捕获的输入。这样可提高测量精度。为此，必须使能 RTC 唤醒中断。

LSI 校准

LSI 校准将遵循与 HSI16 相同的模式，只是参考时钟有所不同。需要将 LSI 时钟连接到 TIM16 的通道 1 输入捕获。然后将 HSE 定义为系统时钟源，借助 LSI 信号连续边沿之间的时钟计数数量，即可对内部低速时钟周期进行测量。

其基本原理是基于相对的测量（例如，HSE/LSI 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。

6.2.18 外设时钟使能寄存器（RCC_AHBxENR、RCC_APBxENRy）

每个外设时钟均可由 RCC_AHBxENR、RCC_APBxENRy 寄存器的 xxxxEN 位使能。

当外设时钟未激活时，不支持外设寄存器进行读写访问。

使能位的同步机制可为外设产生无干扰时钟。使能位置 1 后，在时钟激活前存在时长为 2 个时钟周期的延迟。

注意： 使能外设时钟后，软件必须等待一段延迟才能访问外设寄存器。

6.3 低功耗模式

- 可通过软件禁止 AHB 和 APB 外设时钟，包括 DMA 时钟。
- 在睡眠和低功耗睡眠模式下，CPU 时钟停止工作。在睡眠模式下，可通过软件停止存储器接口时钟（Flash、SRAM1、SRAM2 和 SRAM3 接口）。当连接到 AHB-APB 总线桥时钟的所有外设时钟均被禁止时，睡眠模式期间将通过硬件禁止 AHB-APB 总线桥时钟。
- 在停止模式（停止 0、停止 1 和停止 2）下，将停止 V_{CORE} 域中的所有时钟，并禁止三个 PLL（HSI16、MSI 和 HSE 振荡器）。

  即使 MCU 处于停止模式（如果选择 HSI16 作为该外设的时钟源），所有 U(S)ART、LPUART 和 I^2C 也均可使能 HSI16 振荡器。

  当系统处于停止模式（如果选择 LSE 作为该外设的时钟源）并使能 LSE 振荡器 (LSEON) 时，所有 U(S)ART 和 LPUART 也均可由 LSE 振荡器驱动。在这种情况下，LSE 在停止模式下始终保持开启（它们无法开启 LSE 振荡器）。
- 在待机和关断模式下，将停止 V_{CORE} 域中的所有时钟，并禁止 PLL（HSI16、MSI 和 HSE 振荡器）。

通过将 DBGMCU_CR 寄存器中的 DBG_STOP 或 DBG_STANDBY 位置 1，可以覆盖 CPU 的深度睡眠模式以进行调试。

当退出停止模式（停止 0、停止 1 或停止 2）时，系统时钟为 MSI 或 HSI16，具体取决于 RCC_CFGR 寄存器中 STOPWUCK 位的软件配置。MSI 振荡器的频率（范围和用户微调）是进入停止模式之前配置的频率。HSI16 的用户微调将保留。如果 MSI 在进入停止模式之前处于 PLL 模式，则即使 LSE 在停止模式期间保持开启，也必须在唤醒后等待 PLL 模式稳定时间。

退出待机和关断模式时，系统时钟为 MSI。从待机模式唤醒时的 MSI 频率通过 RCC_CSR 寄存器中的 MSISRANGE 位配置，范围为 1 MHz 到 8 MHz。从关断模式唤醒时的 MSI 频率为 4 MHz。用户微调将丢失。

如果正在执行 Flash 编程操作，则将延迟到 Flash 接口访问结束后再进入停止、待机和关断模式。如果正在访问 APB 域，则将延迟到 APB 访问结束后再进入停止、待机和关断模式。

6.4 RCC 寄存器

6.4.1 时钟控制寄存器 (RCC_CR)

Clock control register

偏移地址：0x00

复位值：0x0000 0063。HSEBYP 不受复位影响。

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
Res.	Res.	PLLSAI 2RDY	PLLSAI 2ON	PLLSAI 1RDY	PLLSAI 1ON	PLLRD Y	PLLON	Res.	Res.	Res.	Res.	CSS ON	HSE BYP	HSE RDY	HSE ON
		r	rw	r	rw	r	rw					rs	rw	r	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	HSIAS FS	HSIRD Y	HSIKER ON	HSION	MSIRANGE[3:0]				MSIRG SEL	MSIPL LEN	MSI RDY	MSION
				rw	r	rw	rw	rw	rw	rw	rw	rs	rw	r	rw

位 31:30 保留，必须保持复位值。

位 29 **PLLSAI2RDY**：SAI2 PLL 时钟就绪标志 (SAI2 PLL clock ready flag)
由硬件置 1，用于指示 PLLSAI2 已锁定。
0：PLLSAI2 未锁定
1：PLLSAI2 已锁定

位 28 **PLLSAI2ON**：SAI2 PLL 使能 (SAI2 PLL enable)
由软件置 1 和清零，用于使能 PLLSAI2。
当进入停止、待机和关断模式时由硬件清零。
0：PLLSAI2 关闭
1：PLLSAI2 开启

位 27 **PLLSAI1RDY**：SAI1 PLL 时钟就绪标志 (SAI1 PLL clock ready flag)
由硬件置 1，用于指示 PLLSAI1 已锁定。
0：PLLSAI1 未锁定
1：PLLSAI1 已锁定

位 26 **PLLSAI1ON**：SAI1 PLL 使能 (SAI1 PLL enable)
由软件置 1 和清零，用于使能 PLLSAI1。
当进入停止、待机和关断模式时由硬件清零。
0：PLLSAI1 关闭
1：PLLSAI1 开启

位 25 **PLLRDY**：主 PLL 时钟就绪标志 (Main PLL clock ready flag)
由硬件置 1，用于指示主 PLL 已锁定。
0：PLL 未锁定
1：PLL 已锁定

位 24 **PLLON**：主 PLL 使能 (Main PLL enable)
由软件置 1 和清零，用于使能主 PLL。
当进入停止、待机和关断模式时由硬件清零。如果 PLL 时钟用作系统时钟，则此位不可清零。
0：PLL 关闭
1：PLL 开启

位 23:20 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，用于使能时钟安全系统。当 CSSON 置 1 时，时钟监测器将在 HSE 振荡器就绪时由硬件使能，并在检出 HSE 时钟故障时由硬件禁止。该位仅置 1，并通过复位清零。

0：时钟安全系统关闭（时钟监测器关闭）

1：时钟安全系统开启（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 晶振旁路 (HSE crystal oscillator bypass)

由软件置 1 和清零，用于用外部时钟旁路振荡器。外部时钟必须通过将 HSEON 位置 1 的方式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 晶振

1：外部时钟旁路 HSE 晶振

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。

0：HSE 振荡器未就绪

1：HSE 振荡器已就绪

注：　在将 HSEON 位清零后，HSERDY 将在 6 个 HSE 时钟周期后转为低电平。

位 16 **HSEON**：HSE 时钟使能 (HSE clock enable)

由软件置 1 和清零。

由硬件清零，用于在进入停止、待机和关断模式时停止 HSE 振荡器。如果 HSE 振荡器直接或间接用作系统时钟，则该位不可复位。

0：HSE 振荡器关闭

1：HSE 振荡器开启

位 15:12 保留，必须保持复位值。

位 11 **HSIASFS**：HSI16 从停止模式自动启动 (HSI16 automatic start from Stop)

由软件置 1 和清零。当系统唤醒时钟为 MSI 时，该位用于在系统唤醒的同时唤醒 HSI16。

0：在 MSI 作为唤醒时钟的情况下退出停止模式时，不通过硬件使能 HSI16 振荡器。

1：在 MSI 作为唤醒时钟的情况下退出停止模式时，通过硬件使能 HSI16 振荡器。

位 10 **HSIRDY**：HSI16 时钟就绪标志 (HSI16 clock ready flag)

由硬件置 1，用于指示 HSI16 振荡器已稳定。仅当通过软件将 HSION 置 1 来使能 HSI16 时，该位才置 1。

0：HSI16 振荡器未就绪

1：HSI16 振荡器已就绪

注：　在将 HSION 位清零后，HSIRDY 将在 6 个 HSI16 时钟周期后转为低电平。

位 9 **HSIKERON**：始终为外设内核使能 HSI16 (HSI16 always enable for peripheral kernels)

由软件置 1 和清零，用于强制 HSI16 开启（即使在停止模式下）。HSI16 只能为将 HSI16 配置为内核时钟的 USART 和 I^2C 外设提供时钟。通过使 HSI16 在停止模式下保持开启，可避免由于 HSI16 启动时间而导致的通信速度变慢。该位对 HSION 值没有任何影响。

0：对 HSI16 振荡器无影响。

1：即使在停止模式下也强制 HSI16 振荡器开启。

位 8 **HSION**：HSI16 时钟使能 (HSI16 clock enable)

由软件置 1 和清零。

由硬件清零，用于在进入停止、待机和关断模式时停止 HSI16 振荡器。

由硬件置 1，用于在退出停止模式时或者在 HSE 晶振发生故障时强制 HSI16 振荡器开启（此时，STOPWUCK=1 或 HSIASFS = 1）。

如果 HSI16 直接或间接用作系统时钟，则此位由硬件置 1。

0：HSI16 振荡器关闭

1：HSI16 振荡器开启

位 7:4 **MSIRANGE[3:0]**：MSI 时钟范围 (MSI clock ranges)

当 MSIRGSEL 置 1 时，这些位由软件配置，用于选择 MSI 的频率范围。总共有 12 种频率范围可供选择：

0000：范围 0，100 kHz 左右

0001：范围 1，200 kHz 左右

0010：范围 2，400 kHz 左右

0011：范围 3，800 kHz 左右

0100：范围 4，1 MHz 左右

0101：范围 5，2 MHz 左右

0110：范围 6，4 MHz 左右（复位值）

0111：范围 7，8 MHz 左右

1000：范围 8，16 MHz 左右

1001：范围 9，24 MHz 左右

1010：范围 10，32 MHz 左右

1011：范围 11，48 MHz 左右

其他值：不允许使用（硬件写保护）

注： 警告：当 MSI 关闭 (MSION=0) 或 MSI 就绪 (MSIRDY=1) 时，可以修改 MSIRANGE。当 MSI 开启且未就绪（MSION = 1 且 MSIRDY = 0）时，不得修改 MSIRANGE。

位 3 **MSIRGSEL**：MSI 时钟范围选择 (MSI clock range selection)

由软件置 1，用于使用 MSIRANGE[3:0] 选择 MSI 时钟范围。写入 0 无影响。

待机或复位后 MSIRGSEL 为 0，MSI 范围值由 CSR 寄存器中的 MSISRANGE 提供。

0：MSI 范围由 RCC_CSR 寄存器中的 MSISRANGE[3:0] 提供

1：MSI 范围由 RCC_CR 寄存器中的 MSIRANGE[3:0] 提供

位 2 **MSIPLLEN**：MSI 时钟 PLL 使能 (MSI clock PLL enable)

由软件置 1 和清零，用于使能/禁止 MSI 时钟源的 PLL 部分。

在 LSE 使能（已使能 LSEON）和就绪（由硬件将 LSERDY 置 1）后，MSIPLLEN 必须使能。为了避免在 LSE 未就绪的情况下使能 MSIPLLEN，提供了相应的硬件保护。

当 LSE 禁止 (LSEON = 0) 或 LSE 上的时钟安全系统检测到 LSE 故障时（请参见 RCC_CSR 寄存器），该位由硬件清零。

0：MSI PLL 关闭

1：MSI PLL 开启

位 1 **MSIRDY**：MSI 时钟就绪标志 (MSI clock ready flag)

该位由硬件置 1，用于指示 MSI 振荡器已稳定。

0：MSI 振荡器未就绪

1：MSI 振荡器已就绪

注： 在将 MSION 位清零后，MSIRDY 将在 6 个 MSI 时钟周期后转为低电平。

位 0 **MSION**：MSI 时钟使能 (MSI clock enable)

此位由软件置 1 和清零。

由硬件清零，用于在进入停止、待机和关断模式时停止 MSI 振荡器。

由硬件置 1，用于在退出待机和关断模式时强制 MSI 振荡器开启。

由硬件置 1，用于在退出停止模式时或者在 HSE 振荡器发生故障时强制 MSI 振荡器开启（此时，STOPWUCK=0）。

直接或间接用作系统时钟时，由硬件置 1。

0：MSI 振荡器关闭

1：MSI 振荡器开启

6.4.2 内部时钟源校准寄存器 (RCC_ICSCR)

Internal clock sources calibration register

偏移地址：0x04

复位值：0x40XX 00XX，其中 X 是出厂前编程的。

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

位 31 保留，必须保持复位值。

位 30:24 **HSITRIM[6:0]**：HSI16 时钟微调 (HSI16 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 HSI16 的频率更为准确。

默认值为 16，加上 HSICAL 值，应能将 HSI16 微调至 16 MHz ± 1 %。

位 23:16 **HSICAL[7:0]**：HSI16 时钟校准 (HSI16 clock calibration)

这些位在启动时初始化为出厂前编程的 HSI16 校准微调值。写入 HSITRIM 后，HSICAL 将更新为 HSITRIM 和出厂微调值之和。

位 15:8 **MSITRIM[7:0]**：MSI 时钟微调 (MSI clock trimming)

通过这些位，可在 MSICAL[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 MSI 的频率更为准确。

位 7:0 **MSICAL[7:0]**：MSI 时钟校准 (MSI clock calibration)

这些位在启动时初始化为出厂前编程的 MSI 校准微调值。写入 MSITRIM 后，MSICAL 将更新为 MSITRIM 和出厂微调值之和。

6.4.3 时钟配置寄存器 (RCC_CFGR)

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。

如果在更新 APB 或 AHB 预分频器值时进行访问，则插入 0 到 15 个等待周期。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	MCOPRE[2:0]			MCOSEL[3:0]				Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	rw	rw	rw	rw	rw	rw	rw								
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STOP WUCK	Res.	PPRE2[2:0]			PPRE1[2:0]			HPRE[3:0]				SWS[1:0]		SW[1:0]	
rw		rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	r	r	rw	rw

位 31 保留，必须保持复位值。

位 30:28 **MCOPRE[2:0]**：微控制器时钟输出预分频器 (Microcontroller clock output prescaler)

这些位将由软件置 1 和清零。

强烈建议在 MCO 输出使能之前更改此预分频器。

000：MCO 1 分频

001：MCO 2 分频

010：MCO 4 分频

011：MCO 8 分频

100：MCO 16 分频

其他值：不允许使用

位 27:24 **MCOSEL[3:0]**：微控制器时钟输出 (Microcontroller clock output)

由软件置 1 和清零。

0000：禁止 MCO 输出，MCO 上无时钟

0001：选择 SYSCLK 系统时钟

0010：选择 MSI 时钟

0011：选择 HSI16 时钟

0100：选择 HSE 时钟

0101：选择主 PLL 时钟

0110：选择 LSI 时钟

0111：选择 LSE 时钟

1000：选择内部 HSI48 时钟

其他值：保留

注： 该时钟输出在启动时或 MCO 时钟源切换期间可能包含一些被截断的周期。

位 23:16 保留，必须保持复位值。

位 15 **STOPWUCK**：从停止模式唤醒时的时钟以及 CSS 备份时钟选择 (Wakeup from Stop and CSS backup clock selection)

由软件置 1 和清零，用于选择退出停止模式时使用的系统时钟。

所选时钟还用作 HSE 上的时钟安全系统的紧急时钟。警告：时钟安全系统已通过 RCC_CR 寄存器中的 HSECSSON 使能且系统时钟为 HSE（SWS=“10”）或者请求开启 HSE（SW=“10”）时，不能修改 STOPWUCK。

0：选择 MSI 振荡器作为从停止模式唤醒时的时钟以及 CSS 备份时钟。

1：选择 HSI16 振荡器作为从停止模式唤醒时的时钟以及 CSS 备份时钟。

位 14 保留，必须保持复位值。

位 13:11 **PPRE2[2:0]**：APB 高速预分频器 (APB2) (APB high-speed prescaler (APB2))
由软件置 1 和清零，用于控制 APB2 时钟 (PCLK2) 的分频系数。
0xx：HCLK 不分频
100：HCLK 2 分频
101：HCLK 4 分频
110：HCLK 8 分频
111：HCLK 16 分频

位 10:8 **PPRE1[2:0]**：APB 低速预分频器 (APB1) (APB low-speed prescaler (APB1))
由软件置 1 和清零，用于控制 APB1 时钟 (PCLK1) 的分频系数。
0xx：HCLK 不分频
100：HCLK 2 分频
101：HCLK 4 分频
110：HCLK 8 分频
111：HCLK 16 分频

位 7:4 **HPRE[3:0]**：AHB 预分频器 (AHB prescaler)
由软件置 1 和清零，用于控制 AHB 时钟的分频系数。

注意： 软件必须根据器件电压范围正确设置这些位，以确保系统频率不会超过允许的最大频率（更多详细信息，请参见*第 5.1.7 节：动态电压调节管理*）。在对这些位执行写操作之后以及减小电压范围之前，必须读取该寄存器以确保其中的值为新值。

0xxx：SYSCLK 不分频
1000：SYSCLK 2 分频
1001：SYSCLK 4 分频
1010：SYSCLK 8 分频
1011：SYSCLK 16 分频
1100：SYSCLK 64 分频
1101：SYSCLK 128 分频
1110：SYSCLK 256 分频
1111：SYSCLK 512 分频

位 3:2 **SWS[1:0]**：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：MSI 振荡器用作系统时钟
01：HSI16 振荡器用作系统时钟
10：HSE 用作系统时钟
11：PLL 用作系统时钟

位 1:0 **SW[1:0]**：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 (SYSCLK)。
由硬件配置，用于在退出待机和关断模式时强制选择 MSI 振荡器。由硬件配置，用于在退出停止模式时或者在 HSE 振荡器发生故障时强制选择 MSI 或 HSI16 振荡器（取决于 STOPWUCK 值）。
00：选择 MSI 作为系统时钟
01：选择 HSI16 作为系统时钟
10：选择 HSE 作为系统时钟
11：选择 PLL 作为系统时钟

6.4.4 PLL 配置寄存器 (RCC_PLLCFGR)

PLL configuration register

偏移地址：0x0C

复位值：0x0000 1000

访问：无等待周期，按字、半字和字节访问

此寄存器用于根据公式配置 PLL 时钟输出：

- f(VCO 时钟) = f(PLL 时钟输入) × (PLLN / PLLM)
- f(PLL_P) = f(VCO 时钟) / PLLP
- f(PLL_Q) = f(VCO 时钟) / PLLQ
- f(PLL_R) = f(VCO 时钟) / PLLR

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PLLPDIV[4:0]					PLLR[1:0]		PLLREN	Res.	PLLQ[1:0]		PLLQEN	Res.	Res.	PLLP	PLLPEN
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.	PLLN[6:0]							PLLM[3:0]				Res.	Res.	PLLSRC[1:0]	
	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw			rw	rw

位 31:27 **PLLPDIV[4:0]**：适用于 PLLSAI3CLK 的主 PLLP 分频系数 (Main PLLP division factor for PLLSAI3CLK)

由软件置 1 和清零，用于控制 SAI1 或 SAI2 时钟频率。PLLSAI3CLK 输出时钟频率 = VCO 频率 / PLLPDIV。

00000：PLLSAI3CLK 由位 PLLP 控制

00001：保留。

00010：PLLSAI3CLK = VCO / 2

....

11111：PLLSAI3CLK = VCO / 31

位 26:25 **PLLR[1:0]**：适用于 PLLCLK（系统时钟）的主 PLL 分频系数 (Main PLL division factor for PLLCLK (system clock))

由软件置 1 和清零，用于控制主 PLL 输出时钟 PLLCLK 的频率。可以选择该输出作为系统时钟。这些位只能在 PLL 已禁止时写入。

PLLCLK 输出时钟频率 = VCO 频率 / PLLR，其中，PLLR = 2、4、6 或 8

00：PLLR = 2

01：PLLR = 4

10：PLLR = 6

11：PLLR = 8

注意： 软件必须正确设置这些位，使其在此域中不超过 120 MHz。

位 24 **PLLREN**：主 PLL PLLCLK 输出使能 (Main PLL PLLCLK output enable)

由软件置 1 和复位，用于使能主 PLL 的 PLLCLK 输出（用作系统时钟）。

当 PLL 的 PLLCLK 输出用作系统时钟时，该位不可写入。

为实现节能，不使用 PLL 的 PLLCLK 输出时，PLLREN 的值应为 0。

0：禁止 PLLCLK 输出

1：使能 PLLCLK 输出

位 23 保留，必须保持复位值。

位 22:21 **PLLQ[1:0]**：适用于 PLL48M1CLK（48 MHz 时钟）的主 PLL 分频系数 (Main PLL division factor for PLL48M1CLK (48 MHz clock))。

由软件置 1 和清零，用于控制主 PLL 输出时钟 PLL48M1CLK 的频率。可为 USB、RNG、SDMMC（48 MHz 时钟）选择此输出。这些位只能在 PLL 已禁止时写入。

PLL48M1CLK 输出时钟频率 = VCO 频率 / PLLQ，其中，PLLQ = 2、4、6 或 8

00：PLLQ = 2

01：PLLQ = 4

10：PLLQ = 6

11：PLLQ = 8

注意： 软件必须正确设置这些位，使其在此域中不超过 120 MHz。

位 20 **PLLQEN**：主 PLL PLL48M1CLK 输出使能 (Main PLL PLL48M1CLK output enable)

由软件置 1 和复位，用于使能主 PLL 的 PLL48M1CLK 输出。

为实现节能，不使用 PLL 的 PLL48M1CLK 输出时，PLLQEN 的值应为 0。

0：禁止 PLL48M1CLK 输出

1：使能 PLL48M1CLK 输出

位 19:18 保留，必须保持复位值。

位 17 **PLLP**：适用于 PLLSAI3CLK（SAI1 和 SAI2 时钟）的主 PLL 分频系数 (Main PLL division factor for PLLSAI3CLK (SAI1 and SAI2 clock))。

由软件置 1 和清零，用于控制主 PLL 输出时钟 PLLSAI3CLK 的频率。可为 SAI1 或 SAI2 选择此输出。这些位只能在 PLL 已禁止时写入。

当 PLLPDIV[4:0] 设置为“00000”时，PLLSAI3CLK 输出时钟频率 = VCO 频率 / PLLP，其中，PLLP = 7 或 17

0：PLLP = 7

1：PLLP = 17

注意： 软件必须正确设置这些位，使其在此域中不超过 120 MHz。

位 16 **PLLPEN**：主 PLL PLLSAI3CLK 输出使能 (Main PLL PLLSAI3CLK output enable)

由软件置 1 和复位，用于使能主 PLL 的 PLLSAI3CLK 输出。

为实现节能，不使用 PLL 的 PLLSAI3CLK 输出时，PLLPEN 的值应为 0。

0：禁止 PLLSAI3CLK 输出

1：使能 PLLSAI3CLK 输出

位 15 保留，必须保持复位值。

位 14:8 **PLLN[6:0]**：适用于 VCO 的主 PLL 倍频系数 (Main PLL multiplication factor for VCO)

由软件置 1 和清零，用于控制 VCO 的倍频系数。这些位只能在 PLL 已禁止时写入。

VCO 输出频率 = VCO 输入频率 x PLLN，其中，8 =< PLLN =< 127

0000000：PLLN = 0，错误配置

0000001：PLLN = 1，错误配置

...

0000111：PLLN = 7，错误配置

0001000：PLLN = 8

0001001：PLLN = 9

...

1111111：PLLN = 127

注意： 软件必须正确设置这些位，确保 VCO 输出频率介于 64 MHz 和 344 MHz 之间。

位 7:4 **PLLM**：适用于主 PLL 输入时钟的分频系数 (Division factor for the main PLL input clock)

由软件置 1 和清零，用于在 VCO 之前对 PLL 输入时钟进行分频。这些位只有在所有 PLL 已禁止时才可写入。

VCO 输入频率 = PLL 输入时钟频率 / PLLM，其中 1 <= PLLM <= 16

0000：PLLM = 1

0001：PLLM = 2

0010：PLLM = 3

0011：PLLM = 4

0100：PLLM = 5

0101：PLLM = 6

0110：PLLM = 7

0111：PLLM = 8

注意： 软件必须正确设置这些位，确保 VCO 输入频率介于 2.66 MHz 和 8 MHz 之间。

位 3:2 保留，必须保持复位值。

位 1:0 **PLLSRC**：主 PLL 输入时钟源 (Main PLL entry clock source)

由软件置 1 和清零，用于选择 PLL 时钟源。这些位只能在 PLL 已禁止时写入。

为实现节能，不使用 PLL 时，PLLSRC 的值应为 00。

00：未向 PLL 发送任何时钟

01：选择 MSI 时钟作为 PLL 时钟输入

10：选择 HSI16 时钟作为 PLL 时钟输入

11：选择 HSE 时钟作为 PLL 时钟输入

6.4.5 PLLSAI1 配置寄存器 (RCC_PLLSAI1CFGR)

PLLSAI1 configuration register

偏移地址：0x10

复位值：0x0000 1000

访问：无等待周期，按字、半字和字节访问

此寄存器用于根据公式配置 PLLSAI1 时钟输出：

- f(VCOSAI1 时钟) = f(PLL 时钟输入) × (PLLSAI1N / PLLSAI1M)
- f(PLLSAI1_P) = f(VCOSAI1 时钟) / PLLSAI1PDIV
- f(PLLSAI1_Q) = f(VCOSAI1 时钟) / PLLSAI1Q
- f(PLLSAI1_R) = f(VCOSAI1 时钟) / PLLSAI1R

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PLLSAI1PDIV[4:0]					PLLSAI1R[1:0]		PLLSAI 1REN	Res.	PLLSAI1Q[1:0]		PLLSAI 1QEN	Res.	Res.	PLLSAI 1P	PLLSAI 1PEN
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.	PLLSAI1N[6:0]							PLLSAI1M[3:0]				Res.	Res.	Res.	Res.
	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw				

位 31:27 **PLLSAI1PDIV[4:0]**：适用于 PLLSAI1CLK 的 PLLSAI1 倍频系数 (PLLSAI1 division factor for PLLSAI1CLK)

由软件置 1 和清零，用于控制 SAI1 或 SAI2 时钟频率。PLLSAI1CLK 输出时钟频率 = VCOSAI1 频率 / PLLSAI1PDIV。

00000：PLLSAI1CLK 由位 PLLSAI1P 控制
00001：保留
00010：PLLSAI1CLK = VCOSAI1 / 2
....
11111：PLLSAI1CLK = VCOSAI1 / 31

注：　只有在禁止 PLLSAI1 后才能写入该位。

位 26:25 **PLLSAI1R[1:0]**：适用于 PLLADC1CLK（ADC 时钟）的 PLLSAI1 分频系数 (PLLSAI1 division factor for PLLADC1CLK (ADC clock))

由软件置 1 和清零，用于控制 PLLSAI1 输出时钟 PLLADC1CLK 的频率。可以选择该输出作为 ADC 时钟。这些位只能在 PLLSAI1 已禁止时写入。

PLLADC1CLK 输出时钟频率 = VCOSAI1 频率 / PLLSAI1R，其中，PLLSAI1R = 2、4、6 或 8

00：PLLSAI1R = 2
01：PLLSAI1R = 4
10：PLLSAI1R = 6
11：PLLSAI1R = 8

位 24 **PLLSAI1REN**：PLLSAI1 PLLADC1CLK 输出使能 (PLLSAI1 PLLADC1CLK output enable)

由软件置 1 和复位，用于使能 PLLSAI1 的 PLLADC1CLK 输出（用作 ADC 时钟）。

为实现节能，不使用 PLLSAI1 的 PLLADC1CLK 输出时，PLLSAI1REN 的值应为 0。

0：禁止 PLLADC1CLK 输出
1：使能 PLLADC1CLK 输出

位 23 保留，必须保持复位值。

位 22:21 **PLLSAI1Q[1:0]**：适用于 PLL48M2CLK（48 MHz 时钟）的 PLLSAI1 分频系数 (PLLSAI1 division factor for PLL48M2CLK (48 MHz clock))

由软件置 1 和清零，用于控制 PLLSAI1 输出时钟 PLL48M2CLK 的频率。可为 USB、RNG、SDMMC（48 MHz 时钟）选择此输出。这些位只能在 PLLSAI1 已禁止时写入。

PLL48M2CLK 输出时钟频率 = VCOSAI1 频率 / PLLSAI1Q，其中，PLLSAI1Q = 2、4、6 或 8

00：PLLSAI1Q = 2

01：PLLSAI1Q = 4

10：PLLSAI1Q = 6

11：PLLSAI1Q = 8

注意： 软件必须正确设置这些位，使其在此域中不超过 120 MHz。

位 20 **PLLSAI1QEN**：PLLSAI1 PLL48M2CLK 输出使能 (PLLSAI1 PLL48M2CLK output enable)

由软件置 1 和复位，用于使能 PLLSAI1 的 PLL48M2CLK 输出。

为实现节能，不使用 PLLSAI1 的 PLL48M2CLK 输出时，PLLSAI1QEN 的值应为 0。

0：禁止 PLL48M2CLK 输出

1：使能 PLL48M2CLK 输出

位 19:18 保留，必须保持复位值。

位 17 **PLLSAI1P**：适用于 PLLSAI1CLK（SAI1 或 SAI2 时钟）的 PLLSAI1 分频系数 (PLLSAI1 division factor for PLLSAI1CLK (SAI1 or SAI2 clock))。

由软件置 1 和清零，用于控制 PLLSAI1 输出时钟 PLLSAI1CLK 的频率。可为 SAI1 或 SAI2 选择此输出。这些位只能在 PLLSAI1 已禁止时写入。

当 PLLSAI1PDIV[4:0] 设置为“00000”时，PLLSAI1CLK 输出时钟频率 = VCOSAI1 频率 / PLLSAI1P，其中，PLLSAI1P = 7 或 17

0：PLLSAI1P = 7

1：PLLSAI1P = 17

位 16 **PLLSAI1PEN**：PLLSAI1 PLLSAI1CLK 输出使能 (PLLSAI1 PLLSAI1CLK output enable)

由软件置 1 和复位，用于使能 PLLSAI1 的 PLLSAI1CLK 输出。

为实现节能，不使用 PLLSAI1 的 PLLSAI1CLK 输出时，PLLSAI1PEN 的值应为 0。

0：禁止 PLLSAI1CLK 输出

1：使能 PLLSAI1CLK 输出

位 15 保留，必须保持复位值。

位 14:8 **PLLSAI1N[6:0]**：适用于 VCO 的 PLLSAI1 倍频系数 (PLLSAI1 multiplication factor for VCO)
由软件置 1 和清零，用于控制 VCO 的倍频系数。这些位只能在 PLLSAI1 已禁止时写入。
VCOSAI1 输出频率 = VCOSAI1 输入频率 x PLLSAI1N
其中，8 =< PLLSAI1N =< 127
0000000：PLLSAI1N = 0，错误配置
0000001：PLLSAI1N = 1，错误配置
...
0000111：PLLSAI1N = 7，错误配置
0001000：PLLSAI1N = 8
0001001：PLLSAI1N = 9
...
1111111：PLLSAI1N = 127

注意： 软件必须正确设置这些位，确保 VCO 输出频率介于 64 MHz 和 344 MHz 之间。

位 7:4 **PLLSAI1M**：适用于 PLLSAI1 输入时钟的分频系数 (Division factor for PLLSAI1 input clock)
由软件置 1 和复位，用于在 VCO 之前对 PLLSAI1 输入时钟进行分频。
这些位只能在 PLLSAI1 已禁止时写入。
VCO 输入频率 = PLLSAI1 输入时钟频率 / PLLSAI1M，其中，1 <= PLLSAI1M <= 16
0000：PLLSAI1M = 1
0001：PLLSAI1M = 2
0010：PLLSAI1M = 3
0011：PLLSAI1M = 4
0100：PLLSAI1M = 5
0101：PLLSAI1M = 6
0110：PLLSAI1M = 7
0111：PLLSAI1M = 8
1000：PLLSAI1M = 9
...
1111：PLLSAI1M = 16

注意： 软件必须正确设置这些位，确保 VCO 输入频率介于 2.66 和 8 MHz 之间。

位 3:0 保留，必须保持复位值。

6.4.6 PLLSAI2 配置寄存器 (RCC_PLLSAI2CFGR)

PLLSAI2 configuration register

偏移地址：0x14

复位值：0x0000 1000

访问：无等待周期，按字、半字和字节访问

此寄存器用于根据公式配置 PLLSAI2 时钟输出：

- f(VCOSAI2 时钟) = f(PLL 时钟输入) × (PLLSAI2N / PLLSAI2M)
- f(PLLSAI2_P) = f(VCOSAI2 时钟) / PLLSAI2PDIV
- f(PLLSAI2_R) = f(VCOSAI2 时钟) / PLLSAI2R

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PLLSAI2PDIV[4:0]					PLLSAI2R[1:0]		PLLSAI 2REN	Res.	PLLSAI2Q		PLLSAI 2QEN	Res.	Res.	PLLSAI 2P	PLLSAI 2PEN
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.	PLLSAI2N[6:0]							PLLSAI2M[3:0]				Res.	Res.	Res.	Res.
	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw				

位 31:27 **PLLSAI2PDIV[4:0]**：适用于 PLLSAI2CLK 的 PLLSAI2 倍频系数 (PLLSAI2 division factor for PLLSAI2CLK)

由软件置 1 和清零，用于控制 SAI1 或 SAI2 时钟频率。PLLSAI2CLK 输出时钟频率 = VCOSAI2 频率 / PLLSAI2PDIV。

00000：PLLSAI2CLK 由位 PLLSAI2P 控制

00001：保留

00010：PLLSAI2CLK = VCOSAI2 / 2

....

11111：PLLSAI2CLK = VCOSAI2 / 31

位 26:25 **PLLSAI2R[1:0]**：适用于 PLLLCDCLK（LTDC 时钟）的 PLLSAI2 分频系数 (PLLSAI2 division factor for PLLLCDCLK (LTDC clock))

由软件置 1 和清零，用于控制 PLLSAI2 输出时钟 PLLLCDCLK 的频率。可以选择该输出作为 ADC 时钟。这些位只能在 PLLSAI2 已禁止时写入。

PLLLCDCLK 输出时钟频率 = VCOSAI2 频率 / PLLSAI2R，其中，PLLSAI2R = 2、4、6 或 8

00：PLLSAI2R = 2

01：PLLSAI2R = 4

10：PLLSAI2R = 6

11：PLLSAI2R = 8

位 24 **PLLSAI2REN**：PLLSAI2 PLLLCDCLK 输出使能 (PLLSAI2 PLLLCDCLK output enable)

由软件置 1 和复位，用于使能 PLLSAI2 的 PLLLCDCLK 输出（用作 ADC 时钟）。

为实现节能，不使用 PLLSAI2 的 PLLLCDCLK 输出时，PLLSAI2REN 的值应为 0。

0：禁止 PLLLCDCLK 输出

1：使能 PLLLCDCLK 输出

位 23 保留，必须保持复位值。

位 22:21 **PLLSAI2Q[1:0]**：PLLSAI2 PLLDSICLK 输出使能 (PLLSAI2 PLLDSICLK output enable)。
由软件置 1 和清零，用于控制 DSI 时钟的频率。
这些位只能在 PLLSAI2 已禁止时写入。
PLLDSICLK 输出时钟频率 = VCOSAI2 频率 / PLLSAI2Q，其中，PLLSAI2Q = 2、4、6 或 8
00：PLLSAI2Q = 2
01：PLLSAI2Q = 4
10：PLLSAI2Q = 6
11：PLLSAI2Q = 8

位 20 **PLLSAI2QEN**：适用于 PLLDSICLK（DSI 时钟）的 PLLSAI2 分频系数 (PLLSAI2 division factor for PLLDSICLK (DSI clock))。
由软件置 1 和复位，用于使能 PLLSAI2 的 PLLDSICLK（DSI 时钟）输出（用作 USB 时钟）。
为实现节能，不使用 PLLSAI2 的 PLLDSICLK 输出时，PLLSAI2PEN 的值应为 0。
0：禁止 PLLDSICLK 输出
1：使能 PLLDSICLK 输出

位 19:18 保留，必须保持复位值。

位 17 **PLLSAI2P**：适用于 PLLSAI2CLK（SAI1 或 SAI2 时钟）的 PLLSAI2 分频系数 (PLLSAI2 division factor for PLLSAI2CLK (SAI1 or SAI2 clock))。
由软件置 1 和清零，用于控制 PLLSAI2 输出时钟 PLLSAI2CLK 的频率。可为 SAI1 或 SAI2 选择此输出。这些位只能在 PLLSAI2 已禁止时写入。
（当 PLLSAI2PDIV[4:0] 设置为“00000”时），PLLSAI2CLK 输出时钟频率 = VCOSAI2 频率 / PLLSAI2P，其中，PLLSAI2P = 7 或 17
0：PLLSAI2P = 7
1：PLLSAI2P = 17

位 16 **PLLSAI2PEN**：PLLSAI2 PLLSAI2CLK 输出使能 (PLLSAI2 PLLSAI2CLK output enable)
由软件置 1 和复位，用于使能 PLLSAI2 的 PLLSAI2CLK 输出。
为实现节能，不使用 PLLSAI2 的 PLLSAI2CLK 输出时，PLLSAI2PEN 的值应为 0。
0：禁止 PLLSAI2CLK 输出
1：使能 PLLSAI2CLK 输出

位 15 保留，必须保持复位值。

位 14:8 **PLLSAI2N[6:0]**：适用于 VCO 的 PLLSAI2 倍频系数 (PLLSAI2 multiplication factor for VCO)
由软件置 1 和清零，用于控制 VCO 的倍频系数。这些位只能在 PLLSAI2 已禁止时写入。
VCOSAI2 输出频率 = VCOSAI2 输入频率 x PLLSAI2N
其中，8 =< PLLSAI2N =< 127
0000000：PLLSAI2N = 0，错误配置
0000001：PLLSAI2N = 1，错误配置
...
0000111：PLLSAI2N = 7，错误配置
0001000：PLLSAI2N = 8
0001001：PLLSAI2N = 9
...
1111111：PLLSAI2N = 127

注意： 软件必须正确设置这些位，确保 VCO 输出频率介于 64 MHz 和 344 MHz 之间。

位 7:4 **PLLSAI2M**：适用于 PLLSAI2 输入时钟的分频系数 (Division factor for PLLSAI2 input clock)
由软件置 1 和复位，用于在 VCO 之前对 PLLSAI2 输入时钟进行分频。
这些位只能在 PLLSAI2 已禁止时写入。
VCO 输入频率 = PLLSAI2 输入时钟频率 / PLLM，其中 1 <= PLLSAI2M <= 16
0000：PLLSAI2M = 1
0001：PLLSAI2M = 2
0010：PLLSAI2M = 3
0011：PLLSAI1M = 4
0100：PLLSAI2M = 5
0101：PLLSAI2M = 6
0110：PLLSAI2M = 7
0111：PLLSAI2M = 8
1000：PLLSAI2M = 9
...
1111：PLLSAI2M = 16

注意： 软件必须正确设置这些位，确保 VCO 输入频率介于 2.66 和 8 MHz 之间。

位 3:0 保留，必须保持复位值。

6.4.7 时钟中断使能寄存器 (RCC_CIER)

Clock interrupt enable 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
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.	Res.	Res.	Res.	HSI48 RDYIE	LSECS SIE	Res.	PLLSAI 2RDYI E	PLLSAI 1RDYI E	PLL RDYIE	HSE RDYIE	HSI RDYIE	MSI RDYIE	LSE RDYIE	LSI RDYIE
					rw	rw		rw	rw	rw	rw	rw	rw	rw	rw

位 31:11 保留，必须保持复位值。

位 10 **HSI48RDYIE**：HSI48 就绪中断使能 (HSI48 ready interrupt enable)
由软件置 1 和清零，用于使能/禁止由内部 HSI48 振荡器所引起的中断。
0：禁止 HSI48 就绪中断
1：使能 HSI48 就绪中断

位 9 **LSECSSIE**：LSE 时钟安全系统中断使能 (LSE clock security system interrupt enable)
由软件置 1 和清零，用于使能/禁止由 LSE 上的时钟安全系统引起的中断。
0：禁止因 LSE 时钟故障而引起的时钟安全中断
1：使能因 LSE 时钟故障而引起的时钟安全中断

位 8 保留，必须保持复位值。

位 7 **PLLSAI2RDYIE**：PLLSAI2 就绪中断使能 (PLLSAI2 ready interrupt enable)
由软件置 1 和清零，用于使能/禁止由 PLLSAI2 锁定引起的中断。
0：禁止 PLLSAI2 锁定中断
1：使能 PLLSAI2 锁定中断

位 6 **PLLSAI1RDYIE**：PLLSAI1 就绪中断使能 (PLLSAI1 ready interrupt enable)
由软件置 1 和清零，用于使能/禁止由 PLSAI1L 锁定引起的中断。
0：禁止 PLLSAI1 锁定中断
1：使能 PLLSAI1 锁定中断

位 5 **PLLRDYIE**：PLL 就绪中断使能 (PLL ready interrupt enable)
由软件置 1 和清零，用于使能/禁止由 PLL 锁定引起的中断。
0：禁止 PLL 锁定中断
1：使能 PLL 锁定中断

位 4 **HSERDYIE**：HSE 就绪中断使能 (HSE 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 3 **HSIRDYIE**：HSI16 就绪中断使能 (HSI16 ready interrupt enable)
由软件置 1 和清零，用于使能/禁止由 HSI16 振荡器稳定所引起的中断。
0：禁止 HSI16 就绪中断
1：使能 HSI16 就绪中断

位 2 **MSIRDYIE**：MSI 就绪中断使能 (MSI ready interrupt enable)
由软件置 1 和清零，用于使能/禁止由 MSI 振荡器稳定所引起的中断。
0：禁止 MSI 就绪中断
1：使能 MSI 就绪中断

位 1 **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 0 **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

6.4.8 时钟中断标志寄存器 (RCC_CIFR)

Clock interrupt flag register

偏移地址：0x1C

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
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.	Res.	Res.	Res.	HSI48 RDYF	LSECS SF	CSSF	PLLSAI 2RDYF	PLLSAI 1RDYF	PLL RDYF	HSE RDYF	HSI RDYF	MSI RDYF	LSE RDYF	LSI RDYF
					r	r	r	r	r	r	r	r	r	r	r

位 31:11 保留，必须保持复位值。

位 10 **HSI48RDYF**：HSI48 就绪中断标志 (HSI48 ready interrupt flag)

当 HSI48 时钟稳定且 HSI48RDYIE 置 1 时由硬件置 1，以对 HSI48ON 置 1 作出响应（请参见*时钟恢复 RC 寄存器 (RCC_CRRCR)*）。

HSI48RDYC 位置 1 时由软件清零。

0：当前未因 HSI48 振荡器引起时钟就绪中断

1：因 HSI48 振荡器引起时钟就绪中断

位 9 **LSECSSF**：LSE 时钟安全系统中断标志 (LSE Clock security system interrupt flag)

当在 LSE 振荡器中检出故障时由硬件置 1。

LSECSSC 位置 1 时由软件清零。

0：当前未因 LSE 时钟故障而引起时钟安全中断

1：因 LSE 时钟故障而引起时钟安全中断

位 8 **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 7 **PLLSAI2RDYF**：PLLSAI2 就绪中断标志 (PLLSAI2 ready interrupt flag)

当 PLLSAI2 锁定并且 PLLSAI2RDYDIE 置 1 时由硬件置 1。

PLLSAI2RDYC 位置 1 时由软件清零。

0：当前未因 PLLSAI2 锁定而引起时钟就绪中断

1：因 PLLSAI2 锁定而引起时钟就绪中断

位 6 **PLLSAI1RDYF**：PLLSAI1 就绪中断标志 (PLLSAI1 ready interrupt flag)

当 PLLSAI1 锁定并且 PLLSAI1RDYDIE 置 1 时由硬件置 1。

PLLSAI1RDYC 位置 1 时由软件清零。

0：当前未因 PLLSAI1 锁定而引起时钟就绪中断

1：因 PLLSAI1 锁定而引起时钟就绪中断

位 5 **PLLRDYF**：PLL 就绪中断标志 (PLL ready interrupt flag)

当 PLL 锁定并且 PLLRDYDIE 置 1 时由硬件置 1。

PLLRDYC 位置 1 时由软件清零。

0：当前未因 PLL 锁定而引起时钟就绪中断

1：因 PLL 锁定而引起时钟就绪中断

位 4 **HSERDYF**：HSE 就绪中断标志 (HSE ready interrupt flag)

当 HSE 时钟稳定且 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 3 **HSIRDYF**：HSI16 就绪中断标志 (HSI16 ready interrupt flag)

当 HSI16 时钟稳定且 HSIRDYDIE 置 1 时由硬件置 1，以对 HSION 置 1 作出响应（请参见 *时钟控制寄存器 (RCC_CR)*）。当 HSION 未置 1，但外设通过时钟请求使能 HSI16 振荡器时，该位不会置 1，也不会生成中断。

HSIRDYC 位置 1 时由软件清零。

0：当前未因 HSI16 振荡器引起时钟就绪中断

1：因 HSI16 振荡器引起时钟就绪中断

位 2 **MSIRDYF**：MSI 就绪中断标志 (MSI ready interrupt flag)

当 MSI 时钟稳定且 MSIRDYDIE 置 1 时由硬件置 1。

MSIRDYC 位置 1 时由软件清零。

0：当前未因 MSI 振荡器引起时钟就绪中断

1：因 MSI 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 LSE 时钟稳定且 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 LSI 时钟稳定且 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

6.4.9 时钟中断清零寄存器 (RCC_CICR)

Clock interrupt clear 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
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.	Res.	Res.	Res.	HSI48RDYC	LSECSSC	CSSC	PLLSAI2RDYC	PLLSAI1RDYC	PLLRDYC	HSERDYC	HSIRDYC	MSIRDYC	LSERDYC	LSIRDYC
						w	w	w	w	w	w	w	w	w	w

位 31:11 保留，必须保持复位值。

位 10 **HSI48RDYC**：HSI48 振荡器就绪中断清零 (HSI48 oscillator ready interrupt clear)
该位由软件置 1，用于将 HSI48RDYF 标志清零。
0：无影响
1：将 HSI48RDYC 标志清零

位 9 **LSECSSC**：LSE 时钟安全系统中断清零 (LSE Clock security system interrupt clear)
此位由软件置 1，用于将 LSECSSF 标志清零。
0：无影响
1：将 LSECSSF 标志清零

位 8 **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：无影响
1：将 CSSF 标志清零

位 7 **PLLSAI2RDYC**：PLLSAI2 就绪中断清零 (PLLSAI2 ready interrupt clear)
此位由软件置 1，用于将 PLLSAI2RDYF 标志清零。
0：无影响
1：将 PLLSAI2RDYF 标志清零

位 6 **PLLSAI1RDYC**：PLLSAI1 就绪中断清零 (PLLSAI1 ready interrupt clear)
此位由软件置 1，用于将 PLLSAI1RDYF 标志清零。
0：无影响
1：将 PLLSAI1RDYF 标志清零

位 5 **PLLRDYC**：PLL 就绪中断清零 (PLL ready interrupt clear)
该位由软件置 1，用于将 PLLRDYF 标志清零。
0：无影响
1：将 PLLRDYF 标志清零

位 4 **HSERDYC**：HSE 就绪中断清零 (HSE ready interrupt clear)
该位由软件置 1，用于将 HSERDYF 标志清零。
0：无影响
1：将 HSERDYF 标志清零

位 3 **HSIRDYC**：HSI16 就绪中断清零 (HSI16 ready interrupt clear)
此位由软件置 1，用于将 HSIRDYF 标志清零。
0：无影响
1：将 HSIRDYF 标志清零

位 2 **MSIRDYC**：MSI 就绪中断清零 (MSI ready interrupt clear)
该位由软件置 1，用于将 MSIRDYF 标志清零。
0：无影响
1：已将 MSIRDYF 清零

位 1 **LSERDYC**：LSE 就绪中断清零 (LSE ready interrupt clear)
该位由软件置 1，用于将 LSERDYF 标志清零。
0：无影响
1：已将 LSERDYF 清零

位 0 **LSIRDYC**：LSI 就绪中断清零 (LSI ready interrupt clear)
该位由软件置 1，用于将 LSIRDYF 标志清零。
0：无影响
1：已将 LSIRDYF 清零

6.4.10 AHB1 外设复位寄存器 (RCC_AHB1RSTR)

AHB1 peripheral reset register

偏移地址：0x28

复位值：0x00000 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
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	GFXMMURST	DMA2DRST	TSCRST
													rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	CRCRST	Res.	Res.	Res.	FLASHRST	Res.	Res.	Res.	Res.	Res.	DMAMUX1RST	DMA2RST	DMA1RST
			rw				rw						rw	rw	rw

位 31:19 保留，必须保持复位值。

位 18 **GFXMMURST**：GFXMMU 复位 (GFXMMU reset)
由软件置 1 和清零。
0：无影响
1：复位 GFXMMU

位 17 **DMA2DRST**：DMA2D 复位 (DMA2D reset)
由软件置 1 和清零。
0：无影响
1：复位 DMA2D

位 16 **TSCRST**：触摸感应控制器复位 (Touch Sensing Controller reset)
由软件置 1 和清零。
0：无影响
1：复位 TSC

位 15:13 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：无影响
1：复位 CRC

位 11:9 保留，必须保持复位值。

位 8 **FLASHRST**：Flash 接口复位 (Flash memory interface reset)
由软件置 1 和清零。仅当 Flash 处于掉电模式时，才可激活此位。
0：无影响
1：复位 Flash 接口

位 7:3 保留，必须保持复位值。

位 2 **DMAMUX1RST**
由软件置 1 和清零。
0：无影响
1：复位 DMAMUX1

位 1 **DMA2RST**：DMA2 复位 (DMA2 reset)
由软件置 1 和清零。
0：无影响
1：复位 DMA2

位 0 **DMA1RST**：DMA1 复位 (DMA1 reset)
由软件置 1 和清零。
0：无影响
1：复位 DMA1

6.4.11 AHB2 外设复位寄存器 (RCC_AHB2RSTR)

AHB2 peripheral reset register

偏移地址：0x2C

复位值：0x00000 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
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SDMMC1RST	Res.	OSPIMRST	Res.	RNGRST	HASHRST	AESRST
													rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	DCMIRST	ADCRST	OTGFSRST	Res.	Res.	Res.	GPIOIRST	GPIOHRST	GPIOGRST	GPIOFRST	GPIOERST	GPIODRST	GPIOCRST	GPIOBRST	GPIOARST
	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 **SDMMC1RST**：SDMMC1 复位 (SDMMC1 reset)
由软件置 1 和清零。
0：无影响
1：复位 SDMMC1

位 21 保留，必须保持复位值。

位 20 **OSPIMRST**：OctoSPI IO 管理器复位 (OctoSPI IO manager reset)
由软件置 1 和清零。
0：无影响
1：复位 OctoSPI IO 管理器

位 18 **RNGRST**：随机数发生器复位 (Random number generator reset)
由软件置 1 和清零。
0：无影响
1：复位 RNG

位 17 **HASHRST**：散列复位 (Hash reset)
由软件置 1 和清零。
0：无影响
1：复位 HASH

位 16 **AESRST**：AES 硬件加速器复位 (AES hardware accelerator reset)
由软件置 1 和清零。
0：无影响
1：复位 AES

位 15 保留，必须保持复位值。

位 14 **DCMIRST**：数字摄像头接口复位 (Digital Camera Interface reset)
由软件置 1 和清零。
0：无影响
1：复位 DCMI 接口

位 13 **ADCRST**：ADC 复位 (ADC reset)
由软件置 1 和清零。
0：无影响
1：复位 ADC 接口

位 12 **OTGFSRST**：USB OTG FS 复位 (USB OTG FS reset)
由软件置 1 和清零。
0：无影响
1：复位 USB OTG FS

位 11:9 保留，必须保持复位值。

位 8 **GPIOIRST**：IO 端口 I 复位 (IO port I reset)
由软件置 1 和清零。
0：无影响
1：复位 IO 端口 I

位 7 **GPIOHRST**：IO 端口 H 复位 (IO port H reset)
由软件置 1 和清零。
0：无影响
1：复位 IO 端口 H

位 6 **GPIOGRST**：IO 端口 G 复位 (IO port G reset)
由软件置 1 和清零。
0：无影响
1：复位 IO 端口 G

位 5 **GPIOFRST**：IO 端口 F 复位 (IO port F reset)
由软件置 1 和清零。
0：无影响
1：复位 IO 端口 F

位 4 **GPIOERST**：IO 端口 E 复位 (IO port E reset)
由软件置 1 和清零。
0：无影响
1：复位 IO 端口 E

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

6.4.12 AHB3 外设复位寄存器 (RCC_AHB3RSTR)

AHB3 peripheral reset register

偏移地址：0x30

复位值：0x00000 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
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.	Res.	Res.	Res.	Res.	OSPI2RST	OSPI1RST	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FMCRST
						rw	rw								rw

位 31:10 保留，必须保持复位值。

位 9 **OSPI2RST**：OctoSPI2 存储器接口复位 (OctoSPI2 memory interface reset)
由软件置 1 和清零。
0：无影响
1：复位 OctoSPI2

位 8 **OSPI1RST**：OctoSPI1 存储器接口复位 (OctoSPI1 memory interface reset)
由软件置 1 和清零。
0：无影响
1：复位 OctoSPI1

位 7:1 保留，必须保持复位值。

位 0 **FMCRST**：灵活的存储控制器复位 (Flexible memory controller reset)
由软件置 1 和清零。
0：无影响
1：复位 FMC

6.4.13 APB1 外设复位寄存器 1 (RCC_APB1RSTR1)

APB1 peripheral reset register 1

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
LPTIM1RST	OPAMPRST	DAC1RST	PWRRST	Res.	Res.	CAN1RST	CRSRST	I2C3RST	I2C2RST	I2C1RST	UART5RST	UART4RST	USART3RST	USART2RST	Res.
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
SPI3RST	SPI2RST	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TIM7RST	TIM6RST	TIM5RST	TIM4RST	TIM3RST	TIM2RST
rw	rw									rw	rw	rw	rw	rw	rw

位 31 **LPTIM1RST**：低功耗定时器 1 复位 (Low Power Timer 1 reset)
由软件置 1 和清零。
0：无影响
1：复位 LPTIM1

位 30 **OPAMPRST**：OPAMP 接口复位 (OPAMP interface reset)
由软件置 1 和清零。
0：无影响
1：复位 OPAMP 接口

位 29 **DAC1RST**：DAC1 接口复位 (DAC1 interface reset)
由软件置 1 和清零。
0：无影响
1：复位 DAC1 接口

位 28 **PWRRST**：电源接口复位 (Power interface reset)
由软件置 1 和清零。
0：无影响
1：复位 PWR

位 27:26 保留，必须保持复位值。

位 25 **CAN1RST**：CAN1 复位 (CAN1 reset)
由软件置 1 和复位。
0：无影响
1：复位 CAN1

位 24 **CRSRST**：CRS 复位 (CRS reset)
由软件置 1 和清零。
0：无影响
1：复位 CRS

位 23 **I2C3RST**：I2C3 复位 (I2C3 reset)
由软件置 1 和复位。
0：无影响
1：复位 I2C3

位 22 **I2C2RST**：I2C2 复位 (I2C2 reset)
由软件置 1 和清零。
0：无影响
1：复位 I2C2

位 21 **I2C1RST**：I2C1 复位 (I2C1 reset)
由软件置 1 和清零。
0：无影响
1：复位 I2C1

位 20 **UART5RST**：UART5 复位 (UART5 reset)
由软件置 1 和清零。
0：无影响
1：复位 UART5

位 19 **UART4RST**：UART4 复位 (UART4 reset)
由软件置 1 和清零。
0：无影响
1：复位 UART4

位 18 **USART3RST**：USART3 复位 (USART3 reset)
由软件置 1 和清零。
0：无影响
1：复位 USART3

位 17 **USART2RST**：USART2 复位 (USART2 reset)
由软件置 1 和清零。
0：无影响
1：复位 USART2

位 16 保留，必须保持复位值。

位 15 **SPI3RST**：SPI3 复位 (SPI3 reset)
由软件置 1 和清零。
0：无影响
1：复位 SPI3

位 14 **SPI2RST**：SPI2 复位 (SPI2 reset)
由软件置 1 和清零。
0：无影响
1：复位 SPI2

位 13:6 保留，必须保持复位值。

位 5 **TIM7RST**：TIM7 定时器复位 (TIM7 timer reset)
由软件置 1 和清零。
0：无影响
1：复位 TIM7

位 4 **TIM6RST**：TIM6 定时器复位 (TIM6 timer reset)
由软件置 1 和清零。
0：无影响
1：复位 TIM6

位 3 **TIM5RST**：TIM5 定时器复位 (TIM5 timer reset)
由软件置 1 和清零。
0：无影响
1：复位 TIM5

位 2 **TIM4RST**：TIM4 定时器复位 (TIM4 timer reset)
由软件置 1 和清零。
0：无影响
1：复位 TIM4

位 1 **TIM3RST**：TIM3 定时器复位 (TIM3 timer reset)
由软件置 1 和清零。
0：无影响
1：复位 TIM3

位 0 **TIM2RST**：TIM2 定时器复位 (TIM2 timer reset)
由软件置 1 和清零。
0：无影响
1：复位 TIM2

6.4.14 APB1 外设复位寄存器 2 (RCC_APB1RSTR2)

APB1 peripheral reset register 2

偏移地址：0x3C

复位值：0x00000 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
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.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LPTIM2RST	Res.	Res.	Res.	I2C4RST	LPUART1RST
										rw				rw	rw

位 31:6 保留，必须保持复位值。

位 5 **LPTIM2RST**：低功耗定时器 2 复位 (Low-power timer 2 reset)
由软件置 1 和清零。
0：无影响
1：复位 LPTIM2

位 4:2 保留，必须保持复位值。

位 1 **I2C4RST**：I2C4 复位 (I2C4 reset)
由软件置 1 和清零。
0：无影响
1：复位 I2C4

位 0 **LPUART1RST**：低功耗 UART 1 复位 (Low-power UART 1 reset)
由软件置 1 和清零。
0：无影响
1：复位 LPUART1

6.4.15 APB2 外设复位寄存器 (RCC_APB2RSTR)

APB2 peripheral reset register

偏移地址：0x40

复位值：0x00000 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
Res.	Res.	Res.	Res.	DSIRST	LTDCRST	Res.	DFSDM1RST	Res.	SAI2RST	SAI1RST	Res.	Res.	TIM17RST	TIM16RST	TIM15RST
				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.	USART1RST	TIM8RST	SPI1RST	TIM1RST	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SYSCFGRST
	rw	rw	rw	rw											rw

位 31:28 保留，必须保持复位值。

位 27 **DSIRST**：DSI 复位 (DSI reset)
由软件置 1 和清零。
0：无影响
1：复位 DSI

位 26 **LTDCRST**：LCD-TFT 复位 (LCD-TFT reset)
由软件置 1 和清零。
0：无影响
1：复位 LCD-TFT

位 25 保留，必须保持复位值。

位 24 **DFSDM1RST**：用于 ∑Δ 调制器的数字滤波器 (DFSDM1) 复位 (Digital filters for sigma-delta modulators (DFSDM1) reset)
由软件置 1 和清零。
0：无影响
1：复位 DFSDM1

位 23 保留，必须保持复位值。

位 22 **SAI2RST**：串行音频接口 2 (SAI2) 复位 (Serial audio interface 2 (SAI2) reset)
由软件置 1 和清零。
0：无影响
1：复位 SAI2

位 21 **SAI1RST**：串行音频接口 1 (SAI1) 复位 (Serial audio interface 2 (SAI1) reset)
由软件置 1 和清零。
0：无影响
1：复位 SAI1

位 20:19 保留，必须保持复位值。

位 18 **TIM17RST**：TIM17 定时器复位 (TIM17 timer reset)
由软件置 1 和清零。
0：无影响
1：复位 TIM17 定时器

位 17 **TIM16RST**：TIM16 定时器复位 (TIM16 timer reset)
由软件置 1 和清零。
0：无影响
1：复位 TIM16 定时器

位 16 **TIM15RST**：TIM15 定时器复位 (TIM15 timer reset)
由软件置 1 和清零。
0：无影响
1：复位 TIM15 定时器

位 15 保留，必须保持复位值。

位 14 **USART1RST**：USART1 复位 (USART1 reset)
由软件置 1 和清零。
0：无影响
1：复位 USART1

位 13 **TIM8RST**：TIM8 定时器复位 (TIM8 timer reset)
由软件置 1 和清零。
0：无影响
1：复位 TIM8 定时器

位 12 **SPI1RST**：SPI1 复位 (SPI1 reset)
由软件置 1 和清零。
0：无影响
1：复位 SPI1

位 11 **TIM1RST**：TIM1 定时器复位 (TIM1 timer reset)
由软件置 1 和清零。
0：无影响
1：复位 TIM1 定时器

位 10:1 保留，必须保持复位值。

位 0 **SYSCFGRST**：SYSCFG + COMP + VREFBUF 复位 (SYSCFG + COMP + VREFBUF reset)
0：无影响
1：复位 SYSCFG + COMP + VREFBUF

6.4.16 AHB1 外设时钟使能寄存器 (RCC_AHB1ENR)

AHB1 peripheral clock enable register

偏移地址：0x48

复位值：0x0000 0100

访问：无等待周期，按字、半字和字节访问

注：　当外设时钟未激活时，不支持外设寄存器进行读写访问。

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.	GFXMMUEN	DMA2DEN	TSCEN
													rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	CRCEN	Res.	Res.	Res.	FLASHEN	Res.	Res.	Res.	Res.	Res.	DMAMUX1EN	DMA2EN	DMA1EN
			rw				rw						rw	rw	rw

位 31:19 保留，必须保持复位值。

位 18 **GFXMMUEN**：图形 MMU 时钟使能 (Graphic MMU clock enable)
由软件置 1 和复位。
0：禁止 GFXMMU 时钟
1：使能 GFXMMU 时钟

位 17 **DMA2DEN**：DMA2D 时钟使能 (DMA2D clock enable)
由软件置 1 和清零。
0：禁止 DMA2D 时钟
1：使能 DMA2D 时钟

位 16 **TSCEN**：触摸感应控制器时钟使能 (Touch Sensing Controller clock enable)
由软件置 1 和清零。
0：禁止 TSC 时钟
1：使能 TSC 时钟

位 15:13 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 11:9 保留，必须保持复位值。

位 8 **FLASHEN**：Flash 接口时钟使能 (Flash memory interface clock enable)
由软件置 1 和清零。仅当 Flash 处于掉电模式时，才可禁止此位。
0：禁止 Flash 接口时钟
1：使能 Flash 接口时钟

位 7:3 保留，必须保持复位值。

位 2 **DMAMUX1EN**：DMAMUX1 时钟使能 (DMAMUX1 clock enable)
由软件置 1 和复位。
0：禁止 DMAMUX1 时钟
1：使能 DMAMUX1 时钟

位 1 **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 0 **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

6.4.17 AHB2 外设时钟使能寄存器 (RCC_AHB2ENR)

AHB2 peripheral clock enable register

偏移地址：0x4C

复位值：0x0000 0000

访问：无等待周期，按字、半字和字节访问

注： *当外设时钟未激活时，不支持外设寄存器进行读写访问。*

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.	SDMMC1EN	Res.	OSPIMEN	Res.	RNGEN	HASHEN	AESEN
									rw		rw		rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	DCMIEN	ADCEN	OTGFSEN	Res.	Res.	Res.	GPIOIEN	GPIOHEN	GPIOGEN	GPIOFEN	GPIOEEN	GPIODEN	GPIOCEN	GPIOBEN	GPIOAEN
	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 **SDMMC1EN**：SDMMC1 时钟使能 (SDMMC1 clock enable)
由软件置 1 和清零。
0：禁止 SDMMC1 时钟
1：使能 SDMMC1 时钟

位 21 保留，必须保持复位值。

位 20 **OSPIMEN**：OctoSPI IO 管理器时钟使能 (OctoSPI IO manager clock enable)
由软件置 1 和清零。
0：禁止 OctoSPI IO 管理器时钟
1：使能 OctoSPI IO 管理器时钟

位 19 保留，必须保持复位值。

位 18 **RNGEN**：随机数发生器时钟使能 (Random Number Generator clock enable)
由软件置 1 和清零。
0：禁止随机数发生器时钟
1：使能随机数发生器时钟

位 17 **HASHEN**：HASH 时钟使能 (HASH clock enable)
由软件置 1 和清零。
0：禁止 HASH 时钟
1：使能 HASH 时钟

位 16 **AESEN**：AES 加速器时钟使能 (AES accelerator clock enable)
由软件置 1 和清零。
0：禁止 AES 时钟
1：使能 AES 时钟

位 15 保留，必须保持复位值。

位 14 **DCMIEN**：DCMI 时钟使能 (DCMI clock enable)

由软件置 1 和清零。

0：禁止 DCMI 时钟

1：使能 DCMI 时钟

位 13 **ADCEN**：ADC 时钟使能 (ADC clock enable)

由软件置 1 和清零。

0：禁止 ADC 时钟

1：使能 ADC 时钟

位 12 **OTGFSEN**：OTG 全速时钟使能 (OTG full speed clock enable)

由软件置 1 和清零。

0：禁止 USB OTG 全速时钟

1：使能 USB OTG 全速时钟

位 11:9 保留，必须保持复位值。

位 8 **GPIOIEN**：IO 端口 I 时钟使能 (IO port I clock enable)

由软件置 1 和清零。

0：禁止 IO 端口 I 时钟

1：使能 IO 端口 I 时钟

位 7 **GPIOHEN**：IO 端口 H 时钟使能 (IO port H clock enable)

由软件置 1 和清零。

0：禁止 IO 端口 H 时钟

1：使能 IO 端口 H 时钟

位 6 **GPIOGEN**：IO 端口 G 时钟使能 (IO port G clock enable)

由软件置 1 和清零。

0：禁止 IO 端口 G 时钟

1：使能 IO 端口 G 时钟

位 5 **GPIOFEN**：IO 端口 F 时钟使能 (IO port F clock enable)

由软件置 1 和清零。

0：禁止 IO 端口 F 时钟

1：使能 IO 端口 F 时钟

位 4 **GPIOEEN**：IO 端口 E 时钟使能 (IO port E clock enable)

由软件置 1 和清零。

0：禁止 IO 端口 E 时钟

1：使能 IO 端口 E 时钟

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

6.4.18 AHB3 外设时钟使能寄存器 (RCC_AHB3ENR)

AHB3 peripheral clock enable register

偏移地址：0x50

复位值：0x00000 0000

访问：无等待周期，按字、半字和字节访问

注：　当外设时钟未激活时，不支持外设寄存器进行读写访问。

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.	Res.	Res.	Res.	Res.	OSPI2EN	OSPI1EN	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FMCEN
						rw	rw								rw

位 31:10 保留，必须保持复位值。

位 9 **OSPI2EN**：OctoSPI2 存储器接口时钟使能 (OctoSPI2 memory interface clock enable)
由软件置 1 和清零。
0：禁止 OctoSPI2 时钟
1：使能 OctoSPI2 时钟

位 8 **OSPI1EN**：OctoSPI1 存储器接口时钟使能 (OctoSPI1 memory interface clock enable)
由软件置 1 和清零。
0：禁止 OctoSPI1 时钟
1：使能 OctoSPI1 时钟

位 7:1 保留，必须保持复位值。

位 0 **FMCEN**：灵活的存储控制器时钟使能 (Flexible memory controller clock enable)
由软件置 1 和清零。
0：禁止 FMC 时钟
1：使能 FMC 时钟

6.4.19 APB1 外设时钟使能寄存器 1 (RCC_APB1ENR1)

APB1 peripheral clock enable register 1

地址：0x58

复位值：0x0000 0400

访问：无等待周期，按字、半字和字节访问

注： *当外设时钟未激活时，不支持外设寄存器进行读写访问。*

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LPTIM1EN	OPAMPEN	DAC1EN	PWREN	Res.	Res.	CAN1EN	CRSEN	I2C3EN	I2C2EN	I2C1EN	UART5EN	UART4EN	USART3EN	USART2EN	Res.
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
SPI3EN	SPI2EN	Res.	Res.	WWDGEN	RTCAPBEN	Res.	Res.	Res.	Res.	TIM7EN	TIM6EN	TIM5EN	TIM4EN	TIM3EN	TIM2EN
rw	rw			rs	rw					rw	rw	rw	rw	rw	rw

位 31 **LPTIM1EN**：低功耗定时器 1 时钟使能 (Low power timer 1 clock enable)
由软件置 1 和清零。
0：禁止 LPTIM1 时钟
1：使能 LPTIM1 时钟

位 30 **OPAMPEN**：OPAMP 接口时钟使能 (OPAMP interface clock enable)
由软件置 1 和清零。
0：禁止 OPAMP 接口时钟
1：使能 OPAMP 接口时钟

位 29 **DAC1EN**：DAC1 接口时钟使能 (DAC1 interface clock enable)
由软件置 1 和清零。
0：禁止 DAC1 接口时钟
1：使能 DAC1 接口时钟

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 27:26 保留，必须保持复位值。

位 25 **CAN1EN**：CAN1 时钟使能 (CAN1 clock enable)
由软件置 1 和清零。
0：禁止 CAN1 时钟
1：使能 CAN1 时钟

位 24 **CRSEN**：时钟恢复系统时钟使能 (Clock Recovery System clock enable)
由软件置 1 和清零。
0：禁止 CRS 时钟
1：使能 CRS 时钟

位 23 **I2C3EN**：I2C3 时钟使能 (I2C3 clock enable)
由软件置 1 和清零。
0：禁止 I2C3 时钟
1：使能 I2C3 时钟

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 **UART5EN**：UART5 时钟使能 (UART5 clock enable)
由软件置 1 和清零。
0：禁止 UART5 时钟
1：使能 UART5 时钟

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 **SPI3EN**：SPI3 时钟使能 (SPI3 clock enable)
由软件置 1 和清零。
0：禁止 SPI3 时钟
1：使能 SPI3 时钟

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，用于使能窗口看门狗时钟。通过硬件系统复位进行复位。如果 WWDG_SW 选项位复位，该位也可由硬件置 1。
0：禁止窗口看门狗时钟
1：使能窗口看门狗时钟

位 10 **RTCAPBEN**：RTC APB 时钟使能 (RTC APB clock enable)
由软件置 1 和清零。
0：禁止 RTC APB 时钟
1：使能 RTC APB 时钟

位 9:6 保留，必须保持复位值。

位 5 **TIM7EN**：TIM7 定时器时钟使能 (TIM7 timer clock enable)
由软件置 1 和清零。
0：禁止 TIM7 时钟
1：使能 TIM7 时钟

位 4 **TIM6EN**：TIM6 定时器时钟使能 (TIM6 timer clock enable)
由软件置 1 和清零。
0：禁止 TIM6 时钟
1：使能 TIM6 时钟

位 3 **TIM5EN**：TIM5 定时器时钟使能 (TIM5 timer clock enable)
由软件置 1 和清零。
0：禁止 TIM5 时钟
1：使能 TIM5 时钟

位 2 **TIM4EN**：TIM4 定时器时钟使能 (TIM4 timer clock enable)
由软件置 1 和清零。
0：禁止 TIM4 时钟
1：使能 TIM4 时钟

位 1 **TIM3EN**：TIM3 定时器时钟使能 (TIM3 timer clock enable)
由软件置 1 和清零。
0：禁止 TIM3 时钟
1：使能 TIM3 时钟

位 0 **TIM2EN**：TIM2 定时器时钟使能 (TIM2 timer clock enable)
由软件置 1 和清零。
0：禁止 TIM2 时钟
1：使能 TIM2 时钟

6.4.20 APB1 外设时钟使能寄存器 2 (RCC_APB1ENR2)

APB1 peripheral clock enable register 2

偏移地址：0x5C

复位值：0x00000 0000

访问：无等待周期，按字、半字和字节访问

注： *当外设时钟未激活时，不支持外设寄存器进行读写访问。*

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.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LPTIM2EN	Res.	Res.	Res.	I2C4EN	LPUART1EN
														rw	rw

位 31:6 保留，必须保持复位值。

位 5 **LPTIM2EN**：低功耗定时器 2 时钟使能 (Low power timer 2 clock enable)
由软件置 1 和清零。
0：禁止 LPTIM2 时钟
1：使能 LPTIM2 时钟

位 4:2 保留，必须保持复位值。

位 1 **I2C4EN**：I2C4 时钟使能 (I2C4 clock enable)
由软件置 1 和清零。
0：禁止 I2C4 时钟
1：使能 I2C4 时钟

位 0 **LPUART1EN**：低功耗 UART 1 时钟使能 (Low power UART 1 clock enable)
由软件置 1 和清零。
0：禁止 LPUART1 时钟
1：使能 LPUART1 时钟

6.4.21 APB2 外设时钟使能寄存器 (RCC_APB2ENR)

APB2 peripheral clock enable register

地址：0x60

复位值：0x0000 0000

访问：按字、半字和字节访问。

注： *当外设时钟未激活时，不支持外设寄存器进行读写访问。*

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	DSIEN	LTDCEN	Res.	DFSDM1EN	Res.	SAI2EN	SAI1EN	Res.	Res.	TIM17EN	TIM16EN	TIM15EN
				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.	USART1EN	TIM8EN	SPI1EN	TIM1EN	Res.	Res.	Res.	FWEN	Res.	Res.	Res.	Res.	Res.	Res.	SYSCFGEN
	rw	rw	rw	rw				rs							rw

位 31:28 保留，必须保持复位值。

位 27 **DSIEN**：DSI 时钟使能 (DSI clock enable)
由软件置 1 和清零。
0：禁止 DSI 时钟
1：使能 DSI 时钟

位 26 **LTDCEN**：LCD-TFT 时钟使能 (LCD-TFT clock enable)
由软件置 1 和清零。
0：禁止 LTDC 时钟
1：使能 LTDC 时钟

位 25 保留，必须保持复位值。

位 24 **DFSDM1EN**：DFSDM1 定时器时钟使能 (DFSDM1 timer clock enable)
由软件置 1 和清零。
0：禁止 DFSDM1 时钟
1：使能 DFSDM1 时钟

位 23 保留，必须保持复位值。

位 22 **SAI2EN**：SAI2 时钟使能 (SAI2 clock enable)
由软件置 1 和清零。
0：禁止 SAI2 时钟
1：使能 SAI2 时钟

位 21 **SAI1EN**：SAI1 时钟使能 (SAI1 clock enable)
由软件置 1 和清零。
0：禁止 SAI1 时钟
1：使能 SAI1 时钟

位 20:19 保留，必须保持复位值。

位 18 **TIM17EN**：TIM17 定时器时钟使能 (TIM17 timer clock enable)
由软件置 1 和清零。
0：禁止 TIM17 定时器时钟
1：使能 TIM17 定时器时钟

位 17 **TIM16EN**：TIM16 定时器时钟使能 (TIM16 timer clock enable)
由软件置 1 和清零。
0：禁止 TIM16 定时器时钟
1：使能 TIM16 定时器时钟

位 16 **TIM15EN**：TIM15 定时器时钟使能 (TIM15 timer clock enable)
由软件置 1 和清零。
0：禁止 TIM15 定时器时钟
1：使能 TIM15 定时器时钟

位 15 保留，必须保持复位值。

位 14 **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 13 **TIM8EN**：TIM8 定时器时钟使能 (TIM8 timer clock enable)
由软件置 1 和清零。
0：禁止 TIM8 定时器时钟
1：使能 TIM8 定时器时钟

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 **TIM1EN**：TIM1 定时器时钟使能 (TIM1 timer clock enable)
由软件置 1 和清零。
0：禁止 TIM1 定时器时钟
1：使能 TIM1 定时器时钟

位 10:8 保留，必须保持复位值。

位 7 **FWEN**：防火墙时钟使能 (Firewall clock enable)

由软件置 1，由硬件复位。软件只能写入 1。写入 0 不起作用。

0：禁止防火墙时钟

1：使能防火墙时钟

位 6:1 保留，必须保持复位值。

位 0 **SYSCFGEN**：SYSCFG + COMP + VREFBUF 时钟使能 (SYSCFG + COMP + VREFBUF clock enable)

由软件置 1 和清零。

0：禁止 SYSCFG + COMP + VREFBUF 时钟

1：使能 SYSCFG + COMP + VREFBUF 时钟

6.4.22 睡眠和停止模式下的 AHB1 外设时钟使能寄存器 (RCC_AHB1SMENR)

AHB1 peripheral clocks enable in Sleep and Stop modes register

偏移地址：0x68

复位值：0x0007 1307

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
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	GFXMMUSMEN	DMA2DSMEN	TSCSMEN
													rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	CRCSMEN	Res.	Res.	SRAM1SMEN	FLASHSMEN	Res.	Res.	Res.	Res.	Res.	DMAMUX1SMEN	DMA2SMEN	DMA1SMEN
			rw			rw	rw							rw	rw

位 31:19 保留，必须保持复位值。

位 18 **GFXMMUSMEN**：睡眠和停止模式期间的 GFXMMU 时钟使能 (GFXMMU clock enable during Sleep and Stop modes)

由软件置 1 和清零

0：睡眠和停止模式期间 GFXMMU 时钟由时钟门控(1) 禁止

1：睡眠和停止模式期间 GFXMMU 时钟由时钟门控(1) 使能

位 17 **DMA2DSMEN**：睡眠和停止模式期间的 DMA2D 时钟使能 (DMA2D clock enable during Sleep and Stop modes)

由软件置 1 和清零

0：睡眠和停止模式期间 DMA2D 时钟由时钟门控(1) 禁止

1：睡眠和停止模式期间 DMA2D 时钟由时钟门控(1) 使能

位 16 **TSCSMEN**：睡眠和停止模式期间的触摸感应控制器时钟使能 (Touch Sensing Controller clocks enable during Sleep and Stop modes)

由软件置 1 和清零。

0：睡眠和停止模式期间 TSC 时钟由时钟门控(1) 禁止

1：睡眠和停止模式期间 TSC 时钟由时钟门控(1) 使能

位 15:13 保留，必须保持复位值。

位 12 **CRCSMEN**：睡眠和停止模式期间的 CRC 时钟使能 (CRC clocks enable during Sleep and Stop modes)

由软件置 1 和清零。

0：睡眠和停止模式期间 CRC 时钟由时钟门控(1) 禁止

1：睡眠和停止模式期间 CRC 时钟由时钟门控(1) 使能

位 11:10 保留，必须保持复位值。

位 9 **SRAM1SMEN**：睡眠和停止模式期间的 SRAM1 接口时钟使能 (SRAM1 interface clocks enable during Sleep and Stop modes)

由软件置 1 和清零。

0：睡眠和停止模式期间 SRAM1 接口时钟由时钟门控(1) 禁止

1：睡眠和停止模式期间 SRAM1 接口时钟由时钟门控(1) 使能

位 8 **FLASHSMEN**：睡眠和停止模式期间的 Flash 接口时钟使能 (Flash memory interface clocks enable during Sleep and Stop modes)

由软件置 1 和清零。

0：睡眠和停止模式期间 Flash 接口时钟由时钟门控(1) 禁止

1：睡眠和停止模式期间 Flash 接口时钟由时钟门控(1) 使能

位 7:3 保留，必须保持复位值。

位 2 **DMAMUX1SMEN**：睡眠和停止模式期间的 DMAMUX1 时钟使能 (DMAMUX1 clock enable during Sleep and Stop modes)

由软件置 1 和清零。

0：睡眠和停止模式期间 DMAMUX1 时钟由时钟门控(1) 禁止

1：睡眠和停止模式期间 DMAMUX1 时钟由时钟门控(1) 使能

位 1 **DMA2SMEN**：睡眠和停止模式期间的 DMA2 时钟使能 (DMA2 clocks enable during Sleep and Stop modes)

睡眠模式期间由软件置 1 和清零。

0：睡眠和停止模式期间 DMA2 时钟由时钟门控(1) 禁止

1：睡眠和停止模式期间 DMA2 时钟由时钟门控(1) 使能

位 0 **DMA1SMEN**：睡眠和停止模式期间的 DMA1 时钟使能 (DMA1 clocks enable during Sleep and Stop modes)

由软件置 1 和清零。

0：睡眠和停止模式期间 DMA1 时钟由时钟门控(1) 禁止

1：睡眠和停止模式期间 DMA1 时钟由时钟门控(1) 使能

1. 该寄存器仅配置时钟门控，而不是时钟源本身。大多数外设的时钟由一个时钟（AHB 或 APB 时钟）提供，该时钟在停止模式下始终处于禁止状态。因此，在停止模式下将该位置 1 不起作用。

6.4.23 睡眠和停止模式下的 AHB2 外设时钟使能寄存器 (RCC_AHB2SMENR)

AHB2 peripheral clocks enable in Sleep and Stop modes register

偏移地址：0x6C

复位值：0x0057 77FF

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
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SDMMC1SMEN	Res.	OSPIMSMEN	Res.	RNGSMEN	HASHSMEN	AESSMEN
									rw				rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	DCMISMEN	ADCSMEN	OTGFSSMEN	Res.	SRAM3SMEN	SRAM2SMEN	GPIOISMEN	GPIOHSMEN	GPIOGSMEN	GPIOFSMEN	GPIOESMEN	GPIODSMEN	GPIOCSMEN	GPIOBSMEN	GPIOASMEN
	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 **SDMMC1SMEN**：睡眠和停止模式期间的 SDMMC1 时钟使能 (SDMMC1 clocks enable during Sleep and Stop modes)

由软件置 1 和清零。

0：睡眠和停止模式期间 SDMMC1 时钟由时钟门控(1) 禁止

1：睡眠和停止模式期间 SDMMC1 时钟由时钟门控(1) 使能

位 21 保留，必须保持复位值。

位 20 **OSPIMSMEN**：睡眠和停止模式期间的 OctoSPI IO 管理器时钟使能 (OctoSPI IO manager clocks enable during Sleep and Stop modes)

由软件置 1 和清零。

0：睡眠和停止模式期间 OCTOSPIM 时钟由时钟门控(1) 禁止

1：睡眠和停止模式期间 OCTOSPIM 时钟由时钟门控(1) 使能

位 19 保留，必须保持复位值。

位 18 **RNGSMEN**：睡眠和停止模式期间的随机数发生器时钟使能 (Random Number Generator clocks enable during Sleep and Stop modes)

由软件置 1 和清零。

0：睡眠和停止模式期间随机数发生器时钟由时钟门控禁止

1：睡眠和停止模式期间随机数发生器时钟由时钟门控使能

位 17 **HASHSMEN**：睡眠和停止模式期间的 HASH 时钟使能 (HASH clock enable during Sleep and Stop modes)

由软件置 1 和清零。

0：睡眠和停止模式期间 HASH 时钟由时钟门控(2) 禁止

1：睡眠和停止模式期间 HASH 时钟由时钟门控(1) 使能

位 16 **AESSMEN**：睡眠和停止模式期间的 AES 加速器时钟使能 (AES accelerator clocks enable during Sleep and Stop modes)

由软件置 1 和清零。

0：睡眠和停止模式期间 AES 时钟由时钟门控(1) 禁止

1：睡眠和停止模式期间 AES 时钟由时钟门控(1) 使能

位 15 保留，必须保持复位值。

位 14 **DCMISMEN**：睡眠和停止模式期间的 DCMI 时钟使能 (DCMI clock enable during Sleep and Stop modes)

由软件置 1 和清零。

0：睡眠和停止模式期间 DCMI 时钟由时钟门控(1) 禁止

1：睡眠和停止模式期间 DCMI 时钟由时钟门控(1) 使能

位 13 **ADCSMEN**：睡眠和停止模式期间的 ADC 时钟使能 (ADC clocks enable during Sleep and Stop modes)

由软件置 1 和清零。

0：睡眠和停止模式期间 ADC 时钟由时钟门控(1) 禁止

1：睡眠和停止模式期间 ADC 时钟由时钟门控(1) 使能

位 12 **OTGFSSMEN**：睡眠和停止模式期间的 OTG 全速时钟使能 (OTG full speed clocks enable during Sleep and Stop modes)

由软件置 1 和清零。

0：睡眠和停止模式期间 USB OTG 全速时钟由时钟门控(1) 禁止

1：睡眠和停止模式期间 USB OTG 全速时钟由时钟门控(1) 使能

位 11 保留，必须保持复位值。

位 10 **SRAM3SMEN**：睡眠和停止模式期间的 SRAM2 接口时钟使能 (SRAM2 interface clocks enable during Sleep and Stop modes)

由软件置 1 和清零。

0：睡眠和停止模式期间 SRAM3 接口时钟由时钟门控(1) 禁止

1：睡眠和停止模式期间 SRAM32 接口时钟由时钟门控(1) 使能

位 9 **SRAM2SMEN**：睡眠和停止模式期间的 SRAM2 接口时钟使能 (SRAM2 interface clocks enable during Sleep and Stop modes)

由软件置 1 和清零。

0：睡眠和停止模式期间 SRAM2 接口时钟由时钟门控(1) 禁止

1：睡眠和停止模式期间 SRAM2 接口时钟由时钟门控(1) 使能

位 8 **GPIOISMEN**：睡眠和停止模式期间的 IO 端口 I 时钟使能 (IO port I clocks enable during Sleep and Stop modes)

由软件置 1 和清零。

0：睡眠和停止模式期间 IO 端口 I 时钟由时钟门控(1) 禁止

1：睡眠和停止模式期间 IO 端口 I 时钟由时钟门控(1) 使能

位 7 **GPIOHSMEN**：睡眠和停止模式期间的 IO 端口 H 时钟使能 (IO port H clocks enable during Sleep and Stop modes)

由软件置 1 和清零。

0：睡眠和停止模式期间 IO 端口 H 时钟由时钟门控(1) 禁止

1：睡眠和停止模式期间 IO 端口 H 时钟由时钟门控(1) 使能

位 6 **GPIOGSMEN**：睡眠和停止模式期间的 IO 端口 G 时钟使能 (IO port G clocks enable during Sleep and Stop modes)

由软件置 1 和清零。

0：睡眠和停止模式期间 IO 端口 G 时钟由时钟门控(1) 禁止

1：睡眠和停止模式期间 IO 端口 G 时钟由时钟门控(1) 使能

位 5 **GPIOFSMEN**：睡眠和停止模式期间的 IO 端口 F 时钟使能 (IO port F clocks enable during Sleep and Stop modes)

由软件置 1 和清零。

0：睡眠和停止模式期间 IO 端口 F 时钟由时钟门控(1) 禁止

1：睡眠和停止模式期间 IO 端口 F 时钟由时钟门控(1) 使能

位 4 **GPIOESMEN**：睡眠和停止模式期间的 IO 端口 E 时钟使能 (IO port E clocks enable during Sleep and Stop modes)

由软件置 1 和清零。

0：睡眠和停止模式期间 IO 端口 E 时钟由时钟门控(1) 禁止

1：睡眠和停止模式期间 IO 端口 E 时钟由时钟门控(1) 使能

位 3 **GPIODSMEN**：睡眠和停止模式期间的 IO 端口 D 时钟使能 (IO port D clocks enable during Sleep and Stop modes)

由软件置 1 和清零。

0：睡眠和停止模式期间 IO 端口 D 时钟由时钟门控(1) 禁止

1：睡眠和停止模式期间 IO 端口 D 时钟由时钟门控(1) 使能

位 2 **GPIOCSMEN**：睡眠和停止模式期间的 IO 端口 C 时钟使能 (IO port C clocks enable during Sleep and Stop modes)

由软件置 1 和清零。

0：睡眠和停止模式期间 IO 端口 C 时钟由时钟门控(1) 禁止

1：睡眠和停止模式期间 IO 端口 C 时钟由时钟门控(1) 使能

位 1 **GPIOBSMEN**：睡眠和停止模式期间的 IO 端口 B 时钟使能 (IO port B clocks enable during Sleep and Stop modes)

由软件置 1 和清零。

0：睡眠和停止模式期间 IO 端口 B 时钟由时钟门控(1) 禁止

1：睡眠和停止模式期间 IO 端口 B 时钟由时钟门控(1) 使能

位 0 **GPIOASMEN**：睡眠和停止模式期间的 IO 端口 A 时钟使能 (IO port A clocks enable during Sleep and Stop modes)

由软件置 1 和清零。

0：睡眠和停止模式期间 IO 端口 A 时钟由时钟门控(1) 禁止

1：睡眠和停止模式期间 IO 端口 A 时钟由时钟门控(1) 使能

1. 该寄存器仅配置时钟门控，而不是时钟源本身。大多数外设的时钟由一个时钟（AHB 或 APB 时钟）提供，该时钟在停止模式下始终处于禁止状态。因此，在停止模式下将该位置 1 不起作用。
2. 该寄存器仅配置时钟门控，而不是时钟源本身。大多数外设的时钟由一个时钟（AHB 或 APB 时钟）提供，该时钟在停止模式下始终处于禁止状态。因此，在停止模式下将该位置 1 不起作用。

6.4.24 睡眠和停止模式下的 AHB3 外设时钟使能寄存器 (RCC_AHB3SMENR)

AHB3 peripheral clocks enable in Sleep and Stop modes register

偏移地址：0x70

复位值：0x00000 0301

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
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.	Res.	Res.	Res.	Res.	OCTOSPI2	OSPI1SMEN	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FMCSMEN
						rw	rw								rw

位 31:10 保留，必须保持复位值。

位 9 **OCTOSPI2**：睡眠和停止模式期间的 OctoSPI2 存储器接口时钟使能 (OctoSPI2 memory interface clocks enable during Sleep and Stop modes)

由软件置 1 和清零。

0：睡眠和停止模式期间 OctoSPI2 时钟由时钟门控(1) 禁止

1：睡眠和停止模式期间 OctoSPI2 时钟由时钟门控(1) 使能

位 8 **OSPI1SMEN**：睡眠和停止模式期间的 OctoSPI1 存储器接口时钟使能 (OctoSPI1 memory interface clocks enable during Sleep and Stop modes)

由软件置 1 和清零。

0：睡眠和停止模式期间 OctoSPI1 时钟由时钟门控(1) 禁止

1：睡眠和停止模式期间 OctoSPI1 时钟由时钟门控(1) 使能

位 7:1 保留，必须保持复位值。

位 0 **FMCSMEN**：睡眠和停止模式期间的灵活存储控制器时钟使能 (Flexible memory controller clocks enable during Sleep and Stop modes)

由软件置 1 和清零。

0：睡眠和停止模式期间 FMC 时钟由时钟门控(1) 禁止

1：睡眠和停止模式期间 FMC 时钟由时钟门控(1) 使能

1. 该寄存器仅配置时钟门控，而不是时钟源本身。大多数外设的时钟由一个时钟（AHB 或 APB 时钟）提供，该时钟在停止模式下始终处于禁止状态。因此，在停止模式下将该位置 1 不起作用。

6.4.25 睡眠和停止模式下的 APB1 外设时钟使能寄存器 1 (RCC_APB1SMENR1)

APB1 peripheral clocks enable in Sleep and Stop modes register 1

地址：0x78

复位值：0xF3FECC3F

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
LPTIM1SMEN	OPAMPSMEN	DAC1SMEN	PWRSMEN	Res.	Res.	CAN1SMEN	CRSSMEN	I2C3SMEN	I2C2SMEN	I2C1SMEN	UART5SMEN	UART4SMEN	USART3SMEN	USART2SMEN	Res.
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
SPI3SMEN	SPI2SMEN	Res.	Res.	WWDGSMEN	RTCAPBSMEN	Res.	Res.	Res.	Res.	TIM7SMEN	TIM6SMEN	TIM5SMEN	TIM4SMEN	TIM3SMEN	TIM2SMEN
rw	rw			rw	rw					rw	rw	rw	rw	rw	rw

位 31 **LPTIM1SMEN**：睡眠和停止模式期间的低功耗定时器 1 时钟使能 (Low power timer 1 clocks enable during Sleep and Stop modes)

由软件置 1 和清零。

0：睡眠和停止模式期间 LPTIM1 时钟由时钟门控(1) 禁止

1：睡眠和停止模式期间 LPTIM1 时钟由时钟门控(1) 使能

位 30 **OPAMPSMEN**：睡眠和停止模式期间的 OPAMP 接口时钟使能 (OPAMP interface clocks enable during Sleep and Stop modes)

由软件置 1 和清零。

0：睡眠和停止模式期间 OPAMP 接口时钟由时钟门控(1) 禁止

1：睡眠和停止模式期间 OPAMP 接口时钟由时钟门控(1) 使能

位 29 **DAC1SMEN**：睡眠和停止模式期间的 DAC1 接口时钟使能 (DAC1 interface clocks enable during Sleep and Stop modes)

由软件置 1 和清零。

0：睡眠和停止模式期间 DAC1 接口时钟由时钟门控(1) 禁止

1：睡眠和停止模式期间 DAC1 接口时钟由时钟门控(1) 使能

位 28 **PWRSMEN**：睡眠和停止模式期间的电源接口时钟使能 (Power interface clocks enable during Sleep and Stop modes)

由软件置 1 和清零。

0：睡眠和停止模式期间电源接口时钟由时钟门控(1) 禁止

1：睡眠和停止模式期间电源接口时钟由时钟门控(1) 使能

位 27 保留，必须保持复位值。

位 25 **CAN1SMEN**：睡眠和停止模式期间的 CAN1 时钟使能 (CAN1 clocks enable during Sleep and Stop modes)

由软件置 1 和清零。

0：睡眠和停止模式期间 CAN1 时钟由时钟门控(1) 禁止

1：睡眠和停止模式期间 CAN1 时钟由时钟门控(1) 使能

位 24 **CRSSMEN**：睡眠和停止模式期间的 CRS 时钟使能 (CRS clock enable during Sleep and Stop modes)

由软件置 1 和清零。

0：睡眠和停止模式期间 CRS 时钟由时钟门控(1) 禁止

1：睡眠和停止模式期间 CRS 时钟由时钟门控(1) 使能

位 23 **I2C3SMEN**：睡眠和停止模式期间的 I2C3 时钟使能 (I2C3 clocks enable during Sleep and Stop modes)

由软件置 1 和清零。

0：睡眠和停止模式期间 I2C3 时钟由时钟门控(1) 禁止

1：睡眠和停止模式期间 I2C3 时钟由时钟门控(1) 使能

位 22 **I2C2SMEN**：睡眠和停止模式期间的 I2C2 时钟使能 (I2C2 clocks enable during Sleep and Stop modes)

由软件置 1 和清零。

0：睡眠和停止模式期间 I2C2 时钟由时钟门控(1) 禁止

1：睡眠和停止模式期间 I2C2 时钟由时钟门控(1) 使能

位 21 **I2C1SMEN**：睡眠和停止模式期间的 I2C1 时钟使能 (I2C1 clocks enable during Sleep and Stop modes)

由软件置 1 和清零。

0：睡眠和停止模式期间 I2C1 时钟由时钟门控(1) 禁止

1：睡眠和停止模式期间 I2C1 时钟由时钟门控(1) 使能

位 20 **UART5SMEN**：睡眠和停止模式期间的 UART5 时钟使能 (UART5 clocks enable during Sleep and Stop modes)

由软件置 1 和清零。

0：睡眠和停止模式期间 UART5 时钟由时钟门控(1) 禁止

1：睡眠和停止模式期间 UART5 时钟由时钟门控(1) 使能

位 19 **UART4SMEN**：睡眠和停止模式期间的 UART4 时钟使能 (UART4 clocks enable during Sleep and Stop modes)

由软件置 1 和清零。

0：睡眠和停止模式期间 UART4 时钟由时钟门控(1) 禁止

1：睡眠和停止模式期间 UART4 时钟由时钟门控(1) 使能

位 18 **USART3SMEN**：睡眠和停止模式期间的 USART3 时钟使能 (USART3 clocks enable during Sleep and Stop modes)

由软件置 1 和清零。

0：睡眠和停止模式期间 USART3 时钟由时钟门控(1) 禁止

1：睡眠和停止模式期间 USART3 时钟由时钟门控(1) 使能

位 17 **USART2SMEN**：睡眠和停止模式期间的 USART2 时钟使能 (USART2 clocks enable during Sleep and Stop modes)

由软件置 1 和清零。

0：睡眠和停止模式期间 USART2 时钟由时钟门控(1) 禁止

1：睡眠和停止模式期间 USART2 时钟由时钟门控(1) 使能

位 16 保留，必须保持复位值。

位 15 **SPI3SMEN**：睡眠和停止模式期间的 SPI3 时钟使能 (SPI3 clocks enable during Sleep and Stop modes)

由软件置 1 和清零。

0：睡眠和停止模式期间 SPI3 时钟由时钟门控(1) 禁止

1：睡眠和停止模式期间 SPI3 时钟由时钟门控(1) 使能

位 14 **SPI2SMEN**：睡眠和停止模式期间的 SPI2 时钟使能 (SPI2 clocks enable during Sleep and Stop modes)

由软件置 1 和清零。

0：睡眠和停止模式期间 SPI2 时钟由时钟门控(1) 禁止

1：睡眠和停止模式期间 SPI2 时钟由时钟门控(1) 使能

位 13:12 保留，必须保持复位值。

位 11 **WWDGSMEN**：睡眠和停止模式期间的窗口看门狗时钟使能 (Window watchdog clocks enable during Sleep and Stop modes)

由软件置 1 和清零。激活硬件 WWDG 选项时，该位由硬件强制置“1”。

0：睡眠和停止模式期间窗口看门狗时钟由时钟门控(1) 禁止

1：睡眠和停止模式期间窗口看门狗时钟由时钟门控(1) 使能

位 10 **RTCAPBSMEN**：睡眠和停止模式期间的 RTC APB 时钟使能 (RTC APB clock enable during Sleep and Stop modes)

由软件置 1 和清零。

0：睡眠和停止模式期间 RTC APB 时钟由时钟门控(1) 禁止

1：睡眠和停止模式期间 RTC APB 时钟由时钟门控(1) 使能

位 6 保留，必须保持复位值。

位 5 **TIM7SMEN**：睡眠和停止模式期间的 TIM7 定时器时钟使能 (TIM7 timer clocks enable during Sleep and Stop modes)

由软件置 1 和清零。
0：睡眠和停止模式期间 TIM7 时钟由时钟门控(1) 禁止
1：睡眠和停止模式期间 TIM7 时钟由时钟门控(1) 使能

位 4 **TIM6SMEN**：睡眠和停止模式期间的 TIM6 定时器时钟使能 (TIM6 timer clocks enable during Sleep and Stop modes)

由软件置 1 和清零。
0：睡眠和停止模式期间 TIM6 时钟由时钟门控(1) 禁止
1：睡眠和停止模式期间 TIM6 时钟由时钟门控(1) 使能

位 3 **TIM5SMEN**：睡眠和停止模式期间的 TIM5 定时器时钟使能 (TIM5 timer clocks enable during Sleep and Stop modes)

由软件置 1 和清零。
0：睡眠和停止模式期间 TIM5 时钟由时钟门控(1) 禁止
1：睡眠和停止模式期间 TIM5 时钟由时钟门控(1) 使能

位 2 **TIM4SMEN**：睡眠和停止模式期间的 TIM4 定时器时钟使能 (TIM4 timer clocks enable during Sleep and Stop modes)

由软件置 1 和清零。
0：睡眠和停止模式期间 TIM4 时钟由时钟门控(1) 禁止
1：睡眠和停止模式期间 TIM4 时钟由时钟门控(1) 使能

位 1 **TIM3SMEN**：睡眠和停止模式期间的 TIM3 定时器时钟使能 (TIM3 timer clocks enable during Sleep and Stop modes)

由软件置 1 和清零。
0：睡眠和停止模式期间 TIM3 时钟由时钟门控(1) 禁止
1：睡眠和停止模式期间 TIM3 时钟由时钟门控(1) 使能

位 0 **TIM2SMEN**：睡眠和停止模式期间的 TIM2 定时器时钟使能 (TIM2 timer clocks enable during Sleep and Stop modes)

由软件置 1 和清零。
0：睡眠和停止模式期间 TIM2 时钟由时钟门控(1) 禁止
1：睡眠和停止模式期间 TIM2 时钟由时钟门控(1) 使能

1. 该寄存器仅配置时钟门控，而不是时钟源本身。大多数外设的时钟由一个时钟（AHB 或 APB 时钟）提供，该时钟在停止模式下始终处于禁止状态。因此，在停止模式下将该位置 1 不起作用。

6.4.26 睡眠和停止模式下的 APB1 外设时钟使能寄存器 2 (RCC_APB1SMENR2)

APB1 peripheral clocks enable in Sleep and Stop modes register 2

偏移地址：0x7C

复位值：0x0000 0023

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
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.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LPTIM 2SMEN	Res.	Res.	Res.	I2C4S MEN	LPUART 1SMEN
										rw				rw	rw

位 31:6 保留，必须保持复位值。

位 5 **LPTIM2SMEN**：睡眠和停止模式期间的低功耗定时器 2 时钟使能 (Low power timer 2 clocks enable during Sleep and Stop modes)

由软件置 1 和清零。

0：睡眠和停止模式期间 LPTIM2 时钟由时钟门控(1) 禁止

1：睡眠和停止模式期间 LPTIM2 时钟由时钟门控(1) 使能

位 4 保留，必须保持复位值。

位 1 **I2C4SMEN**：睡眠和停止模式期间的 I2C4 时钟使能 (I2C4 clocks enable during Sleep and Stop modes)

由软件置 1 和清零。

0：睡眠和停止模式期间 I2C4 时钟由时钟门控(1) 禁止

1：睡眠和停止模式期间 I2C4 时钟由时钟门控(1) 使能

位 0 **LPUART1SMEN**：睡眠和停止模式期间的低功耗 UART 1 时钟使能 (Low power UART 1 clocks enable during Sleep and Stop modes)

由软件置 1 和清零。

0：睡眠和停止模式期间 LPUART1 时钟由时钟门控(1) 禁止

1：睡眠和停止模式期间 LPUART1 时钟由时钟门控(1) 使能

1. 该寄存器仅配置时钟门控，而不是时钟源本身。大多数外设的时钟由一个时钟（AHB 或 APB 时钟）提供，该时钟在停止模式下始终处于禁止状态。因此，在停止模式下将该位置 1 不起作用。

6.4.27 睡眠和停止模式下的 APB2 外设时钟使能寄存器 (RCC_APB2SMENR)

APB2 peripheral clocks enable in Sleep and Stop modes register

地址：0x80

复位值：0x0D67 7801

访问：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
Res.	Res.	Res.	Res.	DSISMEN	LTDCSMEN	Res.	DFSDM1SMEN	Res.	SAI2SMEN	SAI1SMEN	Res.	Res.	TIM17SMEN	TIM16SMEN	TIM15SMEN
				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.	USART1SMEN	TIM8SMEN	SPI1SMEN	TIM1SMEN	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SYSCFGSMEN
	rw	rw	rw	rw											rw

位 31:28 保留，必须保持复位值。

位 27 **DSISMEN**：睡眠和停止模式期间的 DSI 时钟使能 (DSI clocks enable during Sleep and Stop modes)

由软件置 1 和清零。

0：睡眠和停止模式期间 DSI 时钟由时钟门控(1) 禁止

1：睡眠和停止模式期间 DSI 时钟由时钟门控(1) 使能

位 26 **LTDCSMEN**：睡眠和停止模式期间的 LCD-TFT 定时器时钟使能 (LCD-TFT timer clocks enable during Sleep and Stop modes)

由软件置 1 和清零。

0：睡眠和停止模式期间 LCD-TFT 时钟由时钟门控(1) 禁止

1：睡眠和停止模式期间 LCD-TFT 时钟由时钟门控(1) 使能

位 25 保留，必须保持复位值。

位 24 **DFSDM1SMEN**：睡眠和停止模式期间的 DFSDM1 定时器时钟使能 (DFSDM1 timer clocks enable during Sleep and Stop modes)

由软件置 1 和清零。

0：睡眠和停止模式期间 DFSDM1 时钟由时钟门控(1) 禁止

1：睡眠和停止模式期间 DFSDM1 时钟由时钟门控(1) 使能

位 23 保留，必须保持复位值。

位 22 **SAI2SMEN**：睡眠和停止模式期间的 SAI2 时钟使能 (SAI2 clocks enable during Sleep and Stop modes)

由软件置 1 和清零。

0：睡眠和停止模式期间 SAI2 时钟由时钟门控(1) 禁止

1：睡眠和停止模式期间 SAI2 时钟由时钟门控(1) 使能

位 21 **SAI1SMEN**：睡眠和停止模式期间的 SAI1 时钟使能 (SAI1 clocks enable during Sleep and Stop modes)

由软件置 1 和清零。

0：睡眠和停止模式期间 SAI1 时钟由时钟门控(1) 禁止

1：睡眠和停止模式期间 SAI1 时钟由时钟门控(1) 使能

位 20:19 保留，必须保持复位值。

位 18 **TIM17SMEN**：睡眠和停止模式期间的 TIM17 定时器时钟使能 (TIM17 timer clocks enable during Sleep and Stop modes)

由软件置 1 和清零。

0：睡眠和停止模式期间 TIM17 定时器时钟由时钟门控(1) 禁止

1：睡眠和停止模式期间 TIM17 定时器时钟由时钟门控(1) 使能

位 17 **TIM16SMEN**：睡眠和停止模式期间的 TIM16 定时器时钟使能 (TIM16 timer clocks enable during Sleep and Stop modes)

由软件置 1 和清零。

0：睡眠和停止模式期间 TIM16 定时器时钟由时钟门控(1) 禁止

1：睡眠和停止模式期间 TIM16 定时器时钟由时钟门控(1) 使能

位 16 **TIM15SMEN**：睡眠和停止模式期间的 TIM15 定时器时钟使能 (TIM15 timer clocks enable during Sleep and Stop modes)

由软件置 1 和清零。

0：睡眠和停止模式期间 TIM15 定时器时钟由时钟门控(1) 禁止

1：睡眠和停止模式期间 TIM15 定时器时钟由时钟门控(1) 使能

位 15 保留，必须保持复位值。

位 14 **USART1SMEN**：睡眠和停止模式期间的 USART1 时钟使能 (USART1 clocks enable during Sleep and Stop modes)

由软件置 1 和清零。

0：睡眠和停止模式期间 USART1 时钟由时钟门控(1) 禁止

1：睡眠和停止模式期间 USART1 时钟由时钟门控(1) 使能

位 13 **TIM8SMEN**：睡眠和停止模式期间的 TIM8 定时器时钟使能 (TIM8 timer clocks enable during Sleep and Stop modes)

由软件置 1 和清零。

0：睡眠和停止模式期间 TIM8 定时器时钟由时钟门控(1) 禁止

1：睡眠和停止模式期间 TIM8 定时器时钟由时钟门控(1) 使能

位 12 **SPI1SMEN**：睡眠和停止模式期间的 SPI1 时钟使能 (SPI1 clocks enable during Sleep and Stop modes)

由软件置 1 和清零。

0：睡眠和停止模式期间 SPI1 时钟由时钟门控(1) 禁止

1：睡眠和停止模式期间 SPI1 时钟由时钟门控(1) 使能

位 11 **TIM1SMEN**：睡眠和停止模式期间的 TIM1 定时器时钟使能 (TIM1 timer clocks enable during Sleep and Stop modes)

由软件置 1 和清零。

0：睡眠和停止模式期间 TIM1 定时器时钟由时钟门控(1) 禁止

1：睡眠和停止模式期间 TIM1 定时器时钟由时钟门控(1) 使能

位 10:1 保留，必须保持复位值。

位 0 **SYSCFGSMEN**：睡眠和停止模式期间的 SYSCFG + COMP + VREFBUF 时钟使能 (SYSCFG + COMP + VREFBUF clocks enable during Sleep and Stop modes)

由软件置 1 和清零。

0：睡眠和停止模式期间 SYSCFG + COMP + VREFBUF 时钟由时钟门控(1) 禁止

1：睡眠和停止模式期间 SYSCFG + COMP + VREFBUF 时钟由时钟门控(1) 使能

1. 该寄存器仅配置时钟门控，而不是时钟源本身。大多数外设的时钟由一个时钟（AHB 或 APB 时钟）提供，该时钟在停止模式下始终处于禁止状态。因此，在停止模式下将该位置 1 不起作用。

6.4.28 外设专用时钟配置寄存器 (RCC_CCIPR)

Peripherals independent clock configuration register

地址：0x88

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
Res.	Res.	ADCSEL[1:0]		CLK48SEL[1:0]		Res.				LPTIM2SEL[1:0]		LPTIM1SEL[1:0		I2C3SEL[1:0]	
		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
I2C2SEL[1:0]		I2C1SEL[1:0]		LPUART1SEL[1:0]		UART5SEL[1:0]		UART4SEL[1:0]		USART3SEL[1:0]		USART2SEL[1:0]		USART1SEL[1:0]	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31 保留，必须保持复位值。

位 30 保留，必须保持复位值。

位 29:28 **ADCSEL[1:0]**：ADC 时钟源选择 (ADC clock source selection)：
这些位由软件置 1 和清零，用于选择 ADC 接口使用的时钟源。
00：没有选择时钟
01：选择 PLLSAI1“R”时钟 (PLLADC1CLK) 作为 ADC 时钟
10：保留
11：选择系统时钟作为 ADC 时钟

位 27:26 **CLK48SEL[1:0]**：48 MHz 时钟源选择 (48 MHz clock source selection)
这些位由软件置 1 和清零，用于选择 USB OTG FS、RNG 和 SDMMC 使用的 48 MHz 时钟源。
00：选择 HSI48 时钟作为 48 MHz 时钟
01：选择 PLLSAI1“Q”时钟 (PLL48M2CLK) 作为 48 MHz 时钟
10：选择 PLL“Q”时钟 (PLL48M1CLK) 作为 48 MHz 时钟
11：选择 MSI 时钟作为 48 MHz 时钟

位 22:25 保留，必须保持复位值。

位 21:20 **LPTIM2SEL[1:0]**：低功耗定时器 2 时钟源选择 (Low power timer 2 clock source selection)
这些位由软件置 1 和清零，用于选择 LPTIM2 时钟源。
00：选择 PCLK 作为 LPTIM2 时钟
01：选择 LSI 时钟作为 LPTIM2 时钟
10：选择 HSI16 时钟作为 LPTIM2 时钟
11：选择 LSE 时钟作为 LPTIM2 时钟

位 19:18 **LPTIM1SEL[1:0]**：低功耗定时器 1 时钟源选择 (Low power timer 1 clock source selection)
这些位由软件置 1 和清零，用于选择 LPTIM1 时钟源。
00：选择 PCLK 作为 LPTIM1 时钟
01：选择 LSI 时钟作为 LPTIM1 时钟
10：选择 HSI16 时钟作为 LPTIM1 时钟
11：选择 LSE 时钟作为 LPTIM1 时钟

位 17:16 **I2C3SEL[1:0]**：I2C3 时钟源选择 (I2C3 clock source selection)
这些位由软件置 1 和清零，用于选择 I2C3 时钟源。
00：选择 PCLK 作为 I2C3 时钟
01：选择系统时钟 (SYSCLK) 作为 I2C3 时钟
10：选择 HSI16 时钟作为 I2C3 时钟
11：保留

位 15:14 **I2C2SEL[1:0]**：I2C2 时钟源选择 (I2C2 clock source selection)
这些位由软件置 1 和清零，用于选择 I2C2 时钟源。
00：选择 PCLK 作为 I2C2 时钟
01：选择系统时钟 (SYSCLK) 作为 I2C2 时钟
10：选择 HSI16 时钟作为 I2C2 时钟
11：保留

位 13:12 **I2C1SEL[1:0]**：I2C1 时钟源选择 (I2C1 clock source selection)
这些位由软件置 1 和清零，用于选择 I2C1 时钟源。
00：选择 PCLK 作为 I2C1 时钟
01：选择系统时钟 (SYSCLK) 作为 I2C1 时钟
10：选择 HSI16 时钟作为 I2C1 时钟
11：保留

位 11:10 **LPUART1SEL[1:0]**：LPUART1 时钟源选择 (LPUART1 clock source selection)
这些位由软件置 1 和清零，用于选择 LPUART1 时钟源。
00：选择 PCLK 作为 LPUART1 时钟
01：选择系统时钟 (SYSCLK) 作为 LPUART1 时钟
10：选择 HSI16 时钟作为 LPUART1 时钟
11：选择 LSE 时钟作为 LPUART1 时钟

位 9:8 **UART5SEL[1:0]**：UART5 时钟源选择 (UART5 clock source selection)
这些位由软件置 1 和清零，用于选择 UART5 时钟源。
00：选择 PCLK 作为 UART5 时钟
01：选择系统时钟 (SYSCLK) 作为 UART5 时钟
10：选择 HSI16 时钟作为 UART5 时钟
11：选择 LSE 时钟作为 UART5 时钟

位 7:6 **UART4SEL[1:0]**：UART4 时钟源选择 (UART4 clock source selection)
此位由软件置 1 和清零，用于选择 UART4 时钟源。
00：选择 PCLK 作为 UART4 时钟
01：选择系统时钟 (SYSCLK) 作为 UART4 时钟
10：选择 HSI16 时钟作为 UART4 时钟
11：选择 LSE 时钟作为 UART4 时钟

位 5:4 **USART3SEL[1:0]**：USART3 时钟源选择 (USART3 clock source selection)
此位由软件置 1 和清零，用于选择 USART3 时钟源。
00：选择 PCLK 作为 USART3 时钟
01：选择系统时钟 (SYSCLK) 作为 USART3 时钟
10：选择 HSI16 时钟作为 USART3 时钟
11：选择 LSE 时钟作为 USART3 时钟

位 3:2 **USART2SEL[1:0]**：USART2 时钟源选择 (USART2 clock source selection)
此位由软件置 1 和清零，用于选择 USART2 时钟源。
00：选择 PCLK 作为 USART2 时钟
01：选择系统时钟 (SYSCLK) 作为 USART2 时钟
10：选择 HSI16 时钟作为 USART2 时钟
11：选择 LSE 时钟作为 USART2 时钟

位 1:0 **USART1SEL[1:0]**：USART1 时钟源选择 (USART1 clock source selection)
此位由软件置 1 和清零，用于选择 USART1 事件源。
00：选择 PCLK 作为 USART1 时钟
01：选择系统时钟 (SYSCLK) 作为 USART1 时钟
10：选择 HSI16 时钟作为 USART1 时钟
11：选择 LSE 时钟作为 USART1 时钟

6.4.29 备份域控制寄存器 (RCC_BDCR)

Backup domain control register

偏移地址：0x90
复位值：0x0000 0000，除 LSCOSEL、LSCOEN 和 BDRST 只能通过备份域上电复位进行复位以外，其他位均通过备份域复位进行复位。
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。

注： *备份域控制寄存器 (RCC_BDCR) 的位在 V_{CORE} 域之外。因而复位后，这些位受写保护，必须将第 5.4.1 节：电源控制寄存器 1 (PWR_CR1) 中的 DBP 位置 1 才能修改这些位。有关详细信息，请参见第 156 页的第 5.1.5 节：电池备份域。只有备份域复位后，这些位（LSCOSEL、LSCOEN 和 BDRST 除外）才能复位（请参见第 6.1.3 节：备份域复位）。内部复位和外部复位对这些位不会有任何影响。*

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RTCEN	Res.	Res.	Res.	Res.	Res.	RTCSEL[1:0]		Res.	LSECSSD	LSECSSON	LSEDRV[1:0]		LSEBYP	LSERDY	LSEON
rw						rw	rw		r	rw	rw	rw	rw	r	rw

位 31:26 保留，必须保持复位值。

位 25 **LSCOSEL**：低速时钟输出选择 (Low speed clock output selection)
由软件置 1 和清零。
0：选择 LSI 时钟
1：选择 LSE 时钟

位 24 **LSCOEN**：低速时钟输出使能 (Low speed clock output enable)
由软件置 1 和清零。
0：禁止低速时钟输出 (LSCO)
1：使能低速时钟输出 (LSCO)

位 23:17 保留，必须保持复位值。

位 16 **BDRST**：备份域软件复位 (Backup domain software reset)
由软件置 1 和清零。
0：复位未激活
1：复位整个备份域

位 15 **RTCEN**：RTC 时钟使能 (RTC clock enable)
由软件置 1 和清零。
0：禁止 RTC 时钟
1：使能 RTC 时钟

位 14:10 保留，必须保持复位值。

位 9:8 **RTCSEL[1:0]**：RTC 时钟源选择 (RTC clock source selection)

由软件置 1，用于选择 RTC 的时钟源。选择 RTC 时钟源后，除非备份域复位或在 LSE 上检测到故障（LSECSSD 置 1），否则不可再将其更改。可使用 BDRST 位对其进行复位。

00：无时钟

01：LSE 振荡器时钟用作 RTC 时钟

10：LSI 振荡器时钟用作 RTC 时钟

11：32 分频的 HSE 振荡器时钟用作 RTC 时钟

位 7 保留，必须保持复位值。

位 6 **LSECSSD**：LSE 上的 CSS 故障检测 (CSS on LSE failure Detection)

由硬件置 1，用于指示外部 32 kHz 振荡器 (LSE) 上的时钟安全系统何时检测到故障。

0：LSE（32 kHz 振荡器）上未检测到故障

1：LSE（32 kHz 振荡器）上检测到故障

位 5 **LSECSSON**：LSE 上的 CSS 使能 (CSS on LSE enable)

由软件置 1，用于使能 LSE（32 kHz 振荡器）上的时钟安全系统。

在 LSE 振荡器使能（已使能 LSEON 位）和就绪（由硬件将 LSERDY 标志置 1）后以及在选择 RTCSEL 位后，LSECSSON 必须使能。

该位一旦使能便不能禁止，但检测到 LSE 故障（LSECSSD =1）后除外。

此时，软件必须禁止 LSECSSON 位。

0：LSE（32 kHz 外部振荡器）上的 CSS 关闭

1：LSE（32 kHz 外部振荡器）上的 CSS 开启

位 4:3 **LSEDRV[1:0]**：LSE 振荡器驱动能力 (LSE oscillator drive capability)

由软件置 1，用于调整 LSE 振荡器的驱动能力。

00：“Xtal 模式”低驱动能力

01：“Xtal 模式”中低驱动能力

10：“Xtal 模式”中高驱动能力

11：“Xtal 模式”高驱动能力

当振荡器不处于旁路模式时，则处于 Xtal 模式。

位 2 **LSEBYP**：LSE 振荡器旁路 (LSE oscillator bypass)

由软件置 1 和清零，用于旁路调试模式下的振荡器。只有在禁止外部 32 kHz 振荡器（LSEON=0 且 LSERDY=0）后才能写入该位。

0：不旁路 LSE 振荡器

1：旁路 LSE 振荡器

位 1 **LSERDY**：LSE 振荡器就绪 (LSE oscillator ready)

由硬件置 1 和清零，用于指示外部 32 kHz 振荡器已稳定。在 LSEON 位被清零后，LSERDY 将在 6 个外部低速振荡器时钟周期后转为低电平。

0：LSE 振荡器未就绪

1：LSE 振荡器已就绪

位 0 **LSEON**：LSE 振荡器使能 (LSE oscillator enable)

由软件置 1 和清零。

0：LSE 振荡器关闭

1：LSE 振荡器开启

6.4.30 控制/状态寄存器 (RCC_CSR)

Control/status register

地址：0x94

复位值：0x0C00 0600，除复位标志只能通过电源复位进行复位以外，其他标志均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	IWWGRSTF	SFTRSTF	BORRSTF	PINRSTF	OBLRSTF	FWRSTF	RMVF	Res.	Res.	Res.	Res.	Res.	Res.	Res.
r	r	r	r	r	r	r	r	rw							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	MSISRANGE[3:0]				Res.	Res.	Res.	Res.	Res.	Res.	LSIRDY	LSION
				rw	rw	rw	rw							r	rw

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

由于进入非法停止、待机或关断模式而发生复位时，由硬件置 1。
通过写入 RMVF 位清零。
0：未发生非法模式复位
1：发生非法模式复位

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**：独立窗口看门狗复位标志 (Independent window watchdog reset flag)

发生独立看门狗复位时，由硬件置 1。
通过写入 RMVF 位清零。
0：未发生独立看门狗复位
1：发生独立看门狗复位

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 **BORRSTF**：BOR 标志 (BOR flag)

发生 BOR 时，由硬件置 1。
通过写入 RMVF 位清零。
0：未发生 BOR
1：发生 BOR

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

位 25 **OBLRSTF**：选项字节加载复位标志 (Option byte loader reset flag)
发生来自选项字节加载的复位时，由硬件置 1。
通过写入 RMVF 位清零。
0：未发生来自选项字节加载的复位
1：发生来自选项字节加载的复位

位 24 **FWRSTF**：防火墙复位标志 (Firewall reset flag)
发生来自防火墙的复位时，由硬件置 1。
通过写入 RMVF 位清零。
0：未发生来自防火墙的复位
1：发生来自防火墙的复位

位 23 **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：无影响
1：清除复位标志

位 22:12 保留，必须保持复位值。

位 11:8 **MSISRANGE[3:0]**：待机模式后的 MSI 范围 (MSI range after Standby mode)
由软件置 1，用于在启动时选择 MSI 频率。退出待机模式后将使用此范围，直到 MSIRGSEL 置 1 为止。在数据填充或上电复位后，范围始终为 4 MHz。仅可在 MSIRGSEL = “1” 时写入 MSISRANGE。
0100：范围 4，1 MHz 左右
0101：范围 5，2 MHz 左右
0101：范围 6，4 MHz 左右（复位值）
0111：范围 7，8 MHz 左右
其他值：保留

注：　更改 MSISRANGE 时不会更改当前 MSI 频率。

位 7:2 保留，必须保持复位值。

位 1 **LSIRDY**：LSI 振荡器就绪 (LSI oscillator ready)
由硬件置 1 和清零，用于指示 LSI 振荡器已稳定。在将 LSION 位清零后，LSIRDY 将在 3 个 LSI 振荡器时钟周期后转为低电平。如果存在 LSE 上的时钟安全系统、独立看门狗或 RTC 发出的 LSI 请求，即使 LSION = 0，该位也可置 1。
0：LSI 振荡器未就绪
1：LSI 振荡器已就绪

位 0 **LSION**：LSI 振荡器使能 (LSI oscillator enable)
由软件置 1 和清零。
0：LSI 振荡器关闭
1：LSI 振荡器开启

6.4.31 时钟恢复 RC 寄存器 (RCC_CRRCR)

Clock recovery RC register

地址：0x98

复位值：0x0000 XXX0，其中 X 是出厂前编程的。

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
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
HSI48CAL[8:0]									Res.	Res.	Res.	Res.	Res.	HSI48RDY	HSI48ON
r	r	r	r	r	r	r	r	r						r	rw

位 31:16 保留，必须保持复位值

位 15:7 **HSI48CAL[8:0]**：HSI48 时钟校准 (HSI48 clock calibration)

这些位在启动时初始化为出厂前编程的 HSI48 校准微调值。

它们是只读的。

位 6:2 保留，必须保持复位值

位 1 **HSI48RDY**：HSI48 时钟就绪标志 (HSI48 clock ready flag)

由硬件置 1，用以指示 HSI48 振荡器已稳定。仅当通过软件将 HSI48ON 置 1 来使能 HSI48 时，该位才置 1。

0：HSI48 振荡器未就绪

1：HSI48 振荡器已就绪

位 0 **HSI48ON**：HSI48 时钟使能 (HSI48 clock enable)

由软件置 1 和清零。

由硬件清零，用于在进入停止、待机和关断模式时停止 HSI48。

0：HSI48 振荡器关闭

1：HSI48 振荡器开启

6.4.32 外设独立时钟配置寄存器 (RCC_CCIPR2)

Peripherals independent clock configuration register

地址：0x9C

复位值：0x0000 0000

访问：无等待周期，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
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OSPISEL[1:0]		Res.	Res.	PLLSAI2DIVR[1:0]	
										rw				rw	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	SDMMCSEL	Res.	DSISEL	Res.	SAI2SEL[2:0]			SAI1SEL[2:0]			ADFSDMSEL[1:0]		DFSDMSEL	I2C4SEL[1:0]	
	rw		rw		rw			rw			rw		rw	rw	

位 31:22 保留，必须保持复位值。

位 21:20 **OSPISEL**：Octospi 时钟源选择 (Octospi clock source selection)
由软件置 1 和复位。
00：选择系统时钟作为 OctoSPI 内核时钟
01：选择 MSI 时钟作为 OctoSPI 内核时钟
10：选择 PLL48M1CLK 时钟作为 OctoSPI 内核时钟
11：保留

位 19:18 保留，必须保持复位值。

位 17:16 **PLLSAI2DIVR**：适用于 LTDC 时钟的分频系数 (division factor for LTDC clock)
由软件置 1 和复位，用于控制 LTDC 时钟的频率。
这些位只能在 PLLSAI2 已禁止时写入。
LTDC 时钟频率 = f(PLLSAI2_R) / PLLSAI2DIVR，其中，2 <= PLLSAI2DIVR <= 16
00：PLLSAI2DIVR = /2
01：PLLSAI2DIVR = /4
10：PLLSAI2DIVR = /8
11：PLLSAI2DIVR = /16

位 15 保留，必须保持复位值。

位 14 **SDMMCSEL**：SDMMC 时钟选择 (SDMMC clock selection)
由软件置 1 和复位。
此位可用于在 PLLP 时钟 (PLLSAI3CLK) 与来自内部多路复用器的时钟之间选择 SDMMC 内核时钟源。
建议仅在复位之后使能 SDMMC 模块之前对此位进行更改。
0：选择 48 MHz 时钟作为 SDMMC 内核时钟。
1：选择 PLLSAI3CLK 作为 SDMMC 内核时钟，用于频率需要高于 48 MHz 的情况（针对 SDR50 模式）。

位 13 保留，必须保持复位值。

位 12 **DSISEL**：时钟选择 (clock selection)
由软件置 1 和复位。
此位可用于在 PLLDSICLK 时钟于来自 DSI-PHY 的时钟之间选择 DSI 字节通道时钟源。
建议仅在复位之后使能 DSI 模块之前对此位进行更改。
0：选择 DSI-PHY 作为 DSI 字节通道时钟源（通常情况下）。
1：选择 PLLDSICLK 作为 DSI 字节通道时钟源，用于 DSI PLL 和 DSI-PHY 关闭的情况（低功耗模式）。

位 11 保留，必须保持复位值。

位 10:8 **SAI2SEL**：SAI2 时钟源选择 (SAI2 clock source selection)
由软件置 1 和复位。
如果所选时钟为外部时钟，且该时钟已停止，则无法切换到其他时钟。在停止外部时钟之前，用户必须切换到另一个时钟。
000：选择 PLLSAI1CLK 时钟作为 SAI2 时钟
001：选择 PLLSAI2CLK 时钟作为 SAI2 时钟
010：选择 PLLSAI3CLK 时钟作为 SAI2 时钟
011：选择外部时钟 SAI2_EXTCLK 时钟作为 SAI2 时钟
100：选择 HSI 时钟作为 SAI2 时钟
其他配置保留

位 7:5 **SAI1SEL**：SAI1 时钟源选择 (SAI1 clock source selection)
由软件置 1 和复位。
如果所选时钟为外部时钟，且该时钟已停止，则无法切换到其他时钟。
在停止外部时钟之前，用户必须切换到另一个时钟。
000：选择 PLLSAI1CLK 时钟作为 SAI1 时钟
001：选择 PLLSAI2CLK 时钟作为 SAI1 时钟
010：选择 PLLSAI3CLK 时钟作为 SAI11 时钟
011：选择外部时钟 SAI1_EXTCLK 作为 SAI1 时钟
100：选择 HSI 时钟作为 SAI2 时钟
其他配置保留

位 4:3 **ADFSDMSEL**：用于 ∑Δ 调制器的数字滤波器的音频时钟源选择
由软件置 1 和复位。
00：选择 SAI1 时钟作为 DFSDM 音频时钟
01：选择 HSI 时钟作为 DFSDM 音频时钟
10：选择 MSI 时钟作为 DFSDM 音频时钟
11：保留

位 2 **DFSDMSEL**：用于 ∑Δ 调制器的数字滤波器的内核时钟源选择
由软件置 1 和复位。
0：选择 APB2 时钟 (PCLK2) 作为 DFSDM 内核时钟
1：选择系统时钟作为 DFSDM 内核时钟

位 1:0 **I2C4SEL[1:0]**：I2C4 时钟源选择 (I2C4 clock source selection)
这些位由软件置 1 和清零，用于选择 I2C4 时钟源。
00：选择 PCLK 作为 I2C4 时钟
01：选择系统时钟 (SYSCLK) 作为 I2C4 时钟
10：选择 HSI16 时钟作为 I2C4 时钟
11：保留

6.4.33 RCC 寄存器映射

下表列出了 RCC 寄存器映射和复位值。

表 33. RCC 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	RCC_CR	Res.	Res.	PLLSAI2RDY	PLLSAI2ON	PLLSAI1RDY	PLLSAI1ON	PLLRDY	PLLON	Res.	Res.	Res.	Res.	CSSON	HSEBYP	HSERDY	HSEON	Res.	Res.	Res.	Res.	HSIASFS	HSIRDY	HSIKERON	HSION	MSIRANGE [3:0]				MSIRGSEL	MSIPLLEN	MSIRDY	MSION
	Reset value			0	0	0	0	0	0					0	0	0	0					0	0	0	0	0	1	1	0	0	0	1	1
0x04	RCC_ICSCR	Res	HSITRIM[6:0]							HSICAL[7:0]								MSITRIM[7:0]								MSICAL[7:0]							
	Reset value		1	0	0	0	0	0	0	x	x	x	x	x	x	x	x	0	0	0	0	0	0	0	0	x	x	x	x	x	x	x	x
0x08	RCC_CFGR	Res	MCOPRE [2:0]			MCOSEL [3:0]				Res.	Res.	Res.	Res.	Res	Res.	Res.	Res.	STOPWUCK	Res.	PPRE2 [2:0]			PPRE1 [2:0]			HPRE[3:0]				SWS [1:0]		SW [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
0x0C	RCC_PLLCFGR	PLLPDIV[4:0]					PLLR [1:0]		PLLREN	Res.	PLLQ [1:0]		PLLQEN	Res.	Res.	PLLP	PLLPEN	Res.	PLLN [6:0]							PLLM [3:0]				Res.	Res.	PLLSRC [1:0]	
	Reset value	0	0	0	0	0	0	0	0		0	0	0			0	0		0	0	1	0	0	0	0	0	0	0	0			0	0
0x10	RCC_PLLSAI1CFGR	PLLSAI1PDIV [4:0]					PLLSAI1R [1:0]		PLLSAI1REN	Res.	PLLSAI1Q [1:0]		PLLSAI1QEN	Res.	Res.	PLLSAI1P	PLLSAI1PEN	Res.	PLLSAI1N [6:0]							PLLSAI1M [3:0]				Res.	Res.	Res.	Res.
	Reset value	0	0	0	0	0	0	0	0		0	0	0			0	0		0	0	1	0	0	0	0	0	0	0	0				
0x14	RCC_PLLSAI2CFGR	PLLSAI2PDIV [4:0]					PLLSAI2R [1:0]		PLLSAI2REN	Res.	PLLSAI2Q[1:0]		PLLSAI2QEN	Res.	Res.	PLLSAI2P	PLLSAI2PEN	Res.	PLLSAI2N [6:0]							PLLSAI2M [3:0]				Res.	Res.	Res.	Res.
	Reset value	0	0	0	0	0	0	0	0		0	0	0			0	0		0	0	1	0	0	0	0	0	0	0	0				
0x18	RCC_CIER	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	HSI48RDYIE	LSECSSIE	Res.	PLLSAI2RDYIE	PLLSAI1RDYIE	PLLRDYIE	HSERDYIE	HSIRDYIE	MSIRDYIE	LSERDYIE	LSIRDYIE
	Reset value																						0	0		0	0	0	0	0	0	0	0

表 33. RCC 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
0x1C	RCC_CIFR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	HSI48RDYF	LSECSSF	CSSF	PLLSAI2RDYF	PLLSAI1RDYF	PLLRDYF	HSERDYF	HSIRDYF	MSIRDYF	LSERDYF	LSIRDYF
	Reset value																						0	0	0	0	0	0	0	0	0	0	0
0x20	RCC_CICR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	HSI48RDYC	LSECSSC	CSSC	PLLSAI2RDYC	PLLSAI1RDYC	PLLRDYC	HSERDYC	HSIRDYC	MSIRDYC	LSERDYC	LSIRDYC
	Reset value																						0	0	0	0	0	0	0	0	0	0	0
0x28	RCC_ AHB1RSTR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res	GFXMMURST	DMA2DRST	TSCRST	Res	Res.	Res.	CRCRST	Res.	Res	Res.	FLASHRST	Res.	Res.	Res	Res.	Res.	DMAMUX1RST	DMA2RST	DMA1RST
	Reset value														0	0	0				0				0						0	0	0
0x2C	RCC_ AHB2RSTR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SDMMC1RST	Res.	OSPIMRST	Res	RNGRST	HASHRST	AESRST	Res	DCMIRST	ADCRST	OTGFSRST	Res.	Res	Res.	GPIOIRST	GPIOHRST	GPIOGRST	GPIOFRST	GPIOERST	GPIODRST	GPIOCRST	GPIOBRST	GPIOARST
	Reset value										0		0		0	0	0		0	0	0				0	0	0	0	0	0	0	0	0
0x30	RCC_ AHB3RSTR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res	OSPI2RST	OSPI1RST	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FMCRST
	Reset value																							0	0								0
0x38	RCC_ APB1RSTR1	LPTIM1RST	OPAMPRST	DAC1RST	PWRRST	Res.	Res.	CAN1RST	CRSRST	I2C3RST	I2C2RST	I2C1RST	UART5RST	UART4RST	USART3RST	USART2RST	Res.	SPI3RST	SPI2RST	Res.	Res.	Res.	Res	Res.	Res.	Res.	Res.	TIM7RST	TIM6RST	TIM5RST	TIM4RST	TIM3RST	TIM2RST
	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	RCC_ APB1RSTR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res	Res.	Res.	Res.	Res.	LPTIM2RST	Res.	Res.	Res.	I2C4RST	LPUART1RST
	Reset value																											0				0	0

表 33. RCC 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
0x40	RCC_ APB2RSTR	Res.	Res.	Res.	Res.	DSIRST	LTDCRST	Res.	DFSDM1RST	Res.	SAI2RST	SAI1RST	Res.	Res.	TIM17RST	TIM16RST	TIM15RST	Res.	USART1RST	TIM8RST	SPI1RST	TIM1RST	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SYSCFGRST
	Reset value					0	0		0		0	0			0	0	0		0	0	0	0											0
0x48	RCC_AHB1 ENR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	GFXMMUEN	DMA2DEN	TSCEN	Res.	Res.	Res.	CRCEN	Res.	Res.	Res.	FLASHEN	Res.	Res.	Res.	Res.	Res.	DMAMUX1EN	DMA2EN	DMA1EN
	Reset value														0	0	0				0				1						0	0	0
0x4C	RCC_AHB2 ENR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SDMMC1EN	Res.	OSPIMEN	Res.	RNGEN	HASHEN	AESEN	Res.	DCMIEN	ADCEN	OTGFSEN	Res.	Res.	Res.	GPIOIEN	GPIOHEN	GPIOGEN	GPIOFEN	GPIOEEN	GPIODEN	GPIOCEN	GPIOBEN	GPIOAEN
	Reset value										0		0		0	0	0		0	0	0				0	0	0	0	0	0	0	0	0
0x50	RCC_AHB3 ENR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OSPI2EN	OSPI1EN	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FMCEN
	Reset value																							0	0								0
0x58	RCC_ APB1ENR1	LPTIM1EN	OPAMPEN	DAC1EN	PWREN	Res.	Res	CAN1EN	CRSEN	I2C3EN	I2C2EN	I2C1EN	UART5EN	UART4EN	USART3EN	USART2EN	Res.	SP3EN	SPI2EN	Res.	Res.	WWDGEN	RTCAPBEN	Res.	Res.	Res.	Res.	TIM7EN	TIM6EN	TIM5EN	TIM4EN	TIM3EN	TIM2EN
	Reset value	0	0	0	0			0	0	0	0	0	0	0	0	0		0	0			0	1					0	0	0	0	0	0
0x5C	RCC_ APB1ENR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LPTIM2EN	Res.	Res.	Res.	I2C4EN	LPUART1EN
	Reset value																											0				0	0
0x60	RCC_ APB2ENR	Res.	Res.	Res.	Res.	DSIEN	LTDCEN	Res.	DFSDM1EN	Res.	SAI2EN	SAI1EN	Res.	Res	TIM17EN	TIM16EN	TIM15EN	Res	USART1EN	TIM8EN	SPI1EN	TIM1EN	Res	Res	Res.	Res.	FWEN	Res.	Res.	Res.	Res.	Res.	Res.	SYSCFGEN
	Reset value					0	0		0		0	0			0	0	0		0	0	0	0					0							0
0x68	RCC_ AHB1SMENR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	GFXMMUSMEN	DMA2DSMEN	TSCSMEN	Res.	Res.	Res.	CRCSMEN	Res.	Res	SRAM1SMEN	FLASHSMEN	Res.	Res.	Res.	Res.	Res.	DMAMUX1SMEN	DMA2SMEN	DMA1SMEN
	Reset value														1	1	1				1			1	1						1	1	1

表 33. RCC 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
0x6C	RCC_ AHB2SMENR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SDMMC1SMEN	Res.	OSPIMSMEN	Res	RNGSMEN	HASHSMEN	AESSMEN	Res	DCMISMEN	ADCFSSMEN	OTGFSSMEN	Res.	SRAM3SMEN	SRAM2SMEN	GPIOISMEN	GPIOHSMEN	GPIOGSMEN	GPIOFSMEN	GPIOESMEN	GPIODSMEN	GPIOCSMEN	GPIOBSMEN	GPIOASMEN
	Reset value										1		1		1	1	1		1	1	1		1	1	1	1	1	1	1	1	1	1	1
0x70	RCC_ AHB3SMENR	Res.	Res.	Res.	Res.	Res.	Res	Res.	Res.	Res.	Res.	Res.	Res.	Res	Res.	Res.	Res.	Res	Res.	Res.	Res.	Res.	Res.	OSPI2SMEN	OSPI1SMEN	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FMCSMEN
	Reset value																							1	1								1
0x78	RCC_ APB1SM ENR1	LPTIM1SMEN	OPAMPSMEN	DAC1SMEN	PWRSMEN	Res.	Res	CAN1SMEN	CRSSMEN	I2C3SMEN	I2C2SMEN	I2C1SMEN	UART5SMEN	UART4SMEN	USART3SMEN	USART2SMEN	Res.	SP3SMEN	SPI2SMEN	Res.	Res.	WWDGSMEN	RTCAPBSMEN	Res.	Res.	Res.	Res.	Res.	TIM7SMEN	TIM6SMEN	TIM5SMEN	TIM4SMEN	TIM3SMEN	TIM2SMEN
	Reset value	1	1	1	1			1	1	1	1	1	1	1	1	1		1	1			1	1					1	1	1	1	1	1
0x7C	RCC_ APB1SM ENR2	Res.	Res.	Res.	Res.	Res.	Res	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res	Res.	Res.	Res.	Res.	Res	Res.	Res.	Res.	Res.	Res.	LPTIM2SMEN	Res.	Res.	Res.	I2C4SMEN	LPUART1SMEN
	Reset value																											1				1	1
0x80	RCC_ APB2SMENR	Res.	Res.	Res.	Res.	DSISMEN	LTDCSMEN	Res.	DFSDM1SMEN	Res.	SAI2SMEN	SAI1SMEN	Res.	Res	TIM17SMEN	TIM16SMEN	TIM15SMEN	Res	USART1SMEN	TIM8SMEN	SPI1SMEN	TIM1SMEN	Res	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SYSCFGSMEN
	Reset value					1	1		1		1	1			1	1	1		1	1	1	1												1
0x88	RCC_CCIPR	Res.	Res.	ADCSEL		CLK48SEL		Res.	Res.	Res.	Res.	LPTIM2SEL		LPTIM1SEL		I2C3SEL		I2C2SEL		I2C1SEL		LPUART1SEL		UART5SEL		UART4SEL		USART3SEL		USART2SEL		USART1SEL	
	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
0x90	RCC_BDCR	Res.	Res.	Res.	Res.	Res.	Res	LSCOSEL	LSCOEN	Res.	Res.	Res.	Res.	Res	Res.	Res.	BDRST	RTCEN	Res.	Res.	Res.	Res.	Res	Res.	RTC SEL [1:0]		Res.	LSECSSD	LSECSSON	LSE DRV [1:0]		LSEBYP	LSERDY	LSEON
	Reset value							0	0								0	0						0	0		0	0	0	0	0	0	0

表 33. RCC 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
0x94	RCC_CSR	LPWRRSTF	WWDGRSTF	IWDGRSTF	SFTRSTF	BORRSTF	PINRSTF	OBLRSTF	FIREWALLRSTF	RMVF	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MSIS RANGE [3:0]				Res.	Res.	Res.	Res.	Res.	Res.	LSIRDY	LSION
	Reset value	0	0	0	0	0	0	0	0	0												0	1	1	0							0	0
0x98	RCC_CRRCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	HSI48CAL[8:0]									Res.	Res.	Res.	Res.	Res.	Res.	HSI48RDY	HSI48ON
	Reset value																	x	x	x	x	x	x	x	x	x						0	0
0x9C	RCC_CCIPR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OSPISEL [1:0]		Res.	Res.	PLLSAI2DIVR [1:0]		Res.	SDMMCSEL	Res.	DSISEL	Res.	SAI2SEL [2:0]			SAI1SEL [2:0]			ADFSDMSEL [1:0]		DFSDMSEL	I2C4SEL [1:0]	
	Reset value											0	0			0	0		0		0		0	0	0	0	0	0	0	0	0	0	0

7 时钟恢复系统 (CRS)

7.1 简介

时钟恢复系统 (CRS) 是一个作用于内部精细粒度可调 RC 振荡器 HSI48 的高级数字控制器。CRS 提供一种十分强大的方法，可通过与可选同步信号进行比较来评估振荡器输出频率。它能够根据测得的频率误差值自动调整振荡器微调，同时还可以进行手动微调。

CRS 非常适合为 USB 外设提供精密时钟。在这种情况下，同步信号可由 USB 总线上的帧起始 (SOF) 数据包信号提供，USB 主机以 1 ms 的时间间隔精确发送该信号。

同步信号也可由 LSE 振荡器输出提供，或者由用户软件生成。

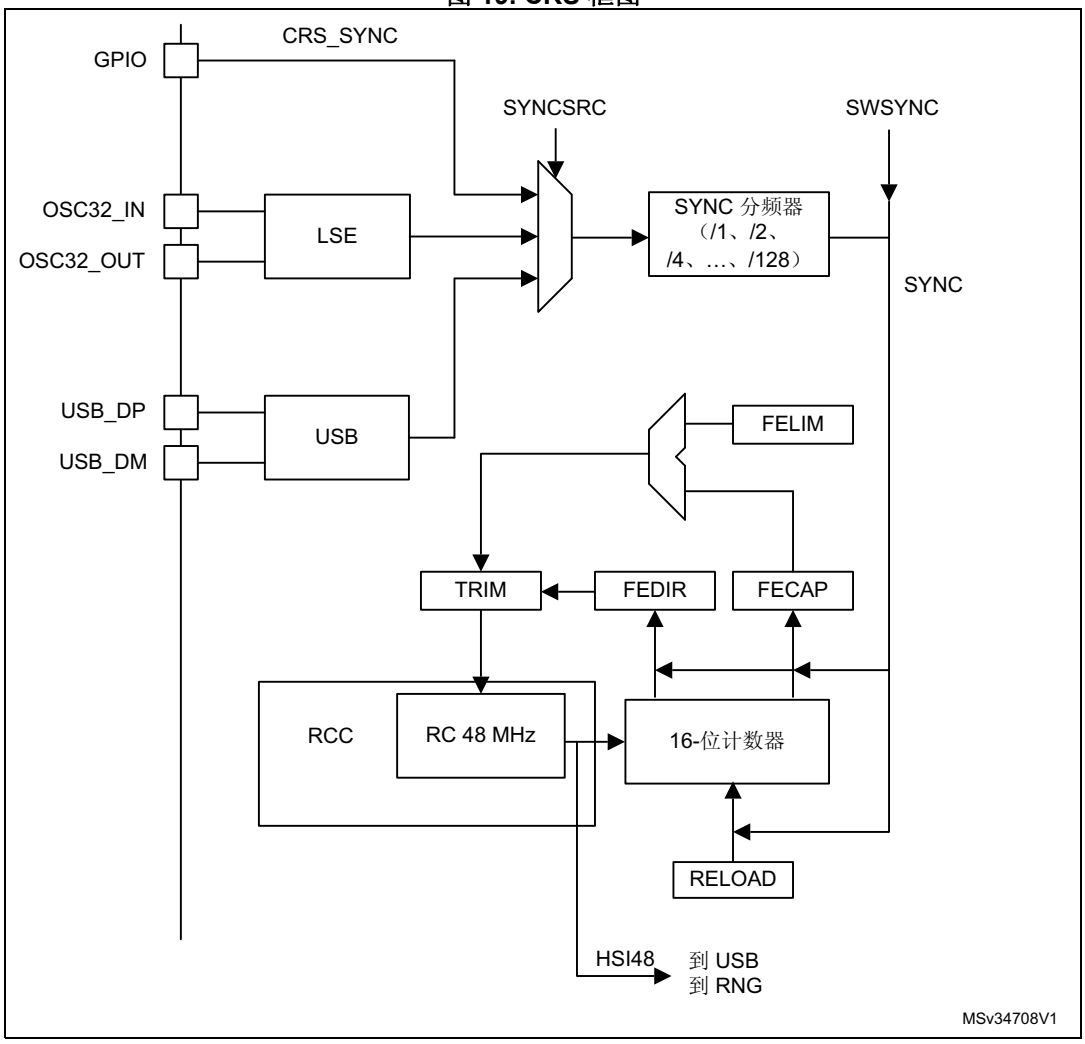
7.2 CRS 主要特性

- 具有可编程预分频器和极性的可选同步源：
  - 外部引脚
  - LSE 振荡器输出
  - USB SOF 数据包接收
- 可由软件生成同步脉冲
- 振荡器自动微调功能，无需 CPU 操作
- 手动控制选项，可加快启动融合
- 16 位频率误差计数器，用于自动误差值捕捉和重载
- 可编程限值，用于自动频率误差值评估和状态报告
- 可屏蔽中断/事件：
  - 预期同步 (ESYNC)
  - 同步正常 (SYNCOK)
  - 同步警告 (SYNCWARN)
  - 同步或微调错误 (ERR)

7.3 CRS 功能说明

7.3.1 CRS 框图

图 19. CRS 框图



7.3.2 同步输入

CRS 同步 (SYNC) 源可通过 CRS_CFGR 寄存器选择，它可以是来自 LSE 时钟的信号或 USB SOF 信号。为使 SYNC 输入更稳定，可使用一个简单的数字滤波器（三分之二多数表决，通过 RC48 时钟采样）来滤除任何毛刺。该信号源还具有可配置极性，可随后由可编程二进制预分频器分频，以获得合理频率范围内的同步信号（通常为 1 kHz 左右）。

有关 CRS 同步源配置的更多信息，请参见*第 7.6.2 节：CRS 配置寄存器 (CRS_CFGR)*。

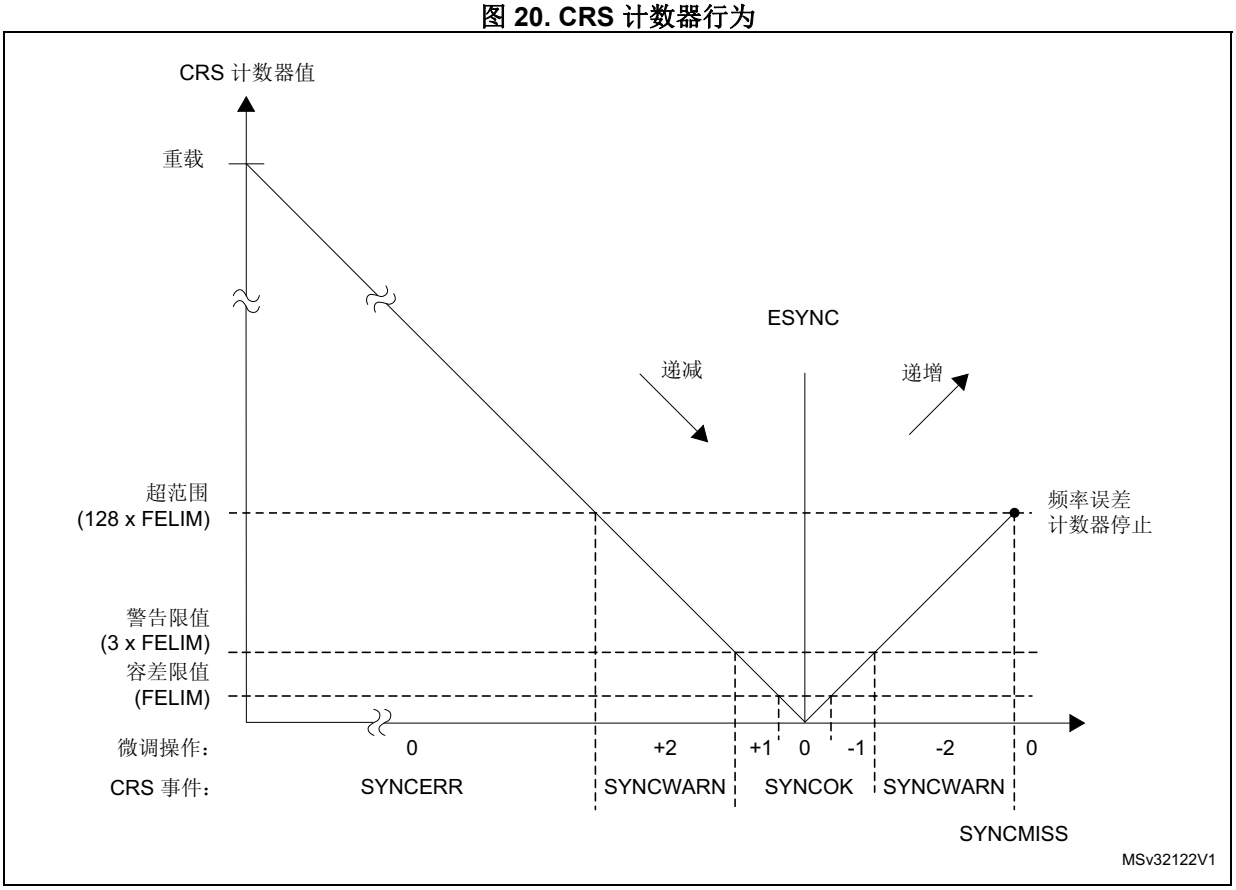
还可通过用软件将 CRS_CR 寄存器中的 SWSYNC 位置 1 的方法来生成同步事件。

7.3.3 频率误差测量

频率误差计数器是一个 16 位递减/递增计数器，会在每个 SYNC 事件发生时重载 RELOAD 值。它开始递减计数，直至达到零值，此时会生成 ESYNC（预期同步）事件。随后它将递增计数到 OUTRANGE 限值，这种情况下最终将停止计数（如果未接收到 SYNC 事件）并生成 SYNCMISS 事件。OUTRANGE 限值定义为频率误差限值（CRS_CFGR 寄存器的 FELIM 字段）乘以 128。

当检测到 SYNC 事件时，频率误差计数器的实际值及其方向存储在 CRS_ISR 寄存器的 FECAP（频率误差捕捉）字段和 FEDIR（频率误差方向）位中。当在递减计数阶段（达到零值之前）期间检测到 SYNC 事件时，意味着实际频率小于目标频率（因此，TRIM 值应递增）；当在递增计数期间检测到 SYNC 事件时，意味着实际频率大于目标频率（因此，TRIM 值应递减）。

图 20. CRS 计数器行为



7.3.4 频率误差评估和自动微调

测得的频率误差通过将频率值与一组限值进行比较进行估算：

- TOLERANCE LIMIT，在 CRS_CFGR 寄存器的 FELIM 字段中直接给出
- WARNING LIMIT，定义为 3 * FELIM 值
- OUTRANGE（误差限值），定义为 128 * FELIM 值

该比较结果用于生成状态指示以及控制自动微调，自动微调通过将 CRS_CR 寄存器中的 AUTOTRIMEN 位置 1 来使能。

- 当频率误差低于容差限值时，意味着 TRIM 字段中的实际微调值是最优值，因此无需任何微调操作。
  - 指示 SYNCOK 状态
  - AUTOTRIM 模式下的 TRIM 值无变化
- 当频率误差低于警告限值但高于或等于容差限值时，意味着需要某种微调操作，但只需一步微调便可达到最优 TRIM 值。
  - 指示 SYNCOK 状态
  - 在 AUTOTRIM 模式下通过一步微调来调整 TRIM 值
- 当频率误差高于或等于警告限值但低于误差限值时，意味着需要更强的微调操作，并且存在下一周期无法达到最优 TRIM 值的风险。
  - 指示 SYNCWARN 状态
  - 在 AUTOTRIM 模式下通过两步微调来调整 TRIM 值
- 当频率误差高于或等于误差限值时，意味着频率超出微调范围。当 SYNC 输入不干净或某个 SYNC 脉冲丢失时（例如，当一个 USB SOF 被破坏时）。也会发生这种情况。
  - 指示 SYNCERR 或 SYNCMISS 状态
  - AUTOTRIM 模式下的 TRIM 值无变化

注： *如果 TRIM 字段的实际值十分接近其限值，则自动微调会强制其上溢或下溢，之后 TRIM 值会恰好设置为限值，此时将指示 TRIMOVF 状态。*

在 AUTOTRIM 模式（CRS_CR 寄存器中的 AUTOTRIMEN 位置 1）下，CRS_CR 的 TRIM 字段由硬件调整并且是只读的。

7.3.5 CRS 初始化和配置

RELOAD 值

RELOAD 值应根据预分频后目标频率与同步源频率之比进行选择。该值随后会减 1，以在零值时达到预期同步。具体公式如下：

$$RELOAD = (f_{TARGET} / f_{SYNC}) - 1$$

RELOAD 字段的复位值对应于 48 MHz 的目标频率和 1 kHz 的同步信号频率（来自 USB 的 SOF 信号）。

FELIM 值

FELIM 值的选择与 HSI48 振荡器的特性及其典型微调步长紧密相关。最优值对应于微调步长的一半，以 HSI48 振荡器时钟节拍数表示。可使用以下公式：

$$FELIM = (f_{TARGET} / f_{SYNC}) * STEP[\%] / 100\% / 2$$

结果应始终舍入为最接近的整数值以获得最佳微调响应。如果应用中不需要频繁的微调操作，可通过稍微增大 FELIM 值来增加微调滞后。

FELIM 字段的复位值对应于 (f_{TARGET} / f_{SYNC}) = 48000 以及 0.14% 的典型微调步长。

注意： 错误配置 RELOAD 和 FELIM 字段无法实现硬件保护，从而导致不定微调响应。预期工作模式需要正确设置 RELOAD 值（根据同步源频率），该值还大于 128 * FELIM 值（OUTRANGE 限值）。

7.4 CRS 低功耗模式

表 34. 低功耗模式对 CRS 的作用

模式	说明
睡眠	无影响。 CRS 中断可使器件退出睡眠模式。
停止	CRS 寄存器被冻结。 CRS 停止工作，直到退出停止或待机模式且 HSI48 振荡器重启。
待机	

7.5 CRS 中断

表 35. 中断控制位

中断事件	事件标志	启用控制位	清零标志位
预期同步	ESYNCF	ESYNCIE	ESYNCC
同步正常	SYNCOKF	SYNCOKIE	SYNCOKC
同步警告	SYNCWARNF	SYNCWARNIE	SYNCWARNC
同步或微调错误 （TRIMOVF、SYNCMISS 和 SYNCERR）	ERRF	ERRIE	ERRC

7.6 CRS 寄存器

有关寄存器说明中使用的缩写，请参见参考手册中的*第 72 页的第 1.1 节*。

外设寄存器可按字（32 位）进行访问。

7.6.1 CRS 控制寄存器 (CRS_CR)

CRS control register

偏移地址：0x00

复位值：0x0000 4000

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.	TRIM[6:0]							SW SYNC	AUTO TRIMEN	CEN	Res.	ESYNC IE	ERRIE	SYNC WARNIE	SYNC OKIE
	rw	rw	rw	rw	rw	rw	rw	rt_w	rw	rw		rw	rw	rw	rw

位 31:15 保留，必须保持复位值。

位 14:8 **TRIM[6:0]**：HSI48 振荡器平滑微调 (HSI48 oscillator smooth trimming)

这些位为 HSI48 振荡器提供用户可编程的微调值。可通过编程使其适应电压和温度的差异，使 HSI48 振荡器的频率更为准确。

默认值为 64，该值对应于微调间隔的中间值。微调步在产品数据手册中指定。较高的 TRIM 值对应于较高的输出频率。

当 AUTOTRIMEN 位置 1 时，该字段由硬件控制并且是只读的。

位 7 **SWSYNC**：生成软件 SYNC 事件 (Generate software SYNC event)

此位由软件置 1 以生成软件 SYNC 事件。它由硬件自动清零。

0：不执行任何操作。

1：生成软件 SYNC 事件。

位 6 **AUTOTRIMEN**：自动微调使能 (Automatic trimming enable)

此位根据在两个 SYNC 事件间测得的频率误差使能 TRIM 位的自动硬件调整。如果此位置 1，则 TRIM 位是只读的。TRIM 值可通过硬件一次调整一步或两步，具体取决于测得的频率误差值。更多详细信息，请参见*第 7.3.4 节：频率误差评估和自动微调*。

0：禁止自动微调，TRIM 位可由用户调整。

1：使能自动微调，TRIM 位是只读的且由硬件控制。

位 5 **CEN**：频率误差计数器使能 (Frequency error counter enable)

此位为频率误差计数器使能振荡器时钟。

0：禁止频率误差计数器

1：使能频率误差计数器

当此位置 1 时，CRS_CFGR 寄存器被写保护，无法修改。

位 4 保留，必须保持复位值。

位 3 **ESYNCIE**：预期 SYNC 中断使能 (Expected SYNC interrupt enable)
0：禁止预期 SYNC (ESYNCF) 中断
1：使能预期 SYNC (ESYNCF) 中断

位 2 **ERRIE**：同步或微调错误中断使能 (Synchronization or trimming error interrupt enable)
0：禁止同步或微调错误 (ERRF) 中断
1：使能同步或微调错误 (ERRF) 中断

位 1 **SYNCWARNIE**：SYNC 警告中断使能 (SYNC warning interrupt enable)
0：禁止 SYNC 警告 (SYNCWARNF) 中断
1：使能 SYNC 警告 (SYNCWARNF) 中断

位 0 **SYNCOKIE**：SYNC 事件正常中断使能 (SYNC event OK interrupt enable)
0：禁止 SYNC 事件正常 (SYNCOKF) 中断
1：使能 SYNC 事件正常 (SYNCOKF) 中断

7.6.2 CRS 配置寄存器 (CRS_CFGR)

CRS configuration register

只有禁止频率误差计数器时，才可写入该寄存器（CRS_CR 中的 CEN 位清零）。当计数器使能时，该寄存器被写保护。

偏移地址：0x04

复位值：0x2022 BB7F

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SYNCPOL	Res.	SYNCSRC[1:0]		Res.	SYNCDIV[2:0]			FELIM[7:0]							
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
RELOAD[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31 **SYNCPOL**：SYNC 极性选择 (SYNC polarity selection)
此位由软件置 1 和清零，以选择 SYNC 信号源的输入极性。
0：SYNC 在上升沿有效（默认）
1：SYNC 在下降沿有效

位 30 保留，必须保持复位值。

位 29:28 **SYNCSRC[1:0]**：SYNC 信号源选择 (SYNC signal source selection)
这些位由软件置 1 和清零，以选择 SYNC 信号源。
00：GPIO 选择为 SYNC 信号源
01：LSE 选择为 SYNC 信号源
10：USB SOF 选择为 SYNC 信号源（默认）
11：保留

注： *当使用 USB LPM（链路层电源管理）且器件处于睡眠模式时，周期性 USB SOF 不会由主机产生。因此，不会向 CRS 提供任何 SYNC 信号来校准运行状态下的 HSI48 振荡器。要确保从睡眠模式唤醒后所需的时钟精度，LSE 或 GPIO 上的参考时钟应用作 SYNC 信号。*

位 27 保留，必须保持复位值。

位 26:24 **SYNCDIV[2:0]**：SYNC 分频器 (SYNC divider)

这些位由软件置 1 和清零，以控制 SYNC 信号的分频系数。

000：SYNC 不进行分频（默认）

001：SYNC 2 分频

010：SYNC 4 分频

011：SYNC 8 分频

100：SYNC 16 分频

101：SYNC 32 分频

110：SYNC 64 分频

111：SYNC 128 分频

位 23:16 **FELIM[7:0]**：频率误差限值 (Frequency error limit)

FELIM 包含用于评估 CRS_ISR 寄存器的 FECAP[15:0] 位中锁存的已捕捉频率误差值的值。

有关 FECAP 评估的更多详细信息，请参见*第 7.3.4 节：频率误差评估和自动微调*。

位 15:0 **RELOAD[15:0]**：计数器重载值 (Counter reload value)

RELOAD 是每次发生 SYNC 事件时要装载到频率误差计数器中的值。

有关计数器行为的更多详细信息，请参见*第 7.3.3 节：频率误差测量*。

7.6.3 CRS 中断和状态寄存器 (CRS_ISR)

CRS interrupt and 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
FECAP[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FEDIR	Res.	Res.	Res.	Res.	TRIM OVF	SYNC MISS	SYNC ERR	Res.	Res.	Res.	Res.	ESYNCF	ERRF	SYNC WARNF	SYNC OKF
r					r	r	r					r	r	r	r

位 31:16 **FECAP[15:0]**：频率误差捕捉 (Frequency error capture)

FECAP 是最后一次 SYNC 事件发生时锁存的频率误差计数器值。

有关 FECAP 使用的更多详细信息，请参见*第 7.3.4 节：频率误差评估和自动微调*。

位 15 **FEDIR**：频率误差方向 (Frequency error direction)

FEDIR 是最后一次 SYNC 事件发生时锁存的频率误差计数器的计数方向。它显示实际频率低于目标频率还是高于目标频率。

0：递增计数方向，实际频率高于目标频率。

1：递减计数方向，实际频率低于目标频率。

位 14:11 保留，必须保持复位值。

位 10 **TRIMOVF**：微调上溢或下溢 (Trimming overflow or underflow)

当自动微调尝试使 TRIM 值上溢或下溢时，该标志由硬件置 1。如果 CRS_CR 寄存器中的 ERRIE 位置 1，将生成中断。此位用软件清零，方法是将 CRS_ICR 寄存器中的 ERRC 位置 1。

0：未发出微调错误信号

1：已发出微调错误信号

位 9 **SYNCMISS**：SYNC 丢失 (SYNC missed)

当频率误差计数器达到值 FELIM * 128 且未检测到 SYNC 时，该标志由硬件置 1，这意味着 SYNC 脉冲丢失或频率误差过大（内部频率过高）而无法通过调整 TRIM 值来补偿，应采取其他措施。此时，频率误差计数器停止（等待下一个 SYNC），并且会在 CRS_CR 寄存器中的 ERRIE 位置 1 时生成中断。此位用软件清零，方法是将 CRS_ICR 寄存器中的 ERRC 位置 1。

0：未发出 SYNC 丢失错误信号

1：已发出 SYNC 丢失错误信号

位 8 **SYNCERR**：SYNC 错误 (SYNC error)

当 SYNC 脉冲在 ESYNC 事件之前达到且所测得的频率误差大于或等于 FELIM * 128 时，该标志由硬件置 1。这意味着频率误差过大（内部频率过低）而无法通过调整 TRIM 值进行补偿，应采取其他措施。如果 CRS_CR 寄存器中的 ERRIE 位置 1，将生成中断。此位用软件清零，方法是将 CRS_ICR 寄存器中的 ERRC 位置 1。

0：未发出 SYNC 错误信号

1：已发出 SYNC 错误信号

位 7:4 保留，必须保持复位值。

位 3 **ESYNCF**：预期 SYNC 标志 (Expected SYNC flag)

当频率误差计数器达到零值时，此标志由硬件置 1。如果 CRS_CR 寄存器中的 ESYNCIE 位置 1，将生成中断。此位用软件清零，方法是将 CRS_ICR 寄存器中的 ESYNCC 位置 1。

0：未发出预期 SYNC 信号

1：已发出预期 SYNC 信号

位 2 **ERRF**：ERRF 错误标志 (Error flag)

发生同步或微调错误时，该标志由硬件置 1。它是 TRIMOVF、SYNCMISS 和 SYNCERR 的逻辑或运算的结果。如果 CRS_CR 寄存器中的 ERRIE 位置 1，将生成中断。此位用软件清零，方法是将 CRS_ICR 寄存器中的 ERRC 位置 1，这将清零 TRIMOVF、SYNCMISS 和 SYNCERR 位。

0：未发出同步或微调错误信号

1：已发出同步或微调错误信号

位 1 **SYNCWARNF**：SYNC 警告标志 (SYNC warning flag)

当所测得的频率误差大于或等于 FELIM * 3 但小于 FELIM * 128 时，该标志由硬件置 1。这意味着要补偿频率误差，必须通过两步或多步来调整 TRIM 值。如果 CRS_CR 寄存器中的 SYNCWARNIE 位置 1，将生成中断。此位用软件清零，方法是将 CRS_ICR 寄存器中的 SYNCWARNC 位置 1。

0：未发出 SYNC 警告信号

1：已发出 SYNC 警告信号

位 0 **SYNCOKF**：SYNC 事件正常标志 (SYNC event OK flag)

当所测得的频率误差小于 FELIM * 3 时，该标志由硬件置 1。这意味着无需调整 TRIM 值或只需一步微调便足以补偿频率误差。如果 CRS_CR 寄存器中的 SYNCOKIE 位置 1，将生成中断。此位用软件清零，方法是将 CRS_ICR 寄存器中的 SYNCOKC 位置 1。

0：未发出 SYNC 事件正常信号

1：已发出 SYNC 事件正常信号

7.6.4 CRS 中断标志清零寄存器 (CRS_ICR)

CRS 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
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.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ESYNCC	ERRC	SYNC WARNC	SYNC OKC
												rw	rw	rw	rw

位 31:4 保留，必须保持复位值。

位 3 **ESYNCC**：预期 SYNC 清零标志 (Expected SYNC clear flag)
将 1 写入此位时，CRS_ISR 寄存器中 ESYNCF 标志将清零。

位 2 **ERRC**：错误清零标志 (Error clear flag)
将 1 写入此位将清零 TRIMOVF、SYNCMISS 和 SYNCERR 位，进而会清零 CRS_ISR 寄存器中的 ERRF 标志。

位 1 **SYNCWARNC**：SYNC 警告清零标志 (SYNC warning clear flag)
将 1 写入此位时，CRS_ISR 寄存器中 SYNCWARNF 标志将清零。

位 0 **SYNCOKC**：SYNC 事件正常清零标志 (SYNC event OK clear flag)
将 1 写入此位时，CRS_ISR 寄存器中 SYNCOKF 标志将清零。

7.6.5 CRS 寄存器映射

表 36. CRS 寄存器映射和复位值

偏移	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	**CRS_CR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TRIM[6:0]							SWSYNC	AUTOTRIMEN	CEN	Res.	ESYNCIE	ERRIE	SYNCWARNIE	SYNCOKIE
	Reset value																		1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x04	**CRS_CFGR**	SYNCPOL	Res.	SYNC SRC [1:0]		Res.	SYNC DIV [2:0]			FELIM[7:0]								RELOAD[15:0]															
	Reset value	0		1	0		0	0	0	0	0	1	0	0	0	1	0	1	0	1	1	1	0	1	1	0	1	1	1	1	1	1	1
0x08	**CRS_ISR**	FECAP[15:0]																FEDIR	Res.	Res.	Res.	Res.	TRIMOVF	SYNCMISS	SYNCERR	Res.	Res.	Res.	Res.	ESYNCF	ERRF	SYNCWARNF	SYNCOKF
	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
0x0C	**CRS_ICR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ESYNCC	ERRC	SYNCWARNC	SYNCOKC
	Reset value																													0	0	0	0

有关寄存器边界地址的信息，请参见*第 78 页的第 2.2.2 节*。

8 通用 I/O (GPIO)

8.1 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）和 1 个 32 位置位/复位寄存器 (GPIOx_BSRR)。此外，所有 GPIO 都包括 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）。

8.2 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 端口配置
- 模拟功能
- 复用功能选择寄存器
- 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

8.3 GPIO 功能描述

根据数据手册中列出的每个 I/O 端口的特性，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
- 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 寄存器和 GPIOx_BRR 寄存器旨在实现对 GPIOx_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 21 和*图 22* 分别显示了标准和 5 V 容限 I/O 端口位的基本结构。*表 37* 给出了可能的端口位配置方案。

图 21. I/O 端口位的基本结构

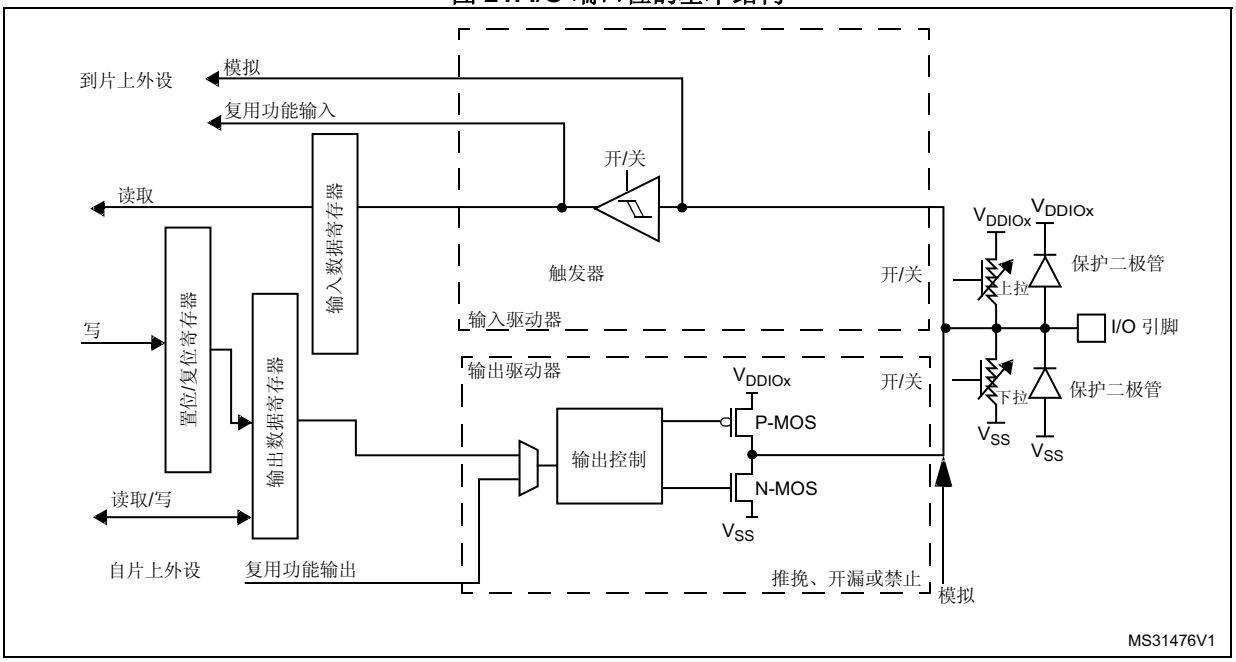
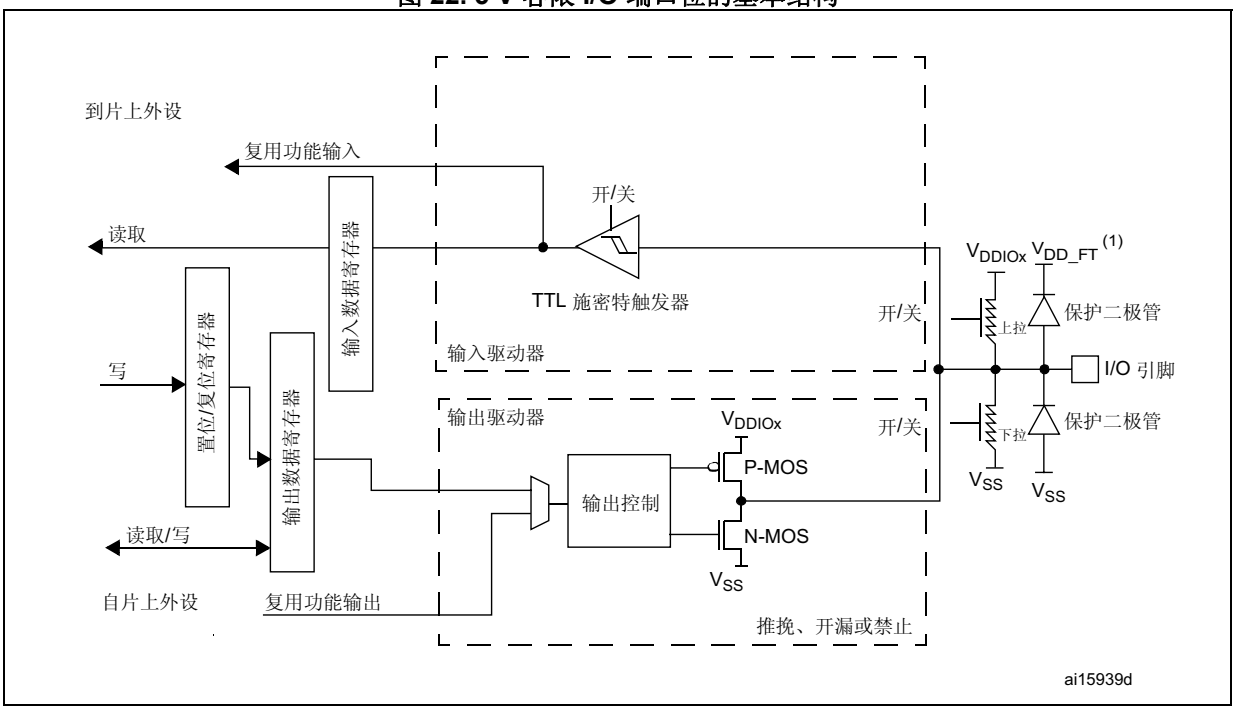


图 22. 5 V 容限 I/O 端口位的基本结构



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

表 37. 端口位配置表(1)

MODE(i) [1:0]	OTYPER(i)	OSPEED(i) [1:0]		PUPD(i) [1:0]		I/O 配置	
01	0	SPEED [1:0]		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 [1:0]		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	
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。

8.3.1 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，大多数 I/O 端口被配置为模拟模式。

复位后，调试引脚处于复用功能上拉/下拉状态：

- PA15：JTDI 处于上拉状态
- PA14：JTCK/SWCLK 处于下拉状态
- PA13：JTMS/SWDAT 处于上拉状态
- PB4：NJTRST 处于上拉状态
- PB3：JTDO 处于浮空状态，未处于上拉或下拉状态

PH3/BOOT0 在复位期间处于输入模式，并至少持续到选项字节加载阶段结束。请参见*第 8.3.15 节：将 PH3 用作 GPIO*。

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（仅驱动低电平，高电平为高阻态）。

输入数据寄存器 (GPIOx_IDR) 每个 AHB 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。

8.3.2 I/O 引脚复用功能复用器和映射

器件 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，该复用器采用多达 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：

- 复位后，复用器选择为复用功能 0 (AF0)。在复用模式下通过 GPIOx_MODER 寄存器配置 I/O。
- 器件数据手册中详细说明了每个引脚的特定复用功能分配。

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。

要在指定配置下使用 I/O，用户必须按照以下步骤操作：

- **调试功能：**每个器件复位后，立即将这些引脚分配为可由调试主机使用的复用功能引脚。
- **GPIO：**在 GPIOx_MODER 寄存器中将所需 I/O 配置为输出、输入或模拟通道。
- **外设复用功能：**
  - 在 GPIOx_AFRL 或 GPIOx_AFRH 寄存器中，将 I/O 连接到所需的 AFx。
  - 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_MODER 寄存器中将所需 I/O 配置为复用功能。
- **其他功能：**
  - 对于 ADC、DAC、OPAMP 和 COMP，在 GPIOx_MODER 寄存器中将所需 I/O 配置为模拟模式，并在 ADC、DAC、OPAMP 和 COMP 寄存器中配置所需功能。
  - 对于 RTC、WKUPx 和振荡器的其他功能，在相关的 RTC、PWR 和 RCC 寄存器中配置所需功能。这些功能优先于标准 GPIO 寄存器中的配置。

有关复用功能 I/O 引脚映射的详细信息，请参见器件数据手册中的“复用功能映射”表。

8.3.3 I/O 端口控制寄存器

每个 GPIO 端口有 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 模式（输入、输出、AF、模拟）。GPIOx_OTYPER 和 GPIOx_OSPEEDR 寄存器用于选择输出类型（推挽或开漏）和速度。无论采用哪种 I/O 方向，GPIOx_PUPDR 寄存器都用于选择上拉/下拉。

8.3.4 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。

有关寄存器说明的详细信息，请参见*第 8.4.5 节：GPIO 端口输入数据寄存器 (GPIOx_IDR) (x = A..I)* 和*第 8.4.6 节：GPIO 端口输出数据寄存器 (GPIOx_ODR) (x = A..I)*。

8.3.5 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：BS(i) 和 BR(i)。当写入 1 时，BS(i) 位会**置位**对应的 ODR(i) 位。当写入 1 时，BR(i) 位会**复位** 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复位操作，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 AHB 写访问中，可以修改一个或多个位。

8.3.6 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 MCU 复位或外设复位之前，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 8.4.8 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。

有关详细信息，请参见*第 8.4.8 节：GPIO 端口配置锁定寄存器 (GPIOx_LCKR) (x = A..I)* 中的 LCKR 寄存器说明。

8.3.7 I/O 复用功能输入/输出

有两个寄存器可用来从每个 I/O 可用的复用功能输入/输出中进行选择。借助这些寄存器，用户可根据应用程序的要求将某个复用功能连接到其他某个引脚。

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，所以只需为指定 I/O 的复用功能输入/输出选择一个通道即可。

要了解在每个 GPIO 引脚上复用了哪些功能，请参见器件数据手册。

8.3.8 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，可将端口配置为输入、输出或复用功能模式（端口不能配置为模拟模式）。请参见*第 16 节：扩展中断和事件控制器 (EXTI)* 和*第 16.3.2 节：唤醒事件管理*。

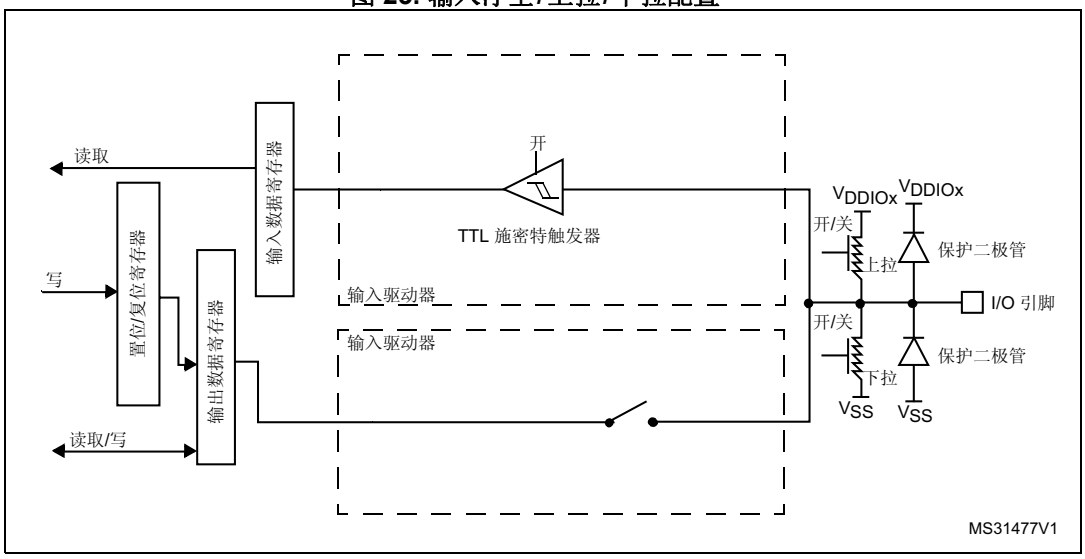
8.3.9 输入配置

对 I/O 端口进行编程作为输入时：

- 输出缓冲器被禁止
- 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
- 输入数据寄存器每个 AHB 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 23 说明了 I/O 端口位的输入配置。

图 23. 输入浮空/上拉/下拉配置



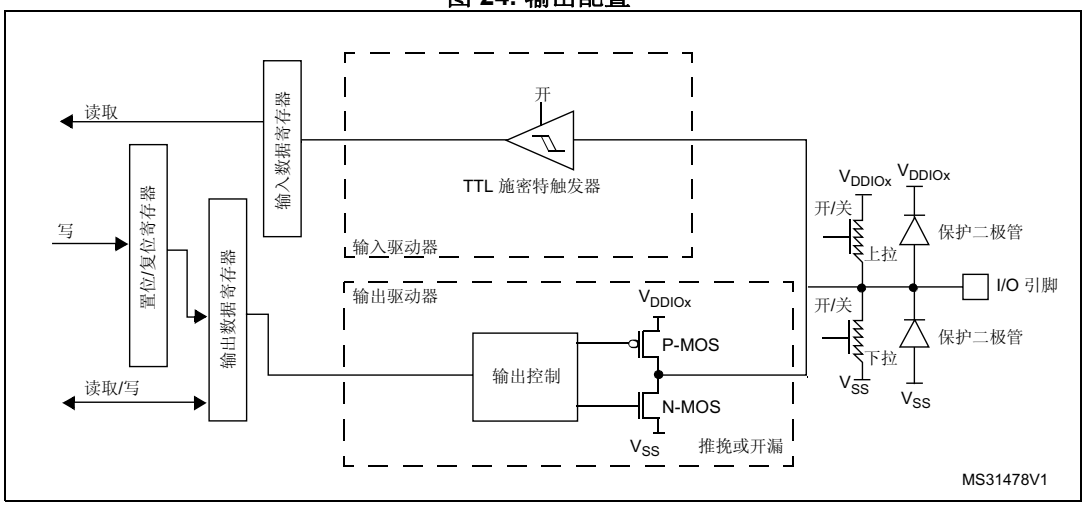
8.3.10 输出配置

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”会使端口保持高阻态 (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上拉和下拉电阻
- 输入数据寄存器每个 AHB 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 24 说明了 I/O 端口位的输出配置。

图 24. 输出配置



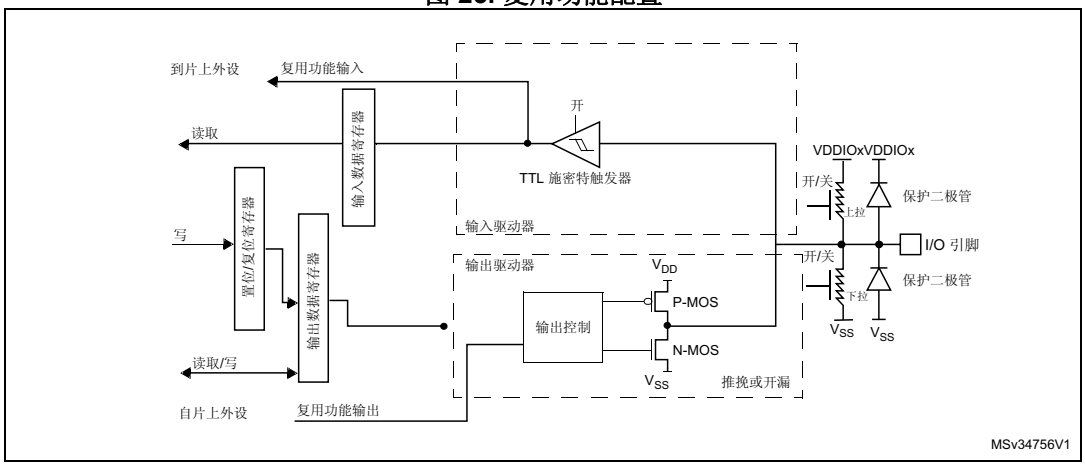
8.3.11 复用功能配置

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模式
- 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
- 输入数据寄存器每个 AHB 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 25 说明了 I/O 端口位的复用功能配置。

图 25. 复用功能配置



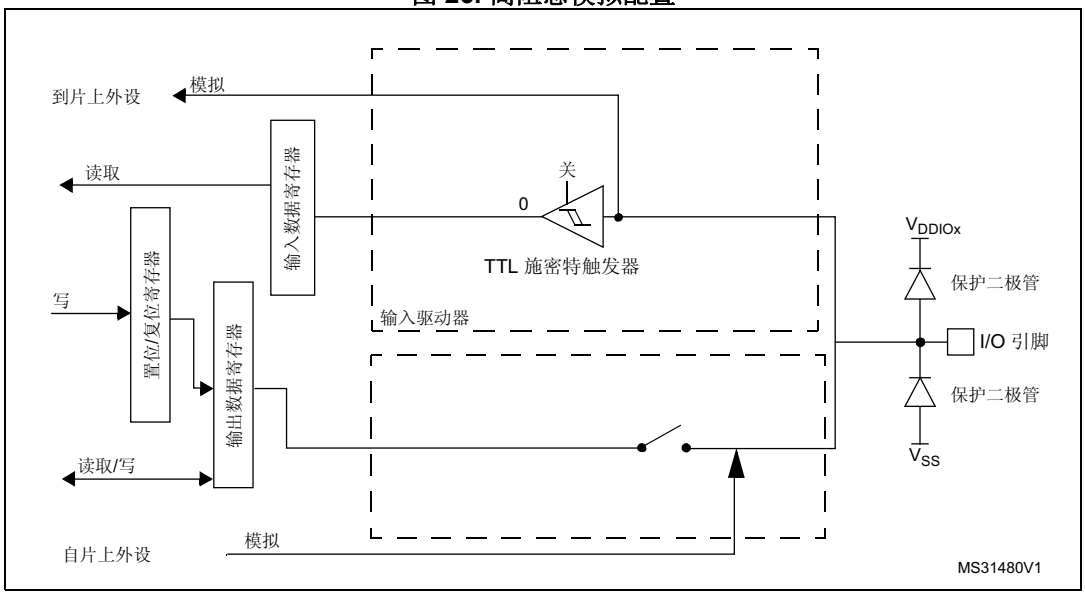
8.3.12 模拟配置

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硬件关闭
- 对输入数据寄存器的读访问值为“0”

图 26 说明了 I/O 端口位的高阻态模拟输入配置。

图 26. 高阻态模拟配置



8.3.13 将 HSE 或 LSE 振荡器引脚用作 GPIO

当 HSE 或 LSE 振荡器关闭（复位后的默认状态）时，可将相关的振荡器引脚用作常规 GPIO。

当 HSE 或 LSE 振荡器打开（通过将 RCC_CSR 寄存器中的 HSEON 或 LSEON 位置 1）时，振荡器会控制与其相关联的引脚，这些引脚的 GPIO 配置不起作用。

将振荡器配置为用户外部时钟模式时，仅为时钟输入保留引脚，OSC_OUT 或 OSC32_OUT 引脚仍可用作常规 GPIO。

8.3.14 在 RTC 电源域中使用 GPIO 引脚

当内核电源域掉电时（器件进入待机模式时），PC13/PC14/PC15 GPIO 功能会丢失。在这种情况下，如果不通过 RTC 配置旁路其 GPIO 配置，这些引脚将被设置为模拟输入模式。

有关 RTC 的 I/O 控制的详细信息，请参见*第 42.3 节：RTC 功能说明*。

8.3.15 将 PH3 用作 GPIO

PH3 可用作自举引脚 (BOOT0) 或 GPIO。它可从输入模式切换到模拟输入模式，具体取决于用户选项字节中的 nSWBOOT0 位：

- 如果 nSWBOOT0 = 1，则在选项字节加载之后切换。
- 如果 nSWBOOT0 = 0，则在复位之后切换。

8.4 GPIO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 38*。

可按字、半字或字节模式写入外设寄存器。

8.4.1 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：

- 0xABFF FFFF（端口 A）
- 0xFFFF FEBF（端口 B）
- 0xFFFF FFFF（端口 C..G 和 I）
- 0x0000 000F（端口 H）

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MODE15[1:0]		MODE14[1:0]		MODE13[1:0]		MODE12[1:0]		MODE11[1:0]		MODE10[1:0]		MODE9[1:0]		MODE8[1:0]	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MODE7[1:0]		MODE6[1:0]		MODE5[1:0]		MODE4[1:0]		MODE3[1:0]		MODE2[1:0]		MODE1[1:0]		MODE0[1:0]	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 2y+1:2y **MODEy[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 模式。
00：输入模式
01：通用输出模式
10：复用功能模式
11：模拟模式（复位状态）

8.4.2 GPIO 端口输出类型寄存器 (GPIOx_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
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
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**：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 输出类型。
0：推挽输出（复位状态）
1：开漏输出

8.4.3 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：

- 0x0C00 0000（端口 A）
- 0x0000 0000（其他端口）

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
OSPEED15 [1:0]		OSPEED14 [1:0]		OSPEED13 [1:0]		OSPEED12 [1:0]		OSPEED11 [1:0]		OSPEED10 [1:0]		OSPEED9 [1:0]		OSPEED8 [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
OSPEED7 [1:0]		OSPEED6 [1:0]		OSPEED5 [1:0]		OSPEED4 [1:0]		OSPEED3 [1:0]		OSPEED2 [1:0]		OSPEED1 [1:0]		OSPEED0 [1:0]	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 2y+1:2y **OSPEEDy[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。
00：低速
01：中速
10：高速
11：超高速
注： 关于每种速度的频率规范以及电源和负载条件，请参见器件数据手册。

8.4.4 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（其他端口）

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

位 2y+1:2y **PUPDy[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

8.4.5 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

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
ID15	ID14	ID13	ID12	ID11	ID10	ID9	ID8	ID7	ID6	ID5	ID4	ID3	ID2	ID1	ID0
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:16 保留，必须保持复位值。

位 15:0 **IDy**：端口输入数据位 (Port input data bit) (y = 0..15)
这些位为只读。它们包含相应 I/O 端口的输入值。

8.4.6 GPIO 端口输出数据寄存器 (GPIOx_ODR) (x = A..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
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
OD15	OD14	OD13	OD12	OD11	OD10	OD9	OD8	OD7	OD6	OD5	OD4	OD3	OD2	OD1	OD0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:16 保留，必须保持复位值。

位 15:0 **ODy**：端口输出数据位 (Port output data bit) (y = 0..15)
这些位可通过软件读取和写入。

注： *对于原子位置 1/复位，通过写入 GPIOx_BSRR 或 GPIOx_BRR 寄存器，可分别置 1 和/或复位 OD 位 (x = A..F)。*

8.4.7 GPIO 端口位置 1/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。读取这些位可返回值 0x0000。
0：不会对相应的 ODx 位执行任何操作
1：复位相应的 ODx 位
注： 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。读取这些位可返回值 0x0000。
0：不会对相应的 ODx 位执行任何操作
1：将相应的 ODx 位置 1

8.4.8 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 MCU 复位或外设复位之前，将无法对该端口位的值进行修改。

注： 可使用特定的写序列对 GPIOx_LCKR 寄存器执行写操作。在此锁定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

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.	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**：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。GPIOx_LCKR 寄存器被锁定，直到下一次 MCU 复位或外设复位。

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： 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 MCU 复位或外设复位为止。

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

8.4.9 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
AFSEL7[3:0]				AFSEL6[3:0]				AFSEL5[3:0]				AFSEL4[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
AFSEL3[3:0]				AFSEL2[3:0]				AFSEL1[3:0]				AFSEL0[3:0]			
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:0 **AFSELy[3:0]**：端口 x 引脚 y 的复用功能选择 (Alternate function selection for port x pin y) (y = 0..7)

这些位通过软件写入，用于配置复用功能 I/O。

AFSELy 选择：

0000: AF0
0001: AF1
0010: AF2
0011: AF3
0100: AF4
0101: AF5
0110: AF6
0111: AF7
1000: AF8
1001: AF9
1010: AF10
1011: AF11
1100: AF12
1101: AF13
1110: AF14
1111: AF15

8.4.10 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
AFSEL15[3:0]				AFSEL14[3:0]				AFSEL13[3:0]				AFSEL12[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
AFSEL11[3:0]				AFSEL10[3:0]				AFSEL9[3:0]				AFSEL8[3:0]			
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:0 **AFSELy[3:0]**：端口 x 引脚 y 的复用功能选择 (Alternate function selection for port x pin y) (y = 8..15)

这些位通过软件写入，用于配置复用功能 I/O。

AFSELy 选择：

0000：AF0
0001：AF1
0010：AF2
0011：AF3
0100：AF4
0101：AF5
0110：AF6
0111：AF7
1000：AF8
1001：AF9
1010：AF10
1011：AF11
1100：AF12
1101：AF13
1110：AF14
1111：AF15

8.4.11 GPIO 端口位复位寄存器 (GPIOx_BRR) (x = A..I)

GPIO port bit reset register

偏移地址：0x28
复位值：0x0000 0000

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

位 31:16 保留

位 15:0 **BRy**：端口 x 复位位 y (Port x Reset bit y) (y = 0..15)

这些位为只写。读取这些位可返回值 0x0000

0：不会对相应的 ODx 位执行任何操作
1：复位相应的 ODx 位

8.4.12 GPIO 寄存器映射

下表提供了 GPIO 寄存器映射和复位值。

表 38. GPIO 寄存器映射和复位值

偏移	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	**GPIOA_MODER**	MODE15[1:0]		MODE14[1:0]		MODE13[1:0]		MODE12[1:0]		MODE11[1:0]		MODE10[1:0]		MODE9[1:0]		MODE8[1:0]		MODE7[1:0]		MODE6[1:0]		MODE5[1:0]		MODE4[1:0]		MODE3[1:0]		MODE2[1:0]		MODE1[1:0]		MODE0[1:0]	
	Reset value	1	0	1	0	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0x00	**GPIOB_MODER**	MODE15[1:0]		MODE14[1:0]		MODE13[1:0]		MODE12[1:0]		MODE11[1:0]		MODE10[1:0]		MODE9[1:0]		MODE8[1:0]		MODE7[1:0]		MODE6[1:0]		MODE5[1:0]		MODE4[1:0]		MODE3[1:0]		MODE2[1:0]		MODE1[1:0]		MODE0[1:0]	
	Reset value	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	0	1	1	1	1	1	1
0x00	**GPIOx_MODER** (where x = C..I)	MODE15[1:0]		MODE14[1:0]		MODE13[1:0]		MODE12[1:0]		MODE11[1:0]		MODE10[1:0]		MODE9[1:0]		MODE8[1:0]		MODE7[1:0]		MODE6[1:0]		MODE5[1:0]		MODE4[1:0]		MODE3[1:0]		MODE2[1:0]		MODE1[1:0]		MODE0[1:0]	
	Reset value	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0x04	**GPIOx_OTYPER** (where x = A..I)	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	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	**GPIOA_OSPEEDR**	OSPEED15[1:0]		OSPEED14[1:0]		OSPEED13[1:0]		OSPEED12[1:0]		OSPEED11[1:0]		OSPEED10[1:0]		OSPEED9[1:0]		OSPEED8[1:0]		OSPEED7[1:0]		OSPEED6[1:0]		OSPEED5[1:0]		OSPEED4[1:0]		OSPEED3[1:0]		OSPEED2[1:0]		OSPEED1[1:0]		OSPEED0[1:0]	
	Reset value	0	0	0	0	1	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
0x08	**GPIOx_OSPEEDR** (where x = B..H)	OSPEED15[1:0]		OSPEED14[1:0]		OSPEED13[1:0]		OSPEED12[1:0]		OSPEED11[1:0]		OSPEED10[1:0]		OSPEED9[1:0]		OSPEED8[1:0]		OSPEED7[1:0]		OSPEED6[1:0]		OSPEED5[1:0]		OSPEED4[1:0]		OSPEED3[1:0]		OSPEED2[1:0]		OSPEED1[1:0]		OSPEED0[1:0]	
	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	**GPIOA_PUPDR**	PUPD15[1:0]		PUPD14[1:0]		PUPD13[1:0]		PUPD12[1:0]		PUPD11[1:0]		PUPD10[1:0]		PUPD9[1:0]		PUPD8[1:0]		PUPD7[1:0]		PUPD6[1:0]		PUPD5[1:0]		PUPD4[1:0]		PUPD3[1:0]		PUPD2[1:0]		PUPD1[1:0]		PUPD0[1:0]	
	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**	PUPD15[1:0]		PUPD14[1:0]		PUPD13[1:0]		PUPD12[1:0]		PUPD11[1:0]		PUPD10[1:0]		PUPD9[1:0]		PUPD8[1:0]		PUPD7[1:0]		PUPD6[1:0]		PUPD5[1:0]		PUPD4[1:0]		PUPD3[1:0]		PUPD2[1:0]		PUPD1[1:0]		PUPD0[1:0]	
	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)	PUPD15[1:0]		PUPD14[1:0]		PUPD13[1:0]		PUPD12[1:0]		PUPD11[1:0]		PUPD10[1:0]		PUPD9[1:0]		PUPD8[1:0]		PUPD7[1:0]		PUPD6[1:0]		PUPD5[1:0]		PUPD4[1:0]		PUPD3[1:0]		PUPD2[1:0]		PUPD1[1:0]		PUPD0[1:0]	
	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)	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	ID15	ID14	ID13	ID12	ID11	ID10	ID9	ID8	ID7	ID6	ID5	ID4	ID3	ID2	ID1	ID0
	Reset value																	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

表 38. GPIO 寄存器映射和复位值（续）

偏移	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
0x14	**GPIOx_ODR** (where x = A..I)	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OD15	OD14	OD13	OD12	OD11	OD10	OD9	OD8	OD7	OD6	OD5	OD4	OD3	OD2	OD1	OD0
	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)	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	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)	AFSEL7[3:0]				AFSEL6[3:0]				AFSEL5[3:0]				AFSEL4[3:0]				AFSEL3[3:0]				AFSEL2[3:0]				AFSEL1[3:0]				AFSEL0[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)	AFSEL15[3:0]				AFSEL14[3:0]				AFSEL13[3:0]				AFSEL12[3:0]				AFSEL11[3:0]				AFSEL10[3:0]				AFSEL9[3:0]				AFSEL8[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
0x28	**GPIOx_BRR** (where x = A..I)	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BR15	BR14	BR13	BR12	BR11	BR10	BR9	BR8	BR7	BR6	BR5	BR4	BR3	BR2	BR1	BR0
	Reset value	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	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*第 2.2 节*。

9 系统配置控制器 (SYSCFG)

9.1 SYSCFG 主要特性

STM32L4Rxxx 和 STM32L4Sxxx 器件配有一组配置寄存器。系统配置控制器的主要用途如下：

- 重新映射存储区域
- 管理 GPIO 的外部中断线连接
- 管理稳健性功能
- 设置 SRAM2 写保护和软件擦除
- 配置 FPU 中断
- 使能防火墙
- 在一些 I/O 和升压器上使能/禁用 I^2C 超快速模式驱动功能，用于 I/O 模拟开关

9.2 SYSCFG 寄存器

9.2.1 SYSCFG 存储器重映射寄存器 (SYSCFG_MEMRMP)

SYSCFG memory remap register

此寄存器用于对存储器重映射进行配置。

偏移地址：0x00

复位值：0x0000 000X（X 是由 BOOT0 引脚和 BOOT1 选项位选择的存储器模式）

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	Res	Res	Res	Res	Res	FB_MODE	Res	Res	Res	Res	Res	MEM_MODE		
							rw						rw	rw	rw

位 31:9 保留，必须保持复位值。

位 8 **FB_MODE**：Flash 存储区模式选择 (Flash Bank mode selection)

0：Flash 存储区 1 映射到 0x0800 0000（并在 0x0000 0000(1) 有别名），Flash 存储区 2 映射到 0x0808 0000（并在 0x0008 0000 处有别名）

1：Flash 存储区 2 映射到 0x0800 0000（并在 0x0000 0000(1) 有别名），Flash 存储区 1 映射到 0x0808 0000（并在 0x0008 0000 有别名）

位 7:3 保留，必须保持复位值。

位 2:0 **MEM_MODE**：存储器映射选择 (Memory mapping selection）

这些位控制在地址 0x0000 0000 的存储器内部映射。它们用于通过软件选择物理重映射，从而旁路 BOOT 引脚和选项位设置。复位后，这些位将使用 BOOT0 引脚（或选项位，具体取决于 nSWBOOT0 选项位）和 BOOT1 选项位选择的值。

000：将主 Flash 映射到地址 0x00000000(1)

001：将系统 Flash 映射到地址 0x00000000

010：将 FMC 存储区 1（NOR/PSRAM 1 和 2）映射到地址 0x00000000

011：将 SRAM1 映射到地址 0x00000000

100：将 OCTOSPI1 存储器映射到地址 0x00000000

101：将 OCTOSPI2 存储器映射到地址 0x00000000

111：保留

1. BFB2 位置 1 后，系统存储器在 0x0000 0000 保持别名

注： *当把 FSMC 重映射到地址 0x0000 0000 时，只有 FSMC 的存储区 1* 的前两个区域（NOR/PSRAM 1 和 NOR/PSRAM 2）可被重映射到低端地址。在重映射模式下，CPU 可以通过 ICode 总线（而不是 System 总线）访问外部存储器来提高性能。

9.2.2 SYSCFG 配置寄存器 1 (SYSCFG_CFGR1)

SYSCFG configuration register 1

偏移地址：0x04

复位值：0x7C00 0001

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FPU_IE[5..0]						Res	Res	I2C4_FMP	I2C3_FMP	I2C2_FMP	I2C1_FMP	I2C_PB9_FMP	I2C_PB8_FMP	I2C_PB7_FMP	I2C_PB6_FMP
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	Res	Res	Res	ANASWVDD	BOOSTEN	Res	Res	Res	Res	Res	Res	Res	FWDIS
						rw	rw								rc_w0

位 31:26 **FPU_IE[5..0]**：浮点单元中断使能位 (Floating Point Unit interrupts enable bits)
FPU_IE[5]：不精确的中断使能
FPU_IE[4]：输入非常规中断使能
FPU_IE[3]：上溢中断使能
FPU_IE[2]：下溢中断使能
FPU_IE[1]：0 分频中断使能
FPU_IE[0]：无效操作中断使能

位 25:24 保留，必须保持复位值。

位 23 **I2C4_FMP**：超快速模式驱动功能激活 (Fast-mode Plus driving capability activation)
该位可在通过 AF 选择位选择的 I2C4 引脚上使能 Fm+ 驱动模式。
0：在通过 AF 选择位选择的 I2C4 引脚上未使能 Fm+ 模式。
1：在通过 AF 选择位选择的 I2C4 引脚上使能 Fm+ 模式。

位 22 **I2C3_FMP**：I2C3 超快速模式驱动功能激活 (I2C3 Fast-mode Plus driving capability activation)
该位可在通过 AF 选择位选择的 I2C3 引脚上使能 Fm+ 驱动模式。
0：在通过 AF 选择位选择的 I2C3 引脚上未使能 Fm+ 模式。
1：在通过 AF 选择位选择的 I2C3 引脚上使能 Fm+ 模式。

位 21 **I2C2_FMP**：I2C2 超快速模式驱动功能激活 (I2C2 Fast-mode Plus driving capability activation)
该位可在通过 AF 选择位选择的 I2C2 引脚上使能 Fm+ 驱动模式。
0：在通过 AF 选择位选择的 I2C2 引脚上未使能 Fm+ 模式。
1：在通过 AF 选择位选择的 I2C2 引脚上使能 Fm+ 模式。

位 20 **I2C1_FMP**：I2C1 超快速模式驱动功能激活 (I2C1 Fast-mode Plus driving capability activation)
该位可在通过 AF 选择位选择的 I2C1 引脚上使能 Fm+ 驱动模式。
0：在通过 AF 选择位选择的 I2C1 引脚上未使能 Fm+ 模式。
1：在通过 AF 选择位选择的 I2C1 引脚上使能 Fm+ 模式。

位 19 **I2C_PB9_FMP**：PB9 上的超快速模式 (Fm+) 驱动功能激活 (Fast-mode Plus (Fm+) driving capability activation on PB9)
该位可针对 PB9 使能 Fm+ 驱动模式。
0：PB9 引脚在标准模式下工作。
1：在 PB9 引脚上使能 Fm+ 模式并旁路速度控制。

位 18 **I2C_PB8_FMP**：PB8 上的超快速模式 (Fm+) 驱动功能激活 (Fast-mode Plus (Fm+) driving capability activation on PB8)
该位可针对 PB8 使能 Fm+ 驱动模式。
0：PB8 引脚在标准模式下工作。
1：在 PB8 引脚上使能 Fm+ 模式并旁路速度控制。

位 17 **I2C_PB7_FMP**：PB7 上的超快速模式 (Fm+) 驱动功能激活 (Fast-mode Plus (Fm+) driving capability activation on PB7)
该位可针对 PB7 使能 Fm+ 驱动模式。
0：PB7 引脚在标准模式下工作。
1：在 PB7 引脚上使能 Fm+ 模式并旁路速度控制。

位 16 **I2C_PB6_FMP**：PB6 上的超快速模式 (Fm+) 驱动功能激活 (Fast-mode Plus (Fm+) driving capability activation on PB6)
该位可针对 PB6 使能 Fm+ 驱动模式。
0：PB6 引脚在标准模式下工作。
1：在 PB6 引脚上使能 Fm+ 模式并旁路速度控制。

位 15:10 保留，必须保持复位值。

位 9 **ANASWVDD**：GPIO 模拟开关控制电压选择 (GPIO analog switch control voltage selection)

0：I/O 模拟开关由 VDDA 或升压器（升压器开启时）供电。

1：I/O 模拟开关由 VDD 供电。

有关位 9 的设置，请参见*表 39*。

位 8 **BOOSTEN**：I/O 模拟开关升压器使能 (I/O analog switch voltage booster enable)

0：I/O 模拟开关由 V_{DDA} 电压供电。这是将 ADC 用于 V_{DDA} 高电压工作条件时的建议配置。

1：I/O 模拟开关由专用升压器供电（由 V_{DD} 供电）。这是将 ADC 用于 V_{DDA} 低电压工作条件时的建议配置。

位 7:1 保留，必须保持复位值。

位 0 **FWDIS**：防火墙禁止 (Firewall disable)

此位由软件清零，以根据防火墙配置对存储器段访问施加保护。使能后，防火墙无法由软件禁止。只能通过系统复位将此位置 1。

0：使能防火墙保护

1：禁止防火墙保护

表 39 介绍了根据电压设置，位 9 (ANASWVDD) 和位 8 (BOOSTEN) 置 1 和复位的条件。

表 39. BOOSTEN 和 ANASWVDD 置 1/复位

VDD	VDDA	BOOSTEN	ANASWVDD
-	> 2.4 V	0	0
> 2.4 V	< 2.4 V	0	1
< 2.4 V	< 2.4 V	1	0

9.2.3 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 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
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 15:12 **EXTI3[3:0]**：EXTI 3 配置位 (EXTI 3 configuration bits)
这些位通过软件写入，以选择 EXTI3 外部中断的源输入。
0000：PA[3] 引脚
0001：PB[3] 引脚
0010：PC[3] 引脚
0011：PD[3] 引脚
0100：PE[3] 引脚
0101：PF[3] 引脚
0110：PG[3] 引脚
0111：PH[3] 引脚
1000：PI[3] 引脚

位 11:8 **EXTI2[3:0]**：EXTI 2 配置位 (EXTI 2 configuration bits)
这些位通过软件写入，以选择 EXTI2 外部中断的源输入。
0000：PA[2] 引脚
0001：PB[2] 引脚
0010：PC[2] 引脚
0011：PD[2] 引脚
0100：PE[2] 引脚
0101：PF[2] 引脚
0110：PG[2] 引脚
0111：PH[2] 引脚
1000：PI[2] 引脚

位 7:4 **EXTI1[3:0]**：EXTI 1 配置位 (EXTI 1 configuration bits)
这些位通过软件写入，以选择 EXTI1 外部中断的源输入。
0000：PA[1] 引脚
0001：PB[1] 引脚
0010：PC[1] 引脚
0011：PD[1] 引脚
0100：PE[1] 引脚
0101：PF[1] 引脚
0110：PG[1] 引脚
0111：PH[1] 引脚
1000：PI[1] 引脚

位 3:0 **EXTI1[3:0]**：EXTI 0 配置位 (EXTI 0 configuration bits)
这些位通过软件写入，以选择 EXTI0 外部中断的源输入。
0000：PA[0] 引脚
0001：PB[0] 引脚
0010：PC[0] 引脚
0011：PD[0] 引脚
0100：PE[0] 引脚
0101：PF[0] 引脚
0110：PG[0] 引脚
0111：PH[0] 引脚
1000：PI[0] 引脚

9.2.4 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 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
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 15:12 **EXTI7[3:0]**：EXTI 7 配置位 (EXTI 7 configuration bits)

这些位通过软件写入，以选择 EXTI7 外部中断的源输入。

0000：PA[7] 引脚
0001：PB[7] 引脚
0010：PC[7] 引脚
0011：PD[7] 引脚
0100：PE[7] 引脚
0101：PF[7] 引脚
0110：PG[7] 引脚
0111：PH[7] 引脚
1000：PI[7] 引脚

位 11:8 **EXTI6[3:0]**：EXTI 6 配置位 (EXTI 6 configuration bits)

这些位通过软件写入，以选择 EXTI6 外部中断的源输入。

0000：PA[6] 引脚
0001：PB[6] 引脚
0010：PC[6] 引脚
0011：PD[6] 引脚
0100：PE[6] 引脚
0101：PF[6] 引脚
0110：PG[6] 引脚
0111：PH[6] 引脚
1000：PI[6] 引脚

位 7:4 **EXTI5[3:0]**：EXTI 5 配置位 (EXTI 5 configuration bits)
这些位通过软件写入，以选择 EXTI5 外部中断的源输入。
0000：PA[5] 引脚
0001：PB[5] 引脚
0010：PC[5] 引脚
0011：PD[5] 引脚
0100：PE[5] 引脚
0101：PF[5] 引脚
0110：PG[5] 引脚
0111：PH[5] 引脚
1000：PI[5] 引脚

位 3:0 **EXTI4[3:0]**：EXTI 4 配置位 (EXTI 4 configuration bits)
这些位通过软件写入，以选择 EXTI4 外部中断的源输入。
0000：PA[4] 引脚
0001：PB[4] 引脚
0010：PC[4] 引脚
0011：PD[4] 引脚
0100：PE[4] 引脚
0101：PF[4] 引脚
0110：PG[4] 引脚
0111：PH[4] 引脚
1000：PI[4] 引脚

注：　上述寄存器中提及的某些 I/O 引脚可能不适用于小型封装。

9.2.5 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 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
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 15:12 **EXTI11[3:0]**：EXTI 11 配置位 (EXTI 11 configuration bits)
这些位通过软件写入，以选择 EXTI11 外部中断的源输入。
0000：PA[11] 引脚
0001：PB[11] 引脚
0010：PC[11] 引脚
0011：PD[11] 引脚
0100：PE[11] 引脚
0101：PF[11] 引脚
0110：PG[11] 引脚
0111：PH[11] 引脚
1000：PI[11] 引脚

位 11:8 **EXTI10[3:0]**：EXTI 10 配置位 (EXTI 10 configuration bits)
这些位通过软件写入，以选择 EXTI10 外部中断的源输入。
0000：PA[10] 引脚
0001：PB[10] 引脚
0010：PC[10] 引脚
0011：PD[10] 引脚
0100：PE[10] 引脚
0101：PF[10] 引脚
0110：PG[10] 引脚
0111：PH[10] 引脚
1000：PI[10] 引脚

位 7:4 **EXTI9[3:0]**：EXTI 9 配置位 (EXTI 9 configuration bits)
这些位通过软件写入，以选择 EXTI9 外部中断的源输入。
0000：PA[9] 引脚
0001：PB[9] 引脚
0010：PC[9] 引脚
0011：PD[9] 引脚
0100：PE[9] 引脚
0101：PF[9] 引脚
0110：PG[9] 引脚
0111：PH[9] 引脚
1000：PI[9] 引脚

位 3:0 **EXTI8[3:0]**：EXTI 8 配置位 (EXTI 8 configuration bits)
这些位通过软件写入，以选择 EXTI8 外部中断的源输入。
0000：PA[8] 引脚
0001：PB[8] 引脚
0010：PC[8] 引脚
0011：PD[8] 引脚
0100：PE[8] 引脚
0101：PF[8] 引脚
0110：PG[8] 引脚
0111：PH[8] 引脚
1000：PI[8] 引脚

注：　　上述寄存器中提及的某些 I/O 引脚可能不适用于小型封装。

9.2.6 SYSCFG 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 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
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 15:12 **EXTI15[3:0]**：EXTI 15 配置位 (EXTI 15 configuration bits)
这些位通过软件写入，以选择 EXTI15 外部中断的源输入。
0000：PA[15] 引脚
0001：PB[15] 引脚
0010：PC[15] 引脚
0011：PD[15] 引脚
0100：PE[15] 引脚
0101：PF[15] 引脚
0110：PG[15] 引脚
0111：PH[15] 引脚
1000：保留

位 11:8 **EXTI14[3:0]**：EXTI 14 配置位 (EXTI 14 configuration bits)
这些位通过软件写入，以选择 EXTI14 外部中断的源输入。
0000：PA[14] 引脚
0001：PB[14] 引脚
0010：PC[14] 引脚
0011：PD[14] 引脚
0100：PE[14] 引脚
0101：PF[14] 引脚
0110：PG[14] 引脚
0111：PH[14] 引脚
1000：保留

位 7:4 **EXTI13[3:0]**：EXTI 13 配置位 (EXTI 13 configuration bits)
这些位通过软件写入，以选择 EXTI13 外部中断的源输入。
0000：PA[13] 引脚
0001：PB[13] 引脚
0010：PC[13] 引脚
0011：PD[13] 引脚
0100：PE[13] 引脚
0101：PF[13] 引脚
0110：PG[13] 引脚
0111：PH[13] 引脚
1000：保留

位 3:0 **EXTI12[3:0]**：EXTI 12 配置位 (EXTI 15 configuration bits)
这些位通过软件写入，以选择 EXTI12 外部中断的源输入。
0000：PA[12] 引脚
0001：PB[12] 引脚
0010：PC[12] 引脚
0011：PD[12] 引脚
0100：PE[12] 引脚
0101：PF[12] 引脚
0110：PG[12] 引脚
0111：PH[12] 引脚
1000：保留

注：　上述寄存器中提及的某些 I/O 引脚可能不适用于小型封装。

9.2.7 SYSCFG SRAM2 控制和状态寄存器 (SYSCFG_SCSR)

SYSCFG SRAM2 control and status register

偏移地址：0x18

系统复位值：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	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	SRAM2 BSY	SRAM2 ER
														r	rw

位 31:2 保留，必须保持复位值

位 1 **SRAM2BSY**：SRAM2 忙于擦除操作 (SRAM2 busy by erase operation)
0：未在执行 SRAM2 擦除操作。
1：正在进行 SRAM2 擦除操作。

位 0 **SRAM2ER**：SRAM2 擦除 (SRAM2 Erase)
将此位置 1 可启动硬件 SRAM2 擦除操作。SRAM2 擦除操作结束后，此位自动清零。
注：　此位受写保护：只有在 SYSCFG_SKR 寄存器中写入正确的密钥序列后，才可将此位置 1。

9.2.8 SYSCFG 配置寄存器 2 (SYSCFG_CFGR2)

SYSCFG configuration register 2

偏移地址：0x1C

系统复位值：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	Res	Res	Res	Res	Res	SPF	Res	Res	Res	Res	ECCL	PVDL	SPL	CLL
							rc_w1					rs	rs	rs	rs

位 31:9 保留，必须保持复位值

位 8 **SPF**：SRAM2 奇偶校验错误标志 (SRAM2 parity error flag)

检测到 SRAM2 奇偶校验错误时，该位由硬件置 1。通过软件写入“1”可将该位清零。

0：未检测到 SRAM2 奇偶校验错误

1：检测到 SRAM2 奇偶校验错误

位 7:4 保留，必须保持复位值

位 3 **ECCL**：ECC 锁定 (ECC Lock)

此位由软件置 1，只能通过系统复位清零。它可用于使能并锁定 TIM1/8/15/16/17 断路输入的 Flash ECC 错误连接。

0：ECC 错误与 TIM1/8/15/16/17 断路输入断开连接。

1：ECC 错误连接至 TIM1/8/15/16/17 断路输入。

位 2 **PVDL**：PVD 锁定使能位 (PVD lock enable bit)

此位由软件置 1，只能通过系统复位清零。它可用于使能并锁定 TIM1/8/15/16/17 断路输入的 PVD 连接以及 PWR_CR2 寄存器中的 PVDE 和 PLS[2:0]。

0：PVD 中断与 TIM1/8/15/16/17 断路输入断开连接。PVDE 和 PLS[2:0] 位可由应用程序设定。

1：PVD 中断连接至 TIM1/8/15/16/17 断路输入，PVDE 和 PLS[2:0] 位为只读位。

位 1 **SPL**：SRAM2 奇偶校验锁定位 (SRAM2 parity lock bit)

此位由软件置 1，只能通过系统复位清零。它可用于使能并锁定 TIM1/8/15/16/17 断路输入的 SRAM2 奇偶校验错误信号连接。

0：SRAM2 奇偶校验错误信号与 TIM1/8/15/16/17 断路输入断开连接

1：SRAM2 奇偶校验错误信号连接至 TIM1/8/15/16/17 断路输入

位 0 **CLL**：Cortex®-M4 LOCKUP (Hardfault) 输出使能位 (Cortex®-M4 LOCKUP (Hardfault) output enable bit)

此位由软件置 1，只能通过系统复位清零。它可用于使能并锁定 TIM1/8/15/16/17 断路输入的 Cortex®-M4 LOCKUP (Hardfault) 输出连接。

0：Cortex®-M4 LOCKUP 输出与 TIM1/8/15/16/17 断路输入断开连接

1：Cortex®-M4 LOCKUP 输出连接至 TIM1/8/15/16/17 断路输入

9.2.9 SYSCFG SRAM2 写保护寄存器 (SYSCFG_SWPR)

SYSCFG SRAM2 write protection register

偏移地址：0x20

系统复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
P31WP	P30WP	P29WP	P28WP	P27WP	P26WP	P25WP	P24WP	P23WP	P22WP	P21WP	P20WP	P19WP	P18WP	P17WP	P16WP
rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
P15WP	P14WP	P13WP	P12WP	P11WP	P10WP	P9WP	P8WP	P7WP	P6WP	P5WP	P4WP	P3WP	P2WP	P1WP	P0WP
rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs

位 31:0 **PxWP**（x = 0 到 31）：SRAM2 页 x 写保护 (SRAM2 page x write protection)
这些位由软件置 1，只能通过系统复位清零。
0：禁止 SRAM2 页 x 的写保护。
1：使能 SRAM2 页 x 的写保护。

9.2.10 SYSCFG SRAM2 密钥寄存器 (SYSCFG_SKR)

SYSCFG SRAM2 key register

偏移地址：0x24

系统复位值：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	Res	Res	Res	Res	Res	Res	KEY[7:0]							
								w	w	w	w	w	w	w	w

位 31:8 保留，必须保持复位值

位 7:0 **KEY[7:0]**：软件擦除的 SRAM2 写保护密钥 (SRAM2 write protection key for software erase)
要解锁 SYSCFG_CFGR2 寄存器中 SRAM2ER 位的写保护，需要执行以下步骤。
1. 向 Key[7:0] 写入“0xCA”
2. 向 Key[7:0] 写入“0x53”
写入一个错误的关键字会再次激活写保护。

9.2.11 SYSCFG SRAM2 写保护寄存器 2 (SYSCFG_SWPR2)

SYSCFG SRAM2 write protection register 2

偏移地址：0x28

系统复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
P63WP	P62WP	P61WP	P60WP	P59WP	P58WP	P57WP	P56WP	P55WP	P54WP	P53WP	P52WP	P51WP	P50WP	P49WP	P48WP
rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
P47WP	P46WP	P45WP	P44WP	P43WP	P42WP	P41WP	P40WP	P39WP	P38WP	P37WP	P36WP	P35WP	P34WP	P33WP	P32WP
rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs	rs

位 31:0 **PxWP**（x = 32 到 63）：SRAM2 页 x 写保护 (SRAM2 page x write protection)

这些位由软件置 1，只能通过系统复位清零。

0：禁止 SRAM2 页 x 的写保护。

1：使能 SRAM2 页 x 的写保护。

9.2.12 SYSCFG 寄存器映射

下表列出了 SYSCFG 寄存器映射和复位值。

表 40. SYSCFG 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	**SYSCFG_ MEMRMP**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FB_MODE	Res.	Res.	Res.	Res.	Res.	MEM_ MODE		
	Reset value																								0						x	x	x
0x04	**SYSCFG_CFGR1**	FPU_IE[5..0]						Res.	Res.	I2C4_FMP	I2C3_FMP	I2C2_FMP	I2C1_FMP	I2C_PB9_FMP	I2C_PB8_FMP	I2C_PB7_FMP	I2C_PB6_FMP	Res.	Res.	Res.	Res.	Res.	Res.	ANASWVDD	BOOSTEN	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FWDIS
	Reset value	0	1	1	1	1	1			0	0	0	0	0	0	0	0							0	0					0	0	0	1
0x08	**SYSCFG_EXTICR1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	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
0x0C	**SYSCFG_EXTICR2**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	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**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	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

表 40. SYSCFG 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
0x14	**SYSCFG_EXTICR4**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	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
0x18	**SYSCFG_SCSR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SRAM2BSY	SRAM2ER
	Reset value																															0	0
0x1C	**SYSCFG_CFGR2**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SPF	Res.	Res.	Res.	Res.	ECCL	PVDL	SPL	CLL
	Reset value																								0					0	0	0	0
0x20	**SYSCFG_SWPR**	P31WP	P30WP	P29WP	P28WP	P27WP	P26WP	P25WP	P24WP	P23WP	P22WP	P21WP	P20WP	P19WP	P18WP	P17WP	P16WP	P15WP	P14WP	P13WP	P12WP	P11WP	P10WP	P9WP	P8WP	P7WP	P6WP	P5WP	P4WP	P3WP	P2WP	P1WP	P0WP
	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	**SYSCFG_SKR**	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	KEY							
	Reset value																									0	0	0	0	0	0	0	0
0x29	**SYSCFG_SWPR2**	P63WP	P62WP	P61WP	P60WP	P59WP	P58WP	P57WP	P56WP	P55WP	P54WP	P53WP	P52WP	P51WP	P50WP	P49WP	P48WP	P47WP	P46WP	P45WP	P44WP	P43WP	P42WP	P41WP	P40WP	P39WP	P38WP	P37WP	P36WP	P35WP	P34WP	P33WP	P32WP
	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*第 78 页的第 2.2.2 节*。

10 外设互连矩阵

10.1 简介

多个外设间有直接连接。

这可以在外设间实现自动通信和/或同步，节省了 CPU 资源，进而降低电源功耗。

此外，这些硬件连接消除了软件延迟，允许设计可预测系统。

这些互连可以在运行、睡眠、低功耗运行和低睡眠、停止 0、停止 1 和停止 2 模式下工作，具体取决于外设。

10.2 连接汇总

表 41. STM32L4Rxxx 和 STM32L4Sxxx 外设互连矩阵(1) (2)

源	目标																				
	TIM1	TIM8	TIM2	TIM3	TIM4	TIM5	TIM6	TIM7	TIM15	TIM16	TIM17	LPTIM1	LPTIM2	ADC1	DFSDM1	OPAMP1	OPAMP2	DAC1	DAC2	COMP1	COMP2	IRTIM
TIM1	-	1	1	1	1	-	-	-	1	-	-	-	-	2	5	-	-	-	-	9	-	-
TIM8	-	-	1	-	1	1	-	-	-	-	-	-	-	2	5	-	-	4	4	-	9	-
TIM2	1	1	-	1	1	1	-	-	-	-	-	-	-	2	-	-	-	4	4	9	-	-
TIM3	1	-	1	-	1	1	-	-	1	-	-	-	-	2	5	-	-	-	-	9	9	-
TIM4	1	1	1	1	-	1	-	-	-	-	-	-	-	2	5	-	-	4	4	-	-	-
TIM5	-	1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	4	4	-	-	-
TIM6	-	-	-	-	-	-	-	-	-	-	-	-	-	2	5	-	-	4	4	-	-	-
TIM7	-	-	-	-	-	-	-	-	-	-	-	-	-	-	5	-	-	4	4	-	-	-
TIM15	1	-	-	1	-	-	-	-	-	-	-	-	-	2	-	-	-	-	-	-	9	-
TIM16	-	-	-	-	-	-	-	-	1	-	-	-	-	-	5	-	-	-	-	-	-	15
TIM17	-	-	-	-	-	-	-	-	1	-	-	-	-	-	-	-	-	-	-	-	-	15
LPTIM1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	5	-	-	-	-	-	-	-
LPTIM2	-	-	-	-	-	-	-	-	-	-	-	-	-	-	5	-	-	-	-	-	-	-
ADC1	3	-	-	-	-	-	-	-	-	-	-	-	-	-	16	-	-	-	-	-	-	-
DFSDM1	6	6	-	-	-	-	-	-	6	6	6	6	6	-	-	-	-	-	-	-	-	-
温度传感器	-	-	-	-	-	-	-	-	-	-	-	-	-	12	-	-	-	-	-	-	-	-
VBAT	-	-	-	-	-	-	-	-	-	-	-	-	-	12	-	-	-	-	-	-	-	-
VREFINT	-	-	-	-	-	-	-	-	-	-	-	-	-	12	-	-	-	-	-	-	-	-
OPAMP1	-	-	-	-	-	-	-	-	-	-	-	-	-	12	-	-	-	-	-	-	-	-

表 41. STM32L4Rxxx 和 STM32L4Sxxx 外设互连矩阵(1) (2)（续）

源	目标																					
	TIM1	TIM8	TIM2	TIM3	TIM4	TIM5	TIM6	TIM7	TIM15	TIM16	TIM17	LPTIM1	LPTIM2	ADC1	DFSDM1	OPAMP1	OPAMP2	DAC1	DAC2	COMP1	COMP2	IRTIM
OPAMP2	-	-	-	-	-	-	-	-	-	-	-	-	-	12	-	-	-	-	-	-	-	-
DAC1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	12	12	-	-	-	-	-
DAC2	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
HSE	-	-	-	-	-	-	-	-	-	-	7	-	-	-	-	-	-	-	-	-	-	-
LSE	-	-	7	-	-	-	-	-	7	7	-	-	-	-	-	-	-	-	-	-	-	-
MSI	-	-	-	-	-	-	-	-	-	-	7	-	-	-	-	-	-	-	-	-	-	-
LSI	-	-	-	-	-	-	-	-	-	7	-	-	-	-	-	-	-	-	-	-	-	-
MCO	-	-	-	-	-	-	-	-	-	-	7	-	-	-	-	-	-	-	-	-	-	-
EXTI	-	-	-	-	-	-	-	-	-	-	-	-	-	2	5	-	-	4	4	-	-	-
RTC	-	-	-	-	-	-	-	-	-	7	-	8	8	-	-	-	-	-	-	-	-	-
COMP1	13	13	13	13	-	-	-	-	13	13	13	8	8	-	-	-	-	-	-	-	-	-
COMP2	13	13	13	13	-	-	-	-	13	13	13	8	8	-	-	-	-	-	-	-	-	-
SYST ERR	14	14	-	-	-	-	-	-	14	14	14	-	-	-	-	-	-	-	-	-	-	-
USB	-	-	11	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

1. 表中的编号是*第 10.3 节：互连详细信息*中相应详细说明小节的链接。
2. 灰色单元格中的“-”符号表示无互连。

10.3 互连详细信息

10.3.1 从定时器 (TIM1/TIM2/TIM3/TIM4/TIM5/TIM8/TIM15/TIM16/TIM17) 到定时器 (TIM1/TIM2/TIM3/TIM4/TIM5/TIM8/TIM15)

目的

某些 TIMx 定时器从内部连接在一起，以实现定时器同步或链接。

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。

*第 35.3.19 节：定时器同步*中对功能进行了说明。

以下章节对同步模式进行了详细说明：

- *第 34.3.26 节：定时器同步*（面向高级控制定时器 (TIM1/TIM8)）
- *第 35.3.18 节：定时器与外部触发同步*（面向通用定时器 (TIM2/TIM3/TIM4/TIM5)）
- *第 36.4.19 节：外部触发同步（仅适用于 TIM15）*（面向通用定时器 (TIM15)）

触发信号

在可配置定时器事件发生后，输出（来自主设备）出现在 TIMx_TRGO 信号上（对于 TIM1/TIM8，还出现在 TIMx_TRGO2 信号上）。

输入（到从设备）位于 TIMx_ITR0/ITR1/ITR2/ITR3 信号上。

TIM1/TIM8 的输入和输出信号如*图 254：高级控制定时器框图*所示。

以下各表介绍了可能的主/从连接：

- *表 215：TIMx 内部触发连接*
- *表 220：TIMx 内部触发连接*
- *表 224：TIMx 内部触发连接*

有效功耗模式

运行、睡眠、低功耗运行、低功耗睡眠。

10.3.2 从定时器 (TIM1/TIM2/TIM3/TIM4/TIM6/TIM8/TIM15) 和 EXTI 到 ADC (ADC1)

目的

通用定时器 (TIM2/TIM3/TIM4)、基本定时器 (TIM6)、高级控制定时器 (TIM1/TIM8)、通用定时器 (TIM15) 和 EXTI 可用于生成 ADC 触发事件。

第 34.3.27 节：ADC 同步 (TIM1/TIM8) 对 TIMx 同步进行了说明。

*第 21.4.18 节：外部触发转换和触发极性（EXTSEL、EXTEN、JEXTSEL、JEXTEN）*对 ADC 同步进行了说明。

触发信号

输出（来自定时器）位于 TIMx_TRGO、TIMx_TRGO2 或 TIMx_CCx 信号事件上。

输入（到 ADC）位于 EXT[15:0] 和 JEXT[15:0] 信号上。

以下各表给出了定时器和 ADC 之间的连接：

- *表 116：ADC1——常规通道的外部触发*
- *表 117：ADC1——注入通道的外部触发*

有效功耗模式

运行、睡眠、低功耗运行、低功耗睡眠。

10.3.3 从 ADC 到定时器 (TIM1/TIM8)

目的

ADC1 可通过看门狗信号向高级控制定时器 (TIM1/TIM8) 提供触发事件。

*第 21.4.29 节：模拟窗口看门狗（AWD1EN、JAWD1EN、AWD1SGL、AWD1CH、AWD2CH、AWD3CH、AWD_HTx、AWD_LTx、AWDx）*中对 ADC 模拟看门狗设置进行了说明。

*第 34.3.4 节：外部触发输入*中对定时器的触发设置进行了说明。

触发信号

输出（来自 ADC）位于信号 ADC_AWDx_OUT n = 1（针对 ADC1）上，其中 x = 1、2、3（每个 ADC 具有 3 个看门狗）；输入（到定时器）位于信号 TIMx_ETR（外部触发信号）上。

有效功耗模式

运行、睡眠、低功耗运行、低功耗睡眠。

10.3.4 从定时器 (TIM2/TIM4/TIM5/TIM6/TIM7/TIM8) 和 EXTI 到 DAC (DAC1/DAC2)

目的

通用定时器 (TIM2/TIM4/TIM5)、基本定时器（TIM6、TIM7）、高级控制定时器 (TIM8) 和 EXTI 可用作触发事件以启动 DAC 转换。

触发信号

输出（来自定时器）位于直接连接至相应 DAC 输入的 TIMx_TRGO 信号上。

*第 22.4.6 节：DAC 触发选择*中给出了 DAC 上输入触发的选择（单/双模式）。

有效功耗模式

运行、睡眠、低功耗运行、低功耗睡眠。

10.3.5 从定时器 (TIM1/TIM3/TIM4/TIM6/TIM7/TIM8/TIM16/LPTIM1/LPTIM2) 和 EXTI 到 DFSDM1

目的

通用定时器 (TIM3/TIM4)、基本定时器 (TIM6/TIM7)、高级控制定时器 (TIM1/TIM8)、通用定时器 (TIM16)、低功耗定时器 (LPTIM1/LPTIM2) 和 EXTI 可用于在 DFSDM1 模块中（在每个可能的数据块 DFSDM1_FLT0/DFSDM1_FLT1/DFSDM1_FLT2/DFSDM1_FLT3 上）生成触发事件并启动 ADC 转换。

*第 27.4.15 节：启动转换*对 DFSDM 触发的转换功能进行了说明。

触发信号

输出（来自定时器）位于 TIMx_TRGO/TIMx_TRGO2 或 TIM16_OC1 信号上。

输入（位于 DFSDM1 中）出现在 DFSDM1_INTRG[0:8] 信号上。

*表 162：DFSDM 触发器连接*中给出了定时器、EXTI 和 DFSDM1 之间的连接。

有效功耗模式

运行、睡眠、低功耗运行、低功耗睡眠。

10.3.6 从 DFSDM1 到定时器 (TIM1/TIM8/TIM15/TIM16/TIM17)

目的

激活看门狗（模拟信号超出最大或最小阈值）或进行短路检测后，DFSDM1 可在高级控制定时器 (TIM1/TIM8) 和通用定时器 (TIM15/TIM16/TIM17) 上生成定时器中断信号。

*第 27.4.10 节：模拟看门狗*对 DFSDM1 看门狗进行了说明。

*第 27.4.11 节：短路检测器*对 DFSDM1 短路检测进行了说明。

以下部分对定时器断路进行了说明：

- *第 34.3.16 节：使用断路功能* (TIM1/TIM8)
- *第 36.4.13 节：使用断路功能* (TIM15/TIM16/TIM17)

触发信号

输出（来自 DFSDM1）位于直接连接至定时器的 dfsdm1_break[0:3] 信号上，并与定时器的其他断路输入信号进行“或”运算。

有效功耗模式

运行、睡眠、低功耗运行、低功耗睡眠。

10.3.7 从 HSE、LSE、LSI、MSI、MCO、RTC 到定时器 (TIM2/TIM15/TIM16/TIM17)

目的

外部时钟（HSE 和 LSE）、内部时钟（LSI 和 MSI）、微控制器输出时钟 (MCO)、GPIO 和 RTC 唤醒中断可用作通用定时器 (TIM15/16/17) 通道 1 的输入。

这可以校准 HSI16/MSI 系统时钟（使用 TIM15/TIM16 和 LSE）或 LSI（使用 TIM16 和 HSE）。也可用于精确测量 LSI（使用 TIM16 和 HSI16）或 MSI（使用 TIM17 和 HSI16）振荡器频率。

使用低速外部 (LSE) 振荡器时，无需额外的硬件连接。

*第 6.2.17 节：基于 TIM15/TIM16/TIM17 的内部/外部时钟测量*对此功能进行了说明。

外部时钟 LSE 可用作 TIM2_ETR 引脚上通用定时器 (TIM2) 的输入，请参见*第 35.4.19 节：TIM2 选项寄存器 1 (TIM2_OR1)*。

有效功耗模式

运行、睡眠、低功耗运行、低功耗睡眠。

10.3.8 从 RTC、COMP1、COMP2 到低功耗定时器 (LPTIM1/LPTIM2)

目的

RTC 闹钟 A/B、RTC_TAMP1/2/3 输入检测、COMP1/2_OUT 可用作触发信号以启动 LPTIM 计数器 (LPTIM1/2)。

触发信号

第 38.4.6 节：触发多路复用器（以及后续部分）对此触发功能进行了说明。

*表 231：LPTIM1 外部触发连接*对输入选择进行了说明。

有效功耗模式

运行、睡眠、低功耗运行、低功耗睡眠、停止 0、停止 1、停止 2（仅限 LPTIM1）。

10.3.9 从定时器 (TIM1/TIM2/TIM3/TIM8/TIM15) 到比较器 (COMP1/COMP2)

目的

高级控制定时器 (TIM1/TIM8)、通用定时器 (TIM2/TIM3) 和通用定时器 (TIM15) 可用作 COMP1/COMP2 的消隐窗口输入

*第 25.3.7 节：比较器输出消隐功能*对消隐功能进行了说明。

以下部分给出了消隐源：

- *第 25.6.1 节：比较器 1 控制和状态寄存器 (COMP1_CSR)* 位 20:18 BLANKING[2:0]
- *第 25.6.2 节：比较器 2 控制和状态寄存器 (COMP2_CSR)* 位 20:18 BLANKING[2:0]

触发信号

定时器输出信号 TIMx_Ocx 是 COMP1/COMP2 的消隐源输入。

有效功耗模式

运行、睡眠、低功耗运行、低功耗睡眠。

10.3.10 从 USB 到定时器 (TIM2)

目的

USB (OTG_FS SOF) 可以生成面向通用定时器 (TIM2) 的触发信号。

*表 220：TIMx 内部触发连接*对 USB 与 TIM2 的连接进行了说明。

触发信号

由 USB FS 帧起始生成的内部信号。

有效功耗模式

运行、睡眠。

10.3.11 从内部模拟源到 ADC 和 OPAMP (OPAMP1/OPAM2)

目的

内部温度传感器 (V_{TS}) 和 V_{BAT} 监测通道连接至 ADC1 输入通道。

内部参考电压 (V_{REFINT}) 连接至 ADC1 输入通道。

OPAMP1 和 OPAMP2 输出可通过 GPIO 连接至 ADC1 输入通道。

DAC1_OUT1 可连接至 OPAMP1_VINP。

DAC1_OUT2 可连接至 OPAMP2_VINP。

其依据包括：

- *第 21.2 节：ADC 主要特性*
- *第 21.4.11 节：通道选择（SQRx、JSQRx）*
- *图 85：ADC1 连接功能*
- *表 155：运算放大器连接情形*

有效功耗模式

运行、睡眠、低功耗运行、低功耗睡眠。

10.3.12 从比较器 (COMP1/COMP2) 到定时器 (TIM1/TIM2/TIM3/TIM8/TIM15/TIM16/TIM17)

目的

比较器 (COMP1/COMP2) 输出值可连接至定时器 (TIM1/TIM2/TIM3/TIM8/TIM15/TIM16/TIM17) 输入捕获或 TIMx_ETR 信号。

*第 34.3.4 节：外部触发输入*对与 ETR 的连接进行了说明。

比较器 (COMP1/COMP2) 输出值还可使用 IO 的开漏连接，通过 GPIO 复用功能选择在输入引脚 TIMx_BKIN 或 TIMx_BKIN2 上为定时器 (TIM1/TIM8) 生成断路输入信号，请参见*第 34.3.17 节：双向断路输入*。

以下部分给出了可能的连接：

- *第 34.4.21 节：TIM1 选项寄存器 1 (TIM1_OR1)*
- *第 34.4.22 节：TIM8 选项寄存器 1 (TIM8_OR1)*
- *第 35.4.19 节：TIM2 选项寄存器 1 (TIM2_OR1)*
- *第 35.4.20 节：TIM3 选项寄存器 1 (TIM3_OR1)*
- *第 35.4.23 节：TIMx 寄存器映射*
- *第 35.4.22 节：TIM3 选项寄存器 2 (TIM3_OR2)*
- *第 36.3 节：TIM16/TIM17 主要特性*

有效功耗模式

运行、睡眠、低功耗运行、低功耗睡眠。

10.3.13 从系统错误到定时器 (TIM1/TIM8/TIM15/TIM16/TIM17)

目的

CSS、CPU HardFault、RAM 奇偶校验错误、FLASH ECC 双重错误检测，PVD 可面向定时器 (TIM1/TIM8/TIM15/TIM16/TIM17) 以定时器断路形式生成系统错误。

断路功能的目的是保护由这些定时器生成的 PWM 信号所驱动的功率器件。

以下部分列出了可能的断路源：

- *第 34.3.16 节：使用断路功能* (TIM1/TIM8)
- *第 36.4.13 节：使用断路功能* (TIM15/TIM16/TIM17)
- *图 364：TIM15 框图*
- *图 365：TIM16/TIM17 框图*

有效功耗模式

运行、睡眠、低功耗运行、低功耗睡眠。

10.3.14 从定时器 (TIM16/TIM17) 到 IRTIM

目的

通用定时器 (TIM16/TIM17) 输出通道 TIMx_OC1 用于生成红外信号输出的波形。

第 39 节：红外接口 (IRTIM) 对此功能进行了介绍。

有效功耗模式

运行、睡眠、低功耗运行、低功耗睡眠。

10.3.15 从 ADC (ADC1) 到 DFSDM

目的

通过一条并行总线，最多可将 3 个内部 ADC 结果直接连接至 DFSDM 输入，从而使用 DFSDM 滤波功能。

该功能作为 DFSDM 外设说明的一部分在*第 27.4.6 节：并行数据输入（来自内部 ADC 的输入）*中有所介绍。

以下部分给出了可能的连接：

- *第 27.7.1 节：DFSDM 通道 y 配置寄存器 (DFSDM_CHyCFGR1)*
  - 位 13:12 DATMPX[1:0]：通道 y 输入数据复用器 (Input data multiplexer for channel y)
- *第 27.7.5 节：DFSDM 通道 y 数据输入寄存器 (DFSDM_CHyDATINR)*
  - 位 31:16 INDAT0[15:0]：通道 y 或通道 y+1 的输入数据 (Input data for channel y or channel y+1)
  - 位 15:0 INDAT0[15:0]：通道 y 的输入数据 (Input data for channel y)

有效功耗模式

运行、睡眠、低功耗运行、低功耗睡眠。

11 直接存储器访问控制器 (DMA)

11.1 简介

直接存储器访问 (DMA) 控制器是一个主控总线系统外设。

当通过控制为 CPU 减轻负担时，DMA 用于在存储器映射的外设和/或存储器之间进行可编程的数据传输。

DMA 控制器采用单 AHB 主控总线架构。

DMA 有两个实例：DMA1（7 通道）和 DMA2（7 通道）。

每个通道都专门管理来自一个或更多外设的存储器访问请求。每个 DMA 都有一个仲裁器，用于处理 DMA 请求间的优先级。

11.2 DMA 主要特性

- 单 AHB 主控总线
- 外设到存储器、存储器到外设、存储器到存储器以及外设到外设的数据传输
- 访问（作为源和目标）片上存储器映射的器件，例如 Flash、SRAM、AHB 和 APB 外设
- 所有 DMA 通道均可单独配置：
  - 每个通道均与来自外设的 DMA 请求信号或存储器到存储器传输中的软件触发信号相关联。由软件来完成配置。
  - 各请求之间的优先级可用软件编程（每通道 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，通道 1 请求的优先级高于通道 2 请求的优先级）。
  - 源和目标的传输大小（字节、半字、字）彼此独立，模拟打包和拆包。源和目标的地址必须根据传输数据的大小进行对齐。
  - 支持外设与存储器之间的双向传输，并支持循环缓冲区管理。
  - 可编程的待传输数据数目：0 到 2^{16} - 1
- 每个通道生成一个中断请求。每个中断请求均因以下三个 DMA 事件中的任意一个而引起：传输完成、半传输或传输错误。

11.3 DMA 实现

11.3.1 DMA1 和 DMA2

DMA1 和 DMA2 使用*表 42* 中所示的硬件配置参数实现。

表 42. DMA1 和 DMA2 实现

特性	DMA1	DMA2
通道数	7	7

11.3.2 DMA 请求映射

DMA 控制器通过 DMAMUX 外设连接至来自 AHB/APB 外设的 DMA 请求。

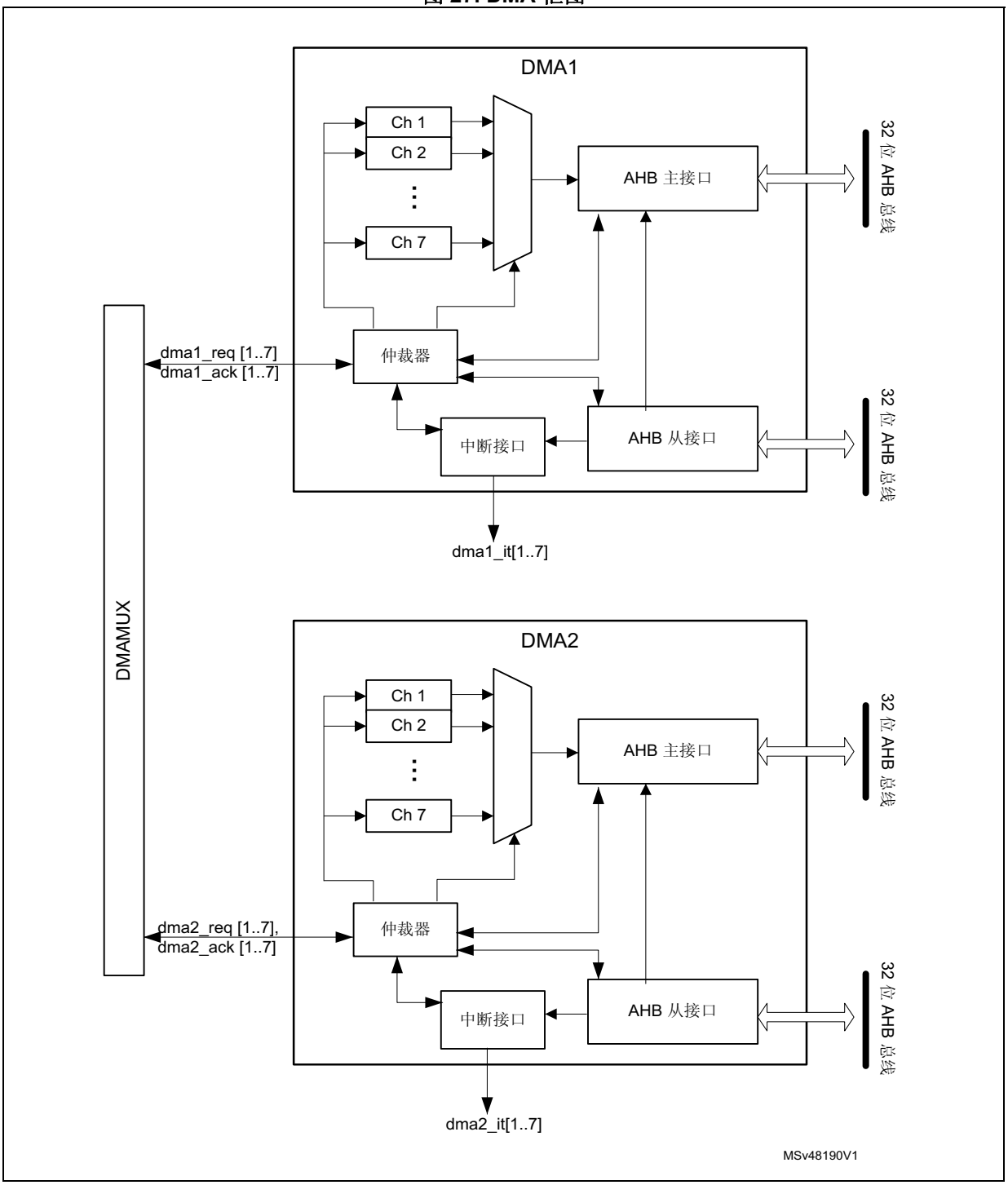
有关不同请求的映射，请参见*第 12.3 节：DMAMUX 实现*。

11.4 DMA 功能说明

11.4.1 DMA 框图

DMA 框图如*图 27* 所示。

图 27. DMA 框图



DMA 控制器通过与其他系统主设备共用 AHB 系统总线来执行直接存储器传输。总线矩阵执行循环调度。当 CPU 与 DMA 访问目标（存储器或外设）相同时，DMA 请求会将 CPU 对系统总线的访问停止数个总线周期。

根据通过 AHB 从接口实现的相应配置，DMA 控制器会在 DMA 通道及其接收的相关请求之间进行仲裁。DMA 控制器还会通过单一 AHB 端口主器件调度 DMA 数据传输。

DMA 控制器按通道向中断控制器生成中断。

11.4.2 DMA 引脚和内部信号

表 43. DMA 内部输入/输出信号

信号名称	信号类型	说明
dma_req[x]	输入	DMA 通道 x 请求
dma_ack[x]	输出	DMA 通道 x 应答
dma_it[x]	输出	DMA 通道 x 中断

11.4.3 DMA 传输

软件在通道级配置 DMA 控制器，以便执行块传输，此类传输由一系列 AHB 总线传输组成。

DMA 块传输可从外设请求，或在进行存储器到存储器传输时由软件触发。

触发该事件后，会按照以下步骤进行单次 DMA 传输：

1. 外设向 DMA 控制器发送单个 DMA 请求信号。
2. DMA 控制器按照与此外设请求相关的通道的优先级来处理该请求。
3. 只要 DMA 控制器授权给外设，DMA 控制器就会向外设发送确认信号。
4. 外设获得 DMA 控制器的确认信号后，便会立即释放其请求。
5. 一旦外设使请求失效，DMA 控制器就会释放确认信号。

外设可继续发送请求，再次启动 DMA 传输。

不论外设是传输源还是传输目标，都使用请求/应答协议。例如，对于存储器到外设传输，外设通过对 DMA 控制器驱动其单个请求信号来启动传输。DMA 控制器随后读取存储器中的单个数据，并将该数据写入外设。

对于给定通道 x，DMA 块传输由以下内容的重复序列组成：

- 单次 DMA 传输，其中封装单个数据的两次 AHB 传输（通过 DMA AHB 总线主控实现）：
  - 通过内部当前外设/存储器地址寄存器进行寻址，从外设数据寄存器或存储器单元中读取单个数据（字节、半字或字）。
    用于首次传输的起始地址是在 DMA_CPARx 或 DMA_CMARx 寄存器中编程的外设或存储器基址。
  - 通过内部当前外设/存储器地址寄存器进行寻址，向外设数据寄存器或存储器单元中写入单个数据（字节、半字或字）。
    用于首次传输的起始地址是在 DMA_CPARx 或 DMA_CMARx 寄存器中编程的外设或存储器基址。
- 编程的 DMA_CNDTRx 寄存器在传输后递减。
  该寄存器包含待传输数据项的剩余数目（AHB“先读后写”传输次数）。

重复执行该序列，直至 DMA_CNDTRx 为空。

注： *AHB 主控总线源地址/目标地址必须根据传输至源/目标的单个数据的设定大小进行对齐。*

11.4.4 DMA 仲裁

DMA 仲裁器管理不同通道间的优先级。

当仲裁器为某个有效通道 x 授予优先权后（硬件请求或软件触发），会发起单次 DMA 传输（例如，单个数据的一次 AHB“先读后写”传输）。随后仲裁器会再次对有效通道组进行仲裁，并选择优先级最高的通道。

优先级管理分为两个阶段：

- 软件：每个通道的优先级均在 DMA_CCRx 寄存器中配置为以下四个不同级别之一：
  - 非常高
  - 高
  - 中
  - 低
- 硬件：如果两个请求的软件优先级相同，则索引编号最低的通道优先。例如，通道 2 的优先级高于通道 4。

当通道 x 被编程为在存储器到存储器模式下进行块传输时，在此通道 x 的各个单次 DMA 传输之间将进行重新仲裁。每当存在其他同时带有有效请求的通道时，DMA 仲裁器便会自动授权切换到其他有有效请求的通道中具有最高优先级的通道，此通道的优先级可能低于存储器到存储器通道。

11.4.5 DMA 通道

各个通道均能处理外设寄存器（位于固定地址中）与存储器单元之间的 DMA 传输。待传输数据项的数目可编程。每个传输完成后，包含待传输的数据项数目的寄存器都会递减。

DMA 通道以块传输的方式进行编程设置。

数据大小可编程

要传输到外设和存储器的单个数据的大小（字节、半字或字）可分别通过 DMA_CCRx 寄存器的 PSIZE[1:0] 和 MSIZE[1:0] 位域进行编程。

指针递增

根据 DMA_CCRx 寄存器的 PINC 位和 MINC 位的状态，外设和存储器指针在每次传输后可自动递增。

如果使能了**递增模式**（PINC 或 MINC 置 1），则下次传输的地址是前一次传输的地址加上 1、2 或 4，具体取决于 PSIZE[1:0] 或 MSIZE[1:0] 中定义的数据大小。首次传输的地址可在 DMA_CPARx 或 DMA_CMARx 寄存器中进行编程。在传输过程中，这些寄存器将保持初始编程的值。软件无法获得当前传输的地址（位于当前内部外设或存储器地址寄存器中）。

如果将通道 x 配置为**非循环模式**，则在最后一次数据传输完成后（即待传输的单个数据数目达到零后），将不处理任何 DMA 请求。必须禁止 DMA 通道，这样才能将新的数据项数目重新加载到 DMA_CNDTRx 寄存器。

注： *如果禁止通道 x，DMA 寄存器不会被复位。DMA 通道寄存器（DMA_CCRx、DMA_CPARx 和 DMA_CMARx）仍保留在通道配置阶段设置的初始值。*

在**循环模式**下，最后一次数据传输完成后，DMA_CNDTRx 寄存器将自动重新加载初始编程值。当前的内部地址寄存器重新加载 DMA_CPARx 和 DMA_CMARx 寄存器中的基址值。

通道配置流程

配置 DMA 通道 x 时需按照以下步骤操作：

1. 在 DMA_CPARx 寄存器中设置外设寄存器地址。
   在外设事件发生后，或在存储器到存储器模式下使能通道后，会将数据从该地址移至存储器或从存储器移至该地址。
2. 设置 DMA_CMARx 寄存器中的存储器地址。
   在外设事件发生后，或在存储器到存储器模式下使能通道后，会将数据写入存储器或从存储器读取数据。
3. 在 DMA_CNDTRx 寄存器中配置待传输的总数据数。
   每次数据传输后，该值都会递减。
4. 在 DMA_CCRx 寄存器中配置下列参数：
   - 通道优先级
   - 数据传输方向
   - 循环模式
   - 外设和存储器递增模式
   - 外设和存储器数据大小
   - 传输完成一半和/或全部完成以及/或者出现传输错误时的中断使能
5. 将 DMA_CCRx 寄存器中的 EN 位置 1 以激活通道。

通道在使能后可处理来自此通道所连接外设的任意 DMA 请求，或者启动存储器到存储器块传输。

注： *通道配置流程的最后两步可合并为对 DMA_CCRx 寄存器进行单次访问来配置和使能通道。*

如果通道已使能且仍处于活动状态（未完成），则软件必须对 DMA_CCRx 寄存器执行两次单独的写访问，以禁用该通道，然后重新编程该通道以进行下一次块传输。
EN 位置 1 后，DMA_CCRx 寄存器的某些位域为只读位域。

停止和恢复通道

软件激活某个通道后，会等待编程的传输完成。DMA 控制器不能恢复总线传输可能挂起的已中止有效通道。

要正确停止并禁止某个通道，软件应将 DMA_CCRx 寄存器的 EN 位清零。软件应确保在传输完成前 DMA 控制器未处理任何来自外设的挂起请求。软件会等待传输完成或传输错误中断。

发生通道传输错误时，DMA_CCRx 寄存器的 EN 位由硬件清零。在 DMA_ISR 寄存器的 TEIFx 位置 1 之前，不能通过软件将该 EN 位置 1 来重新激活通道 x。

循环模式（在存储器到外设/外设到存储器传输中）

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）。可使用 DMA_CCRx 寄存器中的 CIRC 位使能此功能。

注： *在存储器到存储器模式下，不能使用循环模式。在循环模式 (CIRC = 1) 下使能通道前，软件必须将 DMA_CCRx 寄存器的 MEM2MEM 位清零。如果循环模式已激活，则待传输数据的数目将自动重新装载为在通道配置阶段设置的初始值，并继续响应 DMA 请求。*

为停止循环传输，软件需要在禁止 DMA 通道前使外设停止生成 DMA 请求（例如退出 ADC 扫描模式）。
软件必须在启动/使能传输前，以及在停止循环传输后，明确设定 DMA_CNDTRx 值。

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。该模式被称为存储器到存储器模式，由软件启动。

如果 DMA_CCRx 寄存器的 MEM2MEM 位置 1，则通道（如果使能）会启动传输。DMA_CNDTRx 寄存器达到零后，传输停止。

注： *在循环模式下，不得使用存储器到存储器模式。在存储器到存储器模式 (MEM2MEM = 1) 下使能通道前，软件必须将 DMA_CCRx 寄存器的 CIRC 位清零。*

外设到外设模式

在以下情况下，任意 DMA 通道均可工作在外设到外设模式下：

- 选择来自外设的硬件请求触发 DMA 通道时

  此外设为 DMA 发起方，并且对自身与属于其他存储器映射外设（此外设未配置为 DMA 模式）的寄存器之间的数据传输进行调整。
- 未选择任何外设请求和未将任何外设请求连接至 DMA 通道时

  软件通过将 DMA_CCRx 寄存器的 MEM2MEM 位置 1 来配置寄存器到寄存器的传输。

设定传输方向，分配源 / 目标

DMA_CCRx 寄存器的 DIR 位值用于设置传输方向，因此会标识源和目标，这与源 / 目标类型（外设或存储器）无关：

- **DIR = 1** 通常用于定义存储器到外设的传输。一般而言，如果 DIR = 1：
  - **源**属性将由 DMA_MARx 寄存器以及 DMA_CCRx 寄存器的 MSIZE[1:0] 位域和 MINC 位定义。

    无论其常用的命名规则如何，上述“存储器”寄存器、位域和位都用于在外设到外设模式下定义源外设。
  - **目标**属性将由 DMA_PARx 寄存器以及 DMA_CCRx 寄存器的 PSIZE[1:0] 位域和 PINC 位定义。

    无论其常用的命名规则如何，上述“外设”寄存器、位域和位都用于在存储器到存储器模式下定义目标存储器。
- **DIR = 0** 通常用于定义外设到存储器的传输。一般而言，如果 DIR = 0：
  - **源**属性将由 DMA_PARx 寄存器以及 DMA_CCRx 寄存器的 PSIZE[1:0] 位域和 PINC 位定义。

    无论其常用的命名规则如何，上述“外设”寄存器、位域和位都用于在存储器到存储器模式下定义源存储器。
  - **目标**属性将由 DMA_MARx 寄存器以及 DMA_CCRx 寄存器的 MSIZE[1:0] 位域和 MINC 位定义。

    无论其常用的命名规则如何，上述“存储器”寄存器、位域和位都用于在外设到外设模式下定义目标外设。

11.4.6 DMA 数据宽度、对齐和字节序

如果 PSIZE[1:0] 和 MSIZE[1:0] 不相等，则 DMA 控制器将按照 *表 44* 所述方式进行数据对齐。

表 44. 可编程的数据宽度和字节序（PINC = MINC = 1 时）

源端口宽度（DIR = 1 时为 MSIZE，否则为 PSIZE）	目标端口宽度（DIR = 1 时为 PSIZE，否则为 MSIZE）	要传输的数据项的数目 (NDT)	源内容：地址/数据（DIR = 1 时为 DMA_CMARx，否则为 DMA_CPARx）	DMA 传输	目标内容：地址/数据（DIR = 1 时为 DMA_CPARx，否则为 DMA_CMARx）
8	8	8	@0x0/B0 @0x1/B1 @0x2/B2 @0x3/B3	1：读取 B0[7:0] @0x0，然后写入 B0[7:0] @0x0 2：读取 B1[7:0] @0x1，然后写入 B1[7:0] @0x1 3：读取 B2[7:0] @0x2，然后写入 B2[7:0] @0x2 4：读取 B3[7:0] @0x3，然后写入 B3[7:0] @0x3	@0x0/B0 @0x1/B1 @0x2/B2 @0x3/B3
8	16	4	@0x0/B0 @0x1/B1 @0x2/B2 @0x3/B3	1：读取 B0[7:0] @0x0，然后写入 00B0[15:0] @0x0 2：读取 B1[7:0] @0x1，然后写入 00B1[15:0] @0x2 3：读取 B2[7:0] @0x2，然后写入 00B2[15:0] @0x4 4：读取 B3[7:0] @0x3，然后写入 00B3[15:0] @0x6	@0x0/00B0 @0x2/00B1 @0x4/00B2 @0x6/00B3
8	32	4	@0x0/B0 @0x1/B1 @0x2/B2 @0x3/B3	1：读取 B0[7:0] @0x0，然后写入 000000B0[31:0] @0x0 2：读取 B1[7:0] @0x1，然后写入 000000B1[31:0] @0x4 3：读取 B2[7:0] @0x2，然后写入 000000B2[31:0] @0x8 4：读取 B3[7:0] @0x3，然后写入 000000B3[31:0] @0xC	@0x0/000000B0 @0x4/000000B1 @0x8/000000B2 @0xC/000000B3
16	8	4	@0x0/B1B0 @0x2/B3B2 @0x4/B5B4 @0x6/B7B6	1：读取 B1B0[15:0] @0x0，然后写入 B0[7:0] @0x0 2：读取 B3B2[15:0] @0x2，然后写入 B2[7:0] @0x1 3：读取 B5B4[15:0] @0x4，然后写入 B4[7:0] @0x2 4：读取 B7B6[15:0] @0x6，然后写入 B6[7:0] @0x3	@0x0/B0 @0x1/B2 @0x2/B4 @0x3/B6
16	16	4	@0x0/B1B0 @0x2/B3B2 @0x4/B5B4 @0x6/B7B6	1：读取 B1B0[15:0] @0x0，然后写入 B1B0[15:0] @0x0 2：读取 B3B2[15:0] @0x2，然后写入 B3B2[15:0] @0x2 3：读取 B5B4[15:0] @0x4，然后写入 B5B4[15:0] @0x4 4：读取 B7B6[15:0] @0x6，然后写入 B7B6[15:0] @0x6	@0x0/B1B0 @0x2/B3B2 @0x4/B5B4 @0x6/B7B6
16	32	4	@0x0/B1B0 @0x2/B3B2 @0x4/B5B4 @0x6/B7B6	1：读取 B1B0[15:0] @0x0，然后写入 0000B1B0[31:0] @0x0 2：读取 B3B2[15:0] @0x2，然后写入 0000B3B2[31:0] @0x4 3：读取 B5B4[15:0] @0x4，然后写入 0000B5B4[31:0] @0x8 4：读取 B7B6[15:0] @0x6，然后写入 0000B7B6[31:0] @0xC	@0x0/0000B1B0 @0x4/0000B3B2 @0x8/0000B5B4 @0xC/0000B7B6
32	8	4	@0x0/B3B2B1B0 @0x4/B7B6B5B4 @0x8/BBBAB9B8 @0xC/BFBEBDBC	1：读取 B3B2B1B0[31:0] @0x0，然后写入 B0[7:0] @0x0 2：读取 B7B6B5B4[31:0] @0x4，然后写入 B4[7:0] @0x1 3：读取 BBBAB9B8[31:0] @0x8，然后写入 B8[7:0] @0x2 4：读取 BFBEBDBC[31:0] @0xC，然后写入 BC[7:0] @0x3	@0x0/B0 @0x1/B4 @0x2/B8 @0x3/BC
32	16	4	@0x0/B3B2B1B0 @0x4/B7B6B5B4 @0x8/BBBAB9B8 @0xC/BFBEBDBC	1：读取 B3B2B1B0[31:0] @0x0，然后写入 B1B0[15:0] @0x0 2：读取 B7B6B5B4[31:0] @0x4，然后写入 B5B4[15:0] @0x2 3：读取 BBBAB9B8[31:0] @0x8，然后写入 B9B8[15:0] @0x4 4：读取 BFBEBDBC[31:0] @0xC，然后写入 BDBC[15:0] @0x6	@0x0/B1B0 @0x2/B5B4 @0x4/B9B8 @0x6/BDBC
32	32	4	@0x0/B3B2B1B0 @0x4/B7B6B5B4 @0x8/BBBAB9B8 @0xC/BFBEBDBC	1：读取 B3B2B1B0[31:0] @0x0，然后写入 B3B2B1B0[31:0] @0x0 2：读取 B7B6B5B4[31:0] @0x4，然后写入 B7B6B5B4[31:0] @0x4 3：读取 BBBAB9B8[31:0] @0x8，然后写入 BBBAB9B8[31:0] @0x8 4：读取 BFBEBDBC[31:0] @0xC，然后写入 BFBEBDBC[31:0] @0xC	@0x0/B3B2B1B0 @0x4/B7B6B5B4 @0x8/BBBAB9B8 @0xC/BFBEBDBC

解决 AHB 外设不支持字节/半字写传输的问题

如果 DMA 控制器启动 AHB 字节或半字写传输，则 32 位 AHB 主控数据总线 (HWDATA[31:0]) 的未使用数据线上将会重复所传输的数据。

如果 AHB 从外设不支持字节或半字写传输且不会产生任何错误，则 DMA 控制器会对 HWDATA 的 32 个位执行写操作，如下列两个示例所示：

- 要写入半字 0xABCD，DMA 控制器会将 HWDATA 总线设为 0xABCDABCD，并采用半字数据大小（在 AHB 主控总线中，将 HSIZE 设为 HalfWord）。
- 要写入字节 0xAB，DMA 控制器会将 HWDATA 总线设为 0xABABABAB，并采用字节数据大小（在 AHB 主控总线中，将 HSIZE 设为 Byte）。

假设 AHB/APB 桥为 AHB 32 位从外设，且该外设不考虑 HSIZE 数据，则任何 AHB 字节或半字传输都将转换为 32 位 APB 传输，如下所述：

- 将 AHB 字节写传输转换为 APB 字写传输，如向 0x0、0x1、0x2 或 0x3 地址之一写入 0xB0 转换为向 0x0 地址写入 0xB0B0B0B0。
- 将 AHB 半字写传输转换为 APB 字写传输，如向 0x0 或 0x2 地址写入 0xB1B0 转换为向 0x0 地址写入 0xB1B0B1B0。

11.4.7 DMA 错误管理

当对保留的地址空间执行读写操作时，将生成 DMA 传输错误。如果在 DMA 读或写访问过程中生成了 DMA 传输错误，则硬件会将相应 DMA_CCRx 寄存器的 EN 位清零，从而自动禁止出错的通道 x。

DMA_ISR 寄存器的 TEIFx 位置 1。如果 DMA_CCRx 寄存器的 TEIE 位置 1，还将产生中断。

在 DMA_ISR 寄存器的 TEIFx 位清零（通过将 DMA_IFCR 寄存器的 CTEIFx 位置 1）前，DMA_CCRx 寄存器的 EN 位不能再次由软件置 1（通道 x 重新激活）。

软件在收到与外设相关的通道出现传输错误的通知后，首先要停止这个在 DMA 模式下的外设，以禁止任何挂起或后续的 DMA 请求。随后软件可以正常地将 DMA 和外设重新配置为 DMA 模式，以便进行新的传输。

11.5 DMA 中断

对于每个 DMA 通道 x，在发生“半传输”、“传输完成”或“传输错误”时都会生成中断。可以使用单独的中断使能位以提高灵活性。

表 45. DMA 中断请求

中断请求	中断事件	事件标志	中断使能位
通道 x 中断	通道 x 上发生半传输	HTIFx	HTIEx
	通道 x 上传输完成	TCIFx	TCIEx
	通道 x 上发生传输错误	TEIFx	TEIEx
	通道 x 上发生半传输、传输完成或传输错误	GIFx	-

11.6 DMA 寄存器

有关寄存器说明中使用的缩写，请参见*第 1.1 节*。

DMA 寄存器必须按字（32 位）进行访问。

11.6.1 DMA 中断状态寄存器 (DMA_ISR)

DMA interrupt status register

偏移地址：0x00

复位值：0x0000 0000

每个状态位在软件将 DMA_IFCR 寄存器中相应的清零位或相应的全局清零位 CGIFx 置 1 后由硬件清零。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	TEIF7	HTIF7	TCIF7	GIF7	TEIF6	HTIF6	TCIF6	GIF6	TEIF5	HTIF5	TCIF5	GIF5
				r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TEIF4	HTIF4	TCIF4	GIF4	TEIF3	HTIF3	TCIF3	GIF3	TEIF2	HTIF2	TCIF2	GIF2	TEIF1	HTIF1	TCIF1	GIF1
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:28 保留，必须保持复位值。

位 27 **TEIF7**：通道 7 的传输错误 (TE) 标志 (transfer error (TE) flag for channel 7)
0：无 TE 事件
1：发生 TE 事件

位 26 **HTIF7**：通道 7 的半传输 (HT) 标志 (half transfer (HT) flag for channel 7)
0：无 HT 事件
1：发生 HT 事件

位 25 **TCIF7**：通道 7 的传输完成 (TC) 标志 (transfer complete (TC) flag for channel 7)
0：无 TC 事件
1：发生 TC 事件

位 24 **GIF7**：通道 7 的全局中断标志 (global interrupt flag for channel 7)
0：无 TE、HT 或 TC 事件
1：发生 TE、HT 或 TC 事件

位 23 **TEIF6**：通道 6 的传输错误 (TE) 标志 (transfer error (TE) flag for channel 6)
0：无 TE 事件
1：发生 TE 事件

位 22 **HTIF6**：通道 6 的半传输 (HT) 标志 (half transfer (HT) flag for channel 6)
0：无 HT 事件
1：发生 HT 事件

位 21 **TCIF6**：通道 6 的传输完成 (TC) 标志 (transfer complete (TC) flag for channel 6)
0：无 TC 事件
1：发生 TC 事件

位 20 **GIF6**：通道 6 的全局中断标志 (global interrupt flag for channel 6)
0：无 TE、HT 或 TC 事件
1：发生 TE、HT 或 TC 事件

位 19 **TEIF5**：通道 5 的传输错误 (TE) 标志 (transfer error (TE) flag for channel 5)
0：无 TE 事件
1：发生 TE 事件

位 18 **HTIF5**：通道 5 的半传输 (HT) 标志 (half transfer (HT) flag for channel 5)
0：无 HT 事件
1：发生 HT 事件

位 17 **TCIF5**：通道 5 的传输完成 (TC) 标志 (transfer complete (TC) flag for channel 5)
0：无 TC 事件
1：发生 TC 事件

位 16 **GIF5**：通道 5 的全局中断标志 (global interrupt flag for channel 5)
0：无 TE、HT 或 TC 事件
1：发生 TE、HT 或 TC 事件

位 15 **TEIF4**：通道 4 的传输错误 (TE) 标志 (transfer error (TE) flag for channel 4)
0：无 TE 事件
1：发生 TE 事件

位 14 **HTIF4**：通道 4 的半传输 (HT) 标志 (half transfer (HT) flag for channel 4)
0：无 HT 事件
1：发生 HT 事件

位 13 **TCIF4**：通道 4 的传输完成 (TC) 标志 (transfer complete (TC) flag for channel 4)
0：无 TC 事件
1：发生 TC 事件

位 12 **GIF4**：通道 4 的全局中断标志 (global interrupt flag for channel 4)
0：无 TE、HT 或 TC 事件
1：发生 TE、HT 或 TC 事件

位 11 **TEIF3**：通道 3 的传输错误 (TE) 标志 (transfer error (TE) flag for channel 3)
0：无 TE 事件
1：发生 TE 事件

位 10 **HTIF3**：通道 3 的半传输 (HT) 标志 (half transfer (HT) flag for channel 3)
0：无 HT 事件
1：发生 HT 事件

位 9 **TCIF3**：通道 3 的传输完成 (TC) 标志 (transfer complete (TC) flag for channel 3)
0：无 TC 事件
1：发生 TC 事件

位 8 **GIF3**：通道 3 的全局中断标志 (global interrupt flag for channel 3)
0：无 TE、HT 或 TC 事件
1：发生 TE、HT 或 TC 事件

位 7 **TEIF2**：通道 2 的传输错误 (TE) 标志 (transfer error (TE) flag for channel 2)
0：无 TE 事件
1：发生 TE 事件

位 6 **HTIF2**：通道 2 的半传输 (HT) 标志 (half transfer (HT) flag for channel 2)
0：无 HT 事件
1：发生 HT 事件

位 5 **TCIF2**：通道 2 的传输完成 (TC) 标志 (transfer complete (TC) flag for channel 2)
0：无 TC 事件
1：发生 TC 事件

位 4 **GIF2**：通道 2 的全局中断标志 (global interrupt flag for channel 2)
0：无 TE、HT 或 TC 事件
1：发生 TE、HT 或 TC 事件

位 3 **TEIF1**：通道 1 的传输错误 (TE) 标志 (transfer error (TE) flag for channel 1)
0：无 TE 事件
1：发生 TE 事件

位 2 **HTIF1**：通道 1 的半传输 (HT) 标志 (half transfer (HT) flag for channel 1)
0：无 HT 事件
1：发生 HT 事件

位 1 **TCIF1**：通道 1 的传输完成 (TC) 标志 (transfer complete (TC) flag for channel 1)
0：无 TC 事件
1：发生 TC 事件

位 0 **GIF1**：通道 1 的全局中断标志 (global interrupt flag for channel 1)
0：无 TE、HT 或 TC 事件
1：发生 TE、HT 或 TC 事件

11.6.2 DMA 中断标志清零寄存器 (DMA_IFCR)

DMA interrupt flag clear register

偏移地址：0x04

复位值：0x0000 0000

将该 DMA_IFCR 寄存器中通道 x 的全局清零位 CGIFx 置 1 后，DMA 硬件会清零 DMA_ISR 寄存器中的相应 GIFx 位以及各个 TEIFx、HTIFx 和 TCIFx 标志。

将该 DMA_IFCR 寄存器中的各个 CTEIFx、CHTIFx 和 CTCIFx 清零位均置 1 后，DMA 硬件会清零 DMA_ISR 寄存器中相应的单独标志和全局标志 GIFx，前提是其他两个单独的标志均未置 1。

对任何标志清零位写入 0 均不起作用。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	CTEIF7	CHTIF7	CTCIF7	CGIF7	CTEIF6	CHTIF6	CTCIF6	CGIF6	CTEIF5	CHTIF5	CTCIF5	CGIF5
				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
CTEIF4	CHTIF4	CTCIF4	CGIF4	CTEIF3	CHTIF3	CTCIF3	CGIF3	CTEIF2	CHTIF2	CTCIF2	CGIF2	CTEIF1	CHTIF1	CTCIF1	CGIF1
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

位 31:28 保留，必须保持复位值。

位 27 **CTEIF7**：通道 7 的传输错误标志清零 (transfer error flag clear for channel 7)

位 26 **CHTIF7**：通道 7 的半传输标志清零 (half transfer flag clear for channel 7)

位 25 **CTCIF7**：通道 7 的传输完成标志清零 (transfer complete flag clear for channel 7)

位 24 **CGIF7**：通道 7 的全局中断标志清零 (global interrupt flag clear for channel 7)

位 23 **CTEIF6**：通道 6 的传输错误标志清零 (transfer error flag clear for channel 6)

位 22 **CHTIF6**：通道 6 的半传输标志清零 (half transfer flag clear for channel 6)

位 21 **CTCIF6**：通道 6 的传输完成标志清零 (transfer complete flag clear for channel 6)

位 20 **CGIF6**：通道 6 的全局中断标志清零 (global interrupt flag clear for channel 6)

位 19 **CTEIF5**：通道 5 的传输错误标志清零 (transfer error flag clear for channel 5)

位 18 **CHTIF5**：通道 5 的半传输标志清零 (half transfer flag clear for channel 5)

位 17 **CTCIF5**：通道 5 的传输完成标志清零 (transfer complete flag clear for channel 5)

位 16 **CGIF5**：通道 5 的全局中断标志清零 (global interrupt flag clear for channel 5)

位 15 **CTEIF4**：通道 4 的传输错误标志清零 (transfer error flag clear for channel 4)

位 14 **CHTIF4**：通道 4 的半传输标志清零 (half transfer flag clear for channel 4)

位 13 **CTCIF4**：通道 4 的传输完成标志清零 (transfer complete flag clear for channel 4)

位 12 **CGIF4**：通道 4 的全局中断标志清零 (global interrupt flag clear for channel 4)

位 11 **CTEIF3**：通道 3 的传输错误标志清零 (transfer error flag clear for channel 3)

位 10 **CHTIF3**：通道 3 的半传输标志清零 (half transfer flag clear for channel 3)

位 9 **CTCIF3**：通道 3 的传输完成标志清零 (transfer complete flag clear for channel 3)

位 8 **CGIF3**：通道 3 的全局中断标志清零 (global interrupt flag clear for channel 3)

位 7 **CTEIF2**：通道 2 的传输错误标志清零 (transfer error flag clear for channel 2)

位 6 **CHTIF2**：通道 2 的半传输标志清零 (half transfer flag clear for channel 2)

位 5 **CTCIF2**：通道 2 的传输完成标志清零 (transfer complete flag clear for channel 2)

位 4 **CGIF2**：通道 2 的全局中断标志清零 (global interrupt flag clear for channel 2)

位 3 **CTEIF1**：通道 1 的传输错误标志清零 (transfer error flag clear for channel 1)

位 2 **CHTIF1**：通道 1 的半传输标志清零 (half transfer flag clear for channel 1)

位 1 **CTCIF1**：通道 1 的传输完成标志清零 (transfer complete flag clear for channel 1)

位 0 **CGIF1**：通道 1 的全局中断标志清零 (global interrupt flag clear for channel 1)

11.6.3 DMA 通道 x 配置寄存器 (DMA_CCRx)

DMA channel x configuration register

偏移地址：0x08 + 0x14 * (x - 1)，（x = 1 到 7）

复位值：0x0000 0000

EN = 1 时，寄存器位域/位 MEM2MEM、PL[1:0]、MSIZE[1:0]、PSIZE[1:0]、MINC、PINC 和 DIR 为只读。

MEM2MEM 和 CIRC 位的状态不能同时处于高电平。

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.	MEM2MEM	PL[1:0]		MSIZE[1:0]		PSIZE[1:0]		MINC	PINC	CIRC	DIR	TEIE	HTIE	TCIE	EN
	rw	rw	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 **MEM2MEM**：存储器到存储器模式 (memory-to-memory mode)

0：禁止

1：使能

注： 此位由软件置 1 和清零。
使能通道 (EN = 1) 后禁止写入。
使能通道 (EN = 1) 后为只读。

位 13:12 **PL[1:0]**：优先级 (priority level)

00：低

01：中

10：高

11：非常高

注： 此位域由软件置 1 和清零。
使能通道 (EN = 1) 后禁止写入。
使能通道 (EN = 1) 后为只读。

位 11:10 **MSIZE[1:0]**：存储器大小 (memory size)

定义到标识存储器的每次 DMA 传输的数据大小。

在存储器到存储器模式下，如果 DIR = 1，则该位域标识存储器源；如果 DIR = 0，则该位域标识存储器目标。

在外设到外设模式下，如果 DIR = 1，则该位域标识外设源；如果 DIR = 0，则该位域标识外设目标。

00：8 位

01：16 位

10：32 位

11：保留

注： 此位域由软件置 1 和清零。
使能通道 (EN = 1) 后禁止写入。
使能通道 (EN = 1) 后为只读。

位 9:8 **PSIZE[1:0]**：外设大小 (peripheral size)

定义到标识外设的每次 DMA 传输的数据大小。

在存储器到存储器模式下，如果 DIR = 1，则该位域标识存储器目标；如果 DIR = 0，则该位域标识存储器源。

在外设到外设模式下，如果 DIR = 1，则该位域标识外设目标；如果 DIR = 0 则该位域标识外设源。

00：8 位

01：16 位

10：32 位

11：保留

注： 此位域由软件置 1 和清零。
使能通道 (EN = 1) 后禁止写入。
使能通道 (EN = 1) 后为只读。

位 7 **MINC**：存储器递增模式 (memory increment mode)

定义到标识存储器的每次 DMA 传输的递增模式。

在存储器到存储器模式下，如果 DIR = 1，则该位域标识存储器源；如果 DIR = 0，则该位域标识存储器目标。

在外设到外设模式下，如果 DIR = 1，则该位域标识外设源；如果 DIR = 0，则该位域标识外设目标。

0：禁止

1：使能

注：　此位由软件置 1 和清零。
使能通道 (EN = 1) 后禁止写入。
使能通道 (EN = 1) 后为只读。

位 6 **PINC**：外设递增模式 (peripheral increment mode)

定义到标识外设的每次 DMA 传输的递增模式。

在存储器到存储器模式下，如果 DIR = 1，则该位域标识存储器目标；如果 DIR = 0，则该位域标识存储器源。

在外设到外设模式下，如果 DIR = 1，则该位域标识外设目标；如果 DIR = 0 则该位域标识外设源。

0：禁止

1：使能

注：　此位由软件置 1 和清零。
使能通道 (EN = 1) 后禁止写入。
使能通道 (EN = 1) 后为只读。

位 5 **CIRC**：循环模式 (circular mode)

0：禁止

1：使能

注：　此位由软件置 1 和清零。
使能通道 (EN = 1) 后禁止写入。
使能通道 (EN = 1) 后并非只读。

位 4 **DIR**：数据传输方向 (data transfer direction)

该位仅在存储器到外设和外设到存储器模式下才能置 1。

0：从外设读取

- 源属性由 PSIZE 和 PINC 以及 DMA_CPARx 寄存器定义。这同样适用于存储器到存储器模式。
- 目标属性由 MSIZE 和 MINC 以及 DMA_CMARx 寄存器定义。这同样适用于外设到外设模式。

1：从存储器读取

- 目标属性由 PSIZE 和 PINC 以及 DMA_CPARx 寄存器定义。这同样适用于存储器到存储器模式。
- 源属性由 MSIZE 和 MINC 以及 DMA_CMARx 寄存器定义。这同样适用于外设到外设模式。

注：　此位由软件置 1 和清零。
使能通道 (EN = 1) 后禁止写入。
使能通道 (EN = 1) 后为只读。

位 3 **TEIE**：传输错误中断使能 (transfer error interrupt enable)

0：禁止

1：使能

注：　此位由软件置 1 和清零。
使能通道 (EN = 1) 后禁止写入。
使能通道 (EN = 1) 后并非只读。

位 2 **HTIE**：半传输中断使能 (half transfer interrupt enable)

0：禁止

1：使能

注： 此位由软件置 1 和清零。

使能通道 (EN = 1) 后禁止写入。

使能通道 (EN = 1) 后并非只读。

位 1 **TCIE**：传输完成中断使能 (transfer complete interrupt enable)

0：禁止

1：使能

注： 此位由软件置 1 和清零。

使能通道 (EN = 1) 后禁止写入。

使能通道 (EN = 1) 后并非只读。

位 0 **EN**：通道使能 (channel enable)

发生通道传输错误后，该位由硬件清零。在 DMA_ISR 寄存器的 TEIFx 位清零（通过将 DMA_IFCR 寄存器的 CTEIFx 位置 1）前，该位不能再次由软件置 1（通道 x 重新激活）。

0：禁止

1：使能

注： 此位由软件置 1 和清零。

11.6.4 DMA 通道 x 待传输数据数量寄存器 (DMA_CNDTRx)

DMA channel x number of data to transfer register

偏移地址：0x0C + 0x14 * (x - 1)，（x = 1 到 7）

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
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]**：待传输的数据数 (number of data to transfer)（0 到 2^{16} - 1）

通道使能后，该位域由硬件更新：

– 在每次 DMA“先读后写”传输后，该位域都会递减，指示剩余的待传输数据项数目。

– 如果通道未处于循环模式（DMA_CCRx 寄存器中的 CIRC = 0），则在达到设定的待传输数据数目时，该位域会保持为零。

– 如果通道处于循环模式 (CIRC = 1)，则在传输完成后该位域会自动重新装载之前设定的值。

如果该位域为零，则无论通道状态如何（使能或未使能），都不会处理任何传输。

注： 此位域由软件置 1 和清零。

使能通道 (EN = 1) 后禁止写入。

使能通道 (EN = 1) 后为只读。

11.6.5 DMA 通道 x 外设地址寄存器 (DMA_CPARx)

DMA channel x peripheral address register

偏移地址：0x10 + 0x14 * (x - 1)，（x = 1 到 7）

复位值：0x0000 0000

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

位 31:0 **PA[31:0]**：外设地址 (peripheral address)

其中包含读/写数据的外设数据寄存器的基址。

PSIZE[1:0] = 01（16 位）时，忽略 PA[31:0] 的位 0。访问将自动对齐到半字地址。

PSIZE = 10（32 位）时，忽略 PA[31:0] 的位 1 和位 0。访问将自动对齐到字地址。

在存储器到存储器模式下，如果 DIR = 1，则该寄存器标识存储器目标地址；如果 DIR = 0，则该寄存器标识存储器源地址。

在外设到外设模式下，如果 DIR = 1，则该寄存器标识外设目标地址；如果 DIR = 0，则该寄存器标识外设源地址。

注：　此寄存器由软件置 1 和清零。

使能通道 (EN = 1) 后禁止写入。

使能通道 (EN = 1) 后并非只读。

11.6.6 DMA 通道 x 存储器地址寄存器 (DMA_CMARx)

DMA channel x memory address register

偏移地址：0x14 + 0x14 * (x - 1)，（x = 1 到 7）

复位值：0x0000 0000

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

位 31:0 **MA[31:0]**：外设地址 (peripheral address)

其中包含读/写数据的存储器的基址。

MSIZE[1:0] = 01（16 位）时，忽略 MA[31:0] 的位 0。访问将自动对齐到半字地址。

MSIZE = 10（32 位）时，忽略 MA[31:0] 的位 1 和位 0。访问将自动对齐到字地址。

在存储器到存储器模式下，如果 DIR = 1，则该寄存器标识存储器源地址；如果 DIR = 0，则该寄存器标识存储器目标地址。

在外设到外设模式下，如果 DIR = 1，则该寄存器标识外设源地址；如果 DIR = 0，则该寄存器标识外设目标地址。

注：　此寄存器由软件置 1 和清零。

使能通道 (EN = 1) 后禁止写入。

使能通道 (EN = 1) 后并非只读。

11.6.7 DMA 寄存器映射和复位值

表 46 给出了 DMA 寄存器映射和复位值。

表 46. 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_ISR	Res.	Res.	Res.	Res.	TEIF7	HTIF7	TCIF7	GIF7	TEIF6	HTIF6	TCIF6	GIF6	TEIF5	HTIF5	TCIF5	GIF5	TEIF4	HTIF4	TCIF4	GIF4	TEIF3	HTIF3	TCIF3	GIF3	TEIF2	HTIF2	TCIF2	GIF2	TEIF1	HTIF1	TCIF1	GIF1
	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
0x004	DMA_IFCR	Res.	Res.	Res.	Res.	CTEIF7	CHTIF7	CTCIF7	CGIF7	CTEIF6	CHTIF6	CTCIF6	CGIF6	CTEIF5	CHTIF5	CTCIF5	CGIF5	CTEIF4	CHTIF4	CTCIF4	CGIF4	CTEIF3	CHTIF3	CTCIF3	CGIF3	CTEIF2	CHTIF2	CTCIF2	CGIF2	CTEIF1	CHTIF1	CTCIF1	CGIF1
	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
0x008	DMA_CCR1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MEM2MEM	PL[1:0]		MSIZE[1:0]		PSIZE[1:0]		MINC	PINC	CIRC	DIR	TEIE	HTIE	TCIE	EN
	Reset value																		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x00C	DMA_CNDTR1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	NDTR[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x010	DMA_CPAR1	PA[31:0]																															
	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
0x014	DMA_CMAR1	MA[31:0]																															
	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
0x018	Reserved	Reserved																															
0x01C	DMA_CCR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MEM2MEM	PL[1:0]		MSIZE[1:0]		PSIZE[1:0]		MINC	PINC	CIRC	DIR	TEIE	HTIE	TCIE	EN
	Reset value																		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x020	DMA_CNDTR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	NDTR[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x024	DMA_CPAR2	PA[31:0]																															
	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
0x028	DMA_CMAR2	MA[31:0]																															
	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
0x02C	Reserved	Reserved																															
0x030	DMA_CCR3	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MEM2MEM	PL[1:0]		MSIZE[1:0]		PSIZE[1:0]		MINC	PINC	CIRC	DIR	TEIE	HTIE	TCIE	EN
	Reset value																		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x034	DMA_CNDTR3	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	NDTR[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x038	DMA_CPAR3	PA[31:0]																															
	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
0x03C	DMA_CMAR3	MA[31:0]																															
	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
0x040	Reserved	Reserved																															
0x044	DMA_CCR4	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MEM2MEM	PL[1:0]		MSIZE[1:0]		PSIZE[1:0]		MINC	PINC	CIRC	DIR	TEIE	HTIE	TCIE	EN
	Reset value																		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

表 46. DMA 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
0x048	DMA_CNDTR4	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	NDTR[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x04C	DMA_CPAR4	PA[31:0]																															
	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
0x050	DMA_CMAR4	MA[31:0]																															
	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
0x054	Reserved	Reserved																															
0x058	DMA_CCR5	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MEM2MEM	PL[1:0]		MSIZE[1:0]		PSIZE[1:0]		MINC	PINC	CIRC	DIR	TEIE	HTIE	TCIE	EN
	Reset value																		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x05C	DMA_CNDTR5	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	NDTR[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x060	DMA_CPAR5	PA[31:0]																															
	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
0x064	DMA_CMAR5	MA[31:0]																															
	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
0x068	保留	Reserved																															
0x06C	DMA_CCR6	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MEM2MEM	PL[1:0]		MSIZE[1:0]		PSIZE[1:0]		MINC	PINC	CIRC	DIR	TEIE	HTIE	TCIE	EN
	Reset value																		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x070	DMA_CNDTR6	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	NDTR[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x074	DMA_CPAR6	PA[31:0]																															
	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
0x078	DMA_CMAR6	MA[31:0]																															
	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
0x07C	Reserved	Reserved																															
0x080	DMA_CCR7	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MEM2MEM	PL[1:0]		MSIZE[1:0]		PSIZE[1:0]		MINC	PINC	CIRC	DIR	TEIE	HTIE	TCIE	EN
	Reset value																		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x084	DMA_CNDTR7	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	NDTR[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x088	DMA_CPAR7	PA[31:0]																															
	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
0x08C	DMA_CMAR7	MA[31:0]																															
	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*第 2.2.2 节*。

12 DMA 请求复用器 (DMAMUX)

12.1 简介

外设通过设置其 DMA 请求信号来指示存在 DMA 传输请求。DMA 控制器会处理 DMA 请求并生成 DMA 确认信号，而且相应的 DMA 请求信号也将变为无效，但在此之前 DMA 请求一直处于挂起状态。

在本文档中，DMA 请求/确认协议所需的控制信号组并未明确列出和说明，而是称作 DMA 请求线。

DMAMUX 请求复用器可在产品的外设和 DMA 控制器之间重新配置（路由）DMA 请求线。该路由功能通过可编程的多通道 DMA 请求线复用器来确保实现。每条通道可不受限制地选择一个 DMA 请求线，或者按照与来自其 DMAMUX 同步输入的事件相同步的方式选择一个 DMA 请求线。此外，DMAMUX 还可用作其输入触发信号的可编程事件的 DMA 请求发生器。

*第 12.3.1 节*规定了 DMAMUX 实例的数量及其主要特性。

DMAMUX 请求复用器输入到外设 DMA 请求线和到 DMAMUX 请求发生器输出的分配、DMAMUX 请求复用器输出到 DMA 控制器通道的分配以及 DMAMUX 同步和触发输入到内部和外部信号的分配取决于产品实现，具体在*第 12.3.2 节*中进行了介绍。

12.2 DMAMUX 主要特性

- 14 通道可编程 DMA 请求线复用器输出
- 4 通道 DMA 请求发生器
- 26 个 DMA 请求发生器的触发输入
- 26 个同步输入
- 每个 DMA 请求发生器通道均具有：
  - DMA 请求触发输入选择器
  - DMA 请求计数器
  - 所选 DMA 请求触发输入的事件溢出标志
- 每个 DMA 请求线复用器通道输出均具有：
  - 89 个来自外设的输入 DMA 请求线
  - 一个 DMA 请求线输出
  - 同步输入选择器
  - DMA 请求计数器
  - 所选同步输入的事件溢出标志
  - 一个事件输出，用于 DMA 请求链接

12.3 DMAMUX 实现

12.3.1 DMAMUX 实例化

DMAMUX 通过下表中列出的硬件配置参数进行实例化。

表 47. DMAMUX 实例化

特性	DMAMUX
DMAMUX 输出请求通道数	14
DMAMUX 请求发生器通道数	4
DMAMUX 请求触发输入数	26
DMAMUX 同步输入数	26
DMAMUX 外设请求输入数	89

12.3.2 DMAMUX 映射

资源到 DMAMUX 的映射通过硬接线实现。

DMAMUX 与 DMA1 和 DMA2 配合使用：

- DMAMUX 通道 0 到 6 与 DMA1 通道 0 到 6 相连
- DMAMUX 通道 7 到 13 与 DMA2 通道 0 到 6 相连

表 48. DMAMUX：复用器输入到资源的分配

DMA 请求 MUX 输入	资源	DMA 请求 MUX 输入	资源	DMA 请求 MUX 输入	资源
1	dmamux_req_gen0	44	TIM1_CH3	87	DFSDM1_FLT1
2	dmamux_req_gen1	45	TIM1_CH4	88	DFSDM1_FLT2
3	dmamux_req_gen2	46	TIM1_UP	89	DFSDM1_FLT3
4	dmamux_req_gen3	47	TIM1_TRIG	90	DCMI
5	ADC1	48	TIM1_COM	91	AES_IN
6	DAC1	49	TIM8_CH1	92	AES_OUT
7	DAC2	50	TIM8_CH2	93	HASH_IN
8	TIM6_UP	51	TIM8_CH3	94	保留
9	TIM7_UP	52	TIM8_CH4	95	保留
10	SPI1_RX	53	TIM8_UP	96	保留
11	SPI1_TX	54	TIM8_TRIG	97	保留
12	SPI2_RX	55	TIM8_COM	98	保留
13	SPI2_TX	56	TIM2_CH1	99	保留
14	SPI3_RX	57	TIM2_CH2	100	保留
15	SPI3_TX	58	TIM2_CH3	101	保留
16	I2C1_RX	59	TIM2_CH4	102	保留

表 48. DMAMUX：复用器输入到资源的分配（续）

DMA 请求 MUX 输入	资源	DMA 请求 MUX 输入	资源	DMA 请求 MUX 输入	资源
17	I2C1_TX	60	TIM2_UP	103	保留
18	I2C2_RX	61	TIM3_CH1	104	保留
19	I2C2_TX	62	TIM3_CH2	105	保留
20	I2C3_RX	63	TIM3_CH3	106	保留
21	I2C3_TX	64	TIM3_CH4	107	保留
22	I2C4_RX	65	TIM3_UP	108	保留
23	I2C4_TX	66	TIM3_TRIG	109	保留
24	USART1_RX	67	TIM4_CH1	110	保留
25	USART1_TX	68	TIM4_CH2	111	保留
26	USART2_RX	69	TIM4_CH3	112	保留
27	USART2_TX	70	TIM4_CH4	113	保留
28	USART3_RX	71	TIM4_UP	114	保留
29	USART3_TX	72	TIM5_CH1	115	保留
30	UART4_RX	73	TIM5_CH2	116	保留
31	UART4_TX	74	TIM5_CH3	117	保留
32	UART5_RX	75	TIM5_CH4	118	保留
33	UART5_TX	76	TIM5_UP	119	保留
34	LPUART1_RX	77	TIM5_TRIG	120	保留
35	LPUART1_TX	78	TIM15_CH1	121	保留
36	SAI1_A	79	TIM15_UP	122	保留
37	SAI1_B	80	TIM15_TRIG	123	保留
38	SAI2_A	81	TIM15_COM	124	保留
39	SAI2_B	82	TIM16_CH1	125	保留
40	OCTOSPI1	83	TIM16_UP	126	保留
41	OCTOSPI2	84	TIM17_CH1	127	保留
42	TIM1_CH1	85	TIM17_UP	-	-
43	TIM1_CH2	86	DFSDM1_FLT0	-	-

表 49. DMAMUX：触发输入到资源的分配

触发输入	资源	触发输入	资源
0	EXTI 线 0	16	dmamux_evt0
1	EXTI 线 1	17	dmamux_evt1
2	EXTI 线 2	18	dmamux_evt2
3	EXTI 线 3	19	dmamux_evt3
4	EXTI 线 4	20	LPTIM1_OUT
5	EXTI 线 5	21	LPTIM2_OUT

表 49. DMAMUX：触发输入到资源的分配（续）

触发输入	资源	触发输入	资源
6	EXTI 线 6	22	DSI 撕裂效应
7	EXTI 线 7	23	DSI 刷新结束
8	EXTI 线 8	24	DMA2D 传输结束
9	EXTI 线 9	25	LTDC 线中断
10	EXTI 线 10	26	保留
11	EXTI 线 11	27	保留
12	EXTI 线 12	28	保留
13	EXTI 线 13	29	保留
14	EXTI 线 14	30	保留
15	EXTI 线 15	31	保留

表 50. DMAMUX：同步输入到资源的分配

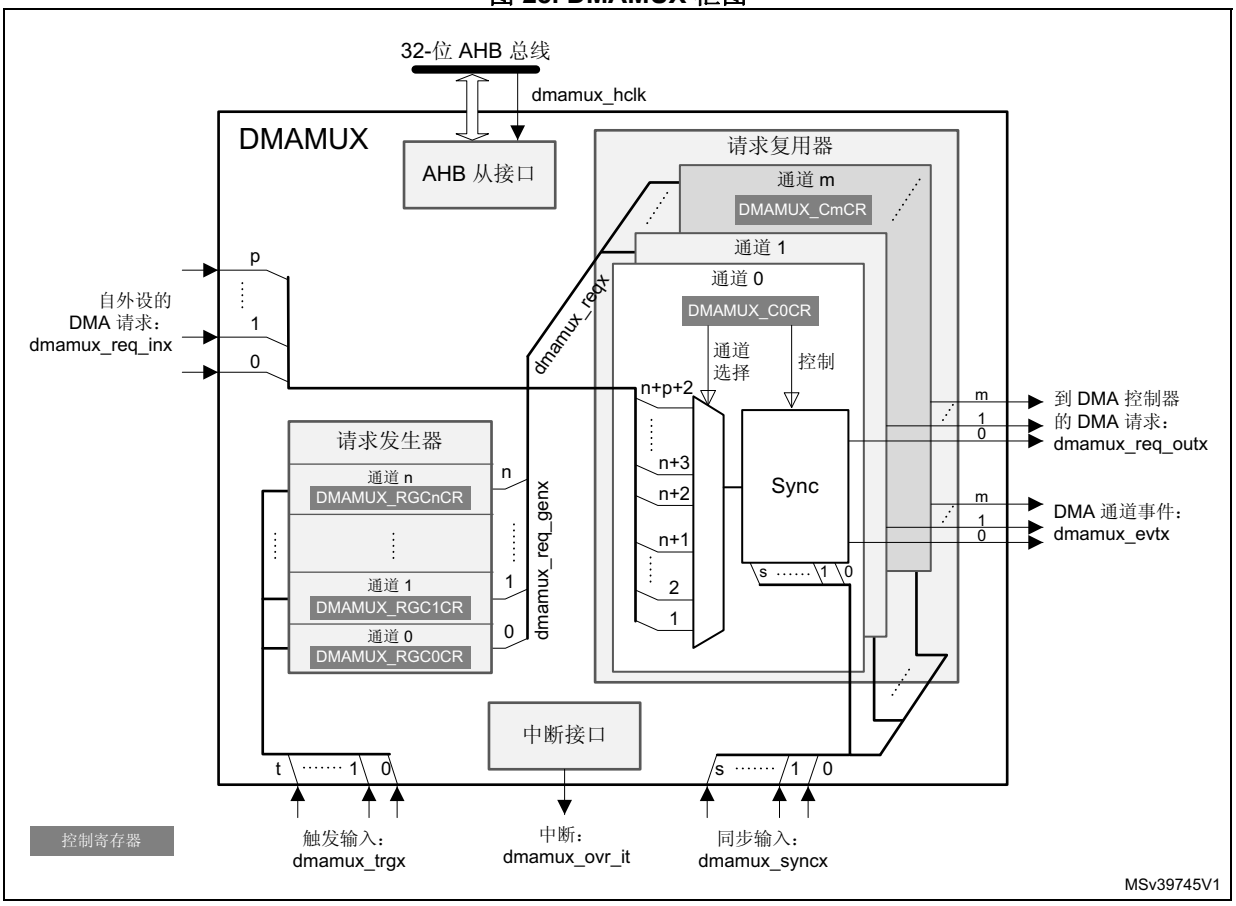
SYNC 输入	资源	SYNC 输入	资源
0	EXTI 线 0	16	dmamux_evt0
1	EXTI 线 1	17	dmamux_evt1
2	EXTI 线 2	18	dmamux_evt2
3	EXTI 线 3	19	dmamux_evt3
4	EXTI 线 4	20	LPTIM1_OUT
5	EXTI 线 5	21	LPTIM2_OUT
6	EXTI 线 6	22	DSI 撕裂效应
7	EXTI 线 7	23	DSI 刷新结束
8	EXTI 线 8	24	DMA2D 传输结束
9	EXTI 线 9	25	LTDC 线中断
10	EXTI 线 10	26	保留
11	EXTI 线 11	27	保留
12	EXTI 线 12	28	保留
13	EXTI 线 13	29	保留
14	EXTI 线 14	30	保留
15	EXTI 线 15	31	保留

12.4 DMAMUX 功能说明

12.4.1 DMAMUX 框图

图 28 显示了 DMAMUX 框图。

图 28. DMAMUX 框图



DMAMUX 具有两个主要子模块：请求线复用器和请求线发生器。

该实现可：

- 分配 DMAMUX 请求复用器子模块输入 (dmamux_reqx)，这些输入来自外设 (dmamux_req_inx) 和 DMAMUX 请求发生器子模块的通道 (dmamux_req_genx)
- 将 DMAMUX 请求输出分配到 DMA 控制器的通道 (dmamux_req_outx)
- 将内部或外部信号分配到 DMA 请求触发输入 (dmamux_trgx)
- 将内部或外部信号分配到同步输入 (dmamux_syncx)

12.4.2 DMAMUX 信号

表 51 列出了 DMAMUX 信号。

表 51. DMAMUX 信号

信号名称	说明
dmamux_hclk	DMAMUX AHB 时钟
dmamux_req_inx	DMAMUX DMA 请求线输入（来自外设）
dmamux_trgx	DMAMUX DMA 请求触发输入（到请求发生器子模块）
dmamux_req_genx	DMAMUX 请求发生器子模块通道输出
dmamux_reqx	DMAMUX 请求复用器子模块输入（来自外设请求和请求发生器通道）
dmamux_syncx	DMAMUX 同步输入（到请求复用器子模块）
dmamux_req_outx	DMAMUX 请求输出（到 DMA 控制器）
dmamux_evtx	DMAMUX 事件输出
dmamux_ovr_it	DMAMUX 溢出中断

12.4.3 DMAMUX 通道

DMAMUX 通道是一个可能包括额外的 DMAMUX 请求发生器通道的 DMAMUX 请求复用器通道，具体取决于所选的请求复用器输入。

DMAMUX 请求复用器通道专用于连接到 DMA 控制器的一个通道。

通道配置流程

请遵循以下顺序来配置 DMAMUX x 通道和相关的 DMA 通道 y：

1. 对 DMA 通道 y 进行完整的设置和配置，但不要使能通道 y。
2. 对相关 DMAMUX y 通道进行完整的设置和配置。
3. 最后，将 DMA y 通道寄存器中的 EN 位置 1 以激活 DMA 通道。

12.4.4 DMAMUX 请求线复用器

DMAMUX 请求复用器及其多条通道可确保 DMA 请求/确认控制信号（称为 DMA 请求线）的实际路由。

每个 DMA 请求线并行连接到 DMAMUX请求线复用器的所有通道。

DMA 请求来自外设或 DMAMUX 请求发生器。

DMAMUX 请求线复用器通道 x 按照 DMAMUX_CxCR 寄存器中的 DMAREQ_ID 字段的配置来选择 DMA 请求线编号。

注： *字段 DMAREQ_ID 为空值表示未选择任何 DMA 请求线。不允许为 DMAMUX 请求线复用器的两个不同通道配置相同的非零 DMAREQ_ID。*

除了 DMA 请求选择外，还可根据需要配置和使能同步模式和/或事件生成。

同步模式和通道事件生成

每个 DMAMUX 请求线复用器通道 x 可单独同步，方法是将 DMAMUX_CxCR 寄存器中的同步使能 (SE) 位置 1。

DMAMUX 具有多个同步输入。同步输入并行连接到请求复用器的所有通道。

同步输入通过给定通道 x 的 DMAMUX_CxCR 寄存器中的 SYNC_ID 字段选择。

当某个通道处于此同步模式时，如果通过 DMAMUX_CxCR 寄存器的 SPOL[1:0] 字段检测到所选输入同步信号上的可编程上升沿/下降沿，则所选输入 DMA 请求线信号将被传送到复用器通道输出。

此外，DMAMUX 请求复用器内部有一个可编程 DMA 请求计数器，可用于实现通道请求输出生成功能以及事件生成功能。通道 x 输出上的事件生成功能通过 DMAMUX_CxCR 寄存器的 EGE 位（事件生成使能）使能。

如*图 30* 所示，检测到同步输入的边沿后，挂起的所选输入 DMA 请求线将连接到 DMAMUX 复用器通道 x 输出。

注： *如果在不存在挂起的所选输入 DMA 请求线的情况下，发生同步事件，则会将其丢弃。再次发生同步事件之前，后续有效的输入请求线将不会连接至 DMAMUX 复用器通道输出。*

此后，DMA 控制器每次处理连接的 DMAMUX 请求时（已处理请求被禁止），DMAMUX 请求计数器都会递减。下溢时，DMA 请求计数器将自动装入 DMAMUX_CxCR 寄存器的 NBREQ 字段中的值，输入 DMA 请求线将与复用器通道 x 输出断开。

因此，检测到同步事件后传送到复用器通道 x 输出的 DMA 请求数等于 NBREQ 字段中的值加 1。

注： *只能通过软件在相应复用器通道 x 的同步使能位 SE 和事件生成使能位 EGE 均禁止时写入 NBREQ 字段的值。*

图 29. DMAMUX 请求线复用器通道的同步模式

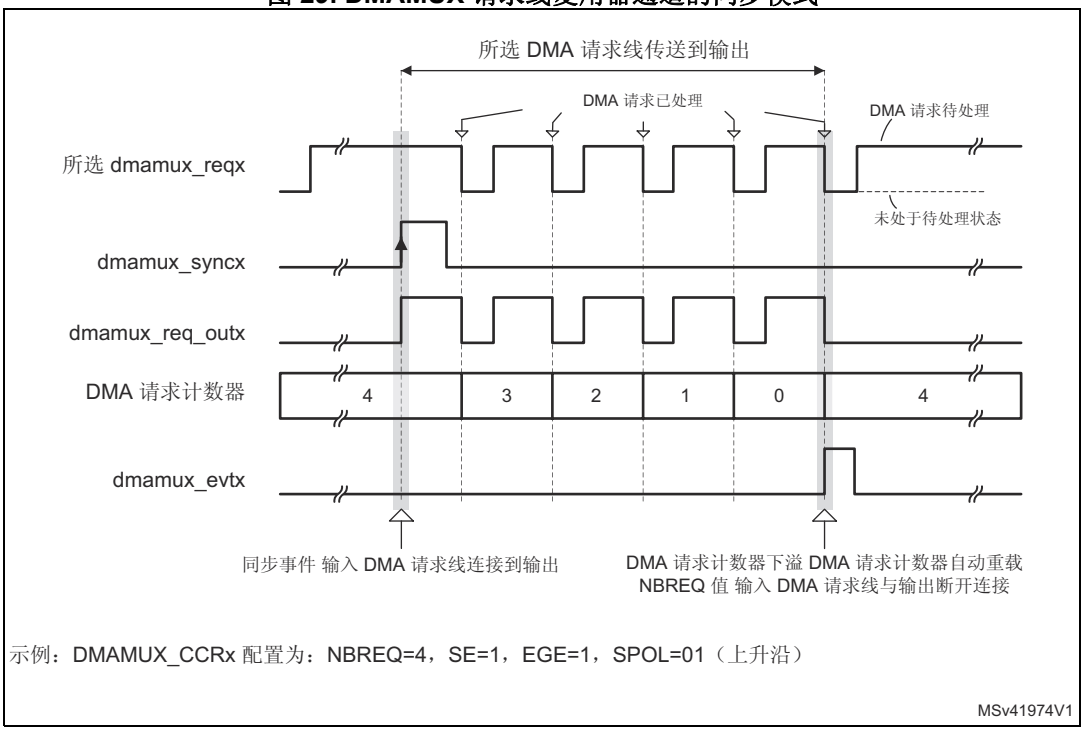
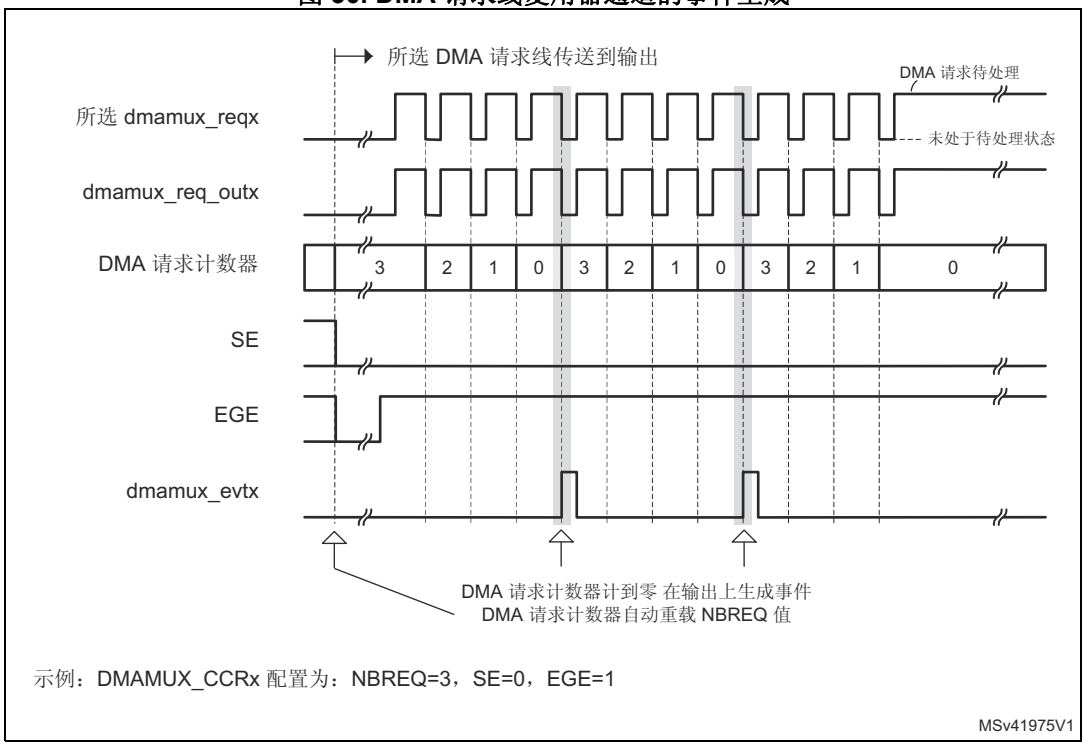


图 30. DMA 请求线复用器通道的事件生成



如果 EGE 使能，当复用器通道的 DMA 请求计数器自动重新装入已编程 NBREQ 字段的值时，复用器通道会生成一个通道事件（即，一个 AHB 时钟周期的脉冲），如*图 29* 和*图 30* 所示。

注： *如果 EGE 使能且 NBREQ = 0，则在每个处理的 DMA 请求后都会生成一个事件。*

注： *如果边沿后的状态保持稳定的时间持续 2 个 AHB 时钟周期以上，则会检测到同步事件（边沿）。*

写入 DMAMUX_CxCR 寄存器后，同步事件将被屏蔽 3 个 AHB 时钟周期。

同步溢出和中断

如果在 DMA 请求计数器的值小于复用器通道 x 已编程的 NBREQ 字段值时发生新的同步事件，则 DMAMUX_CSR 状态寄存器中的同步溢出标志位 SOFx 将置 1。

注： *DMA 控制器的相关通道使用结束后，应禁止请求复用器通道 x 同步 (DMAMUX_CxCR.SE = 0)。否则，在新检测到的同步事件后，由于未从 DMA 控制器收到 DMA 确认（即无已处理请求），将会发生同步溢出。*

溢出标志 SOFx 的复位方式是将 DMAMUX_CFR 寄存器中的相关清除同步溢出标志位 CSOFx 置 1。

如果 DMAMUX_CxCR 寄存器中的同步溢出中断使能位 SOIE 置 1，则将同步溢出标志置 1 会产生中断。

12.4.5 DMAMUX 请求发生器

DMAMUX 请求发生器会在其 DMA 请求触发输入上出现触发事件后产生 DMA 请求。

DMAMUX 请求发生器有多个通道。DMA 请求触发输入并行连接到所有通道。

DMAMUX 请求发生器通道的输出是 DMAMUX 请求线复用器的输入。

每个 DMAMUX 请求发生器通道 x 在相应的 DMAMUX_RGxCR 寄存器中都有一个使能位 GE（发生器使能）。

DMAMUX 请求发生器通道 x 的 DMA 请求触发输入通过相应 DMAMUX_RGxCR 寄存器中的 SIG_ID（触发信号 ID）字段选择。

DMA 请求触发输入上的触发事件可以是上升沿、下降沿或任一边沿。有效边沿通过相应 DMAMUX_RGxCR 寄存器中的 GPOL（发生器极性）字段选择。

触发事件后，相应发生器通道开始在其输出上生成 DMA 请求。连接的 DMA 控制器每次处理 DMAMUX 生成的请求时（已处理请求被禁止），内置（DMAMUX 请求发生器内）DMA 请求计数器都会递减。下溢时，请求发生器通道会停止生成 DMA 请求，DMA 请求计数器将在出现下一个触发事件时自动装入其设定值。

因此，触发事件后生成的 DMA 请求的数量为 GNBREQ+1。

注： *只能通过软件在相应发生器通道 x 的使能位 GE 被禁止时写入 GNBREQ 字段的值。*

如果边沿后的状态保持稳定的时间持续 2 个 AHB 时钟周期以上，则会检测到触发事件（边沿）。

写入 DMAMUX_RGxCR 寄存器后，触发事件将被屏蔽 3 个 AHB 时钟周期。

触发溢出和中断

如果在 DMAMUX 请求发生器计数器的值小于相应请求发生器通道 x 已编程的 GNBREQ 字段值时发生新的 DMA 请求触发事件，并且请求发生器通道 x 通过 GE 使能，则状态 DMAMUX_RGSR 寄存器中的请求触发事件溢出标志位 OFx 将通过硬件置为有效。

注： *DMA 控制器的相关通道使用结束后，应禁止请求发生器通道 x (DMAMUX_RGxCR.GE = 0)。否则，在新检测到的触发事件后，由于未从 DMA 收到确认（即无已处理请求），将会发生触发溢出。*

溢出标志 OFx 的复位方式是将 DMAMUX_RGCFR 寄存器中的相关清除溢出标志位 COFx 置 1。

如果 DMAMUX_RGxCR 寄存器中的 DMA 请求触发事件溢出中断使能位 OIE 置 1，则将 DMAMUX 请求触发溢出标志置 1 会产生中断。

12.5 DMAMUX 中断

发生以下事件时会产生中断：

- 每个 DMA 请求线复用器通道发生同步事件溢出
- 每个 DMA 请求发生器通道发生触发事件溢出

对任一事件，每个通道的中断使能、状态和清除标志寄存器位均可单独使用。

表 52. DMAMUX 中断

中断信号	中断事件	事件标志	清除位	使能位
dmamuxovr_it	DMAMUX 请求线复用器的通道 x 上发生同步事件溢出	SOFx	CSOFx	SOIE
	DMAMUX 请求发生器的通道 x 上发生触发事件溢出	OFx	COFx	OIE

12.6 DMAMUX 寄存器

有关 DMAMUX 基址的信息，请参见包括寄存器边界地址的表格。

DMAMUX 寄存器可按（8 位）字节、（16 位）半字或（32 位）字访问。地址应与数据大小对齐。

12.6.1 DMAMUX 请求线复用器通道 x 配置寄存器 (DMAMUX_CxCR)

DMAMUX request line multiplexer channel x configuration register

偏移地址：0x04 * x（x = 0 到 13）

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	SYNC_ID[4:0]					NBREQ[4:0]					SPOL[1:0]		SE
			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.	Res.	Res.	Res.	EGE	SOIE	Res.	DMAREQ_ID[6:0]						
						rw	rw		rw	rw	rw	rw	rw	rw	rw

位 31:29 保留，必须保持复位值。

位 28:24 **SYNC_ID[4:0]**：同步标识 (Synchronization identification)
选择同步输入（请参见*表 50：DMAMUX：同步输入到资源的分配*）。

位 23:19 **NBREQ[4:0]**：要转发的 DMA 请求数减 1 (Number of DMA requests minus 1 to forward)
定义在同步事件后要转发到 DMA 控制器的 DMA 请求的数量，和/或生成输出事件前的 DMA 请求的数量。
只有当 SE 和 EGE 位均为低电平时，才能写入该字段。

位 18:17 **SPOL[1:0]**：同步极性 (Synchronization polarity)
定义所选同步输入的边沿极性：
00：无事件，即，无同步也无检测
01：上升沿
10：下降沿
11：上升沿和下降沿

位 16 **SE**：同步使能 (Synchronization enable)
0：禁止同步
1：使能同步

位 15:10 保留，必须保持复位值。

位 9 **EGE**：事件生成使能 (Event generation enable)
0：禁止事件生成
1：使能事件生成

位 8 **SOIE**：同步溢出中断使能 (Synchronization overrun interrupt enable)
0：禁止中断
1：使能中断

位 7 保留，必须保持复位值。

位 6:0 **DMAREQ_ID[6:0]**：DMA 请求标识 (DMA request identification)
选择输入 DMA 请求。有关复用器输入到资源的分配的信息，请参见 DMAMUX 表。

12.6.2 DMAMUX 请求线复用器中断通道状态寄存器 (DMAMUX_CSR)

DMAMUX request line multiplexer interrupt channel status register

偏移地址：0x080

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.	SOF13	SOF12	SOF11	SOF10	SOF9	SOF8	SOF7	SOF6	SOF5	SOF4	SOF3	SOF2	SOF1	SOF0
		r	r	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 13:0 **SOF[13:0]**：同步溢出事件标志 (Synchronization overrun event flag)
当 DMA 请求线复用器通道 x 上发生同步事件且 DMA 请求计数器的值小于 NBREQ 时，该标志将置 1。
该标志的清零方式是向 DMAMUX_CFR 寄存器中的相应 CSOFx 位写入 1。

12.6.3 DMAMUX 请求线复用器中断清除标志寄存器 (DMAMUX_CFR)

DMAMUX request line multiplexer interrupt clear flag register

偏移地址：0x084

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.	CSOF 13	CSOF 12	CSOF 11	CSOF 10	CSOF 9	CSOF 8	CSOF 7	CSOF 6	CSOF 5	CSOF 4	CSOF 3	CSOF 2	CSOF 1	CSOF 0
		w	w	w	w	w	w	w	w	w	w	w	w	w	w

位 31:14 保留，必须保持复位值。

位 13:0 **CSOF[13:0]**：清除同步溢出事件标志 (Clear synchronization overrun event flag)
将 1 写入每个位时，DMAMUX_CSR 寄存器中相应的溢出标志 SOFx 将清零。

12.6.4 DMAMUX 请求发生器通道 x 配置寄存器 (DMAMUX_RGxCR)

DMAMUX request generator channel x configuration register

偏移地址：0x100 + 0x04 * (x-0)（x = 0 到 3）

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.	GNBREQ[4:0]					GPOL[1:0]		GE
								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.	Res.	Res.	Res.	Res.	OIE	Res.	Res.	Res.	SIG_ID[4:0]				
							rw				rw	rw	rw	rw	rw

位 31:24 保留，必须保持复位值。

位 23:19 **GNBREQ[4:0]**：要生成的 DMA 请求数（减 1）(Number of DMA requests to be generated (minus 1))

定义触发事件后生成的 DMA 请求的数量。生成的 DMA 请求的实际数量为 GNBREQ+1。

注： 只有在 GE 位禁止时才能写入此字段。

位 18:17 **GPOL[1:0]**：DMA 请求发生器触发极性 (DMA request generator trigger polarity)

定义所选触发输入的边沿极性

00：无事件。即，无触发检测和生成

01：上升沿

10：下降沿

11：上升沿和下降沿

位 16 **GE**：DMA 请求发生器通道 x 使能 (DMA request generator channel x enable)

0：禁止 DMA 请求发生器通道 x

1：使能 DMA 请求发生器通道 x

位 15:9 保留，必须保持复位值。

位 8 **OIE**：触发溢出中断使能 (Trigger overrun interrupt enable)

0：禁止在出现触发溢出事件时产生中断。

1：使能在出现触发溢出事件时产生中断。

位 7:5 保留，必须保持复位值。

位 4:0 **SIG_ID[4:0]**：信号标识 (Signal identification)

选择用于 DMA 请求发生器的通道 x 的 DMA 请求触发输入。

12.6.5 DMAMUX 请求发生器中断状态寄存器 (DMAMUX_RGSR)

DMAMUX request generator interrupt status register

偏移地址：0x140

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.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OF3	OF2	OF1	OF0
												r	r	r	r

位 31:4 保留，必须保持复位值。

位 3:0 **OF[3:0]**：触发溢出事件标志 (Trigger overrun event flag)

当 DMA 请求发生器通道 x 上发生触发事件且 DMA 请求发生器计数器的值小于 GNBREQ 时，该标志将置 1。

该标志的清零方式是向 DMAMUX_RGCFR 寄存器中的相应 COFx 位写入 1。

12.6.6 DMAMUX 请求发生器中断清除标志寄存器 (DMAMUX_RGCFR)

DMAMUX request generator interrupt clear flag register

偏移地址：0x144

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.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	COF3	COF2	COF1	COF0
												w	w	w	w

位 31:4 保留，必须保持复位值。

位 3:0 **COF[3:0]**：清除触发溢出事件标志 (Clear trigger overrun event flag)

将 1 写入每个位时，DMAMUX_RGSR 寄存器中相应的溢出标志 OFx 将清零。

12.6.7 DMAMUX寄存器映射

下表汇总了 DMAMUX 寄存器和复位值。有关 DMAMUX 寄存器的基本地址，请参见寄存器边界地址表。

表 53. DMAMUX 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	DMAMUX_C0CR	Res.	Res.	Res.	SYNC_ID[4:0]					NBREQ[4:0]					SPOL[1:0]		SE	Res.	Res.	Res.	Res.	Res.	Res.	EGE	SOIE	Res.	DMAREQ_ID[6:0]						
	Reset value				0	0	0	0	0	0	0	0	0	0	0	0	0							0	0		0	0	0	0	0	0	0
0x004	DMAMUX_C1CR	Res.	Res.	Res.	SYNC_ID[4:0]					NBREQ[4:0]					SPOL[1:0]		SE	Res.	Res.	Res.	Res.	Res.	Res.	EGE	SOIE	Res.	DMAREQ_ID[6:0]						
	Reset value				0	0	0	0	0	0	0	0	0	0	0	0	0							0	0		0	0	0	0	0	0	0
0x008	DMAMUX_C2CR	Res.	Res.	Res.	SYNC_ID[4:0]					NBREQ[4:0]					SPOL[1:0]		SE	Res.	Res.	Res.	Res.	Res.	Res.	EGE	SOIE	Res.	DMAREQ_ID[6:0]						
	Reset value				0	0	0	0	0	0	0	0	0	0	0	0	0							0	0		0	0	0	0	0	0	0
0x00C	DMAMUX_C3CR	Res.	Res.	Res.	SYNC_ID[4:0]					NBREQ[4:0]					SPOL[1:0]		SE	Res.	Res.	Res.	Res.	Res.	Res.	EGE	SOIE	Res.	DMAREQ_ID[6:0]						
	Reset value				0	0	0	0	0	0	0	0	0	0	0	0	0							0	0		0	0	0	0	0	0	0
0x010	DMAMUX_C4CR	Res.	Res.	Res.	SYNC_ID[4:0]					NBREQ[4:0]					SPOL[1:0]		SE	Res.	Res.	Res.	Res.	Res.	Res.	EGE	SOIE	Res.	DMAREQ_ID[6:0]						
	Reset value				0	0	0	0	0	0	0	0	0	0	0	0	0							0	0		0	0	0	0	0	0	0
0x014	DMAMUX_C5CR	Res.	Res.	Res.	SYNC_ID[4:0]					NBREQ[4:0]					SPOL[1:0]		SE	Res.	Res.	Res.	Res.	Res.	Res.	EGE	SOIE	Res.	DMAREQ_ID[6:0]						
	Reset value				0	0	0	0	0	0	0	0	0	0	0	0	0							0	0		0	0	0	0	0	0	0
0x018	DMAMUX_C6CR	Res.	Res.	Res.	SYNC_ID[4:0]					NBREQ[4:0]					SPOL[1:0]		SE	Res.	Res.	Res.	Res.	Res.	Res.	EGE	SOIE	Res.	DMAREQ_ID[6:0]						
	Reset value				0	0	0	0	0	0	0	0	0	0	0	0	0							0	0		0	0	0	0	0	0	0
0x01C	DMAMUX_C7CR	Res.	Res.	Res.	SYNC_ID[4:0]					NBREQ[4:0]					SPOL[1:0]		SE	Res.	Res.	Res.	Res.	Res.	Res.	EGE	SOIE	Res.	DMAREQ_ID[6:0]						
	Reset value				0	0	0	0	0	0	0	0	0	0	0	0	0							0	0		0	0	0	0	0	0	0
0x020	DMAMUX_C8CR	Res.	Res.	Res.	SYNC_ID[4:0]					NBREQ[4:0]					SPOL[1:0]		SE	Res.	Res.	Res.	Res.	Res.	Res.	EGE	SOIE	Res.	DMAREQ_ID[6:0]						
	Reset value				0	0	0	0	0	0	0	0	0	0	0	0	0							0	0		0	0	0	0	0	0	0
0x024	DMAMUX_C9CR	Res.	Res.	Res.	SYNC_ID[4:0]					NBREQ[4:0]					SPOL[1:0]		SE	Res.	Res.	Res.	Res.	Res.	Res.	EGE	SOIE	Res.	DMAREQ_ID[6:0]						
	Reset value				0	0	0	0	0	0	0	0	0	0	0	0	0							0	0		0	0	0	0	0	0	0
0x028	DMAMUX_C10CR	Res.	Res.	Res.	SYNC_ID[4:0]					NBREQ[4:0]					SPOL[1:0]		SE	Res.	Res.	Res.	Res.	Res.	Res.	EGE	SOIE	Res.	DMAREQ_ID[6:0]						
	Reset value				0	0	0	0	0	0	0	0	0	0	0	0	0							0	0		0	0	0	0	0	0	0
0x02C	DMAMUX_C11CR	Res.	Res.	Res.	SYNC_ID[4:0]					NBREQ[4:0]					SPOL[1:0]		SE	Res.	Res.	Res.	Res.	Res.	Res.	EGE	SOIE	Res.	DMAREQ_ID[6:0]						
	Reset value				0	0	0	0	0	0	0	0	0	0	0	0	0							0	0		0	0	0	0	0	0	0
0x030	DMAMUX_C12CR	Res.	Res.	Res.	SYNC_ID[4:0]					NBREQ[4:0]					SPOL[1:0]		SE	Res.	Res.	Res.	Res.	Res.	Res.	EGE	SOIE	Res.	DMAREQ_ID[6:0]						
	Reset value				0	0	0	0	0	0	0	0	0	0	0	0	0							0	0		0	0	0	0	0	0	0
0x034	DMAMUX_C13CR	Res.	Res.	Res.	SYNC_ID[4:0]					NBREQ[4:0]					SPOL[1:0]		SE	Res.	Res.	Res.	Res.	Res.	Res.	EGE	SOIE	Res.	DMAREQ_ID[6:0]						
	Reset value				0	0	0	0	0	0	0	0	0	0	0	0	0							0	0		0	0	0	0	0	0	0
0x038 - 0x07C	Reserved	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
0x080	DMAMUX_CSR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SOF13	SOF12	SOF11	SOF10	SOF9	SOF8	SOF7	SOF6	SOF5	SOF4	SOF3	SOF2	SOF1	SOF0
	Reset value																			0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x084	DMAMUX_CFR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CSOF13	CSOF12	CSOF11	CSOF10	CSOF9	CSOF8	CSOF7	CSOF6	CSOF5	CSOF4	CSOF3	CSOF2	CSOF1	CSOF0
	Reset value																			0	0	0	0	0	0	0	0	0	0	0	0	0	0

表 53. DMAMUX 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
0x088 - 0x0FC	Reserved	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
0x100	DMAMUX_RG0CR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	GNBREQ[4:0]					GPOL[1:0]		GE	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OIE	Res.	Res.	Res.	SIG_ID[4:0]				
	Reset value									0	0	0	0	0	0	0	0								0				0	0	0	0	0
0x104	DMAMUX_RG1CR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	GNBREQ[4:0]					GPOL[1:0]		GE	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OIE	Res.	Res.	Res.	SIG_ID[4:0]				
	Reset value									0	0	0	0	0	0	0	0								0				0	0	0	0	0
0x108	DMAMUX_RG2CR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	GNBREQ[4:0]					GPOL[1:0]		GE	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OIE	Res.	Res.	Res.	SIG_ID[4:0]				
	Reset value									0	0	0	0	0	0	0	0								0				0	0	0	0	0
0x10C	DMAMUX_RG3CR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	GNBREQ[4:0]					GPOL[1:0]		GE	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OIE	Res.	Res.	Res.	SIG_ID[4:0]				
	Reset value									0	0	0	0	0	0	0	0								0				0	0	0	0	0
0x110 - 0x13C	Reserved	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
0x140	DMAMUX_RGSR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OF3	OF2	OF1	OF0
	Reset value																													0	0	0	0
0x144	DMAMUX_RGCFR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	COF3	COF2	COF1	COF0
	Reset value																													0	0	0	0
0x148 0x3FC	Reserved	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

13 Chrom-ART Accelerator™ 控制器 (DMA2D)

13.1 DMA2D 简介

Chrom-ART Accelerator™ (DMA2D) 是专用于图像处理的专业 DMA。它可以执行下列操作：

- 用特定颜色填充目标图像的一部分或全部
- 将源图像的一部分（或全部）复制到目标图像的一部分（或全部）中
- 通过像素格式转换将源图像的一部分或全部复制到目标图像的一部分或全部中
- 将像素格式不同的两个源图像部分和/或全部混合，再将结果复制到颜色格式不同的部分或整个目标图像中

在索引颜色模式或直接颜色模式下，支持所有经典颜色编码方案，并支持每像素 4 位到最高 32 位。DMA2D 自身具有专门的 CLUT（颜色查找表）存储器。

13.2 DMA2D 的主要特性

DMA2D 的主要特性有：

- 采用单 AHB 主设备总线架构
- AHB 从设备编程接口支持 8/16/32 位访问（32 位的 CLUT 访问除外）
- 用户可编程工作区大小
- 用户可编程源区域和目标区域的偏移（以像素或字节表示）
- 用户可编程整个存储空间的源地址和目标地址
- 最多支持 2 个源的混合操作
- Alpha 值可修改（源值、固定值或调制的值）
- 用户可编程源颜色格式和目标的颜色格式
- 采用间接或直接颜色编码时，支持多达 11 种颜色格式，且支持每像素 4 位到最高 32 位
- 间接颜色模式下使用 2 个内部存储器存储 CLUT
- 通过 CPU 自动加载 CLUT 或对 CLUT 进行编程
- 用户可编程 CLUT 大小
- 使用内部定时器控制 AHB 带宽
- 支持 6 种工作模式：寄存器到存储器；存储器到存储器；存储器到存储器并执行像素格式转换；存储器到存储器并执行像素格式转换和混合；存储器到存储器并支持像素格式转换、混合以及固定颜色前景层；以及存储器到存储器并支持像素格式转换、混合以及固定颜色背景层
- 可使用固定颜色进行区域填充
- 可从一个区域复制到另一个区域
- 在源图像和目标图像之间进行复制时进行像素格式转换
- 支持从颜色格式不同的两幅源图像复制并混合
- 可进行输出缓冲器字节交换，以支持通过并行接口刷新显示画面
- 可中止并挂起 DMA2D 操作
- 支持在传输用户可编程的目标行时生成水印中断
- 支持发生总线错误或访问冲突时生成中断
- 支持处理完成时生成中断

13.3 DMA2D 功能说明

13.3.1 概述

DMA2D 控制器执行直接存储器传输。作为一个 AHB 主设备，它可以控制 AHB 总线矩阵来启动 AHB 事务。

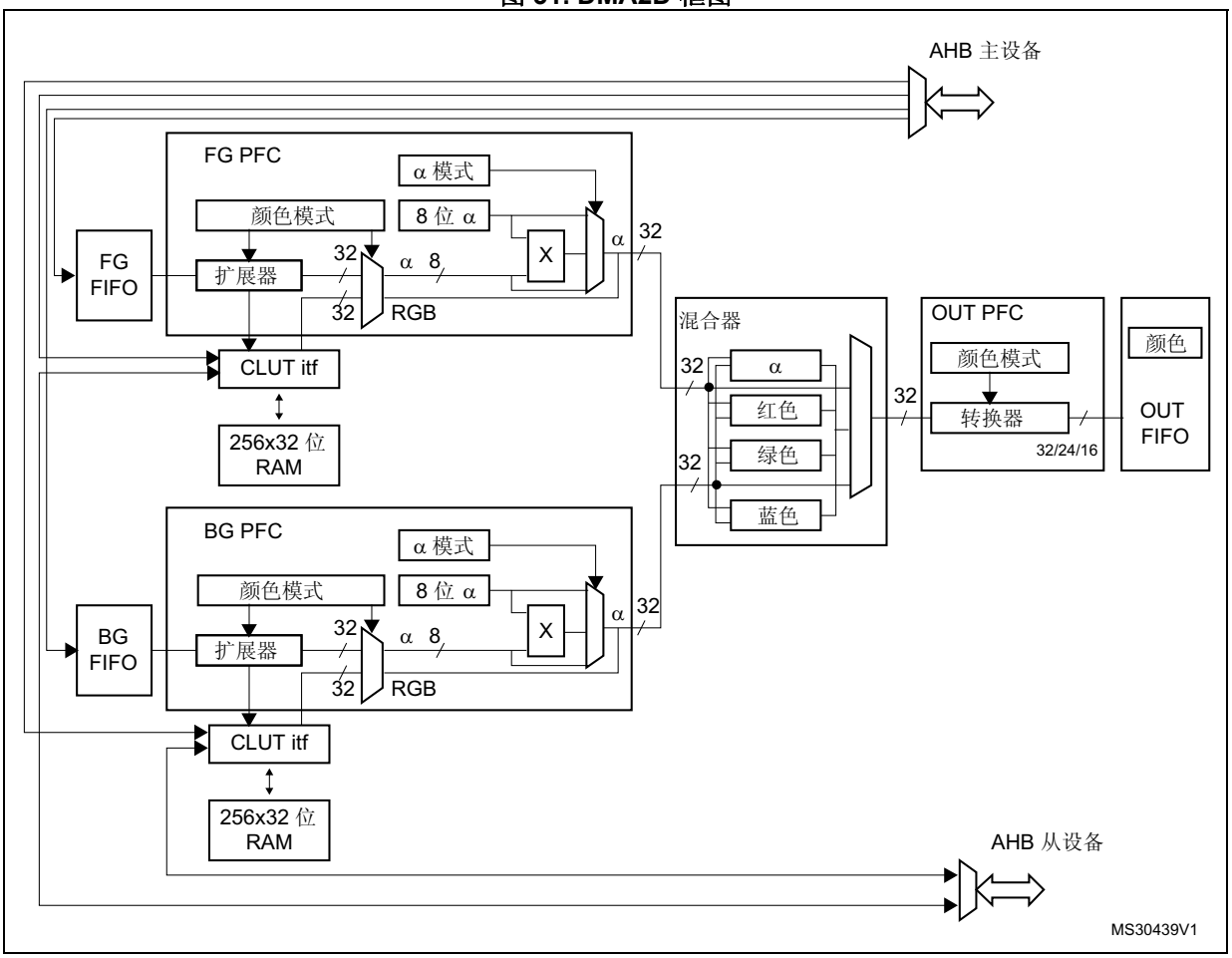
DMA2D 可在以下四种模式下工作：

- 寄存器到存储器
- 存储器到存储器
- 存储器到存储器并执行像素格式转换
- 存储器到存储器并执行像素格式转换和混合
- 存储器到存储器并执行像素格式转换、混合和固定颜色前景层
- 存储器到存储器并支持像素格式转换、混合和固定颜色背景层

AHB 从设备端口用于编程 DMA2D 控制器。

DMA2D 的框图如*图 31：DMA2D 框图*所示。

图 31. DMA2D 框图



13.3.2 DMA2D 控制

通过 DMA2D 控制寄存器 (DMA2D_CR) 配置 DMA2D 控制器，允许选择：

用户应用可以执行下列操作：

- 选择工作模式
- 使能/禁止 DMA2D 中断
- 启动/挂起/中止进行中的数据传输

13.3.3 DMA2D 前景层 FIFO 和背景层 FIFO

DMA2D 前景层 (FG) FIFO 和背景层 (BG) FIFO 获取要复制和/或处理的输入数据。

这些 FIFO 根据相应像素格式转换器 (PFC) 中定义的颜色格式获取像素。

通过如下一组寄存器对它们进行编程：

- DMA2D 前景层存储器地址寄存器 (DMA2D_FGMAR)
- DMA2D 前景层偏移寄存器 (DMA2D_FGOR)
- DMA2D 背景层存储器地址寄存器 (DMA2D_BGMAR)
- DMA2D 背景层偏移寄存器 (DMA2D_BGBOR)
- DMA2D 行数寄存器（行数和每行像素数）(DMA2D_NLR)

DMA2D 在寄存器到存储器模式下工作时，不激活任何 FIFO。

DMA2D 在存储器到存储器模式下工作时（无像素格式转换和混合操作），仅激活 FG FIFO，并将其用作缓冲区。

DMA2D 在存储器到存储器模式下工作时并支持像素格式转换时（无混合操作），不会激活 BG FIFO。

13.3.4 DMA2D 前景层和背景层像素格式转换器 (PFC)

DMA2D 前景层和背景层像素格式转换器 (PFC) 执行像素格式转换，以生成每像素 32 位的值。PFC 还能够修改 alpha 通道。

转换器在第一阶段转换颜色格式。前景层像素和背景层像素的原始颜色格式分别通过 DMA2D_FGPFCCR 和 DMA2D_BGPFCCR 的 CM[3:0] 位来配置。

*表 54：输入时支持的颜色模式*给出了支持的输入格式。

表 54. 输入时支持的颜色模式

CM[3:0]	颜色模式
0000	ARGB8888
0001	RGB888
0010	RGB565
0011	ARGB1555
0100	ARGB4444
0101	L8
0110	AL44
0111	AL88

表 54. 输入时支持的颜色模式（续）

CM[3:0]	颜色模式
1000	L4
1001	A8
1010	A4

颜色格式的编码方式如下：

- Alpha 值字段：透明
  0xFF 值对应不透明像素，0x00 对应透明像素。
- R 字段代表红色
- G 字段代表绿色
- B 字段代表蓝色
- L 字段：亮度
  该字段是 CLUT 的索引，用于检索三个/四个 RGB/ARGB 分量。

如果原始格式为直接颜色模式 (ARGB/RGB)，则通过将 MSB 复制到 LSB 扩展为每通道 8 位。这可以确保转换具有良好的线性。

如果原始格式为间接颜色模式 (L/AL)，则需要使用 CLUT，并且每个像素格式转换器与一个 256 个 32 位条目的 CLUT 相关联。

如果原始格式不包括 alpha 通道，则会自动将 alpha 值设为 0xFF（不透明）。

对于特定的 alpha 模式 A4 和 A8，既不存储颜色信息，也不编制索引。用于生成图像的颜色是固定的，并且在 DMA2D_FGCOLR 寄存器中定义前景层像素的颜色，在 DMA2D_BGCOLR 寄存器中定义背景层像素的颜色。

系统存储器中的字段顺序如*表 55：存储器中的数据顺序*所示。

表 55. 存储器中的数据顺序

颜色模式	@ + 3	@ + 2	@ + 1	@ + 0
ARGB8888	A_0[7:0]	R_0[7:0]	G_0[7:0]	B_0[7:0]
RGB888	B_1[7:0]	R_0[7:0]	G_0[7:0]	B_0[7:0]
	G_2[7:0]	B_2[7:0]	R_1[7:0]	G_1[7:0]
	R_3[7:0]	G_3[7:0]	B_3[7:0]	R_2[7:0]
RGB565	R_1[4:0]G_1[5:3]	G_1[2:0]B_1[4:0]	R_0[4:0]G_0[5:3]	G_0[2:0]B_0[4:0]
ARGB1555	A_1[0]R_1[4:0]G_1[4:3]	G_1[2:0]B_1[4:0]	A_0[0]R_0[4:0]G_0[4:3]	G_0[2:0]B_0[4:0]
ARGB4444	A_1[3:0]R_1[3:0]	G_1[3:0]B_1[3:0]	A_0[3:0]R_0[3:0]	G_0[3:0]B_0[3:0]
L8	L_3[7:0]	L_2[7:0]	L_1[7:0]	L_0[7:0]
AL44	A_3[3:0]L_3[3:0]	A_2[3:0]L_2[3:0]	A_1[3:0]L_1[3:0]	A_0[3:0]L_0[3:0]
AL88	A_1[7:0]	L_1[7:0]	A_0[7:0]	L_0[7:0]
L4	L_7[3:0]L_6[3:0]	L_5[3:0]L_4[3:0]	L_3[3:0]L_2[3:0]	L_1[3:0]L_0[3:0]
A8	A_3[7:0]	A_2[7:0]	A_1[7:0]	A_0[7:0]
A4	A_7[3:0]A_6[3:0]	A_5[3:0]A_4[3:0]	A_3[3:0]A_2[3:0]	A_1[3:0]A_0[3:0]

通过 ARGB8888 模式支持按 32 位对齐 24 位 RGB888。

生成 32 位值后，即可根据 DMA2D_FGPFCCR/DMA2D_BGPFCCR 寄存器的 AM[1:0] 字段修改 alpha 通道，如*表 56：Alpha 模式配置*所示。

Alpha 通道可以：

- 保持不变（不做修改）
- 替换为 DMA2D_FGPFCCR/DMA2D_BGPFCCR 的 ALPHA[7:0] 值
- 或替换为原始 alpha 值与 DMA2D_FGPFCCR/DMA2D_BGPFCCR 的 ALPHA [7:0] 值除以 255 所得商的乘积

表 56. Alpha 模式配置

AM[1:0]	Alpha 模式
00	不做修改
01	替换为 DMA2D_xxPFCCR 中的值
10	替换为原始值与 DMA2D_xxPFCCR 中的值/255 所得商的乘积
11	保留

注： *要支持备用格式，可通过将 DMA2D_FGPFCCR/DMA2D_BGPFCCR 寄存器的 AI 位置 1 将传入的 alpha 值取反。这同样适用于 DMA2D_FGPFCCR/DMA2D_BGPFCCR 和 CLUT 中存储的 Alpha 值。*

此外，还可通过将 DMA2D_FGPFCCR/DMA2D_BGPFCCR 寄存器的 RBS 位置 1 来交换 R 字段和 B 字段。这同样适用于 CLUT 和 DMA2D_FGCOLR/DMA2D_BGCOLR 寄存器中使用的 RGB 顺序。

13.3.5 DMA2D 前景层 FIFO 和背景层 CLUT 接口

CLUT 接口可管理对 CLUT 存储器的访问以及 CLUT 的自动加载。

支持如下三种访问：

- PFC 在像素格式转换期间读取 CLUT
- CPU 对 CLUT 进行数据的读取或写入时，通过 AHB 从设备端口访问 CLUT
- 执行自动加载 CLUT 时，通过 AHB 主设备端口进行 CLUT 写入

可通过两种不同方法执行 CLUT 存储器加载：

- 自动加载

  加载 CLUT 时应遵守以下顺序：

  a) 将 CLUT 地址编程到 DMA2D_FGCMAR 寄存器（前景层 CLUT）或 DMA2D_BGCMAR 寄存器（背景层 CLUT）

  b) 将 CLUT 大小编程到 DMA2D_FGPFCCR 寄存器（前景层 CLUT）或 MA2D_BGPFCCR 寄存器（背景层 CLUT）的 S[7:0] 字段。

  c) 将 MA2D_FGPFCCR 寄存器（前景层 CLUT）或 DMA2D_BGPFCCR 寄存器（背景层 CLUT）的 START 位置 1 以启动传输。自动加载过程期间，不可通过 CPU 访问 CLUT。如果出现冲突，若 DMA2D_CR 中 CAEIE 被置 1，则发生 CLUT 访问错误中断。

- 手动加载

  应用程序必须通过 DMA2D AHB 从设备端口手动编程本地 CLUT 存储器映射到的 CLUT。前景层 CLUT 从偏移地址 0x0400 开始，背景层 CLUT 从偏移地址 0x0800 开始。

CLUT 格式可以是 24 位或 32 位。通过 DMA2D_FGPFCCR 寄存器（前景层 CLUT）或 DMA2D_BGPFCCR 寄存器（背景层 CLUT）的 CCM 位配置格式，如*表 57：支持的 CLUT 颜色模式*所示。

表 57. 支持的 CLUT 颜色模式

CCM	CLUT 颜色模式
0	32 位 ARGB8888
1	24 位 RGB888

*表 58：系统存储器中的 CLUT 数据顺序*给出了系统存储器中 CLUT 数据的组织方式。

表 58. 系统存储器中的 CLUT 数据顺序

CLUT 颜色模式	@ + 3	@ + 2	@ + 1	@ + 0
ARGB8888	A_0[7:0]	R_0[7:0]	G_0[7:0]	B_0[7:0]
RGB888	B_1[7:0]	R_0[7:0]	G_0[7:0]	B_0[7:0]
	G_2[7:0]	B_2[7:0]	R_1[7:0]	G_1[7:0]
	R_3[7:0]	G_3[7:0]	B_3[7:0]	R_2[7:0]

13.3.6 DMA2D 混合器

DMA2D 混合器成对混合源像素以计算结果像素。

混合将按以下公式执行：

$$\text{其中 } \alpha_{Mult} = \frac{\alpha_{FG} \cdot \alpha_{BG}}{255}$$

$$\alpha_{OUT} = \alpha_{FG} + \alpha_{BG} - \alpha_{Mult}$$

$$C_{OUT} = \frac{C_{FG} \cdot \alpha_{FG} + C_{BG} \cdot \alpha_{BG} - C_{BG} \cdot \alpha_{Mult}}{\alpha_{OUT}} \quad \text{其中 C = R 或 G 或 B}$$

商将向下取整

混合器不需要任何配置寄存器。是否使用混合器取决于 DMA2D_CR 寄存器的 MODE[2:0] 字段中定义的 DMA2D 工作模式。

13.3.7 DMA2D 输出 PFC

输出 PFC 将像素格式从 32 位转换为指定的输出格式，输出格式在 DMA2D 输出像素格式转换器配置寄存器 (DMA2D_OPFCCR) 的 CM[2:0] 字段中定义。

*表 59：输出时支持的颜色模式*给出了支持的输出格式。

表 59. 输出时支持的颜色模式

CM[2:0]	颜色模式
000	ARGB8888
001	RGB888
010	RGB565
011	ARGB1555
100	ARGB4444

注： *要支持备用格式，可通过将 DMA2D_OPFCCR 寄存器的 AI 位置 1 将计算的 alpha 值取反。这同样适用于 DMA2D_OCOLR 中使用的 Alpha 值。*

此外，还可通过将 DMA2D_OPFCCR 寄存器的 RBS 位置 1 来交换 R 字段和 B 字段。这同样适用于 DMA2D_OCOLR 中使用的 RGB 顺序。

13.3.8 DMA2D 输出 FIFO

输出 FIFO 根据输出 PFC 中定义的颜色格式对像素进行编程。

通过如下一组寄存器定义目标区域：

- DMA2D 输出存储器地址寄存器 (DMA2D_OMAR)
- DMA2D 输出偏移寄存器 (DMA2D_OOR)
- DMA2D 行数寄存器（行数和每行像素数）(DMA2D_NLR)

如果 DMA2D 在寄存器到存储器模式下工作，则配置的输出矩形将以 DMA2D 输出颜色寄存器 (DMA2D_OCOLR) 中指定的颜色填充，该寄存器中包含固定的 32 位、24 位或 16 位值。通过 DMA2D_OPFCCR 寄存器的 CM[2:0] 字段选择格式。

将按照*表 60：存储器中的数据顺序*中定义的顺序在存储器中存储数据

表 60. 存储器中的数据顺序

颜色模式	@ + 3	@ + 2	@ + 1	@ + 0
ARGB8888	A_0[7:0]	R_0[7:0]	G_0[7:0]	B_0[7:0]
RGB888	B_1[7:0]	R_0[7:0]	G_0[7:0]	B_0[7:0]
	G_2[7:0]	B_2[7:0]	R_1[7:0]	G_1[7:0]
	R_3[7:0]	G_3[7:0]	B_3[7:0]	R_2[7:0]
RGB565	R_1[4:0]G_1[5:3]	G_1[2:0]B_1[4:0]	R_0[4:0]G_0[5:3]	G_0[2:0]B_0[4:0]
ARGB1555	A_1[0]R_1[4:0]G_1[4:3]	G_1[2:0]B_1[4:0]	A_0[0]R_0[4:0]G_0[4:3]	G_0[2:0]B_0[4:0]
ARGB4444	A_1[3:0]R_1[3:0]	G_1[3:0]B_1[3:0]	A_0[3:0]R_0[3:0]	G_0[3:0]B_0[3:0]

通过 ARGB8888 模式支持按 32 位对齐 RGB888。

13.3.9 DMA2D 输出 FIFO 字节重新排序

输出 FIFO 字节可重新排序，以支持直接通过 DMA2D 的并行接口 (F(S)MC) 对帧缓冲区进行显示更新。

可通过以下方式对字节重新排序：

- 使用 RBS 位交换红色和蓝色分量
- 使用 SB 位两个两个地交换输出 FIFO 中的字节

激活字节交换功能（DMA2D_OPFCR 的 SB 字段置 1）后，每行的像素数（DMA2D_NLR 的 PL 字段）必须是偶数，输出存储器地址（DMA2D_OMAR 的 MA 字段）必须是偶数，并且以字节为单位计算的输出行偏移（由 DMA2D_CR 的 LOM 字段和 DMA2D_OOR 的 LO 字段值而得）必须是偶数。否则，将检测到配置错误。

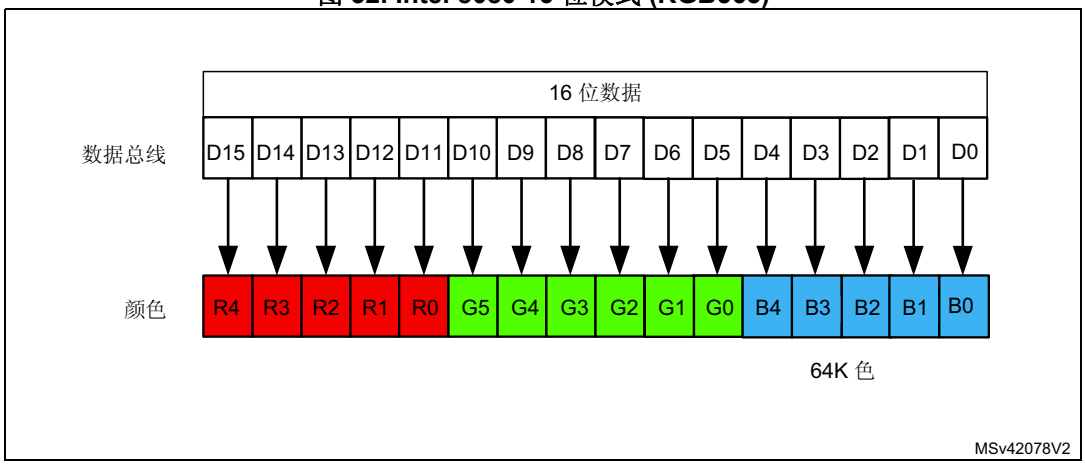
表 61. 存储器中的标准数据顺序

颜色模式	@ + 3	@ + 2	@ + 1	@ + 0
RGB888	B_1[7:0]	R_0[7:0]	G_0[7:0]	B_0[7:0]
	G_2[7:0]	B_2[7:0]	R_1[7:0]	G_1[7:0]
	R_3[7:0]	G_3[7:0]	B_3[7:0]	R_2[7:0]
RGB565	R_1[4:0]G_1[5:3]	G_1[2:0]B_1[4:0]	R_0[4:0]G_0[5:3]	G_0[2:0]B_0[4:0]

16 位模式 (RGB565)

无需通过 DMA2D 对字节重新排序即可支持此模式。

图 32. Intel 8080 16 位模式 (RGB565)



18/24 位模式 (RGB888)

此模式需要数据重新排序。

1. 必须交换红色和蓝色分量（将 RBS 位置 1）
2. 必须交换半字的 MSB 和 LSB 字节（将 SB 位置 1）

图 33. Intel 8080 18/24 位模式 (RGB888)

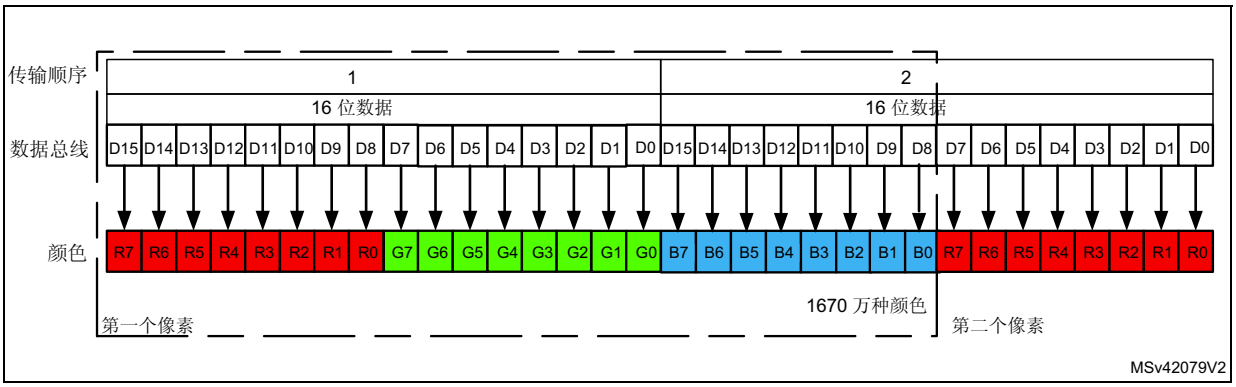


表 62. 输出 FIFO 字节重新排序步骤

步骤	@ + 3	@ + 2	@ + 1	@ + 0
原始数据排序	B_1[7:0]	R_0[7:0]	G_0[7:0]	B_0[7:0]
	G_2[7:0]	B_2[7:0]	R_1[7:0]	G_1[7:0]
	R_3[7:0]	G_3[7:0]	B_3[7:0]	R_2[7:0]
将 RBS 位置 1				
红色和蓝色分量交换（RBS 置 1）后的数据排序	R_1[7:0]	B_0[7:0]	G_0[7:0]	R_0[7:0]
	G_2[7:0]	R_2[7:0]	B_1[7:0]	G_1[7:0]
	B_3[7:0]	G_3[7:0]	R_3[7:0]	B_2[7:0]
将 SB 位置 1				
字节交换（SB 置 1）后的数据排序	B_0[7:0]	R_1[7:0]	R_0[7:0]	G_0[7:0]
	R_2[7:0]	G_2[7:0]	G_1[7:0]	B_1[7:0]
	G_3[7:0]	B_3[7:0]	B_2[7:0]	R_3[7:0]

13.3.10 DMA2D AHB 主设备端口定时器

AHB 主设备端口内嵌一个 8 位定时器，以便可选择限制交叉开关矩阵的带宽。

此定时器由 AHB 时钟驱动，对两个连续访问之间的死区进行计数。这样可限制带宽的使用。

通过 AHB 主设备端口定时器配置寄存器 (DMA2D_AMPTCR) 配置定时器使能和死区值。

13.3.11 DMA2D 事务

DMA2D 事务由给定数目的数据传输序列组成。可通过软件对数据数目和宽度进行编程。

每个 DMA2D 数据传输最多需要 4 个步骤：

1. 从 DMA2D_FGMAR 寄存器寻址的存储单元加载数据并按照 DMA2D_FGCR 中的定义进行像素格式转换。
2. 从 DMA2D_BGMAR 寄存器寻址的存储单元加载数据并按照 DMA2D_BGCR 中的定义进行像素格式转换。
3. 根据对 alpha 值进行 PFC 操作所得到的 alpha 通道，将所有检索到的像素混合。
4. 根据 DMA2D_OCR 寄存器对合成像素进行像素格式转换，然后将数据编程到通过 DMA2D_OMAR 寄存器寻址的存储单元。

13.3.12 DMA2D 配置

源和目标数据传输在整个 4 GB 区域（地址范围在 0x0000 0000 和 0xFFFF FFFF 之间）都可以寻址外设和存储器。

DMA2D 可在以下四种模式下工作，通过 DMA2D_CR 寄存器的 MODE[2:0] 位选择工作模式：

- 寄存器到存储器
- 存储器到存储器
- 存储器到存储器并执行 PFC
- 存储器到存储器并执行 PFC 和混合
- 存储器到存储器并执行 PFC、混合和固定颜色 FG
- 存储器到存储器并支持 PFC、混合和固定颜色 BG

寄存器到存储器

寄存器到存储器模式用于以预定义颜色填充用户自定义区域。

颜色格式在 DMA2D_OPFCCR 中设置。

DMA2D 不从任何源获取数据。它只将 DMA2D_OCOLR 寄存器中定义的颜色写入通过 DMA2D_OMAR 寻址以及 DMA2D_NLR 和 DMA2D_OOR 定义的区域。

存储器到存储器

在存储器到存储器模式下，DMA2D 不执行任何图形数据转换。前景层输入 FIFO 充当缓冲区，数据从 DMA2D_FGMAR 中定义的源存储单元传输到 DMA2D_OMAR 寻址的目标存储单元。

DMA2D_FGPFCCR 寄存器的 CM[3:0] 位中编程的颜色模式决定输入和输出的每像素位数。

对于要传输的区域大小，源区域大小由 DMA2D_NLR 和 DMA2D_FGOR 寄存器定义，目标区域大小则由 DMA2D_NLR 和 DMA2D_OOR 寄存器定义。

存储器到存储器并执行 PFC

此模式下，DMA2D 对源数据执行像素格式转换并将结果存储在目标存储单元。

对于要传输的区域大小，源区域大小由 DMA2D_NLR 和 DMA2D_FGOR 寄存器定义，目标区域大小则由 DMA2D_NLR 和 DMA2D_OOR 寄存器定义。

从 DMA2D_FGMAR 寄存器定义的位置获取数据，并由前景层 PFC 进行处理。原始像素格式通过 DMA2D_FGPFCCR 寄存器配置。

如果原始像素格式是直接颜色模式，则所有颜色通道都扩展到 8 位。

如果像素格式是间接颜色模式，则必须将相关 CLUT 加载到 CLUT 存储器中。

CLUT 加载可按如下顺序自动完成：

1. 在 DMA2D_FGCMAR 中设置 CLUT 地址。
2. 在 DMA2D_FGPFCCR 寄存器的 CS[7:0] 位设置 CLUT 大小。
3. 在 DMA2D_FGPFCCR 寄存器的 CCM 位设置 CLUT 格式（24 或 32 位）。
4. 将 DMA2D_FGPFCCR 寄存器的 START 位置 1 启动 CLUT 加载。

CLUT 加载完成时，DMA2D_ISR 寄存器将置位 CTCIF 标志；如果 DMA2D_CR 中的 CTCIE 位置 1，还将产生中断。CLUT 自动加载过程无法与传统的 DMA2D 传输同时进行。

CLUT 还可由 CPU 填充，或由任意其他主设备通过 APB 端口填充。在 DMA2D 传输进行期间和使用 CLUT（间接颜色格式）时无法访问 CLUT。

在颜色转换执行期间，可根据 DMA2D_FGPFCCR 寄存器中编程的值添加或更改 alpha 值。如果原始图像没有 alpha 通道，则会自动添加一个默认的 alpha 值 0xFF 以获得完全不透明的像素。可根据 DMA2D_FGPFCCR 寄存器的 AM[1:0] 位修改 alpha 值：

- 保持不变。
- 替换为 DMA2D_FGPFCCR 寄存器的 ALPHA[7:0] 值中定义的值。
- 替换为原始值与 DMA2D_FGPFCCR 寄存器的 ALPHA[7:0] 值除以 255 所得商的乘积。

结果得到的 32 位数据由 OUT PFC 编码成 DMA2D_OPFCCR 寄存器的 CM[2:0] 字段所指定的格式。输出像素格式不可是间接模式，原因是输出时不支持 CLUT 生成过程。

数据经处理后，将写入 DMA2D_OMAR 寻址的目标存储单元。

存储器到存储器并执行 PFC 和混合

此模式下，将在前景层 FIFO 和背景层 FIFO（分别在 DMA2D_FGMAR 和 DMA2D_BGMAR 中定义）获取 2 个源图像。

必须按存储器到存储器模式中所述配置两个像素格式转换器。由于这两个像素格式转换器各自独立且自身具有 CLUT 存储器，因此其配置可以不同。

在每个像素都通过相应的 PFC 转换为 32 位后，将根据以下公式进行混合：

$$\text{其中 } \alpha_{Mult} = \frac{\alpha_{FG} \cdot \alpha_{BG}}{255}$$

$$\alpha_{OUT} = \alpha_{FG} + \alpha_{BG} - \alpha_{Mult}$$

$$C_{OUT} = \frac{C_{FG} \cdot \alpha_{FG} + C_{BG} \cdot \alpha_{BG} - C_{BG} \cdot \alpha_{Mult}}{\alpha_{OUT}} \quad \text{其中 C = R 或 G 或 B}$$

商将向下取整

输出 PFC 将根据指定的输出格式对得到的 32 位像素值进行编码，并且编码数据将写入 DMA2D_OMAR 寻址的目标存储单元。

存储器到存储器并支持 PFC、混合和固定颜色 FG

在此模式下，仅在背景层 FIFO（DMA2D_BGMAR 定义的存储单元中）获取 1 个源图像。

前景层颜色的值由 DMA2D_FGCOLR 寄存器给出，alpha 值设置为 0xFF（不透明）。

Alpha 值可根据 DMA2D_FGPFCR 的 AM[1:0] 和 ALPHA[7:0] 字段替换或修改。

必须按存储器到存储器模式中所述配置两个像素格式转换器。由于这两个像素格式转换器各自独立且自身具有 CLUT 存储器，因此其配置可以不同。

在每个像素由相应的 PFC 转换为 32 位后，它们将混合在一起，得到的 32 位像素值由输出 PFC 根据指定的输出格式进行编码，并且数据将被写入到 DMA2D_OMAR 指向的目标存储单元。

存储器到存储器并支持 PFC、混合和固定颜色 BG

在此模式下，仅在前景层 FIFO（DMA2D_FGMAR 定义的存储单元中）获取 1 个源图像。

背景层颜色的值由 DMA2D_BGCOLR 寄存器给出，alpha 值设置为 0xFF（不透明）。

Alpha 值可根据 DMA2D_BGPFCR 的 AM[1:0] 和 ALPHA[7:0] 字段替换或修改。

必须按存储器到存储器模式中所述配置两个像素格式转换器。由于这两个像素格式转换器各自独立且自身具有 CLUT 存储器，因此其配置可以不同。

在每个像素由相应的 PFC 转换为 32 位后，它们将混合在一起，得到的 32 位像素值由输出 PFC 根据指定的输出格式进行编码，并且数据将被写入到 DMA2D_OMAR 指向的目标存储单元。

配置错误检测

DMA2D 将在每次执行传输前检查配置是否正确。开始新的传输/自动加载时，如果检测到配置错误，硬件将设置配置错误中断标志。如果 DMA2D_CR 寄存器的 CEIE 位置 1，还将产生中断。

可检测到的错误配置如下：

- 前景层 CLUT 自动加载：DMA2D_FGCMAR 的 MA 位与 DMA2D_FGPFCCR 的 CCM 位不匹配
- 背景层 CLUT 自动加载：DMA2D_BGCMAR 的 MA 位与 DMA2D_BGPFCCR 的 CCM 位不匹配
- 存储器传输（寄存器到存储器模式以及存储器到存储器模式并支持混合和固定颜色 FG 除外）：DMA2D_FGMAR 的 MA 位与 DMA2D_FGPFCCR 的 CM 不匹配
- 存储器传输（寄存器到存储器模式以及存储器到存储器模式并支持混合和固定颜色 FG 除外）：DMA2D_FGPFCCR 的 CM 位无效
- 存储器传输（寄存器到存储器模式以及存储器到存储器模式并支持混合和固定颜色 FG 除外）：DMA2D_NLR 的 PL 位为奇数，而 DMA2D_FGPFCCR 的 CM 为 A4 或 L4
- 存储器传输（寄存器到存储器模式以及存储器到存储器模式并支持混合和固定颜色 FG 除外）：DMA2D_FGOR 的 LO 位为奇数，而 DMA2D_FGPFCCR 的 CM 为 A4 或 L4，且 DMA2D_CR 的 LOM 位为像素模式
- 存储器传输（仅限混合模式以及存储器到存储器模式并支持混合和固定颜色 FG）：DMA2D_BGMAR 的 MA 位与 DMA2D_BGPFCCR 的 CM 不匹配
- 存储器传输（仅限混合模式以及混合并支持固定颜色 FG 模式）：DMA2D_BGPFCCR 的 CM 位无效
- 存储器传输（仅限混合模式以及混合并支持固定颜色 FG 模式）：DMA2D_NLR 的 PL 位为奇数，而 DMA2D_BGPFCCR 的 CM 位为 A4 或 L4
- 存储器传输（仅限混合模式以及混合并支持固定颜色 FG 模式）：DMA2D_BGOR 的 LO 位为奇数，而 DMA2D_BGPFCCR 的 CM 为 A4 或 L4，且 DMA2D_CR 的 LOM 位为像素模式
- 存储器传输（存储器到存储器模式除外）：DMA2D_OMAR 的 MA 位与 DMA2D_OPFCCR 的 CM 位不匹配
- 存储器传输（存储器到存储器模式除外）：DMA2D_OPFCCR 的 CM 位无效
- 进行字节交换的存储器传输：DMA2D_NLR 的 PL 位为奇数，或 DMA2D_OMAR 的 MA 位为奇数，或者当 DMA2D_OPFCCR 的 SB 位置 1 的时候，且字节中的 LO（由 DMA2D_CR 的 LOM 位和 DMA2D_OOR 的 LO 位值而得）为奇数时
- 存储器传输：DMA2D_NLR 的 NL 位 = 0
- 存储器传输：DMA2D_NLR 的 PL 位 = 0
- 存储器传输：DMA2D_CR 的 MODE 位无效

13.3.13 DMA2D 传输控制（启动、挂起、中止和完成）

配置 DMA2D 后，通过将 DMA2D_CR 寄存器的 START 位置 1 可启动传输。传输完成时，START 位自动复位且 DMA2D_ISR 寄存器置位 TCIF 标志。如果 DMA2D_CR 寄存器中的 TCIE 位置 1，还将产生中断。

用户应用程序随时都可通过将 DMA2D_CR 寄存器的 SUSP 位置 1 来挂起 DMA2D。随即可通过将 DMA2D_CR 寄存器的 ABORT 位置 1 中止事务，或通过将 DMA2D_CR 寄存器的 SUSP 位复位重新启动事务。

用户应用程序随时都可通过将 DMA2D_CR 寄存器的 ABORT 位置 1 来中止处理中的事务。在这种情况下，不会置位 TCIF 标志。

还可通过 DMA2D_CR 寄存器的 ABORT 或 SUSP 位中止或挂起 CLUT 自动传输过程。

13.3.14 水印

可对水印编程，以在指定行的最后一个像素写入目标存储区域时产生中断。

行号在 DMA2D_LWR 寄存器的 LW[15:0] 字段中定义。

在该行的最后一个像素传输完成时，DMA2D_ISR 将置位 TWIF 标志；在 DMA2D_CR 的 TWIE 位置 1 的情况下，还将产生中断。

13.3.15 错误管理

可触发两种错误：

- AHB 主设备端口错误，通过 DMA2D_ISR 寄存器的 TEIF 标志指示。
- CLUT 访问引发的冲突（CPU 在 CLUT 加载或 DMA2D 传输执行期间尝试访问 CLUT），通过 DMA2D_ISR 寄存器的 CAEIF 标志指示。

这两个标志都与其在 DMA2D_CR 寄存器中的中断使能标志（控制在需要时产生中断的 TEIE 和 CAEIE）相关。

13.3.16 AHB 死区

要限制 AHB 带宽的使用，可在两个连续的 AHB 访问之间编程一个死区。

可通过将 DMA2D_AMTCR 寄存器中的 EN 位置 1 使能此特性。

死区值存储在 DMA2D_AMTCR 寄存器的 DT[7:0] 字段中。该值表示 AHB 总线上两个连续事务之间允许占用的最少周期数。

在 DMA2D 运行过程中对死区值所做的更新将在下一次 AHB 传输时生效。

13.4 DMA2D 中断

发生如下事件时可生成中断：

- 配置错误
- CLUT 传输完成
- CLUT 访问错误
- 到达传输水印
- 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 63. DMA2D 中断请求

中断事件	事件标志	使能控制位
配置错误	CEIF	CEIE
CLUT 传输完成	CTCIF	CTCIE
CLUT 访问错误	CAEIF	CAEIE
传输水印	TWF	TWIE
传输完成	TCIF	TCIE
传输错误	TEIF	TEIE

13.5 DMA2D 寄存器

13.5.1 DMA2D 控制寄存器 (DMA2D_CR)

DMA2D 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
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MODE[2:0]		
													rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	CEIE	CTCIE	CAEIE	TWIE	TCIE	TEIE	Res.	LOM	Res.	Res.	Res.	ABORT	SUSP	START
		rw	rw	rw	rw	rw	rw		rw				rs	rw	rs

位 31:19 保留，必须保持复位值

位 18:16 **MODE[2:0]**：DMA2D 模式 (DMA2D mode)
这些位将由软件置 1 和清零。无法在进行传输时修改这些位。
000：存储器到存储器（仅限 FG 获取）
001：存储器到存储器并执行 PFC（仅限 FG PFC 激活时的 FG 获取）
010：存储器到存储器并执行混合（执行 PFC 和混合时的 FG 和 BG 获取）
011：寄存器到存储器（无 FG 和 BG，仅输出阶段激活）
100：存储器到存储器并支持混合与固定颜色 FG（仅限 FG 和 BG PFC 激活时的 BG 获取）
101：存储器到存储器并支持混合与固定颜色 BG（仅限 FG 和 BG PFC 激活时的 BG 获取）
其他：无意义

位 15:14 保留，必须保持复位值

位 13 **CEIE**：配置错误中断使能 (Configuration Error Interrupt Enable)
此位由软件置 1 和清零。
0：CE 中断禁止
1：CE 中断使能

位 12 **CTCIE**：CLUT 传输完成中断使能 (CLUT transfer complete interrupt enable)
此位由软件置 1 和清零。
0：CTC 中断禁止
1：CTC 中断使能

位 11 **CAEIE**：CLUT 访问错误中断使能 (CLUT access error interrupt enable)
此位由软件置 1 和清零。
0：CAE 中断禁止
1：CAE 中断使能

位 10 **TWIE**：传输水印中断使能 (Transfer watermark interrupt enable)
此位由软件置 1 和清零。
0：TW 中断禁止
1：TW 中断使能

位 9 **TCIE**：传输完成中断使能 (Transfer complete interrupt enable)
此位由软件置 1 和清零。
0：TC 中断禁止
1：TC 中断使能

位 8 **TEIE**：传输错误中断使能 (Transfer error interrupt enable)
此位由软件置 1 和清零。
0：TE 中断禁止
1：TE 中断使能

位 7 保留，必须保持复位值

位 6 **LOM**：行偏移模式 (Line Offset Mode)
该位用于配置前景层、背景层和输出的行偏移的表示形式（像素或字节）。
此位由软件置 1 和清零。无法在进行传输时修改此位。
0：行偏移以像素表示
1：行偏移以字节表示

位 5:3 保留，必须保持复位值

位 2 **ABORT**：中止 (Abort)
此位可用于中止当前传输。此位可通过软件置 1 并在 START 位复位时由硬件自动复位。
0：不请求传输中止
1：请求传输中止

位 1 **SUSP**：挂起 (Suspend)
此位可用于挂起当前传输。此位由软件置 1 和复位。此位在 START 位复位时由硬件自动复位。
0：传输不挂起
1：传输挂起

位 0 **START**：启动 (Start)
此位可用于根据各种配置寄存器中加载的参数启动 DMA2D。在下列情况下将自动复位此位：
– 传输结束时
– 通过用户应用程序将 DMA2D_CR 中的 ABORT 位置 1 中止数据传输时
– 数据传输出错时
– 因配置错误或已经在进行其他传输操作（CLUT 自动加载）导致数据传输未启动时

13.5.2 DMA2D 中断状态寄存器 (DMA2D_ISR)

DMA2D Interrupt Status 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.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CEIF	CTCIF	CAEIF	TWIF	TCIF	TEIF
										r	r	r	r	r	r

位 31:6 保留，必须保持复位值

位 5 **CEIF**：配置错误中断标志 (Configuration error interrupt flag)
DMA2D_CR、DMA2DFGPFCCR 或 DMA2D_BGPFCCR 的 START 位置 1 以及编程了错误的配置时，此位置 1。

位 4 **CTCIF**：CLUT 传输完成中断标志 (CLUT transfer complete interrupt flag)
完成将 CLUT 从系统存储区复制到 DMA2D 内部存储器时，此位置 1。

位 3 **CAEIF**：CLUT 访问错误中断标志 (CLUT access error interrupt flag)
在从系统存储器自动将 CLUT 复制到 DMA2D 内部存储器期间，CPU 若访问 CLUT，此位将置 1。

位 2 **TWIF**：传输水印中断标志 (Transfer watermark interrupt flag)
带水印行的最后一个像素完成传输时，此位置 1。

位 1 **TCIF**：传输完成中断标志 (Transfer complete interrupt flag)
DMA2D 传输操作完成（仅限数据传输）时此位置 1。

位 0 **TEIF**：传输错误中断标志 (Transfer error interrupt flag)
DMA 传输期间（数据传输或 CLUT 自动加载）出错时此位置 1。

13.5.3 DMA2D 中断标志清零寄存器 (DMA2D_IFCR)

DMA2D interrupt flag clear 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.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CCEIF	CCTCIF	CAECIF	CTWIF	CTCIF	CTEIF
										rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1

位 31:6 保留，必须保持复位值

位 5 **CCEIF**：清除配置错误中断标志 (Clear configuration error interrupt flag)
将此位编程为 1 可清除 DMA2D_ISR 寄存器中的 CEIF 标志

位 4 **CCTCIF**：清除 CLUT 传输完成中断标志 (Clear CLUT transfer complete interrupt flag)
将此位编程为 1 可清除 DMA2D_ISR 寄存器中的 CTCIF 标志

位 3 **CAECIF**：清除 CLUT 访问错误中断标志 (Clear CLUT access error interrupt flag)
将此位编程为 1 可清除 DMA2D_ISR 寄存器中的 CAEIF 标志

位 2 **CTWIF**：清除传输水印中断标志 (Clear transfer watermark interrupt flag)
将此位编程为 1 可清除 DMA2D_ISR 寄存器中的 TWIF 标志

位 1 **CTCIF**：清除传输完成中断标志 (Clear transfer complete interrupt flag)
将此位编程为 1 可清除 DMA2D_ISR 寄存器中的 TCIF 标志

位 0 **CTEIF**：清除传输错误中断标志 (Clear Transfer error interrupt flag)
将此位编程为 1 可清除 DMA2D_ISR 寄存器中的 TEIF 标志

13.5.4 DMA2D 前景层存储器地址寄存器 (DMA2D_FGMAR)

DMA2D foreground memory address register

偏移地址：0x000C

复位值：0x0000 0000

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

位 31:0 **MA[31:0]**：存储器地址 (Memory address)

前景层图像所用数据的地址。只有在禁止数据传输的情况下才能写入该寄存器。数据传输一旦启动，此寄存器即变为只读。

地址对齐必须与所选图像格式相匹配，例如每像素 32 位格式必须为 32 位对齐，每像素 16 位格式必须为 16 位对齐，而每像素 4 位格式必须为 8 位对齐。

13.5.5 DMA2D 前景层偏移寄存器 (DMA2D_FGOR)

DMA2D foreground offset 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
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
LO[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 **LO[15:0]**：行偏移 (Line offset)

对于前景层图像所使用的行偏移，在 LOM 位复位时以像素表示，在 LOM 位置 1 时以字节表示。

以像素表示时，仅考虑 LO[13:0]，忽略 LO[15:14]。

此值用于生成地址。行偏移将添加到各行末尾，用于确定下一行的起始地址。

只有在禁止数据传输的情况下才能写入这些位。数据传输一旦启动，这些位将变为只读。

如果图像格式为每像素 4 位，则行偏移值必须为偶数。

13.5.6 DMA2D 背景层存储器地址寄存器 (DMA2D_BGMAR)

DMA2D background memory address 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
MA[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
MA[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:0 **MA[31:0]**：存储器地址 (Memory address)

背景层图像所用数据的地址。只有在禁止数据传输的情况下才能写入该寄存器。数据传输一旦启动，此寄存器即变为只读。

地址对齐必须与所选图像格式相匹配，例如每像素 32 位格式必须为 32 位对齐，每像素 16 位格式必须为 16 位对齐，而每像素 4 位格式必须为 8 位对齐。

13.5.7 DMA2D 背景层偏移寄存器 (DMA2D_BGOR)

DMA2D background offset 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
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
LO[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 **LO[15:0]**：行偏移 (Line offset)

对于背景层图像所使用的行偏移，在 LOM 位复位时以像素表示，在 LOM 位置 1 时以字节表示。

以像素表示时，仅考虑 LO[13:0]，忽略 LO[15:14]。

此值用于生成地址。行偏移将添加到各行末尾，用于确定下一行的起始地址。

只有在禁止数据传输的情况下才能写入这些位。数据传输一旦启动，这些位将变为只读。

如果图像格式为每像素 4 位，则行偏移值必须为偶数。

13.5.8 DMA2D 前景层 PFC 控制寄存器 (DMA2D_FGPFCCR)

DMA2D foreground PFC control register

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
ALPHA[7:0]								Res.	Res.	RBS	AI	Res.	Res.	AM[1:0]	
rw	rw	rw	rw	rw	rw	rw	rw			rw	rw			rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CS[7:0]								Res.	Res.	START	CCM	CM[3:0]			
rw	rw	rw	rw	rw	rw	rw	rw			rs	rw	rw	rw	rw	rw

位 31:24 **ALPHA[7:0]**：Alpha 值 (Alpha value)

这些位定义固定的 alpha 通道值，该值可替代原始的 alpha 值或与原始的 alpha 值相乘，具体取决于通过 AM[1:0] 位选择的 alpha 模式。

只有在禁止数据传输的情况下才能写入这些位。传输一旦启动，这些位将变为只读。

位 23:22 保留，必须保持复位值

位 21 **RBS**：红蓝交换 (Red Blue Swap)

此位可交换 R 和 B，从而支持 BGR 或 ABGR 颜色格式。传输一旦启动，此位将变为只读。

0：常规模式（RGB 或 ARGB）

1：交换模式（BGR 或 ABGR）

位 20 **AI**：将 Alpha 取反 (Alpha Inverted)

此位用于将 alpha 值取反。传输一旦启动，此位将变为只读。

0：常规 Alpha

1：取反 Alpha

位 19:18 保留，必须保持复位值

位 17:16 **AM[1:0]**：Alpha 模式 (Alpha mode)

这些位用于选择将用于前景层图像的 alpha 通道值。只有在禁止数据传输的情况下才能写入这些位。传输一旦启动，这些位将变为只读。

00：不修改前景层图像的 alpha 通道值

01：原始前景层图像的 alpha 通道值替换为 ALPHA[7: 0]

10：原始前景层图像的 alpha 通道值替换为 ALPHA[7: 0] 与原始 alpha 通道值的乘积

其他配置无意义

位 15:8 **CS[7:0]**：CLUT 大小 (CLUT size)

这些位定义前景层图像所用的 CLUT 的大小。CLUT 传输一旦启动，此字段将变为只读。CLUT 条目数等于 CS[7:0] + 1。

位 7:6 保留，必须保持复位值

位 5 **START**：启动 (Start)

可将此位置 1 以启动 CLUT 的自动加载过程。该位在以下情况下自动复位：

- 传输结束时
- 通过用户应用程序将 DMA2D_CR 中的 ABORT 位置 1 中止传输时
- 传输出错时
- 因配置错误或已经在进行其他传输操作（数据传输或自动背景层 CLUT 传输）导致传输未启动时。

位 4 **CCM**：CLUT 颜色模式 (CLUT color mode)

此位定义 CLUT 的颜色格式。只有在禁止数据传输的情况下才能写入该位。CLUT 传输一旦启动，此位将变为只读。

0：ARGB8888

1：RGB888

其他：无意义

位 3:0 **CM[3:0]**：颜色模式 (Color mode)

这些位定义前景层图像的颜色格式。只有在禁止数据传输的情况下才能写入这些位。传输一旦启动，这些位将变为只读。

0000：ARGB8888

0001：RGB888

0010：RGB565

0011：ARGB1555

0100：ARGB4444

0101：L8

0110：AL44

0111：AL88

1000：L4

1001：A8

1010：A4

其他：无意义

13.5.9 DMA2D 前景层颜色寄存器 (DMA2D_FGCOLR)

DMA2D foreground color 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
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RED[7:0]							
								rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GREEN[7:0]								BLUE[7:0]							
rw	rw	rw	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 23:16 **RED[7:0]**：红色值 (Red Value)

这些位定义前景层图像的 A4 或 A8 模式的红色值。只有在禁止数据传输的情况下才能写入这些位。传输一旦启动，这些位将变为只读。

还用于固定颜色 FG（在存储器到存储器并支持混合与固定颜色 FG（仅限 FG 和 BG PFC 激活时的 BG 获取）模式下）。

位 15:8 **GREEN[7:0]**：绿色值 (Green Value)

这些位定义前景层图像的 A4 或 A8 模式的绿色值。只有在禁止数据传输的情况下才能写入这些位。传输一旦启动，这些位将变为只读。

还用于固定颜色 FG（在存储器到存储器并支持混合与固定颜色 FG（仅限 FG 和 BG PFC 激活时的 BG 获取）模式下）。

位 7:0 **BLUE[7:0]**：蓝色值 (Blue Value)

这些位定义前景层图像的 A4 或 A8 模式的蓝色值。只有在禁止数据传输的情况下才能写入这些位。传输一旦启动，这些位将变为只读。

还用于固定颜色 FG（在存储器到存储器并支持混合与固定颜色 FG（仅限 FG 和 BG PFC 激活时的 BG 获取）模式下）。

13.5.10 DMA2D 背景层 PFC 控制寄存器 (DMA2D_BGPFCCR)

DMA2D background PFC control 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
ALPHA[7:0]								Res.	Res.	RBS	AI	Res.	Res.	AM[1:0]	
rw	rw	rw	rw	rw	rw	rw	rw			rw	rw			rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CS[7:0]								Res.	Res.	START	CCM	CM[3:0]			
rw	rw	rw	rw	rw	rw	rw	rw			rs	rw	rw	rw	rw	rw

位 31:24 **ALPHA[7:0]**：Alpha 值 (Alpha value)

这些位定义固定的 alpha 通道值，该值可替代原始的 alpha 值或与原始的 alpha 值相乘，具体取决于通过 AM[1:0] 位选择的 alpha 模式。只有在禁止数据传输的情况下才能写入这些位。传输一旦启动，这些位将变为只读。

位 23:22 保留，必须保持复位值

位 21 **RBS**：红蓝交换 (Red Blue Swap)

此位可交换 R 和 B，从而支持 BGR 或 ABGR 颜色格式。传输一旦启动，此位将变为只读。

0：常规模式（RGB 或 ARGB）

1：交换模式（BGR 或 ABGR）

位 20 **AI**：将 Alpha 取反

此位用于将 alpha 值取反。传输一旦启动，此位将变为只读。

0：常规 Alpha

1：取反 Alpha

位 19:18 保留，必须保持复位值

位 17:16 **AM[1:0]**：Alpha 模式 (Alpha mode)

这些位定义将用于背景层图像的 alpha 通道值。只有在禁止数据传输的情况下才能写入这些位。传输一旦启动，这些位将变为只读。

00：不修改前景层图像的 alpha 通道值

01：原始背景层图像的 alpha 通道值替换为 ALPHA[7: 0]

10：原始背景层图像的 alpha 通道值替换为 ALPHA[7: 0] 与原始 alpha 通道值的乘积

其他：无意义

位 15:8 **CS[7:0]**：CLUT 大小 (CLUT size)

这些位定义背景层图像所用的 CLUT 的大小。CLUT 传输一旦启动，此字段将变为只读。CLUT 条目数等于 CS[7:0] + 1。

位 7:6 保留，必须保持复位值

位 5 **START**：启动 (Start)

可将此位置 1 以启动 CLUT 的自动加载过程。该位在以下情况下自动复位：

- 传输结束时
- 通过用户应用程序将 DMA2D_CR 中的 ABORT 位置 1 中止传输时
- 传输出错时
- 因配置错误或已经在进行其他传输操作（数据传输或自动前景层 CLUT 传输）导致传输未启动时。

位 4 **CCM**：CLUT 颜色模式 (CLUT Color mode)

这些位定义 CLUT 的颜色格式。只有在禁止传输的情况下才能写入该寄存器。CLUT 传输一旦启动，此位将变为只读。

0：ARGB8888

1：RGB888

其他：无意义

位 3:0 **CM[3:0]**：颜色模式 (Color mode)

这些位定义前景层图像的颜色格式。只有在禁止数据传输的情况下才能写入这些位。传输一旦启动，这些位将变为只读。

0000：ARGB8888

0001：RGB888

0010：RGB565

0011：ARGB1555

0100：ARGB4444

0101：L8

0110：AL44

0111：AL88

1000：L4

1001：A8

1010：A4

其他：无意义

13.5.11 DMA2D 背景层颜色寄存器 (DMA2D_BGCOLR)

DMA2D background color 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
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RED[7:0]							
								rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GREEN[7:0]								BLUE[7:0]							
rw	rw	rw	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 23:16 **RED[7:0]**：红色值 (Red Value)

这些位定义背景层图像的 A4 或 A8 模式的红色值。只有在禁止数据传输的情况下才能写入这些位。传输一旦启动，这些位将变为只读。

还用于固定颜色 FG（在存储器到存储器并支持混合与固定颜色 FG（仅限 FG 和 BG PFC 激活时的 BG 获取）模式下）。

位 15:8 **GREEN[7:0]**：绿色值 (Green Value)

这些位定义背景层图像的 A4 或 A8 模式的绿色值。只有在禁止数据传输的情况下才能写入这些位。传输一旦启动，这些位将变为只读。

还用于固定颜色 FG（在存储器到存储器并支持混合与固定颜色 FG（仅限 FG 和 BG PFC 激活时的 BG 获取）模式下）。

位 7:0 **BLUE[7:0]**：蓝色值 (Blue Value)

这些位定义背景层图像的 A4 或 A8 模式的蓝色值。只有在禁止数据传输的情况下才能写入这些位。传输一旦启动，这些位将变为只读。

还用于固定颜色 FG（在存储器到存储器并支持混合与固定颜色 FG（仅限 FG 和 BG PFC 激活时的 BG 获取）模式下）。

13.5.12 DMA2D 前景层 CLUT 存储器地址寄存器 (DMA2D_FGCMAR)

DMA2D foreground CLUT memory address register

偏移地址：0x002C

复位值：0x0000 0000

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

位 31:0 **MA[31:0]**：存储器地址 (Memory Address)

专用于前景层图像的 CLUT 地址所使用的数据地址。只有不存在进行中传输的情况下才能写入该寄存器。CLUT 传输一旦启动，此寄存器将变为只读。

如果前景层 CLUT 格式是 32 位，则地址必须是 32 位对齐。

13.5.13 DMA2D 背景层 CLUT 存储器地址寄存器 (DMA2D_BGCMAR)

DMA2D background CLUT memory address register

偏移地址：0x0030

复位值：0x0000 0000

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

位 31:0 **MA[31:0]**：存储器地址 (Memory address)

专用于背景层图像的 CLUT 地址所使用的数据地址。只有不存在进行中传输的情况下才能写入该寄存器。CLUT 传输一旦启动，此寄存器将变为只读。

如果背景层 CLUT 格式是 32 位，则地址必须是 32 位对齐。

13.5.14 DMA2D 输出 PFC 控制寄存器 (DMA2D_OPFCCR)

DMA2D output PFC control register

偏移地址：0x0034

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

位 31:22 保留，必须保持复位值

位 21 **RBS**：红蓝交换 (Red Blue Swap)
此位可交换 R 和 B，从而支持 BGR 或 ABGR 颜色格式。传输一旦启动，此位将变为只读。
0：常规模式（RGB 或 ARGB）
1：交换模式（BGR 或 ABGR）

位 20 **AI**：将 Alpha 取反 (Alpha Inverted)
此位用于将 alpha 值取反。传输一旦启动，此位将变为只读。
0：常规 Alpha
1：取反 Alpha

位 19:9 保留，必须保持复位值

位 8 **SB**：交换字节 (Swap Bytes)
该位置 1 后，输出 FIFO 中的字节两个两个地进行交换。
如果该位置 1，则每行像素数 (PL) 必须是偶数，且输出存储器地址 (OMAR) 必须是偶数。
只有在传输禁止时才能写入此寄存器。传输一旦启动，此寄存器即变为只读。
0：输出 FIFO 中的字节采用常规顺序
1：输出 FIFO 中的字节两个两个地进行交换

位 7:3 保留，必须保持复位值

位 2:0 **CM[2:0]**：颜色模式 (Color mode)
这些位定义背景层图像的颜色格式。只有在禁止数据传输的情况下才能写入这些位。传输一旦启动，这些位将变为只读。
000：ARGB8888
001：RGB888
010：RGB565
011：ARGB1555
100：ARGB4444
其他：无意义

13.5.15 DMA2D 输出颜色寄存器 (DMA2D_OCOLR)

DMA2D output color register

偏移地址：0x0038

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ALPHA[7:0]								RED[7:0]							
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
GREEN[7:0]								BLUE[7:0]							
RED[4:0]					GREEN[5:0]					BLUE[4:0]					
A	RED[4:0]				GREEN[4:0]					BLUE[4:0]					
ALPHA[3:0]				RED[3:0]				GREEN[3:0]				BLUE[3:0]			
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:24 **ALPHA[7:0]**：Alpha 通道值 (Alpha Channel Value)

这些位定义输出颜色的 alpha 通道。只有在禁止数据传输的情况下才能写入这些位。传输一旦启动，这些位将变为只读。

位 23:16 **RED[7:0]**：红色值 (Red Value)

这些位定义输出图像的红色值。只有在禁止数据传输的情况下才能写入这些位。传输一旦启动，这些位将变为只读。

位 15:8 **GREEN[7:0]**：绿色值 (Green Value)

这些位定义输出图像的绿色值。只有在禁止数据传输的情况下才能写入这些位。传输一旦启动，这些位将变为只读。

位 7:0 **BLUE[7:0]**：蓝色值 (Blue Value)

这些位定义输出图像的蓝色值。只有在禁止数据传输的情况下才能写入这些位。传输一旦启动，这些位将变为只读。

13.5.16 DMA2D 输出存储器地址寄存器 (DMA2D_OMAR)

DMA2D output memory address register

偏移地址：0x003C

复位值：0x0000 0000

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

位 31:0 **MA[31:0]**：存储器地址 (Memory Address)

输出 FIFO 所用数据的地址。只有在禁止数据传输的情况下才能写入这些位。传输一旦启动，这些位将变为只读。

地址对齐必须与所选图像格式相匹配，例如每像素 32 位格式必须为 32 位对齐，每像素 16 位格式必须为 16 位对齐。

13.5.17 DMA2D 输出偏移寄存器 (DMA2D_OOR)

DMA2D output offset register

偏移地址：0x0040

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
LO[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 **LO[15:0]**：行偏移 (Line offset)

对于输出所使用的行偏移，在 LOM 位复位时以像素表示，在 LOM 位置 1 时以字节表示。
以像素表示时，仅考虑 LO[13:0]，忽略 LO[15:14]。
此值用于生成地址。行偏移将添加到各行末尾，用于确定下一行的起始地址。
只有在禁止数据传输的情况下才能写入这些位。数据传输一旦启动，这些位将变为只读。

13.5.18 DMA2D 行数寄存器 (DMA2D_NLR)

DMA2D number of line register

偏移地址：0x0044

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	PL[13:0]													
		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
NL[15:0]															
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 29:16 **PL[13:0]**：每行像素数 (Pixel per lines)

待传输区域的每行像素数。只有在禁止数据传输的情况下才能写入这些位。传输一旦启动，这些位将变为只读。
如果任何一个输入图像格式为每像素 4 位，则每行像素数必须为偶数。

位 15:0 **NL[15:0]**：行数 (Number of lines)

待传输区域的行数。只有在禁止数据传输的情况下才能写入这些位。传输一旦启动，这些位将变为只读。

13.5.19 DMA2D 行水印寄存器 (DMA2D_LWR)

DMA2D line watermark register

偏移地址：0x0048

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
LW[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 **LW[15:0]**：行水印 (Line watermark)
这些位可用以配置可产生中断的行水印。
在带水印行的最后一个像素传输完成时产生中断。只有在禁止数据传输的情况下才能写入这些位。传输一旦启动，这些位将变为只读。

13.5.20 DMA2D AHB 主设备定时器配置寄存器 (DMA2D_AMTCR)

DMA2D AHB master timer configuration register

偏移地址：0x004C

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
DT[7:0]								Res.	Res.	Res.	Res.	Res.	Res.	Res.	EN
rw	rw	rw	rw	rw	rw	rw	rw								rw

位 31:16 保留

位 15:8 **DT[7:0]**：死区 (Dead Time)
在 AHB 主设备端口上两个连续访问之间所插入的死区值，以 AHB 时钟周期数表示。这些位表示两个连续 AHB 访问之间允许占用的最少周期数。

位 7:1 保留

位 0 **EN**：使能 (Enable)
使能死区功能。

13.5.21 DMA2D 寄存器映射

下表对 DMA2D 寄存器进行了汇总。有关 DMA2D 寄存器基地址的信息，请参见*第 78 页的第 2.2.2 节*。

表 64. DMA2D 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
0x0000	DMA2D_CR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MODE[2:0]			Res.	Res.	CEIE	CTCIE	CAEIE	TWIE	TCIE	TEIE	Res.	LOM	Res.	Res.	Res.	ABORT	SUSP	START
	Reset value														0	0	0			0	0	0	0	0	0		0				0	0	0
0x0004	DMA2D_ISR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CEIF	CTCIF	CAEIF	TWIF	TCIF	TEIF
	Reset value																											0	0	0	0	0	0
0x0008	DMA2D_IFCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CCEIF	CCTCIF	CAECIF	CTWIF	CTCIF	CTEIF
	Reset value																											0	0	0	0	0	0
0x000C	DMA2D_FGMAR	MA[31:0]																															
	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
0x0010	DMA2D_FGOR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LO[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0014	DMA2D_BGMAR	MA[31:0]																															
	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
0x0018	DMA2D_BGOR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LO[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x001C	DMA2D_FGPFCCR	ALPHA[7:0]								Res.	Res.	RBS	AI	Res.	Res.	AM[1:0]		CS[7:0]								Res.	Res.	START	CCM	CM[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
0x0020	DMA2D_FGCOLR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RED[7:0]								GREEN[7:0]								BLUE[7:0]							
	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
0x0024	DMA2D_BGPFCCR	ALPHA[7:0]								Res.	Res.	RBS	AI	Res.	Res.	AM[1:0]		CS[7:0]								Res.	Res.	START	CCM	CM[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
0x0028	DMA2D_BGCOLR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RED[7:0]								GREEN[7:0]								BLUE[7:0]							
	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
0x002C	DMA2D_FGCMAR	MA[31:0]																															
	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
0x0030	DMA2D_BGCMAR	MA[31:0]																															
	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
0x0034	DMA2D_OPFCCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RBS	AI	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CM[2:0]		
	Reset value											0	0																		0	0	0

表 64. DMA2D 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
0x0038	DMA2D_OCOLR	ALPHA[7:0]								RED[7:0]								GREEN[7:0]								BLUE[7:0]							
		Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RED[4:0]					GREEN[5:0]						BLUE[4:0]				
		Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	A	RED[4:0]				GREEN[4:0]						BLUE[4:0]				
		Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ALPHA[3:0]				RED[3:0]				GREEN[3:0]				BLUE[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
0x003C	DMA2D_OMAR	MA[31:0]																															
	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
0x0040	DMA2D_OOR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LO[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0044	DMA2D_NLR	Res.	Res.	PL[13:0]														NL[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
0x0048	DMA2D_LWR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LW[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x004C	DMA2D_AMTCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DT[7:0]								Res.	Res.	Res.	Res.	Res.	Res.	Res.	EN
	Reset value																	0	0	0	0	0	0	0	0								0
0x0050-Ox03FC	Reserved	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
0x0400-0x07FC	DMA2D_FGCLUT	ALPHA[7:0][255:0]								RED[7:0][255:0]								GREEN[7:0][255:0]								BLUE[7:0][255:0]							
	Reset value	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
0x0800-0x0BFF	DMA2D_BGCLUT	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RED[7:0][255:0]								GREEN[7:0][255:0]								BLUE[7:0][255:0]							
	Reset value									X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

14 Chrom-GRC™ (GFXMMU)

14.1 简介

图形 MMU 是面向图形的存储器管理单元，旨在根据显示形状优化存储器使用率。

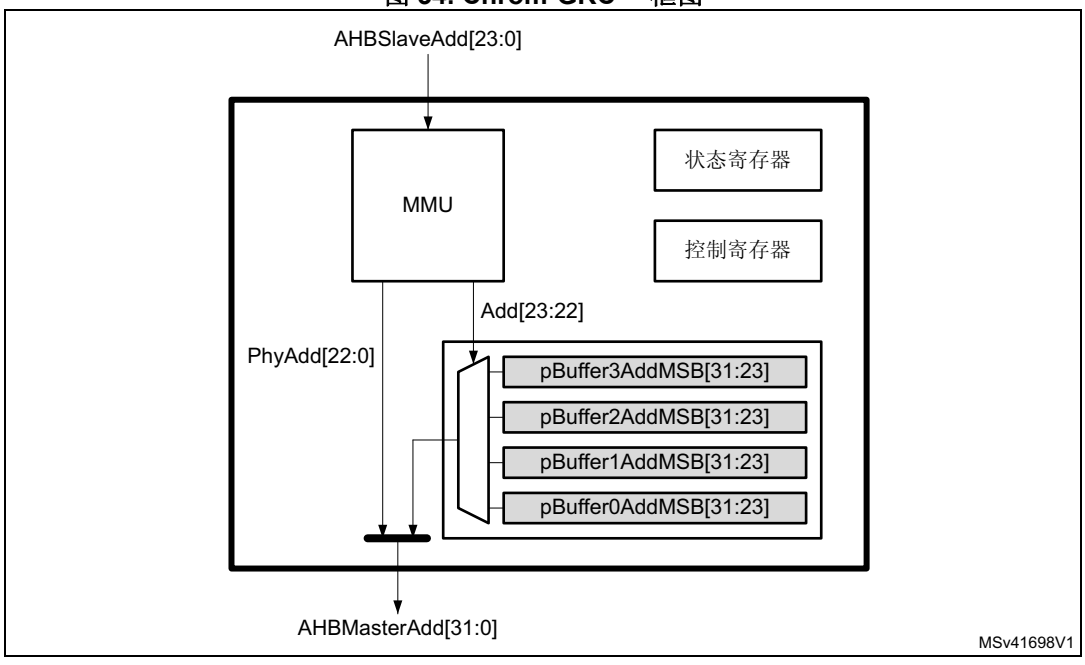
14.2 Chrom-GRC™ 主要特性

- 显示形状可完全编程，用于在物理上仅存储可见像素
- 多达 4 个虚拟缓冲区
- 对于每个虚拟缓冲区，每行有 3072 或 4096 个字节，共有 1024 行
- 每个虚拟缓冲区都可在物理上映射到任一系统存储器
- 缓冲区溢出时会生成中断（每个缓冲区 1 个中断）
- 发生存储器传输错误时生成中断

14.3 Chrom-GRC™ 功能和架构说明

图形 MMU 用于进行地址解析，以将虚拟缓冲区地址转换为物理缓冲区地址。

图 34. Chrom-GRC™ 框图



14.3.1 虚拟存储器

图形 MMU 提供了一个虚拟存储空间对所有系统主设备可见。此虚拟存储空间分为四个虚拟缓冲区。

虚拟缓冲区

系统主设备将虚拟缓冲区视为表示 1024 行虚拟帧缓冲区的连续存储空间。
每行分为 192 或 256 个 16 字节块，具体取决于图形 MMU 配置寄存器 (GFXMMU_CR) 的 192BM 位。

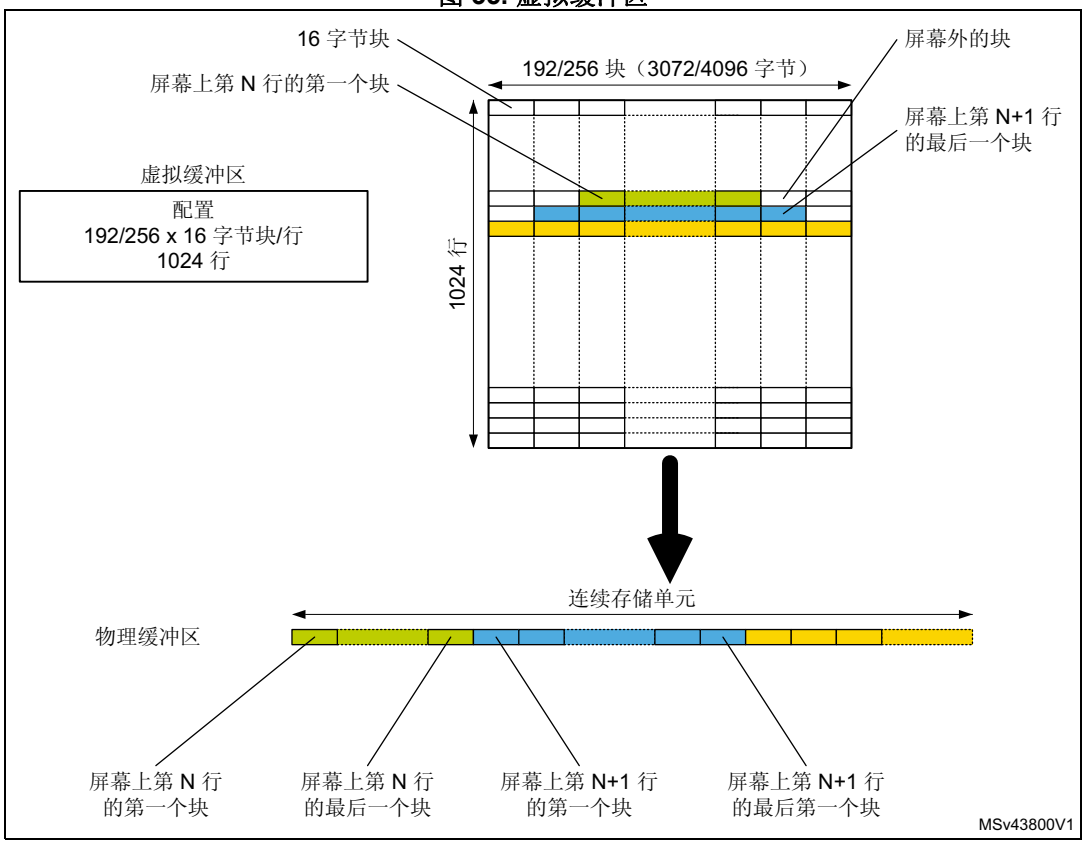
根据显示形状和大小，只会将必要的块映射到物理存储单元。此映射通过为每个行编程 LUT 条目实现：

- 行的使能
- 第一个“可见”块的编号
- 最后一个“可见”块的编号
- 物理缓冲区内行的偏移地址

通过编程每个行的偏移地址，可使“可见”块在物理缓冲区中连续排列。

LUT 由所有缓冲区共用，即，所有缓冲区具有相同的“形状”。

图 35. 虚拟缓冲区



虚拟缓冲区概述

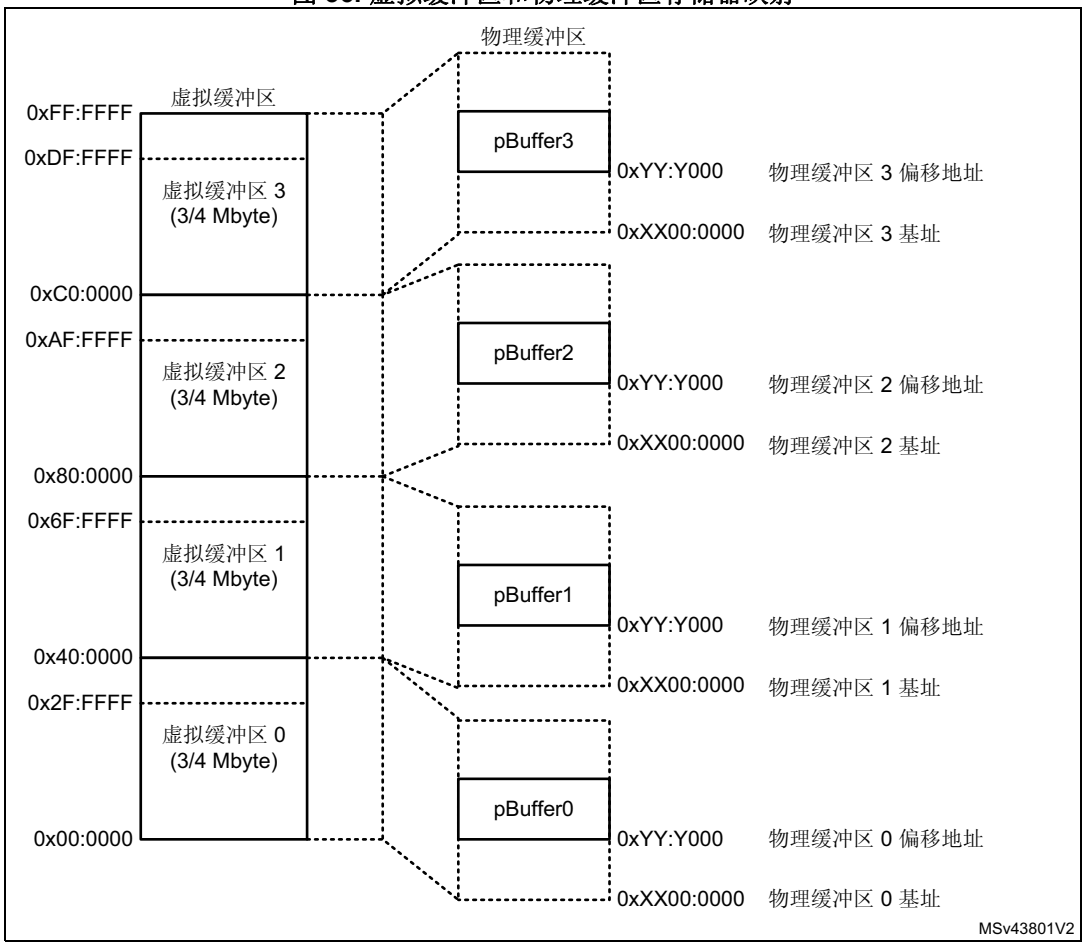
对于以 32bpp 或 16bpp 编码的帧缓冲区，虚拟缓冲区可配置为包含 192 或 256 个块。因此，对于 32bpp，虚拟帧缓冲区为 768 x 1024 或 1024 x 1024 像素；对于 16bpp 则为 1536 x 1024 或 2048 x 1024 像素。

对于以 24bpp 编码的帧缓冲区，虚拟缓冲区应配置为包含 192 个块，以使每行像素数为整数。因此，对于 24bpp，虚拟帧缓冲区为 1024 x 1024 像素。

基于以下内容，每个缓冲区均可在物理上映射到物理存储器中的任意位置：

- 图形 MMU 缓冲区 x 配置寄存器 (GFXMMU_BxCR) 的物理缓冲区基址 (PBBA) 字段。该字段配置缓冲区映射到的 8 MB 区域的物理单元。
- 相对于物理缓冲区基址的物理缓冲区单元由图形 MMU 缓冲区 x 配置寄存器 (GFXMMU_BxCR) 的物理缓冲区偏移 (PBO) 字段定义。

图 36. 虚拟缓冲区和物理缓冲区存储器映射



缓冲区不能溢出其基址所定义区域的 8 MB 边界。如果溢出，则图形 MMU 状态寄存器 (GFXMMU_SR) 的缓冲区 x 溢出标志 (BxOF) 将置 1，并且在图形 MMU 配置寄存器 (GFXMMU_CR) 的缓冲区 x 溢出中断使能 (BxOIE) 位置 1 时会生成中断。

虚拟缓冲区应用用例

物理单元可单独配置，因此四个虚拟缓冲区可在物理上映射到不连续单元。例如，这会使四个缓冲区映射到四个不同的 SDRAM 存储区，避免用于访问 SDRAM 的额外预充电周期。

因此，CPU/ChromART® 应使用其中一个缓冲区进行帧缓冲区计算，而 LTDC 应使用另一个缓冲区。

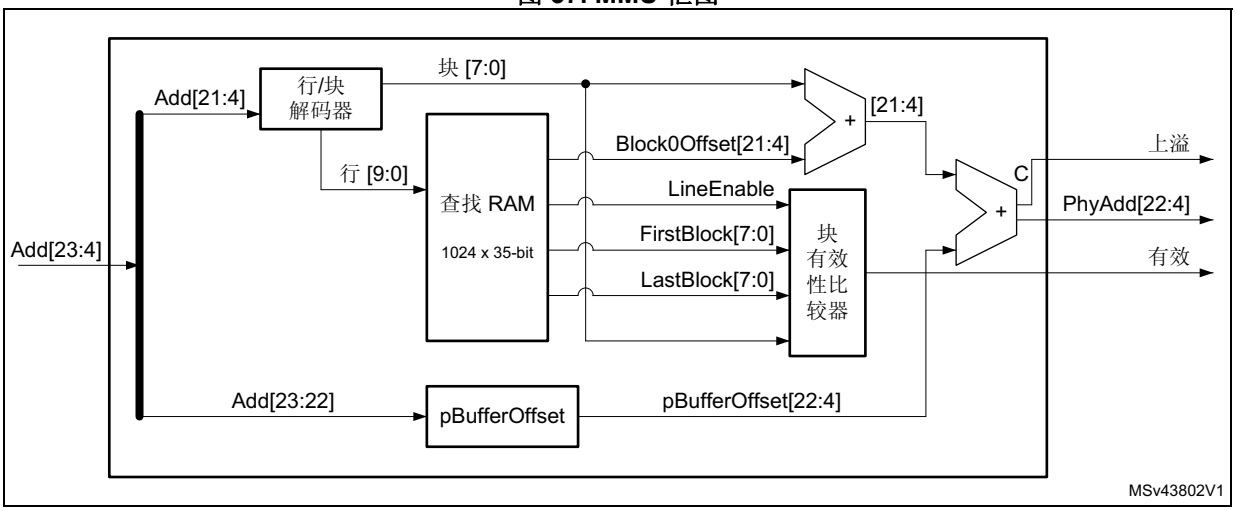
剩余的两个缓冲区可用作需要额外绘制缓冲区的图形库。

14.3.2 MMU 架构

MMU 块用于进行地址解析。它接收 24 位地址并返回 23 位物理地址和一个有效信号，以指示地址是否已进行物理映射。MMU 还会检查区域边界的溢出情况。

MMU LUT 以 1024 x 35 位 RAM 的形式实现。

图 37. MMU 框图



行块解码器

行块解码器根据地址生成块编号以及行编号。

查找表 RAM

查找表 RAM 是一个包含以下字段的 1024 x 35 位 RAM：

- 1 位用于行使能
- 8 位用于第一个有效块
- 8 位用于最后一个有效块
- 18 位用于行偏移

由于 RAM 的大小大于一个字，因此每个条目在存储器映射时均被拆分为两个字。写访问分两步执行：

1. 在 GFXMMU_LUTxL 存储单元中写入包含行使能/第一个有效块/最后一个有效块的第一个字（在内部缓冲）
2. 在 GFXMMU_LUTxH 存储单元中写入包含行偏移的第二个字（与在内部缓冲的值一起有效写入存储器中）

可随时对 LUT 执行写操作，但若与此同时主设备正在使用 MMU 则会导致不一致。由于在 LUT 编程期间 CPU 具有优先权，因此这可能会降低 MMU 的计算速度。

在执行读取操作期间无限制，但这可能会降低 CPU 速度，因为 MMU 具有访问 LUT 的优先权。

块验证/比较器

对块进行检查，确认块是否有效。

在以下情况下，会将块视为有效（即，已进行物理映射）

- 行已使能
- 块编号大于等于第一个有效块编号
- 块编号小于等于最后一个有效块编号

如果块有效，则生成的物理地址将视为正确地址。

如果 MMU 计值结果为无效，则会忽略写操作，读操作将返回图形 MMU 默认值寄存器 (GFXMMU_DVR) 的默认值 (DV) 字段中存储的默认 32 位值。

缓冲区中的块偏移地址计算

块编号加上行偏移可得到物理缓冲区中的块偏移。

因此，存储在 LUT 中的行偏移由以下公式得出：

行偏移 = [(已使用的可见块数) - (第一个可见块)] x 块大小

其中：

- 使用所有行的所有块时，行偏移的值最大。因此，最后一行的行偏移可为最大值：1023 x 256 x 6 = 0x3F:F00x
- 第一行的最后一个块是第一个有效块时，行偏移的值最小：-255*16 = - 0xFFx，即 0x3F:F01x

因此，使用 LUT 的行偏移条目的完整范围。

不考虑进位，因为此阶段能够执行负偏移计算（0x3F:F01x 到 0x3F:FFFx 之间的值）

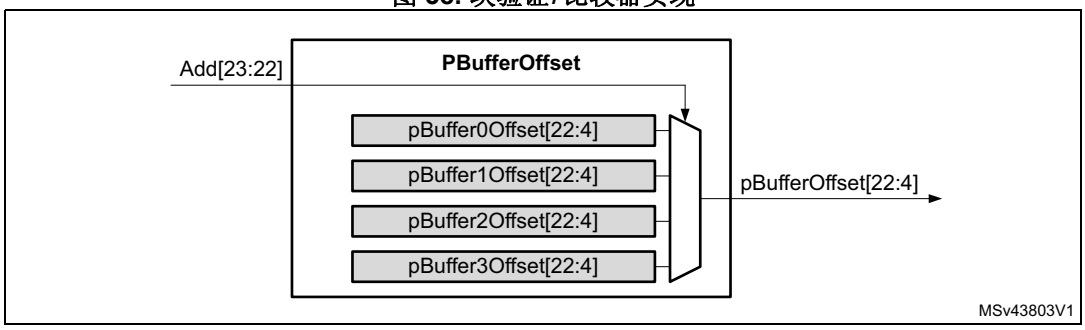
由于块偏移处于 4 MB 缓冲区内，因此生成的地址为 22 位宽。

块偏移地址计算

计算出缓冲区内的块偏移后，该值将与块相对于物理缓冲区基址的偏移相加。

块的偏移在寄存器中定义，如下图所示。

图 38. 块验证/比较器实现



得到的地址和缓冲区偏移地址应为 23 位。

考虑进位以触发地址溢出。进位被传播到图形 MMU 状态寄存器 (GFXMMU_SR)，以将缓冲区 x 溢出标志 (BxOF) 置 1。

计算示例

如果虚拟缓冲区 0 采用以下配置

- 行 0 的第一个可见块：块 7
- 行 0 的可见块数：10
- 行 1 的第一个可见块：块 6
- 行 1 的可见块数：12
- 物理缓冲区地址：0xC020:0000

配置应为：

- 物理缓冲区 0 基址：0xC000:0000
- 缓冲区 0 的偏移：0x20:0000
- 行 0 的第一个可见块：块 7
- 行 0 的最后一个可见块：块 16
- 行 0 的块 0 偏移：(0 - 7) x 0x10 = -0x70 = 0x3F:FF90
- 行 1 的第一个可见块：块 6
- 行 1 的最后一个可见块：块 17
- 行 1 的块 0 偏移：(10 - 6) x 0x10 = (0xA - 0x6) x 0x10 = 0x40

因此：

- 行 0 的块 7 的物理地址为：

0xC000:0000 + 0x20:0000 + (0x3F:FF90 + 0x70 不进位) = 0xC020:0000

- 行 0 的块 16 的物理地址为：

0xC000:0000 + 0x20:0000 + (0x3F:FF90 + 0x100 不进位) = 0xC020:0090

- 行 1 的块 6 的物理地址为：

0xC000:0000 + 0x20:0000 + (0x40 + 0x60 不进位) = 0xC020:00A0

- 行 1 的块 17 的物理地址为：

0xC000:0000 + 0x20:0000 + (0x40 + 0x110 不进位) = 0xC020:0150

14.4 图形 MMU 中断

发生如下事件时可生成中断：

- 缓冲区 0 溢出
- 缓冲区 1 溢出
- 缓冲区 2 溢出
- 缓冲区 3 溢出
- AHB 主设备错误

可以使用单独的中断使能位以提高灵活性。

表 65. 图形 MMU 中断请求

中断事件	事件标志	使能控制位
缓冲区 0 溢出	B0OF	B0OIE
缓冲区 1 溢出	B1OF	B1OIE
缓冲区 2 溢出	B2OF	B2OIE
缓冲区 3 溢出	B3OF	B3OIE
AHB 主设备错误	AMEF	AMEIE

14.5 图形 MMU 寄存器

14.5.1 图形 MMU 配置寄存器 (GFXMMU_CR)

Graphic MMU configuration 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
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.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	192BM	Res.	AMEIE	B3OIE	B2OIE	B1OIE	B0OIE
									rw		rw	rw	rw	rw	rw

位 31:7 保留，必须保持复位值。

位 6 **192BM**：192 块模式 (192 Block mode)
该位定义每行的块数。
0：每行 256 个块
1：每行 192 个块

位 5 保留，必须保持复位值。

位 4 **AMEIE**：AHB 主设备错误中断使能 (AHB master error interrupt enable)
该位使能 AHB 主设备错误中断。
0：禁止中断
1：使能中断

位 3 **B3OIE**：缓冲区 3 溢出中断使能 (Buffer 3 overflow interrupt enable)
该位使能缓冲区 3 溢出中断。
0：禁止中断
1：使能中断

位 2 **B2OIE**：缓冲区 2 溢出中断使能 (Buffer 2 overflow interrupt enable)
该位使能缓冲区 2 溢出中断。
0：禁止中断
1：使能中断

位 1 **B1OIE**：缓冲区 1 溢出中断使能 (Buffer 1 overflow interrupt enable)
该位使能缓冲区 1 溢出中断。
0：禁止中断
1：使能中断

位 0 **B0OIE**：缓冲区 0 溢出中断使能 (Buffer 0 overflow interrupt enable)
该位使能缓冲区 0 溢出中断。
0：禁止中断
1：使能中断

14.5.2 图形 MMU 状态寄存器 (GFXMMU_SR)

Graphic MMU status 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.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	AMEF	B3OF	B2OF	B1OF	B0OF
											r	r	r	r	r

位 31:5 保留，必须保持复位值。

位 4 **AMEF**：AHB 主设备错误标志 (AHB master error flag)
在事务期间发生 AHB 错误时该位置 1。向 CAMEF 写入 1 可将该位清零。

位 3 **B3OF**：缓冲区 3 溢出标志 (Buffer 3 overflow flag)
在对缓冲区 3 的偏移进行计算期间发生溢出时该位置 1。向 CB3OF 写入 1 可将该位清零。

位 2 **B2OF**：缓冲区 2 溢出标志 (Buffer 2 overflow flag)
在对缓冲区 2 的偏移进行计算期间发生溢出时该位置 1。向 CB2OF 写入 1 可将该位清零。

位 1 **B1OF**：缓冲区 1 溢出标志 (Buffer 1 overflow flag)
在对缓冲区 1 的偏移进行计算期间发生溢出时该位置 1。向 CB1OF 写入 1 可将该位清零。

位 0 **B0OF**：缓冲区 0 溢出标志 (Buffer 0 overflow flag)
在对缓冲区 0 的偏移进行计算期间发生溢出时该位置 1。向 CB0OF 写入 1 可将该位清零。

14.5.3 图形 MMU 标志清零寄存器 (GFXMMU_FCR)

Graphic MMU flag clear 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.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CAMEF	CB3OF	CB2OF	CB1OF	CB0OF
											rc_w1	rc_w1	rc_w1	rc_w1	rc_w1

位 31:5 保留，必须保持复位值。

位 4 **CAMEF**：清零 AHB 主设备错误标志 (Clear AHB master error flag)
写入 1 可将 GFXMMU_SR 寄存器中的 AHB 主设备错误标志清零。

位 3 **CB3OF**：清零缓冲区 3 溢出标志 (Clear buffer 3 overflow flag)
写入 1 可将 GFXMMU_SR 寄存器中的缓冲区 3 溢出标志清零。

位 2 **CB2OF**：清零缓冲区 2 溢出标志 (Clear buffer 2 overflow flag)
写入 1 可将 GFXMMU_SR 寄存器中的缓冲区 2 溢出标志清零。

位 1 **CB1OF**：清零缓冲区 1 溢出标志 (Clear buffer 1 overflow flag)
写入 1 可将 GFXMMU_SR 寄存器中的缓冲区 1 溢出标志清零。

位 0 **CB0OF**：清零缓冲区 0 溢出标志 (Clear buffer 0 overflow flag)
写入 1 可将 GFXMMU_SR 寄存器中的缓冲区 0 溢出标志清零。

14.5.4 图形 MMU 默认值寄存器 (GFXMMU_DVR)

Graphic MMU default value 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
DV[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
DV[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:0 **DV[31:0]**：默认值 (Default value)
该字段指示默认的 32 位值，主设备访问未进行物理映射的虚拟存储单元时返回该值。

14.5.5 图形 MMU 缓冲区 0 配置寄存器 (GFXMMU_B0CR)

Graphic MMU buffer 0 configuration 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
PBBA[31:23]									PBO[22:16]						
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
PBO[15:4]												Res.	Res.	Res.	Res.
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw				

位 31:23 **PBBA[31:23]**：物理缓冲区基址 (Physical buffer base address)
物理缓冲区的基址 MSB。

位 22:4 **PBO[22:4]**：物理缓冲区偏移 (Physical buffer offset)
物理缓冲区的偏移。

位 3:0 保留，必须保持复位值。

14.5.6 图形 MMU 缓冲区 1 配置寄存器 (GFXMMU_B1CR)

Graphic MMU buffer 1 configuration 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
PBBA[31:23]									PBO[22:16]						
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
PBO[15:4]												Res.	Res.	Res.	Res.
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw				

位 31:23 **PBBA[31:23]**：物理缓冲区基址 (Physical buffer base address)
物理缓冲区的基址 MSB。

位 22:4 **PBO[22:4]**：物理缓冲区偏移 (Physical buffer offset)
物理缓冲区的偏移。

位 3:0 保留，必须保持复位值。

14.5.7 图形 MMU 缓冲区 2 配置寄存器 (GFXMMU_B2CR)

Graphic MMU buffer 2 configuration 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
PBBA[31:23]									PBO[22:16]						
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
PBO[15:4]												Res.	Res.	Res.	Res.
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw				

位 31:23 **PBBA[31:23]**：物理缓冲区基址 (Physical buffer base address)
物理缓冲区的基址 MSB。

位 22:4 **PBO[22:4]**：物理缓冲区偏移 (Physical buffer offset)
物理缓冲区的偏移。

位 3:0 保留，必须保持复位值。

14.5.8 图形 MMU 缓冲区 3 配置寄存器 (GFXMMU_B3CR)

Graphic MMU buffer 3 configuration register

偏移地址：0x002C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PBBA[31:23]									PBO[22:16]						
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
PBO[15:4]												Res.	Res.	Res.	Res.
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw				

位 31:23 **PBBA[31:23]**：物理缓冲区基址 (Physical buffer base address)
物理缓冲区的基址 MSB。

位 22:4 **PBO[22:4]**：物理缓冲区偏移 (Physical buffer offset)
物理缓冲区的偏移。

位 3:0 保留，必须保持复位值。

14.5.9 图形 MMU LUT 条目 x 低位 (GFXMMU_LUTxL)

Graphic MMU LUT entry x low

偏移地址：0x1000 + 8 * x，x = 0...1023

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

位 31:24 保留，必须保持复位值。

位 23:16 **LVB[7:0]**：最后一个有效块 (Last Valid Block)
行编号 x 的最后一个有效块的编号。

位 15:8 **FVB[7:0]**：第一个有效块 (First Valid Block)
行编号 x 的第一个有效块的编号。

位 7:1 保留，必须保持复位值。

位 0 **EN**：使能 (Enable)
行使能。
0：禁止行（不执行 MMU 计值）
1：使能行（执行 MMU 计值）

14.5.10 图形 MMU LUT 条目 x 高位 (GFXMMU_LUTxH)

Graphic MMU LUT entry x high

偏移地址：0x1000 + 8 * x + 4，x = 0...1023

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.	LO[21:16]					
										rw	rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LO[15:4]												Res.	Res.	Res.	Res.
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 21:4 **LO[21:4]**：行偏移 (Line offset)
行编号 x 的行偏移（即，行 x 的块 0 的偏移）

位 3:0 保留，必须保持复位值。

14.5.11 图形 MMU 寄存器映射

下表对图形 MMU 寄存器进行了汇总。有关图形 MMU 寄存器的基址，请参见寄存器边界地址表。

表 66. 图形 MMU 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
0x0000	**GFXMMU_CR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	192BM	Res.	AMEIE	B3OIE	B2OIE	B1OIE	B0OIE
	Reset value																										0		0	0	0	0	0
0x0004	**GFXMMU_ISR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	AMEF	B3OF	B2OF	B1OF	B0OF
	Reset value																												0	0	0	0	0
0x0008	**GFXMMU_FCR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CAMEF	CB3OF	CB2OF	CB1OF	CB0OF
	Reset value																												0	0	0	0	0
0x000C	**Reserved**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
0x0010	**GFXMMU_DRR**	DV[31:0]																															
	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
0x0014 0x001C	**Reserved**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
0x0020	**GFXMMU_B0CR**	PBBA[31:23]									PBO[22:4]																			Res.	Res.	Res.	Res.
	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				
0x0024	**GFXMMU_B1CR**	PBBA[31:23]									PBO[22:4]																			Res.	Res.	Res.	Res.
	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				
0x0028	**GFXMMU_B2CR**	PBBA[31:23]									PBO[22:4]																			Res.	Res.	Res.	Res.
	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				
0x002C	**GFXMMU_B3CR**	PBBA[31:23]									PBO[22:4]																			Res.	Res.	Res.	Res.
	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				
0x0030 0x0FEC 0x0FF0	**Reserved**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
0x1000	**GFXMMU_ LUT0L**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LVB[7:0]								FVB[7:0]								Res.	Res.	Res.	Res.	Res.	Res.	Res.	EN
	Reset value									0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0								0
0x1004	**GFXMMU_ LUT0H**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LO[21:4]																		Res.	Res.	Res.	Res.
	Reset value										0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				
...																																	
0x2FF8	**GFXMMU_ LUT1023L**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LVB[7:0]								FVB[7:0]								Res.	Res.	Res.	Res.	Res.	Res.	Res.	EN
	Reset value									0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0								0
0x2FFC	**GFXMMU_ LUT1023H**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LO[21:4]																		Res.	Res.	Res.	Res.
	Reset value										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*第 78 页的第 2.2.2 节*。

15 嵌套向量中断控制器 (NVIC)

15.1 NVIC 主要特性

- 95 个可屏蔽中断通道（不包括带 FPU 的 Cortex®-M4 的 16 根中断线）
- 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。有关异常和 NVIC 编程的更多信息，请参见 Cortex™-M4 产品的 PM0214 编程手册。

15.2 SysTick 校准值寄存器

SysTick 校准值设置为 0x40003A97，当 SysTick 时钟设置为 15 MHz（最大 $f_{HCLK}/8$）时，会产生 1 ms 时间基准。

15.3 中断和异常向量

表 67 中的灰色行表示无特定位置的向量。

表 67. STM32L4Rxxx 和 STM32L4Sxxx 向量表

位置	优先级	优先级类型	缩略语	说明	地址
-	-	-	-	保留	0x0000 0000
-	-3	固定	复位	复位	0x0000 0004
-	-2	固定	NMI	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 0028
-	3	-	-	通过 SWI 指令调用的系统服务	0x0000 002C
-	4	-	-	监控	0x0000 0030
-	-	-	-	保留	0x0000 0034
-	5	可设置	PendSV	可挂起的系统服务请求	0x0000 0038
-	6	可设置	SysTick	系统节拍定时器	0x0000 003C
0	7	可设置	WWDG	窗口看门狗中断	0x0000 0040
1	8	可设置	PVD_PVM	PVD/PVM1/PVM2/PVM3/PVM4（通过 EXTI线 16/35/36/37/38 中断实现）	0x0000 0044
2	9	可设置	RTC_TAMP_STAMP/ CSS_LSE	RTC 入侵或时间戳/LSE 上的 CSS（通过 EXTI 线 19 中断实现）	0x0000 0048
3	10	可设置	RTC_WKUP	RTC 唤醒定时器，通过 EXTI 线 20 中断实现	0x0000 004C
4	11	可设置	闪存	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_CH1	DMA1 通道 1 中断	0x0000 006C

表 67. STM32L4Rxxx 和 STM32L4Sxxx 向量表（续）

位置	优先级	优先级类型	缩略语	说明	地址
12	19	可设置	DMA1_CH2	DMA1 通道 2 中断	0x0000 0070
13	20	可设置	DMA1_CH3	DMA1 通道 3 中断	0x0000 0074
14	21	可设置	DMA1_CH4	DMA1 通道 4 中断	0x0000 0078
15	22	可设置	DMA1_CH5	DMA1 通道 5 中断	0x0000 007C
16	23	可设置	DMA1_CH6	DMA1 通道 6 中断	0x0000 0080
17	24	可设置	DMA1_CH7	DMA1 通道 7 中断	0x0000 0084
18	25	可设置	ADC1	ADC1 全局中断	0x0000 0088
19	26	可设置	CAN1_TX	CAN1_TX 中断	0x0000 008C
20	27	可设置	CAN1_RX0	CAN1_RX0 中断	0x0000 0090
21	28	可设置	CAN1_RX1	CAN1_RX1 中断	0x0000 0094
22	29	可设置	CAN1_SCE	CAN1_SCE 中断	0x0000 0098
23	30	可设置	EXTI9_5	EXTI 线 [9:5] 中断	0x0000 009C
24	31	可设置	TIM1_BRK/TIM15	TIM1 刹车/TIM15 全局中断	0x0000 00A0
25	32	可设置	TIM1_UP/TIM16	TIM1 更新/TIM16 全局中断	0x0000 00A4
26	33	可设置	TIM1_TRG_COM/ TIM17	TIM1 触发和换向/TIM17 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	I2C1 事件中断	0x0000 00BC
32	39	可设置	I2C1_ER	I2C1 错误中断	0x0000 00C0
33	40	可设置	I2C2_EV	I2C2 事件中断	0x0000 00C4
34	41	可设置	I2C2_ER	I2C2 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
41	48	可设置	RTC_ALARM	RTC 闹钟（通过 EXTI 线 18 中断实现）	0x0000 00E4
42	49	可设置	DFSDM1_FLT3	DFSDM1_FLT3 全局中断	0x0000 00E8
43	50	可设置	TIM8_BRK	TIM8 刹车中断	0x0000 00EC

表 67. STM32L4Rxxx 和 STM32L4Sxxx 向量表（续）

位置	优先级	优先级类型	缩略语	说明	地址
44	51	可设置	TIM8_UP	TIM8 更新中断	0x0000 00F0
45	52	可设置	TIM8_TRG_COM	TIM8 触发和换向中断	0x0000 00F4
46	53	可设置	TIM8_CC	TIM8 捕获/比较中断	0x0000 00F8
47	54	可设置	保留	保留	0x0000 00FC
48	55	可设置	FMC	FMC 全局中断	0x0000 0100
49	56	可设置	SDMMC1	SDMMC1 全局中断	0x0000 0104
50	57	可设置	TIM5	TIM5 全局中断	0x0000 0108
51	58	可设置	SPI3	SPI3 全局中断	0x0000 010C
52	59	可设置	UART4	UART4 全局中断	0x0000 0110
53	60	可设置	UART5	UART5 全局中断	0x0000 0114
54	61	可设置	TIM6_DACUNDER	TIM6 全局和 DAC1 下溢中断	0x0000 0118
55	62	可设置	TIM7	TIM7 全局中断	0x0000 011C
56	63	可设置	DMA2_CH1	DMA2 通道 1 中断	0x0000 0120
57	64	可设置	DMA2_CH2	DMA2 通道 2 中断	0x0000 0124
58	65	可设置	DMA2_CH3	DMA2 通道 3 中断	0x0000 0128
59	66	可设置	DMA2_CH4	DMA2 通道 4 中断	0x0000 012C
60	67	可设置	DMA2_CH5	DMA2 通道 5 中断	0x0000 0130
61	68	可设置	DFSDM1_FLT0	DFSDM1_FLT0 全局中断	0x0000 0134
62	69	可设置	DFSDM1_FLT1	DFSDM1_FLT1 全局中断	0x0000 0138
63	70	可设置	DFSDM1_FLT2	DFSDM1_FLT2 全局中断	0x0000 013C
64	71	可设置	COMP	COMP1/COMP2（通过 EXTI 线 21/22 中断实现）	0x0000 0140
65	72	可设置	LPTIM1	LPTIM1 全局中断	0x0000 0144
66	73	可设置	LPTIM2	LPTIM2 全局中断	0x0000 0148
67	74	可设置	OTG_FS	OTG_FS 全局中断	0x0000 014C
68	75	可设置	DMA2_CH6	DMA2 通道 6 中断	0x0000 0150
69	76	可设置	DMA2_CH7	DMA2 通道 7 中断	0x0000 0154
70	77	可设置	LPUART1	LPUART1 全局中断	0x0000 0158
71	78	可设置	OCTOSPI1	OCTOSPI1 全局中断	0x0000 015C
72	79	可设置	I2C3_EV	I2C3 事件中断	0x0000 0160
73	80	可设置	I2C3_ER	I2C3 错误中断	0x0000 0164
74	81	可设置	SAI1	SAI1 全局中断	0x0000 0168
75	82	可设置	SAI2	SAI2 全局中断	0x0000 016C

表 67. STM32L4Rxxx 和 STM32L4Sxxx 向量表（续）

位置	优先级	优先级类型	缩略语	说明	地址
76	83	可设置	OCTOSPI2	OCTOSPI2 全局中断	0x0000 0170
77	84	可设置	TSC	TSC 全局中断	0x0000 0174
78	85	可设置	DSIHSOT	DSI 全局中断	0x0000 0178
79	86	可设置	AES	AES 全局中断	0x0000 017C
80	87	可设置	RNG	RNG 全局中断	0x0000 0180
81	88	可设置	FPU	浮点中断	0x0000 0184
82	89	可设置	HASH 和 CRS	HASH 和 CRS 中断	0x0000 0188
83	90	可设置	I2C4_EV	I2C4 事件中断	0x0000 018C
84	91	可设置	I2C4_ER	I2C4 错误中断	0x0000 0190
85	92	可设置	DCMI	DCMI 全局中断	0x0000 0194
86	93	可设置	保留	保留	0x0000 0198
87	94	可设置	保留	保留	0x0000 019C
88	95	可设置	保留	保留	0x0000 01A0
89	96	可设置	保留	保留	0x0000 01A4
90	97	可设置	DMA2D	DMA2D 全局中断	0x0000 01A8
91	98	可设置	LCD-TFT	LTDC 全局中断	0x0000 019C
92	99	可设置	LCD-TFT_ER	LTDC 全局错误中断	0x0000 01A0
93	100	可设置	GFXMMU	GFXMMU 全局错误中断	0x0000 01A4
94	101	可设置	DMAMUX1_OVR	DMAMUX 溢出中断	0x0000 01A8

16 扩展中断和事件控制器 (EXTI)

16.1 简介

EXTI 的主要特性如下：

- 可生成多达 39 个事件/中断请求
  - 26 条可配置线
  - 12 条无需配置的线
- 单独屏蔽每根事件/中断线
- 可配置上升沿或下降沿（仅限可配置线）
- 专用状态位（仅限可配置线）
- 事件/中断仿真请求（仅限可配置线）

16.2 EXTI 主要特性

扩展中断和事件控制器 (EXTI) 用于管理内外部异步事件/中断，并生成对 CPU/中断控制器的事件请求和对电源控制器的唤醒请求。

EXTI 最多可管理 39 根事件线，这些事件线可将器件从停止 0 和停止 1 模式唤醒。并非所有事件都能将器件从停止 2 模式唤醒（请参见*表 68：EXTI 线连接*）。

线可以是可配置线，也可以是无需配置的线：

- 线可配置：可以独立选择有效边沿，并通过状态标志指示中断源。可配置线供 I/O 外部中断和少部分外设使用。
- 线无需配置：它们供部分外设使用，以生成从停止模式唤醒的事件或中断。状态标志由外设提供。

每根线均可单独屏蔽生成的中断或事件。

该控制器还允许通过写入专用寄存器，来在与其相对应的硬件事件线上对事件或中断进行软件仿真。

16.3 EXTI 功能说明

对于可配置的中断线，应配置好并使能中断线才能产生中断。根据需要的边沿检测设置 2 个触发寄存器，同时在中断屏蔽寄存器的相应位写“1”使能中断请求。当中断线上出现选定信号沿时，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。

对于无需配置的中断线：默认在中断屏蔽寄存器中使能中断，挂起寄存器中没有对应的挂起位。

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。

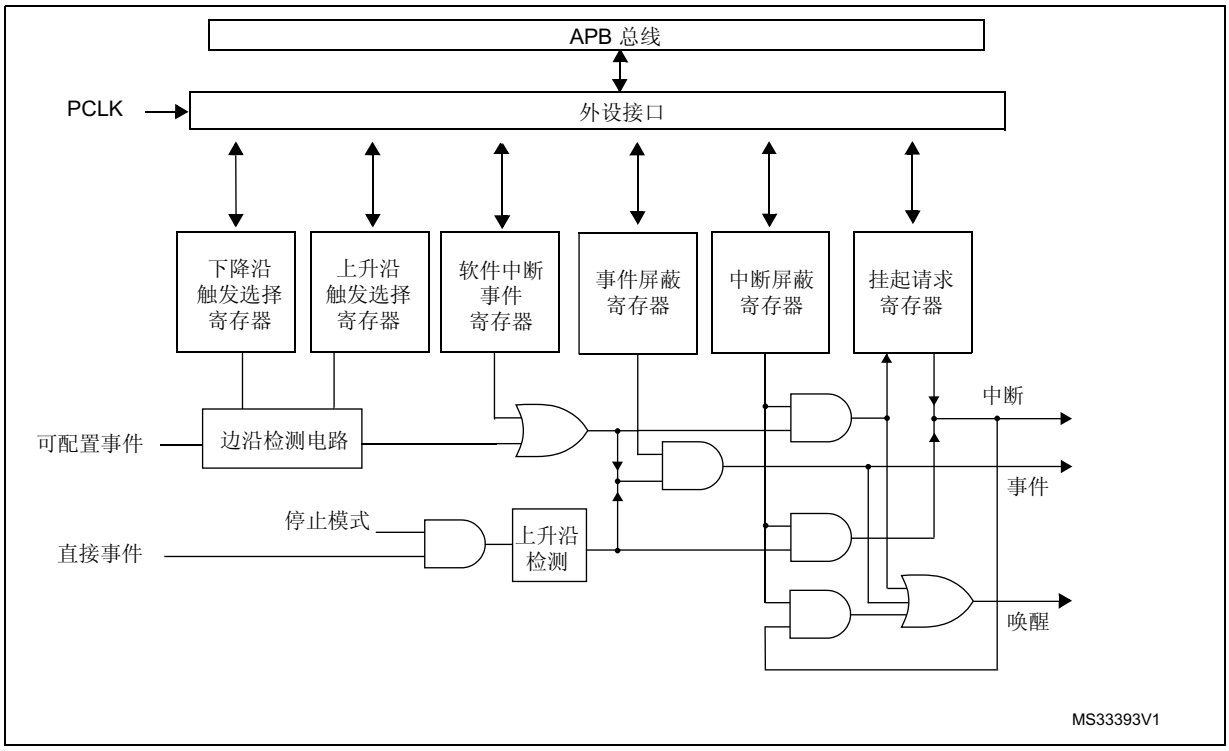
对于可配置线，通过在软件中对软件中断/事件寄存器写“1”，也可以产生中断/事件请求。

注： *与无需配置的线相关的中断或事件仅在系统处于停止模式时触发。如果系统仍在运行，EXTI 不会产生任何中断/事件。*

16.3.1 EXTI 框图

扩展中断/事件框图如*图 39* 所示。

图 39. 可配置中断/事件框图



16.3.2 唤醒事件管理

STM32L4Rxxx 和 STM32L4Sxxx 能够处理唤醒内核 (WFE) 的外部或内部事件。唤醒事件可通过以下方式产生：

- 在外设的控制寄存器中使能一个中断，但不在 NVIC 中使能，同时使能 Cortex™-M4 系统控制寄存器中的 SEVONPEND 位。当 MCU 从 WFE 恢复时，需要清除相应 EXTI 外设的中断挂起位和外设 NVIC IRQ 通道挂起位（在 NVIC 中断清除挂起寄存器中）。
- 将一个 EXTI 线配置为事件模式。当 CPU 从 WFE 恢复时，因为对应事件线的挂起位没有被置 1，不必清除相应外设的中断挂起位或 NVIC IRQ 通道挂起位。

16.3.3 外设异步中断

有些外设能够在系统处于运行模式和停止模式时都能产生事件，从而可将系统从停止模式唤醒。

为实现此目的，外设会同时生成同步（与系统时钟同步，即 APB 时钟）和异步两种事件。此异步事件连接到 EXTI 无需配置的线。

注： *少部分具有从停止模式唤醒功能的外设连接到 EXTI 可配置线。对于这类外设，需要配置 EXTI 才能从停止模式唤醒。*

16.3.4 硬件中断选择

要将一根线配置为中断源，请执行以下步骤：

1. 配置 EXTI_IMR 寄存器中相应的屏蔽位。
2. 配置中断线的触发选择位（EXTI_RTSR 和 EXTI_FTSR）。
3. 配置用于控制映射到 EXTI 的 NVIC IRQ 通道的使能和屏蔽位，从而使任一中断线中的请求可以被正确地响应。

注： *无需配置的线不需要进行任何 EXTI 配置。*

16.3.5 硬件事件选择

要将一根线配置为事件源，请执行以下步骤：

1. 配置 EXTI_EMR 寄存器中相应的屏蔽位。
2. 配置事件线的触发选择位（EXTI_RTSR 和 EXTI_FTSR）。

16.3.6 软件中断/事件选择

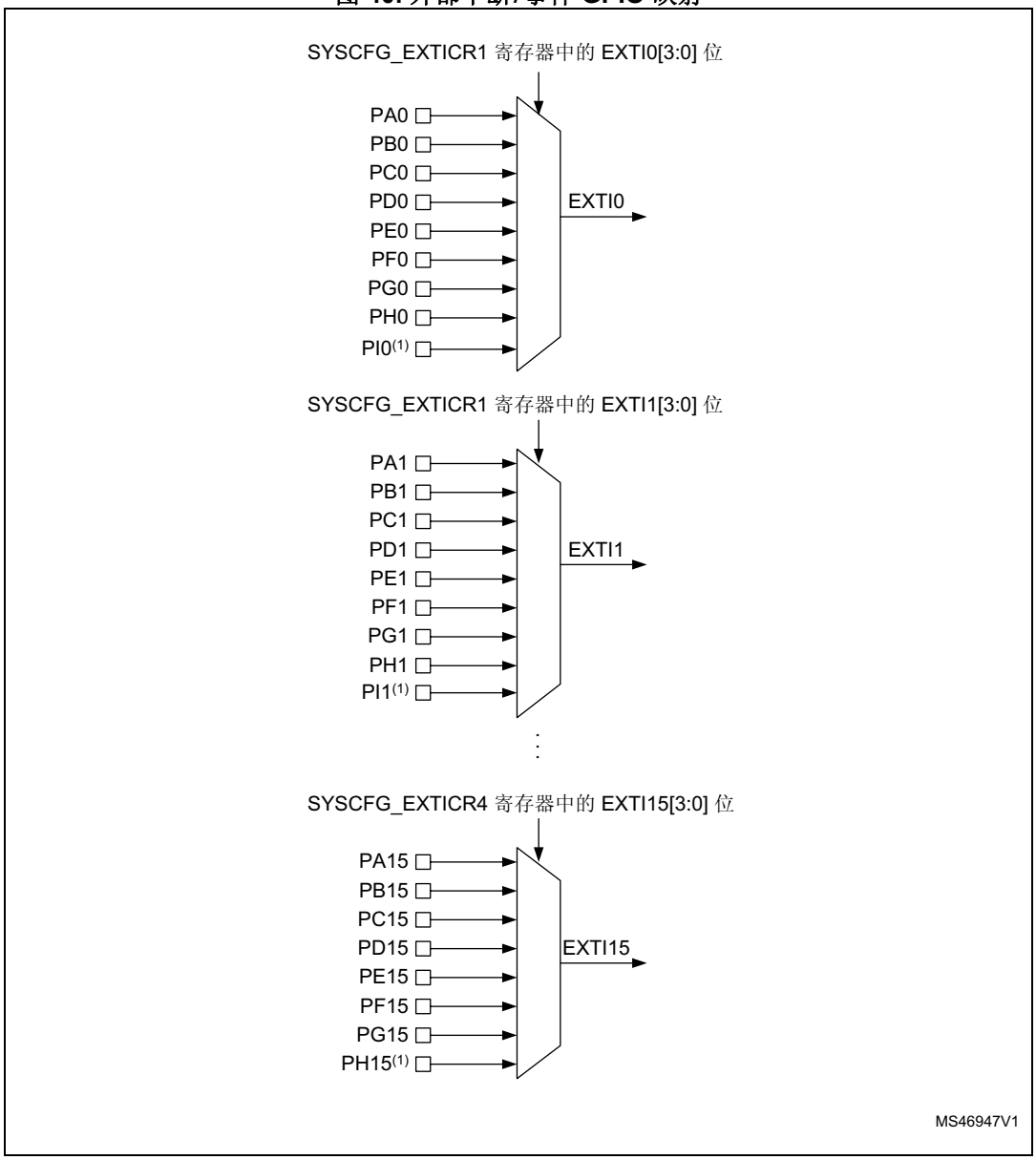
可将任一可配置线配置为软件中断/事件线。要产生软件中断，必须执行以下步骤：

1. 配置相应的屏蔽位（EXTI_IMR、EXTI_EMR）。
2. 在软件中断寄存器 (EXTI_SWIER) 设置相应的请求位。

16.4 EXTI 中断/事件线映射

在 STM32L4Rxxx 和 STM32L4Sxxx 中，有 39 根中断/事件线可用。GPIO 连接到 16 根可配置中断/事件线（请参见*图 40*）。

图 40. 外部中断/事件 GPIO 映射



2. GPIO PI12 到 PI15 不可用。

*表 68：EXTI 线连接*列出了 EXTI 线的连接方式。

表 68. EXTI 线连接

EXTI 线	线源(1)	线类型
0-15	GPIO	可配置
16	PVD	可配置
17	OTG FS 唤醒事件(2) (OTG_FS_WKUP)	无需配置
18	RTC 闹钟	可配置
19	RTC 入侵、时间戳或 CSS_LSE	可配置
20	RTC 唤醒定时器	可配置
21	COMP1 输出	可配置
22	COMP2 输出	可配置
23	I2C1 唤醒(2)	无需配置
24	I2C2 唤醒(2)	无需配置
25	I2C3 唤醒	无需配置
26	USART1 唤醒(2)	无需配置
27	USART2 唤醒(2)	无需配置
28	USART3 唤醒(2)	无需配置
29	UART4 唤醒(2)	无需配置
30	UART5 唤醒(2)	无需配置
31	LPUART1 唤醒	无需配置
32	LPTIM1	无需配置
33	LPTIM2(2)	无需配置
34	保留	保留
35	PVM1 唤醒	可配置
36	PVM2 唤醒	可配置
37	PVM3 唤醒	可配置
38	PVM4 唤醒	可配置
39	保留	保留
40	I2C4 唤醒	无需配置

1. 所有线均可将器件从停止 0 和停止 1 模式唤醒。除上述线之外，其余所有线均可将器件从停止 2 模式唤醒。
2. 此线源不能从停止 2 模式唤醒。

16.5 EXTI 寄存器

有关寄存器说明中使用的缩写，请参见*第 72 页的第 1.1 节*。

外设寄存器必须按字（32 位）进行访问。

16.5.1 中断屏蔽寄存器 1 (EXTI_IMR1)

Interrupt mask register 1

偏移地址：0x00

复位值：0xFF82 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IM31	IM30	IM29	IM28	IM27	IM26	IM25	IM24	IM23	IM22	IM21	IM20	IM19	IM18	IM17	IM16
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
IM15	IM14	IM13	IM12	IM11	IM10	IM9	IM8	IM7	IM6	IM5	IM4	IM3	IM2	IM1	IM0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:0 **IMx**：线 x 上的中断屏蔽 (Interrupt Mask on line x)（x = 0 到 31）
0：屏蔽来自线 x 的中断请求
1：开放来自线 x 的中断请求

注： *无需配置线（线 17、线 23 到 34、线 39）的复位值置“1”，以在默认情况下使能中断。*

16.5.2 事件屏蔽寄存器 1 (EXTI_EMR1)

Event mask 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
EM31	EM30	EM29	EM28	EM27	EM26	EM25	EM24	EM23	EM22	EM21	EM20	EM19	EM18	EM17	EM16
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
EM15	EM14	EM13	EM12	EM11	EM10	EM9	EM8	EM7	EM6	EM5	EM4	EM3	EM2	EM1	EM0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:0 **EMx**：线 x 上的事件屏蔽 (Event mask on line x)（x = 0 到 31）
0：屏蔽来自 x 线的事件请求
1：开放来自 x 线的事件请求

16.5.3 上升沿触发选择寄存器 1 (EXTI_RTSR1)

Rising trigger selection register 1

偏移地址：0x08

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RT22	RT21	RT20	RT19	RT18	Res.	RT16
									rw	rw	rw	rw	rw		rw

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

位 31:23 保留，必须保持复位值。

位 22:18 **RTx**：线 x 的上升沿触发事件配置位 (Rising trigger event configuration bit of line x)（x = 18 到 22）
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）

位 17 保留，必须保持复位值。

位 16:0 **RTx**：线 x 的上升沿触发事件配置位 (Rising trigger event configuration bit of line x)（x = 0 到 16）
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： *可配置唤醒线采用边沿触发。在这些线上不能出现毛刺信号。如果在对 EXTI_RTSR 寄存器执行写入操作时可配置的中断线上产生上升沿，挂起位不会置 1。*

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。

16.5.4 下降沿触发选择寄存器 1 (EXTI_FTSR1)

Falling trigger selection 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
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FT22	FT21	FT20	FT19	FT18	Res.	FT16
									rw	rw	rw	rw	rw		rw

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

位 31:23 保留，必须保持复位值。

位 22:18 **FTx**：线 x 的下降沿触发事件配置位 (Falling trigger event configuration bit of line x)（x = 18 到 22）
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）

位 17 保留，必须保持复位值。

位 16:0 **FTx**：线 x 的下降沿触发事件配置位 (Falling trigger event configuration bit of line x)（x = 0 到 16）
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： *可配置唤醒线采用边沿触发。在这些线上不能出现毛刺信号。如果在对 EXTI_FTSR 寄存器执行写入操作时可配置的中断线上产生下降沿，挂起位不会置 1。*

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。

16.5.5 软件中断事件寄存器 1 (EXTI_SWIER1)

Software interrupt event register 1

偏移地址：0x10

复位值：0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SWI 15	SWI 14	SWI 13	SWI 12	SWI 11	SWI 10	SWI 9	SWI 8	SWI 7	SWI 6	SWI 5	SWI 4	SWI 3	SWI 2	SWI 1	SWI 0
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: 18 **SWIx**：线 x 上的软件中断 (Software interrupt on line x)（x = 18 到 22）
如果在 EXTI_IMR 寄存器中允许在 x 线上产生该中断，则当此位设置为“0”时将“1”写入该位会将 EXTI_PR 中的相应挂起位置 1，进而产生中断请求。
通过清除 EXTI_PR 寄存器的相应位（对此位写入“1”），可以将该位清零。

位 17 保留，必须保持复位值。

位 16:0 **SWIx**：线 x 上的软件中断 (Software interrupt on line x)（x = 0 到 16）
如果在 EXTI_IMR 寄存器中允许在 x 线上产生该中断，则当此位设置为“0”时将“1”写入该位会将 EXTI_PR 中的相应挂起位置 1，进而产生中断请求。
通过清除 EXTI_PR 的相应位（对此位写入“1”），可以将该位清零。

16.5.6 挂起寄存器 1 (EXTI_PR1)

Pending register 1

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
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PIF22	PIF21	PIF20	PIF19	PIF18	Res.	PIF16
									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
PIF15	PIF14	PIF13	PIF12	PIF11	PIF10	PIF9	PIF8	PIF7	PIF6	PIF5	PIF4	PIF3	PIF2	PIF1	PIF0
rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1

位 31:23 保留，必须保持复位值。

位 22:18 **PIFx**：线 x 上的挂起中断标志 (Pending interrupt flag on line x)（x = 18 到 22）
0：未发生触发请求
1：发生了选择的触发请求
当中断线上发生了选择的边沿事件时，此位被置 1。向此位写入 1 可将其清零。

位 17 保留，必须保持复位值。

位 16:0 **PIFx**：线 x 上的挂起中断标志 (Pending interrupt flag on line x)（x = 0 到 16）
0：未发生触发请求
1：发生了选择的触发请求
当中断线上发生了选择的边沿事件时，此位被置 1。向此位写入 1 可将其清零。

16.5.7 中断屏蔽寄存器 2 (EXTI_IMR2)

Interrupt mask register 2

偏移地址：0x20

复位值：0x0000 0187

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.	Res.	Res.	Res.	Res.	Res.	IM40	IM39	IM38	IM37	IM36	IM35	IM34	IM33	IM32
							rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:8 保留，必须保持复位值

位 8:0 **IMx**：线 x 上的中断屏蔽 (Interrupt mask on line x)（x = 32 到 40）
0：屏蔽来自 x 线的中断请求
1：开放来自 x 线的中断请求

注： *无需配置线（线 17、线 23 到 34、线 39、线 40）的复位值置“1”，以在默认情况下使能中断。*

16.5.8 事件屏蔽寄存器 2 (EXTI_EMR2)

Event mask register 2

偏移地址：0x24

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	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.	Res.	Res.	Res.	Res.	Res.	EM40	EM39	EM38	EM37	EM36	EM35	EM34	EM33	EM32
							rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:8 保留，必须保持复位值

位 8:0 **EMx**：线 x 上的事件屏蔽 (Event mask on line x)（x = 32 到 40）
0：屏蔽来自 x 线的事件请求
1：开放来自 x 线的事件请求

16.5.9 上升沿触发选择寄存器 2 (EXTI_RTSR2)

Rising trigger selection register 2

偏移地址：0x28

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	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.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RT38	RT37	RT36	RT35	Res.	Res.	Res.
									rw	rw	rw	rw			

位 31:8 保留，必须保持复位值。

位 7:3 **RTx**：线 x 的上升沿触发事件配置位 (Rising trigger event configuration bit of line x)（x = 35 到 38）
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）

位 2:0 保留，必须保持复位值。

注： *可配置唤醒线采用边沿触发。在这些线上不能出现毛刺信号。如果在对 EXTI_RTSR 寄存器执行写入操作时可配置的中断线上产生上升沿，挂起位不会置 1。*

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。

16.5.10 下降沿触发选择寄存器 2 (EXTI_FTSR2)

Falling trigger selection register 2

偏移地址：0x2C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	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.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FT38	FT37	FT36	FT35	Res.	Res.	Res.
									rw	rw	rw	rw			

位 31:8 保留，必须保持复位值。

位 7:3 **FTx**：线 x 的下降沿触发事件配置位 (Falling trigger event configuration bit of line x)（x = 35 到 38）

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）

位 2:0 保留，必须保持复位值。

注： *可配置唤醒线采用边沿触发。在这些线上不能出现毛刺信号。如果在对 EXTI_FTSR 寄存器执行写入操作时可配置的中断线上产生下降沿，挂起位不会置 1。*

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。

16.5.11 软件中断事件寄存器 2 (EXTI_SWIER2)

Software interrupt event 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
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.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SWI 38	SWI 37	SWI 36	SWI 35	Res.	Res.	Res.
									rw	rw	rw	rw			

位 31:8 保留，必须保持复位值。

位 7 **SWIx**：线 x 上的软件中断 (Software interrupt on line x)（x = 35 到 38）

如果在 EXTI_IMR 寄存器中允许在 x 线上产生该中断，则当此位设置为“0”时将“1”写入该位会将 EXTI_PR 中的相应挂起位置 1，进而产生中断请求。

通过清除 EXTI_PR 的对应位（对此位写入“1”），可以将该位清零。

位 2:0 保留，必须保持复位值。

16.5.12 挂起寄存器 2 (EXTI_PR2)

Pending register 2

偏移地址：0x34

复位值：未定义

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.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PIF38	PIF37	PIF36	PIF35	Res.	Res.	Res.
									rc_w1	rc_w1	rc_w1	rc_w1			

位 31:8 保留，必须保持复位值。

位 7 **PIFx**：线 x 上的挂起中断标志 (Pending interrupt flag on line x)（x = 35 到 38）

0：未发生触发请求

1：发生了选择的触发请求

当中断线上发生了选择的边沿事件时，此位被置 1。向此位写入 1 可将其清零。

位 2:0 保留，必须保持复位值。

16.5.13 EXTI 寄存器映射

表 69 给出了 EXTI 寄存器映射和复位值。

表 69. 扩展中断/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_IMR1**	IM31	IM30	IM29	IM28	IM27	IM26	IM25	IM24	IM23	IM22	IM21	IM20	IM19	IM18	IM17	IM16	IM15	IM14	IM13	IM12	IM11	IM10	IM9	IM8	IM7	IM6	IM5	IM4	IM3	IM2	IM1	IM0
	Reset value	1	1	1	1	1	1	1	1	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x04	**EXTI_EMR1**	EM31	EM30	EM29	EM28	EM27	EM26	EM25	EM24	EM23	EM22	EM21	EM20	EM19	EM18	EM17	EM16	EM15	EM14	EM13	EM12	EM11	EM10	EM9	EM8	EM7	EM6	EM5	EM4	EM3	EM2	EM1	EM0
	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	**EXTI_RTSR1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RT22	RT21	RT20	RT19	RT18	Res.	RT16	RT15	RT14	RT13	RT12	RT11	RT10	RT9	RT8	RT7	RT6	RT5	RT4	RT3	RT2	RT1	RT0
	Reset value										0	0	0	0	0		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0C	**EXTI_FTSR1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FT22	FT21	FT20	FT19	FT18	Res.	FT16	FT15	FT14	FT13	FT12	FT11	FT10	FT9	FT8	FT7	FT6	FT5	FT4	FT3	FT2	FT1	FT0
	Reset value										0	0	0	0	0		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x10	**EXTI_SWIER1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SWI22	SWI21	SWI20	SWI19	SWI18	Res.	SWI16	SWI15	SWI14	SWI13	SWI12	SWI11	SWI10	SWI9	SWI8	SWI7	SWI6	SWI5	SWI4	SWI3	SWI2	SWI1	SWI0
	Reset value										0	0	0	0	0		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x14	**EXTI_PR1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PIF22	PIF21	PIF20	PIF19	PIF18	Res.	PIF16	PIF15	PIF14	PIF13	PIF12	PIF11	PIF10	PIF9	PIF8	PIF7	PIF6	PIF5	PIF4	PIF3	PIF2	PIF1	PIF0
	Reset value										0	0	0	0	0		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x20	**EXTI_IMR2**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	IM40	IM39	IM38	IM37	IM36	IM35	IM34	IM33	IM32
	Reset value																								1	1	0	0	0	0	1	1	1
0x24	**EXTI_EMR2**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	EM40	EM39	EM38	EM37	EM36	EM35	EM34	EM33	EM32
	Reset value																								0	0	0	0	0	0	0	0	0
0x28	**EXTI_RTSR2**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RT38	RT37	RT36	RT35	Res.	Res.	Res.
	Reset value																										0	0	0	0			
0x2C	**EXTI_FTSR2**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FT38	FT37	FT36	FT35	Res.	Res.	Res.
	Reset value																										0	0	0	0			
0x30	**EXTI_SWIER2**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SWI38	SWI37	SWI36	SWI35	Res.	Res.	Res.
	Reset value																										0	0	0	0			
0x34	**EXTI_PR2**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PIF38	PIF37	PIF36	PIF35	Res.	Res.	Res.
	Reset value																										0	0	0	0			

17 循环冗余校验计算单元 (CRC)

17.1 简介

CRC（循环冗余校验）计算单元使用多项式发生器从一个 8 位/16 位/32 位的数据字中产生 CRC 码。

在众多的应用中，基于 CRC 的技术还常用来验证数据传输或存储的完整性。根据功能安全标准的规定，这些技术提供了验证 Flash 完整性的方法。CRC 计算单元有助于在运行期间计算软件的签名，并将该签名与链接时生成并存储在指定存储单元的参考签名加以比较。

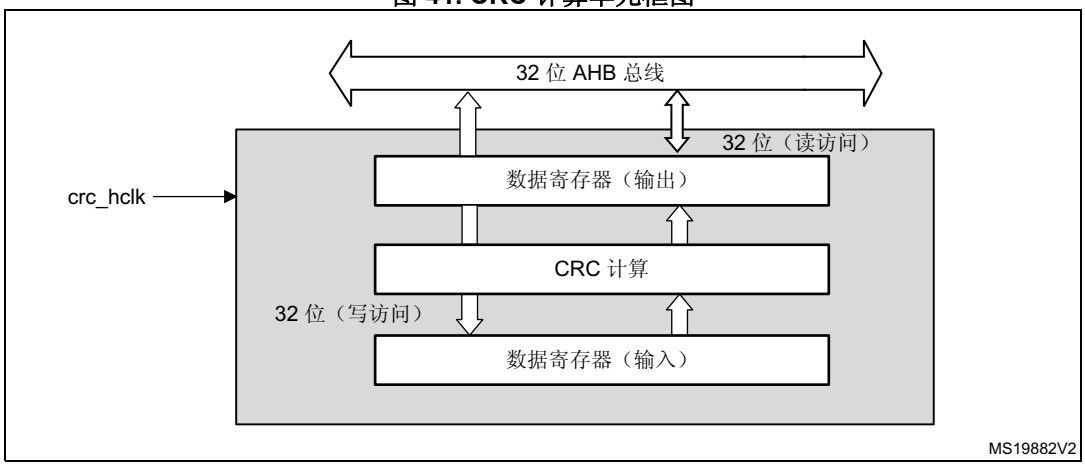
17.2 CRC 主要特性

- 使用 CRC-32（以太网）多项式：0x4C11DB7
  $X^{32} + X^{26} + X^{23} + X^{22} + X^{16} + X^{12} + X^{11} + X^{10} + X^8 + X^7 + X^5 + X^4 + X^2 + X + 1$
- 或者，使用位数可编程的（7 位、8 位、16 位和 32 位）的完全可编程多项式
- 处理 8 位、16 位、32 位数据大小
- 可编程 CRC 初始值
- 单输入/输出 32 位数据寄存器
- 输入缓冲器可避免计算期间发生总线阻塞
- 对于 32 位数据大小，CRC 计算在 4 个 AHB 时钟周期 (HCLK) 内完成
- 8 位通用寄存器（可用于临时存储）
- I/O 数据的可逆性选项

17.3 CRC 功能说明

17.3.1 CRC 框图

图 41. CRC 计算单元框图



17.3.2 CRC 内部信号

表 70. CRC 内部输入/输出信号

信号名称	信号类型	说明
crc_hclk	数字输入	AHB 时钟

17.3.3 CRC 操作

CRC 计算单元具有单个 32 位读/写数据寄存器 (CRC_DR)。它用于输入新数据（写访问）和保存之前 CRC 计算的结果（读访问）。

对数据寄存器的每个写操作都会对之前的 CRC 值（存储在 CRC_DR 中）和新值再做一次 CRC 计算。CRC 计算针对整个 32 位数据字或逐个字节完成，具体取决于数据的写入格式。

CRC_DR 寄存器可按字、右对齐半字和右对齐字节进行访问。对于其他寄存器，只允许进行 32 位访问。

计算时间取决于数据宽度：

- 32 位数据需要 4 个 AHB 时钟周期
- 16 位数据需要 2 个 AHB 时钟周期
- 8 位数据需要 1 个 AHB 时钟周期

输入缓冲器中可立即写入第二个数据，无需因之前的 CRC 计算而等待任何等待状态。

可动态调整数据大小，从而能最大程度地减少给定字节数的写访问次数。例如，对 5 个字节进行 CRC 计算时，可先写入字，然后写入字节。

输入数据的顺序可反转，以管理各种数据存放方式（双字/单字/字节，大端/小端等等）。可对 8 位、16 位和 32 位数据执行反转操作，具体取决于 CRC_CR 寄存器中的 REV_IN[1:0] 位。

例如，输入数据 0x1A2B3C4D 在 CRC 计算中用作：

按字节执行位反转的 0x58D43CB2

按半字执行位反转的 0xD458B23C

按全字执行位反转的 0xB23CD458

通过将 CRC_CR 寄存器中 REV_OUT 位置 1 也可以将输出数据反转。

该操作按位进行：例如，输出数据 0x11223344 将转换为 0x22CC4488。

使用 CRC_CR 寄存器中的 RESET 控制位可将 CRC 计算器初始化为可编程值（默认值为 0xFFFFFFFF）。

可使用 CRC_INIT 寄存器对 CRC 初始值进行编程。对 CRC_INIT 寄存器进行写访问时会自动初始化 CRC_DR 寄存器。

CRC_IDR 寄存器可用于保存与 CRC 计算相关的临时值。它不受 CRC_CR 寄存器中的 RESET 位影响。

多项式可编程

多项式系数可完全通过 CRC_POL 寄存器进行编程，通过编程 CRC_CR 寄存器中的 POLYSIZE[1:0] 位可将多项式大小配置为 7 位、8 位、16 位或 32 位。不支持偶数多项式。

如果 CRC 数据小于 32 位，可从 CRC_DR 寄存器的最低有效位读取 CRC 的值。

为实现可靠的 CRC 计算，CRC 计算期间不能实时更改多项式的值或大小。因此，如果正在执行 CRC 计算，则在更改多项式前，应用程序必须先复位，或者先执行 CRC_DR 读操作。

默认的多项式值为 CRC-32（以太网）多项式：0x4C11DB7。

17.4 CRC 寄存器

17.4.1 数据寄存器 (CRC_DR)

Data register

偏移地址：0x00

复位值：0xFFFF FFFF

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

位 31:0 **DR[31:0]**：数据寄存器位 (Data register bits)

该寄存器用于向 CRC 计算器写入新数据。

读取寄存器时可读出之前的 CRC 计算结果。

如果数据大小小于 32 位，则最低有效位可用于写入/读取正确值。

17.4.2 独立数据寄存器 (CRC_IDR)

Independent data 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
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.	Res.	Res.	Res.	Res.	Res.	Res.	IDR[7:0]							
								rw							

位 31:8 保留，必须保持复位值。

位 7:0 **IDR[7:0]**：通用的 8 位数据寄存器位 (General-purpose 8-bit data register bits)
这些位可用作一个字节的临时存储单元。
此寄存器不受 CRC_CR 寄存器中 RESET 位产生的 CRC 复位影响。

17.4.3 控制寄存器 (CRC_CR)

Control 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
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.	Res.	Res.	Res.	Res.	Res.	Res.	REV_OUT	REV_IN[1:0]		POLYSIZE[1:0]		Res.	Res.	RESET
								rw	rw	rw	rw	rw			rs

位 31:8 保留，必须保持复位值。

位 7 **REV_OUT**：反转输出数据 (Reverse output data)
该位用于控制输出数据位顺序的反转。
0：不影响位顺序
1：位反转输出格式

位 6:5 **REV_IN[1:0]**：反转输入数据 (Reverse input data)
这些位用于控制输入数据位顺序的反转。
00：不影响位顺序
01：按字节执行位反转
10：按半字执行位反转
11：按字执行位反转

位 4:3 **POLYSIZE[1:0]**：多项式大小 (Polynomial size)
这些位用于控制多项式的大小。
00：32 位多项式
01：16 位多项式
10：8 位多项式
11：7 位多项式

位 2:1 保留，必须保持复位值。

位 0 **RESET**：RESET 位
此位由软件置 1，用于复位 CRC 计算单元并将数据寄存器设置为存储在 CRC_INIT 寄存器中的值。此位只能置 1，将由硬件自动进行清零。

17.4.4 CRC 初始值 (CRC_INIT)

Initial CRC value

偏移地址：0x10

复位值：0xFFFF FFFF

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

位 31:0 **CRC_INIT[31:0]**：*可编程 CRC 初始值 (Programmable initial CRC value)*
此寄存器用于写入 CRC 初始值。

17.4.5 CRC 多项式 (CRC_POL)

CRC polynomial

偏移地址：0x14

复位值：0x04C1 1DB7

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

位 31:0 **POL[31:0]**：可编程多项式 (Programmable polynomial)
此寄存器用于写入要用于 CRC 计算的多项式系数。
如果多项式大小小于 32 位，则必须使用最低有效位编程正确值。

17.4.6 CRC 寄存器映射

表 71. CRC 寄存器映射和复位值

偏移	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	**CRC_DR**	DR[31:0]																															
	Reset value	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0x04	**CRC_IDR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	IDR[7:0]							
	Reset value																									0	0	0	0	0	0	0	0
0x08	**CRC_CR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REV_OUT	REV_IN[1:0]		POLYSIZE[1:0]		Res.	Res.	RESET
	Reset value																									0	0	0	0	0			0
0x10	**CRC_INIT**	CRC_INIT[31:0]																															
	Reset value	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0x14	**CRC_POL**	Polynomial coefficients																															
	Reset value	0x04C11DB7																															

有关寄存器边界地址的信息，请参见*第 78 页的第 2.2.2 节*。

18 灵活的静态存储控制器 (FSMC)

灵活的静态存储控制器 (FSMC) 包括以下两个存储控制器：

- NOR/PSRAM 存储控制器
- NAND 存储控制器

18.1 FSMC 主要特性

FSMC 功能块可连接：同步/异步静态存储器和 NAND Flash。其主要用途有：

- 将 AHB 数据通信事务转换为适当的外部器件协议
- 满足外部存储器器件的访问时间要求

所有外部存储器共享地址、数据和控制信号，但有各自的片选信号。FSMC 一次只能访问一个外部器件。

FSMC 控制器的主要特性如下：

- 连接静态存储器映射的器件：
  - 静态随机访问存储器 (SRAM)
  - NOR Flash/OneNAND Flash
  - PSRAM（4 个存储区域）
  - 铁电 RAM (FRAM)
  - 带有硬件 ECC 的 NAND Flash 存储器，可检查多达 8 KB 的数据
- 可与并行 LCD 模块连接，支持 Intel 8080 和 Motorola 6800 模式
- 支持突发模式，能够更快速地访问同步器件（如 NOR Flash 和 PSRAM）
- 可编程连续时钟输出以支持异步和同步访问
- 具有 8 位、16 位或 32 位宽的数据总线
- 每个存储区域有独立的片选控制
- 每个存储区域可独立配置
- 写使能和字节通道选择输出，可配合 PSRAM 和 SRAM 器件使用
- 外部异步等待控制
- 16 x 32 位深度写 FIFO

写 FIFO 由所有存储控制器所共用，包括：

- 写数据 FIFO，用于存储要写入存储器的 AHB 数据（最多 32 位）以及 AHB 传输的一个控制位（突发或非连续模式）。
- 写地址 FIFO，用于存储 AHB 地址（最多 28 位）以及 AHB 数据大小（最多 2 位）。在突发模式下工作时，将仅存储起始地址，但越过页边界时除外（适用于 PSRAM）。在此情况下，AHB 突发传输将分成两个 FIFO 条目。

通过将 FSMC_BCR1 寄存器中的 WFDIS 位置 1 可禁止写 FIFO。

启动时，必须通过用户应用程序对 FSMC 引脚进行配置。应用程序未使用的 FSMC I/O 引脚可用于其他用途。

定义外部器件类型和其特性的 FSMC 寄存器通常在启动时进行设置，并且在下次上电或复位前保持不变。但是，可随时更改设置。

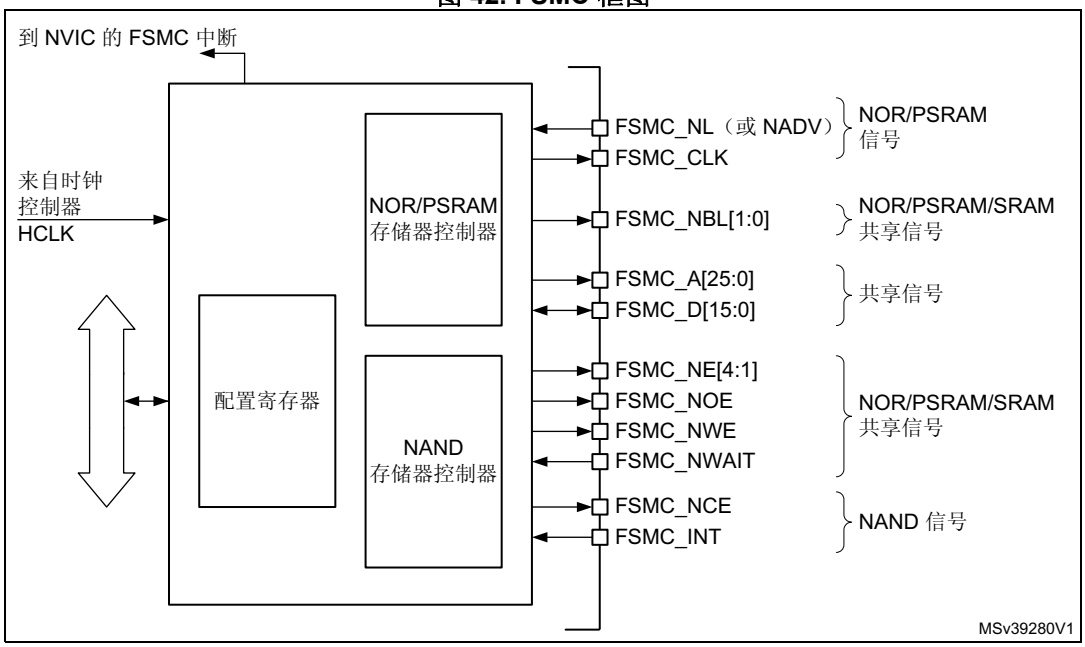
18.2 FMC 框图

FSMC 包含以下主要模块：

- AHB 接口（包括 FSMC 配置寄存器）
- NOR Flash/PSRAM/SRAM 控制器
- 外部器件接口
- NAND Flash 控制器

框图如下图所示。

图 42. FSMC 框图



18.3 AHB 接口

CPU 和其他 AHB 总线主设备可通过该 AHB 从设备接口访问外部存储器。

AHB 事务会转换为外部器件协议。尤其是当所选外部存储器的宽度为 16 位或 8 位时，AHB 中的 32 位宽事务将被划分成多个连续的 16 或 8 位访问。除非在使能扩展模式时在 D 模式下进行访问，否则 FSMC 片选 (FSMC_NEx) 在连续两次访问之间不会翻转。

出现以下条件时，FSMC 将产生 AHB 错误：

- 读取或写入未使能的 FSMC 存储区域。
- 在 FSMC_BCRx 寄存器中的 FACCEN 位复位时读取或写入 NOR Flash 存储区域。

此 AHB 错误的影响具体取决于尝试进行读写访问的 AHB 主设备：

- 如果为 Cortex®-M4 CPU，则会生成硬性故障中断。
- 如果为 DMA 控制器，则会生成 DMA 传输错误，并会自动禁止相应的 DMA 通道。

AHB 时钟 (HCLK) 是 FSMC 的参考时钟。

18.3.1 支持的存储器和事务

通用事务规则

所请求的 AHB 事务数据宽度可以是 8、16 或 32 位，但访问的外部器件具有固定的数据宽度。这可能会导致传输不一致。

因此，必须遵循一些简单的事务规则：

- AHB 事务数据宽度和存储器数据宽度相等

  在此情况下没有任何问题。
- AHB 事务数据宽度大于存储器宽度：

  在此情况下，FSMC 会将 AHB 事务分为多个较小的连续存储器访问，以符合外部数据宽度。连续访问之间，FSMC 片选 (FSMC_NEx) 不会翻转。如果将总线周转时间配置为除 0 外的任何其他值，则 FMC 片选 (FMC_NEx) 会在连续访问之间翻转。与 FRAM 存储器相连时需要此功能。
- AHB 事务数据宽度小于存储器宽度：

  传送可能一致，也可能不一致，具体取决于外部器件的类型：
  - 对具有字节选择功能的器件（SRAM、ROM、PSRAM）进行访问

    在这种情况下，FSMC 允许读/写事务并通过其字节选择通道 NBL[1:0] 访问恰当的数据。

    通过 NBL[1:0] 寻址要写入的字节。

    读取所有存储器字节（NBL[1:0] 在读取事务期间保持为低电平），并丢弃无用的字节。
  - 访问不具有字节选择功能的器件（NOR 和 NAND Flash）

    当请求对 16 位宽的 Flash 存储器进行字节访问时会发生此情形。由于无法在字节模式下访问器件（只能向 Flash 读取或写入 16 位字），因此允许读事务和写事务（控制器会读取全部 16 位存储器字，但只使用所需字节）。

NOR Flash/PSRAM 的回卷支持

不支持同步存储器的回绕突发模式。存储器必须按未定义长度的线性突发模式进行配置。

配置寄存器

可通过一组寄存器配置 FSMC。有关 NOR Flash/PSRAM 控制寄存器的详细说明，请参见*第 18.5.6 节*。有关 NAND Flash 寄存器的详细说明，请参见*第 18.6.7 节*。

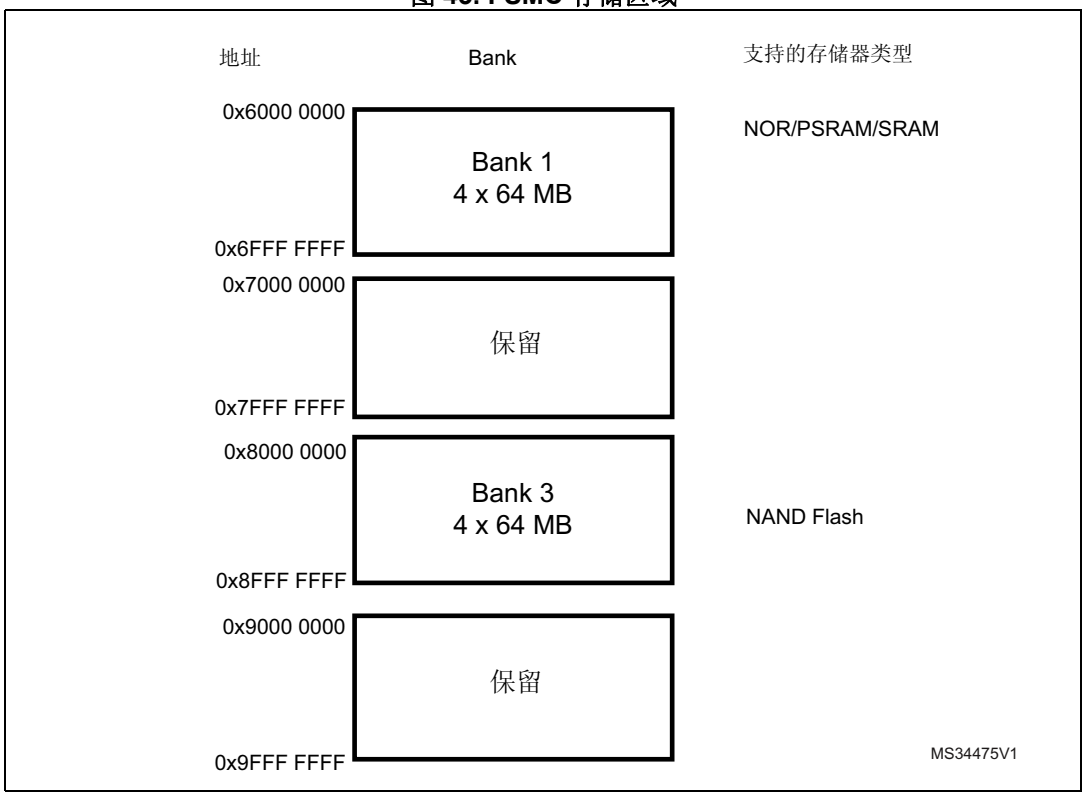
18.4 外部器件地址映射

从 FSMC 的角度，外部存储器被划分为固定大小的存储区域，每个存储区域的大小为 256 MB（请参见*图 43*）：

- 存储区域 1 可连接多达 4 个 NOR Flash 或 PSRAM 器件。此存储区域被划分为如下 4 个 NOR/PSRAM 子区域，带 4 个专用片选信号：
  - 存储区域 1 - NOR/PSRAM 1
  - 存储区域 1 - NOR/PSRAM 2
  - 存储区域 1 - NOR/PSRAM 3
  - 存储区域 1 - NOR/PSRAM 4
- 存储区域 3 用于连接 NAND Flash 器件。此空间的 MPU 存储器特性必须通过软件重新配置到器件中。

对于每个存储区域，所要使用的存储器类型可由用户应用程序通过配置寄存器配置。

图 43. FSMC 存储区域



18.4.1 NOR/PSRAM 地址映射

HADDR[27:26] 位用于从*表 72* 中所示的四个存储区域之中选择其中一个存储区域。

表 72. NOR/PSRAM 存储区域选择

HADDR[27:26](1)	选择的存储区域
00	存储区域 1 - NOR/PSRAM 1
01	存储区域 1 - NOR/PSRAM 2
10	存储区域 1 - NOR/PSRAM 3
11	存储区域 1 - NOR/PSRAM 4

1. HADDR 是 AHB 内部地址线，但也会参与对外部存储器的寻址。

HADDR[25:0] 位包含外部存储器地址。由于 HADDR 为字节地址，而存储器按字寻址，所以根据存储器数据宽度不同，实际向存储器发送的地址也将有所不同，如下表所示。

表 73. NOR/PSRAM 外部存储器地址

存储器宽度(1)	向存储器发出的数据地址	最大存储器容量（位）
8 位	HADDR[25:0]	64 MB x 8 = 512 Mb
16 位	HADDR[25:1] >> 1	64 MB/2 x 16 = 512 Mb

1. 如果外部存储器的宽度为 16 位，FSMC 将使用内部的 HADDR[25:1] 地址来作为对外部存储器的寻址地址 FSMC_A[24:0]。
   无论外部存储器的宽度是多少，FSMC_A[0] 都应连接到外部存储器地址 A[0]。

18.4.2 NAND Flash 地址映射

NAND 存储区域分为几个存储器区域，如*表 74* 所示。

表 74. NAND 存储映射和时序寄存器

起始地址	结束地址	FSMC 存储区域	存储空间	时序寄存器
0x8800 0000	0x8BFF FFFF	存储区域 3 - NAND Flash	特性区	FSMC_PATT (0x8C)
0x8000 0000	0x83FF FFFF		通用区	FSMC_PMEM (0x88)

对于 NAND Flash 存储器，通用区和特性区存储空间分为三个部分，均位于低位 256 KB 中（见下面的*表 75*）：

- 数据区域（通用/特性存储空间中的第一个 64 KB）
- 命令区域（通用/特性存储空间中的第二个 64 KB）
- 地址区域（通用/特性存储空间中的下一个 128 KB）

表 75. NAND 存储区域选择

部分名称	HADDR[17:16]	地址范围
地址区域	1X	0x020000-0x03FFFF
命令区域	01	0x010000-0x01FFFF
数据区域	00	0x000000-0x0FFFF

应用程序软件使用这 3 个区域来访问 NAND Flash 存储器：

- **向 NAND Flash 发送命令**，软件必须向命令区域中的任意存储单元写入命令值。
- **指定读取或写入的 NAND Flash 地址**，软件必须向地址区域中的任意存储单元写入地址值。由于地址的长度可以是 4 或 5 个字节（具体取决于实际存储器大小），要指定完整的地址，需要对地址区域执行多个连续写入操作。
- **读取或写入数据**，软件将从数据区域中的任意存储单元读取数据，或者向其中写入数据。

由于 NAND Flash 存储器会自动递增地址，所以在访问连续存储器位置时，无需递增数据区域的地址。

18.5 NOR Flash/PSRAM 控制器

FSMC 会生成适当的信号时序，以驱动以下类型的存储器：

- 异步 SRAM、FRAM 和 ROM
  - 8 位
  - 16 位
- PSRAM (CellularRAM™)
  - 异步模式
  - 同步访问的突发模式
  - 复用或非复用
- NOR Flash
  - 异步模式
  - 同步访问的突发模式
  - 复用或非复用

FSMC 会为每个存储区域输出唯一的片选信号 NE[4:1]。所有其他信号（地址、数据和控制）均为共享信号。

FSMC 通过可编程时序支持多种器件，其中：

- 等待周期可编程（最多 15 个时钟周期）
- 总线周转周期可编程（最多 15 个时钟周期）
- 输出使能和写入使能延迟可编程（最多 15 个时钟周期）
- 独立的读和写时序和协议，以支持各种存储器和时序
- 可编程连续时钟 (FSMC_CLK) 输出

FSMC 时钟 (FSMC_CLK) 是 HCLK 时钟的约数。该时钟可传送到选定的外部器件，根据 FSMC_BCR1 寄存器中 CCKEN 位的配置，可决定只在同步访问期间传送还是同步异步访问期间都传送：

- 如果 CCLKEN 位复位，则 FSMC 仅在同步访问（读/写事务）期间生成时钟 (CLK)。
- 如果 CCLKEN 位置 1，则 FSMC 将在异步和同步访问期间生成连续时钟。要生成 FSMC_CLK 连续时钟，存储区域 1 还必须配置为支持同步模式（请参见*第 18.5.6 节：NOR/PSRAM 控制器寄存器*）。由于所有同步存储器均使用同一个时钟，因此在生成连续输出时钟和执行同步访问时，AHB 数据大小必须与存储器数据宽度 (MWID) 相同，否则 FSMC_CLK 频率将根据 AHB 数据事务发生变化（有关 FSMC_CLK 分频比公式的信息，请参见*第 18.5.5 节：同步事务*）。

每个存储区域的大小固定，均为 64 MB。每个存储区域都通过专用的寄存器配置（请参见*第 18.5.6 节：NOR/PSRAM 控制器寄存器*）。

存储器的可编程参数包括访问时间（请参见*表 76*）和对等待管理的支持（用于在突发模式下访问 NOR Flash 和 PSRAM）。

表 76. NOR/PSRAM 的可编程访问参数

参数	功能	访问模式	单位	最小值	最大值
地址建立	地址建立阶段的持续时间	异步	AHB 时钟周期 (HCLK)	0	15
地址保持	地址保持阶段的持续时间	异步，复用 I/O	AHB 时钟周期 (HCLK)	1	15
NBL 建立	字节通道建立阶段的持续时间	异步	AHB 时钟周期 (HCLK)	0	3
数据建立	数据建立阶段的持续时间	异步	AHB 时钟周期 (HCLK)	1	256
数据保持	数据保持阶段的持续时间	异步	AHB 时钟周期 (HCLK)	0	3
总线周转	总线周转阶段的持续时间	异步和同步读/写	AHB 时钟周期 (HCLK)	0	15
时钟分频比	构建一个存储器时钟周期 (CLK) 所需的 AHB 时钟周期 (HCLK) 数量	同步	AHB 时钟周期 (HCLK)	2	16
数据延迟	在发出突发的第一个数据前向存储器发出的时钟周期数量	同步	存储器时钟周期 (CLK)	2	17

18.5.1 外部存储器接口信号

表 77、*表 78* 和 *表 79* 列出了通常用于连接 NOR Flash、SRAM 和 PSRAM 的信号。

注： *前缀“N”标识低电平有效的信号。*

NOR Flash，非复用 I/O

表 77. 非复用 I/O NOR Flash

FSMC 信号名称	I/O	功能
CLK	O	时钟（用于同步访问）
A[25:0]	O	地址总线
D[15:0]	I/O	双向数据总线
NE[x]	O	片选，x = 1..4
NOE	O	输出使能
NWE	O	写入使能
NL(= NADV)	O	锁存使能（对于部分 NOR Flash 器件，此信号也称为地址有效 (NADV)）
NWAIT	I	FSMC 的 NOR Flash 等待输入信号

最大容量为 512 Mb（26 个地址线）。

NOR Flash，16 位复用 I/O

表 78. 16 位复用 I/O NOR Flash

FSMC 信号名称	I/O	功能
CLK	O	时钟（用于同步访问）
A[25:16]	O	地址总线
AD[15:0]	I/O	16 位复用，双向地址/数据总线 （16 位地址 A[15:0] 和数据 D[15:0] 在数据总线上复用）
NE[x]	O	片选，x = 1..4
NOE	O	输出使能
NWE	O	写入使能
NL(= NADV)	O	锁存使能 （对于部分 NOR Flash 器件，此信号也称为地址有效 (NADV)）
NWAIT	I	FSMC 的 NOR Flash 等待输入信号

最大容量为 512 Mb。

PSRAM/FRAM/SRAM，非复用 I/O

表 79. 非复用 I/O PSRAM/SRAM

FSMC 信号名称	I/O	功能
CLK	O	时钟（仅用于 PSRAM 同步访问）
A[25:0]	O	地址总线
D[15:0]	I/O	数据双向总线
NE[x]	O	片选，x = 1..4 （在 PSRAM 应用中被称作 NCE（CellularRAM™，即 CRAM））
NOE	O	输出使能
NWE	O	写入使能
NL(= NADV)	O	仅用于 PSRAM 输入的地址有效信号（存储器信号名称：NADV）
NWAIT	I	FSMC 的 PSRAM 等待输入信号
NBL[1:0]	O	字节通道输出。字节 0 和字节 1 控制（高字节和低字节使能）

最大容量为 512 Mb。

PSRAM，16 位复用 I/O

表 80. 16 位复用 I/O PSRAM

FSMC 信号名称	I/O	功能
CLK	O	时钟（用于同步访问）
A[25:16]	O	地址总线
AD[15:0]	I/O	16 位复用，双向地址/数据总线 （16 位地址 A[15:0] 和数据 D[15:0] 在数据总线上复用）
NE[x]	O	片选，x = 1..4 （在 PSRAM 应用中被称作 NCE（CellularRAM™，即 CRAM））
NOE	O	输出使能
NWE	O	写入使能
NL(= NADV)	O	用于 PSRAM 输入的地址有效信号（存储器信号名称：NADV）
NWAIT	I	FSMC 的 PSRAM 等待输入信号
NBL[1:0]	O	字节通道输出。字节 0 和字节 1 控制（高字节和低字节使能）

最大容量为 512 Mb（26 个地址线）。

18.5.2 支持的存储器和事务

下面的*表 81* 显示的是当 NOR Flash、PSRAM 和 SRAM 的存储器数据总线宽度为 16 位时所支持的器件、访问模式和事务的示例。本示例中 FSMC 不允许（或不支持）的事务以灰色显示。

表 81. NOR Flash/PSRAM：支持的存储器和事务示例

器件	模式	R/W	AHB 数据大小	存储器数据大小	是否允许	注释
NOR Flash（复用 I/O 和非复用 I/O）	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
	异步页	R	-	16	否	不支持该模式
	同步	R	8	16	否	-
	同步	R	16	16	是	-
	同步	R	32	16	是	-
PSRAM（复用 I/O 和非复用 I/O）	异步	R	8	16	是	-
	异步	W	8	16	是	使用字节通道 NBL[1:0]
	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
	异步页	R	-	16	否	不支持该模式
	同步	R	8	16	否	-
	同步	R	16	16	是	-
	同步	R	32	16	是	-
	同步	W	8	16	是	使用字节通道 NBL[1:0]
	同步	W	16/32	16	是	-
SRAM 和 ROM	异步	R	8 / 16	16	是	-
	异步	W	8 / 16	16	是	使用字节通道 NBL[1:0]
	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，使用字节通道 NBL[1:0]

18.5.3 通用时序规则

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
- 在同步模式（读取或写入）下，所有输出信号在 HCLK 的上升沿变化。无论 CLKDIV 值为何，所有输出均会按以下方式变化：
  - NOEL/NWEL/NEL/NADVL/NADVH/NBLL/ 地址有效输出在 FSMC_CLK 时钟的下降沿变化。
  - NOEH/NWEH/NEH/NOEH/NBLH/ 地址有效输出在 FSMC_CLK 时钟的上升沿变化。

18.5.4 NOR Flash/PSRAM 控制器异步事务

异步静态存储器（NOR Flash、PSRAM、SRAM、FRAM）

- 信号通过内部时钟 HCLK 进行同步。此时钟不会发送到存储器。
- FSMC 始终会在 NOE 禁止前进行数据采样，而后再禁止 NOE 信号。这样可以确保符合存储器数据保持时序的要求（数据转换的最小芯片使能高电平通常为 0 ns）。
- 如果使能扩展模式（FSMC_BCRx 寄存器中的 EXTMOD 位置 1），则最多可提供四种扩展模式（A、B、C 和 D）。可以混合使用 A、B、C 和 D 模式来进行读取和写入操作。例如，可以在模式 A 下执行读取操作，而在模式 B 下执行写入操作。
- 如果禁用扩展模式（FSMC_BCRx 寄存器中的 EXTMOD 位复位），则 FSMC 可以在模式 1 或模式 2 下运行，如下所述：
  - 当选择 SRAM/PSRAM 存储器类型时，模式 1 为默认模式（FSMC_BCRx 寄存器中 MTYP = 0x0 或 0x01）。
  - 当选择 NOR 存储器类型时，模式 2 为默认模式（FSMC_BCRx 寄存器中 MTYP = 0x10）。

模式 1 - SRAM/FRAM/PSRAM (CRAM)

下图显示了所支持模式的读取和写入事务以及所需的 FSMC _BCRx 和 FSMC_BTRx/FSMC_BWTRx 寄存器配置。

图 44. 模式 1 读取访问波形

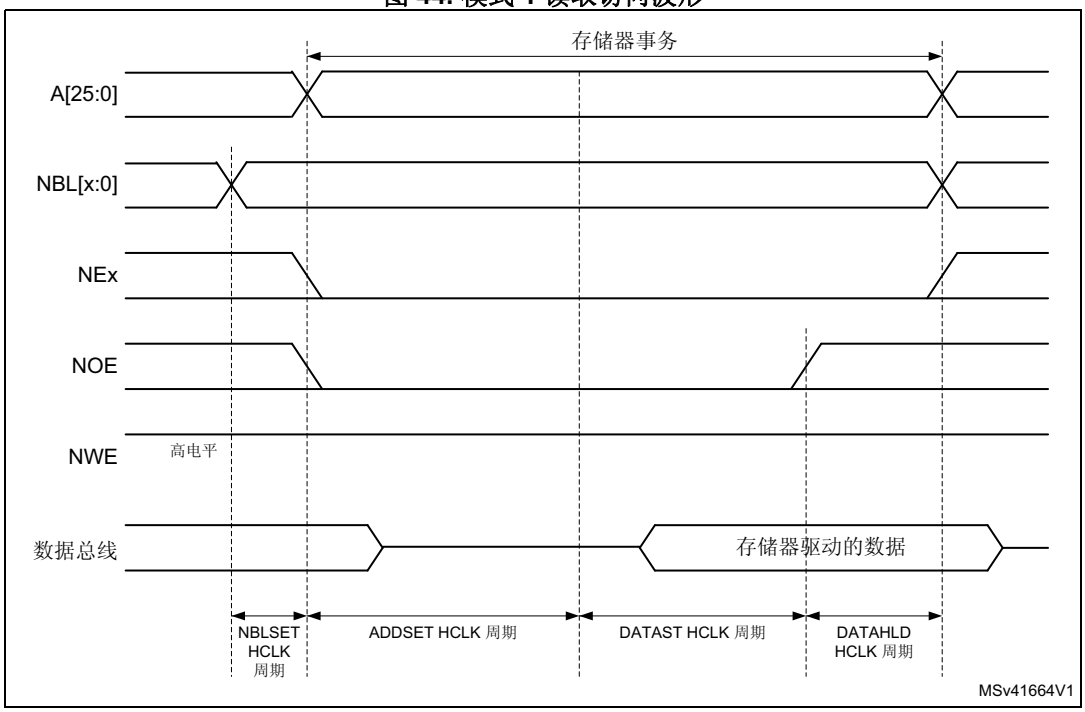
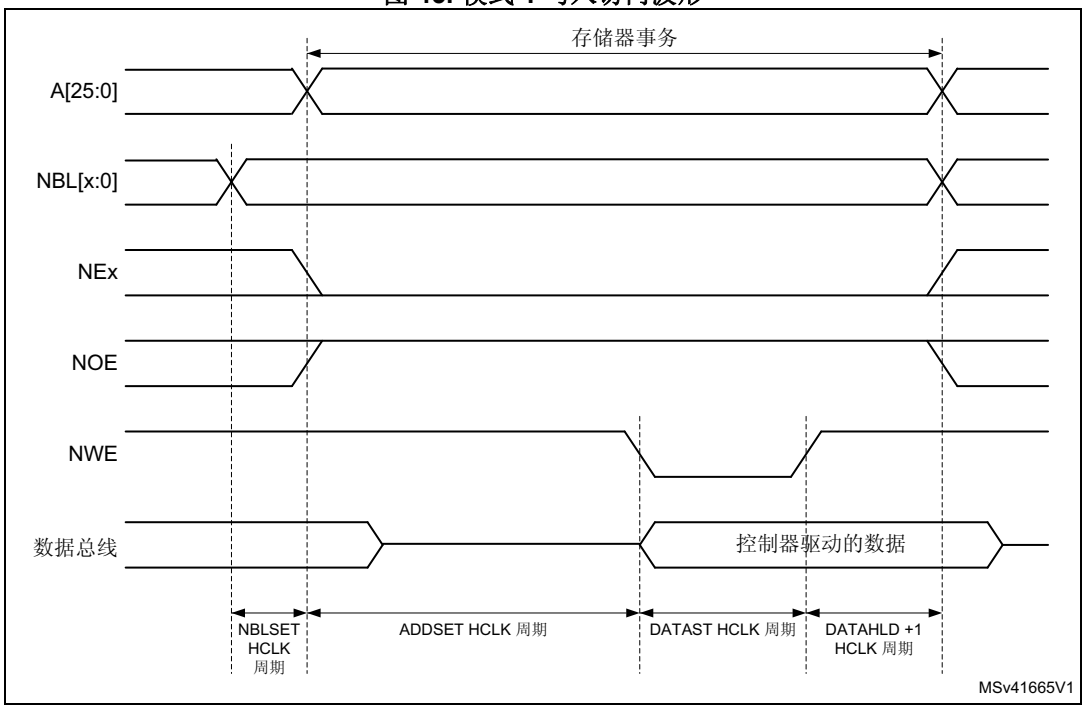


图 45. 模式 1 写入访问波形



位于读写事务末尾的 DATAHLD 时间有助于确保 NOE/NWE 上升沿之后的地址和数据保持时间。DATAST 值必须大于零 (DATAST > 0)。

表 82. FSMC_BCRx 位域

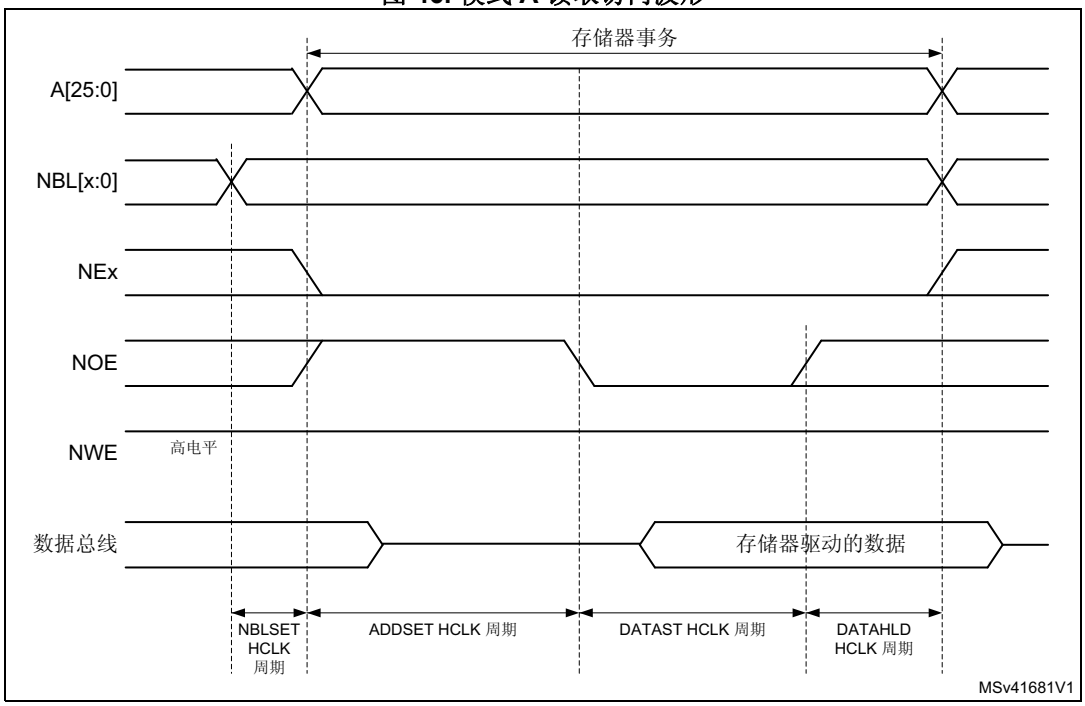
位号	位名	要设置的值
31:24	保留	0x000
23:22	NBLSET[1:0]	根据需要进行设置
21	WFDIS	根据需要进行设置
20	CCLKEN	根据需要进行设置
19	CBURSTRW	0x0（对异步模式没有影响）
18:16	CPSIZE	0x0（对异步模式没有影响）
15	ASYNCWAIT	如果存储器支持该特性，则置为 1。否则，保持为 0
14	EXTMOD	0x0
13	WAITEN	0x0（对异步模式没有影响）
12	WREN	根据需要进行设置
11	保留	0x0
10	保留	0x0
9	WAITPOL	仅当位 15 为 1 时才有意义
8	BURSTEN	0x0
7	保留	0x1
6	FACCEN	无关
5:4	MWID	根据需要进行设置
3:2	MTYP	根据需要进行设置，0x2 除外 (NOR Flash)
1	MUXE	0x0
0	MBKEN	0x1

表 83. FSMC_BTRx 位域

位号	位名	要设置的值
31:30	DATAHLD	数据保持阶段的持续时间（读取访问为 DATAHLD 个 HCLK 周期，写入访问为 DATAHLD+1 个 HCLK 周期）。
29:28	ACCMOD	无关
27:24	DATLAT	无关
23:20	CLKDIV	无关
19:16	BUSTURN	NEx 变为高电平到 NEx 变为低电平之间的时间 (BUSTURN HCLK)。
15:8	DATAST	第二个访问阶段的持续时间（DATAST 个 HCLK 周期）。
7:4	ADDHLD	无关
3:0	ADDSET	第一个访问阶段的持续时间（ADDSET 个 HCLK 周期）。 ADDSET 的最小值为 0。

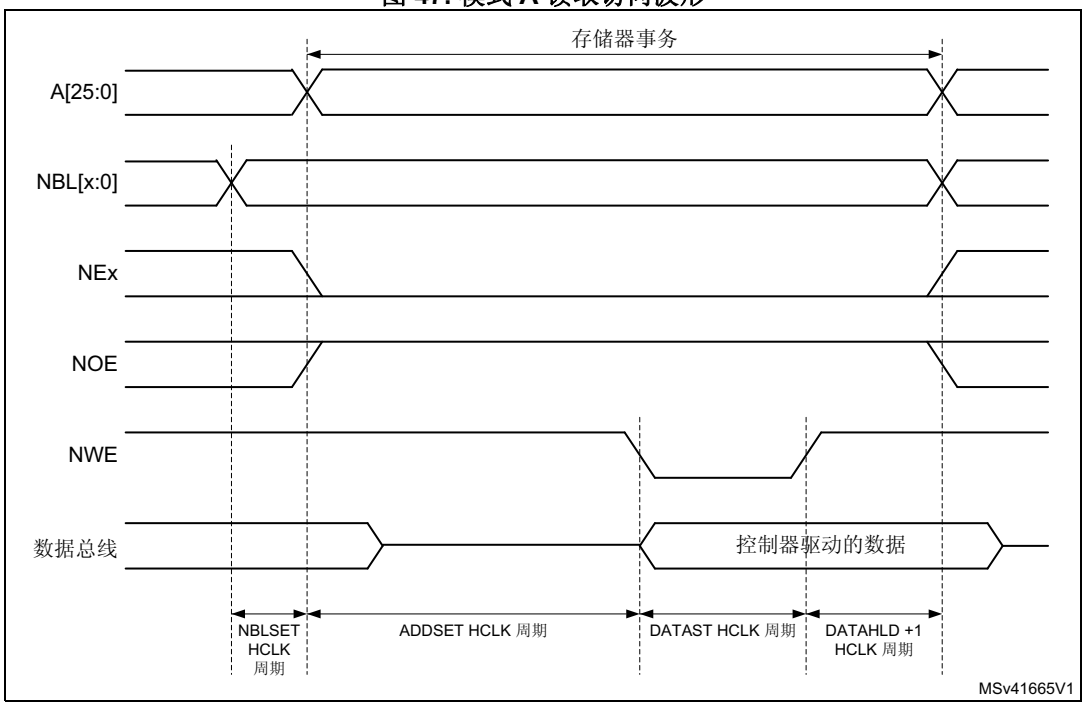
模式 A - SRAM/FRAM/PSRAM (CRAM) OE 切换

图 46. 模式 A 读取访问波形



1. NBL[1:0] 在进行读取访问时为低电平

图 47. 模式 A 读取访问波形



与模式 1 的不同之处在于 NOE 的切换与独立的读取和写入时序。

表 84. FSMC_BCRx 位域

位号	位名	要设置的值
31:24	保留	0x000
23:22	NBLSET[1:0]	根据需要进行设置
21	WFDIS	根据需要进行设置
20	CCLKEN	根据需要进行设置
19	CBURSTRW	0x0（对异步模式没有影响）
18:16	CPSIZE	0x0（对异步模式没有影响）
15	ASYNCWAIT	如果存储器支持该特性，则置为 1。否则，保持为 0
14	EXTMOD	0x1
13	WAITEN	0x0（对异步模式没有影响）
12	WREN	根据需要进行设置
11	WAITCFG	无关
10	保留	0x0
9	WAITPOL	仅当位 15 为 1 时才有意义
8	BURSTEN	0x0
7	保留	0x1
6	FACCEN	无关
5:4	MWID	根据需要进行设置
3:2	MTYP	根据需要进行设置，0x2 除外 (NOR Flash)
1	MUXEN	0x0
0	MBKEN	0x1

表 85. FSMC_BTRx 位域

位号	位名	要设置的值
31:30	DATAHLD	数据保持阶段的持续时间（对于读取访问，为 DATAHLD 个 HCLK 周期）。
29:28	ACCMOD	0x0
27:24	DATLAT	无关
23:20	CLKDIV	无关
19:16	BUSTURN	NEx 变为高电平到 NEx 变为低电平之间的时间 (BUSTURN HCLK)。
15:8	DATAST	写入访问第二个阶段的持续时间（DATAST+1 个 HCLK 周期）
7:4	ADDHLD	无关
3:0	ADDSET	写入访问第一个阶段的持续时间（ADDSET+1 个 HCLK 周期）。ADDSET 的最小值为 0。

表 86. FSMC_BWTRx 位域

位号	位名	要设置的值
31:30	DATAHLD	数据保持阶段的持续时间（对于写入访问，为 DATAHLD+1 个 HCLK 周期）。
29:28	ACCMOD	0x0
27:24	DATLAT	无关
23:20	CLKDIV	无关
19:16	BUSTURN	NEx 变为高电平到 NEx 变为低电平之间的时间 (BUSTURN HCLK)。
15:8	DATAST	写入访问第二个阶段的持续时间（DATAST 个 HCLK 周期）。
7:4	ADDHLD	无关
3:0	ADDSET	写入访问第一个阶段的持续时间（ADDSET 个 HCLK 周期）。 ADDSET 的最小值为 0。

模式 2/B - NOR Flash

图 48. 模式 2 和模式 B 读取访问波形

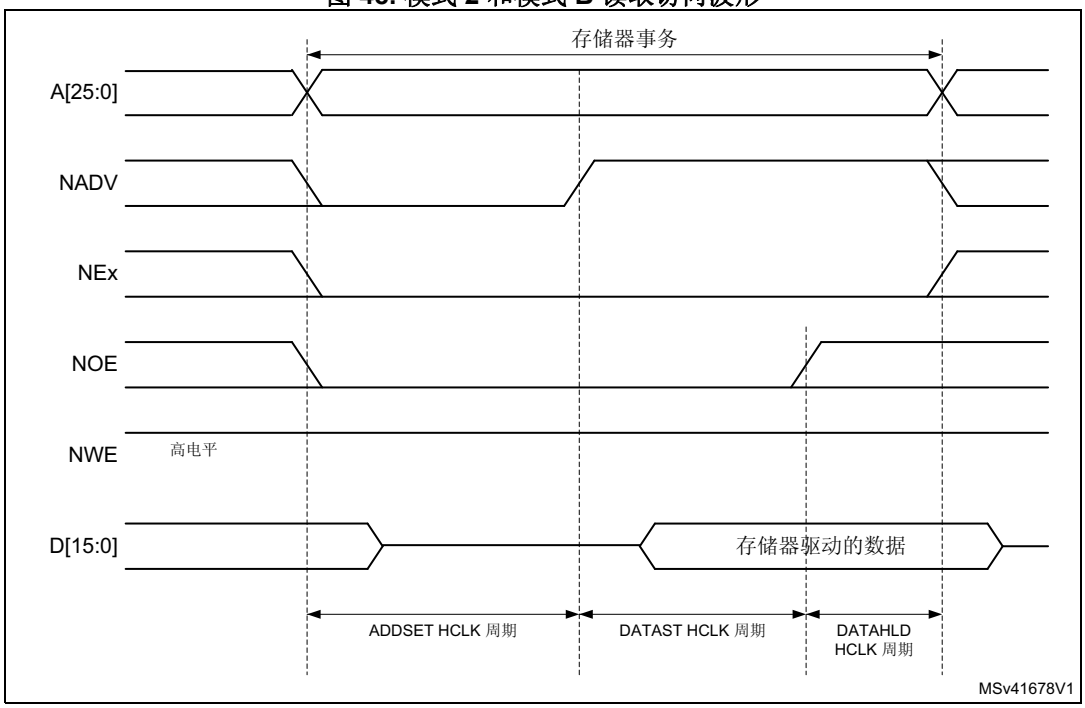


图 49. 模式 2 读取访问波形

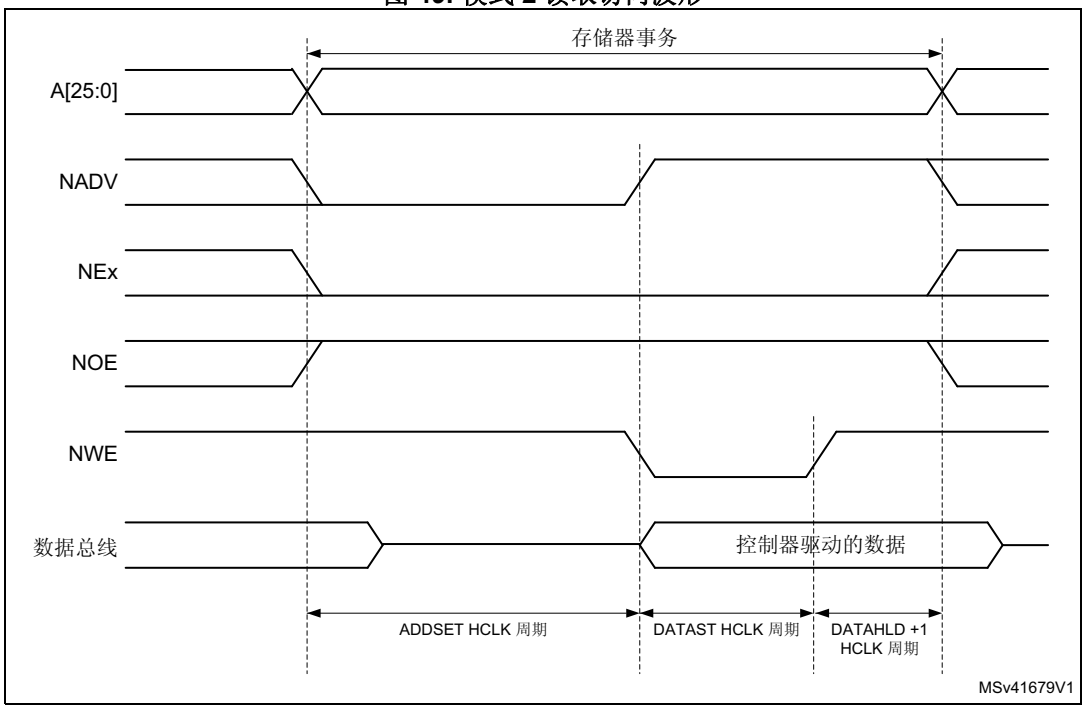
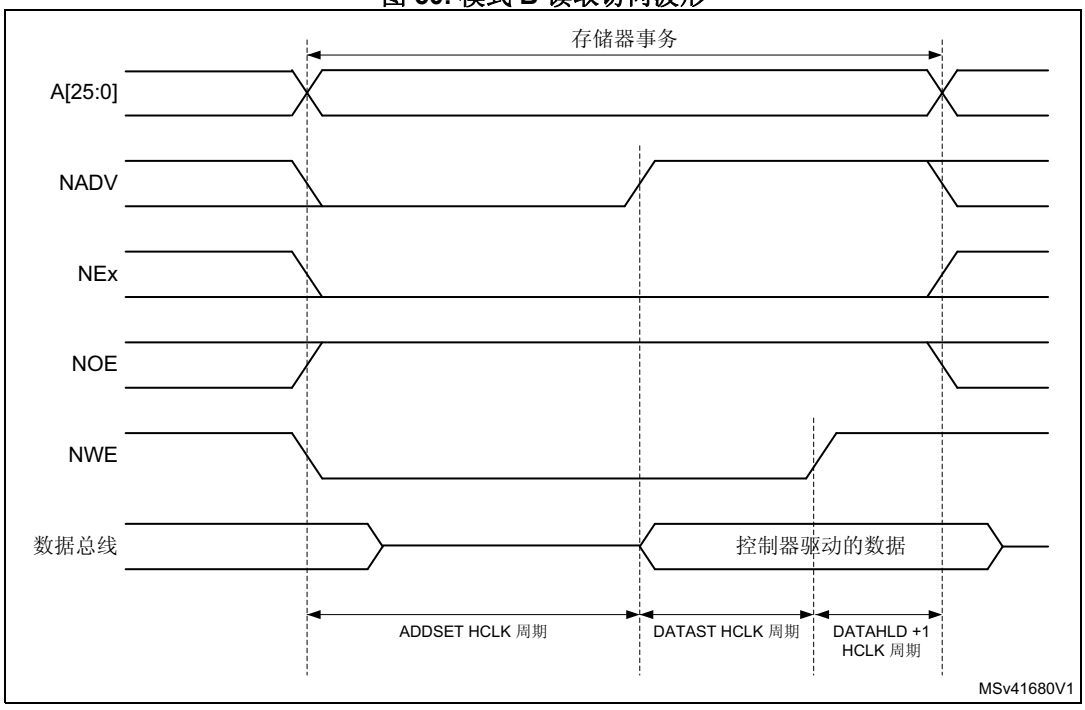


图 50. 模式 B 读取访问波形



与模式 1 的不同之处在于设置扩展模式（模式 B）时的 NWE 切换与独立的读取和写入时序。

表 87. FSMC_BCRx 位域

位号	位名	要设置的值
31:24	保留	0x000
23:22	NBLSET[1:0]	无关
21	WFDIS	根据需要进行设置
20	CCLKEN	根据需要进行设置
19	CBURSTRW	0x0（对异步模式没有影响）
18:16	CPSIZE	0x0（对异步模式没有影响）
15	ASYNCWAIT	如果存储器支持该特性，则置为 1。否则，保持为 0
14	EXTMOD	模式 B 为 0x1，模式 2 为 0x0
13	WAITEN	0x0（对异步模式没有影响）
12	WREN	根据需要进行设置
11	WAITCFG	无关
10	保留	0x0
9	WAITPOL	仅当位 15 为 1 时才有意义
8	BURSTEN	0x0
7	保留	0x1
6	FACCEN	0x1
5:4	MWID	根据需要进行设置
3:2	MTYP	0x2 (NOR Flash)
1	MUXEN	0x0
0	MBKEN	0x1

表 88. FSMC_BTRx 位域

位号	位名	要设置的值
31:30	DATAHLD	数据保持阶段的持续时间（禁止扩展模式时，读取访问为 DATAHLD 个 HCLK 周期，写入访问为 DATAHLD+1 个 HCLK 周期）。
29:28	ACCMOD	如果设置了扩展模式，则为 0x1
27:24	DATLAT	无关
23:20	CLKDIV	无关
19:16	BUSTURN	NEx 变为高电平到 NEx 变为低电平之间的时间 (BUSTURN HCLK)。
15:8	DATAST	读取访问第二个阶段的持续时间（DATAST 个 HCLK 周期）。
7:4	ADDHLD	无关
3:0	ADDSET	读取访问第一个阶段的持续时间（ADDSET 个 HCLK 周期）。ADDSET 的最小值为 0。

表 89. FSMC_BWTRx 位域

位号	位名	要设置的值
31:30	DATAHLD	数据保持阶段的持续时间（对于写入访问，为 DATAHLD+1 个 HCLK 周期）。
29:28	ACCMOD	如果设置了扩展模式，则为 0x1
27:24	DATLAT	无关
23:20	CLKDIV	无关
19:16	BUSTURN	NEx 变为高电平到 NEx 变为低电平之间的时间 (BUSTURN HCLK)。
15:8	DATAST	写入访问第二个阶段的持续时间（DATAST 个 HCLK 周期）。
7:4	ADDHLD	无关
3:0	ADDSET	写入访问第一个阶段的持续时间（ADDSET 个 HCLK 周期）。ADDSET 的最小值为 0。

注： *仅当设置了扩展模式（模式 B）时，FSMC_BWTRx 寄存器才有效，否则其内容均为“无关”。*

模式 C - NOR Flash - OE 切换

图 51. 模式 C 读取访问波形

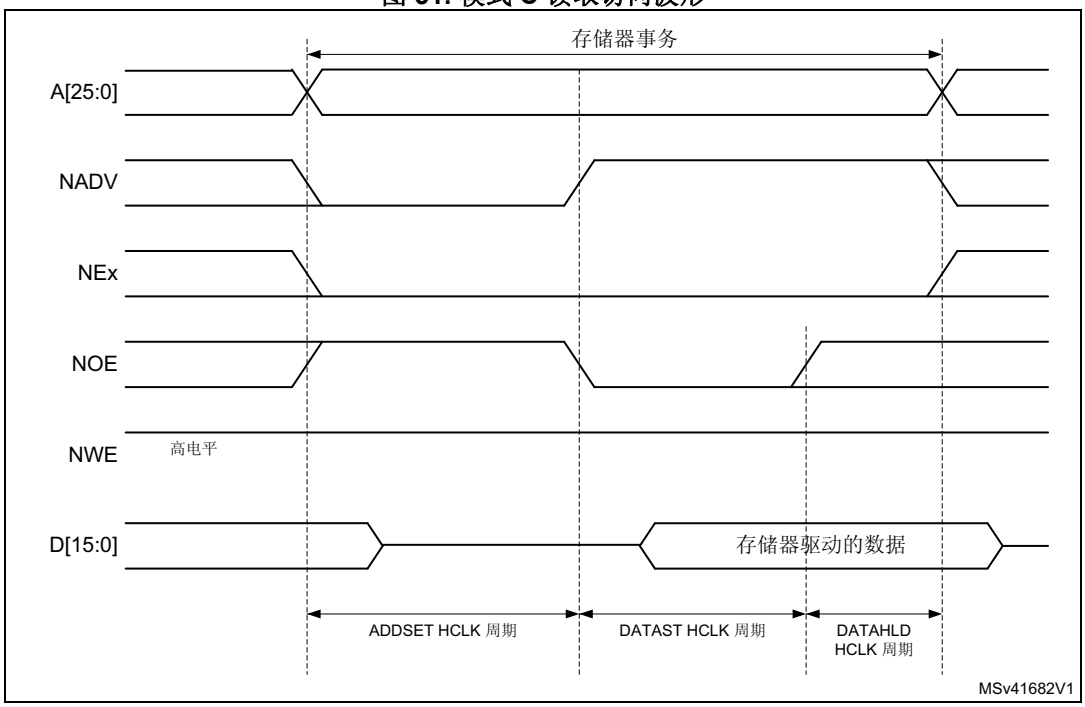
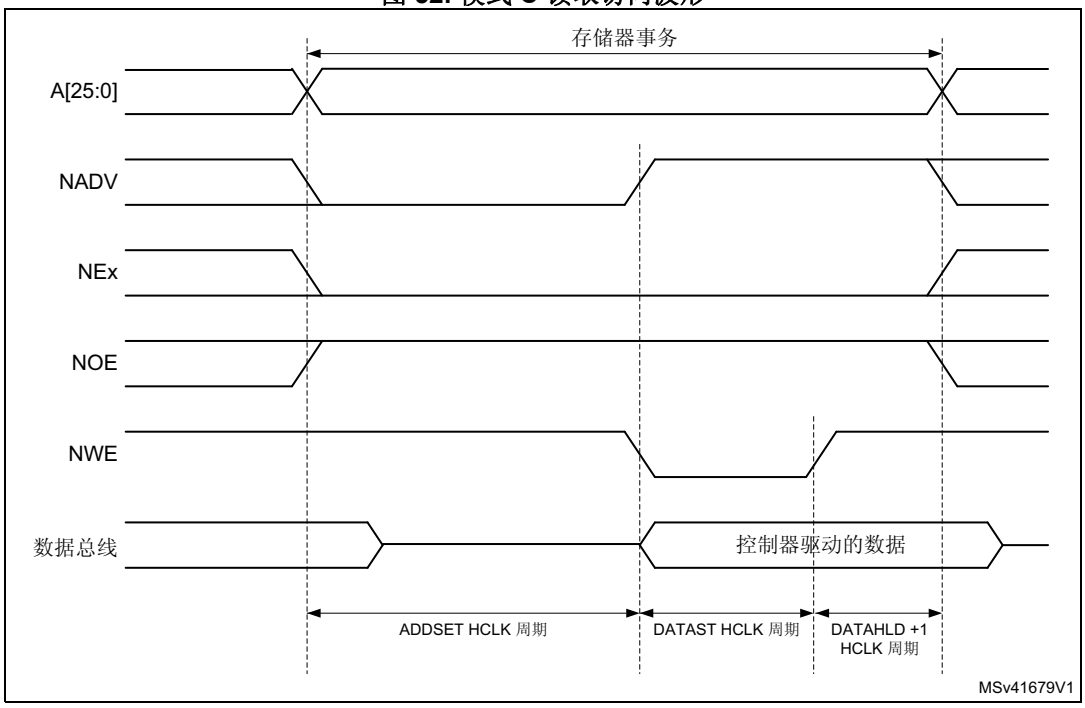


图 52. 模式 C 读取访问波形



与模式 1 的不同之处在于 NOE 的切换与独立的读取和写入时序。

表 90. FSMC_BCRx 位域

位号	位名	要设置的值
31:24	保留	0x000
23:22	NBLSET[1:0]	无关
21	WFDIS	根据需要进行设置
20	CCLKEN	根据需要进行设置
19	CBURSTRW	0x0（对异步模式没有影响）
18:16	CPSIZE	0x0（对异步模式没有影响）
15	ASYNCWAIT	如果存储器支持该特性，则置为 1。否则，保持为 0
14	EXTMOD	0x1
13	WAITEN	0x0（对异步模式没有影响）
12	WREN	根据需要进行设置
11	WAITCFG	无关
10	保留	0x0
9	WAITPOL	仅当位 15 为 1 时才有意义
8	BURSTEN	0x0
7	保留	0x1
6	FACCEN	0x1

表 90. FSMC_BCRx 位域（续）

位号	位名	要设置的值
5:4	MWID	根据需要进行设置
3:2	MTYP	0x02 (NOR Flash)
1	MUXEN	0x0
0	MBKEN	0x1

表 91. FSMC_BTRx 位域

位号	位名	要设置的值
31:30	DATAHLD	数据保持阶段的持续时间（对于读取访问，为 DATAHLD 个 HCLK 周期）。
29:28	ACCMOD	0x2
27:24	DATLAT	0x0
23:20	CLKDIV	0x0
19:16	BUSTURN	NEx 变为高电平到 NEx 变为低电平之间的时间 (BUSTURN HCLK)。
15:8	DATAST	写入访问第二个阶段的持续时间（DATAST+1 个 HCLK 周期）
7:4	ADDHLD	无关
3:0	ADDSET	写入访问第一个阶段的持续时间（ADDSET+1 个 HCLK 周期）。ADDSET 的最小值为 0。

表 92. FSMC_BWTRx 位域

位号	位名	要设置的值
31:30	DATAHLD	数据保持阶段的持续时间（对于写入访问，为 DATAHLD+1 个 HCLK 周期）。
29:28	ACCMOD	0x2
27:24	DATLAT	无关
23:20	CLKDIV	无关
19:16	BUSTURN	NEx 变为高电平到 NEx 变为低电平之间的时间 (BUSTURN HCLK)。
15:8	DATAST	写入访问第二个阶段的持续时间（DATAST 个 HCLK 周期）。
7:4	ADDHLD	无关
3:0	ADDSET	写入访问第一个阶段的持续时间（ADDSET 个 HCLK 周期）。ADDSET 的最小值为 0。

模式 D - 扩展地址异步访问

图 53. 模式 D 读取访问波形

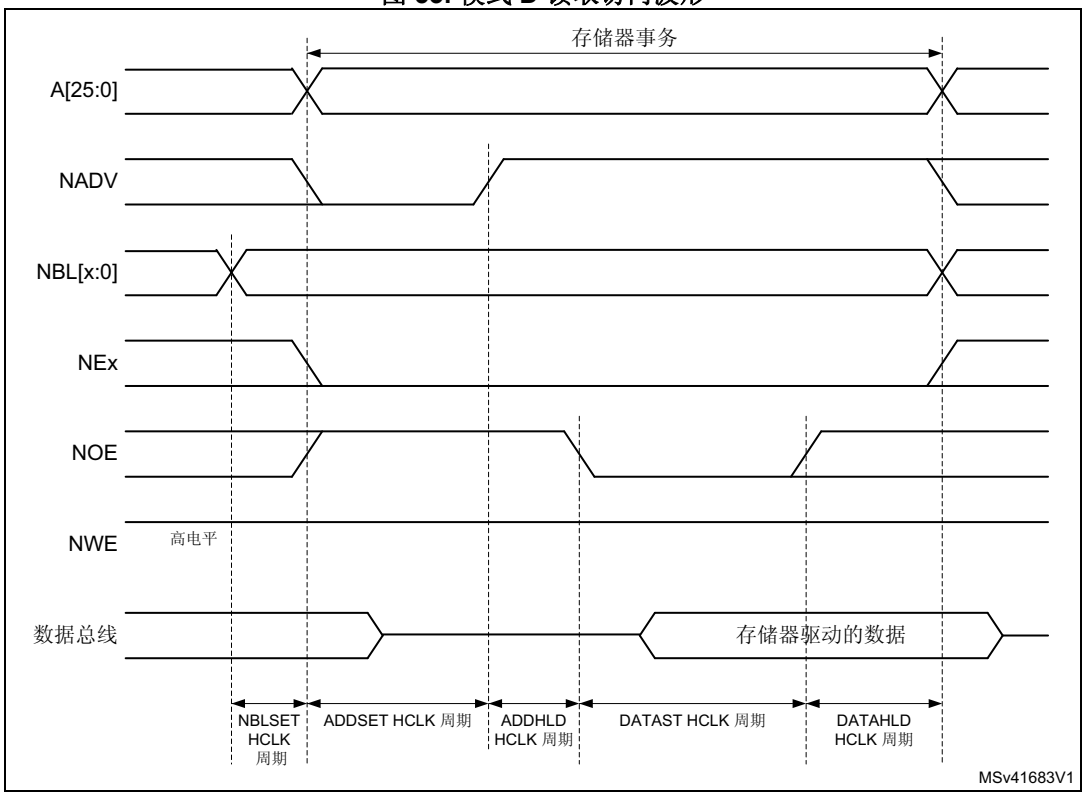
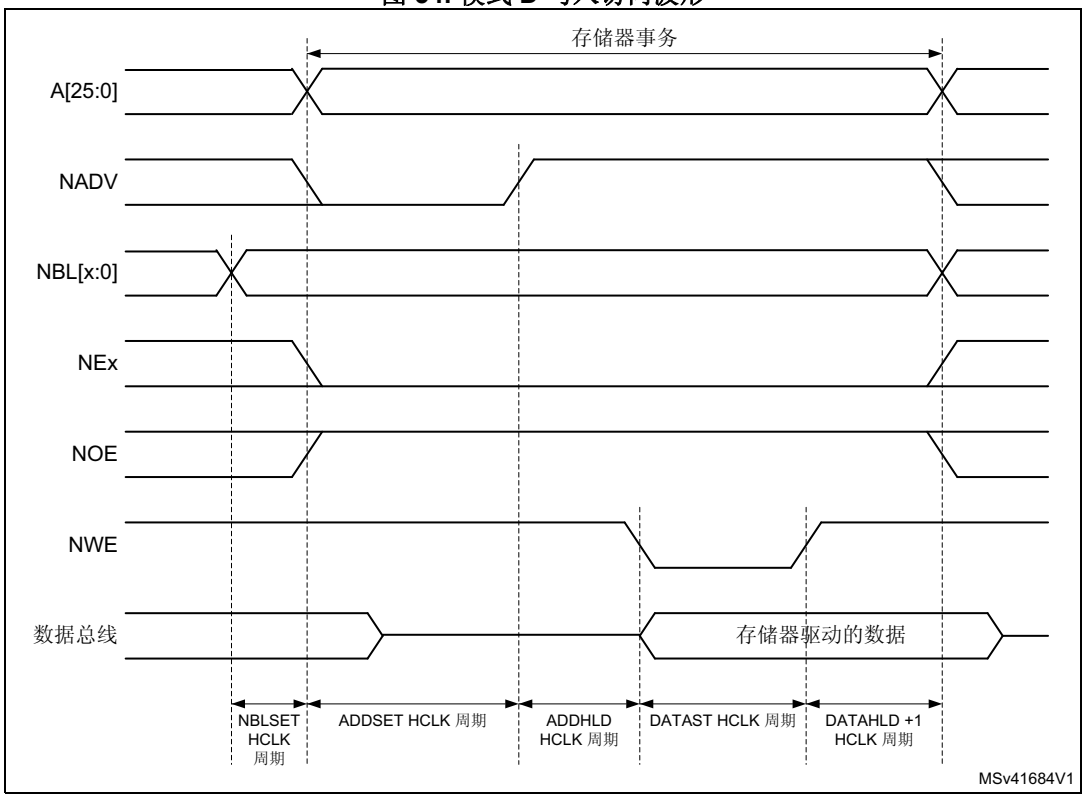


图 54. 模式 D 写入访问波形



与模式 1 的不同之处在于 NADV 变化后 NOE 的切换与独立的读取和写入时序。

表 93. FSMC_BCRx 位域

位号	位名	要设置的值
31:24	保留	0x000
23:22	NBLSET[1:0]	根据需要进行设置
21	WFDIS	根据需要进行设置
20	CCLKEN	根据需要进行设置
19	CBURSTRW	0x0（对异步模式没有影响）
18:16	CPSIZE	0x0（对异步模式没有影响）
15	ASYNCWAIT	如果存储器支持该特性，则置为 1。否则，保持为 0
14	EXTMOD	0x1
13	WAITEN	0x0（对异步模式没有影响）
12	WREN	根据需要进行设置
11	WAITCFG	无关
10	保留	0x0
9	WAITPOL	仅当位 15 为 1 时才有意义
8	BURSTEN	0x0

表 93. FSMC_BCRx 位域（续）

位号	位名	要设置的值
7	保留	0x1
6	FACCEN	根据存储器支持情况进行设置
5:4	MWID	根据需要进行设置
3:2	MTYP	根据需要进行设置
1	MUXEN	0x0
0	MBKEN	0x1

表 94. FSMC_BTRx 位域

位号	位名	要设置的值
31:30	DATAHLD	数据保持阶段的持续时间（对于读取访问，为 DATAHLD 个 HCLK 周期）。
29:28	ACCMOD	0x3
27:24	DATLAT	无关
23:20	CLKDIV	无关
19:16	BUSTURN	NEx 变为高电平到 NEx 变为低电平之间的时间 (BUSTURN HCLK)。
15:8	DATAST	写入访问第二个阶段的持续时间（DATAST+1 个 HCLK 周期）
7:4	ADDHLD	读取访问中间阶段的持续时间（ADDHLD 个 HCLK 周期）。
3:0	ADDSET	写入访问第一个阶段的持续时间（ADDSET+1 个 HCLK 周期）。ADDSET 的最小值为 1。

表 95. FSMC_BWTRx 位域

位号	位名	要设置的值
31:30	DATAHLD	数据保持阶段的持续时间（对于写入访问，为 DATAHLD+1 个 HCLK 周期）。
29:28	ACCMOD	0x3
27:24	DATLAT	无关
23:20	CLKDIV	无关
19:16	BUSTURN	NEx 变为高电平到 NEx 变为低电平之间的时间 (BUSTURN HCLK)。
15:8	DATAST	第二个访问阶段的持续时间（DATAST 个 HCLK 周期）。
7:4	ADDHLD	写入访问中间阶段的持续时间（ADDHLD 个 HCLK 周期）
3:0	ADDSET	写入访问第一个阶段的持续时间（ADDSET 个 HCLK 周期）。ADDSET 的最小值为 1。

复用模式 - 复用异步访问 NOR Flash

图 55. 复用读取访问波形

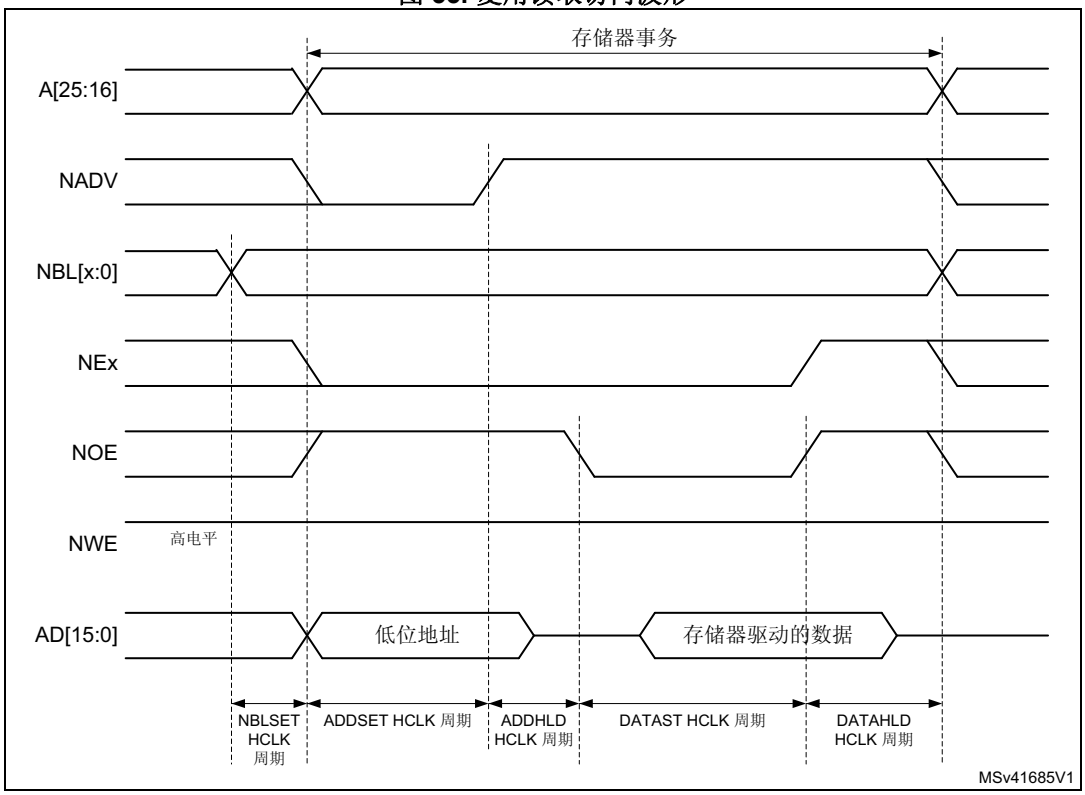
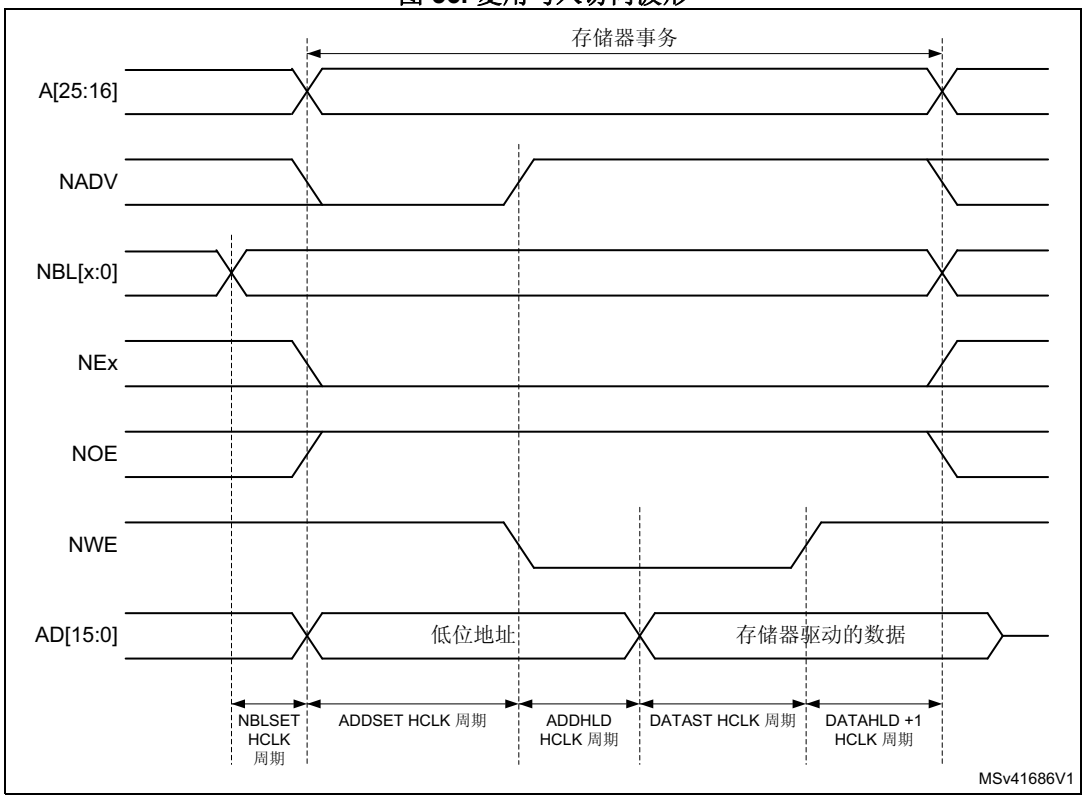


图 56. 复用写入访问波形



与模式 D 的不同之处在于在数据总线上驱动低地址字节。

表 96. FSMC_BCRx 位域

位号	位名	要设置的值
31:24	保留	0x000
23:22	NBLSET[1:0]	根据需要进行设置
21	WFDIS	根据需要进行设置
20	CCLKEN	根据需要进行设置
19	CBURSTRW	0x0（对异步模式没有影响）
18:16	CPSIZE	0x0（对异步模式没有影响）
15	ASYNCWAIT	如果存储器支持该特性，则置为 1。否则，保持为 0
14	EXTMOD	0x0
13	WAITEN	0x0（对异步模式没有影响）
12	WREN	根据需要进行设置
11	WAITCFG	无关
10	保留	0x0
9	WAITPOL	仅当位 15 为 1 时才有意义
8	BURSTEN	0x0
7	保留	0x1

表 96. FSMC_BCRx 位域（续）

位号	位名	要设置的值
6	FACCEN	0x1
5:4	MWID	根据需要进行设置
3:2	MTYP	0x2 (NOR Flash) 或 0x1 (PSRAM)
1	MUXEN	0x1
0	MBKEN	0x1

表 97. FSMC_BTRx 位域

位号	位名	要设置的值
31:30	DATAHLD	数据保持阶段的持续时间（读取访问为 DATAHLD 个 HCLK 周期，写入访问为 DATAHLD+1 个 HCLK 周期）。
29:28	ACCMOD	0x0
27:24	DATLAT	无关
23:20	CLKDIV	无关
19:16	BUSTURN	NEx 变为高电平到 NEx 变为低电平之间的时间 (BUSTURN HCLK)。
15:8	DATAST	第二个访问阶段的持续时间（DATAST 个 HCLK 周期）。
7:4	ADDHLD	访问中间阶段的持续时间（ADDHLD 个 HCLK 周期）。
3:0	ADDSET	第一个访问阶段的持续时间（ADDSET 个 HCLK 周期）。ADDSET 的最小值为 1。

异步访问中的 WAIT 管理

如果异步存储器发出 WAIT 信号，指示尚未准备好接受或提供数据，则 FSMC_BCRx 寄存器中的 ASYNCWAIT 位必须置 1。

如果 WAIT 信号处于有效状态（电平高低取决于 WAITPOL 位），则由 DATAST 位控制的第二个访问阶段（数据建立阶段）将延长，直到 WAIT 变为无效状态。与数据建立阶段不同，由 ADDSET 和 ADDHLD 位控制的第一个访问阶段（地址建立和地址保持阶段）对 WAIT 不敏感，因此第一个访问阶段不会延长。

必须配置数据建立阶段，以便在存储器事务结束前 4 个 HCLK 周期检测到 WAIT。必须考虑以下情况：

1. 存储器发出的 WAIT 信号和 NOE/NWE 信号对齐：

$$DATAST \geq (4 \times HCLK) + max_wait_assertion_time$$

2. 存储器发出的 WAIT 信号和 NEx 对齐（或者 NOE/NWE 信号不翻转）：
   如果

$$max_wait_assertion_time > address_phase + hold_phase$$

   那么：

$$DATAST \geq (4 \times HCLK) + (max_wait_assertion_time - address_phase - hold_phase)$$

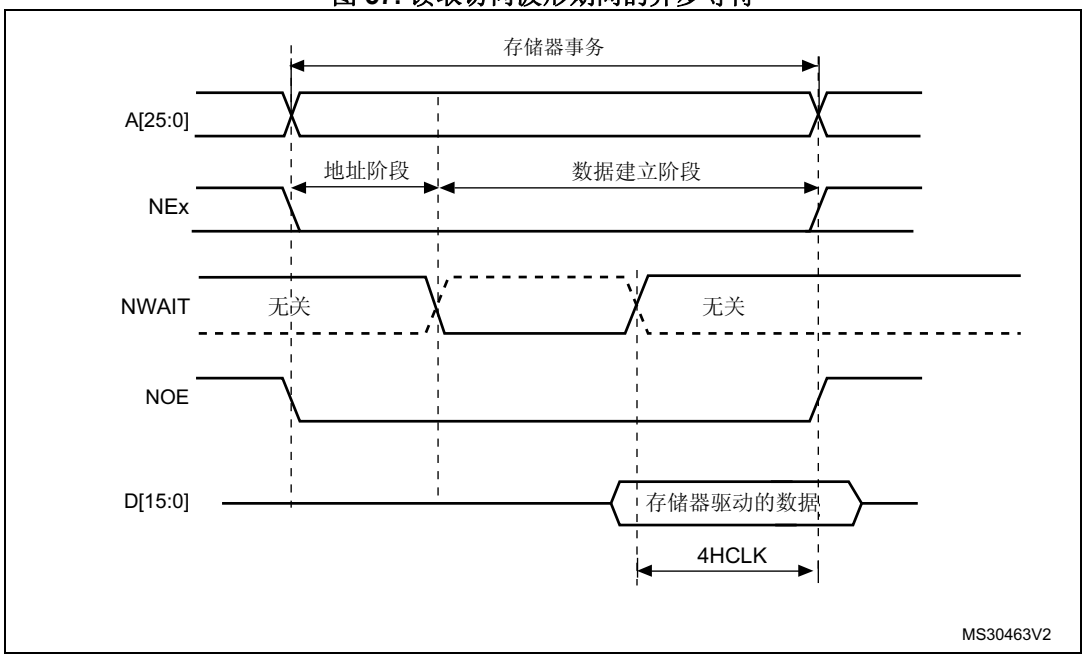
   否则

$$DATAST \geq 4 \times HCLK$$

   其中，max_wait_assertion_time 是在 NEx/NOE/NWE 变为低电平后存储器使能 WAIT 信号所花费的最长时间。

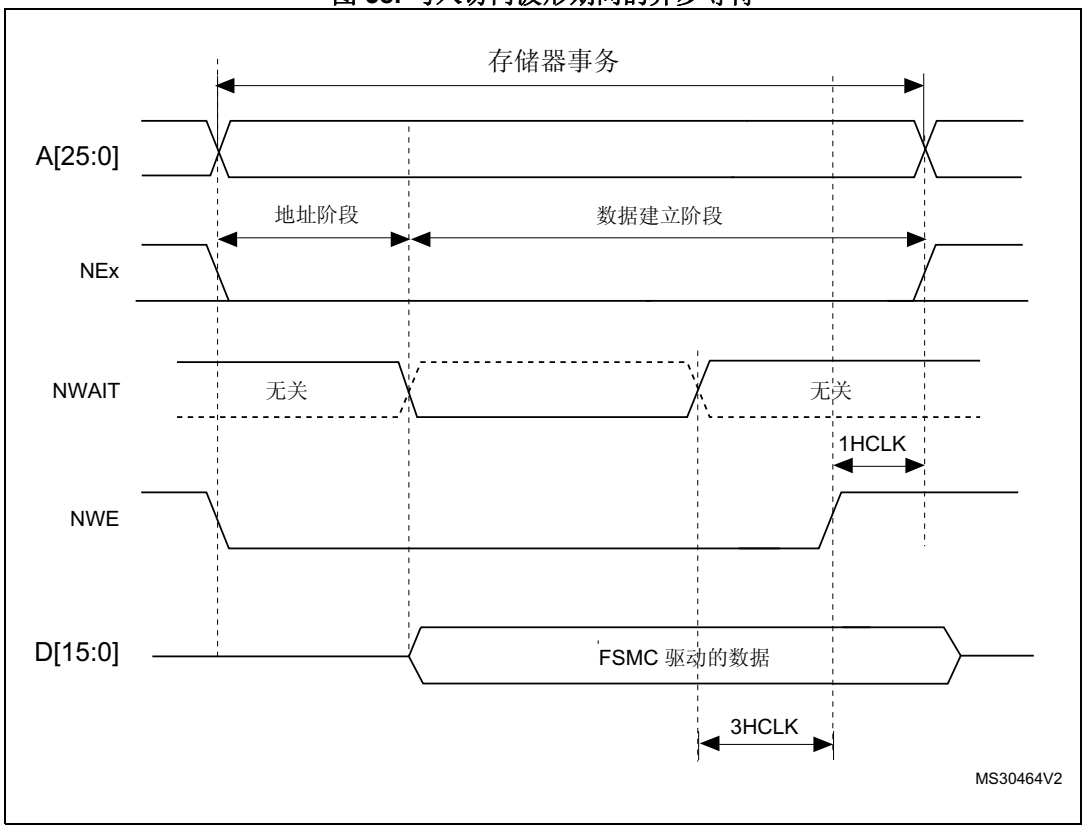
图 57 和*图 58* 显示了异步存储器释放 WAIT 之后，在存储器访问阶段增加的 HCLK 时钟周期的个数（与上述情况无关）。

图 57. 读取访问波形期间的异步等待



1. NWAIT 极性取决于 FSMC_BCRx 寄存器中的 WAITPOL 位设置。

图 58. 写入访问波形期间的异步等待



1. NWAIT 极性取决于 FSMC_BCRx 寄存器中的 WAITPOL 位设置。

18.5.5 同步事务

存储器时钟 FSMC_CLK 是 HCLK 的约数。它取决于 CLKDIV 的值和 MWID/AHB 数据大小，如以下公式所示：

$$\text{FMC_CLK 分频比} = \max(\text{CLKDIV} + 1, \text{MWID(AHB 数据大小)})$$

MWID 为 16 位或 8 位时，FSMC_CLK 分频比始终由 CLKDIV 的设定值定义。

示例：

- CLKDIV = 1，MWID = 16 位，AHB 数据大小 = 8 位时，FSMC_CLK=HCLK/2。

NOR Flash 指定了从 NADV 使能到 CLK 高电平的最短时间。为了符合这一限制，FSMC 不会在同步访问的第一个内部时钟周期内（NADV 使能之前）将时钟发到存储器。这样可以确保存储器时钟的上升沿出现在 NADV 低脉冲的中间。

数据延迟与 NOR 延迟

数据延迟是对数据进行采样之前需要等待的周期数。DATLAT 的值必须与 NOR FLASH 配置寄存器中指定的延迟值一致。当数据延迟计数中 NADV 为低电平时，FSMC 不会计入时钟周期。

注意： 一些 NOR Flash 将 NADV 低电平周期计入数据延迟计数，这样 NOR Flash 延迟和 FSMC DATLAT 参数之间的确切关系可以是以下任一种：

- NOR Flash 延迟 = (DATLAT + 2) 个 CLK 时钟周期
- NOR Flash 延迟 = (DATLAT + 3) 个 CLK 时钟周期

近来有一些存储器会在延迟阶段使能 NWAIT。在这种情况下，可以将 DATLAT 设置为最小值。然后，FSMC 会对数据进行采样，并且等待足够长的时间来评估数据是否有效。这样，FSMC 就能检测到存储器存在延迟的时间，从而处理真实数据。

其他存储器不会在延迟期间使能 NWAIT。在这种情况下，必须正确设置 FSMC 和存储器的延迟，否则可能会将无效数据误用为有效数据，或者在存储器访问初始阶段丢失有效数据。

单次突发传输

当所选存储区域配置为同步突发模式时，例如，如果向 16 位存储器请求了一个 AHB 单次突发事务，则 FSMC 会执行长度为 1 的突发事务（如果 AHB 传输为 16 位）或者长度为 2 的突发事务（如果 AHB 传输为 32 位），然后在最后一个数据选通时禁止片选信号。

与异步读取操作相比，就周期而言这并不是最有效的传输方法。但是，随机异步读取需要先重新编程存储器访问模式，这样总时间会更长。

越过 CellularRAM™ 1.5 的边界页

CellularRAM™ 1.5 不允许突发访问越过页边界。根据存储器页大小配置 FSMC_BCR1 寄存器中的 CPSIZE 位来达到存储器页大小时，FSMC 控制器会自动分离突发访问。

等待管理

对于同步 NOR Flash，会在配置的延迟周期（相当于 (DATLAT+2) 个 CLK 时钟周期）之后对 NWAIT 进行评估。

如果 NWAIT 有效（WAITPOL = 0 时为低电平，WAITPOL = 1 时为高电平），会插入等待状态，直到 NWAIT 无效（WAITPOL = 0 时为高电平，WAITPOL = 1 时为低电平）。

当 NWAIT 无效时，数据将立即（位 WAITCFG = 1）或在下一个时钟边沿（位 WAITCFG = 0）被视为有效。

通过 NWAIT 信号插入等待周期期间，控制器会继续将时钟脉冲发送到存储器，保持片选和输出使能信号有效。但不将数据视为有效。

突发模式下，NOR Flash NWAIT 信号有两种时序配置：

- Flash 在等待周期之前一个数据周期发出 NWAIT 信号（复位后的默认值）。
- Flash 在等待周期期间发出 NWAIT 信号。

FSMC 支持这两种 NOR Flash 等待周期配置，通过 FSMC_BCRx 寄存器的 WAITCFG 位 (x = 0..3) 针对每个片选进行配置。

图 59. 等待配置波形

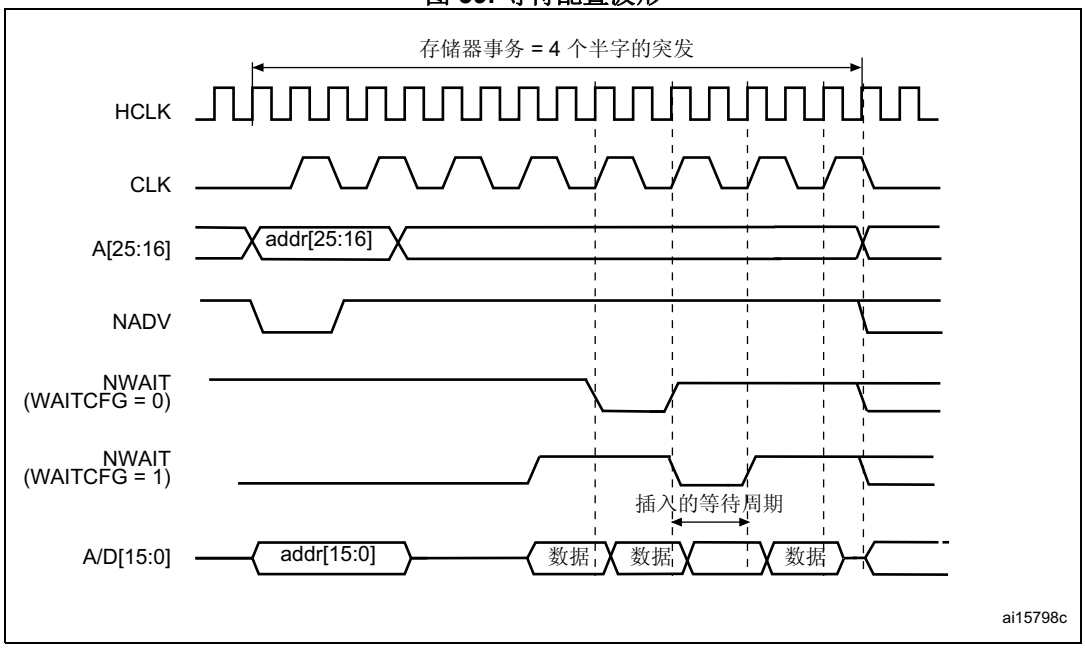
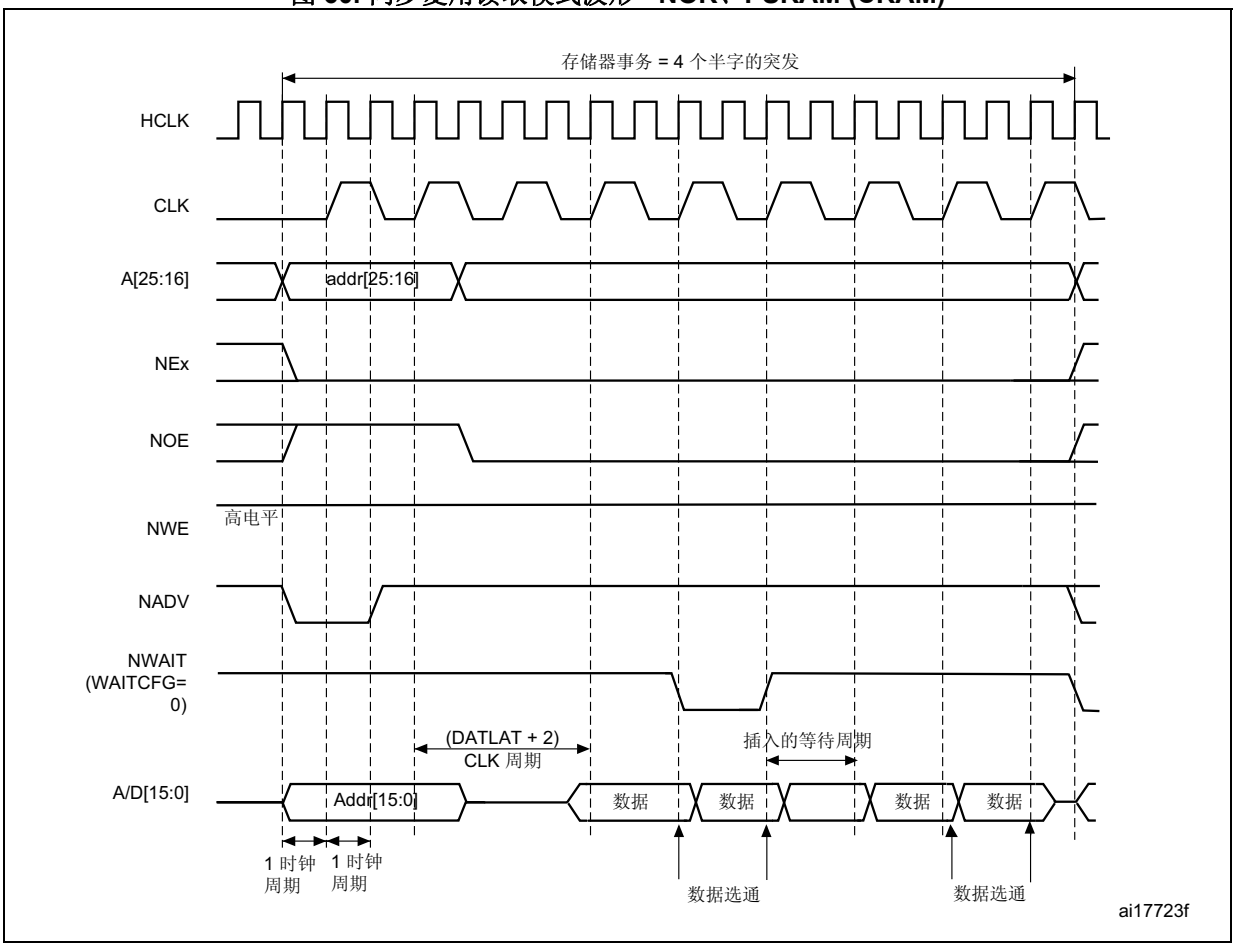

存储器事务 = 4 个半字的突发
HCLK
CLK
A[25:16]
addr[25:16]
NADV
NWAIT
(WAITCFG = 0)
NWAIT
(WAITCFG = 1)
插入的等待周期
A/D[15:0]
addr[15:0]
数据
数据
数据
ai15798c

图 60. 同步复用读取模式波形 - NOR、PSRAM (CRAM)



1. 字节通道输出 NBL 未显示，它们对于 NOR 访问保持高电平，对于 PSRAM (CRAM) 访问则保持低电平。

表 98. FSMC_BCRx 位域

位号	位名	要设置的值
31-24	保留	0x000
23:22	NBLSET[1:0]	无关
21	WFDIS	根据需要进行设置
20	CCLKEN	根据需要进行设置
19	CBURSTRW	对同步读取没有影响
18:16	CPSIZE	0x0（对异步模式没有影响）
15	ASYNCWAIT	0x0
14	EXTMOD	0x0
13	WAITEN	在存储器支持该特性的情况下设置为 1，否则保持为 0
12	WREN	对同步读取没有影响
11	WAITCFG	是否置 1 视存储器情况而定

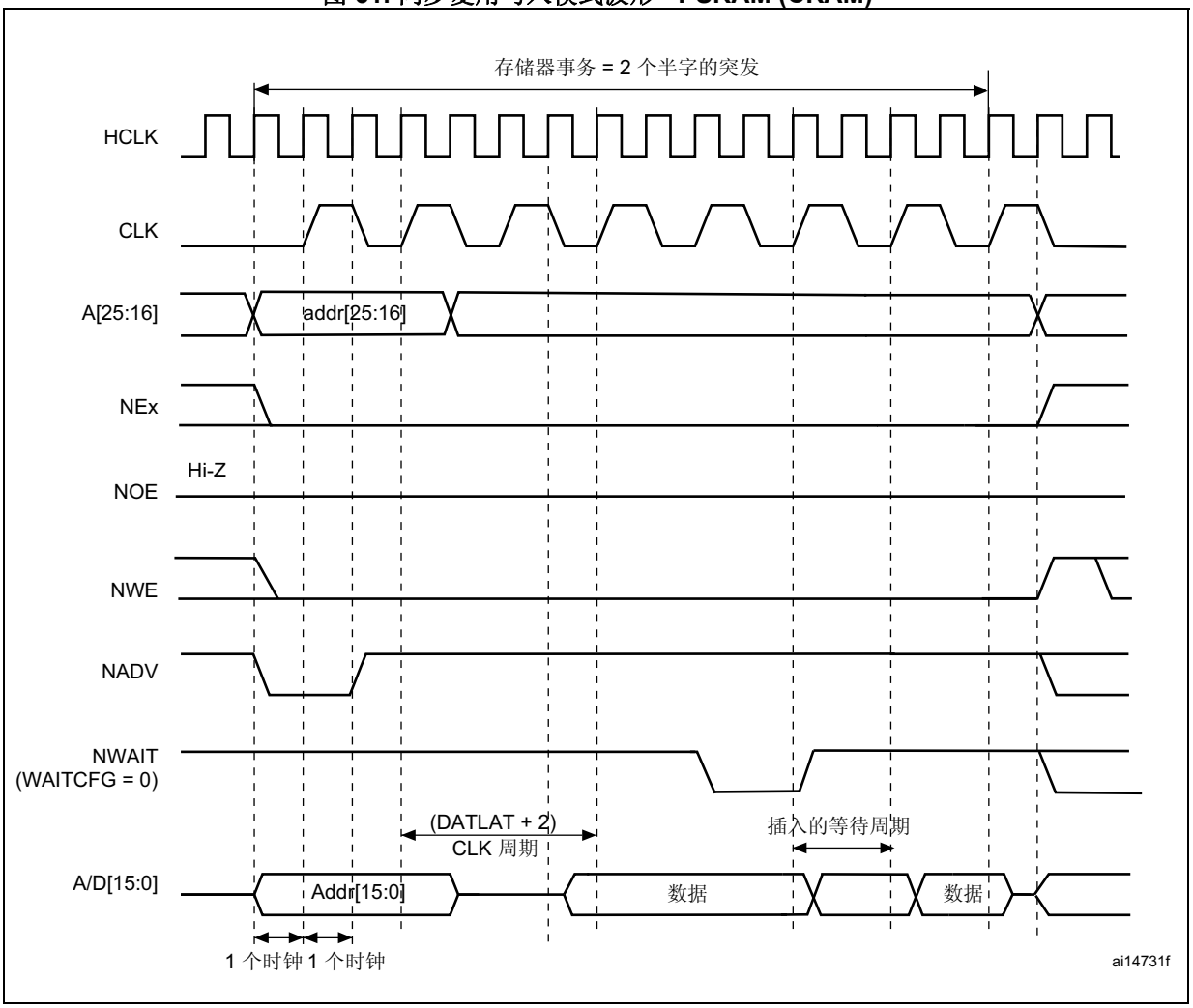
表 98. FSMC_BCRx 位域（续）

位号	位名	要设置的值
10	保留	0x0
9	WAITPOL	是否置 1 视存储器情况而定
8	BURSTEN	0x1
7	保留	0x1
6	FACCEN	在存储器支持的情况下置 1 (NOR Flash)
5-4	MWID	根据需要进行设置
3-2	MTYP	0x1 或 0x2
1	MUXEN	根据需要进行设置
0	MBKEN	0x1

表 99. FSMC_BTRx 位域

位号	位名	要设置的值
31:30	DATAHLD	无关
29:28	ACCMOD	0x0
27-24	DATLAT	数据延迟
27-24	DATLAT	数据延迟
23-20	CLKDIV	0x0，使 CLK = HCLK 0x1，使 CLK = 2 × HCLK ..
19-16	BUSTURN	NEx 变为高电平到 NEx 变为低电平之间的时间 (BUSTURN HCLK)。
15-8	DATAST	无关
7-4	ADDHLD	无关
3-0	ADDSET	无关

图 61. 同步复用写入模式波形 - PSRAM (CRAM)



1. 存储器必须提前一个周期发出 NWAIT 信号，相应地 WAITCFG 必须编程为 0。
2. 字节通道 (NBL) 输出未显示，当 NEx 有效时它们保持低电平。

表 100. FSMC_BCRx 位域

位号	位名	要设置的值
31-24	保留	0x000
23:22	NBLSET[1:0]	无关
21	WFDIS	根据需要进行设置
20	CCLKEN	根据需要进行设置
19	CBURSTRW	0x1
18:16	CPSIZE	根据需要进行设置（CRAM 1.5 为 0x1）
15	ASYNCWAIT	0x0
14	EXTMOD	0x0

表 100. FSMC_BCRx 位域（续）

位号	位名	要设置的值
13	WAITEN	在存储器支持该特性的情况下设置为 1，否则保持为 0
12	WREN	0x1
11	WAITCFG	0x0
10	保留	0x0
9	WAITPOL	是否置 1 视存储器情况而定
8	BURSTEN	对同步写入没有影响
7	保留	0x1
6	FACCEN	根据存储器支持情况进行设置
5-4	MWID	根据需要进行设置
3-2	MTYP	0x1
1	MUXEN	根据需要进行设置
0	MBKEN	0x1

表 101. FSMC_BTRx 位域

位号	位名	要设置的值
31-30	DATAHLD	无关
29:28	ACCMOD	0x0
27-24	DATLAT	数据延迟
23-20	CLKDIV	0x0，使 CLK = HCLK 0x1，使 CLK = 2 × HCLK
19-16	BUSTURN	NEx 变为高电平到 NEx 变为低电平之间的时间 (BUSTURN HCLK)。
15-8	DATAST	无关
7-4	ADDHLD	无关
3-0	ADDSET	无关

18.5.6 NOR/PSRAM 控制器寄存器

SRAM/NOR-Flash 片选控制寄存器 1..4 (FSMC_BCR1..4)

SRAM/NOR-Flash chip-select control registers 1..4

偏移地址：8 * (x – 1)，x = 1...4

复位值：对于存储区域 1 为 0x0000 30DB，对于存储区域 2 到 4 为 0x0000 30D2

该寄存器包含每个存储区域的控制信息，用于 SRAM、PSRAM、FRAM 和 NOR Flash。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	NBLSET[1:0]		WFDIS	CCLK EN	CBURST rw	CPSIZE[2:0]		
								rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ASYNC WAIT	EXT MOD	WAIT EN	WREN	WAIT CFG	Res.	WAIT POL	BURST EN	Res.	FACC EN	MWID[1:0]		MTYP[1:0]		MUX EN	MBK EN
rw	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 23:22 **NBLSET[1:0]**：字节通道 (NBL) 建立 (Byte lane (NBL) SETUP)

这些位将 NBL 建立时间配置为从 NBLx 变为低电平到片选 NEx 变为低电平。

00：NBL 建立时间为 0 个 AHB 时钟周期

01：NBL 建立时间为 1 个 AHB 时钟周期

10：NBL 建立时间为 2 个 AHB 时钟周期

11：NBL 建立时间为 3 个 AHB 时钟周期

位 21 **WFDIS**：写 FIFO 禁止 (Write FIFO Disable)

此位可禁止 FSMC 控制器使用的写 FIFO。

0：使能写 FIFO（复位后的默认值）

1：禁止写 FIFO

注： FSMC_BCR2..4 寄存器的 WFDIS 位为“无关”位。只能通过 FSMC_BCR1 寄存器使能。

位 20 **CCLKEN**：连续时钟使能 (Continuous Clock Enable)。

该位可使能向外部存储器器件输出 FSMC_CLK 时钟。

0：仅在同步存储器访问（读/写事务）期间生成 FSMC_CLK。FSMC_CLK 时钟分频比由 FSMC_BCRx 寄存器中配置的 CLKDIV 值指定（复位后的默认值）。

1：异步和同步访问期间连续生成 FSMC_CLK。CCLKEN 置 1 将激活 FSMC_CLK 时钟。

注： FSMC_BCR2..4 寄存器的 CCLKEN 位为“无关”位。只能通过 FSMC_BCR1 寄存器使能。存储区域 1 必须配置为支持同步模式才能生成 FSMC_CLK 连续时钟。

注： 如果 CCLKEN 位置 1，则 FSMC_CLK 时钟分频比将由 FSMC_BTR1 寄存器的 CLKDIV 值指定。FSMC_BWTR1 寄存器中的 CLKDIV 为无关位。

注： 如果采用同步模式且 CCLKEN 位置 1，则与存储区域 1 之外的其他存储区域相连的同步存储器将共用同一个时钟源（FSMC_BTR2..4 寄存器和 FSMC_BWTR2..4 寄存器中的 CLKDIV 值对其他存储区域不起作用。）

位 19 **CBURSTRW**：突发写使能 (Write burst enable)。

PSRAM (CRAM) 在突发模式下工作时，该位可使能同步写访问。同步读取访问的使能位为 FSMC_BCRx 寄存器中的 BURSTEN 位。

0：始终在异步模式下写入

1：在同步模式下写入

位 18:16 **CPSIZE[2:0]**：CRAM 页大小 (CRAM page size)。

这些位用于 CellularRAM™ 1.5，CellularRAM™ 1.5 不允许突发访问越过页面之间的地址边界。如果配置这些位，FSMC 控制器会在达到存储器页大小后自动分离突发访问（请参见存储器数据手册了解页大小）。

000：越过页边界后不进行突发分离（复位后的默认值）

001：128 字节

010：256 字节

011：512 字节

100：1024 字节

其他值：保留

位 15 **ASYNCWAIT**：异步传输期间的等待信号 (Wait signal during asynchronous transfers)

该位可使能/禁止 FSMC 使用等待信号，即使在异步协议期间也有效。

0：运行异步协议时不考虑 NWAIT 信号（复位后的默认值）

1：运行异步协议时考虑 NWAIT 信号

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

通过此位，可对 FSMCFSMC_BWTR 寄存器中非复用异步访问的写入时序进行配置，进而使读取和写入操作采用不同时序。

0：不考虑 FSMC_BWTR 寄存器中的值（复位后的默认值）

1：考虑 FSMC_BWTR 寄存器中的值

注： 如果禁用扩展模式，FSMC 可以在模式 1 或模式 2 下运行，如下所述：

- *当选择 SRAM/PSRAM 存储器类型时，模式 1 为默认模式（MTYP = 0x0 或 0x01）*
- *当选择 NOR 存储器类型时，模式 2 为默认模式（MTYP = 0x10）*

位 13 **WAITEN**：等待使能位 (Wait enable bit)。

该位可使能/禁止在同步模式下访问存储器时通过 NWAIT 信号插入等待周期。

0：禁止 NWAIT 信号（不考虑其电平，不在配置过的 Flash 延迟周期后插入等待周期）

1：使能 NWAIT 信号（考虑其电平，如果使能，在配置过的延迟周期后插入等待周期）（复位后的默认值）

位 12 **WREN**：写入使能位 (Write enable bit)。

该位指示 FSMC 是否使能/禁止在存储区域内写入：

0：FSMC 禁止在存储区域内写入，如果进行写操作将报告 AHB 错误

1：FSMC 使能在存储区域内写入（复位后的默认值）

位 11 **WAITCFG**：等待时序配置 (Wait timing configuration)。

NWAIT 信号指示存储器中的数据是否有效，或者在同步模式下访问存储器时是否必须插入等待周期。该配置位决定存储器是在等待周期之前的一个时钟周期还是等待周期期间使能 NWAIT：

0：NWAIT 信号在等待周期之前的一个数据周期有效（复位后的默认值）

1：NWAIT 信号在等待周期期间有效（不适用于 PSRAM）

位 10 保留，必须保持复位值。

位 9 **WAITPOL**：等待信号极性位 (Wait signal polarity bit)。

定义同步或异步模式下使用的存储器的等待信号极性：
0：NWAIT 低电平有效（复位后的默认值）
1：NWAIT 高电平有效

位 8 **BURSTEN**：突发使能位 (Burst enable bit)。

该位可使能/禁止同步读取访问。该位仅对突发模式下工作的同步存储器有效：
0：禁止突发模式（复位后的默认值）。在异步模式下进行读取访问。
1：使能突发模式。在同步模式下进行读取访问。

位 7 保留，必须保持复位值。

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

使能 NOR Flash 访问操作。
0：禁止相应的 NOR Flash 访问
1：使能相应的 NOR Flash 访问（复位后的默认值）

位 5:4 **MWID[1:0]**：存储器数据总线宽度 (Memory data bus width)。

定义外部存储器器件宽度，对所有类型的存储器均有效。
00：8 位
01：16 位（复位后的默认值）
10：保留
11：保留

位 3:2 **MTYP[1:0]**：存储器类型 (Memory type)。

定义与相应存储区域相连的外部存储器类型：
00：SRAM/FRAM（对于存储区域 2...4，复位后的默认值）
01：PSRAM (CRAM)/FRAM
10：NOR Flash/OneNAND Flash（对于存储区域 1，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)。

该位置 1 时，地址和数据值在数据总线上复用，仅对 NOR 和 PSRAM 存储器有效：
0：地址/数据非复用
1：地址/数据在数据总线上复用（复位后的默认值）

位 0 **MBKEN**：存储区域使能位 (Memory bank enable bit)。

使能存储区域。复位后使能存储区域 1，其他存储区域均禁止。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
1：使能相应的存储区域

SRAM/NOR-Flash 片选时序寄存器 1..4 (FSMC_BTR1..4)

SRAM/NOR-Flash chip-select timing registers 1..4

偏移地址：0x04 + 8 * (x – 1)，x = 1..4

复位值：0x0FFF FFFF

该寄存器包含每个存储区域的控制信息，用于 SRAM、PSRAM 和 NOR Flash。如果 FSMC_BCRx 寄存器中的 EXTMOD 位置 1，该寄存器将和另外一个寄存器配合来配置写入和读取访问，也就是说有 2 个寄存器可用：一个用于配置读取访问（此寄存器），另一个用于配置写入访问（FSMC_BWTRx 寄存器）。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DATAHLD[1:0]		ACCMOD[1:0]		DATLAT[3:0]				CLKDIV[3:0]				BUSTURN[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
DATAST[7:0]								ADDHLD[3:0]				ADDSET[3:0]			
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:30 **DATAHLD[1:0]**：数据保持阶段持续时间 (Data hold phase duration)

通过软件写入这些位可定义以 HCLK 周期表示的数据保持阶段持续时间（请参见*图 44* 到*图 56*），适用于异步访问模式：

对于读取访问

00：DATAHLD 阶段的持续时间 = 0 × HCLK 时钟周期（默认值）

01：DATAHLD 阶段的持续时间 = 1 × HCLK 时钟周期

10：DATAHLD 阶段的持续时间 = 2 × HCLK 时钟周期

11：DATAHLD 阶段的持续时间 = 3 × HCLK 时钟周期

对于写入访问

00：DATAHLD 阶段的持续时间 = 1 × HCLK 时钟周期（默认值）

01：DATAHLD 阶段的持续时间 = 2 × HCLK 时钟周期

10：DATAHLD 阶段的持续时间 = 3 × HCLK 时钟周期

11：DATAHLD 阶段的持续时间 = 4 × HCLK 时钟周期

位 29:28 **ACCMOD[1:0]**：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 EXTMOD 位置 1 时，这些位才有效。

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 27:24 **DATLAT[3:0]**：同步存储器的数据延迟 (Data latency for synchronous memory)（请参见位说明下的注释）

对于使能了读/写突发模式（BURSTEN/CBURSTRW 位置 1）的同步访问，该位域定义读写首个数据前要发送给存储器的存储器时钟周期数 (+2)：

该时序参数以 FSMC_CLK 周期而非 HCLK 周期表示。

该值与异步访问模式无关。

0000：首次突发访问时，2 个 CLK 时钟周期的数据延迟

1111：首次突发访问时，17 个 CLK 时钟周期的数据延迟（复位后的默认值）

位 23:20 **CLKDIV[3:0]**：FMC_CLK 信号的时钟分频比 (Clock divide ratio (for FSMC_CLK signal))

定义 FSMC_CLK 时钟输出信号的周期，以 HCLK 周期数表示：

0000：FSMC_CLK 周期 = 1 × HCLK 周期

0001：FSMC_CLK 周期 = 2 × HCLK 周期

0010：FSMC_CLK 周期 = 3 × HCLK 周期

1111：FSMC_CLK 周期 = 16 × HCLK 周期（复位后的默认值）

在异步 NOR Flash、SRAM 或 PSRAM 访问模式下，该值为无关值。

注： 有关 FSMC_CLK 分频比公式的信息，请参见第 18.5.5 节：同步事务

位 19:16 **BUSTURN[3:0]**：总线周转阶段的持续时间 (Bus turnaround phase duration)

通过软件写入这些位可在当前读取/写入事务的结尾与同一存储区域的下一事务之间添加延迟。该延迟可以匹配连续事务之间的最短时间（tEHEL 由 NEx 高电平变为 NEx 低电平）以及存储器在读取访问后释放数据总线所需的最长时间（tEHQZ，芯片使能高电平到输出高阻态所经过的时间）：对于模式 D 和复用模式，建议使用此延迟。对于非复用存储器，总线周转延迟可设置为最小值。

(BUSTURN + 1) 个 HCLK 周期 ≥ max(tEHEL 最小值, tEHQZ 最大值)

对于 FRAM 存储器，总线周转延迟应配置为匹配最小 tPC（预充电时间）时序。总线周转延迟插入到同一存储区域的任意连续事务（读/读、写/写、读/写和写/读）之间，以匹配 tPC 存储器时序。片选在任意连续访问之间翻转。

(BUSTURN + 1) 个 HCLK 周期 ≥ tPC 最小值

0000：BUSTURN 阶段的持续时间 = 增加 1 个 HCLK 时钟周期

...

1111：BUSTURN 阶段的持续时间 = 增加 16 个 HCLK 时钟周期（复位后的默认值）

位 15:8 **DATAST[7:0]**：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（请参见*图 44* 到*图 56*），适用于异步访问模式：

0000 0000：保留

0000 0001：DATAST 阶段的持续时间 = 1 × HCLK 时钟周期

0000 0010：DATAST 阶段的持续时间 = 2 × HCLK 时钟周期

...

1111 1111：DATAST 阶段的持续时间 = 255 × 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 44* 到*图 56*）。

例如：在模式 1、写入访问以及 DATAST=1 条件下，数据阶段的持续时间 = DATAST+1 = 2 个 HCLK 时钟周期。

注：　在同步访问模式下，该值为无关值。

位 7:4 **ADDHLD[3:0]**：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 44* 到*图 56*），适用于模式 D 或复用访问：

0000：保留

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 44* 到*图 56*）。

注：　在同步访问模式下，该值不使用，因为地址保持阶段的持续时间始终是 1 个存储器时钟周期。

位 3:0 **ADDSET[3:0]**：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 44* 到*图 56*），适用于 SRAM、ROM、异步 NOR Flash 和 PSRAM：

0000：ADDSET 阶段的持续时间 = 0 × HCLK 时钟周期

...

1111：ADDSET 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设置阶段持续时间的信息，请参见相应图片（*图 44* 到*图 56*）。

注：　在同步访问模式下，该值为无关值。

复用模式或模式 D 下，ADDSET 的最小值为 1。

注：　PSRAM (CRAM) 由于内部刷新而导致数据延时时间长度不确定。因此，这些存储器会在整个延迟阶段发送 NWAIT 信号，以便按照需要延长延迟。

对于 PSRAM (CRAM)，字段 DATLAT 必须设置为 0，这样 FSMC 会立即退出延迟阶段，开始对存储器中的 NWAIT 采样，然后在存储器准备就绪后开始读取或写入。

此方法也适用于最新一代的同步 Flash，同早期 Flash 不同的是，此类 Flash 会发送 NWAIT 信号（检查所用的具体 Flash 数据表）。

SRAM/NOR-Flash 写入时序寄存器 1..4 (FSMC_BWTR1..4)

SRAM/NOR-Flash write timing registers 1..4

偏移地址：0x104 + 8 * (x – 1)，x = 1...4

复位值：0x0FFF FFFF

此寄存器包含每个存储区域的控制信息，用于 SRAM、PSRAM 和 NOR Flash。
当 FSMC_BCRx 寄存器中的 EXTMOD 位置 1 时，该寄存器将处于有效状态，可以进行写入访问。

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

位 31:30 **DATAHLD[1:0]**：数据保持阶段持续时间 (Data hold phase duration)

通过软件写入这些位可定义以 HCLK 周期表示的数据保持阶段持续时间（请参见*图 44* 到*图 56*），适用于异步写访问模式：

00：DATAHLD 阶段的持续时间 = 1 × HCLK 时钟周期（默认值）

01：DATAHLD 阶段的持续时间 = 2 × HCLK 时钟周期

10：DATAHLD 阶段的持续时间 = 3 × HCLK 时钟周期

11：DATAHLD 阶段的持续时间 = 4 × HCLK 时钟周期

位 29:28 **ACCMOD[1:0]**：访问模式 (Access mode)

指定异步访问模式，如下一个时序图所示。仅当 FSMC_BCRx 寄存器中的 EXTMOD 位置 1 时，这些位才有效。

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 27:20 保留，必须保持复位值。

位 19:16 **BUSTURN[3:0]**：总线周转阶段的持续时间 (Bus turnaround phase duration)

通过软件写入这些位可在当前写入事务的结尾与同一存储区域的下一事务之间添加延迟。

对于 FRAM 存储器，总线周转延迟应配置为匹配最小 tPC（预充电时间）时序。总线周转延迟插入到同一存储区域的任意连续事务（读/读、写/写、读/写和写/读）之间。片选在任意连续访问之间翻转。

(BUSTURN + 1) 个 HCLK 周期 ≥ tPC 最小值

0000：BUSTURN 阶段的持续时间 = 增加 1 个 HCLK 时钟周期

...

1111：BUSTURN 阶段的持续时间 = 增加 16 个 HCLK 时钟周期（复位后的默认值）

位 15:8 **DATAST[7:0]**：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（请参见*图 44* 到*图 56*），适用于异步 SRAM、PSRAM 和 NOR Flash 访问模式：

0000 0000：保留

0000 0001：DATAST 阶段的持续时间 = 1 × HCLK 时钟周期

0000 0010：DATAST 阶段的持续时间 = 2 × HCLK 时钟周期

...

1111 1111：DATAST 阶段的持续时间 = 255 × HCLK 时钟周期（复位后的默认值）

位 7:4 **ADDHLD[3:0]**：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 53* 到*图 56*），适用于异步复用访问：

0000：保留

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）

注： 在同步 NOR Flash 访问模式下，该值不使用，因为地址保持阶段的持续时间始终是 1 个 Flash 时钟周期。

位 3:0 **ADDSET[3:0]**：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以 HCLK 周期表示的*地址建立*阶段持续时间（请参见*图 44* 到*图 56*），适用于异步访问模式：

0000：ADDSET 阶段的持续时间 = 0 × HCLK 时钟周期

...

1111：ADDSET 阶段的持续时间 = 15 × HCLK 时钟周期（复位后的默认值）

注： 在同步访问模式下，该值不使用，因为地址建立阶段的持续时间始终是 1 个 Flash 时钟周期。复用模式下，ADDSET 的最小值为 1。

18.6 NAND Flash 控制器

FSMC 会生成适当的信号时序，以驱动以下类型的器件：

- 8 位和 16 位 NAND Flash

NAND 存储区域通过专用的寄存器配置（*第 18.6.7 节*）。可编程的存储器参数包括访问时序（如*表 102* 所示）和 ECC 配置。

表 102. 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所需时钟周期 (HCLK) 数	读/写	AHB 时钟周期 (HCLK)	1	255
存储器等待	命令使能的最小持续时间（按 HCLK 时钟周期计）	读/写	AHB 时钟周期 (HCLK)	2	255
存储器保持	命令禁止后，必须保持地址（如果进行了写访问还需保持数据）的时钟周期 (HCLK) 数	读/写	AHB 时钟周期 (HCLK)	1	254
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)	1	255

18.6.1 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。

注： *前缀“N”标识低电平有效的信号。*

8 位 NAND Flash

表 103. 8 位 NAND Flash

FSMC 信号名称	I/O	功能
A[17]	O	NAND Flash 地址锁存使能 (ALE) 信号
A[16]	O	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	O	片选
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	I	输入 FSMC 的 NAND Flash 就绪/繁忙信号

由于 FSMC 能够管理足够多的地址周期，因此理论上不存在容量限制。

16 位 NAND Flash

表 104. 16 位 NAND Flash

FSMC 信号名称	I/O	功能
A[17]	O	NAND Flash 地址锁存使能 (ALE) 信号
A[16]	O	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	O	片选
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	I	输入 FSMC 的 NAND Flash 就绪/繁忙信号

由于 FSMC 能够管理足够多的地址周期，因此理论上不存在容量限制。

18.6.2 NAND Flash 支持的存储器和事务

表 105 介绍了所支持的器件、访问模式和事务。NAND Flash 控制器不允许（或不支持）的事务以灰色显示。

表 105. 支持的存储器和事务

器件	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

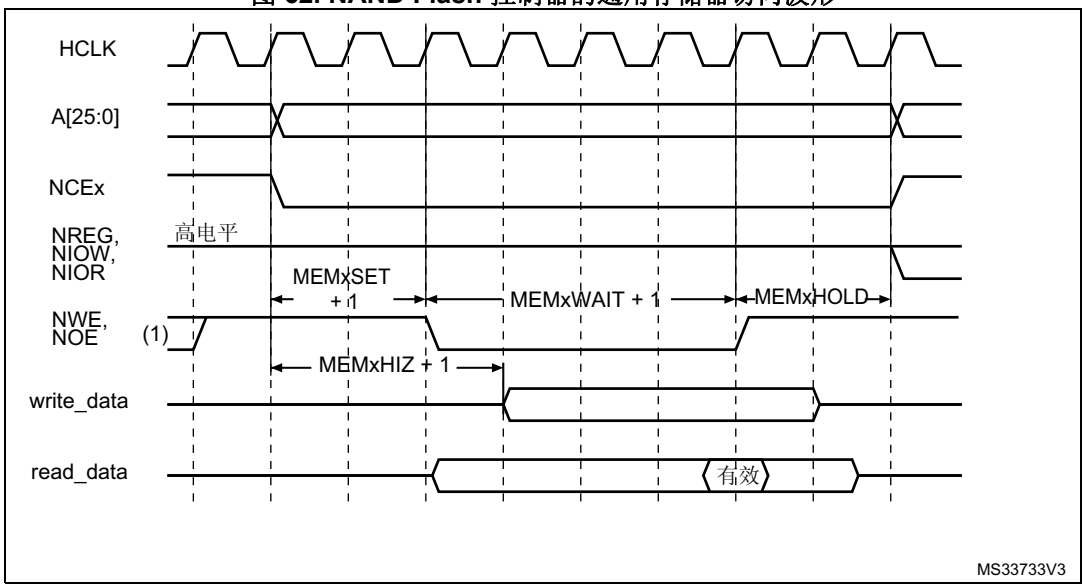
18.6.3 NAND Flash 的时序图

NAND Flash 存储区域通过以下一组寄存器进行管理：

- 控制寄存器：FSMC_PCR
- 中断状态寄存器：FSMC_SR
- ECC 寄存器：FSMC_ECCR
- 通用存储器空间的时序寄存器：FSMC_PMEM
- 特性存储器空间的时序寄存器：FSMC_PATT

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 的三个访问阶段的 HCLK 周期数，另一个参数用于定义进行写访问时开始驱动数据总线的时序。*图 62* 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 62. NAND Flash 控制器的通用存储器访问波形



1. 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）。
2. 对于写访问，保持阶段延迟为 (MEMHOLD) 个 HCLK 周期，而对于读访问，则为 (MEMHOLD + 2) 个 HCLK 周期。

18.6.4 NAND Flash 操作

NAND Flash 器件的命令锁存使能 (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特定地址执行写操作。

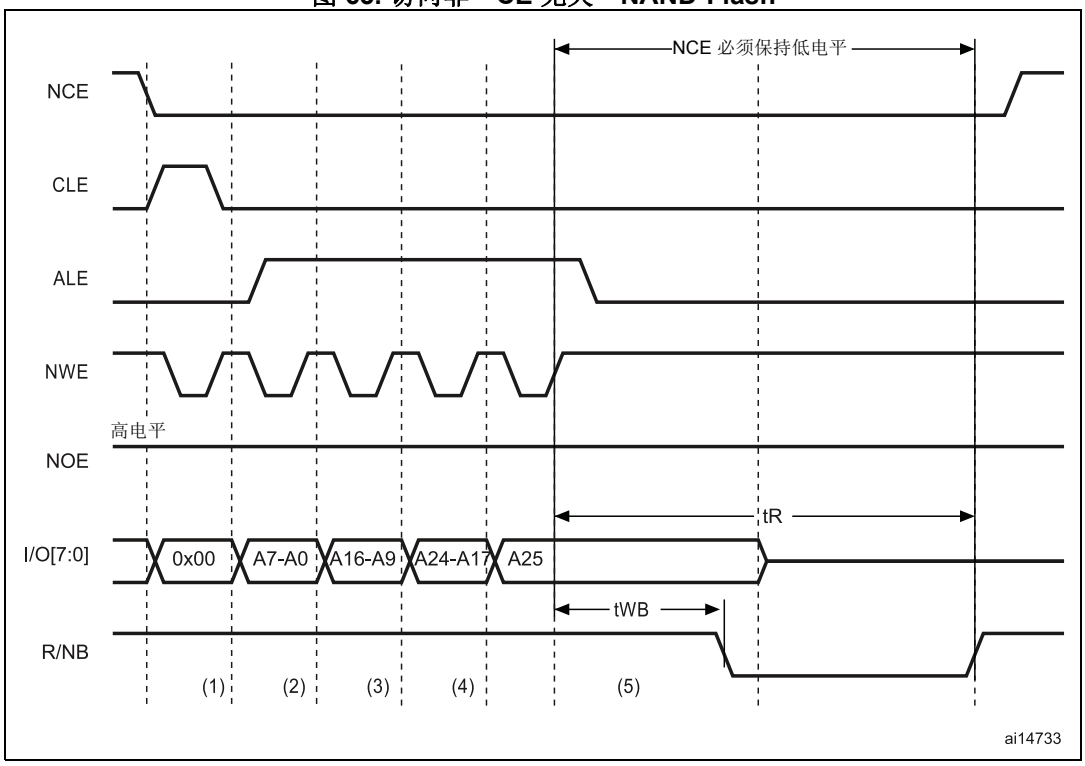
从 NAND Flash 器件进行的典型页读取操作需要执行以下步骤：

1. 根据 NAND Flash 的特性（PWID 位指示 NAND Flash 数据总线宽度，PTYP = 1，根据需要 PWAITEN = 0 或 1；有关时序配置，请参见*第 18.4.2 节：NAND Flash 地址映射*），配置 FSMC_PCR 和 FSMC_PMEM（对于某些器件，配置 FSMC_PATT，请参见*第 18.6.5 节：NAND Flash 预等待功能*）寄存器，进而编程和使能相应的存储区域。
2. CPU 向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 Samsung NAND Flash 器件的字节为 0x00）。在写入选通期间（NWE 上为低电平脉冲），NAND Flash 的 LE 输入有效，因此将已写入的字节视为 NAND Flash 的一个命令。该命令由存储器器件锁存后，随后进行页读取操作时，不需要再次写入该命令。
3. 通过在通用存储器空间或特性空间写入 STARTAD[7:0]、STARTAD[16:9]、STARTAD[24:17] 以及 STARTAD[25]（针对 64 Mb x 8 位的 NAND Flash）这 4 个字节（对于容量较小的器件写入 3 个字节），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 器件的 ALE 输入有效，因此将已写入的字节视为读操作的起始地址。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（有关详细信息，请参见*第 18.6.5 节：NAND Flash 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）。等待期间，控制器保持 NCE 信号有效（低电平）。
5. 然后 CPU 能够从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. 可读取下一个 NAND Flash 页，而无需任何 CPU 命令或地址写操作。可采用以下三种不同的方式实现：
   - 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 器件发送新命令

18.6.5 NAND Flash 预等待功能

一些 NAND Flash 器件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。（见*图 63*）。

图 63. 访问非“CE 无关”NAND-Flash



1. CPU 在地址 0x7001 0000 处写入字节 0x00。
2. CPU 在地址 0x7002 0000 处写入字节 A7~A0。
3. CPU 在地址 0x7002 0000 处写入字节 A16~A9。
4. CPU 在地址 0x7002 0000 处写入字节 A24~A17。
5. CPU 在地址 0x7802 0000 处写入字节 A25：FSMC 通过 FSMC_PATT 时序定义执行写访问，其中 ATTHOLD ≥ 7（假设 (7+1) × HCLK = 112 ns > t_{WB} 最大值）。这可确保 NCE 保持低电平，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 t_{WB} 时序。然而 CPU 对 NAND Flash 进行的所有读取访问均会经过 (MEMHOLD + 2) 个 HCLK 周期的保持延迟，而 CPU 写入访问会经过 (MEMHOLD) 个 HCLK 周期延迟（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 t_{WB} 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 器件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。

18.6.6 NAND Flash 中的纠错码 (ECC) 计算

FSMC NAND 卡控制器包括两个纠错码计算硬件模块，每个存储区域各有一个。这些模块可在软件处理 ECC 时减少主机 CPU 工作负载。

这两个 ECC 模块相同，分别与存储区域 2 和存储区域 3 相关联。因此，硬件 ECC 计算均不适用于与存储区域 4 相连的存储器。

对 NAND Flash 执行读取或写入操作时，相应每 256、512、1 024、2 048、4 096 或 8 192 个字节，FSMC 中使用的 ECC 算法可修正 1 位错误并且检测出 2 位错误。该操作基于 Hamming 编码算法，并且包括计算行和列奇偶校验。

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（NCE 和 NWE）。

ECC 按如下说明操作：

- 当访问 NAND Flash 存储区域 2 或存储区域 3 时，将锁存 D[15:0] 总线上出现的数据并将其用于 ECC 计算。
- 当访问 NAND Flash 中的任何其他地址时，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 对 NAND Flash 完成所需字节数的读取/写入操作后，必须读取 FSMC_ECCR 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_PCR 寄存器中的 ECCEN 位置 1。

要进行 ECC 计算：

1. 使能 FSMC_PCR 寄存器中的 ECCEN 位。
2. 将数据写入 NAND Flash 页。在写入 NAND 页期间，ECC 模块将计算 ECC 值。
3. 读取 FSMC_ECCR 寄存器中所提供的 ECC 值，并将其存储到变量。
4. 将 FSMC_PCR 寄存器中的 ECCEN 位清零后使能，然后从 NAND 页回读写入的数据。在读取 NAND 页期间，ECC 模块将计算 ECC 值。
5. 读取 FSMC_ECCR 寄存器中所提供的新 ECC 值。
6. 如果两次读取的 ECC 值相同，则无需校正，否则说明存在 ECC 错误，并且软件校正例程将返回有关该错误是否能够得到校正的信息。

18.6.7 NAND Flash 控制器寄存器

NAND Flash 控制寄存器 (FSMC_PCR)

NAND Flash control registers

偏移地址：0x80

复位值：0x0000 0018

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.	ECCPS[2:0]			TAR3
												rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAR[2:0]			TCLR[3:0]				Res.	Res.	ECCEN	PWID[1:0]		PTYP	PBKEN	PWAITEN	Res.
rw	rw	rw	rw	rw	rw	rw			rw	rw	rw	rw	rw	rw	

位 31:20 保留，必须保持复位值。

位 19:17 **ECCPS[2:0]**：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 16:13 **TAR[3:0]**：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) × THCLK，其中 THCLK 是 HCLK 时钟周期。
0000：1 个 HCLK 周期（默认）
1111：16 个 HCLK 周期
注： 根据寻址空间，SET 为 MEMSET 或 ATTSET。

位 12:9 **TCLR[3:0]**：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) × THCLK，其中 THCLK 是 HCLK 时钟周期。
0000：1 个 HCLK 周期（默认）
1111：16 个 HCLK 周期
注： 根据寻址空间，SET 为 MEMSET 或 ATTSET。

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[1:0]**：数据总线宽度 (Data bus width)。
定义外部存储器器件宽度。
00：8 位
01：16 位（复位后的默认值）
10：保留
11：保留

位 3 **PTYP**：存储器类型 (Memory type)。
定义附加到相应存储区域的器件的类型：
0：保留，必须保持复位值
1：NAND Flash（复位后为默认值）

位 2 **PBKEN**：NAND Flash 存储区域使能位 (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 1 **PWAITEN**：等待特性使能位 (Wait feature enable bit)。
使能 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 (FSMC_SR)

FIFO status and interrupt register

偏移地址：0x84

复位值：0x0000 0040

该寄存器包含有关 FIFO 状态和中断的信息。FSMC 具有一个 FIFO，当向存储器执行写入操作以传输来自 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 FIFO，然后释放 AHB 供 FSMC 以外的外设的事务使用。这些寄存器位中有一位用来指示 FIFO 的状态，供 ECC 使用。

在将数据写入存储器时计算 ECC。因此，为了读取正确的 ECC，软件必须等到 FIFO 为空。

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.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	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
注： 如果该位由软件写入 1，则将被置 1。

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
注： 如果该位由软件写入 1，则将被置 1。

通用存储器空间时序寄存器 2..4 (FSMC_PMEM)

Common memory space timing register 2..4

偏移地址：地址：0x88

复位值：0xFCFC FCFC

FSMC_PMEM 读/写寄存器包含 NAND Flash 存储区域的时序信息。该信息用于访问 NAND Flash 的通用存储空间来实现命令、地址写访问和数据读/写访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MEMHIZ[7:0]								MEMHOLD[7:0]							
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
MEMWAIT[7:0]								MEMSET[7:0]							
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:24 **MEMHIZ[7:0]**：通用存储器 x 数据总线高阻态时间 (Common memory x databus Hi-Z time)
定义在存储区上对 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：
0000 0000：1 个 HCLK 周期
1111 1110：255 个 HCLK 周期
1111 1111：保留

位 23:16 **MEMHOLD[7:0]**：通用存储器保持时间 (Common memory hold time)
针对在存储区 x 上对 NAND Flash 的通用存储空间执行的读或写访问，在命令（NWE、NOE）被禁止之后定义的保持地址（和写访问数据）的写入访问 HCLK 时钟周期数和读取访问 HCLK (+2) 时钟周期数：
0000 0000：保留
0000 0001：写入访问 1 个 HCLK 周期/读取访问 3 个 HCLK 周期
1111 1110：写入访问 254 个 HCLK 周期/读取访问 256 个 HCLK 周期
1111 1111：保留

位 15:8 **MEMWAIT[7:0]**：通用存储器等待时间 (Common memory wait time)
针对在存储区上对 NAND Flash 的通用存储空间执行的读或写访问，定义使能命令（NWE、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：
0000 0000：保留
0000 0001：2 个 HCLK 周期（+ 禁止 NWAIT 时引入的等待周期）
1111 1110：255 个 HCLK 周期（+ 禁止 NWAIT 时引入的等待周期）
1111 1111：保留

位 7:0 **MEMSET[7:0]**：通用存储器 x 建立时间 (Common memory x setup time)
针对在存储区 x 上对 NAND Flash 的通用存储空间执行的读或写访问，定义使能命令（NWE，NOE）前建立地址所需的 HCLK (+1) 时钟周期数：
0000 0000：1 个 HCLK 周期
1111 1110：255 个 HCLK 周期
1111 1111：保留

特性存储器空间时序寄存器 (FSMC_PATT)

Attribute memory space timing registers

偏移地址：0x8C

复位值：0xFCFC FCFC

FSMC_PATT 读/写寄存器包含 NAND Flash 存储区域的时序信息。当对 NAND Flash 的特性存储空间进行 8 位访问时，如果最后一次地址写访问的时序必须与先前访问的时序不同，则使用该寄存器（有关就绪/繁忙管理的信息，请参见*第 18.6.5 节：NAND Flash 预等待功能*）。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ATTHIZ[7:0]								ATTHOLD[7:0]							
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
ATTWAIT[7:0]								ATTSET[7:0]							
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:24 **ATTHIZ[7:0]**：特性存储器数据总线高阻态时间 (Attribute memory data bus Hi-Z time)

定义在存储区上对 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：

0000 0000：0 个 HCLK 周期

1111 1110：255 个 HCLK 周期

1111 1111：保留

位 23:16 **ATTHOLD[7:0]**：特性存储器保持时间 (Attribute memory hold time)

针对在存储区上对 NAND Flash 的特性存储空间执行的读或写访问，在命令（NWE、NOE）被禁止之后定义的保持地址（和写访问数据）的写入访问 HCLK 时钟周期数和读取访问 HCLK (+2) 时钟周期数：

0000 0000：保留

0000 0001：写入访问 1 个 HCLK 周期/读取访问 3 个 HCLK 周期

1111 1110：写入访问 254 个 HCLK 周期/读取访问 256 个 HCLK 周期

1111 1111：保留

位 15:8 **ATTWAIT[7:0]**：特性存储器等待时间 (Attribute memory wait time)

针对在存储区 x 上对 NAND Flash 的特性存储空间执行的读或写访问，定义使能命令（NWE、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。

0000 0000：保留

0000 0001：2 个 HCLK 周期（+ 禁止 NWAIT 时引入的等待周期）

1111 1110：255 个 HCLK 周期（+ 禁止 NWAIT 时引入的等待周期）

1111 1111：保留

位 7:0 **ATTSET[7:0]**：特性存储器建立时间 (Attribute memory setup time)

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

0000 0000：1 个 HCLK 周期

1111 1110：255 个 HCLK 周期

1111 1111：保留

ECC 结果寄存器 (FSMC_ECCR)

ECC result registers

偏移地址：0x94

复位值：0x0000 0000

该寄存器包含由 FSMC NAND 控制器的 ECC 计算模块计算所得的当前纠错码值。当 CPU 从 NAND Flash 页的正确地址读取数据时（请参见*第 18.6.6 节：NAND Flash 中的纠错码 (ECC) 计算*），ECC 计算模块会自动处理读取自/写入 NAND Flash 的数据。读取 X 个字节后（依据 FSMC_PCR 寄存器的 ECCPS 位域），CPU 必须从 FSMC_ECC 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，如不同则进行校正。FSMC_ECCR 寄存器被读取后应通过将 ECCEN 位置 0 进行清零。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
ECC[31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ECC[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:0 **ECC[31:0]**：ECC 结果 (ECC result)

该位域包含由 ECC 计算逻辑计算所得的值。*表 106* 介绍了这些位域的内容。

表 106. ECC 结果相关位

ECCPS[2:0]	以字节为单位的页大小	ECC 位
000	256	ECC[21:0]
001	512	ECC[23:0]
010	1024	ECC[25:0]
011	2048	ECC[27:0]
100	4096	ECC[29:0]
101	8192	ECC[31:0]

18.7 FSMC 寄存器映射

表 107. FSMC 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
0x00	**FSMC_BCR1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	NBL SET [1:0]		WFDIS	CCLKEN	CBURSTRW	CPSIZE [2:0]			ASYNCWAIT	EXTMOD	WAITEN	WREN	WAITCFG	Res.	WAITPOL	BURSTEN	Res.	FACCEN	MWID [1:0]		MTYP [1:0]		MUXEN	MBKEN
	Reset value									0	0	0	0	0	0	0	0	0	0	1	1	0		0	0		1	0	1	1	0	1	1
0x08	**FSMC_BCR2**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	NBL SET [1:0]		Res.	Res.	CBURSTRW	CPSIZE [2:0]			ASYNCWAIT	EXTMOD	WAITEN	WREN	WAITCFG	Res.	WAITPOL	BURSTEN	Res.	FACCEN	MWID [1:0]		MTYP [1:0]		MUXEN	MBKEN
	Reset value									0	0			0	0	0	0	0	0	1	1	0		0	0		1	0	1	0	0	1	0
0x10	**FSMC_BCR3**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	NBL SET [1:0]		Res.	Res.	CBURSTRW	CPSIZE [2:0]			ASYNCWAIT	EXTMOD	WAITEN	WREN	WAITCFG	Res.	WAITPOL	BURSTEN	Res.	FACCEN	MWID [1:0]		MTYP [1:0]		MUXEN	MBKEN
	Reset value									0	0			0	0	0	0	0	0	1	1	0		0	0		1	0	1	0	0	1	0
0x18	**FSMC_BCR4**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	NBL SET [1:0]		Res.	Res.	CBURSTRW	CPSIZE [2:0]			ASYNCWAIT	EXTMOD	WAITEN	WREN	WAITCFG	Res.	WAITPOL	BURSTEN	Res.	FACCEN	MWID [1:0]		MTYP [1:0]		MUXEN	MBKEN
	Reset value									0	0			0	0	0	0	0	0	1	1	0		0	0		1	0	1	0	0	1	0
0x04	**FSMC_BTR1**	DATAHLD [1:0]		ACCMOD[1:0]		DATLAT[3:0]				CLKDIV[3:0]				BUSTURN[3:0]				DATAST[7:0]								ADDHLD[3:0]				ADDSET[3:0]			
	Reset value	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0x0C	**FSMC_BTR2**	DATAHLD [1:0]		ACCMOD[1:0]		DATLAT[3:0]				CLKDIV[3:0]				BUSTURN[3:0]				DATAST[7:0]								ADDHLD[3:0]				ADDSET[3:0]			
	Reset value	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0x14	**FSMC_BTR3**	DATAHLD [1:0]		ACCMOD[1:0]		DATLAT[3:0]				CLKDIV[3:0]				BUSTURN[3:0]				DATAST[7:0]								ADDHLD[3:0]				ADDSET[3:0]			
	Reset value	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0x1C	**FSMC_BTR4**	DATAHLD [1:0]		ACCMOD[1:0]		DATLAT[3:0]				CLKDIV[3:0]				BUSTURN[3:0]				DATAST[7:0]								ADDHLD[3:0]				ADDSET[3:0]			
	Reset value	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0x104	**FSMC_BWTR1**	DATAHLD [1:0]		ACCMOD[1:0]		Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BUSTURN[3:0]				DATAST[7:0]								ADDHLD[3:0]				ADDSET[3:0]			
	Reset value	0	0	0	0									1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

表 107. FSMC 寄存器映射（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
0x10C	**FSMC_BWTR2**	DATAHLD [1:0]		ACCMOD[1:0]		Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BUSTURN[3:0]				DATAST[7:0]								ADDHLD[3:0]				ADDSET[3:0]			
	Reset value	0	0	0	0									1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0x114	**FSMC_BWTR3**	DATAHLD [1:0]		ACCMOD[1:0]		Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BUSTURN[3:0]				DATAST[7:0]								ADDHLD[3:0]				ADDSET[3:0]			
	Reset value	0	0	0	0									1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0x11C	**FSMC_BWTR4**	DATAHLD [1:0]		ACCMOD[1:0]		Res	Res	Res	Res	Res	Res	Res	Res	BUSTURN[3:0]				DATAST[7:0]								ADDHLD[3:0]				ADDSET[3:0]			
	Reset value	0	0	0	0									1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0x80	**FSMC_PCR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ECCPS [2:0]				TAR[3:0]				TCLR[3:0]				Res.	Res.	ECCEN	PWID [1:0]		PTYP	PBKEN	PWAITEN	Res.
	Reset value													0	0	0	0	0	0	0	0	0	0	0	0			0	0	1	1	0	0	
0x84	**FSMC_SR**	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	FEMPT	IFEN	ILEN	IREN	IFS	ILS	IRS
	Reset value																										1	0	0	0	0	0	0
0x88	**FSMC_PMEM**	MEMHIZx[7:0]								MEMHOLDx[7:0]								MEMWAITx[7:0]								MEMSETx[7:0]							
	Reset value	1	1	1	1	1	1	0	0	1	1	1	1	1	1	0	0	1	1	1	1	1	1	0	0	1	1	1	1	1	1	0	0
0x8C	**FSMC_PATT**	ATTHIZ[7:0]								ATTHOLD[7:0]								ATTWAIT[7:0]								ATTSET[7:0]							
	Reset value	1	1	1	1	1	1	0	0	1	1	1	1	1	1	0	0	1	1	1	1	1	1	0	0	1	1	1	1	1	1	0	0
0x94	**FSMC_ECCR**	ECCx[31:0]																															
	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*第 78 页的第 2.2.2 节*。

19 八线 SPI 接口 (OCTOSPI)

OCTOSPI 是一种专用的通信接口，连接单线、双线、四线或八线 SPI 存储器。该接口可以在以下三种模式下工作：

- 间接模式：使用 OCTOSPI 寄存器执行全部操作
- 状态轮询模式：周期性读取外部存储器状态寄存器，而且标志位置 1 时会产生中断
- 内存映射模式：外部存储器进行存储映射，系统将其视作支持读写操作的内部存储器

OCTOSPI 支持两种帧格式：

- 传统帧格式，包括命令、地址、交替字节、空指令周期和数据阶段
- HyperBus™ 帧格式

19.1 OCTOSPI 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
- 在内存映射模式下支持读写
- 支持单线、双线、四线和八线通信
- 双片四线模式，通过并行访问两个四线存储器，可同时发送/接收 8 位数据
- 支持 SDR 和 DTR 模式
- 支持数据选通
- 完全可编程操作码
- 完全可编程帧格式
- HyperBus™ 支持
- 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

19.2 OCTOSPI 功能说明

图 64. OCTOSPI 框图（八线通信）

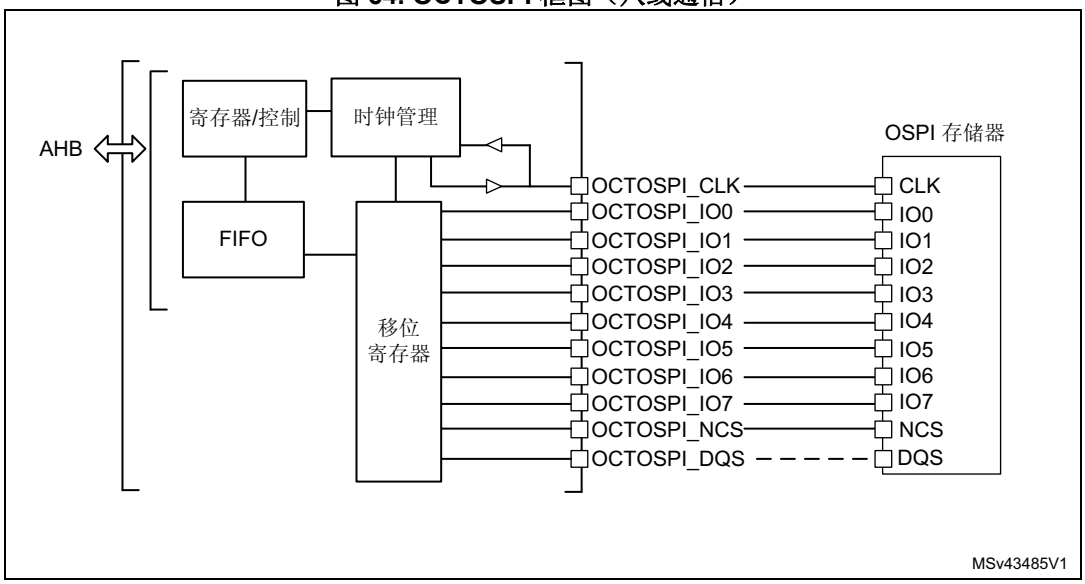


图 65. OCTOSPI 框图（四线模式）

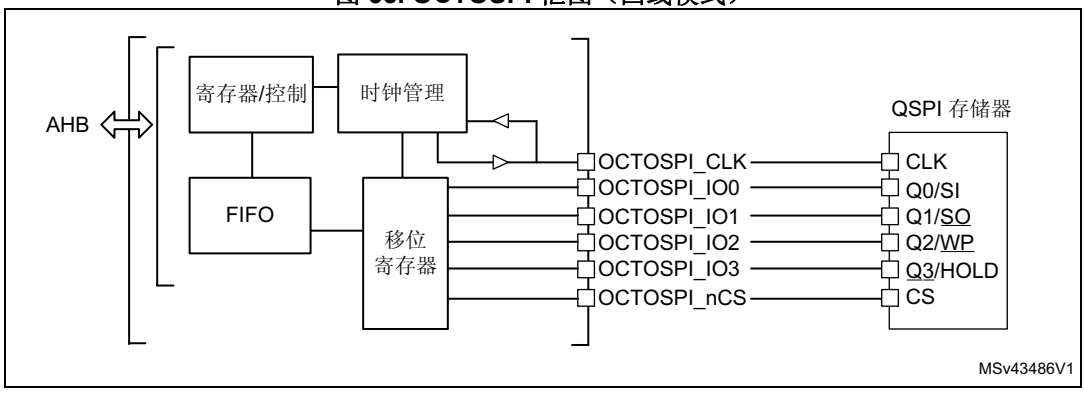
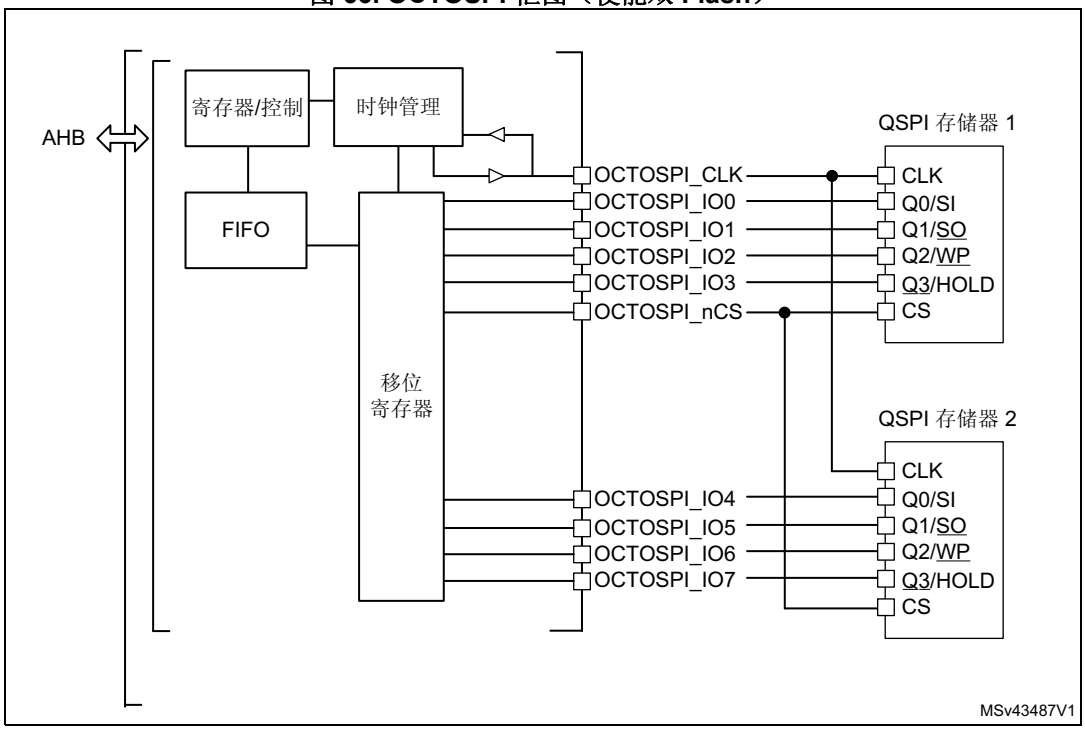


图 66. OCTOSPI 框图（使能双 Flash）



OCTOSPI 具有以下两种工作模式：

- 常规命令模式
- HyperBus™ 模式

在这两种模式下，OCTOSPI 均根据功能模式使用 6 到 12 个信号连接存储器：

- nCS：用于所有模式的片选
- CLK：用于所有模式的通信时钟
- DQS：主要用于八线模式的数据选通
- IO[3:0]：用于所有模式的数据总线 LSB
- IO[7:4]：用于双片四线和八线模式的数据总线 MSB

19.2.1 OCTOSPI 常规命令模式

在常规命令模式下，OCTOSPI 使用命令与外部器件通信。每个命令可以包括 5 个阶段：

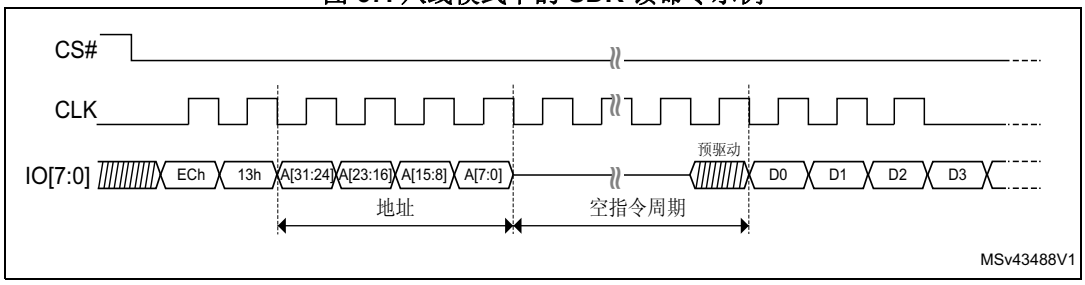
- 指令阶段
- 地址阶段
- 交替字节阶段
- 空指令阶段
- 数据阶段

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。

在内存映射模式下，支持读写操作，因此将复制一些配置寄存器以指定写入操作（使用常规寄存器对读取操作进行配置）。

图 67. 八线模式下的 SDR 读命令示例



特定的常规命令模式功能通过寄存器在 0x0100-0x01FC 偏移范围内进行配置。

指令阶段

这一阶段，将 1 到 4 个字节的指令发送到外部器件，指定待执行操作的类型。在 ISIZE[1:0] (OCTOSPI_CCR[5:4]) 位域中配置待发送指令的大小，在 OCTOSPI_IR 寄存器的 INSTRUCTION[31:0] 位域中编程指令。

尽管大多数器件从 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）、一次发送 4 位（在四线 SPI 模式中通过 IO0 到 IO3）或一次发送 8 位（在八线 SPI 模式中通过 IO0 到 IO7）。这可通过 IMODE[2:0] (OCTOSPI_CCR[2:0]) 位域进行配置。

通过将 IDTR (OCTOSPI_CCR[3]) 位置 1，可在时钟的每个上升沿和下降沿以 DRT（双倍传输速率）模式发送该指令。

若 IMODE = 000，则跳过指令阶段，命令序列从地址阶段（如果存在）开始。

在内存映射模式下，用于写入操作的指令在 OCTOSPI_WIR 寄存器中指定，指令格式在 OCTOSPI_WCCR 寄存器中指定。用于读取操作的指令和指令格式在常规寄存器（OCTOSPI_IR 和 OCTOSPI_CCR）中指定。

地址阶段

在地址阶段，将 1-4 字节发送到外部器件，指示操作地址。待发送的地址字节数在 ADSIZE[1:0] (OCTOSPI_CCR[13:12]) 位域中进行配置。

在间接模式和自动轮询模式下，待发送的地址字节在 ADDRESS[31:0] (OCTOSPI_AR) 位域中指定。在内存映射模式下，则通过 AHB（来自于 Cortex®-M 内核或 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）、4 位（在四线 SPI 模式中通过 IO0 到 IO3）或 8 位（在八线 SPI 模式中通过 IO0 到 IO7）。这可通过 ADMODE[2:0] (OCTOSPI_CCR[10:8]) 位域进行配置。

通过设定 ADDTR(OCTOQSPI_CCR[11])，地址可以在 DTR 模式（在时钟的每个上升沿和下降沿）被发送。

若 ADMODE = 000，则跳过地址阶段，命令序列直接进入下一阶段（如果存在）。

在内存映射模式下，写入操作的地址格式在 OCTOSPI_WCCR 寄存器中指定。读取操作的地址格式在常规寄存器 (OCTOSPI_CCR) 中指定。

交替字节阶段

在交替字节阶段，将 1-4 字节发送到外部器件，一般用于控制操作模式。待发送的交替字节数在 ABSIZE[1:0] (OCTOSPI_CCR[21:20]) 位域中进行配置。待发送的字节在 OCTOSPI_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 SO）、2 位（在双线 SPI 模式中通过 IO0 和 IO1）、4 位（在四线 SPI 模式中通过 IO0 到 IO3）或 8 位（在八线 SPI 模式中通过 IO0 到 IO7）。这可通过 ABMODE[2:0] (OCTOSPI_CCR[18:16]) 位域进行配置。

通过将 ABDTR (OCTOSPI_CCR[19]) 位置 1，可（在时钟的每个上升沿和下降沿）以 DTR 模式发送交替字节。

若 ABMODE = 000，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 SPI 模式 (ABMODE=011) 并发送一个字节，方法是 ALTERNATE 的位 7 和 3 置“1”（IO3 保持高电平）且位 6 和 2 置“0”（IO2 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 和位 0 中。例如，如果半字节 2 (0010) 通过 IO0/IO1 发送，则 ALTERNATE 应设置为 0x8A (1000_1010)。

在内存映射模式下，用于写入操作的交替字节在 OCTOSPI_WABR 寄存器中指定，交替字节格式在 OCTOSPI_WCCR 寄存器中指定。用于读取操作的交替字节和交替字节格式在常规寄存器（OCTOSPI_ABR 和 OCTOSPI_CCR）中指定。

空指令周期阶段

在空指令周期阶段，给定的 1 到 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，给外部器件留出准备数据阶段的时间。这一阶段中给定的周期数在 DCYC[4:0] (OCTOSPI_TCR[4:0]) 位域中指定。在 SDR 和 DTR 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）。

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、四线或八线模式从外部器件接收数据时，至少需要指定一个空指令周期。

当使用激活了 DQS 的存储器时，建议至少有五个空指令周期。

在内存映射模式下，写入操作的空指令周期在 OCTOSPI_WTCR 寄存器中指定。读取操作的空指令周期在常规寄存器 (OCTOSPI_TCR) 中指定。

数据阶段

在数据阶段，可从外部器件接收或向其发送任意数量的字节。

在间接模式下，待发送/接收的字节数在 OCTOSPI_DLR 寄存器中指定。

在间接写入模式下，发送到外部器件的数据必须写入 OCTOSPI_DR 寄存器。在间接读取模式下，通过读取 OCTOSPI_DR 寄存器获得从外部器件接收的数据。

在自动轮询模式下，待接收的字节数在 OCTOSPI_DLR 寄存器中指定，可通过读取 OCTOSPI_DR 寄存器获得从外部器件接收的数据。

在内存映射模式下，读取或写入的数据通过 AHB 直接发送到或接收自 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/SI）、2 位（在双线 SPI 模式中通过 IO0/IO1）、4 位（在四线 SPI 模式中通过 IO0 到 IO3）或 8 位（在八线 SPI 模式中通过 IO0 到 IO7）。这可通过 DMODE[2:0] (OCTOSPI_CCR[26:24]) 位域进行配置。

通过将 DDTR (OCTOSPI_CCR[27]) 位置 1，可（在时钟的每个上升沿和下降沿）以 DTR 模式发送或接收数据。

若 DMODE = 000，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。

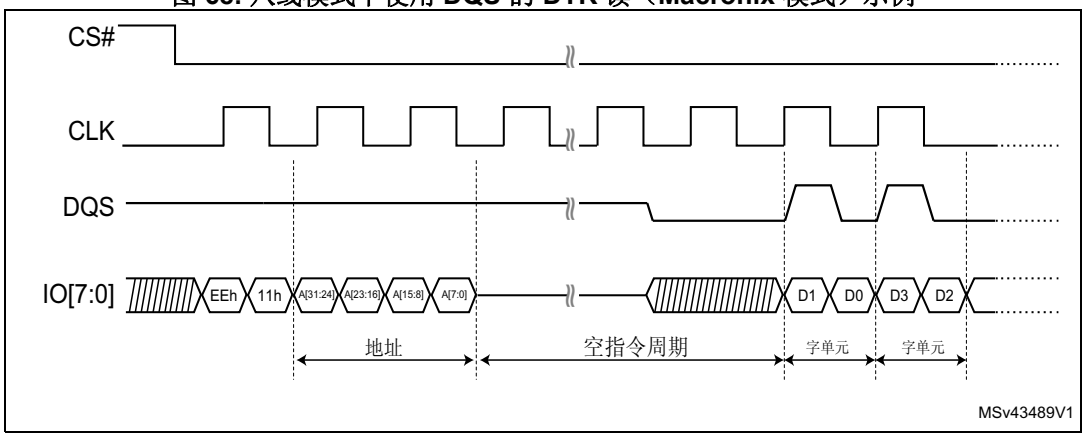
在内存映射模式下，写入操作的数据格式在 OCTOSPI_WCCR 寄存器中指定。读取操作的数据格式在常规寄存器 (OCTOSPI_CCR) 中指定。

DQS 的用法

当器件根据数据切换 DQS 时，DQS 信号可用于读取事务期间的数据选通。

通过将 OCTOSPI 通信配置寄存器 (OCTOSPI_CCR) 的 DQS 使能 (DQSE) 位置 1，可使能 DQS 管理。

图 68. 八线模式下使用 DQS 的 DTR 读（Macronix 模式）示例



19.2.2 OCTOSPI 常规命令模式信号接口

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 IO0 共享）发送到外部器件。从外部器件接收到的数据通过 SI（其 I/O 与 IO1 共享）送达。

通过将（OCTOSPI_CCR 和 OCTOSPI_WCCR 中的）IMODE/ADMODE/ABMODE/DMODE 位域设置为 001，可对不同的命令阶段分别进行配置，以使用此单比特模式。

在每个已配置为单线 SPI 模式的阶段中：

- IO0 (SO) 处于输出模式
- IO1 (SI) 处于输入模式（高阻抗）
- IO2 处于输出模式并强制置“0”（以禁止“写保护”功能）
- IO3 处于输出模式并强制置“1”（以禁止“保持”功能）
- IO4/IO5/IO6/IO7 处于输出模式并强制置“0”

若 DMODE = 001，这对于空指令阶段也同样如此。

双线 SPI 模式

在双线 SPI 模式下，通过 IO0/IO1 信号同时发送/接收两位。

通过将（OCTOSPI_CCR 和 OCTOSPI_WCCR 中的）IMODE/ADMODE/ABMODE/DMODE 位域设置为 010，可对不同的命令阶段分别进行配置，以使用双线 SPI 模式。

在每个已配置为双线 SPI 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”
- IO4/IO5/IO6/IO7 处于输出模式并强制置“0”

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

通过将（OCTOSPI_CCR 和 OCTOSPI_WCCR 中的）IMODE/ADMODE/ABMODE/DMODE 位域设置为 011，可对不同的命令阶段分别进行配置，以使用四线 SPI 模式。

在每个已配置为四线 SPI 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。IO4/IO5/IO6/IO7 处于输出模式并强制置“0”。

在空指令阶段中，若 DMODE = 011，则 IO0/IO1/IO2/IO3 均为高阻态。

IO2 和 IO3 仅用于四线 SPI 模式。如果未配置任何阶段使用四线 SPI 模式，即使 OCTOSPI 激活，对应 IO2 和 IO3 的引脚也可用于其他功能。

八线 SPI 模式

在常规八线 SPI 模式下，通过 IO[0:7] 信号同时发送/接收高位。

通过将（OCTOSPI_CCR 和 OCTOSPI_WCCR 中的）IMODE/ADMODE/ABMODE/DMODE 位域设置为 100，可对不同的命令阶段分别进行配置，以使用八线 SPI 模式。

在每个已配置为八线模式的阶段中，IO[0:7] 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，若 DMODE = 100，则 IO[0:7] 均为高阻态。

IO[4:7] 仅用于八线 SPI 模式。如果未配置任何阶段使用八线 SPI 模式，即使 OCTOSPI 激活，对应 IO[4:7] 的引脚也可用于其他功能。

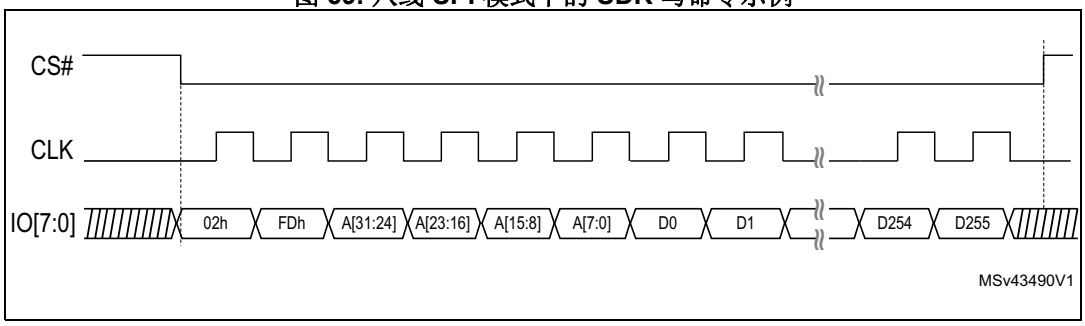
SDR 模式

默认情况下，所有阶段均在单倍数据速率 (SDR) 模式下工作。

在 SDR 模式下，当 OCTOSPI 驱动 IO0/SO、IO1、IO2、IO3、IO4、IO5、IO6 和 IO7 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，OCTOSPI 假定外部器件也通过 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：

图 69. 八线 SPI 模式下的 SDR 写命令示例



DTR 模式

通过将 OCTOSPI_CCR 寄存器中的 IDTR/ADDTR/ABDTR/DDTR 位置 1，可以将指令/地址/交替/数据阶段配置为在双倍传输速率 (DTR) 模式下工作。

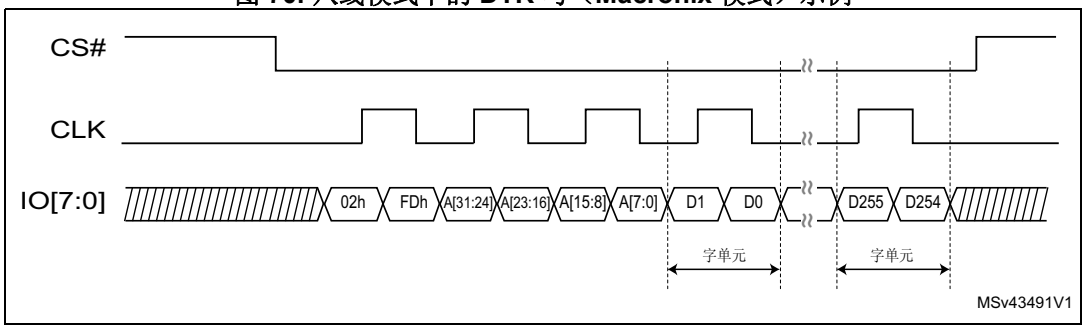
在内存映射模式下，写入操作各个阶段的 DTR 模式在 OCTOSPI_WCCR 寄存器中指定。读取操作各个阶段的 DTR 模式在常规寄存器 (OCTOSPI_CCR) 中指定。

在 DTR 模式下，当 OCTOSPI 在指令/地址/交替字节/数据阶段驱动 IO0/SO、IO1、IO2、IO3、IO4、IO5、IO6 和 IO7 信号时，将在 CLK 的每个上升沿和下降沿发送或接收 1 位。

在 DTR 模式下接收数据时，OCTOSPI 假定外部器件也通过 CLK 的上升沿和下降沿发送数据。若 DDTR = 1，固件必须清零 SSHIFT (OCTOSPI_TCR[30])。因此，在半个 CLK 周期后（下一个反向边沿）对信号采样。

在 DTR 模式下，建议将 OctoSPI 时序配置寄存器 (OCTOSPI_TCR) 的延迟保持四分之一周期 (DHQC) 位置 1，以将输出移位四分之一周期，避免保持存储器端出现的问题。

图 70. 八线模式下的 DTR 写（Macronix 模式）示例



双片四线模式

若 DQM (OCTOSPI_CR[6]) 位为 1，OCTOSPI 处于双片四线模式。OCTOSPI 使用两个外部四线 SPI 器件 （器件 A 和器件 B），在每个周期中发送/接收 8 位 （在 DTR 模式下为 16 位），能够有效地将吞吐量和容量扩大一倍。

每个器件使用同一个 CLK 和 nCS 信号，但其 IO0、IO1、IO2 和 IO3 信号是各自独立的。

双片四线模式可与单比特模式、双比特模式以及四比特模式结合使用，也可与 SDR 或 DTR 模式相结合。

器件的大小在 DEVSIZE[4:0] (OCTOSPI_DCR1[20:16]) 中指定，指定的值应能够反映外部器件的总容量，即单个组件容量的 2 倍。

如果地址 X 为偶数，OCTOSPI 赋给地址 X 的字节是存放于器件 A 的地址 X/2 中的字节，OCTOSPI 赋给地址 X+1 的字节是存放于器件 B 的地址 X/2 中的字节。也就是说，偶地址中的字节存储于器件 A，奇地址中的字节存储于器件 B。

在双片四线模式下读取器件的状态寄存器时，需要读取的字节数是常规模式下的 2 倍。这意味着在状态寄存器获取指令到达后，如果每个器件给出八个有效位，则 OCTOSPI 必须配置为 2 个字节（16 位）的数据长度，它将从每个器件接收 1 个字节。

如果每个器件给出一个 16 位的状态，则 OCTOSPI 必须配置为读取四字节，以在双片四线模式下可获取两个器件的所有状态位。结果（在数据寄存器中）的最低有效字节是器件 A 的状态寄存器的最低有效字节，而下一个字节是器件 B 的状态寄存器的最低有效字节。数据寄存器的第三个字节是器件 A 的第二个字节，第四个字节是器件 B 的第二个字节（器件具有 16 位状态寄存器时）。

偶数个字节必须始终在双片四线模式下访问。因此，若 DQM = 1，则数据长度位域 (OCTOSPI_DLR[0]) 的位 0 始终保持为 1。

在双片四线模式下，器件 A 的接口信号的行为基本上与正常模式下相同。在指令、地址、交替字节以及空指令周期阶段，器件 B 的接口信号具有与器件 A 的接口信号完全相同的波形。也就是说，每个器件总是接收相同的指令与地址。然后，在数据阶段，AIOx 和 BIOx 总线并行传输数据，但发送到器件 A（或从其接收）的数据与器件 B 中的不同。

19.2.3 HyperBus™ 模式

在 HyperBus™ 模式下，OCTOSPI 可使用 HyperBus™ 协议与外部器件通信。

HyperBus™ 根据具体的工作电压选择使用 11 到 12 个引脚：

- IO[7:0] 用作双向数据总线
- RWDS 用于读写数据选通和延迟插入（映射到 DQS 引脚）
- nCS
- CLK

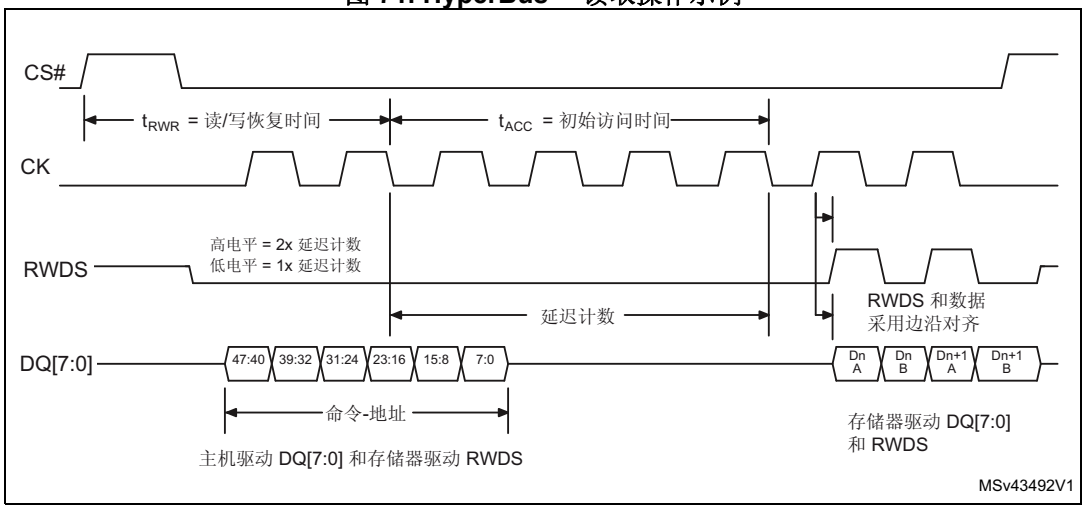
HyperBus™ 不需要任何命令规范，也不需要任何交替字节，因此使用单独的寄存器组来定义事务的时序。

HyperBus™ 帧由两个阶段组成：

- 命令/地址阶段
- 数据阶段

nCS 在每个事务开始前下降，在每个事务完成后再次上升。

图 71. HyperBus™ 读取操作示例



特定的 HyperBus™ 功能通过寄存器在 0x0200-0x02FC 偏移范围内进行配置。

命令/地址阶段

在此初始阶段，OCTOSPI 通过 IO[7:0] 发送 48 位，以指定将与外部器件之间执行的操作。

表 108. 命令/地址阶段说明

CA 位	位名	说明
47	R/W#	将事务标识为读取操作或写入操作
46	地址空间	指示事务正在访问存储器空间还是寄存器空间
45	突发类型	指示突发是线性还是回卷
44-16	行和高位列地址	选择行和高位列地址
15-3	保留	-
2-0	低位列地址	选择半页内的起始 16 位字

地址空间通过 OCTOSPI 器件配置寄存器 (OCTOSPI_DCR1) 的存储器类型 (MTYP) 位域进行配置。

器件的总大小在 OCTOSPI 器件配置寄存器 (OCTOSPI_DCR1) 的器件大小 (DEVSIZE) 位域中进行配置。对于 MCP，器件大小为 MCP 所有尺寸的所有晶片的总和。

具有初始延迟的读写操作

HyperBus™ 读写操作需要遵循两个时序：

- 器件最短读/写恢复时间 t_{RWR}
- 器件访问时间 t_{ACC}

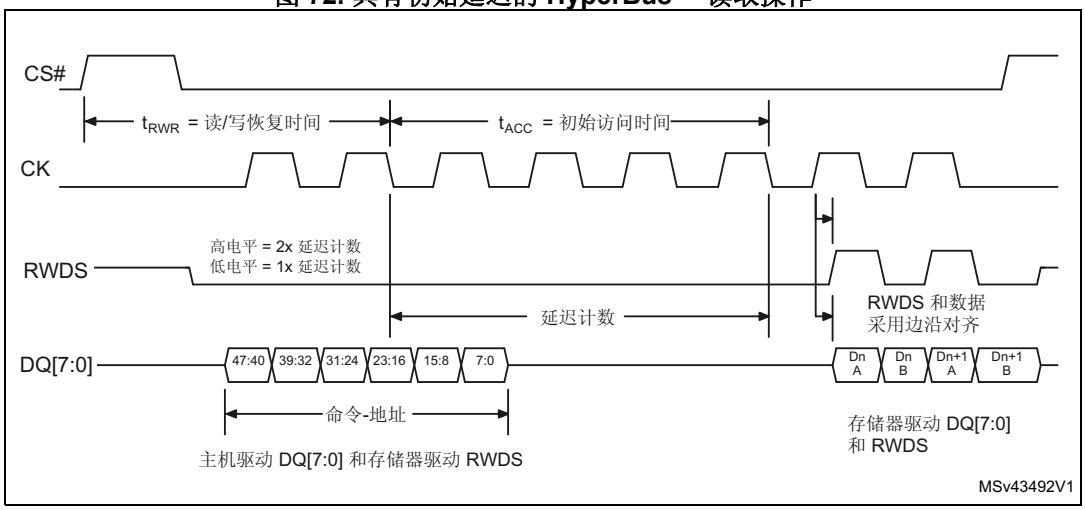
t_{RWR} 在 OCTOSPI HyperBus™ 延迟配置寄存器 (OCTOSPI_HLCR) 的 TRWR 位域中定义。

t_{ACC} 在 OCTOSPI HyperBus™ 延迟配置寄存器 (OCTOSPI_HLCR) 的 TACC 位域中定义。

在读取操作期间，RWDS 有以下两种用途：

- 供器件在命令/地址阶段请求额外延迟。
- 供设备在数据阶段进行数据选通。

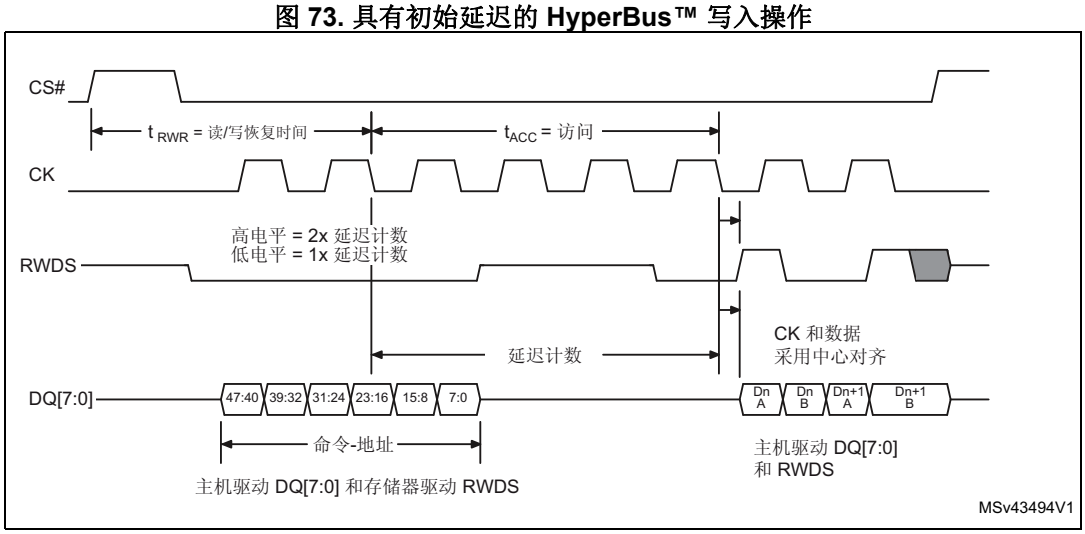
图 72. 具有初始延迟的 HyperBus™ 读取操作



在写入操作期间，RWDS 有以下两种用途：

- 供器件在命令/地址阶段请求额外延迟。
- 供 OCTOSPI 在数据阶段进行写入数据屏蔽。

图 73. 具有初始延迟的 HyperBus™ 写入操作



具有额外延迟的读写操作

如果器件需要额外延迟（例如，在 SDRAM 的刷新周期期间），则必须在命令/地址阶段将其连接到其中一个 RWDS 信号。

OCTOSPI 增加了一段额外的 t_{ACC} 时间，以满足器件要求。

图 74. 具有额外延迟的 HyperBus™ 读取操作

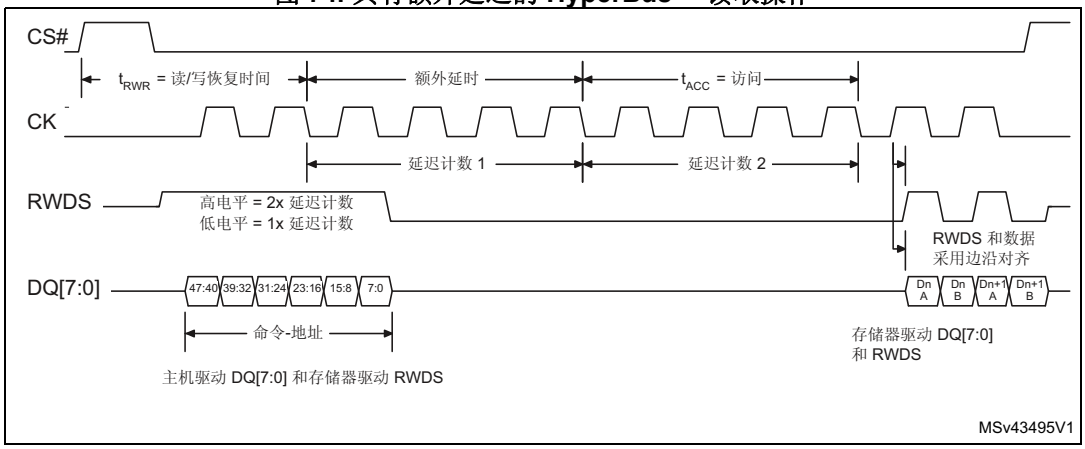
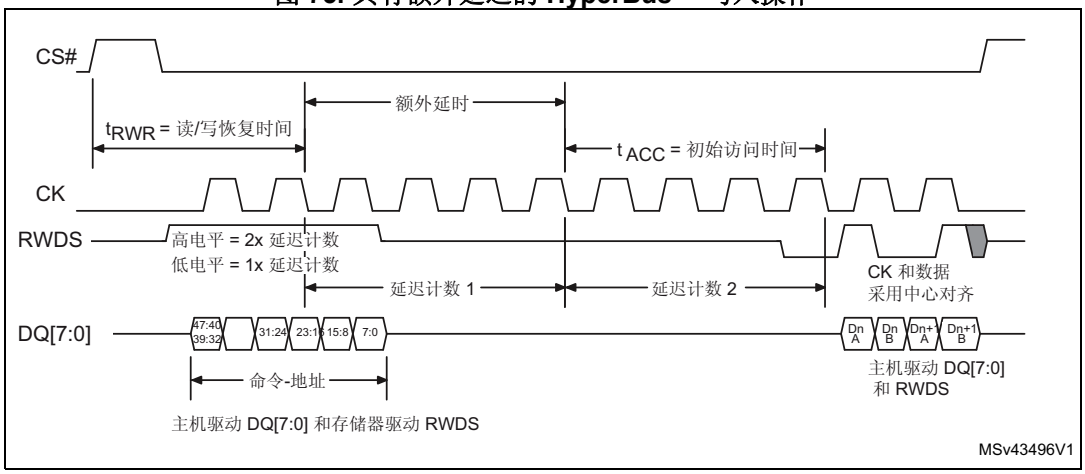


图 75. 具有额外延迟的 HyperBus™ 写入操作



固定延迟模式

一些器件或应用可能不需要上述可变延迟。

通过将 OCTOSPI HyperBus™ 延迟配置寄存器 (OCTOSPI_HLCR) 的延迟模式 (LM) 位置 1，可以将延迟强制设为固定值 2 x t_{ACC}。

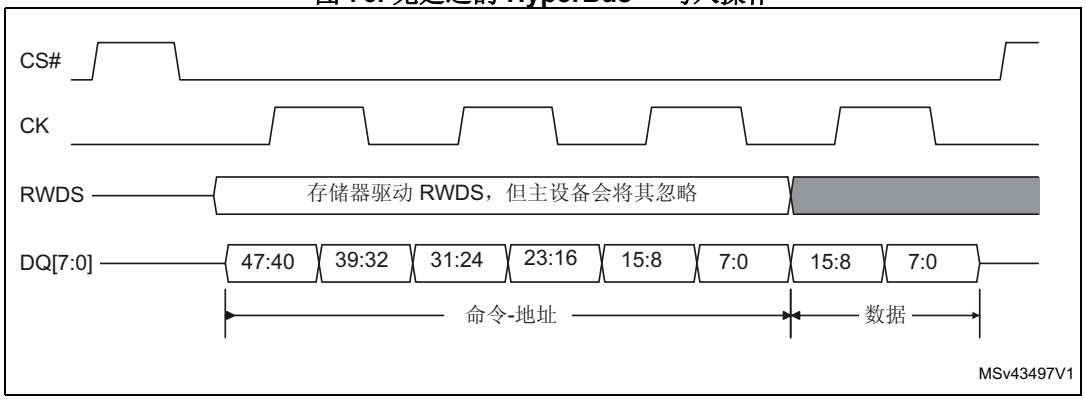
在此固定延迟模式下，OCTOSPI 不考虑 RWDS 信号的状态，并且始终添加一段延迟时间为固定值 2 x t_{ACC} 的额外延迟。

无延迟的写入操作

一些器件可能还要求写入操作的延迟为零。

通过将 OCTOSPI HyperBus™ 延迟配置寄存器 (OCTOSPI_HLCR) 的写入零延迟 (WZL) 位置 1，可以强制设为零写入延迟。

图 76. 无延迟的 HyperBus™ 写入操作

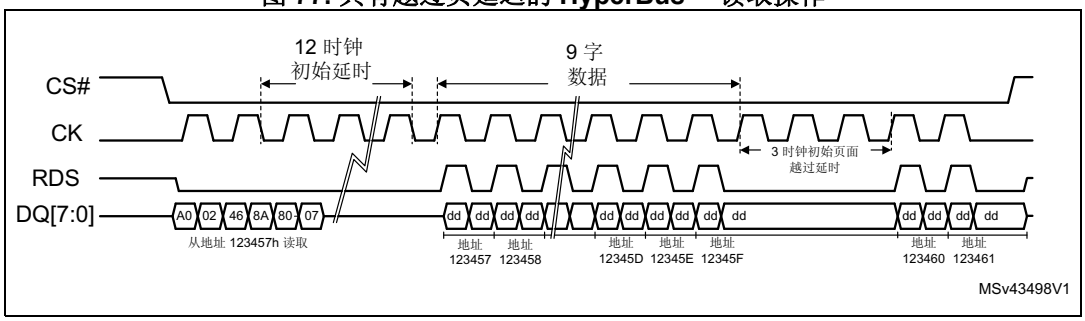


读取操作期间的越过页延迟

一些器件在越过页时可能需要额外延迟来执行读取操作。

对于任何页访问，都必须遵循初始延迟，因此若第一次访问接近页边界，则在越过页时自动添加延迟以遵循 t_{ACC} 时间。

图 77. 具有越过页延迟的 HyperBus™ 读取操作



19.2.4 常规命令模式与 HyperBus™ 的通用功能

OCTOSPI 支持一些对常规命令模式和 HyperBus™ 通用的特定功能，例如：

- 回卷支持
- CS 边界和调整

回卷支持

OCTOSPI 支持 HyperBus™ 协议所定义的混合回卷。在常规命令模式下，同样也支持混合回卷。

在混合回卷中，事务可通过增量访问的方式在初始回卷之后继续。

目标存储器支持的回卷大小在 OCTOSPI 器件配置寄存器 2 (OCTOSPI_DCR2) 的回卷大小 (WRAPSIZE) 位域中进行配置。

如果目标存储器不支持混合回卷，则必须将 WRAPSIZE 位域设置为 0。

CS 边界、刷新和通信调整功能

可通过激活三种过程来调整 OCTOSPI 事务：

- CS 边界
- 刷新
- 通信调整

CS 边界功能会将事务限制在已对齐地址的边界。要对齐的地址的大小在八线 SPI 器件配置寄存器 3 (OCTOSPI_DCR3) 的 CS 边界 (CSBOUND) 位域中进行配置，其值等于 $2^{CSBOUND}$。

例如，如果 CSBOUND = 4，则将边界设置为 2^4 = 16 字节。因此，每当 LSB 地址等于 0xF 时，以及每当发出新事务以寻址下一个数据时，都会释放 nCS。

如果 CSBOUND 为零，则禁用该功能，建议该值最小为 3。

由于编程 Flash 的其他页需要使用命令，因此 CSBOUND 功能无法用于写入模式下的 Flash 器件。

刷新功能会将事务的持续时间限制为 OCTOSPI 器件配置寄存器 4 (OCTOSPI_DCR4) 的 REFRESH 位域中编程的值。持续时间以周期数表示。这样可使外部 RAM 定期执行其内部刷新操作。

刷新值必须大于最小的事务大小（以周期数表示，包括命令/地址/交替/空指令阶段）。

通信调整功能会将事务的最大长度限制为 OCTOSPI 器件配置寄存器 3 (OCTOSPI_DCR3) 的最大传输 (MAXTRAN) 位域中编程的值。

时钟周期数达到 MAXTRAN+1 后，如果第二个 OCTOSPI 请求访问，则释放 nCS，并发出新事务以寻址下一个数据。如果第二个 OCTOSPI 没有请求访问，则事务不停止，且不释放 nCS。

如果 MAXTRAN 为零，则没有限制。

MAXTRAN 值必须大于最小的事务大小（以周期数表示，包括命令/地址/交替/空指令阶段）。

由于编程 Flash 的其他页需要使用命令，因此 MAXTRAN 功能无法用于写入模式下的 Flash 器件。

如果 CS 边界、刷新和通信调整这三种同时使能，则在第一次满足条件时即释放 nCS。

通信调整功能

可通过 CS 边界功能调整 OCTOSPI 事务：

CS 边界功能会将事务限制在已对齐地址的边界。要对齐的地址的大小在 OCTOSPI 器件配置寄存器 3 (OCTOSPI_DCR3) 的 CS 边界 (CSBOUND) 位域中进行配置，其值等于 $2^{CSBOUND}$。

例如，如果 CSBOUND = 4，则将边界设置为 2^4 = 16 字节。因此，每当 LSB 地址等于 0xF 时，以及每当发出新事务以寻址下一个数据时，都会释放 nCS。

如果 CSBOUND 为零，则禁用该功能，建议该值最小为“3”。

由于编程 Flash 的其他页需要使用命令，因此 CSBOUND 功能无法用于写入模式下的 Flash 器件。

19.3 OCTOSPI 工作模式

OCTOSPI 有三种工作模式（无论使用的低层协议是常规协议还是 HyperBus™ 协议）

- 间接模式（读或写）
- 状态轮询模式
- 内存映射模式

19.3.1 OCTOSPI 间接模式

在间接模式下，通过写入 OCTOSPI 寄存器来触发命令；并通过读写数据寄存器来传输数据，与对待其他通信外设相似。

若 FMODE = 00 (OCTOSPI_CR[29:28])，则 OCTOSPI 处于间接写入模式，字节在数据阶段中发送到外部器件。通过写入数据寄存器 (OCTOSPI_DR) 的方式提供数据。

若 FMODE = 01，则 OCTOSPI 处于间接读取模式，在数据阶段中从外部器件接收字节。通过读取 OCTOSPI_DR 来获取数据。

在间接模式下，当通过 8 个通道在 DTR 模式下配置 OCTOSPI 时，给定的起始地址必须为偶数。

读取/写入的字节数在数据长度寄存器 (OCTOSPI_DLR) 中指定。如果 OCTOSPI_DLR = 0xFFFF_FFFF（全为“1”），则数据长度视为未定义，OCTOSPI 将继续传输数据，直到到达（由 DEVSIZE 定义的）外部器件的结尾。如果不传输任何字节，DMODE (OCTOSPI_CCR[26:24]) 应设置为 00。

如果 OCTOSPI_DLR = 0xFFFF_FFFF 并且 DEVSIZE=0x1F（最大值指示 4 GB 的器件），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 OCTOSPI 被禁止后停止。在读取最后一个存储器地址后（地址为 0xFFFF_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 OCTOSPI_DCR1 寄存器中定义的器件大小，在达到外部 SPI 存储器的限制时，TCF 置 1。

在常规命令模式下触发传输启动

根据 OCTOSPI 的配置，当使用常规模式时，有三种在间接模式下触发传输启动的方式。通常，在固件给出命令所需的最后一点信息时，即会触发传输。更具体来讲，满足以下条件时即启动在间接模式下的通信：

- 对 INSTRUCTION[31:0] (OCTOSPI_IR) 执行写入操作，如果没有地址是必需的（当 ADMODE = 000）并且不需要固件提供数据（当 FMODE = 01 或 DMODE = 000）
- 对 ADDRESS[31:0] (OCTOSPI_AR) 执行写入操作，如果地址是必需的（当 ADMODE != 000）并且不需要固件提供数据（当 FMODE = 01 或 DMODE = 000）
- 对 DATA[31:0] (OCTOSPI_DR) 执行写入操作，如果地址是必需的（当 ADMODE != 000）并且不需要固件提供数据（当 FMODE = 00 并且 DMODE!=000）

写入交替字节寄存器 (OCTOSPI_ABR) 始终不会触发通信启动。如果需要交替字节，必须预先进行编程。

如果命令启动，BUSY 位 (OCTOSPI_SR[5]) 将自动置 1。

在 HyperBus™ 模式下触发传输启动

根据 OCTOSPI 的配置，在间接模式下有三种触发命令启动的方式。通常，在固件给出命令所需的最后一点信息时，即会触发传输。更具体来讲，满足以下条件时即启动在间接模式下的通信：

- 在间接读取模式下（当 FMODE = 01 时）对 ADDRESS[31:0] (OCTOSPI_AR) 执行写入操作
- 在间接写入模式下（当 FMODE = 00 时）对 DATA[31:0] (OCTOSPI_DR) 执行写入操作

如果传输启动，BUSY 位 (OCTOSPI_SR[5]) 将自动置 1。

FIFO 和数据管理

在间接模式中，数据将通过 OCTOSPI 内部的一个 32 字节 FIFO。FLEVEL[5:0] (OCTOSPI_SR[12:8]) 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 OCTOSPI_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 个字节，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 OCTOSPI_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[4:0] 用于定义一个 FIFO 阈值，如果超过该阈值，FIFO 阈值标志 FTF 将置 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。从外部器件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 DMA 置 1 时启动 DMA 传输。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 已满时，OCTOSPI 将暂时停止从外部器件读取字节以避免上溢。请注意，只有在 FIFO 中的 4 个字节为空 (FLEVEL <= 28) 时才会重新开始读取外部器件。因此，若 FTHRES >= 29，应用程序必须读取足够的字节以确保 OCTOSPI 再次从外部器件检索数据。否则，只要 FLEVEL < FTHRES，FTF 标志将保持为“0”。

19.3.2 OCTOSPI 状态标志轮询模式

在自动轮询模式下，OCTOSPI 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。

对器件访问的启动方式与在间接读取模式下相同。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] (OCTOSPI_PSMAR) 的内容用于屏蔽来自外部器件的数据。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] (OCTOSPI_PSMAR) 相同，说明存在位 n 匹配。

如果轮询匹配模式位 (PMM, OCTOSPI_CR[23]) 为 0，将激活“AND”匹配模式。这意味着状态匹配标志 (SMF) 仅在全部未屏蔽位均存在匹配时置 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 OCTOSPI (EN = 0) 前继续进行周期性访问。

数据寄存器 (OCTOSPI_DR) 包含最新接收的状态字节（FIFO 禁用）。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。

在自动轮询模式下，不支持可变延迟。因此，必须将存储器配置为固定延迟。

19.3.3 OCTOSPI 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。请注意，即使外部器件容量更大，寻址空间也无法超过 256 MB。

如果访问的地址超出 DEVSIZE 定义的范围但仍在 256 MB 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。

支持芯片内执行 (XIP) 操作，即 OCTOSPI 在完成最近一次访问后会继续将字节加载到地址。如果随后连续访问后续的字节，由于结果已经预取，这一系列访问操作将很快完成。

默认情况下，即便在很长时间内不访问外部器件，OCTOSPI 也不会停止预取操作，它会将之前的读取操作保持激活状态且 nCS 保持低电平，或重新启动新传输。

由于 nCS 保持低电平时，外部器件功耗增加，应用程序可能会激活超时计数器（TCEN=1，OCTOSPI_CR[3]）。从而在一个 TIMEOUT[15:0] (OCTOSPI_LPTR) 周期后，若从时钟无效起在 x 个周期的时长内没有访问，则释放 nCS。

BUSY 在第一次内存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。

19.4 OCTOSPI 配置

OCTOSPI 配置分三步：

1. OCTOSPI 系统配置
2. OCTOSPI 器件配置
3. OCTOSPI 模式配置

19.4.1 OCTOSPI 系统配置

OCTOSPI 使用 OCTOSPI_CR 进行配置。用户必须编程：

- 功能模式（FMODE 位域）
- 轮询模式行为（根据需要，PMM 和 APMS 位）
- 生成 DMA 请求或生成中断的 FIFO 电平（FTHRES 位域）
- 使用 DMA（DMAEN 位）
- 使用超时计数器（TCEN 位）
- 双片四线模式（根据需要，DQM 位）（仅适用于四线 SPI 配置）

若是用于中断，则相关使能位也可在该阶段置 1。

如果使用超时计数器，则在 OCTOSPI_LPTR 寄存器中编程超时值。

在 OCTOSPI 配置期间不得使能 DMA 通道，只有在操作完全配置后才能使能 DMA 通道，以避免生成任何意外请求。

19.4.2 OCTOSPI 器件配置

通过 OCTOSPI_DCR1 和 OCTOSPI_DCR2 寄存器配置与目标外部器件相关的参数。用户必须编程：

- 存储器类型（MTYP 位域）
- 器件大小（DEVSIZE 位域）
- 片选最短高电平时间（CSHT 位域）
- 时钟模式（FRCK 和 CKMODE 位）
- 支持的回卷大小（WRAPSIZE 位域）
- 器件频率（PRESCALER 位域）
- MTYP[2:0] 位域定义用于 8 线模式的存储器类型
- 在 8 数据位模式下按 D0/D1 排序的 Micron 模式 (DMODE = 100)
- 在 8 数据位模式下按 D1/D0 排序的 Macronix 模式 (DMODE = 100)
- HyperBus™ 存储器模式。协议遵循 Spansion 的 HyperBus™ 规范，必须选择 8 数据位 DDR 模式
- HyperBus™ 寄存器模式。寻址寄存器空间，此模式下的内存映射访问必须是不可缓存的，或者必须使用间接读取/写入模式

DEVSIZE[4:0] 位域使用下面的公式定义外部存储器的大小：

器件中的字节数 = $2^{[DEVSIZE+1]}$

DEVSIZE+1 是对外部器件寻址所需的地址位数。在间接模式下，外部器件容量最高可达 4 GB（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 256 MB。

如果 DQM = 1，则 DEVSIZE 表示两个器件容量的总和。

OCTOSPI 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。

如果外部器件需要命令之间的时间更长，可使用片选高电平时间 (CSHT) 位域指定 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 电平（nCS = 1 时）。

在 HyperBus™ 模式下，必须在 OCTOSPI HyperBus™ 延迟配置寄存器 (OCTOSPI_HLCR) 中配置器件时序（t_{ACC} 和 t_{RWR}）和延迟模式。

19.4.3 OCTOSPI 常规命令模式配置

间接模式配置

FMODE 编程为 00 可选择间接写入模式，将数据发送到外部器件。FMODE 编程为 01 可选择间接读取模式，读取外部器件中的数据。

OCTOSPI 用于间接模式时，采用以下方式构建帧：

1. 在 OCTOSPI_DLR 中指定待读取或写入的字节数
2. 在 OCTOSPI_TCR 中指定帧时序
3. 在 OCTOSPI_CCR 中指定帧格式
4. 在 OCTOSPI_IR 中指定指令
5. 在 OCTOSPI_ABR 中指定要在地址阶段后立即发送的可选交替字节
6. 在 OCTOSPI_AR 中指定目标地址
7. 如果需要，使能 DMA 通道
8. 通过 OCTOSPI_DR 从 FIFO 读取数据/向 FIFO 写入数据（如果未使用 DMA）

如果无需为某个命令更新地址寄存器 (OCTOSPI_AR) 与数据寄存器 (OCTOSPI_DR)，则在写入 OCTOSPI_CCR 时，该命令序列便立即启动。在 ADMODE 和 DMODE 均为 000 时，或在间接读取模式 (FMODE = 01) 下仅 ADMODE = 000 时，便属于此情况。

在需要地址（ADMODE 不为 000），但无需写入数据寄存器（FMODE = 01 或 DMODE = 000）时，通过写入 OCTOSPI_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 != 000）中，通过 OCTOSPI_DR 写入 FIFO 触发通信启动。

状态标志轮询模式配置

将 FMODE 位域 (OCTOSPI_CR[29:28]) 设置为 10，使能状态标志轮询模式。在此模式下，发送编程的帧并周期性检索数据。

每帧中读取的最大数据量为 4 字节。如果 OCTOSPI_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 OCTOSPI_PIR 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 SMF（状态匹配标志）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 OCTOSPI_PSMKR 中的值进行屏蔽，并与存储在 OCTOSPI_PSMAR 中的值进行或运算或与运算。

若是存在匹配，则 SMF 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 OCTOSPI 自动停止。在任何情况下，最新的检索值都在 OCTOSPI_DR 中可用。

OCTOSPI 用于自动轮询模式时，采用以下方式构建帧：

1. 在 OCTOSPI_PSMKR 中指定输入屏蔽
2. 在 OCTOSPI_PSMAR 中指定比较值
3. 在 OCTOSPI_PIR 中指定读取周期
4. 在 OCTOSPI_DLR 中指定待读取的数据字节数
5. 在 OCTOSPI_TCR 中指定帧时序
6. 在 OCTOSPI_CCR 中指定帧格式
7. 在 OCTOSPI_IR 中指定指令
8. 在 OCTOSPI_ABR 中指定要在地址阶段后立即发送的可选交替字节
9. 在 OCTOSPI_AR 中指定可选目标地址

如果无需为某个命令更新地址寄存器 (OCTOSPI_AR)，则在写入 OCTOSPI_CCR 时，该命令序列便立即启动。这是 ADMODE = 000 时的情况。

在需要地址（ADMODE 不为 000）时，通过写入 OCTOSPI_AR 更新地址后，命令序列便立即启动。

内存映射模式配置

在内存映射模式下，外部器件被视为内部存储器，只是存在访问延迟。在该模式下，允许对外部器件执行读写操作。

由于必须轮询擦除或编程状态的内部标志，因此不建议使用内存映射写入进行 Flash 编程。

如果存储器不带写入选通或者处于双片四线模式，则不支持内存映射写入操作。

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

当 AHB 主器件访问内存映射空间时，将发送已编程的指令和帧。

FIFO 用作预取缓冲区以接受所有线性读取。在此模式中，对于 OCTOSPI_DR 的任何访问均返回零。

数据长度寄存器 (OCTOSPI_DLR) 在内存映射模式中无意义。

OCTOSPI 用于内存映射模式时，采用以下方式构建帧：

1. 在 OCTOSPI_TCR 中指定读取操作的帧时序
2. 在 OCTOSPI_CCR 中指定读取操作的帧格式
3. 在 OCTOSPI_IR 中指定指令
4. 在 OCTOSPI_ABR 中指定读取操作要在地址阶段后立即发送的可选交替字节
5. 在 OCTOSPI_WTCR 中指定写入操作的帧时序
6. 在 OCTOSPI_WCCR 中指定写入操作的帧格式
7. 在 OCTOSPI_WIR 中指定指令
8. 在 OCTOSPI_WABR 中指定读取操作要在地址阶段后立即发送的可选交替字节

所有配置操作必须在首次访问存储器区域之前完成。第一次访问时，OCTOSPI 繁忙，不允许进一步配置。

OCTOSPI 延迟数据采样

默认情况下，不使用 DQS 时，OCTOSPI 在外部器件驱动信号后过半个 CLK 周期才对外部器件驱动的数据采样。

在外部信号延迟时，这有利于推迟数据采样。使用 SSHIFT (OCTOSPI_TCR[30]) 位，可将数据采样移位半个 CLK 周期。

当数据阶段配置为 DTR 模式 (DDTR = 1) 时，固件必须清零 SSHIFT。

指令仅发送一次

Flash（例如 Winbond）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 (OCTOSPI_CCR[31]) 可利用此类功能的优势。

SIOO 仅在间接模式下有效。如果 SIOO 位置 1，则在对 OCTOSPI_CCR 执行写入操作后，仅针对第一条命令发送指令。

后续命令序列都将跳过指令阶段，直到 OCTOSPI_CCR 被写入为止。在 IMODE = 00（无指令）时，SIOO 不起作用。

19.4.4 OCTOSPI HyperBus™ 模式配置

间接模式配置

FMODE 编程为 00 可选择间接写入模式，将数据发送到外部器件。FMODE 编程为 01 可选择间接读取模式，读取外部器件中的数据。

OCTOSPI 用于间接模式时，采用以下方式构建帧：

1. 在 OCTOSPI_DLR 中指定待读取或写入的字节数
2. 在 OCTOSPI_AR 中指定目标地址
3. 如果需要，使能 DMA 通道
4. 通过 OCTOSPI_DR 从 FIFO 读取数据/向 FIFO 写入数据（如果未使用 DMA）

在间接读取模式下（当 FMODE = 01 时），通过写入 OCTOSPI_AR 更新地址后，命令序列便立即启动。

在间接写入模式下（当 FMODE = 00 时），通过 OCTOSPI_DR 写入 FIFO 触发通信启动。

状态标志轮询模式配置

将 FMODE 位域 (OCTOSPI_CR[29:28]) 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索数据。

每帧中读取的最大数据量为 4 字节。如果 OCTOSPI_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 OCTOSPI_PIR 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 SMF 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 OCTOSPI_PSMKR 中的值进行屏蔽，并与存储在 OCTOSPI_PSMAR 中的值进行或运算或与运算。

若是存在匹配，则 SMF 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 OCTOSPI 自动停止。

在任何情况下，最新的检索值都在 OCTOSPI_DR 中可用。

OCTOSPI 用于自动轮询模式时，采用以下方式构建帧：

1. 在 OCTOSPI_PSMKR 中指定输入屏蔽
2. 在 OCTOSPI_PSMAR 中指定比较值
3. 在 OCTOSPI_PIR 中指定读取周期
4. 在 OCTOSPI_DLR 中指定待读取的数据字节数
5. 在 OCTOSPI_AR 中指定目标地址

通过写入 OCTOSPI_AR 更新地址后，命令序列便立即启动。

内存映射模式配置

在内存映射模式下，外部器件被视为内部存储器，只是存在访问延迟。在该模式下，允许对外部器件执行读写操作。

将 OCTOSPI_CR 寄存器中的 FMODE 设置为 11 可进入内存映射模式。当 AHB 主器件访问内存映射空间时，将发送已编程的指令和帧。

FIFO 用作预取缓冲区以接受所有线性读取。在此模式中，对于 OCTOSPI_DR 的任何访问均返回零。

数据长度寄存器 (OCTOSPI_DLR) 在内存映射模式中无意义。

所有配置操作必须在首次访问存储器区域之前完成。第一次访问时，OCTOSPI 繁忙，不允许进行配置。

19.4.5 OCTOSPI 错误管理

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 OCTOSPI_AR 中编程了错误的地址（根据 OCTOSPI_DCR1 中 DEVSIZE[4:0] 定义的器件大小）。TEF 将置 1，如果使能，还将产生中断。
- 在间接模式下，如果地址加数据的长度超过器件大小，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，将产生 AHB 错误，以响应故障 AHB 请求。
- 当内存映射模式被禁止时访问内存映射区域，将产生 AHB 错误，以响应故障 AHB 请求。

19.4.6 OCTOSPI 的繁忙位和中止功能

在 OCTOSPI 启动对外部器件的操作时，OCTOSPI_SR 中的 BUSY 位自动置 1。

在间接模式下，在 OCTOSPI 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 OCTOSPI_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 清空。

在将 ABORT 位置 1 之前，软件必须确保使用同步屏障完成所有当前事务。

请注意，如果中止对状态寄存器的写入操作，有些器件可能发生错误行为。

19.4.7 OCTOSPI 重新配置或禁用

在重新配置 OCTOSPI 之前，软件必须确保完成所有事务：

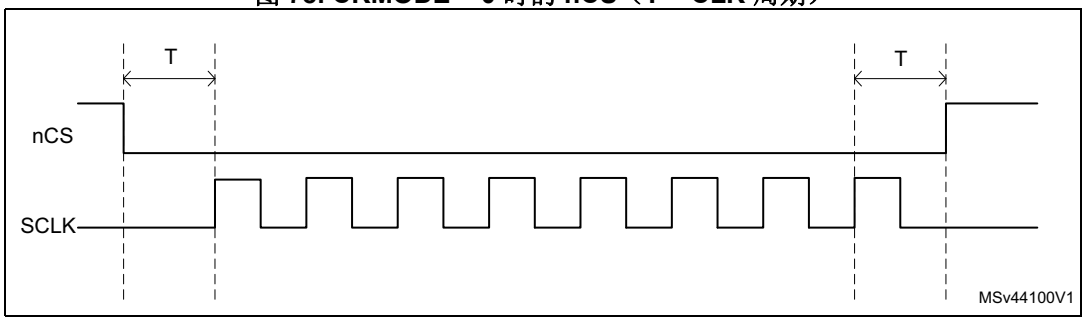
- 内存映射写入后，软件应执行虚拟读取，然后执行同步屏障，最后执行中止。
- 内存映射读取后，软件应执行同步屏障，然后执行中止。

19.4.8 nCS 行为

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

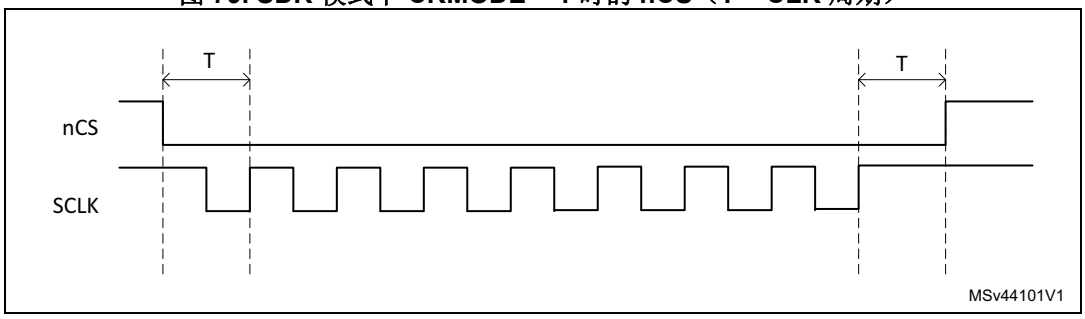
当 CKMODE = 0（“模式 0”，在未进行任何操作时 CLK 保持低电平）时，nCS 在操作首次升高 CLK 边沿时的一个 CLK 周期前降至低电平，在操作最后一次升高 CLK 边沿时的一个 CLK 周期后升至高电平，如*图 78* 所示。

图 78. CKMODE = 0 时的 nCS（T = CLK 周期）



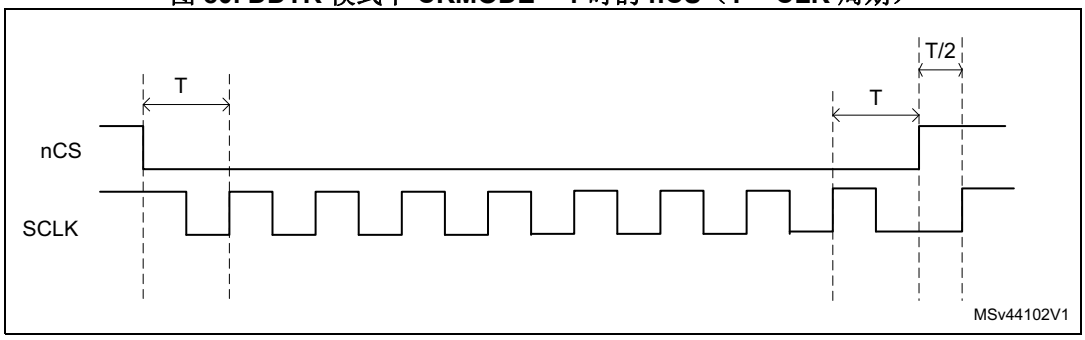
当 CKMODE = 1（“模式 3”，在未进行任何操作时 CLK 升至高电平）以及在 SDR 模式下时，nCS 在操作首次升高 CLK 边沿时的一个 CLK 周期前降至低电平，在操作最后一次升高 CLK 边沿时的一个 CLK 周期后升至高电平，如*图 79* 所示。

图 79. SDR 模式下 CKMODE = 1 时的 nCS（T = CLK 周期）



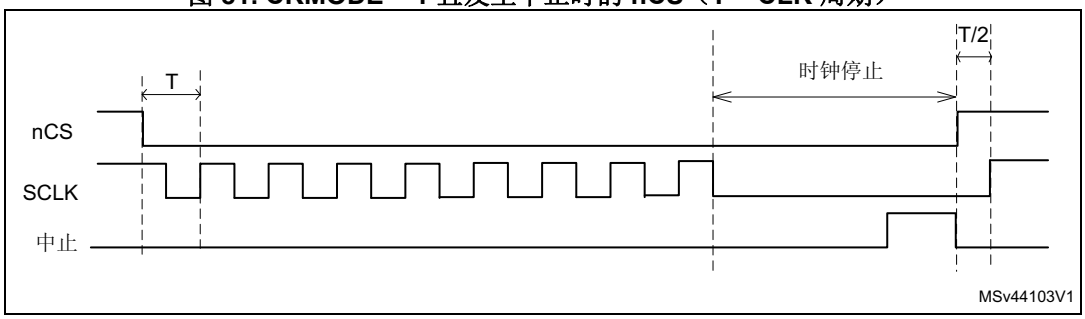
当 CKMODE = 1（“模式 3”）且 DDTR = 1（数据 DTR 模式）时，nCS 在操作首次升高 CLK 边沿时的一个 CLK 周期前降至低电平，在操作最后一次升高 CLK 边沿时的一个 CLK 周期后升至高电平，如*图 80* 所示。由于 DDR 操作必须伴随下降沿完成，当 nCS 变为高电平时，CLK 为低电平并在经过半个周期后恢复高电平。

图 80. DDTR 模式下 CKMODE = 1 时的 nCS（T = CLK 周期）



若 FIFO 在读取操作期间保持写满状态或在写入操作期间保持为空，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 CLK 周期后升至高电平，如*图 81* 所示。

图 81. CKMODE = 1 且发生中止时的 nCS（T = CLK 周期）



不处于双片四线模式 (DQM = 0) 时，仅访问器件 A，因此 BnCS 保持高电平。在双片四线模式下，BnCS 与 AnCS 的行为完全相同。因此，如果存在器件 B 并且应用程序始终处于双片四线模式，则器件 B 可使用 AnCS，而 BnCS 引脚输出可用于其他功能。

19.5 OCTOSPI 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供更大的灵活性。

表 109. OCTOSPI 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

19.6 OCTOSPI 寄存器

19.6.1 OCTOSPI 控制寄存器 (OCTOSPI_CR)

OCTOSPI 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
Res.		FMODE		Res.				PMM	APMS	Res.	TOIE	SMIE	FTIE	TCIE	TEIE
		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.			FTHRES[4:0]					FSEL	DQM	Res.		TCEN	DMAEN	ABORT	EN
			rw					rw	rw			rw	w1s	rw	w1s

位 31:30 保留，必须保持复位值

位 29:28 **FMODE[1:0]**：功能模式 (Functional mode)

该位域定义 OCTOSPI 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 时写入该位域

位 27:24 保留，必须保持复位值

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，如果从器件接收的所有未屏蔽位均与匹配寄存器中的对应位相匹配，则 SMF 置 1

1：OR 匹配模式，如果从器件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，则 SMF 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 OCTOSPI 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**：超时中断使能 (Timeout interrupt enable)

该位使能超时中断

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，OCTOSPI_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 select)
该位选择单 Flash 模式 (DFM = 0) 下单线/双线/四线模式中要寻址的 Flash
0：选择 FLASH 1（通过 IO[3:0] 交换数据）
1：选择 FLASH 2（通过 IO[3:0] 交换数据）
当 DFM = 1 或选择八线模式时，忽略该位
仅可在 BUSY = 0 时修改该位

位 6 **DQM**：双片四线模式 (Dual-quad mode)
该位激活双片四线模式，同时使用两个外部器件以将吞吐量和容量扩大一倍
0：禁止双片四线模式
1：使能双片四线模式
仅可在 BUSY = 0 时修改该位。

位 5:4 保留，必须保持复位值

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。使能超时计数器

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，外部器件持续不活动 TIMEOUT[15:0] 个周期后释放片选

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 OCTOSPI_DR 寄存器可使用 DMA 输入或输出数据。当 FIFO 阈值标志 (FTF) 置 1 时，将触发 DMA 传输

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，该位也用于复位 APM 位或 DM 位

0：不请求中止

1：请求中止

位 0 **EN**：使能 (Enable)

使能 OCTOSPI

0：禁止 OCTOSPI

1：使能 OCTOSPI

19.6.2 OCTOSPI 器件配置寄存器 1 (OCTOSPI_DCR1)

OCTOSPI device configuration register 1

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.					MTYP			Res.			DEVSIZE				
					rw						rw				

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.					CSHT			Res.						FRCK	CK-MODE
					rw									rw	rw

位 31:27 保留，必须保持复位值

位 26:24 **MTYP**：存储器类型 (Memory type)

该位指示将支持的存储器类型

000：Micron 模式，在 DTR 8 数据位模式下按 D0/D1 排序。八线/四线/双线/单线模式下的常规 SPI 协议

001：Macronix 模式，在 DTR 8 数据位模式下按 D1/D0 排序。八线/四线/双线/单线模式下的常规 SPI 协议

010-011：保留

010：APMemory 模式，具有反向 DQS 极性

011：Macronix RAM 模式，在 DTR 8 数据位模式下按 D1/D0 排序。具有专用地址映射的八线/四线/双线/单线模式下的常规 SPI 协议

100：HyperBus™ 存储器模式，协议遵循 Spansion 的 HyperBus™ 规范。必须选择 8 数据位 DDR 模式

101：HyperBus™ 寄存器模式，寻址寄存器空间。此模式下的内存映射访问必须是不可缓存的，或者必须使用间接读取/写入模式

110-111：保留，必须保持复位值

位 23:21 保留，必须保持复位值

位 20:16 **DEVSIZE[4:0]**：器件大小 (Device size)

该位域使用下面的公式定义了外部器件的大小：

器件中的字节数 = $2^{[DEVSIZE+1]}$

DEVSIZE+1 是对外部器件寻址所需的地址位数。在间接模式下，外部器件容量最高可达 4 GB（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 256 MB

在常规命令模式下，如果 DQM = 1，则 DEVSIZE 表示两个器件容量的总和

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) 在发送至外部器件的命令之间必须保持高电平的最少 CLK 周期数

0：nCS 在外部器件命令之间保持高电平至少 1 个周期

1：nCS 在外部器件命令之间保持高电平至少 2 个周期

...

7：nCS 在外部器件命令之间保持高电平至少 8 个周期

位 7:2 保留，必须保持复位值

位 1 **FRCK**：自由运行时钟 (Free running clock)

该位用于配置自由运行时钟

0：CLK 不自由运行

1：CLK 自由运行（始终提供）

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

19.6.3 OCTOSPI 器件配置寄存器 2 (OCTOSPI_DCR2)

OCTOSPI device configuration register 2

偏移地址：0x000C

复位值：0x0000 0000

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

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

位 31:19 保留，必须保持复位值

位 18:16 **WRAPSIZE**：回卷大小 (Wrap size)

该位域指示为存储器配置的回卷大小。如果存储器具有对应于回卷指令的独立命令，该位域指示与 OCTOSPI1_WPIR 寄存器中保存的命令相关联的回卷大小

000：存储器不支持回卷读取

001：保留，必须保持复位值

010：外部存储器支持 16 字节的回卷大小（重要）

011：外部存储器支持 32 字节的回卷大小（最为可能）

100：外部存储器支持 64 字节的回卷大小 (L1 MPU)

101：外部存储器支持 128 字节的回卷大小

110-111：保留，必须保持复位值

位 15:8 保留，必须保持复位值

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

该位域定义基于内核时钟生成 CLK 所用的分频系数（值 + 1）

0：$F_{CLK} = F_{KERNEL}$，AHB 时钟直接用作 OCTOSPI 的 CLK（预分频器被旁路）

1：$F_{CLK} = F_{KERNEL}/2$

2：$F_{CLK} = F_{KERNEL}/3$

...

255：$F_{CLK} = F_{KERNEL}/256$

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

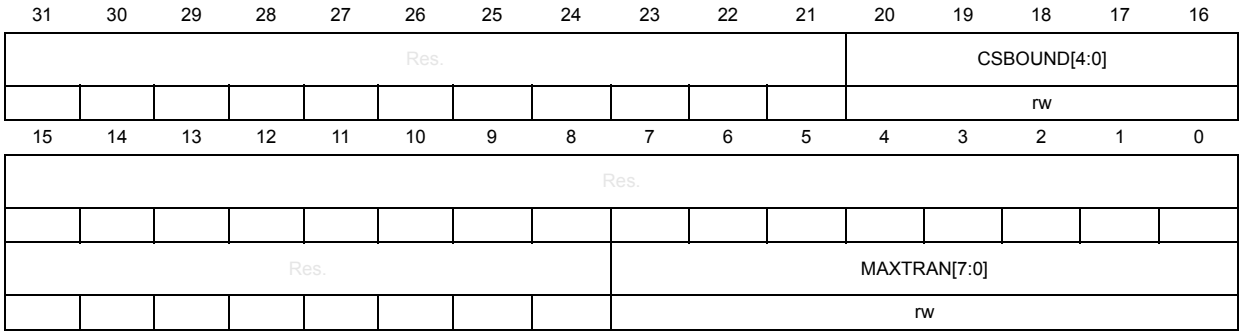
仅可在 BUSY = 0 时修改该位域

19.6.4 OCTOSPI 器件配置寄存器 3 (OCTOSPI_DCR3)

OCTOSPI device configuration register 3

偏移地址：0x0010

复位值：0x0000 0000



位 31:21 保留，必须保持复位值

位 20:16 **CSBOUND[4:0]**：CS 边界 (CS boundary)
该位用于使能事务边界功能。激活时，建议最小值为 3
在每个 $2^{CSBOUND}$ 字节边界释放 nCS
0：禁止 CS 边界
其他值：CS 边界设置为 $2^{CSBOUND}$ 字节

位 15:0 保留，必须保持复位值

位 15:8 保留，必须保持复位值

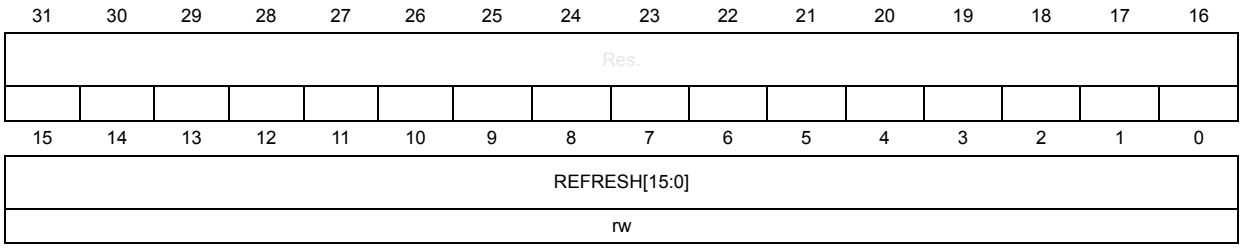
位 7:0 **MAXTRAN[7:0]**：最大传输 (Maximum transfer)
该位用于使能通信调整功能
当另一个 OCTOSPI 请求访问总线时，每 MAXTRAN+1 个时钟周期释放一次 nCS
0：禁止最大通信
其他值：最大通信设置为 MAXTRAN+1 字节

19.6.5 OCTOSPI 器件配置寄存器 4 (OCTOSPI_DCR4)

OCTOSPI device configuration register 4

偏移地址：0x0014

复位值：0x0000 0000



位 31:16 保留，必须保持复位值

位 15:0 **REFRESH[15:0]**：刷新速率 (Refresh rate)
该位用于使能刷新速率功能
每 REFRESH+1 个时钟周期释放一次 nCS
0：禁止刷新
其他值：最大通信长度设置为 REFRESH+1 个时钟周期

19.6.6 OCTOSPI 状态寄存器 (OCTOSPI_SR)

OCTOSPI status 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
Res.															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.		FLEVEL[5:0]						Res.		BUSY	TOF	SMF	FTF	TCF	TEF
		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)
该位域给出 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
在对外部器件的操作完成并且 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 1
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 (OCTOSPI_PSMAR) 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，或从外部器件读取完成后，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，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 complete 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 可将该位清零

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

19.6.7 OCTOSPI 标志清零寄存器 (OCTOSPI_FCR)

OCTOSPI flag clear 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
Res.															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.											CTOF	CSMF	Res.	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 OCTOSPI_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 OCTOSPI_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**：清零传输完成标志 (Clear transfer complete flag)
写入 1 可将 OCTOSPI_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 OCTOSPI_SR 寄存器中的 TEF 标志清零

19.6.8 OCTOSPI 数据长度寄存器 (OCTOSPI_DLR)

OCTOSPI data length register

偏移地址：0x0040

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															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DL[15:0]															
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，OCTOSPI 将继续传输数据直到到达由 DEVSIZE 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

...

0xFFFF_FFFD：传输 4,294,967,294 (4G-2) 个字节

0xFFFF_FFFE：传输 4,294,967,295 (4G-1) 个字节

0xFFFF_FFFF：未定义长度；传输所有字节直到到达由 DEVSIZE 定义的外部器件的结尾。如果 DEVSIZE = 0x1F，则读取无限继续下去

在双片四线模式 (DQM = 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

该位域在内存映射模式 (FMODE = 10) 下不起作用

仅可在 BUSY = 0 时写入该位域

19.6.9 OCTOSPI 地址寄存器 (OCTOSPI_AR)

OCTOSPI address register

偏移地址：0x0048

复位值：0x0000 0000

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

位 31:0 **ADDRESS[31:0]**：地址 (Address)

指定要发送到外部器件的地址

在 Hyperflash ™ 模式下，该位域必须为偶数，因为该协议面向 16 位字

BUSY = 0 或 FMODE = 11（内存映射模式）时，将忽略写入该位域

19.6.10 OCTOSPI 数据寄存器 (OCTOSPI_DR)

OCTOSPI data register

偏移地址：0x0050

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

位 31:0 **DATA[31:0]**：数据 (Data)

指定要与外部 SPI 器件交换的数据。

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外部器件，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）已从外部器件接收的数据。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外部器件读取的数据（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 DATA[7:0] 而半字读取必须读取 DATA[15:0]。

19.6.11 OCTOSPI 轮询状态屏蔽寄存器 (OCTOSPI _PSMKR)

OCTOSPI polling status mask register

偏移地址：0x0080

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															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MASK[15:0]															
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 时写入该位域

19.6.12 OCTOSPI 轮询状态匹配寄存器 (OCTOSPI _PSMAR)

OCTOSPI polling status match register

偏移地址：0x0088

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															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MATCH[15:0]															
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 时写入该位域

19.6.13 OCTOSPI 轮询间隔寄存器 (OCTOSPI _PIR)

OCTOSPI polling interval register

偏移地址：0x0090

复位值：0x0000 0000

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

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

位 31:16 保留，必须保持复位值

位 15:0 **INTERVAL[15:0]**：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 时写入该位域

19.6.14 OCTOSPI 通信配置寄存器 (OCTOSPI_CCR)

OCTOSPI communication configuration register

偏移地址：0x0100

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SIOO	Res.	DQSE	Res.	DDTR	DMODE			Res.		ABSIZE		ABDTR	ABMODE		
rw		rw		rw	rw					rw		rw	rw		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.		ADSIZE		ADDTR	ADMODE			Res.		ISIZE		IDTR	IMODE		
		rw		rw	rw					rw		rw	rw		

位 31 **SIOO**：仅发送指令一次模式
请参见*第 513 页的指令仅发送一次*一节。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 时写入该位域

位 30 保留，必须保持复位值

位 29 **DQSE**：DQS 使能 (DQS enable)
该位使能数据选通管理
0：禁止 DQS
1：使能 DQS

位 28 保留，必须保持复位值

位 27 **DDTR**：交替字节双倍传输速率 (Alternate bytes double transfer rate)
该位为交替字节阶段设置 DTR 模式：
0：交替字节阶段禁止 DTR 模式
1：交替字节阶段使能 DTR 模式
仅可在 BUSY = 0 时写入该位域

位 26:24 **DMODE[2:0]**：数据模式 (Data mode)
该位域定义数据阶段的操作模式：
000：无指令
001：单线传输数据
010：双线传输数据
011：四线传输数据
100：八线传输数据
101-111：保留，必须保持复位值
仅可在 BUSY = 0 时写入该位域

位 23:22 保留，必须保持复位值

位 21:20 **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 时写入该位域

位 19 **ABDTR**：交替字节双倍传输速率 (Alternate bytes double transfer rate)
该位为交替字节阶段设置 DTR 模式：
0：交替字节阶段禁止 DTR 模式
1：交替字节阶段使能 DTR 模式
仅可在 BUSY = 0 时写入该位域

位 18:16 **ABMODE[2:0]**：交替字节模式 (Alternate byte mode)
该位域定义交替字节阶段的操作模式：
000：无指令
001：单线传输交替字节
010：双线传输交替字节
011：四线传输交替字节
011：四线传输交替字节
100：八线传输交替字节
101-111：保留，必须保持复位值
仅可在 BUSY = 0 时写入该位域

位 15:14 保留，必须保持复位值

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 时写入该位域

位 11 **ADDTR**：地址双倍传输速率 (Address double transfer rate)
该位为地址阶段设置 DTR 模式：
0：地址阶段禁止 DTR 模式
1：地址阶段使能 DTR 模式
仅可在 BUSY = 0 时写入该位域

位 10:8 **ADMODE[2:0]**：地址模式 (Address mode)
该位域定义地址阶段的操作模式：
000：无地址
001：单线传输地址
010：双线传输地址
011：四线传输地址
100：八线传输地址
101-111：保留，必须保持复位值
仅可在 BUSY = 0 时写入该位域。

位 7:6 保留，必须保持复位值

位 5:4 **ISIZE[1:0]**：指令大小 (Instruction size)
该位定义指令大小：
00：8 位指令
01：16 位指令
10：24 位指令
11：32 位指令
仅可在 BUSY = 0 时写入该位域

位 3 **IDTR**：指令双倍传输速率 (Instruction double transfer rate)
该位为指令阶段设置 DTR 模式：
0：指令阶段禁止 DTR 模式
1：指令阶段启用 DTR 模式
仅可在 BUSY = 0 时写入该位域

位 2:0 **IMODE[2:0]**：指令模式 (Instruction mode)
该位域定义指令阶段的操作模式：
000：无指令
001：单线传输指令
010：双线传输指令
011：四线传输指令
100：八线传输指令
101-111：保留，必须保持复位值
仅可在 BUSY = 0 时写入该位域

19.6.15 OCTOSPI 时序配置寄存器 (OCTOSPI_TCR)

OCTOSPI timing configuration register

偏移地址：0x0108

复位值：0x0000 0000

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

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

位 31 保留，必须保持复位值

位 30 **SSHIFT**：采样移位 (Sample shift)
默认情况下，OCTOSPI 在外部器件驱动数据后过半个 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
当数据阶段配置为 DTR 模式 (DDTR = 1) 时，固件必须确保 SSHIFT = 0
仅可在 BUSY = 0 时修改该位域

位 29 保留，必须保持复位值

位 28 **DHQC**：延迟保持四分之一周期 (Delay hold quarter cycle)

位 27:5 保留，必须保持复位值

位 4:0 **DCYC[4:0]**：空指令周期数 (Number of dummy cycles)
该位域定义空指令阶段的持续时间
在 SDR 和 DTR 模式下，它指定 CLK 周期数 (0-31)
当使用激活了 DQS 的存储器时，建议至少有五个空指令周期
仅可在 BUSY = 0 时写入该位域

19.6.16 OCTOSPI 指令寄存器 (OCTOSPI_IR)

OCTOSPI instruction register

偏移地址：0x0110

复位值：0x0000 0000

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

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

位 31:0 **INSTRUCTION[31:0]**：指令 (Instruction)
指定要发送到外部 SPI 器件的指令
仅可在 BUSY = 0 时写入该位域

19.6.17 OCTOSPI 交替字节寄存器 (OCTOSPI_ABR)

OCTOSPI alternate bytes register

偏移地址：0x0120

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															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ALTERNATE[15:0]															
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 器件的可选数据。
仅可在 BUSY = 0 时写入该位域。

19.6.18 OCTOSPI 低功耗超时寄存器 (OCTOSPI_LPTR)

OCTOSPI low-power timeout register

偏移地址：0x00130

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TIMEOUT[15:0]															
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，OCTOSPI 将预取后续字节并将其存放在 FIFO 中
该位域指示在时钟变为无效后，OCTOSPI 等待多少个 CLK 时钟周期才让 nCS 升至高电平，将外部器件置为低功耗状态
仅可在 BUSY = 0 时写入该位域

19.6.19 OCTOSPI 回卷通信配置寄存器 (OCTOSPI_WPCCR)

OCTOSPI wrap communication configuration register

偏移地址：0x0140

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.		DQSE	Res.	DDTR	DMODE			Res.		ABSIZE		ABDTR	ABMODE		
		rw		rw	rw					rw		rw	rw		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.		ADSIZE		ADDTR	ADMODE			Res.		ISIZE		IDTR	IMODE		
		rw		rw	rw					rw		rw	rw		

位 31:30 保留，必须保持复位值

位 29 **DQSE**：DQS 使能 (DQS enable)
该位使能数据选通管理
0：禁止 DQS
1：使能 DQS

位 28 保留，必须保持复位值

位 27 **DDTR**：数据双倍传输速率 (Data double transfer rate)
该位为数据阶段设置 DTR 模式：
0：数据阶段禁止 DTR 模式
1：数据阶段使能 DTR 模式
仅可在 BUSY = 0 时写入该位域

位 26:24 **DMODE[2:0]**：数据模式 (Data mode)
该位域定义数据阶段的操作模式：
000：无指令
001：单线传输数据
010：双线传输数据
011：四线传输数据
100：八线传输数据
101-111：保留
仅可在 BUSY = 0 时写入该位域

位 23:22 保留，必须保持复位值

位 21:20 **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 时写入该位域

位 19 **ABDTR**：交替字节双倍传输速率 (Alternate bytes double transfer rate)
该位为交替字节阶段设置 DTR 模式：
0：交替字节阶段禁止 DTR 模式
1：交替字节阶段使能 DTR 模式
仅可在 BUSY = 0 时写入该位域

位 18:16 **ABMODE[2:0]**：交替字节模式 (Alternate byte mode)
该位域定义交替字节阶段的操作模式：
000：无指令
001：单线传输交替字节
010：双线传输交替字节
011：四线传输交替字节
011：四线传输交替字节
100：八线传输交替字节
101-111：保留，必须保持复位值
仅可在 =0 时写入该位域

位 15:14 保留，必须保持复位值

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 时写入该位域

位 11 **ADDTR**：地址双倍传输速率 (Address double transfer rate)
该位为地址阶段设置 DTR 模式：
0：地址阶段禁止 DTR 模式
1：地址阶段使能 DTR 模式
仅可在 BUSY = 0 时写入该位域

位 10:8 **ADMODE[2:0]**：地址模式 (Address mode)
该位域定义地址阶段的操作模式：
000：无地址
001：单线传输地址
010：双线传输地址
011：四线传输地址
100：八线传输地址
101-111：保留，必须保持复位值
仅可在 BUSY = 0 时写入该位域

位 7:6 保留，必须保持复位值

位 5:4 **ISIZE[1:0]**：指令大小 (Instruction size)
该位定义指令大小：
00：8 位指令
01：16 位指令
10：24 位指令
11：32 位指令
仅可在 BUSY = 0 时写入该位域

位 3 **IDTR**：指令双倍传输速率 (Instruction double transfer rate)
该位为指令阶段设置 DTR 模式：
0：指令阶段禁止 DTR 模式
1：指令阶段启用 DTR 模式
仅可在 BUSY = 0 时写入该位域

位 2:0 **IMODE[2:0]**：指令模式 (Instruction mode)
该位域定义指令阶段的操作模式：
000：无指令
001：单线传输指令
010：双线传输指令
011：四线传输指令
100：八线传输指令
101-111：保留，必须保持复位值
仅可在 BUSY = 0 时写入该位域

19.6.20 OCTOSPI 回卷时序配置寄存器 (OCTOSPI_WPTCR)

OCTOSPI wrap timing configuration register

偏移地址：0x0148

复位值：0x0000 0000

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

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

位 31 保留，必须保持复位值

位 30 **SSHIFT**：采样移位 (Sample shift)
默认情况下，OCTOSPI 在外部器件驱动数据后过半个 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
当数据阶段配置为 DTR 模式 (DDTR = 1) 时，固件必须确保 SSHIFT = 0。
仅可在 BUSY = 0 时修改该位域

位 29 保留，必须保持复位值

位 28 **DHQC**：延迟保持四分之一周期 (Delay hold quarter cycle)
在 DTR 通信的输出中增加四分之一周期延迟，以满足保持要求。
0：无四分之一周期延迟
1：插入四分之一周期延迟

位 27:5 保留，必须保持复位值

位 4:0 **DCYC[4:0]**：空指令周期数 (Number of dummy cycles)
该位域定义空指令阶段的持续时间
在 SDR 和 DTR 模式下，它指定 CLK 周期数 (0-31)
当使用激活了 DQS 的存储器时，建议至少有 5 个空指令周期
仅可在 BUSY = 0 时写入该位域

19.6.21 OCTOSPI 回卷指令寄存器 (OCTOSPI_WPIR)

OCTOSPI wrap instruction register

偏移地址：0x0150

复位值：0x0000 0000

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

位 31:0 **INSTRUCTION[31:0]**：指令 (Instruction)
指定要发送到外部 SPI 器件的指令
仅可在 BUSY = 0 时写入该位域

19.6.22 OCTOSPI 回卷交替字节寄存器 (OCTOSPI_WPABR)

OCTOSPI wrap alternate bytes register

偏移地址：0x0160

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															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ALTERNATE[15:0]															
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 器件的可选数据
仅可在 BUSY = 0 时写入该位域

19.6.23 OCTOSPI 写入通信配置寄存器 (OCTOSPI_WCCR)

OCTOSPI write communication configuration register

偏移地址：0x0180

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.		DQSE	Res.	DDTR	DMODE			Res.		ABSIZE		ABDTR	ABMODE		
		rw		rw	rw					rw		rw	rw		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.		ADSIZE		ADDTR	ADMODE			Res.		ISIZE		IDTR	IMODE		
		rw		rw	rw					rw		rw	rw		

位 31:30 保留，必须保持复位值

位 29 **DQSE**：DQS 使能 (DQS enable)
该位使能数据选通管理
0：禁止 DQS
1：使能 DQS

位 28 保留，必须保持复位值

位 27 **DDTR**：数据双倍传输速率 (data double transfer rate)
该位为数据阶段设置 DTR 模式：
0：数据阶段禁止 DTR 模式
1：数据阶段使能 DTR 模式
仅可在 BUSY = 0 时写入该位域

位 26:24 **DMODE[2:0]**：数据模式 (Data mode)
该位域定义数据阶段的操作模式：
000：无指令
001：单线传输数据
010：双线传输数据
011：四线传输数据
011：四线传输数据
100：八线传输数据
101-111：保留，必须保持复位值
仅可在 BUSY = 0 时写入该位域

位 23:22 保留，必须保持复位值

位 21:20 **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 时写入该位域

位 19 **ABDTR**：交替字节双倍传输速率 (Alternate bytes double transfer rate)
该位为交替字节阶段设置 DTR 模式：
0：交替字节阶段禁止 DTR 模式
1：交替字节阶段使能 DTR 模式
仅可在 BUSY = 0 时写入该位域

位 18:16 **ABMODE[2:0]**：交替字节模式 (Alternate byte mode)
该位域定义交替字节阶段的操作模式：
000：无指令
001：单线传输交替字节
010：双线传输交替字节
011：四线传输交替字节
011：四线传输交替字节
100：八线传输交替字节
101-111：保留，必须保持复位值
仅可在 BUSY = 0 时写入该位域

位 15:14 保留，必须保持复位值

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 时写入该位域

位 11 **ADDTR**：地址双倍传输速率 (Address double transfer rate)
该位为地址阶段设置 DTR 模式：
0：地址阶段禁止 DTR 模式
1：地址阶段使能 DTR 模式
仅可在 BUSY = 0 时写入该位域

位 10:8 **ADMODE[2:0]**：地址模式 (Address mode)
该位域定义地址阶段的操作模式：
000：无地址
001：单线传输地址
010：双线传输地址
011：四线传输地址
011：四线传输地址
100：八线传输地址
101-111：保留，必须保持复位值
仅可在 BUSY = 0 时写入该位域

位 7:6 保留，必须保持复位值

位 5:4 **ISIZE[1:0]**：指令大小 (Instruction size)
该位定义指令大小：
00：8 位指令
01：16 位指令
10：24 位指令
11：32 位指令
仅可在 BUSY = 0 时写入该位域

位 3 **IDTR**：指令双倍传输速率 (Instruction double transfer rate)
该位为指令阶段设置 DTR 模式：
0：指令阶段禁止 DTR 模式
1：指令阶段启用 DTR 模式
仅可在 BUSY = 0 时写入该位域

位 2:0 **IMODE[2:0]**：指令模式 (Instruction mode)
该位域定义指令阶段的操作模式：
000：无指令
001：单线传输指令
010：双线传输指令
011：四线传输指令
011：四线传输指令
100：八线传输指令
101-111：保留
仅可在 BUSY = 0 时写入该位域

19.6.24 OCTOSPI 写入时序配置寄存器 (OCTOSPI_WTCR)

OCTOSPI write timing configuration register

偏移地址：0x0188

复位值：0x0000 0000

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

位 31:5 保留，必须保持复位值

位 4:0 **DCYC[4:0]**：空指令周期数 (Number of dummy cycles)
该位域定义空指令阶段的持续时间
在 SDR 和 DTR 模式下，它指定 CLK 周期数 (0-31)
当使用激活了 DQS 的存储器时，建议至少有 5 个空指令周期
仅可在 BUSY = 0 时写入该位域

19.6.25 OCTOSPI 写入指令寄存器 (OCTOSPI_WIR)

OCTOSPI write instruction register

偏移地址：0x0190

复位值：0x0000 0000

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

位 31:0 **INSTRUCTION[31:0]**：指令 (Instruction)
指定要发送到外部 SPI 器件的指令
仅可在 BUSY = 0 时写入该位域

19.6.26 OCTOSPI 写入交替字节寄存器 (OCTOSPI_WABR)

OCTOSPI write alternate bytes register

偏移地址：0x1A0

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

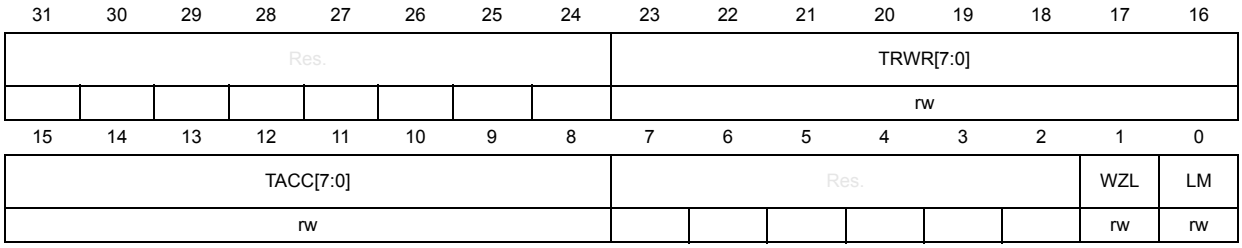
位 31:0 **ALTERNATE[31:0]**：交替字节 (Alternate Bytes)
指定要在地址后立即发送到外部 SPI 器件的可选数据
仅可在 BUSY = 0 时写入该位域

19.6.27 OCTOSPI HyperBus™ 延迟配置寄存器 (OCTOSPI_HLCR)

OCTOSPI HyperBus™ latency configuration register

偏移地址：0x0200

复位值：0x0000 0000



位 31:24 保留，必须保持复位值

位 23:16 **TRWR**：读写恢复时间 (Read write recovery time)
器件读写恢复时间以通信时钟周期数表示

位 15:8 **TACC[7:0]**：访问时间 (Access time)
器件访问时间以通信时钟周期数表示

位 7:2 保留，必须保持复位值

位 1 **WZL**：写入零延迟 (Write zero latency)
该位为写入操作使能零延迟
0：写入访问有延迟
1：写入访问无延迟

位 0 **LM**：延迟模式 (Latency mode)
该位选择延迟模式
0：可变初始延迟
1：固定延迟

19.6.28 OCTOSPI 寄存器映射

下表对 OCTOSPI 寄存器进行了汇总。有关 OCTOSPI 寄存器的基址，请参见寄存器边界地址表。

表 110. OCTOSPI 寄存器映射

偏移	寄存器名称和复位值	寄存器大小																														
		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
0x0000	OCTOSPI_CR	Res.	Res.	FMDOE[1:0]		Res.	Res.	Res.	Res.	PMM	APMS	Res.	TOIE	SMIE	FTIE	TCIE	TEIE	Res.	Res.	Res.	FTHRES[4:0]					FSEL	DQM	Res.	Res.	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
0x0004	Reserved	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
0x0008	OCTOSPI_DCR1	Res.	Res.	Res.	Res.	Res.	MTYP [2:0]			Res.	Res.	Res.	DEVSIZE[4:0]					Res.	Res.	Res.	Res.	Res.	CSHT [2:0]			Res.	Res.	Res.	Res.	Res.	Res.	FRCK.	CKMODE.
	Reset value						0	0	0				0	0	0	0	0						0	0	0							0	0
0x000C	OCTOSPI_DCR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WRA¨PS IZE[2:0]			Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PRESCALER[7:0]							
	Reset value														0	0	0									0	0	0	0	0	0	0	0
0x0010	OCTOSPI_DCR3	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CSBOUND[4:0].					Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	Reset value												0	0	0	0	0																
0x0010	OCTOSPI_DCR3	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CSBOUND[4:0].					Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MAXTRAN[7:0]							
	Reset value												0	0	0	0	0									0	0	0	0	0	0	0	0
0x0014 - 0x001C	Reserved	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
0x0014	OCTOSPI_DCR4	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REFRESH[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0018 - 0x001C	Reserved	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
0x0020	OCTOSPI_SR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FLEVEL[5:0]						Res.	Res.	BUSY	TOF	SMF	FTF	TCF	TEF
	Reset value																			0	0	0	0	0	0			0	0	0	0	0	0
0x0024	OCTOSPI_FCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CTOF	CSMF	Res.	CTCF	CTEF
	Reset value																												0	0		0	0
0x0028 - 0x003C	Reserved	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

表 110. OCTOSPI 寄存器映射（续）

偏移	寄存器名称和复位值	寄存器大小																													
		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
0x0040	OCTOSPI_DLR	DL[31:0]																															
	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
0x0044	Reserved	Res.																															
0x0048	OCTOSPI_AR	ADDRESS[31:0]																															
	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
0x004C	Reserved	Res.																															
0x0050	OCTOSPI_DR	DATA[31:0]																															
	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
0x0054 - 0x007C	Reserved	Res.																															
0x0080	OCTOSPI_PSMKR	MASK[31:0]																															
	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
0x0084	Reserved	Res.																															
0x0088	OCTOSPI_PSMAR	MATCH[31:0]																															
	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
0x008C	Reserved	Res.																															
0x0090	OCTOSPI_PIR	Res.																INTERVAL[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0094 - 0x00FC	Reserved	Res.																															
0x0100	OCTOSPI_CCR	SIOO		DQSE		DDTR	DMODE [2:0]					ABSIZE [1:0]		ABDTR	ABMODE [2:0]					ADSIZE [1:0]		ADDTR	ADMODE [2:0]				ISIZE[1:0]		IDTR	IMODE[2:0]		
	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
0x0104	Reserved	Res.																															
0x0108	OCTOSPI_TCR		SSHIFT		DHQC																								DCYC[4:0]				
	Reset value		0		0																							0	0	0	0	0
0x010C	Reserved	Res.																															
0x0110	OCTOSPI_IR	INSTRUCTION[31:0]																															
	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

表 110. OCTOSPI 寄存器映射（续）

偏移	寄存器名称和复位值	寄存器大小																													
		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
0x0114 -0x011C	Reserved	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
0x0120	OCTOSPI_ABR	ALTERNATE[31:0]																													
	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
0x0124 - 0x012C	Reserved	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
0x0130	OCTOSPI_LPTR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TIMEOUT[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0134 - 0x013C	Reserved	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
0x0140	OCTOSPI_WPCCR	Res.	Res.	DQSE	Res.	DDTR	DMODE [2:0]			Res.	Res.	ABSIZE [1:0]		ABDTR	ABMODE [2:0]			Res.	Res.	ADSIZE [1:0]		ADDTR.	ADMODE [2:0]			Res.	Res.	ISIZE [1:0]		IDTR	IMODE [2:0]		
	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
0x0144	Reserved	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
0x0148	OCTOSPI_WPTCR	Res.	SSHIFT	Res.	DHQC	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DCYC[4:0]				
	Reset value		0		0																								0	0	0	0	0
0x014C	Reserved	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
0x0150	OCTOSPI_WPIR	INSTRUCTION[31:0]																													
	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
0x0154 - 0x015C	Reserved	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
0x0160	OCTOSPI_WPABR	ALTERNATE[31:0]																													
	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
0x0164 - 0x017C	Reserved	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
0x0134 - 0x017C	Reserved	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
0x0180	OCTOSPI_WCCR	Res.	Res.	DQSE	Res.	DDTR	DMODE [2:0]			Res.	Res.	ABSIZE [1:0]		ABDTR	ABMODE [2:0]			Res.	Res.	ADSIZE [1:0]		ADDTR	ADMODE [2:0]			Res.	Res.	ISIZE [1:0]		IDTR	IMODE [2:0]		
	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
0x0184	Reserved	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

表 110. OCTOSPI 寄存器映射（续）

偏移	寄存器名称和复位值	寄存器大小																													
		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
0x0188	OCTOSPI_WTCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DCYC[4:0]				
	Reset value																																
0x018C	Reserved	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
0x0190	OCTOSPI_WIR	INSTRUCTION[31:0]																															
	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
0x0194 - 0x019C	Reserved	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
0x01A0	OCTOSPI_WABR	ALTERNATE[31:0]																															
	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
0x01A4 - 0x01FC	Reserved	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	Reset value																																
0x0200	OCTOSPI_HLCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TRWR[7:0]								TACC[7:0]								Res.	Res.	Res.	Res.	Res.	Res.	WZL	LM
	Reset value									0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0							0	0

20 OctoSPI IO 管理器 (OCTOSPIM)

20.1 简介

OctoSPI IO 管理器是一个低级别使能：

- 高效的 OctoSPI 全 IO 矩阵引脚分配（在复用功能映射前）

20.2 OctoSPI IO 管理器的主要特性

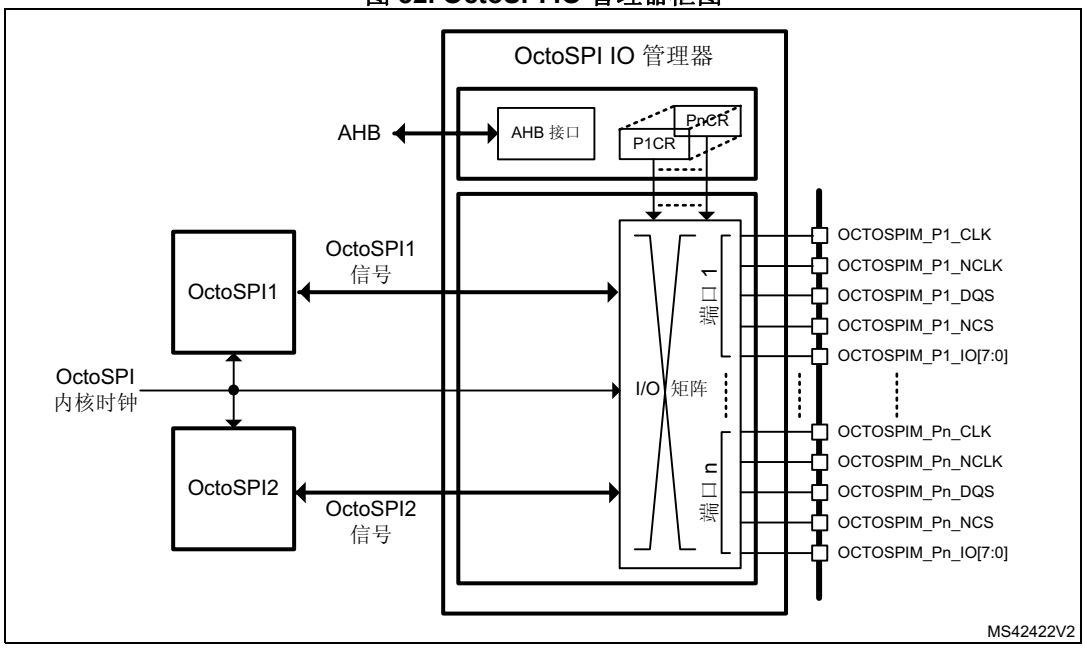
- 最多支持两个单/双/四/八 SPI 接口
- 最多支持使用 3 个端口进行引脚分配
- 完全可编程 IO 矩阵，支持按功能（数据/控制/时钟）进行引脚分配

20.3 OctoSPI IO 管理器功能说明

20.3.1 概述

OctoSPI IO 管理器的框图如*图 82* 所示。

图 82. OctoSPI IO 管理器框图



1. 端口数量 (n) 的取值范围是 1 到 3。

20.3.2 OctoSPI IO 管理器矩阵

用户可使用 OctoSPI IO 管理器矩阵设置完全可编程的功能预映射：

- 所有 OCTOSPIM_Pn_CLK/OCTOSPIM_Pn_NCLK 对均可独立映射到 OctoSPI1_CLK/OctoSPI1_NCLK 或 OctoSPI2_CLK/OctoSPI2_NCLK
- 所有 OCTOSPIM_Pn_DQS 均可独立映射到 OctoSPI1_DQS 或 OctoSPI2_DQS
- 所有 OCTOSPIM_Pn_NCS 均可独立映射到 OctoSPI1_NCS 或 OctoSPI2_NCS
- 所有 OCTOSPIM_Pn_IO[3:0] 和 OCTOSPIM_Pn_IO[7:4] 均可独立映射到 OctoSPI1_IO[3:0]、OctoSPI1_IO[7:4]、OctoSPI2_IO1[3:0] 或 OctoSPI2_IO[7:4]

每个 OctoSPI IO 管理器端口均由独立信号使能，通过相应的 OctoSPI IO 管理器端口 n 配置寄存器 (OCTOSPIM_PnCR) 对映射进行配置。

当多个 I/O 引脚拥有相同配置，且在同一时间被使能时，结果会不可预测。

复位后默认 OctoSPI1 和 OctoSPI2 的所有信号分别在端口 1 和端口 2 映射。

当 OctoSPI 被禁用时，I/O 矩阵应该被配置，以便面总线中出现意外事务。

20.4 OctoSPI IO 管理器寄存器

20.4.1 OctoSPI IO 管理器端口 n 配置寄存器 (OCTOSPIM_PnCR)

OctoSPI IO Manager Port n configuration register

偏移地址：0x0000 + 4 x n

复位值：n = 1 时为 0x0301 0111
复位值：n = 2 时为 0x0705 0333

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	IOHSRC[1:0]		IOHEN	Res.	Res.	Res.	Res.	Res.	IOLSRC[1:0]		IOLEN
					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.	Res.	Res.	Res.	NCSSRC	NCSEN	Res.	Res.	DQSSRC	DQSEN	Res.	Res.	CLKSRC	CLKEN
						rw	rw			rw	rw			rw	rw

位 31:27 保留，必须保持复位值。

位 26:25 **IOHSRC[1:0]**：端口 n 的 IO[7:4] 源 (IO[7:4] Source for Port n)
该位用于选择端口 n 的 IO[7:4] 源。
00：OCTOSPI1_IO[3:0]
01：OCTOSPI1_IO[7:4]
10：OCTOSPI2_IO[3:0]
11：OCTOSPI2_IO[7:4]

位 24 **IOHEN**：端口 n 的 IO[7:4] 使能 (IO[7:4] Enable for Port n)
该位用于使能端口 n 的 IO[7:4]。
0：禁止端口 n 的 IO[7:4]
1：使能端口 n 的 IO[7:4]

位 23:19 保留，必须保持复位值。

位 18:17 **IOLSRC[1:0]**：端口 n 的 IO[3:0] 源 (IO[3:0] Source for Port n)
该位用于选择端口 n 的 IO[3:0] 源。
00：OCTOSPI1_IO[3:0]
01：OCTOSPI1_IO[7:4]
10：OCTOSPI2_IO[3:0]
11：OCTOSPI2_IO[7:4]

位 16 **IOLEN**：端口 n 的 IO[3:0] 使能 (IO[3:0] Enable for Port n)
该位用于使能端口 n 的 IO[3:0]。
0：禁止端口 n 的 IO[3:0]
1：使能端口 n 的 IO[3:0]

位 15:10 保留，必须保持复位值。

位 9 **NCSSRC**：端口 n 的 nCS 源 (nCS Source for Port n)
该位用于选择端口 n 的 nCS 源。
0：OCTOSPI1_nCS
1：OCTOSPI2_nCS

位 8 **NCSEN**：端口 n 的 nCS 使能 (nCS Enable for Port n)
该位用于使能端口 n 的 nCS。
0：禁止端口 n 的 nCS
1：使能端口 n 的 nCS

位 7:6 保留，必须保持复位值。

位 5 **DQSSRC**：端口 n 的 DQS 源 (DQS Source for Port n)
该位用于选择端口 n 的 DQS 源。
0：OCTOSPI1_DQS
1：OCTOSPI2_DQS

位 4 **DQSEN**：端口 n 的 DQS 使能 (DQS Enable for Port n)
该位用于使能端口 n 的 DQS。
0：禁止端口 n 的 DQS
1：使能端口 n 的 DQS

位 3:2 保留，必须保持复位值。

位 1 **CLKSRC**：端口 n 的 CLK/CLKn 源 (CLK/CLKn Source for Port n)
该位用于选择端口 n 的 CLK/CLKn 源。
0：OCTOSPI1_CLK/CLKn
1：OCTOSPI2_CLK/CLKn

位 0 **CLKEN**：端口 n 的 CLK/CLKn 使能 (CLK/CLKn Enable for Port n)
该位用于使能端口 n 的 CLK/CLKn。
0：禁止端口 n 的 CLK/CLKn
1：使能端口 n 的 CLK/CLKn

20.4.2 OctoSPI IO 管理器寄存器映射

下表对 OctoSPI IO 管理器寄存器进行了汇总。有关 OctoSPI IO 管理器寄存器的基址，请参见寄存器边界地址表。

表 111. OCTOSPI IO 管理器寄存器映射和复位值

偏移	寄存器名称 复位值	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
0x0004	**OCTOSPIM _P1CR**	Res.	Res.	Res.	Res.	Res.	IOHSRC [1:0]		IOHEN	Res.	Res.	Res.	Res.	Res.	IOLSRC [1:0]		IOLEN	Res.	Res.	Res.	Res.	Res.	Res.	NCSSRC	NCSEN	Res.	Res.	DQSSRC	DQSEN	Res.	Res.	CLKSRC	CLKEN
	Reset value						0	1	1						0	0	1							0	1			0	1			0	1
0x0008	**OCTOSPIM _P2CR**	Res.	Res.	Res.	Res.	Res.	IOHSRC [1:0]		IOHEN	Res.	Res.	Res.	Res.	Res.	IOLSRC [1:0]		IOLEN	Res.	Res.	Res.	Res.	Res.	Res.	NCSSRC	NCSEN	Res.	Res.	DQSSRC	DQSEN	Res.	Res.	CLKSRC	CLKEN
	Reset value						1	1	1						1	0	1							1	1			1	1			1	1

21 模数转换器 (ADC)

21.1 简介

ADC 由 12 位逐次逼近模数转换器组成。

ADC 的复用通道多达 20 条。各种不同通道的 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。

ADC 映射到 AHB 总线，从而可实现快速数据处理。

ADC 具有模拟看门狗特性，允许应用检测输入电压是否超过了用户自定义的阈值上限或下限。

内置硬件过采样器，可提高模拟性能，同时还能减轻 CPU 进行相关计算的负担。

采用了高效的低功耗模式，在低频下可实现极低的功耗。

21.2 ADC 主要特性

- 高性能特性
  - 可配置 12 位、10 位、8 位或 6 位分辨率
  - ADC 转换时间：
    快速通道：12 位分辨率时为 0.188 µs (5.33 Ms/s)
    慢速通道：12 位分辨率时为 0.238 µs (4.21 Ms/s)
  - ADC 转换时间与 AHB 总线时钟频率无关
  - 可通过降低分辨率来缩短转换时间：10 位分辨率时为 0.16 µs
  - 可管理单端或差分输入
  - AHB 从总线接口，可实现快速数据处理
  - 自校准
  - 可独立设置各通道采样时间
  - 多达四条注入通道（对常规通道或注入通道的模拟输入分配完全可配置）
  - 硬件辅助准备注入通道的上下文，从而实现快速上下文切换
  - 数据对齐以保持内置数据一致性
  - 数据可由 GP-DMA 管理，以实现常规通道转换
  - 数据可连接到 DFSDM 进行后期处理
  - 4 个专用数据寄存器供注入通道使用
- 过采样器
  - 16 位数据寄存器
  - 过采样率可在 2x 到 256x 之间进行调整
  - 可编程数据最多可移位 8 位
- 低功耗特性
  - 速度自适应低功耗模式，可降低 ADC 在低频下工作时的功耗
  - 可在实现慢速总线频率应用的同时保持最佳 ADC 性能（无论 AHB 总线时钟频率为何，快速通道均可保持 0.188 µs 的转换时间）
  - 提供自动控制，可避免 ADC 在低 AHB 总线时钟频率应用中溢出（自动延迟模式）

- 多个外部模拟输入通道
  - GPIO 焊盘有多达 5 条快速通道
  - GPIO 焊盘有多达 11 条慢速通道
- 此外，还有多条内部专用通道
  - 内部参考电压 (V_{REFINT})
  - 内部温度传感器 (V_{SENSE})
  - V_{BAT} 监测通道 ($V_{BAT}/3$)
  - DAC1 和 DAC2 内部通道，连接到 ADC1
- 可通过以下方式启动转换过程：
  - 通过软件启动常规转换和注入转换
  - 通过极性可配置的硬件触发器（内部定时器事件或 GPIO 输入事件）启动常规转换和注入转换
- 转换模式
  - ADC 可转换单条通道，也可扫描一系列通道
  - 单次模式会在每次触发时对选定的输入执行一次转换
  - 连续模式可连续转换选定的输入
  - 不连续采样模式
- 在 ADC 准备就绪、采样结束、转换结束（常规转换或注入转换）、序列转换结束（常规转换或注入转换）、模拟看门狗 1/2/3 事件或溢出事件时生成中断
- 3 个模拟看门狗
- ADC 输入范围：$V_{REF-} \leq V_{IN} \leq V_{REF+}$

图 83 给出了一个 ADC 的框图。

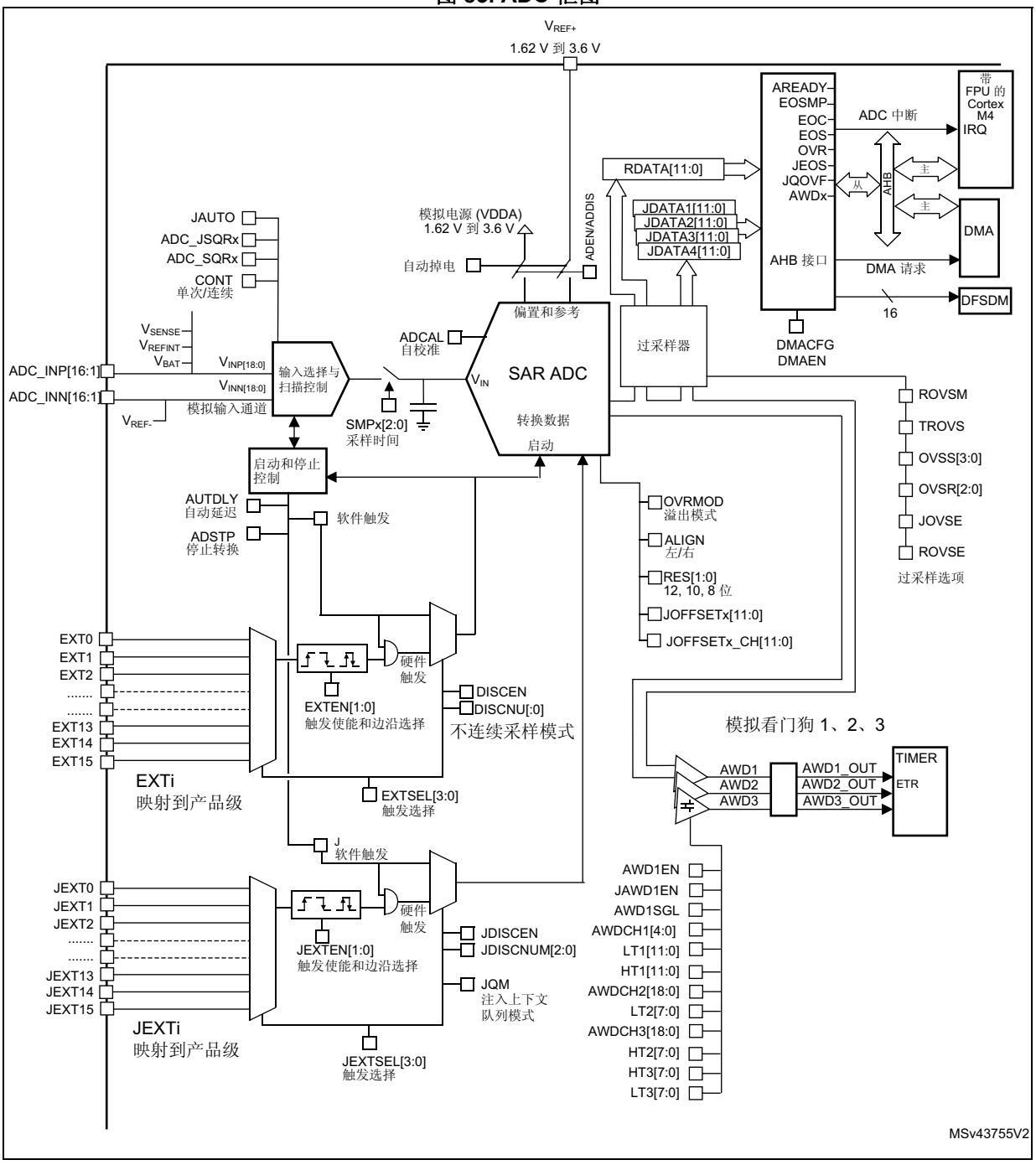
21.3 ADC 实现

21.4 ADC 功能说明

21.4.1 ADC 框图

图 83 给出了 ADC 框图，*表 113* 列出了 ADC 的引脚说明。

图 83. ADC 框图


VREF+
1.62 V 到 3.6 V
AREADY
EOSMP
EOC
EOS
OVR
JEOS
JQOVF
AWDx
ADC 中断
带 FPU 的 Cortex M4 IRQ
AHB
主
从
主
DMA
AHB 接口
DMA 请求
DFSDM
16
DMACFG
DMAEN
RDATA[11:0]
JDATA1[11:0]
JDATA2[11:0]
JDATA3[11:0]
JDATA4[11:0]
ADEN/ADDIS
模拟电源 (VDDA)
1.62 V 到 3.6 V
自动掉电
JAUTO
ADC_JSQRx
ADC_SQRx
CONT
单次/连续
VSENSE
VREFINT
VBAT
VINP[18:0]
VINN[18:0]
ADC_INP[16:1]
ADC_INN[16:1]
模拟输入通道
VREF-
输入选择与扫描控制
ADCAL
自校准
偏置和参考
SAR ADC
VIN
转换数据
启动
过采样器
SMPx[2:0]
采样时间
启动和停止控制
AUTDLY
自动延迟
ADSTP
停止转换
软件触发
ROVSM
TROVS
OVSS[3:0]
OVSR[2:0]
JOVSE
ROVSE
过采样选项
OVRMOD
溢出模式
ALIGN
左/右
RES[1:0]
12, 10, 8 位
JOFFSETx[11:0]
JOFFSETx_CH[11:0]
EXT0
EXT1
EXT2
EXT13
EXT14
EXT15
EXTi
映射到产品级
硬件触发
EXTEN[1:0]
触发使能和边沿选择
DISCEN
DISCNU[:0]
不连续采样模式
EXTSEL[3:0]
触发选择
模拟看门狗 1、2、3
AWD1
AWD2
AWD3
AWD1_OUT
AWD2_OUT
AWD3_OUT
TIMER
ETR
J
软件触发
JEXT0
JEXT1
JEXT2
JEXT13
JEXT14
JEXT15
JEXTi
映射到产品级
硬件触发
JEXTEN[1:0]
触发使能和边沿选择
JDISCEN
JDISCNUM[2:0]
JQM
注入上下文队列模式
JEXTSEL[3:0]
触发选择
AWD1EN
JAWD1EN
AWD1SGL
AWDCH1[4:0]
LT1[11:0]
HT1[11:0]
AWDCH2[18:0]
LT2[7:0]
AWDCH3[18:0]
HT2[7:0]
HT3[7:0]
LT3[7:0]
MSv43755V2

21.4.2 ADC 引脚和内部信号

表 112. ADC 内部输入/输出信号

内部信号名称	信号类型	说明
EXT[15:0]	输入	共有多达 16 个外部触发输入用于常规转换（可连接至片上定时器）。
JEXT[15:0]	输入	共有多达 16 个外部触发输入用于注入转换（可连接至片上定时器）。
ADC_AWDx_OUT	输出	内部模拟看门狗输出信号，连接至片上定时器。（x = 模拟看门狗编号 1、2、3）
V_{SENSE}	输入	内部温度传感器的输出电压
V_{REFINT}	输入	内部参考电压的输出电压
V_{BAT}	输入电源	外部电池电压

表 113. ADC 输入/输出引脚

引脚名称	信号类型	注释
V_{REF+}	正模拟参考电压输入	ADC 高/正参考电压， 1.62 V ≤ V_{REF+} ≤ V_{DDA}
V_{DDA}	模拟电源输入	模拟电源等于 V_{DDA}： 1.62 V ≤ V_{DDA} ≤ 3.6 V
V_{REF-}	负模拟参考电压输入	ADC 的低/负参考电压。V_{REF-} 在内部连接到 V_{SSA}
V_{SSA}	模拟电源地输入	模拟电源接地引脚。对于无专用 V_{SSA} 引脚的器件封装，V_{SSA} 在内部连接到 V_{SS}。
V_{INP}[18:0]	正输入模拟通道	连接到外部通道：ADC1_INP*i* 或内部通道。
V_{INN}[18:0]	负输入模拟通道	连接到 V_{REF-} 或外部通道：ADC1_INN*i* 和 ADC1_INP*[i+1]*
ADC1_INN[16:1]	外部模拟输入信号	最多 16 个模拟输入通道： – 5 个快速通道（ADC1_INN1 到 IN5） – 11 个慢速通道（ADC1_INN6 到 INN16）
ADC1_INP[16:1]	外部模拟输入信号	最多 16 个模拟输入通道： – 5 个快速通道（ADC1_INP1 到 INP5） – 11 个慢速通道（ADC1_INP6 到 INP16）

21.4.3 时钟

双时钟域架构

双时钟域架构意味着 ADC 时钟独立于 AHB 总线时钟。

三个 ADC 的输入时钟相同，可从两个不同的时钟源中选择（请参见*图 84：ADC 时钟方案*）：

1. ADC 时钟可为特定时钟源。它可来自于以下时钟源：
   - 系统时钟
   - PLLSAI1

   有关如何生成 ADC 专用时钟的更多信息，请参见 RCC 部分。要选择此时钟方案，必须将 ADCx_CCR 寄存器的 CKMODE[1:0] 位复位。
2. ADC 时钟可由 ADC 总线接口的 AHB 时钟除以一个可编程的因数（1、2 或 4）来提供。在这种模式下，可选择可编程的分频系数（根据位 CKMODE[3:0]，选择 1、2 或 4）。

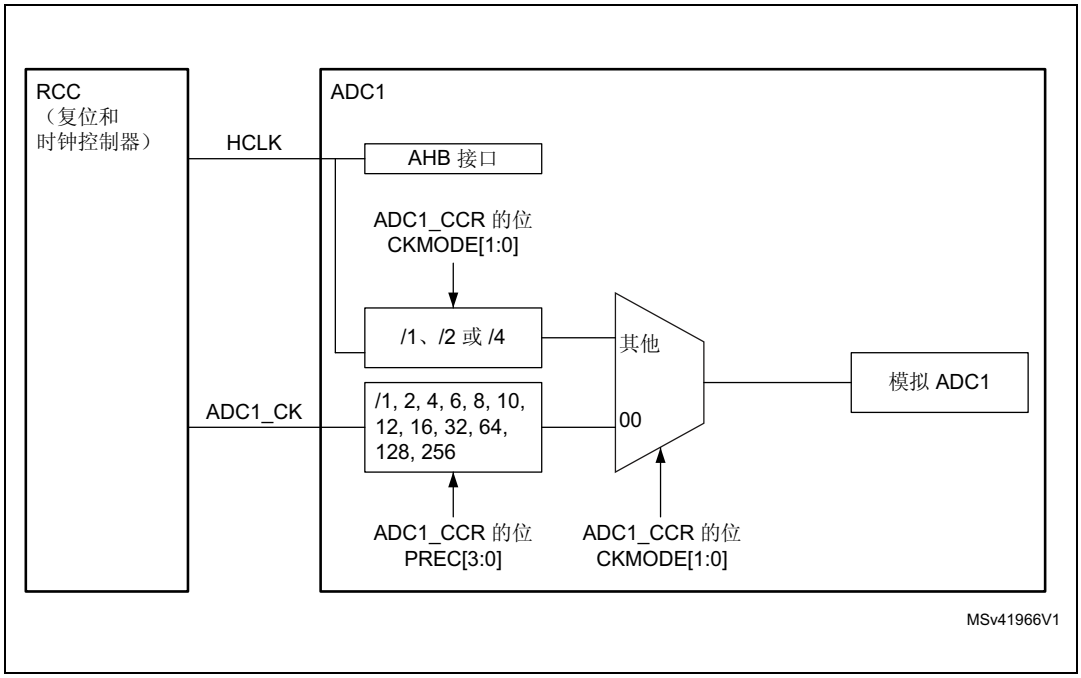
   要选择此时钟方案，ADCx_CCR 寄存器的 CKMODE[1:0] 位不得为“00”。

注： *对于选项 2)，仅当 AHB 预分频器置 1（RCC_CFGR 寄存器中的 HPRE[3:0] = 0xxx）时，才能使用预分频系数 1 (CKMODE[1:0]=01)。*

选项 1) 的优点在于无论选择哪种 AHB 时钟方案，都可以达到最大 ADC 时钟频率。使用通过 ADCx_CCR 寄存器中的 PRESC[3:0] 位配置的预分频器时，ADC 时钟最后会除以以下比率：1、2、4、6、8、12、16、32、64、128、256。

选项 2) 的优势在于绕过了时钟域重新同步。如果 ADC 由定时器触发，并且应用要求 ADC 精确触发（不存在任何不确定性），可使用此选项（否则，重新同步两个时钟域会为触发时刻带来不确定性）。

图 84. ADC 时钟方案



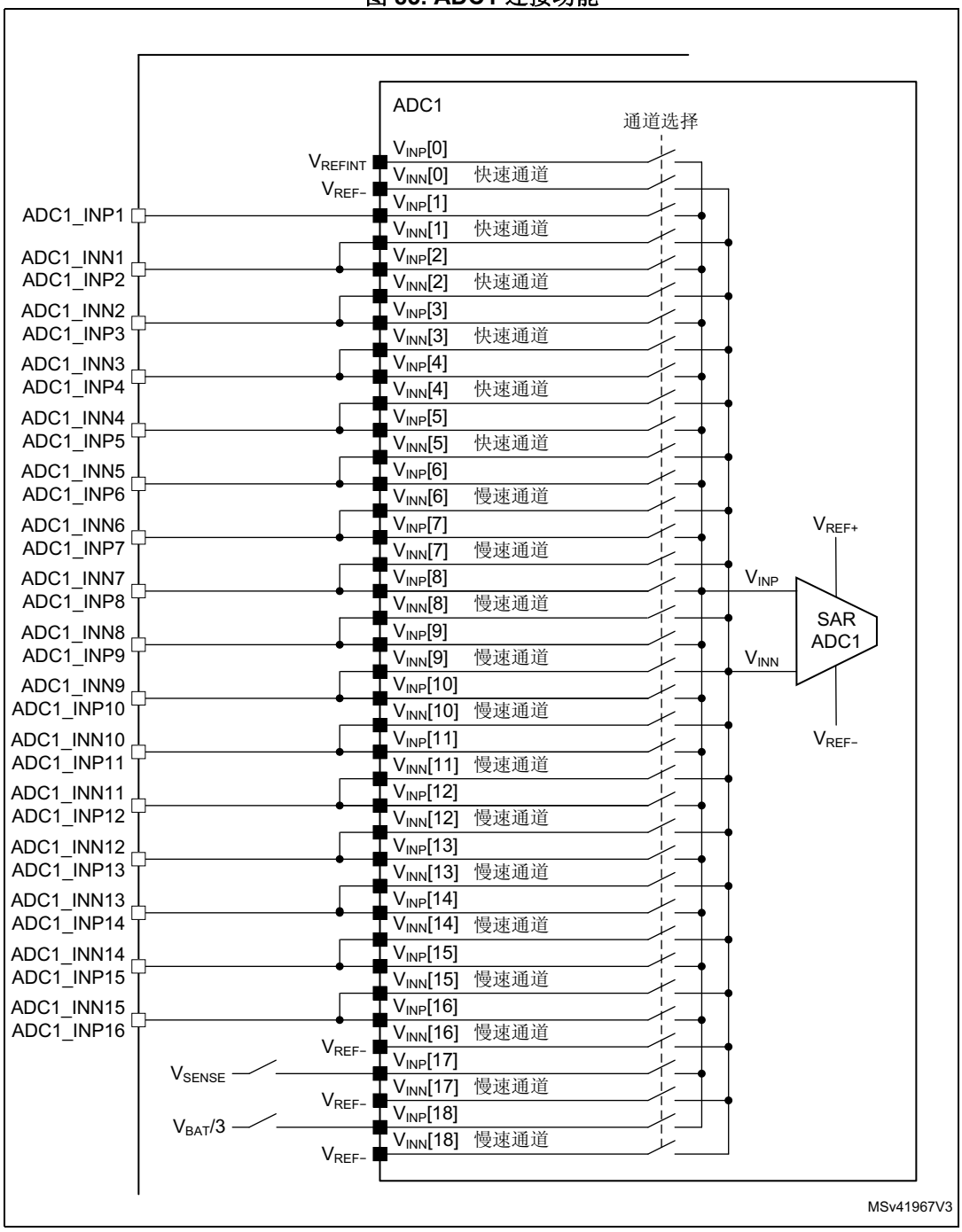
ADC 时钟与 AHB 时钟的时钟比例限制

通常来讲，ADC 时钟与 AHB 时钟之比没有限制，但一些注入通道已编程的情况除外，此时，ADC 时钟与 AHB 时钟之比必须遵循以下规定：

- 如果所有通道的分辨率均为 12 位或 10 位，则 $F_{HCLK} >= F_{ADC}/4$
- 如果一些通道的分辨率为 8 位（且所有通道的分辨率均不低于 8 位），则 $F_{HCLK} >= F_{ADC}/3$
- 如果一些通道的分辨率为 6 位，则 $F_{HCLK} >= F_{ADC}/2$

21.4.4 ADC1 连接功能

图 85. ADC1 连接功能



21.4.5 从设备 AHB 接口

ADC 利用 AHB 从设备端口实现控制/状态寄存器访问和数据访问。AHB 接口的特性如下：

- 字（32 位）访问
- 单周期响应
- 以零等待状态响应对寄存器的所有读/写访问

AHB 从设备接口不支持分离/重试请求，且绝不会生成 AHB 错误。

21.4.6 ADC 深度掉电模式 (DEEPPWD) 和 ADC 稳压器 (ADVREGEN)

ADC 默认处于深度掉电模式，在该模式下，其电源会在内部切断以降低泄漏电流（ADCx_CR 寄存器中的 DEEPPWD 位的复位状态为 1）。

要开始 ADC 操作，需要先将 DEEPPWD 位清零，使 ADC 退出深度掉电模式。

然后，必须将 ADCx_CR 寄存器中的 ADVREGEN 位置 1，以使能 ADC 内部稳压器。软件必须在 ADC 稳压器的启动时间 ($T_{ADCVREG_STUP}$) 内等待，之后才能开始校准或使能 ADC。该延迟必须通过软件实现。

有关 ADC 稳压器的启动时间，请参见器件数据手册中的 $T_{ADCVREG_STUP}$ 参数。

ADC 操作完成后，可禁止 ADC (ADEN=0)。此外，还可通过禁止 ADC 稳压器实现节能。这可通过向 ADVREGEN 位写入 0 来实现。

之后，为了通过降低泄漏电流的方式进一步实现节能，还可以将 ADCx_CR 寄存器中的 DEEPPWD 位置 1，以使 ADC 再次进入深度掉电模式。进入停止模式之前，这一点特别值得关注。

注： *写入 DEEPPWD=1 会自动禁止 ADC 稳压器，并且 ADVREGEN 位会自动清零。*

如果内部稳压器已禁止 (ADVREGEN=0)，则会保持内部模拟校准。

在 ADC 深度掉电模式下 (DEEPPWD=1)，内部模拟校准会丢失，需要重新启动校准或重新应用之前保存的校准系数（请参见*第 21.4.8 节：校准（ADCAL、ADCALDIF、ADCx_CALFACT）*）。

21.4.7 单端通道和差分输入通道

可通过写入 ADCx_DIFSEL 寄存器中的 DIFSEL[15:1] 位将通道配置为单端输入或差分输入。必须在禁止 ADC (ADEN=0) 时写入此配置。请注意，DIFSEL[18:16,0] 固定为单端通道，始终读为 0。

在单端输入模式下，要通过通道“i”转换的模拟电压是外部电压 $V_{INP[i]}$（正输入）和 V_{REF-}（负输入）之差。

在差分输入模式下，要通过通道“i”转换的模拟电压是外部电压 $V_{INP[i]}$（正输入）与 $V_{INN[i]}$（负输入）之差。

当 ADC 配置为差分模式时，两路输入的偏置电压均应为 $(V_{REF+})/2$。

输入信号应为差分信号（共模电压应固定）。

有关输入通道连接方式的完整说明，请参见*第 21.4.4 节：ADC1 连接功能*。

注意： 将通道“i”配置为差分输入模式时，其负输入电压 $V_{INN[i]}$ 连接至 $V_{INP[i+1]}$。因此，通道“*i+1*”不再可用于单端模式或差分模式下，并且不得配置为进行转换。

注： *由于 ADC 通道 0、16、17、18 连接至单端外部模拟输入或内部通道，因此其强制采用单端配置（相应的 DIFSEL[i] 位始终为零）。*

21.4.8 校准（ADCAL、ADCALDIF、ADCx_CALFACT）

ADC 提供有自动校准过程，用于驱动包括 ADC 上电/掉电序列在内的所有校准序列。在校准过程中，ADC 会计算 7 位宽的校准系数，并会在下一次掉电之前在自身内部应用该值。在校准过程中，应用不得使用 ADC，必须等待至校准完成。

执行任何 ADC 操作之前都必须进行校准。校准可消除偏移误差，由于制造工艺或带隙的不同，各芯片的偏移误差也有所不同。

单端输入转换与差分输入转换应用的校准系数有所不同：

- 启动要为单端输入转换应用的校准之前，写入 ADCALDIF=0。
- 启动要为差分输入转换应用的校准之前，写入 ADCALDIF=1。

随后，通过软件将 ADCAL 位置 1 发起校准。仅当 ADC 禁止 (ADEN=0) 后，才能发起校准。ADCAL 位在所有校准序列过程中保持为 1。校准完成后，此位会立即由硬件清零。此时，相关校准系数会存储在模拟 ADC 内部，同时还会存储在 ADCx_CALFACT 寄存器的 CALFACT_S[6:0] 或 CALFACT_D[6:0] 位中（具体取决于是单端输入校准还是差分输入校准）

如果禁止 ADC (ADEN=0)，则会保留内部模拟校准。但如果长时间禁止 ADC，建议在重新使能 ADC 之前运行新的校准周期。

每次 ADC 掉电时，内部模拟校准都会丢失（例如，产品进入 STANDBY 模式或 VBAT 模式时）。这种情况下，为了避免浪费时间重新校准 ADC，可将校准系数重新写入 ADCx_CALFACT 寄存器，但前提是软件之前保存了上次校准时得出的校准系数。

如果 ADC 已使能但未进行转换（ADEN=1、ADSTART=0 且 JADSTART=0），可写入校准系数。随后，下次转换启动时，校准系数会自动添加到模拟 ADC 中。这一载入过程是透明的，不会对转换的启动造成延迟。建议在 V_{REF+} 电压变化超过 10% 时重新进行校准。

ADC 校准的软件流程

1. 确保 DEEPPWD=0、ADVREGEN=1，并确保 ADC 稳压器启动时间已过。
2. 确保 ADEN=0。
3. 设置 ADCALDIF=0（单端输入）或 ADCALDIF=1（差分输入），选择此校准的输入模式。
4. 将 ADCAL 置 1。
5. 等待 ADCAL=0。
6. 可从 ADCx_CALFACT 寄存器读取校准系数。

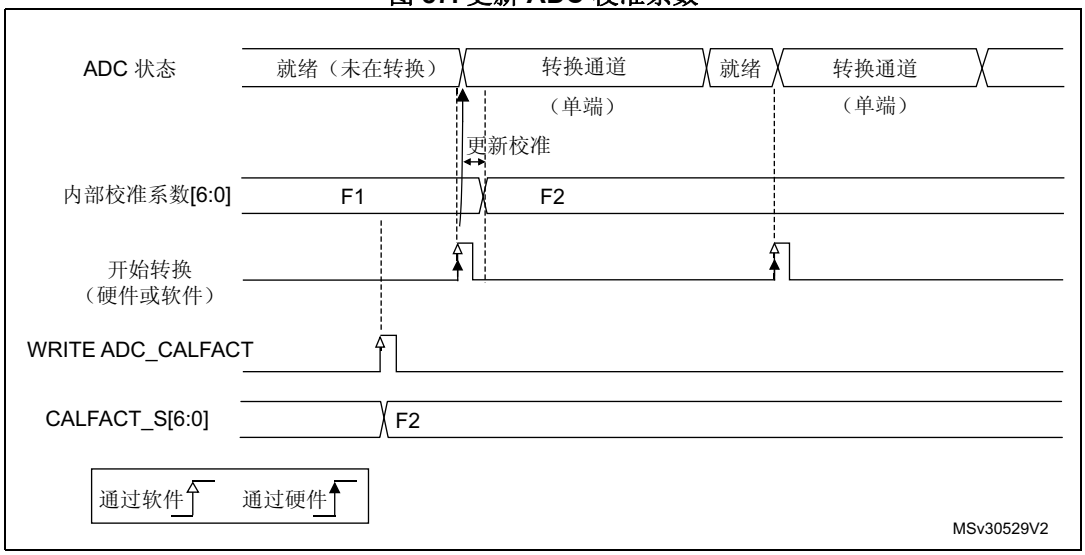
图 86. ADC 校准



将校准系数重新注入 ADC 的软件流程

1. 确保 ADEN=1、ADSTART=0 且 JADSTART=0（ADC 已使能，且未进行任何转换）。
2. 将新的校准系数写入 CALFACT_S 和 CALFACT_D。
3. 启动转换时，仅当内部模拟校准系数与 CALFACT_S 位（单端输入通道）或 CALFACT_D 位（差分输入通道）中存储的值不同时，校准系数才会注入到模拟 ADC。

图 87. 更新 ADC 校准系数

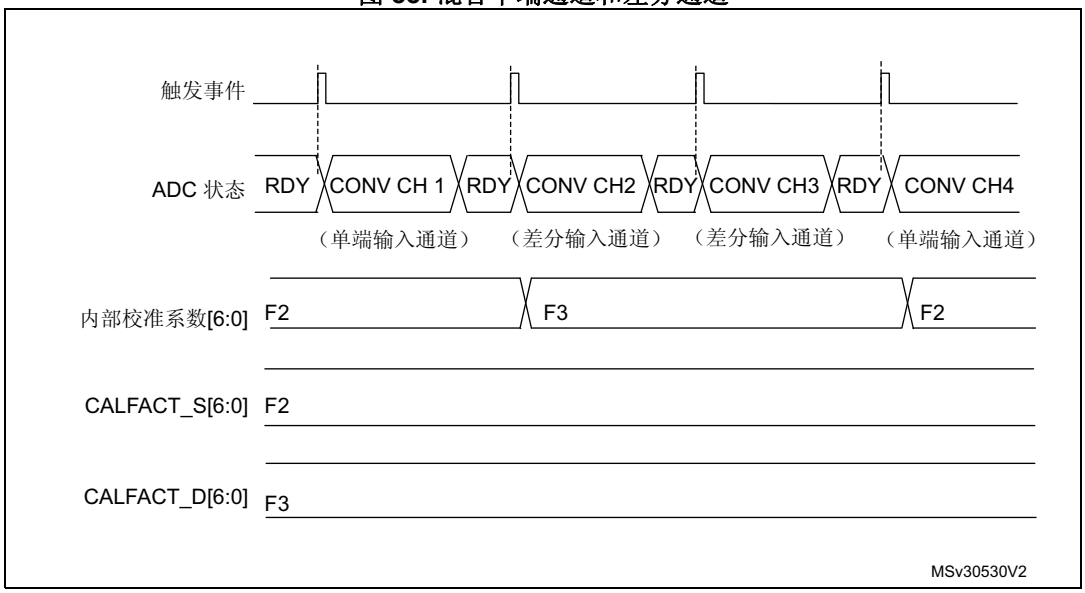


单个 ADC 转换单端模拟输入和差分模拟输入

如果 ADC 既要转换差分输入，也要转换单端输入，则必须执行两次校准，两次校准分别在 ADCALDIF=0 和 ADCALDIF=1 的情况下执行。操作流程如下：

1. 禁止 ADC。
2. 在单端输入模式下 (ADCALDIF=0) 校准 ADC。此操作会更新寄存器 CALFACT_S[6:0]。
3. 在差分输入模式下 (ADCALDIF=1) 校准 ADC。此操作会更新寄存器 CALFACT_D[6:0]。
4. 使能 ADC、配置通道并启动转换。每次从单端输入通道切换为差分输入通道（或进行相反切换）时，校准都将自动注入到模拟 ADC 中。

图 88. 混合单端通道和差分通道



21.4.9 ADC 开关控制（ADEN、ADDIS、ADRDY）

首先按照*第 21.4.6 节：ADC 深度掉电模式 (DEEPPWD) 和 ADC 稳压器 (ADVREGEN)*中的流程执行操作。

DEEPPWD=0 且 ADVREGEN=1 后，可使能 ADC，并且 ADC 在开始精确转换之前需要 t_{STAB} 的稳定时间，如*图 89* 所示。以下两个控制位可使能或禁止 ADC：

- 将 ADEN 置 1 可使能 ADC。ADC 准备就绪后，ADRDY 标志会立即置 1。
- 将 ADDIS 置 1 可禁止 ADC。随后，模拟 ADC 被完全禁止后，ADEN 位和 ADDIS 位会自动由硬件清零。

随后可通过将 ADSTART 置 1（请参见*第 21.4.18 节：外部触发转换和触发极性（EXTSEL、EXTEN、JEXTSEL、JEXTEN）*）开始进行常规转换，如果触发器已使能，也可在发生外部触发事件时开始进行常规转换。

可通过将 JADSTART 置 1 开始进行注入转换，如果注入触发器已使能，也可在发生外部注入触发事件时开始进行注入转换。

通过软件使能 ADC 的流程

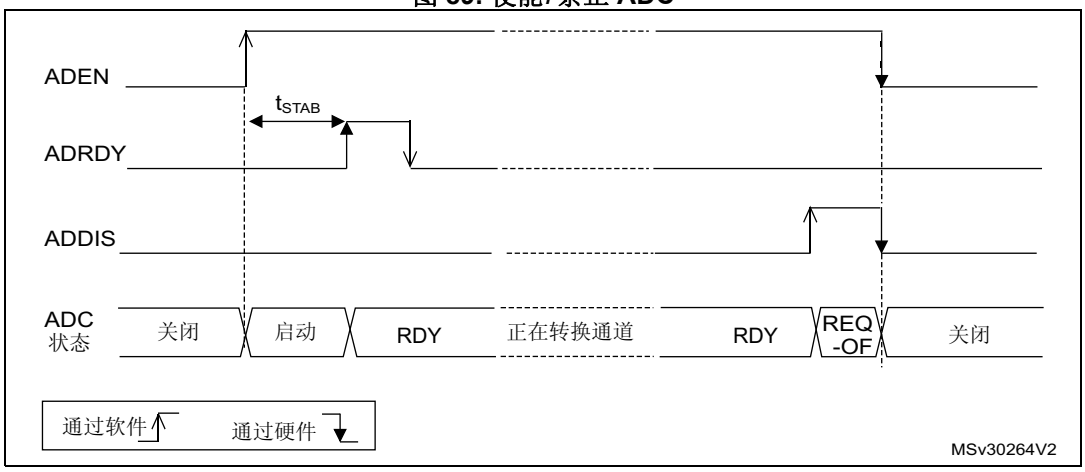
1. 向 ADCx_ISR 寄存器中的 ADRDY 位写入“1”，将其清零。
2. 将 ADEN 置 1。
3. 等待，直至 ADRDY=1（ADRDY 会在 ADC 启动时间后置 1）。这可以使用关联中断来实现（将 ADRDYIE 置 1）。
4. 向 ADCx_ISR 寄存器中的 ADRDY 位写入“1”，将其清零（可选）。

注意： ADCAL=1 期间和硬件清零 ADCAL 位后（校准结束）的 4 个 ADC 时钟周期内，ADEN 位无法置 1。

通过软件禁止 ADC 的流程

1. 检查 ADSTART 和 JADSTART 是否均为 0，以确保当前未执行任何转换。如有需要，可将 ADSTP 置 1，并将 JADSTP 置 1，随后等待至 ADSTP=0 且 JADSTP=0，以停止任何正在进行的常规转换和注入转换。
2. 将 ADDIS 置 1。
3. 如果应用要求，可等待 ADEN=0，直至模拟 ADC 已有效禁止（ADEN=0 后，ADDIS 将自动复位）。

图 89. 使能/禁止 ADC



21.4.10 写入 ADC 控制位时的限制

仅当 ADC 已禁止（ADEN 必须等于 0）时，软件才可通过写入 RCC 控制位的方式配置和使能 ADC 时钟（请参见 RCC 部分）、ADCx_DIFSEL 寄存器中的控制位 DIFSEL、ADCx_CR 寄存器中的控制位 ADCAL 和 ADEN。

只有在 ADC 已使能、并且没有待处理的禁止 ADC 的请求（ADEN 必须等于 1 且 ADDIS 必须等于 0）时，才允许软件向 ADCx_CR 寄存器的控制位 ADSTART、JADSTART 和 ADDIS 执行写操作。

对于 ADCx_CFGR、ADCx_SMPRx、ADCx_SQRy、ADCx_JDRy、ADCx_OFRy、ADCx_OFCHRy 和 ADCx_IER 寄存器的所有其他控制位：

- 对于与常规转换配置相关的控制位，仅当 ADC 已使能 (ADEN=1) 且未进行常规转换（ADSTART 必须等于 0）时，才允许软件对这些位执行写操作。
- 对于与注入转换配置相关的控制位，仅当 ADC 已使能 (ADEN=1) 且未进行注入转换（JADSTART 必须等于 0）时，才允许软件对这些位执行写操作。

仅当 ADC 已使能（最终会进行转换）且没有待处理的禁止 ADC 的请求（ADSTART 或 JADSTART 必须等于 1 且 ADDIS 必须等于 0）时，软件才可以对 ADCx_CR 寄存器的 ADSTP 或 JADSTP 控制位执行写操作。

如果 ADC 已使能 (ADEN=1)，软件可随时对 ADCx_JSQR 寄存器执行写操作。

注： *没有硬件保护可以防止执行此类禁止的写操作，ADC 操作可能进入未知状态。要从此状态恢复，必须禁止 ADC（将 ADEN 清零，并且将 ADCx_CR 寄存器的所有位都清零）。*

21.4.11 通道选择（SQRx、JSQRx）

复用通道多达 20 条：

- 5 路来自 GPIO 焊盘的快速模拟输入 (ADC1_INP/INN[1..5])
- 最多 10 路来自 GPIO 焊盘的慢速模拟输入 (ADC1_INP/INN[6..16])。并非所有这些输入都连到 GPIO 的焊盘，具体取决于产品。
- ADC 连接到以下内部模拟输入：
  - 内部参考电压 (V_{REFINT}) 连接到 ADC1_INP0/INN0。
  - 内部温度传感器 (V_{SENSE}) 连接到 ADC1_INP17/INN17。
  - V_{BAT} 监测通道 (V_{BAT}/3) 连接到 ADC1_INP18/INN18。

注： *要对其中一条内部模拟通道进行转换，必须先对 ADCx_CCR 寄存器中的 VREFEN、CH17SEL 或 CH18SEL 位进行编程，以使能相应的模拟源。*

可以将转换分为两组：常规转换和注入转换。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：ADC1_INP/INN3、 ADC1_INP/INN8、ADC1_INP/INN2、 ADC1/INN2、 ADCx_INP/INN0、 ADC1_INP/INN2、 ADC1_INP/INN2、ADC1_INP/INN15。

- 一个**常规转换组**最多由 16 个转换构成。必须在 ADCx_SQRy 寄存器中选择转换序列的常规通道及其顺序。常规转换组中的转换总数必须写入 ADCx_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 ADCx_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 ADCx_JSQR 寄存器中的 L[1:0] 位。

不得在可能进行常规转换时对 ADCx_SQRy 寄存器进行修改。因此，必须先写入 ADSTP=1 停止 ADC 常规转换（请参见*第 21.4.17 节：停止正在进行的转换（ADSTP、JADSTP）*）。

进行注入转换时，可实时修改 ADCx_JSQR 寄存器。请参见第 21.4.21 节：注入转换的上下文队列。

21.4.12 可独立设置各通道采样时间（SMPR1、SMPR2）

开始转换之前，ADC 必须在待测量电压源与 ADC 内置采样电容之间建立直接连接。该采样时间必须足以使输入电压源为嵌入式电容充电至输入电压水平。

对各通道进行采样时可以使用不同的采样时间，采样时间可通过 ADCx_SMPR1 和 ADCx_SMPR2 寄存器中的 SMP[2:0] 位编程，可选采样时间值如下：

- SMP = 000：2.5 个 ADC 时钟周期
- SMP = 001：6.5 个 ADC 时钟周期
- SMP = 010：12.5 个 ADC 时钟周期
- SMP = 011：24.5 个 ADC 时钟周期
- SMP = 100：47.5 个 ADC 时钟周期
- SMP = 101：92.5 个 ADC 时钟周期
- SMP = 110：247.5 个 ADC 时钟周期
- SMP = 111：640.5 个 ADC 时钟周期

总转换时间的计算公式如下：

T_{CONV} = 采样时间 + 12.5 个 ADC 时钟周期

示例：

如果 F_{ADC_CLK} = 80 MHz，采样时间为 2.5 个 ADC 时钟周期：

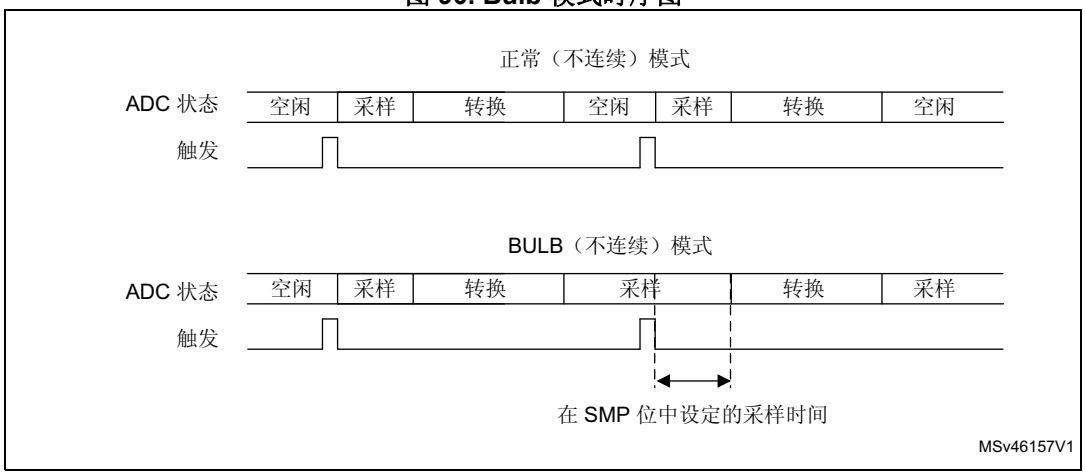
T_{CONV} = (2.5 + 12.5) 个 ADC 时钟周期 = 15 个 ADC 时钟周期 = 187.5 ns（对于快速通道）

ADC 通过将状态位 EOSMP 置 1 来指示采样阶段结束（仅限常规转换）。

快速通道和慢速通道的采样时间限制

对于每条通道，必须对 SMP[2:0] 位进行编程，以符合数据手册 ADC 特性部分规定的最短采样时间要求。

图 90. Bulb 模式时序图



I/O 模拟开关升压器

当 V_{DDA} 电压过低时，I/O 模拟开关电阻会增大，此时需要相应地调整采样时间（请参见数据手册中的电气特性）。可通过 SYSCFG_CFGR1 寄存器中的 BOOSTEN 位使能内部升压器，从而在低 V_{DDA} 条件下最大限度减小该电阻值。

21.4.13 单次转换模式 (CONT=0)

在单次转换模式下，ADC 会将通道的所有转换执行一次。CONT 位为 0 时，可通过以下方式启动此模式：

- 将 ADCx_CR 寄存器中的 ADSTART 位置 1（适用于常规通道）
- 将 ADCx_CR 寄存器中的 JADSTART 位置 1（适用于注入通道）
- 外部硬件触发事件（适用于常规通道或注入通道）

在常规序列中，每次转换完成后：

- 转换数据存储在 16 位 ADCx_DR 寄存器中
- EOC（常规转换结束）标志置 1
- EOCIE 位置 1 时将产生中断

在注入序列中，每次转换完成后：

- 转换数据存储在四个 16 位 ADCx_JDRy 寄存器的其中一个寄存器中
- JEOC（注入转换结束）标志置 1
- JEOCIE 位置 1 时将产生中断

常规序列完成后：

- EOS（常规序列结束）标志置 1
- EOSIE 位置 1 时将产生中断

注入序列完成后：

- JEOS（注入序列结束）标志置 1
- JEOSIE 位置 1 时将产生中断

随后，ADC 会停止工作，直至发生新的外部常规或注入触发，或者 ADSTART 或 JADSTART 位再次置 1。

注： *要转换单个通道，可将序列长度编程为 1。*

21.4.14 连续转换模式 (CONT=1)

该模式仅适用于常规通道。

在连续转换模式下，如果发生软件或硬件常规触发事件，ADC 会将通道的所有常规转换执行一次，随后会自动重启并持续执行序列的每个转换。CONT 位为 1 时，可通过外部触发或将 ADCx_CR 寄存器中的 ADSTART 位置 1 来启动此模式。

在常规序列中，每次转换完成后：

- 转换数据存储在 16 位 ADCx_DR 寄存器中
- EOC（转换结束）标志置 1
- EOCIE 位置 1 时将产生中断

转换序列完成后：

- EOS（序列结束）标志置 1
- EOSIE 位置 1 时将产生中断

随后，会立即重启新序列，ADC 会继续重复执行转换序列。

注： *要转换单个通道，可将序列长度编程为 1。*

不能同时使能不连续模式和连续模式：禁止同时将 DISCEN 和 CONT 位置 1。

注入通道不能连续转换，唯一例外的是，在连续转换模式下（使用 JAUTO 位）注入通道配置为在常规通道后的自动转换，请参见自动注入模式一节。

21.4.15 开始转换（ADSTART、JADSTART）

软件通过将 ADSTART 置 1 的方式开始进行 ADC 常规转换。

ADSTART 置 1 后，会开始进行转换：

- 立即开始转换：EXTEN = 0x0 时（软件触发）
- 在所选常规硬件触发的下一有效边沿开始转换：EXTEN 不等于 0x0 时

软件通过将 JADSTART 置 1 的方式开始进行 ADC 注入转换。

JADSTART 置 1 后，会开始进行转换：

- 立即开始转换：JEXTEN = 0x0 时（软件触发）
- 在所选注入硬件触发的下一有效边沿开始转换：JEXTEN 不等于 0x0 时

注： *在自动注入模式下 (JAUTO=1)，使用 ADSTART 位开始常规转换，然后再进行自动注入转换（JADSTART 必须保持清零）。*

ADSTART 位和 JADSTART 位还提供当前是否正在进行 ADC 操作的信息。可以在 ADSTART=0 且 JADSTART=0（指示 ADC 处于空闲状态）时重新配置 ADC。

ADSTART 通过硬件清零：

- 在使用软件常规触发的单次模式下（CONT=0，EXTSEL=0x0）
  - 如果 DISCEN = 1，只要常规转换序列结束（EOS 置 1）或子组处理结束就清零
- 在所有情况下（CONT=x、EXTSEL=x）
  - 执行由软件调用的 ADSTP 程序之后清零

注： *在连续模式下 (CONT=1)，由于序列会自动重新启动，因此，当 EOS 置 1 时，ADSTART 位不会通过硬件清零。*

如果在单次模式下选择了硬件触发（CONT=0 且 EXTSEL 不等于 0x00），当 EOS 置 1 时，ADSTART 位不会通过硬件清零，借此软件无需再次为下一个硬件触发事件复位 ADSTART。这样可确保不会错过任何后续的硬件触发。

JADSTART 通过硬件清零：

- 在使用软件注入触发的单次模式下 (JEXTSEL=0x0)
  - 如果 JDISCEN = 1，只要注入转换序列结束（JEOS 置 1）或子组处理结束就清零
- 在所有情况下 (JEXTSEL=x)
  - 执行由软件调用的 JADSTP 程序之后清零

注： *选择软件触发时，如果 EOC 标志仍为高电平，则不应将 ADSTART 位置 1。*

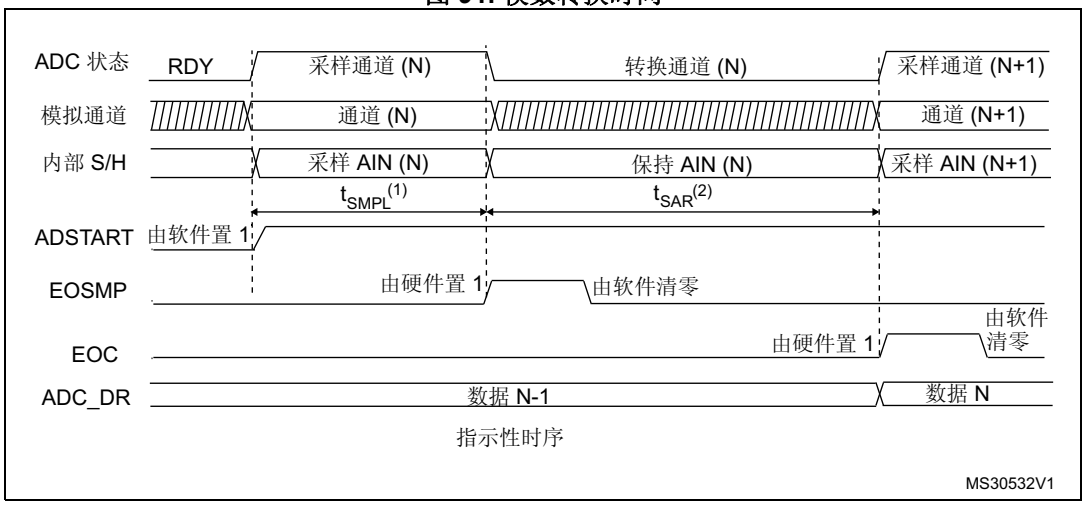
21.4.16 ADC 时序

从转换开始到转换结束所经过的时间是配置的采样时间与逐次逼近时间（具体取决于数据分辨率）的总和：

$$T_{CONV} = T_{SMPL} + T_{SAR} = [2.5\ _{|min} + 12.5\ _{|12\ 位}] \times T_{ADC_CLK}$$

$$T_{CONV} = T_{SMPL} + T_{SAR} = 31.25\ ns\ _{|min} + 156.25\ ns\ _{|12\ 位} = 187.5\ ns（对于\ F_{ADC_CLK} = 80\ MHz）$$

图 91. 模数转换时间



1. T_{SMPL} 取决于 SMP[2:0]
2. T_{SAR} 取决于 RES[2:0]

21.4.17 停止正在进行的转换（ADSTP、JADSTP）

软件决定是否停止转换，要停止正在进行的常规转换，应将 ADSTP 置 1；要停止正在进行的注入转换，应将 JADSTP 置 1。

停止转换将复位正在进行的 ADC 操作。随后可重新配置 ADC（例如：更改通道选择或触发），为新操作做好准备。

请注意，可以在常规转换仍在执行时停止注入转换，反之亦然。这样便可在常规转换仍在进行时重新配置注入转换序列和触发（反之亦然）。

如果 ADSTP 位由软件置 1，则会中止任何正在进行的常规转换，并会丢弃部分转换结果（ADCx_DR 寄存器不会更新为当前转换结果）。

如果 JADSTP 位由软件置 1，则会中止任何正在进行的注入转换，并会丢弃部分转换结果（ADCx_JDRy 寄存器不会更新为当前转换结果）。扫描序列也会中止并会复位（这意味着重启 ADC 将重新开始新的序列）。

该程序执行完毕后，ADSTP/ADSTART 位（常规转换时）或 JADSTP/JADSTART 位（注入转换时）会由硬件清零，软件必须轮询 ADSTART（或 JADSTART）直至其复位，然后才能判定 ADC 已完全停止运行。

注： *在自动注入模式下 (JAUTO=1)，将 ADSTP 位置 1 会中止常规转换和注入转换（不得使用 JADSTP）。*

图 92. 停止正在进行的常规转换

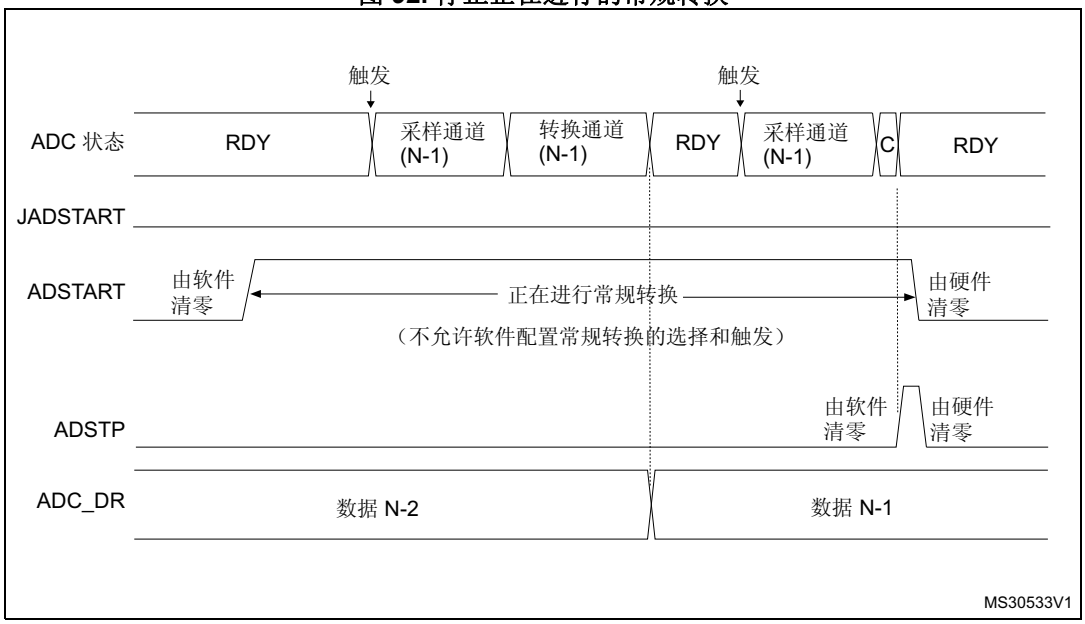
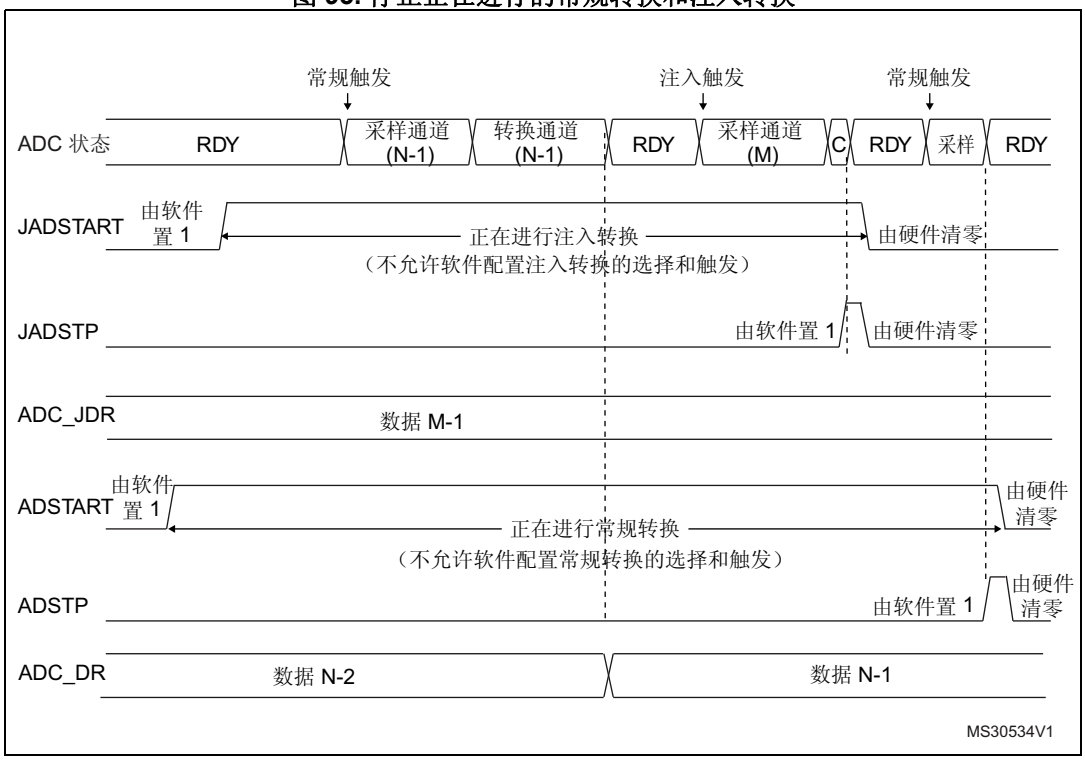


图 93. 停止正在进行的常规转换和注入转换



21.4.18 外部触发转换和触发极性（EXTSEL、EXTEN、JEXTSEL、JEXTEN）

可通过软件或外部事件（例如定时器捕获、输入引脚）触发转换或转换序列。如果 EXTEN[1:0] 控制位（对于常规转换）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。

注入队列使能时（位 JQDIS=0），不能实现注入软件触发。

软件将 ADSTART 位置 1 后，常规触发选择有效；软件将 JADSTART 位置 1 后，注入触发选择有效。

在转换进行时发生的硬件触发会被忽略。

- 如果位 ADSTART=0，则会忽略发生的任何常规硬件触发。
- 如果位 JADSTART=0，则会忽略发生的任何注入硬件触发。

表 114 提供了 EXTEN[1:0] 和 JEXTEN[1:0] 值与触发极性之间的对应关系。

表 114. 为常规外部触发配置触发极性

EXTEN[1:0]	源
00	禁止硬件触发检测，使能软件触发检测
01	在上升沿执行硬件触发检测
10	在下降沿执行硬件触发检测
11	在上升沿和下降沿均执行硬件触发检测

注： *不能实时更改常规触发的极性。*

表 115. 为注入外部触发配置触发极性

JEXTEN[1:0]	源
00	– 如果 JQDIS=1（队列禁止）：禁止硬件触发检测，使能软件触发检测 – 如果 JQDIS=0（队列使能）：同时禁止硬件和软件触发检测
01	在上升沿执行硬件触发检测
10	在下降沿执行硬件触发检测
11	在上升沿和下降沿均执行硬件触发检测

注： *如果队列已使能 (JQDIS=0)，可预计并实时更改注入触发的极性。请参见第 21.4.21 节：注入转换的上下文队列。*

EXTSEL 和 JEXTSEL 控制位用于从 16 个可能事件中选择可触发常规组转换和注入组转换的事件。

可通过注入触发中断常规组转换。

注： *不能实时更改常规触发选择。*
可以预计并实时更改注入触发选择。请参见第 574 页的第 21.4.21 节：注入转换的上下文队列。

表 116 到*表 117* 列出了用于常规转换和注入转换的三个 ADC 的所有可能存在的外部触发。

表 116. ADC1——常规通道的外部触发

名称	源	类型	EXTSEL[3:0]
EXT0	TIM1_CH1	片上定时器的内部信号	0000
EXT1	TIM1_CH2	片上定时器的内部信号	0001
EXT2	TIM1_CH3	片上定时器的内部信号	0010
EXT3	TIM2_CH2	片上定时器的内部信号	0011
EXT4	TIM3_TRGO	片上定时器的内部信号	0100
EXT5	TIM4_CH4	片上定时器的内部信号	0101
EXT6	EXTI 线 11	外部引脚	0110
EXT7	TIM8_TRGO	片上定时器的内部信号	0111
EXT8	TIM8_TRGO2	片上定时器的内部信号	1000
EXT9	TIM1_TRGO	片上定时器的内部信号	1001
EXT10	TIM1_TRGO2	片上定时器的内部信号	1010
EXT11	TIM2_TRGO	片上定时器的内部信号	1011
EXT12	TIM4_TRGO	片上定时器的内部信号	1100
EXT13	TIM6_TRGO	片上定时器的内部信号	1101
EXT14	TIM15_TRGO	片上定时器的内部信号	1110
EXT15	TIM3_CH4	片上定时器的内部信号	1111

表 117. ADC1——注入通道的外部触发

名称	源	类型	JEXTSEL[3..0]
JEXT0	TIM1_TRGO	片上定时器的内部信号	0000
JEXT1	TIM1_CH4	片上定时器的内部信号	0001
JEXT2	TIM2_TRGO	片上定时器的内部信号	0010
JEXT3	TIM2_CH1	片上定时器的内部信号	0011
JEXT4	TIM3_CH4	片上定时器的内部信号	0100
JEXT5	TIM4_TRGO	片上定时器的内部信号	0101
JEXT6	EXTI 线 15	外部引脚	0110
JEXT7	TIM8_CH4	片上定时器的内部信号	0111
JEXT8	TIM1_TRGO2	片上定时器的内部信号	1000
JEXT9	TIM8_TRGO	片上定时器的内部信号	1001
JEXT10	TIM8_TRGO2	片上定时器的内部信号	1010

表 117. ADC1——注入通道的外部触发（续）

名称	源	类型	JEXTSEL[3..0]
JEXT11	TIM3_CH3	片上定时器的内部信号	1011
JEXT12	TIM3_TRGO	片上定时器的内部信号	1100
JEXT13	TIM3_CH1	片上定时器的内部信号	1101
JEXT14	TIM6_TRGO	片上定时器的内部信号	1110
JEXT15	TIM15_TRGO	片上定时器的内部信号	1111

21.4.19 注入通道管理

触发注入模式

要使用触发注入，必须将 ADCx_CFGR 寄存器中的 JAUTO 位清零。

1. 通过外部触发或将 ADCx_CR 寄存器中的 ADSTART 位置 1 来启动常规通道组转换。
2. 如果在常规通道组转换期间出现外部注入触发或者 ADCx_CR 寄存器中的 JADSTART 位置 1，则当前的转换会复位，并会启动注入通道序列切换（所有注入通道都会转换一次）。
3. 然后，常规通道组的常规转换会从上次中断的常规转换处恢复。
4. 如果在注入转换期间出现常规事件，注入转换不会中断，但在注入序列结束时会执行常规序列。*图 94* 显示了相应的时序图。

注： *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 28 个 ADC 时钟周期（即，采样时间为 1.5 个时钟周期的两次转换），则触发事件的最小间隔不能小于 29 个 ADC 时钟周期。*

自动注入模式

如果将 ADCx_CFGR 寄存器中的 JAUTO 位置 1，则注入组中的通道会在常规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 ADCx_SQRy 和 ADCx_JSQR 寄存器中编程。

在该模式下，必须将 ADCx_CR 寄存器中的 ADSTART 位置 1 以开始常规转换，然后再进行注入转换（JADSTART 必须保持清零）。将 ADSTP 位置 1 会中止常规转换和注入转换（不得使用 JADSTP 位）。

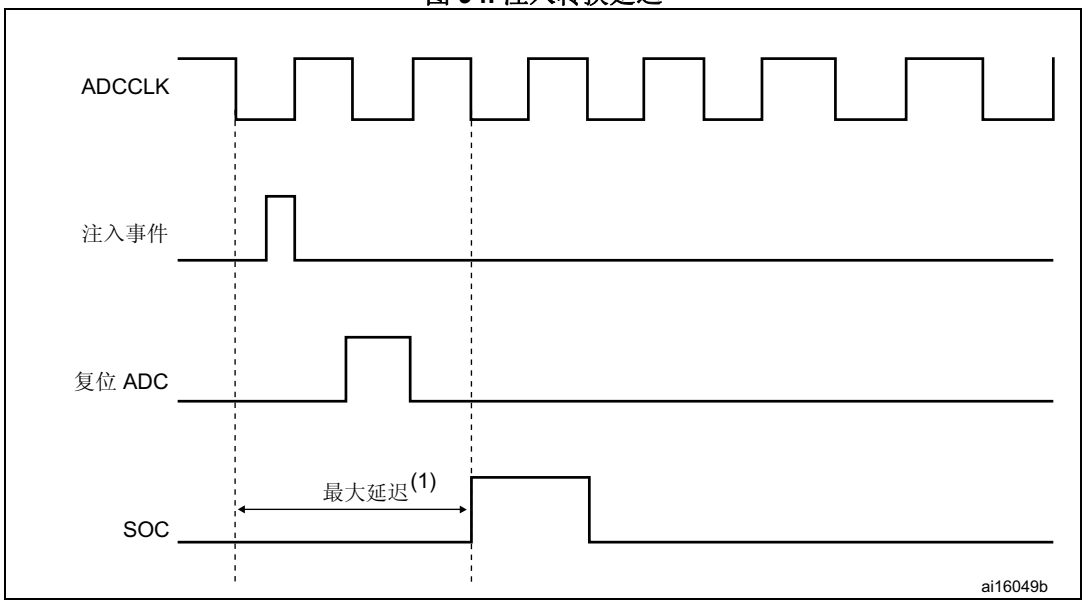
在此模式下，必须禁止注入通道上的外部触发。

如果 CONT 位和 JAUTO 位均已置 1，则常规通道及随后的注入通道会被连续转换。

注： *不能同时使用自动注入和不连续采样模式。*

如果在 JAUTO 模式下使用 DMA 导出常规定序器的数据，需要将其设定为循环模式（将 DMA_CCRx 寄存器中的 CIRC 位置 1）。如果 CIRC 位复位（单发模式），JAUTO 序列将在出现 DMA 传输完成事件时停止。

图 94. 注入转换延迟



1. 有关延迟的最大值，请参见 STM32L4Rxxx 和 STM32L4Sxxx 数据手册中的电气特性部分。

21.4.20 不连续模式（DISCEN、DISCNUM、JDISCEN）

常规组模式

可将 ADCx_CFGR 寄存器中的 DISCEN 位置 1 来使能此模式。

该模式用于转换含有 n (n ≤ 8) 个转换的短序列（子组），该短序列是在 ADCx_SQRy 寄存器中选择的转换序列的一部分。可通过写入 ADCx_CFGR 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 ADCx_SQRy 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 ADCx_SQR1 寄存器中的 L[3:0] 位定义总序列长度。

示例：

- DISCEN=1，n = 3，要转换的通道 = 1、2、3、6、7、8、9、10、11
  - 第一次触发：转换的通道为 1、2、3（每次转换时都生成 EOC 事件）。
  - 第二次触发：转换的通道为 6、7、8（每次转换时都生成 EOC 事件）。
  - 第三次触发：转换的通道为 9、10、11（每次转换时都生成 EOC 事件），并会在通道 11 转换完成后生成 EOS 事件。
  - 第四次触发：转换的通道为 1、2、3（每次转换时都生成 EOC 事件）。
  - ...

- DISCEN=0，要转换的通道 = 1、2、3、6、7、8、9、10、11
  - 第一次触发：转换整个序列：通道 1、然后是通道 2、3、6、7、8、9、10 和 11。每次转换都会生成 EOC 事件，最后一次转换还会生成 EOS 事件。
  - 所有后续触发事件都将重启整个序列。

注： *在不连续模式下转换常规组时，不会出现翻转（序列最后一个子组的转换次数少于 n 次）。*

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，第四次触发重新转换了第一个子组中的通道 1、2 和 3。

不能同时使能不连续模式和连续模式。如果同时使能两种模式（即 DISCEN=1、CONT=1），ADC 会认定连续模式已禁止并继续执行相关操作。

注入组模式

可将 ADCx_CFGR 寄存器中的 JDISCEN 位置 1 来使能此模式。在出现外部注入触发事件之后，该模式会逐通道转换在 ADCx_JSQR 寄存器中选择的序列，相当于不连续模式下常规通道“n”固定为 1 的情况。

出现外部触发时，将启动在 ADCx_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 ADCx_JSQR 寄存器中的 JL[1:0] 位定义总序列长度。

示例：

- JDISCEN=1，要转换的通道 = 1、2、3
  - 第一次触发：转换通道 1（生成 JEOC 事件）
  - 第二次触发：转换通道 2（生成 JEOC 事件）
  - 第三次触发：转换通道 3 并生成 JEOC 事件和 JEOS 事件
  - ...

注： *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 1。*

不能同时使用自动注入模式和不连续模式：当 JAUTO 置 1 时，DISCEN 和 JDISCEN 位必须通过软件保持清零状态。

21.4.21 注入转换的上下文队列

实现上下文队列可为下一个注入转换序列准备多达 2 个上下文。必须将 ADCx_CFGR 寄存器的 JQDIS 位复位才能使能此功能。上下文队列使能时，只能进行硬件触发转换。

该上下文包括：

- 注入触发的配置（ADCx_JSQR 寄存器中的 JEXTEN[1:0] 位和 JEXTSEL 位）
- 注入序列的定义（ADCx_JSQR 寄存器中的 JSQx[4:0] 位和 JL[1:0] 位）

上下文的所有参数都会在 ADCx_JSQR 这一寄存器中定义，该寄存器会实现一个双缓冲区队列，可缓冲多达 2 组参数：

- JSQR 寄存器可随时写入，正在进行注入转换时也不例外。
- 每个写入到 JSQR 寄存器中的数据均会存储在上下文队列中。
- 队列开始时为空，对 JSQR 寄存器进行的第一次写访问时会立即更改上下文，随即 ADC 会准备好接收注入触发。
- 注入序列完成后，队列会被占用，上下文会根据队列中存储的后续 JSQR 参数进行更改。这一新的上下文会用于下一个注入转换序列。

- 如果在队列已满的情况下向 JSQR 寄存器执行写操作，会发生队列溢出。这种溢出情况会通过 JQOVF 标志置为有效来指示。发生溢出时，会忽略造成溢出的 JSQR 寄存器写访问，并且上下文队列保持不变。如果 JQOVFIE 位置 1，可产生中断。
- 队列变空时可能执行两种操作，具体取决于寄存器 ADCx_CFGR 的控制位 JQM 的值。
  - 如果 JQM=0，队列刚好在使能 ADC 后为空，但在随后的运行操作期间绝不会变空：队列始终保留上一个有效的上下文，并会根据上一个有效的上下文处理后续的有效注入序列启动。
  - 如果 JQM=1，队列会在注入序列结束后或队列被清空时变空。这种情况下，队列中不存在任何上下文，硬件触发也会被禁止。因此会忽略后续的所有硬件注入触发，直至软件重新向 JSQR 寄存器写入新的注入上下文。
- 读取 JSQR 寄存器会返回当前有效的 JSQR 上下文。如果 JSQR 上下文为空，JSQR 的读出值为 0x0000。
- 如果通过将 JADSTP 置 1 的方式停止注入转换、或者通过将 ADDIS 置 1 的方式禁止 ADC，队列会被清空。
  - 如果 JQM=0，队列会保留上一个有效的上下文。
  - 如果 JQM=1，队列会变空，并会忽略触发。

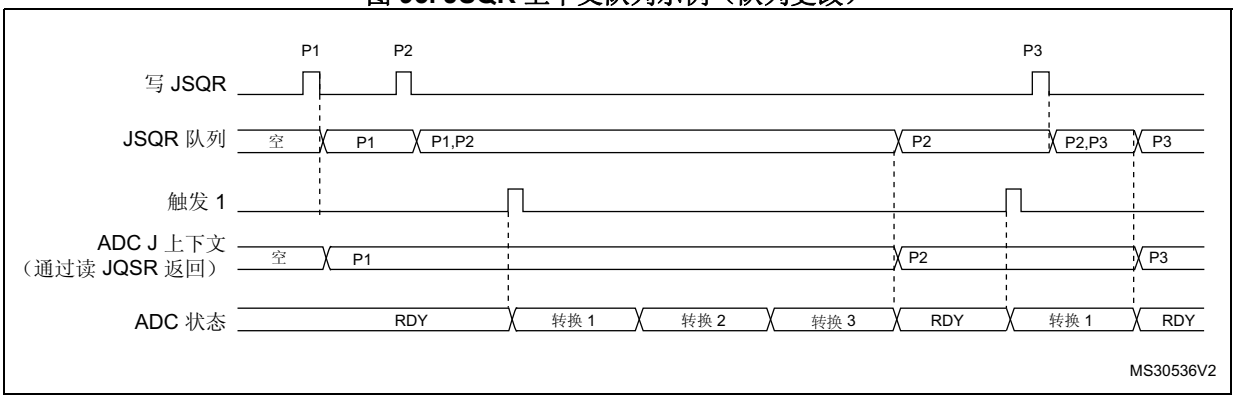
注： *如果配置为不连续模式（位 JDISCEN=1），只有注入序列的最后一次触发会更改上下文并占用队列。第一次触发仅会占用队列，但其他触发仍为有效触发，如下文中的不连续模式示例所示（两个上下文的长度均为 3）：*

- *第一次触发，不连续。序列 1：上下文 1 已占用，已执行第一次转换。*
- *第二次触发，不连续。序列 1：第二次转换。*
- *第三次触发，不连续。序列 1：第三次转换。*
- *第四次触发，不连续。序列 2：上下文 2 已占用，已执行第一次转换。*
- *第五次触发，不连续。序列 2：第二次转换。*
- *第六次触发，不连续。序列 2：第三次转换。*

更改触发或序列上下文时的操作

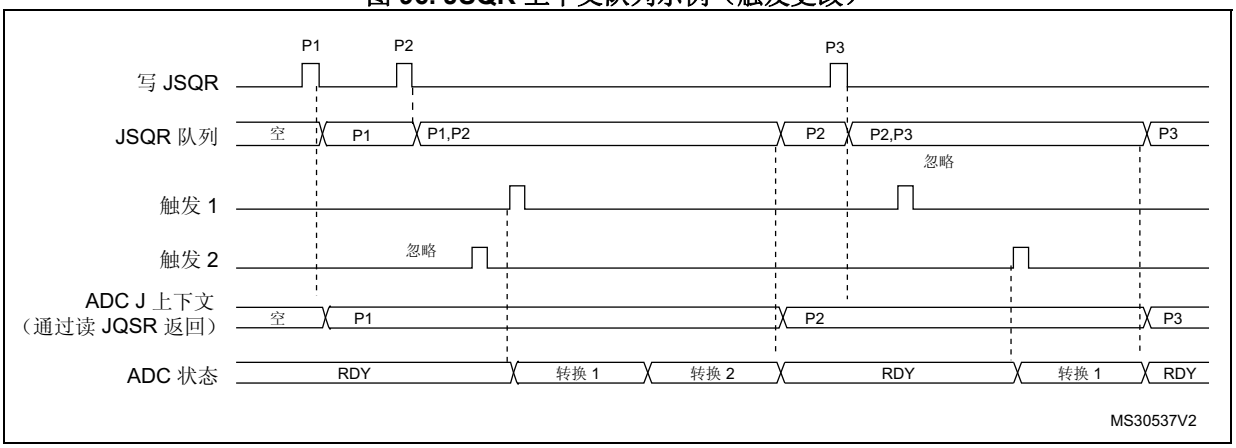
图 95 和*图 96* 显示了更改序列或触发时上下文队列的操作。

图 95. JSQR 上下文队列示例（队列更改）



1. 参数：
   P1：3 个转换组成的序列，硬件触发 1
   P2：1 个转换组成的序列，硬件触发 1
   P3：4 个转换组成的序列，硬件触发 1

图 96. JSQR 上下文队列示例（触发更改）

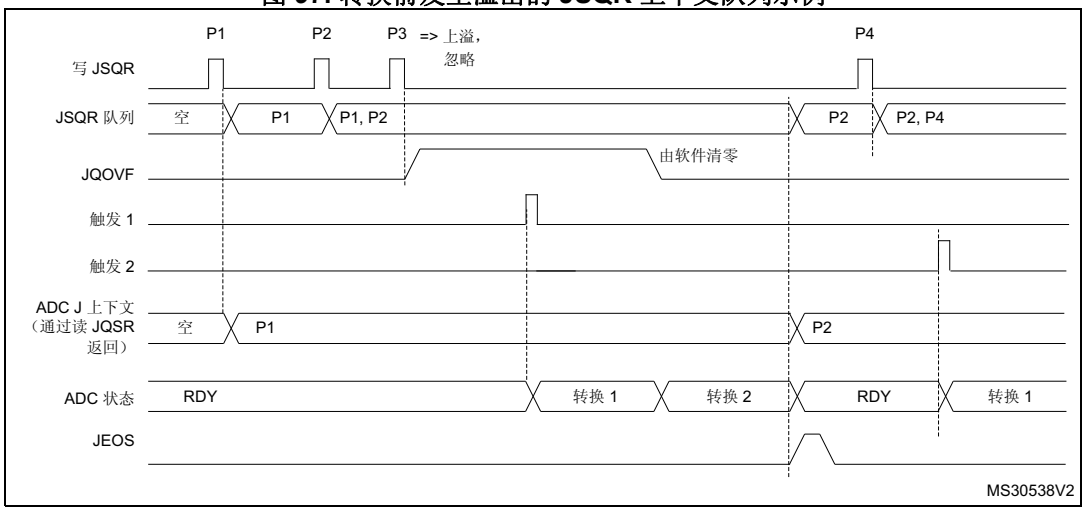


1. 参数：
   P1：2 个转换组成的序列，硬件触发 1
   P2：1 个转换组成的序列，硬件触发 2
   P3：4 个转换组成的序列，硬件触发 1

上下文队列：发生队列溢出时的操作

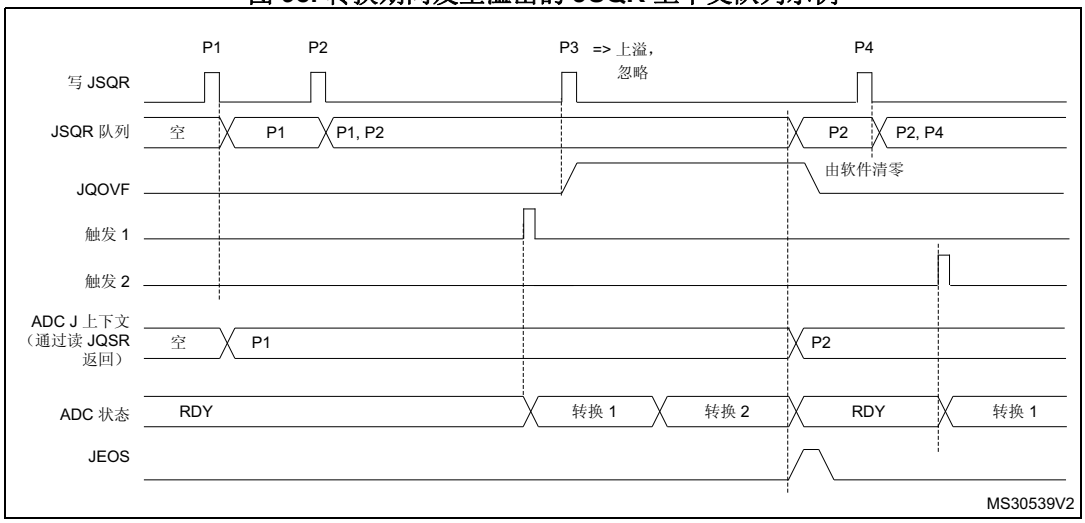
图 97 和*图 98* 显示了转换之前或之后发生溢出时上下文队列的操作。

图 97. 转换前发生溢出的 JSQR 上下文队列示例



1. 参数：
   P1：2 个转换组成的序列，硬件触发 1
   P2：1 个转换组成的序列，硬件触发 2
   P3：3 个转换组成的序列，硬件触发 1
   P4：4 个转换组成的序列，硬件触发 1

图 98. 转换期间发生溢出的 JSQR 上下文队列示例



1. 参数：
   P1：2 个转换组成的序列，硬件触发 1
   P2：1 个转换组成的序列，硬件触发 2
   P3：3 个转换组成的序列，硬件触发 1
   P4：4 个转换组成的序列，硬件触发 1

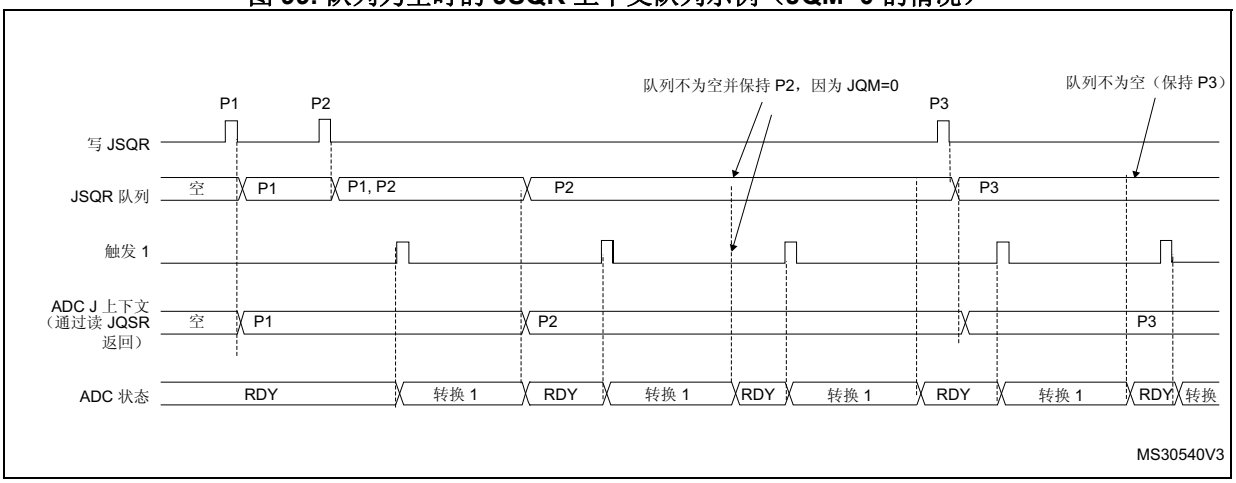
建议按照下述方法管理队列溢出：

- 将每个 P 上下文写入 JSQR 寄存器后，通过标志 JQOVF 指示写操作是否已被忽略（可生成中断）。
- 为避免队列溢出，仅在上一个上下文 P2 的 JEOS 标志置 1 后写入第三个上下文 (P3)。这样可确保上一个上下文已被占用且队列未满。

上下文队列：队列变空时的操作

图 99 和*图 100* 显示了在 JQM=0 或 JQM=1 这两种情况下队列变空时上下文队列的操作。

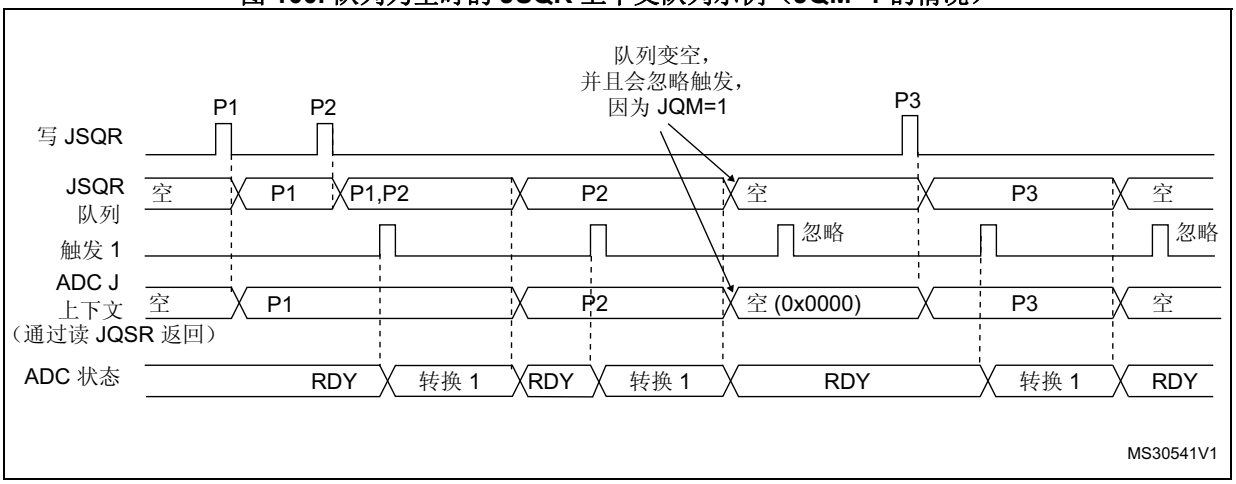
图 99. 队列为空时的 JSQR 上下文队列示例（JQM=0 的情况）



1. 参数：
   P1：1 个转换组成的序列，硬件触发 1
   P2：1 个转换组成的序列，硬件触发 1
   P3：1 个转换组成的序列，硬件触发 1

注： *写入 P3 时，上下文会立即改变。但由于内部重新同步的原因，会存在一定的延迟，如果刚好在写入 P3 之后或之前发生触发，可能会发生启动的转换被视为上下文 P2 的情况。为了避免出现这种情况，用户必须确保写入立即应用的新上下文时没有发生 ADC 触发。*

图 100. 队列为空时的 JSQR 上下文队列示例（JQM=1 的情况）

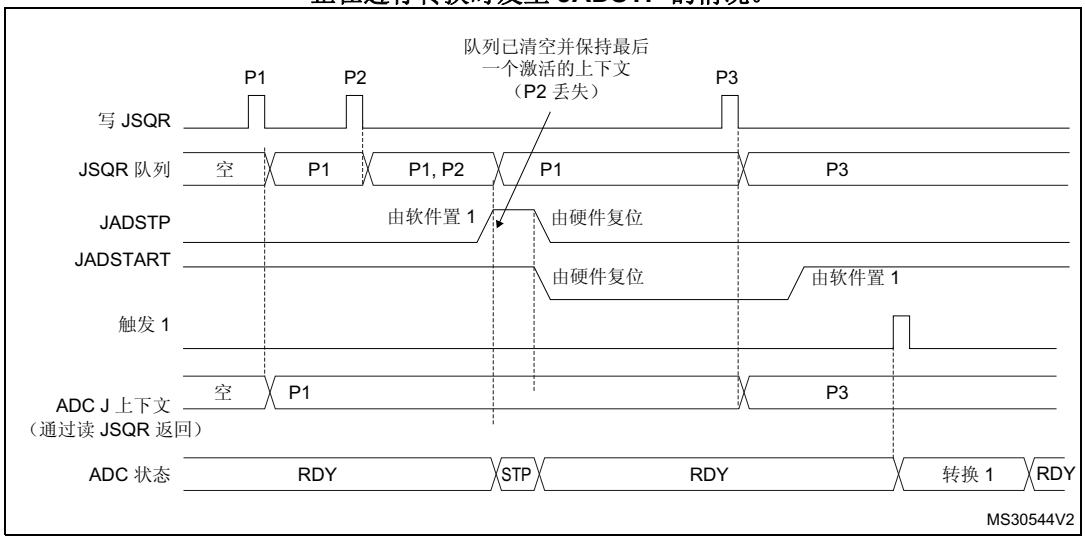


1. 参数：
   P1：1 个转换组成的序列，硬件触发 1
   P2：1 个转换组成的序列，硬件触发 1
   P3：1 个转换组成的序列，硬件触发 1

清空上下文队列

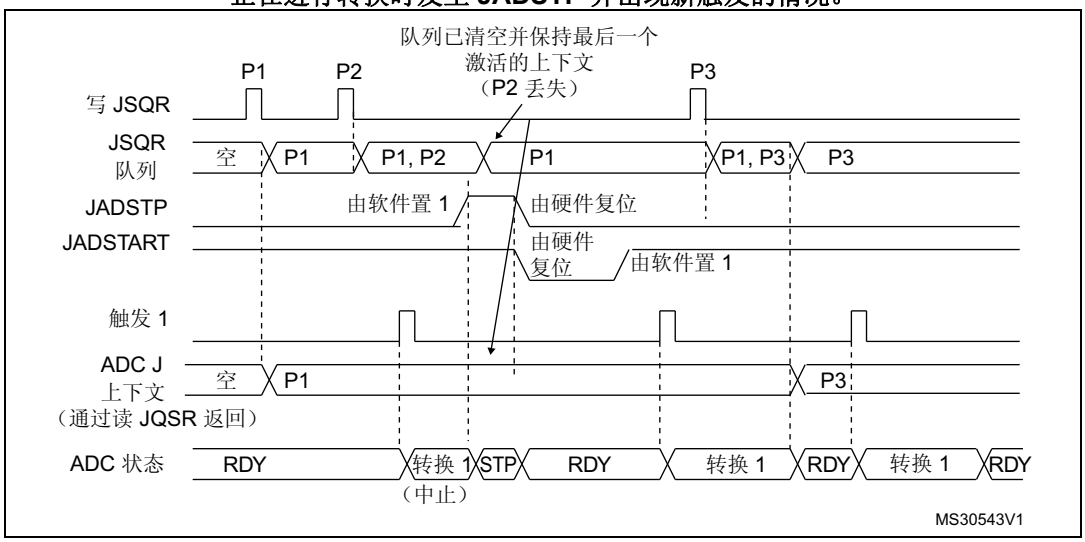
下图显示的是各种情况下清空队列时上下文队列的操作。

图 101. 通过将 JADSTP 置 1 的方式清空 JSQR 上下文队列 (JQM=0)。正在进行转换时发生 JADSTP 的情况。



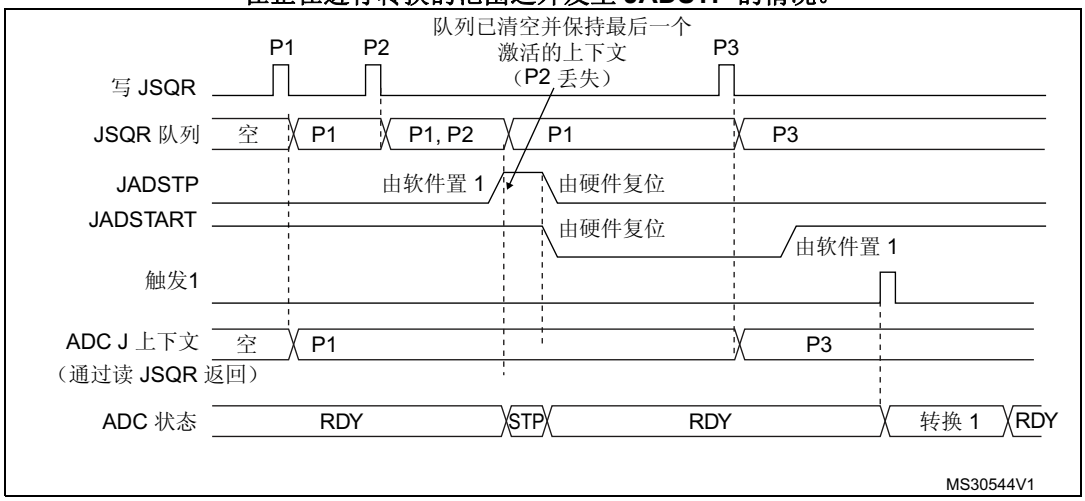
1. 参数：
   P1：1 个转换组成的序列，硬件触发 1
   P2：1 个转换组成的序列，硬件触发 1
   P3：1 个转换组成的序列，硬件触发 1

图 102. 通过将 JADSTP 置 1 的方式清空 JSQR 上下文队列 (JQM=0)。正在进行转换时发生 JADSTP 并出现新触发的情况。



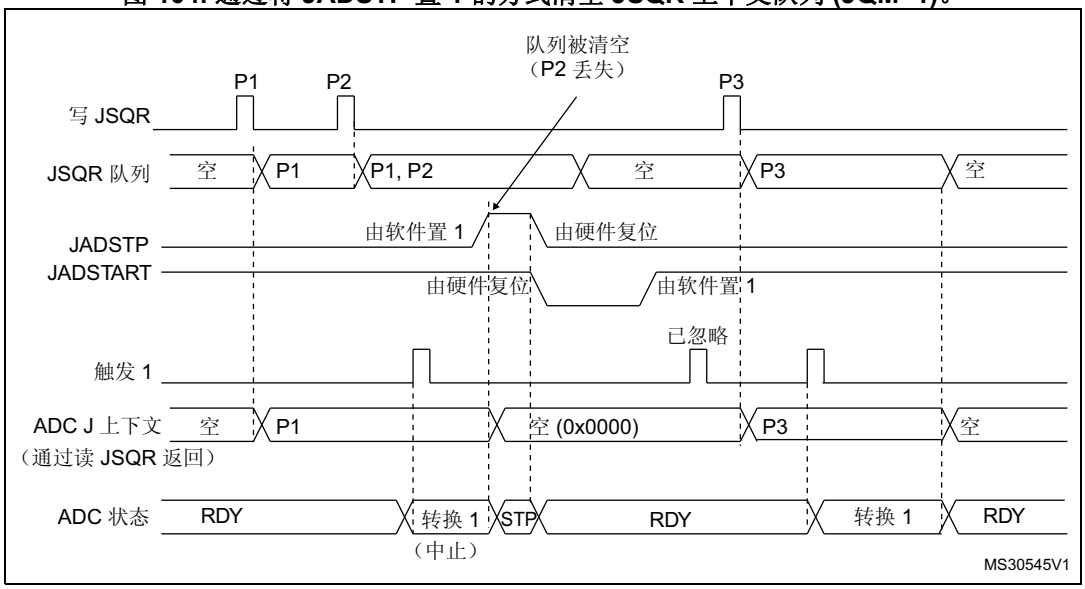
1. 参数：
   P1：1 个转换组成的序列，硬件触发 1
   P2：1 个转换组成的序列，硬件触发 1
   P3：1 个转换组成的序列，硬件触发 1

图 103. 通过将 JADSTP 置 1 的方式清空 JSQR 上下文队列 (JQM=0)。在正在进行转换的范围之外发生 JADSTP 的情况。



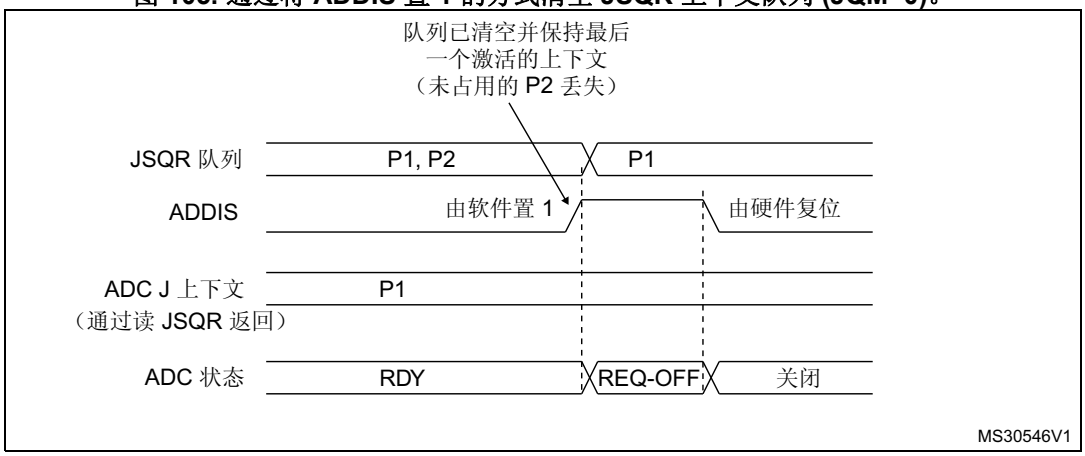
1. 参数：
   P1：1 个转换组成的序列，硬件触发 1
   P2：1 个转换组成的序列，硬件触发 1
   P3：1 个转换组成的序列，硬件触发 1

图 104. 通过将 JADSTP 置 1 的方式清空 JSQR 上下文队列 (JQM=1)。



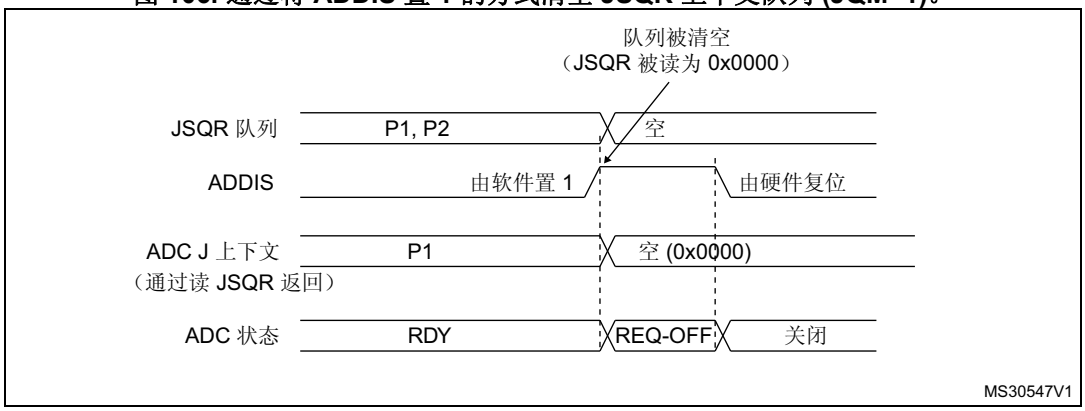
1. 参数：
   P1：1 个转换组成的序列，硬件触发 1
   P2：1 个转换组成的序列，硬件触发 1
   P3：1 个转换组成的序列，硬件触发 1

图 105. 通过将 ADDIS 置 1 的方式清空 JSQR 上下文队列 (JQM=0)。



1. 参数：
   P1：1 个转换组成的序列，硬件触发 1
   P2：1 个转换组成的序列，硬件触发 1
   P3：1 个转换组成的序列，硬件触发 1

图 106. 通过将 ADDIS 置 1 的方式清空 JSQR 上下文队列 (JQM=1)。



1. 参数：
   P1：1 个转换组成的序列，硬件触发 1
   P2：1 个转换组成的序列，硬件触发 1
   P3：1 个转换组成的序列，硬件触发 1

上下文队列：在队列为空时启动 ADC

要在队列为空时启动 ADC 操作，必须按照以下程序进行操作，以免在 ADC 初始化时无法获悉第一个上下文。此程序仅在 JQM 位复位的情况下适用：

1. 写入空 JSQR，JEXTEN 不等于 0（否则会触发软件转换）
2. 将 JADSTART 置 1
3. 将 JADSTP 置 1
4. 等待，直至 JADSTART 复位
5. 将 JADSTART 置 1

禁止队列

可通过将 ADCx_CFGR 寄存器中的 JQDIS 位置 1 的方式禁止队列。

21.4.22 可编程分辨率 (RES)——快速转换模式

可通过降低 ADC 分辨率来执行快速转换。

通过对控制位 RES[1:0] 进行编程，可将分辨率配置为 12 位、10 位、8 位或 6 位。*图 111*、*图 112*、*图 113* 和*图 114* 显示了转换结果格式与分辨率和数据对齐方式的对应关系。

对于不要求使用高数据精度的应用，分辨率越低，转换时间越短，降低分辨率可缩短逐次逼近步骤所需的转换时间，如*表 118* 所示。

表 118. T_{SAR} 与分辨率的对应关系

RES（位）	T_{SAR}（ADC 时钟周期）	F_{ADC}=80 MHz 时的 T_{SAR} (ns)	T_{CONV}（ADC 时钟周期）（采样时间 = 2.5 个 ADC 时钟周期）	F_{ADC}=80 MHz 时的 T_{CONV} (ns)
12	12.5 个 ADC 时钟周期	156.25 ns	15 个 ADC 时钟周期	187.5 ns
10	10.5 个 ADC 时钟周期	131.25 ns	13 个 ADC 时钟周期	162.5 ns
8	8.5 个 ADC 时钟周期	106.25 ns	11 个 ADC 时钟周期	137.5 ns
6	6.5 个 ADC 时钟周期	81.25 ns	9 个 ADC 时钟周期	112.5 ns

21.4.23 转换结束、采样阶段结束（EOC、JEOC、EOSMP）

每次出现常规转换结束 (EOC) 事件和注入转换结束 (JEOC) 事件时，ADC 都会通知应用。

新的常规转换数据出现在 ADCx_DR 寄存器中后，ADC 会立即将 EOC 标志置 1。如果 EOCIE 位置 1，可产生中断。EOC 标志可通过由软件向其写入 1 或读取 ADCx_DR 的方式来清零。

新的注入转换数据出现在 ADCx_JDRy 寄存器中后，ADC 会立即将 JEOC 标志置 1。如果 JEOCIE 位置 1，可产生中断。JEOC 标志可通过由软件向其写入 1 或读取相应 ADCx_JDRy 寄存器的方式来清零。

ADC 还通过将状态位 EOSMP 置 1 来指示采样阶段结束（仅限常规转换）。EOSMP 标志可通过由软件向其写入 1 的方式来清零。如果 EOSMPIE 位置 1，可产生中断。

21.4.24 转换序列结束（EOS、JEOS）

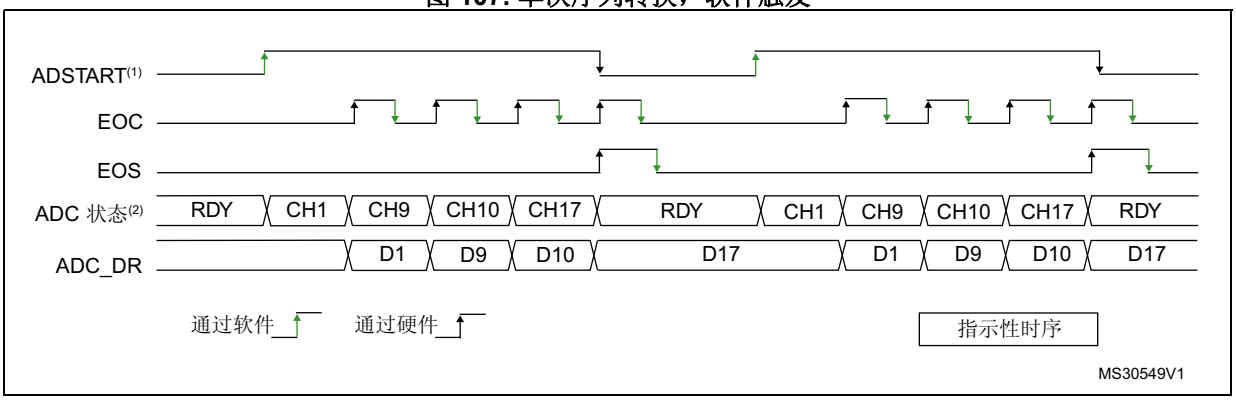
每次出现常规序列结束 (EOS) 事件和注入序列结束 (JEOS) 事件时，ADC 都会通知应用。

常规转换序列的最后一个数据出现在 ADCx_DR 寄存器中时，ADC 会立即将 EOS 标志置 1。如果 EOSIE 位置 1，可产生中断。EOS 标志可通过由软件向其写入 1 的方式来清零。

注入转换序列的最后一个数据完成时，ADC 会立即将 JEOS 标志置 1。如果 JEOSIE 位置 1，可产生中断。JEOS 标志可通过由软件向其写入 1 的方式来清零。

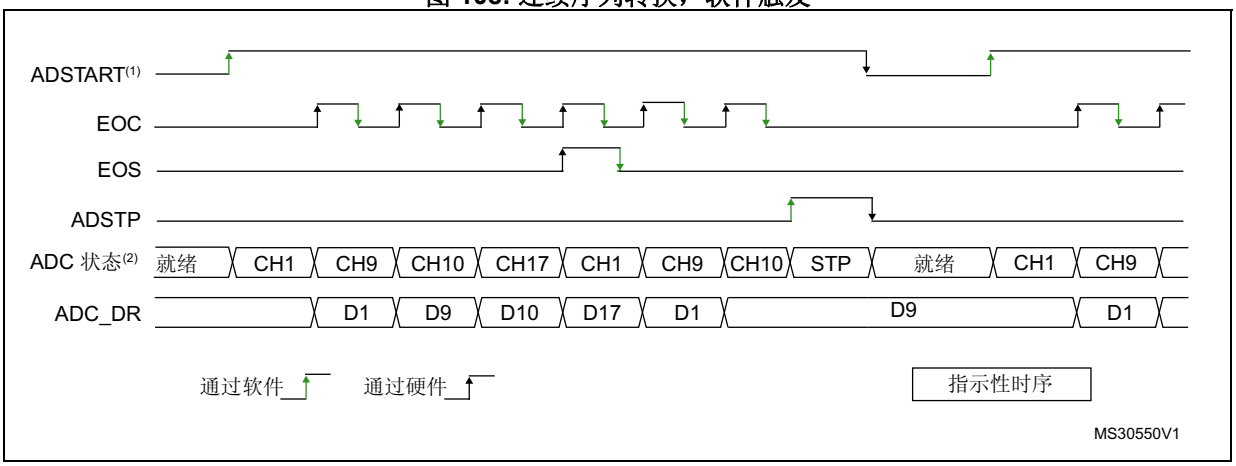
21.4.25 时序图示例（单次模式/连续模式，硬件/软件触发）

图 107. 单次序列转换，软件触发



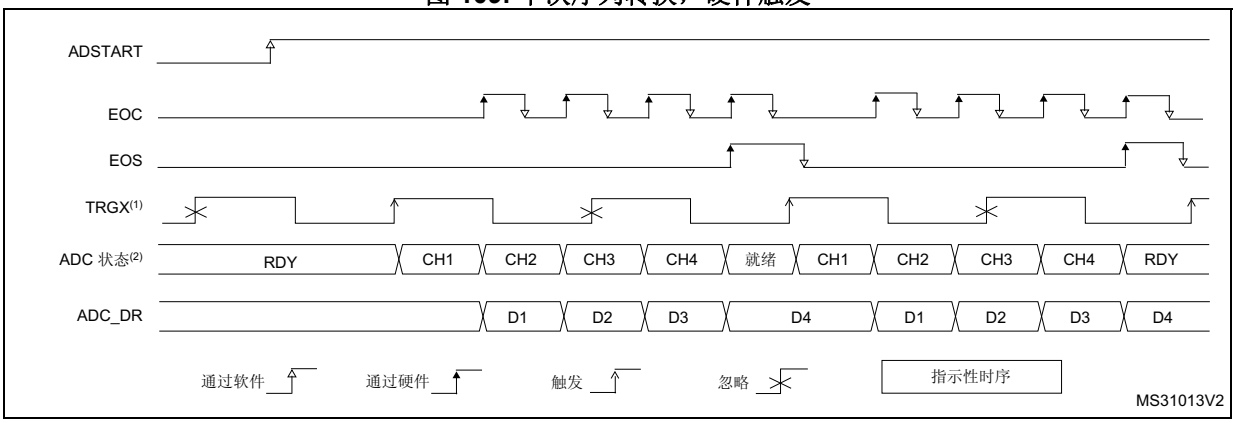
1. EXTEN=0x0，CONT=0
2. 所选通道 = 1、9、10、17；AUTDLY=0

图 108. 连续序列转换，软件触发



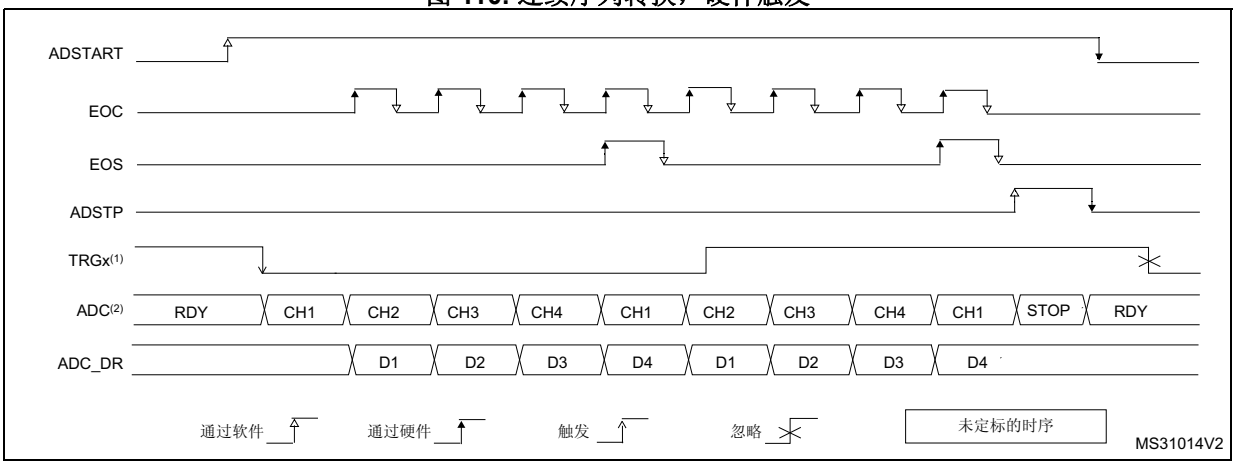
1. EXTEN=0x0，CONT=1
2. 所选通道 = 1、9、10、17；AUTDLY=0

图 109. 单次序列转换，硬件触发



1. 选择 TRGx（过频）作为触发源，EXTEN = 01，CONT = 0。
2. 所选通道 = 1、2、3、4；AUTDLY=0。

图 110. 连续序列转换，硬件触发



1. 选择 TRGx 作为触发源，EXTEN = 10，CONT = 1。
2. 所选通道 = 1、2、3、4；AUTDLY=0。

21.4.26 数据管理

数据寄存器、数据对齐和偏移（ADCx_DR、OFFSETy、OFFSETy_CH、ALIGN）

数据和对齐

每次常规转换通道结束时（发生 EOC 事件时），转换后数据的结果都会存储在宽度为 16 位的 ADCx_DR 数据寄存器中。

每次注入转换通道结束时（发生 JEOC 事件时），转换后数据的结果都会存储在宽度为 16 位的相应 ADCx_JDRy 数据寄存器中。

ADCx_CFGR 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 111*、*图 112*、*图 113*、和*图 114* 所示。

特例：采用左对齐时，数据基于半字对齐，分辨率设置为 6 位时除外。当分辨率设置为 6 位时，数据基于字节对齐，如*图 113* 和*图 114* 所示。

注： *过采样模式下不支持左对齐。当 ROVSE 和/或 JOVSE 位置 1 时，忽略 ALIGN 位的值，并且 ADC 仅提供右对齐数据。*

偏移

可以通过将 ADCx_OFRy 寄存器中的 OFFSETy_EN 置 1，对通道应用偏移 y（y=1、2、3、4）。应用偏移的通道会编程到 ADCx_OFRy 寄存器的 OFFSETy_CH[4:0] 位中。这种情况下，转换后的值会减去写入到 OFFSETy[11:0] 位中的用户自定义偏移。结果可能是负值，因此读取的数据为有符号值，SEXT 位代表扩展符号值。

注： *过采样模式下不支持偏移校准。如果 ROVSE 和/或 JOVSE 位置 1，ADCx_OFRy 寄存器中 OFFSETy_EN 位的值会被忽略（视为复位）。*

表 121 介绍了如何对模拟看门狗 1 的所有可能的分辨率进行比较。

表 119. 偏移计算与数据分辨率

分辨率（位 RES[1:0]）	原始转换数据与偏移相减		结果	注释
	原始转换数据，左对齐	偏移		
00：12 位	DATA[11:0]	OFFSET[11:0]	有符号 12 位数据	-
01：10 位	DATA[11:2]，00	OFFSET[11:0]	有符号 10 位数据	用户必须将 OFFSET[1:0] 配置为“00”
10：8 位	DATA[11:4]，0000	OFFSET[11:0]	有符号 8 位数据	用户必须将 OFFSET[3:0] 配置为“0000”
11：6 位	DATA[11:6]，000000	OFFSET[11:0]	有符号 6 位数据	用户必须将 OFFSET[5:0] 配置为“000000”

从对应于通道“i”的 ADCx_DR（常规通道）或 ADCx_JDRy（注入通道，y=1、2、3、4）读取数据时：

- 如果相应通道的其中一个偏移已使能（位 OFFSETy_EN=1），则读取的数据为有符号数据。
- 如果此通道的四个偏移均未使能，则读取的数据为无符号数据。

图 111、*图 112*、*图 113* 和*图 114* 中显示了有符号数据和无符号数据的对齐方式。

图 111. 右对齐（偏移禁止，无符号值）

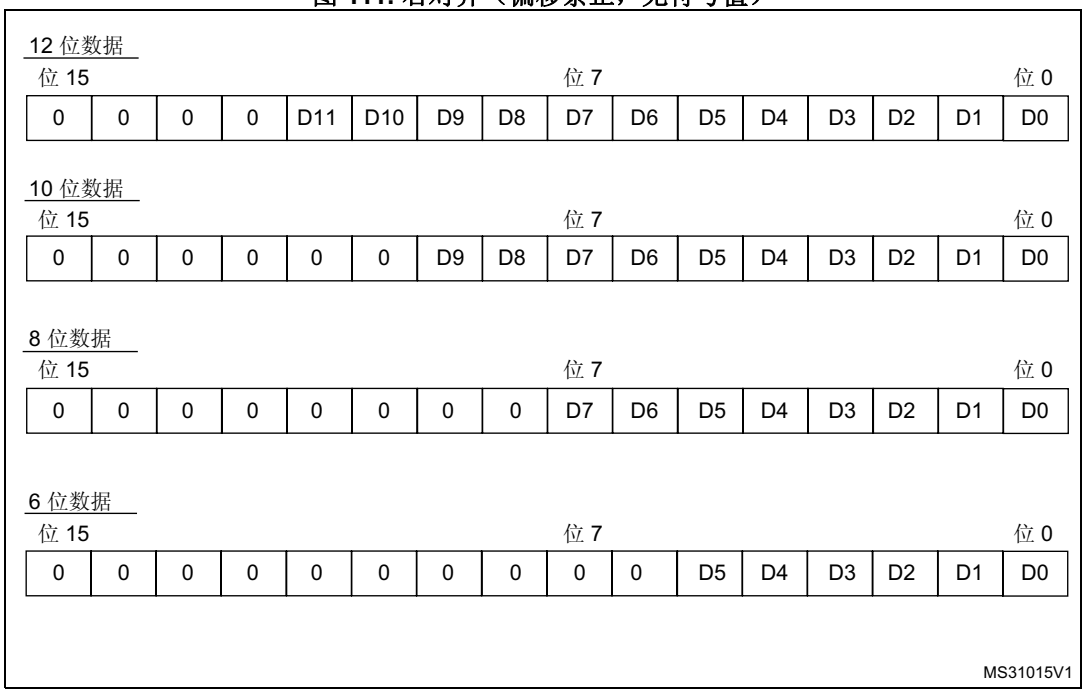


图 112. 右对齐（偏移使能，有符号值）

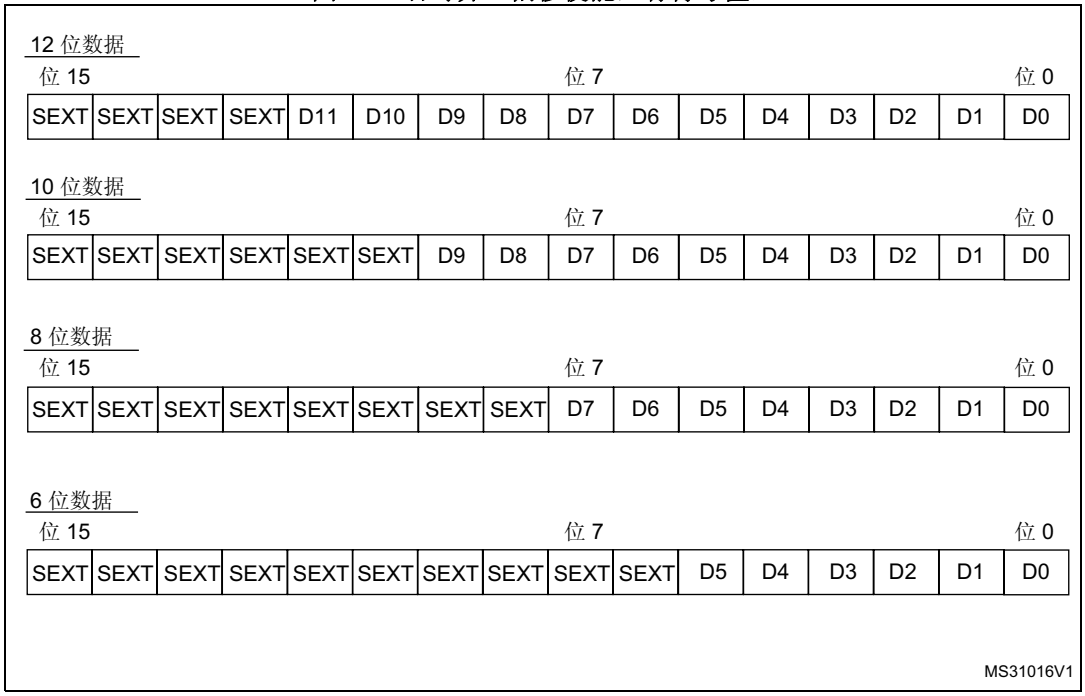


图 113. 左对齐（偏移禁止，无符号值）

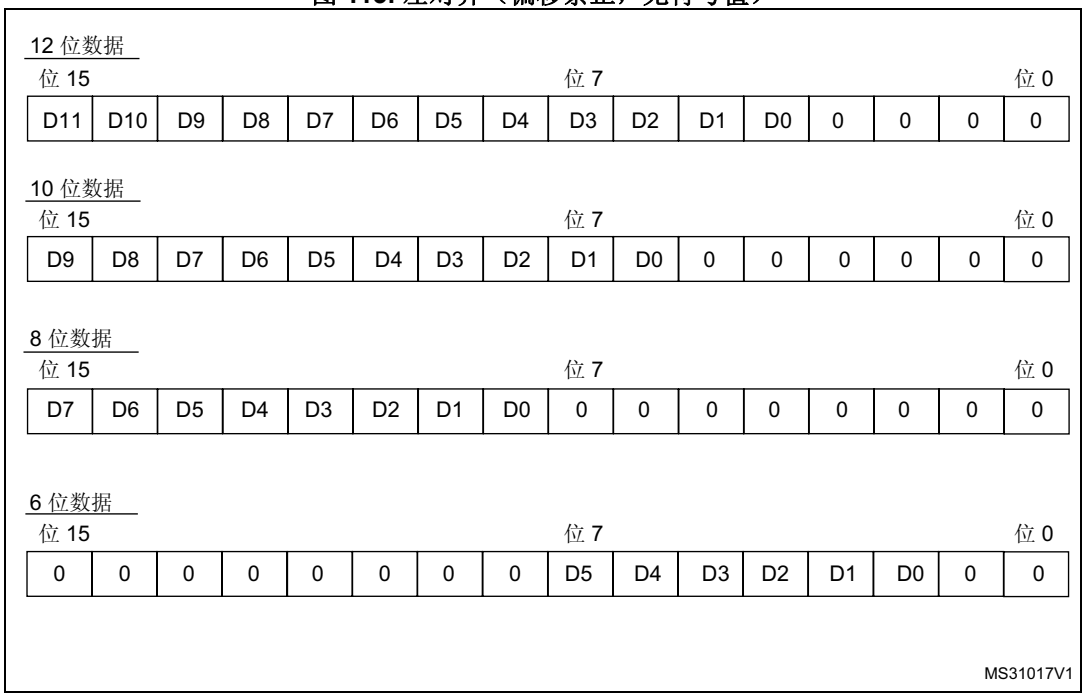
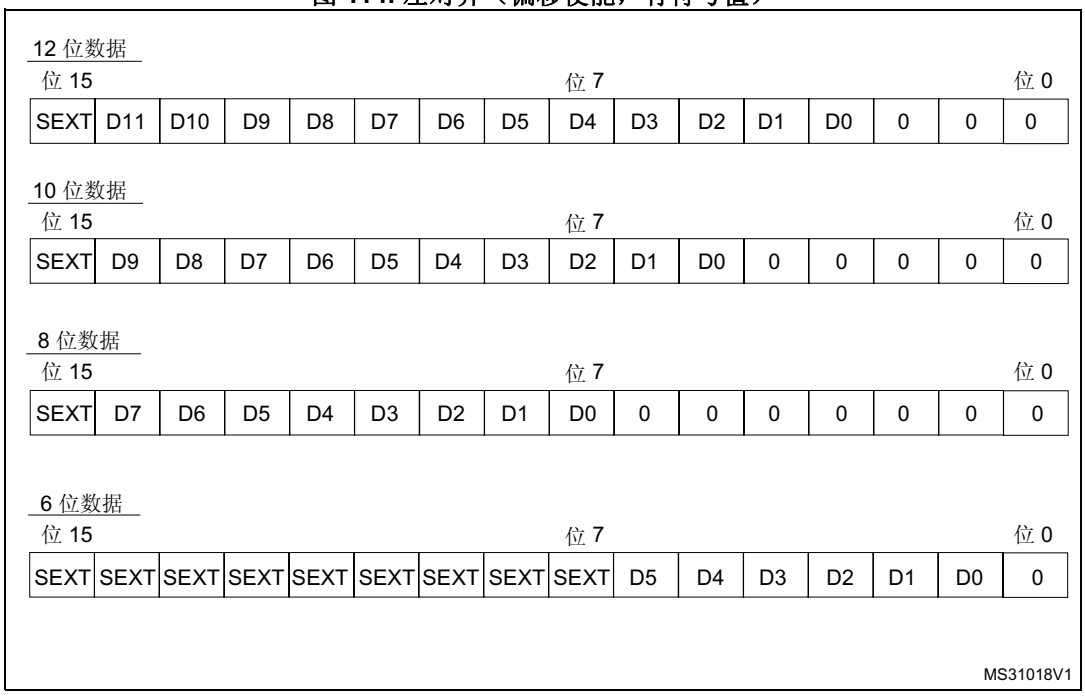


图 114. 左对齐（偏移使能，有符号值）



ADC 溢出（OVR、OVRMOD）

如果常规转换后的数据未在新转换数据可用之前（由 CPU 或 DMA）读取，会由溢出标志 (OVR) 指示缓冲区溢出事件。

如果新转换完成时 EOC 标志仍为 1，则 OVR 标志会置 1。如果 OVRIE 位置 1，可产生中断。

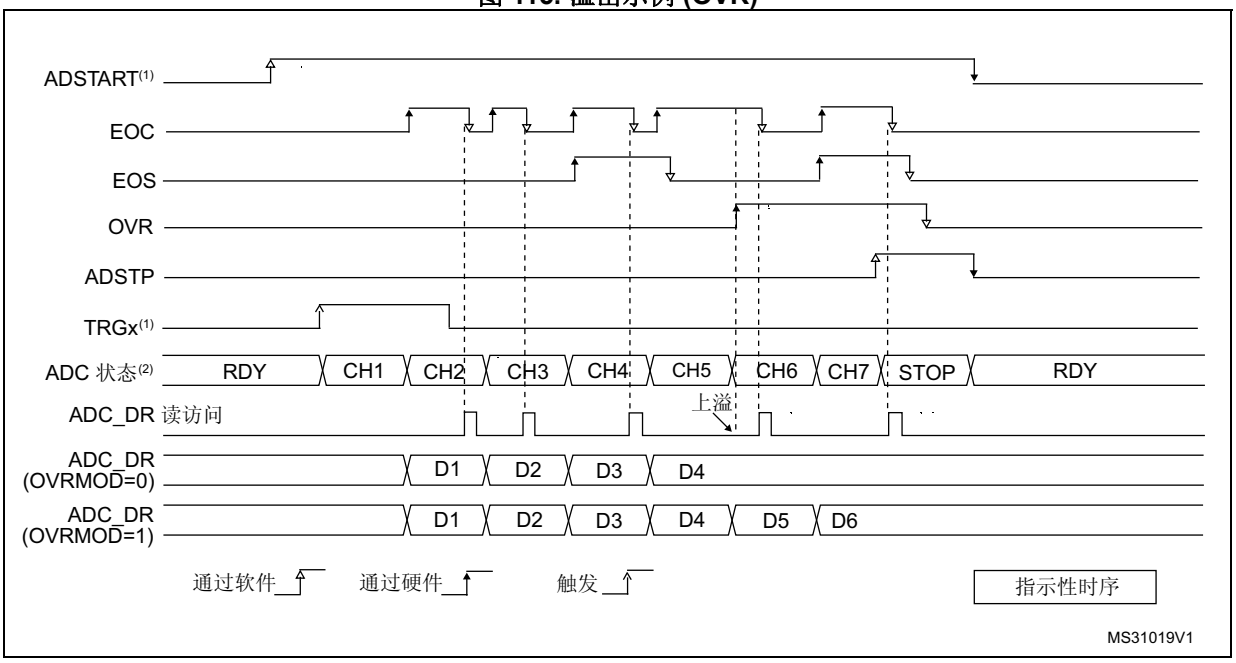
如果发生溢出情况，ADC 仍会保持工作状态并可继续进行转换，除非通过软件将 ADSTP 位置 1，从而停止并复位序列。

OVR 标志可通过由软件向其写入 1 的方式来清零。

可对控制位 OVRMOD 进行编程，从而配置发生溢出事件时是要保留数据还是要覆盖数据：

- OVRMOD=0：溢出事件会保留数据寄存器的数据，防止其被覆盖：会保留原数据，并会丢弃新的转换结果。如果 OVR 保持为 1，可继续进行转换，但还是会丢弃结果数据。
- OVRMOD=1：数据寄存器会由上一次转换结果覆盖，之前未读取的数据会丢失。如果 OVR 保持为 1，可继续正常进行转换，并且 ADCx_DR 寄存器将始终包含最新的转换数据。

图 115. 溢出示例 (OVR)



注：　由于四条注入通道均有专用的数据寄存器，因此不会对注入通道进行溢出检测。

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，软件必须使用 EOC 标志及其相关中断来处理各个数据。每当转换结束时，EOC 都会置 1，并且可以读取 ADCx_DR 寄存器。OVRMOD 应配置为 0，以便将溢出事件作为错误进行管理。

在不使用 DMA 且不发生溢出的情况下管理转换

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（例如，存在模拟看门狗时）。在这种情况下，OVRMOD 位必须配置为 1，OVR 标志应被软件忽略。溢出事件不会阻止 ADC 继续进行转换，ADCx_DR 寄存器始终包含最新的转换。

使用 DMA 管理转换

由于转换得出的通道值会存储到唯一的数据寄存器中，因此，对于多个通道的转换，使用 DMA 非常有帮助。这样可以避免丢失已存储在 ADCx_DR 寄存器中的数据。

若 DMA 模式已使能（ADCx_CFGR 寄存器中的 DMAEN 位置 1），则会在每个通道转换后生成 DMA 请求。这样便可将转换的数据从 ADCx_DR 寄存器传输到用软件选择的目标位置。

尽管如此，如果因 DMA 无法及时处理 DMA 传输请求而发生溢出 (OVR=1)，ADC 会停止生成 DMA 请求，新转换对应的数据不会通过 DMA 进行传输。这意味着可将传输到 RAM 的所有数据都视为有效数据。

根据 OVRMOD 位的配置，可保留或覆盖数据（请参见 *ADC 溢出（OVR、OVRMOD）*一节）。

DMA 传输请求会禁止，直至软件将 OVR 位清零。

根据应用用途的不同，推荐使用两种不同的 DMA 模式，并使用 ADCx_CFGR 寄存器的 DMACFG 位配置相应的模式：

- DMA 单次模式 (DMACFG=0)
  如果通过编程将 DMA 设置为传输固定数目的数据，应选择此模式。
- DMA 循环模式 (DMACFG=1)
  如果在循环模式下对 DMA 进行编程，应选择此模式。

DMA 单次模式 (DMACFG=0)

在该模式下，每次出现新的转换数据时，ADC 都会生成 DMA 传输请求，DMA 到达最后一个 DMA 传输操作时（发生 DMA_EOT 中断，请参见 DMA 一段），即使转换已再次开始，ADC 也会停止生成 DMA 请求。

DMA 传输完成后（在 DMA 控制器中配置的所有传输操作均已完成后）：

- ADC 数据寄存器的内容会冻结。
- 任何正在进行的转换都会中止，其部分结果会被丢弃。
- 不会将任何新的 DMA 请求发送到 DMA 控制器。如果仍存在已开始的转换，这样可避免生成溢出错误。
- 扫描序列会停止并复位。
- DMA 会停止。

DMA 循环模式 (DMACFG=1)

在该模式下，每次数据寄存器中出现新的转换数据时，ADC 都会生成 DMA 传输请求，即使 DMA 已到达最后一次 DMA 传输操作也不例外。这样可将处于循环模式的 DMA 配置为处理连续的模拟输入数据流。

21.4.27 使用 DFSDM 管理转换

ADC 转换结果可直接传送到 ΣΔ 滤波器 (DFSDM)。

在这种情况下，DFSDMCFG 位必须置 1，DMAEN 位必须清 0。

ADC 会将常规数据寄存器数据的所有 16 位传输到 DFSDM，传输完成后，会立即复位 EOC 标志。

数据必须为 16 位有符号格式：

ADCx_DR[15:12] = 符号扩展

ADCx_DR[11] = 符号

ADCx_DR[11:0] = 数据

要在 12 位 ADC 模式下实现 16 位有符号格式，软件需要在 OFFSETy_EN 置 1 后将 OFFSETy[11:0] 配置为 0x800。

仅右对齐数据格式适用于 DFSDM 接口（请参见*图 112：右对齐（偏移使能，有符号值）*）。

21.4.28 动态低功耗特性

自动延迟转换模式 (AUTDLY)

ADC 会执行由 AUTDLY 配置位控制的自动延迟转换模式。自动延迟转换可用于简化软件，并可优化采用低频时钟的应用（此类应用可能存在 ADC 溢出的风险）的性能。

AUTDLY=1 时，仅当同一组内所有之前的数据已处理完毕时，才能开始新的转换：

- 对于常规转换：ADCx_DR 寄存器已读取或者 EOC 位已清零时（请参见*图 116*）。
- 对于注入转换：JEOS 位已清零时（请参见*图 117*）。

通过这种方式，可自动调整 ADC 的速度，使其适应系统读取数据的速度。

延时插入到每次常规转换后（无论 DISCEN=0 还是 1）和每个注入转换序列后（无论 JDISCEN=0 还是 1）。

注： *不会在注入转换序列之间插入延时，但会在最后一个序列之后插入延时。*

转换过程中，此延时期间发生的硬件触发事件（针对同一组转换）会被忽略。

注： *如果延时期间发生软件触发，则不会被忽略，仍可在该延时时间内通过将 ADSTART 或 JADSTART 位置 1 的方式重新启动转换；启动新转换之前，会由软件读取数据。*

不会在不同组的转换之间插入延时（常规转换后即进行注入转换，反之亦然）：

- 如果在常规转换的自动延时期间发生注入触发，注入转换会立即开始（请参见*图 117*）。
- 注入序列完成后，ADC 会等待上一常规转换的延时（如果未结束），然后才会启动新的常规转换（请参见*图 119*）。

自动注入模式 (JAUTO=1) 下的操作略有不同，仅当上一注入转换序列的自动延时已结束时（JEOS 已清零时），才能开始进行新的常规转换。这是为了确保软件可在启动新序列之前读取给定序列的所有数据（请参见*图 120*）。

要在连续自动注入和自动延时组合模式下停止转换（JAUTO=1、CONT=1 且 AUTDLY=1），请按照以下程序进行操作：

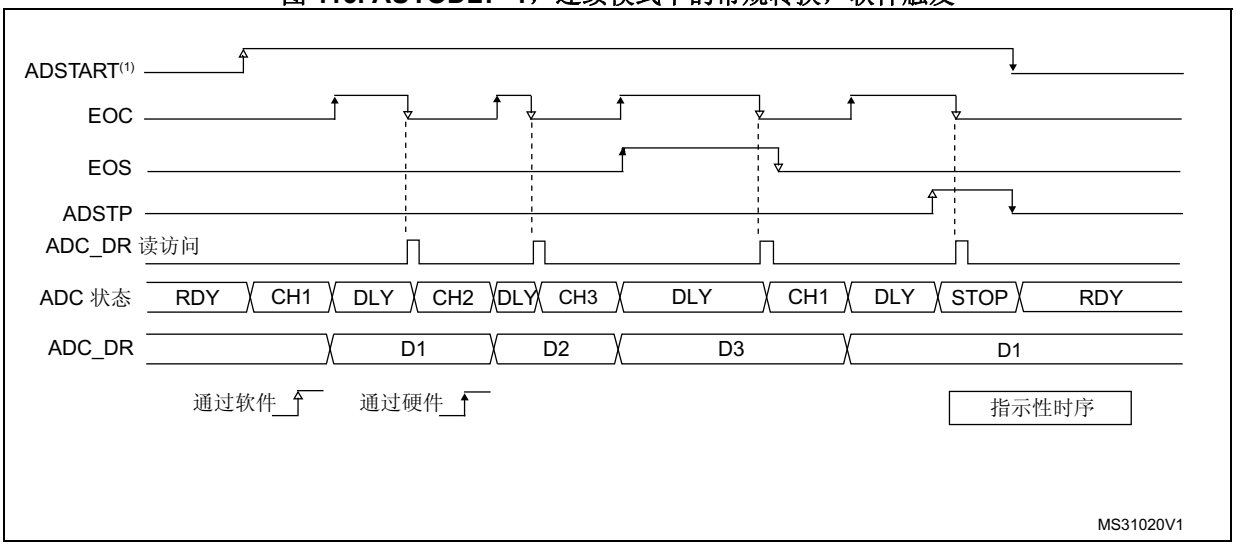
1. 等待，直至 JEOS=1（不会重新启动任何转换）
2. 将 JEOS 清零
3. 将 ADSTP 置 1
4. 读取常规数据

否则，如果 JEOS 在 ADSTP 已置 1 后清零，可能重新启动新的常规序列。

在 AUTDLY 模式下，如果在正在进行的常规序列期间或序列的最后一次常规转换之后的延时期间发生硬件常规触发事件，则会忽略该触发事件。但是，如果触发事件发生在该延时之后，即使发生在随后的注入序列的延时期间，也会将该触发事件视为待处理事件。随后，会在注入序列的延时结束时开始转换。

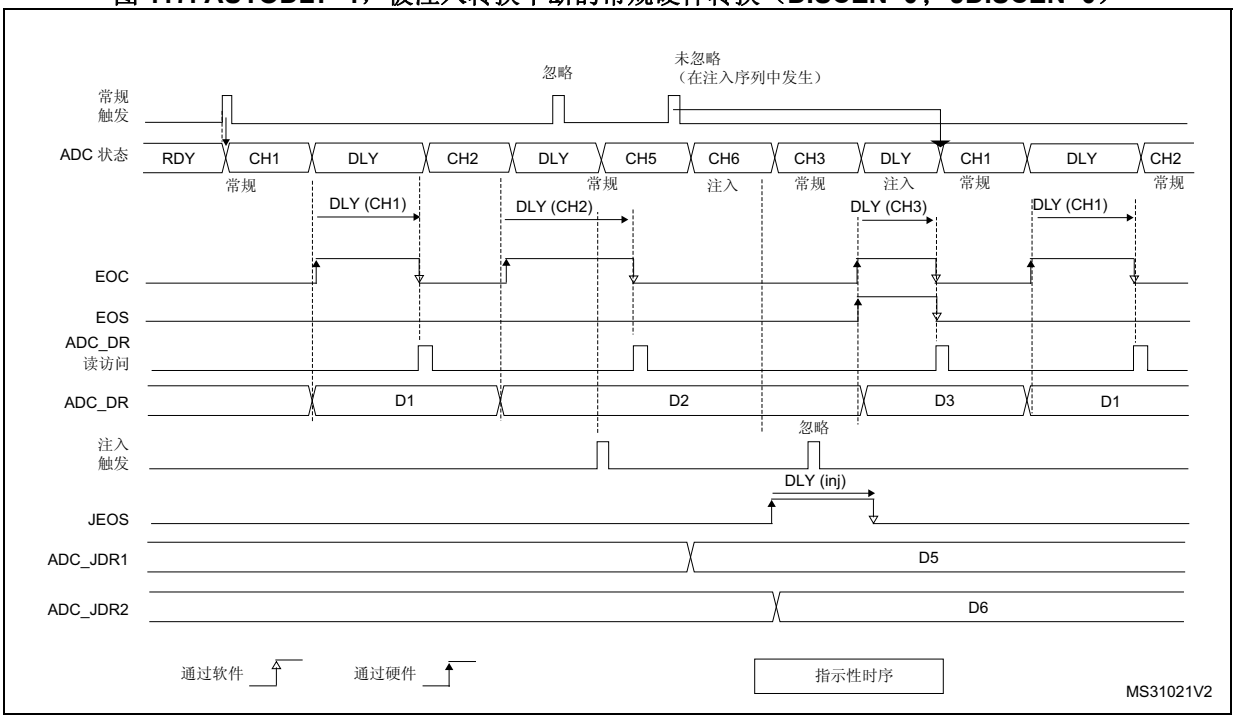
在 AUTDLY 模式下，如果在正在进行的注入序列期间或序列的最后一次注入转换之后的延时期间发生硬件注入触发事件，则会忽略该触发事件。

图 116. AUTODLY=1，连续模式下的常规转换，软件触发



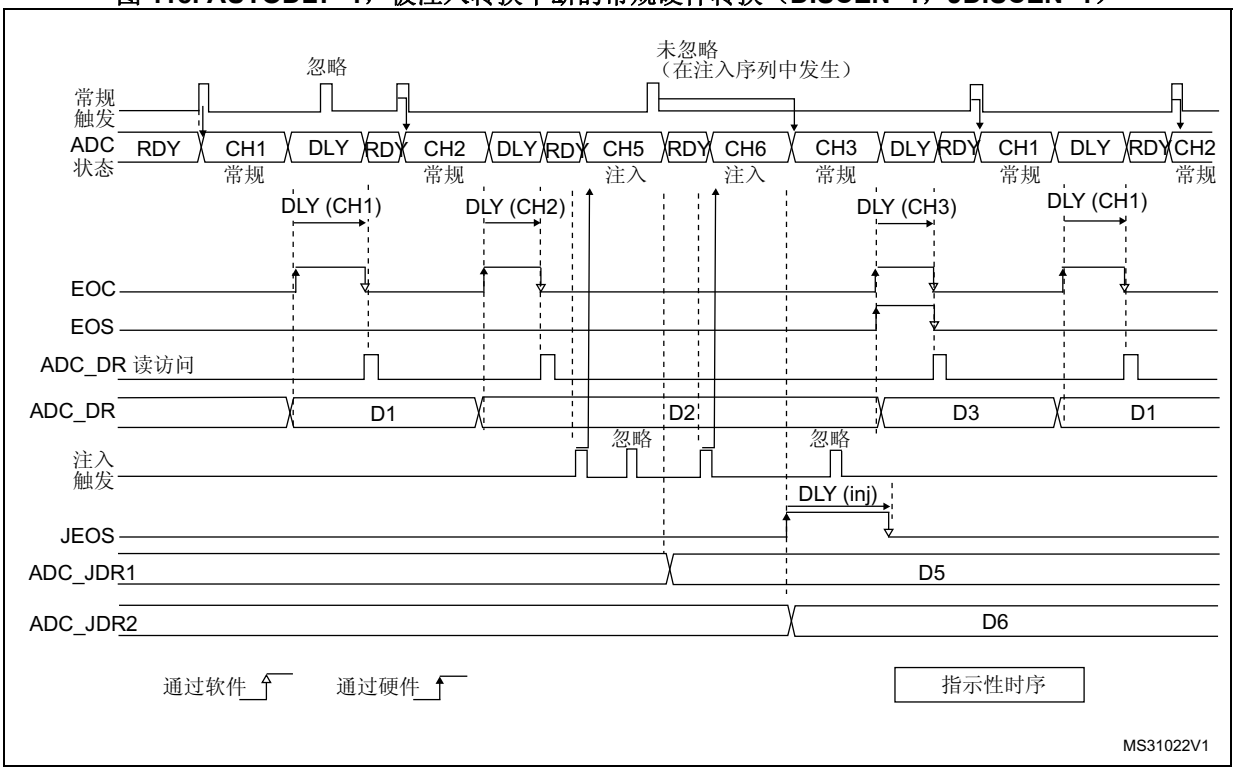
1. AUTDLY=1
2. 常规配置：EXTEN=0x0（软件触发），CONT=1，CHANNELS = 1、2、3
3. 注入配置禁止

图 117. AUTODLY=1，被注入转换中断的常规硬件转换（DISCEN=0；JDISCEN=0）



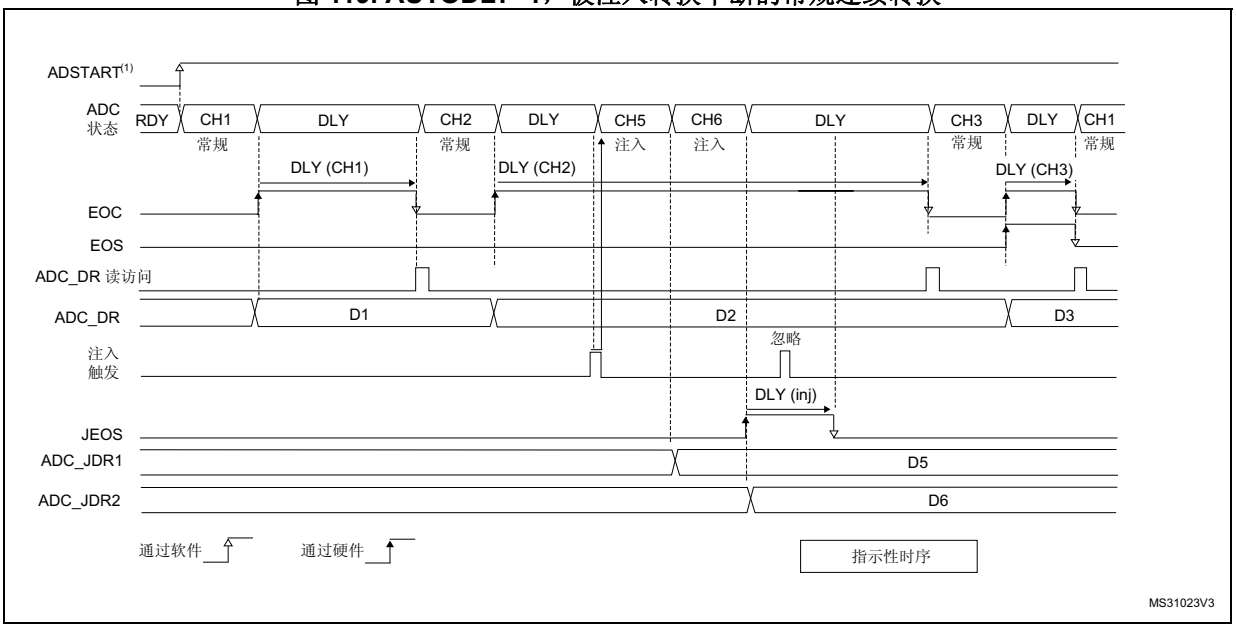
1. AUTDLY=1
2. 常规配置：EXTEN=0x1（硬件触发），CONT=0，DISCEN=0，CHANNELS = 1、2、3
3. 注入配置：JEXTEN=0x1（硬件触发），JDISCEN=0，CHANNELS = 5、6

图 118. AUTODLY=1，被注入转换中断的常规硬件转换（DISCEN=1，JDISCEN=1）



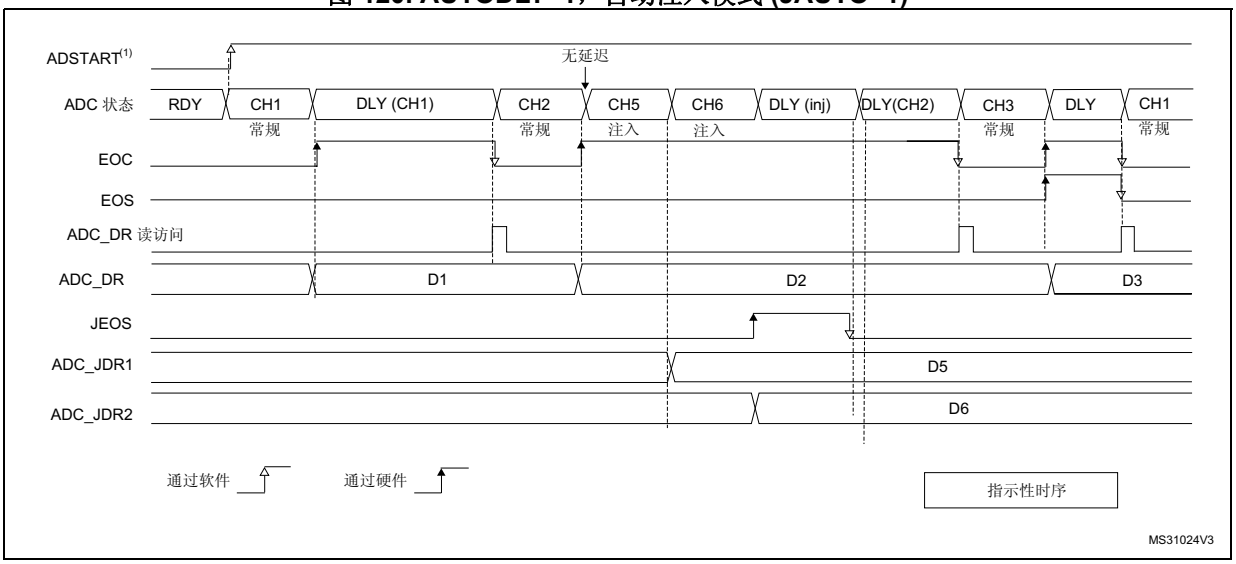
1. AUTDLY=1
2. 常规配置：EXTEN=0x1（硬件触发），CONT=0，DISCEN=1，DISCNUM=1、CHANNELS = 1、2、3
3. 注入配置：JEXTEN=0x1（硬件触发），JDISCEN=1，CHANNELS = 5、6

图 119. AUTODLY=1，被注入转换中断的常规连续转换



1. AUTDLY=1
2. 常规配置：EXTEN=0x0（软件触发），CONT=1，DISCEN=0，CHANNELS = 1、2、3
3. 注入配置：JEXTEN=0x1（硬件触发），JDISCEN=0，CHANNELS = 5、6

图 120. AUTODLY=1，自动注入模式 (JAUTO=1)

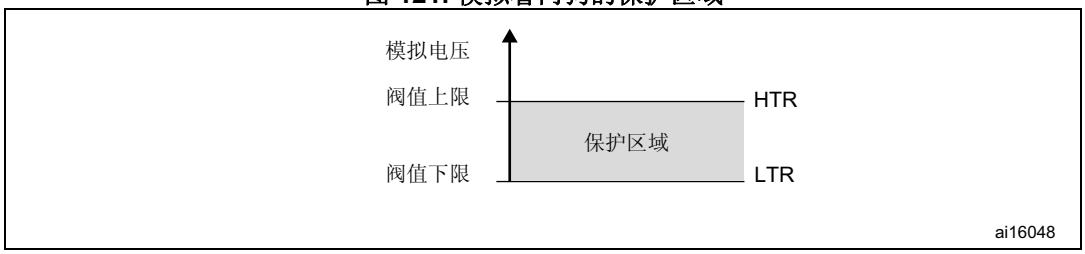


1. AUTDLY=1
2. 常规配置：EXTEN=0x0（软件触发），CONT=1，DISCEN=0，CHANNELS = 1、2
3. 注入配置：JAUTO=1，CHANNELS = 5、6

21.4.29 模拟窗口看门狗（AWD1EN、JAWD1EN、AWD1SGL、AWD1CH、AWD2CH、AWD3CH、AWD_HTx、AWD_LTx、AWDx）

三个 AWD 模拟看门狗会监测一些通道是否保持在配置的电压范围（窗口）内。

图 121. 模拟看门狗的保护区域



AWDx 标志和中断

可通过将 ADCx_IER 寄存器（x=1、2、3）中的 AWDxIE 置 1 的方式分别为 3 个模拟看门狗使能中断。

AWDx（x=1、2、3）标志可通过由软件向其写入 1 的方式来清零。

在对齐之前，会将 ADC 转换结果与阈值上限和下限进行比较。

模拟看门狗 1 说明

将 ADCx_CFGR 寄存器中的 AWD1EN 位置 1，可使能 AWD 模拟看门狗 1。该看门狗监测一条已选通道或所有已使能通道(1) 是否仍在配置的电压范围（窗口）内。

表 120 介绍了应如何配置 ADCx_CFGR 寄存器才能在一条或多个通道上使能模拟看门狗。

表 120. 模拟看门狗通道选择

模拟看门狗保护的通道	AWD1SGL 位	AWD1EN 位	JAWD1EN 位
无	x	0	0
所有注入通道	0	0	1
所有常规通道	0	1	0
所有常规通道和注入通道	0	1	1
单个(1) 注入通道	1	0	1
单个(1) 常规通道	1	1	0
单个(1) 常规通道或注入通道	1	1	1

1. 通过 AWD1CH[4:0] 位选择。此外，还必须将通道编程为在合适的常规序列或注入序列中进行转换。

如果 ADC 转换的模拟电压低于阈值下限或高于阈值上限，则 AWD1 模拟看门狗状态位会置 1。

这些阈值编程到模拟看门狗 1 的 ADCx_TR1 寄存器的 HT1[11:0] 和 LT1[11:0] 位。如果转换的数据分辨率小于 12 位（取决于 RES[1:0] 位），由于始终会在内部对完整的 12 位原始转换数据进行比较（左对齐），因此编程阈值的 LSB 必须保持清零状态。

表 121 介绍了如何对模拟看门狗 1 的所有可能的分辨率进行比较。

表 121. 模拟看门狗 1 比较

分辨率（位 RES[1:0]）	模拟看门狗比较对象：		注释
	原始转换数据，左对齐(1)	阈值	
00：12 位	DATA[11:0]	LT1[11:0] 和 HT1[11:0]	-
01：10 位	DATA[11:2]，00	LT1[11:0] 和 HT1[11:0]	用户必须将 LT1[1:0] 和 HT1[1:0] 配置为 00
10：8 位	DATA[11:4]，0000	LT1[11:0] 和 HT1[11:0]	用户必须将 LT1[3:0] 和 HT1[3:0] 配置为 0000
11：6 位	DATA[11:6]，000000	LT1[11:0] 和 HT1[11:0]	用户必须将 LT1[5:0] 和 HT1[5:0] 配置为 000000

1. 进行任何对齐计算以及应用任何偏移之前，会对原始转换数据进行看门狗比较（进行比较的数据是无符号数据）。

模拟看门狗 2 和 3 说明

第二个和第三个模拟看门狗更加灵活，可通过编程 AWDCHx[19:0]（x=2、3）中的相应位来保护多条已选通道。

AWDCHx[19:0]（x=2、3）的任何位置 1 时，会使能相应的看门狗。

其分辨率被限制为 8 位，仅高 8 位阈值可编程到 HTx[7:0] 和 LTx[7:0] 中。*表 122* 介绍了如何对所有可能的分辨率进行比较。

表 122. 模拟看门狗 2 和 3 比较

分辨率（RES[1:0] 位）	模拟看门狗比较对象：		注释
	原始转换数据，左对齐(1)	阈值	
00：12 位	DATA[11:4]	LTx[7:0] 和 HTx[7:0]	DATA[3:0] 与比较无关
01：10 位	DATA[11:4]	LTx[7:0] 和 HTx[7:0]	DATA[3:2] 与比较无关
10：8 位	DATA[11:4]	LTx[7:0] 和 HTx[7:0]	-
11：6 位	DATA[11:6]，00	LTx[7:0] 和 HTx[7:0]	用户必须将 LTx[1:0] 和 HTx[1:0] 配置为 00

1. 进行任何对齐计算以及应用任何偏移之前，会对原始转换数据进行看门狗比较（进行比较的数据是无符号数据）。

ADCy_AWDx_OUT 信号输出生成

每个模拟看门狗都关联到一个内部硬件信号 ADCy_AWDx_OUT（y = ADC 编号，x = 看门狗编号），该信号直接连接到一些片上定时器的 ETR 输入（外部触发）。请参见片上定时器部分以了解如何选择 ADCy_AWDx_OUT 信号作为 ETR。

当关联的模拟看门狗使能时，ADCy_AWDx_OUT 会激活：

- 当受保护的转换超出编程阈值时，ADCy_AWDx_OUT 会置 1。
- 在编程阈值范围内的下一受保护转换结束后，ADCy_AWDx_OUT 会复位（如果下一受保护转换仍超出编程阈值范围，该位仍保持置 1）。
- 禁止 ADC 时（将 ADDIS 置 1 时），ADCy_AWDx_OUT 也保持复位状态。请注意，停止常规转换或注入转换（将 ADSTP 或 JADSTP 置 1）对 ADCy_AWDx_OUT 的生成没有任何影响。

注： *AWDx 标志由硬件置 1，并由软件复位：AWDx 标志对 ADCy_AWDx_OUT 的生成没有影响（例如：如果软件未将 AWDx 标志清零，即 AWDx 标志会保持为 1，此时 ADCy_AWDx_OUT 仍可翻转）。*

图 122. ADCy_AWDx_OUT 信号生成（所有常规通道上）

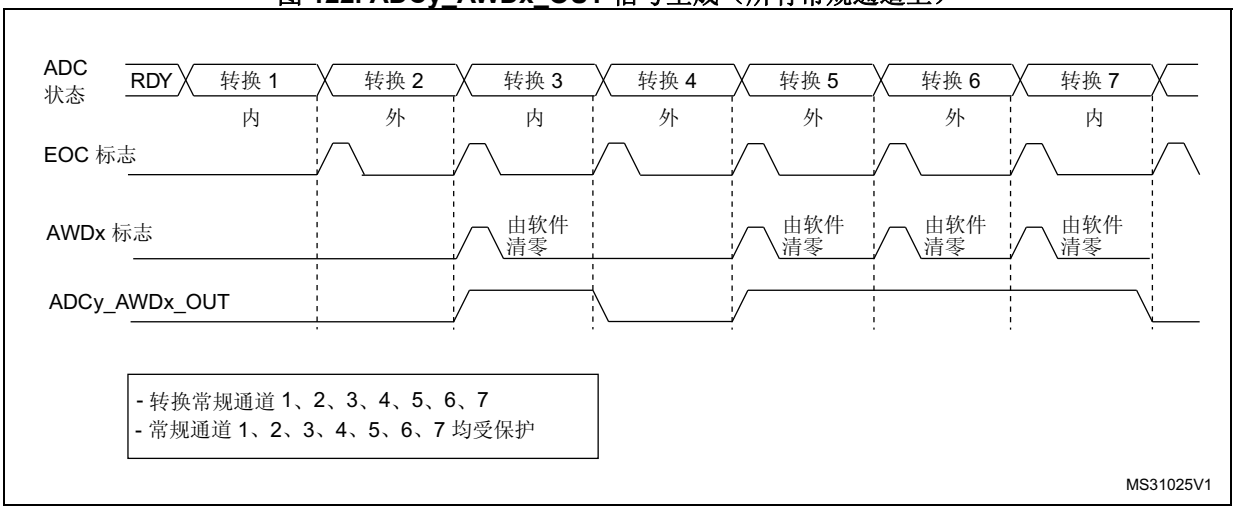


图 123. ADCy_AWDx_OUT 信号生成（AWDx 标志未通过软件清零）

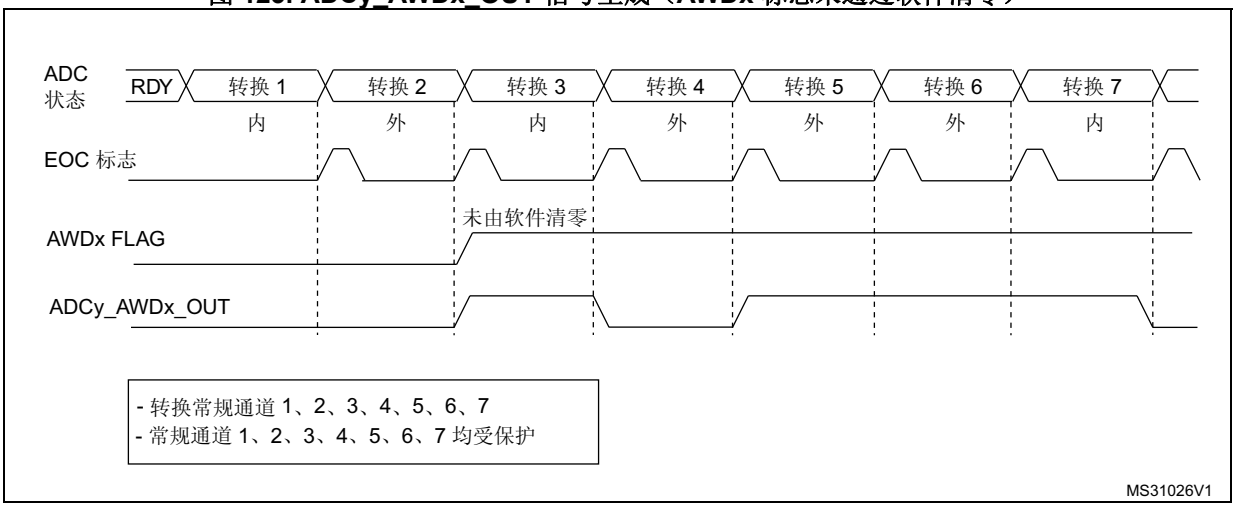


图 124. ADCy_AWDx_OUT 信号生成（单条常规通道上）

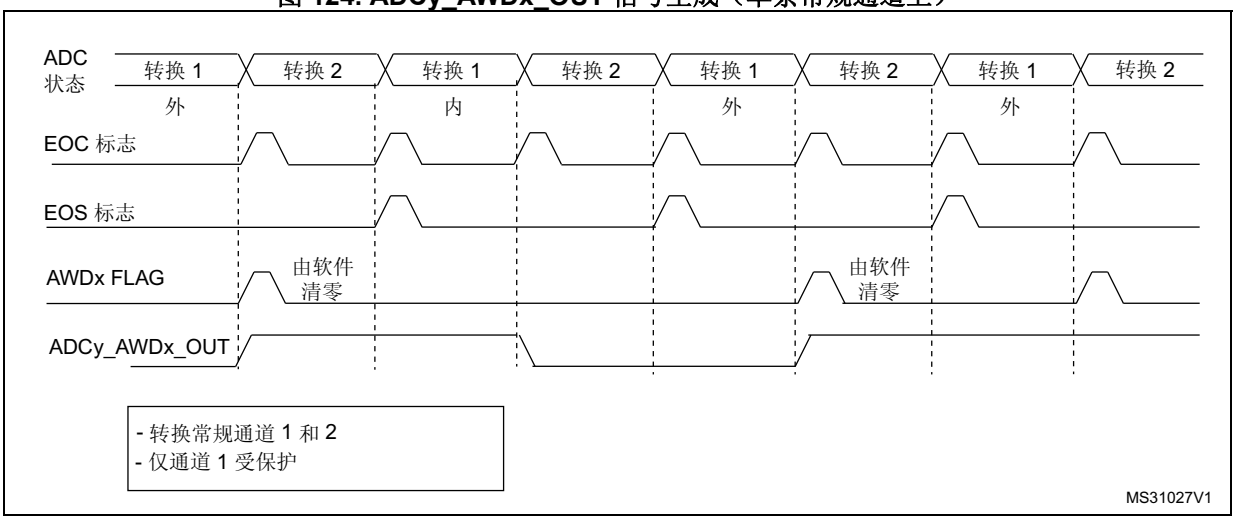
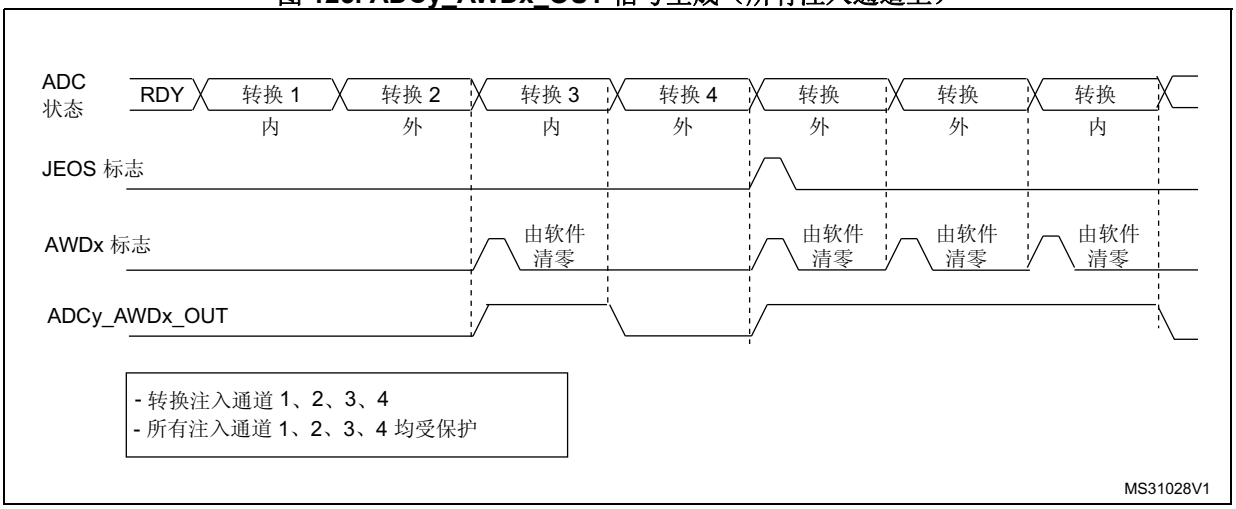


图 125. ADCy_AWDx_OUT 信号生成（所有注入通道上）



21.4.30 过采样器

过采样单元会进行数据预处理，以减轻 CPU 的负担。过采样单元可处理多个转换，并计算多个转换结果的平均值，得到数据宽度增大（高达 16 位）的单个数据。

它提供的结果采用以下形式，其中的 N 和 M 可以进行调整：

$$\text{结果} = \frac{1}{M} \times \sum_{n=0}^{n=N-1} \text{转换}(t_n)$$

允许通过硬件执行以下功能：计算平均值、降低数据速率、改进 SNR 以及基本滤波。

过采样率 N 通过 ADCx_CFGR2 寄存器中的 OVFS[2:0] 位进行定义，范围为 2x 到 256x。分频系数 M 通过向右移位来实现（最多可移 8 位），并且通过 ADCx_CFGR2 寄存器中的 OVSS[3:0] 位定义。

求和单元可得出多达 20 位的结果（256x 12 位结果），结果会先右移。随后会截断为 16 个最低有效位，这些位使用移位后剩下的最低有效位四舍五入为最接近的数值，然后将最终得到的结果传输到 ADCx_DR 数据寄存器中。

注： *如果移位后得到的中间结果超过 16 位，则会将结果按原样截断（无饱和）。*

图 126. 20 位到 16 位结果的截断过程

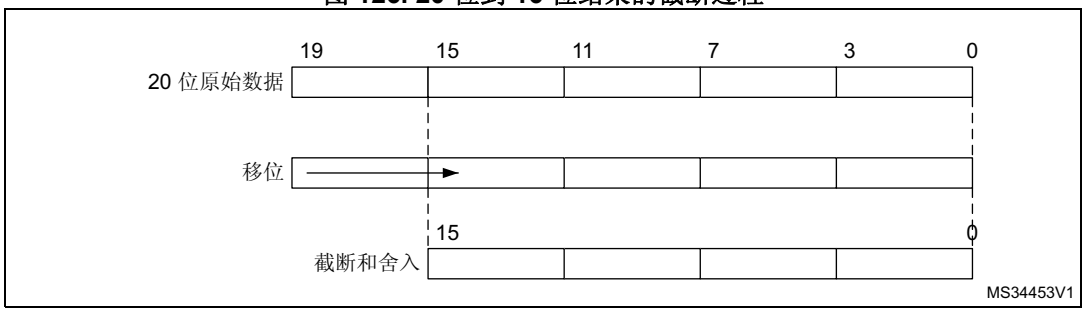


图 127 介绍了从原始的 20 位累加数据到最终 16 位结果的数值处理过程。

图 127. 移 5 位并进行舍入的数值示例

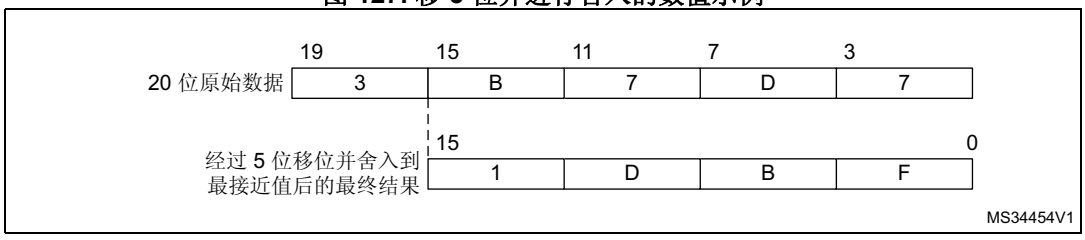


表 123 列出了原始转换数据等于 0xFFF 时对应的各种 N 和 M 组合的数据格式。

表 123. 最大输出结果与 N 和 M 的对应关系（灰色单元格表示截断的部分）

过采样率	最大原始数据	不移位 OVSS = 0000	移 1 位 OVSS = 0001	移 2 位 OVSS = 0010	移 3 位 OVSS = 0011	移 4 位 OVSS = 0100	移 5 位 OVSS = 0101	移 6 位 OVSS = 0110	移 7 位 OVSS = 0111	移 8 位 OVSS = 1000
2x	0x1FFE	0x1FFE	0x0FFF	0x0800	0x0400	0x0200	0x0100	0x0080	0x0040	0x020
4x	0x3FFC	0x3FFC	0x1FFE	0x0FFF	0x0800	0x0400	0x0200	0x0100	0x0080	0x0040
8x	0x7FF8	0x7FF8	0x3FFC	0x1FFE	0x0FFF	0x0800	0x0400	0x0200	0x0100	0x0080
16x	0xFFF0	0xFFF0	0x7FF8	0x3FFC	0x1FFE	0x0FFF	0x0800	0x0400	0x0200	0x0100
32x	0x1FFE0	0xFFE0	0xFFF0	0x7FF8	0x3FFC	0x1FFE	0x0FFF	0x0800	0x0400	0x0200
64x	0x3FFC0	0xFFC0	0xFFE0	0xFFF0	0x7FF8	0x3FFC	0x1FFE	0x0FFF	0x0800	0x0400
128x	0x7FF80	0xFF80	0xFFC0	0xFFE0	0xFFF0	0x7FF8	0x3FFC	0x1FFE	0x0FFF	0x0800
256x	0xFFF00	0xFF00	0xFF80	0xFFC0	0xFFE0	0xFFF0	0x7FF8	0x3FFC	0x1FFE	0x0FFF

过采样模式下的转换时序不会发生变化：在整个过采样序列中，采样时间保持不变。每完成 N 次转换都会提供新数据，等效延迟等于 N x T_{CONV} = N x (t_{SMPL} + t_{SAR})。各标志的置 1 情况如下：

- 每个采样阶段后都会将采样阶段结束标志 (EOSMP) 置 1
- 如果过采样结果可用，每完成 N 次转换都会发生转换结束事件 (EOC)
- 过采样数据序列完成后（即 N x 序列长度次转换之后），会发生序列结束事件 (EOS)

过采样时支持的 ADC 工作模式

在过采样模式下，大部分 ADC 工作模式都会保留：

- 单次转换或连续模式转换
- 可由软件或触发器启动 ADC 转换
- ADC 在转换过程中停止（中止）
- 通过 CPU 或 DMA 在支持溢出检测的情况下读取数据
- 低功耗模式 (AUTDLY)
- 可编程分辨率：在这种情况下，会按照与 12 位转换相同的方式对分辨率降低的转换值（根据 ADCx_CFGR1 寄存器中的 RES[1:0] 位）进行累加、截断、四舍五入和移位

注： *处理过采样数据时，不可使用对齐模式。ADCx_CFGR1 中的 ALIGN 位会被忽略，且数据始终采用右对齐格式。*

过采样模式下不支持偏移校准。如果 ROVSE 和/或 JOVSE 位置 1，ADCx_OFRy 寄存器中 OFFSETy_EN 位的值会被忽略（视为复位）。

模拟看门狗

模拟看门狗功能会保留（AWDSGL 位和 AWDEN 位），但存在以下区别：

- 会忽略 RES[1:0] 位，始终会使用完整的 12 位值 HT[11:0] 和 LT[11:0] 进行比较
- 会比较 16 位过采样结果 ADCx_DR[15:4] 的 12 个最高有效位

注： *移位位数较大时必须多加留意，因为这样会缩小比较范围。例如，如果过采样结果移了 4 位，得到 12 位右对齐数据，那么只能对 8 个数据位执行有效的模拟看门狗比较。比较操作会在 ADCx_DR[11:4] 与 HT[0:7]/LT[0:7] 之间进行，并且 HT[11:8]/LT[11:8] 必须保持复位状态。*

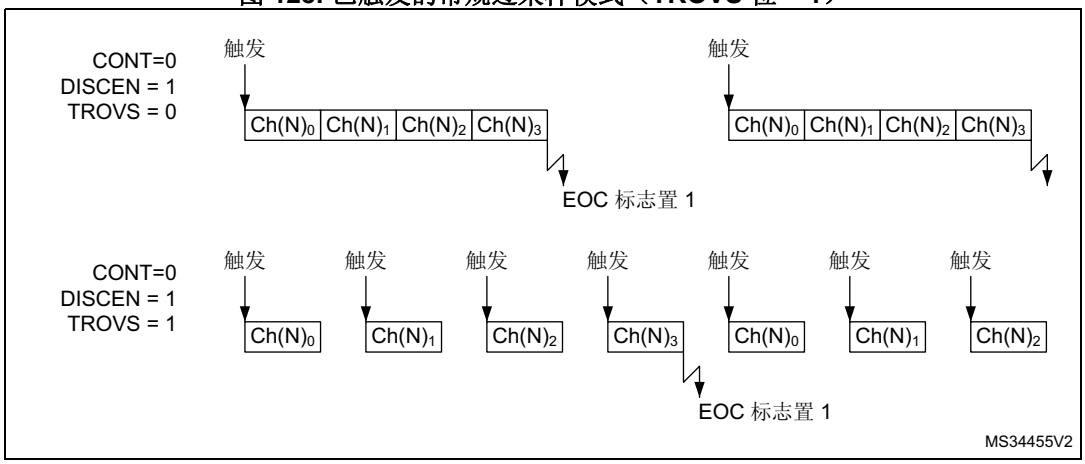
触发模式

平均值计算单元还可用于基本滤波，虽然它不是非常强大的滤波器（衰减缓慢、停止频段衰减受限），但可用作陷波滤波器，用于抑制恒定的寄生频率（通常来自输电线或开关电源）。为此，可使用 ADCx_CFGR2 中的 TROVS 位使能特定的不连续模式，以获得由用户定义、且与转换时间本身无关的过采样频率。

图 128 显示了在不连续模式下如何响应触发从而开始转换。

如果 TROVS 位置 1，则会忽略 DISCEN 位的内容，并将该位视为 1。

图 128. 已触发的常规过采样模式（TROVS 位 = 1）



过采样时的注入和常规定序器管理

在过采样模式下，注入定序器和常规定序器可以执行不同操作。如果两个定序器必须同时使用，则可为它们使能过采样并设定一些限制条件（与唯一的累加单元相关）。

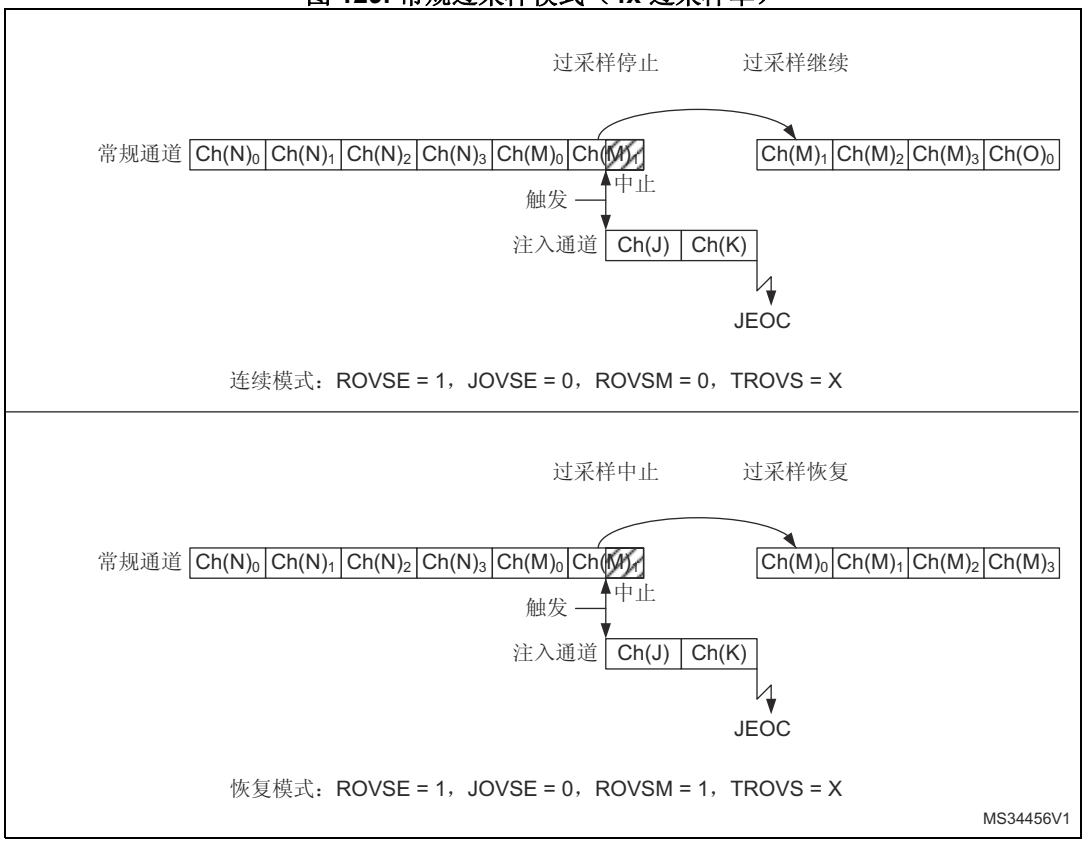
仅对常规通道进行过采样

常规过采样模式位 ROVSM 定义了常规过采样序列在被注入转换中断的情况下如何恢复：

- 在连续模式下，会从上一有效数据开始重新累加（在由于注入触发而发出的转换中止请求之前）。这样可确保在任何注入频率下均可而完成过采样（假设触发之间至少可完成一次常规转换）；
- 在恢复模式下，会从 0 开始重新累加（会忽略之前的转换结果）。该模式可确保所有用于过采样的数据在单个时隙内进行了背靠背转换。需要注意的是，注入触发周期必须超过过采样时长。如果该条件未得到满足，将无法完成过采样，常规定序器将被禁用。

图 129 以 4x 过采样率为例进行了说明。

图 129. 常规过采样模式（4x 过采样率）



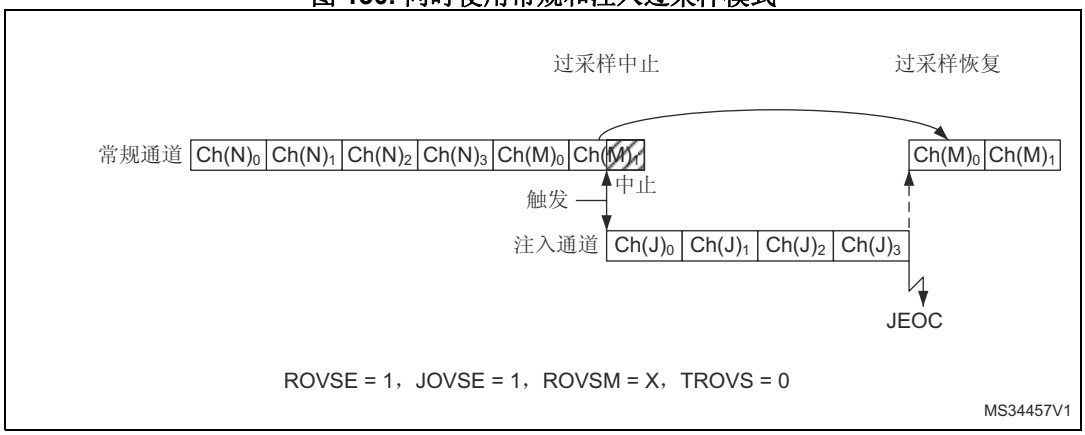
仅对注入通道进行过采样

注入过采样模式位 JOVSE 专为注入定序器中的转换使能过采样。

对常规通道和注入通道进行过采样

可以将 ROVSE 和 JOVSE 位都置 1。在这种情况下，常规过采样模式会强制进入恢复模式（忽略 ROVSM 位），如*图 130* 所示。

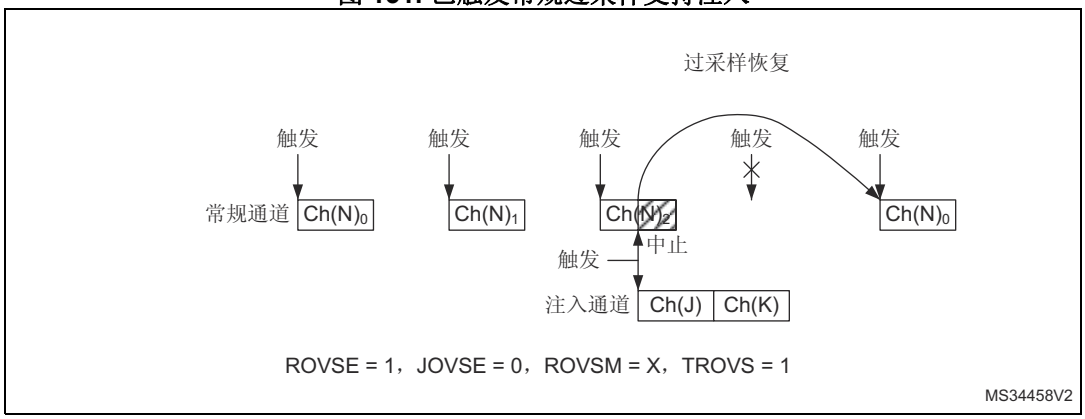
图 130. 同时使用常规和注入过采样模式



已触发常规过采样支持注入转换

已触发常规模式下支持注入转换。在这种情况下，必须禁止注入过采样模式，并且会忽略 ROVSM 位（强制进入恢复模式）。JOVSE 位必须复位。具体操作如*图 131* 所示。

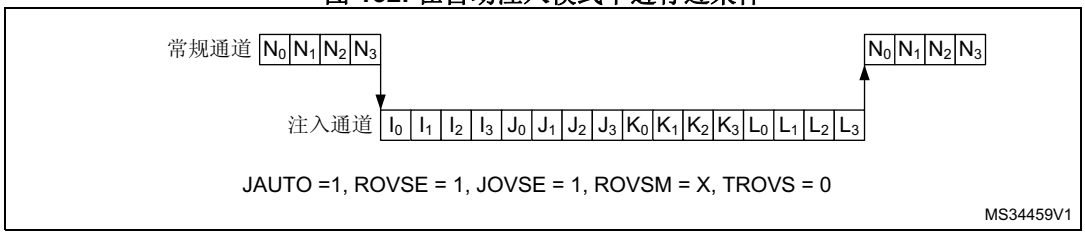
图 131. 已触发常规过采样支持注入



自动注入模式

可以对自动注入序列进行过采样，并将所有转换结果存储在寄存器中以节省 DMA 资源。使用该模式的前提条件是常规和注入过采样均激活：JAUTO = 1、ROVSE = 1 且 JOVSE = 1，不支持其他组合。在自动注入模式下，会忽略 ROVSM 位。*图 132* 显示了转换的排序方式。

图 132. 在自动注入模式下进行过采样



此外，还可以使用 TROVS 位使能触发模式。在这种情况下，ADC 必须进行如下配置：JAUTO = 1、DISCEN = 0、JDISCEN = 0、ROVSE = 1、JOVSE = 1 和 TROVSE = 1。

21.4.31 温度传感器

温度传感器可用于测量器件的结温 (Tj)。温度传感器内部连接到 ADC 输入通道，该通道用于将传感器输出电压转换为数字值。不使用时可将传感器置于掉电模式。支持的温度范围：-40 °C 到 125 °C。

图 133 显示的是温度传感器与 ADC 之间连接的框图。

温度传感器的输出电压随温度线性变化。由于工艺不同，该线的偏移量取决于各个芯片（芯片之间的温度变化可达 45 °C）。

未校准的内部温度传感器更适用于对温度变量而非绝对温度进行测量的应用。为提高温度传感器测量的准确性，ST 在生产过程中将校准值存储在每个器件的系统存储器中。

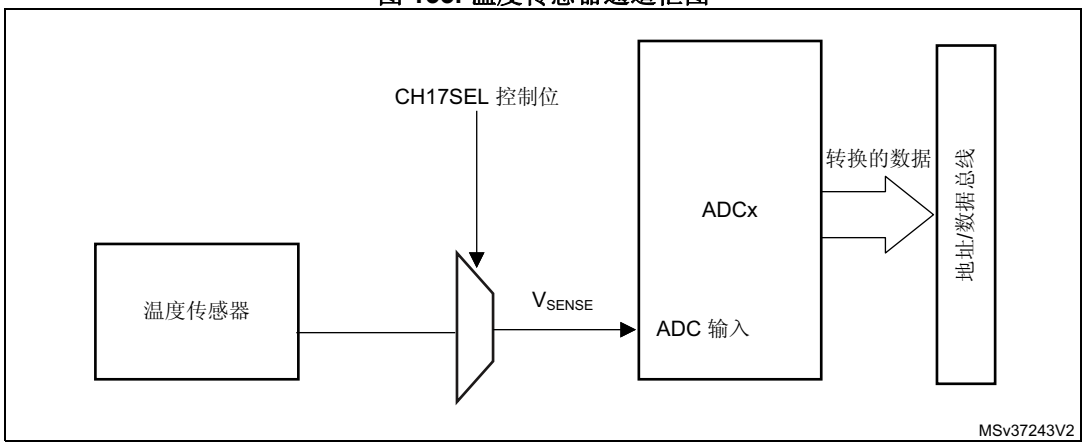
在制造过程中，会将温度传感器的校准数据和内部参考电压的校准数据存储在系统存储区。随后，用户应用可读取这些数据，并使用这些数据提高温度传感器或内部参考的准确性（更多相关信息，请参见数据手册）。

温度传感器在内部连接到 ADC 输入通道，该通道用于将传感器输出电压转换为数字值。有关转换内部温度传感器时要应用的采样时间值，请参见器件数据手册的电气特性部分。

不使用时可将传感器置于掉电模式。

图 133 显示了温度传感器框图。

图 133. 温度传感器通道框图



读取温度

要使用传感器，请执行以下操作：

1. 选择连接到 V_{SENSE} 的 ADC 输入通道。
2. 设定合适的采样时间（请参见器件数据手册中的电气特性部分）。
3. 在 ADCx_CCR 寄存器中将 CH17SEL 位置 1，以便将温度传感器从掉电模式中唤醒。
4. 开始 ADC 转换。
5. 读取 ADC 数据寄存器中生成的 V_{SENSE} 数据。
6. 使用以下公式计算实际温度：

$$\text{温度（以 °C 为单位）} = \frac{110\ °C - 30\ °C}{TS_CAL2 - TS_CAL1} \times (TS_DATA - TS_CAL1) + 30\ °C$$

其中：

- TS_CAL2 是在 110 °C 下获得的温度传感器校准值
- TS_CAL1 是在 30 °C 下获得的温度传感器校准值
- TS_DATA 是由 ADC 转换得到的实际温度传感器输出值

  更多关于 TS_CAL1 和 TS_CAL2 校准点的信息，请参见器件数据手册。

注： *传感器从掉电模式中唤醒需要一个启动时间，启动时间过后其才能正确输出 V_{SENSE}。ADC 在上电后同样需要一个启动时间，因此，为尽可能减少延迟，应同时将 ADEN 和 CH17SEL 位置 1。*

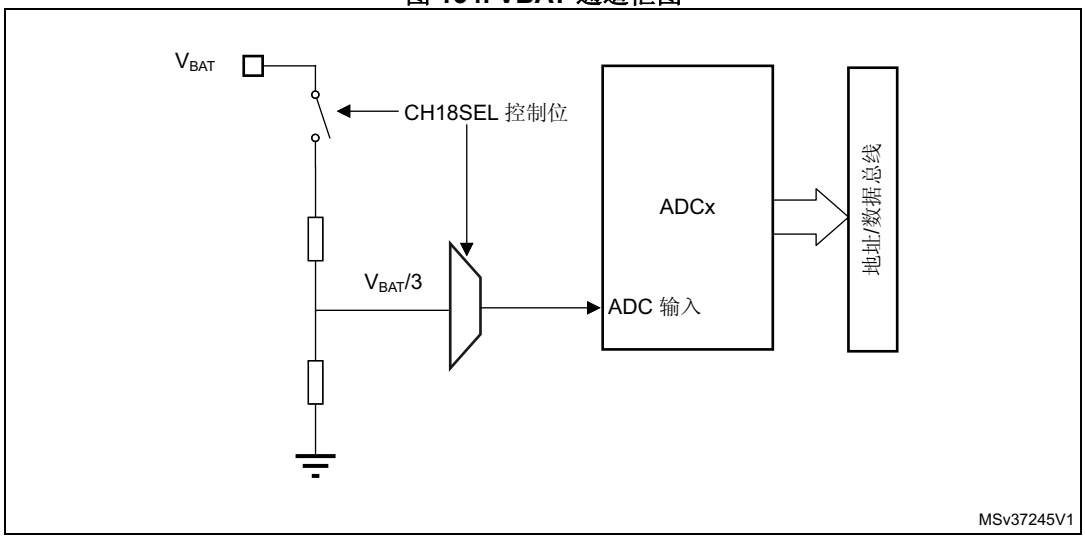
21.4.32 VBAT 电源监测

ADCx_CCR 寄存器中的 CH18SEL 位用于切换到电池电压。由于 V_{BAT} 电压可能高于 V_{DDA}，为确保 ADC 正常工作，V_{BAT} 引脚需要在内部连接到桥接分配器（除以 3）。CH18SEL 置 1 时，会自动使能此桥，以将 $V_{BAT}/3$ 连接到 ADC 输入通道。因此，转换出的数字值为 V_{BAT} 电压的三分之一。为防止电池出现意外的电能消耗，建议仅在必要时为 ADC 转换使能桥接分配器。

有关转换 $V_{BAT}/3$ 电压时要应用的采样时间值，请参见器件数据手册的电气特性部分。

下图显示了 V_{BAT} 传感特性的框图。

图 134. VBAT 通道框图



1. 必须将 CH18SEL 位置 1 才能为 $V_{BAT}/3$ 使能内部通道转换。

21.4.33 监测内部参考电压

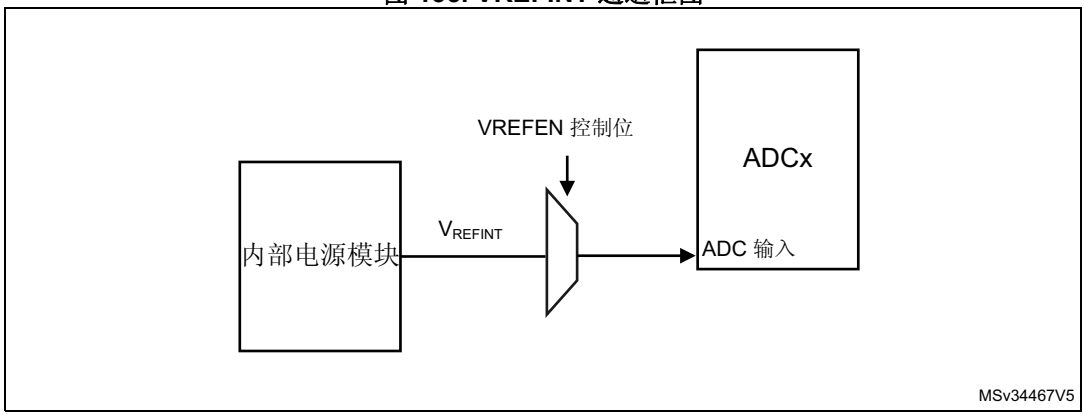
可以通过监测内部参考电压 (V_{REFINT}) 来获得用于评估 ADC V_{REF+} 电压的参考值。

内部参考电压在内部连接到 ADC1 的输入通道 0 (ADC1_INP0)。

有关转换内部参考电压时要应用的采样时间值，请参见产品数据手册的电气特性部分。

图 135 显示了 V_{REFINT} 传感特性的框图。

图 135. VREFINT 通道框图



1. 必须将 ADCx_CCR 寄存器中的 VREFEN 位置 1 才能使能内部通道 (V_{REFINT}) 的转换。

使用内部参考电压计算实际的 V_{DDA} 电压

施加给微控制器的 V_{DDA} 电源电压可能会有变化，或无法获得准确值。在制造过程中由 ADC 在 V_{DDA} = 3.0 V 的条件下获得的内置内部参考电压 (V_{REFINT}) 及其校准数据可用于评估实际的 V_{DDA} 电压水平。

以下公式可求得为器件供电的实际 V_{DDA} 电压：

$$V_{DDA} = 3.0\ V \times VREFINT_CAL/VREFINT_DATA$$

其中：

- VREFINT_CAL 是 VREFINT 校准值
- VREFINT_DATA 是由 ADC 转换得到的实际 VREFINT 输出值

将电源相关的 ADC 测量值转换为绝对电压值

ADC 用于提供对应于模拟电源与施加给转换通道的电压之比的数字值。对于大部分应用用例，需要将该比值转换成与 V_{DDA} 无关的电压。对于 V_{DDA} 已知、ADC 转换值进行了右对齐的应用，可使用以下公式得到该绝对值：

$$V_{CHANNELx} = \frac{V_{DDA}}{FULL_SCALE} \times ADCx_DATA$$

对于 V_{DDA} 值未知的应用，必须使用内部参考电压，V_{DDA} 可替换为*使用内部参考电压计算实际的 VDDA 电压*部分提供的表达式，从而得出以下公式：

$$V_{CHANNELx} = \frac{3.0\ V \times VREFINT_CAL \times ADCx_DATA}{VREFINT_DATA \times FULL_SCALE}$$

其中：

- VREFINT_CAL 是 VREFINT 校准值。
- ADCx_DATA 是由 ADC 在通道 x 上测得的值（右对齐）。
- VREFINT_DATA 是由 ADC 转换得到的实际 VREFINT 输出值。
- FULL_SCALE 是 ADC 输出的最大数字值。例如，如果分辨率为 12 位，该值为 2^{12} - 1 = 4095，如果分辨率为 8 位，该值为 2^8 - 1 = 255。

注： *如果执行 ADC 测量时使用的是输出格式而非 12 位右对齐格式，那么必须先将所有参数转换为兼容格式，然后再进行计算。*

21.5 ADC 中断

在以下情况下会生成中断：

- ADC 上电后，在 ADC 就绪时（ADRDY 标志）
- 常规组的任何转换结束时（EOC 标志）
- 常规组的转换序列结束时（EOS 标志）
- 注入组的任何转换结束时（JEOC 标志）
- 注入组的转换序列结束时（JEOS 标志）
- 发生模拟看门狗检测时（AWD1、AWD2 和 AWD3 标志）
- 采样阶段结束时（EOSMP 标志）
- 发生数据溢出时（OVR 标志）
- 注入序列上下文队列溢出时（JQOVF 标志）

可以使用单独的中断使能位以提高灵活性。

表 124. ADC 中断

中断事件	事件标志	使能控制位
ADC 就绪	ADRDY	ADRDYIE
常规组的转换结束	EOC	EOCIE
常规组的转换序列结束	EOS	EOSIE
注入组的转换结束	JEOC	JEOCIE
注入组的转换序列结束	JEOS	JEOSIE
模拟看门狗 1 状态位置 1	AWD1	AWD1IE
模拟看门狗 2 状态位置 1	AWD2	AWD2IE
模拟看门狗 3 状态位置 1	AWD3	AWD3IE
采样阶段结束	EOSMP	EOSMPIE
上溢	OVR	OVRIE
注入上下文队列溢出	JQOVF	JQOVFIE

21.6 ADC 寄存器

有关寄存器说明中使用的缩写，请参见*第 72 页的第 1.1 节*。

21.6.1 ADC 中断和状态寄存器 (ADC_ISR)

ADC interrupt and 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
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.	Res.	Res.	Res.	JQOVF	AWD3	AWD2	AWD1	JEOS	JEOC	OVR	EOS	EOC	EOSMP	ADRDY
					rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1

位 31:11 保留，必须保持复位值。

位 10 **JQOVF**：注入上下文队列溢出 (Injected context queue overflow)
当注入上下文队列溢出时，该位会由硬件置 1。通过软件写入 1 可将该位清零。更多信息，请参见*第 21.4.21 节：注入转换的上下文队列*。
0：未发生注入上下文队列溢出（或标志事件已通过软件确认并清零）
1：发生注入上下文队列溢出

位 9 **AWD3**：模拟看门狗 3 标志 (Analog watchdog 3 flag)
当转换电压超过在 ADCx_TR3 寄存器的 LT3[7:0] 和 HT3[7:0] 字段中编程的值时，硬件会将该位置 1。通过软件写入 1 可将该位清零。
0：未发生模拟看门狗 3 事件（或标志事件已通过软件确认并清零）
1：发生模拟看门狗 3 事件

位 8 **AWD2**：模拟看门狗 2 标志 (Analog watchdog 2 flag)
当转换电压超过在 ADCx_TR2 寄存器的 LT2[7:0] 和 HT2[7:0] 字段中编程的值时，硬件会将该位置 1。通过软件写入 1 可将该位清零。
0：未发生模拟看门狗 2 事件（或标志事件已通过软件确认并清零）
1：发生模拟看门狗 2 事件

位 7 **AWD1**：模拟看门狗 1 标志 (Analog watchdog 1 flag)
当转换电压超过在 ADCx_TR1 寄存器的 LT1[11:0] 和 HT1[11:0] 字段中编程的值时，硬件会将该位置 1。通过软件 写入 1 可将该位清零。
0：未发生模拟看门狗 1 事件（或标志事件已通过软件确认并清零）
1：发生模拟看门狗 1 事件

位 6 **JEOS**：注入通道序列结束标志 (Injected channel end of sequence flag)
组内所有注入通道转换结束时，硬件将该位置 1。通过软件写入 1 可将该位清零。
0：注入转换序列未完成（或标志事件已通过软件确认并清零）
1：注入转换已完成

位 5 **JEOC**：注入通道转换结束标志 (Injected channel end of conversion flag)

当通道的每次注入转换结束，新数据出现在相应的 ADCx_JDRy 寄存器时，会通过硬件将该位置 1。通过软件向该位写入 1，或读取相应的 ADCx_JDRy 寄存器都可将该位清零。

0：注入通道转换未完成（或标志事件已通过软件确认并清零）

1：注入通道转换已完成

位 4 **OVR**：ADC 溢出 (ADC overrun)

该位在常规通道上发生溢出事件时由硬件置 1，这意味着在 EOC 标志已置 1 时，新转换已完成。通过软件写入 1 可将该位清零。

0：未发生溢出事件（或标志事件已通过软件确认并清零）

1：发生溢出

位 3 **EOS**：常规序列结束标志 (End of regular sequence flag)

常规通道序列转换结束后，硬件将该位置 1。通过软件写入 1 可将该位清零。

0：常规转换序列未完成（或标志事件已通过软件确认并清零）

1：常规转换序列已完成

位 2 **EOC**：转换结束标志 (End of conversion flag)

当通道的每次常规转换结束，新数据出现在 ADCx_DR 寄存器时，会通过硬件将该位置 1。通过软件向该位写入 1，或读取 ADCx_DR 寄存器都可将该位清零。

0：常规通道转换未完成（或标志事件已通过软件确认并清零）

1：常规通道转换已完成

位 1 **EOSMP**：采样结束标志 (End of sampling flag)

在任何通道（仅限常规通道）的转换过程中，当采样阶段结束时，会通过硬件将该位置 1。

0：采样阶段未结束（或标志事件已通过软件确认并清零）

1：采样阶段已结束

位 0 **ADRDY**：ADC 就绪 (ADC ready)

ADC 使能后（位 ADEN=1）以及 ADC 达到准备好接收转换请求的状态时，会通过硬件将该位置 1。通过软件写入 1 可将该位清零。

0：ADC 未准备好开始转换（或标志事件已通过软件确认并清零）

1：ADC 已准备好开始转换

21.6.2 ADC 中断使能寄存器 (ADC_IER)

ADC interrupt enable 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
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.	Res.	Res.	Res.	JQOVFIE	AWD3IE	AWD2IE	AWD1IE	JEOSIE	JEOCIE	OVRIE	EOSIE	EOCIE	EOSMPIE	ADRDYIE
					rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:11 保留，必须保持复位值。

位 10 **JQOVFIE**：注入上下文队列溢出中断使能 (Injected context queue overflow interrupt enable)

此位由软件置 1 和清零，用于使能/禁止注入上下文队列溢出中断。

0：禁止注入上下文队列溢出中断。

1：使能注入上下文队列溢出中断。JQOVF 位置 1 时产生中断。

注： 仅当 JADSTART=0 时（这可确保当前未进行任何注入转换），才允许通过软件对此位执行写操作。

位 9 **AWD3IE**：模拟看门狗 3 中断使能 (Analog watchdog 3 interrupt enable)

此位由软件置 1 和清零，用于使能/禁止模拟看门狗 3 中断。

0：禁止模拟看门狗 3 中断

1：使能模拟看门狗 3 中断

注： 仅当 ADSTART=0 且 JADSTART=0 时（这可确保当前未进行任何转换），才允许通过软件对此位执行写操作。

位 8 **AWD2IE**：模拟看门狗 2 中断使能 (Analog watchdog 2 interrupt enable)

此位由软件置 1 和清零，用于使能/禁止模拟看门狗 2 中断。

0：禁止模拟看门狗 2 中断

1：使能模拟看门狗 2 中断

注： 仅当 ADSTART=0 且 JADSTART=0 时（这可确保当前未进行任何转换），才允许通过软件对此位执行写操作。

位 7 **AWD1IE**：模拟看门狗 1 中断使能 (Analog watchdog 1 interrupt enable)

此位由软件置 1 和清零，用于使能/禁止模拟看门狗 1 中断。

0：禁止模拟看门狗 1 中断

1：使能模拟看门狗 1 中断

注： 仅当 ADSTART=0 且 JADSTART=0 时（这可确保当前未进行任何转换），才允许通过软件对此位执行写操作。

位 6 **JEOSIE**：注入转换序列结束中断使能 (End of injected sequence of conversions interrupt enable)

此位由软件置 1 和清零，用于使能/禁止注入转换序列结束中断。

0：禁止 JEOS 中断。

1：使能 JEOS 中断。JEOS 位置 1 时产生中断。

注： 仅当 JADSTART=0 时（这可确保当前未进行任何注入转换），才允许通过软件对此位执行写操作。

位 5 **JEOCIE**：注入转换结束中断使能 (End of injected conversion interrupt enable)

此位由软件置 1 和清零，用于使能/禁止注入转换结束中断。

0：禁止 JEOC 中断。

1：使能 JEOC 中断。JEOC 位置 1 时产生中断。

注： 仅当 JADSTART=0 时（这可确保当前未进行任何常规转换），才允许通过软件对此位执行写操作。

位 4 **OVRIE**：溢出中断使能 (Overrun interrupt enable)

此位由软件置 1 和清零，用于使能/禁止常规转换的溢出中断。

0：禁止溢出中断。

1：使能溢出中断。OVR 位置 1 时产生中断。

注： 仅当 ADSTART=0 时（这可确保当前未进行任何常规转换），才允许通过软件对此位执行写操作。

位 3 **EOSIE**：常规转换序列结束中断使能 (End of regular sequence of conversions interrupt enable)

此位由软件置 1 和清零，用于使能/禁止常规转换序列结束中断。

0：禁止 EOS 中断。

1：使能 EOS 中断。EOS 位置 1 时产生中断。

注： 仅当 ADSTART=0 时（这可确保当前未进行任何常规转换），才允许通过软件对此位执行写操作。

位 2 **EOCIE**：常规转换结束中断使能 (End of regular conversion interrupt enable)

此位由软件置 1 和清零，用于使能/禁止常规转换结束中断。

0：禁止 EOC 中断。

1：使能 EOC 中断。EOC 位置 1 时产生中断。

注： 仅当 ADSTART=0 时（这可确保当前未进行任何常规转换），才允许通过软件对此位执行写操作。

位 1 **EOSMPIE**：常规转换采样结束中断使能 (End of sampling flag interrupt enable for regular conversions)

此位由软件置 1 和清零，用于使能/禁止常规转换采样阶段结束中断。

0：禁止 EOSMP 中断。

1：使能 EOSMP 中断。EOSMP 位置 1 时产生中断。

注： 仅当 ADSTART=0 时（这可确保当前未进行任何常规转换），才允许通过软件对此位执行写操作。

位 0 **ADRDYIE**：ADC 就绪中断使能 (ADC ready interrupt enable)

此位由软件置 1 和清零，用于使能/禁止 ADC 就绪中断。

0：禁止 ADRDY 中断。

1：使能 ADRDY 中断。ADRDY 位置 1 时产生中断。

注： 仅当 ADSTART=0 且 JADSTART=0 时（这可确保当前未进行任何转换），才允许通过软件对此位执行写操作。

21.6.3 ADC 控制寄存器 (ADC_CR)

ADC control register

偏移地址：0x08

复位值：0x2000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ADCAL	ADCALDIF	DEEPPWD	ADVREGEN	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
rs	rw	rw	rw												
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	JADSTP	ADSTP	JADSTART	ADSTART	ADDIS	ADEN
										rs	rs	rs	rs	rs	rs

位 31 **ADCAL**：ADC 校准 (ADC calibration)

此位由软件置 1，用于开始 ADC 校准。先对 ADCALDIF 位编程，以确定此校准应用于单端输入模式还是差分输入模式。

校准完成后，该位由硬件清零。

0：校准已完成。

1：写入 1 可校准 ADC。读取值为 1 表示正在进行校准。

注： *仅当 ADEN=0 时，才允许通过软件将 ADCAL 置 1 的方式启动校准。*

仅当 ADEN=1、ADSTART=0 且 JADSTART=0（ADC 已使能，当前未进行任何转换）时，才允许通过软件对 ADCx_CALFACT 执行写操作来更新校准系数。

位 30 **ADCALDIF**：校准的差分模式 (Differential mode for calibration)

此位由软件置 1 和清零，用于为校准配置单端输入模式或差分输入模式。

0：写入 ADCAL 将在单端输入模式下启动校准。

1：写入 ADCAL 将在差分输入模式下启动校准。

注： *仅当 ADC 已禁止、且当前未进行校准时（ADCAL=0、JADSTART=0、JADSTP=0、ADSTART=0、ADSTP=0、ADDIS=0 且 ADEN=0），才允许通过软件对此位执行写操作。*

位 29 **DEEPPWD**：深度掉电使能 (Deep-power-down enable)

此位由软件置 1 和清零，用于使 ADC 进入深度掉电模式。

0：ADC 未处于深度掉电模式

1：ADC 处于深度掉电模式（默认复位状态）

注： *仅当 ADC 已禁止时（ADCAL=0、JADSTART=0、JADSTP=0、ADSTART=0、ADSTP=0、ADDIS=0 且 ADEN=0），才允许通过软件对此位执行写操作。*

位 28 **ADVREGEN**：ADC 稳压器使能 (ADC voltage regulator enable)

此位由软件置 1，用于使能 ADC 稳压器。

执行启动校准或使能 ADC 等操作之前，必须先使能 ADC 稳压器，并且软件必须等待稳压器启动时间。

0：禁止 ADC 稳压器

1：使能 ADC 稳压器。

有关 ADC 稳压器使能和禁止序列的更多详细信息，请参见*第 21.4.6 节：ADC 深度掉电模式 (DEEPPWD) 和 ADC 稳压器 (ADVREGEN)*。

仅当 ADC 已禁止时（ADCAL=0、JADSTART=0、ADSTART=0、ADSTP=0、ADDIS=0 且 ADEN=0），才能通过软件对该位域进行编程。

位 27:6 保留，必须保持复位值。

位 5 **JADSTP**：ADC 停止注入转换命令 (ADC stop of injected conversion command)

该位由软件置 1，用于停止和丢弃正在进行的注入转换（JADSTP 命令）。

当转换已有效丢弃、并且可重新配置 ADC 注入序列和触发时，会通过硬件将该位清零。随后，ADC 会准备好接收新的开始注入转换命令（JADSTART 命令）。

0：当前未执行 ADC 停止注入转换命令。

1：写入 1 可停止正在进行的注入转换。读取值为 1 表示正在执行 ADSTP 命令。

注： *仅当 JADSTART=1 且 ADDIS=0 时（ADC 已使能、最终会进行注入转换、并且没有任何待处理的禁止 ADC 的请求），才允许通过软件将 JADSTP 置 1*

在自动注入模式下 (JAUTO=1)，将 ADSTP 位置 1 会中止常规转换和注入转换（不要使用 JADSTP）

位 4 **ADSTP**：ADC 停止常规转换命令 (ADC stop of regular conversion command)

该位由软件置 1，用于停止和丢弃正在进行的常规转换（ADSTP 命令）。

当转换已有效丢弃、并且可重新配置 ADC 常规序列和触发时，会通过硬件将该位清零。随后，ADC 会准备好接收新的开始常规转换命令（ADSTART 命令）。

0：当前未执行 ADC 停止常规转换命令。

1：写入 1 可停止正在进行的常规转换。读取值为 1 表示正在执行 ADSTP 命令。

注： *仅当 ADSTART=1 且 ADDIS=0 时（ADC 已使能、最终会进行常规转换、并且没有任何待处理的禁止 ADC 的请求），才允许通过软件将 ADSTP 置 1。*

在自动注入模式下 (JAUTO=1)，将 ADSTP 位置 1 会中止常规转换和注入转换（不要使用 JADSTP）。

位 3 **JADSTART**：ADC 开始注入转换 (ADC start of injected conversion)

此位由软件置 1，用于开始 ADC 的注入通道转换。根据配置位 JEXTEN 的值，可以立即开始转换（软件触发配置），也可以在发生注入硬件触发事件后开始转换（硬件触发配置）。

该位通过硬件清零：

– 在单次转换模式下，如果选择了软件触发 (JEXTSEL=0x0)：出现注入转换序列结束 (JEOS) 标志时清零。

– 在所有情况下：执行完 JADSTP 命令后，由硬件将 JADSTP 位清零的同时清零。

0：当前未进行 ADC 注入转换。

1：写入 1 可开始注入转换。读取值为 1 表示 ADC 正在运行，最终会转换注入通道。

注： *仅当 ADEN=1 且 ADDIS=0 时（ADC 已使能，并且没有任何待处理的禁止 ADC 的请求），才允许通过软件将 JADSTART 置 1。*

在自动注入模式下 (JAUTO=1)，通过将 ADSTART 位置 1 开始常规转换和自动注入转换（JADSTART 必须保持清零）。

位 2 **ADSTART**：ADC 开始常规转换 (ADC start of regular conversion)

此位由软件置 1，用于开始 ADC 的常规通道转换。根据配置位 EXTEN 的值，可以立即开始转换（软件触发配置），也可以在发生常规硬件触发事件后开始转换（硬件触发配置）。

该位通过硬件清零：

– 在单次转换模式下，如果选择了软件触发 (EXTSEL=0x0)：出现常规转换序列结束 (EOS) 标志时清零。

– 在所有情况下：执行完 ADSTP 命令后，由硬件将 ADSTP 位清零的同时清零。

0：当前未进行 ADC 常规转换。

1：写入 1 可开始常规转换。读取值为 1 表示 ADC 正在运行，最终会转换常规通道。

注： *仅当 ADEN=1 且 ADDIS=0 时（ADC 已使能，并且没有任何待处理的禁止 ADC 的请求），才允许通过软件将 ADSTART 置 1*

在自动注入模式下 (JAUTO=1)，通过将 ADSTART 位置 1 开始常规转换和自动注入转换（JADSTART 必须保持清零）

位 1 **ADDIS**：ADC 禁止命令 (ADC disable command)

该位由软件置 1，用于禁止 ADC（ADDIS 命令）并使其进入掉电状态（OFF 状态）。

ADC 已有效禁止后，会立即通过硬件将该位清零（此时也会通过硬件将 ADEN 清零）。

0：当前未执行 ADDIS 命令

1：写入 1 可禁止 ADC。读取值为 1 表示正在执行 ADDIS 命令。

注： *仅当 ADEN=1、ADSTART=0 且 JADSTART=0 时（这可确保当前未进行任何转换），才允许通过软件将 ADDIS 置 1*

位 0 **ADEN**：ADC 使能控制 (ADC enable control)

该位由软件置 1，用于使能 ADC。ADRDY 标志置 1 后，ADC 将立即准备好运行。

如果 ADC 已禁止，则执行 ADDIS 命令后，将通过硬件对该位清零。

0：禁止 ADC（OFF 状态）

1：写入 1 来使能 ADC。

注： *仅当 ADCx_CR 寄存器的所有位均为 0 时（ADCAL=0、JADSTART=0、ADSTART=0、ADSTP=0、ADDIS=0 且 ADEN=0，但 ADVREGEN 位除外，此位必须为 1），才允许通过软件将 ADEN 置 1（并且软件必须等待稳压器的启动时间）*

21.6.4 ADC 配置寄存器 (ADC_CFGR)

ADC configuration register

偏移地址：0x0C

复位值：0x8000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
JQDIS	AWD1CH[4:0]					JAUTO	JAWD1EN	AWD1EN	AWD1SGL	JQM	JDISCEN	DISCNUM[2:0]			DISCEN
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
Res.	AUTDLY	CONT	OVRMOD	EXTEN[1:0]		EXTSEL3	EXTSEL2	EXTSEL1	EXTSEL0	ALIGN	RES[1:0]		DFSDMCFG	DMACFG	DMAEN
	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31 **JQDIS**：注入队列禁止 (Injected Queue disable)

此位由软件置 1 和清零，用于禁止注入队列机制：

0：使能注入队列

1：禁止注入队列

注： *仅当 ADSTART=0 且 JADSTART=0 时（这可确保当前未进行常规转换、也未进行注入转换），才允许通过软件对此位执行写操作。*

将 JQDIS 位置 1 或复位会导致注入队列被清空，JSQR 寄存器也会被清空。

位 30:26 **AWD1CH[4:0]**：模拟看门狗 1 通道选择 (Analog watchdog 1 channel selection)

这些位将由软件置 1 和清零。它们用于选择由模拟看门狗监控的输入通道。

00000：通过 AWD1 监控 ADC 模拟输入通道 0（仅在 ADC1 上可用）

00001：通过 AWD1 监控 ADC 模拟输入通道 1

.....

10010：通过 AWD1 监控 ADC 模拟输入通道 18

其他：保留，不得使用

注： *通过 AWD1CH 选择的通道必须也在 SQRi 或 JSQRi 寄存器中选择。*

仅当 ADSTART=0 且 JADSTART=0 时（这可确保当前未进行任何转换），才允许通过软件对这些位执行写操作。

位 25 **JAUTO**：注入组自动转换 (Automatic injected group conversion)

通过软件将该位置 1 和清零可在常规组转换后分别使能/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

注： *仅当 ADSTART=0 且 JADSTART=0 时（这可确保当前未进行常规转换、也未进行注入转换），才允许通过软件对此位执行写操作。*

位 24 **JAWD1EN**：注入通道上的模拟看门狗 1 使能 (Analog watchdog 1 enable on injected channels)

此位由软件置 1 和清零

0：在注入通道上禁止模拟看门狗 1

1：在注入通道上使能模拟看门狗 1

注： *仅当 JADSTART=0 时（这可确保当前未进行任何注入转换），才允许通过软件对此位执行写操作。*

位 23 **AWD1EN**：常规通道上的模拟看门狗 1 使能 (Analog watchdog 1 enable on regular channels)

此位由软件置 1 和清零

0：在常规通道上禁止模拟看门狗 1

1：在常规通道上使能模拟看门狗 1

注： 仅当 ADSTART=0 时（这可确保当前未进行任何常规转换），才允许通过软件对此位执行写操作。

位 22 **AWD1SGL**：在单一通道或所有通道上使能看门狗 1 (Enable the watchdog 1 on a single channel or on all channels)

此位由软件置 1 和清零，用于在通过 AWD1CH[4:0] 位确定的通道或所有通道上使能模拟看门狗。

0：在所有通道上使能模拟看门狗 1

1：在单一通道上使能模拟看门狗 1

注： 仅当 ADSTART=0 且 JADSTART=0 时（这可确保当前未进行任何转换），才允许通过软件对这些位执行写操作。

位 21 **JQM**：JSQR 队列模式 (JSQR queue mode)

此位由软件置 1 和清零。

此位定义空队列的管理方式。

0：JSQR 模式 0：队列从不为空，并会保留上一次写入 JSQR 的配置。

1：JSQR 模式 1：队列可以为空，队列为空时，会在上一个有效注入序列完成后立即在内部禁止注入序列的软件和硬件触发。

更多信息，请参见*第 21.4.21 节：注入转换的上下文队列*。

注： 仅当 JADSTART=0 时（这可确保当前未进行任何注入转换），才允许通过软件对此位执行写操作。

位 20 **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/禁止组中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

注： 仅当 JADSTART=0 时（这可确保当前未进行任何注入转换），才允许通过软件对此位执行写操作。

不能同时使用自动注入模式和不连续模式：当 JAUTO 置 1 时，DISCEN 和 JDISCEN 位必须通过软件保持清零状态。

位 19:17 **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常规通道数。

000：1 个通道

001：2 个通道

...

111：8 个通道

注： 仅当 ADSTART=0 时（这可确保当前未进行任何常规转换），才允许通过软件对这些位执行写操作。

位 16 **DISCEN**：常规通道的不连续模式 (Discontinuous mode for regular channels)

此位由软件置 1 和清零，用于使能/禁止常规通道的不连续模式。

0：禁止常规通道的不连续模式

1：使能常规通道的不连续模式

注： *不能同时使能不连续模式和连续模式：禁止同时将 DISCEN 和 CONT 位置 1。*

不能同时使用自动注入模式和不连续模式：当 JAUTO 置 1 时，DISCEN 和 JDISCEN 位必须通过软件保持清零状态。

仅当 ADSTART=0 时（这可确保当前未进行任何常规转换），才允许通过软件对此位执行写操作。

位 15 保留，必须保持复位值。

位 14 **AUTDLY**：延迟转换模式 (Delayed conversion mode)

此位由软件置 1 和清零，用于使能/禁止自动延迟转换模式。

0：自动延迟转换模式关闭

1：自动延迟转换模式开启

注： *仅当 ADSTART=0 且 JADSTART=0 时（这可确保当前未进行任何转换），才允许通过软件对此位执行写操作。*

位 13 **CONT**：常规转换的单次/连续转换模式 (Single/continuous conversion mode for regular conversions)

此位由软件置 1 和清零。该位置 1 时，常规转换将持续进行，直到该位清零。

0：单次转换模式

1：连续转换模式

注： *不能同时使能不连续模式和连续模式：禁止同时将 DISCEN 和 CONT 位置 1。*

仅当 ADSTART=0 时（这可确保当前未进行任何常规转换），才允许通过软件对此位执行写操作。

位 12 **OVRMOD**：溢出模式 (Overrun Mode)

该位由软件置 1 和清零，用于配置数据溢出的管理方式。

0：如果检测到溢出，ADCx_DR 寄存器会保留原有数据。

1：如果检测到溢出，ADCx_DR 寄存器会被上一转换结果覆盖。

注： *仅当 ADSTART=0 时（这可确保当前未进行任何常规转换），才允许通过软件对此位执行写操作。*

位 11:10 **EXTEN[1:0]**：常规通道的外部触发使能和极性选择 (External trigger enable and polarity selection 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常规组的触发。

00：禁止硬件触发检测（可通过软件启动转换）

01：在上升沿执行硬件触发检测

10：在下降沿执行硬件触发检测

11：在上升沿和下降沿都执行硬件触发检测

注： *仅当 ADSTART=0 时（这可确保当前未进行任何常规转换），才允许通过软件对这些位执行写操作。*

位 9:6 **EXTSEL[3:0]**：常规组的外部触发选择 (External trigger selection for regular group)

这些位可选择用于触发常规组转换的外部事件。

0000：事件 0
0001：事件 1
0010：事件 2
0011：事件 3
0100：事件 4
0101：事件 5
0110：事件 6
0111：事件 7
...
1111：事件 15

注： 仅当 ADSTART=0 时（这可确保当前未进行任何常规转换），才允许通过软件对这些位执行写操作。

位 5 **ALIGN**：数据对齐 (Data alignment)

此位由软件置 1 和清零，用于选择右对齐或左对齐。请参见*数据寄存器、数据对齐和偏移（ADCx_DR、OFFSETy、OFFSETy_CH、ALIGN）*一节。

0：右对齐
1：左对齐

注： 仅当 ADSTART=0 且 JADSTART=0 时（这可确保当前未进行任何转换），才允许通过软件对此位执行写操作。

位 4:3 **RES[1:0]**：数据分辨率 (Data resolution)

通过软件写入这些位可选择转换的分辨率。

00：12 位
01：10 位
10：8 位
11：6 位

注： 仅当 ADSTART=0 且 JADSTART=0 时（这可确保当前未进行任何转换），才允许通过软件对这些位执行写操作。

位 2 **DFSDMCFG**：DFSDM 模式配置 (DFSDM mode configuration)

此位由软件置 1 和清零，用于使能 DFSDM 模式。仅当 DMAEN=0 时有效。

0：禁止 DFSDM 模式
1：使能 DFSDM 模式

注： 为确保当前未执行任何转换，仅在 ADSTART = 0 且 JADSTART = 0 时才允许通过软件对此位执行写操作。

位 1 **DMACFG**：直接存储器访问配置 (Direct memory access configuration)

此位由软件置 1 和清零，用于在 DMA 两种工作模式之间进行选择，仅当 DMAEN=1 时，该位才有效。

0：选择 DMA 单次模式
1：选择 DMA 循环模式

更多详细信息，请参见*使用 DMA 管理转换*一节

注： 仅当 ADSTART=0 且 JADSTART=0 时（这可确保当前未进行任何转换），才允许通过软件对此位执行写操作。

位 0 **DMAEN**：直接存储器访问使能 (Direct memory access enable)

此位由软件置 1 和清零，用于使能 DMA 请求的生成。这样便可使用 GP-DMA 自动管理转换的数据。有关详细信息，请参见*使用 DMA 管理转换*一节。

0：禁止 DMA

1：使能 DMA

注： 仅当 ADSTART=0 且 JADSTART=0 时（这可确保当前未进行任何转换），才允许通过软件对此位执行写操作。

21.6.5 ADC 配置寄存器 2 (ADC_CFGR2)

ADC configuration 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
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.	Res.	Res.	Res.	ROVSM	TROVS	OVSS[3:0]				OVSR[2:0]			JOVSE	ROVSE
					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 保留，必须保持复位值。

位 27:17 保留，必须保持复位值。

位 16:11 保留，必须保持复位值。

位 10 **ROVSM**：常规过采样模式 (Regular Oversampling mode)

此位由软件置 1 和清零，用于选择常规过采样模式。

0：连续模式：如果触发注入转换，过采样会暂时停止，并会在注入序列完成后继续（注入序列过程中会保留过采样缓冲区）

1：恢复模式：如果触发注入转换，当前过采样会中止，并会在注入序列完成后从头开始进行过采样（注入序列开始后，过采样缓冲区会清零）

注： 仅当 ADSTART=0 时（这可确保当前未进行任何转换），才允许通过软件对此位执行写操作。

位 9 **TROVS**：已触发常规过采样 (Triggered Regular Oversampling)

此位由软件置 1 和清零，以使能已触发过采样

0：会在触发后连续完成某一通道的所有过采样转换

1：某一通道的每个过采样转换都需要重新触发

注： 仅当 ADSTART=0 时（这可确保当前未进行任何转换），才允许通过软件对此位执行写操作。

位 8:5 **OVSS[3:0]**：过采样移位 (Oversampling shift)

此位域由软件置 1 和清零，用于定义应用到原始过采样结果的右移位数。

0000：不发生移位
0001：移 1 位
0010：移 2 位
0011：移 3 位
0100：移 4 位
0101：移 5 位
0110：移 6 位
0111：移 7 位
1000：移 8 位
其他代码保留

注： 仅当 ADSTART=0 时（这可确保当前未进行任何转换），才允许通过软件对这些位执行写操作。

位 4:2 **OVSR[2:0]**：过采样率 (Oversampling ratio)

此位域由软件置 1 和清零，用于定义过采样率。

000：2x
001：4x
010：8x
011：16x
100：32x
101：64x
110：128x
111：256x

注： 仅当 ADSTART=0 时（这可确保当前未进行任何转换），才允许通过软件对这些位执行写操作。

位 1 **JOVSE**：注入过采样使能 (Injected Oversampling Enable)

此位由软件置 1 和清零，用于使能注入过采样。

0：禁止注入过采样
1：使能注入过采样

注： 仅当 ADSTART=0 且 JADSTART=0 时（这可确保当前未进行任何转换），才允许通过软件对此位执行写操作

位 0 **ROVSE**：常规过采样使能 (Regular Oversampling Enable)

此位由软件置 1 和清零，用于使能常规过采样。

0：禁止常规过采样
1：使能常规过采样

注： 仅当 ADSTART=0 且 JADSTART=0 时（这可确保当前未进行任何转换），才允许通过软件对此位执行写操作

21.6.6 ADC 采样时间寄存器 1 (ADC_SMPR1)

ADC sample time register 1

偏移地址：0x14

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	SMP9[2:0]			SMP8[2:0]			SMP7[2:0]			SMP6[2:0]			SMP5[2:1]	
		rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

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

000：2.5 个 ADC 时钟周期
001：6.5 个 ADC 时钟周期
010：12.5 个 ADC 时钟周期
011：24.5 个 ADC 时钟周期
100：47.5 个 ADC 时钟周期
101：92.5 个 ADC 时钟周期
110：247.5 个 ADC 时钟周期
111：640.5 个 ADC 时钟周期

注： 仅当 ADSTART=0 且 JADSTART=0 时（这可确保当前未进行任何转换），才允许通过软件对这些位执行写操作。

21.6.7 ADC 采样时间寄存器 2 (ADC_SMPR2)

ADC sample time register 2

偏移地址：0x18

复位值：0x0000 0000

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

000：2.5 个 ADC 时钟周期
001：6.5 个 ADC 时钟周期
010：12.5 个 ADC 时钟周期
011：24.5 个 ADC 时钟周期
100：47.5 个 ADC 时钟周期
101：92.5 个 ADC 时钟周期
110：247.5 个 ADC 时钟周期
111：640.5 个 ADC 时钟周期

注： 仅当 ADSTART=0 且 JADSTART=0 时（这可确保当前未进行任何转换），才允许通过软件对这些位执行写操作。

21.6.8 ADC 看门狗阈值寄存器 1 (ADC_TR1)

ADC watchdog threshold register 1

偏移地址：0x20

复位值：0x0FFF 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	HT1[11:0]											

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	LT1[11:0]											
				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 27:16 **HT1[11:0]**：模拟看门狗 1 阈值上限 (Analog watchdog 1 higher threshold)

这些位由软件写入，用于定义模拟看门狗 1 的阈值上限。

请参见*第 21.4.29 节：模拟窗口看门狗（AWD1EN、JAWD1EN、AWD1SGL、AWD1CH、AWD2CH、AWD3CH、AWD_HTx、AWD_LTx、AWDx）*。

注： 仅当 ADSTART=0 且 JADSTART=0 时（这可确保当前未进行任何转换），才允许通过软件对这些位执行写操作。

位 15:12 保留，必须保持复位值。

位 11:0 **LT1[11:0]**：模拟看门狗 1 阈值下限 (Analog watchdog 1 lower threshold)

这些位由软件写入，用于定义模拟看门狗 1 的阈值下限。

请参见*第 21.4.29 节：模拟窗口看门狗（AWD1EN、JAWD1EN、AWD1SGL、AWD1CH、AWD2CH、AWD3CH、AWD_HTx、AWD_LTx、AWDx）*。

注： 仅当 ADSTART=0 且 JADSTART=0 时（这可确保当前未进行任何转换），才允许通过软件对这些位执行写操作。

21.6.9 ADC 看门狗阈值寄存器 2 (ADC_TR2)

ADC watchdog threshold register 2

偏移地址：0x24

复位值：0x00FF 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.	HT2[7:0]							
								rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LT2[7:0]							
								rw	rw	rw	rw	rw	rw	rw	rw

位 31:24 保留，必须保持复位值。

位 23:16 **HT2[7:0]**：模拟看门狗 2 阈值上限 (Analog watchdog 2 higher threshold)

这些位由软件写入，用于定义模拟看门狗 2 的阈值上限。

请参见*第 21.4.29 节：模拟窗口看门狗（AWD1EN、JAWD1EN、AWD1SGL、AWD1CH、AWD2CH、AWD3CH、AWD_HTx、AWD_LTx、AWDx）*。

注：仅当 ADSTART=0 且 JADSTART=0 时（这可确保当前未进行任何转换），才允许通过软件对这些位执行写操作。

位 15:8 保留，必须保持复位值。

位 7:0 **LT2[7:0]**：模拟看门狗 2 阈值下限 (Analog watchdog 2 lower threshold)

这些位由软件写入，用于定义模拟看门狗 2 的阈值下限。

请参见*第 21.4.29 节：模拟窗口看门狗（AWD1EN、JAWD1EN、AWD1SGL、AWD1CH、AWD2CH、AWD3CH、AWD_HTx、AWD_LTx、AWDx）*。

注：仅当 ADSTART=0 且 JADSTART=0 时（这可确保当前未进行任何转换），才允许通过软件对这些位执行写操作。

21.6.10 ADC 看门狗阈值寄存器 3 (ADC_TR3)

ADC watchdog threshold register 3

偏移地址：0x28

复位值：0x00FF 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.	HT3[7:0]							
								rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LT3[7:0]							
								rw	rw	rw	rw	rw	rw	rw	rw

位 31:24 保留，必须保持复位值。

位 23:16 **HT3[7:0]**：模拟看门狗 3 阈值上限 (Analog watchdog 3 higher threshold)
这些位由软件写入，用于定义模拟看门狗 3 的阈值上限。
请参见*第 21.4.29 节：模拟窗口看门狗（AWD1EN、JAWD1EN、AWD1SGL、AWD1CH、AWD2CH、AWD3CH、AWD_HTx、AWD_LTx、AWDx）*。
注：仅当 ADSTART=0 且 JADSTART=0 时（这可确保当前未进行任何转换），才允许通过软件对这些位执行写操作。

位 15:8 保留，必须保持复位值。

位 7:0 **LT3[7:0]**：模拟看门狗 3 阈值下限 (Analog watchdog 3 lower threshold)
这些位由软件写入，用于定义模拟看门狗 3 的阈值下限。
此看门狗将 LT3 的 8 位与转换后数据的高 8 位进行比较。
注：仅当 ADSTART=0 且 JADSTART=0 时（这可确保当前未进行任何转换），才允许通过软件对这些位执行写操作。

21.6.11 ADC 常规序列寄存器 1 (ADC_SQR1)

ADC regular sequence register 1

偏移地址：0x30

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	SQ4[4:0]					Res.	SQ3[4:0]					Res.	SQ2[4]
			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
SQ2[3:0]				Res.	SQ1[4:0]					Res.	Res.	L[3:0]			
rw	rw	rw	rw		rw	rw	rw	rw	rw			rw	rw	rw	rw

位 31:29 保留，必须保持复位值。

位 28:24 **SQ4[4:0]**：常规序列中的第四次转换 (4th conversion in regular sequence)
通过软件写入这些位，并将通道编号 (0..18) 分配为常规转换序列中的第四次转换。
注：仅当 ADSTART=0 时（这可确保当前未进行任何常规转换），才允许通过软件对这些位执行写操作。

位 23 保留，必须保持复位值。

位 22:18 **SQ3[4:0]**：常规序列中的第三次转换 (3rd conversion in regular sequence)
通过软件写入这些位，并将通道编号 (0..18) 分配为常规转换序列中的第三次转换。
注：仅当 ADSTART=0 时（这可确保当前未进行任何常规转换），才允许通过软件对这些位执行写操作。

位 17 保留，必须保持复位值。

位 16:12 **SQ2[4:0]**：常规序列中的第二次转换 (2nd conversion in regular sequence)
通过软件写入这些位，并将通道编号 (0..18) 分配为常规转换序列中的第二次转换。
注：仅当 ADSTART=0 时（这可确保当前未进行任何常规转换），才允许通过软件对这些位执行写操作。

位 11 保留，必须保持复位值。

位 10:6 **SQ1[4:0]**：常规序列中的第一次转换 (1st conversion in regular sequence)
通过软件写入这些位，并将通道编号 (0..18) 分配为常规转换序列中的第一次转换。
注： 仅当 ADSTART=0 时（这可确保当前未进行任何常规转换），才允许通过软件对这些位执行写操作。

位 5:4 保留，必须保持复位值。

位 3:0 **L[3:0]**：常规通道序列长度 (Regular channel sequence length)
通过软件写入这些位可定义常规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
注： 仅当 ADSTART=0 时（这可确保当前未进行任何常规转换），才允许通过软件对这些位执行写操作。

21.6.12 ADC 常规序列寄存器 2 (ADC_SQR2)

ADC regular sequence register 2

偏移地址：0x34

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	SQ9[4:0]					Res.	SQ8[4:0]					Res.	SQ7[4]
			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
SQ7[3:0]				Res.	SQ6[4:0]					Res.	SQ5[4:0]				
rw	rw	rw	rw		rw	rw	rw	rw	rw		rw	rw	rw	rw	rw

位 31:29 保留，必须保持复位值。

位 28:24 **SQ9[4:0]**：常规序列中的第九次转换 (9th conversion in regular sequence)
通过软件写入这些位，并将通道编号 (0..18) 分配为常规转换序列中的第九次转换。
注： 仅当 ADSTART=0 时（这可确保当前未进行任何常规转换），才允许通过软件对这些位执行写操作。

位 23 保留，必须保持复位值。

位 22:18 **SQ8[4:0]**：常规序列中的第八次转换 (8th conversion in regular sequence)
通过软件写入这些位，并将通道编号 (0..18) 分配为常规转换序列中的第八次转换
注： 仅当 ADSTART=0 时（这可确保当前未进行任何常规转换），才允许通过软件对这些位执行写操作。

位 17 保留，必须保持复位值。

位 16:12 **SQ7[4:0]**：常规序列中的第七次转换 (7th conversion in regular sequence)
通过软件写入这些位，并将通道编号 (0..18) 分配为常规转换序列中的第七次转换。
注： 仅当 ADSTART=0 时（这可确保当前未进行任何常规转换），才允许通过软件对这些位执行写操作。

位 11 保留，必须保持复位值。

位 10:6 **SQ6[4:0]**：常规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) 分配为常规转换序列中的第六次转换。
注： 仅当 ADSTART=0 时（这可确保当前未进行任何常规转换），才允许通过软件对这些位执行写操作。

位 5 保留，必须保持复位值。

位 4:0 **SQ5[4:0]**：常规序列中的第五次转换 (5th conversion in regular sequence)
通过软件写入这些位，并将通道编号 (0..18) 分配为常规转换序列中的第五次转换。
注： 仅当 ADSTART=0 时（这可确保当前未进行任何常规转换），才允许通过软件对这些位执行写操作。

21.6.13 ADC 常规序列寄存器 3 (ADC_SQR3)

ADC regular sequence register 3

偏移地址：0x38

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	SQ14[4:0]					Res.	SQ13[4:0]					Res.	SQ12[4]
			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
SQ12[3:0]				Res.	SQ11[4:0]					Res.	SQ10[4:0]				
rw	rw	rw	rw		rw	rw	rw	rw	rw		rw	rw	rw	rw	rw

位 31:29 保留，必须保持复位值。

位 28:24 **SQ14[4:0]**：常规序列中的第十四次转换 (14th conversion in regular sequence)
通过软件写入这些位，并将通道编号 (0..18) 分配为常规转换序列中的第十四次转换。
注： 仅当 ADSTART=0 时（这可确保当前未进行任何常规转换），才允许通过软件对这些位执行写操作。

位 23 保留，必须保持复位值。

位 22:18 **SQ13[4:0]**：常规序列中的第十三次转换 (13th conversion in regular sequence)
通过软件写入这些位，并将通道编号 (0..18) 分配为常规转换序列中的第十三次转换。
注： 仅当 ADSTART=0 时（这可确保当前未进行任何常规转换），才允许通过软件对这些位执行写操作。

位 17 保留，必须保持复位值。

位 16:12 **SQ12[4:0]**：常规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) 分配为常规转换序列中的第十二次转换。
注： 仅当 ADSTART=0 时（这可确保当前未进行任何常规转换），才允许通过软件对这些位执行写操作。

位 11 保留，必须保持复位值。

位 10:6 **SQ11[4:0]**：常规序列中的第十一次转换 (11th conversion in regular sequence)
通过软件写入这些位，并将通道编号 (0..18) 分配为常规转换序列中的第十一次转换。
注： 仅当 ADSTART=0 时（这可确保当前未进行任何常规转换），才允许通过软件对这些位执行写操作。

位 5 保留，必须保持复位值。

位 4:0 **SQ10[4:0]**：常规序列中的第十次转换 (10th conversion in regular sequence)
通过软件写入这些位，并将通道编号 (0..18) 分配为常规转换序列中的第十次转换。
注： 仅当 ADSTART=0 时（这可确保当前未进行任何常规转换），才允许通过软件对这些位执行写操作。

21.6.14 ADC 常规序列寄存器 4 (ADC_SQR4)

ADC regular sequence register 4

偏移地址：0x3C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	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.	Res.	Res.	Res.	SQ16[4:0]					Res.	SQ15[4:0]				
					rw	rw	rw	rw	rw		rw	rw	rw	rw	rw

位 31:11 保留，必须保持复位值。

位 10:6 **SQ16[4:0]**：常规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) 分配为常规转换序列中的第十六次转换。
注： 仅当 ADSTART=0 时（这可确保当前未进行任何常规转换），才允许通过软件对这些位执行写操作。

位 5 保留，必须保持复位值。

位 4:0 **SQ15[4:0]**：常规序列中的第十五次转换 (15th conversion in regular sequence)
通过软件写入这些位，并将通道编号 (0..18) 分配为常规转换序列中的第十五次转换。
注： 仅当 ADSTART=0 时（这可确保当前未进行任何常规转换），才允许通过软件对这些位执行写操作。

21.6.15 ADC 常规数据寄存器 (ADC_DR)

ADC regular Data Register

偏移地址：0x40

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	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
RDATA[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 **RDATA[15:0]**：已转换常规数据 (Regular Data converted)

这些位为只读。其中包含上一转换常规通道的转换结果。数据有左对齐和右对齐两种方式，如*第 21.4.26 节：数据管理*所述。

21.6.16 ADC 注入序列寄存器 (ADC_JSQR)

ADC injected sequence 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
Res.	JSQ4[4:0]					Res.	JSQ3[4:0]					Res.	JSQ2[4:2]		
	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
JSQ2[1:0]		Res.	JSQ1[4:0]					JEXTEN[1:0]		JEXTSEL[3:0]				JL[1:0]	
rw	rw		rw	rw	rw	rw	rw	rw		rw	rw	rw	rw	rw	rw

位 31 保留，必须保持复位值。

位 30:26 **JSQ4[4:0]**：注入序列中的第四次转换 (4th conversion in the injected sequence)

通过软件写入这些位，并将通道编号 (0..18) 分配为注入转换序列中的第四次转换。

注： ADC 使能后 (ADEN=1)，允许随时通过软件写入这些位。

位 25 保留，必须保持复位值。

位 24:20 **JSQ3[4:0]**：注入序列中的第三次转换 (3rd conversion in the injected sequence)

通过软件写入这些位，并将通道编号 (0..18) 分配为注入转换序列中的第三次转换。

注： ADC 使能后 (ADEN=1)，允许随时通过软件写入这些位。

位 19 保留，必须保持复位值。

位 18:14 **JSQ2[4:0]**：注入序列中的第二次转换 (2nd conversion in the injected sequence)

通过软件写入这些位，并将通道编号 (0..18) 分配为注入转换序列中的第二次转换。

注： ADC 使能后 (ADEN=1)，允许随时通过软件写入这些位。

位 13 保留，必须保持复位值。

位 12:8 **JSQ1[4:0]**：注入序列中的第一次转换 (1st conversion in the injected sequence)
通过软件写入这些位，并将通道编号 (0..18) 分配为注入转换序列中的第一次转换。
注： ADC 使能后 (ADEN=1)，允许随时通过软件写入这些位。

位 7:6 **JEXTEN[1:0]**：注入通道的外部触发使能和极性选择 (External Trigger Enable and Polarity Selection 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：如果 JQDIS=0（队列使能），同时禁止硬件和软件触发检测
00：如果 JQDIS=1（队列禁止），禁止硬件触发检测（可通过软件启动转换）
01：在上升沿执行硬件触发检测
10：在下降沿执行硬件触发检测
11：在上升沿和下降沿都执行硬件触发检测
注： ADC 使能后 (ADEN=1)，允许随时通过软件写入这些位。
如果 JQM=1 且上下文队列为空，则会在内部禁止注入序列的软件和硬件触发（请参见第 21.4.21 节：注入转换的上下文队列）

位 5:2 **JEXTSEL[3:0]**：注入组的外部触发选择 (External Trigger Selection for injected group)
这些位可选择用于触发注入组转换的外部事件：
0000：事件 0
0001：事件 1
0010：事件 2
0011：事件 3
0100：事件 4
0101：事件 5
0110：事件 6
0111：事件 7
...
1111：事件 15
注： ADC 使能后 (ADEN=1)，允许随时通过软件写入这些位。

位 1:0 **JL[1:0]**：注入通道序列长度 (Injected channel 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
注： ADC 使能后 (ADEN=1)，允许随时通过软件写入这些位。

21.6.17 ADC 偏移 y 寄存器 (ADC_OFRy)

ADC offset y register

偏移地址：0x60 + x * 0x100 + 0x04 * (y−1)，y = 1 到 4

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
OFFSETy_EN	OFFSETy_CH[4:0]					Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
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.	Res.	OFFSETy[11:0]											
				rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31 **OFFSETy_EN**：偏移 y 使能 (Offset y Enable)

该位通过软件写入，用于使能和禁止编程到 OFFSETy[11:0] 位中的偏移。

注： 仅当 ADSTART=0 且 JADSTART=0 时（这可确保当前未进行任何转换），才允许通过软件对此位执行写操作。

位 30:26 **OFFSETy_CH[4:0]**：数据偏移 y 的通道选择 (Channel selection for the Data offset y)

这些位通过软件写入，用于定义编程到 OFFSETy[11:0] 位中的偏移将应用于的通道。

注： 仅当 ADSTART=0 且 JADSTART=0 时（这可确保当前未进行任何转换），才允许通过软件对这些位执行写操作。

位 25:12 保留，必须保持复位值。

位 11:0 **OFFSETy[11:0]**：编程到 OFFSETy_CH[4:0] 位中的通道对应的数据偏移 y (Data offset y for the channel programmed into bits OFFSETy_CH[4:0])

这些位通过软件写入，用于定义在转换通道（可以是常规通道或注入通道）时从原始转换数据中减去的偏移 y。应用数据偏移 y 的通道必须在 OFFSETy_CH[4:0] 位中编程。转换结果可从 ADCx_DR（常规转换）或 ADCx_JDRyi 寄存器（注入转换）中读取。

注： 仅当 ADSTART=0 且 JADSTART=0 时（这可确保当前未进行任何转换），才允许通过软件对这些位执行写操作。

如果多个偏移 (OFFSETy) 指向同一通道，相减时只会考虑使用 x 值最小的偏移。
例如：如果 OFFSET1_CH[4:0]=4 且 OFFSET2_CH[4:0]=4，转换通道 4 时会减去 OFFSET1[11:0]。

21.6.18 ADC 注入通道 y 数据寄存器 (ADC_JDRy)

ADC injected channel y data register

偏移地址：0x80 + 0x04 * (y-1)，y= 1 到 4

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
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 y 的转换结果。数据有左对齐和右对齐两种方式，如*第 21.4.26 节：数据管理*所述。

21.6.19 ADC 模拟看门狗 2 配置寄存器 (ADC_AWD2CR)

ADC Analog Watchdog 2 Configuration Register

偏移地址：0xA0

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.	AWD2CH[18:16]		
													rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AWD2CH[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:19 保留，必须保持复位值。

位 18:0 **AWD2CH[18:0]**：模拟看门狗 2 通道选择 (Analog watchdog 2 channel selection)
这些位将由软件置 1 和清零。它们用于使能和选择由模拟看门狗 2 监控的输入通道。
AWD2CH[i] = 0：不通过 AWD2 监控 ADC 模拟输入通道 i
AWD2CH[i] = 1：通过 AWD2 监控 ADC 模拟输入通道 i
当 AWD2CH[18:0] = 000..0 时，会禁止模拟看门狗 2

注： 通过 AWD2CH 选择的通道必须也在 SQRi 或 JSQRi 寄存器中选择。
仅当 ADSTART=0 且 JADSTART=0 时（这可确保当前未进行任何转换），才允许通过软件对这些位执行写操作。

21.6.20 ADC 模拟看门狗 3 配置寄存器 (ADC_AWD3CR)

ADC Analog Watchdog 3 Configuration Register

偏移地址：0xA4

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.	AWD3CH[18:16]		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AWD3CH[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:19 保留，必须保持复位值。

位 18:0 **AWD3CH[18:0]**：模拟看门狗 3 通道选择 (Analog watchdog 3 channel selection)
这些位将由软件置 1 和清零。它们用于使能和选择由模拟看门狗 3 监控的输入通道。
AWD3CH[i] = 0：不通过 AWD3 监控 ADC 模拟输入通道 i
AWD3CH[i] = 1：通过 AWD3 监控 ADC 模拟输入通道 i
当 AWD3CH[18:0] = 000..0 时，会禁止模拟看门狗 3

注： 通过 AWD3CH 选择的通道必须也在 SQRi 或 JSQRi 寄存器中选择。
仅当 ADSTART=0 且 JADSTART=0 时（这可确保当前未进行任何转换），才允许通过软件对这些位执行写操作。

21.6.21 ADC 差分模式选择寄存器 (ADC_DIFSEL)

ADC Differential mode Selection Register

偏移地址：0xB0

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.	DIFSEL[18:16]		
													r	r	r

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

位 31:19 保留，必须保持复位值。

位 18:16 **DIFSEL[18:16]**：通道 18 到 16 的差分模式 (Differential mode for channels 18 to 16)。
这些位为只读。强制这些通道进入单端输入模式（连接到单端 I/O 端口或内部通道）。

位 15:1 **DIFSEL[15:1]**：通道 15 到 1 的差分模式 (Differential mode for channels 15 to 1)
这些位将由软件置 1 和清零。它们用于选择将通道配置为单端模式或差分模式。
DIFSEL[i] = 0：将 ADC 模拟输入通道 i 配置为单端模式
DIFSEL[i] = 1：将 ADC 模拟输入通道 i 配置为差分模式

注： 仅当 ADC 已禁止时（ADCAL=0、JADSTART=0、JADSTP=0、ADSTART=0、ADSTP=0、ADDIS=0 且 ADEN=0），才允许通过软件对这些位执行写操作。

位 0 **DIFSEL[0]**：通道 0 的差分模式 (Differential mode for channel 0)
该位为只读。强制该通道进入单端输入模式（连接到内部通道）。

21.6.22 ADC 校准系数 (ADC_CALFACT)

ADC Calibration Factors

偏移地址：0xB4

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.	CALFACT_D[6:0]						
									rw	rw	rw	rw	rw	rw	rw

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

位 31:23 保留，必须保持复位值。

位 22:16 **CALFACT_D[6:0]**：差分模式下的校准系数 (Calibration Factors in differential mode)

这些位可由硬件或软件写入。

差分输入校准完成后，会立即由硬件更新为校准系数。

软件可向这些位写入新的校准系数。如果新校准系数不同于当前存储于模拟 ADC 中的校准系数，启动新的差分校准后，会立即应用新校准系数。

注： 仅当 ADEN=1、ADSTART=0 且 JADSTART=0 时（ADC 已使能、当前未执行任何校准和转换），才允许通过软件对这些位执行写操作。

位 15:7 保留，必须保持复位值。

位 6:0 **CALFACT_S[6:0]**：单端模式下的校准系数 (Calibration Factors In single-ended mode)

这些位可由硬件或软件写入。

单端输入校准完成后，会立即由硬件更新为校准系数。

软件可向这些位写入新的校准系数。如果新校准系数不同于当前存储于模拟 ADC 中的校准系数，启动新的单端校准后，会立即应用新校准系数。

注： 仅当 ADEN=1、ADSTART=0 且 JADSTART=0 时（ADC 已使能、当前未执行任何校准和转换），才允许通过软件对这些位执行写操作。

21.7 ADC 通用寄存器

这些寄存器定义主 ADC 和从 ADC 共用的控制和状态寄存器：

21.7.1 ADC 通用状态寄存器 (ADC_CSR)

ADCCommon status register

偏移地址：0x00（该偏移地址与主 ADC 基址 + 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 ADCx_ISR 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
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.	Res.	Res.	Res.	JQOVF_MST	AWD3_MST	AWD2_MST	AWD1_MST	JEOS_MST	JEOC_MST	OVR_MST	EOS_MST	EOC_MST	EOSMP_MST	ADRDY_MST
					r	r	r	r	r	r	r	r	r	r	r

位 31:11 保留，必须保持复位值。

位 10 **JQOVF_MST**：主 ADC 的注入上下文队列溢出标志 (Injected Context Queue Overflow flag of the master ADC)

该位是相应 ADCx_ISR 寄存器中 JQOVF 位的副本。

位 9 **AWD3_MST**：主 ADC 的模拟看门狗 3 标志 (Analog watchdog 3 flag of the master ADC)

该位是相应 ADCx_ISR 寄存器中 AWD3 位的副本。

位 8 **AWD2_MST**：主 ADC 的模拟看门狗 2 标志 (Analog watchdog 2 flag of the master ADC)

该位是相应 ADCx_ISR 寄存器中 AWD2 位的副本。

位 7 **AWD1_MST**：主 ADC 的模拟看门狗 1 标志 (Analog watchdog 1 flag of the master ADC)
该位是相应 ADCx_ISR 寄存器中 AWD1 位的副本。

位 6 **JEOS_MST**：主 ADC 注入序列结束标志 (End of injected sequence flag of the master ADC)
该位是相应 ADCx_ISR 寄存器中 JEOS 位的副本。

位 5 **JEOC_MST**：主 ADC 注入转换结束标志 (End of injected conversion flag of the master ADC)
该位是相应 ADCx_ISR 寄存器中 JEOC 位的副本。

位 4 **OVR_MST**：主 ADC 的溢出标志 (Overrun flag of the master ADC)
该位是相应 ADCx_ISR 寄存器中 OVR 位的副本。

位 3 **EOS_MST**：主 ADC 常规序列结束标志 (End of regular sequence flag of the master ADC)
该位是相应 ADCx_ISR 寄存器中 EOS 位的副本。

位 2 **EOC_MST**：主 ADC 常规转换结束标志 (End of regular conversion of the master ADC)
该位是相应 ADCx_ISR 寄存器中 EOC 位的副本。

位 1 **EOSMP_MST**：主 ADC 采样阶段结束标志 (End of Sampling phase flag of the master ADC)
该位是相应 ADCx_ISR 寄存器中 EOSMP 位的副本。

位 0 **ADRDY_MST**：主 ADC 就绪 (Master ADC ready)
该位是相应 ADCx_ISR 寄存器中 ADRDY 位的副本。

21.7.2 ADC 通用控制寄存器 (ADC_CCR)

ADCcommon control register

偏移地址：0x08（该偏移地址与主 ADC 基址 + 0x300 相关）

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	CH18SEL	CH17SEL	VREFEN	PRESC[3:0]				CKMODE[1:0]	
							rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

位 31:25 保留，必须保持复位值。

位 24 **CH18SEL**：CH18 选择 (CH18 selection)
通过软件将该位置 1 和清零可控制通道 18
0：禁止 V_{BAT} 通道
1：使能 V_{BAT} 通道

位 23 **CH17SEL**：CH17 选择 (CH17 selection)
通过软件将该位置 1 和清零可控制通道 17
0：禁止温度传感器通道
1：使能温度传感器通道

位 22 **VREFEN**：VREFINT 使能 (VREFINT enable)

通过软件将该位置 1 和清零可使能/禁止 V_{REFINT} 通道

0：禁止 V_{REFINT} 通道

1：使能 V_{REFINT} 通道

位 21:18 **PRESC[3:0]**：ADC 预分频系数 (ADC prescaler)

这些位由软件置 1 和清零，用于选择 ADC 的时钟频率。该时钟为所有 ADC 所共用。

0000：输入 ADC 时钟未分频
0001：输入 ADC 时钟 2 分频
0010：输入 ADC 时钟 4 分频
0011：输入 ADC 时钟 6 分频
0100：输入 ADC 时钟 8 分频
0101：输入 ADC 时钟 10 分频
0110：输入 ADC 时钟 12 分频
0111：输入 ADC 时钟 16 分频
1000：输入 ADC 时钟 32 分频
1001：输入 ADC 时钟 64 分频
1010：输入 ADC 时钟 128 分频
1011：输入 ADC 时钟 256 分频
其他：保留

注： 仅当 ADC 已禁止时（ADCAL=0、JADSTART=0、ADSTART=0、ADSTP=0、ADDIS=0 且 ADEN=0），才允许通过软件对这些位执行写操作。仅当 CKMODE[1:0] = 0b00 时，才会应用 ADC 预分频值。

位 17:16 **CKMODE[1:0]**：ADC 时钟模式 (ADC clock mode)

这些位由软件置 1 和清零，用于定义 ADC 时钟方案（主 ADC 和从 ADC 共用）：

00：CK_ADCx (x=123)（异步时钟模式），在产品级生成（请参见*第 6 节：复位和时钟控制 (RCC)*）

01：HCLK/1（同步时钟模式）。仅当 AHB 时钟预分频器设为 1（RCC_CFGR 寄存器中的 HPRE[3:0] = 0xxx）且系统时钟占空比为 50% 时，才必须使能此配置。

10：HCLK/2（同步时钟模式）

11：HCLK/4（同步时钟模式）

在所有同步时钟模式下，从定时器触发到转换开始的延迟过程中，不存在抖动。

注： 仅当 ADC 已禁止时（ADCAL=0、JADSTART=0、ADSTART=0、ADSTP=0、ADDIS=0 且 ADEN=0），才允许通过软件对这些位执行写操作。

位 15:0 保留，必须保持复位值。

21.7.3 ADC 寄存器映射

表 125. 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_ISR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	JQOVF	AWD3	AWD2	AWD1	JEOS	JEOC	OVR	EOS	EOC	EOSMP	ADRDY
	Reset value																						0	0	0	0	0	0	0	0	0	0	0

表 125. 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
0x04	**ADC_IER**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	JQOVFIE	AWD3IE	AWD2IE	AWD1IE	JEOSIE	JEOCIE	OVRIE	EOSIE	EOCIE	EOSMPIE	ADRDYIE
	Reset value																						0	0	0	0	0	0	0	0	0	0	0
0x08	**ADC_CR**	ADCAL	ADCALDIF	DEEPPWD	ADVREGEN	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	JADSTP	ADSTP	JADSTART	ADSTART	ADDIS	ADEN
	Reset value	0	0	1	0																							0	0	0	0	0	0
0x0C	**ADC_CFGR**	JQDIS.	AWD1CH[4:0]					JAUTO	JAWD1EN	AWD1EN	AWD1SGL	JQM	JDISCEN	DISCNUM [2:0]			DISCEN	Res.	AUTDLY	CONT	OVRMOD	EXTEN[1:0]		EXTSEL3	EXTSEL2	EXTSEL1	EXTSEL0	ALIGN	RES [1:0]		DFSDMCFG	DMACFG	DMAEN
	Reset value	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	0	0	0
0x0C	**ADC_CFGR2**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ROVSM	TROVS	OVSS[3:0]				OVSR [2:0]			JOVSE	ROVSE
	Reset value																						0	0	0	0	0	0	0	0	0	0	0
0x14	**ADC_SMPR1**	Res.	Res.	SMP9 [2:0]			SMP8 [2:0]			SMP7 [2:0]			SMP6 [2:0]			SMP5 [2:0]			SMP4 [2:0]			SMP3 [2:0]			SMP2 [2:0]			SMP1 [2:0]			SMP0 [2:0]		
	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
0x18	**ADC_SMPR2**	Res.	Res.	Res.	Res.	Res.	SMP18 [2:0]			SMP17 [2:0]			SMP16 [2:0]			SMP15 [2:0]			SMP14 [2:0]			SMP13 [2:0]			SMP12 [2:0]			SMP11 [2:0]			SMP10 [2:0]		
	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
0x1C	Reserved	Res.																															
0x20	**ADC_TR1**	Res.	Res.	Res.	Res.	HT1[11:0]												Res.	Res.	Res.	Res.	LT1[11:0]												
	Reset value					1	1	1	1	1	1	1	1	1	1	1	1					0	0	0	0	0	0	0	0	0	0	0	0
0x24	**ADC_TR2**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	HT2[7:0]								Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LT2[7:0]							
	Reset value									1	1	1	1	1	1	1	1									0	0	0	0	0	0	0	0
0x28	**ADC_TR3**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	HT3[7:0]								Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LT3[7:0]							
	Reset value									1	1	1	1	1	1	1	1									0	0	0	0	0	0	0	0
0x2C	Reserved	Res.																															
0x30	**ADC_SQR1**	Res.	Res.	Res.	SQ4[4:0]					Res.	SQ3[4:0]					Res.	SQ2[4:0]					Res.	SQ1[4:0]					Res.	Res.	L[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
0x34	**ADC_SQR2**	Res.	Res.	Res.	SQ9[4:0]					Res.	SQ8[4:0]					Res.	SQ7[4:0]					Res.	SQ6[4:0]					Res.	SQ5[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	0	0	0	0
0x38	**ADC_SQR3**	Res.	Res.	Res.	SQ14[4:0]					Res.	SQ13[4:0]					Res.	SQ12[4:0]					Res.	SQ11[4:0]					Res.	SQ10[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	0	0	0	0
0x3C	**ADC_SQR4**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SQ16[4:0]					Res.	SQ15[4:0]				
	Reset value																						0	0	0	0	0		0	0	0	0	0
0x40	**ADC_DR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	regular RDATA[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x44-0x48	Reserved	Res.																															
0x4C	**ADC_JSQR**	Res.	JSQ4[4:0]					Res.	JSQ3[4:0]					Res.	JSQ2[4:0]			Res.	JSQ1[4:0]				JEXTEN[1:0]	JEXTSEL [3:0]						JL[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
0x50-0x5C	Reserved	Res.																															

表 125. 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
0x60	**ADC_OFR1**	OFFSET1_EN	OFFSET1_CH[4:0]					Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OFFSET1[11:0]											
	Reset value	0	0	0	0	0	0															0	0	0	0	0	0	0	0	0	0	0	0
0x64	**ADC_OFR2**	OFFSET2_EN	OFFSET2_CH[4:0]					Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OFFSET2[11:0]											
	Reset value	0	0	0	0	0	0															0	0	0	0	0	0	0	0	0	0	0	0
0x68	**ADC_OFR3**	OFFSET3_EN	OFFSET3_CH[4:0]					Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OFFSET3[11:0]											
	Reset value	0	0	0	0	0	0															0	0	0	0	0	0	0	0	0	0	0	0
0x6C	**ADC_OFR4**	OFFSET4_EN	OFFSET4_CH[4:0]					Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OFFSET4[11:0]											
	Reset value	0	0	0	0	0	0															0	0	0	0	0	0	0	0	0	0	0	0
0x70-0x7C	Reserved	Res.																															
0x80	**ADC_JDR1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	JDATA1[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x84	**ADC_JDR2**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	JDATA2[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x88	**ADC_JDR3**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	JDATA3[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x8C	**ADC_JDR4**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	JDATA4[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x8C-0x9C	Reserved	Res.																															
0xA0	**ADC_AWD2CR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	AWD2CH[18:0]																		
	Reset value														0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0xA4	**ADC_AWD3CR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	AWD3CH[18:0]																		
	Reset value														0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0xA8-0xAC	Reserved	Res.																															
0xB0	**ADC_DIFSEL**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DIFSEL[18:0]																		
	Reset value														0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0xB4	**ADC_CALFACT**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CALFACT_D[6:0]							Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CALFACT_S[6:0]							
	Reset value										0	0	0	0	0	0	0									0	0	0	0	0	0	0

表 126. ADC 寄存器映射和复位值（主 ADC 和从 ADC 通用寄存器，偏移 = 0x300）

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**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	JQOVF_MST	AWD3_MST	AWD2_MST	AWD1_MST	JEOS_MST	JEOC_MST	OVR_MST	EOS_MST	EOC_MST	EOSMP_MST	ADRDY_MST
	Reset value																						0	0	0	0	0	0	0	0	0	0	0
0x04	Reserved	Res.																															
0x08	**ADC_CCR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CH18SEL	CH17SEL	VREFEN	PRESC[3:0]				CKMODE[1:0]		Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	Reset value								0	0	0	0	0	0	0	0	0																

有关寄存器边界地址的信息，请参见*第 2.2.2 节：存储器映射和寄存器边界地址*。

22 数模转换器 (DAC)

22.1 简介

DAC 模块是 12 位电压输出数模转换器。DAC 可以按 8 位或 12 位模式进行配置，并且可与 DMA 控制器配合使用。在 12 位模式下，数据可以采用左对齐或右对齐。DAC 有两个输出通道，每个通道各有一个转换器。在 DAC 双通道模式下，每个通道可以单独进行转换；当两个通道组合在一起同步执行更新操作时，也可以同时进行转换。可通过一个输入参考电压引脚 V_{REF+}（与其他模拟外设共享）来提高分辨率。还可以在同一输入上设置内部参考。请参见电压参考缓冲器 *(VREFBUF)* 一节。

如果 DAC 输出与输出焊盘断开连接并连接到片上外设，DAC_OUTx 引脚可用作通用输入/输出 (GPIO)。可选择使能 DAC 输出缓冲器，从而支持高驱动输出电流。可分别对每条 DAC 输出通道应用校准。DAC 输出通道支持低功耗模式，即采样和保持模式。

22.2 DAC 主要特性

DAC 的主要特性如下（请参见*图 136：双通道 DAC 框图*）

- 一个 DAC 接口，最多对应两个输出通道
- 12 位模式下数据采用左对齐或右对齐
- 同步更新功能
- 噪声波和三角波生成
- DAC 双通道单独或同时转换
- 每条通道都有 DMA 功能，包括 DMA 下溢错误检测
- 通过外部触发信号进行转换
- DAC 输出通道缓冲/非缓冲模式
- 缓冲器偏移校准
- 每路 DAC 输出均可与 DAC_OUTx 输出引脚断开连接
- DAC 输出可与片上外设连接
- 可在停止模式下通过采样和保持模式实现低功耗运行
- 输入参考电压 V_{REF+}

图 136 所示为 DAC 通道的框图，*表 128* 则给出了引脚说明。

22.3 DAC 实现

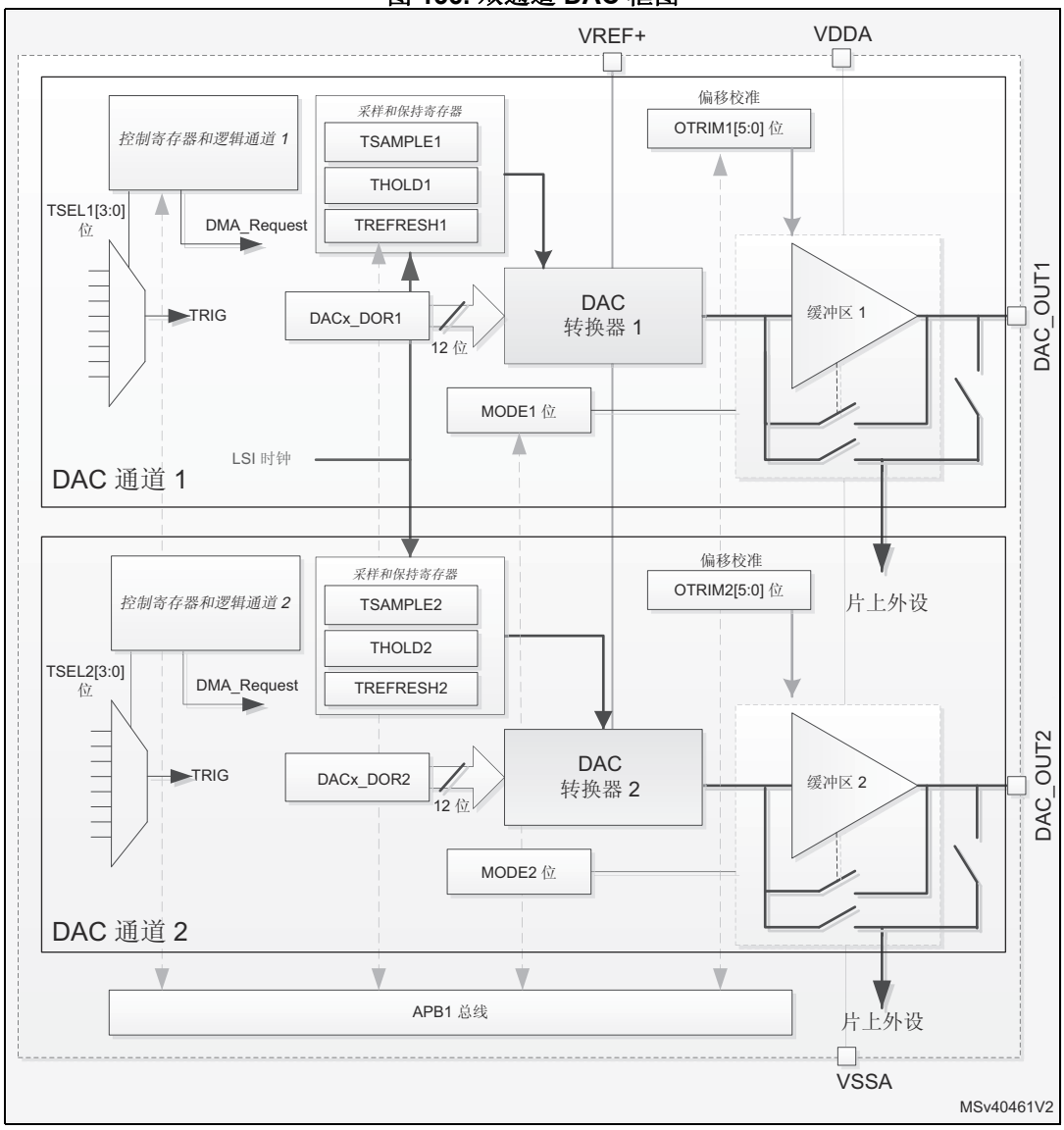
表 127. DAC 实现

DAC 特性	DAC1
双通道	X
I/O 连接	PA4 上的 DAC1_OUT1，PA5 上的 DAC1_OUT2

22.4 DAC 功能说明

22.4.1 DAC框图

图 136. 双通道 DAC 框图



1. 输出模式控制器在正常模式（采用缓冲/非缓冲配置）与采样和保持模式之间切换。
2. 在单通道 DAC 上，通道 2 不可用。

DAC 包含：

- 多达两条输出通道
- DAC_OUTx 可与输出引脚断开连接并用作普通 GPIO
- DAC_OUTx 可使用与片上外设（如比较器和 OPAMP）的内部引脚连接
- DAC 输出通道（缓冲/非缓冲）
- 采样保持模块及寄存器使用 LSI 时钟源。可以运行在停止模式以实现静态转换

DAC 包含多达两条独立的输出通道。每条输出通道均可连接到片上外设，如 COMP、OPAMP 和 ADC。在这种情况下，DAC 输出通道可与 DAC_OUTx 输出引脚断开连接，相应的 GPIO 可用于其他用途。

DAC 输出可缓冲、也可以不缓冲。采样和保持模块及其关联寄存器可在停止模式下使用 LSI 时钟源运行。

表 128. DAC 输入/输出引脚

引脚名称	信号类型	注释
V_{REF+}	正模拟参考电压输入	DAC 高/正参考电压，$V_{REF+} \le V_{DDAmax}$（请参见数据手册）
VDDA	模拟电源输入	模拟电源
VSSA	模拟电源地输入	模拟电源地
DAC_OUTx	模拟输出信号	DAC 通道 x 模拟输出

22.4.2 DAC 通道使能

将 DAC_CR 寄存器中的相应 ENx 位置 1，即可使能对应 DAC 通道。经过一段启动时间 t_{WAKEUP} 后，DAC 通道被真正使能。

注： *ENx 位只会使能模拟 DAC Channelx。即使 ENx 位复位，DAC Channelx 数字接口仍处于使能状态。*

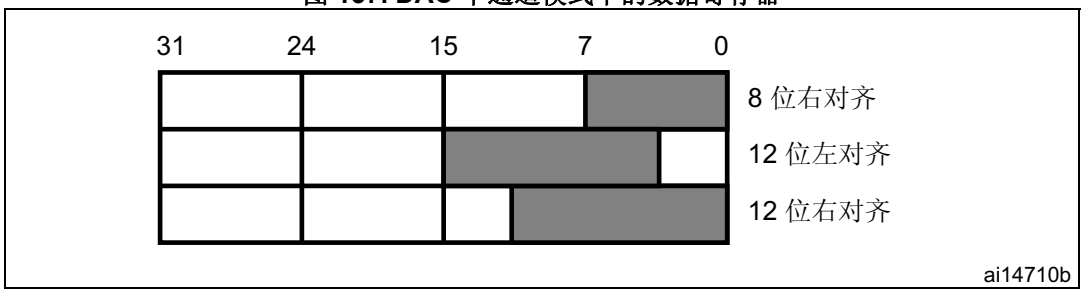
22.4.3 DAC 数据格式

根据所选配置模式，数据必须按如下方式写入指定寄存器：

- 对于 DAC 单通道 x，有三种可能的方式：
  - 8 位右对齐：软件必须将数据加载到 DAC_DHR8Ry[7:0] 位（存储到 DHRy[11:4] 位）
  - 12 位左对齐：软件必须将数据加载到 DAC_DHR12Ly [15:4] 位（存储到 DHRy[11:0] 位）
  - 12 位右对齐：软件必须将数据加载到 DAC_DHR12Ry [11:0] 位（存储到 DHRx[11:0] 位）

根据加载的 DAC_DHRyyyw 寄存器，用户写入的数据将移位并存储到相应的 DHRx（数据保持寄存器 x，即内部非存储器映射寄存器）。之后，DHRx 寄存器将被自动加载，或者通过软件或外部事件触发加载到 DORx 寄存器。

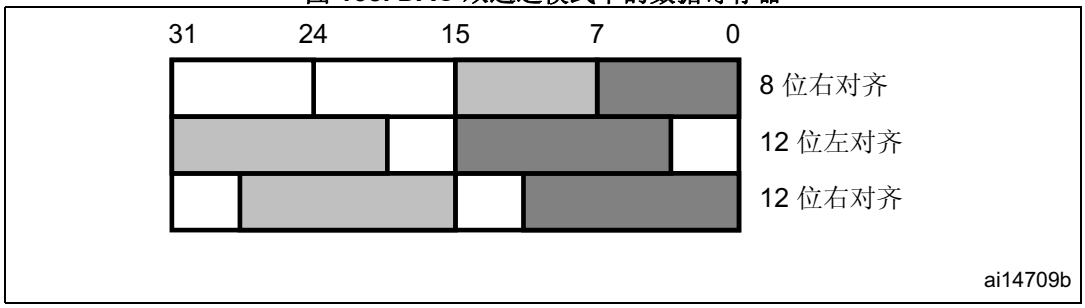
图 137. DAC 单通道模式下的数据寄存器



- 对于 DAC 双通道（可用时），有三种可能的方式：
  - 8 位右对齐：将 DAC 1 通道的数据加载到 DAC_DHR8RD [7:0] 位（存储到 DHR1[11:4] 位），将 DAC 2 通道的数据加载到 DAC_DHR8RD [15:8] 位（存储到 DHR2[11:4] 位）
  - 12 位左对齐：将 DAC 1 通道的数据加载到 DAC_DHR12RD [15:4] 位（存储到 DHR1[11:0] 位），将 DAC 2 通道的数据加载到 DAC_DHR12RD [31:20] 位（存储到 DHR2[11:0] 位）
  - 12 位右对齐：将 DAC 1 通道的数据加载到 DAC_DHR12RD [11:0] 位（存储到 DHR1[11:0] 位），将 DAC 2 通道的数据加载到 DAC_DHR12RD [27:16] 位（存储到 DHR2[11:0] 位）

根据加载的 DAC_DHRyyyD 寄存器，用户写入的数据将移位并存储到 DHR1 和 DHR2（数据保持寄存器，即内部非存储器映射寄存器）。之后，DHR1 和 DHR2 寄存器将被自动加载，或者通过软件或外部事件触发分别被加载到 DOR1 和 DOR2 寄存器。

图 138. DAC 双通道模式下的数据寄存器



22.4.4 DAC 转换

DAC_DORy 无法直接写入，任何数据都必须通过加载 DAC_DHRy 寄存器（写入 DAC_DHR8Ry、DAC_DHR12Ly、DAC_DHR12Ry、DAC_DHR8RD、DAC_DHR12RD 或 DAC_DHR12LD）才能传输到 DAC 通道 x。

如果未选择硬件触发（DAC_CR 寄存器中的 TENx 位复位），那么经过一个 APB1 时钟周期后，DAC_DHRy 寄存器中存储的数据将自动转移到 DAC_DORy 寄存器。但是，如果选择硬件触发（DAC_CR 寄存器中的 TENx 位置 1）且触发条件到来，将在触发信号后经过三个 APB1 时钟周期进行转移。

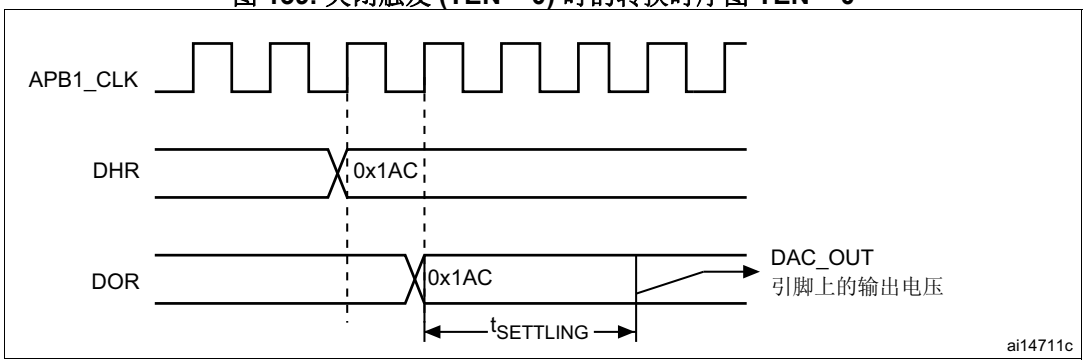
当 DAC_DORy 加载了 DAC_DHRy 内容时，模拟输出电压将在一段时间 $t_{SETTLING}$ 后可用，具体时间取决于电源电压和模拟输出负载。

当 APB1 时钟速度大于 80 MHz 时，DAC_CR 的 HFSEL 位必须置 1。从 DAC_DHRy 寄存器传输到 DAC_DORy 寄存器会增加三个 APB1 时钟周期的额外延时 ($t_{SETTLING}$)。

DAC_DORy 更新频率限制为 APB1 时钟频率的 1/3。当 HFSEL 位置 1 时，该频率限制为 APB1 时钟频率的 1/8。

如果 HFSEL 置 1，则在 ENx 位置 1 后的八个时钟周期内不允许对 DHRx 寄存器进行写入操作。在此期间，也不允许进行软件/硬件触发。

图 139. 关闭触发 (TEN = 0) 时的转换时序图 TEN = 0



22.4.5 DAC 输出电压

经过线性转换后，数字输入会转换为 0 到 V_{REF+} 之间的输出电压。

各 DAC 通道引脚的模拟输出电压通过以下公式确定：

$$\text{DAC 输出} = V_{REF} \times \frac{DOR}{4096}$$

22.4.6 DAC 触发选择

如果 TENx 控制位置 1，可通过外部事件（定时计数器、外部中断线）触发转换。由 TSELx[2:0] 控制位确定 8 个可能的事件中将触发转换的事件，如*第 22.7.1 节：DAC 控制寄存器 (DAC_CR)* 中的 TSEL1[2:0] 和 TSEL2[2:0] 位所示。

每当 DAC 接口在所选触发源上检测到上升沿时（请参见下表），DAC_DHRy 寄存器中存储的最后一个数据即会传输到 DAC_DORy 寄存器中。发生触发后再经过三个 APB1 周期，DAC_DORy 寄存器将会得到更新。

如果选择软件触发，一旦 SWTRIG 位置 1，转换即会开始。DAC_DHRy 寄存器内容加载到 DAC_DORy 寄存器中后，SWTRIG 即由硬件复位。

注： *ENx 位置 1 时，无法更改 TSELx[2:0] 位。*

如果选择软件触发，DAC_DHRy 寄存器的内容只需一个 APB1 时钟周期即可转移到 DAC_DORy 寄存器。

表 129. DAC 触发选择

源	类型	TSELx[3:0]
SWTRIG	软件控制位	0000
TIM1_TRGO	片上定时器的内部信号	0001
TIM2_TRGO	片上定时器的内部信号	0010
TIM4_TRGO	片上定时器的内部信号	0011
TIM5_TRGO	片上定时器的内部信号	0100
TIM6_TRGO	片上定时器的内部信号	0101
TIM7_TRGO	片上定时器的内部信号	0110
TIM8_TRGO	片上定时器的内部信号	0111
TIM15_TRGO	片上定时器的内部信号	1000
保留	-	1001
保留	-	1010
LPTIM1_OUT	片上定时器的内部信号	1011
LPTIM2_OUT	片上定时器的内部信号	1100
EXTI9	外部引脚	1101
保留	-	1110
保留	-	1111

22.4.7 DMA 请求

每个 DAC 通道都具有 DMA 功能。两个 DMA 通道用于处理 DAC 通道的 DMA 请求。

当 DMAENx 位置 1 时，如果发生外部触发（而不是软件触发），则传输结束后 DAC_DHRy 寄存器的值会转移到 DAC_DORy 寄存器，并且会产生 DMA 请求。

在双通道模式下，如果两个 DMAENx 位均置 1，则将产生两个 DMA 请求。如果只需要一个 DMA 请求，应仅将相应 DMAENx 位置 1。这样，应用程序可以在双通道模式下通过一个 DMA 请求和一个特定 DMA 通道来管理两个 DAC 通道。

由于 DAC_DHRy 已在 DMA 请求之前向 DAC_DORy 传输数据，必须在发生第一个触发事件之前将第一个数据写入 DAC_DHRy。

DMA 下溢

DAC DMA 请求没有缓冲队列。这样，如果第二个外部触发到达时尚未收到第一个外部触发的应答，将不会发出新的请求，并且 DAC_SR 寄存器中的 DAM 通道下溢标志 DMAUDRx 将置 1，以报告这一错误状况。DAC 通道仍将继续转换旧有数据。

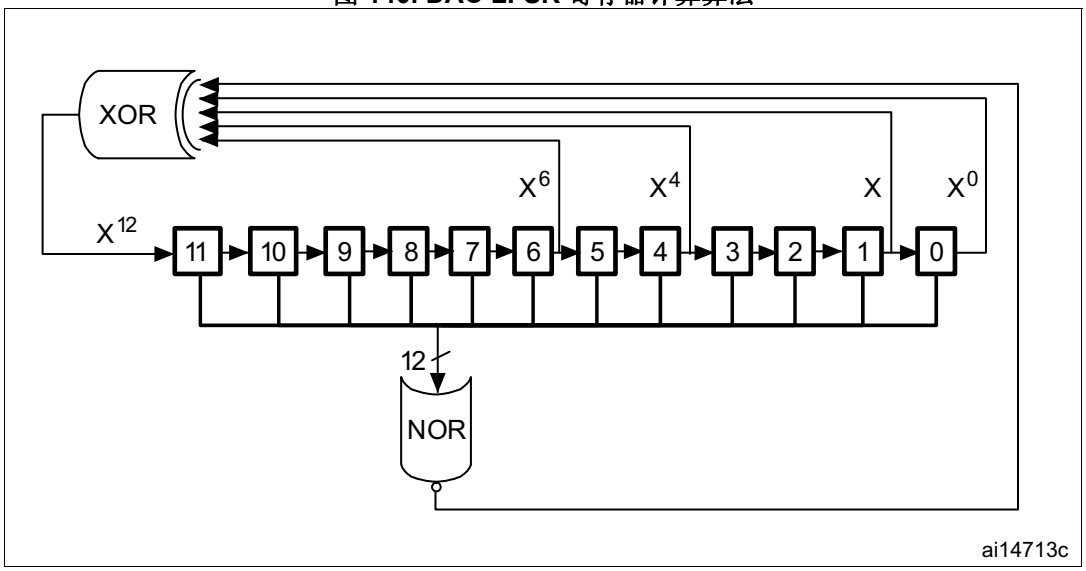
软件应通过写入 1 来将 DMAUDRx 标志清零，将所用 DMA 数据流的 DMAEN 位清零，并重新初始化 DMA 和 DAC 通道 x，以便正确地重新开始传输。软件应修改 DAC 触发转换频率或减轻 DMA 工作负载，以避免再次发生 DMA 下溢。最后，可通过使能 DMA 数据传输和转换触发来继续完成 DAC 转换。

对于各 DAC 通道，如果使能 DAC_CR 寄存器中相应的 DMAUDRIEx 位，还将产生中断。

22.4.8 生成噪声

为了生成可变振幅的伪噪声，可使用 LFSR（线性反馈移位寄存器）。将 WAVEx[1:0] 置为“01”即可选择生成 DAC 噪声。LFSR 中的预加载值为 0xAAA。在每次发生触发事件后，经过三个 APB1 时钟周期，该寄存器会依照特定的计算算法完成更新。

图 140. DAC LFSR 寄存器计算算法

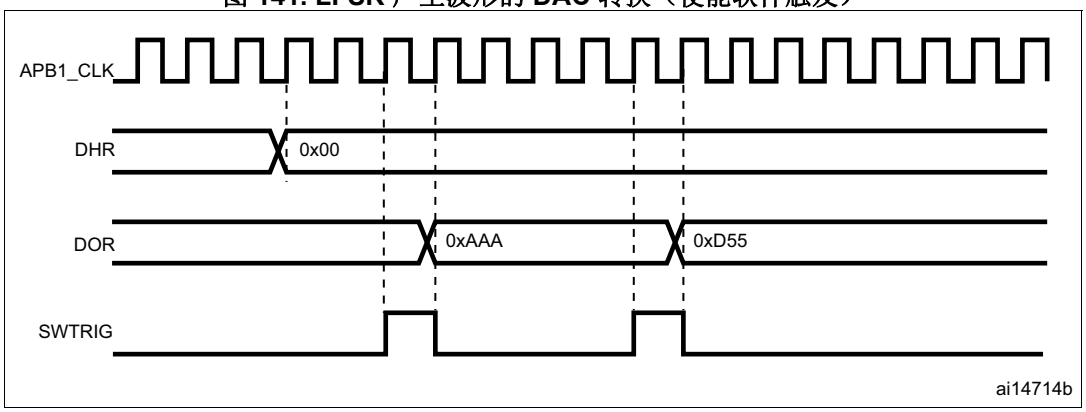


LFSR 值可以通过 DAC_CR 寄存器中的 MAMPx[3:0] 位来部分或完全屏蔽，在不发生溢出的情况下，该值将与 DAC_DHRy 的内容相加，然后存储到 DAC_DORy 寄存器中。

如果 LFSR 为 0x0000，将向其注入“1”（防锁定机制）。

可以通过复位 WAVEx[1:0] 位来将 LFSR 波形产生功能关闭。

图 141. LFSR 产生波形的 DAC 转换（使能软件触发）



注： *要生成噪声，必须通过将 DAC_CR 寄存器中的 TENx 位置 1 来使能 DAC 触发。*

22.4.9 生成三角波

可以在直流电流或慢变信号上叠加一个小幅三角波。将 WAVEx[1:0] 置为“10”即可选择生成 DAC 三角波。振幅通过 DAC_CR 寄存器中的 MAMPx[3:0] 位进行配置。每次发生触发事件后，经过三个 APB1 时钟周期，内部三角波计数器将会递增。在不发生溢出的情况下，该计数器的值将与 DAC_DHRy 寄存器内容相加，所得总和将传输到 DAC_DORy 寄存器中。只要小于 MAMPx[3:0] 位定义的最大振幅，三角波计数器就会一直递增。一旦达到配置的振幅，计数器将递减至零，然后再递增，以此类推。

可以通过复位 WAVEx[1:0] 位来将三角波产生功能关闭。

图 142. 生成 DAC 三角波

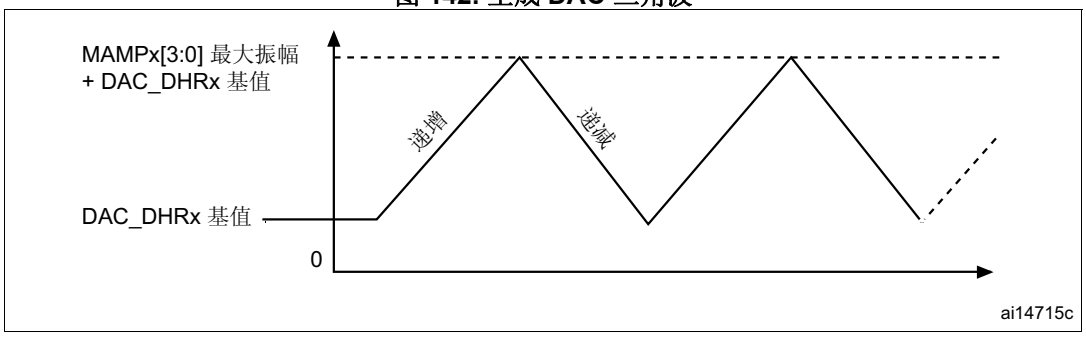
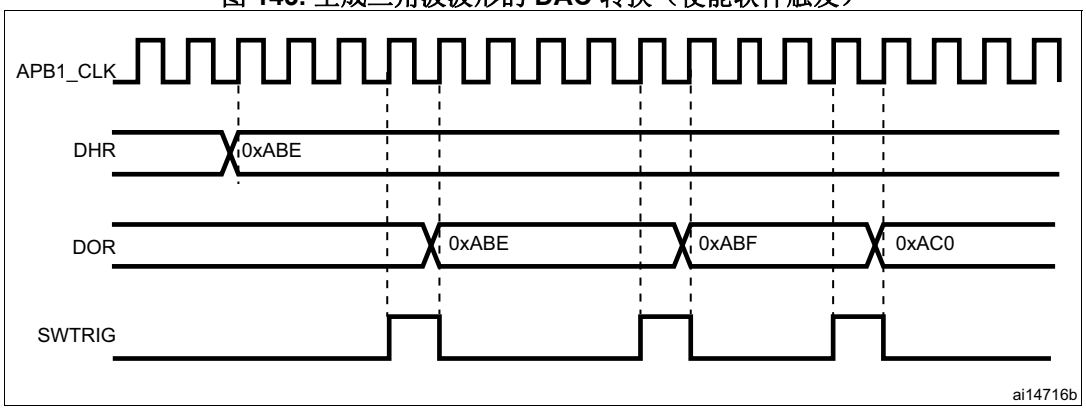


图 143. 生成三角波波形的 DAC 转换（使能软件触发）



注： *要生成三角波，必须通过将 DAC_CR 寄存器中的 TENx 位置 1 来使能 DAC 触发。MAMPx[3:0] 位必须在使能 DAC 之前进行配置，否则将无法更改。*

22.4.10 DAC 通道模式

每条 DAC 通道均可配置为正常模式或采样和保持模式。可使能输出缓冲器，以实现高驱动能力。使能输出缓冲器之前，需要校准电压偏移。此项校准是在出厂时执行的（复位后加载），可在应用运行期间通过软件进行调整。

正常模式

在正常模式下，可通过更改缓冲器状态以及更改 DAC_OUTx 引脚互连实现四种组合。

要使能输出缓冲器，DAC_MCR 寄存器中的 MODEx[2:0] 位应满足：

- 000：DAC 连接到外部引脚
- 001：DAC 连接到外部引脚以及片上外设

要禁止输出缓冲器，DAC_MCR 寄存器中的 MODEx[2:0] 位应满足：

- 010：DAC 连接到外部引脚
- 011：DAC 连接到片上外设

采样和保持模式

在采样和保持模式下，DAC 内核在触发转换后对数据进行转换，然后使电容上保持转换后的电压。不进行转换时，DAC 内核和缓冲器在两次采样之间是完全关闭的，DAC 输出处于三态，因此可降低整体功耗。每次新转换之前都需要一个新的稳定周期（$T_{stab-BON}$ 或 $T_{stab-BOFF}$，具体视缓冲器状态而定）。

在该模式下，DAC 内核以及所有相应的逻辑和寄存器除了受 APB1 时钟驱动外，还受低速时钟 (LSI) 驱动，从而可在深度低功耗模式（如停止模式）下使用 DAC 通道。

采样/保持模式运行可分为 3 个阶段：

1. 采样阶段：采样/保持元件充电到所需电压。充电时间取决于电容值（内部还是外部，由用户选择）。采样时间是通过 DAC_SHSRy 寄存器中的 TSAMx[9:0] 位配置的。对 TSAMx[9:0] 位执行写操作期间，DAC_SR 寄存器中的 BWSTx 位会置 1，从而可在两个时钟域（APB 和低速时钟）之间实现同步，并允许软件在 DAC 通道运行期间更改采样阶段的值
2. 保持阶段：DAC 输出通道处于三态，DAC 内核和缓冲器关闭，以降低电流消耗。保持时间是通过 DAC_SHHR 寄存器中的 THOLDx[9:0] 位配置的。
3. 刷新阶段：刷新时间是通过 DAC_SHRR 寄存器中的 TREFx[7:0] 位配置的。

以上三个阶段的时间以 LSI 时钟为单位。举例来说，假设选择了 LSI ~32 KHz，要配置 350 µs 的采样时间，2 ms 的保持时间和 100 µs 的刷新时间：

采样阶段需要 12 个周期：SAMx[9:0] = 11

保持阶段需要 62 个周期：THOLDx[9:0] = 62

刷新阶段需要 4 个周期：TREFx[7:0] = 4

本例中，与正常模式相比，功耗几乎降低了 15 倍。

下表列出了计算正确的采样时间和刷新时间的公式，保持时间取决于漏电流。

表 130. 采样时间和刷新时间

缓冲器状态	$t_{sampling}$ (1)(3)	$t_{refresh}$ (2)(3)
启用	$T_{stab-BON} + (10*R_{BON}*C_{load})$	$T_{stab-BON} + (R_{BON}*C_{load})*ln(2*Nlsb)$
禁用	$T_{stab-BOFF} + (10*R_{BOFF}*Cload)$	$T_{stab-BOFF} + (R_{BOFF}*Cload)*ln(2*Nlsb)$

注： *上述公式中，要实现 12 位分辨率，稳定为理想代码值（½ LSB 或精度）需要 10 个时间常数。要实现 8 位分辨率，稳定时间为 7 个时间常数。*

*电容以输出的漏电流放电后，保持阶段的容许压降“Vd”以 LSB 数表示。稳定回理想值（½ LSB 误差精度）需要 ln(2*Nlsb) 个 DAC 时间常数。*

参数 $T_{stab-BON}$、$T_{stab-BOFF}$、R_{BON} 和 R_{BOFF} 在数据手册中作出了说明。

输出缓冲器开启时的采样和刷新时间计算示例

注： *下例中使用的数值仅用于说明目的。有关产品数据，请参见产品数据手册。*

C_{load} = 100 nF

V_{DDA} = 3.0 V

采样阶段：

$t_{sampling}$ = 7 µs + (10 * 2000 * 100 * 10^{-9}) = 2.007 ms

（其中，$T_{stab-BON}$ = 7 µs，RBON = 2 kΩ）

刷新阶段：

$t_{refresh}$ = 7 µs + (2000 * 100 * 10^{-9}) * ln(2*10) = 606.1 µs

（其中，N_{lsb} = 10，保持阶段会减少 10 个 LSB）

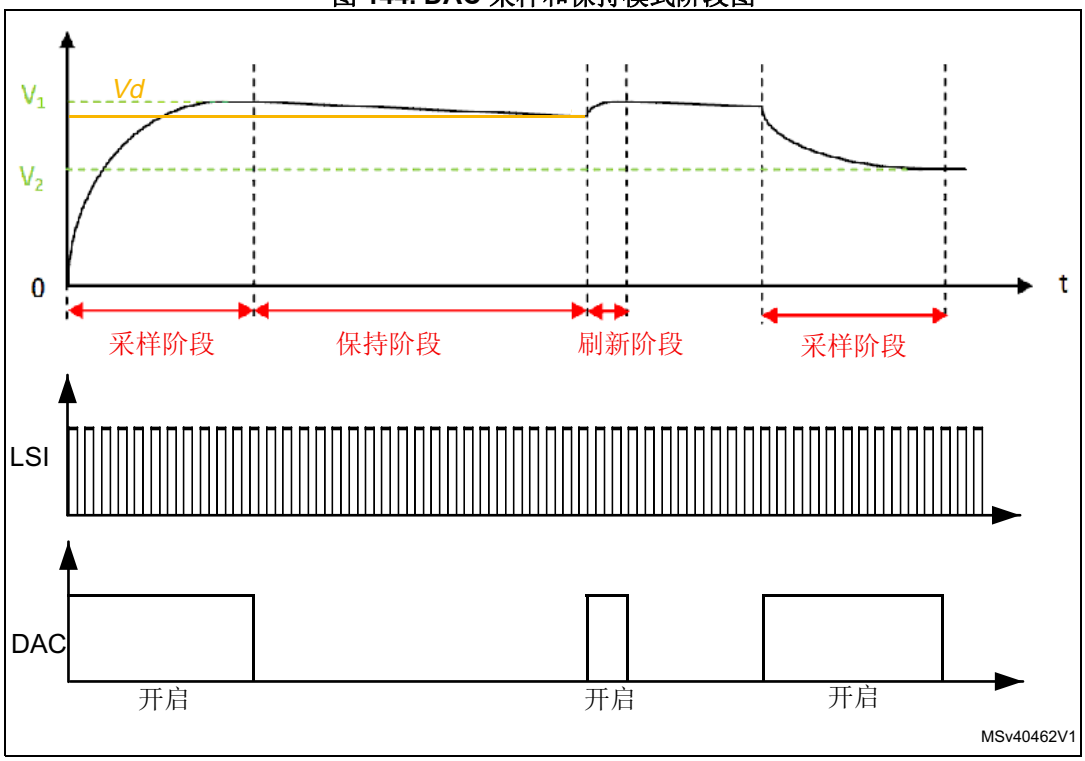
保持阶段：

D_v = i_{leak} * t_{hold} / C_{load} = 0.0073 V（3 V 时 10 个 LSB 在 12 位分辨率下）

i_{leak} = 150 nA（所有温度范围内 IO 泄漏的最坏情况）

t_{hold} = 0.0073 * 100 * 10^{-9} / (150 * 10^{-9}) = 4.867 ms

图 144. DAC 采样和保持模式阶段图



与正常模式相似，采样和保持模式也具有不同配置。

要使能输出缓冲器，DAC_MCR 寄存器中的 MODEx[2:0] 位应满足：

- 100：DAC 连接到外部引脚
- 101：DAC 连接到外部引脚以及片上外设

要禁止输出缓冲器，DAC_MCR 寄存器中的 MODEx[2:0] 位应满足：

- 110：DAC 连接到外部引脚以及片上外设
- 111：DAC 连接到片上外设

DAC_MCR 寄存器中的 MODEx[2:0] 位等于 111 时。内部电容“$C_{loadint}$”将保持 DAC 内核的电压输出，随后会将该电压驱动到片上外设。

所有采样和保持阶段均可中断，DAC_DHRy 中的任何更改都会立即触发新的采样阶段。

表 131. 通道输出模式汇总

MODEx[2:0]			模式	缓冲器	输出连接
0	0	0	正常模式	启用	连接到外部引脚
0	0	1			连接到外部引脚以及片上外设（例如，比较器）
0	1	0		输出关闭	连接到外部引脚
0	1	1			连接到片上外设（例如，比较器）
1	0	0	采样和保持模式	启用	连接到外部引脚
1	0	1			连接到外部引脚以及片上外设（例如，比较器）
1	1	0		输出关闭	连接到外部引脚以及片上外设（例如，比较器）
1	1	1			连接到片上外设（例如，比较器）

22.4.11 DAC 通道缓冲器校准

N 位数模转换器 (DAC) 的传递函数为：

$$V_{out} = ((D / 2^{N-1}) \times G \times V_{ref}) + V_{OS}$$

其中，V_{OUT} 为模拟输出，D 为数字输入，G 为增益，V_{ref} 为标称满量程电压，V_{os} 为偏移电压。对于理想的 DAC 通道，G = 1 且 V_{os} = 0。

由于输出缓冲器所具有的特性，不同芯片的电压偏移可能有所不同，并会在模拟输出上引入绝对偏移误差。为补偿 V_{os}，需要通过调整技术进行校准。

仅当 DAC 通道 x 在缓冲器使能的条件下运行时，校准才有效（MODEx[2:0] = 000b、001b、100b 或 101b）。如果在缓冲器关闭的条件下应用于其他模式，校准无效。校准过程中：

- 缓冲器输出将与引脚内部/外部连接断开，并会进入三态模式 (HiZ)。
- 缓冲器将作为比较器来检测中间代码值 0x800，并通过内部桥将其与 VREF+/2 信号进行比较，然后根据比较结果（CAL_FLAGx 位）将其输出信号切换为 0 或 1。

提供以下两种校准技术：

- 出厂调整（始终使能）

  DAC 缓冲器偏移是在出厂时进行调整的。DAC_CCR 寄存器中 OTRIMx[4:0] 位的默认值是出厂调整值，会在 DAC 数字接口复位时载入。

- 用户调整

  如果运行条件不同于标称出厂调整条件，特别是 V_{DD}/V_{DDA} 电压、温度、VREF+ 值发生变化时，可进行用户调整，并且用户可在应用过程中随时通过软件进行调整。

注： *更多关于标称出厂调整条件的详细信息，请参见数据手册。*

另外，如果 V_{DD}/V_{DDA} 移除（例如器件进入 STANDBY 或 VBAT 模式），则需要进行校准。

执行用户调整校准的步骤如下：

1. 如果 DAC 通道激活，可向 DAC_CR 中的 ENx 位写入 0 来禁止通道
2. 写入 DAC_MCR 寄存器，选择使能了缓冲器的模式（MODEx[2:0] = 000b、001b、100b 或 101b）
3. 将 DAC_CR 寄存器中的 CENx 位置 1，开始进行 DAC 通道 x 校准
4. 应用调整算法：
   a) 向 OTRIMx[4:0] 位写入以 00000b 开头的代码
   b) 等待 $t_{OFFTRIMmax}$ 延时
   c) 检查 DAC_SR 中的 CAL_FLAGx 位是否已置 1
   d) 如果 CAL_FLAGx 已置 1，调整代码 OTRIMx[4:0] 已找到，并将在运行过程中用来补偿输出值，否则会使 OTRIMx[4:0] 递增并重复执行 (a) 到 (d) 的步骤

软件算法可使用逐次逼近法或二分法较快地计算和设置 OTRIMx[4:0] 位的内容。

CAL_FLAGx 位的换向/切换指示偏移已正确补偿，相应的调整代码必须保留在 DAC_CCR 寄存器的 OTRIMx[4:0] 位中。

注： *为了获得正确的值，对 DAC_SR 寄存器中的 OTRIMx[4:0] 位进行的写操作与对 CAL_FLAGx 位进行的读操作之间必须保留 $t_{OFFTRIMmax}$ 延时 。此参数在数据手册的电气特性部分进行了说明。*

如果 V_{DD}/V_{DDA}、VREF+ 和温度条件在器件运行期间不发生更改，同时器件更为频繁地进入待机和 VBAT 模式，软件可将第一次用户校准时找到的 OTRIMx[4:0] 位存储在 flash 或备用寄存器中。然后在器件电源再次恢复时直接加载/写入这些位，从而无需等待新的校准时间。

当 CENx 位置 1 时，不允许将 ENx 位置 1。

22.4.12 DAC 双通道转换（如果可用）

为了在同时需要两个 DAC 通道的应用中有效利用总线带宽，DAC 模块有三个双寄存器可供操作：DHR8RD、DHR12RD 和 DHR12LD。这样，只需一个寄存器访问即可同时驱动两个 DAC 通道。要生成相应波形，无需访问 DHRxxxD 寄存器。因此，可单独也可同时使用两个输出通道。

通过两个 DAC 通道和这三个双寄存器可以实现 11 种转换模式。但如果需要，所有这些转换模式也都可以通过单独的 DHRx 寄存器来实现。

下面几段内容将介绍所有这些模式。

独立触发（不产生波形）

要将 DAC 配置为此转换模式，需要遵循以下顺序：

1. 将两个 DAC 通道触发使能位 TEN1 和 TEN2 置 1。
2. 将 TSEL1[2:0] 和 TSEL2[2:0] 设置为不同的值，以配置不同的触发源。
3. 将 DAC 双通道数据加载到所需 DHR 寄存器（DAC_DHR12RD、DAC_DHR12LD 或 DAC_DHR8RD）。

DAC 通道 1 触发信号到达时，DHR1 寄存器的内容转移到 DAC_DOR1（三个 APB1 时钟周期之后）。

DAC 通道 2 触发信号到达时，DHR2 寄存器的内容转移到 DAC_DOR2（三个 APB1 时钟周期之后）。

独立触发（生成单个 LFSR）

要将 DAC 配置为此转换模式，需要遵循以下顺序：

1. 将两个 DAC 通道触发使能位 TEN1 和 TEN2 置 1。
2. 将 TSEL1[2:0] 和 TSEL2[2:0] 设置为不同的值，以配置不同的触发源。
3. 将两个 DAC 通道的 WAVEx[1:0] 设置为 01，并在 MAMPx[3:0] 位中配置相同的 LFSR 掩码值。
4. 将 DAC 双通道数据加载到所需 DHR 寄存器（DHR12RD、DHR12LD 或 DHR8RD）。

DAC 通道 1 触发信号到达时，LFSR1 计数器内容（使用相同的掩码）与 DHR1 寄存器内容相加，所得总和转移到 DAC_DOR1 中（三个 APB1 时钟周期之后）。LFSR1 计数器随即更新。

DAC 通道 2 触发信号到达时，LFSR2 计数器内容（使用相同的掩码）与 DHR2 寄存器内容相加，所得总和转移到 DAC_DOR2 中（三个 APB1 时钟周期之后）。LFSR2 计数器随即更新。

独立触发（生成不同 LFSR）

要将 DAC 配置为此转换模式，需要遵循以下顺序：

1. 将两个 DAC 通道触发使能位 TEN1 和 TEN2 置 1。
2. 将 TSEL1[2:0] 和 TSEL2[2:0] 设置为不同的值，以配置不同的触发源。
3. 将两个 DAC 通道的 WAVEx[1:0] 设置为 01，并在 MAMP1[3:0] 和 MAMP2[3:0] 位中设置不同的 LFSR 掩码值。
4. 将 DAC 双通道数据加载到所需 DHR 寄存器（DAC_DHR12RD、DAC_DHR12LD 或 DAC_DHR8RD）。

DAC 通道 1 触发信号到达时，LFSR1 计数器内容（使用 MAMP1[3:0] 配置的掩码）与 DHR1 寄存器内容相加，所得总和转移到 DAC_DOR1 中（三个 APB1 时钟周期之后）。LFSR1 计数器随即更新。

DAC 通道 2 触发信号到达时，LFSR2 计数器内容（使用 MAMP2[3:0] 配置的掩码）与 DHR2 寄存器内容相加，所得总和转移到 DAC_DOR2 中（三个 APB1 时钟周期之后）。LFSR2 计数器随即更新。

独立触发（生成单个三角波）

要将 DAC 配置为此转换模式，需要遵循以下顺序：

1. 将两个 DAC 通道触发使能位 TEN1 和 TEN2 置 1。
2. 将 TSEL1[2:0] 和 TSEL2[2:0] 设置为不同的值，以配置不同的触发源。
3. 将两个 DAC 通道的 WAVEx[1:0] 设置为 1x，并在 MAMPx[3:0] 位中配置相同的最大振幅值。
4. 将 DAC 双通道数据加载到所需 DHR 寄存器（DAC_DHR12RD、DAC_DHR12LD 或 DAC_DHR8RD）。

DAC 通道 1 触发信号到达时，DAC 通道 1 三角波计数器内容（使用相同的三角波振幅）与 DHR1 寄存器内容相加，所得总和转移到 DAC_DOR1 中（三个 APB1 时钟周期之后）。DAC 1 通道 三角波计数器随即更新。

DAC 通道 2 触发信号到达时，DAC 通道 2 三角波计数器内容（使用相同的三角波振幅）与 DHR2 寄存器内容相加，所得总和转移到 DAC_DOR2 中（三个 APB1 时钟周期之后）。DAC 2 通道 三角波计数器随即更新。

独立触发（生成不同三角波）

要将 DAC 配置为此转换模式，需要遵循以下顺序：

1. 将两个 DAC 通道触发使能位 TEN1 和 TEN2 置 1。
2. 将 TSEL1[2:0] 和 TSEL2[2:0] 设置为不同的值，以配置不同的触发源。
3. 将两个 DAC 通道的 WAVEx[1:0] 设置为 1x，并在 MAMP1[3:0] 和 MAMP2[3:0] 位中设置不同的最大振幅值。
4. 将 DAC 双通道数据加载到所需 DHR 寄存器（DAC_DHR12RD、DAC_DHR12LD 或 DAC_DHR8RD）。

DAC 通道 1 触发信号到达时，DAC 通道 1 三角波计数器内容（使用 MAMP1[3:0] 配置的三角波振幅）与 DHR1 寄存器内容相加，所得总和转移到 DAC_DOR1 中（三个 APB1 时钟周期之后）。DAC 1 通道 三角波计数器随即更新。

DAC 通道 2 触发信号到达时，DAC 通道 2 三角波计数器内容（使用 MAMP2[3:0] 配置的三角波振幅）与 DHR2 寄存器内容相加，所得总和转移到 DAC_DOR2 中（三个 APB1 时钟周期之后）。DAC 2 通道 三角波计数器随即更新。

同步软件启动

要将 DAC 配置为此转换模式，需要遵循以下顺序：

- 将 DAC 双通道数据加载到所需 DHR 寄存器（DAC_DHR12RD、DAC_DHR12LD 或 DAC_DHR8RD）

在此配置中，DHR1 和 DHR2 寄存器内容会在一个 APB1 时钟周期后分别转移到 DAC_DOR1 和 DAC_DOR2 中。

同步触发（不产生波形）

要将 DAC 配置为此转换模式，需要遵循以下顺序：

- 将两个 DAC 通道触发使能位 TEN1 和 TEN2 置 1
- 将 TSEL1[2:0] 和 TSEL2[2:0] 设置为相同的值，以便为两个 DAC 通道配置相同的触发源
- 将 DAC 双通道数据加载到所需 DHR 寄存器（DAC_DHR12RD、DAC_DHR12LD 或 DAC_DHR8RD）

当触发信号到达时，DHR1 和 DHR2 寄存器内容将分别转移到 DAC_DOR1 和 DAC_DOR2 中（三个 APB1 时钟周期之后）。

同步触发（生成单个 LFSR）

要将 DAC 配置为此转换模式，需要遵循以下顺序：

- 将两个 DAC 通道触发使能位 TEN1 和 TEN2 置 1
- 将 TSEL1[2:0] 和 TSEL2[2:0] 设置为相同的值，以便为两个 DAC 通道配置相同的触发源
- 将两个 DAC 通道的 WAVEx[1:0] 设置为 01，并在 MAMPx[3:0] 位中配置相同的 LFSR 掩码值
- 将 DAC 双通道数据加载到所需 DHR 寄存器（DHR12RD、DHR12LD 或 DHR8RD）

触发信号到达时， LFSR1 计数器内容 （使用相同的掩码）与 DHR1 寄存器内容相加，所得总和转移到 DAC_DOR1 中 （三个 APB1 时钟周期之后）。 LFSR1 计数器随即更新。同时，LFSR2 计数器内容（使用相同的掩码）与 DHR2 寄存器内容相加，所得总和转移到 DAC_DOR2 中 （三个 APB1 时钟周期之后）。 LFSR2 计数器随即更新。

同步触发（生成不同 LFSR）

要将 DAC 配置为此转换模式，需要遵循以下顺序：

- 将两个 DAC 通道触发使能位 TEN1 和 TEN2 置 1
- 将 TSEL1[2:0] 和 TSEL2[2:0] 设置为相同的值，以便为两个 DAC 通道配置相同的触发源
- 将两个 DAC 通道的 WAVEx[1:0] 设置为 01，并在 MAMP1[3:0] 和 MAMP2[3:0] 位中设置不同的 LFSR 掩码值
- 将 DAC 双通道数据加载到所需 DHR 寄存器（DAC_DHR12RD、DAC_DHR12LD 或 DAC_DHR8RD）

触发信号到达时，LFSR1 计数器内容（使用 MAMP1[3:0] 配置的掩码）与 DHR1 寄存器内容相加，所得总和转移到 DAC_DOR1 中（三个 APB1 时钟周期之后）。LFSR1 计数器随即更新。
同时，LFSR2 计数器内容（使用 MAMP2[3:0] 配置的掩码）与 DHR2 寄存器内容相加，所得总和转移到 DAC_DOR2 中（三个 APB1 时钟周期之后）。LFSR2 计数器随即更新。

同步触发（生成单个三角波）

要将 DAC 配置为此转换模式，需要遵循以下顺序：

- 将两个 DAC 通道触发使能位 TEN1 和 TEN2 置 1
- 将 TSEL1[2:0] 和 TSEL2[2:0] 设置为相同的值，以便为两个 DAC 通道配置相同的触发源
- 将两个 DAC 通道的 WAVEx[1:0] 设置为 1x，并在 MAMPx[3:0] 位中配置相同的最大振幅值
- 将 DAC 双通道数据加载到所需 DHR 寄存器（DAC_DHR12RD、DAC_DHR12LD 或 DAC_DHR8RD）

触发信号到达时，DAC 通道 1 三角波计数器内容（使用相同的三角波振幅）与 DHR1 寄存器内容相加，所得总和转移到 DAC_DOR1 中（三个 APB1 时钟周期之后）。DAC 1 通道三角波计数器随即更新。

同时，DAC 通道 2 三角波计数器内容（使用相同的三角波振幅）与 DHR2 寄存器内容相加，所得总和转移到 DAC_DOR2 中（三个 APB1 时钟周期之后）。DAC 2 通道 三角波计数器随即更新。

同步触发（生成不同三角波）

要将 DAC 配置为此转换模式，需要遵循以下顺序：

- 将两个 DAC 通道触发使能位 TEN1 和 TEN2 置 1
- 将 TSEL1[2:0] 和 TSEL2[2:0] 设置为相同的值，以便为两个 DAC 通道配置相同的触发源
- 将两个 DAC 通道的 WAVEx[1:0] 设置为 1x，并在 MAMP1[3:0] 和 MAMP2[3:0] 位中设置不同的最大振幅值
- 将 DAC 双通道数据加载到所需 DHR 寄存器（DAC_DHR12RD、DAC_DHR12LD 或 DAC_DHR8RD）

触发信号到达时，DAC 通道 1 三角波计数器内容（使用 MAMP1[3:0] 配置的三角波振幅）与 DHR1 寄存器内容相加，所得总和转移到 DAC_DOR1 中（三个 APB 时钟周期之后）。DAC 通道 1 三角波计数器随即更新。

同时，DAC 通道 2 三角波计数器内容（使用 MAMP2[3:0] 配置的三角波振幅）与 DHR2 寄存器内容相加，所得总和转移到 DAC_DOR2 中（三个 APB1 时钟周期之后）。DAC 通道 2 三角波计数器随即更新。

22.5 DAC 低功耗模式

表 132. DAC 低功耗模式的作用

模式	说明
睡眠	无影响，DAC 与 DMA 结合使用。
低功耗运行	无影响。
低功耗睡眠	无影响。DAC 与 DMA 结合使用。
停止 0/停止 1	如果通过 LSI 时钟选择了采样和保持模式，则 DAC 将以静态值继续工作。
停止 2	DAC 寄存器内容保留。必须在进入停止 2 模式之前禁止 DAC。
待机	DAC 外设掉电，退出待机模式或关断模式后必须重新初始化。
关断	

22.6 DAC 中断

表 133. DAC 中断

中断事件	事件标志	使能控制位
DMA 下溢	DMAUDRx	DMAUDRIEx

22.7 DAC 寄存器

有关寄存器说明中使用的缩写，请参见*第 72 页的第 1 节*。

外设寄存器必须按字（32 位）进行访问。

22.7.1 DAC 控制寄存器 (DAC_CR)

DAC control 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
Res.	CEN2	DMAUDRIE2	DMAEN2	MAMP2[3:0]				WAVE2[1:0]		TSEL22	TSEL21	TSEL20	TEN2	Res.	EN2
	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
HFSEL	CEN1	DMAUDRIE1	DMAEN1	MAMP1[3:0]				WAVE1[1:0]		TSEL12	TSEL11	TSEL10	TEN1	Res.	EN1
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw		rw

位 31 保留，必须保持复位值。

位 30 **CEN2**：DAC 通道 2 校准使能 (DAC Channel 2 calibration enable)

此位由软件置 1 和清零，用于使能/禁止 DAC 通道 2 校准，仅当 DAC_CR 中的位 EN2 设置为 0 时才可写入（仅当 DAC 通道已禁止时才可以进入/退出校准模式），否则会忽略写操作。

0：DAC 通道 2 处于正常运行模式

1：DAC 通道 2 处于校准模式

注： 此位仅适用于双通道 DAC。请参见第 22.3 节：DAC 实现。

位 29 **DMAUDRIE2**：DAC 2 通道 DMA 下溢中断使能 (DAC channel2 DMA underrun interrupt enable)

此位由软件置 1 和清零。

0：禁止 DAC 2 通道 DMA 下溢中断

1：使能 DAC 2 通道 DMA 下溢中断

注： 此位仅适用于双通道 DAC。请参见第 22.3 节：DAC 实现。

位 28 **DMAEN2**：DAC 2 通道 DMA 使能 (DAC channel2 DMA enable)

此位由软件置 1 和清零。

0：禁止 DAC 2 通道 DMA 模式

1：使能 DAC 2 通道 DMA 模式

注： 此位仅适用于双通道 DAC。请参见第 22.3 节：DAC 实现。

位 27:24 **MAMP2[3:0]**：DAC 2 通道掩码/振幅选择器 (DAC channel2 mask/amplitude selector)

这些位由软件写入，用于在生成噪声波模式下选择掩码，或者在生成三角波模式下选择振幅。

0000：不屏蔽 LFSR 的位 0/三角波振幅等于 1

0001：不屏蔽 LFSR 的位 [1:0]/三角波振幅等于 3

0010：不屏蔽 LFSR 的位 [2:0]/三角波振幅等于 7

0011：不屏蔽 LFSR 的位 [3:0]/三角波振幅等于 15

0100：不屏蔽 LFSR 的位 [4:0]/三角波振幅等于 31

0101：不屏蔽 LFSR 的位 [5:0]/三角波振幅等于 63

0110：不屏蔽 LFSR 的位 [6:0]/三角波振幅等于 127

0111：不屏蔽 LFSR 的位 [7:0]/三角波振幅等于 255

1000：不屏蔽 LFSR 的位 [8:0]/三角波振幅等于 511

1001：不屏蔽 LFSR 的位 [9:0]/三角波振幅等于 1023

1010：不屏蔽 LFSR 的位 [10:0]/三角波振幅等于 2047

≥ 1011：不屏蔽 LFSR 的位 [11:0]/三角波振幅等于 4095

注： 这些位仅适用于双通道 DAC。请参见第 22.3 节：DAC 实现。

位 23:22 **WAVE2[1:0]**：DAC 2 通道噪声/三角波生成使能 (DAC channel2 noise/triangle wave generation enable)

这些位由软件置 1 或清零。

00：禁止生成波

01：使能生成噪声波

1x：使能生成三角波

注： 只在位 TEN2 = 1（使能 DAC 2 通道触发）时使用

这些位仅适用于双通道 DAC。请参见第 22.3 节：DAC 实现。

位 21:19 **TSEL2[2:0]**：DAC 2 通道触发器选择 (DAC channel2 trigger selection)
这些位用于选择 DAC 2 通道的外部触发事件
有关触发配置和映射的详细信息，请参见触发选择表。
注： 只在位 TEN2 = 1（使能 DAC 2 通道触发）时使用。
这些位仅适用于双通道 DAC。请参见第 22.3 节：DAC 实现。

位 18 **TEN2**：DAC 2 通道触发使能 (DAC channel2 trigger enable)
此位由软件置 1 和清零，以使能/禁止 DAC 2 通道触发
0：禁止 DAC 2 通道触发，写入 DAC_DHR2 寄存器的数据在一个 APB1 时钟周期之后转移到 DAC_DOR2 寄存器
1：使能 DAC 2 通道触发，DAC_DHR2 寄存器的数据在三个 APB1 时钟周期之后转移到 DAC_DOR2 寄存器
注： 如果选择软件触发，DAC_DHR2 寄存器的内容只需一个 APB1 时钟周期即可转移到 DAC_DOR2 寄存器。
这些位仅适用于双通道 DAC。请参见第 22.3 节：DAC 实现。

位 17 保留，必须保持复位值。

位 16 **EN2**：DAC 2 通道使能 (DAC channel2 enable)
此位由软件置 1 和清零，以使能/禁止 DAC 2 通道。
0：禁止 DAC 2 通道
1：使能 DAC 2 通道
注： 这些位仅适用于双通道 DAC。请参见第 22.3 节：DAC 实现。

位 15 **HFSEL**：高频接口模式使能 (High frequency interface mode enable)
此位由软件置 1 和清零，用于使能/禁止 DAC 接口高速模式
当 APB1 时钟频率高于 80 MHz 时，该位需置 1。
0：禁止高频接口模式
1：使能高频接口模式

位 14 **CEN1**：DAC 通道 1 校准使能 (DAC Channel 1 calibration enable)
此位由软件置 1 和清零，用于使能/禁止 DAC 通道 1 校准，仅当 DAC_CR 中的位 EN1=0 时才可写入（仅当 DAC 通道已禁止时才可以进入/退出校准模式），否则会忽略写操作。
0：DAC 通道 1 处于正常运行模式
1：DAC 通道 1 处于校准模式

位 13 **DMAUDRIE1**：DAC 1 通道 DMA 下溢中断使能 (DAC channel1 DMA Underrun Interrupt enable)
此位由软件置 1 和清零。
0：禁止 DAC 1 通道 DMA 下溢中断
1：使能 DAC 1 通道 DMA 下溢中断

位 12 **DMAEN1**：DAC 1 通道 DMA 使能 (DAC channel1 DMA enable)
此位由软件置 1 和清零。
0：禁止 DAC 1 通道 DMA 模式
1：使能 DAC 1 通道 DMA 模式

位 11:8 **MAMP1[3:0]**：DAC 1 通道掩码/振幅选择器 (DAC channel1 mask/amplitude selector)

这些位由软件写入，用于在生成噪声波模式下选择掩码，或者在生成三角波模式下选择振幅。

0000：不屏蔽 LFSR 的位 0/三角波振幅等于 1
0001：不屏蔽 LFSR 的位 [1:0]/三角波振幅等于 3
0010：不屏蔽 LFSR 的位 [2:0]/三角波振幅等于 7
0011：不屏蔽 LFSR 的位 [3:0]/三角波振幅等于 15
0100：不屏蔽 LFSR 的位 [4:0]/三角波振幅等于 31
0101：不屏蔽 LFSR 的位 [5:0]/三角波振幅等于 63
0110：不屏蔽 LFSR 的位 [6:0]/三角波振幅等于 127
0111：不屏蔽 LFSR 的位 [7:0]/三角波振幅等于 255
1000：不屏蔽 LFSR 的位 [8:0]/三角波振幅等于 511
1001：不屏蔽 LFSR 的位 [9:0]/三角波振幅等于 1023
1010：不屏蔽 LFSR 的位 [10:0]/三角波振幅等于 2047
≥ 1011：不屏蔽 LFSR 的位 [11:0]/三角波振幅等于 4095

位 7:6 **WAVE1[1:0]**：DAC 1 通道噪声/三角波生成使能 (DAC channel1 noise/triangle wave generation enable)

这些位将由软件置 1 和清零。
00：禁止生成波
01：使能生成噪声波
1x：使能生成三角波
只在位 TEN1 = 1（使能 DAC 1 通道触发）时使用。

位 5:3 **TSEL1[2:0]**：DAC 1 通道触发器选择 (DAC channel1 trigger selection)

这些位用于选择 DAC 1 通道的外部触发事件。
有关触发配置和映射的详细信息，请参见触发选择表。

注：　只在位 TEN1 = 1（使能 DAC 1 通道触发）时使用。

位 2 **TEN1**：DAC 1 通道触发使能 (DAC channel1 trigger enable)

此位由软件置 1 和清零，以使能/禁止 DAC 1 通道触发。
0：禁止 DAC 1 通道触发，写入 DAC_DHR1 寄存器的数据在一个 APB1 时钟周期之后转移到 DAC_DOR1 寄存器
1：使能 DAC 1 通道触发，DAC_DHR1 寄存器的数据在三个 APB1 时钟周期之后转移到 DAC_DOR1 寄存器

*注：　如果选择软件触发，*DAC_DHR1 *寄存器的内容只需一个 APB1 时钟周期即可转移到 DAC_DOR1 寄存器。*

位 1 保留，必须保持复位值。

位 0 **EN1**：DAC 1 通道使能 (DAC channel1 enable)

此位由软件置 1 和清零，以使能/禁止 DAC 1 通道。
0：禁止 DAC 1 通道
1：使能 DAC 1 通道

22.7.2 DAC 软件触发寄存器 (DAC_SWTRGR)

DAC software trigger 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
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.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SWTRIG2	SWTRIG1
														w	w

位 31:2 保留，必须保持复位值。

位 1 **SWTRIG2**：DAC 2 通道软件触发 (DAC channel2 software trigger)

该位由软件置 1，用于在软件触发模式下触发 DAC。

0：不触发

1：触发

注： 一旦 DAC_DHR2 寄存器值加载到 DAC_DOR2 寄存器中，该位即会由硬件清零（一个 APB1 时钟周期之后）。

此位仅适用于双通道 DAC。请参见第 22.3 节：DAC 实现。

位 0 **SWTRIG1**：DAC 1 通道软件触发 (DAC channel1 software trigger)

该位由软件置 1，用于在软件触发模式下触发 DAC。

0：不触发

1：触发

注： 一旦 DAC_DHR1 寄存器值加载到 DAC_DOR1 寄存器中，该位即会由硬件清零（一个 APB1 时钟周期之后）。

22.7.3 DAC 1 通道 12 位右对齐数据保持寄存器 (DAC_DHR12R1)

DAC channel1 12-bit right-aligned data holding 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
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.	Res.	Res.	DACC1DHR[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 **DACC1DHR[11:0]**：DAC 1 通道 12 位右对齐数据 (DAC channel1 12-bit right-aligned data)

这些位通过软件写入，用于指定 DAC 1 通道的 12 位数据。

22.7.4 DAC 1 通道 12 位左对齐数据保持寄存器 (DAC_DHR12L1)

DAC channel1 12-bit left aligned data holding 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
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
DACC1DHR[11:0]												Res.	Res.	Res.	Res.
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 **DACC1DHR[11:0]**：DAC 1 通道 12 位左对齐数据 (DAC channel1 12-bit left-aligned data)
这些位通过软件写入。
用于指定 DAC 1 通道的 12 位数据。

位 3:0 保留，必须保持复位值。

22.7.5 DAC 1 通道 8 位右对齐数据保持寄存器 (DAC_DHR8R1)

DAC channel1 8-bit right aligned data holding 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
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.	Res.	Res.	Res.	Res.	Res.	Res.	DACC1DHR[7:0]							
								rw	rw	rw	rw	rw	rw	rw	rw

位 31:8 保留，必须保持复位值。

位 7:0 **DACC1DHR[7:0]**：DAC 1 通道 8 位右对齐数据 (DAC channel1 8-bit right-aligned data)
这些位通过软件写入，用于指定 DAC 1 通道的 8 位数据。

22.7.6 DAC 2 通道 12 位右对齐数据保持寄存器 (DAC_DHR12R2)

DAC channel2 12-bit right aligned data holding register

此寄存器仅适用于双通道 DAC。请参见*第 22.3 节：DAC 实现*。

偏移地址：0x14

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	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.	Res.	Res.	DACC2DHR[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 **DACC2DHR[11:0]**：DAC 2 通道 12 位右对齐数据 (DAC channel2 12-bit right-aligned data)
这些位通过软件写入，用于指定 DAC 2 通道的 12 位数据。

22.7.7 DAC 2 通道 12 位左对齐数据保持寄存器 (DAC_DHR12L2)

DAC channel2 12-bit left aligned data holding register

此寄存器仅适用于双通道 DAC。请参见*第 22.3 节：DAC 实现*。

偏移地址：0x18

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	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
DACC2DHR[11:0]												Res.	Res.	Res.	Res.
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 **DACC2DHR[11:0]**：DAC 2 通道 12 位左对齐数据 (DAC channel2 12-bit left-aligned data)
这些位由软件写入，用于为 DAC 2 通道指定 12 位数据。

位 3:0 保留，必须保持复位值。

22.7.8 DAC 2 通道 8 位右对齐数据保持寄存器 (DAC_DHR8R2)

DAC channel2 8-bit right-aligned data holding register

此寄存器仅适用于双通道 DAC。请参见*第 22.3 节：DAC 实现*。

偏移地址：0x1C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	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.	Res.	Res.	Res.	Res.	Res.	Res.	DACC2DHR[7:0]							
								rw	rw	rw	rw	rw	rw	rw	rw

位 31:8 保留，必须保持复位值。

位 7:0 **DACC2DHR[7:0]**：DAC 2 通道 8 位右对齐数据 (DAC channel2 8-bit right-aligned data)
这些位由软件写入，用于为 DAC 2 通道指定 8 位数据。

22.7.9 双 DAC 12 位右对齐数据保持寄存器 (DAC_DHR12RD)

Dual DAC 12-bit right-aligned data holding 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
Res.	Res.	Res.	Res.	DACC2DHR[11:0]											
				rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	DACC1DHR[11:0]											
				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 27:16 **DACC2DHR[11:0]**：DAC 2 通道 12 位右对齐数据 (DAC channel2 12-bit right-aligned data)
这些位由软件写入，用于为 DAC 2 通道指定 12 位数据。

位 15:12 保留，必须保持复位值。

位 11:0 **DACC1DHR[11:0]**：DAC 1 通道 12 位右对齐数据 (DAC channel1 12-bit right-aligned data)
这些位由软件写入，用于为 DAC 1 通道指定 12 位数据。

22.7.10 双 DAC 12 位左对齐数据保持寄存器 (DAC_DHR12LD)

Dual DAC 12-bit left aligned data holding 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
DACC2DHR[11:0]												Res.	Res.	Res.	Res.
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
DACC1DHR[11:0]												Res.	Res.	Res.	Res.
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw				

位 31:20 **DACC2DHR[11:0]**：DAC 2 通道 12 位左对齐数据 (DAC channel2 12-bit left-aligned data)
这些位由软件写入，用于为 DAC 2 通道指定 12 位数据。

位 19:16 保留，必须保持复位值。

位 15:4 **DACC1DHR[11:0]**：DAC 1 通道 12 位左对齐数据 (DAC channel1 12-bit left-aligned data)
这些位由软件写入，用于为 DAC 1 通道指定 12 位数据。

位 3:0 保留，必须保持复位值。

22.7.11 双 DAC 8 位右对齐数据保持寄存器 (DAC_DHR8RD)

Dual DAC 8-bit right aligned data holding register

偏移地址：0x28

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	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
DACC2DHR[7:0]								DACC1DHR[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 **DACC2DHR[7:0]**：DAC 2 通道 8 位右对齐数据 (DAC channel2 8-bit right-aligned data)
这些位由软件写入，用于为 DAC 2 通道指定 8 位数据。

位 7:0 **DACC1DHR[7:0]**：DAC 1 通道 8 位右对齐数据 (DAC channel1 8-bit right-aligned data)
这些位由软件写入，用于为 DAC 1 通道指定 8 位数据。

22.7.12 DAC 1 通道数据输出寄存器 (DAC_DOR1)

DAC channel1 data output register

偏移地址：0x2C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	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.	Res.	Res.	DACC1DOR[11:0]											
				r	r	r	r	r	r	r	r	r	r	r	r

位 31:12 保留，必须保持复位值。

位 11:0 **DACC1DOR[11:0]**：DAC 1 通道数据输出 (DAC channel1 data output)
这些位为只读，其中包含 DAC 1 通道的数据输出。

22.7.13 DAC 2 通道数据输出寄存器 (DAC_DOR2)

DAC channel2 data output register

此寄存器仅适用于双通道 DAC。请参见*第 22.3 节：DAC 实现*。

偏移地址：0x30

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	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.	Res.	Res.	DACC2DOR[11:0]											
				r	r	r	r	r	r	r	r	r	r	r	r

位 31:12 保留，必须保持复位值。

位 11:0 **DACC2DOR[11:0]**：DAC 2 通道数据输出 (DAC channel2 data output)
这些位为只读，其中包含 DAC 2 通道的数据输出。

22.7.14 DAC 状态寄存器 (DAC_SR)

DAC status register

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
BWST2	CAL_FLAG2	DMAUDR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
r	r	rc_w1													
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BWST1	CAL_FLAG1	DMAUDR1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
r	r	rc_w1													

位 31 **BWST2**：DAC 通道 2 写入采样时间忙标志 (DAC Channel 2 busy writing sample time flag)
此位会在采样和保持模式使能后由系统置 1，每次软件写入 DAC_SHSR2 寄存器时，此位都会置 1，对 DAC_SHSR2 的写操作完成后，此位通过硬件清零。（需要约 3 个 LSI 同步周期）。
0：目前未对 DAC_SHSR2 进行写操作：DAC_SHSR2 可写入
1：目前正在对 DAC_SHSR2 进行写操作：DAC_SHSR2 不可写入
注： 此位仅适用于双通道 DAC。请参见第 22.3 节：DAC 实现。

位 30 **CAL_FLAG2**：DAC 通道 2 校准偏移状态 (DAC Channel 2 calibration offset status)
此位通过硬件置 1 和清零。
0：校准调整值小于偏移校正值
1：校准调整值等于或大于偏移校正值
注： 此位仅适用于双通道 DAC。请参见第 22.3 节：DAC 实现。

位 29 **DMAUDR2**：DAC 2 通道 DMA 下溢标志 (DAC channel2 DMA underrun flag)
此位由硬件置 1，由软件清零（写入 1）。
0：DAC 2 通道未发生 DMA 下溢错误状况
1：DAC 2 通道发生 DMA 下溢错误状况（当前所选触发源以高于 DMA 服务能力的频率驱动 DAC 2 通道转换）。
注： 此位仅适用于双通道 DAC。请参见第 22.3 节：DAC 实现。

位 28 保留，必须保持复位值。

位 27 保留，必须保持复位值。

位 26:16 保留，必须保持复位值。

位 15 **BWST1**：DAC 通道 1 写入采样时间忙标志 (DAC Channel 1 busy writing sample time flag)
此位会在采样和保持模式使能后由系统置 1，每次软件写入 DAC_SHSR1 寄存器时，此位都会置 1，对 DAC_SHSR1 的写操作完成后，此位通过硬件清零。（需要约 3 个 LSI 同步周期）。
0：目前未对 DAC_SHSR1 进行写操作：DAC_SHSR1 可写入
1：目前正在对 DAC_SHSR1 进行写操作：DAC_SHSR1 不可写入

位 14 **CAL_FLAG1**：DAC 通道 1 校准偏移状态 (DAC Channel 1 calibration offset status)
此位通过硬件置 1 和清零。
0：校准调整值小于偏移校正值
1：校准调整值等于或大于偏移校正值

位 13 **DMAUDR1**：DAC 1 通道 DMA 下溢标志 (DAC channel1 DMA underrun flag)
此位由硬件置 1，由软件清零（写入 1）。
0：DAC 1 通道未发生 DMA 下溢错误状况
1：DAC 1 通道发生 DMA 下溢错误状况（当前所选触发源以高于 DMA 服务能力的频率驱动 DAC 1 通道转换）

位 12 保留，必须保持复位值。

位 11 保留，必须保持复位值。

位 10:0 保留，必须保持复位值。

22.7.15 DAC 校准控制寄存器 (DAC_CCR)

DAC calibration control register

偏移地址：0x38

复位值：0x00XX 00XX

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.	OTRIM2[4:0]				
											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.	Res.	Res.	Res.	Res.	Res.	Res.	OTRIM1[4:0]				
											rw	rw	rw	rw	rw

位 31:21 保留，必须保持复位值。

位 20:16 **OTRIM2[4:0]**：DAC 通道 2 偏移调整值 (DAC Channel 2 offset trimming value)
这些位仅适用于双通道 DAC。请参见*第 22.3 节：DAC 实现*。

位 15:5 保留，必须保持复位值。

位 4:0 **OTRIM1[4:0]**：DAC 通道 1 偏移调整值 (DAC Channel 1 offset trimming value)

22.7.16 DAC 模式控制寄存器 (DAC_MCR)

DAC mode control register

偏移地址：0x3C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MODE2[2:0]		
													rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MODE1[2:0]		
													rw	rw	rw

位 31:26 保留，必须保持复位值。

位 25 保留，必须保持复位值。

位 24 保留，必须保持复位值。

位 23:19 保留，必须保持复位值。

位 18:16 **MODE2[2:0]**：DAC 通道 2 模式 (DAC Channel 2 mode)

仅当 DAC 已禁止且不处于校准模式时（DAC_CR 寄存器中的位 EN2=0 且位 CEN2 =0），才可写入这些位。如果 EN2=1 或 CEN2 =1，则会忽略写操作。

这些位可由软件置 1 和清零，用于选择 DAC 通道 2 模式。

– DAC 通道 2 处于正常模式

000：DAC 通道 2 连接到外部引脚且使能了缓冲器

001：DAC 通道 2 连接到外部引脚以及片上外设且使能了缓冲器

010：DAC 通道 2 连接到外部引脚且禁止了缓冲器

011：DAC 通道 2 连接到片上外设且禁止了缓冲器

– DAC 通道 2 处于采样和保持模式

100：DAC 通道 2 连接到外部引脚且使能了缓冲器

101：DAC 通道 2 连接到外部引脚以及片上外设且使能了缓冲器

110：DAC 通道 2 连接到外部引脚以及片上外设且禁止了缓冲器

111：DAC 通道 2 连接到片上外设且禁止了缓冲器

注：　仅可在 EN2 = 0 时修改该寄存器。

这些位仅适用于双通道 DAC。请参见第 22.3 节：DAC 实现。

位 15:14 保留，必须保持复位值。

位 13:10 保留，必须保持复位值。

位 9 保留，必须保持复位值。

位 8 保留，必须保持复位值。

位 7:3 保留，必须保持复位值。

位 2:0 **MODE1[2:0]**：DAC 通道 1 模式 (DAC Channel 1 mode)

仅当 DAC 已禁止且不处于校准模式时（DAC_CR 寄存器中的位 EN1=0 且位 CEN1 =0），才可写入这些位。如果 EN1=1 或 CEN1 =1，则会忽略写操作。

这些位可由软件置 1 和清零，用于选择 DAC 通道 1 模式。

– DAC 通道 1 处于正常模式

000：DAC 通道 1 连接到外部引脚且使能了缓冲器

001：DAC 通道 1 连接到外部引脚以及片上外设且使能了缓冲器

010：DAC 通道 1 连接到外部引脚且禁止了缓冲器

011：DAC 通道 1 连接到片上外设且禁止了缓冲器

– DAC 通道 1 处于采样和保持模式

100：DAC 通道 1 连接到外部引脚且使能了缓冲器

101：DAC 通道 1 连接到外部引脚以及片上外设且使能了缓冲器

110：DAC 通道 1 连接到外部引脚以及片上外设且禁止了缓冲器

111：DAC 通道 1 连接到片上外设且禁止了缓冲器

注：　仅可在 EN1 = 0 时修改该寄存器。

22.7.17 DAC 通道 1 采样和保持采样时间寄存器 (DAC_SHSR1)

DAC channel 1 sample and hold sample time register

偏移地址：0x40

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	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.	Res.	Res.	Res.	Res.	TSAMPLE1[9:0]									
						rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:10 保留，必须保持复位值。

位 9:0 **TSAMPLE1[9:0]**：DAC 通道 1 采样时间 (DAC Channel 1 sample Time)（仅在采样和保持模式下有效）

当 DAC 通道 1 已禁止或者处于正常运行期间时，可写入这些位。对于后一种情况，仅当 DAC_SCR 寄存器的 BWSTx 为低电平时，才可进行写操作，如果 BWSTx=1，会忽略写操作。

注： *它代表执行采样阶段的 LSI 时钟数。采样时间* = (TSAMPLE1[9:0] + 1) x *LSI 时钟周期。*

22.7.18 DAC 通道 2 采样和保持采样时间寄存器 (DAC_SHSR2)

DAC channel 2 sample and hold sample time register

此寄存器仅适用于双通道 DAC。请参见*第 22.3 节：DAC 实现*。

偏移地址：0x44

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.	Res.	Res.	Res.	Res.	TSAMPLE2[9:0]									
						rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:10 保留，必须保持复位值。

位 9:0 **TSAMPLE2[9:0]**：DAC 通道 2 采样时间 (DAC Channel 2 sample Time)（仅在采样和保持模式下有效）

当 DAC 通道 2 已禁止或者处于正常运行期间时，可写入这些位。对于后一种情况，仅当 DAC_SR 寄存器的 BWSTx 为低电平时，才可进行写操作，如果 BWSTx=1，会忽略写操作。

注： *它代表执行采样阶段的 LSI 时钟数。采样时间* = (TSAMPLE1[9:0] + 1) x *LSI 时钟周期。*

22.7.19 DAC 采样和保持时间寄存器 (DAC_SHHR)

DAC sample and hold time register

偏移地址：0x48

复位值：0x0001 0001

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	THOLD2[9:0]									
						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.	Res.	Res.	Res.	THOLD1[9:0]									
						rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:26 保留，必须保持复位值。

位 25:16 **THOLD2[9:0]**：DAC 通道 2 保持时间 (DAC Channel 2 hold time)（仅在采样和保持模式下有效）。

保持时间 = (THOLD[9:0]) x LSI 时钟周期

注： 仅可在 EN2 = 0 时修改该寄存器。

这些位仅适用于双通道 DAC。请参见第 22.3 节：DAC 实现。

位 15:10 保留，必须保持复位值。

位 9:0 **THOLD1[9:0]**：DAC 通道 1 保持时间 (DAC Channel 1 hold Time)（仅在采样和保持模式下有效）

保持时间 = (THOLD[9:0]) x LSI 时钟周期

注： 仅可在 EN2 = 0 时修改该寄存器。

注： 仅当 DAC 通道已禁止且处于正常运行模式时（DAC_CR 寄存器中的位 ENx=0 且位 CEN2x=0），才可写入这些位。如果 ENx=1 或 CENx =1，则会忽略写操作。

22.7.20 DAC 采样和保持刷新时间寄存器 (DAC_SHRR)

DAC sample and hold refresh time register

偏移地址：0x4C

复位值：0x0001 0001

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TREFRESH2[7:0]							
								rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TREFRESH1[7:0]							
								rw	rw	rw	rw	rw	rw	rw	rw

位 31:24 保留，必须保持复位值。

位 23:16 **TREFRESH2[7:0]**：DAC 通道 2 刷新时间 (DAC Channel 2 refresh Time)（仅在采样和保持模式下有效）

刷新时间 = (TREFRESH[7:0]) x LSI 时钟周期

注： 仅可在 EN2 = 0 时修改该寄存器。

这些位仅适用于双通道 DAC。请参见第 22.3 节：DAC 实现。

位 15:8 保留，必须保持复位值。

位 7:0 **TREFRESH1[7:0]**：DAC 通道 1 刷新时间 (DAC Channel 1 refresh Time)（仅在采样和保持模式下有效）

刷新时间 = (TREFRESH[7:0]) x LSI 时钟周期

注： 仅可在 EN2 = 0 时修改该寄存器。

注： 仅当 DAC 通道已禁止且处于正常运行模式时（DAC_CR 寄存器中的位 ENx=0 且位 CEN2x=0），才可写入这些位。如果 ENx=1 或 CENx =1，则会忽略写操作。

22.7.21 DAC 寄存器映射

表 134 汇总了 DAC 寄存器。

表 134. DAC 寄存器映射和复位值

偏移	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	**DAC_CR**	Res.	CEN2	DMAUDRIE2	DMAEN2	MAMP2[3:0]				WAVE2[2:0]		TSEL22	TSEL21	TSEL20	TEN2	Res.	EN2	HFSEL	CEN1	DMAUDRIE1	DMAEN1	MAMP1[3:0]				WAVE1[2:0]		TSEL12	TSEL11	TSEL10	TEN1	Res.	EN1
	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
0x04	**DAC_ SWTRGR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SWTRIG2	SWTRIG1
	Reset value																															0	0
0x08	**DAC_ DHR12R1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DACC1DHR[11:0]											
	Reset value																					0	0	0	0	0	0	0	0	0	0	0	0
0x0C	**DAC_ DHR12L1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DACC1DHR[11:0]												Res.	Res.	Res.	Res.
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0				
0x10	**DAC_ DHR8R1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DACC1DHR[7:0]							
	Reset value																									0	0	0	0	0	0	0	0
0x14	**DAC_ DHR12R2**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DACC2DHR[11:0]											
	Reset value																					0	0	0	0	0	0	0	0	0	0	0	0
0x18	**DAC_ DHR12L2**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DACC2DHR[11:0]												Res.	Res.	Res.	Res.
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0				
0x1C	**DAC_ DHR8R2**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DACC2DHR[7:0]							
	Reset value																									0	0	0	0	0	0	0	0
0x20	**DAC_ DHR12RD**	Res.	Res.	Res.	Res.	DACC2DHR[11:0]												Res.	Res.	Res.	Res.	DACC1DHR[11:0]											
	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
0x24	**DAC_ DHR12LD**	DACC2DHR[11:0]												Res.	Res.	Res.	Res.	DACC1DHR[11:0]												Res.	Res.	Res.	Res.
	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				
0x28	**DAC_ DHR8RD**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DACC2DHR[7:0]				DACC1DHR[7:0]											
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x2C	**DAC_ DOR1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DACC1DOR[11:0]											
	Reset value																					0	0	0	0	0	0	0	0	0	0	0	0
0x30	**DAC_ DOR2**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DACC2DOR[11:0]											
	Reset value																					0	0	0	0	0	0	0	0	0	0	0	0
0x34	**DAC_SR**	BWST2	CAL_FLAG2	DMAUDR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BWST1	CAL_FLAG1	DMAUDR1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	Reset value	0	0	0														0	0	0													

表 134. DAC 寄存器映射和复位值（续）

偏移	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
0x38	**DAC_CCR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OTRIM2[4:0]					Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OTRIM1[4:0]				
	Reset value												X	X	X	X	X												X	X	X	X	X
0x3C	**DAC_MCR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MODE2[2:0]			Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MODE1[2:0]		
	Reset value														0	0	0														0	0	0
0x40	**DAC_SHSR1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TSAMPLE1[9:0]									
	Reset value																							0	0	0	0	0	0	0	0	0	0
0x44	**DAC_SHSR2**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TSAMPLE2[9:0]									
	Reset value																							0	0	0	0	0	0	0	0	0	0
0x48	**DAC_SHHR**	Res.	Res.	Res.	Res.	Res.	Res.	THOLD2[9:0]										Res.	Res.	Res.	Res.	Res.	Res.	THOLD1[9:0]									
	Reset value							0	0	0	0	0	0	0	0	0	1							0	0	0	0	0	0	0	0	0	1
0x4C	**DAC_SHRR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TREFRESH2[7:0]								Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TREFRESH1[7:0]							
	Reset value									0	0	0	0	0	0	0	1									0	0	0	0	0	0	0	1

有关寄存器边界地址的信息，请参见*第 2.2.2 节：存储器映射和寄存器边界地址*。

23 数字摄像头接口 (DCMI)

23.1 DCMI 简介

数字摄像头接口是一个同步并行接口，能够接收外部 8 位、10 位、12 位或 14 位 CMOS 摄像头模块发出的高速数据流。可支持不同的数据格式：YCbCr4:2:2/RGB565 逐行视频和压缩数据 (JPEG)。

此接口适用于黑白摄像头、X24 和 X5 摄像头，并假定所有预处理（如调整大小）都在摄像头模块中执行。

23.2 DCMI 主要特性

- 8 位、10 位、12 位或 14 位并行接口
- 内嵌码/外部行同步和帧同步
- 连续模式或快照模式
- 裁剪功能
- 支持以下数据格式：
  - 8/10/12/14 位逐行视频：单色或原始拜尔格式
  - YCbCr 4:2:2 逐行视频
  - RGB 565 逐行视频
  - 压缩数据：JPEG

23.3 DCMI 时钟

数字摄像头接口使用 DCMI_PIXCLK 和 HCLK 这两个时钟域。伴随 DCMI_PIXCLK 产生的信号稳定之后，将在 HCLK 上升沿时对这些信号进行采样。HCLK 域中的一个使能信号用于指示摄像头发出的数据已稳定，可进行采样。DCMI_PIXCLK 的最大周期必须大于 2.5 倍 HCLK 周期。

23.4 DCMI 功能概述

数字摄像头接口是一个同步并行接口，可接收高速数据流。该接口包含多达 14 条数据线 (D13-D0) 和一条像素时钟线 (DCMI_PIXCLK)。像素时钟的极性可以编程，因此可以在像素时钟的上升沿或下降沿捕获数据。

这些数据以 32 位（4 字节）对齐，存储到 32 位数据寄存器 (DCMI_DR) 中，然后通过通用 DMA 进行传输。图像缓冲区由 DMA 管理，而不是由摄像头接口管理。

从摄像头接收的数据可以按行/帧来组织（原始 YUB/RGB/拜尔模式），也可以是一系列 JPEG 图像。要使能 JPEG 图像接收，必须将 JPEG 位（DCMI_CR 寄存器的位 3）置 1。

数据流可由可选的 DCMI_HSYNC（水平同步）信号和 DCMI_VSYNC（垂直同步）信号硬件同步，或者通过数据流中嵌入的同步码同步。

23.4.1 DCMI 框图

图 145 显示了 DCMI 框图。

图 145. DCMI 框图

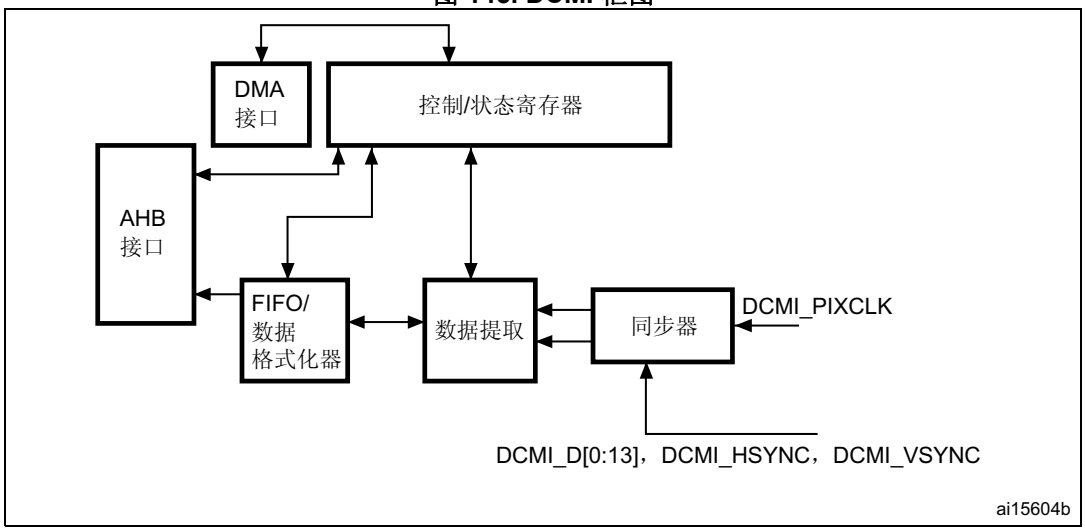
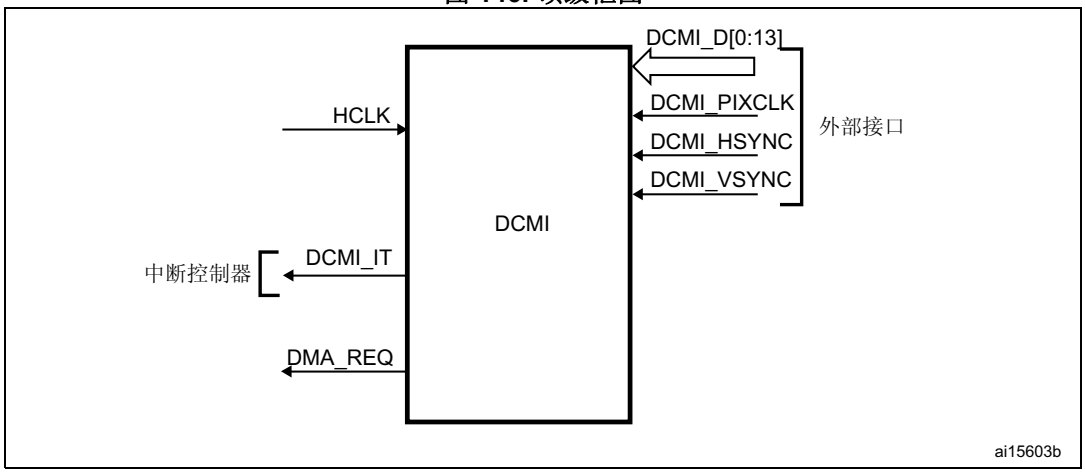


图 146. 顶级框图



23.4.2 DMA 接口

当 DCMI_CR 寄存器中的 CAPTURE 位置 1 时，激活 DMA 接口。摄像头接口每次在其寄存器中收到一个完整的 32 位数据块时，都将触发一个 DMA 请求。

23.4.3 DCMI 物理接口

该接口由 11/13/15/17 个输入信号组成。仅支持从模式。

根据 DCMI_CR 寄存器中 EDM[1:0] 位的设置，摄像头接口可以捕获 8 位、10 位、12 位或 14 位数据。如果使用的位数少于 14，则必须将未使用的输入引脚接地。

表 135 显示了 DCMI 引脚。

表 135. DCMI 外部信号

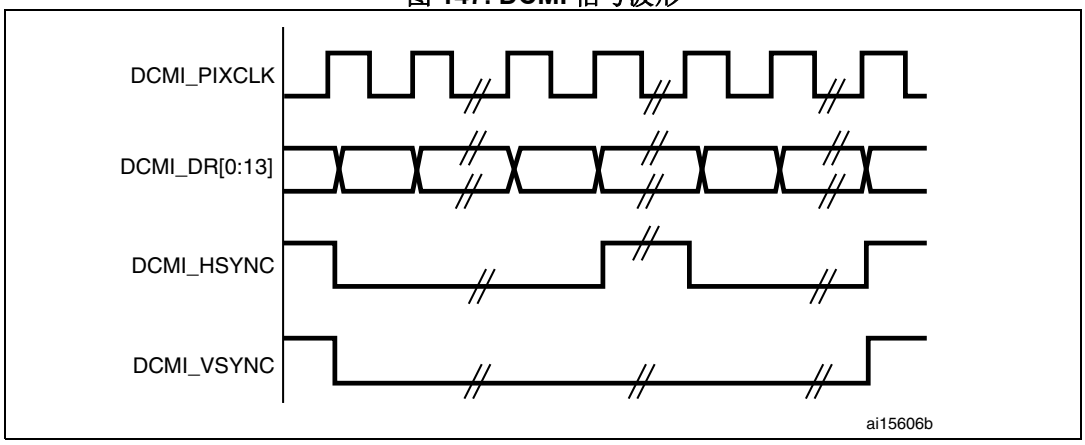
信号名称		信号类型	信号说明
8 位 10 位 12 位 14 位	DCMI_D[0..7] DCMI_D[0..9] DCMI_D[0..11] DCMI_D[0..13]	数字输入	DCMI 数据
DCMI_PIXCLK		数字输入	像素时钟
DCMI_HSYNC		数字输入	水平同步/数据有效
DCMI_VSYNC		数字输入	垂直同步

数据与 DCMI_PIXCLK 保持同步，并根据像素时钟的极性在像素时钟上升沿/下降沿发生变化。

DCMI_HSYNC 信号指示行的开始/结束。

DCMI_VSYNC 信号指示帧的开始/结束。

图 147. DCMI 信号波形



1. DCMI_PIXCLK 的捕获沿为下降沿，DCMI_HSYNC 和 DCMI_VSYNC 的有效状态为 1。
2. DCMI_HSYNC 和 DCMI_VSYNC 的状态可同时发生更改。

8 位数据

当 DCMI_CR 中的 EDM[1:0] 编程为“00”时，接口将捕获其输入 (DCMI_D[0:7]) 的 8 个 LSB，并将其存储为 8 位数据。DCMI_D[13:8] 输入则忽略。在此情况下，要捕获 32 位字，摄像头接口需要花费四个像素时钟周期。

捕获的第一个数据字节放置在 32 位字的 LSB 位置，捕获的第四个数据字节放置在 32 位字的 MSB 位置。*表 136* 列举了捕获到的数据字节在两个 32 位字中的位置排布。

表 136. 捕获的数据字节在 32 位字（宽 8 位）中的位置排布

字节地址	31:24	23:16	15:8	7:0
0	D_{n+3}[7:0]	D_{n+2}[7:0]	D_{n+1}[7:0]	D_n[7:0]
4	D_{n+7}[7:0]	D_{n+6}[7:0]	D_{n+5}[7:0]	D_{n+4}[7:0]

10 位数据

当 DCMI_CR 中的 EDM[1:0] 编程为“01”时，摄像头接口将捕获其输入 DCMI_D[0..9] 的 10 位数据，并将其存储为 16 位字的 10 个最低有效位。DCMI_DR 寄存器中的其余最高有效位（位 11 到 15）将清零。因此，在此情况下，每两个像素时钟周期会生成一个 32 位数据字。

捕获的第一个数据放置在 32 位字的 LSB 位置，捕获的第二个数据放置在 32 位字的 MSB 位置，如*表 137* 中所示。

表 137. 捕获的数据字节在 32 位字（宽 10 位）中的位置排布

字节地址	31:26	25:16	15:10	9:0
0	0	D_{n+1}[9:0]	0	D_n[9:0]
4	0	D_{n+3}[9:0]	0	D_{n+2}[9:0]

12 位数据

当 DCMI_CR 中的 EDM[1:0] 编程为“10”时，摄像头接口将捕获其输入 DCMI_D[0..11] 的 12 位数据，并将其存储为 16 位字的 12 个最低有效位。其余最高有效位将清零。因此，在此情况下，每两个像素时钟周期会生成一个 32 位数据字。

捕获的第一个数据放置在 32 位字的 LSB 位置，捕获的第二个数据放置在 32 位字的 MSB 位置，如*表 138* 中所示。

表 138. 捕获的数据字节在 32 位字（宽 12 位）中的位置排布

字节地址	31:28	27:16	15:12	11:0
0	0	D_{n+1}[11:0]	0	D_n[11:0]
4	0	D_{n+3}[11:0]	0	D_{n+2}[11:0]

14 位数据

当 DCMI_CR 中的 EDM[1:0] 编程为“11”时，摄像头接口将捕获其输入 DCMI_D[0..13] 的 14 位数据，并将其存储为 16 位字的 14 个最低有效位。其余最高有效位将清零。因此，在此情况下，每两个像素时钟周期会生成一个 32 位数据字。

捕获的第一个数据放置在 32 位字的 LSB 位置，捕获的第二个数据放置在 32 位字的 MSB 位置，如*表 139* 中所示。

表 139. 捕获的数据字节在 32 位字（宽 14 位）中的位置排布

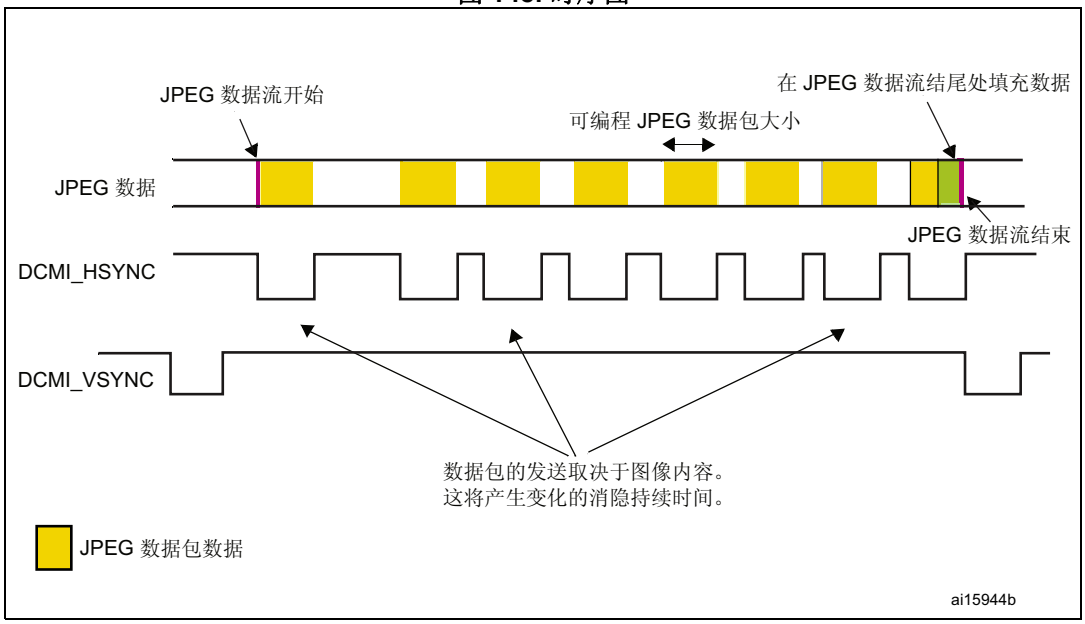
字节地址	31:30	29:16	15:14	13:0
0	0	D_{n+1}[13:0]	0	D_n[13:0]
4	0	D_{n+3}[13:0]	0	D_{n+2}[13:0]

23.4.4 同步

数字摄像头接口支持内嵌码同步或硬件（DCMI_HSYNC 和 DCMI_VSYNC）同步。使用内嵌码同步时，由数字摄像头模块确保 0x00 和 0xFF 值仅用于同步（不用于数据中）。只有 8 位并行数据接口宽度支持内嵌同步码（即，DCMI_CR 寄存器中的 EDM[1:0] 位应为“00”）。

对于压缩数据，DCMI 仅支持硬件同步模式。这种情况下，DCMI_VSYNC 指示图像的开始/结束，DCMI_HSYNC 则用作“数据有效”信号。*图 148* 显示了相应的时序图。

图 148. 时序图



硬件同步模式

硬件同步模式下将使用两个同步信号 (DCMI_HSYNC/DCMI_VSYNC)。

根据摄像头模块/模式的不同，可能在水平/垂直同步期间内发送数据。由于系统会忽略 DCMI_HSYNC/DCMI_VSYNC 信号有效电平期间内接收的所有数据，DCMI_HSYNC/DCMI_VSYNC 信号相当于消隐信号。

为了正确地将图像传输到 DMA/RAM 缓冲区，数据传输将与 DCMI_VSYNC 信号同步。选择硬件同步模式并启用捕获（DCMI_CR 中的 CAPTURE 位置 1）时，数据传输将与 DCMI_VSYNC 信号的无效电平同步 （开始下一帧时）。

之后传输便可以连续执行，由 DMA 将连续帧传输到多个连续的缓冲区或一个具有循环特性的缓冲区。为了允许 DMA 管理连续帧，每一帧结束时都将激活 VSIF（垂直同步中断标志）。

内嵌码同步模式

在此同步模式下，将使用数据流中嵌入的 32 位码来同步数据流。这些码使用数据中*不*再使用的值 0x00/0xFF。共有 4 种同步码类型，均采用 0xFF0000XY 格式。只有 8 位并行数据接口支持内嵌码同步（DCMI_CR 寄存器中的 EDM[1:0] 位应编程为“00”）。对于其他数据宽度，此模式将造成无法预知的结果，因此不得使用。

注： *（在隔行扫描模式下）摄像头模块具有 8 个此类编码。因此，摄像头接口不支持隔行扫描模式（否则每帧数据的一半都会被丢弃）。*

- 模式 2

  四个内嵌码通知以下事件

  – 帧开始 (FS)
  – 帧结束 (FE)
  – 行开始 (LS)
  – 行结束 (LE)

  4 个同步码采用的格式 0xFF0000XY 中的 XY 值可编程（参见*第 23.7.7 节：DCMI 内嵌同步码寄存器 (DCMI_ESCR)*）。

  将 0xFF 编程为“帧结束”意味着所有未使用的编码均被解释为有效帧结束编码。

  在此模式下，一旦使能摄像头接口，将在首次出现帧结束 (FE) 同步码并且后接帧开始 (FS) 同步码之后开始捕获帧。

- 模式 1

  摄像头模式 1 是另一种编码。此模式与 ITU656 兼容。

  这些同步码通知另一组事件：

  – SAV（有效行）——行开始
  – EAV（有效行）——行结束
  – SAV（消隐）——帧间消隐期内的行开始
  – EAV（消隐）——帧间消隐期内的行结束

可通过对同步码进行如下编程来支持此模式：

- FS ≤ 0xFF
- FE ≤ 0xFF
- LS ≤ SAV（有效）
- LE ≤ EAV（有效）

此外还针对帧/行开始和帧/行结束同步码实现了非屏蔽码功能。这样可以仅使用同步码中未被屏蔽的位进行比较。因此，可以选择一个位用于同步码的比较，来检测帧/行起始和帧/行结束。这意味着可以多个同步码表示帧/行的起始和结束，它们仅在未被屏蔽的位相同即可。

示例

FS = 0xA5

FS 的非屏蔽码 = 0x10

这种情况下，只需要比较数据码的第 4 位来检测是否是 FS 信号。

23.4.5 捕获模式

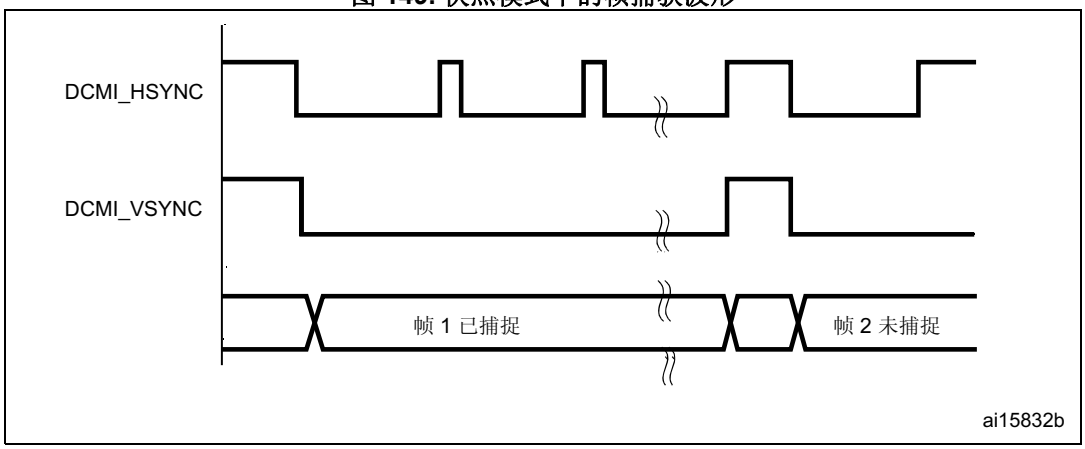
此接口支持两种类型的捕获：快照（单帧）和连续采集。

快照模式（单帧）

此模式下只捕获单帧（DCMI_CR 寄存器中的 CM = “1”）。在 DCMI_CR 中的 CAPTURE 位置 1 后，该接口将等待系统检测帧开始，然后再对数据进行采样。收到完整的第一帧后，将自动禁止摄像头接口（DCMI_CR 中的 CAPTURE 位清零）。如果使能相应中断，将生成中断 (IT_FRAME)。

如果发生溢出，这帧将丢失并且 CAPTURE 位清零。

图 149. 快照模式下的帧捕获波形

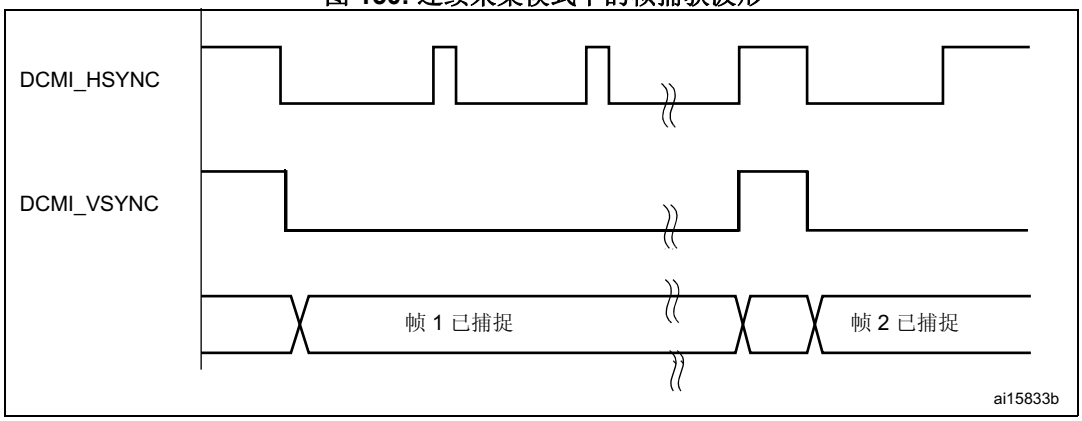


1. 此例中，DCMI_HSYNC 和 DCMI_VSYNC 的有效状态为 1。
2. DCMI_HSYNC 和 DCMI_VSYNC 的状态可同时发生更改。

连续采集模式

在此模式下（DCMI_CR 中的 CM 位 = “0”），一旦 DCMI_CR 中的 CAPTURE 位置 1，将在下一个 DCMI_VSYNC 或内嵌同步码帧起始时启动采集过程，具体取决于同步模式。该过程一直持续到 DCMI_CR 中的 CAPTURE 位清零。CAPTURE 位清零后，采集过程将持续到当前帧结束。

图 150. 连续采集模式下的帧捕获波形



1. 此例中，DCMI_HSYNC 和 DCMI_VSYNC 的有效状态为 1。
2. DCMI_HSYNC 和 DCMI_VSYNC 的状态可同时发生更改。

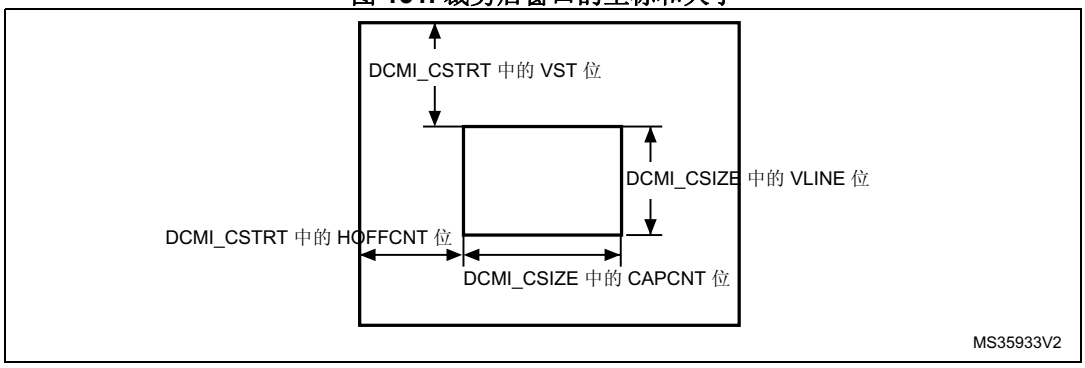
在连续采集模式下，可以通过配置 DCMI_CR 中的 FCRC 位来选择采集所有图片，或每两帧采集一次图片，或每四帧采集一张图片，以此降低帧捕获率。

注： *在硬件同步模式下（DCMI_CR 中的 ESS = “0”），即使 DCMI_CR 中的 CAPTURE = “0”，也将生成 IT_VSYNC 中断（如果使能）。因此为进一步降低帧捕获率，IT_VSYNC 中断可与快照模式结合使用，用来统计 2 次捕获之间的帧数。这并不适用于内嵌码同步模式。*

23.4.6 裁剪功能

摄像头接口可以使用裁剪功能从收到的图像中选择一个矩形窗口。起始（左上角）坐标和窗口大小（用像素时钟数表示的水平尺寸以及用行数表示的垂直尺寸）由两个 32 位寄存器（DCMI_CWSTRT 和 DCMI_CWSIZE）指定。窗口大小用像素时钟数（水平尺寸）和行数（垂直尺寸）表示。

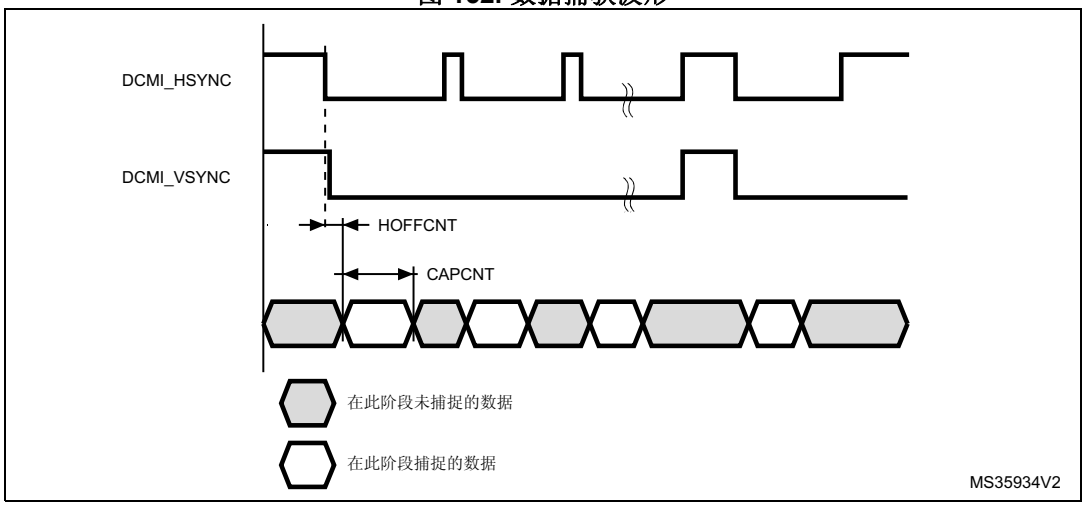
图 151. 裁剪后窗口的坐标和大小



这些寄存器将捕获窗口的起点坐标指定为某一行号（从 0 开始）和某一个像素时钟数（从 0 开始），窗口大小则指定为行数和像素时钟数。CAPCNT 值只能是 4 的倍数（两个最低有效位强制为 0），才能通过 DMA 正确传输数据。

如果在捕获 DCMI_CWSIZE 寄存器中指定行数完成之前，DCMI_VSYNC 信号已有效，那么捕获将停止，并且在中断使能时生成 IT_FRAME 中断。

图 152. 数据捕获波形



1. 此例中，DCMI_HSYNC 和 DCMI_VSYNC 的有效状态为 1。
2. DCMI_HSYNC 和 DCMI_VSYNC 的状态可同时发生更改。

23.4.7 JPEG 格式

要允许接收 JPEG 图像，必须将 DCMI_CR 寄存器中的 JPEG 位置 1。JPEG 图像不按行和帧存储，因此 DCMI_VSYNC 信号用于启动捕获过程，而 DCMI_HSYNC 则用作数据使能信号。行中包含的字节数可能不是 4 的倍数，因此处理此类情况时应十分谨慎，因为需要每次从捕获的数据形成一个完整的 32 位字时，才生成一个 DMA 请求。检测到帧结束并且尚未凑成 32 位字时，将使用“0”进行填充，并触发一个 DMA 请求。

裁剪功能和内嵌码同步不适用于 JPEG 格式。

23.4.8 FIFO

输入模式

为了对 AHB 上的数据传输率加以管理，在摄像头接口上实现了 4 个字深度的 FIFO。DCMI 配有一个简单的 FIFO 控制器，每次摄像头接口从 AHB 读取数据时读指针递增，每次摄像头接口向 FIFO 写入数据时写指针递增。因为没有溢出保护，如果数据传输率超过了 AHB 接口能够承受的速率，FIFO 中的数据就会被覆盖。

如果同步信号出错，或者 FIFO 发生溢出，FIFO 将复位，DCMI 接口将等待新的数据帧开始。

23.5 数据格式说明

23.5.1 数据格式

支持三种类型的数据：

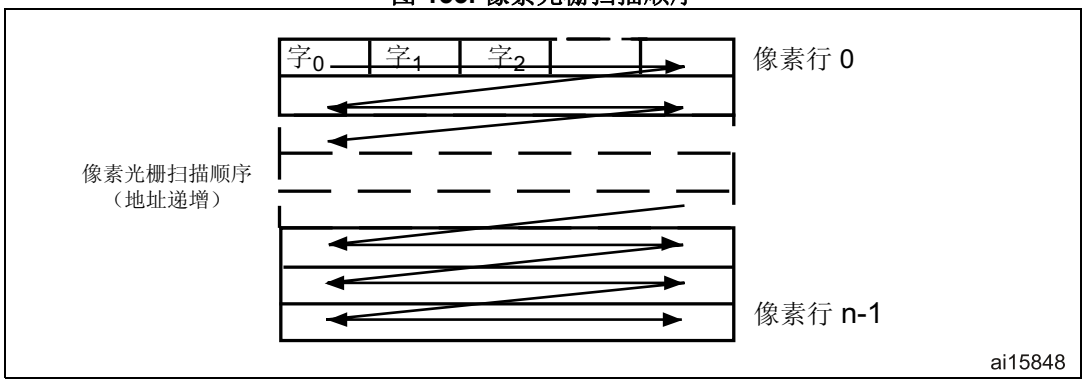
- 8 位逐行视频：单色或原始拜尔格式。
- YCbCr 4:2:2 逐行视频。
- RGB565 逐行视频。采用 16 位（5 位表示蓝色，5 位表示红色，6 位表示绿色）编码的像素需要两个时钟周期传输。

压缩数据：JPEG

对于 B&W、YCbCr 或 RGB 数据，最大输入大小为 2048 × 2048 像素。JPEG 压缩模式无限制。

对于单色、RGB 和 YCbCr，帧缓冲区以光栅模式存储。使用 32 位字。仅支持小端对齐格式。

图 153. 像素光栅扫描顺序



23.5.2 单色格式

特性：

- 光栅格式
- 每个像素 8 位

表 140 显示了数据的存储方式。

表 140. 单色逐行视频格式的数据存储

字节地址	31:24	23:16	15:8	7:0
0	n + 3	n + 2	n + 1	n
4	n + 7	n + 6	n + 5	n + 4

23.5.3 RGB 格式

特性：

- 光栅格式
- RGB
- 交替：一个缓冲区：R、G 和 B 交替：BRGBRGBRG：等
- 对显示输出的优化

RGB 平面格式与标准的 OS 帧缓冲显示格式兼容。
仅支持 16 BPP（每个像素 16 位）：RGB565（每 32 位字表示 2 个像素）。
不支持 24 BPP（调色板格式）和灰度格式。像素按照光栅扫描顺序进行存储，即从顶部像素行到底部像素行，从像素行的左侧到右侧。像素分量包括 R（红色）、G（绿色）和 B（蓝色）。所有分量的空间分辨率都相同（格式 4:4:4）。一帧数据中各个分量交替间隔存储。

表 141 显示了数据的存储方式。

表 141. 以 RGB 逐行视频格式存储数据

字节地址	31:27	26:21	20:16	15:11	10:5	4:0
0	红色 n + 1	绿色 n + 1	蓝色 n + 1	红色 n	绿色 n	蓝色 n
4	红色 n + 4	绿色 n + 3	蓝色 n + 3	红色 n + 2	绿色 n + 2	蓝色 n + 2

23.5.4 YCbCr 格式

特性：

- 光栅格式
- YCbCr 4:2:2
- 交替：一个缓冲区：Y、Cb 和 Cr 交替：CbYCrYCbYCr：等

像素分量包括 Y（亮度）、Cb 和 Cr（蓝色色度和红色色度）。每个分量都采用 8 位进行编码。亮度和色度（交替）存储在一起，如*表 142* 中所示。

表 142. YCbCr 逐行视频格式下的数据存储

字节地址	31:24	23:16	15:8	7:0
0	Y n + 1	Cr n	Y n	Cb n
4	Y n + 3	Cr n + 2	Y n + 2	Cb n + 2

23.5.5 YCbCr 格式——仅含 Y 分量

特性：

- 光栅格式
- YCbCr 4:2:2
- 缓冲区仅包含 Y 分量信息——单色图像

像素分量包括 Y（亮度）、Cb 和 Cr（蓝色色度和红色色度）。在此模式下，将丢弃色度信息。仅存储每个像素采用 8 位进行编码的亮度分量，如*表 143* 所示。

结果为单色图像，其分辨率与原始 YCbCr 数据的分辨率相同。

表 143. YCbCr 逐行视频格式下的数据存储——Y 分量提取模式

字节地址	31:24	23:16	15:8	7:0
0	Y n + 3	Y n + 2	Y n + 1	Y n
4	Y n + 7	Y n + 6	Y n + 5	Y n + 4

23.5.6 半分辨率图像提取

这是对先前接收模式的修改，适用于单色、RGB 或 Y 分量提取模式。

此模式仅允许存储半分辨率图像。可通过 OELS 和 LSM 控制位进行选择。

23.6 DCMI 中断

有五种中断源。所有中断都可通过软件屏蔽。全局中断 (IT_DCMI) 是所有单个中断的逻辑或运算所得结果。*表 144* 列出了所有中断。

表 144. DCMI 中断

中断名称	中断事件
IT_LINE	表示行结束
IT_FRAME	表示帧捕获结束
IT_OVR	表示接收的数据发生溢出错误
IT_VSYNC	表示同步帧
IT_ERR	表示内嵌码同步帧检测期间检测到错误
IT_DCMI	以上中断的逻辑或结果

23.7 DCMI 寄存器说明

所有 DCMI 寄存器都必须作为 32 位字访问，否则将发生总线错误。

23.7.1 DCMI 控制寄存器 (DCMI_CR)

DCMI control 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
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OELS	LSM	OEBS	BSM[1:0]	
											rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	ENABLE	Res.	Res.	EDM[1:0]		FCRC[1:0]		VSPOL	HSPOL	PCKPOL	ESS	JPEG	CROP	CM	CAPTURE
	rw			rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:21 保留，必须保持复位值。

位 20 **OELS**：奇数/偶数行选择（行选择开始）(Odd/Even Line Select (Line Select Start))
此位与 LSM 字段 (LSM = 1) 一起使用
0：接口在帧开始后捕获第一行，同时丢弃第二行
1：接口在帧开始后捕获第二行，同时丢弃第一行

位 19 **LSM**：行选择模式 (Line Select mode)
0：接口捕获所有接收到的行
1：接口每两行捕获一行

位 18 **OEBS**：奇数/偶数字节选择（字节选择开始）(Odd/Even Byte Select (Byte Select Start))

此位与 BSM 字段 (BSM <> 00) 一起使用

0：接口从帧/行开始捕获第一个数据（字节或双字节），同时丢弃第二个字节

1：接口从帧/行开始捕获第二个数据（字节或双字节），同时丢弃第一个字节

位 17:16 **BSM[1:0]**：字节选择模式 (Byte Select mode)

00：接口捕获所有接收到的数据

01：接口对接收到的数据每隔一个字节进行捕获

10：接口每四个字节捕获一个字节

11：接口每四个字节捕获两个字节

注：　此模式仅在 EDM[1:0]=00 时有效。对于所有其他 EDM 值，必须将此位字段编程为复位值。

位 15 保留，必须保持复位值。

位 14 **ENABLE**：DCMI 使能 (DCMI enable)

0：禁止 DCMI

1：使能 DCMI

注：　使能此位之前，应对 DCMI 配置寄存器进行适当的设置。

位 13:12 保留，必须保持复位值。

位 11:10 **EDM[1:0]**：扩展数据模式 (Extended data mode)

00：接口每个像素时钟捕获 8 位数据

01：接口每个像素时钟捕获 10 位数据

10：接口每个像素时钟捕获 12 位数据

11：接口每个像素时钟捕获 14 位数据

位 9:8 **FCRC[1:0]**：帧捕获率控制 (Frame capture rate control)

这些位定义了帧捕获频率。仅在连续采集模式下有效。快照模式下将被忽略。

00：捕获所有帧

01：每隔一帧捕获一次（带宽降低 50%）

10：每隔三帧捕获一次（带宽降低 75%）

11：保留

位 7 **VSPOL**：垂直同步极性 (Vertical synchronization polarity)

此位指示数据在并行接口上无效时 DCMI_VSYNC 引脚的电平。

0：DCMI_VSYNC 低电平有效

1：DCMI_VSYNC 高电平有效

位 6 **HSPOL**：水平同步极性 (Horizontal synchronization polarity)

此位指示数据在并行接口上无效时 DCMI_HSYNC 引脚的电平。

0：DCMI_HSYNC 低电平有效

1：DCMI_HSYNC 高电平有效

位 5 **PCKPOL**：像素时钟极性 (Pixel clock polarity)

此位用来配置像素时钟的捕获沿

0：下降沿有效。

1：上升沿有效。

位 4 **ESS**：内嵌码同步选择 (Embedded synchronization select)

0：硬件同步，数据捕获（帧/行开始/停止）由 DCMI_HSYNC/DCMI_VSYNC 信号同步。

1：内嵌码同步，数据捕获由数据流中嵌入的同步码同步。

注：　仅对 8 位并行数据有效。ESS 位置 1 时，将忽略 HSPOL/VSPOL。

JPEG 模式下会禁止此位。

位 3 **JPEG**：JPEG 格式 (JPEG format)

0：未经压缩的视频格式

1：此位用于 JPEG 数据传输。DCMI_HSYNC 信号用作数据使能信号。此模式下无法使用裁剪和内嵌码同步功能（ESS 位）。

位 2 **CROP**：裁剪功能 (Crop feature)

0：捕获完整图像。这种情况下，图像帧包含的字节总数应该为 4 的倍数

1：仅捕获剪裁寄存器所指定的窗口中的数据。如果窗口大小超出图片大小，则仅捕获图片大小。

位 1 **CM**：捕获模式 (Capture mode)

0：连续采集模式——收到的数据将通过 DMA 传输到目标存储区。缓冲区位置和模式（线性或循环缓冲区）由系统 DMA 控制。

1：快照模式（单帧）——一旦激活，接口将等待帧开始，然后通过 DMA 传输单帧。帧结束时将自动复位 CAPTURE 位。

位 0 **CAPTURE**：使能捕获 (Capture enable)

0：禁止捕获。

1：使能捕获。

摄像头接口等待第一帧开始，然后生成一个 DMA 请求以将收到的数据传输到目标存储器中。

在快照模式下，收到的第一帧结束时将自动使 CAPTURE 位清零。

在连续采集模式下，如果在执行捕获操作时通过软件将此位清零，则帧结束后此位的清零才生效。

注：　使能此位之前，应对 DMA 控制器和所有 DCMI 配置寄存器进行适当的编程。

23.7.2 DCMI 状态寄存器 (DCMI_SR)

DCMI 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
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.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FNE	VSYNC	HSYNC
													r	r	r

位 31:3 保留，必须保持复位值。

位 2 **FNE**：FIFO 非空 (FIFO not empty)

此位指示 FIFO 的状态

1：FIFO 包含有效数据

0：FIFO 为空

位 1 **VSYNC**：

此位指示配置了适当极性的 DCMI_VSYNC 引脚的状态。

使用内嵌码同步时，此位的含义如下：

0：有效帧

1：在帧之间同步

如果使用内嵌码同步，则仅当 DCMI_CR 中的 CAPTURE 位置 1 时，此位才有意义。

位 0 **HSYNC**：

此位指示配置了适当极性的 DCMI_HSYNC 引脚的状态。

使用内嵌码同步时，此位的含义如下：

0：有效行

1：在行之间同步

如果使用内嵌码同步，则仅当 DCMI_CR 中的 CAPTURE 位置 1 时，此位才有意义。

23.7.3 DCMI 原始中断状态寄存器 (DCMI_RIS)

DCMI raw interrupt 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
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.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LINE_RIS	VSYNC_RIS	ERR_RIS	OVR_RIS	FRAME_RIS
											r	r	r	r	r

DCMI_RIS 用来指示原始中断状态，以只读方式访问。执行读操作时，此寄存器返回的是对应中断未被 DCMI_IER 寄存器屏蔽的原始状态。

位 31:5 保留，必须保持复位值。

位 4 **LINE_RIS**：行原始中断状态 (Line raw interrupt status)

当 DCMI_HSYNC 信号从无效状态更改为有效状态时，此位置 1。即使行无效也会变成高电平。

如果使用内嵌码同步，则仅当 DCMI_CR 中的 CAPTURE 位置 1 时，此位才置 1。

向 DCMI_ICR 中的 LINE_ISC 位写入“1”即可将此位清零。

位 3 **VSYNC_RIS**：DCMI_VSYNC 原始中断状态 (DCMI_VSYNC raw interrupt status)

当 DCMI_VSYNC 信号从无效状态更改为有效状态时，此位置 1。

如果使用内嵌码同步，则仅当 DCMI_CR 中的 CAPTURE 位置 1 时，此位才置 1。

向 DCMI_ICR 中的 VSYNC_ISC 位写入“1”即可将此位清零。

位 2 **ERR_RIS**：同步错误原始中断状态 (Synchronization error raw interrupt status)

0：未检测到同步错误。

1：未按正确顺序接收内嵌码同步字符。

此位仅在内嵌码同步模式下有效。向 DCMI_ICR 中的 ERR_ISC 位写入“1”即可将此位清零。

注： 此位仅适用于内嵌码同步模式。

位 1 **OVR_RIS**：溢出原始中断状态 (Overrun raw interrupt status)

0：未发生数据缓冲区溢出。

1：发生数据缓冲区溢出，数据 FIFO 损坏。

向 DCMI_ICR 中的 OVR_ISC 位写入“1”即可将此位清零。

位 0 **FRAME_RIS**：捕获完成原始中断状态 (Capture complete raw interrupt status)

0：没有新的捕获数据。

1：已捕获一帧。

对一帧数据或裁剪窗口内的数据捕获完毕后，此位置 1。

如果捕获裁剪窗口，则将在裁剪窗口最后一行的行结束时将此位置 1。即使捕获的帧为空（例如，窗口超出帧范围），此位也将置 1。

向 DCMI_ICR 中的 FRAME_ISC 位写入“1”即可将此位清零。

23.7.4 DCMI 中断使能寄存器 (DCMI_IER)

DCMI interrupt enable 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
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.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LINE_IE	VSYNC_IE	ERR_IE	OVR_IE	FRAME_IE
											rw	rw	rw	rw	rw

DCMI_IER 寄存器用于使能中断。当其中任一位置 1 时，将使能相应中断。此寄存器可读写。

位 31:5 保留，必须保持复位值。

位 4 **LINE_IE**：使能行中断 (Line interrupt enable)
0：接收到行时不生成中断
1：完全接收到行时生成一个中断

位 3 **VSYNC_IE**：DCMI_VSYNC 中断使能 (DCMI_VSYNC interrupt enable)
0：不生成中断
1：DCMI_VSYNC 每次从无效电平变为有效电平时都生成一个中断
DCMI_VSYNC 信号的有效电平由 VSPOL 位定义。

位 2 **ERR_IE**：同步错误中断使能 (Synchronization error interrupt enable)
0：不生成中断
1：如果未按正确顺序接收内嵌码同步代码，则生成一个中断
注： 此位仅适用于内嵌码同步模式。

位 1 **OVR_IE**：溢出中断使能 (Overrun interrupt enable)
0：不生成中断
1：如果 DMA 无法在收到新数据（32 位）之前传输上一个数据，则生成一个中断

位 0 **FRAME_IE**：捕获完成中断使能 (Capture complete interrupt enable)
0：不生成中断
1：接收完成一帧数据或裁剪窗口内的数据（在裁剪模式下），生成一个中断

23.7.5 DCMI 屏蔽中断状态寄存器 (DCMI_MIS)

DCMI masked interrupt status register

此 DCMI_MIS 寄存器是一个只读寄存器。执行读操作时，此寄存器将返回相应中断的当前屏蔽状态（取决于 DCMI_IER 中相应中断的值）。如果 DCMI_IER 中的相应使能位和 DCMI_RIS 中的相应位都置 1，则此寄存器中的对应位将置 1。

偏移地址：0x10

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	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.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LINE_MIS	VSYNC_MIS	ERR_MIS	OVR_MIS	FRAME_MIS
											r	r	r	r	r

位 31:5 保留，必须保持复位值。

位 4 **LINE_MIS**：行屏蔽中断状态 (Line masked interrupt status)
此位指示屏蔽行中断的状态
0：接收到行时不生成中断。
1：收到完整一行数据且 DCMI_IER 中的 LINE_IE 位置 1 时生成一个中断。

位 3 **VSYNC_MIS**：VSYNC 屏蔽中断状态 (VSYNC masked interrupt status)
此位指示屏蔽 VSYNC 中断的状态
0：DCMI_VSYNC 电平跳变时不生成中断。
1：DCMI_VSYNC 每次从无效电平变为有效电平且 DCMI_IER 中的 VSYNC_IE 位置 1 时都生成一个中断。
DCMI_VSYNC 信号的有效电平由 VSPOL 位定义。

位 2 **ERR_MIS**：同步错误屏蔽中断状态 (Synchronization error masked interrupt status)
此位指示屏蔽同步错误中断
0：发生同步错误时不生成中断。
1：如果未按正确顺序接收内嵌同步码且 DCMI_IER 中的 ERR_IE 位置 1，则生成一个中断。
注： 此位仅适用于内嵌码同步模式。

位 1 **OVR_MIS**：溢出屏蔽中断状态 (Overrun masked interrupt status)
此位指示屏蔽溢出中断的状态
0：发生溢出时不生成中断。
1：如果 DMA 无法在收到新数据（32 位）之前传输上一个数据且 DCMI_IER 中的 OVR_IE 位置 1，则生成一个中断。

位 0 **FRAME_MIS**：捕获完成屏蔽中断状态 (Capture complete masked interrupt status)
此位指示屏蔽捕获完成中断的状态
0：完成捕获后不生成中断。
1：接收完成一帧数据或裁剪窗口内的数据（在裁剪模式下），且 DCMI_IER 中的 FRAME_IE 位置 1 时生成一个中断。

23.7.6 DCMI 中断清零寄存器 (DCMI_ICR)

DCMI interrupt clear 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
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.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LINE_ISC	VSYNC_ISC	ERR_ISC	OVR_ISC	FRAME_ISC
											w	w	w	w	w

DCMI_ICR 寄存器为只写寄存器。向此寄存器的某一位写入“1”可将 DCMI_RIS 和 DCMI_MIS 寄存器中的相应位清零。写入 “0”则不会带来任何影响。

位 31:5 保留，必须保持复位值。

位 4 **LINE_ISC**：线中断状态清零 (line interrupt status clear)
在此位中写入“1”可将 DCMI_RIS 寄存器中的 LINE_RIS 清零

位 3 **VSYNC_ISC**：垂直同步中断状态清零 (Vertical Synchronization interrupt status clear)
在此位中写入“1”可将 DCMI_RIS 中的 VSYNC_RIS 位清零

位 2 **ERR_ISC**：同步错误中断状态清零 (Synchronization error interrupt status clear)
在此位中写入“1”可将 DCMI_RIS 中的 ERR_RIS 位清零
注： 此位仅适用于内嵌码同步模式。

位 1 **OVR_ISC**：溢出中断状态清零 (Overrun interrupt status clear)
在此位中写入“1”可将 DCMI_RIS 中的 OVR_RIS 位清零

位 0 **FRAME_ISC**：捕获完成中断状态清零 (Capture complete interrupt status clear)
在此位中写入“1”可将 DCMI_RIS 中的 FRAME_RIS 位清零

23.7.7 DCMI 内嵌同步码寄存器 (DCMI_ESCR)

DCMI embedded synchronization code 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
FEC[7:0]								LEC[7:0]							
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
LSC[7:0]								FSC[7:0]							
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:24 **FEC[7:0]**：帧结束分隔码 (Frame end delimiter code)

此字节指定帧结束分隔码。由 4 个字节组成，格式为 0xFF 0x00 0x00 FEC。
如果将 FEC 编程为 0xFF，所有未使用的其他代码 (0xFF0000XY) 都将视为帧结束分隔符。

位 23:16 **LEC[7:0]**：行结束分隔码 (Line end delimiter code)

此字节指定行结束分隔码。由 4 个字节组成，格式为 0xFF 0x00 0x00 LEC。

位 15:8 **LSC[7:0]**：行开始分隔码 (Line start delimiter code)

此字节指定行开始分隔码。由 4 个字节组成，格式为 0xFF 0x00 0x00 LSC。

位 7:0 **FSC[7:0]**：帧开始分隔码 (Frame start delimiter code)

此字节指定帧开始分隔码。由 4 个字节组成，格式为 0xFF 0x00 0x00 FSC。
如果将 FSC 编程为 0xFF，将检测不到任何帧开始分隔符。但在 FEC 代码后第一次出现 LSC 时将视为帧分隔符的开始。

23.7.8 DCMI 内嵌码同步取消屏蔽寄存器 (DCMI_ESUR)

DCMI embedded synchronization unmask register

偏移地址：0x1C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FEU[7:0]								LEU[7:0]							
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
LSU[7:0]								FSU[7:0]							
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:24 **FEU[7:0]**：帧结束分隔符取消屏蔽 (Frame end delimiter unmask)

此字节指定对帧结束分隔码的屏蔽。

0：将帧结束分隔符与收到的数据进行比较时，将屏蔽 DCMI_ESCR 中 FEC 字节中的相应位

1：将帧结束分隔符与收到的数据进行比较时，将比较 DCMI_ESCR 中 FEC 字节中的相应位

位 23:16 **LEU[7:0]**：行结束分隔符取消屏蔽 (Line end delimiter unmask)

此字节指定对行结束分隔码的屏蔽。

0：将行结束分隔符与收到的数据进行比较时，将屏蔽 DCMI_ESCR 中 LEC 字节中的相应位

1：将行结束分隔符与收到的数据进行比较时，将比较 DCMI_ESCR 中 LEC 字节中的相应位

位 15:8 **LSU[7:0]**：行开始分隔符取消屏蔽 (Line start delimiter unmask)

此字节指定对行开始分隔码的屏蔽。

0：将行开始分隔符与收到的数据进行比较时，将屏蔽 DCMI_ESCR 中 LSC 字节中的相应位

1：将行开始分隔符与收到的数据进行比较时，将比较 DCMI_ESCR 中 LSC 字节中的相应位

位 7:0 **FSU[7:0]**：帧开始分隔符取消屏蔽 (Frame start delimiter unmask)

此字节指定对帧开始分隔码的屏蔽。

0：将帧开始分隔符与收到的数据进行比较时，将屏蔽 DCMI_ESCR 中 FSC 字节中的相应位

1：将帧开始分隔符与收到的数据进行比较时，将比较 DCMI_ESCR 中 FSC 字节中的相应位

23.7.9 DCMI 裁剪窗口起点 (DCMI_CWSTRT)

DCMI crop window start

偏移地址：0x20

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	VST[12:0]												
			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.	HOFFCNT[13:0]													
		rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:29 保留，必须保持复位值。

位 28:16 **VST[12:0]**：窗口开始的行数 (Window start line count)

图像捕获从此行开始。对之前行的数据不予捕获。

0x0000 => 行 1

0x0001 => 行 2

0x0002 => 行 3

....

位 15:14 保留，必须保持复位值。

位 13:0 **HOFFCNT[13:0]**：窗口开始的水平方向偏移值 (Horizontal offset count)

窗口行内，每行在捕获数据前需空出的像素时钟个数。

23.7.10 DCMI 裁剪窗口大小 (DCMI_CWSIZE)

DCMI crop window size

偏移地址：0x24

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	VLINE[13:0]													
		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
Res.	Res.	CAPCNT[13:0]													
		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 29:16 **VLINE[13:0]**：垂直行计数 (Vertical line count)

窗口内包含的行数。

0x0000 => 1 行

0x0001 => 2 行

0x0002 => 3 行

....

位 15:14 保留，必须保持复位值。

位 13:0 **CAPCNT[13:0]**：捕获计数 (Capture count)

窗口内要捕获的像素时钟数。其值应与表示不同并行接口宽度的字对齐数据相对应。

0x0000 => 1 个像素

0x0001 => 2 个像素

0x0002 => 3 个像素

....

23.7.11 DCMI 数据寄存器 (DCMI_DR)

DCMI data register

偏移地址：0x28

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Byte3[7:0]								Byte2[7:0]							
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Byte1[7:0]								Byte0[7:0]							
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:24 **Byte3[7:0]**：数据字节 3 (Data byte 3)

位 23:16 **Byte2[7:0]**：数据字节 2 (Data byte 2)

位 15:8 **Byte1[7:0]**：数据字节 1 (Data byte 1)

位 7:0 **Byte0[7:0]**：数据字节 0 (Data byte 0)

数字摄像头接口每收到 32 位数据，才触发一次 DMA 请求。4 字深度的 FIFO 可为 DMA 传输留出足够时间并避免出现 DMA 溢出情况。

23.7.12 DCMI 寄存器映射

表 145 汇总了 DCMI 寄存器。

表 145. DCMI 寄存器地址映射和复位值

偏移	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	DCMI_CR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OELS	LSM	OEBS	BSM		Res.	ENABLE	Res.	Res.	EDM		FCRC		VSPOL	HSPOL	PCKPOL	ESS	JPEG	CROP	CM	CAPTURE
	Reset value												0	0	0	0	0		0			0	0	0	0	0	0	0	0	0	0	0	0
0x04	DCMI_SR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FNE	VSYNC	HSYNC
	Reset value																														0	0	0
0x08	DCMI_RIS	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LINE_RIS	VSYNC_RIS	ERR_RIS	OVR_RIS	FRAME_RIS
	Reset value																												0	0	0	0	0
0x0C	DCMI_IER	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LINE_IE	VSYNC_IE	ERR_IE	OVR_IE	FRAME_IE
	Reset value																												0	0	0	0	0
0x10	DCMI_MIS	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LINE_MIS	VSYNC_MIS	ERR_MIS	OVR_MIS	FRAME_MIS
	Reset value																												0	0	0	0	0
0x14	DCMI_ICR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LINE_ISC	VSYNC_ISC	ERR_ISC	OVR_ISC	FRAME_ISC
	Reset value																												0	0	0	0	0
0x18	DCMI_ESCR	FEC								LEC								LSC								FSC							
	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	DCMI_ESUR	FEU								LEU								LSU								FSU							
	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
0x20	DCMI_CWSTRT	Res.	Res.	Res.	VST[12:0]													Res.	Res.	HOFFCNT[13:0]													
	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
0x24	DCMI_CWSIZE	Res.	Res.	VLINE13:0]														Res.	Res.	CAPCNT[13:0]													
	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
0x28	DCMI_DR	Byte3								Byte2								Byte1								Byte0							
	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*第 78 页的第 2.2.2 节*。

24 电压参考缓冲器 (VREFBUF)

24.1 简介

STM32L4Rxxx 和 STM32L4Sxxx 器件内置有电压参考缓冲器，既可用作 ADC 和 DAC 的参考电压，也可通过 VREF+ 引脚用作外部元件的参考电压。当 VREF + 引脚与 VDDA 引脚在一个封装中互相绑定时，电压参考缓冲器不可用且必须禁用（关于封装引脚分配说明，请参考数据手册）。

24.2 VREFBUF 功能说明

内部电压参考缓冲器支持两种电压值[a]，可利用 VREFBUF_CSR 寄存器中的 VRS 位进行配置：

- VRS = 0：V_{REF_OUT1} 大约为 2.048 V。
- VRS = 1：V_{REF_OUT2} 大约为 2.5 V。

内部参考电压配置为四种不同模式，具体取决于 ENVR 和 HIZ 位的配置。下表列出了这些模式：

表 146. VREF 缓冲器模式

ENVR	HIZ	VREF 缓冲器配置
0	0	VREFBUF 缓冲器关闭： – V_{REF+} 引脚下拉到 V_{SSA}
0	1	外部参考电压模式（默认值）： – VREFBUF 缓冲器关闭 – V_{REF+} 引脚上拉模式
1	0	内部参考电压模式： – VREFBUF 缓冲器开启 – V_{REF+} 引脚连接到 VREFBUF 缓冲器输出
1	1	保持模式： – VREFBUF 缓冲器关闭 – V_{REF+} 引脚浮空。借助外部电容来保持电压 – 禁止 VRR 检测，VRR 位保持最后一个状态

通过将 VREFBUF_CSR 寄存器中的 ENVR 位置 1 并将 HIZ 位清零使能 VREFBUF 后，用户必须等待至 VRR 位置 1，指示参考电压输出已达到预期值。

a. 最小 V_{DDA} 电压取决于 VRS 设置，请参见产品数据手册。

24.3 VREFBUF 寄存器

24.3.1 VREFBUF 控制和状态寄存器 (VREFBUF_CSR)

VREFBUF control and status register

偏移地址：0x00

复位值：0x0000 0002

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.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	VRR	VRS	HIZ	ENVR
												r	rw	rw	rw

位 31:4 保留，必须保持复位值。

位 3 **VRR**：电压参考缓冲器就绪 (Voltage reference buffer ready)
0：电压参考缓冲器输出未就绪。
1：电压参考缓冲器输出已达到请求值。

位 2 **VRS**：参考电压调节 (Voltage reference scale)
该位选择由电压参考缓冲器生成的值。
0：参考电压设为 V_{REF_OUT1}（约 2.048 V）。
1：参考电压设为 V_{REF_OUT2}（约 2.5 V）。

位 1 **HIZ**：高阻态模式 (High impedance mode)
此位控制模拟开关是否连接到 V_{REF+} 引脚。
0：V_{REF+} 从内部连接到电压参考缓冲器输出。
1：V_{REF+} 引脚为高阻态。
关于 ENVR 位配置及模式说明，请参见*表 146：VREF 缓冲器模式*。

位 0 **ENVR**：电压参考缓冲器模式使能 (Voltage reference buffer mode enable)
此位用于使能电压参考缓冲器模式。
0：禁止内部参考电压模式（外部参考电压模式）。
1：使能内部参考电压模式（参考缓冲器使能或保持模式）。

24.3.2 VREFBUF 校准控制寄存器 (VREFBUF_CCR)

VREFBUF calibration control register

偏移地址：0x04

复位值：0x0000 00XX

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.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TRIM[5:0]					
										rw	rw	rw	rw	rw	rw

位 31:6 保留，必须保持复位值。

位 5:0 **TRIM[5:0]**：微调代码 (Trimming code)

在生产测试期间，使用 Flash 中存储的的微调值进行复位后，这些位将自动初始化。写入这些位可调整内部参考缓冲器电压。

24.3.3 VREFBUF 寄存器映射

下表列出了 VREFBUF 寄存器映射和复位值。

表 147. VREFBUF 寄存器映射和复位值

偏移	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	**VREFBUF_CSR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	VRR	VRS	HIZ	ENVR
	Reset value																													0	0	1	0
0x04	**VREFBUF_CCR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TRIM[5:0]					
	Reset value																											x	x	x	x	x	x

有关寄存器边界地址的信息，请参见*第 78 页的第 2.2.2 节*。

25 比较器 (COMP)

25.1 简介

器件嵌入两个超低功耗比较器 COMP1 和 COMP2

这两个比较器可用于多种功能，包括：

- 在模拟信号的触发下从低功耗模式唤醒
- 模拟信号调理
- 与定时器的 PWM 输出结合使用时，构成逐周期电流控制环路

25.2 COMP 主要特性

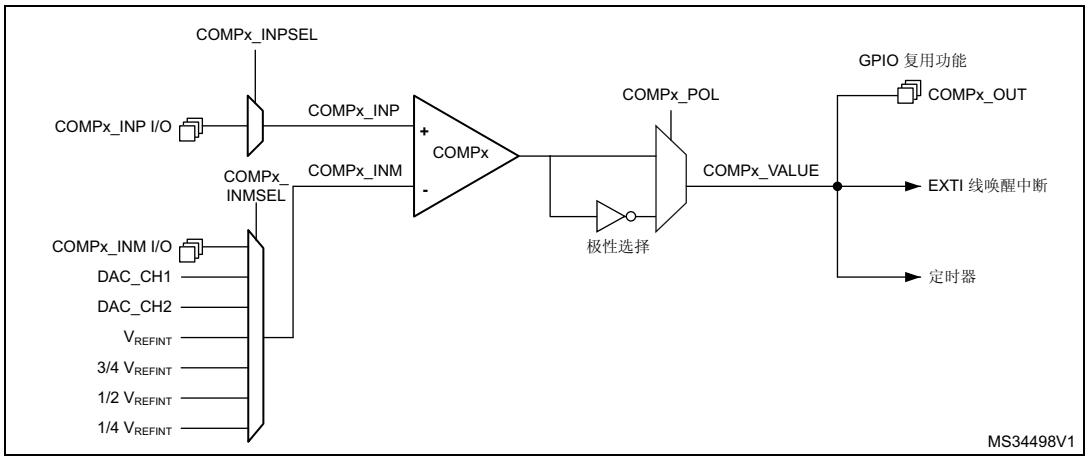
- 每个比较器都具有可配置的正输入和负输入，用于灵活选择电压：
  - 复用 I/O 引脚
  - DAC 通道 1 和通道 2
  - 通过调节器（缓冲分压器）提供的内部参考电压和三个因数分压值（1/4、1/2、3/4）
- 可编程迟滞
- 可编程速度/功耗
- 输出可以重定向到用于触发以下事件的 I/O 或定时器输入：
  - 断路事件（用于快速 PWM 关断）
- 消隐源比较器输出
- 两个比较器可以组合构成窗口比较器
- 每个比较器都可产生中断，用于使器件从睡眠模式和停止模式唤醒（通过 EXTI 控制器）

25.3 COMP 功能说明

25.3.1 COMP 框图

比较器的框图如*图 154：比较器框图*所示。

图 154. 比较器框图



25.3.2 COMP 引脚和内部信号

用作比较器输入的 I/O 必须在 GPIO 寄存器中配置为模拟模式。

比较器输出可以使用数据手册的“复用功能映射”表中给出的复用功能通道连接到 I/O。

输出也可以在内部重定向到用于以下用途的各种定时器输入：

- 使用 BKIN 和 BKIN2 输入紧急关断 PWM 信号
- 使用 OCREF_CLR 输入进行逐周期电流控制
- 用于时序测量的输入捕捉

可以在内部和外部同时对比较器输出进行重定向。

表 148. COMP1 正输入分配

COMP1_INP	COMP1_INPSEL
PC5	0
PB2	1

表 149. COMP1 负输入分配

COMP1_INM	COMP1_INMSEL[2:0]
¼ V_{REFINT}	000
½ V_{REFINT}	001

表 149. COMP1 负输入分配（续）

COMP1_INM	COMP1_INMSEL[2:0]
¾ V_{REFINT}	010
V_{REFINT}	011
DAC 通道 1	100
DAC 通道 2	101
PB1	110
PC4	111

表 150. COMP2 正输入分配

COMP2_INP	COMP2_INPSEL
PB4	0
PB6	1

表 151. COMP2 负输入分配

COMP2_INM	COMP2_INMSEL[2:0]
¼ V_{REFINT}	000
½ V_{REFINT}	001
¾ V_{REFINT}	010
V_{REFINT}	011
DAC 通道 1	100
DAC 通道 2	101
PB3	110
PB7	111

25.3.3 COMP 复位和时钟

时钟控制器提供的 COMP 时钟与 APB2 时钟同步。

RCC 控制器中不单独提供其时钟使能控制位。COMP 和 SYSCFG 共用复位和时钟使能位。

注： ***重要提示：****极性选择逻辑和到端口的输出重定向独立于 APB2 时钟。因此，即使在停止模式下比较器仍能正常工作。*

25.3.4 比较器锁定机制

这两个比较器可用于过流或热保护等安全用途。对于具有特定功能安全要求的应用，必须保证在发生意外寄存器访问或程序计数器损坏时，不能更改比较器编程。

为此，可以对比较器控制和状态寄存器进行写保护（只读）。

一旦编程完成，COMPx LOCK 位便会置 1。这将导致整个寄存器变成只读，包括 COMPx LOCK 位。

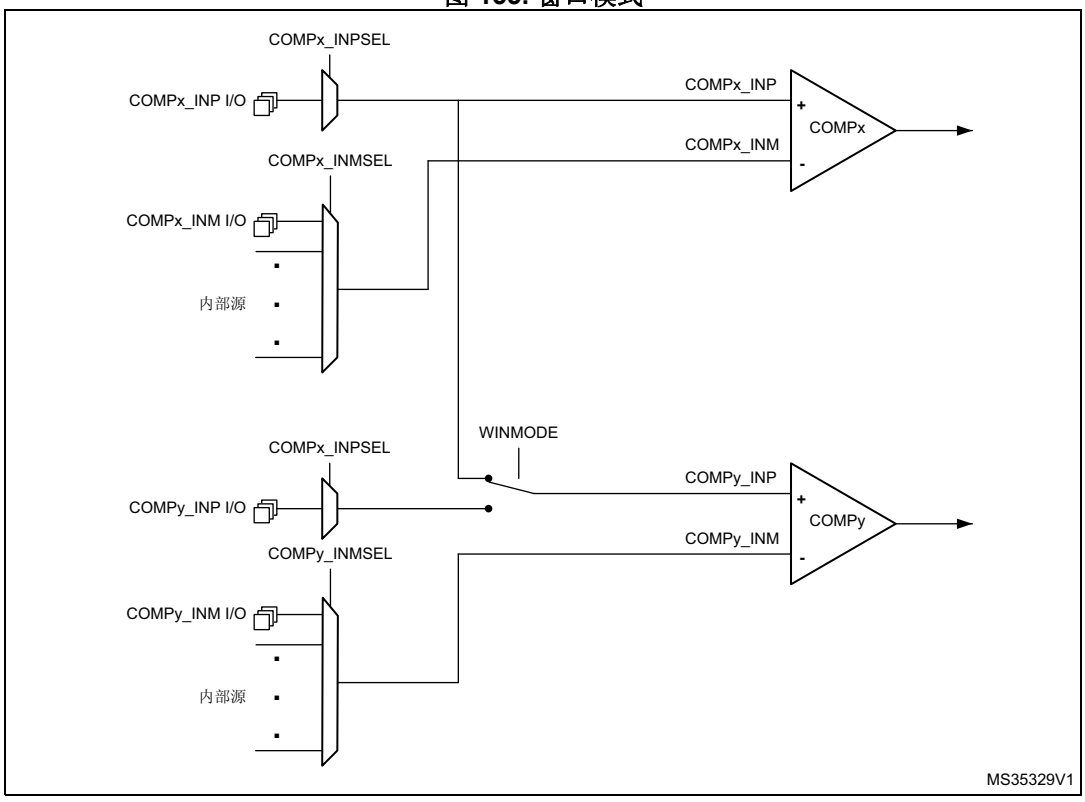
只能通过 MCU 复位来复位写保护。

25.3.5 窗口比较器

窗口比较器用于监视模拟电压是否处于阈值上下限所定义的特定电压范围内。

可使用两个嵌入式比较器创建窗口比较器。受监视的模拟电压连接到连在一起的比较器的非反相 (+) 输入，阈值上下限电压连接到比较器的反相 (-) 输入。可通过使能 WINMODE 位将两个非反相输入在内部连在一起，进而保留一个 IO 用于其他用途。

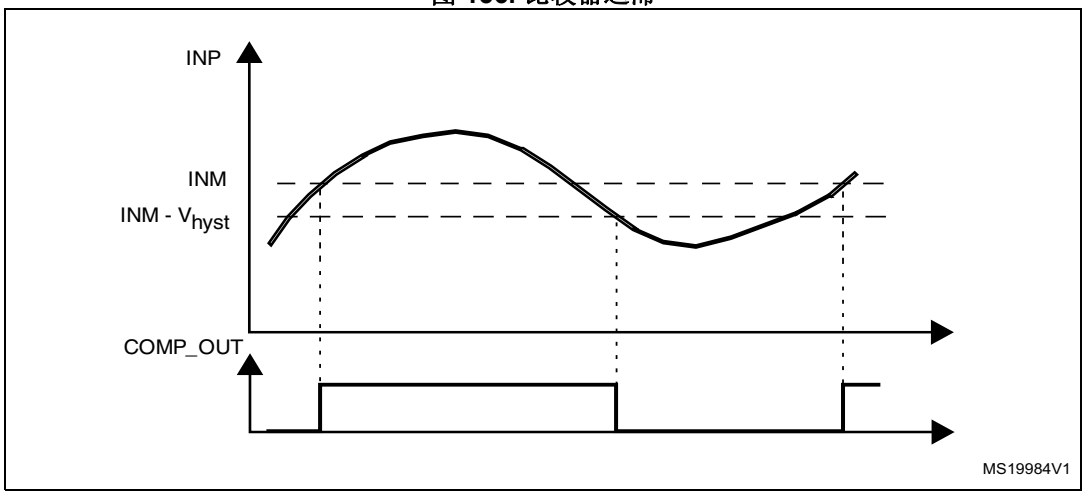
图 155. 窗口模式



25.3.6 迟滞

比较器具有可编程迟滞，可在有噪声信号时避免发生意外输出转换。迟滞可在不需要时（例如，退出低功耗模式时）禁止，以便使用外部组件强制迟滞值。

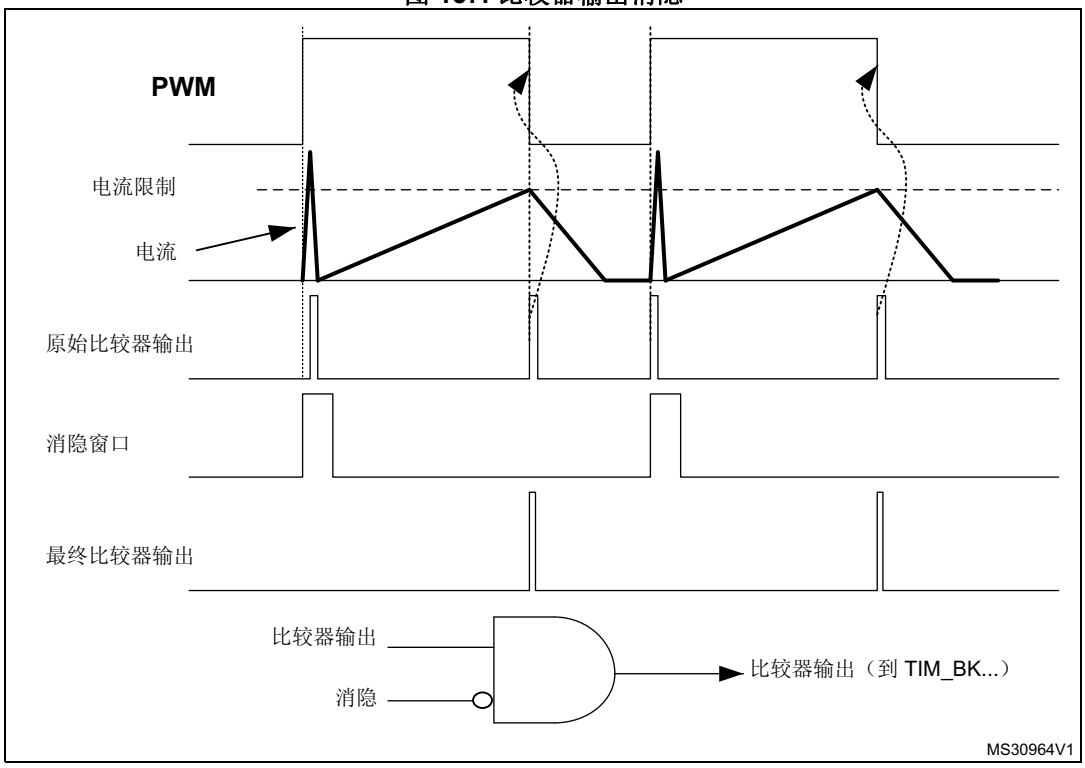
图 156. 比较器迟滞



25.3.7 比较器输出消隐功能

消隐功能的目的是防止电流调节在 PWM 周期开始处出现短暂电流尖峰（通常为功率开关反向并联二极管中的恢复电流）时发生跳闸。其包括选择的消隐窗口，对应定时器比较输出信号。通过软件进行选择（请参见比较器寄存器说明，了解可能的消隐信号）。然后，将消隐信号的补码与比较器输出执行逻辑“与”运算，以提供所需比较器输出。请参见下图所示的示例。

图 157. 比较器输出消隐



25.3.8 COMP 功耗和速度模式

对于给定的应用，可调节 COMP1 和 COMP2 功耗与传播延迟以获得最佳平衡。

COMPx_CSR 寄存器的位 PWRMODE[1:0] 可编程为如下值：

00：高速/全功耗

01 或 10：中速/中等功耗

11：低速/超低功耗

25.4 COMP 低功耗模式

表 152. 低功耗模式下的比较器行为

模式	说明
睡眠	对比较器无影响。 比较器中断可使器件退出睡眠模式。
低功耗运行	无影响。
低功耗睡眠	无影响。COMP 中断可使器件退出低功耗睡眠模式。
停止 0	对比较器无影响。 比较器中断可使器件退出停止模式。
停止 1	
停止 2	
待机	COMP 寄存器掉电，退出待机或关断模式后必须重新初始化。
关断	

25.5 COMP 中断

比较器输出从内部连接到扩展中断和事件控制器。每个比较器都有其各自的 EXTI 线，能够产生中断或事件。该机制还可用于退出低功耗模式。

更多详细信息，请参见中断和事件部分。

要使能 COMPx 中断，需要遵循以下顺序：

1. 将对应于 COMPx 输出事件的 EXTI 线配置为中断模式并将其使能，然后选择上升沿有效、下降沿有效或二者均有效
2. 配置并使能映射到相应 EXTI 线的 NVIC IRQ 通道
3. 使能 COMPx

表 153. 中断控制位

中断事件	事件标志	使能控制位	退出睡眠模式	退出停止模式	退出待机模式
COMP1 输出	COMP1_CSR 中的 VALUE	通过 EXTI	是	是	N/A
COMP2 输出	COMP2_CSR 中的 VALUE	通过 EXTI	是	是	N/A

25.6 COMP 寄存器

25.6.1 比较器 1 控制和状态寄存器 (COMP1_CSR)

Comparator 1 control and status register

COMP1_CSR 为比较器 1 控制/状态寄存器。此寄存器包含与比较器 1 相关的所有位/标志。

偏移地址：0x00

系统复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LOCK	VALUE	Res.	Res.	Res.	Res.	Res.	Res.	SCALEN	BRGEN	Res.	BLANKING			HYST	
rs	r							rw	rw		rw			rw	

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

位 31 **LOCK**：COMP1_CSR 寄存器锁定位 (COMP1_CSR register lock bit)

此位由软件置 1，通过硬件系统复位清零。它将锁定比较器 1 控制寄存器 (COMP1_CSR[31:0]) 的全部内容。

0：比较器 1 的 COMP1_CSR[31:0] 可读/写

1：比较器 1 的 COMP1_CSR[31:0] 只读

位 30 **VALUE**：比较器 1 输出状态位 (Comparator 1 output status bit)

此位为只读。它反映当前比较器 1 输出的状态（受到 POLARITY 位的影响）。

位 29:24 保留，必须保持复位值。

位 23 **SCALEN**：电压调节器使能位 (Voltage scaler enable bit)

此位由软件置 1 和清零。该位使能比较器 1 负输入上的 V_{REFINT} 分压器输出。

0：禁止带隙调节器（COMP2_CSR 寄存器的 SCALEN 位也复位时）

1：使能带隙调节器

位 22 **BRGEN**：调节器电阻桥使能 (Scaler bridge enable)
此位由软件置 1 和清零（仅限 LOCK 未置 1 的情况）。该位可使能调节器电阻桥。
0：禁止调节器电阻桥（COMP2_CSR 寄存器的 BRGEN 位也复位时）
1：使能调节器电阻桥
如果 SCALEN 置 1 且 BRGEN 复位，则 BG 参考电压可用，但 1/4 BGAP、1/2 BGAP 和 3/4 BGAP 不可用。发送 BGAP 值，而非 1/4 BGAP、1/2 BGAP、3/4 BGAP。
如果 SCALEN 和 BRGEN 置 1，则 1/4 BGAP、1/2 BGAP、3/4 BGAP 和 BGAP 参考电压可用。

位 21 保留，必须保持复位值

位 20:18 **BLANKING[2:0]**：比较器 1 消隐源选择位 (Comparator 1 blanking source selection bits)
这些位选择哪个定时器输出控制比较器 1 输出消隐。
000：无消隐
001：选择 TIM1 OC5 作为消隐源
010：选择 TIM2 OC3 作为消隐源
所有其他值：保留

位 17:16 **HYST[1:0]**：比较器 1 迟滞选择位 (Comparator 1 hysteresis selection bits)
这些位由软件置 1 和清零（仅限 LOCK 未置 1 的情况）。他们选择比较器 1 的迟滞电压。
00：无迟滞
01：低迟滞
10：中等迟滞
11：高迟滞

位 15 **POLARITY**：比较器 1 极性选择位 (Comparator 1 polarity selection bit)
此位由软件置 1 和清零（仅限 LOCK 未置 1 的情况）。它使比较器 1 的极性反相。
0：比较器 1 输出值不反相
1：比较器 1 输出值反相

位 14:8 保留，必须保持复位值。

位 7 **INPSEL**：比较器 1 正输入选择位 (Comparator 1 input plus selection bit)
此位由软件置 1 和清零（仅限 LOCK 未置 1 的情况）。
0：外部 IO - PC5
1：PB2

位 6:4 **INMSEL**：比较器 1 负输入选择位 (Comparator 1 input minus selection bits)
这些位由软件置 1 和清零（仅限 LOCK 未置 1 的情况）。他们选择哪个输入连接到比较器 1 的负输入。
000 = 1/4 V_{REFINT}
001 = 1/2 V_{REFINT}
010 = 3/4 V_{REFINT}
011 = V_{REFINT}
100 = DAC 通道 1
101 = DAC 通道 2
110 = PB1111 = PC4

位 3:2 **PWRMODE[1:0]**：比较器 1 的功耗模式 (Power Mode of the comparator 1)
这些位由软件置 1 和清零（仅限 LOCK 未置 1 的情况）。他们控制比较器 1 的功耗/速度。
00： 高速
01 或 10： 中速
11： 超低功耗

位 1 保留，必须保持清零。

位 0 **EN**：比较器 1 使能位 (Comparator 1 enable bit)
此位由软件置 1 和清零（仅限 LOCK 未置 1 的情况）。可开启比较器 1。
0：关闭比较器 1
1：开启比较器 1

25.6.2 比较器 2 控制和状态寄存器 (COMP2_CSR)

Comparator 2 control and status register

COMP2_CSR 为比较器 2 控制/状态寄存器。此寄存器包含与比较器 2 相关的所有位/标志。

偏移地址：0x04

系统复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LOCK	VALUE	Res.	Res.	Res.	Res.	Res.	Res.	SCAL EN	BRG EN	Res.	BLANKING			HYST	
rs	r							rw	rw		rw			rw	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
POLA RITY	Res.	Res.	Res.	Res.	Res.	WIN MODE	Res.	INP SEL.	INMSEL			PWRMODE		Res.	EN
rw						rw		rw	rw			rw			rw

位 31 **LOCK**：CSR 寄存器锁定位 (CSR register lock bit)
此位由软件置 1，通过硬件系统复位清零。它将锁定比较器 2 控制寄存器 (COMP2_CSR[31:0]) 的全部内容。
0：比较器 2 的 COMP2_CSR[31:0] 可读/写
1：比较器 2 的 COMP2_CSR[31:0] 只读

位 30 **VALUE**：比较器 2 输出状态位 (Comparator 2 output status bit)
此位为只读。它反映当前比较器 2 输出的状态（受到 POLARITY 位的影响）。

位 29:24 保留，必须保持复位值

位 23 **SCALEN**：电压调节器使能位 (Voltage scaler enable bit)
此位由软件置 1 和清零。该位使能比较器 2 负输入上的 V_{REFINT} 分压器输出。
0：禁止带隙调节器（COMP1_CSR 寄存器的 SCALEN 位也复位时）
1：使能带隙调节器

位 22 **BRGEN**：调节器电阻桥使能 (Scaler bridge enable)
此位由软件置 1 和清零（仅限 LOCK 未置 1 的情况）。该位可使能调节器电阻桥。
0：禁止调节器电阻桥（COMP1_CSR 寄存器的 BRGEN 位也复位时）
1：使能调节器电阻桥
如果 SCALEN 置 1 且 BRGEN 复位，则 BG 参考电压可用，但 1/4 BGAP、1/2 BGAP 和 3/4 BGAP 不可用。发送 BGAP 值，而非 1/4 BGAP、1/2 BGAP、3/4 BGAP。
如果 SCALEN 和 BRGEN 置 1，则 1/4 BGAP、1/2 BGAP、3/4 BGAP 和 BGAP 参考电压可用。

位 21 保留，必须保持复位值

位 20:18 **BLANKING[2:0]**：比较器 2 消隐源选择位 (Comparator 2 blanking source selection bits)
这些位选择哪个定时器输出控制比较器 2 输出消隐。
000：无消隐
100：选择 TIM15 OC1 作为消隐源
所有其他值：保留

位 17:16 **HYST[1:0]**：比较器 2 迟滞选择位 (Comparator 2 hysteresis selection bits)
这些位由软件置 1 和清零（仅限 LOCK 未置 1 的情况）。选择比较器 2 的迟滞电压。
00：无迟滞
01：低迟滞
10：中等迟滞
11：高迟滞

位 15 **POLARITY**：比较器 2 极性选择位 (Comparator 2 polarity selection bit)
此位由软件置 1 和清零（仅限 LOCK 未置 1 的情况）。它使比较器 2 的极性反相。
0：比较器 2 输出值不反相
1：比较器 2 输出值反相

位 14:10 保留，必须保持复位值。

位 9 **WINMODE**：窗口模式选择位 (Windows mode selection bit)
此位由软件置 1 和清零（仅限 LOCK 未置 1 的情况）。该位选择比较器的窗口模式。如果该位置 1，则比较器的两个正输入将连接在一起。
0：比较器 2 的正输入不连接到比较器 1
1：比较器 2 的正输入连接到比较器 1 的正输入

位 8 保留，必须保持复位值。

位 7 **INPSEL**：比较器 2 正输入选择位 (Comparator 2 input plus selection bit)
此位由软件置 1 和清零（仅限 LOCK 未置 1 的情况）。
0：PB4
1：PB6

位 6:4 **INMSEL**：比较器 2 负输入选择位 (Comparator 2 input minus selection bits)
这些位由软件置 1 和清零（仅限 LOCK 未置 1 的情况）。他们选择哪个输入连接到比较器 2 的负输入。
000 = 1/4 V_{REFINT}
001 = 1/2 V_{REFINT}
010 = 3/4 V_{REFINT}
011 = V_{REFINT}
100 = DAC 通道 1
101 = DAC 通道 2
110 = PB3
111 = PB7

位 3:2 **PWRMODE[1:0]**：比较器 2 的功耗模式 (Power Mode of the comparator 2)
这些位由软件置 1 和清零（仅限 LOCK 未置 1 的情况）。他们控制比较器 2 的功耗/速度。
00： 高速
01 或 10： 中速
11： 超低功耗

位 1 保留，必须保持清零。

位 0 **EN**：比较器 2 使能位 (Comparator 2 enable bit)
此位由软件置 1 和清零（仅限 LOCK 未置 1 的情况）。可开启比较器 2。
0：关闭比较器 2
1：开启比较器 2

25.6.3 COMP 寄存器映射

下表对比较器寄存器进行了汇总。

表 154. COMP 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	**COMP1_CSR**	LOCK	VALUE	Res.	Res.	Res.	Res.	Res.	Res.	SCALEN	BRGEN.	Res.	BLANKING			HYST		POLARITY.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	INPSEL.	INMSEL			PWRMODE		Res.	EN
	Reset value	0	0							0	0		0	0	0	0	0	0								0	0	0	0	0	0		0
0x04	**COMP2_CSR**	LOCK	VALUE	Res.	Res.	Res.	Res.	Res.	Res.	SCALEN	BRGEN	Res.	BLANKING			HYST		POLARITY.	Res.	Res.	Res.	Res.	Res.	WINMODE	Res.	INPSEL	INMSEL			PWRMODE		Res.	EN
	Reset value	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*第 78 页的第 2.2.2 节*。

26 运算放大器 (OPAMP)

26.1 简介

器件内置两个运算放大器，每个运算放大器具有两个输入和一个输出。这三个 I/O 可连接到外部引脚，因此能实现任意类型的外部互连。可以将运算放大器在内部配置为跟随器或放大器，放大器具有从 2 到 16 的非反相增益。

正输入可连接到内部 DAC。

输出可连接到内部 ADC。

26.2 OPAMP 主要特性

- 轨到轨输入和输出电压范围
- 低输入失调电流（低至 1 nA）
- 低输入失调电压（校准后为 1.5 mV，出厂校准时为 3 mV）
- 低功耗模式（电流损耗降到 30 µA 而不是 100 µA）
- 快速唤醒时间（正常模式下为 10 µs，低功耗模式下为 30 µs）
- 增益带宽为 1.6 MHz

26.3 OPAMP 功能说明

OPAMP 有多种模式。

每个 OPAMP 均可单独使能，如果被禁止，则输出呈高阻态。

使能后，其既可以处于校准模式，此模式下 OPAMP 的所有输入都断开，也可以处于功能模式。

有两种功能模式，低功耗模式和正常模式。在功能模式下，OPAMP 的输入和输出的连接如*第 26.3.3 节：信号走线*中所述。

26.3.1 OPAMP 复位和时钟

访问寄存器时，需要运算放大器时钟。如果应用程序无需对这些寄存器进行读或写访问，则可使用外设时钟使能寄存器关闭此时钟（请参见*第 6.4.19 节：APB1 外设时钟使能寄存器 1 (RCC_APB1ENR1)* 中的 OPAMPEN 位）。

OPAEN 位使能和禁止 OPAMP 工作。应先更改 OPAMP 寄存器配置，然后再使能 OPAEN 位，以避免对输出产生寄生效应。

如果不再需要运算放大器的输出，则可禁止运算放大器以节能。禁止 OPAMP 时，会保持之前设置的所有配置（包括校准）不变。

26.3.2 初始配置

运算放大器默认配置为功能模式，在该模式下，三个 IO 连接到外部引脚。在默认模式下，运算放大器使用出厂微调值。有关出厂微调的条件，请参见数据手册的电气特性部分，通常温度为 30 °C，电压为 3 V。可对微调值进行调节，有关更改微调值的信息，请参见*第 26.3.5 节：校准*。默认配置使用正常模式，该模式提供最高性能。可以将 OPALPM 位置 1，从而将运算放大器切换到低功耗模式并降低性能。数据手册的“电气特性”部分对正常和低功耗模式特性进行了定义。使用前，如果 V_{DDA} 高于 2.4V，则 OPAMP_CSR 的 OPA_RANGE 位必须置 1，否则保持为 0。

一旦 OPAMP_CSR 寄存器中的 OPAEN 位置 1，运算放大器即可工作。两个输入引脚和一个输出引脚可按照*第 26.3.3 节：信号走线*中的定义进行连接，并且可对默认连接设置进行更改。

注： *必须在相应 GPIOx_MODER 寄存器中将输入和输出引脚配置为模拟模式（默认状态）。*

26.3.3 信号走线

运算放大器引脚走线由 OPAMP_CSR 寄存器确定。

下表介绍了两个运算放大器（OPAMP1 和 OPAMP2）的引脚连接。

表 155. 运算放大器连接情形

信号	引脚	内部	注释
OPAMP1_VINM	PA1 或专用引脚(1)	OPAMP1_OUT 或 PGA	通过 OPAMODE 和 VM_SEL 位控制。
OPAMP1_VINP	PA0	DAC1_OUT1	通过 VP_SEL 位控制。
OPAMP1_VOUT	PA3	ADC1_IN8	在 OPAMP 使能时连接该引脚。ADC 输入通过 ADC 控制。
OPAMP2_VINM	PA7 或专用引脚(1)	OPAMP2_OUT 或 PGA	通过 OPAMODE 和 VM_SEL 位控制。
OPAMP2_VINP	PA6	DAC1_OUT2	通过 VP_SEL 位控制
OPAMP2_VOUT	PB0	ADC1_IN15	在 OPAMP 使能时连接该引脚。ADC 输入通过 ADC 控制。

1. 仅 BGA132 和 BGA169 封装提供专用引脚。本配置提供最低输入偏置电流（参见数据手册）。

26.3.4 OPAMP 模式

运算放大器的输入和输出都可在引脚上访问。放大器可用于多种配置环境：

- 独立模式（外部增益设置模式）
- 跟随器配置模式
- PGA 模式

注： *放大器输出引脚直接连接到输出引脚的焊盘，以最大限度地降低输出阻抗。即使放大器被配置为 PGA 并仅连接到 ADC 通道，放大器输出引脚也不能用作通用 I/O。*

注： *输入信号的阻抗必须保持在不能使输入泄漏产生明显伪信号（由于源极阻降而引起）的水平以下。有关更多详细信息，请参见数据手册的电气特性部分。*

独立模式（外部增益设置模式）

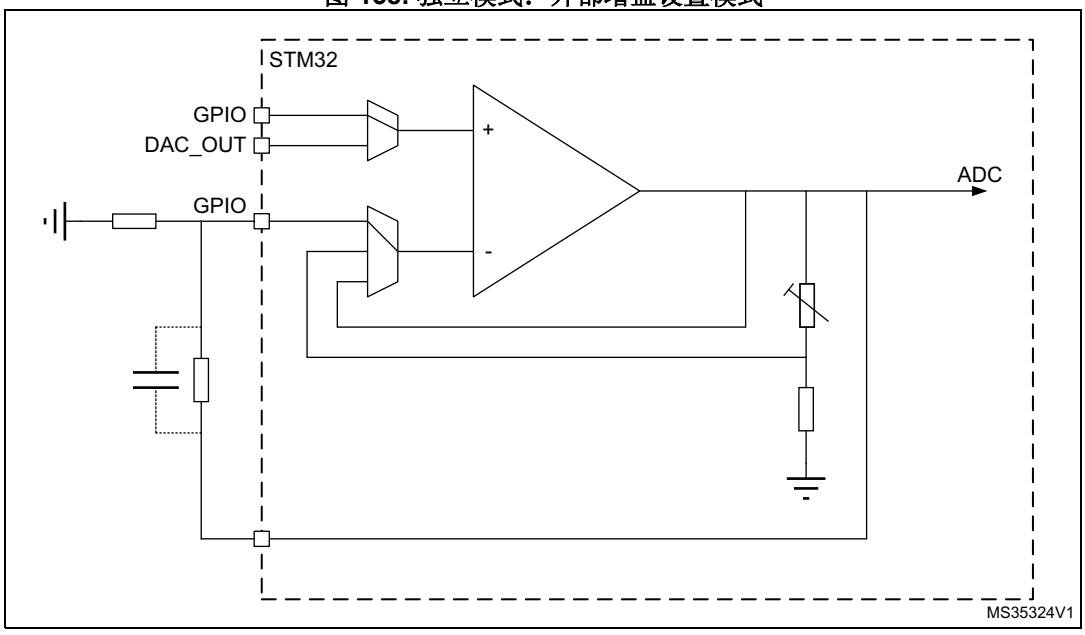
下面介绍在独立模式下使用 OPAMP 的程序。

从 OPAMP_CSR 的默认值和 GPIOx_MODER 的默认状态开始，根据 V_{DDA} 电压配置 OPA_RANGE 位。OPAEN 位置 1 后，两个输入引脚和一个输出引脚会连接到运算放大器。

此默认配置使用工厂微调值，并在正常模式下运行（达到最高性能）。可按如下方式更改 OPAMP 的行为：

- 可以将 OPALPM 设置为“运算放大器低功耗”模式，以节省电能。
- 可以将 USERTRIM 置 1，以修改输入偏移的微调值。

图 158. 独立模式：外部增益设置模式



跟随器配置模式

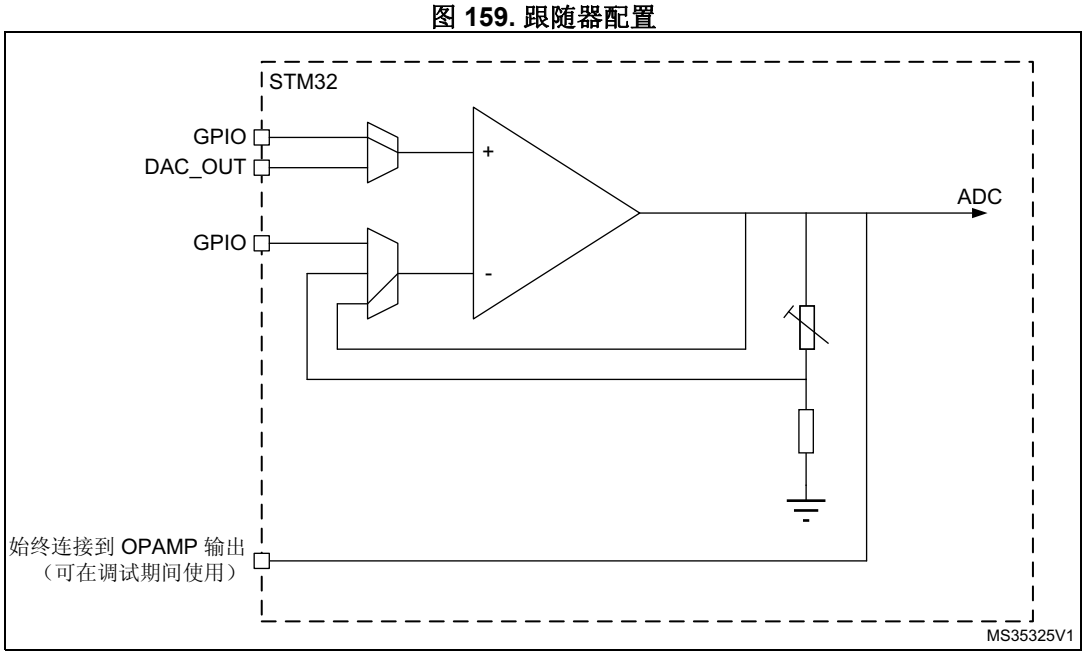
下面介绍在跟随器模式下使用 OPAMP 的过程。

- 将 OPAMODE 位配置为“内部跟随器”。
- 将 VP_SEL 位配置为“GPIO 连接到 VINP”。
- OPAEN 位置 1 后，OPAMP_VINP 引脚上的信号就会被复制到 OPAMP_VOUT 引脚。

注： *对应于 OPAMP_VINM 的引脚可供另外使用。*

注： *运算放大器输出上的信号也视为 ADC 输入。因此，如果输入信号频率与运算放大器增益带宽积规范相符，则配置为跟随器模式的 OPAMP 可用于对输入信号进行阻抗调整，然后再将这些信号馈送至 ADC 输入。*

图 159. 跟随器配置



可编程增益放大器模式

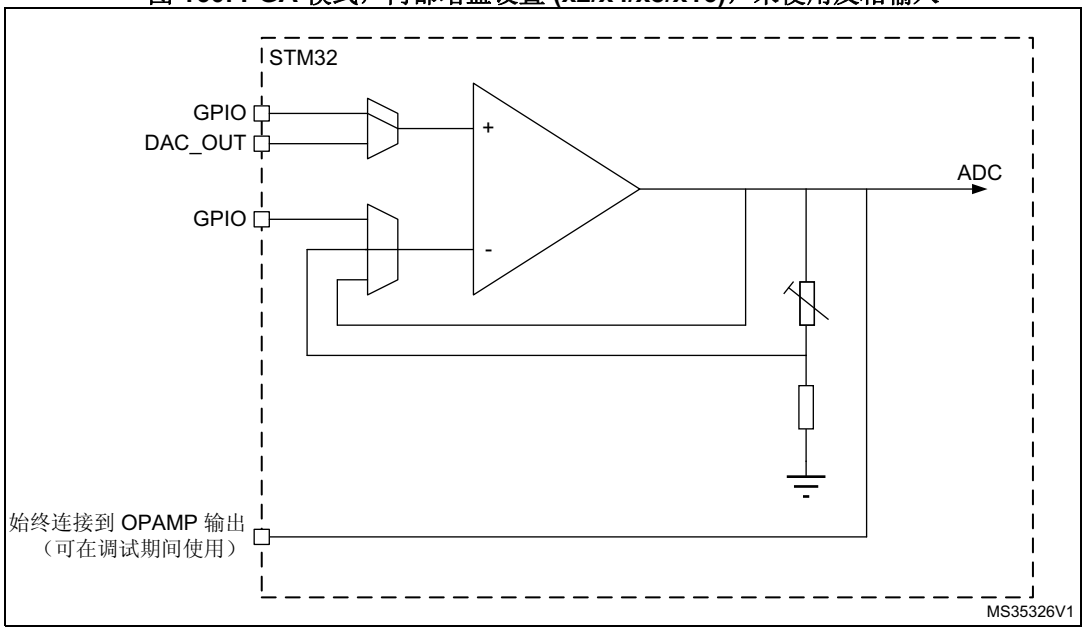
下面介绍使用 OPAMP 来放大输入信号幅值的过程。

- 将 OPAMODE 位配置为“内部 PGA 使能”
- 将 PGA_GAIN 位配置为“内部 PGA 增益 2、4、8 或 16”
- 将 VM_SEL 位配置为“未在外部连接反相输入”
- 将 VP_SEL 位配置为“GPIO 连接到 VINP”

一旦 OPAEN 位置 1，OPAMP_VINP 引脚上的信号就会放大所选增益的倍数，并可在 OPAMP_VOUT 引脚上测得放大后的电压。

注： *为避免饱和，输入电压应保持低于：V_{DDA} 除以所选增益。*

图 160. PGA 模式，内部增益设置 (x2/x4/x8/x16)，未使用反相输入



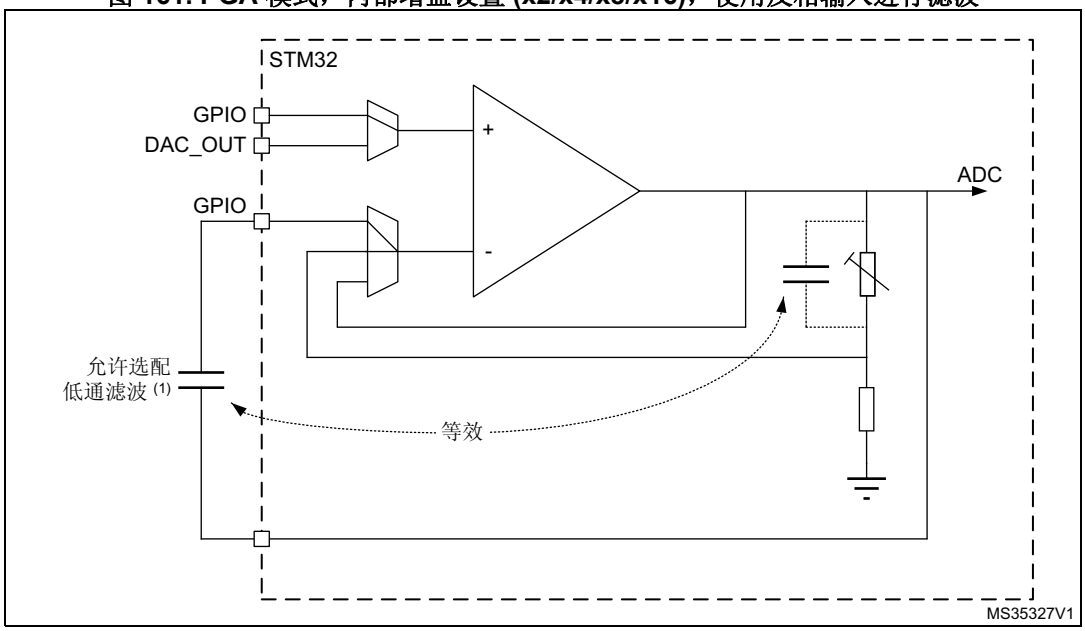
使用外部滤波的可编程增益放大器模式

下面介绍使用 OPAMP 和外部滤波来放大输入信号幅值的步骤。

- 将 OPAMODE 位配置为“内部 PGA 使能”
- 将 PGA_GAIN 位配置为“内部 PGA 增益 2、4、8 或 16”
- 将 VM_SEL 位配置为“GPIO 连接到 VINM”
- 将 VP_SEL 位配置为“GPIO 连接到 VINP”

VINP 上的任何外部连接均可与内部 PGA 并联使用，例如，可在 VOUT 和 VINM 之间连接一个电容来实现滤波目的（有关 PGA 电阻网络中使用的电阻值，请参见数据手册）。

图 161. PGA 模式，内部增益设置 (x2/x4/x8/x16)，使用反相输入进行滤波



1. 增益取决于截止频率。

26.3.5 校准

启动时，使用预设的“出厂”微调值初始化了微调值。

每个运算放大器偏移均可由用户进行微调。对于正常模式和低功耗模式，特定寄存器允许有不同的微调值。

校准的目的在于尽可能地减小 OPAMP 输入失调电压。在稳定的电压和温度条件下，校准电路可将输入失调电压减至 +/-1.5 mV 以下。

对于每个运算放大器和每种模式，需要对两个微调值进行微调，一个值用于 N 差分对，另一个值用于 P 差分对。

对于每个运算放大器，有两个寄存器可用于微调偏移值，一个用于正常模式 (OPAMP_OTR)，另一个用于低功耗模式 (OPAMP_LPOTR)。每个寄存器都包含 5 个用于 P 差分对微调的位和 5 个用于 N 差分对微调的位。这些均为“用户”微调的值。

用户可使用 OPAMP_CSR 寄存器的 USERTRIM 位从“出厂”值切换到“用户”微调的值。该位在启动时复位，因此默认情况下，“出厂”值被应用于 OPAMP 微调寄存器。

用户可在校准或功能模式下轻松更改微调值。

通常在将 CALON 位设置为 1 来初始化校准操作之后，才配置偏移微调寄存器。当 CALON = 1 时，运算放大器的输入从工作环境中断开。

- 将 CALSEL 置为 1 可初始化 P 差分对的偏移校准（使用低参考电压）。
- 将 CALSEL 复位为 0 可初始化 N 差分对的偏移校准（使用高参考电压）。

CALON = 1 时，CALOUT 位将反映 CALSEL 和 OPALPM 所选择的微调值的影响。CALOUT 的值在两个连续微调值之间切换时，表示这两个值均是最佳微调值。在更改微调值之后，CALOUT 标志最多需要 1 ms 的时间达到稳定状态（请参见数据手册电气特性部分的 $t_{OFFTRIM}$max 延迟规范）。

注： *微调值越接近于最佳微调值，稳定所用的时间就越长（但在任何情况下，最长稳定时间都低于 1 ms）。*

表 156. 工作模式和校准

模式	控制位				输出	
	OPAEN	OPALPM	CALON	CALSEL	V_{OUT}	CALOUT 标志
正常工作模式	1	0	0	X	模拟	0
低功耗模式	1	1	0	X	模拟	0
掉电	0	X	X	X	Z	0
正常模式的偏移校准高值	1	0	1	0	模拟	X
正常模式的偏移校准低值	1	0	1	1	模拟	X
低功耗模式的偏移校准高值	1	1	1	0	模拟	X
低功耗模式的偏移校准低值	1	1	1	1	模拟	X

校准过程

以下为对任一运算放大器执行完全校准的步骤：

1. 选择 OPAMP_CSR 中的合适 OPA_RANGE，然后将 OPAMP_CSR 中的 OPAEN 位置 1，以使能运算放大器。
2. 将 OPAMP_CSR 寄存器中的 USERTRIM 位置 1。
3. 选择一种校准模式（请参见*表 156：工作模式和校准*）。步骤 3 到 4 必须重复 4 次。首先我们选择
   - 正常模式，偏移校准高值（N 差分对）

   上述校准模式对应于 OPAMP_CSR 寄存器中的 OPALPM = 0，CALSEL = 0。
4. 从 00000b 开始递增 OPAMP_OTR 的 TRIMOFFSETN[4:0]，直至 OPAMP_CSR 的 CALOUT 变为 1。

注： *对于偏移校准高值，CALOUT 从 0 变为 1；对于偏移校准低值，则从 1 变为 0。*

注： *在写入 OPAMP_OTR 寄存器和读取 CALOUT 值之间，确保等待数据手册电气特性部分中指定的 $t_{OFFTRIM}max$ 延迟，以获得正确的 CALOUT 值。*

CALOUT 反转即表示，偏移已正确补偿，必须在 OPAMP_OTR 寄存器中保存相应的微调值。

针对以下模式重复步骤 3 到 4：

- 正常模式和偏移校准低值
- 低功耗模式和偏移校准高值
- 低功耗模式和偏移校准低值

如果未使用某种模式，则无需执行相应的校准。

所有运算放大器均可同时校准。

注： *在整个校准阶段，运算放大器输出的外部连接不能上拉或下拉高于 500 µA 的电流。*

校准过程中，须将 OPAMODE 位置为 00、01（禁止 PGA）或 11（内部跟随器）。

26.4 OPAMP 低功耗模式

表 157. 低功耗模式对 OPAMP 的影响

模式	说明
睡眠	无影响。
低功耗运行	无影响。
低功耗睡眠	无影响。
停止 0/停止 1	无影响，OPAMP 寄存器内容保持不变。
停止 2	OPAMP 寄存器内容仍被保持。OPAMP 必须在进入停止 2 模式前禁止。
待机	OPAMP 寄存器掉电，退出待机或关断模式后必须重新初始化。
关断	

26.5 OPAMP 寄存器

26.5.1 OPAMP1 控制/状态寄存器 (OPAMP1_CSR)

OPAMP1 control/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
OPA_RANGE	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
rw															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CAL OUT	USER TRIM	CAL SEL	CALON	Res.	VP_SEL	VM_SEL		Res.	Res.	PGA_GAIN		OPAMODE		OPA LPM	OPAEN
r	rw	rw	rw		rw	rw	rw			rw	rw	rw	w	rw	rw

位 31 **OPA_RANGE**：运算放大器电源的稳定范围 (Operational amplifier power supply range for stability)
为进行 OPA_RANGE 位写操作，所有 OPA 必须处于掉电模式。这适用于产品中嵌入的全部 OPA。
0：低范围 (VDDA < 2.4V)
1：高范围 (VDDA > 2.4V)

位 30:16 保留，必须保持复位值。

位 15 **CALOUT**：运算放大器校准输出 (Operational amplifier calibration output)
校准模式期间，此信号翻转时，微调偏移。

位 14 **USERTRIM**：用于从“出厂”OPA 偏移微调值切换到“用户”OPA 偏移微调值
对于正常模式和低功耗模式，该位均有效。
0：使用“出厂”微调代码
1：使用“用户”微调代码

位 13 **CALSEL**：校准选择 (Calibration selection)
0：NMOS 校准（对 OPAMP 输入施加 200 mV 电压）
1：PMOS 校准（对 OPAMP 输入施加 VDDA-200 mV 电压）

位 12 **CALON**：校准模式使能 (Calibration mode enabled)
0：正常模式
1：校准模式（通过硬件打开所有开关）

位 11 保留，必须保持复位值。

位 10 **VP_SEL**：非反相输入选择 (Non inverted input selection)
0：GPIO 连接到 VINP
1：DAC 连接到 VINP

位 9:8 **VM_SEL**：反相输入选择 (Inverting input selection)
只有在 OPAMODE = 00、01 或 10 时才会使用这些位。
00：连接到 VINM 的 GPIO（在 PGA 模式下也可用于滤波）
01：专用低漏型输入（仅 BGA132 提供），连接到 VINM（在 PGA 模式下也可用于滤波）
1x：未在外部连接的反相输入。仅在 OPAMODE = 10 时（PGA 模式），这些配置才有效

位 7:6 保留，必须保持复位值。

位 5:4 **PGA_GAIN**：运算放大器可编程放大器增益值 (Operational amplifier Programmable amplifier gain value)
00：内部 PGA 增益 2
01：内部 PGA 增益 4
10：内部 PGA 增益 8
11：内部 PGA 增益 16

位 3:2 **OPAMODE**：运算放大器 PGA 模式 (Operational amplifier PGA mode)
00：禁止内部 PGA
01：禁止内部 PGA
10：使能内部 PGA，增益在 PGA_GAIN 中编程
11：内部跟随器

位 1 **OPALPM**：运算放大器低功耗模式 (Operational amplifier Low Power Mode)
要更改此配置，必须禁止运算放大器。
0：运算放大器处于正常模式
1：运算放大器处于低功耗模式

位 0 **OPAEN**：运算放大器使能 (Operational amplifier Enable)
0：运算放大器已禁止
1：运算放大器已使能

26.5.2 正常模式下的 OPAMP1 偏移微调寄存器 (OPAMP1_OTR)

OPAMP1 offset trimming register in normal mode

偏移地址：0x04

复位值：0x0000 XXXX（出厂微调值）

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.	Res.	TRIMOFFSETP[4:0]					Res.	Res.	Res.	TRIMOFFSETN[4:0]				
			rw	rw	rw	rw	rw				rw	rw	rw	rw	rw

位 31:13 保留，必须保持复位值。

位 12:8 **TRIMOFFSETP[4:0]**：PMOS 差分对的微调 (Trim for PMOS differential pairs)

位 7:5 保留，必须保持复位值。

位 4:0 **TRIMOFFSETN[4:0]**：NMOS 差分对的微调 (Trim for NMOS differential pairs)

26.5.3 低功耗模式下的 OPAMP1 偏移微调寄存器 (OPAMP1_LPOTR)

OPAMP1 offset trimming register in low-power mode

偏移地址：0x08

复位值：0x0000 XXXX（出厂微调值）

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.	Res.	TRIMLPOFFSETP[4:0]					Res.	Res.	Res.	TRIMLPOFFSETN[4:0]				
			rw	rw	rw	rw	rw				rw	rw	rw	rw	rw

位 31:13 保留，必须保持复位值。

位 12:8 **TRIMLPOFFSETP[4:0]**：PMOS 差分对的低功耗模式微调 (Low-power mode trim for PMOS differential pairs)

位 7:5 保留，必须保持复位值。

位 4:0 **TRIMLPOFFSETN[4:0]**：NMOS 差分对的低功耗模式微调 (Low-power mode trim for NMOS differential pairs)

26.5.4 OPAMP2 控制/状态寄存器 (OPAMP2_CRS)

OPAMP2 control/status 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
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
CALOUT	USERTRIM	CALSEL	CALON	Res.	VP_SEL	VM_SEL		Res.	Res.	PGA_GAIN		OPAMODE		OPALPM	OPAEN
r	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 **CALOUT**：运算放大器校准输出 (Operational amplifier calibration output)
校准模式期间，此信号翻转时，微调偏移。

位 14 **USERTRIM**：用于从“出厂”OPA 偏移微调值切换到“用户”OPA 偏移微调值
对于正常模式和低功耗模式，该位均有效。
0：使用“出厂”微调代码
1：使用“用户”微调代码

位 13 **CALSEL**：校准选择 (Calibration selection)
0：NMOS 校准（对 OPAMP 输入施加 200 mV 电压）
1：PMOS 校准（对 OPAMP 输入施加 VDDA-200 mV 电压）

位 12 **CALON**：校准模式使能 (Calibration mode enabled)
0：正常模式
1：校准模式（通过硬件打开所有开关）

位 11 保留，必须保持复位值。

位 10 **VP_SEL**：非反相输入选择 (Non inverted input selection)
0：GPIO 连接到 VINP
1：DAC 连接到 VINP

位 9:8 **VM_SEL**：反相输入选择 (Inverting input selection)
只有在 OPAMODE = 00、01 或 10 时才会使用这些位。
00：连接到 VINM 的 GPIO（在 PGA 模式下也可用于滤波）
01：专用低漏型输入，（仅 BGA132 提供）连接到 VINM（在 PGA 模式下也可用于滤波）
1x：未在外部连接的反相输入。仅在 OPAMODE = 10 时（PGA 模式），这些配置才有效。

位 7:6 保留，必须保持复位值。

位 5:4 **PGA_GAIN**：运算放大器可编程放大器增益值 (Operational amplifier Programmable amplifier gain value)
00：内部 PGA 增益 2
01：内部 PGA 增益 4
10：内部 PGA 增益 8
11：内部 PGA 增益 16

位 3:2 **OPAMODE**：运算放大器 PGA 模式 (Operational amplifier PGA mode)
00：禁止内部 PGA
01：禁止内部 PGA
10：使能内部 PGA，增益在 PGA_GAIN 中编程
11：内部跟随器

位 1 **OPALPM**：运算放大器低功耗模式 (Operational amplifier Low Power Mode)
要更改此配置，必须禁止运算放大器。
0：运算放大器处于正常模式
1：运算放大器处于低功耗模式

位 0 **OPAEN**：运算放大器使能 (Operational amplifier Enable)
0：运算放大器已禁止
1：运算放大器已使能

26.5.5 正常模式下的 OPAMP2 偏移微调寄存器 (OPAMP2_OTR)

OPAMP2 offset trimming register in normal mode

偏移地址：0x14

复位值：0x0000 XXXX（出厂微调值）

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.	Res.	TRIMOFFSETP[4:0]					Res.	Res.	Res.	TRIMOFFSETN[4:0]				
			rw	rw	rw	rw	rw				rw	rw	rw	rw	rw

位 31:13 保留，必须保持复位值。

位 12:8 **TRIMOFFSETP[4:0]**：PMOS 差分对的微调 (Trim for PMOS differential pairs)

位 7:5 保留，必须保持复位值。

位 4:0 **TRIMOFFSETN[4:0]**：NMOS 差分对的微调 (Trim for NMOS differential pairs)

26.5.6 低功耗模式下的 OPAMP2 偏移微调寄存器 (OPAMP2_LPOTR)

OPAMP2 offset trimming register in low-power mode

偏移地址：0x18

复位值：0x0000 XXXX（出厂微调值）

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.	Res.	TRIMLPOFFSETP[4:0]					Res.	Res.	Res.	TRIMLPOFFSETN[4:0]				
			rw	rw	rw	rw	rw				rw	rw	rw	rw	rw

位 31:13 保留，必须保持复位值。

位 12:8 **TRIMLPOFFSETP[4:0]**：PMOS 差分对的低功耗模式微调 (Low-power mode trim for PMOS differential pairs)

位 7:5 保留，必须保持复位值。

位 4:0 **TRIMLPOFFSETN[4:0]**：NMOS 差分对的低功耗模式微调 (Low-power mode trim for NMOS differential pairs)

26.5.7 OPAMP 寄存器映射

表 158. OPAMP 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	**OPAMP1_CSR**	OPA_RANGE	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CALOUT	USERTRIM	CALSEL	CALON	Res.	VP_SEL	VM_SEL		Res.	Res.	PGA_GAIN		OPAMODE		OPALPM	OPAEN
	Reset value	0																0	0	0	0		0	0	0			0	0	0	0	0	0
0x04	**OPAMP1_OTR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TRIM OFFSETP[4:0]					Res.	Res.	Res.	TRIM OFFSETN[4:0]				
	Reset value																				(1)								(1)				
0x08	**OPAMP1_ LPOTR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TRIMLP OFFSETP[4:0]					Res.	Res.	Res.	TRIMLP OFFSETN[4:0]				
	Reset value																				(1)								(1)				
0x10	**OPAMP2_CSR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CALOUT	USERTRIM	CALSEL	CALON	Res.	VP_SEL	VM_SEL		Res.	Res.	PGA_GAIN		OPAMODE		OPALPM	OPAEN
	Reset value																	0	0	0	0		0	0	0			0	0	0	0	0	0
0x14	**OPAMP2_OTR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TRIM OFFSETP[4:0]					Res.	Res.	Res.	TRIM OFFSETN[4:0]				
	Reset value																				(1)								(1)				
0x18	**OPAMP2_LPO TR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TRIMLP OFFSETP[4:0]					Res.	Res.	Res.	TRIMLP OFFSETN[4:0]				
	Reset value																				(1)								(1)				

1. 出厂微调值。

有关寄存器边界地址的信息，请参见*第 2.2.2 节：存储器映射和寄存器边界地址*。

27 数字滤波器，用于ΣΔ调制器 (DFSDM)

27.1 简介

ΣΔ 调制器数字滤波器 (DFSDM) 是一种高性能模块，专用于将外部 ΣΔ 调制器连接到微控制器。它具有多达 8 个外部数字串行接口（通道）和多达 4 个数字滤波器，具有灵活的 ΣΔ 流数字处理选项，可提供高达 24 位 ADC 最终分辨率。DFSDM 还特有并行数据流输入，输入可选择来自内部 ADC 外设或来自微控制器存储器。

外部 ΣΔ 调制器将其模拟输入的模拟值转换为数字数据流。此数字数据流通过串行接口发送到 DFSDM 输入通道。DFSDM 支持多种标准来连接各种 ΣΔ 调制器输出：SPI 接口和曼彻斯特编码单线接口（都具有可调参数）。DFSDM 模块支持连接多达 8 个复用输入数字串行通道，这些通道可与多达 4 个 DFSDM 模块共享。DFSDM 模块还支持来自多达 8 个内部 16 位数据通道（可选择来自内部 ADC 或来自微控制器存储器）的并行数据输入。

DFSDM 将输入数据流转换为最终数字数据字，该数字字表示 ΣΔ 调制器模拟输入上的模拟输入值。该转换基于以下可配置的数字处理而完成：对输入串行数据流进行数字滤波和抽取。

转换速度和分辨率可根据以下可配置的数字处理参数进行调整：滤波器类型、滤波器阶数、滤波器长度和积分器长度。最大输出数据分辨率高达 24 位。有两种转换模式：单次转换模式和连续转换模式。数据可通过 DMA 自动存储在系统 RAM 缓冲区中，因此降低了软件开销。

可使用灵活的定时器触发系统来控制启动 DFSDM 转换。此定时控制可触发即时转换或延时转换，延时时间可以编程。

DFSDM 具有模拟看门狗功能。模拟看门狗可以分配给任何输入通道数据流或最终输出数据。模拟看门狗本身可以对输入数据流进行数字滤波，以达到被监视数据的所需速度和分辨率。

为了检测控制应用中的短路情况，提供了一个短路检测器。该模块监视各个输入通道数据流是否在定义的持续时间内保持稳定（即输入数据流中有若干 0 或 1）。

极值检测器模块监视最终输出数据并存储最大和最小输出数据值。存储的极值可通过软件重新启动。

支持两种功率模式：正常模式和停止模式。

27.2 DFSDM主要特性

- 多达 8 个复用输入数字串行通道：
  - 可配置的 SPI 接口，用于连接各种 ΣΔ 调制器
  - 支持可配置的曼彻斯特编码单线接口
  - 用于 ΣΔ 调制器的时钟输出
- 来自多达 8 个内部数字并行通道的可选输入：
  - 分辨率高达 16 位的输入
  - 内部源：ADC 数据或存储器（CPU/DMA 写）数据流
- 可调节的数字信号处理：
  - $Sinc^x$ 滤波器：滤波器阶数/类型 (1..5)，过采样率（高达 1..1024）
  - 积分器：过采样率 (1..256)
- 高达 24 位的输出数据分辨率：
  - 最终数据的右移位器（0..31 位）
- 有符号输出数据格式
- 自动数据偏移校正（偏移值由用户存储在寄存器中）
- 连续或单次转换
- 可通过以下途径同步启动转换：
  - 软件触发
  - 内部定时器
  - 外部事件
  - 使用第一个 DFSDM 滤波器 (DFSDM_FLT0) 同步启动转换
- 模拟看门狗功能：
  - 下限和上限数据阈值寄存器
  - 自带可配置的 $Sinc^x$ 数字滤波器（阶数 = 1..3，过采样率 = 1..32）
  - 输入来自输出数据寄存器或来自一个或多个输入数字串行通道
  - 独立于标准转换的连续监视
- 短路检测器，用于检测饱和的模拟输入值（下限和上限）：
  - 高达 8 位计数器，用于检测输入数据流中 1..256 个连续的 0 或 1
  - 连续监视各个通道（8 个串行通道收发器输出）
- 当发生模拟看门狗事件或短路检测器事件时，生成断路
- 极值检测器：
  - 存储最小和最大输出数据值
  - 由软件更新
- DMA 可用于读取转换数据
- 中断：结束转换，溢出，模拟看门狗，短路，通道时钟缺失
- “常规”或“注入”转换：
  - “常规”转换可随时进行请求，即使在连续模式下亦如此，且不会对“注入”转换的时序产生影响

27.3 DFSDM 实现

本部分介绍在 DFSDMx 中实现的配置。

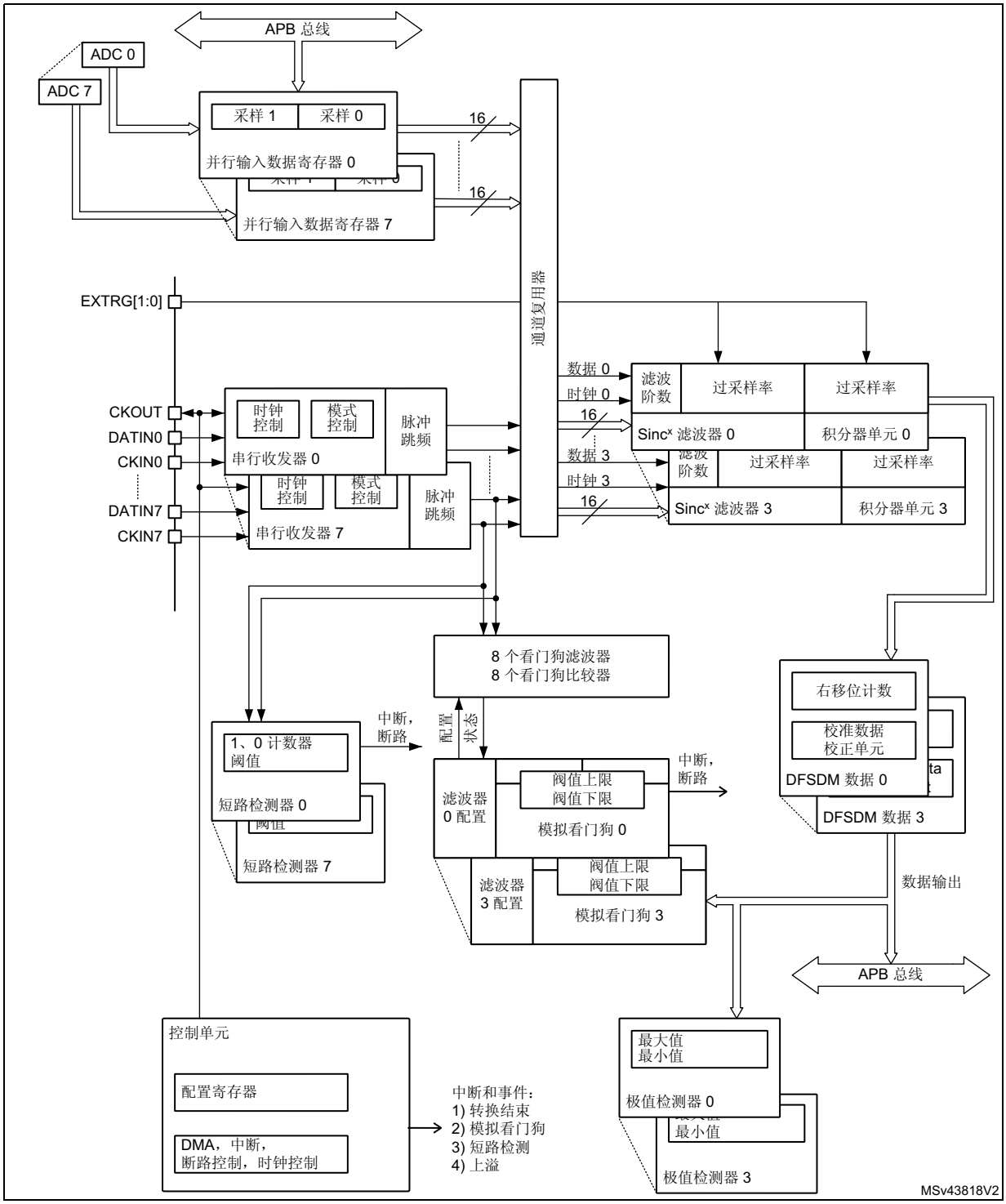
表 159. DFSDM1 实现

DFSDM 功能	DFSDM1
通道数	8
滤波器数	4
来自内部 ADC 的输入	X
支持的触发源	32
脉冲跳过器	X
ID 寄存器支持	-

27.4 DFSDM 功能说明

27.4.1 DFSDM框图

图 162. 单个 DFSDM 框图



1. 本示例显示了 4 个 DFSDM 滤波器和 8 个输入通道（最大 配置）。

27.4.2 DFSDM 引脚和内部信号

表 160. DFSDM 外部引脚

名称	信号类型	注释
VDD	电源	数字电源。
VSS	电源	数字接地电源。
CKIN[7:0]	时钟输入	从外部 ΣΔ 调制器提供的时钟信号。FT 输入。
DATIN[7:0]	数据输入	从外部 ΣΔ 调制器提供的数据信号。FT 输入。
CKOUT	时钟输出	时钟输出，用于为外部 ΣΔ 调制器提供时钟信号。
EXTRG[1:0]	外部触发信号	来自两个 EXTI 信号的输入触发，用于启动模拟转换（来自 GPIO：EXTI11 和 EXTI15）。

表 161. DFSDM 内部信号

名称	信号类型	注释
dfsdm_jtrg[31:0]	内部/ 外部触发信号	来自内部/外部触发源的输入触发，用于启动模拟转换（来自内部源：同步输入，来自外部源：使用时钟同步的异步输入）。详细信息，请参见*表 162*。
dfsdm_break[3:0]	断路信号输出	由模拟看门狗或短路检测器触发的断路信号事件。
dfsdm_dma[3:0]	DMA 请求信号	来自各个 DFSDM_FLTx (x=0..3) 的 DMA 请求信号：结束注入转换事件。
dfsdm_it[3:0]	中断请求信号	各个 DFSDM_FLTx (x=0..3) 的中断信号。
dfsdm_dat_adc[15:0]	ADC输入数据	多达 4 个内部 ADC 数据总线用作并行输入。

表 162. DFSDM 触发器连接

触发器名称	触发源：
dfsdm_jtrg0	TIM1_TRGO
dfsdm_jtrg1	TIM1_TRGO2
dfsdm_jtrg2	TIM8_TRGO
dfsdm_jtrg3	TIM8_TRGO2
dfsdm_jtrg4	TIM3_TRGO
dfsdm_jtrg5	TIM4_TRGO
dfsdm_jtrg6	TIM16_OC1
dfsdm_jtrg7	TIM6_TRGO
dfsdm_jtrg8	TIM7_TRGO
dfsdm_jtrg[23:9]	保留
dfsdm_jtrg24	EXTI11
dfsdm_jtrg25	EXTI15

表 162. DFSDM 触发器连接（续）

触发器名称	触发源：
dfsdm_jtrg26	LTIMER1
dfsdm_jtrg[31:27]	保留

表 163. DFSDM 断路连接

断路名称	断路目标
dfsdm_break[0]	TIM1/TIM15 断路
dfsdm_break[1]	TIM1 断路 2/TIM16 断路
dfsdm_break[2]	TIM8/TIM17 断路
dfsdm_break[3]	TIM8 断路 2

27.4.3 DFSDM 复位和时钟

DFSDM 开关控制

通过将 DFSDM_CH0CFGR1 寄存器的 DFSDMEN 置 1，全局使能 DFSDM 接口。DFSDM 被全局使能后，所有输入通道 y (y=0..7) 和数字滤波器 DFSDM_FLTx (x=0..3) 会在其使能位（DFSDM_CHyCFGR1 中的通道使能位 CHEN 和 DFSDM_FLTxCR1 中的 DFSDM_FLTx 使能位 DFEN）置 1 时开始工作。

数字滤波器 x DFSDM_FLTx (x=0..3) 通过将 DFSDM_FLTxCR1 寄存器的 DFEN 置 1 来使能。使能 DFSDM_FLTx (DFEN=1) 后，$Sinc^x$ 数字滤波器单元和积分器单元会重新初始化。

通过将 DFEN 清零，正在进行的所有转换都会立即停止，DFSDM_FLTx 会置于停止模式。除 DFSDM_FLTxAWSR 和 DFSDM_FLTxISR（均被复位）外，所有寄存器设置保持不变。

通过将 DFSDM_CHyCFGR1 寄存器的 CHEN 置 1 使能通道 y (y=0..7)。通道使能后，会从外部 ΣΔ 调制器或并行内部数据源（ADC 或存储器的 CPU/DMA 写操作）接收串行数据。

在停止系统时钟，使器件进入停止模式之前，必须全局禁止 DFSDM（通过在 DFSDM_CH0CFGR1 中设置 DFSDMEN = 0）。

DFSDM 时钟

内部 DFSDM 时钟 $f_{DFSDMCLK}$ 用于驱动通道收发器、数字处理模块（数字滤波器、积分器）和后续附加模块（模拟看门狗、短路检测器、极值检测器、控制模块），此时钟由 RCC 模块生成，源自系统时钟 SYSCLK 或外设时钟 PCLK2（请参见*第 6.4.32 节：外设独立时钟配置寄存器 (RCC_CCIPR2)* 中的 DFSDMSEL 位说明）。DFSDM 时钟在停止模式下自动停止（对于所有 DFSDM_FLTx，x=0..3，DFEN = 0 时）。

DFSDM 串行通道收发器可以接收外部串行时钟，以采样外部串行数据流。如果使用标准 SPI 编码，内部 DFSDM 时钟必须至少比外部串行时钟快 4 倍；如果使用曼彻斯特编码，则须比外部串行时钟快 6 倍。

DFSDM 可提供一个外部输出时钟信号，以驱动外部 ΣΔ 调制器时钟输入。它通过 CKOUT 引脚提供。该输出时钟信号必须在给定器件手册中的指定范围之内，可以通过分频器取自 DFSDM 时钟，也可以取自音频时钟（请参见 DFSDM_CH0CFGR1 寄存器 CKOUTSRC 位），分频器可以编程，范围为 2 - 256（DFSDM_CH0CFGR1 寄存器中的 CKOUTDIV）。音频时钟源为 SAI1 时钟，可通过 RCC 配置中 SAI1SEL[1:0] 字段进行选择（请参见*第 6.4.32 节：外设独立时钟配置寄存器 (RCC_CCIPR2)*）。

27.4.4 串行通道收发器

有 8 个复用串行数据通道，可以通过各个滤波器、模拟看门狗或者短路检测器将其选择用于转换。这些串行收发器接收来自外部 ΣΔ 调制器的数据流。数据流能以 SPI 格式或曼彻斯特编码格式进行发送（请参见 DFSDM_CHyCFGR1 寄存器的 SITP[1:0] 位）。
通过将 DFSDM_CHyCFGR1 寄存器的 CHEN 置 1 可以使能通道。

通道输入选择

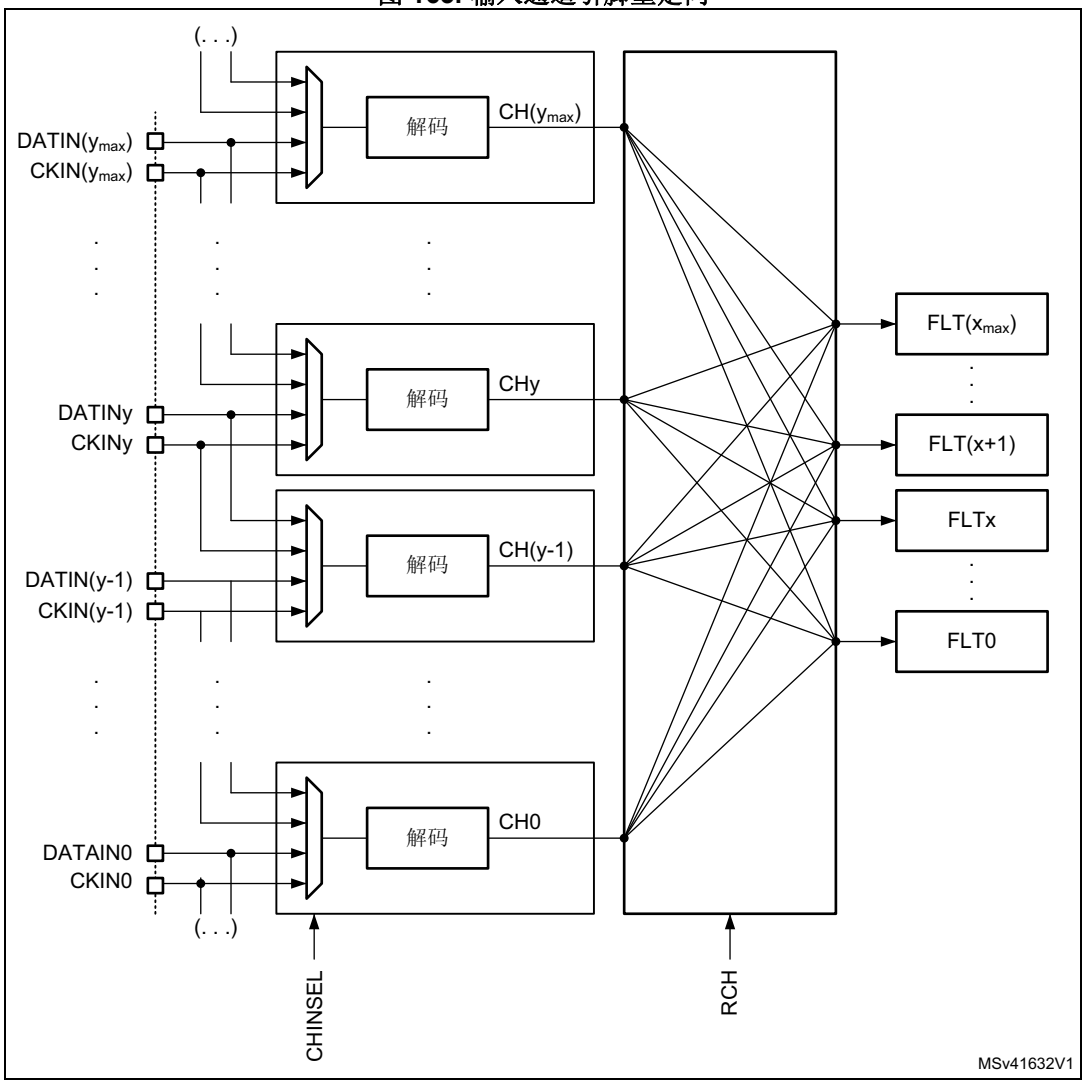
DATINy 和 CKINy 引脚的串行输入（数据和时钟信号）可从后续通道引脚重定向。此串行输入通道重定向可通过 DFSDM_CHyCFGR1 寄存器中的 CHINSEL 位设置。

通道重定向可用于从 PDM（脉冲密度调制）立体声麦克风中采集音频数据。PDM 立体声麦克风包含一个数据和一个时钟信号。数据信号为左右音频通道提供信息（时钟上升沿采样用于左通道，时钟下降沿采样用于右通道）。

PDM 麦克风输入串行通道的配置：

- PDM 麦克风信号（数据、时钟）将连接到 DFSDM 输入串行通道 y（DATINy，CKOUT）引脚。
- 将通道 y 进行以下配置：CHINSEL = 0（输入来自给定通道引脚：DATINy 和 CKINy）。
- 将通道 (y-1)（模8）进行以下配置：CHINSEL = 1（输入来自后续通道 ((y-1)+1) 引脚：DATINy 和 CKINy）。
- 通道 y：SITP[1:0] = 0（上升沿选通数据）=> 通道 y 的左音频通道。
- 通道 (y-1)：SITP[1:0] = 1（下降沿选通数据）=> 通道 y-1 的右音频通道。
- 两个 DFSDM 滤波器将分配至通道 y 和通道 (y-1)（用于对 PDM 麦克风的左右通道进行滤波）。

图 163. 输入通道引脚重定向



输出时钟生成

可在 CKOUT 引脚上提供一个时钟信号，来驱动外部 ΣΔ 调制器时钟输入。此 CKOUT 信号的频率通过预分频器（请参见 DFSDM_CH0CFGR1 寄存器 CKOUTDIV 位）取自 DFSDM 时钟或音频时钟（请参见 DFSDM_CH0CFGR1 寄存器 CKOUTSRC 位）。如果输出时钟停止，则 CKOUT 信号会被置于低电平状态（通过在 DFSDM_CHyCFGR1 寄存器中设置 CKOUTDIV = 0 或 在 DFSDM_CH0CFGR1 寄存器中设置 DFSDMEN = 0，可以停止输出时钟）。在以下时间执行输出时钟停止：

- DFSDMEN 清零后 4 个系统时钟（当 CKOUTSRC =0 时）
- DFSDMEN 清零后 1 个系统时钟和 3 个音频时钟（CKOUTSRC = 1 时）

更改 CKOUTSRC 之前，软件必须等待 CKOUT 停止，以避免在 CKOUT 引脚上产生毛刺信号。输出时钟信号频率必须介于 0 - 20 MHz 范围内。

SPI 数据输入格式操作

在 SPI 格式下，数据流通过数据和时钟信号以串行格式进行发送。数据信号始终由 DATINy 引脚提供。时钟信号可通过 CKINy 引脚从外部提供，也可以通过取自 CKOUT 信号源的信号从内部提供。

如果选择外部时钟源 (SPICKSEL[1:0] = 0)，则根据（DFSDM_CHyCFGR1 寄存器中的）SITP[1:0] 位设置在（CKINy 引脚的）时钟上升沿或下降沿采样（DATINy 引脚上的）数据信号。

内部时钟源 —— 请参见 DFSDM_CHyCFGR1 寄存器中的 SPICKSEL[1:0]：

- CKOUT 信号
  - 用于连接外部 ΣΔ 调制器，调制器直接使用其时钟输入（来自 CKOUT）来生成输出串行通信时钟。
  - 采样点：上升沿/下降沿，具体取决于 SITP[1:0] 设置。
- CKOUT/2 信号（在 CKOUT 上升沿生成）：
  - 用于连接外部 ΣΔ 调制器，调制器将时钟输入（来自 CKOUT）除以 2，来生成输出串行通信时钟（该输出时钟变化在各个时钟输入上升沿有效）。
  - 采样点：每第二个 CKOUT 下降沿。
- CKOUT/2 信号（在 CKOUT 下降沿生成）：
  - 用于连接外部 ΣΔ 调制器，调制器将时钟输入（来自 CKOUT）除以 2，来生成输出串行通信时钟（该输出时钟变化在各个时钟输入下降沿有效）。
  - 采样点：每第二个 CKOUT 上升沿。

注： *只有在外部 ΣΔ 调制器将 CKOUT 信号用作时钟输入（以实现同步时钟和数据操作）时，才能使用内部时钟源。*

使用内部时钟源可省去 CKINy 引脚连接（CKINy 引脚可用于其他用途）。

如果采用 SPI 编码，时钟源信号频率必须介于 0 - 20 MHz 范围内，且小于 $f_{DFSDMCLK}/4$。

曼彻斯特编码数据输入格式操作

如果采用曼彻斯特编码格式，数据流仅通过 DATINy 引脚以串行格式进行发送。经过曼彻斯特解码之后，数据和时钟信号从串行流中恢复。可以有两种曼彻斯特编码设置（请参见 DFSDM_CHyCFGR1 寄存器 SITP[1:0] 位）：

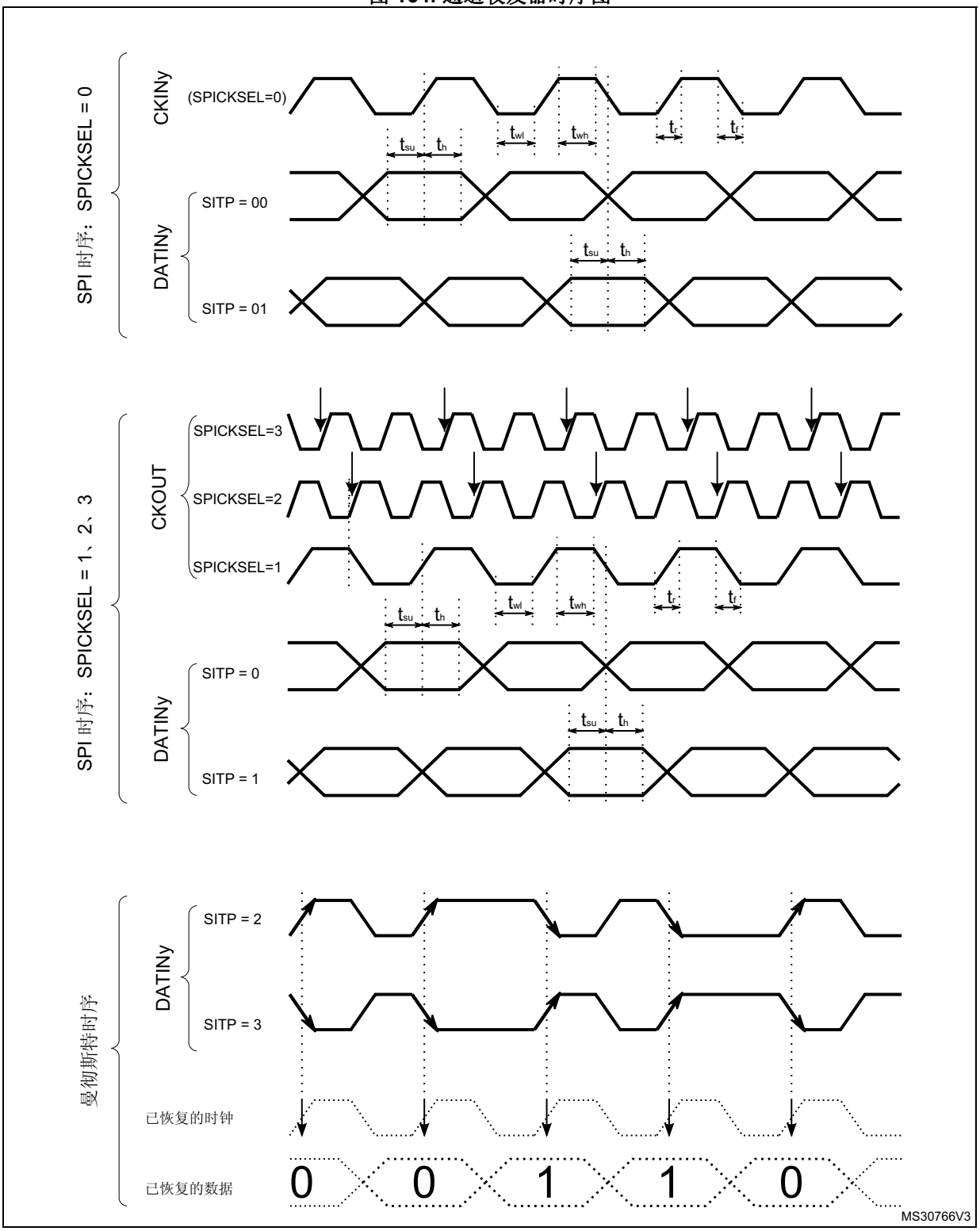
- 信号上升沿 = 逻辑 0；信号下降沿 = 逻辑 1
- 信号上升沿 = 逻辑 1；信号下降沿 = 逻辑 0

如果采用曼彻斯特编码，则恢复后的时钟信号频率必须介于 0 - 10 MHz 范围内，且小于 $f_{DFSDMCLK}/6$。

为了正确接收曼彻斯特编码数据，必须根据预期的曼彻斯特数据速率对（DFSDM_CH0CFGR1 寄存器中的）CKOUTDIV 分频器进行设置，设置公式为：

$$((CKOUTDIV + 1) \times T_{SYSCLK}) < T_{曼彻斯特时钟} < (2 \times CKOUTDIV \times T_{SYSCLK})$$

图 164. 通道收发器时序图



时钟缺失检测

可检查通道串行时钟输入的时钟缺失/存在情况，以确保转换和错误报告正常运行。可通过 DFSDM_CHyCFGR1 寄存器 CKABEN 位在各个输入通道 y 上使能或禁止时钟缺失检测。如果使能了给定通道的时钟缺失检测，则会对其连续执行时钟缺失检测。如果出现输入时钟错误，则时钟缺失标志会置 1 (CKABF[y] = 1)，并会调用中断（CKABIE = 1 时）（请参见 DFSDM_FLT0ISR 寄存器的 CKABF[7:0] 和 DFSDM_CHyCFGR1 的 CKABEN）。时钟缺失标志清零（通过 DFSDM_FLT0ICR 寄存器的 CLRCKABF）后，会刷新时钟缺失标志。相应通道 y 禁止时，则时钟缺失状态位 CKABF[y] 还会由硬件置 1（如果 CHEN[y] = 0，则 CKABF[y] 保持在置 1 状态）。

如果发生了时钟缺失事件，则数据转换（和/或模拟看门狗和短路检测器）会提供错误的数据。当报告了时钟缺失时，用户应管理此事件并丢弃给定数据。

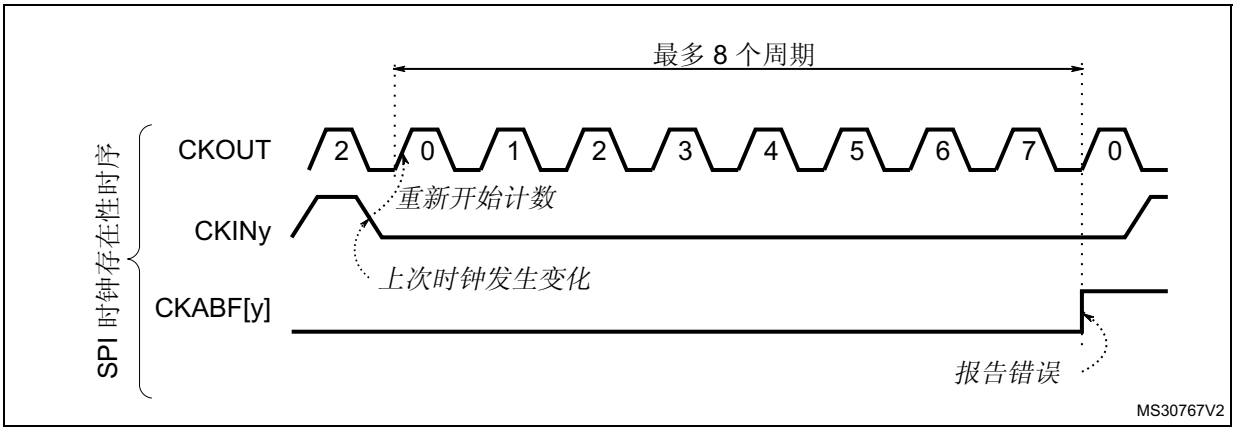
只有在将系统时钟用于 CKOUT 信号（DFSDM_CH0CFGR1 寄存器中的 CKOUTSRC = 0）时，时钟缺失功能才可用。

当收发器尚未被同步时，时钟缺失标志会置 1，且无法由（DFSDM_FLT0ICR 寄存器中）CLRCKABF[y] 位清零。与时钟缺失检测功能相关的软件序列应：

- 通过 CHEN = 1 使能给定通道。
- 尝试将时钟缺失标志清零（通过 CLRCKABF = 1），直至时钟缺失标志已实际清零 (CKABF = 0)。此时，收发器被同步（信号时钟有效），能够接收数据。
- 使能时钟缺失功能 CKABEN = 1 且相关中断 CKABIE = 1，以检测 SPI 时钟是否丢失或曼彻斯特数据边沿是否缺失。

如果使用 SPI 数据格式，将基于外部输入时钟与输出时钟生成（CKOUT 信号）的比较进行时钟缺失检测。输入通道中的外部输入时钟信号必须每经过 CKOUT 信号（由 DFSDM_CH0CFGR1 寄存器 CKOUTDIV 字段控制）的 8 个信号周期更改一次。

图 165. SPI 时钟缺失时序图



如果使用曼彻斯特数据格式，则时钟缺失表示无法从曼彻斯特编码信号中恢复时钟。为了能够正常恢复时钟，首先必须接收带 1 - 0 或 0 - 1 转换的数据（有关曼彻斯特同步，请参见*图 167*）。

如果使用曼彻斯特编码格式，则时钟缺失检测（在首次成功同步之后）是基于编码串行数据输入信号与输出时钟生成（CKOUT 信号）之间的变化比较。在 CKOUT 信号（由 DFSDM_CH0CFGR1 寄存器 CKOUTDIV 位控制）的两个周期期间，DATINy 引脚电平必须发生变化。该条件还定义了能正确恢复曼彻斯特编码的数据和时钟信号的最小数据速率。

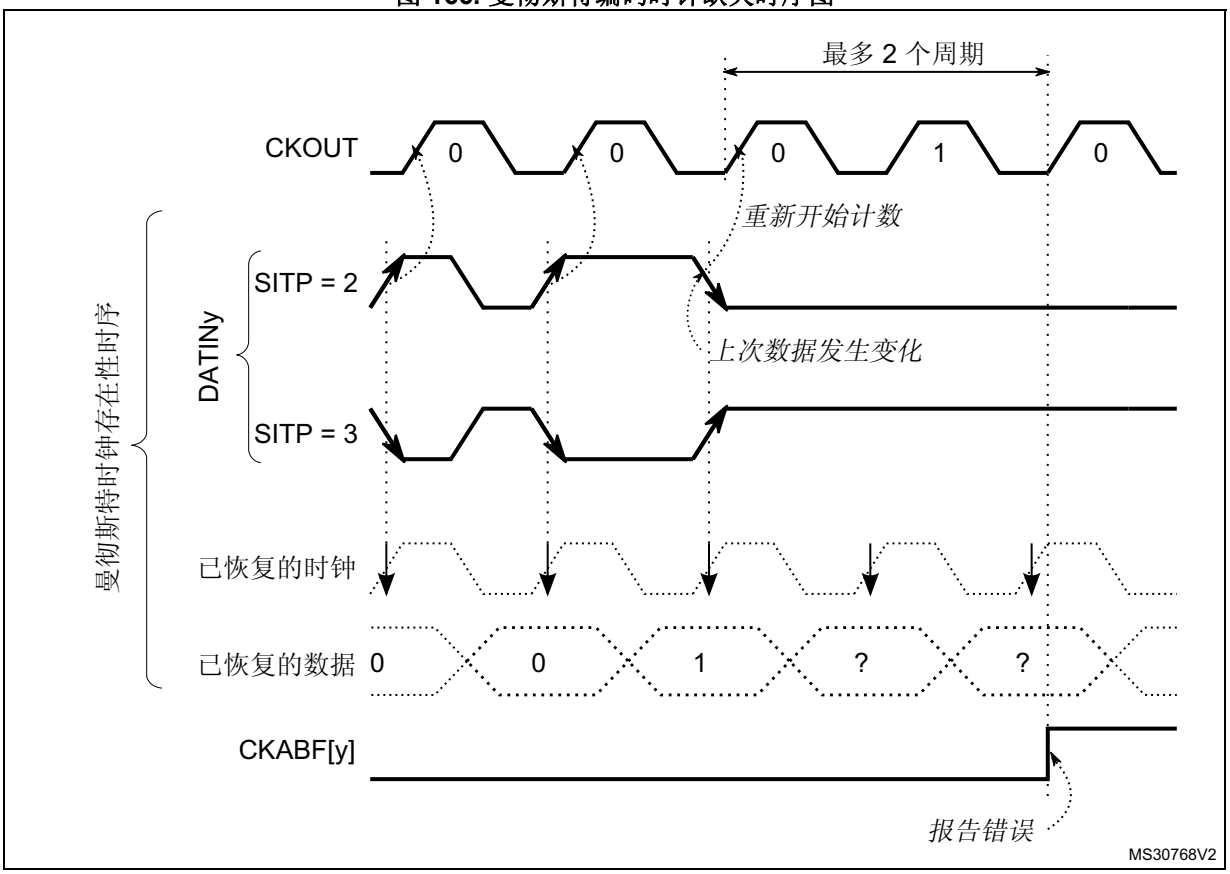
曼彻斯特编码数据的最大数据速率必须小于 CKOUT 信号。

因此，为正确接收曼彻斯特编码数据，必须根据以下公式设置 CKOUTDIV 分频器：

$$((CKOUTDIV + 1) \times T_{SYSCLK}) < T_{曼彻斯特时钟} < (2 \times CKOUTDIV \times T_{SYSCLK})$$

如果出现输入时钟恢复错误，则时钟缺失标志会置 1 (CKABF[y] = 1)，并会调用中断（CKABIE=1 时）（请参见 DFSDM_FLT0ISR 寄存器的 CKABF[7:0] 和 DFSDM_CHyCFGR1 的 CKABEN）。时钟缺失标志清零（通过 DFSDM_FLT0ICR 寄存器的 CLRCKABF）后，会刷新时钟缺失标志。

图 166. 曼彻斯特编码时钟缺失时序图



曼彻斯特/SPI 编码同步

必须在使能通道（DFSDM_CHyCFGR1 寄存器 CHEN = 1）后，首次同步曼彻斯特编码数据流。当接收到 0 - 1 或 1 - 0 数据转换（以便能检测有效数据边沿）时，同步结束。在通过 DFSDM_FLT0ICR 的 CLRCKABF [y] 将给定通道的 CKABF[y] 清零之后，可以按照以下详细描述的软件序列通过轮询 CKABF [y] = 0 来检查同步是否结束：

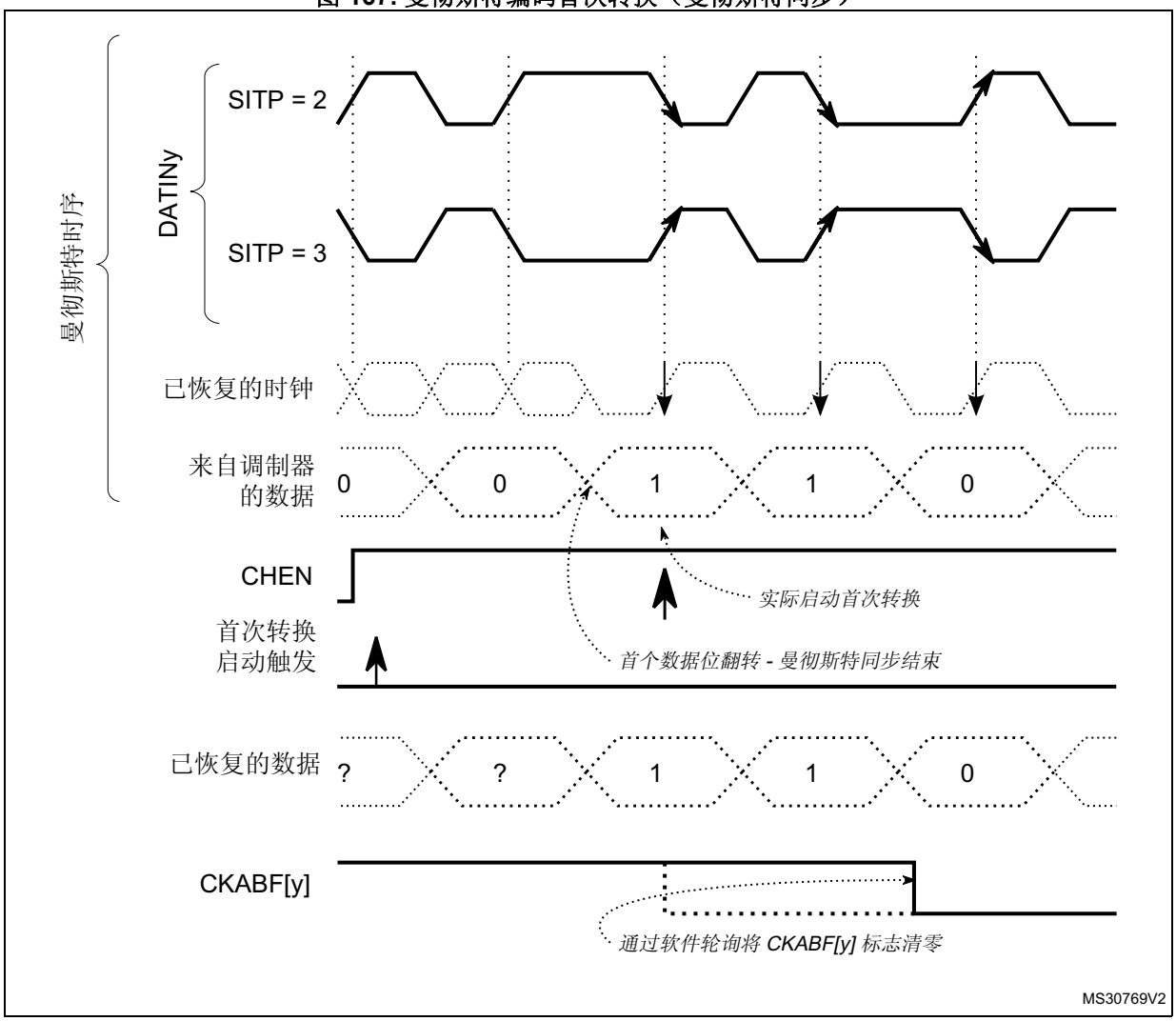
通过将 CLRCKABF[y] 位置 1 将 CKABF[y] 标志清零。如果通道 y 尚未同步，则硬件会立即将 CKABF[y] 标志置 1。软件随后将回读 CKABF[y] 标志，如果该标志置 1，则会再次通过将 CLRCKABF[y] 位置 1 来将其清零。此软件序列（CKABF[y] 标志的轮询）将继续进行，直至 CKABF[y] 标志置 1（表示曼彻斯特码流已同步）。为了能够同步/接收曼彻斯特编码数据，必须根据预期的曼彻斯特数据速率对（DFSDM_CH0CFGR1 寄存器中的）CKOUTDIV 分频器进行设置，以下为设置公式：

$$((CKOUTDIV + 1) \times T_{SYSCLK}) < T_{曼彻斯特时钟} < (2 \times CKOUTDIV \times T_{SYSCLK})$$

SPI 码流在第一次检测到时钟输入信号（有效的上升沿/下降沿）后同步。

注： *当收发器尚未被同步时，时钟缺失标志会置 1，且无法由（DFSDM_FLT0ICR 寄存器中）CLRCKABF[y] 位清零。*

图 167. 曼彻斯特编码首次转换（曼彻斯特同步）



外部串行时钟频率测量

通过测量通道串行时钟输入频率可提供来自外部 ΣΔ 调制器的实时数据速率，这对应用目的很重要。

外部串行时钟输入频率可以通过在一个转换持续时间内对 DFSDM 时钟（$f_{DFSDMCLK}$）进行计数的定时器来测量。计数在转换触发（常规或注入）后的第一个输入数据时钟处开始，并在转换结束（转换结束标志置 1）前的最后一个输入数据时钟处结束。转换完成（JEOCF =1 或 REOCF = 1）时，会在寄存器 DFSDM_FLTxCNVTIMR 的计数器 CNVCNT[27:0] 中更新每次转换的时间（第一个串行采样和最后一个串行采样之间的时间）。然后用户可以根据数字滤波器设置（FORD、FOSR、IOSR、FAST）计算数据速率。只有在滤波器被旁路（FOSR = 0，只有积分器有效，DFSDM_FLTxCNVTIMR 寄存器 CNVCNT[27:0] = 0）时，才会停止外部串行频率测量。

对于并行数据输入（*第 27.4.6 节：并行数据输入*），测量的频率为一次转换期间内的平均输入数据速率。

注： *转换被中断（例如，通过禁止/使能所选通道）时，还会在 CNVCNT[27:0] 中对中断时间进行计数。因此，为得到正确的转换时间结果，建议不要中断转换。*

转换时间：

FAST = 0 时的注入转换或常规转换（或者 FAST=1 时的首次转换）：

对于 $Sinc^x$ 滤波器 (x =1..5)：

$t = CNVCNT/f_{DFSDMCLK} = [F_{OSR} * (I_{OSR}-1 + F_{ORD}) + F_{ORD}] / f_{CKIN}$

对于 FastSinc 滤波器：

$t = CNVCNT/f_{DFSDMCLK} = [F_{OSR} * (I_{OSR}-1 + 4) + 2] / f_{CKIN}$

FAST = 1 时的常规转换（首次转换除外）：

对于 $Sinc^x$ 和 FastSinc 滤波器：

$t = CNVCNT/f_{DFSDMCLK} = [F_{OSR} * I_{OSR}] / f_{CKIN}$

如果 F_{OSR} = FOSR[9:0] + 1 = 1（滤波器旁路，仅积分器有效），则：

$t = I_{OSR} / f_{CKIN}$（... 但是 CNVCNT = 0）

其中：

- f_{CKIN} 为（给定通道 CKINy 引脚上的）通道输入时钟频率或（并行数据输入时的）输入数据速率
- F_{OSR} 为滤波器过采样率：F_{OSR} = FOSR[9:0] + 1（请参见 DFSDM_FLTxFCR 寄存器）
- I_{OSR} 为积分器过采样率：I_{OSR} = IOSR[7:0] + 1（请参见 DFSDM_FLTxFCR 寄存器）
- F_{ORD} 为滤波器阶数：F_{ORD} = FORD[2:0]（请参见 DFSDM_FLTxFCR 寄存器）

通道偏移设置

每个通道都有自身的偏移设置（在寄存器中），该偏移值最终会从给定通道的各个转换结果（注入或常规）中减去。在数据右移位后执行偏移校正。偏移值以 24 位有符号值的形式存储在 DFSDM_CHyCFGR2 寄存器的 OFFSET[23:0] 字段中。

数据右移位

要使结果与 24 位值对齐，每个通道均定义了一系列右移位，这些右移位将应用于给定通道的各个转换结果（注入或常规）。右移位数存储在 DFSDM_CHyCFGR2 寄存器的 DTRBS[4:0] 位中。

右移位将结果四舍五入为最接近的整数值。移位结果的符号保留，以便得到有效 24 位有符号格式的结果数据。

脉冲跳频

脉冲跳频的目的在于实现指定输入通道的延迟线等特性。输入串行数据流（仅串行流）中指定数量的样品可在进入滤波器前被丢弃。该数据丢弃通过跳过指定数量的采样输入时钟脉冲（指定的串行数据样品未被滤波器采样）来执行。采样时钟由脉冲跳频功能门控，以获得指定数量的时钟脉冲。指定的时钟脉冲被跳过时，输入数据的滤波会继续。与未跳过的数据流相比，该操作导致来自滤波器的最终输出样品（及下一个样品）从后面的输入数据计算得出。由于是从较新的输入样品而不是“未跳过”样品计算得出，因此，最终的样品看起来是向前移动的。最终的“跳过样品”将在稍后转换，因为跳过的输入数据样品必须由随后的输入数据样品替换。由于两个数据缓冲区都相移了，因此，最终的数据缓冲区特性（跳过和未跳过的输出数据缓冲区比较）就好像未跳过的数据流稍微延迟了。

要跳过的时钟脉冲数应写入 DFSDM_CHyDLYR 寄存器的 PLSSKP[5:0] 字段。写入 PLSSKP[5:0] 字段后，将开始在指定通道上执行脉冲跳过。可对 PLSSKP[5:0] 字段进行读取，来检查脉冲跳频的进度。PLSSKP[5:0]=0 表示已执行脉冲跳频。

对 PLSSKP[5:0] 执行单次写操作时，最多可跳过 63 个时钟脉冲。如果需要跳过更多脉冲，则用户必须多次写入 PLSSKP[5:0] 字段。应用程序软件应处理每个滤波器的累积跳过时钟数。

27.4.5 配置输入串行接口

必须为输入串行接口配置以下参数：

- **输出时钟预分频器。**提供了可编程预分频器，用来从 DFSDM 时钟生成输出时钟 (2 - 256)。它由 DFSDM_CH0CFGR1 寄存器中的 CKOUTDIV[7:0] 位定义。
- **串行接口类型和输入时钟相位。**选择 SPI 或曼彻斯特编码以及输入时钟的采样边沿。它由 DFSDM_CHyCFGR1 寄存器中的 SITP [1:0] 位定义。
- **输入时钟源。**来自 CKINy 引脚的外部源或来自 CKOUT 引脚的内部源。它由 DFSDM_CHyCFGR1 寄存器中的 SPICKSEL[1:0] 字段定义。
- **最终数据右移位。**定义最终数据右移位，以将结果与 24 位值对齐。它由 DFSDM_CHyCFGR2 寄存器中的 DTRBS[4:0] 定义。
- **每个通道的通道偏移。**定义给定串行通道的模拟偏移（所连接外部 ΣΔ 调制器的偏移）。它由 DFSDM_CHyCFGR2 寄存器中的 OFFSET[23:0] 位定义。
- **每个通道的短路检测器和时钟缺失使能。**用于在 DFSDM_CHyCFGR1 寄存器中使能或禁止给定串行通道的短路检测器（通过 SCDEN 位）和时钟缺失检测（通过 CKABEN 位）。
- **模拟看门狗滤波器和短路检测器阈值设置。**用于配置通道模拟看门狗滤波器参数和通道短路检测器参数。在 DFSDM_CHyAWSCDR 寄存器中定义这些配置。

27.4.6 并行数据输入

每个输入通道均为 16 位并行数据输入提供一个寄存器（除了串行数据输入外）。每个 16 位并行输入只能源自内部数据源：

- 内部 ADC 结果。
- 直接 CPU/DMA 写操作。

通过 DFSDM_CHyCFGR1 寄存器的 DATMPX[1:0] 字段选择是对给定通道使用串行还是并行数据输入。DATMPX[1:0] 中还定义了并行数据源：内部 ADC 或直接由 CPU/DMA 进行的写操作。

每个通道均包含一个 32 位数据输入寄存器 DFSDM_CHyDATINR，在该寄存器中可写入 16 位数据。数据采用 16 位有符号格式。这些数据还可用作接受 16 位并行数据的数字滤波器的输入。

如果选择串行数据输入 (DATMPX[1:0] = 0)，则 DFSDM_CHyDATINR 寄存器被写保护。

来自内部 ADC 的输入

对于并行 ADC 数据输入 (DATMPX[1:0]=1)，ADC[y+1] 结果被分配至通道 y 输入（ADC1 填充 DFSDM_CHDATIN0R 寄存器，ADC2 填充 DFSDM_CHDATIN1R 寄存器，...，ADC8 填充 DFSDM_CHDATIN7R 寄存器）。来自 ADC[y+1] 的转换结束事件会导致更新通道 y 的数据（来自 ADC[y+1] 的并行数据被用作数字滤波器的下一个采样）。转换结束事件发生时，来自 ADC[y+1] 的数据被写入 DFSDM_CHyDATINR 寄存器（INDAT0[15:0] 字段）。

数据封装模式设置（DFSDM_CHyCFGR1 寄存器中的 DATPACK[1:0]）对 ADC 数据输入无影响。

来自存储器的输入（直接由 CPU/DMA 写入）

对于由 CPU 或 DMA (DATMPX[1:0] = 2) 直接写入到 DFSDM_CHyDATINR 寄存器中的数据，可以将其用作数据输入，以处理来自存储器或外设的数字数据流。

数据可由 CPU 或 DMA 写入到 DFSDM_CHyDATINR 寄存器中：

1. **CPU 数据写入：**

   输入数据直接由 CPU 写入到 DFSDM_CHyDATINR 寄存器中。

2. **DMA 数据写入：**

   DMA 应配置为存储器到存储器传输模式，以将存储器缓冲区中的数据传输至 DFSDM_CHyDATINR 寄存器中。目标存储器地址为 DFSDM_CHyDATINR 寄存器的地址。数据以 DMA 传输速度从存储器传输至 DFSDM 并行输入。

   此 DMA 不同于读取 DFSDM 转换结果所使用的 DMA。这两个 DMA 可同时使用，第一个 DMA（配置为存储器到存储器传输）用于输入数据写入，第二个 DMA（配置为外设到存储器传输）用于数据结果读取。

对 DFSDM_CHyDATINR 的访问可为 16 位或 32 位宽，允许分别在一次写操作中加载一个或两个采样。32 位输入数据寄存器 (DFSDM_CHyDATINR) 可用一个或两个 16 位数据采样进行填充，具体取决于在 DFSDM_CHyCFGR1 寄存器的 DATPACK[1:0] 字段中定义的数据封装操作模式：

1. **标准模式 (DATPACK[1:0] = 0)：**

   只有一个样本存储在 DFSDM_CHyDATINR 寄存器的 INDAT0[15:0] 字段中，其用作通道 y 的输入数据。高 16 位 (INDAT1[15:0]) 被忽略，且被写保护。数字滤波器必须执行一次输入采样（通过 INDAT0[15:0]），以便清空由 CPU/DMA 填充后的数据寄存器。该模式与对 DFSDM_CHyDATINR 寄存器的 16 位 CPU/DMA 访问结合使用，以在每次写操作时加载一个采样。

2. **交错模式 (DATPACK[1:0] = 1)：**

   DFSDM_CHyDATINR 寄存器被用作两个采样缓冲区。第一个采样存储在 INDAT0[15:0] 中，第二个采样存储在 INDAT1[15:0] 中。数字滤波器必须通过通道 y 执行两次输入采样，以清空 DFSDM_CHyDATINR 寄存器。该模式与对 DFSDM_CHyDATINR 寄存器的 32 位 CPU/DMA 访问结合使用，以在每次写操作时加载两个采样。

3. **双通道模式 (DATPACK[1:0] = 2)：**

   将两个采样写入到 DFSDM_CHyDATINR 寄存器中。INDAT0[15:0] 中的数据用于通道 y，INDAT1[15:0] 中的数据用于通道 y+1。INDAT1[15:0] 中的数据会被自动复制到下一 (y+1) 通道数据寄存器 DFSDM_CH[y+1]DATINR) 的 INDAT0[15:0] 中。数字滤波器必须执行两次采样（一次来自通道 y，另一次来自通道 (y+1)），以清空 DFSDM_CHyDATINR 寄存器。

   只能针对偶数通道数（y = 0、2、4、6）进行双通道模式设置 (DATPACK[1:0] = 2)。如果将奇数通道（y = 1、3、5、7）设为双通道模式，则对于该通道，INDAT0[15:0] 和 INDAT1[15:0] 部分均被写保护。如果偶数通道被设为双通道模式，则下一奇数通道必须设置成标准模式 (DATPACK[1:0] = 0)，以便与偶数通道正确配合。

有关 DFSDM_CHyDATINR 寄存器数据模式和通道数据样本分配，请参见*图 168*。

图 168. DFSDM_CHyDATINR 寄存器操作模式和分配



首先使能所选输入通道（通道 y），以便进行数据采集（启动通道 y 转换），然后必须向 DFSDM_CHyDATINR 寄存器进行写操作，以加载一个或两个采样。否则，在下次处理时会丢失写入的数据。

例如：对于单次转换和交错模式，在启动单次转换前，不要开始向 DFSDM_CHyDATINR 写入成对数据样本（开始转换前 DFSDM_CHyDATINR 中存在的数据均会被丢弃）。

27.4.7 通道选择

存在 8 个复用通道，可选择用于注入通道组转换和/或常规通道转换。

注入通道组可以选择 8 个通道中的任一通道或其全部。DFSDM_FLTxJCHGR 寄存器中的 JCHG[7:0] 选择注入组通道，其中 JCHG[y] =1 表示已选择通道 y。

注入转换可工作在扫描模式 (JSCAN = 1) 或单个模式 (JSCAN = 0) 下。在扫描模式下，每个所选通道都会依次转换。最低通道（通道 0，如果已选择）最先转换，紧接着转换下一个较高通道，直至 JCHG[7:0] 选择的所有通道均被转换。在单个模式 (JSCAN=0) 下，只会转换所选的一个通道，通道选择会移至下一通道。JSCAN = 0 时，对 JCHG[7:0] 进行写入操作会将通道选择复位为选择的最低通道。

注入转换可由软件或触发器启动。此类转换从不会被常规转换中断。

常规通道只选择 8 个通道的其中一个。DFSDM_FLTxCR1 寄存器中的 RCH[2:0] 指示所选通道。

常规转换只能由软件启动（不能由触发器启动）。请求注入转换时，会暂时中断连续常规转换序列。

对已禁止通道（DFSDM_CHyCFGR1 寄存器中的 CHEN = 0）执行转换会导致转换将永远不会结束，这是因为未提供输入数据的原因（无时钟信号）。在这种情况下，需要使能给定通道（DFSDM_CHyCFGR1 寄存器中的 CHEN = 1），或停止转换（通过 DFSDM_FLTxCR1 寄存器的 DFEN = 0）。

27.4.8 数字滤波器配置

DFSDM 包含 $Sinc^x$ 类型数字滤波器实现。此 $Sinc^x$ 滤波器执行输入数字数据流滤波，这会导致减小输出数据速率（抽取）以及增大输出数据分辨率。可对 $Sinc^x$ 数字滤波器进行配置，以达到所需输出数据速率和所需输出数据分辨率。可配置参数有：

- 滤波器阶数/类型：（请参见 DFSDM_FLTxFCR 寄存器中的 FORD[2:0] 位）：
  - FastSinc
  - $Sinc^1$
  - $Sinc^2$
  - $Sinc^3$
  - $Sinc^4$
  - $Sinc^5$
- 滤波器过采样/抽取率：（请参见 DFSDM_FLTxFCR 寄存器中的 FOSR[9:0] 位）：
  - FOSR = 1-1024 —— 适用于 FastSinc 滤波器 和 $Sinc^x$ 滤波器 x = F_{ORD} = 1..3
  - FOSR = 1-215 —— 适用于 $Sinc^x$ 滤波器 x = F_{ORD} = 4
  - FOSR = 1-73 —— 适用于 $Sinc^x$ 滤波器 x = F_{ORD} = 5

滤波器支持以下传递函数（H 域中的冲激响应）：

- $Sinc^x$ 滤波器类型： $$H(z) = \left(\frac{1 - z^{-FOSR}}{1 - z^{-1}}\right)^x$$
- FastSinc 滤波器类型： $$H(z) = \left(\frac{1 - z^{-FOSR}}{1 - z^{-1}}\right)^2 \cdot (1 + z^{-(2 \cdot FOSR)})$$

图 169. 示例：$Sinc^3$ 滤波器响应

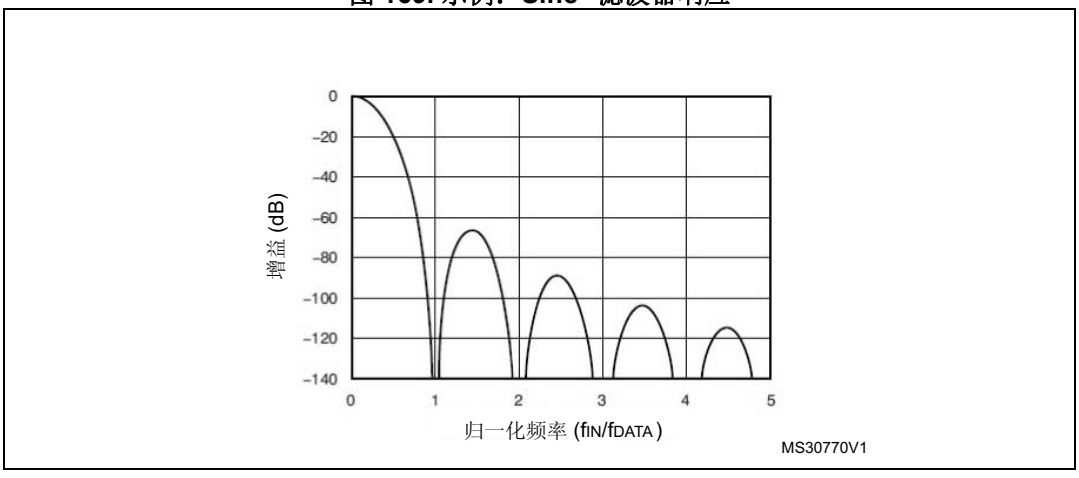


表 164. 滤波器最大输出分辨率（来自滤波器输出的峰值数据值），基于某些 FOSR 值

FOSR	Sinc^1	Sinc^2	FastSinc	Sinc^3	Sinc^4	Sinc^5
x	+/- x	+/- x^2	+/- $2x^2$	+/- x^3	+/- x^4	+/- x^5
4	+/- 4	+/- 16	+/- 32	+/- 64	+/- 256	+/- 1024
8	+/- 8	+/- 64	+/- 128	+/- 512	+/- 4096	-
32	+/- 32	+/- 1024	+/- 2048	+/- 32768	+/- 1048576	+/- 33554432
64	+/- 64	+/- 4096	+/- 8192	+/- 262144	+/- 16777216	+/- 1073741824
128	+/- 128	+/- 16384	+/- 32768	+/- 2097152	+/- 268435456	在满量程输入的条件下，结果会溢出（> 32位有符号整数值）
256	+/- 256	+/- 65536	+/- 131072	+/- 16777216		
1024	+/- 1024	+/- 1048576	+/- 2097152	+/- 1073741824		

有关 Sinc 滤波器类型属性和使用的更多信息，建议了解下数字滤波器的理论知识（可从 Internet 上下载更多资源）。

27.4.9 积分器单元

积分器会对来自数字滤波器的数据进一步提升抽取率和分辨率。对于来自滤波器的给定数量的数据采样，积分器对其进行简单的求和。

积分器过采样率参数定义了将多少个数据的和作为积分器的一个数据输出。IOSR 可设置为介于 1-256 范围内（请参见 DFSDM_FLTxFCR 寄存器中的 IOSR[7:0] 位说明）。

表 165. 积分器最大输出分辨率（来自积分器输出的峰值数据值），基于某些 IOSR 值，FOSR = 256 以及 Sinc^3 滤波器类型（最大数据）

IOSR	Sinc^1	Sinc^2	FastSinc	Sinc^3	Sinc^4	Sinc^5
x	+/- FOSR. x	+/- $FOSR^2$.x	+/- 2.$FOSR^2$. x	+/- $FOSR^3$. x	+/- $FOSR^4$. x	+/- $FOSR^5$. x
4	-	-	-	+/- 67 108 864	-	-
32	-	-	-	+/- 536 870 912	-	-
128	-	-	-	+/- 2 147 483 648	-	-
256	-	-	-	+/- 2^{32}	-	-

27.4.10 模拟看门狗

模拟看门狗用于在达到或超出给定阈值上限或者达到或低于给定阈值下限时，触发外部信号（断路或中断）。随即可调用中断/事件/断路生成。

每个模拟看门狗都会根据（DFSDM_FLTxCR1 寄存器中）AWFSEL 位设置监视串行数据接收器输出（在每个通道上模拟看门狗滤波器之后）或数据输出寄存器（当前注入或常规转换结果）。是否需要通过模拟看门狗 x 监视的输入通道可以通过 DFSDM_FLTxCR2 寄存器中的 AWDCH[7:0] 选择。

输入通道上的模拟看门狗转换与标准转换无关。在标准转换模式下，模拟看门狗在每个输入通道上使用自己的滤波器和信号处理，与注入或常规主转换无关。在连续转换模式下，对所选输入通道执行模拟看门狗转换，以便在注入或常规主转换暂停（RCIP = 0，JCIP = 0）时也仍能对通道进行监视。

阈值上限和阈值下限寄存器与给定数据值（通过 DFSDM_FLTxAWHTR 寄存器的 AWHT[23:0] 位和 DFSDM_FLTxAWLTR 寄存器的 AWLT[23:0] 位进行设置）进行比较。

可以选择 2 种情况将阈值寄存器与数据值进行比较：

- 选择 1：在这种情况下，输入数据取自最终输出数据寄存器 (AWFSEL = 0)。该选择具有以下特性：
  - 高输入数据分辨率（高达 24 位）
  - 响应时间慢——不适用于过流检测等快速响应应用
  - 为了进行比较，最终数据需要经过移位和偏移数据校正
  - 最终数据仅在执行常规或注入主转换后才可用
  - 可在并行输入数据源（DFSDM_CHyCFGR1 寄存器中的 DATMPX[1:0] ≠ 0）情况下使用
- 选择 2：在这种情况下，输入数据取自任一串行数据接收器输出 (AWFSEL = 1)。该选择具有以下特性：
  - 输入串行数据通过专用模拟看门狗 $Sinc^x$ 通道滤波器进行处理，滤波器的过采样率 (1..32) 和滤波器阶数 (1..3) 可以配置（请参见 DFSDM_CHyAWSCDR 寄存器中的 AWFOSR[4:0] 和 AWFORD[1:0] 位设置）
  - 较低分辨率（最高为 16 位）
  - 响应时间快——适用于需要快速响应的应用，如过流/过压检测
  - 数据以连续模式提供，与常规或注入主转换活动无关

进行输入通道监视 (AWFSEL = 1) 时，用于跟阈值进行比较的数据取自 AWDCH[7:0] 字段（DFSDM_FLTxCR2 寄存器）所选的通道。将每个所选通道滤波器结果与一个阈值对 (AWHT[23:0] / AWLT[23:0]) 进行比较。此时，只有高 16 位 (AWHT[23:8] / AWLT[23:8]) 会定义与模拟看门狗滤波器输出进行比较的 16 位阈值，因为来自模拟看门狗滤波器的数据的分辨率最高为 16 位。在这种情况下 (AWFSEL=1)，不对 AWHT[7:0] / AWLT[7:0] 位进行比较。

在 DFSDM_CHyAWSCDR 寄存器中设置各个输入通道的模拟看门狗滤波器配置的参数（滤波器阶数 AWFORD[1:0] 和滤波器过采样率 AWFOSR[4:0]）。

每个输入通道都配有自身的比较器，用于比较模拟看门狗数据 （来自模拟看门狗滤波器）和模拟看门狗阈值 (AWHT/AWLT)。选择多个通道（DFSDM_FLTxCR2 寄存器的字段 AWDCH[7:0]）时，可同时接收多个比较请求。在这种情况下，首先管理编号最小的通道请求，然后继续管理编号较大的所选通道。对于每个通道，结果均可记录在单独的标志（DFSDM_FLTxAWSR 寄存器的字段 AWHTF[7:0] 和字段 AWLTF[7:0]）中。每个通道请求均在 8 个 DFSDM 时钟周期内执行。因此，各个通道的带宽被限制为 8 个 DFSDM 时钟周期 （AWDCH[7:0] = 0xFF 时）。因为最大输入通道采样时钟频率为 DFSDM 时钟频率的四分之一，因此在该输入时钟速度下，不能对模拟看门狗配置 AWFOSR = 0 （模拟看门狗滤波器被旁路）。因此，用户必须根据输入采样时钟速度和 DFSDM 频率正确配置监视的通道数和模拟看门狗滤波器参数。

给定通道 y 的模拟看门狗滤波器数据可通过固件从 DFSDM_CHyWDATR 寄存器 WDATA[15:0] 字段中读取。如果 DFSDM_CHyCFGR1 寄存器中的 CHEN = 1，则该模拟看门狗滤波器数据被连续转换，其数据速率由模拟看门狗滤波器设置和通道输入时钟频率给定。

模拟看门狗滤波器转换的工作方式类似于常规的快速连续转换（无积分器）。模拟看门狗滤波器输出（通道输入时钟频率为 f_{CKIN}）一个结果所需的串行采样数量：

首次转换：

对于 Sinc^x 滤波器 (x = 1..5)：采样数 = [F_{OSR} * F_{ORD} + F_{ORD} + 1]

对于 FastSinc 滤波器：采样数 = [F_{OSR} * 4 + 2 + 1]

后续转换：

对于 Sinc^x 和 FastSinc 滤波器：采样数 = [FOSR * IOSR]

其中：

F_{OSR} .....滤波器过采样率：F_{OSR} = AWFOSR[4:0] + 1（请参见 DFSDM_CHyAWSCDR 寄存器）

F_{ORD} .....滤波器阶数：F_{ORD} = AWFORD[1:0]（请参见 DFSDM_CHyAWSCDR 寄存器）

对于输出数据寄存器监视 (AWFSEL = 0)，在完成最终数据的右移位和偏移校正（请参见 DFSDM_CHyCFGR2 寄存器的 OFFSET[23:0] 和 DTRBS[4:0] 字段）后进行比较。在由（DFSDM_FLTxCR2 寄存器中）AWDCH[7:0] 字段所选通道的每次注入或常规转换结束后，进行比较。

模拟看门狗事件的状态反映在 DFSDM_FLTxAWSR 寄存器中，该寄存器中锁定了给定事件。AWHTF[y] = 1 标志指示在通道 y 上是否超出了 AWHT[23:0] 值。AWLTF[y] = 1 标志指示在通道 y 上是否超出了 AWLT[23:0] 值。对于 DFSDM_FLTxAWSR 寄存器中锁存的事件，将通过向 DFSDM_FLTxAWCFR 寄存器中的相应清零位 CLRAWHTF[y] 或 CLRAWLTF[y] 写入“1”来清除。

模拟看门狗的全局状态通过 DFSDM_FLTxISR 寄存器中的 AWDF 标志位来反映（其用于快速检测中断源）。AWDF = 1 表示至少发生了一个看门狗事件（至少对于一个通道，AWHTF[y] = 1 或 AWLTF[y] = 1）。所有 AWHTF[7:0] 和 AWLTF[7:0] 均清零时，AWDF 位才清零。

可将模拟看门狗事件分配至断路输出信号。可将四个断路输出分配至超出阈值上限或低于阈值下限的事件 (dfsdm_break[3:0])。通过 DFSDM_FLTxAWHTR 和 DFSDM_FLTxAWLTR 寄存器中的 BKAWH[3:0] 和 BKAWL[3:0] 字段将断路信号分配至给定模拟看门狗事件。

27.4.11 短路检测器

使用短路检测器的目的在于，当模拟信号达到饱和值（超出满量程范围）并保持给定的时间时，短路检测器可以在极快的响应时间内发出信号。该特性可检测短路或断路故障（例如过流或过压）。可调用中断/事件/断路生成。

短路检测器的输入数据取自通道收发器输出。

各个输入通道上都有一个递增计数器，用于对串行数据接收器输出上的连续 0 或 1 进行计数。如果接收的数据流发生变化（数据信号从 1 到 0 或从 0 到 1），则会重新启动计数器。如果该计数器达到短路阈值寄存器值（DFSDM_CHyAWSCDR 寄存器中的 SCDT[7:0] 位），则会调用短路事件。每个输入通道都配有短路检测器。通过将 SCDEN 位（DFSDM_CHyCFGR1 寄存器中）置 1，可选择对任一通道进行连续监视，每个通道都有其自己的短路检测器设置（阈值位 SCDT[7:0]、状态位 SCDF[7:0]、状态清零位 CLRSCDF[7:0]）。禁止相应通道 y (CHEN[y] = 0) 时，状态标志 SCDF[y] 还可由硬件清零。

对于各个通道，可将短路检测器事件分配至断路输出信号 dfsdm_break[3:0]。可将四个断路输出分配至短路检测器事件。通过 DFSDM_CHyAWSCDR 寄存器中的 BKSCD[3:0] 字段将断路信号分配至给定通道短路检测器事件。

如果选择了并行输入数据通道（DFSDM_CHyCFGR1 寄存器中的 DATMPX[1:0] ≠ 0），则无法使用短路检测器。

总共有四个断路输出（与模拟看门狗功能共享）。

27.4.12 极值检测器

极值检测器用于采集最终输出数据字的最小值和最大值（峰峰值）。

如果输出数据字高于极值检测器最大值寄存器（DFSDM_FLTxEXMAX 寄存器中的 EXMAX[23:0] 位）中存储的值，则该寄存器将用当前输出数据字值进行更新，且存储其数据的通道位于（DFSDM_FLTxEXMAX 寄存器的）EXMAXCH[2:0] 位中。

如果输出数据字低于极值检测器最小值寄存器（DFSDM_FLTxEXMIN 寄存器中的 EXMIN[23:0] 位）中存储的值，则该寄存器将用当前输出数据字值进行更新，且存储其数据的通道位于（DFSDM_FLTxEXMIN 寄存器的）EXMINCH[2:0] 位中。

最小值和最大值寄存器值可由软件（通过读取给定的 DFSDM_FLTxEXMAX 或 DFSDM_FLTxEXMIN 寄存器）进行刷新。刷新后，极值检测器最小值数据寄存器 DFSDM_FLTxEXMIN 用 0x7FFFFF（最大正值）进行填充，极值检测器最大值寄存器 DFSDM_FLTxEXMAX 用 0x800000（最小负值）进行填充。

在完成右移位和偏移数据校正之后，极值检测器才进行比较。对于每个极值检测器，在（DFSDM_FLTxCR2 寄存器中的）EXCH[7:0] 位中选择计算极值时所涉及的输入通道。

27.4.13 数据单元模块

数据单元模块是整个处理路径的最后一个模块：外部 ΣΔ 调制器 —— 串行收发器 —— Sinc 滤波器 —— 积分器 —— 数据单元块。

输出数据速率取决于串行数据流速率以及滤波器和积分器设置。最大输出数据速率为：

$$\text{数据速率[采样/秒]} = \frac{f_{CKIN}}{F_{OSR} \cdot (I_{OSR} - 1 + F_{ORD}) + (F_{ORD} + 1)} \quad \text{...FAST = 0，Sincx 滤波器}$$

$$\text{数据速率[采样/秒]} = \frac{f_{CKIN}}{F_{OSR} \cdot (I_{OSR} - 1 + 4) + (2 + 1)} \quad \text{...FAST = 0，FastSinc 滤波器}$$

或

$$\text{数据速率[采样/秒]} = \frac{f_{CKIN}}{F_{OSR} \cdot I_{OSR}} \quad \text{...FAST = 1}$$

并行数据输入时的最大输出数据速率：

$$\text{数据速率[采样/秒]} = \frac{f_{DATAIN_RATE}}{F_{OSR} \cdot (I_{OSR} - 1 + F_{ORD}) + (F_{ORD} + 1)} \quad \text{...FAST = 0，Sincx 滤波器}$$

或

$$\text{数据采样[采样/秒]} = \frac{f_{DATAIN_RATE}}{F_{OSR} \cdot (I_{OSR} - 1 + 4) + (2 + 1)} \quad \text{...FAST = 0，FastSinc 滤波器}$$

或

$$\text{数据采样[采样/秒]} = \frac{f_{DATAIN_RATE}}{F_{OSR} \cdot I_{OSR}} \quad \text{...FAST = 1，或任何滤波器旁路情况（FOSR = 1）}$$

其中：f_{DATAIN_RATE} = 来自 ADC 或 CPU/DMA 的输入数据速率

在该模块中执行最终数据的右移位，因为最终数据宽度为 24 位，而来自处理路径的数据最高可为 32 位。对于每个所选输入通道，此右移位均可在 0-31 位的范围内进行配置（请参见 DFSDM_CHyCFGR2 寄存器的 DTRBS[4:0] 位）。右移位将结果四舍五入为最接近的整数值。移位结果的符号保留，以便得到有效 24 位有符号格式的结果数据。

接下来对结果进行偏移校正。从给定通道的输出数据中减去偏移校正值（存储在寄存器 DFSDM_CHyCFGR2 中的 OFFSET[23:0]）。OFFSET[23:0] 字段中的数据由软件通过相应的校准程序进行设置。

因数字处理过程中的所有操作均在 32 位有符号寄存器上执行，因此为保证结果不会发生溢出，必须满足以下条件：

$FOSR^{FORD} \cdot IOSR <= 2^{31}$ ... 适用于 $Sinc^x$ 滤波器 (x = 1..5)
$2 \cdot FOSR^2 \cdot IOSR <= 2^{31}$ ... 适用于 FastSinc 滤波器

注： *滤波器和积分器旁路时（IOSR[7:0] = 0，FOSR[9:0] = 0），输入数据速率 (f_{DATAIN_RATE}) 必须被限制为能够读取所有输出数据：*
$f_{DATAIN_RATE} \le f_{APB}$ 其中，f_{APB} 为 DFSDM 外设连接的总线频率。

27.4.14 有符号数据格式

每个 DFSDM 输入串行通道均可连接至一个外部 ΣΔ 调制器。一个外部 ΣΔ 调制器可配有 2 个差分输入（正负），这两个输入可用于差分或单端信号测量。

始终假定 ΣΔ 调制器输出采用有符号格式（来自 ΣΔ 调制器的 0/1 数据流表示值 -1 和 +1）。

寄存器的有符号数据格式：在最终输出数据、模拟看门狗、极值检测器和偏移校正的寄存器中的数据是有符号格式的。输出数据字的 msb 表示值的符号（二进制补码格式）。

27.4.15 启动转换

注入转换可使用以下方法启动：

- 软件：向 DFSDM_FLTxCR1 寄存器中的 JSWSTART 写入“1”。
- 触发：JEXTSEL[4:0] 在 JEXTEN 激活触发时选择触发信号，同时选择有效边沿（请参见 DFSDM_FLTxCR1 寄存器）。
- 如果 JSYNC = 1，则使用 DFSDM_FLT0 同步启动：对于 DFSDM_FLTx (x > 0)，注入转换会在 DFSDM_FLT0 中自动启动；注入转换由软件启动（DFSDM_FLT0CR2 寄存器中的 JSWSTART = 1）。DFSDM_FLTx (x > 0) 中的每个注入转换都始终根据其本地配置设置（JSCAN 和 JCHG 等）执行。

如果使能扫描转换（位 JSCAN = 1），则每次触发注入转换时，注入组（DFSDM_FLTxJCHGR 寄存器中的 JCHG[7:0] 位）的全部所选通道将从最低通道（通道 0，如果已选择）开始，被依次转换。

如果禁止扫描转换（位 JSCAN = 0），则每次触发注入转换时，只会转换注入组（DFSDM_FLTxJCHGR 寄存器中的 JCHG[7:0] 位）的所选通道之一，通道选择随即会移至下一通道。JSCAN = 0 时，对 JCHG[7:0] 位进行写操作会将通道选择置为选择的最低注入通道。

在给定时间内只能进行一次注入转换。因此，如果已发出但尚未完成某个注入转换请求，则会忽略启动注入转换的任何请求。

常规转换可使用以下方法启动：

- 软件：向 DFSDM_FLTxCR1 寄存器中的 RSWSTART 写入“1”。
- 如果 RSYNC = 1，则使用 DFSDM_FLT0 同步启动：对于 DFSDM_FLTx (x > 0)，常规转换会在 DFSDM_FLT0 中自动启动；常规转换由软件启动（DFSDM_FLT0CR2 寄存器中的 RSWSTART = 1）。DFSDM_FLTx (x > 0) 中的每个常规转换都始终根据其本地配置设置（RCONT 和 RCH 等）执行。

在给定时间内只能有一个常规转换处于待处理或正在执行状态。因此，如果已发出但尚未完成某个常规转换请求，则会忽略启动常规转换的任何请求。如果常规转换被注入转换中断，或者常规转换在进行注入转换的过程中启动，则常规转换处于待处理状态。此待处理状态的常规转换随即被延迟，并在完成所有注入转换时执行。延迟的常规转换通过 DFSDM_FLTxRDATAR 寄存器的 RPEND 位进行表示。

27.4.16 连续和快速连续模式

如果将 DFSDM_FLTxCR1 寄存器中的 RCONT 置 1，则会在连续模式下执行常规转换。RCONT = 1 表示向 RSWSTART 写入“1”后，重复转换 RCH[2:0] 所选的通道。

可通过向 RCONT 写入“0”停止在连续模式下执行的常规转换。将 RCONT 清零后，会立即停止正在进行的转换。

在连续模式下，可通过将 DFSDM_FLTxCR1 寄存器的 FAST 位置 1 来提高数据速率。此时，如果对一个通道进行连续转换，则滤波器无需用新的刷新数据重新填充，因为滤波器内部数据有效，这些数据是先前通过对连续数据进行采样而得。速度提升与所选的滤波器阶数有关。通过 RSWSTART = 1 启动连续转换后，将充分进行快速模式 (FAST = 1) 下的首次转换（与 FAST = 0 时相同），之后会在较短的间隔内完成各个后续转换。

连续模式下的转换时间：

FAST = 0 时（或 FAST = 1 时的快速转换）：

对于 $Sinc^x$ 滤波器：

$$t = CNVCNT/f_{DFSDMCLK} = [F_{OSR} * (I_{OSR}-1 + F_{ORD}) + F_{ORD}] / f_{CKIN}$$

对于 FastSinc 滤波器：

$$t = CNVCNT/f_{DFSDMCLK} = [F_{OSR} * (I_{OSR}-1 + 4) + 2] / f_{CKIN}$$

FAST = 1 时（首次转换除外）：

对于 $Sinc^x$ 和 FastSinc 滤波器：

$$t = CNVCNT/f_{DFSDMCLK} = [F_{OSR} * I_{OSR}] / f_{CKIN}$$

F_{OSR} = FOSR[9:0] + 1 = 1 时（滤波器旁路，仅积分器有效）：

$$t = I_{OSR} / f_{CKIN}$$（... 但是 CNVCNT = 0）

连续模式不适用于注入转换。注入转换可通过定时器触发进行启动，以便使用精确时序仿真连续模式。

如果常规连续转换正在进行 (RCONT = 1)，并且如果对请求常规连续转换 (RCONT = 1) 的 DFSDM_FLTxCR1 寄存器进行写访问，则会从下一转换周期重新启动常规连续转换（类似于新的常规连续转换应用于新通道选择，即使 DFSDM_FLTxCR1 寄存器中无更改，亦如此）。

27.4.17 请求优先级

注入转换的优先级高于常规转换。正在进行的常规转换会立即被注入转换请求中断；在注入转换完成后将重新启动此常规转换。

如果某个注入转换处于待处理状态或正在进行，则不能启动其他注入转换：只要（DFSDM_FLTxISR 寄存器中）位 JCIP 为“1”，就会忽略启动注入转换（通过 JSWSTART 或触发）的请求。

同理，如果某个常规转换处于待处理状态或正在进行，则不能启动其他常规转换：只要（DFSDM_FLTxISR 寄存器中）位 RCIP 为“1”，就会忽略启动常规转换（使用 RSWSTART）的请求。

不过，如果在进行某个常规转换时请求注入转换，则常规转换会立即停止，并会启动注入转换。常规转换之后会重新启动，并且此延迟的重新启动会通过位 RPEND 进行表示。

注入转换的优先级高于常规转换，因为注入转换可暂时中断连续的常规转换序列。完成注入转换序列时，如果 RCONT 仍置 1 则会重新启动连续常规转换（RPEND 位将发出基于首次常规转换结果的延迟启动信号）。

通过对 DFSDM 进行相同的写操作启动相关操作时，或者，如果在其他操作结束时有多个操作处于待处理状态，则优先级也至关重要。例如，假设注入转换正在进行 (JCIP = 1)，则对 DFSDM_FLTxCR1 进行单次写操作将向 RSWSTART 写入“1”，以请求常规转换。完成注入序列时，优先级指示接下来执行常规转换，并且通过 RPEND 位指示相应的延迟启动。

27.4.18 在运行模式下的功耗优化

为降低功耗，DFSDM 滤波器和积分器会在未用于转换时自动进入空闲状态（RCIP = 0，JCIP = 0）。

27.5 DFSDM 中断

为提升 CPU 性能，已实现了一组与 CPU 事件发生相关的中断：

- 注入转换结束中断：
  - 由 DFSDM_FLTxCR2 寄存器中的 JEOCIE 位使能
  - 由 DFSDM_FLTxISR 寄存器中的 JEOCF 位指示
  - 通过读取 DFSDM_FLTxJDATAR 寄存器（注入数据）清除
  - 指示哪一个通道发生了转换结束，并在 DFSDM_FLTxJDATAR 寄存器的 JDATACH[2:0] 位中进行报告
- 常规转换结束中断：
  - 由 DFSDM_FLTxCR2 寄存器中的 REOCIE 位使能
  - 由 DFSDM_FLTxISR 寄存器中的 REOCF 位指示
  - 通过读取 DFSDM_FLTxRDATAR 寄存器（常规数据）清除
  - 指示哪一个通道发生了转换结束，并在 DFSDM_FLTxRDATAR 寄存器的 RDATACH[2:0] 位中进行报告

- 注入转换数据溢出中断：
  - 当注入的转换数据未从 DFSDM_FLTxJDATAR 寄存器中读取 （通过 CPU 或 DMA），并被新注入转换覆盖时，会发生此中断
  - 由 DFSDM_FLTxCR2 寄存器中的 JOVRIE 位使能
  - 由 DFSDM_FLTxISR 寄存器中的 JOVRF 位指示
  - 通过向 DFSDM_FLTxICR 寄存器中的 CLRJOVRF 位写入“1”进行清除
- 常规转换数据溢出中断：
  - 当常规转换数据未从 DFSDM_FLTxRDATAR 寄存器中读取（通过 CPU 或 DMA），并被新常规转换覆盖时，会发生此中断
  - 由 DFSDM_FLTxCR2 寄存器中的 ROVRIE 位使能
  - 由 DFSDM_FLTxISR 寄存器中的 ROVRF 位指示
  - 通过向 DFSDM_FLTxICR 寄存器中的 CLRROVRF 位写入“1”进行清除
- 模拟看门狗中断：
  - 当转换数据（输出数据或模拟看门狗滤波器中的数据——具体取决于 DFSDM_FLTxCR1 寄存器中的 AWFSEL 位设置）超出 DFSDM_FLTxAWHTR/ DFSDM_FLTxAWLTR 寄存器中的阈值上限或低于相应阈值下限时，会发生此中断
  - 通过（所选通道 AWDCH[7:0] 的）DFSDM_FLTxCR2 寄存器 AWDIE 位使能
  - 由 DFSDM_FLTxISR 寄存器中的 AWDF 位指示
  - 通过 DFSDM_FLTxAWSR 寄存器中的 AWHTF[7:0] 和 AWLTF[7:0] 字段单独指示模拟看门狗阈值上限或下限错误
  - 通过向 DFSDM_FLTxAWCFR 寄存器中的相应 CLRAWHTF[7:0] 或 CLRAWLTF[7:0] 位写入“1”进行清除
- 短路检测器中断：
  - 稳定的数据数超出 DFSDM_CHyAWSCDR 寄存器中的阈值时会发生此中断
  - 通过（通过 DFSDM_CHyCFGR1 寄存器 SCDEN 位选择的通道的） DFSDM_FLTxCR2 寄存器的 SCDIE 位使能
  - 由 DFSDM_FLTxISR 寄存器的 SCDF[7:0] 位指示（还会报告发生短路检测器事件的通道）
  - 通过向 DFSDM_FLTxICR 寄存器中的相应 CLRSCDF[7:0] 位写入“1”进行清除
- 通道时钟缺失中断：
  - CKINy 引脚上出现时钟缺失时会发生此中断（请参见*第 27.4.4 节：串行通道收发器的时钟缺失检测*）
  - 通过（通过 DFSDM_CHyCFGR1 寄存器 CKABEN 位选择的通道的） DFSDM_FLTxCR2 寄存器的 CKABIE 位使能
  - 由 DFSDM_FLTxISR 寄存器中的 CKABF[y] 位指示
  - 通过向 DFSDM_FLTxICR 寄存器中的 CLRCKABF[y] 位写入“1”进行清除

表 166. DFSDM 中断请求

中断事件	事件标志	事件/中断清除方法	中断使能控制位
注入转换结束	JEOCF	读取 DFSDM_FLTxJDATAR	JEOCIE
常规转换结束	REOCF	读取 DFSDM_FLTxRDATAR	REOCIE
注入数据溢出	JOVRF	写入 CLRJOVRF = 1	JOVRIE
常规数据溢出	ROVRF	写入 CLRROVRF = 1	ROVRIE

表 166. DFSDM 中断请求（续）

中断事件	事件标志	事件/中断清除方法	中断使能控制位
模拟看门狗	AWDF、AWHTF[7:0]、AWLTF[7:0]	写入 CLRAWHTF[7:0] = 1 写入 CLRAWLTF[7:0] = 1	AWDIE、(AWDCH[7:0])
短路检测器	SCDF[7:0]	写入 CLRSCDF[7:0] = 1	SCDIE、(SCDEN)
通道时钟缺失	CKABF[7:0]	写入 CLRCKABF[7:0] = 1	CKABIE、(CKABEN)

27.6 DFSDM DMA 传输

为降低 CPU 干预，可使用 DMA 传输将转换传输至存储器。注入转换的 DMA 传输通过将 DFSDM_FLTxCR1 寄存器的 JDMAEN 位置 1 使能。常规转换的 DMA 传输通过将 DFSDM_FLTxCR1 寄存器的 RDMAEN 位置 1 使能。

注： *通过 DMA 传输，中断标志在注入或常规转换结束时自动清零（DFSDM_FLTxISR 寄存器中的 JEOCF 或 REOCF 位），因为 DMA 正在读取 DFSDM_FLTxJDATAR 或 DFSDM_FLTxRDATAR 寄存器。*

27.7 DFSDM 通道 y 寄存器 (y=0..7)

27.7.1 DFSDM 通道 y 配置寄存器 (DFSDM_CHyCFGR1)

DFSDM channel y configuration register

该寄存器指定通道 y 所使用的参数。

偏移地址：0x00 + 0x20 * y，（y = 0 到 7）

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DFSDMEN	CKOUTSRC	Res.	Res.	Res.	Res.	Res.	Res.	CKOUTDIV[7:0]							
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
DATPACK[1:0]		DATMPX[1:0]		Res.	Res.	Res.	CHINSEL	CHEN	CKABEN	SCDEN	Res.	SPICKSEL[1:0]		SITP[1:0]	
rw	rw	rw	rw				rw	rw	rw	rw		rw	rw	rw	rw

位 31 **DFSDMEN**：DFSDM 接口全局使能 (Global enable for DFSDM interface)

0：禁止 DFSDM 接口

1：使能 DFSDM 接口

如果已使能 DFSDM 接口，则该接口将根据使能的 y 通道和使能的 x 滤波器设置（DFSDM_CHyCFGR1 的 CHEN 位和 DFSDM_FLTxCR1 的 DFEN 位）进行工作。通过设置 DFSDMEN = 0 清除数据：

– 所有寄存器 DFSDM_FLTxISR 均被设为复位状态 (x = 0..3)

– 所有寄存器 DFSDM_FLTxAWSR 均被设为复位状态 (x = 0..3)

注： DFSDMEN 仅存在于 DFSDM_CH0CFGR1 寄存器（通道 y = 0）中

位 30 **CKOUTSRC**：输出串行时钟源选择 (Output serial clock source selection)

0：输出时钟的源来自系统时钟

1：输出时钟的源来自音频时钟

– 可通过 RCC 配置中 SAI1SEL[1:0] 字段选择 SAI1 时钟（请参见*第 6.4.32 节：外设独立时钟配置寄存器 (RCC_CCIPR2)*）。

只有在（DFSDM_CH0CFGR1 寄存器中）DFSDMEN = 0时，才能修改该值。

注： CKOUTSRC 仅存在于 DFSDM_CH0CFGR1 寄存器（通道 y = 0）中

位 29:24 保留，必须保持复位值。

位 23:16 **CKOUTDIV[7:0]**：输出串行时钟分频器 (Output serial clock divider)

0：禁止输出时钟生成（CKOUT 信号被设为低电平状态）

1 - 255：为 CKOUT 信号定义串行时钟输出的系统时钟分频，范围为 2 - 256（分频器值 = CKOUTDIV + 1）。

CKOUTDIV 还定义了时钟缺失检测的阈值。

只有在（DFSDM_CH0CFGR1 寄存器中）DFSDMEN = 0 时，才能修改该值。

如果 DFSDMEN = 0（DFSDM_CH0CFGR1 寄存器中），则 CKOUT 信号被置为低电平状态（在 DFSDMEN = 0 后的一个 DFSDM 时钟周期内执行设置）。

注： CKOUTDIV 仅存在于 DFSDM_CH0CFGR1 寄存器（通道 y = 0）中

位 15:14 **DATPACK[1:0]**：DFSDM_CHyDATINR 寄存器数据封装模式 (Data packing mode in DFSDM_CHyDATINR register)

0：标准：DFSDM_CHyDATINR 寄存器中的输入数据仅存储在 INDAT0[15:0] 中。要清空 DFSDM_CHyDATINR 寄存器，必须由 DFSDM 滤波器从通道 y 中读取一个采样。

1：交错：DFSDM_CHyDATINR 寄存器中的输入数据存储为两个采样：

– 第一个采样位于 INDAT0[15:0] 中（已分配至通道 y）

– 第二个采样位于 INDAT1[15:0] 中（已分配至通道 y）

要清空 DFSDM_CHyDATINR 寄存器，必须由数字滤波器从通道 y 中读取两个采样（INDAT0[15:0] 部分被读为第一个采样，而 INDAT1[15:0] 部分则被读为下一个采样）。

2：双通道：DFSDM_CHyDATINR 寄存器中的输入数据存储为两个采样：

– 第一个采样位于 INDAT0[15:0] 中（已分配至通道 y）

– 第二个样采样于 INDAT1[15:0] 中（已分配至通道 y+1）

要清空 DFSDM_CHyDATINR 寄存器，必须由数字滤波器从通道 y 中读取第一个样本，且必须由另一数字滤波器从通道 y+1 中读取第二个样本。只有在通道数为偶数（y = 0、2、4、6）时才支持双通道模式，对于通道数为奇数（y = 1、3、5、7）的情况，DFSDM_CHyDATINR 被写保护。如果偶数通道被设为双通道模式，则下一奇数通道必须进入标准模式 (DATPACK[1:0] = 0)，以便与偶数通道正确配合。

3：保留

只有在（DFSDM_CHyCFGR1 寄存器中）CHEN = 0时，才能修改该值。

位 13:12 **DATMPX[1:0]**：通道 y 输入数据复用器 (Input data multiplexer for channel y)

0：通道 y 输入数据取自外部串行输入，为 1 位值。DFSDM_CHyDATINR 寄存器被写保护。

1：通道 y 输入数据取自内部模数转换器 ADC_{y+1} 输出寄存器更新，为 16 位值（若 ADC_{y+1} 存在）。ADC 输出数据被写入 DFSDM_CHyDATINR 寄存器的 INDAT0[15:0] 部分。

2：通道 y 输入数据取自内部 DFSDM_CHyDATINR 寄存器（通过 CPU/DMA 直接写操作实现）。可写入一个或两个 16 位数据采样，具体取决于 DATPACK[1:0] 位域设置。

3：保留。

注： 只有在（DFSDM_CHyCFGR1 寄存器中）CHEN = 0时，才能修改该值。

位 11:9 保留，必须保持复位值。

位 8 **CHINSEL**：通道输入选择 (Channel inputs selection)

0：通道输入取自同一通道 y 的引脚。

1：通道输入取自下一通道（通道 (y+1) 取 8 的模）的引脚。

只有在（DFSDM_CHyCFGR1 寄存器中）CHEN = 0时，才能修改该值。

位 7 **CHEN**：通道 y 使能 (Channel y enable)

0：禁止通道 y

1：使能通道 y

如果使能了通道 y，则会根据给定通道设置开始接收串行数据。

位 6 **CKABEN**：通道 y 时钟缺失检测器使能 (Clock absence detector enable on channel y)

0：禁止通道 y 时钟缺失检测器

1：使能通道 y 时钟缺失检测器

位 5 **SCDEN**：通道 y 短路检测器使能 (Short-circuit detector enable on channel y)

0：输入通道 y 将不受短路检测器保护

1：输入通道 y 将持续受短路检测器保护

位 4 保留，必须保持复位值。

位 3:2 **SPICKSEL[1:0]**：通道 y SPI 时钟选择 (SPI clock select for channel y)

0：时钟来自外部 CKINy 输入——基于 SITP[1:0] 的采样点

1：时钟来自内部 CKOUT 输出——基于 SITP[1:0] 的采样点

2：时钟来自内部 CKOUT——在 CKOUT 每个第二个下降沿的采样点。
用于连接外部 ΣΔ 调制器，调制器将时钟输入（来自 CKOUT）除以 2，来生成输出串行通信时钟（该输出时钟变化在各个时钟输入上升沿有效）。

3：时钟来自内部 CKOUT 输出——在 CKOUT 每个第二个上升沿的采样点。
用于连接外部 ΣΔ 调制器，调制器将时钟输入（来自 CKOUT）除以 2，来生成输出串行通信时钟（该输出时钟变化在各个时钟输入下降沿有效）。

只有在（DFSDM_CHyCFGR1 寄存器中）CHEN = 0时，才能修改该值。

位 1:0 **SITP[1:0]**：通道 y 串行接口类型 (Serial interface type for channel y)

00：上升沿选通数据 SPI

01：下降沿选通数据 SPI

10：DATINy 引脚曼彻斯特编码输入：上升沿 = 逻辑 0，下降沿 = 逻辑 1

11：DATINy 引脚曼彻斯特编码输入：上升沿 = 逻辑 1，下降沿 = 逻辑 0

只有在（DFSDM_CHyCFGR1 寄存器中的）CHEN = 0 时，才能修改该值。

27.7.2 DFSDM 通道 y 配置寄存器 (DFSDM_CHyCFGR2)

DFSDM channel y configuration register

该寄存器指定通道 y 所使用的参数。

偏移地址：0x04 + 0x20 * y，（y = 0 到 7）

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
OFFSET[23:8]															
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
OFFSET[7:0]								DTRBS[4:0]					Res.	Res.	Res.
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw			

位 31:8 **OFFSET[23:0]**：通道 y 24 位校准偏移 (24-bit calibration offset for channel y)

对于通道 y 的每一次转换结果，都执行 OFFSET。

该值由软件设置。

位 7:3 **DTRBS[4:0]**：通道 y 数据右移位 (Data right bit-shift for channel y)

0-31：定义积分器输出数据结果的移位——将向右移多少位以获得最终结果。在进行偏移校正之前执行移位。数据移位将结果四舍五入为最接近的整数值。移位结果的符号保留（以便得到有效 24 位有符号格式的结果数据）。

只有在（DFSDM_CHyCFGR1 寄存器中）CHEN = 0 时，才能修改该值。

位 2:0 保留，必须保持复位值。

27.7.3 DFSDM 通道 y 模拟看门狗和短路检测器寄存器 (DFSDM_CHyAWSCDR)

DFSDM channel y analog watchdog and short-circuit detector register

通道 y 短路检测器和模拟看门狗设置

偏移地址：0x08 + 0x20 * y，（y = 0 到 7）

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.	AWFORD[1:0]		Res.	AWFOSR[4:0]				
								rw	rw		rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BKSCD[3:0]				Res.	Res.	Res.	Res.	SCDT[7:0]							
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 23:22 **AWFORD[1:0]**：通道 y 模拟看门狗 Sinc 滤波器阶数 (Analog watchdog Sinc filter order on channel y)

0：FastSinc 滤波器类型
1：$Sinc^1$ 滤波器类型
2：$Sinc^2$ 滤波器类型
3：$Sinc^3$ 滤波器类型

$Sinc^x$ 滤波器类型传递函数：

$$H(z) = \left(\frac{1 - z^{-FOSR}}{1 - z^{-1}}\right)^x$$

FastSinc 滤波器类型传递函数：

$$H(z) = \left(\frac{1 - z^{-FOSR}}{1 - z^{-1}}\right)^2 \cdot (1 + z^{-(2 \cdot FOSR)})$$

只有在（DFSDM_CHyCFGR1 寄存器中）CHEN = 0时，才能修改该位。

位 21 保留，必须保持复位值。

位 20:16 **AWFOSR[4:0]**：通道 y 模拟看门狗滤波器过采样率（抽取率）(Analog watchdog filter oversampling ratio (decimation rate) on channel y)

0 - 31：定义 Sinc 类型滤波器的长度，长度范围为 1 - 32 (AWFOSR + 1)。该数字还表示模拟数据速率的抽取率。

只有在（DFSDM_CHyCFGR1 寄存器中）CHEN = 0 时，才能修改该位。

注： 如果 AWFOSR = 0，则滤波器无影响（滤波器旁路）。

位 15:12 **BKSCD[3:0]**：通道 y 短路检测器断路信号分配 (Break signal assignment for short-circuit detector on channel y)

BKSCD[i] = 0：断路 i 信号未分配至通道 y 短路检测器
BKSCD[i] = 1：断路 i 信号已分配至通道 y 短路检测器

位 11:8 保留，必须保持复位值。

位 7:0 **SCDT[7:0]**：通道 y 的短路检测器阈值 (short-circuit detector threshold for channel y)

这些位可由软件写入，用来定义短路检测器的计数器阈值。如果达到该值，则在给定通道上发生短路检测器事件。

27.7.4 DFSDM 通道 y 看门狗滤波器数据寄存器 (DFSDM_CHyWDATR)

DFSDM channel y watchdog filter data register

该寄存器包含的数据来自于与输入通道 y 关联的模拟看门狗滤波器。

偏移地址：0x0C + 0x20 * y，（y = 0 到 7）

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
WDATA[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 **WDATA[15:0]**：输入通道 y 看门狗数据 (Input channel y watchdog data)
由输入通道 y 模拟看门狗滤波器转换的数据。该通道进行连续的数据转换（无触发），转换分辨率受一定限制（OSR=1...32/sinc 阶数 = 1...3）。

27.7.5 DFSDM 通道 y 数据输入寄存器 (DFSDM_CHyDATINR)

DFSDM channel y data input register

该寄存器包含的 16 位输入数据将通过 DFSDM 滤波器模块进行处理。

偏移地址：0x10 + 0x20 * y，（y = 0 到 7）

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
INDAT1[15:0]															
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
INDAT0[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:16 **INDAT1[15:0]**：通道 y 或通道 y+1 的输入数据 (Input data for channel y or channel y+1)

当 DATMPX[1:0] = 1 或 DATMPX[1:0] = 2 时，输入并行通道数据将通过数字滤波器进行处理。

数据可由 CPU/DMA 写入（当 DATMPX[1:0] = 2 时），也可以由内部 ADC 直接写入（当 DATMPX[1:0] = 1 时）。

当 DATPACK[1:0] = 0（标准模式）时

INDAT0[15:0] 被写保护（不用于输入采样）。

当 DATPACK[1:0] = 1（交错模式）时

第二个通道 y 数据采样存储到 INDAT1[15:0] 中。第一个通道 y 数据采样存储到 INDAT0[15:0] 中。

DFSDM_FLTx 滤波器依次读取该两个采样，作为两个通道 y 数据采样。

当 DATPACK[1:0] = 2（双模式）时

对于偶数 y 通道：INDAT1[15:0] 中的采样自动复制到通道 (y+1) 的 INDAT0[15:0] 中。

对于奇数 y 通道：INDAT1[15:0] 被写保护。

请参见*第 27.4.6 节：并行数据输入*以获得更多信息。

INDAT0[15:1] 采用 16 位带符号格式。

位 15:0 **INDAT0[15:0]**：通道 y 输入数据 (Input data for channel y)

当 DATMPX[1:0] = 1 或 DATMPX[1:0] = 2 时，输入并行通道数据将通过数字滤波器进行处理。

数据可由 CPU/DMA 写入（当 DATMPX[1:0] = 2 时），也可以由内部 ADC 直接写入（当 DATMPX[1:0] = 1 时）。

当 DATPACK[1:0] = 0（标准模式）时

通道 y 数据采样存储到 INDAT0[15:0] 中。

当 DATPACK[1:0] = 1（交错模式）时

第一个通道 y 数据采样存储到 INDAT0[15:0] 中。第二个通道 y 数据采样存储到 INDAT1[15:0] 中。

DFSDM_FLTx 滤波器依次读取该两个采样，作为两个通道 y 数据采样。

当 DATPACK[1:0] = 2（双模式）时

对于偶数 y 通道：通道 y 数据采样存储到 INDAT0[15:0] 中。

对于奇数 y 通道：INDAT0[15:0] 被写保护。

请参见*第 27.4.6 节：并行数据输入*以获得更多信息。

INDAT0[15:0] 采用 16 位带符号格式。

27.7.6 DFSDM 通道 y 延迟寄存器 (DFSDM_CHyDLYR)

DFSDM channel y delay register

偏移地址：0x14 + 0x20 * y，（y = 0 到 7）

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.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PLSSKP[5:0]					
										rw	rw	rw	rw	rw	rw

位 31:6 保留，必须保持复位值。

位 5:0 **PLSSKP[5:0]**：面向输入数据跳过功能的要跳过脉冲数 (Pulses to skip for input data skipping function)

0-63：定义要跳过的串行输入样品数。写入该字段后，立即应用跳过功能。读取 PLSSKP[5:0]，可返回要跳过的脉冲的当前值。如果 PLSSKP[5:0]=0，则已跳过全部所需数据样品。

注： PLSSKP[5:0] 不为零时，用户也可更新 PLSSKP[5:0]。

27.8 DFSDM 滤波器 x 模块寄存器 (x=0..3)

27.8.1 DFSDM 滤波器 x 控制寄存器 1 (DFSDM_FLTxCR1)

DFSDM filter x control register 1

偏移地址：0x100 + 0x80 * x，（x = 0 到 3）

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	AWFSEL	FAST	Res.	Res.	RCH[2:0]			Res.	Res.	RDMAEN	Res.	RSYNC	RCONT	RSWSTART	Res.
	rw	rw			rw	rw	rw			rw		rw	rw	rt_w1	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	JEXTEN[1:0]		JEXTSEL[4:0]					Res.	Res.	JDMAEN	JSCAN	JSYNC	Res.	JSWSTART	DFEN
	rw	rw	rw	rw	rw	rw	rw			rw	rw	rw		rt_w1	rw

位 31 保留，必须保持复位值。

位 30 **AWFSEL**：模拟看门狗快速模式选择 (Analog watchdog fast mode select)

0：数据输出值的模拟看门狗（在数字滤波器之后）。在偏移校正和移位后进行比较

1：通道收发器值的模拟看门狗（在看门狗滤波器之后）

位 29 **FAST**：常规转换的快速转换模式选择 (Fast conversion mode selection for regular conversions)

0：禁止快速转换模式

1：使能快速转换模式

在连续模式下进行常规转换时，如果使能了快速模式，则每次转换（第一次转换除外）都比在标准模式下的转换执行得快。该位对于不连续的转换没有影响。

只有在 DFEN = 0 (DFSDM_FLTxCR1) 时才能修改该位。

如果 FAST=0（或者 FAST=1 连续模式下的第一次转换），则：

$t = [F_{OSR} * (I_{OSR}-1 + F_{ORD}) + F_{ORD}] / f_{CKIN}$.....（适用于 $Sinc^x$ 滤波器）

$t = [F_{OSR} * (I_{OSR}-1 + 4) + 2] / f_{CKIN}$.....（适用于 FastSinc 滤波器）

如果在连续模式下 FAST=1（第一次转换除外），则：

$t = [F_{OSR} * I_{OSR}] / f_{CKIN}$

如果 $F_{OSR} = F_{OSR}[9:0] + 1 = 1$ 时（滤波器旁路，仅积分器有效），则：

$t = I_{OSR} / f_{CKIN}$（... 但是 CNVCNT = 0）

其中，f_{CKIN} 为（给定通道 CKINy 引脚上的）通道输入时钟频率，在并行数据输入的情况下，其表示输入数据速率。

位 28:27 保留，必须保持复位值。

位 26:24 **RCH[2:0]**：常规通道选择 (Regular channel selection)

0：将通道 0 选作常规通道

1：将通道 1 选作常规通道

...

7：将通道 7 选作常规通道

当 RCIP = 1 时写入这些位，并在下一个常规转换开始时生效。这在连续模式下（当 RCONT = 1 时）特别有用。它还会影响待处理的常规转换（因正在进行注入转换而暂停）。

位 23:22 保留，必须保持复位值。

位 21 **RDMAEN**：使能 DMA 通道以读取常规转换数据 (DMA channel enabled to read data for the regular conversion)

0：未使能 DMA 通道来读取常规数据

1：已使能 DMA 通道来读取常规数据

只有在 DFEN = 0 (DFSDM_FLTxCR1) 时才能修改该位。

位 20 保留，必须保持复位值。

位 19 **RSYNC**：使用 DFSDM_FLT0 同步启动常规转换 (Launch regular conversion synchronously with DFSDM_FLT0)

0：不使用 DFSDM_FLT0 同步启动常规转换

1：在 DFSDM_FLT0 中启动常规转换的同时，在该 DFSDM_FLTx 中启动常规转换

只有在 DFEN = 0 (DFSDM_FLTxCR1) 时才能修改该位。

位 18 **RCONT**：常规转换的连续模式选择 (Continuous mode selection for regular conversions)

0：对于每个转换请求，只转换一次常规通道

1：在发出每个转换请求后，重复转换常规通道

在连续常规转换进行过程中，向该位写入“0”将立即停止连续模式。

位 17 **RSWSTART**：软件启动常规通道转换 (Software start of a conversion on the regular channel)

0：写入“0”无影响

1：写入“1”会发出启动常规通道转换的请求，并会使 RCIP 变为“1”。如果已经设置 RCIP=1，则对 RSWSTART 进行的写操作无效。如果 RSYNC = 1，则写入“1”无效。

此位始终读为“0”。

位 16:15 保留，必须保持复位值。

位 14:13 **JEXTEN[1:0]**：注入转换触发使能和触发边沿选择 (Trigger enable and trigger edge selection for injected conversions)

00：禁止触发检测

01：所选触发的每个上升沿都会发出启动注入转换的请求

10：所选触发的每个下降沿都会发出启动注入转换的请求

11：所选触发的上升沿和下降沿都会发出启动注入转换的请求

只有在 DFEN = 0 (DFSDM_FLTxCR1) 时才能修改该位。

位 12:8 **JEXTSEL[4:0]**：启动注入转换的触发信号选择 (Trigger signal selection for launching injected conversions)

0x0-0x1F：根据下表选择的触发输入（内部或外部触发）。

只有在 DFEN = 0 (DFSDM_FLTxCR1) 时才能修改该位。

注： 同步触发的延迟最大为一个 $f_{DFSDMCLK}$ 时钟周期（存在确定的抖动），异步触发的延迟为 2-3 个 $f_{DFSDMCLK}$ 时钟周期（抖动长达 1 个周期）。

	DFSDM_FLT0	DFSDM_FLT1	DFSDM_FLT2	DFSDM_FLT3
0x00	dfsdm_jtrg0	dfsdm_jtrg0	dfsdm_jtrg0	dfsdm_jtrg0
0x01	dfsdm_jtrg1	dfsdm_jtrg1	dfsdm_jtrg1	dfsdm_jtrg1
...				
0x1E	dfsdm_jtrg30	dfsdm_jtrg30	dfsdm_jtrg30	dfsdm_jtrg30
0x1F	dfsdm_jtrg31	dfsdm_jtrg31	dfsdm_jtrg31	dfsdm_jtrg31

请参见*表 162：DFSDM 触发器连接*。

位 7:6 保留，必须保持复位值。

位 5 **JDMAEN**：为注入通道组使能 DMA 以读取数据 (DMA channel enabled to read data for the injected channel group)

0：未使能 DMA 通道来读取注入数据

1：已使能 DMA 通道来读取注入数据

只有在 DFEN = 0 (DFSDM_FLTxCR1) 时才能修改该位。

位 4 **JSCAN**：注入转换的扫描转换模式 (Scanning conversion mode for injected conversions)

0：对注入通道组执行一次通道转换，然后选择该通道组下一个通道。

1：从选择的最低通道开始，对注入通道组执行连续转换。

只有在 DFEN = 0 (DFSDM_FLTxCR1) 时才能修改该位。

如果 JSCAN=0，则对 JCHG 进行写入操作会将通道选择复位为最小所选通道。

位 3 **JSYNC**：使用 DFSDM_FLT0 JSWSTART 触发同步启动注入转换 (Launch an injected conversion synchronously with the DFSDM_FLT0 JSWSTART trigger)

0：不使用 DFSDM_FLT0 同步启动注入转换

1：在 DFSDM_FLT0 中通过 JSWSTART 触发启动注入转换的同时，在该 DFSDM_FLTx 中启动注入转换

只有在 DFEN = 0 (DFSDM_FLTxCR1) 时才能修改该位。

位 2 保留，必须保持复位值。

位 1 **JSWSTART**：启动注入通道组转换 (Start a conversion of the injected group of channels)

0：写入“0”无影响。

1：写入“1”会发出转换注入通道组请求，同时会使 JCIP 变为“1”。如果已经设置 JCIP = 1，则对 JSWSTART 进行的写操作无效。如果 JSYNC = 1，则写入“1”无效。

此位始终读为“0”。

位 0 **DFEN**：DFSDM_FLTx 使能 (DFSDM_FLTx enable)

0：禁止 DFSDM_FLTx。给定 DFSDM_FLTx 的所有转换均立即停止，DFSDM_FLTx 的所有功能均停止。

1：使能 DFSDM_FLTx。如果使能了 DFSDM_FLTx，则 DFSDM_FLTx 会根据其设置开始工作。

通过设置 DFEN = 0 清零的数据：

– 寄存器 DFSDM_FLTxISR 被设为复位状态

– 寄存器 DFSDM_FLTxAWSR 被设为复位状态

27.8.2 DFSDM 滤波器 x 控制寄存器 2 (DFSDM_FLTxCR2)

DFSDM filter x control register 2

偏移地址：0x104 + 0x80 * x，（x = 0 到 3）

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.	AWDCH[7:0]							
								rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EXCH[7:0]								Res.	CKABIE	SCDIE	AWDIE	ROVRIE	JOVRIE	REOCIE	JEOCIE
rw	rw	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 23:16 **AWDCH[7:0]**：模拟看门狗通道选择 (Analog watchdog channel selection)
这些位用于选择由模拟看门狗持续监控的输入通道。
AWDCH[y] = 0：在通道 y 上禁止模拟看门狗
AWDCH[y] = 1：在通道 y 上使能模拟看门狗

位 15:8 **EXCH[7:0]**：极值检测器通道选择 (Extremes detector channel selection)
这些位选择极值检测器要采用的输入通道。
EXCH[y] = 0：极值检测器不接受通道 y 的数据
EXCH[y] = 1：极值检测器接受通道 y 的数据

位 7 保留，必须保持复位值。

位 6 **CKABIE**：时钟缺失中断使能 (Clock absence interrupt enable)
0：禁止检测通道输入时钟缺失中断
1：使能检测通道输入时钟缺失中断
请参见 DFSDM_FLTxISR 中的 CKABF[7:0] 说明。
注： CKABIE 仅存在于 DFSDM_FLT0CR2 寄存器中（滤波器 x = 0）

位 5 **SCDIE**：短路检测器中断使能 (Short-circuit detector interrupt enable)
0：禁止短路检测器中断
1：使能短路检测器中断
请参见 DFSDM_FLTxISR 中的 SCDF[7:0] 说明。
注： SCDIE 仅存在于 DFSDM_FLT0CR2 寄存器中（滤波器 x = 0）

位 4 **AWDIE**：模拟看门狗中断使能 (Analog watchdog interrupt enable)
0：禁止模拟看门狗中断
1：使能模拟看门狗中断
请参见 DFSDM_FLTxISR 中的 AWDF 说明。

位 3 **ROVRIE**：常规数据溢出中断使能 (Regular data overrun interrupt enable)
0：禁止常规数据溢出中断
1：使能常规数据溢出中断
请参见 DFSDM_FLTxISR 中的 ROVRF 说明。

位 2 **JOVRIE**：注入数据溢出中断使能 (Injected data overrun interrupt enable)
0：禁止注入数据溢出中断
1：使能注入数据溢出中断
请参见 DFSDM_FLTxISR 中的 JOVRF 说明。

位 1 **REOCIE**：常规转换结束中断使能 (Regular end of conversion interrupt enable)
0：禁止常规转换结束中断
1：使能常规转换结束中断
请参见 DFSDM_FLTxISR 中的 REOCF 说明。

位 0 **JEOCIE**：注入转换结束中断使能 (Injected end of conversion interrupt enable)
0：禁止注入转换结束中断
1：使能注入转换结束中断
请参见 DFSDM_FLTxISR 中的 JEOCF 说明。

27.8.3 DFSDM 滤波器 x 中断和状态寄存器 (DFSDM_FLTxISR)

DFSDM filter x interrupt and status register

偏移地址：0x108 + 0x80 * x，x = 0..3

复位值：0x00FF 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SCDF[7:0]								CKABF[7:0]							
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	RCIP	JCIP	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	AWDF	ROVRF	JOVRF	REOCF	JEOCF
	r	r									r	r	r	r	r

位 31:24 **SCDF[7:0]**：短路检测器标志 (short-circuit detector flag)
SDCF[y] = 0：通道 y 上未发生任何短路检测器事件
SDCF[y] = 1：通道 y 上的短路检测器计数器达到在 DFSDM_CHyAWSCDR 寄存器中编程的值
此位将由硬件置 1，该位可由软件使用 DFSDM_FLTxICR 寄存器中的相应 CLRSCDF[y] 位清零。当 CHEN[y] = 0（给定通道被禁止）时，SCDF[y] 还可以由硬件清零。
注： SCDF[7:0] 仅存在于 DFSDM_FLT0ISR 寄存器中（滤波器 x = 0）

位 23:16 **CKABF[7:0]**：时钟缺失标志 (Clock absence flag)
CKABF[y] = 0：通道 y 上存在时钟信号。
CKABF[y] = 1：通道 y 上不存在时钟信号。
当检测到通道 y 上不存在时钟信号时，给定 y 位由硬件置 1。当 CHEN = 0 时（请参见 DFSDM_CHyCFGR1 寄存器），它由硬件保持为 CKABF[y] = 1 状态。当收发器尚未被同步时，它由硬件保持为 CKABF[y] = 1 状态。它可由软件使用 DFSDM_FLTxICR 寄存器中的相应 CLRCKABF[y] 位清零。
注： CKABF[7:0] 仅存在于 DFSDM_FLT0ISR 寄存器中（滤波器 x = 0）

位 15 保留，必须保持复位值。

位 14 **RCIP**：常规转换正在进行状态 (Regular conversion in progress status)
0：未发出转换常规通道的请求
1：常规通道转换正在进行，或一个常规转换请求正在等待处理
当 RCIP = 1 时，忽略启动常规转换的请求。

位 13 **JCIP**：注入转换正在进行状态 (Injected conversion in progress status)

0：未发出转换注入通道组的请求（软件和触发器均未发出请求）

1：注入通道组转换处于正在进行状态（向 JSWSTART 写入了“1”）或者一个注入转换请求正在等待处理（由于触发检测）

当 JCIP = 1 时，忽略启动注入转换的请求。

位 12:5 保留，必须保持复位值。

位 4 **AWDF**：模拟看门狗 (Analog watchdog)

0：未发生模拟看门狗事件

1：模拟看门狗模块检测到电压超出在 DFSDM_FLTxAWLTR 或 DFSDM_FLTxAWHTR 寄存器中编程的值。

此位将由硬件置 1，它由软件通过以下方式清零：将 DFSDM_FLTxAWSR 寄存器中的所有源标志位 AWHTF[7:0] 和 AWLTF[7:0] 清零（通过向 DFSDM_FLTxAWCFR 寄存器中的清零位写入“1”实现）。

位 3 **ROVRF**：常规转换溢出标志 (Regular conversion overrun flag)

0：未发生常规转换溢出

1：发生了常规转换溢出，也就是说，常规转换已经完成，但 REOCF 仍然为“1”。RDATAR 不受溢出影响

此位将由硬件置 1，该位可由软件使用 DFSDM_FLTxICR 寄存器中的 CLRROVRF 位清零。

位 2 **JOVRF**：注入转换溢出标志 (Injected conversion overrun flag)

0：未发生注入转换溢出

1：发生了注入转换溢出，也就是说，注入转换已经完成，但 JEOCF 仍然为“1”。JDATAR 不受溢出影响

此位将由硬件置 1，该位可由软件使用 DFSDM_FLTxICR 寄存器中的 CLRJOVRF 位清零。

位 1 **REOCF**：常规转换结束标志 (End of regular conversion flag)

0：未完成常规转换

1：已完成常规转换并可以读取其数据

此位将由硬件置 1，当软件或 DMA 读取 DFSDM_FLTxRDATAR 时，该位清零。

位 0 **JEOCF**：注入转换结束标志 (End of injected conversion flag)

0：未完成注入转换

1：已完成注入转换并可以读取其数据

此位将由硬件置 1，当软件或 DMA 读取 DFSDM_FLTxJDATAR 时，该位清零。

注： *对于每个标志位，可通过将 DFSDM_FLTxCR2 中的相应位置 1 来使能中断。如果调用了中断，则必须在退出中断服务程序之前将该标志清零。*

当 DFEN = 0 时，DFSDM_FLTxISR 的所有位都会自动复位。

27.8.4 DFSDM 滤波器 x 中断标志清零寄存器 (DFSDM_FLTxICR)

DFSDM filter x interrupt flag clear register

偏移地址：0x10C + 0x80 * x，（x = 0 到 3）

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CLRSCDF[7:0]								CLRCKABF[7:0]							
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
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CLRROVRF	CLRJOVRF	Res.	Res.
												rc_w1	rc_w1		

位 31:24 **CLRSCDF[7:0]**：清零短路检测器标志 (Clear the short-circuit detector flag)

CLRSCDF[y] = 0：写入“0”无影响

CLRSCDF[y] = 1：向位置 y 写入“1”会将 DFSDM_FLTxISR 寄存器中的相应 SCDF[y] 位清零

注： CLRSCDF[7:0] 仅存在于 DFSDM_FLT0ICR 寄存器中（滤波器 x = 0）

位 23:16 **CLRCKABF[7:0]**：清零时钟缺失标志 (Clear the clock absence flag)

CLRCKABF[y] = 0：写入“0”无影响

CLRCKABF[y] = 1：向位置 y 写入“1”会将 DFSDM_FLTxISR 寄存器中的相应 CKABF[y] 位清零。当收发器尚未被同步时，时钟缺失标志会置 1，且无法由 CLRCKABF[y] 清零。

注： CLRCKABF[7:0] 仅存在于 DFSDM_FLT0ICR 寄存器中（滤波器 x = 0）

位 15:4 保留，必须保持复位值。

位 3 **CLRROVRF**：将常规转换溢出标志清零 (Clear the regular conversion overrun flag)

0：写入“0”无影响

1：写入“1”会将 DFSDM_FLTxISR 寄存器中的 ROVRF 位清零

位 2 **CLRJOVRF**：将注入转换溢出标志清零 (Clear the injected conversion overrun flag)

0：写入“0”无影响

1：写入“1”会将 DFSDM_FLTxISR 寄存器中的 JOVRF 位清零

位 1:0 保留，必须保持复位值。

注： DFSDM_FLTxICR 的位始终读为“0”。

27.8.5 DFSDM 滤波器 x 注入通道组选择寄存器 (DFSDM_FLTxJCHGR)

DFSDM filter x injected channel group selection register

偏移地址：0x110 + 0x80 * x，（x = 0 到 3）

复位值：0x0000 0001

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.	Res.	Res.	Res.	Res.	Res.	Res.	JCHG[7:0]							
								rw	rw	rw	rw	rw	rw	rw	rw

位 31:8 保留，必须保持复位值。

位 7:0 **JCHG[7:0]**：注入通道组选择 (Injected channel group selection)

JCHG[y] = 0：通道 y 不属于注入组

JCHG[y] = 1：通道 y 属于注入组

如果 JSCAN = 1，则依次转换每一个所选通道。首先转换最小通道（通道 0，如果已选择），最后转换最大所选通道。

如果 JSCAN = 0，则只转换其中一个所选通道，然后通道选择移至下一个通道。如果 JSCAN=0，则对 JCHG 进行写入操作会将通道选择复位为最小所选通道。

必须始终至少为注入组选择一个通道。如果写操作导致所有 JCHG 位均变为零，则忽略写操作。

27.8.6 DFSDM 滤波器 x 控制寄存器 (DFSDM_FLTxFCR)

DFSDM filter x control register

偏移地址：0x114 + 0x80 * x，x = 0..3

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FORD[2:0]			Res.	Res.	Res.	FOSR[9:0]									
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.	Res.	Res.	Res.	Res.	Res.	IOSR[7:0]							
								rw	rw	rw	rw	rw	rw	rw	rw

位 31:29 **FORD[2:0]**：Sinc 滤波器阶数 (Sinc filter order)

0：FastSinc 滤波器类型
1：$Sinc^1$ 滤波器类型
2：$Sinc^2$ 滤波器类型
3：$Sinc^3$ 滤波器类型
4：$Sinc^4$ 滤波器类型
5：$Sinc^5$ 滤波器类型
6-7：保留
$Sinc^x$ 滤波器类型传递函数：

$$H(z) = \left(\frac{1 - z^{-FOSR}}{1 - z^{-1}}\right)^x$$

FastSinc 滤波器类型传递函数：

$$H(z) = \left(\frac{1 - z^{-FOSR}}{1 - z^{-1}}\right)^2 \cdot (1 + z^{-(2 \cdot FOSR)})$$

只有在 DFEN = 0 (DFSDM_FLTxCR1) 时才能修改该位。

位 28:26 保留，必须保持复位值。

位 25:16 **FOSR[9:0]**：Sinc 滤波器过采样率（抽取率）(Sinc filter oversampling ratio (decimation rate))

0 - 1023：定义 Sinc 类型滤波器的长度，长度范围为 1 - 1024 (F_{OSR} = FOSR[9:0] + 1)。该数字还表示滤波器输出数据速率的抽取率。
只有在 DFEN = 0 (DFSDM_FLTxCR1) 时才能修改该位。
注： 如果 FOSR = 0，则滤波器无影响（滤波器旁路）。

位 15:8 保留，必须保持复位值。

位 7:0 **IOSR[7:0]**：积分器过采样率（平均长度）(Integrator oversampling ratio (averaging length))

0 - 255：积分器的长度，长度范围为 1 - 256 (IOSR + 1)。定义一个积分器输出数据采样是由多少个 Sinc 滤波器采样而合成的。积分器的输出数据速率将被减去该数值（额外数据抽取率）。
只有在 DFEN = 0 (DFSDM_FLTxCR1) 时才能修改该位。
注： 如果 IOSR = 0，则积分器无影响（积分器旁路）。

27.8.7 注入组的 DFSDM 滤波器 x 数据寄存器 (DFSDM_FLTxJDATAR)

DFSDM filter x data register for injected group

偏移地址：0x118 + 0x80 * x，（x = 0 到 3）

复位值：0x0000 0000

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

位 31:8 **JDATA[23:0]**：注入组转换数据 (Injected group conversion data)
每次转换完注入组中的一个通道时，得到的数据都会存储在该字段中。JEOCF =1 时数据有效。读取该寄存器会将相应的 JEOCF 清零。

位 7:3 保留，必须保持复位值。

位 2:0 **JDATACH[2:0]**：最近转换的注入通道 (Injected channel most recently converted)
每次转换完注入组中的一个通道时，都会更新 JDATACH[2:0] 以指示转换了哪一个通道。因此，JDATA[23:0] 保持的数据对应于 JDATACH[2:0] 所指示的通道。

注： *可以使用 DMA 从该寄存器中读取数据。可以使用半字访问只读取转换数据的最高有效位 (MSB)。*

读取该寄存器还会将 DFSDM_FLTxISR 的 JEOCF 清零。因此，如果为从该寄存器中读取数据而激活了 DMA，则固件不能读取该寄存器。

27.8.8 常规通道的 DFSDM 滤波器 x 数据寄存器 (DFSDM_FLTxRDATAR)

DFSDM filter x data register for the regular channel

偏移地址：0x11C + 0x80 * x，（x = 0 到 3）

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RDATA[23:8]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

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

位 31:8 **RDATA[23:0]**：常规通道转换数据 (Regular channel conversion data)
每次完成常规转换时，相应的数据都会存储在该寄存器中。当 REOCF =1 时，数据有效。读取该寄存器会将相应的 REOCF 清零。

位 7:5 保留，必须保持复位值。

位 4 **RPEND**：常规通道待处理数据 (Regular channel pending data)
因在转换期间发生注入通道触发而导致 RDATA[23:0] 中的常规数据被延迟处理

位 3 保留，必须保持复位值。

位 2:0 **RDATACH[2:0]**：最近转换的常规通道 (Regular channel most recently converted)
每次完成常规转换时，都会更新 RDATACH[2:0] 以指示转换了哪一个通道（因为在常规转换期间，可更新 DFSDM_FLTxCR1 寄存器中的常规通道选择 RCH[2:0]）。因此，RDATA[23:0] 保持的数据对应于 RDATACH[2:0] 所指示的通道。

注： *可以使用半字访问只读取转换数据的最高有效位 (MSB)。*

读取该寄存器还会将 DFSDM_FLTxISR 的 REOCF 清零。

27.8.9 DFSDM 滤波器 x 模拟看门狗阈值上限寄存器 (DFSDM_FLTxAWHTR)

DFSDM filter x analog watchdog high threshold register

偏移地址：0x120 + 0x80 * x，（x = 0 到 3）

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
AWHT[23:8]															
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
AWHT[7:0]								Res.	Res.	Res.	Res.	BKAWH[3:0]			
rw	rw	rw	rw	rw	rw	rw	rw					rw	rw	rw	rw

位 31:8 **AWHT[23:0]**：模拟看门狗阈值上限 (Analog watchdog high threshold)

这些位由软件写入，用于定义模拟看门狗的阈值上限。

注： 对于通道收发器监视器 (AWFSEL = 1)，高 16 位 (AWHT[23:8]) 会定义 16 位阈值，该阈值与模拟看门狗滤波器输出进行比较（因为来自模拟看门狗滤波器的数据的分辨率最高为 16 位）。在这种情况下，不对位 AWHT[7:0] 进行比较。

位 7:4 保留，必须保持复位值。

位 3:0 **BKAWH[3:0]**：模拟看门狗阈值上限事件的断路信号分配 (Break signal assignment to analog watchdog high threshold event)

BKAWH[i] = 0：断路 i 信号不分配到模拟看门狗阈值上限事件

BKAWH[i] = 1：断路 i 信号分配到模拟看门狗阈值上限事件

27.8.10 DFSDM 滤波器 x 模拟看门狗阈值下限寄存器 (DFSDM_FLTxAWLTR)

DFSDM filter x analog watchdog low threshold register

偏移地址：0x124 + 0x80 * x，（x = 0 到 3）

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
AWLT[23:8]															
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
AWLT[7:0]								Res.	Res.	Res.	Res.	BKAWL[3:0]			
rw	rw	rw	rw	rw	rw	rw	rw					rw	rw	rw	rw

位 31:8 **AWLT[23:0]**：模拟看门狗阈值下限 (Analog watchdog low threshold)

这些位由软件写入，用于定义模拟看门狗的阈值下限。

注： 对于通道收发器监视 (AWFSEL = 1)，仅高 16 位 (AWLT[23:8]) 定义 16 位阈值，该阈值与模拟看门狗滤波器输出进行比较（因为来自模拟看门狗滤波器的数据的分辨率最高为 16 位）。在这种情况下，不对位 AWLT[7:0] 进行比较。

位 7:4 保留，必须保持复位值。

位 3:0 **BKAWL[3:0]**：模拟看门狗阈值下限事件的断路信号分配 (Break signal assignment to analog watchdog low threshold event)

BKAWL[i] = 0：断路 i 信号不分配到模拟看门狗阈值下限事件
BKAWL[i] = 1：断路 i 信号分配到模拟看门狗阈值下限事件

27.8.11 DFSDM 滤波器 x 模拟看门狗状态寄存器 (DFSDM_FLTxAWSR)

DFSDM filter x analog watchdog status register

偏移地址：0x128 + 0x80 * x，（x = 0 到 3）

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
AWHTF[7:0]								AWLTF[7:0]							
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 15:8 **AWHTF[7:0]**：模拟看门狗阈值上限标志 (Analog watchdog high threshold flag)

AWHTF[y] = 1 指示通道 y 上存在阈值上限错误。该位由硬件置 1。该位可由软件使用 DFSDM_FLTxAWCFR 寄存器中的相应 CLRAWHTF[y] 位清零。

位 7:0 **AWLTF[7:0]**：模拟看门狗阈值下限标志 (Analog watchdog low threshold flag)

AWLTF[y] = 1 指示通道 y 上存在阈值下限错误。该位由硬件置 1。该位可由软件使用 DFSDM_FLTxAWCFR 寄存器中的相应 CLRAWLTF[y] 位清零。

注： DFEN = 0 时，DFSDM_FLTxAWSR 的所有位都会自动复位。

27.8.12 DFSDM 滤波器 x 模拟看门狗清零标志寄存器 (DFSDM_FLTxAWCFR)

DFSDM filter x analog watchdog clear flag register

偏移地址：0x12C + 0x80 * x，（x = 0 到 3）

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
CLRAWHTF[7:0]								CLRAWLTF[7:0]							
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 31:16 保留，必须保持复位值。

位 15:8 **CLRAWHTF[7:0]**：清零模拟看门狗阈值上限标志 (Clear the analog watchdog high threshold flag)
CLRAWHTF[y] = 0：写入“0”无影响
CLRAWHTF[y] = 1：向位置 y 写入“1”会将 DFSDM_FLTxAWSR 寄存器中的相应 AWHTF[y] 位清零。

位 7:0 **CLRAWLTF[7:0]**：将模拟看门狗阈值下限标志清零 (Clear the analog watchdog low threshold flag)
CLRAWLTF[y] = 0：写入“0”无影响
CLRAWLTF[y] = 1：向位置 y 写入“1”会将 DFSDM_FLTxAWSR 寄存器中的相应 AWLTF[y] 位清零。

27.8.13 DFSDM 滤波器 x 极值检测器最大值寄存器 (DFSDM_FLTxEXMAX)

DFSDM filter x extremes detector maximum register

偏移地址：0x130 + 0x80 * x，（x = 0 到 3）

复位值：0x8000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
EXMAX[23:8]															
rs_r	rc_r	rc_r	rc_r	rc_r	rc_r	rc_r	rc_r	rc_r	rc_r	rc_r	rc_r	rc_r	rc_r	rc_r	rc_r

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

位 31:8 **EXMAX[23:0]**：极值检测器最大值 (Extremes detector maximum value)
这些位由硬件置 1，用于指示 DFSDM_FLTx 所转换的最大值。通过读取该寄存器将 EXMAX[23:0] 位复位为值 (0x800000)。

位 7:3 保留，必须保持复位值。

位 2:0 **EXMAXCH[2:0]**：极值检测器最大值数据通道 (Extremes detector maximum data channel)
这些位包含的信息涉及哪一个通道的数据被存储到 EXMAX[23:0]。可通过读取该寄存器将这些位清零。

27.8.14 DFSDM 滤波器 x 极值检测器最小值寄存器(DFSDM_FLTxEXMIN)

DFSDM filter x extremes detector minimum register

偏移地址：0x134 + 0x80 * x，（x = 0 到 3）

复位值：0x7FFF FF00

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

位 31:8 **EXMIN[23:0]**：极值检测器最小值 (Extremes detector minimum value)
这些位由硬件置 1，用于指示 DFSDM_FLTx 所转换的最小值。通过读取该寄存器将 EXMIN[23:0] 位复位为值 (0x7FFFFF)。

位 7:3 保留，必须保持复位值。

位 2:0 **EXMINCH[2:0]**：极值检测器最小值数据通道 (Extremes detector minimum data channel)
这些位包含的信息涉及哪一个通道的数据被存储到 EXMIN[23:0]。可通过读取该寄存器将这些位清零。

27.8.15 DFSDM 滤波器 x 转换定时器寄存器 (DFSDM_FLTxCNVTIMR)

DFSDM filter x conversion timer register

偏移地址：0x138 + 0x80 * x，（x = 0 到 3）

复位值：0x0000 0000

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

位 31:4 **CNVCNT[27:0]**：28 位定时器计数转换时间 (28-bit timer counting conversion time) t = CNVCNT[27:0] / $f_{DFSDMCLK}$

定时器的输入时钟来自 DFSDM 时钟（系统时钟 $f_{DFSDMCLK}$）。转换时间测量始于每次转换开始，并止于每次转换结束（即第一次和最后一次串行采样之间的间隔）。只有在滤波器旁路 (FOSR[9:0] = 0) 的情况下，转换时间测量才会停止，且 CNVCNT[27:0] = 0。时间计时如下：

如果 FAST=0（或者 FAST=1 连续模式下的第一次转换），则：

$t = [F_{OSR} * (I_{OSR}-1 + F_{ORD}) + F_{ORD}] / f_{CKIN}$.....（适用于 $Sinc^x$ 滤波器）

$t = [F_{OSR} * (I_{OSR}-1 + 4) + 2] / f_{CKIN}$.....（适用于 FastSinc 滤波器）

如果在连续模式下 FAST=1（第一次转换除外），则：

$t = [F_{OSR} * I_{OSR}] / f_{CKIN}$

如果 F_{OSR} = FOSR[9:0] + 1 = 1（滤波器旁路，仅积分器有效），则：

CNVCNT = 0（时间计时停止，转换时间：$t = I_{OSR} / f_{CKIN}$）

其中，f_{CKIN} 为（给定通道 CKINy 引脚上的）通道输入时钟频率；在并行数据输入（来自内部 ADC 或来自 CPU/DMA 写操作）的情况下，其表示输入数据速率。

注： 当转换被中断时（例如通过禁止/使能所选通道），定时器也将该中断时间计算在内。

位 3:0 保留，必须保持复位值。

27.8.16 DFSDM寄存器映射

下表对 DFSDM 寄存器进行了汇总。

表 167. DFSDM 寄存器映射和复位值

偏移	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	**DFSDM_CH0CFGR1**	DFSDMEN	CKOUTSRC	Res.	Res.	Res.	Res.	Res.	Res.	CKOUTDIV[7:0]								DATPACK[1:0]		DATMPX[1:0]		Res.	Res.	Res.	CHINSEL	CHEN	CKABEN	SCDEN	Res.	SPICKSEL[1:0]		SITP[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
0x04	**DFSDM_CH0CFGR2**	OFFSET[23:0]																								DTRBS[4:0]						Res.	Res.	Res.
	reset value	0																								0								
0x08	**DFSDM_CH0AWSCDR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	AWFORD[1:0]		Res.	AWFOSR[4:0]					BKSCD[3:0]				Res.	Res.	Res.	Res.	SCDT[7:0]							
	reset value									0	0		0	0	0	0	0	0	0	0	0					0	0	0	0	0	0	0	0	0
0x0C	**DFSDM_CH0WDATR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WDATA[15:0]															
	reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x10	**DFSDM_CH0DATINR**	INDAT1[15:0]																INDAT0[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
0x14	**DFSDM_CH0DLYR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PLSSKP[5:0]					
	reset value																											0	0	0	0	0	0
0x18 - 0x1C	Reserved	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

表 167. DFSDM 寄存器映射和复位值（续）

偏移	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
0x20	**DFSDM_CH1CFGR1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DATPACK[1:0]		DATMPX[1:0]		Res.	Res.	Res.	CHINSEL	CHEN	CKABEN	SCDEN	Res.	SPICKSEL[1:0]		SITP[1:0]	
	reset value																	0	0	0	0				0	0	0	0		0	0	0	0
0x24	**DFSDM_CH1CFGR2**	OFFSET[23:0]																								DTRBS[4:0]						Res.	Res.	Res.
	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			
0x28	**DFSDM_CH1AWSCDR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	AWFORD[1:0]		Res.	AWFOSR[4:0]					BKSCD[3:0]				Res.	Res.	Res.	Res.	SCDT[7:0]							
	reset value									0	0		0	0	0	0	0	0	0	0	0					0	0	0	0	0	0	0	0	0
0x2C	**DFSDM_CH1WDATR**	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	WDATA[15:0]															
	reset value																0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x30	**DFSDM_CH1DATINR**	INDAT1[15:0]																INDAT0[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
0x34	**DFSDM_CH1DLYR**	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	PLSSKP[5:0]					
	reset value																											0	0	0	0	0	0
0x38 - 0x3C	Reserved	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
0x40	**DFSDM_CH2CFGR1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DATPACK[1:0]		DATMPX[1:0]		Res.	Res.	Res.	CHINSEL	CHEN	CKABEN	SCDEN	Res.	SPICKSEL[1:0]		SITP[1:0]	
	reset value																	0	0	0	0				0	0	0	0		0	0	0	0
0x44	**DFSDM_CH2CFGR2**	OFFSET[23:0]																								DTRBS[4:0]						Res.	Res.	Res.
	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			
0x48	**DFSDM_CH2AWSCDR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	AWFORD[1:0]		Res.	AWFOSR[4:0]					BKSCD[3:0]				Res.	Res.	Res.	Res.	SCDT[7:0]							
	reset value									0	0		0	0	0	0	0	0	0	0	0					0	0	0	0	0	0	0	0	0
0x4C	**DFSDM_CH2WDATR**	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	WDATA[15:0]															
	reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x50	**DFSDM_CH2DATINR**	INDAT1[15:0]																INDAT0[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
0x54	**DFSDM_CH2DLYR**	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	PLSSKP[5:0]					
	reset value																											0	0	0	0	0	0
0x58 - 0x5C	Reserved	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

表 167. DFSDM 寄存器映射和复位值（续）

偏移	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
0x60	**DFSDM_ CH3CFGR1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DATPACK[1:0]		DATMPX[1:0]		Res.	Res.	Res.	CHINSEL	CHEN	CKABEN	SCDEN	Res.	SPICKSEL[1:0]		SITP[1:0]	
	reset value																	0	0	0	0				0	0	0	0		0	0	0	0
0x64	**DFSDM_ CH3CFGR2**	OFFSET[23:0]																								DTRBS[4:0]					Res.	Res.	Res.
	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			
0x68	**DFSDM_ CH3AWSCDR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	AWFORD[1:0]		Res.	AWFOSR[4:0]					BKSCD[3:0]				Res.	Res.	Res.	Res.	SCDT[7:0]							
	reset value									0	0		0	0	0	0	0	0	0	0	0					0	0	0	0	0	0	0	0	0
0x6C	**DFSDM_ CH3WDATR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WDATA[15:0]															
	reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x70	**DFSDM_ CH3DATINR**	INDAT1[15:0]																INDAT0[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
0x74	**DFSDM_ CH3DLYR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PLSSKP[5:0]					
	reset value																											0	0	0	0	0	0
0x78 - 0x7C	Reserved	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
0x80	**DFSDM_ CH4CFGR1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DATPACK[1:0]		DATMPX[1:0]		Res.	Res.	Res.	CHINSEL	CHEN	CKABEN	SCDEN	Res.	SPICKSEL[1:0]		SITP[1:0]	
	reset value																	0	0	0	0				0	0	0	0		0	0	0	0
0x84	**DFSDM_ CH4CFGR2**	OFFSET[23:0]																								DTRBS[4:0]					Res.	Res.	Res.
	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			
0x88	**DFSDM_ CH4AWSCDR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	AWFORD[1:0]		Res.	AWFOSR[4:0]					BKSCD[3:0]				Res.	Res.	Res.	Res.	SCDT[7:0]							
	reset value									0	0		0	0	0	0	0	0	0	0	0					0	0	0	0	0	0	0	0	0
0x8C	**DFSDM_ CH4WDATR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WDATA[15:0]															
	reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x90	**DFSDM_ CH4DATINR**	INDAT1[15:0]																INDAT0[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
0x94	**DFSDM_ CH4DLYR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PLSSKP[5:0]					
	reset value																											0	0	0	0	0	0
0x98 - 0x9C	Reserved	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

表 167. DFSDM 寄存器映射和复位值（续）

偏移	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
0xA0	**DFSDM_CH5CFGR1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DATPACK[1:0]		DATMPX[1:0]		Res.	Res.	Res.	CHINSEL	CHEN	CKABEN	SCDEN	Res.	SPICKSEL[1:0]		SITP[1:0]	
	reset value																	0	0	0	0				0	0	0	0		0	0	0	0
0xA4	**DFSDM_CH5CFGR2**	OFFSET[23:0]																								DTRBS[4:0]					Res.	Res.	Res.
	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			
0xA8	**DFSDM_CH5AWSCDR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	AWFORD[1:0]		Res.	AWFOSR[4:0]					BKSCD[3:0]				Res.	Res.	Res.	Res.	SCDT[7:0]							
	reset value									0	0		0	0	0	0	0	0	0	0	0					0	0	0	0	0	0	0	0
0xAC	**DFSDM_CH5WDATR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.			Res.	Res.	Res.	Res.	Res.	Res.	WDATA[15:0]															
	reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0xB0	**DFSDM_CH5DATINR**	INDAT1[15:0]																INDAT0[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
0xB4	**DFSDM_CH5DLYR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PLSSKP[5:0]					
	reset value																											0	0	0	0	0	0
0xB8 - 0xBC	Reserved	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
0xC0	**DFSDM_CH6CFGR1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DATPACK[1:0]		DATMPX[1:0]		Res.	Res.	Res.	CHINSEL	CHEN	CKABEN	SCDEN	Res.	SPICKSEL[1:0]		SITP[1:0]	
	reset value																	0	0	0	0				0	0	0	0		0	0	0	0
0xC4	**DFSDM_CH6CFGR2**	OFFSET[23:0]																								DTRBS[4:0]					Res	Res	Res
	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			
0xC8	**DFSDM_CH6AWSCDR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	AWFORD[1:0]		Res.	AWFOSR[4:0]					BKSCD[3:0]				Res.	Res.	Res.	Res.	SCDT[7:0]							
	reset value									0	0		0	0	0	0	0	0	0	0	0					0	0	0	0	0	0	0	0
0xCC	**DFSDM_CH6WDATR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WDATA[15:0]															
	reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0xD0	**DFSDM_CH6DATINR**	INDAT1[15:0]																INDAT0[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
0xD4	**DFSDM_CH6DLYR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PLSSKP[5:0]					
	reset value																											0	0	0	0	0	0
0xD8 - 0xDC	Reserved	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

表 167. DFSDM 寄存器映射和复位值（续）

偏移	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
0xE0	**DFSDM_ CH7CFGR1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DATPACK[1:0]		DATMPX[1:0]		Res.	Res.	Res.	CHINSEL	CHEN	CKABEN	SCDEN	Res.	SPICKSEL[1:0]		SITP[1:0]	
	reset value																	0	0	0	0				0	0	0	0		0	0	0	0
0xE4	**DFSDM_ CH7CFGR2**	OFFSET[23:0]																								DTRBS[4:0]						Res.	Res.	Res.
	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			
0xE8	**DFSDM_ CH7AWSCDR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	AWFORD[1:0]		Res.	AWFOSR[4:0]					BKSCD[3:0]				Res.	Res.	Res.	Res.	SCDT[7:0]							
	reset value									0	0		0	0	0	0	0	0	0	0	0					0	0	0	0	0	0	0	0	0
0xEC	**DFSDM_ CH7WDATR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WDATA[15:0]															
	reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0xF0	**DFSDM_ CH7DATINR**	INDAT1[15:0]																INDAT0[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
0xF4	**DFSDM_ CH7DLYR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PLSSKP[5:0]					
	reset value																											0	0	0	0	0	0
0xF8 - 0xFC	Reserved	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
0x100	**DFSDM_ FLT0CR1**	Res.	AWFSEL	FAST	Res.	Res.	RCH[2:0]			Res.	Res.	RDMAEN	Res.	RSYNC	RCONT	RSW START	Res.	Res.	JEXTEN[1:0]		JEXTSEL[4:0]				Res.	Res.	JDMAEN	JSCAN	JSYNC	Res.	JSW START	DFEN	
	reset value		0	0			0	0	0			0		0	0	0			0	0	0	0	0	0	0			0	0	0		0	0
0x104	**DFSDM_ FLT0CR2**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	AWDCH[7:0]								EXCH[7:0]								Res.	CKABIE	SCDIE	AWDIE	ROVRIE	JOVRIE	REOCIE	JEOCIE
	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
0x108	**DFSDM_ FLT0ISR**	SCDF[7:0]								CKABF[7:0]								Res.	RCIP	JCIP	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	AWDF	ROVRF	JOVRF	REOCF	JEOCF
	reset value	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1		0	0									0	0	0	0	0
0x10C	**DFSDM_ FLT0ICR**	CLRSCDF[7:0]								CLRCKABF[7:0]								Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CLR ROVRF	CLR JOVRF	Res.	Res.
	reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0													0	0		
0x110	**DFSDM_ FLT0JCHGR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	JCHG[7:0]							
	reset value																									0	0	0	0	0	0	0	1
0x114	**DFSDM_ FLT0FCR**	FORD[2:0]			Res.	Res.	Res.	FOSR[9:0]										Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	IOSR[7:0]							
	reset value	0	0	0				0	0	0	0	0	0	0	0	0	0									0	0	0	0	0	0	0	0

表 167. DFSDM 寄存器映射和复位值（续）

偏移	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
0x118	**DFSDM_ FLT0JDATAR**	JDATA[23:0]																								Res.	Res.	Res.	Res.	Res.	JDATACH [2:0]		
	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
0x11C	**DFSDM_ FLT0RDATAR**	RDATA[23:0]																								Res.	Res.	Res.	RPEND	Res.	RDATA CH[2:0]		
	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
0x120	**DFSDM_ FLT0AWHTR**	AWHT[23:0]																								Res.	Res.	Res.	Res.	BKAWH[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
0x124	**DFSDM_ FLT0AWLTR**	AWLT[23:0]																								Res.	Res.	Res.	Res.	BKAWL[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
0x128	**DFSDM_ FLT0AWSR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	AWHTF[7:0]								AWLTF[7:0]							
	reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x12C	**DFSDM_ FLT0AWCFR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CLRAWHTF[7:0]								CLRAWLTF[7:0]							
	reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x130	**DFSDM_ FLT0EXMAX**	EXMAX[23:0]																								Res.	Res.	Res.	Res.	Res.	EXMAXCH[2:0]		
	reset value	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
0x134	**DFSDM_ FLT0EXMIN**	EXMIN[23:0]																								Res.	Res.	Res.	Res.	Res.	EXMINCH[2:0]		
	reset value	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1						0	0	0
0x138	**DFSDM_ FLT0CNVTIMR**	CNVCNT[27:0]																												Res.	Res.	Res.	Res.
	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				
0x13C - 0x17C	Reserved	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
0x180	**DFSDM_ FLT1CR1**	Res.	AWFSEL	FAST	Res.	Res.	RCH[2:0]			Res.	Res.	RDMAEN	Res.	RSYNC	RCONT	RSW START	Res.	Res.	JEXTEN[1:0]		JEXTSEL[4:0]					Res.	Res.	JDMAEN	JSCAN	JSYNC	Res.	JSW START	DFEN
	reset value		0	0			0	0	0			0		0	0	0			0	0	0	0	0	0	0			0	0	0		0	0
0x184	**DFSDM_ FLT1CR2**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	AWDCH[7:0]								EXCH[7:0]								Res.	Res.	Res.	AWDIE	ROVRIE	JOVRIE	REOCIE	JEOCIE
	reset value									0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				0	0	0	0	0
0x188	**DFSDM_ FLT1ISR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RCIP	JCIP	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	AWDF	ROVRF	JOVRF	REOCF	JEOCF
	reset value																		0	0									0	0	0	0	0

表 167. DFSDM 寄存器映射和复位值（续）

偏移	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
0x18C	**DFSDM_ FLT1ICR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CLR ROVRF	CLR JOVRF	Res.	Res.
	reset value																													0	0		
0x190	**DFSDM_ FLT1JCHGR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	JCHG[7:0]							
	reset value																									0	0	0	0	0	0	0	1
0x194	**DFSDM_ FLT1FCR**	FORD[2:0]			Res.	Res.	Res.	FOSR[9:0]										Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	IOSR[7:0]							
	reset value	0	0	0				0	0	0	0	0	0	0	0	0	0									0	0	0	0	0	0	0	0
0x198	**DFSDM_ FLT1JDATAR**	JDATA[23:0]																								Res.	Res.	Res.	Res.	Res.	JDATACH[2:0]		
	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
0x19C	**DFSDM_ FLT1RDATAR**	RDATA[23:0]																								Res.	Res.	Res.	RPEND	Res.	RDATA CH[2:0]		
	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
0x1A0	**DFSDM_ FLT1AWHTR**	AWHT[23:0]																								Res.	Res.	Res.	Res.	BKAWH[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
0x1A4	**DFSDM_ FLT1AWLTR**	AWLT[23:0]																								Res.	Res.	Res.	Res.	BKAWL[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
0x1A8	**DFSDM_ FLT1AWSR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	AWHTF[7:0]								AWLTF[7:0]							
	reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x1AC	**DFSDM_ FLT1AWCFR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CLRAWHTF[7:0]								CLRAWLTF[7:0]							
	reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x1B0	**DFSDM_ FLT1EXMAX**	EXMAX[23:0]																								Res.	Res.	Res.	Res.	Res.	EXMAXCH[2:0]		
	reset value	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
0x1B4	**DFSDM_ FLT1EXMIN**	EXMIN[23:0]																								Res.	Res.	Res.	Res.	Res.	EXMINCH[2:0]		
	reset value	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1						0	0	0
0x1B8	**DFSDM_ FLT1CNVTIMR**	CNVCNT[27:0]																												Res.	Res.	Res.	Res.
	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				
0x1BC - 0x1FC	Reserved	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

表 167. DFSDM 寄存器映射和复位值（续）

偏移	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
0x200	**DFSDM_FLT2CR1**	Res.	AWFSEL	FAST	Res.	Res.	RCH[2:0]			Res.	Res.	RDMAEN	Res.	RSYNC	RCONT	RSW START	Res.	Res.	JEXTEN[1:0]		JEXTSEL[4:0]					Res.	Res.	JDMAEN	JSCAN	JSYNC	Res.	JSW START	DFEN
	reset value		0	0			0	0	0			0		0	0	0			0	0	0	0	0	0	0			0	0	0		0	0
0x204	**DFSDM_FLT2CR2**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	AWDCH[7:0]								EXCH[7:0]								Res.	Res.	Res.	AWDIE	ROVRIE	JOVRIE	REOCIE	JEOCIE
	reset value									0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				0	0	0	0	0
0x208	**DFSDM_FLT2ISR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RCIP	JCIP	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	AWDF	ROVRF	JOVRF	REOCF	JEOCF
	reset value																		0	0									0	0	0	0	0
0x20C	**DFSDM_FLT2ICR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CLR ROVRF	CLR JOVRF	Res.	Res.
	reset value																													0	0		
0x210	**DFSDM_FLT2JCHGR**	Res	Res.	Res.	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	JCHG[7:0]							
	reset value																									0	0	0	0	0	0	0	1
0x214	**DFSDM_FLT2FCR**	FORD[2:0]			Res.	Res.	Res.	FOSR[9:0]										Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	IOSR[7:0]							
	reset value	0	0	0				0	0	0	0	0	0	0	0	0	0									0	0	0	0	0	0	0	0
0x218	**DFSDM_FLT2JDATAR**	JDATA[23:0]																								Res.	Res.	Res.	Res.	Res.	JDATACH[2:0]		
	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
0x21C	**DFSDM_FLT2RDATAR**	RDATA[23:0]																								Res.	Res.	Res.	RPEND	Res.	RDATACH[2:0]		
	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
0x220	**DFSDM_FLT2AWHTR**	AWHT[23:0]																								Res.	Res.	Res.	Res.	BKAWH[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
0x224	**DFSDM_FLT2AWLTR**	AWLT[23:0]																								Res.	Res.	Res.	Res.	BKAWL[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
0x228	**DFSDM_FLT2AWSR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	AWHTF[7:0]								AWLTF[7:0]							
	reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x22C	**DFSDM_FLT2AWCFR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CLRAWHTF[7:0]								CLRAWLTF[7:0]							
	reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

表 167. DFSDM 寄存器映射和复位值（续）

偏移	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
0x230	**DFSDM_ FLT2EXMAX**	EXMAX[23:0]																								Res.	Res.	Res.	Res.	Res.	EXMAXCH[2:0]		
	reset value	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
0x234	**DFSDM_ FLT2EXMIN**	EXMIN[23:0]																								Res.	Res.	Res.	Res.	Res.	EXMINCH[2:0]		
	reset value	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1						0	0	0
0x238	**DFSDM_ FLT2CNVTIMR**	CNVCNT[27:0]																												Res.	Res.	Res.	Res.
	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				
0x23C - 0x27C	Reserved	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
0x280	**DFSDM_ FLT3CR1**	Res.	AWFSEL	FAST	Res.	Res.	RCH[2:0]			Res.	Res.	RDMAEN	Res.	RSYNC	RCONT	RSW START	Res.	Res.	JEXTEN[1:0]		JEXTSEL[4:0]					Res.	Res.	JDMAEN	JSCAN	JSYNC	Res.	JSW START	DFEN
	reset value		0	0			0	0	0			0		0	0	0			0	0	0	0	0	0	0			0	0	0		0	0
0x284	**DFSDM_ FLT3CR2**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	AWDCH[7:0]								EXCH[7:0]								Res.	Res.	Res.	AWDIE	ROVRIE	JOVRIE	REOCIE	JEOCIE
	reset value									0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				0	0	0	0	0
0x288	**DFSDM_ FLT3ISR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RCIP	JCIP	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	AWDF	ROVRF	JOVRF	REOCF	JEOCF
	reset value																		0	0									0	0	0	0	0
0x28C	**DFSDM_ FLT3ICR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CLR ROVRF	CLR JOVRF	Res.	Res.
	reset value																													0	0		
0x290	**DFSDM_ FLT3JCHGR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	JCHG[7:0]							
	reset value																									0	0	0	0	0	0	0	1
0x294	**DFSDM_ FLT3FCR**	FORD[2:0]			Res.	Res.	Res.	FOSR[9:0]										Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	IOSR[7:0]							
	reset value	0	0	0				0	0	0	0	0	0	0	0	0	0									0	0	0	0	0	0	0	0
0x298	**DFSDM_ FLT3JDATAR**	JDATA[23:0]																								Res.	Res.	Res.	Res.	Res.	JDATACH[2:0]		
	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

表 167. DFSDM 寄存器映射和复位值（续）

偏移	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
0x29C	**DFSDM_ FLT3RDATAR**	RDATA[23:0]																								Res.	Res.	Res.	RPEND	Res.	RDATACH[2:0]		
	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
0x2A0	**DFSDM_ FLT3AWHTR**	AWHT[23:0]																								Res.	Res.	Res.	Res.	BKAWH[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
0x2A4	**DFSDM_ FLT3AWLTR**	AWLT[23:0]																								Res.	Res.	Res.	Res.	BKAWL[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
0x2A8	**DFSDM_ FLT3AWSR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	AWHTF[7:0]								AWLTF[7:0]							
	reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x2AC	**DFSDM_ FLT3AWCFR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CLRAWHTF[7:0]								CLRAWLTF[7:0]							
	reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x2B0	**DFSDM_ FLT3EXMAX**	EXMAX[23:0]																								Res.	Res.	Res.	Res.	Res.	EXMAXCH[2:0]		
	reset value	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
0x2B4	**DFSDM_ FLT3EXMIN**	EXMIN[23:0]																								Res.	Res.	Res.	Res.	Res.	EXMINCH[2:0]		
	reset value	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1						0	0	0
0x2B8	**DFSDM_ FLT3CNVTIMR**	CNVCNT[27:0]																												Res.	Res.	Res.	Res.
	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				
0x2BC - 0x3FC	Reserved	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res

有关寄存器边界地址的信息，请参见*第 2.2.2 节：存储器映射和寄存器边界地址*。

28 LCD-TFT 显示控制器 (LTDC)

28.1 简介

LCD-TFT（液晶显示器——薄膜晶体管）显示器控制器提供并行数字 RGB（红色、绿色、蓝色）以及水平同步、垂直同步、像素时钟和数据使能信号，这些信号直接输出到不同 LCD 和 TFT 面板的接口。

28.2 LTDC 主要特性

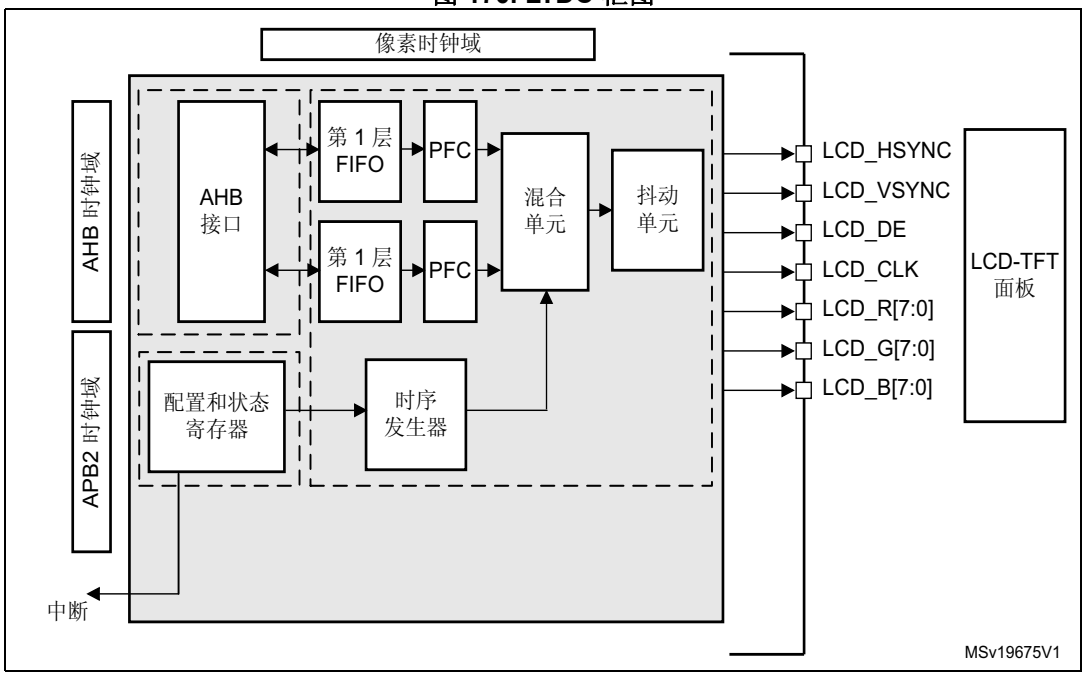
- 24 位 RGB 并行像素输出；每像素 8 位 (RGB888)
- 2 个带有专用 FIFO 的显示层（FIFO 深度 64x32 位）
- 查色表 (CLUT)，每个图层最高 256 种颜色（256x24 位）
- 可针对不同显示面板编程时序
- 可编程背景色
- 可编程 HSYNC、VSYNC 和数据使能信号的极性
- 每层有多达 8 个输入颜色格式可供选择
  - ARGB8888
  - RGB888
  - RGB565
  - ARGB1555
  - ARGB4444
  - L8（8 位 luminance 或 CLUT）
  - AL44（4 位 alpha + 4 位 luminance）
  - AL88（8 位 alpha + 8 位 luminance）
- 每通道的低位采用伪随机抖动输出
  - 红色、绿色、蓝色的抖动宽度为 2 位
- 使用 alpha 值（每像素或常数）在两层之间灵活混合
- 色键（透明颜色）
- 可编程窗口位置和大小
- 支持薄膜晶体管 (TFT) 彩色显示器
- AHB 主接口支持 16 个字的突发
- 最高 4 个可编程中断事件

28.3 LTDC 功能说明

28.3.1 LTDC 框图

LTDC 的框图如*图 170：LTDC 框图*所示。

图 170. LTDC 框图



层 FIFO：每层一个 64x32 位 FIFO。

PFC：执行像素格式转换的像素格式转换器，从层的所选输入像素格式转换为字。

AHB 接口：用于将数据从存储器传输到 FIFO。

有关混合单元、抖动单元和时序发生器的信息：请参见*第 28.4.1 节*和*第 28.4.2 节*。

28.3.2 LTDC 引脚和外部信号接口

表 168 总结了 LTDC 信号接口。

表 168. LTDC 引脚和信号接口

LCD-TFT 信号	I/O	说明
LCD_CLK	O	时钟输出
LCD_HSYNC	O	水平同步
LCD_VSYNC	O	垂直同步
LCD_DE	O	非数据使能
LCD_R[7:0]	O	数据：8 位红色数据
LCD_G[7:0]	O	数据：8 位绿色数据
LCD_B[7:0]	O	数据：8 位蓝色数据

必须通过用户程序配置 LTDC-TFT 控制器引脚。未使用的引脚可用于其他功能。

对于高达 24 位 (RGB888) 的 LTDC 输出，如果使用低于 8 bpp 的像素深度将 RGB565 或 RGB666 输出到 16 位或 18 位显示器，则 RGB 显示数据线必须连接到 LCD-TFT 控制器 RGB 数据线的 MSB。例如，当 LCD-TFT 控制器与 RGB565 16 位显示器相连时，LCD 显示器的 R[4:0]、G[5:0] 和 B[4:0] 数据线引脚必须连接至 LCD-TFT 控制器的 LCD_R[7:3]、LCD_G[7:2] 和 LCD_B[7:3]。

28.3.3 LTDC 复位和时钟

LCD-TFT 控制器外设使用 3 个时钟域：

- AHB 时钟域 (HCLK)

  该时钟域包含 LCD-TFT AHB 主接口，该接口用于将数据从存储器传输到 FIFO 层和帧缓冲区配置寄存器中。
- APB2 时钟域 (PCLK2)

  该时钟域包含全局配置寄存器和中断寄存器。
- 像素时钟域 (LCD_CLK)

  该时钟域包含像素数据生成、层配置寄存器以及 LCD-TFT 接口信号发生器。LCD_CLK 输出应按照面板要求配置。LCD_CLK 由特定的 PLL 输出生成（请参见“复位和时钟控制”部分）。

表 169 汇总了各个寄存器的时钟域。

表 169. 各个寄存器的时钟域

LTDC 寄存器	时钟域
LTDC_LxCR	HCLK
LTDC_LxCFBAR	
LTDC_LxCFBLR	
LTDC_LxCFBLNR	
LTDC_SRCR	PCLK2
LTDC_IER	
LTDC_ISR	
LTDC_ICR	

表 169. 各个寄存器的时钟域（续）

LTDC 寄存器	时钟域
LTDC_SSCR	像素时钟 (LCD_CLK)
LTDC_BPCR	
LTDC_AWCR	
LTDC_TWCR	
LTDC_GCR	
LTDC_BCCR	
LTDC_LIPCR	
LTDC_CPSR	
LTDC_CDSR	
LTDC_LxWHPCR	
LTDC_LxWVPCR	
LTDC_LxCKCR	
LTDC_LxPFCR	
LTDC_LxCACR	
LTDC_LxDCCR	
LTDC_LxBFCR	
LTDC_LxCLUTWR	

访问 LTDC 寄存器时必须谨慎，APB2 总线在以下期间处于停止状态：

- 6 个 PCKL2 周期 + 5 个 LCD_CLK 周期（AHB 时钟域寄存器的 5 个 HCLK 周期），用于寄存器写访问和更新；
- 7 个 PCKL2 周期 + 5 个 LCD_CLK 周期（AHB 时钟域寄存器的 5 个 HCLK 周期），用于寄存器读访问。

对于 PCLK2 时钟域寄存器，APB2 总线会在进行寄存器写访问时停止 6 个 PCKL2 周期，在读访问时停止 7 个 PCKL2 周期。

通过将 RCC_APB2RSTR 寄存器中的相应位置 1 可将 LCD 控制器复位。这将复位三个时钟域。

28.4 LTDC 可编程参数

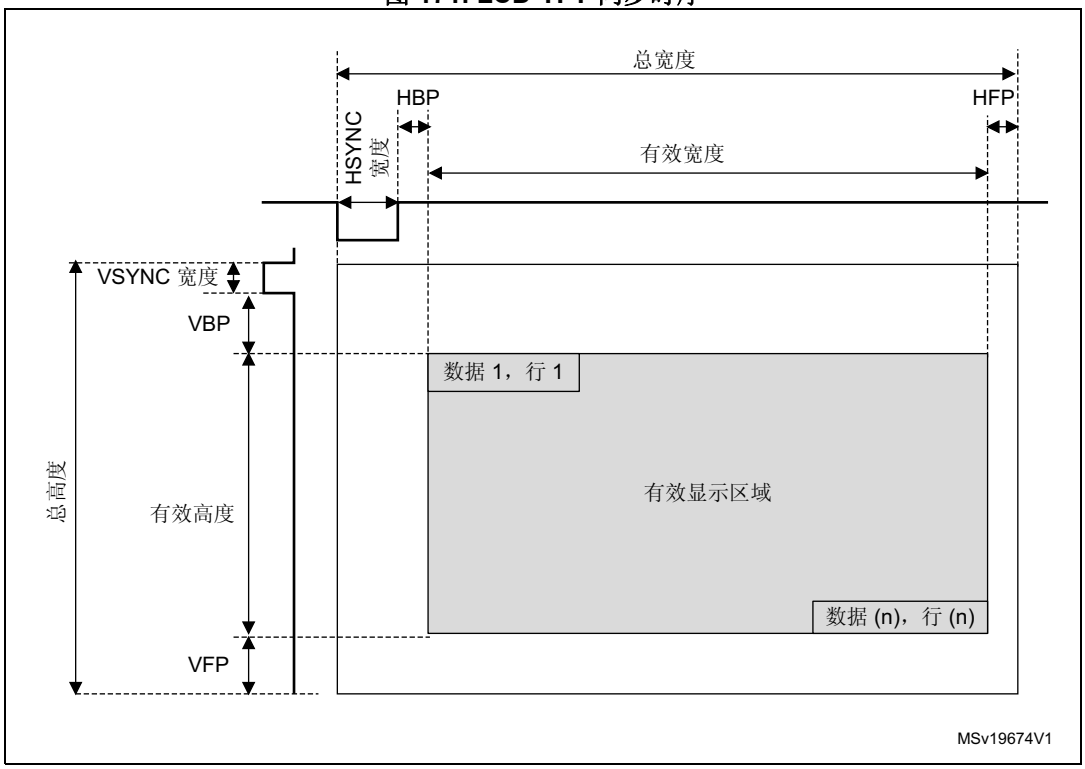
LCD-TFT 控制器提供灵活的可配置参数。其可通过 LTDC_GCR 寄存器使能或禁止。

28.4.1 LTDC 全局配置参数

同步时序

图 171 显示了框图*图 170* 中所示的同步时序发生器模块生成的可配置时序参数。该模块生成水平和垂直同步时序面板信号、像素时钟和数据使能信号。

图 171. LCD-TFT 同步时序



注： *HBP 和 HFP 分别为水平后沿周期和水平前沿周期。*

VBP 和 VFP 分别为垂直后沿周期和垂直前沿周期。

LCD-TFT 可编程同步时序包括：

- HSYNC 和 VSYNC 宽度：水平和垂直同步宽度，通过编程 LTDC_SSCR 寄存器中的 **HSYNC 宽度 - 1** 和 **VSYNC 宽度 - 1** 的值进行配置。
- HBP 和 VBP：水平和垂直同步后沿宽度，通过编程 LTDC_BPCR 寄存器中的累加值 **HSYNC 宽度 + HBP - 1** 和累加值 **VSYNC 宽度 + VBP - 1** 进行配置。
- 有效宽度和有效高度：有效宽度和有效高度通过编程 LTDC_AWCR 寄存器中的累加值 **HSYNC 宽度 + HBP + 有效宽度 - 1** 和累加值 **VSYNC 宽度 + VBP + 有效高度 - 1** 进行配置（仅支持最高 1024x768）。
- 总宽度：总宽度通过编程 LTDC_TWCR 寄存器中的累加值 **HSYNC 宽度 + HBP + 有效宽度 + HFP - 1** 进行配置。HFP 为水平前沿周期。
- 总高度：总高度通过编程 LTDC_TWCR 寄存器中的累加值 **VSYNC 高度 + VBP + 有效高度 + VFP - 1** 进行配置。VFP 为垂直前沿周期。

注： *使能 LTDC 时，产生的时序以 X/Y=0/0 位置作为垂直同步区域中的第一个水平同步像素，随后是后沿、有效数据显示区域和前沿。*

禁止 LTDC 时，时序发生器模块复位为 X = *总宽度 - 1*、Y = *总高度 - 1*，并在垂直同步阶段和 FIFO 清理前保持上一个像素。因此，仅连续输出消隐数据。

同步时序配置示例

TFT-LCD 时序（必须从面板数据手册中提取）：

- 水平和垂直同步宽度：0xA 像素，0x2 行
- 水平和垂直后沿：0x14 像素，0x2 行
- 有效宽度和有效高度：0x140 像素，0xF0 行 (320x240)
- 水平前沿：0xA 像素
- 垂直前沿：0x4 行

LTDC 时序寄存器中编程的值为：

- LTDC_BPCR 寄存器：编程为 0x001D0003
  （AHBP[11:0] 为 0x1D(0xA+ 0x13)，AVBP[10:0]A 为 0x3(0x2 + 0x1)）。
- LTDC_AWCR 寄存器：编程为 0x015D00F3
  （AAW[11:0] 为 0x15D(0xA +0x14 +0x13F)，AAH[10:0] 为 0xF3(0x2 +0x2 + 0xEF)）。
- LTDC_TWCR 寄存器：编程为 0x00000167
  （TOTALW[11:0] 为 0x167(0xA +0x14 +0x140 + 0x9)）。
- LTDC_THCR 寄存器：编程为 0x000000F7
  （TOTALH[10:0] 为 0xF7(0x2 +0x2 + 0xF0 + 3)）。

可编程极性

水平和垂直同步、数据使能和像素时钟输出信号的极性可通过 LTDC_GCR 寄存器编程为高电平有效或低电平有效。

背景色

背景色 (RGB888) 可通过 LTDC_BCCR 寄存器编程。它用于与底层混合。

抖动

使用 LFSR 的伪随机抖动技术用于向各个像素颜色通道值（R、G 或 B）添加小的随机值（阈值），从而当 18 位显示器上显示 24 位数据时，可在某些情况下对 MSB 进行舍入操作。因此，抖动技术用于对帧与帧之间不同的数据进行舍入操作。

伪随机抖动技术的过程是将 LSB 与阈值比较，并在 LSB 部分 >= 阈值时，仅向 MSB 部分加 1。一旦应用抖动技术，通常会减少 LSB。

添加的伪随机值的宽度为每个颜色通道 2 位：红色 2 位、绿色 2 位及蓝色 2 位。

使能 LCD-TFT 控制器后，LFSR 以第一个有效像素开始运行，并且即使在消隐周期内和抖动关闭时也保持运行状态。如果禁止 LTDC，LFSR 将复位。

可通过 LTDC_GCR 寄存器实时开启和关闭抖动。

重载影子寄存器

一些配置寄存器执行影子操作。根据 LTDC_SRCR 寄存器的配置，影子寄存器的值可以立即或者在场消隐周期开始时装载到活动寄存器。如果选择了立即重载配置，则只应在所有新寄存器完成写操作后激活重载。

在重载完成前不应再次修改影子寄存器。读取影子寄存器将返回实际有效值。新写入的值只能在重载发生后读取。

如果在 LTDC_IER 寄存器中相应使能，则可产生寄存器重载中断。

影子寄存器均为第 1 层和第 2 层寄存器，但 LTDC_LxCLUTWR 寄存器除外。

中断产生事件

有关中断配置，请参见*第 28.5 节：LTDC 中断*。

28.4.2 层可编程参数

最多可单独使能、禁止和配置两个层。层显示顺序固定为自下而上。如果使能两个层，则层 2 为顶部显示窗口。

窗口

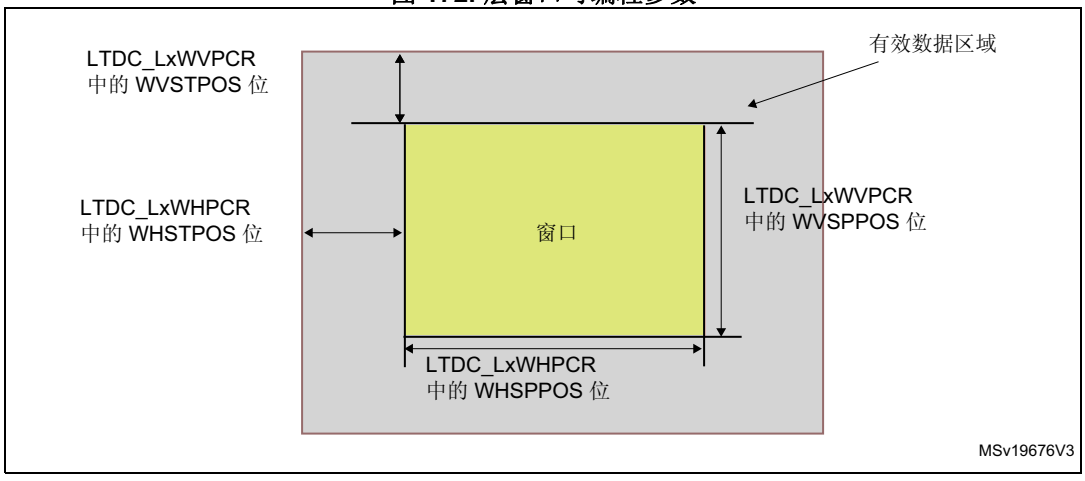
可为每个层定位和调整大小，各个层必须位于有效显示区域内。

窗口位置和大小通过左上和右下的 X/Y 位置以及包含同步、后沿大小和有效数据区域的内部时序发生器配置。请参见 LTDC_LxWHPCR 和 LTDC_WVPCR 寄存器。

可编程层位置和大小定义了一行中的第一个/最后一个可见像素和窗口中的第一个/最后一个可见行。它允许显示完整的图像帧，也允许只显示图像帧的一部分。请参见*图 172*。

- 层中的第一个和最后一个可见像素通过配置 LTDC_LxWHPCR 寄存器中的 WHSTPOS[11:0] 和 WHSPPOS[11:0] 进行设置。
- 层中的第一个和最后一个可见行通过配置 LTDC_LxWVPCR 寄存器中的 WVSTPOS[10:0] 和 WVSPPOS[10:0] 进行设置。

图 172. 层窗口可编程参数



像素输入格式

可编程像素格式用于层的帧缓冲区中存储的数据。

可通过 LTDC_LxPFCR 寄存器为每个层配置多达 8 个输入像素格式。

像素数据从帧缓冲区中读取，随后按照以下方式转换为内部 8888 (ARGB) 格式：宽度低于 8 位的分量通过位重复扩展到 8 位。所选位范围多次拼接，直至其超过 8 位。在得到的向量中，选择高 8 位。例如：5 位 RGB565 红色通道将变为（位位置）：43210432（低 3 位由 5 位中的高 3 位填充）

表 170 说明了像素数据映射与所选格式的关系。

表 170. 像素数据映射与颜色格式的关系

ARGB8888			
@+3 A_x[7:0]	@+2 R_x[7:0]	@+1 G_x[7:0]	@ B_x[7:0]
@+7 A_{x+1}[7:0]	@+6 R_{x+1}[7:0]	@+5 G_{x+1}[7:0]	@+4 B_{x+1}[7:0]
RGB888			
@+3 B_{x+1}[7:0]	@+2 R_x[7:0]	@+1 G_x[7:0]	@ B_x[7:0]
@+7 G_{x+2}[7:0]	@+6 B_{x+2}[7:0]	@+5 R_{x+1}[7:0]	@+4 G_{x+1}[7:0]
RGB565			
@+3 R_{x+1}[4:0] G_{x+1}[5:3]	@+2 G_{x+1}[2:0] B_{x+1}[4:0]	@+1 R_x[4:0] G_x[5:3]	@ G_x[2:0] B_x[4:0]
@+7 R_{x+3}[4:0] G_{x+3}[5:3]	@+6 G_{x+3}[2:0] B_{x+3}[4:0]	@+5 R_{x+2}[4:0] G_{x+2}[5:3]	@+4 G_{x+2}[2:0] B_{x+2}[4:0]
ARGB1555			
@+3 A_{x+1}[0]R_{x+1}[4:0] G_{x+1}[4:3]	@+2 G_{x+1}[2:0] B_{x+1}[4:0]	@+1 A_x[0] R_x[4:0] G_x[4:3]	@ G_x[2:0] B_x[4:0]
@+7 A_{x+3}[0]R_{x+3}[4:0] G_{x+3}[4:3]	@+6 G_{x+3}[2:0] B_{x+3}[4:0]	@+5 A_{x+2}[0]R_{x+2}[4:0]G_{x+2}[4:3]	@+4 G_{x+2}[2:0] B_{x+2}[4:0]
ARGB4444			
@+3 A_{x+1}[3:0]R_{x+1}[3:0]	@+2 G_{x+1}[3:0] B_{x+1}[3:0]	@+1 A_x[3:0] R_x[3:0]	@ G_x[3:0] B_x[3:0]
@+7 A_{x+3}[3:0]R_{x+3}[3:0]	@+6 G_{x+3}[3:0] B_{x+3}[3:0]	@+5 A_{x+2}[3:0]R_{x+2}[3:0]	@+4 G_{x+2}[3:0] B_{x+2}[3:0]

表 170. 像素数据映射与颜色格式的关系（续）

L8			
@+3 L_{x+3}[7:0]	@+2 L_{x+2}[7:0]	@+1 L_{x+1}[7:0]	@ L_x[7:0]
@+7 L_{x+7}[7:0]	@+6 L_{x+6}[7:0]	@+5 L_{x+5}[7:0]	@+4 L_{x+4}[7:0]
AL44			
@+3 A_{x+3}[3:0] L_{x+3}[3:0]	@+2 A_{x+2}[3:0] L_{x+2}[3:0]	@+1 A_{x+1}[3:0] L_{x+1}[3:0]	@ A_x[3:0] L_x[3:0]
@+7 A_{x+7}[3:0] L_{x+7}[3:0]	@+6 A_{x+6}[3:0] L_{x+6}[3:0]	@+5 A_{x+5}[3:0] L_{x+5}[3:0]	@+4 A_{x+4}[3:0] L_{x+4}[3:0]
AL88			
@+3 A_{x+1}[7:0]	@+2 L_{x+1}[7:0]	@+1 A_x[7:0]	@ L_x[7:0]
@+7 A_{x+3}[7:0]	@+6 L_{x+3}[7:0]	@+5 A_{x+2}[7:0]	@+4 L_{x+2}[7:0]

查色表 (CLUT)

可在运行时通过 LTDC_LxCR 寄存器为每个层使能 CLUT，CLUT 仅在使用 L8、AL44 和 AL88 输入像素格式时适用于索引色。

首先，CLUT 必须加载用于替换相应像素（索引色）的原始 R、G 和 B 值的 R、G 和 B 值。每个颜色（RGB 值）在 CLUT 内都有自己对应的地址。

R、G 和 B 值及其各自的地址均通过 LTDC_LxCLUTWR 寄存器编程。

- 在使用 L8 和 AL88 输入像素格式时，CLUT 必须加载 256 个颜色。各颜色的地址在 LTDC_LxCLUTWR 寄存器的 CLUTADD 位中配置。
- 在使用 AL44 输入像素格式时，CLUT 必须仅加载 16 个颜色。各颜色的地址必须通过将 4 位 L 通道重复为 8 位进行填充，具体如下：
  - L0（索引色 0），地址 0x00 处
  - L1，地址 0x11 处
  - L2，地址 0x22 处
  - .....
  - L15，地址 0xFF 处

颜色帧缓冲区地址

每个层的颜色帧缓冲区均有一个起始地址，该地址通过 LTDC_LxCFBAR 寄存器进行配置。

当使能某个层时，将从颜色帧缓冲区中获取该数据。

颜色帧缓冲区长度

每层均设置颜色帧缓冲区的总行长（单位为字节）和行数，二者可分别通过 LTDC_LxCFBLR 和 LTDC_LxCFBLNR 寄存器进行配置。

行长和行数的设置用于阻止超出帧缓存结尾的数据被预取到层对应的 FIFO 中。

- 如果设置为低于所需字节，则会产生 FIFO 下溢中断（如果之前已使能）。
- 如果设置为高于实际所需字节，则将丢弃从 FIFO 中读取的无用数据。无用数据不会显示。

颜色帧缓冲区间距

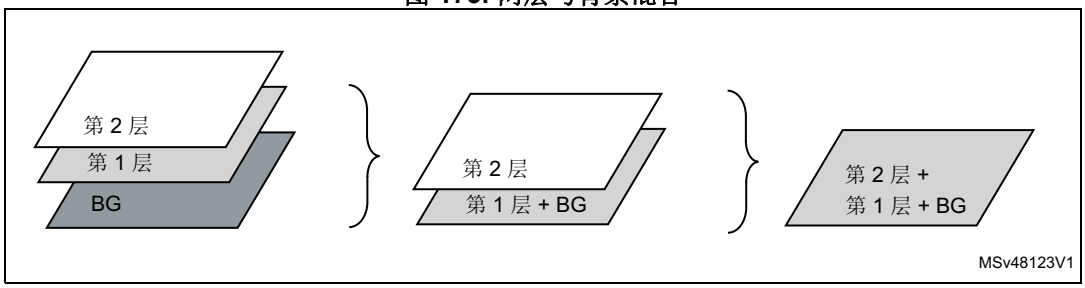
每层的颜色帧缓冲区均具有可配置间距，此间距是一行的开始与下一行开始的距离（以字节为单位）。它通过 LTDC_LxCFBLR 寄存器配置。

层混合

混合操作始终有效，两层可按照 LTDC_LxBFCR 寄存器中配置的混合系数进行混合。

混合顺序固定，即由下至上。如果使能了两层，首先第 1 层将与背景色混合，随后第 2 层与第 1 层和背景的混合颜色结果再次混合。请参见*图 173*。

图 173. 两层与背景混合



默认颜色

每层可具有 ARGB 格式的默认颜色，该颜色在定义的层窗口外使用或在层禁止时使用。

默认颜色通过 LTDC_LxDCCR 寄存器配置。

始终在两层间执行混合操作，即便其中一层禁止也是如此。要避免层禁止时显示默认颜色，需将 LTDC_LxBFCR 寄存器中此层的混合系数设置为其复位值。

色键

色键 (RGB) 可配置为代表透明像素。

使能色键后，当前像素（格式转换后、CLUT 分别混合前的像素）将与色键进行比较。如果当前像素与编程的 RGB 值相匹配，则该像素的所有通道 (ARGB) 均设置为 0。

运行时，可配置色键值并用其替换像素 RGB 值。

色键通过 LTDC_LxCKCR 寄存器使能。

色键通过 LTDC_LxCKCR 寄存器配置。编程值取决于像素格式，因为它会在像素格式转换为 ARGB888 后与当前像素进行比较。

示例：如果将中黄色（50% 红 + 50% 绿）用作透明色键，则：

- 在 RGB565 中，中黄色为 0x8400。将 LTDC_LxCKCR 设为 0x848200。
- 在 ARGB8888 中，中黄色为 0x808000，将 LTDC_LxCKCR 设为 0x808000。
- 在所有基于 CLUT 的颜色模式（L8、AL88、AL44）下，将其中一个调色板条目设为中黄色 0x808000，将 LTDC_LxCKCR 设为 0x808000。

28.5 LTDC 中断

LTDC 提供四个可屏蔽中断，这些中断经逻辑或运算到两个中断向量。

中断源可通过 LTDC_IER 寄存器单独使能或禁止。将相应的屏蔽位置 1 可使能相应中断。

发生如下事件时会产生两个中断：

- 行中断：达到被编程的行时产生。行中断的位置在 LTDC_LIPCR 寄存器中编程
- 寄存器重载中断：在垂直消隐周期内执行影子寄存器重载时产生
- FIFO 下溢中断：从空的层 FIFO 中请求像素时产生
- 传输错误中断：数据传输期间出现 AHB 总线错误时产生

这些中断事件与 NVIC 控制器相连，如*图 174* 所示。

图 174. 中断事件

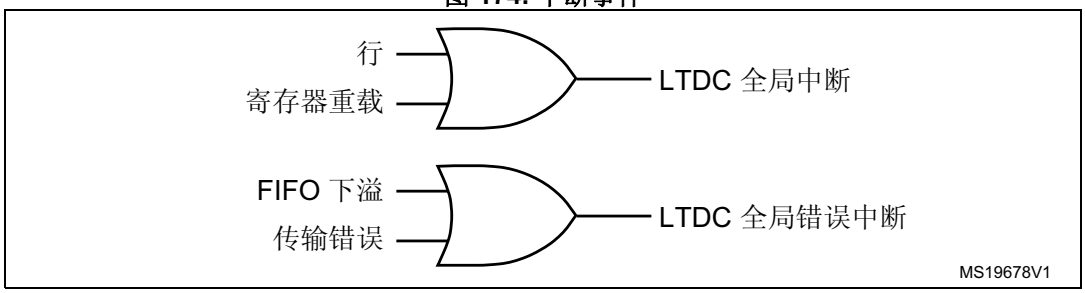


表 171. LTDC 中断请求

中断事件	事件标志	使能控制位
行	LIF	LIE
寄存器重载	RRIF	RRIEN
FIFO 下溢	FUDERRIF	FUDERRIE
传输错误	TERRIF	TERRIE

28.6 LTDC 编程步骤

- 在 RCC 寄存器中使能 LTDC 时钟。
- 按照面板数据表配置所需像素时钟。
- 按照*第 28.4.1 节：LTDC 全局配置参数*中所述的面板数据表配置同步时序：VSYNC、HSYNC、垂直和水平后沿、有效数据区域以及前沿时序。
- 配置 LTDC_GCR 寄存器中的同步信号和时钟极性。
- 必要时，配置 LTDC_BCCR 寄存器中的背景色。
- 配置 LTDC_IER 和 LTDC_LIPCR 寄存器中的所需中断。
- 通过执行以下编程操作配置第 1/2 层的参数：
  - 编程 LTDC_LxWHPCR 和 LTDC_WVPCR 寄存器中的层窗口的水平和垂直位置。层窗口必须位于有效数据区域
  - 编程 LTDC_LxPFCR 寄存器中的像素输入格式
  - 编程 LTDC_LxCFBAR 寄存器中的颜色帧缓冲区起始地址
  - 编程 LTDC_LxCFBLR 寄存器中的颜色帧缓冲区的行长和间距
  - 编程 LTDC_LxCFBLNR 寄存器中的颜色帧缓冲区的行数
  - 必要时，在 LTDC_LxCLUTWR 寄存器中为 CLUT 加载 RGB 值及其地址
  - 必要时，分别在 LTDC_LxDCCR 和 LTDC_LxBFCR 寄存器中配置默认颜色和混合系数
- 使能 LTDC_LxCR 寄存器中的第 1/2 层，必要时使能 CLUT。
- 如果需要，可分别在 LTDC_GCR 和 LTDC_LxCKCR 寄存器中使能抖动和色键。也可以实时使能这两个功能。
- 通过 LTDC_SRCR 寄存器将影子寄存器重载到活动寄存器中。
- 使能 LTDC_GCR 寄存器中的 LCD-TFT 控制器。
- 除 CLUT 外，所有层参数均可实时修改。新配置必须通过配置 LTDC_SRCR 寄存器立即重载或在垂直消隐周期内重载。

注： *所有层的寄存器均执行影子操作。一旦对某个寄存器执行写操作，在重载完成前不应再次进行修改。因此，如果在尚未重载时对同一寄存器执行新的写操作，则将覆盖之前的配置。*

28.7 LTDC 寄存器

28.7.1 LTDC 同步大小配置寄存器 (LTDC_SSCR)

LTDC synchronization size configuration register

此寄存器定义水平同步像素数减 1 以及垂直同步行数减 1。有关配置的示例，请参见*图 171*和*第 28.4 节：LTDC 可编程参数*。

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
Res.	Res.	Res.	Res.	HSW[11:0]											
				rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

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

位 31:28 保留，必须保持复位值。

位 27:16 **HSW[11:0]**：水平同步宽度 (horizontal synchronization width)（以像素时钟周期为单位）
这些位定义水平同步像素数减 1。

位 15:11 保留，必须保持复位值。

位 10:0 **VSH[10:0]**：垂直同步高度 (vertical synchronization height)（以水平扫描行为单位）
这些位定义垂直同步高度减 1。它表示水平同步行数。

28.7.2 LTDC 后沿配置寄存器 (LTDC_BPCR)

LTDC back porch configuration register

此寄存器定义水平同步像素加水平后沿像素的累加数减 1（**HSYNC 宽度 + HBP - 1**）以及垂直同步行加垂直后沿行的累加数减 1（**VSYNC 高度 + VBP - 1**）。有关配置的示例，请参见*图 171*和*第 28.4 节：LTDC 可编程参数*。

偏移地址：0x0C

复位值：0x0000 0000

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

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

位 31:28 保留，必须保持复位值。

位 27:16 **AHBP[11:0]**：累加水平后沿 (accumulated horizontal back porch)（以像素时钟周期为单位）
这些位定义累加水平后沿宽度（水平同步像素加水平后沿像素减 1）。
水平后沿是水平同步信号变为无效到下一扫描行的有效显示开始之间的间隔。

位 15:11 保留，必须保持复位值。

位 10:0 **AVBP[10:0]**：累加垂直后沿 (accumulated Vertical back porch)（以水平扫描行为单位）
这些位定义累加垂直后沿宽度（垂直同步行加垂直后沿行减 1）。
垂直后沿是帧开始到下一帧的首个有效扫描行开始所包含的水平扫描行的数量。

28.7.3 LTDC 有效宽度配置寄存器 (LTDC_AWCR)

LTDC active width configuration register

此寄存器定义水平同步加水平后沿加有效像素的累加数减 1（**HSYNC 宽度 + HBP + 有效宽度 - 1**）以及垂直同步行加垂直后沿行加有效行的累加数减 1（**VSYNC 高度 + BVBP + 有效高度 - 1**）。有关配置的示例，请参见*图 171*和*第 28.4 节：LTDC 可编程参数*。

偏移地址：0x10

复位值：0x0000 0000

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

位 31:28 保留，必须保持复位值。

位 27:16 **AAW[11:0]**：累加有效宽度 (accumulated active width)（以像素时钟周期为单位）
这些位定义累加有效宽度（水平同步像素加水平后沿像素加有效像素减 1）。
有效宽度是面板扫描行的有效显示区中的像素数。
有关最大像素时钟支持的最大有效宽度，请参见器件数据手册。

位 15:11 保留，必须保持复位值。

位 10:0 **AAH[10:0]**：累加有效高度 (accumulated active height)（以水平扫描行为单位）
这些位定义累加高度（垂直同步行加垂直后沿行加有效高度行减 1）。有效高度是面板中的有效行数。
有关最大像素时钟支持的最大有效高度，请参见器件数据手册。

28.7.4 LTDC 总宽度配置寄存器 (LTDC_TWCR)

LTDC total width configuration register

此寄存器定义水平同步加水平后沿加有效像素加水平前沿的像素累加数减 1（**HSYNC 宽度 + HBP + 有效宽度 + HFP - 1**）以及垂直同步行加垂直后沿行加有效行加垂直前沿行的累加数减 1（**VSYNC 高度 + BVBP + 有效高度 + VFP - 1**）。有关配置的示例，请参见*图 171* 和*第 28.4 节：LTDC 可编程参数*。

偏移地址：0x14

复位值：0x0000 0000

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

位 31:28 保留，必须保持复位值。

位 27:16 **TOTALW[11:0]**：总宽度 (total width)（以像素时钟周期为单位）
这些位定义累加总宽度（水平同步像素加水平后沿像素加有效宽度加水平前沿像素减 1）。

位 15:11 保留，必须保持复位值。

位 10:0 **TOTALH[10:0]**：总高度 (total height)（以水平扫描行为单位）
这些位定义累加高度（垂直同步行加垂直后沿行加有效高度加垂直前沿行减 1）。

28.7.5 LTDC 全局控制寄存器 (LTDC_GCR)

LTDC global control register

此寄存器定义 LCD-TFT 控制器的全局配置。

偏移地址：0x18

复位值：0x0000 2220

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
HSPOL	VSPOL	DEPOL	PCPOL	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DEN
rw	rw	rw	rw												rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	DRW[2:0]			Res.	DGW[2:0]			Res.	DBW[2:0]			Res.	Res.	Res.	LTDCEN
	r	r	r		r	r	r		r	r	r				rw

位 31 **HSPOL**：水平同步极性 (horizontal synchronization polarity)
此位由软件置 1 和清零。
0：水平同步极性低电平有效。
1：水平同步极性高电平有效。

位 30 **VSPOL**：垂直同步极性 (vertical synchronization polarity)
此位由软件置 1 和清零。
0：垂直同步低电平有效。
1：垂直同步高电平有效。

位 29 **DEPOL**：非数据使能极性 (not data enable polarity)
此位由软件置 1 和清零。
0：非数据使能极性低电平有效。
1：非数据使能极性高电平有效。

位 28 **PCPOL**：像素时钟极性 (pixel clock polarity)
此位由软件置 1 和清零。
0：像素时钟极性低电平有效。
1：像素时钟高电平有效。

位 27:17 保留，必须保持复位值。

位 16 **DEN**：抖动使能 (dither enable)
此位由软件置 1 和清零。
0：禁止抖动
1：使能抖动

位 15 保留，必须保持复位值。

位 14:12 **DRW[2:0]**：抖动红色宽度 (dither red width)
这些位返回抖动红色位。

位 11 保留，必须保持复位值。

位 10:8 **DGW[2:0]**：抖动绿色宽度 (dither green width)
这些位返回抖动绿色位。

位 7 保留，必须保持复位值。

位 6:4 **DBW[2:0]**：抖动蓝色宽度 (dither blue width)
这些位返回抖动蓝色位。

位 3:1 保留，必须保持复位值。

位 0 **LTDCEN**：LCD-TFT 控制器使能 (LCD-TFT controller enable)
此位由软件置 1 和清零。
0：禁止 LTDC
1：使能 LTDC

28.7.6 LTDC 影子重载配置寄存器 (LTDC_SRCR)

LTDC shadow reload configuration register

此寄存器允许立即或在垂直消隐周期内将影子寄存器的值重载到活动寄存器中。影子寄存器均为第 1 层和第 2 层寄存器，但 LTDC_L1CLUTWR 和 LTDC_L2CLUTWR 除外。

偏移地址：0x24

复位值：0x0000 0000

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

位 31:2 保留，必须保持复位值。

位 1 **VBR**：垂直消隐重载 (vertical blanking reload)

此位由软件置 1，只有重载后才由硬件清零（一旦置 1，便无法通过寄存器写操作清零）。

0：无影响

1：影子寄存器在垂直消隐周期（有效显示区后的第一行开始时）内重载。

位 0 **IMR**：立即重载 (immediate reload)

此位由软件置 1，只有重载后才由硬件清零。

0：无影响

1：影子寄存器立即重载。

注： *影子寄存器回读有效值。在重载完成前，读取寄存器得到的是上次的数值。*

28.7.7 LTDC 背景色配置寄存器 (LTDC_BCCR)

LTDC background color configuration register

此寄存器定义背景色 (RGB888)。

偏移地址：0x2C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BCRED[7:0]							
								rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BCGREEN[7:0]								BCBLUE[7:0]							
rw	rw	rw	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 23:16 **BCRED[7:0]**：背景红色值 (background color red value)
这些位配置背景红色值。

位 15:8 **BCGREEN[7:0]**：背景绿色值 (background color green value)
这些位配置背景绿色值。

位 7:0 **BCBLUE[7:0]**：背景蓝色值 (background color blue value)
这些位配置背景蓝色值。

28.7.8 LTDC 中断使能寄存器 (LTDC_IER)

LTDC interrupt enable register

此寄存器通过将对应的位置 1 来确定哪一个状态标志位产生中断请求。

偏移地址：0x34

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	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.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RRIE	TERRIE	FUIE	LIE
												rw	rw	rw	rw

位 31:4 保留，必须保持复位值。

位 3 **RRIE**：寄存器重载中断使能 (register reload interrupt enable)
此位由软件置 1 和清零。
0：禁止寄存器重载中断
1：使能寄存器重载中断

位 2 **TERRIE**：传输错误中断使能 (transfer error interrupt enable)
此位由软件置 1 和清零。
0：禁止传输错误中断
1：使能传输错误中断

位 1 **FUIE**：FIFO 下溢中断使能 (FIFO underrun interrupt enable)
此位由软件置 1 和清零。
0：禁止 FIFO 下溢中断
1：使能 FIFO 下溢中断

位 0 **LIE**：行中断使能 (line interrupt enable)
此位由软件置 1 和清零。
0：禁止行中断
1：使能行中断

28.7.9 LTDC 中断状态寄存器 (LTDC_ISR)

LTDC interrupt status register

此寄存器返回中断状态标志。

偏移地址：0x38

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	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.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RRIF	TERRIF	FUIF	LIF
												r	r	r	r

位 31:4 保留，必须保持复位值。

位 3 **RRIF**：寄存器重载中断标志 (register reload interrupt flag)

0：未产生寄存器重载中断

1：发生垂直消隐重载时（以及到达有效区域后的第一行时）产生寄存器重载中断

位 2 **TERRIF**：传输错误中断标志 (transfer error interrupt flag)

0：未产生传输错误中断

1：出现总线错误时产生传输错误中断

位 1 **FUIF**：FIFO 下溢中断标志 (FIFO underrun interrupt flag)

0：未产生 FIFO 下溢中断。

1：当其中一个层 FIFO 为空并从 FIFO 读取像素数据时，将产生 FIFO 下溢中断

位 0 **LIF**：行中断标志 (line interrupt flag)

0：未产生行中断

1：到达编程的行时产生行中断

28.7.10 LTDC 中断清零寄存器 (LTDC_ICR)

LTDC Interrupt Clear Register

偏移地址：0x3C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	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.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CRRIF	CTERRIF	CFUIF	CLIF
												w	w	w	w

位 31:4 保留，必须保持复位值。

位 3 **CRRIF**：寄存器重载中断清零标志 (clears register reload interrupt flag)
0：无影响
1：将 LTDC_ISR 寄存器中的 RRIF 标志清零

位 2 **CTERRIF**：传输错误中断清零标志 (clears the transfer error interrupt flag)
0：无影响
1：将 LTDC_ISR 寄存器中的 TERRIF 标志清零。

位 1 **CFUIF**：FIFO 下溢中断清零标志 (clears the FIFO underrun interrupt flag)
0：无影响
1：将 LTDC_ISR 寄存器中的 FUDERRIF 标志清零。

位 0 **CLIF**：行中断清零标志 (clears the line interrupt flag)
0：无影响
1：将 LTDC_ISR 寄存器中的 LIF 标志清零。

28.7.11 LTDC 行中断位置配置寄存器 (LTDC_LIPCR)

LTDC line interrupt position configuration register

此寄存器定义行中断的位置。要编程的行值取决于时序参数。请参见*图 171*。

偏移地址：0x40

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	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.	Res.	Res.	Res.	LIPOS[10:0]										
					rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:11 保留，必须保持复位值。

位 10:0 **LIPOS[10:0]**：行中断位置 (line interrupt position)
这些位配置行中断位置。

28.7.12 LTDC 当前位置状态寄存器 (LTDC_CPSR)

LTDC current position status register

偏移地址：0x44

复位值：0x0000 0000

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

位 31:16 **CXPOS[15:0]**：当前 X 位置 (current X position)
这些位返回当前 X 位置。

位 15:0 **CYPOS[15:0]**：当前 Y 位置 (current Y position)
这些位返回当前 Y 位置。

28.7.13 LTDC 当前显示状态寄存器 (LTDC_CDSR)

LTDC current display status register

此寄存器返回由 HSYNC、VSYNC 和水平/垂直 DE 信号控制的当前显示阶段的状态。

示例：如果当前显示阶段为垂直同步阶段，则 VSYNCS 位置 1（高电平有效）。示例：如果当前显示阶段为水平同步阶段，则 HSYNCS 位高电平有效。

偏移地址：0x48

复位值：0x0000 000F

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.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	HSYNCS	VSYNCS	HDES	VDES
												r	r	r	r

位 31:4 保留，必须保持复位值。

位 3 **HSYNCS**：水平同步显示状态 (horizontal synchronization display status)
0：低电平有效
1：高电平有效

位 2 **VSYNCS**：垂直同步显示状态 (vertical synchronization display status)
0：低电平有效
1：高电平有效

位 1 **HDES**：水平数据使能显示状态 (horizontal data enable display status)
0：低电平有效
1：高电平有效

位 0 **VDES**：垂直数据使能显示状态 (vertical data enable display status)
0：低电平有效
1：高电平有效

注： *返回的状态不取决于* ***LTDC_GCR*** *寄存器中配置的极性，而是返回当前的有效显示阶段。*

28.7.14 LTDC 第 x 层控制寄存器 (LTDC_LxCR)

LTDC layer x control register

偏移地址：0x84 + 0x80 * (x - 1)，（x = 1 到 2）

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.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CLUTEN	Res.	Res.	COLKEN	LEN
											rw			rw	rw

位 31:5 保留，必须保持复位值。

位 4 **CLUTEN**：查色表使能 (color look-up table enable)
此位由软件置 1 和清零。
0：禁止查色表
1：使能查色表
CLUT 仅对 L8、AL44 和 AL88 像素格式有意义。请参见*查色表 (CLUT)*。

位 3:2 保留，必须保持复位值。

位 1 **COLKEN**：色键使能 (color keying enable)
此位由软件置 1 和清零。
0：禁止色键
1：使能色键

位 0 **LEN**：层使能 (layer enable)
此位由软件置 1 和清零。
0：禁止层
1：使能层

28.7.15 LTDC 第 x 层窗口水平位置配置寄存器 (LTDC_LxWHPCR)

LTDC layer x window horizontal position configuration register

此寄存器定义第 1 层或第 2 层窗口的水平位置（第一个和最后一个像素）。

一行的第一个可见像素是在 LTDC_BPCR 寄存器中编程的 AHBP[11:0] 位 + 1 的值。

一行的最后一个可见像素是在 LTDC_AWCR 寄存器中编程的 AAW[10:0] 位的值。

偏移地址：0x88 + 0x80 * (x - 1)，（x = 1 到 2）

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.	WHSPPOS[11:0]											
				rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	WHSTPOS[11:0]											
				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 27:16 **WHSPPOS[11:0]**：窗口水平结束位置 (window horizontal stop position)
这些位配置层窗口的一行的最后一个可见像素。
WHSPPOS[11:0] 必须 ≥ ***AHBP[11:0] 位 + 1***（在 LTDC_BPCR 寄存器中编程）。

位 15:12 保留，必须保持复位值。

位 11:0 **WHSTPOS[11:0]**：窗口水平起始位置 (window horizontal start position)
这些位配置层窗口的一行的第一个可见像素。
WHSTPOS[11:0] 必须 ≤ ***AAW[11:0] 位***（在 LTDC_AWCR 寄存器中编程）。

示例：

LTDC_BPCR 寄存器配置为 0x000E0005（AHBP[11:0] 为 0xE），LTDC_AWCR 寄存器配置为 0x028E01E5（AAW[11:0] 为 0x28E）。要配置大小为 630x460 的窗口的水平位置（有效数据区域中的水平起始偏移为 5 个像素）：

1. 层窗口的第一个像素：WHSTPOS[11:0] 必须编程为 0x14 (0xE+1+0x5)。
2. 层窗口的最后一个像素：WHSPPOS[11:0] 必须编程为 0x28A。

28.7.16 LTDC 第 x 层窗口垂直位置配置寄存器 (LTDC_LxWVPCR)

LTDC layer x window vertical position configuration register

此寄存器定义第 1 层或第 2 层窗口的垂直位置（第一行或最后一行）。

一个帧的第一个可见行是在 LTDC_BPCR 寄存器中编程的 AVBP[10:0] 位 + 1 的值。

一个帧的最后一个可见行是在 LTDC_AWCR 寄存器中编程的 AAH[10:0] 位的值。

偏移地址：0x8C + 0x80 * (x - 1)，（x = 1 到 2）

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.	WVSPPOS[10:0]										
					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
Res.	Res.	Res.	Res.	Res.	WVSTPOS[10:0]										
					rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:27 保留，必须保持复位值。

位 26:16 **WVSPPOS[10:0]**：窗口垂直结束位置 (window vertical stop position)
这些位配置层窗口的最后一个可见行。
WVSPPOS[10:0] 必须 ≥ ***AVBP[10:0] 位 + 1***（在 LTDC_BPCR 寄存器中编程）。

位 15:11 保留，必须保持复位值。

位 10:0 **WVSTPOS[10:0]**：窗口垂直起始位置 (window vertical start position)
这些位配置层窗口的第一个可见行。
WVSTPOS[10:0] 必须 ≤ ***AAH[10:0] 位***（在 LTDC_AWCR 寄存器中编程）。

示例：

LTDC_BPCR 寄存器配置为 0x000E0005（AVBP[10:0] 为 0x5），LTDC_AWCR 寄存器配置为 0x028E01E5（AAH[10:0] 为 0x1E5）。

要配置大小为 630x460 的窗口的垂直位置（有效数据区域中的垂直起始偏移为 8 行）：

1. 层窗口的第一行：WVSTPOS[10:0] 必须编程为 0xE (0x5 + 1 + 0x8)。
2. 层窗口的最后一行：WVSTPOS[10:0] 必须编程为 0x1DA。

28.7.17 LTDC 第 x 层色键配置寄存器 (LTDC_LxCKCR)

LTDC layer x color keying configuration register

此寄存器定义色键使用的色键值 (RGB)。

偏移地址：0x90 + 0x80 * (x - 1)，（x = 1 到 2）

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.	CKRED[7:0]							
								rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CKGREEN[7:0]								CKBLUE[7:0]							
rw	rw	rw	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 23:16 **CKRED[7:0]**：色键红色值 (color key red value)

位 15:8 **CKGREEN[7:0]**：色键绿色值 (color key green value)

位 7:0 **CKBLUE[7:0]**：色键蓝色值 (color key blue value)

28.7.18 LTDC 第 x 层像素格式配置寄存器 (LTDC_LxPFCR)

LTDC layer x pixel format configuration register

此寄存器定义层的帧缓冲区中存储的数据所使用的像素格式。像素数据从帧缓冲区读取，随后转换为内部格式 8888 (ARGB)。

偏移地址：0x94 + 0x80 * (x - 1)，（x = 1 到 2）

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.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PF[2:0]		
													rw	rw	rw

位 31:3 保留，必须保持复位值。

位 2:0 **PF[2:0]**：像素格式 (pixel format)

这些位配置像素格式

000：ARGB8888

001：RGB888

010：RGB565

011：ARGB1555

100：ARGB4444

101：L8（8 位 luminance）

110：AL44（4 位 alpha，4 位 luminance）

111：AL88（8 位 alpha，8 位 luminance）

28.7.19 LTDC 第 x 层常数 alpha 配置寄存器 (LTDC_LxCACR)

LTDC layer x constant alpha configuration register

此寄存器定义在 alpha 混合中使用的常数 alpha 值（由硬件实现 255 等分）。请参见 LTDC_LxBFCR 寄存器。

偏移地址：0x98 + 0x80 * (x - 1)，（x = 1 到 2）

复位值：0x0000 00FF

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.	Res.	Res.	Res.	Res.	Res.	Res.	CONSTA[7:0]							
								rw	rw	rw	rw	rw	rw	rw	rw

位 31:8 保留，必须保持复位值。

位 7:0 **CONSTA[7:0]**：常数 alpha (constant alpha)

这些位配置混合时使用的常数 alpha。常数 alpha 由硬件实现 255 等分。

示例：如果编程的常数 alpha 为 0xFF，则常数 alpha 值为 255/255 = 1。

28.7.20 LTDC 第 x 层默认颜色配置寄存器 (LTDC_LxDCCR)

LTDC layer x default color configuration register

此寄存器定义采用 ARGB 格式的层的默认颜色。默认颜色在定义的层窗口外使用或在层禁止时使用。复位值 0x00000000 定义了透明黑色。

偏移地址：0x9C + 0x80 * (x - 1)，（*x* = 1 到 2）

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DCALPHA[7:0]								DCRED[7:0]							
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
DCGREEN[7:0]								DCBLUE[7:0]							
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:24 **DCALPHA[7:0]**：默认 alpha 值 (default color alpha)
这些位配置默认 alpha 值。

位 23:16 **DCRED[7:0]**：默认颜色红色 (default color red)
这些位配置默认红色值。

位 15:8 **DCGREEN[7:0]**：默认颜色绿色 (default color green)
这些位配置默认绿色值。

位 7:0 **DCBLUE[7:0]**：默认颜色蓝色 (default color blue)
这些位配置默认蓝色值。

28.7.21 LTDC 第 x 层混合系数配置寄存器 (LTDC_LxBFCR)

LTDC layer x blending factors configuration register

此寄存器定义混合系数 F1 和 F2。

通用混合公式为：BC = BF1 x C + BF2 x Cs

- BC = 混合后的颜色
- BF1 = 混合系数 1
- C = 当前层颜色
- BF2 = 混合系数 2
- Cs = 底层混合后的颜色

偏移地址：0xA0 + 0x80 * (x - 1)，（x = 1 到 2）

复位值：0x0000 0607

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.	Res.	Res.	Res.	BF1[2:0]			Res.	Res.	Res.	Res.	Res.	BF2[2:0]		
					rw	rw	rw						rw	rw	rw

位 31:11 保留，必须保持复位值。

位 10:8 **BF1[2:0]**：混合系数 1 (blending factor 1)

这些位选择混合系数 F1。

000：保留
001：保留
010：保留
011：保留
100：常数 alpha
101：保留
110：像素 alpha x 常数 alpha
111：保留

位 7:3 保留，必须保持复位值。

位 2:0 **BF2[2:0]**：混合系数 2 (blending factor 2)

这些位选择混合系数 F2

000：保留
001：保留
010：保留
011：保留
100：保留
101：1 - 常数 alpha
110：保留
111：1 -（像素 alpha x 常数 alpha）

注： *常数 alpha 值是在 LxCACR 寄存器中编程的值，由硬件实现 255 等分。*

示例：仅使能第 1 层，BF1 配置为常数 alpha。BF2 配置为 1 - 常数 alpha。在 LxCACR 寄存器中编程的常数 alpha 为 240 (0xF0)。因此，常数 alpha 值为 240/255 = 0.94。C：当前层颜色为 128。
Cs：背景色为 48。第 1 层与背景色混合。
BC = 常数 alpha x C + (1 - 常数 alpha) x Cs = 0.94 x 128 + (1 - 0.94) x 48 = 123。

28.7.22 LTDC 第 x 层颜色帧缓冲区地址寄存器 (LTDC_LxCFBAR)

LTDC layer x color frame buffer address register

此寄存器定义颜色帧缓冲区的起始地址，该地址必须指向帧缓冲区中存储的层的左上角像素的像素数据地址。

偏移地址：0xAC + 0x80 * (x - 1)，（x = 1 到 2）

复位值：0x0000 0000

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

位 31:0 **CFBADD[31:0]**：颜色帧缓冲区起始地址 (color frame buffer start address)
这些位定义颜色帧缓冲区的起始地址。

28.7.23 LTDC 第 x 层颜色帧缓冲区长度寄存器 (LTDC_LxCFBLR)

LTDC layer x color frame buffer length register

此寄存器定义颜色帧缓冲区的行长和行间距。

偏移地址：0xB0 + 0x80 * (x - 1)，（x = 1 到 2）

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	CFBP[12:0]												
			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.	CFBLL[12:0]												
			rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:29 保留，必须保持复位值。

位 28:16 **CFBP[12:0]**：颜色帧缓冲区间距（以字节为单位）(color frame buffer pitch in bytes)
这些位定义从像素某行的起始处到下一行的起始处的增量（以字节为单位）。

位 15:13 保留，必须保持复位值。

位 12:0 **CFBLL[12:0]**：颜色帧缓冲区行长 (color frame buffer line length)
这些位定义一行像素的长度（以字节为单位）+ 3。
行长的计算方法为：
有效宽度 * 每像素的字节数 + 3。

示例：

- 采用 RGB565（每像素 2 个字节）格式且宽度为 256 像素的帧缓冲区（每行总字节数为 256 * 2 = 512）需要向此寄存器写入值 0x02000203（其中，间距 = 行长）。
- 采用 RGB888（每像素 3 个字节）格式且宽度为 320 像素的帧缓冲区（每行总字节数为 320 * 3 = 960）需要向此寄存器写入值 0x03C003C3（其中，间距 = 行长）。

28.7.24 LTDC 第 x 层颜色帧缓冲区行数寄存器 (LTDC_LxCFBLNR)

LTDC layer x color frame buffer line number register

此寄存器定义颜色帧缓冲区中的行数。

偏移地址：0xB4 + 0x80 * (x - 1)，（x = 1 到 2）

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.	Res.	Res.	Res.	CFBLNBR[10:0]										
					rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:11 保留，必须保持复位值。

位 10:0 **CFBLNBR[10:0]**：帧缓冲区行数 (frame buffer line number)
这些位用来定义缓冲区中的行数，行数跟有效高度对应。

注： *行数和行长设置定义针对每层从每个帧中获取的数据量。如果配置为低于所需字节，则会产生 FIFO 下溢中断（如果使能）。*
另一方面，起始地址和间距设置定义存储器中每行的正确起始位置。

28.7.25 LTDC 第 x 层 CLUT 写寄存器 (LTDC_LxCLUTWR)

LTDC layer x CLUT write register

此寄存器定义 CLUT 地址和 RGB 值。

偏移地址：0xC4 + 0x80 * (x - 1)，（x = 1 到 2）

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CLUTADD[7:0]								RED[7:0]							
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
GREEN[7:0]								BLUE[7:0]							
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

位 31:24 **CLUTADD[7:0]**：CLUT 地址 (CLUT address)
这些位配置每个 RGB 值的 CLUT 地址（CLUT 内的颜色位置）。

位 23:16 **RED[7:0]**：红色值 (red value)
这些位配置红色值。

位 15:8 **GREEN[7:0]**：绿色值 (green value)
这些位配置绿色值。

位 7:0 **BLUE[7:0]**：蓝色值 (blue value)
这些位配置蓝色值。

注： *CLUT 写寄存器只应在消隐周期内或在层禁止时配置。CLUT 可通过 LTDC_LxCR 寄存器使能和禁止。*

CLUT 仅对 L8、AL44 和 AL88 像素格式有意义。

28.7.26 LTDC 寄存器映射

下表对 LTDC 寄存器进行了汇总。有关 LTDC 寄存器的基本地址，请参见寄存器边界地址表。

表 172. LTDC 寄存器映射和复位值

偏移	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
0x0008	LTDC_SSCR	Res.	Res.	Res.	Res.	HSW[11:0]												Res.	Res.	Res.	Res.	Res.	VSH[10:0]										
	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
0x000C	LTDC_BPCR	Res.	Res.	Res.	Res.	AHBP[11:0]												Res.	Res.	Res.	Res.	Res.	AVBP[10:0]										
	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
0x0010	LTDC_AWCR	Res.	Res.	Res.	Res.	AAW[11:0]												Res.	Res.	Res.	Res.	Res.	AAH[10:0]										
	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
0x0014	LTDC_TWCR	Res.	Res.	Res.	Res.	TOTALW[11:0]												Res.	Res.	Res.	Res.	Res.	TOTALH[10:0]										
	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
0x0018	LTDC_GCR	HSPOL	VSPOL	DEPOL	PCPOL	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DEN	Res.	DRW[2:0]			Res.	DGW[2:0]			Res.	DBW[2:0]			Res.	Res.	Res.	LTDCEN
	Reset value	0	0	0	0												0		0	1	0		0	1	0		0	1	0				0
0x0024	LTDC_SRCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	VBR	IMR
	Reset value																															0	0
0x002C	LTDC_BCCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BCRED[7:0]								BCGREEN[7:0]								BCBLUE[7:0]							
	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
0x0034	LTDC_IER	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RRIE	TERRIE	FUIE	LIE
	Reset value																													0	0	0	0
0x0038	LTDC_ISR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RRIF	TERRIF	FUIF	LIF
	Reset value																													0	0	0	0
0x003C	LTDC_ICR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CRRIF	CTERRIF	CFUIF	CLIF
	Reset value																													0	0	0	0
0x0040	LTDC_LIPCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LIPOS[10:0]										
	Reset value																						0	0	0	0	0	0	0	0	0	0	0
0x0044	LTDC_CPSR	CXPOS[15:0]																CYPOS[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

表 172. LTDC 寄存器映射和复位值（续）

偏移	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
0x0048	LTDC_CDSR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.		HSYNCS	VSYNCS	HDES	VDES
	Reset value																													1	1	1	1
0x0084	LTDC_L1CR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CLUTEN	Res.	Res.	COLKEN	LEN
	Reset value																												0			0	0
0x0088	LTDC_L1WHPCR	Res.	Res.	Res.	Res.	WHSPPOS[11:0]												Res.	Res.	Res.	Res.	WHSTPOS[11:0]											
	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
0x008C	LTDC_L1WVPCR	Res.	Res.	Res.	Res.	Res.	WVSPPOS[10:0]											Res.	Res.	Res.	Res.	Res.	WVSTPOS[10:0]										
	Reset value						0	0	0	0	0	0	0	0	0	0	0						0	0	0	0	0	0	0	0	0	0	0
0x0090	LTDC_L1CKCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CKRED[7:0]								CKGREEN[7:0]								CKBLUE[7:0]							
	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
0x0094	LTDC_L1PFCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PF[2:0]		
	Reset value																														0	0	0
0x0098	LTDC_L1CACR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CONSTA[7:0]							
	Reset value																									1	1	1	1	1	1	1	1
0x009C	LTDC_L1DCCR	DCALPHA[7:0]								DCRED[7:0]								DCGREEN[7:0]								DCBLUE[7:0]							
	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
0x00A0	LTDC_L1BFCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BF1[2:0]			Res.	Res.	Res.	Res.	Res.	BF2[2:0]		
	Reset value																						1	1	0						1	1	1
0x00AC	LTDC_L1CFBAR	CFBADD[31:0]																															
	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
0x00B0	LTDC_L1CFBLR	Res.	Res.	Res.	CFBP[12:0]													Res.	Res.	Res.	CFBLL[12:0]												
	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
0x00B4	LTDC_L1CFBLNR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CFBLNBR[10:0]										
	Reset value																						0	0	0	0	0	0	0	0	0	0	0
0x00C4	LTDC_L1CLUTWR	CLUTADD[7:0]								RED[7:0]								GREEN[7:0]								BLUE[7:0]							
	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
0x0104	LTDC_L2CR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CLUTEN	Res.	Res.	COLKEN	LEN
	Reset value																												0			0	0

表 172. LTDC 寄存器映射和复位值（续）

偏移	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
0x0108	LTDC_L2WHPCR	Res.	Res.	Res.	Res.	WHSPPOS[11:0]												Res.	Res.	Res.	Res.	WHSTPOS[11:0]											
	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
0x010C	LTDC_L2WVPCR	Res.	Res.	Res.	Res.	Res.	WVSPPOS[10:0]											Res.	Res.	Res.	Res.	Res.	WVSTPOS[10:0]										
	Reset value						0	0	0	0	0	0	0	0	0	0	0						0	0	0	0	0	0	0	0	0	0	0
0x0110	LTDC_L2CKCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CKRED[7:0]								CKGREEN[7:0]								CKBLUE[7:0]							
	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
0x0114	LTDC_L2PFCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PF[2:0]		
	Reset value																														0	0	0
0x0118	LTDC_L2CACR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CONSTA[7:0]							
	Reset value																									1	1	1	1	1	1	1	1
0x011C	LTDC_L2DCCR	DCALPHA[7:0]								DCRED[7:0]								DCGREEN[7:0]								DCBLUE[7:0]							
	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
0x0120	LTDC_L2BFCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BF1[2:0]			Res.	Res.	Res.	Res.	Res.	BF2[2:0]		
	Reset value																						1	1	0						1	1	1
0x012C	LTDC_L2CFBAR	CFBADD[31:0]																															
	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
0x0130	LTDC_L2CFBLR	Res.	Res.	Res.	CFBP[12:0]													Res.	Res.	Res.	CFBLL[12:0]												
	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
0x0134	LTDC_L2CFBLNR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CFBLNBR[10:0]										
	Reset value																						0	0	0	0	0	0	0	0	0	0	0
0x0144	LTDC_L2CLUTWR	CLUTADD[7:0]								RED[7:0]								GREEN[7:0]								BLUE[7:0]							
	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*第 2.2.2 节*。

29 DSI 主机 (DSIHOST)

29.1 简介

显示串行接口 (DSI) 是 MIPI® 联盟定义的通信协议组的一部分。MIPI® DSI 主机控制器是一种数字内核，可实现 MIPI® DSI 规范中定义的所有协议功能。

DSI 提供系统与 MIPI® D-PHY 之间的接口，允许与 DSI 兼容的显示屏进行通信。

29.2 标准和参考文献

- MIPI® 联盟显示串行接口 (DSI) 规范
  v1.1——2011 年 11 月 22 日
- MIPI® 联盟显示总线接口 (DBI-2) 规范
  v2.00——2005 年 11 月 16 日
- MIPI® 联盟显示命令集 (DCS) 规范
  v1.1——2011 年 11 月 22 日
- MIPI® 联盟显示像素接口 (DPI-2) 规范
  v2.00——2005 年 9 月 15 日
- MIPI® 联盟立体显示格式 (SDF) 规范
  v1.0——2011 年 11 月 22 日
- MIPI® 联盟 D-PHY 规范
  v1.1——2011 年 11 月 7 日

29.3 DSI 主机主要特性

- 符合 MIPI® 联盟标准（参见*第 29.2 节：标准和参考文献*）
- MIPI® D-PHY 接口
- 支持 MIPI® 联盟 DCS 规范中定义的所有命令：
  - 通过 APB 接口发送所有命令模式数据包
  - 在视频模式下以低功耗、高速度发送命令
- 最多支持两条 D-PHY 数据通道
- 通过数据通道 0 支持双向通信和逸出模式
- 支持 D-PHY 时钟通道中的非连续时钟，实现进一步节能
- 支持禁止 PLL 的超低功耗模式
- ECC 和校验和功能
- 支持发送结束数据包 (EoTp)
- 故障恢复方案
- 系统接口的可配置选择：
  - 用于控制和可选支持通用和 DCS 命令的 AMBA APB
  - 通过 LTDC 的视频模式接口
  - 通过 LTDC 的适配命令模式接口
  - 视频模式、适配命令模式和 APB 从模式下的独立可编程虚拟通道 Id
- 视频模式接口特性：
  - LTDC 接口颜色编码映射到 24 位接口：
    - 16 位 RGB，配置 1、2 和 3
    - 18 位 RGB，配置 1 和 2
    - 24 位 RGB
  - 所有 LTDC 接口信号的极性可编程
  - 超出 DPI 标准最大分辨率 800x480 像素的扩展分辨率
  - 最大分辨率受可用 DSI 物理链路带宽的限制：
    - 通道数：2
    - 每个通道的最大速度：500 Mbps
    - 请参见*第 29.4.2 节：支持的分辨率和帧速率*中的示例
- 适配接口特性：
  - 支持通过 *memory_write_start* (WMS) 和 *memory_write_continue* (WMC) DCS 命令发送大量数据
  - LTDC 接口颜色编码映射到 24 位接口：
    - 16 位 RGB，配置 1、2 和 3
    - 18 位 RGB，配置 1 和 2
    - 24 位 RGB
- 视频模式测试图发生器：
  - 无 LTDC 刺激的垂直和水平彩条生成
  - 无 LTDC 刺激的 BER 模式

29.4 DSI 主机功能说明

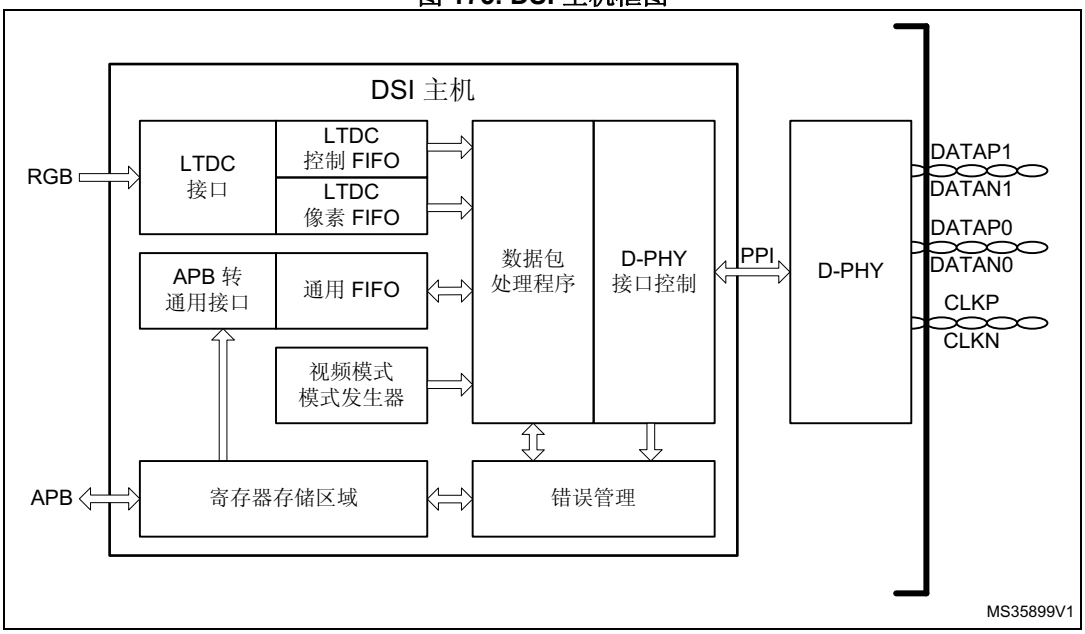
29.4.1 概述

MIPI® DSI 主机包括从内部连接到 LTDC 的专用视频接口和可用于向显示屏发送信息的通用 APB 接口。更多详细信息：

- LTDC 接口：
  - 在视频模式下使用此接口发送信息，其中视频模式是指采用实时像素流 (DPI) 的形式进行从主机处理器到显示模块的传输。
  - 通过自定义模式，此接口可用于在适配命令模式 (DBI) 下以全带宽发送信息。
- APB 从接口：该接口允许在命令模式下发送一般信息，跟随在属性寄存器接口之后。该接口可以在视频模式或适配命令模式下与 LTDC 接口同时运行。
- 视频模式测试图发生器：此接口允许在无任何刺激的情况下发送水平/垂直彩条和 D-PHY BER 测试模式。

DSI 主机的框图如*图 175* 所示。

图 175. DSI 主机框图



29.4.2 支持的分辨率和帧速率

DSI 规范未定义支持的标准分辨率或帧速率。显示分辨率、消隐周期、同步事件持续时间、帧速率和像素颜色深度在所需带宽中起着基本作用。此外，与链路相关的其他属性会影响链路支持 DSI 特定设备的能力。这些属性可以是：显示输入缓冲能力、视频发送模式（突发或非突发）、总线转向 (BTA) 时间、视频模式发送中的并发命令模式流或显示设备详情。所有这些变量让定义标准程序以估算支持特定显示设备的最小通道速率和最小通道数变得极其困难。

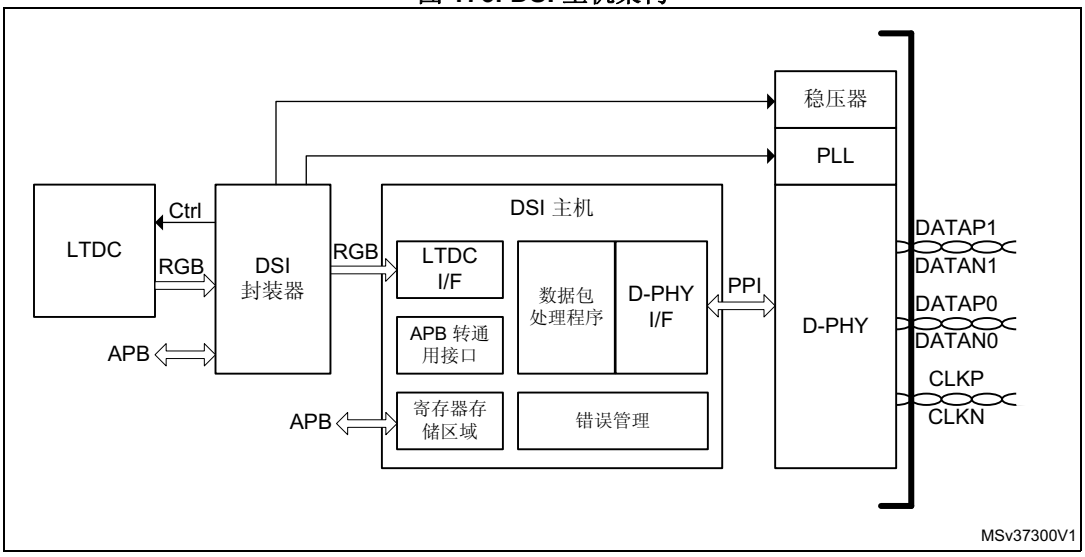
估算的基本假设为：

- 时钟通道频率为 250 MHz，每条数据通道的带宽为 500 Mbps；
- 显示屏应能够以在 DSI 链路中传送像素数据的速度缓冲像素数据；
- 发送像素数据时，链路上没有明显的控制流。

29.4.3 系统级架构

图 176 为 DSI 主机的架构

图 176. DSI 主机架构



不同部分具有以下功能：

- DSI 封装器确保 LTDC 和 DSI 主机内核连接。它可调整颜色模式、信号极性以及管理撕裂效应 (TE)，以在适配命令模式下自动更新帧缓冲区。DSI 封装器还控制 DSI 稳压器、DSI PLL 和 MIPI® D-PHY 的一些特定功能。
- LTDC 接口捕获来自 LTDC 的数据和控制信号，并将它们传送到用于视频控制信号的 FIFO 和用于像素数据的 FIFO。然后，此数据将用于构建以下对象之一：
  - 视频模式下的视频数据包（请参见*第 29.5 节*）
  - 适配命令模式下的 *memory_write_start* 和 *memory_write_continue* DCS 命令（请参见*第 29.6 节*）
- 寄存器库可通过标准 AMBA-APB 从接口访问，提供对 DSI 主机寄存器的访问以便进行配置和控制。此外，还有一个完全可编程的中断发生器，用于通知系统某些事件。
- PHY 接口控制负责管理 D-PHY 接口。此接口确认当前操作，并使能低功耗发送/接收或高速发送，还可在可用 D-PHY 通道之间执行数据分割以实现高速发送。
- 数据包处理程序调度链路内的活动。它基于当前可操作的接口和使用的视频发送模式（突发模式或非突发模式，带同步脉冲或同步事件）执行多种功能。它构建长数据包或短数据包，同时生成相应的 ECC 和 CRC 代码。此块还执行以下功能：
  - 数据包接收
  - 通过检查 ECC 验证数据包报头
  - 单比特错误的报头校正和通知
  - 终止接收
  - 多报头错误通知
  - 根据传入数据包的虚拟通道，处理程序将输出数据路由到相应的端口。
- APB 到通用块将 APB 操作桥接到保存通用命令的 FIFO 中。该块与以下 FIFO 连接：
  - 命令 FIFO
  - 写入负载 FIFO
  - 读取负载 FIFO

- 错误管理模块通知并监视 DSI 链路上的错误情况。其控制定时器来确定是否发生超时、执行内部软复位并触发中断通知。

29.5 功能说明：LTDC 接口上的视频模式

LTDC 接口捕获数据和控制信号，并将这些信号传送到 FIFO 接口，进而发送到 DSI 链路。

接口上存在两种不同数据流：视频控制信号和像素数据。根据接口颜色编码，像素数据在整个 LTDC 总线的配置方式不同。

界面像素颜色编码如*表 173* 所示。

表 173. LTDC 接口中色彩分量的位置

位置	16 位			18 位		24 位
	配置 1	配置 2	配置 3	配置 1	配置 2	
D23	-	-	-	-	-	R[7]
D22	-	-	-	-	-	R[6]
D21	-	-	R[4]	-	R[5]	R[5]
D20	-	R[4]	R[3]	-	R[4]	R[4]
D19	-	R[3]	R[2]	-	R[3]	R[3]
D18	-	R[2]	R[1]	-	R[2]	R[2]
D17	-	R[1]	R[0]	R[5]	R[1]	R[1]
D16	-	R[0]	-	R[4]	R[0]	R[0]
D15	R[4]	-	-	R[3]	-	G[7]
D14	R[3]	-	-	R[2]	-	G[6]
D13	R[2]	G[5]	G[5]	R[1]	G[5]	G[5]
D12	R[1]	G[4]	G[4]	R[0]	G[4]	G[4]
D11	R[0]	G[3]	G[3]	G[5]	G[3]	G[3]
D10	G[5]	G[2]	G[2]	G[4]	G[2]	G[2]
D9	G[4]	G[1]	G[1]	G[3]	G[1]	G[1]
D8	G[3]	G[0]	G[0]	G[2]	G[0]	G[0]
D7	G[2]	-	-	G[1]	-	B[7]
D6	G[1]	-	-	G[0]	-	B[6]
D5	G[0]	-	B[4]	B[5]	B[5]	B[5]
D4	B[4]	B[4]	B[3]	B[4]	B[4]	B[4]
D3	B[3]	B[3]	B[2]	B[3]	B[3]	B[3]
D2	B[2]	B[2]	B[1]	B[2]	B[2]	B[2]

位置	16 位			18 位		24 位
	配置 1	配置 2	配置 3	配置 1	配置 2	
D1	B[1]	B[1]	B[0]	B[1]	B[1]	B[1]
D0	B[0]	B[0]	-	B[0]	B[0]	B[0]

可以配置 LTDC 接口以提高灵活性并促使该接口在多个系统中正确使用。以下配置选项可用：

- 极性控制：所有控制信号均可编程，以根据 LTDC 配置改变极性。
- 内核复位后，DSI 主机等待第一个 VSYNC 有效跳变以开始信号采样（包括像素数据），从而避免在帧中间开始图像数据的发送。
- 如果接口像素颜色编码为 18 位且禁止 18 位松散封装流，则 VPSIZE 字段中编程的像素数必须是 4 的倍数。这意味着在这种模式下，配置中的两个 LSB 始终被推断为零。规范规定，在此模式下，像素行的大小应该是四的倍数。
- 为了避免 FIFO 下溢和上溢，假定始终从 LTDC 接收配置的像素数。
- 为了使存储器相对于数据包调度保持有序，数据包像素数参数用于将内存空间分解到多个视频数据包。

对于 SHTDN 和 COLM 采样和发送，来自 LTDC 的视频流必须有效。这意味着如果 LTDC 未主动生成 VSYNC 和 HSYNC 等视频信号，这些信号就不会通过 DSI 链路发送。由于这种限制，为了正确发送命令，第一个 VSYNC 有效脉冲应当出现以进行命令采样和发送。关闭显示屏时，LTDC 必须在命令发出后在一帧内保持有效。这样可确保在实际禁止 LTDC 接口上的视频生成之前正确发送命令。

SHTDN 和 COLM 值可以在 DSI 封装器控制寄存器 (DSI_WCR) 中编程。

对于所有数据类型，每个时钟周期接收一个完整像素。有效负载的像素数被限制为某个值的倍数，如*表 174* 所示。

表 174. 每个数据类型的有效负载大小（以像素为单位）的多样性

值	数据类型
1	16 位 18 位松散封装 24 位
2	松散封装像素流
4	18 位非松散封装

29.5.1 视频发送模式

有不同的视频发送模式，即：

- 突发模式
- 非突发模式
  - 使用同步脉冲的非突发模式
  - 使用同步事件的非突发模式

突发模式

在该模式下，整个激活像素行在 FIFO 中进行缓冲并以单个数据包进行发送，无中断。此发送模式要求 DPI 像素 FIFO 容量可以存储一整行像素数据。当像素所需带宽和 DSI 链路带宽之间的差异很大时，最好使用此模式，这样可使 DSI 主机能够在单个数据突发中快速调度整个的视频激活行，然后返回到低功耗模式。

非突发模式

在此模式下，处理器使用 DSI 主机的分区属性将视频行发送划分为多个 DSI 数据包。这样做的目的是使所需像素带宽与 DSI 链路带宽相匹配。在此模式下，控制器配置无需在 LTDC 接口像素 FIFO 内部存储整行像素数据，只需存储一个视频数据包的内容。

选择突发模式或非突发模式的指南

突发模式和非突发模式的选择主要取决于系统配置和设备要求。选择适合应用场景的视频发送模式。突发模式更有利，因为它增加了链路在低功率模式下花费更多时间的概率，从而降低了功耗。但是，为了充分利用突发模式的最大收益，应满足以下条件：

- DSI 主机模块应具有足够的像素存储空间来存储整个像素行，以避免内部 FIFO 上溢。
- 显示设备应支持在单次数据包突发中接收完整像素行，以避免接收缓冲区发生上溢。
- DSI 输出带宽应高于 LTDC 接口输入带宽，以使链路每行进入低功耗状态一次。

如果系统无法满足这些要求，像素数据可能丢失，从而导致显示设备在使用突发模式时出现故障。这些错误与系统存储临时像素数据的能力有关。

如果无法满足使用突发模式的所有条件，请使用非突发模式来避免由突发模式导致的错误。非突发模式为像素发送提供了更好的速率匹配，允许：

- 只在存储器中存储一定量的像素，无需完整像素行（对 DSI 主机中 LTDC 接口 RAM 的要求更低）。
- 使用仅支持少量像素缓冲（少于一个完整像素行）的设备进行操作。

DSI 非突发模式的配置应使 DSI 输出像素比与 LTDC 接口输入像素比相匹配，从而降低主机和/或设备端的存储器要求。这是通过将像素行划分为若干像素块并选择性地将它们与空包进行交错来实现的。

以下等式显示了如何在非突发模式下编程 DSI 主机模块发送参数，以匹配 DSI 链路像素输出比（“=”符号的左侧）和 LTDC 接口像素输入（“=”符号的右侧）。

当空包使能时：

lanebyteclkperiod * NUMC (VPSIZE * bytes_per_pixel + 12 + NPSIZE) / number_of_lanes

= pixels_per_line * LTDC_Clock_period

当空包禁止时：

lanebyteclkperiod * NUMC (VPSIZE * bytes_per_pixel + 6) / number_of_lanes

= pixels_per_line * LTDC_Clock_period

29.5.2 在视频模式下更新 LTDC 接口配置

可以在不影响当前帧的情况下实时更新 LTDC 接口配置。这是在影子寄存器的帮助下完成的。此功能由 DSI 主机视频影子控制寄存器 (DSI_VSCR) 控制。

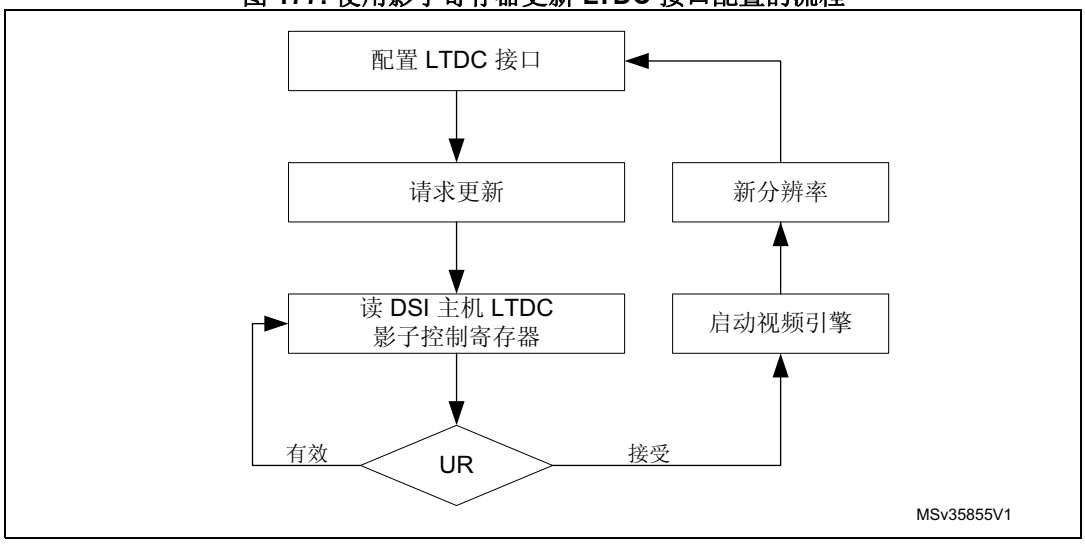
新配置仅在系统请求时使用。要在发送视频帧期间更新视频配置，需要将该帧的配置存储在辅助寄存器中。这样，可以通过 APB 接口设置新的帧配置，而不会损坏当前帧。

默认情况下，此功能被禁止。要使能此功能，需要将 DSI 主机视频子影控制寄存器 (DSI_VSCR) 的使能 (EN) 位置 1。

使能此功能后，系统将提供存储在辅助寄存器中的配置。

图 177 显示了更新 LTDC 接口配置的必要步骤。

图 177. 使用影子寄存器更新 LTDC 接口配置的流程

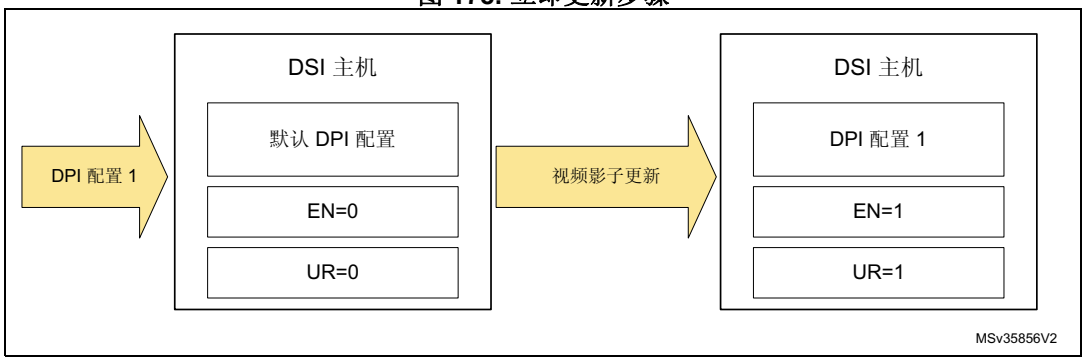


立即更新

当影子寄存器功能激活时，在视频引擎启动之前，辅助寄存器需要 LTDC 配置。这意味着，在复位之后，立即对更新寄存器 (UR) 位授予权限。

在需要立即更新有效寄存器而不进行复位的情况下（如*图 178* 所示），需要确保 DSI 主机视频影子控制寄存器 (DSI_VSCR) 的使能 (EN) 位和更新寄存器 (UR) 位设置为 0。

图 178. 立即更新步骤

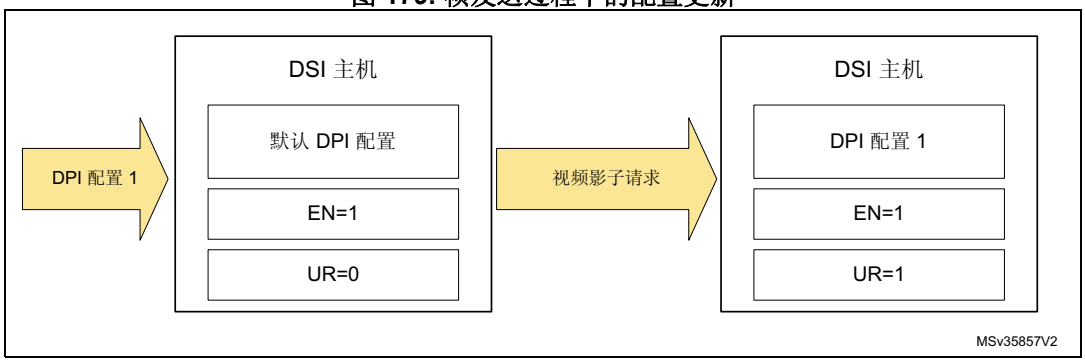


使用 APB 在帧发送期间更新配置

要更新 LTDC 接口配置，请按照*图 179* 所示的步骤操作：

1. 确保 DSI 主机视频影子控制寄存器 (DSI_VSCR) 的使能 (EN) 位置 1。
2. 将 DSI 主机视频影子控制寄存器 (DSI_VSCR) 的更新寄存器 (UR) 位置 1。
3. 监视更新寄存器 (UR) 位。更新结束后此位会置 0。

图 179. 帧发送过程中的配置更新



请求配置更新

可以在帧的任何部分请求 LTDC 接口配置更新。DSI 主机等待直到帧结束才能更改配置。但是，要避免在帧的第一行发送更新请求，因为数据必须在时钟域之间传播。

29.6 功能说明：LTDC 接口上的适配命令模式

适配命令模式使系统能够输入由 DSI 主机使用命令模式发送（使用 DCS 数据包）从 LTDC 传送的像素流。适配命令模式还支持像素输入控制率信号和撕裂效应报告机制。

适配命令模式允许通过 *memory_write_start* (WMS) 和 *memory_write_continue* (WMC) DCS 命令发送大量数据。该模式有助于为在命令模式下发送到 MIPI® 显示屏的存储器写操作提供更宽的数据带宽，并刷新高分辨率显示屏中的大面积像素。如果要传输额外的命令，如显示配置命令、回读命令和撕裂效应初始化，则应使用 APB 从通用接口补偿适配命令模式的功能。

适配命令操作模式支持 16 bpp、18 bpp 和 24 bpp RGB。

在适配命令模式下发送图像数据：

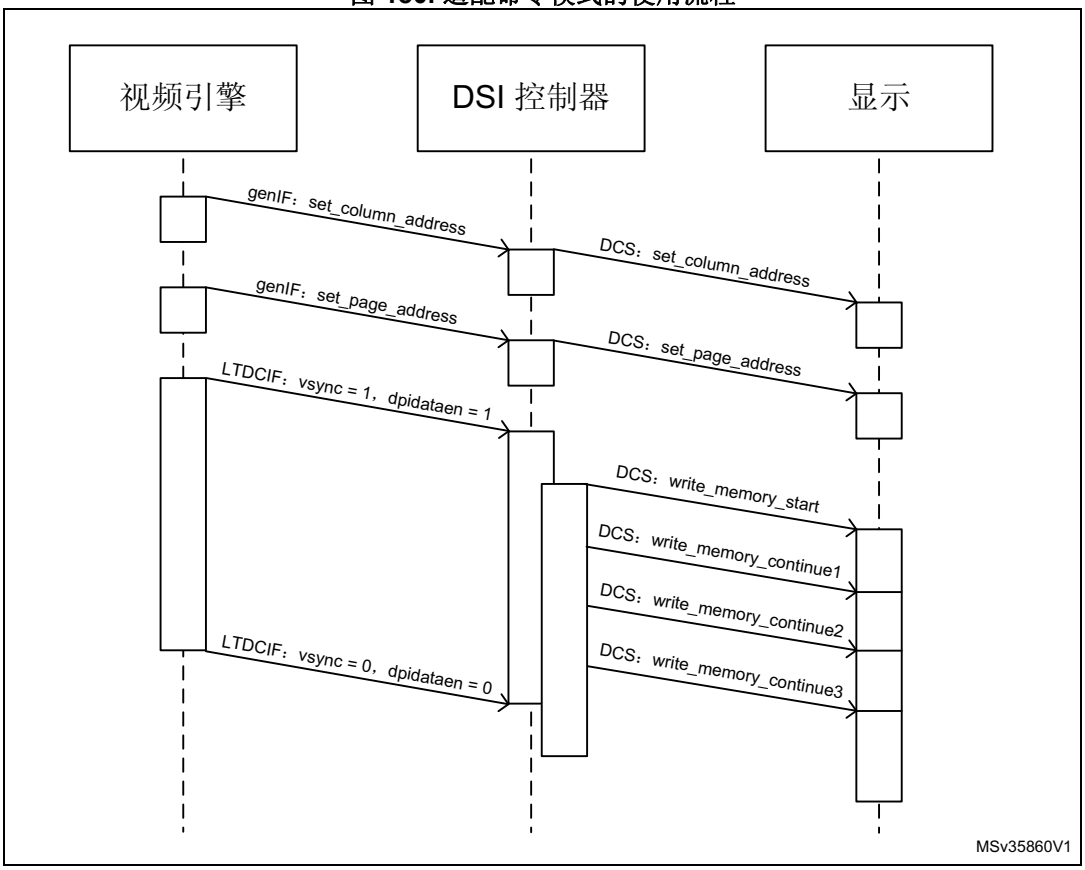
- 将 DSI 主机模式配置寄存器 (DSI_MCR) 的命令模式 (CMDM) 位置 1。
- 将 DSI 封装器配置寄存器 (DSI_WCFGR) 的 DSI 模式 (DSIM) 位置 1。

要发送图像数据，请按下列步骤操作：

- 使用 *set_column_address* 和 *set_page_address* DCS 命令定义要刷新的图像区域。图像区域只需要定义一次，并且在定义不同的值之前保持有效。
- 使用 DSI 主机 LTDC 颜色编码寄存器 (DSI_LCOLCR) 中的颜色编码 (COLC) 字段定义要使用的像素颜色编码。
- 使用 DSI 主机 LTDC VCID 寄存器 (DSI_LVCIDR) 的虚拟通道 ID (VCID) 定义 LTDC 接口生成数据包的虚拟通道 ID。以上值也只需要定义一次。
- 开始从设置 DSI_WCR 寄存器的 LTDC 使能 (LTDCEN) 位的 LTDC 发送数据。

图 180 显示了适配命令模式的使用流程。

图 180. 适配命令模式的使用流程



当 DSI 主机模式配置寄存器 (DSI_CFGR) 的命令模式 (CMDM) 位置 1 时，LTDC 接口采取与适配命令模式相应的行为。

在此模式下，主处理器可以使用 LTDC 接口发送要写入外设帧缓冲器中的连续像素流。它使用像素输入总线接收像素并自动控制流量以限制连续像素流。当接收到第一个像素时，DSI 主机 LTDC 命令配置寄存器 (DSI_LCCR) 的命令大小 (CMDSIZE) 字段的当前值被映射到内部接口功能。接口在通过接口输入的每个有效像素上使计数器递增。当此像素计数器达到命令大小 (CMDSIZE) 时，命令被写入命令 FIFO，并且数据包准备通过 DSI 链路发送。

如果最后一个像素在计数器达到影子命令大小 (CMDSIZE) 值之前到达，则向命令 FIFO 发出 WMS 命令，其中字数 (WC) 设置为对应于计数器值的字节数。如果接收到的像素数量超过 CMDSIZE（阴影值），则将 WMS 命令发送到命令 FIFO，将 WC 设置为与命令大小 (CMDSIZE) 对应的字节数，并重新启动计数器。

将第一个 WMS 命令写入 FIFO 后，电路以类似的方式工作，但发出的是 WMC 命令而不是 WMS 命令。重复该过程，直到接收到图像的最后一个像素。当接收到图像的最后一个像素或达到命令大小 (CMDSIZE) 限制时，模块自动开始发送新的数据包。

与 LTDC 同步

DSI 封装器通过以下方式执行传输过程的同步：

- 控制 LTDC 的启动/停止。
- 在 LTDC 和 DSI 主机之间进行数据流控制。

可以触发刷新显示帧缓冲区的传输

- 手动将 DSI 封装器控制寄存器 (DSI_WCR) 的 LTDC 使能 (LTDCEN) 位置 1。
- 发生撕裂效应 (TEIF) 事件时自动启用自动刷新 (AR)。

通过 DSI 封装器配置寄存器 (DSI_WCFGR) 的自动刷新 (AR) 位完成在手动模式和自动模式之间的选择。在自动刷新模式下，DSI 封装器控制寄存器 (DSI_WCR) 的 LTDC 使能 (LTDCEN) 位由撕裂效应 (TEIF) 事件自动置 1。

在手动或自动刷新模式下完成一帧的传输后，DSI 封装器会停止 TFT 显示控制器 (LTDC)，复位 DSI 封装器控制寄存器 (DSI_WCR) 的 LTDC 使能 (LTDCEN) 位，并设置 DSI 封装器状态寄存器 (DSI_WSR) 的刷新中断结束标志 (ERIF)。如果 DSI 封装器配置寄存器 (DSI_WCFGR) 的刷新结束中断使能 (ERIE) 位置 1，则产生中断。

可以将 DSI 封装器状态寄存器 (DSI_WSR) 的刷新结束中断标志 (ERIF) 标志复位，将 DSI 封装器清零中断标志寄存器 (DSI_WCIFR) 的刷新结束中断标志 (CERIF) 位置 1。

根据 DSI 封装器配置寄存器 (DSI_WCFGR) 的 VSync 极性 (VSPOL) 位，DSI 封装器在 VSync 的上升沿或下降沿同步停止 TFT 显示控制器 (LTDC)。

支持撕裂效应

DSI 规范支持命令模式显示屏中的撕裂效应功能。它使主机处理器能够知道显示屏外设读取其帧缓冲区内容过程中的精确时序信息。

撕裂效果可以通过以下方式管理：

- 单独的引脚（未在 DSI 规范中介绍）
- DSI 撕裂效应功能：应使用通用接口寄存器通过 APB 接口发出 *set_tear_on* DCS 命令

通过 GPIO 管理撕裂效应

当 DSI 封装器配置寄存器 (DSI_WCFGR) 的撕裂效应源 (TESRC) 位置 1 时，通过 GPIO 发出撕裂效应信号。

输入信号的极性可以通过 DSI 封装器配置寄存器 (DSI_WCFGR) 的撕裂效应极性 (TEPOL) 位进行配置。

检测到编程边沿时，DSI 封装器中断和状态寄存器 (DSI_WISR) 的撕裂效应中断标志 (TEIF) 位置 1。

如果 DSI 封装器中断使能寄存器 (DSI_WIER) 的撕裂效应中断使能 (TEIE) 位置 1，则产生中断。

通过 DSI 链路管理撕裂效应

当 DSI 封装器配置寄存器 (DSI_WCFGR) 的 TESRC 位置 1 时，通过 DSI 链路管理撕裂效应：

发送 *set_tear_on* 命令后，DSI 主机执行双总线转向 (BTA)，将链路的所有权授予 DSI 显示屏。显示屏保留总线的所有权，直到发生撕裂事件，该事件由 D-PHY 触发事件报告给 DSI 主机。然后，DSI 主机对触发器进行解码并报告将 DSI 封装器中断和状态寄存器 (DSI_WISR) 的撕裂效应中断标志 (TEIF) 位置 1 的事件。

如果 DSI 封装器中断使能寄存器 (DSI_WIER) 的撕裂效应中断使能 (TEIE) 位置 1，则产生中断。

要使用此功能，必须在使用 WMS 和 WMC DCS 命令更新显示后发出 *set_tear_on* 命令。此过程将暂停 DSI 链路，直到显示屏准备好接收新的帧更新。

为实现灵活性，DSI 主机不会在 WMS/WMC 序列后自动生成撕裂效应请求（双 BTA）。这样可以更新显示屏的几个区域，从而改善 DSI 带宽利用率。撕裂效应请求必须始终由 DSI 主机实现的 *set_tear_on* 命令触发。

配置以下寄存器以激活撕裂效应：

- DSI 主机命令模式配置寄存器 (DSI_CMCR)：TEARE；
- DSI 主机协议配置寄存器 (DSI_PCR)：BTAE。

29.7 功能说明：APB 从通用接口

APB 从接口允许在命令模式下发送一般信息，跟随在属性寄存器接口之后。通过此接口发送的命令不限于 DCS 规范规定的命令，还包括 DSI 规范中介绍的制造商特定通用命令。

如 DSI 规范中所述，DSI 主机支持写入和读取命令模式数据包的发送。这些数据包使用 APB 寄存器访问进行构建。根据操作，DSI 主机通用有效负载数据寄存器 (DSI_GPDR) 具有两种不同的功能。发送命令模式数据包时写入该寄存器会将数据作为有效负载发送。读取此寄存器将返回回读操作的有效负载。DSI 主机通用报头配置寄存器 (DSI_GHCR) 包含命令模式数据包报头类型和报头数据。写入此寄存器会触发数据包的发送，这意味着对于长命令模式数据包，需要在 DSI 主机通用有效负载数据寄存器 (DSI_GPDR) 中预先写入数据包的有效负载。

可通过通用接口发送的有效数据包如下：

- 通用短写数据包 0 参数
- 通用短写数据包 1 参数
- 通用短写数据包 2 参数
- 通用短读数据包 0 参数
- 通用短读数据包 1 参数
- 通用短读数据包 2 参数
- 最大读数据包配置
- 通用长写数据包
- DCS 短写数据包 0 参数
- DCS 短写数据包 1 参数
- DCS 短读数据包 0 参数
- DCS 长写数据包

DSI 主机通用数据包状态寄存器 (DSI_GPSR) 中的一组位报告与 APB 接口支持相关的 FIFO 状态。

始终使用 DSI 发送模式之一（即，视频模式或命令模式）传输通用接口数据包。如果没有选择这两种模式，数据包不会通过链路发送，相关 FIFO 最终会上溢。

29.7.1 使用通用接口进行数据包发送

通过 APB 总线传输数据包需要满足以下条件：

- APB 协议定义了写和读过程各执行两个时钟周期。这意味着通过 APB 接口的最大输入数据速率始终为 APB 时钟速度的一半。
- 数据输入总线的最大宽度为 32 位。这允许在输入 APB 时钟频率和 APB 接口可实现的最大比特率之间定义关系。
- 单独使用 APB 时的 DSI 链路像素比特率为（APB 时钟频率）* 16 Mb。
- 仅使用 APB 接口时，理论 DSI 链路的最大比特率可以表示为：DSI 链路最大比特率 = APB 时钟频率（以 MHz 为单位）* 32/2 Mbps。在该公式中，数字 32 表示 APB 数据总线宽度，由于每个 APB 写入过程需要执行两个时钟周期，因此除以 2。
- 带宽取决于 APB 时钟频率；可用带宽随时钟频率增加而增加。

要驱动 APB 接口以实现 DSI 链路传输的高带宽命令模式流，DSI 主机应仅在命令模式下运行，且 APB 接口应该是当前正在使用的唯一数据源。因此，APB 接口具有 DSI 链路的整个带宽，并且不会与任何其他输入接口源共享。

存储器写入命令需要来自 APB 接口的最大吞吐量，因为这些命令包含由 DSI 链路传送的大量数据。写入数据包信息时，首先使用 DSI 主机通用有效负载数据寄存器 (DSI_GPDR) 将给定数据包的有效负载写入有效负载 FIFO。有效负载数据用于命令参数时，将要发送的第一个字节置于 APB 数据总线的最低有效字节位置。

写入有效负载后，将数据报头写入命令 FIFO，这样便可正确存储在命令 FIFO 中。有关 32 位 APB 数据总线上的数据包报头组织的更多信息，请参见 DCS 规范附录 A。

用于存储器写入命令时，有效负载数据包含像素信息，并且应遵循 DCS 规范附录 A 中的像素到字节转换构成。

图 *181* 到 *185* 展示了如何在 APB 数据写入总线中组织像素数据。

存储器写入命令在 DCS 长数据包中传送，封装在 DSI 数据包中。DSI 指定 DCS 命令应存在于数据包的第一个有效负载字节中。图中也包含这些信息。在图 *181* 到 *185* 中，*写入存储器*命令可以由 DCS 命令*写入存储器开始*和*写入存储器继续*替换。

图 181. 24 bpp APB 像素到字节构成

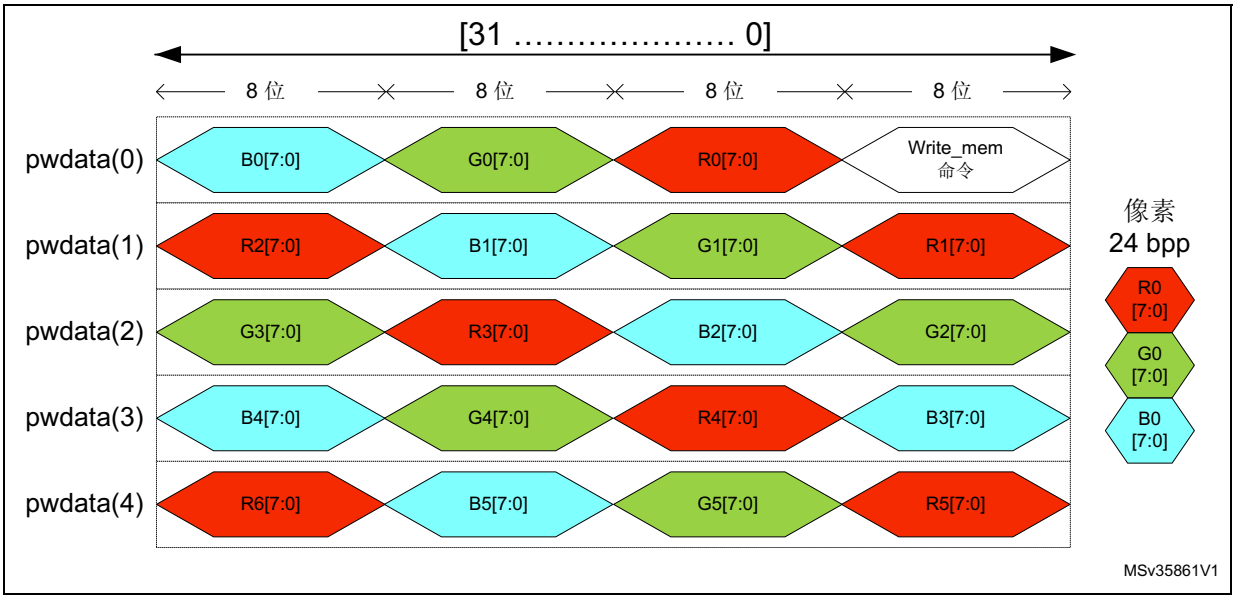


图 182. 18 bpp APB 像素到字节构成

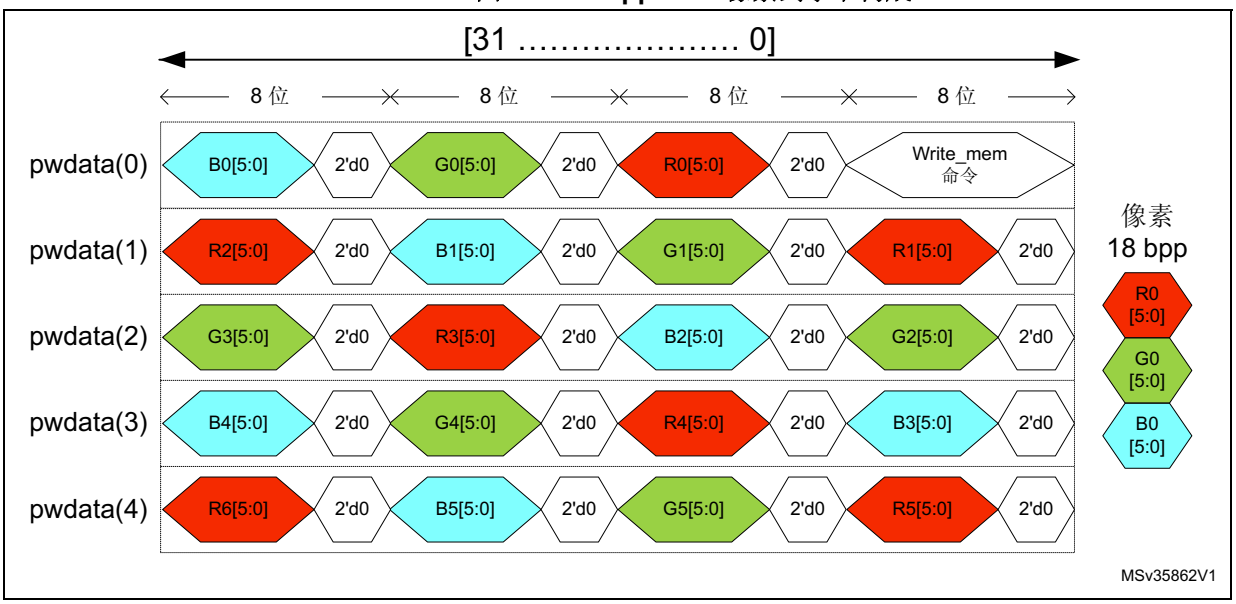


图 183. 16 bpp APB 像素到字节构成

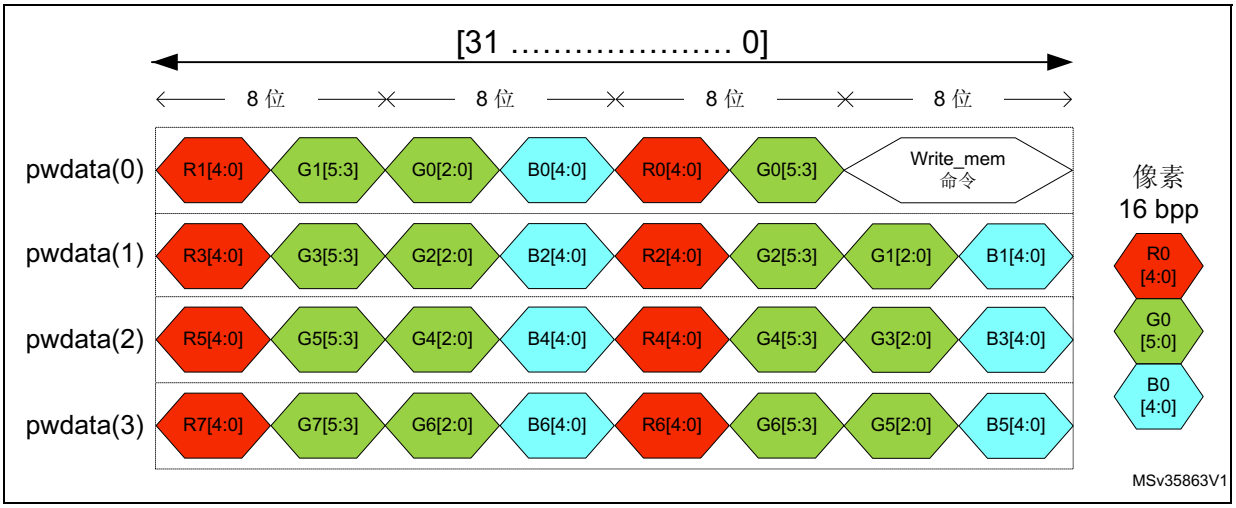


图 184. 12 bpp APB 像素到字节构成

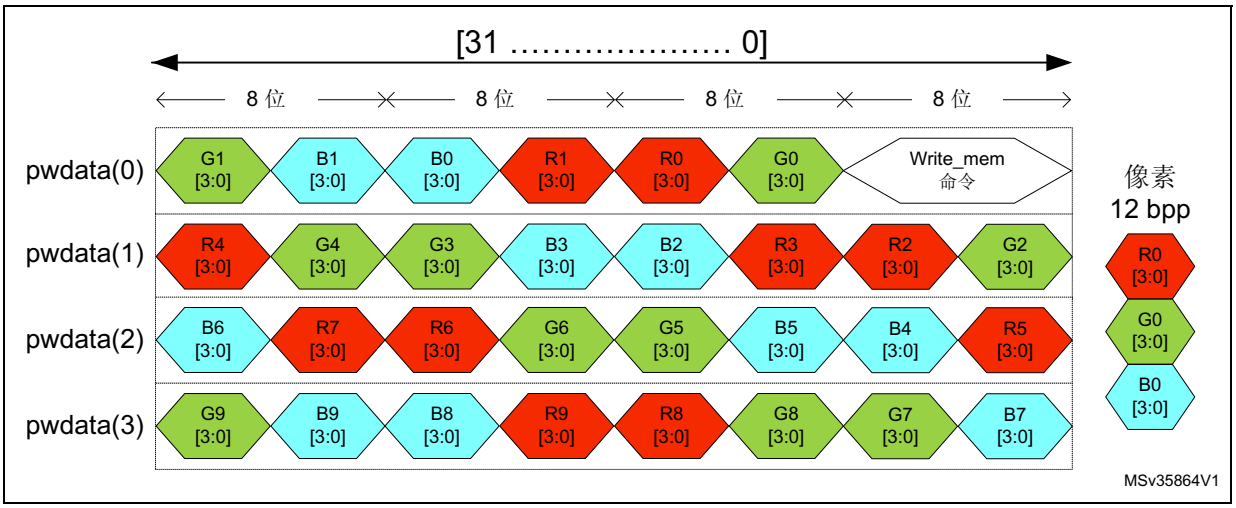
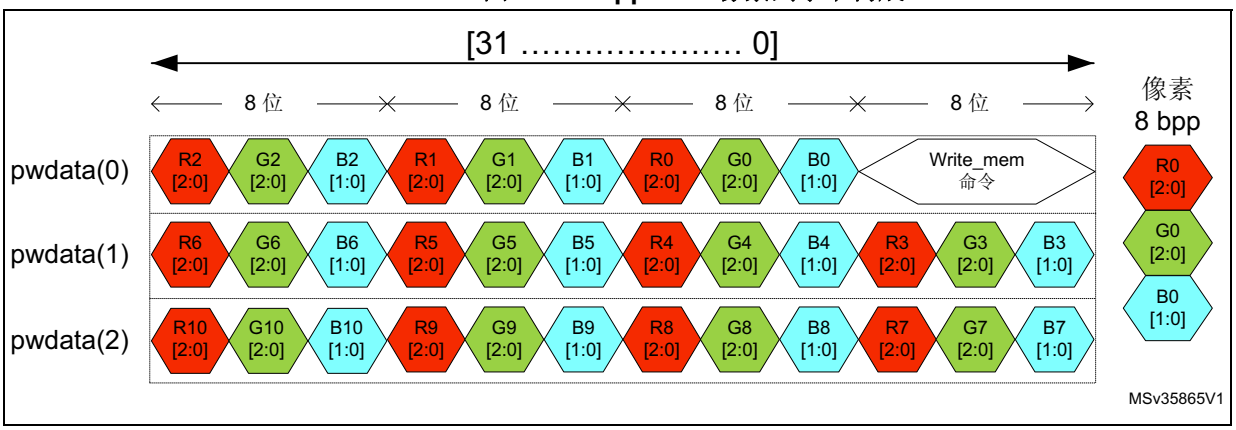


图 185. 8 bpp APB 像素到字节构成



29.8 功能说明：超时计数器

DSI 主机包括用于管理各种通信阶段超时的计数器。每个超时的持续时间可由 6 个 DSI 主机超时计数器配置寄存器 (DSI_TCCR0..5) 进行配置。

有两种类型的计数器：

- 争用错误检测超时计数器（*第 29.8.1 节*）；
- 外设响应超时计数器（*第 29.8.2 节*）。

29.8.1 争用错误检测超时计数器

DSI 主机实现通过一组计数器和条件来通知错误。它具有一组寄存器，可控制用于确定是否发生超时的定时器，还包含一组在读操作时被清零的中断状态寄存器（详细信息请参见*表 175*）。这些寄存器还会触发中断信号，当 DSI 连接发生错误时，系统可以使用该中断信号来激活。

表 175. 争用检测超时计数器配置

超时计数器	值寄存器	值域	标志寄存器	标志域
高速发送	DSI_TCCR0	TOHSTX	DSI_ISR1	TOHSTX
低功耗接收	DSI_TCCR0	TOLPRX	DSI_ISR1	TOLPRX

这些 16 位计数器的时间单位配置为 DSI 主机时钟控制寄存器 (DSI_CCR) 的超时时钟分频 (TOCKDIV) 字段中定义的周期。

写入 DSI 主机时钟控制寄存器 (DSI_CCR) 中的超时时钟分频 (TOCKDIV) 字段的值定义了超时限制（使用通道字节时钟作为输入）的时间单位。

这种机制扩大了定义限制的范围。

高速发送争用检测

超时持续时间在 DSI 主机超时计数器配置寄存器 1 (DSI_TCCR0) 的高速发送超时计数器 (HSTX_TOCNT) 字段中配置。一个 16 位计数器测量高速模式激活的时间。

如果该计数器达到 DSI 主机超时计数器配置寄存器 1 (DSI_TCCR0) 的高速发送超时计数 (HSTX_TOCNT) 字段定义的值，则 DSI 主机中断和状态寄存器 1 (DSI_ISR1) 中的超时高速传输 (TOHSTX) 位置位，并向 DSI 主机生成内部软复位。

如果 DSI 主机中断使能寄存器 1 (DSI_IER1) 的超时高速发送中断使能 (TOHSTXIE) 位置 1，则会产生中断。

低功耗接收争用检测

该超时在 DSI 主机超时计数器配置寄存器 1 (DSI_TCCR1) 的低功耗接收超时计数器 (LPRX_TOCNT) 字段中配置。一个 16 位计数器测量低功耗接收激活的时间。

如果该计数器达到 DSI 主机超时计数器配置寄存器 1 (DSI_TCCR0) 的低功耗接收超时计数器 (LPRX_TOCNT) 字段定义的值，则 DSI 主机中断和状态寄存器 1 (DSI_ISR1) 中的超时低功耗接收 (TOLPRX) 位置位，并向 DSI 主机生成内部软复位。

如果 DSI 主机中断使能寄存器 1 (DSI_IER1) 的超时低功耗接收中断使能 (TOLPRXIE) 位置 1，则会产生中断。软件收到中断通知后，必须通过禁止并置位 DSI 主机 PHY 控制寄存器 (DSI_PCTLR) 的数字使能 (DEN) 位来复位 D-PHY。

29.8.2 外设响应超时计数器

外设可能不会对某些收到的数据包立即作出正确响应。例如，外设接收读取请求，但由于其架构而暂时无法访问 RAM。原因可能是正在刷新面板并需要一段时间响应。在这种情况下，设置超时以确保主机等待足够长的时间，以便设备能够在接收新数据或正确响应新请求之前处理先前的数据。

表 176 列出了属于具有相关外设响应超时的各类事件。

表 176. PRESP_TO 计数器不同类别的事件列表

类别	事件
表示 BTA PRESP_TO 的项目	总线转向
指示 PRESP_TO 的读取请求（为 HS 和 LP 复制）	(0x04) 通用读命令，无短参数 (0x14) 通用读命令，1 个短参数 (0x24) 通用读命令，2 个短参数 (0x06) DCS 读取，无短参数
指示 PRESP_TO 的写入请求（为 HS 和 LP 复制）	(0x03) 通用短写命令，无短参数 (0x13) 通用短写命令，1 个短参数 (0x23) 通用短写命令，2 个短参数 (0x29) 通用长写命令过长 (0x05) DCS 短写命令，无短参数 (0x15) DCS 短写命令，1 个短参数 (0x39) DCS 长写命令/write_LUT，命令数据包过长 (0x37) 设置最大返回数据包大小

DSI 主机确保在发送触发超时的事件时，D-PHY 切换为停止状态，计数器开始运行，直到达到该超时的值。链路保持 LP-11 状态，即使有其他事件准备好发送，也需要在超时后使用。

图 *186* 到 *188* 说明了*表 176* 中列出的三个类别的 PRESP_TO 计数器的计数流程。

图 186. 总线转向后的 PRESP_TO 时序

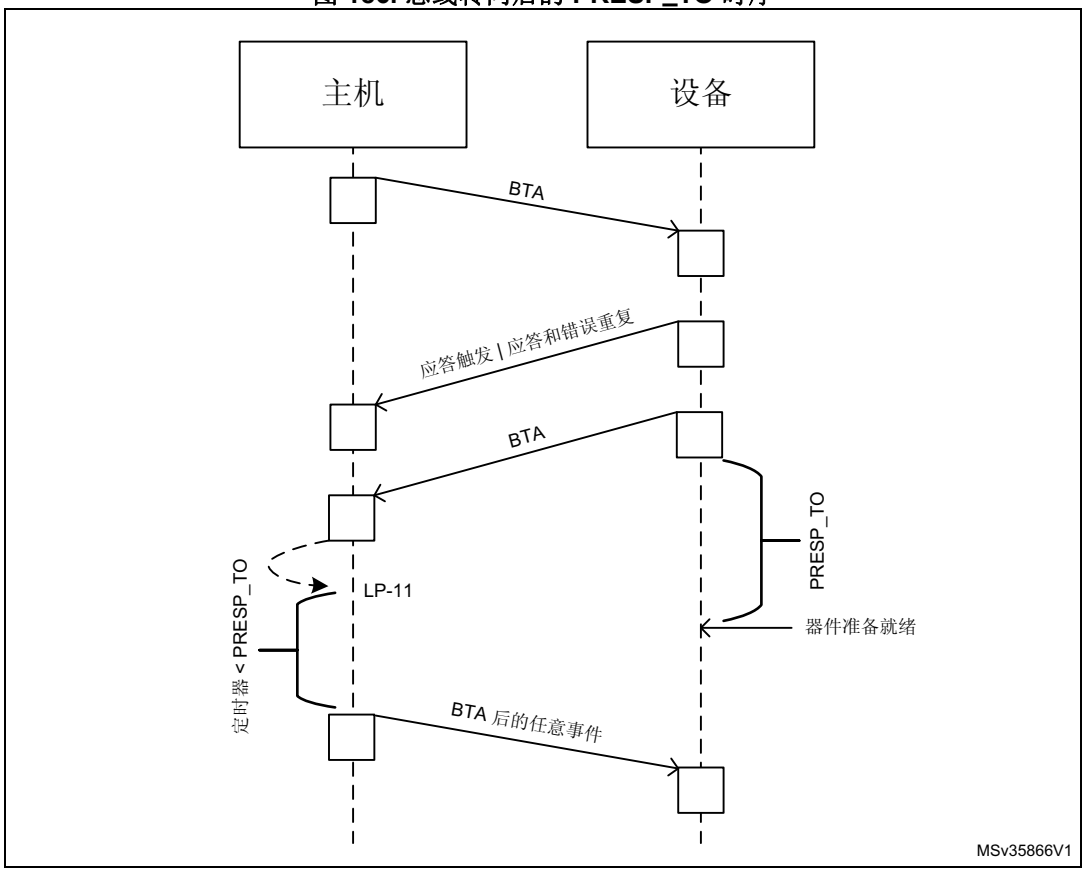


图 187. 读取请求（HS 或 LP）后的 PRESP_TO 时序

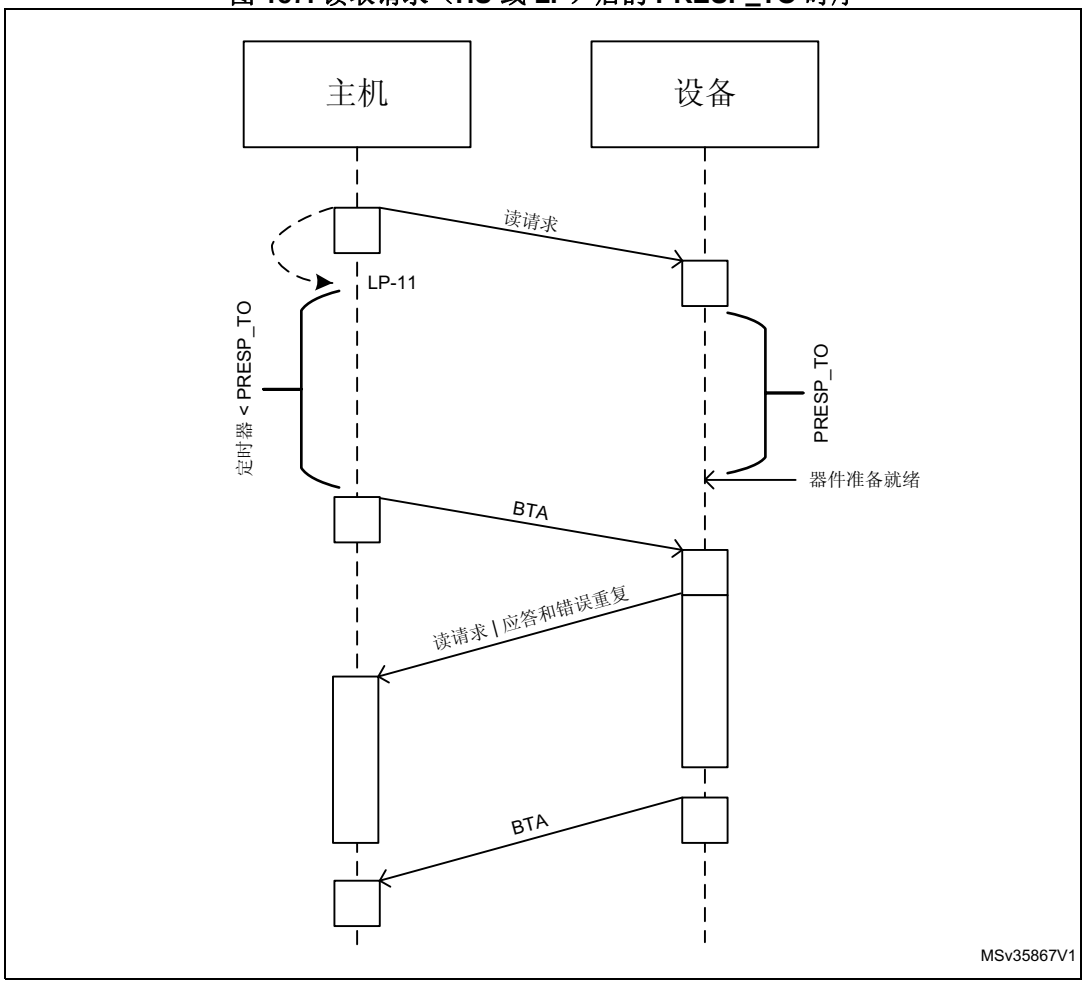


图 188. 写入请求（HS 或 LP）后的 PRESP_TO 时序

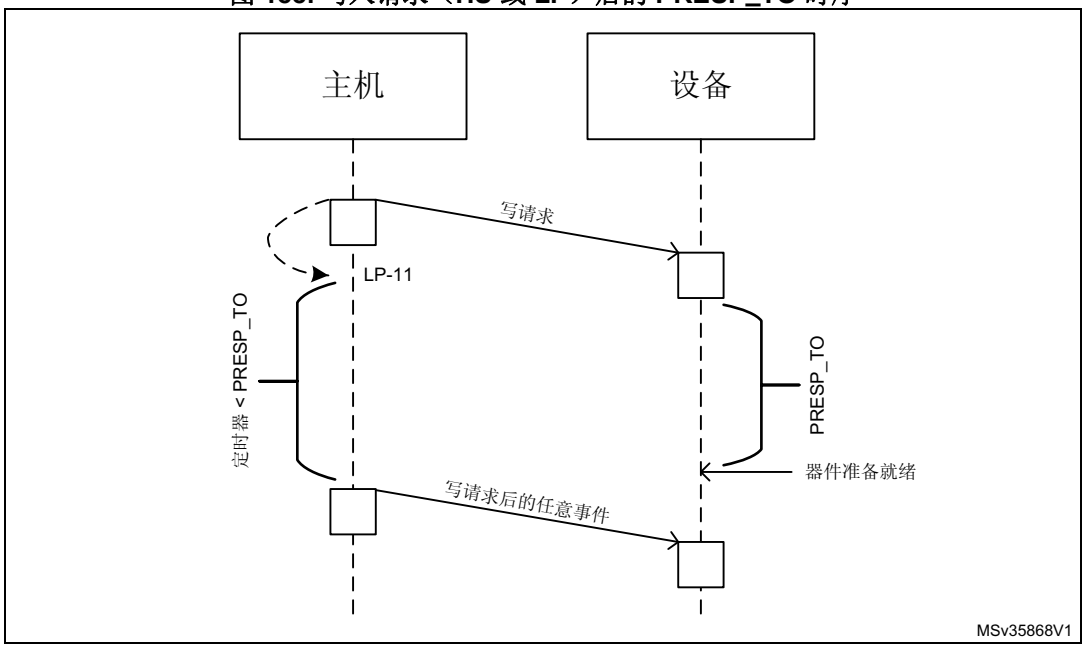


表 177 介绍用于配置 PRESP_TO 计数器的字段。

表 177. PRESP_TO 计数器配置

说明		寄存器	字段
DSI 主机使链路保持静止状态的周期	发送高速读操作后	DSI_TCCR1	HSRD_TOCNT
	发送低功耗读操作后	DSI_TCCR2	LPRD_TOCNT
	完成总线转向 (BTA) 后	DSI_TCCR5	BTA_TOCNT
DSI 主机使链路保持无效状态的周期	发送高速写操作后	DSI_TCCR3	HSWR_TOCNT
	发送低功耗写操作后	DSI_TCCR4	LPWR_TOCNT

这些寄存器中的值以通道字节时钟的周期数为单位。这些寄存器仅在命令模式下使用，因为在视频模式下，为了保持显示屏正确刷新，必须满足严格的时序调度，并且不能因这些调度或任何其他超时而断开。将给定超时设置为 0 将禁止进入 LP-11 状态和该类别事件的超时。

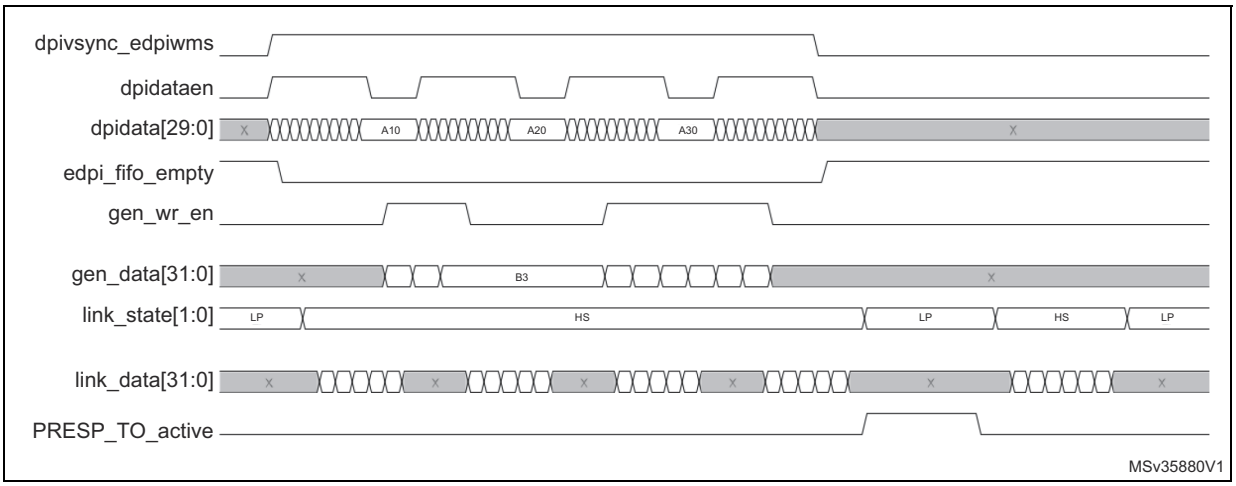
高速模式下的读取和写入请求与低功耗模式下的读取和写入请求不同。例如，如果 HSRD_TOCNT 设置为零且 LPRD_TOCNT 设置为非零值，则无参数的通用读取不会在高速模式下激活 PRESP_TO 计数器，但会在低功耗模式下激活 PRESP_TO。

DSI 主机超时计数器配置寄存器 4 (DSI_TCCR3) 包含一个特殊的 Presp 模式 (PM) 位，用于更改适配命令中 PRESP_TO 的正常行为模式，以实现高速写入操作超时。该位置 1 时，允许仅使用一次来自 HSWR_TOCNT 的 PRESP_TO，当满足以下两个条件时：

- LTDC VSYNC 信号上升和下降；
- 发送适配命令模式下源自 LTDC 接口的数据包，并且其 FIFO 再次为空。

在这种情况下，即使有来自通用接口的通信准备好发送，也不会将非适配命令模式请求发送到 D-PHY，同时会使 D-PHY 返回到停止状态。将非适配命令模式请求发送到 D-PHY 时，会激活 PRESP_TO 计数器，并且仅当此发送完成时，DSI 主机才会发送其他准备就绪的通信，如*图 189* 所示。

图 189. 准备模式为 1 时的影响

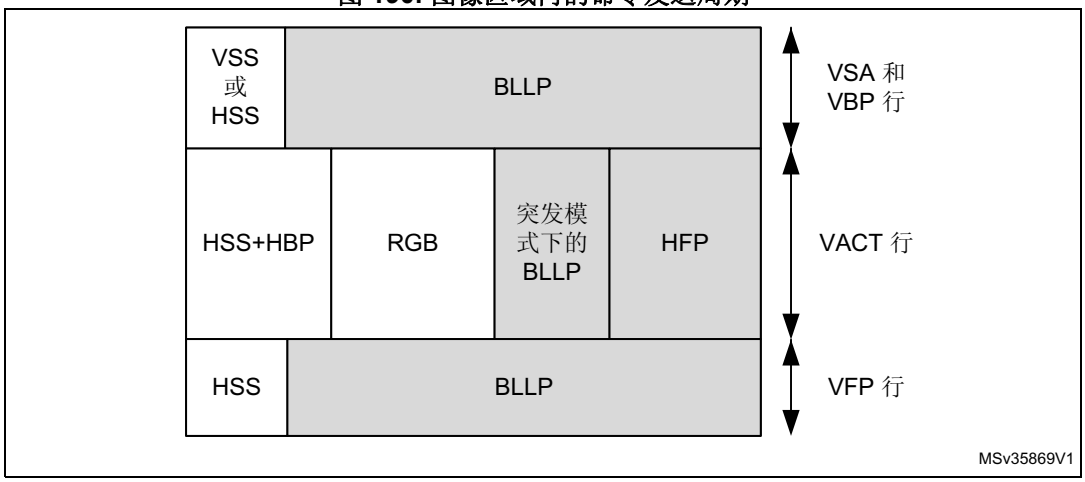


29.9 功能说明：命令发送

29.9.1 视频模式下的命令发送

在视频模式下，DSI 主机支持以高速和低功耗发送命令。DSI 主机使用消隐或低功耗 (BLLP) 周期发送通过 APB 通用接口插入的命令。上述周期对应于*图 190* 的灰色区域 。

图 190. 图像区域内的命令发送周期

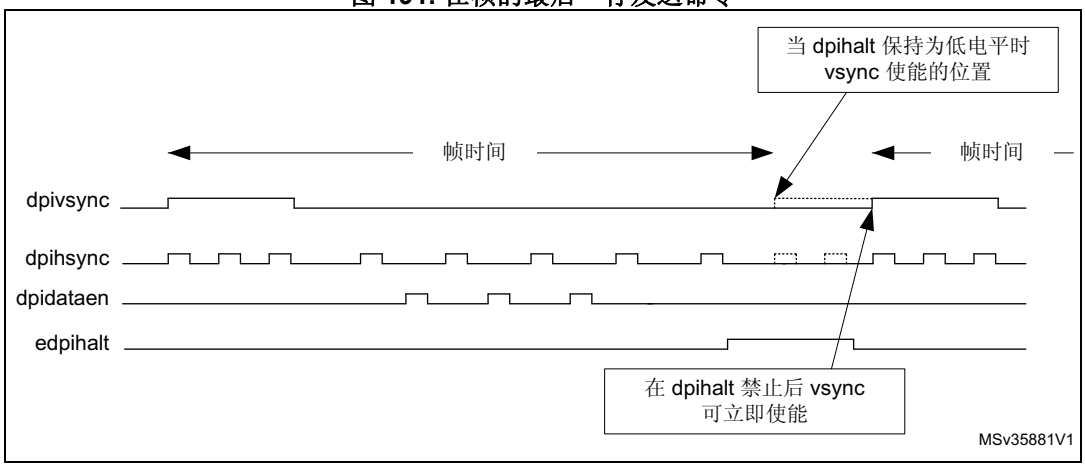


在以下数据包/状态之后的消隐周期中发送命令：

- 垂直同步起始 (VSS) 数据包（如果未使能视频同步脉冲）
- VSA、VBP 和 VFP 区域中的水平同步结束 (HSE) 数据包
- 水平同步起始 (HSS) 数据包（如果未在 VSA、VBP 和 VFP 区域中使能视频同步脉冲）
- 水平激活 (HACT) 状态

除了与 BLLP 对应的区域外，还可以在帧的最后一行期间发送较大的命令。在这种情况下，将违反视频模式的行时间，并且会将 edpihalt 信号设置为请求 DPI 视频时序信号以保持未激活状态。仅当命令不适合任何 BLLP 区域时，才会延迟到最后一行，从而导致违反视频模式的行时间，如*图 191* 所示。

图 191. 在帧的最后一行发送命令



即使是在帧的最后一行，每行也只发送一条命令，但每行都可以有一条命令。

在低功耗模式下每行只能发送一条命令。但是，每行都可以有一条低功耗命令。在高速模式下，DSI 主机可以发送多条命令，数量应尽可能多，以使其适应可用时间。

DSI 主机避免在最后一行发送命令，因为最后一行可能比其他行短。例如，行时间 (t_L) 可能比 LTDC 接口的 t_L 长半个周期，也就是说，帧中的每一行的时间为最后一行加上半个周期。这导致最后一行（½ 周期）×（行数 - 1）短于 t_L。

DSI 封装器控制寄存器 (DSI_WCR) 的 COLM 和 SHTDN 位也能触发命令数据包的发送。命令为：

- 颜色模式开启
- 颜色模式关闭
- 关闭外设
- 开启外设

这些命令不会在 VACT 区域中发送。如果 DSI 主机视频模式配置寄存器 (DSI_VMCR) 的低功耗命令使能 (LPCE) 位置 1，则这些命令将以低功耗模式发送。

在低功耗模式下，DSI 主机低功耗模式配置寄存器 (DSI_LPMCR) 的最大数据包大小 (LPSIZE) 字段用于确定是否可以发送这些命令。假定最大数据包大小 (LPSIZE) 大于或等于四个字节（短数据包中的字节数），因为 DSI 主机不在最后一行上发送这些命令。

如果 DSI 主机低功耗模式配置寄存器 (DSI_LPMCR) 的帧总线转向应答使能 (FBTAAE) 位置 1，则 DSI 主机会在帧的最后一行后生成 BTA。这可能与写入命令或读取命令一致。在任何一种情况下，LTDC 接口都会停止，直到收到应答（DSI 总线的控制权返回给主机）。

29.9.2 低功耗模式下的命令发送

DSI 主机可以配置为在高速视频模式发送期间发送低功耗命令。

要使能此功能，需要将 DSI 主机视频模式配置寄存器 (DSI_VMCR) 的低功耗命令使能位 (LPCE) 置 1。在这种情况下，需要计算可用时间（以字节为单位），从而在低功耗模式下将命令发送到水平前沿 (HFP)、垂直同步有效 (VSA)、垂直后沿 (VBP) 和垂直前沿 (VFP) 区域。

视频模式配置寄存器 (DSI_VMCR) 寄存器的第 8 位到第 13 位指示 DSI 主机是否可以在空闲时转到 LP。如果低功耗命令使能 (LPCE) 位置 1 且非视频数据包在队列中，则即使不允许 DSI 主机进入低功耗模式，也会在特定区域忽略低功耗配置并发送低功耗命令。在低功耗命令发送后，DSI 主机保持低功耗状态，直到发生同步事件。

例如，考虑将 VFP 选择为高速区域 (LPVFPE = 1'b0)，将 LPCE 设置为在 VPF 区域中以低功耗发送的命令。该命令在低功耗模式下发送，行保持低功耗模式，直到新的 HSS 到达。

计算在 VSA、VBP 和 VFP 区域中以 LP 模式发送命令的时间

DSI 主机低功耗模式配置寄存器 (DSI_LPMCR) 的最大数据包大小 (LPSIZE) 字段表示在低功耗模式下（基于逸出时钟）在 VSA、VBP 和 VFP 区域的行内发送命令的可用时间（以字节为单位）。

最大数据包大小 (LPSIZE) 的计算取决于使用的视频模式。

图 192 说明了使用同步脉冲的非突发视频模式的时序间隔，*图 193* 表示使用同步事件的突发和非突发视频模式的时序间隔。

图 192. 使用同步脉冲的非突发模式下的 LPSIZE 计算

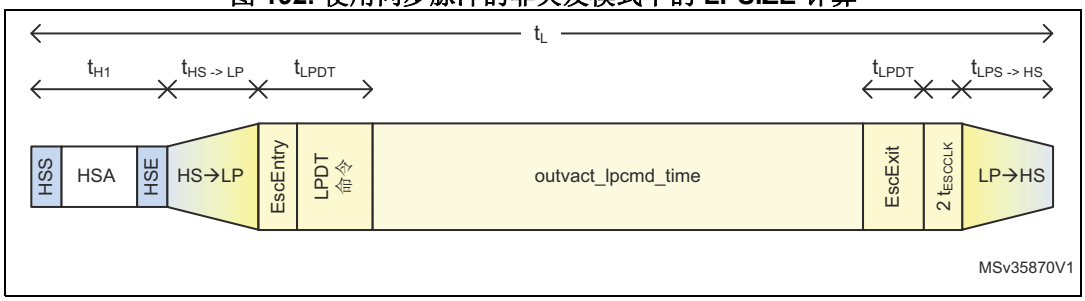
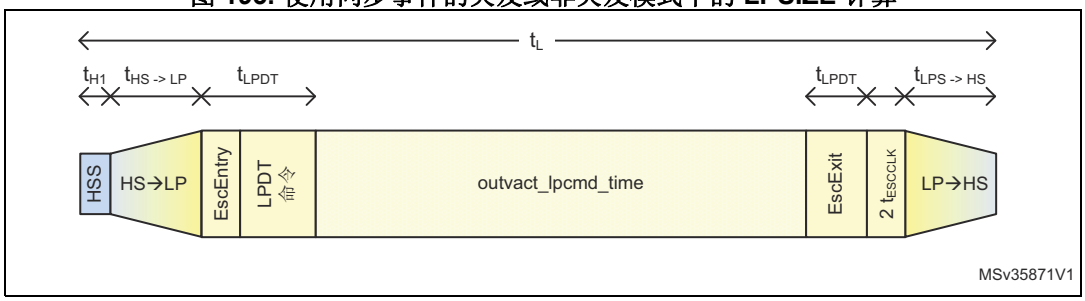


图 193. 使用同步事件的突发或非突发模式下的 LPSIZE 计算



该时间的计算方式如下：

$LPSIZE = (t_L - (t_{H1} + t_{HS->LP} + t_{LPHS} + t_{LPDT} + 2\ t_{ESCCLK})) / (2 \times 8 \times t_{ESCCLK})$，其中

- t_L = 行时间；
- t_{H1} = 同步脉冲模式下 HSA 脉冲的时间（*图 192*）或发送 HSS 数据包的时间，包括 EoTp（*图 193*）；
- $t_{HS->LP}$ = 进入低功耗模式的时间；
- $t_{LP->HS}$ = 退出低功耗模式的时间；
- t_{LPDT} = 与逸出模式进入、LPDT 命令和逸出退出相关的 D-PHY 时序。根据 D-PHY 规范，此值在 LP 模式下始终为 11 位（或 22 个 TX 逸出时钟周期）；
- t_{ESCCLK} = 在 DSI_CCR 寄存器的 TXECKDIV 字段中编程的逸出时钟周期；
- t_{ESCCLK} = DSI 主机实现引起的延迟。

在上面的等式中，除以 8 表示将可用时间转换为字节。除以 2 表示每两个逸出时钟周期发送一位。最大数据包大小 (LPSIZE) 字段可以直接与要发送命令的大小进行比较，从而确定是否有足够的时间发送命令。对于该字段，可在低功耗模式下发送的命令的最大大小限制为 255 个字节。发送 DCTRL 命令（例如低功耗模式下的关断和颜色）时，必须将此寄存器编程为大于或等于 4 个字节的值。

考虑帧速率为每行 12.4 µs，并假定逸出时钟频率为 20 MHz，通道比特率为 800 Mb/s。在这种情况下，可以在逸出模式下发送 124 位（即 124 位 = 12.4 µs * 20 MHz/2）。不过，需要考虑 D-PHY 协议和 PHY 时序。

做出以下假设：

- 通道字节时钟周期为 10 ns（每通道 800 Mb）；
- 逸出时钟周期为 50 ns (DSI_CCR.TXECKDIV = 5)；
- 视频以非突发模式发送，同步脉冲由 HSS 和 HSE 数据包限定；
- DSI 配置为双通道；
- D-PHY 需要 180 ns 从低功耗模式切换到高速模式 (DSI_DLTCR.LS2HS_TIME = 18)；
- D-PHY 需要 200 ns 从高速模式切换到低功耗模式 (DSI_DLTCR.HS2LP_TIME = 20)；
- t_{HSA} = 420 ns。

在此例中，13 个字节的命令可以按如下方式发送：

LPSIZE = (12.4 µs - (420 ns + 180 ns +200 ns + (22 × 50 ns + 2 × 50 ns))) / (2 × 8 × 50 ns) = 13 字节。

计算低功耗模式下 HFP 区域内的到发送命令的时间

DSIHOST 低功耗模式配置寄存器 (DSI_LPMCR) 的 VACT 最大数据包大小 (VLPSIZE) 字段表示在低功耗模式下（基于逸出时钟）在垂直活动区域 (VACT) 内发送命令的可用时间（以字节为单位）。

要计算 VACT 最大数据包大小 (VLPSIZE) 的值，请考虑使用的视频模式。*图 194* 显示了使用同步脉冲的非突发视频模式的定时间隔，*图 195* 显示了使用同步事件的非突发视频模式，*图 196* 显示了突发视频模式。

图 194. 使用同步脉冲的非突发模式下的 VLPSIZE 计算

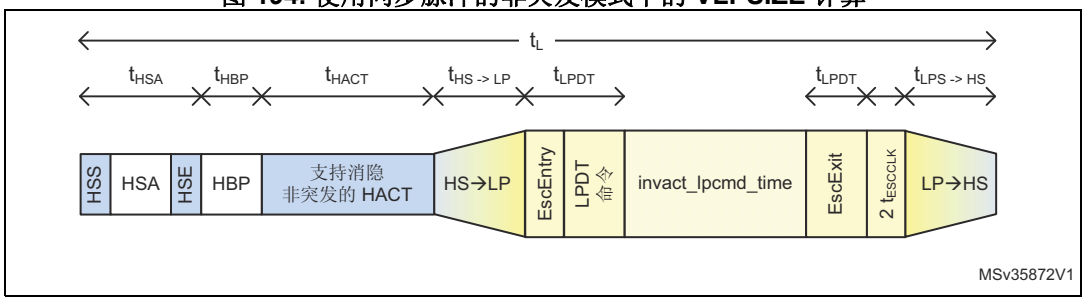


图 195. 使用同步事件的非突发模式下的 VLPSIZE 计算

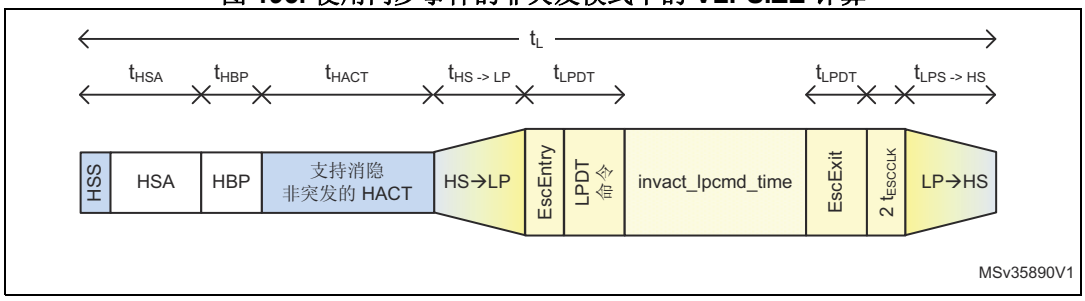
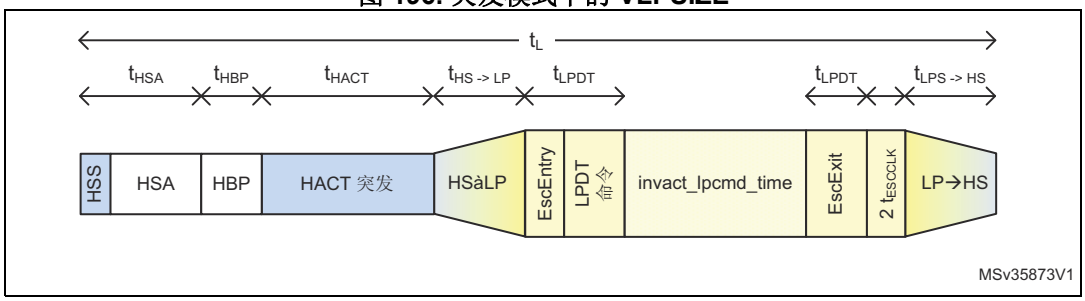


图 196. 突发模式下的 VLPSIZE



该时间的计算方式如下：

$VLPSIZE = (t_L - (t_{HSA} + t_{HBP} + t_{HACT} + t_{HS->LP} + t_{LP->HS} + t_{LPDT} + 2\ t_{ESCCLK})) / (2 \times 8 \times t_{ESCCLK})$

其中

- t_L = 行时间；
- t_{HSA} = HSA 脉冲 (DSI_VHSACR.HSA) 的时间；
- t_{HBP} = 水平后沿 (DSI_VHBPCR.HBP) 的时间；
- t_{HACT} = 视频激活的时间。对于突发模式，视频激活的时间经过压缩，计算方式为：t_{HACT} = VPSIZE * Bytes_per_Pixel /Number_Lanes * $t_{Lane_byte_clk}$；
- t_{ESCCLK} = 在 DSI_CCR 寄存器的 TXECKDIV 字段中编程的逸出时钟周期。

VLPSIZE 字段可以直接与要发送命令的大小进行比较，从而确定是否有时间发送命令。

考虑帧速率为每行 16.4 µs，并假定逸出时钟频率为 20 MHz，通道比特率为 800 Mb/s。在这种情况下，可以在逸出模式下发送 420 位（即 164 位 = 16.4 µs * 20 MHz/2）。不过，由于它是帧的垂直活动区域，因此除了 D-PHY 协议和 PHY 时序外，还要考虑 HSA、HBP 和 HACT 时序。做出以下假设：

- 有效通道数为 4；
- 通道字节时钟周期 (lanebyteclkperiod) 为 10 ns（每通道 800 Mb）；
- 逸出时钟周期为 50 ns (DSI_CCR.TXECKDIV = 5)；
- D-PHY 需要 180 ns 从低功耗模式转换到高速模式 (DSI_DLTCR.LP2HS_TIME = 18)；
- D-PHY 需要 200 ns 从高速模式转换到低功耗模式 (DSI_DLTCR.HS2LP_TIME = 20)；
- t_{HSA} = 420 ns；
- t_{HBP} = 800 ns；
- t_{HACT} = 12800 ns，以便以 24 bpp 发送 1280 像素；
- 在非突发模式下发送视频；
- DSI 主机配置为四通道。

在此示例中，考虑在非突发模式下发送视频。VLPSIZE 的计算方式如下：

VLPSIZE = (16.4 µs -(420 ns + 800 ns + 12.8 µs + 180 ns +200 ns +
(22 × 50 ns + 2 × 50 ns)) / (2 × 8 × 50 ns) = 1 字节

在此期间只能发送一个字节。一个短数据包（例如，通用短写）至少需要四个字节。因此，在本示例中，命令不会在 VACT 区域中发送。

如果使能了突发模式，则将有更多时间在 VACT 区域中发送命令，因为 HACT 的时间经过压缩。

VLPSIZE = (16.4 µs - (420 ns + 800 ns + (1280 × 3 / 4 × 10 ns) + 180 ns + 200 ns +
(22 × 50 ns + 2 × 50 ns) / (2 × 8 × 50 ns) = 5 字节

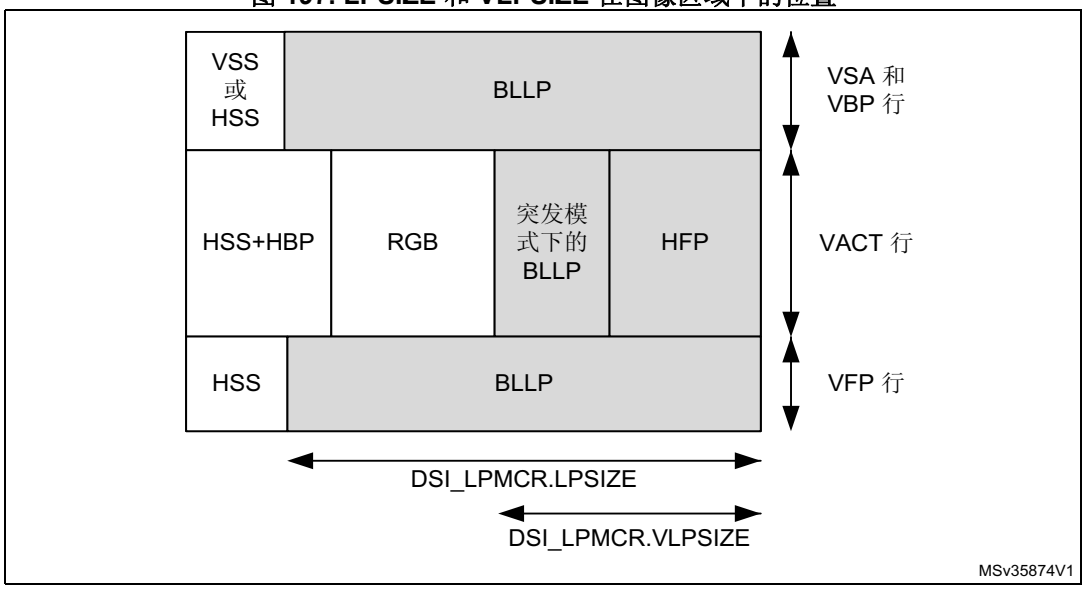
对于突发模式，VLPSIZE 为 5 个字节，可以发送一个 4 字节的短数据包。

不同周期的命令发送

允许对 LPSIZE 和 VLPSIZE 字段进行简单比较，以确定是否可以在任何 BLLP 周期发送命令。

图 197 说明了 VLPSIZE 和 LPSIZE 的含义，同时将它们与阴影区域和 VACT 区域相匹配。

图 197. LPSIZE 和 VLPSIZE 在图像区域中的位置



29.9.3 高速模式下的命令发送

如果DSI_VMCR 寄存器的 LPCE 位为 0，则在视频模式下高速发送命令。在这种情况下，DSI 主机将自动确定每个命令可以发送的区域，无需编程或计算。

29.9.4 读取命令发送

DSI_DLTCR 寄存器的 MRD_TIME 字段配置执行读取命令所需的最长时间（以通道字节周期为单位），计算方式为：

MRD_TIME = 低功耗模式下发送读取命令的时间 + 进入和退出低功耗模式的时间 + 从外设设备返回读取数据包的时间。

从外设返回读取数据包的时间取决于读取字节数和外设的逸出时钟频率，与主机的逸出时钟无关。MRD_TIME 字段用于高速和低功耗模式，以确定是否有时间在 BLLP 周期内完成读取命令。

在高速模式下 (LPCE = 0)，MRD_TIME 的计算公式如下：

$MRD_TIME = (t_{HS->LP} + t_{LP->HS} + t_{read} + 2 \times t_{BTA}) / lanebyteclkperiod$

在低功耗模式下 (LPCE = 1)，MRD_TIME 的计算公式如下：

MRD_TIME = ($t_{HS->LP}$ + $t_{LP->HS}$ + t_{LPDT} + t_{lprd} + t_{read} + 2 x t_{BTA}) / lanebyteclkperiod，其中：

- $t_{HS->LP}$ = 进入低功耗模式的时间；
- $t_{LP->HS}$ = 退出低功耗模式的时间；
- t_{LPDT} = 与逸出模式进入、LPDT 命令和逸出模式退出相关的 D-PHY 时序（根据 D-PHY 规范，该值在 LP 中始终为 11 位或 22 个 TX 逸出时钟周期）；
- t_{lprd} = 低功耗模式下的读取命令时间（64 * TX 逸出时钟）；
- t_{read} = 从外设返回读取数据包的时间；
- t_{BTA} = 执行总线转向的时间（取决于 D-PHY）。

建议将从外设读取的最大字节数保持为最小值，以便有足够的时间在行时间内发出读取命令。确保 MRD_TIME x 通道字节时钟周期小于主机 LPSIZE x 16 x 逸出时钟周期，否则读取命令将发送到帧的最后一行。如果需要读取大量参数 (> 16)，则在执行读取命令时增加 MRD_TIME。读取完成后，将 MRD_TIME 降至较低值。

如果在帧的最后一行发出读取命令，则 LTDC 接口停止并保持停止状态，直到读取命令正在执行。视频发送应在此期间停止。

29.9.5 低功耗模式下的时钟通道

为了降低 D-PHY 的功耗，不在高速模式下发送 DSI 主机时，允许时钟通道进入低功耗模式。控制器自动处理时钟通道从 HS（时钟通道有效发送时钟）到 LP 状态的转换，无需软件直接干预。可通过配置 DSI_CLCR 寄存器的 DPCC 和 ACR 位使能此功能。

在命令模式下，当 DSI 主机没有任何 HS 数据包要发送时，可以将时钟通道置于低功耗模式。

在视频模式（LTDC 接口）下，DSI 主机控制器使用其内部视频和 PHY 时序配置来确定时钟线是否有时间进入低功耗模式，并且不会影响像素数据和同步事件的视频数据发送。

视频模式正确配置后（请参见*第 29.5 节：功能说明：LTDC 接口上的视频模式*），DSI 主机需要了解时钟通道在高速模式与低功耗模式之间切换所需的时间。可从 D-PHY 规范获得所需的值：使用以下值编程 DSI_CLTCR 寄存器：

- HS2LP_TIME = 时钟通道中从 HS 到 LP 的时间/HS 中的字节时钟周期 (lanebyteclk)
- LP2HS_TIME = 时钟通道中从 LP 到 HS 的时间/HS 中的字节时钟周期 (lanebyteclk)

根据编程的值，DSI 主机计算视频帧的未激活区域中是否有足够的时间使时钟通道进入低功耗模式。

针对以下三种可能的情况，DSI 主机确定了实现节能的最佳方式：

- 没有足够的时间进入低功耗模式。因此，按照*图 198* 所示添加消隐周期；
- 数据通道有足够的时间进入低功耗模式，但没有足够的时间使时钟通道进入低功耗模式，请参见*图 199*。
- 数据通道和时钟通道有足够的时间进入低功耗模式，如*图 200* 所示。

图 198. HS 中的时钟通道和数据通道



图 199. HS 中的时钟通道和 LP 中的数据通道

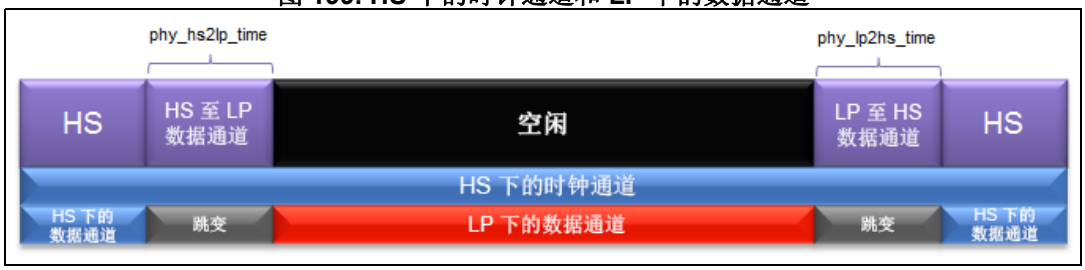


图 200. LP 中的时钟通道和数据通道

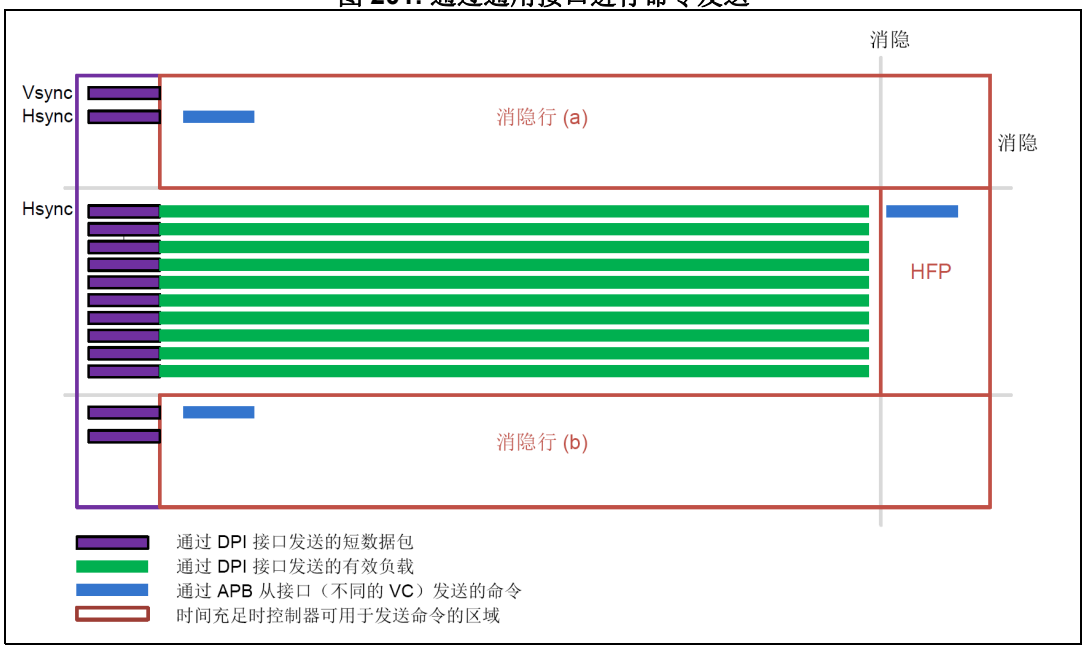


29.10 功能说明：虚拟通道

DSI 主机支持选择虚拟通道 (VC) 用于每个接口。使用多个虚拟频道，当每个显示屏具有不同的虚拟通道标识符时，系统可以同时寻址多个显示屏。

为特定的虚拟通道配置 LTDC 接口时，可以使用 APB 从通用接口在视频流发送时发出命令。这样，便可通过正在进行的视频流发送命令，同时寻址不同的虚拟通道，从而使接口支持多个显示屏。视频模式下，视频流发送具有最高优先级。因此，只有在视频流发送中存在可用时间时，才会传输边带数据包（例如，通用接口数据包）。DSI 主机识别可用时间段并加以利用来传输通用接口数据包。*图 201* 显示了 DSI 主机在 LTDC 接口发送的视频流中插入来自 APB 通用接口的数据包的位置。

图 201. 通过通用接口进行命令发送



也可以使用不同的虚拟通道寻址仅有一个通用接口的多个显示屏。由于通用接口可通过配置不限于任何特定的虚拟通道，因此可以发送具有不同虚拟通道的数据包。这使接口能够将数据包的时间复用，提供给具有不同虚拟通道的显示屏。

可以使用以下配置寄存器选择与 LTDC 和 APB 从通用接口上的发送相关的虚拟通道 ID：

- DSI_LVCID.VCID 字段配置使用 LTDC 接口索引到视频模式数据包的虚拟通道 ID。
- DSI_GHCR 寄存器配置使用 APB 从通用接口进行发送的数据包报头（包含要使用的虚拟通道 ID）。
- DSI_GVIDR.VCID 字段配置读取响应的虚拟通道 ID 以存储并返回到通用接口。

29.11 功能说明：视频模式测试图发生器

视频模式测试图发生器允许在无任何刺激的情况下发送水平/垂直彩条和 D-PHY BER 测试模式。

帧请求必须在*表 178* 列出的视频寄存器中定义。

表 178. 帧请求配置寄存器

寄存器名称	说明
DSI 主机视频模式配置寄存器	视频模式配置
DSI 主机视频数据包配置寄存器	视频数据包大小
DSI 主机视频块配置寄存器	块数
DSI 主机视频空包配置寄存器	空包大小
DSI 主机视频 HSA 配置寄存器	水平同步活动时间
DSI 主机视频 HBP 配置寄存器	水平后沿时间
DSI 主机视频行配置寄存器	行时间
DSI 主机视频 VSA 配置寄存器	垂直同步活动周期
DSI 主机视频 VBP 配置寄存器	垂直后沿周期
DSI 主机视频 VFP 配置寄存器	垂直前沿周期
DSI 主机视频 VA 配置寄存器	垂直分辨率

29.11.1 彩条测试卡

彩条模式包含白色、黄色、青色、绿色、紫色、红色、蓝色和黑色等八种彩条。

通过将行像素大小（垂直模式）或行数（水平模式）除以 8 来计算各颜色宽度。在垂直彩条模式下（*图 202*），每个彩条的宽度为行中的像素数除以 8。如果某行中的像素数不能被 8 整除，则最后一种颜色（黑色）包含不能整除部分的像素数。

在水平彩条模式下（*图 203*），每个彩条的宽度为行中的帧数除以 8。如果某帧中的行数不能被 8 整除，则最后一种颜色（黑色）包含不能整除部分的行数。

图 202. 垂直彩条模式

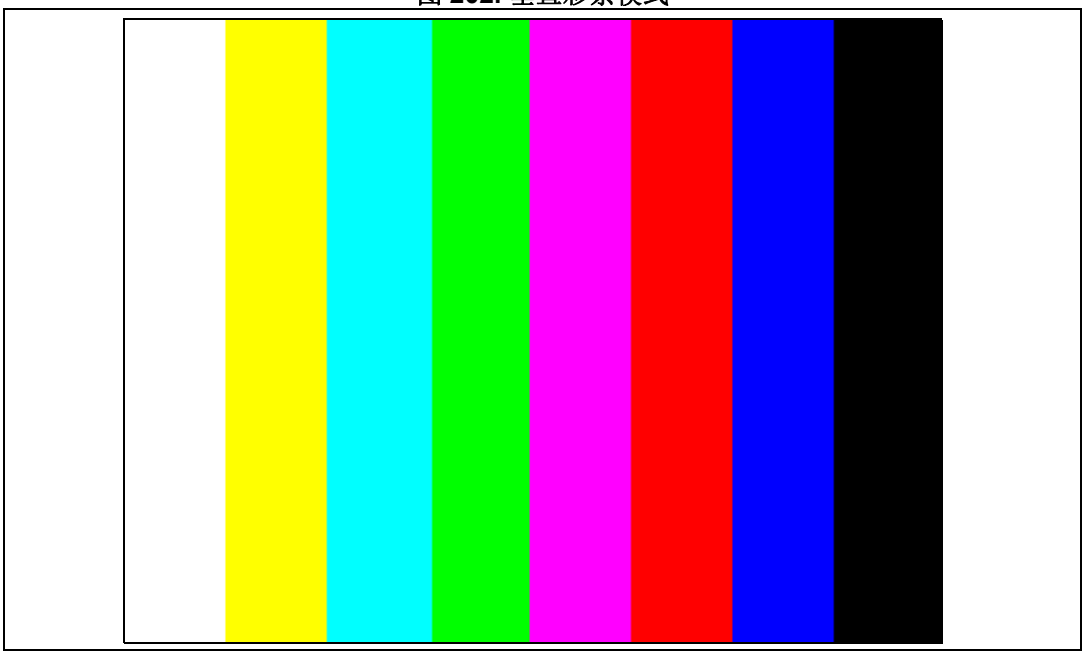


图 203. 水平彩条模式



29.11.2 颜色编码

表 179 显示使用的 RGB 分量。

表 179. RGB 分量

	白色	黄色	青色	绿色	品红色	红色	蓝色	黑色
R	高	高	低	低	高	高	低	低
G	高	高	高	高	低	低	低	低
B	高	低	高	低	高	低	高	低

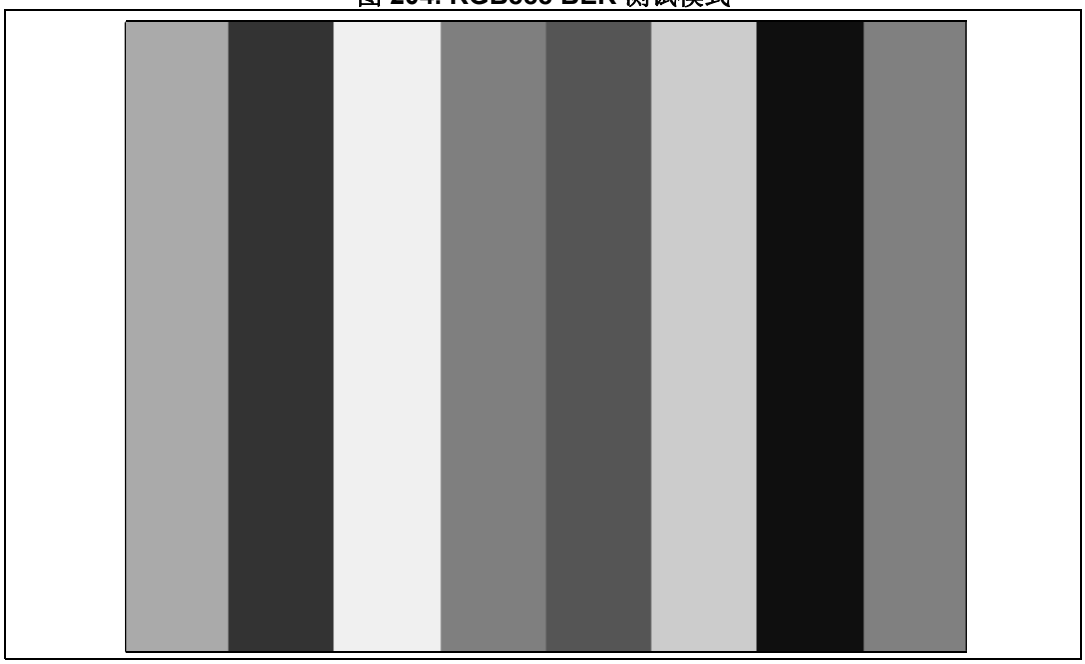
29.11.3 BER 测试模式

BER 测试模式简化了一致性测试。该模式测试 RX D-PHY 功能是否能够正确接收数据。需要下列数据模式：

- 0xAA 的 X 个字节（高频模式，反相）；
- 0x33 的 X 个字节（中频模式）；
- 0xF0 的 X 个字节（低频模式，反相）；
- 0x7F 的 X 个字节（通道 0 模式）；
- 0x55 的 X 个字节（高频模式）；
- 0xCC 的 X 个字节（中频模式，反相）；
- 0x0F 的 X 个字节（低频模式）；
- 0x80 的 Y 个字节（通道 1 模式）。

大多数情况下，Y 等于 X。但是，根据使用的行长和颜色编码，Y 可能和 X 不同。使用 RGB888 颜色编码且水平分辨率为 8 的倍数时，*图 204* 中显示的模式出现在 DSI 显示屏上。

图 204. RGB888 BER 测试模式



29.11.4 视频模式测试图发生器分辨率

根据方向、BER 模式和颜色编码，视频模式测试图发生器接受的最小分辨率为：

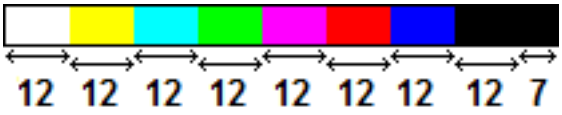
- BER 模式：8x8；
- 水平彩条模式：8x8；
- 垂直彩条模式：8x8。

垂直模式

各颜色栏的宽度由八种测试模式颜色的水平分辨率（像素）的除法确定。如果水平分辨率不能被 8 整除，则会延长最后一种颜色（黑色）以填充分辨率。

在*图 205* 的示例中，水平分辨率为 103。

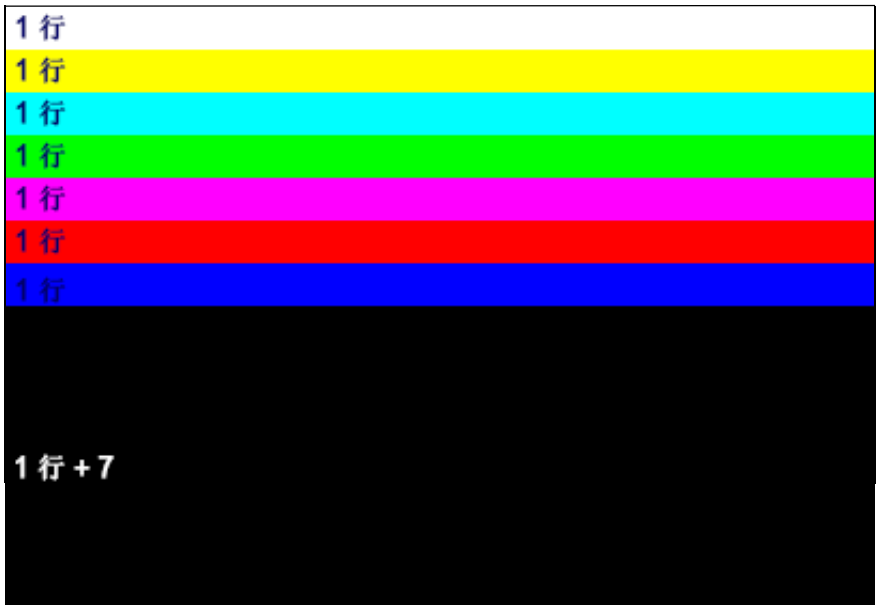
图 205. 垂直模式 (103x15)



水平模式

各彩条的宽度由八种测试模式颜色的垂直分辨率数（行）的除法确定。如果垂直分辨率不能被 8 整除，则会延长最后一种颜色（黑色）以填充分辨率，如*图 206* 所示。

图 206. 水平模式 (103x15)



29.12 功能说明：D-PHY 管理

嵌入式 MIPI® D-PHY 由 DSI 主机直接控制，并通过 DSI 封装器配置。

此位，还内置专用 PLL 和专用 1.2 V 稳压器，为 DSI 主机和 D-PHY 提供时钟和电源。

29.12.1 D-PHY 配置

凭借 DSI_WPCRx 寄存器，通过 DSI 封装器执行 D-PHY 配置。

时序定义

MIPI® D-PHY 使用专用定时器管理所有通信时序。由于所有时序都以纳秒 (ns) 为单位，因此必须配置单位间隔字段以确保所有时序的持续时间正常。

通过 DSI_WPCR0.UIX4 字段配置单位间隔。此值以 0.25 ns 为单位定义高速模式下的位周期。如果该周期不是 0.25 ns 的倍数，则应将驱动值向下舍入。

例如，对于 300 Mb/s 的链路，单位间隔为 3.33 ns，因此 UIX4 应为 13.33。在这种情况下，写入值为 13 (0x0D)。

引脚的压摆率和延迟调整

要微调 DSI 通信，需要微调压摆率和延迟：

- 数据通道和时钟通道中高速发送的压摆率
- 数据通道和时钟通道中低功耗发送的压摆率
- 数据通道和时钟通道中高速发送的发送延迟

表 180. 压摆率和延迟调整

功能	通道	DSI_WPCR1 中的值域
高速发送下的压摆率	时钟通道	HSTXSRCCL
	数据通道	HSTXSRCDL
低功耗发送下的压摆率	时钟通道	LPSRCCL
	数据通道	LPSRCDL
高速发送延迟	时钟通道	HSTXDCL
	数据通道	HSTXDDL

所有此参数的默认值为 2'h00。所有值应仅在 DSI 停止（DSI_WCR.DSIEN = 0 且 CR.EN = 0）时编程。

低功耗接收过滤器调整

可通过 DSI_WPCR1 寄存器的 LPRXFT 字段微调低功耗接收器的低通截止频率。默认值为应 2'h00，且应仅在 DSI 停止（CR.DSIEN = 0 且 CR.EN = 0）时编程。

特殊 Sdd 控制

可以在时钟通道和数据通道上激活其他电流路径以满足 MIPI® D-PHY 规范中定义的 Sdd_{TX} 参数。

可通过将 DSI_WPCR1 寄存器中的 SDDC 位置 1 来完成此操作。

自定义通道配置

为了简化 DSI 集成，可按照*表 181* 所述交换通道引脚和/或反转通道上的高速信号。

表 181. 自定义通道配置

功能	通道	DSI_WPCR0 中的使能位
交换通道引脚	时钟通道	SWCL
	数据通道 0	SWDL0
	数据通道 1	SWDL1
反转通道上的高速信号	时钟通道	HSICL
	数据通道 0	HSIDL0
	数据通道 1	HSIDL1

自定义时序配置

可根据特定需要调整某些 MIPI® D-PHY 时序，如*表 182* 所述。

表 182. 自定义时序参数

MIPI® 时序	DSI_WPCR0 中的使能位	配置寄存器	字段	默认值	默认持续时间
$t_{CLK-POST}$	TCLKPOSTEN	DSI_WPCR4	TCLKPOST	200	100 ns + 120*UI
t_{LPX}（时钟通道）	TLPXCEN	DSI_WPCR3	TLPXC	100	50 ns
t_{HS_EXIT}	THSEXITEN		THSEXIT	200	100 ns + 40*UI
t_{LPX}（数据通道）	TLPXDEN		TLPXD	100	50 ns
$t_{HS-ZERO}$	THSZEROEN		THSZERO	175	175 ns + 8*UI
$t_{HS-TRAIL}$	THSTRAIL	DSI_WPCR2	THSTRAIL	140	70 ns + 8*UI
$t_{HS-PREPARE}$	THSPREPEN		THSPREP	126	63 ns + 12*UI
$t_{CLK-ZERO}$	TCLKZEROEN		TCLKZERO	195	390 ns
$t_{CLK-PREPARE}$	TCLKPREPEN		TCLKPREP	120	60 ns + 20*UI

所有值应仅在 DSI 停止（CR.DSIEN = 0 且 CR.EN = 0）时编程。

29.12.2 特殊 D-PHY 操作

DSI 封装器具有某些控制位，可在某些特殊状态和/或行为下强制 D-PHY。

强制通道状态

可通过 DSI_WPCR1 寄存器的 FTXSMDL 和 FTXSMCL 位强制 TX 停止模式下的数据通道和/或时钟通道。

将此位置 1 可使相应通道模块立即跳转到发送控制模式并开始发送停止状态 (LP-11)。

此功能可用于在错误 BTA 序列后返回到 TX 模式。

在低功耗模式下强制低功耗接收器

DSI_WPCR1 寄存器的 FLPRXLPM 位可使能低功耗接收器 (LPRX) 的低功耗模式。此位置 1 时，LPRX 始终在低功耗模式下运行。此位未置 1 时，LPRX 仅在 ULPS 期间在低功耗模式下运行。

禁止开启数据通道

此位置 1 时，即使从另一端接收总线转向请求 (BTA)，DSI_WPCR0 寄存器的 TDDL 位也会强制数据通道保持在接收模式下。

29.12.3 特殊低功耗 D-PHY 功能

嵌入式 D-PHY 提供了两个特定功能来优化功耗。

通道上的下拉电阻

每条通道上的 D-PHY 嵌入式下拉电阻可避免未使用通道时出现浮空状态。

DSI_WPCR0 寄存器的 PDEN 位置 1 时，可使能通道上的下拉电阻。

禁止数据通道争用检测

可关闭数据通道上的争用检测器来降低 D-PHY 的整体功耗。

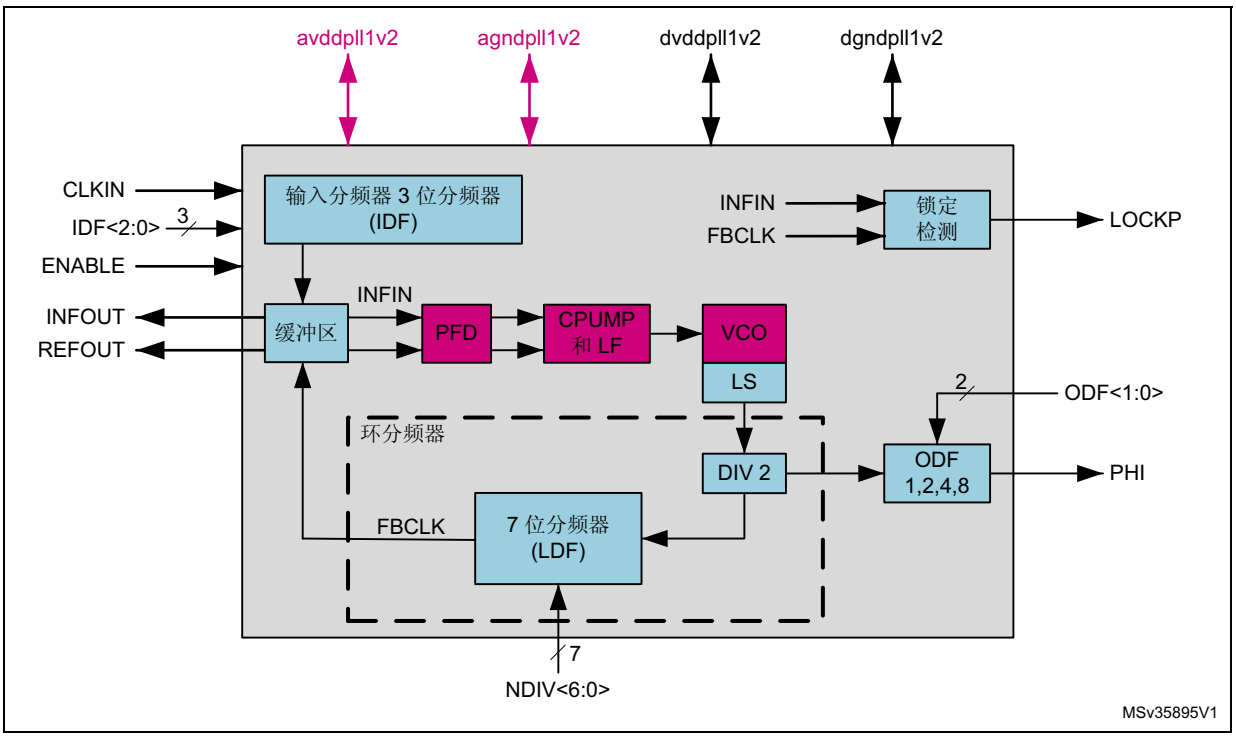
DSI_WPCR0 寄存器的 CDOFFDL 位置 1 时，会禁止数据通道上的争用检测。

可在正向逃逸模式中通过此操作降低静态功耗。

29.12.4 DSI PLL 控制

专用 DSI PLL 通过 DSI 封装器进行控制，如*图 207* 所示（模拟块和信号为粉色，数字信号为黑色，数字块为浅蓝色）。

图 207. PLL 框图



PLL 输出频率通过 DSI_WRPCR 寄存器字段配置。VCO 频率和 PLL 输出频率的计算方式如下：

$$F_{VCO} = (CLK_{IN} / IDF) * 2 * NDIV$$

$$PHI = F_{VCO} / (2 *ODF)$$

其中：

- CLK_{IN} 的范围是 4 到 100 MHz；
- DSI_WRPCR.NDIV 的范围是 10 到 125；
- DSI_WRPCR.IDF 的范围是 1 到 7；
- INFIN 的范围是 4 到 25 MHz；
- F_{VCO} 的范围是 500 MHz 到 1 GHz；
- DSI_WRPCR.ODF 可以是 1、2、4 或 8；
- PHI 的范围是 31.25 到 500 MHz。

通过将 DSI_WRPCR 寄存器中的 PLLEN 位置 1 来使能 PLL。

PLL 锁定后，DSI_WISR 中的 PLLLIF 位置 1。如果 DSI_WIER 中的 PLLLIE 位置 1，则会产生中断。

可通过 DSI_WISR 寄存器中的 PLLLS 标志监视 PLL 状态（锁定或解锁）。

当 PLL 解锁时，DSI_WISR 的 PLLUIF 位置 1。如果 DSI_WIER 的 PLLUIE 位置 1，则会产生中断。

仅当禁止 PLL 时，才能更改 DSI PLL 设置。

29.12.5 稳压器控制

通过 DSI Wrapper 控制提供 1.2 V 电压的 DSI 稳压器。

通过将 DSI_WRPCR 寄存器的 REGEN 位置 1 来使能稳压器。

当稳压器就绪时，将 DSI_WISR 寄存器的 RRIF 位置 1。如果 DSI_WIER 的 RRIE 位置 1，则会产生中断。

可通过 DSI_WISR 寄存器中的 RRS 标志监视稳压器状态（就绪或未就绪）。

请注意，D-PHY 没有单独的上电控制位。通过直接使能 1.2 V 稳压器完成 D-PHY 的电源开/关。

禁止 1.2 V 稳压器时，D-PHY 的 3.3 V 部件自动断电。

29.13 功能说明：中断和错误

DSI 主机或 DSI 封装器均可以产生中断。

所有中断均合并成一个中断通道进入中断控制器。

29.13.1 DSI 封装器中断

发生如下事件时可产生中断：

- 撕裂效应事件；
- 刷新结束；
- PLL 已锁定；
- PLL 未锁定；
- 稳压器就绪。

可以使用单独的中断使能位以提高灵活性。

表 183. DSI 封装器中断请求

中断事件	DSI_WISR 中的事件标志	DSI_WIER 中的使能控制位
撕裂效应	TEIF	TEIE
刷新结束	ERIF	ERIE
PLL 已锁定	PLLLIF	PLLLIE
PLL 未锁定	PLLUIF	PLLUIE
稳压器就绪	RRIF	RRIE

29.13.2 DSI 主机中断和错误

DSI_ISR0 和 DSI_ISR1 寄存器与错误状况报告相关。这些寄存器可触发通知系统发生错误的中断。

DSI_ISR0 或 DSI_ISR1 寄存器中发生错误时，DSI 主机中的一条中断线置为高电平。

可通过对屏蔽寄存器 DSI_IER0 和 DSI_IER1 进行编程来屏蔽中断触发。默认情况下会屏蔽所有错误。当上述寄存器中的任意位置 1 时，可使能特定错误中断。错误位始终在相应 DSI_ISR 寄存器中置 1。DSI_ISR0 和 DSI_ISR1 寄存器始终在读取操作后清零。读取所有造成中断的寄存器后，中断线清零。

中断强制寄存器（DSI_FIR0 和 DSI_FIR1）用于测试，允许单独触发中断事件，无需激活触发中断源的条件；原因是为测试产生刺激的过程非常复杂。此功能还有助于开发和测试与中断事件相关的软件。将这些寄存器的任意位置 1 将触发相应的中断。

图 208 中的浅黄色框表示某些错误的位置。

图 208. 错误源

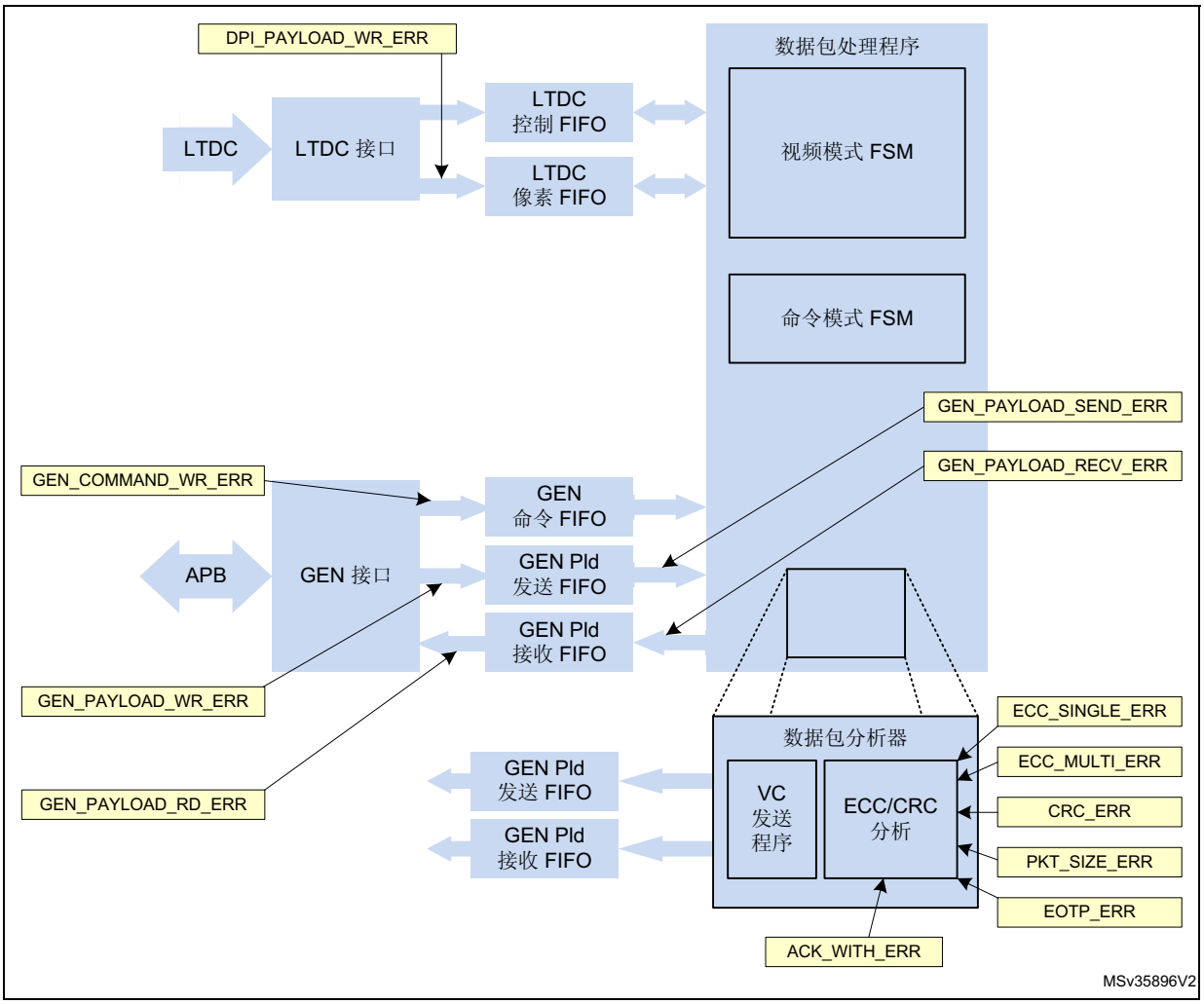


表 184 解释了引发这些中断的原因，并介绍了如何从这些中断恢复。

表 184. 错误原因和恢复

DSI 主机中断和状态寄存器	位	名称	错误原因	处理错误的推荐方法
0	20	PE4	D-PHY 报告 LP1 争用错误。 D-PHY 主机在尝试驱动线到高电平时检测争用。	从争用恢复 D-PHY。 再次复位 DSI 主机并发送数据包。如果此错误反复发生，请仔细分析主机与设备之间的连接。
0	19	PE3	D-PHY 报告 LP0 争用错误。 D-PHY 主机在尝试将线驱动到低电平时检测争用。	从争用恢复 D-PHY。 再次复位 DSI 主机并发送数据包。如果此错误反复发生，请仔细分析主机与设备之间的连接。
0	18	PE2	D-PHY 报告故障控制错误。 D-PHY 在通道 0 线路中检测到错误的线路状态序列。	设备未按预期运行，未与设备正确建立通信。这是一个不可恢复的错误。 复位 DSI 主机和 D-PHY。如果此错误反复发生，请分析设备的行为。
0	17	PE1	D-PHY 报告 LPDT 错误。 D-PHY 检测到 LDPT 不匹配 8 位的倍数。	这种数据接收不可靠。D-PHY 恢复，但从设备接收的数据可能不可靠。 建议复位 DSI 主机并重复 RX 发送。
0	16	PE0	D-PHY 报告逸出进入错误。 D-PHY 未识别接收的逸出进入代码。	D-PHY 未识别逸出进入代码。发送被忽略。 D-PHY 主机恢复，但系统应重复 RX 接收。
0	15	AE15	此错误直接从错误数据包的应答中检索。 设备在接收中检测到协议违例。	请参见显示文档。当此错误激活时，设备应执行另一条回读命令报告关于此错误的附加信息。读取附加信息并采取相应操作。
0	14	AE14	错误数据包的应答中包含此错误。 设备选择将该位用于错误报告。	请参见设备文档，了解产生此错误的可能原因并采取适当措施。
0	13	AE13	错误数据包的应答中包含此错误。 设备报告的发送长度与数据包长度不匹配。	这种错误的可能原因是数据包报头中存在多个错误（超过 2 个），因此错误检测失败，设备未丢弃数据包。在这种情况下，数据包报头损坏，可能造成解码不匹配。 再次发送数据包。如果此错误反复发生，请仔细分析主机与设备之间的连接。
0	12	AE12	错误数据包的应答中包含此错误。 设备无法识别至少一个接收数据包中的 VC ID。	这种错误的可能原因是数据包报头中存在多个错误（超过 2 个），因此错误检测失败，设备未丢弃数据包。在这种情况下，数据包报头损坏，可能造成解码不匹配。 再次发送数据包。如果此错误反复发生，请仔细分析主机与设备之间的连接。

表 184. 错误原因和恢复（续）

DSI 主机中断和状态寄存器	位	名称	错误原因	处理错误的推荐方法
0	11	AE11	错误数据包的应答中包含此错误。 设备无法识别至少一个接收数据包的数据类型。	检查设备功能。设备可能不支持某些数据包。 重复发送。
0	10	AE10	错误数据包的应答中包含此错误。 设备检测到至少一个接收数据包中存在 CRC 错误。	在最后的应答请求之后发送的某些长数据包可能包含有效负载中的 CRC 错误。 如果有效负载内容非常重要，则再次发送数据包。 如果此错误反复发生，请仔细分析主机与设备之间的连接。
0	9	AE9	错误数据包的应答中包含此错误。 设备检测到至少一个接收数据包中存在多位 ECC 错误。	设备不解析在最后的应答请求之后发送的数据包。 如果数据包非常重要，则再次发送数据包。 如果此错误反复发生，请仔细分析主机与设备之间的连接。
0	8	AE8	错误数据包的应答中包含此错误。 设备检测到至少一个接收数据包中存在 1 位 ECC 错误并加以更正。	无需任何操作。 设备应答数据包。 如果此错误反复发生，请分析信号完整性或链路的噪声条件。
0	7	AE7	错误数据包的应答中包含此错误。 设备通过 LP0/LP1 检测检测到线路争用。	此错误可能会影响低功耗数据接收和发送。忽略数据包，并再次发送。设备自动恢复。 如果此错误反复发生，请检查设备功能以及主机与设备之间的连接。 请参见 DSI 规范 1.1 的第 7.2.1 节。
0	6	AE6	错误数据包的应答中包含此错误。 设备检测到故障控制错误。	设备检测到以下情况之一： – LP-10（LP 请求）后面不带有效逸出或转向序列的剩余部分。 – LP-01（HS 请求）后面不带桥状态 (LP-00)。 D-PHY 通信损坏。此错误不可恢复。 复位 DSI 主机和 D-PHY。 请参见 DSI 规范 1.1 的第 7.1.6 节。
0	5	AE5	错误数据包的应答中包含此错误。 HS 接收和 LP 发送的显示超时计数器到期。	主机和设备超时计数器可能未正确配置。设备 HS_TX 超时应短于主机 HS_RX 超时。主机 LP_RX 超时应长于设备 LP_TX 超时。检查并确认主机配置与设备规范一致。该错误会自动恢复，尽管无法确保发送或接收的所有数据包都已完成。有关此错误的更多信息，请参见 DSI 规范 1.1 的第 7.2.2 节。

表 184. 错误原因和恢复（续）

DSI 主机中断和状态寄存器	位	名称	错误原因	处理错误的推荐方法
0	4	AE4	错误数据包的应答中包含此错误。 设备报告 LPDT 未以 8 位边界对齐	无法确保设备正确接收数据包。 再次发送数据包。有关此错误的更多信息，请参见 DSI 规范的第 7.1.5 节。
0	3	AE3	错误数据包的应答中包含此错误。 设备未识别逸出模式进入命令。	设备未识别逸出模式进入代码。 检查设备功能。有关此错误的更多信息，请参见 DSI 规范的第 7.1.4 节。 重复向设备发送。
0	2	AE2	错误数据包的应答中包含此错误。 检测到 EoT 序列时，设备检测到 HS 发送未以 8 位边界结束。	无法确保设备正确接收数据包。应重新发送。再次发送数据包。有关此错误的更多信息，请参见 DSI 规范 1.1 的第 7.1.3 节。
0	1	AE1	错误数据包的应答中包含此错误。 设备检测到 SoT 前导序列已损坏。	设备丢弃传入的发送。主机应执行重新发送。有关此错误的更多信息，请参见 DSI 规范 1.1 的第 7.1.2 节。
0	0	AE0	错误数据包的应答中包含此错误。 设备报告接收到 SoT 序列且存在错误，但仍可实现同步。	设备容许 SoT 序列中存在单个位和某些多位错误，但数据包的正确性受到影响。如果数据包内容非常重要，则再次发送数据包。有关此错误的更多信息，请参见 DSI 规范 1.1 的第 7.1.1 节。
1	12	GPRXE	通用读取 FIFO 中发生上溢。	未针对最大回读数据包大小正确标注读取 FIFO 大小。 配置设备以返回大小适合主机标注 FIFO 的读取数据。存储在 FIFO 中的数据已损坏。 复位 DSI 主机并重复读取步骤。
1	11	GPRDE	通用读取 FIFO 中发生下溢。	系统不等待读取过程结束并开始从 FIFO 检索数据。在完全接收之前请求读取数据。数据被损坏。 复位 DSI 主机并重复读取步骤。读取数据之前通过 APB 接口检查读取步骤是否完成。
1	10	GPTXE	通用写入有效负载 FIFO 中发生下溢。	在相应的数据包有效负载完全加载到有效负载 FIFO 之前，系统写入数据包报头。此错误不可恢复，发送的数据包已损坏。 复位 DSI 主机并重复写入步骤。
1	9	GPWRE	通用写入有效负载 FIFO 中发生上溢。	有效负载 FIFO 大小标注错误，无法存储长数据包的总有效负载。存储在 FIFO 中的数据被损坏。 复位 DSI 主机并重复写入步骤。

表 184. 错误原因和恢复（续）

DSI 主机中断和状态寄存器	位	名称	错误原因	处理错误的推荐方法
1	8	GCWRE	通用命令 FIFO 中发生上溢。	命令 FIFO 大小标注错误，无法存储突发数据包的总报头。存储在 FIFO 中的数据被损坏。 复位 DSI 主机并重复写入步骤。
1	7	LPWRE	DPI 像素有效负载 FIFO 中发生上溢。	控制器 FIFO 尺寸的运行分辨率未正确设置。检查视频模式配置寄存器。此分辨率应与 LTDC 视频分辨率保持一致。像素数据序列被损坏。 复位 DSI 主机并重新启动视频发送。
1	6	EOTPE	主机接收的发送未以发送结束数据包结束。	此错误对于已接收数据包的数据完整性不重要。 检查设备是否支持 EoTp 数据包的发送。
1	5	PSE	主机接收的发送未按边界的预期结束。	无法确保接收数据的完整性。 复位 DSI 主机并重复读取步骤。
1	4	CRCE	主机报告接收的长数据包的有效负载中存在 CRC 错误。	接收的有效负载数据已损坏。复位 DSI 主机并重复读取步骤。如果此错误反复发生，请检查 DSI 连接链路的噪声级别。
1	3	ECCME	主机报告接收到的数据包中包含多个 ECC 错误。	接收到的数据包已损坏。DSI 主机忽略以下所有数据包。DSI 主机应重复读取步骤。
1	2	ECCSE	主机报告接收到的数据包中包含一个位错误。	此错误并不重要，因为 DSI 主机可以纠正该错误并正确解码数据包。 如果此错误反复发生，请检查 DSI 连接链路的信号完整性和噪声级别。
1	1	TOLPRX	主机报告配置的低功耗接收超时计数器已到期。	一旦配置的超时计数器结束，DSI 主机将自动复位控制器端并恢复正常操作。在此事件期间发生的数据包发送将丢失。 如果此错误反复发生，请检查定时器配置是否存在任何问题。该定时器应大于设备生成的最大低功耗发送。
1	0	TOHSTX	主机报告配置的高速发送超时计数器已到期。	一旦配置的超时计数器结束，DSI 主机将自动复位控制器端并恢复正常操作。在此事件期间发生的数据包发送将丢失。 如果此错误反复发生，请检查定时器配置是否存在任何问题。该定时器应大于主机生成的最大高速发送突发。
DSI 封装器	10	PLLUF	D-PHY 的 PLL 已解锁。	此错误非常重要。 应重新配置并重新启动图形子系统。

29.14 编程步骤

要操作 DSI 主机，必须熟悉 MIPI® DSI 规范。每个软件可编程寄存器均可通过 APB 接口访问。

29.14.1 编程步骤概述

视频模式或适配命令模式的编程过程必须遵循以下顺序：

1. 配置 RCC（请参见 RCC 一章）
   - 使能 DSI 和 LTDC 的时钟
   - 配置 LTDC PLL，将其打开并等待其锁定
2. 可选择配置 GPIO（例如，如果撕裂效应需要使用 GPIO）
3. 可选择有效 ISR
4. 配置 LTDC（请参见 LTDC 一章）
   - 编程面板时序
   - 使能相关层
5. 打开 DSI 稳压器并等待稳压器就绪，如*第 29.12.5 节*所述
6. 配置 DSI PLL，将其打开并等待其锁定，如*第 29.12.4 节*中所述
7. 在 DSI 主机和 DSI 封装器中配置 D-PHY 参数以定义 D-PHY 配置和时序，详细信息请参见*第 29.14.2 节*
8. 配置 DSI 主机时序，如*第 29.14.3 节*中所述
9. 配置 DSI 主机流控制和 DBI 接口，如*第 29.14.4 节*中所述
10. 配置 DSI 主机 LTDC 接口，如*第 29.14.5 节*中所述
11. 配置视频模式下的 DSI 主机（如*第 29.14.6 节*中所述）或配置适配命令模式（如*第 29.14.7 节*中所述）
12. 使能 D-PHY，将 DSI_PCTLR 的 DEN 位置 1
13. 使能 D-PHY 时钟通道，将 DSI_PCTLR 的 CKEN 位置 1
14. 使能 DSI 主机，将 DSI_CR 的 EN 位置 1
15. 使能 DSI 封装器，将 DSI_WCR 的 DSIEN 位置 1
16. 可以通过 APB 通用接口发送 DCS 命令来配置显示屏
17. 在 LTDC 中使能 LTDC
18. 通过 DSI 封装器启动 LTDC 流程 (CR.LTDCEN = 1)

在视频模式下，只要使能 LTDC，数据流就会启动。

在适配命令模式下，只要 CR.LTDCEN 位置 1，就会启动帧缓冲区更新。

29.14.2 配置 D-PHY 参数

在开始任何通信之前，需要对 D-PHY 进行特定配置。配置参数存储在 DSI 主机或 DSI 封装器中。

在 DSI 封装器中配置 D-PHY 参数

DSI 封装器可用于微调 D-PHY 的时序或物理参数。标准用途的 D-PHY 不需要此操作。所有字段和参数都在 DSI 封装器的寄存器说明中进行介绍。

要正确启动 D-PHY，只有一个字段是必需的：单位间隔乘以 DSI 封装器 PHY 配置寄存器 1 (DSI_WPCR0) 的 4 (UIX4) 字段。

此字段以 0.25 ns 为单位定义了高速模式下的位周期，它被用作由 D-PHY 管理的所有时序的时基。

如果链路速度为 600 Mb/s，则单位间隔应为 1.667 ns，乘以 4 即为 6.667 ns。向下舍入时，必须在 DSI_WPCR0 寄存器的 UIX4 字段中写入值 6。

在 DSI 主机中配置 D-PHY 参数

DSI 主机存储 D-PHY 时序参数的配置和通道数量。

启动前必须配置以下字段：

- DSI_PCONFR 寄存器中的数据通道数
- DSI_CLCR 寄存器中的自动时钟通道控制 (ACR)
- DSI_CLCR 寄存器中的时钟控制 (DPCC)
- DSI_CLTCR 和 DSI_DLTCR 寄存器中时钟通道和数据通道的 LP/HS 和 HS/LP 转换时间
- DSI_PCONFR 寄存器中的停止等待时间

29.14.3 配置 DSI 主机时序

应在 DSI 主机中配置全部协议时序。

时钟分频器配置

内部生成两个时钟

- 超时时钟；
- TX 逸出时钟。

超时时钟用作 HS 到 LP 和 LP 到 HS 转换错误配置中的定时单元。其分频系数由 DSI 主机时钟控制寄存器 (DSI_CCR) 的超时时钟分频 (TOCKDIV) 字段配置。

TX 逸出时钟在低功耗模式下使用。其分频系数由 DSI 主机时钟控制寄存器 (DSI_CCR) 的 TX 逸出时钟分频 (TXECKDIV) 字段配置（相对于通道字节时钟）。典型值约为 20 MHz。

超时配置

*第 29.8 节*所述的超时管理的时序在 DSI 主机超时计数器配置寄存器（DSI_TCCR0 到 DSI_TCCR5）中配置。

29.14.4 配置流控制和 DBI 接口

借助 DSI 主机协议配置寄存器 (DSI_PCR) 来配置流控制。配置参数如下

- CRC 接收使能（CRCRXE 位）
- ECC 接收使能（ECCRXE 位）
- BTA 使能（BTAE 位）
- EoTp 接收使能（ETRXE 位）
- EoTp 发送使能（ETTXE 位）

这些参数值取决于用于与 DSI 显示屏通信的协议。

用于通用 DBI 接口的虚拟通道 ID 应由 DSI 主机通用 VCID 寄存器 (DSI_GVCIDR) 的虚拟通道 ID (VCID) 字段配置。

可在高速或低功耗模式下发送或接收所有 DCS 命令，具体取决于命令类型。对于每一个命令，在 DSI 主机命令模式配置寄存器 (DSI_CMCR) 中编程专用配置位。

还应在 DSI 主机命令模式配置寄存器 (DSI_CMCR) 中配置对数据包或撕裂效应事件的应答请求。

29.14.5 配置 DSI 主机 LTDC 接口

由于 DSI 主机通过视频模式或适配命令模式下的 LTDC 连接到系统，因此 DSI 封装器在这两种模式之间执行低级连接。

编程到 DSI 封装器中的参数必须与编程到 LTDC 和 DSI 主机中的参数一致。

必须配置以下字段：

- DSI 主机 LTDC VCID 寄存器 (DSI_LVCIDR) 的虚拟通道 ID (VCID) 字段中的虚拟通道 ID。
- DSI 主机 LTDC 颜色编码寄存器 (DSI_LCOLCR) 的颜色编码 (COLC) 字段和 DSI 封装器配置寄存器 (DSI_WCFGR) 中的颜色多路复用 (COLMUX) 字段。
- 如果松散数据包用于 18 位模式，则必须将 DSI 主机 LTDC 颜色编码寄存器 (DSI_LCOLCR) 的松散数据包使能 (LPE) 位置 1。
- DSI 主机 LTDC 极性配置寄存器 (DSI_LPCR) 的 HSync 极性 (HSP) 位中的 HSYNC 极性。
- DSI 主机 LTDC 极性配置寄存器 (DSI_LPCR) 的 VSync 极性 (VSP) 位和 DSI 封装器配置寄存器 (DSI_WCFGR) 的 VSync 极性 (VSPOL) 位中的 VSYNC 极性。
- DSI 主机 LTDC 极性配置寄存器 (DSI_LPCR) 的 DATA ENABLE 极性数据使能极性 (DEP) 位。

29.14.6 配置视频模式

视频模式配置应定义控制器在低功耗命令发送、视频传输类型（突发或非突发模式）及面板水平和垂直时序中的行为：

- 选择视频传输模式以定义处理器如何要求通过 DSI 链路传输视频行。
  - 在 DSI_VMCR 中配置低功耗转换以定义允许进入低功耗模式的视频周期——如果有时间这样做。
  - 配置控制器是否应在帧结束时请求外设应答消息 (DSI_VMCR.FBTAAE)。
  - 配置是否在低功耗模式下发送命令 (DSI_VMCR.LPE)。
- 选择视频模式类型
  - 突发模式：

    将视频模式类型 (DSI_VMCR.VMT) 配置为值 2'b1x。

    将视频数据包大小 (DSI_VPCR.VPSIZE) 配置为有效行周期的大小（以像素为单位）。

    DSI 主机会忽略寄存器 DSI_VCCR 和 DSI_VNPCR。
  - 非突发模式：

    将视频模式类型 (DSI_VMCR.VMT) 配置为 2'b00 以启用同步脉冲的发送，或者配置为 2'b01 以启用同步事件的发送。

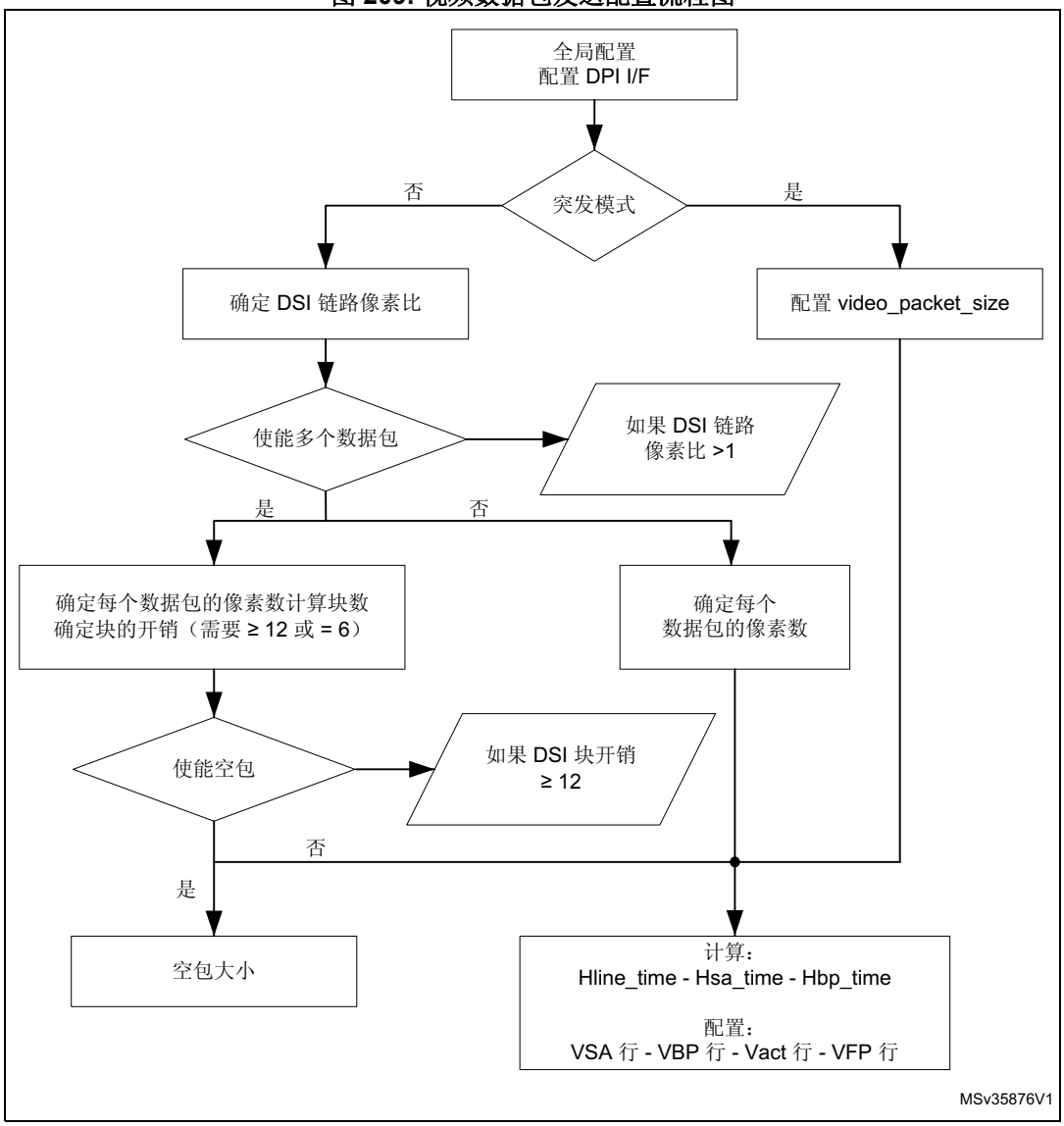
    将视频数据包大小 (DSI_VPCR.VPSIZE) 配置为要在单个数据包中发送的像素数。选择此值取决于所连接外设的可用存储器（如果首先存储数据），或者取决于要在 DSI 主机中为 FIFO 选择的存储器。

    将块数 (DSI_VCCR.NUMC) 配置为每个视频行要发送的数据包数。VPSIZE * NUMC 的值是视频每行的像素数，除非 NUMC 为 0，此时或禁止多数据包。如果将其设置为 1，则每行仍然只有一个数据包，但它可以是块的一部分，后面带有一个空包。

    将空包大小 (DSI_VNPCR.NPSIZE) 配置为作为块的一部分插入的空包的大小。将其设置为 0 将禁止空数据包。
- 按如下方式定义视频水平时序配置：
  - 将水平行时间 (DSI_VLCR.HLINE) 配置为LTDC 视频行所需的时间（以通道字节时钟周期为单位，时钟通道为 500 MHz 时，通道字节时钟周期为 8 ns）。当 LTDC 时钟和通道字节时钟周期不是倍数关系时，用于编程 DSI_VLCR.HLINE 的值需要舍入。由于配置值的舍入，将导致各行之间的时序不匹配。如果 DSI 主机配置为不进入低功耗模式，则此时序差异会在每一行上累积，从而在帧结束时引起大量不匹配。原因是 DSI 主机无法在每个新行上重新同步，因为它会在 LTDC 接口上发生水平同步事件时发送消隐数据包。但是，累积的不匹配应当在帧的最后一行消失，根据 DSI 规范，在垂直同步事件上开始新帧时，链路应始终返回到低功耗以重新获得同步。如果累计时序不匹配大于低功耗模式下最后一行的时间，则会发生故障。通过将 DSI 主机配置为每行进入低功耗模式一次，可以避免这种现象。
  - 将水平同步持续时间 (DSI_VHSACR.HSA) 配置为的 LTDC 水平同步活动周期所需的时间（以通道字节时钟周期为单位，通常周期为 8 ns）。
  - 将水平后沿持续时间 (DSI_VHBPCR.HBP) 配置为 LTDC 水平后沿周期所需的时间（以通道字节时钟周期为单位，通常周期为 8 ns）。应特别注意该参数的计算。
- 定义垂直行配置：
  - 使用 LTDC 垂直同步活动周期中现有的行数配置垂直同步持续时间 (DSI_VVSACR.VSA)。
  - 使用 LTDC 垂直后沿周期中现有的行数配置垂直后沿持续时间 (DSI_VVBPCR.VBP)。
  - 使用 LTDC 垂直前沿周期中现有的行数配置垂直前沿持续时间 (DSI_VVFPCR.VFP)。
  - 使用 LTDC 垂直激活周期中现有的行数配置垂直激活持续时间 (DSI_VVACR.VA)。

图 209 说明了配置 DPI 数据包发送的步骤。

图 209. 视频数据包发送配置流程图



视频配置示例

以下为视频数据包发送配置的示例：

视频分辨率：

- PCLK 周期 = 50 ns
- HSA = 8 PCLK
- HBP = 8 PCLK
- HACT = 480 PCLK
- HFP = 24 PCLK
- VSA = 2 行
- VBP = 2 行
- VACT = 640 行
- VFP = 4 行

配置步骤：

- 视频发送模式配置：
  a) 配置低功耗转换：
  DSI_VMCR[13:8] = 6'b111111，以在所有视频周期内使能 LP。
  b) DSI_VMCR.FBTAAE = 1，以便 DSI 主机在每帧结束时从外设请求应答响应消息。
- 要使用突发模式，请按以下步骤操作：
  DSI_VMCR.VMT = 2'b1x
  DSI_VPCR.VPSIZE = 480
- 水平时序配置：
  – DSI_VLCR.HLINE = (HSA + HBP + HACT + HFP) *（PCLK 周期/ Clk 通道字节周期）= (8 + 8 + 480 + 24) * (50 / 8) = 3250
  – DSI_VHSACR.HSA = HSA *（PCLK 周期/Clk 通道字节周期）= 8 * (50 / 8) = 50
  – DSI_VHDPCR.HBP = HBP *（PCLK 周期/Clk 通道字节周期）= 8 * (50 / 8) = 50
- 垂直行配置：
  – DSI_VVSACR.VSA = 2
  – DSI_VVBPCR.VBP = 2
  – DSI_VVFPCR.VFP = 4
  – DSI_VVACR.VA = 640

29.14.7 配置适配命令模式

适配命令模式需要配置以下参数：

- DSI 主机 LTDC 命令配置寄存器 (DSI_LCCR) 的命令大小 (CMDSIZE) 字段，用于定义写入存储器命令最大允许大小。
- DSI 封装器配置寄存器 (DSI_WCFGR) 的撕裂效应源 (TESRC) 和可选撕裂效应极性 (TEPOL) 位。
- 如果每次接收到撕裂效应事件时需要自动更新显示屏，则将 DSI 封装器配置寄存器 (DSI_WCFGR) 的自动刷新 (AR) 位置 1。

29.14.8 配置视频模式测试图发生器

DSI 主机可发送不带水平/垂直彩条的彩条模式和不带任何类型刺激的 D-PHY BER 测试模式。

图 210 显示了发送测试模式的编程顺序：

1. 配置 DSI_MCR 寄存器以使能视频模式。使用 DSI_VMCR.VMT 配置视频模式类型。
2. 配置 DSI_LCOLCR 寄存器。
3. 使用*图 211* 中所示的寄存器配置帧，其中灰色区域表示传输的像素。
4. 配置模式生成模式 (DSI_VMCR.PGM) 和模式方向 (DSI_VMCR.PGO)，并加以使能 (DSI_VMCR.PGE)。

图 210. 发送测试模式的编程顺序

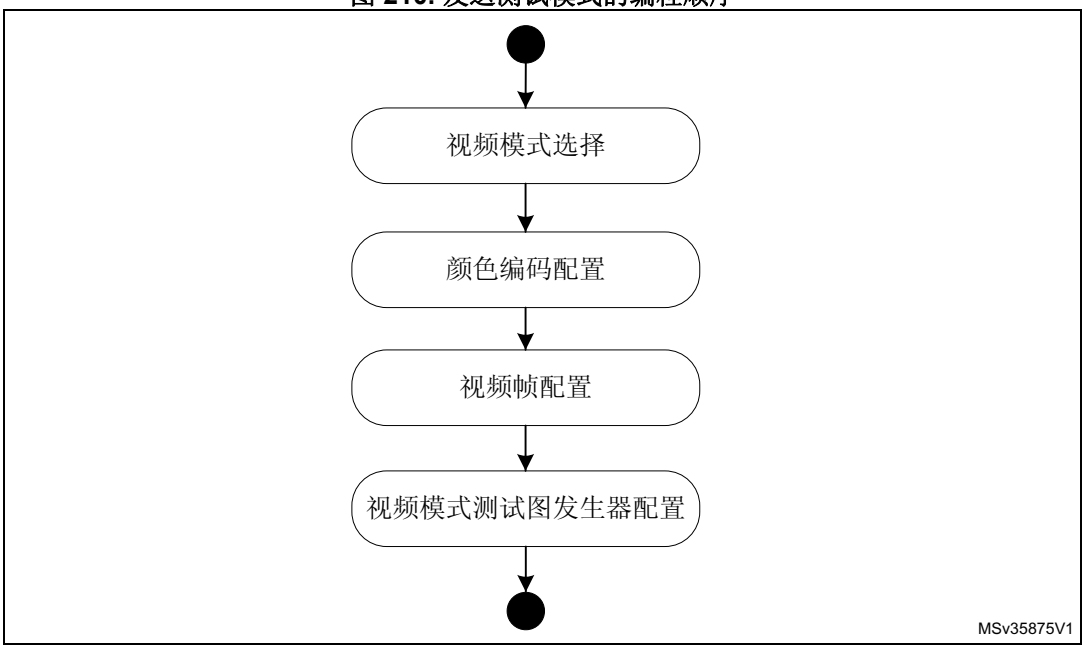
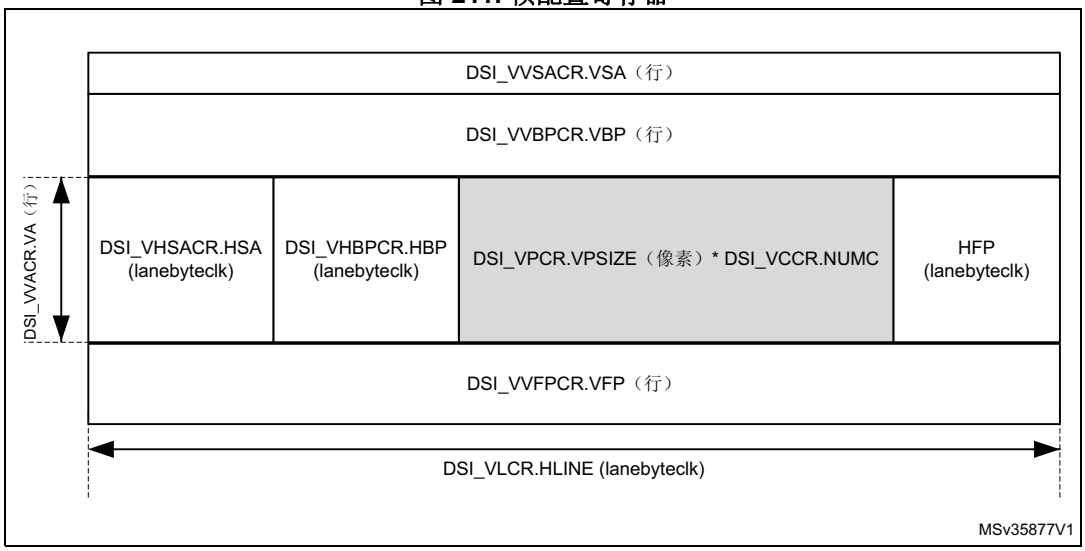


图 211. 帧配置寄存器



注： *有效负载的像素数限制为表 174 中提供的值的倍数。*

29.14.9 管理 ULPM

配置软件进入和退出 ULPM 的方式有两种：

- 在 D-PHY PLL 运行期间进入和退出 ULPM。此过程较快。
- 在 D-PHY PLL 关闭时进入和退出 ULPM。就功耗而言，此过程更高效。

ULPM 序列的时钟管理

ULPM 管理状态机使用由 D-PHY 提供的通道字节时钟。

由于 D-PHY 仅在时钟通道未处于 ULPM 状态时提供通道字节时钟，因此必须在启动 ULPM 模式进入序列之前切换 DSI 主机的通道字节时钟源。

通道字节时钟源由 RCC 控制。它可以

- 是 D-PHY 的提供通道字节时钟（适用于 ULPM 外的所有模式）
- 由系统 PLL 生成的时钟（适用于 ULPM）

进入 ULPM 的流程

执行以下步骤中详细介绍的过程，以在时钟通道和数据通道上进入 ULPM：

1. 验证 DSI 主机的初始状态：
   - DSI_PCTLR[2:1] = 2'h3
   - DSI_WRPCR.PLLEN = 1'h1 且 DSI_WRPCR.REGEN = 1'h1
   - DSI_PUCR[3:0] = 4'h0
   - DSI_PTTCR[3:0] = 4'h0
   - 验证全部有效通道是否处于停止状态且 D-PHY PLL 是否锁定：
     单通道配置：DSI_PSR[6:4] = 3'h3、DSI_PSR[1] = 1'h0 且 DSI_WISR.PLLS = 1'h1
     双通道配置：DSI_PSR[8:4] = 5'h1B、DSI_PSR[1] = 1'h0 和 DSI__WISR.PLLS = 1'h1
2. 将 RCC 中的通道字节时钟源从 D-PHY 切换到系统 PLL。
3. 设置 DSI_PUCR[3:0] = 4'h5，以在数据通道和时钟通道上进入 ULPM。

4. 等待直到 D-PHY 有效通道进入 ULPM：
   – 单通道配置：DSI_PSR[6:1] = 6'h00
   – 双通道配置：DSI_PSR[8:1] = 8'h00

   DSI 主机现在处于 ULPM 中。
5. 设置 DSI_WRPCR.PLLEN = 1'b0，关闭 D-PHY PLL。

退出 ULPM 的流程

执行以下步骤中介绍的流程，以在时钟通道和数据通道上退出 ULPM：

1. 验证全部有效通道是否在 ULPM 中：
   – 单通道配置：DSI_PSR[6:1] = 6'h00
   – 双通道配置：DSI_PSR[8:1] = 8'h00
2. 设置 DSI_WRPCR.PLLEN = 1'b1，打开 D-PHY PLL。
3. 等待直到 D-PHY PLL 锁定
   – DSI_WISR.PLLS = 1'b1
4. 在不禁止 ULPM 请求位的情况下，通过设置 DSI_PUCR[3:0] = 4'hF 将 ULPM 退出位置为有效。
5. 等待直到全部有效通道退出 ULPM：
   – 单通道配置：
     DSI_PSR[5] = 1'b1
     DSI_PSR[3] = 1'b1
   – 双通道配置：
     DSI_PSR[8] = 1'b1
     DSI_PSR[5] = 1'b1
     DSI_PSR[3] = 1'b1
6. 等待 1 ms。
7. 通过设置 DSI_PUCR [3:0] = 4'h0，将 ULPM 请求和 ULPM 退出位置为无效。
8. 将 RCC 中的通道字节时钟源从 PLL 切换到 D-PHY。
9. DSI 主机现在处于停止状态且 D-PHY PLL 已锁定：
   – 单通道配置：
     DSI_PSR[6:4] = 3'h3
     DSI_PSR[1] = 1'h0
     DSI_WRPCR.PLLEN = 1'b1
   – 双通道配置：
     DSI_PSR[8:4] = 5'h1B
     DSI_PSR[1] = 1'h0
     DSI_WRPCR.PLLEN = 1'b1

29.15 DSI 主机寄存器

29.15.1 DSI 主机版本寄存器 (DSI_VR)

DSI Host Version Register

偏移地址：0x0000

复位值：0x3133 302A

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

位 31:0 **VERSION**：DSI 主机的版本 (Version of the DSI Host)
RO 寄存器包含 DSI 主机的版本

29.15.2 DSI 主机控制寄存器 (DSI_CR)

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

位 31:1 保留，必须保持复位值

位 0 **EN**：使能 (Enable)
此位将 DSI 主机配置为上电模式或复位。
0：禁止 DSI 主机（复位时）。
1：使能 DSI 主机。

29.15.3 DSI 主机时钟控制寄存器 (DSI_CCR)

DSI HOST Clock Control 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
TOCKDIV[7:0]								TXECKDIV[7:0]							
rw								rw							

位 31:16 保留，必须保持复位值

位 15:8 **TOCKDIV**：超时时钟分频 (Timeout Clock Division)
此字段指示配置 HS 到 LP 和 LP 到 HS 转换错误时，用作定时单元的超时时钟的分频系数。

位 7:0 **TXECKDIV**：TX 逸出时钟分频 (TX Escape Clock Division)
此字段指示 TX 逸出时钟源 (lanebyteclk) 的分频系数。值 0 和 1 停止 TX_ESC 时钟生成。

29.15.4 DSI 主机 LTDC VCID 寄存器 (DSI_LVCIDR)

DSI Host LTDC VCID Register

偏移地址：0x000C

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.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	VCID	
														rw	

位 31:2 保留，必须保持复位值

位 1:0 **VCID**：虚拟通道 ID (Virtual Channel ID)
这些位配置 LTDC 接口通信的虚拟通道 ID。

29.15.5 DSI 主机 LTDC 颜色编码寄存器 (DSI_LCOLCR)

DSI Host LTDC Color Coding 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
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.	Res.	Res.	Res.	Res.	Res.	LPE	Res.	Res.	Res.	Res.	COLC[3:0]			
												rw			

位 31:9 保留，必须保持复位值

位 8 **LPE**：松散封装使能 (Loosely Packet Enable)
此位使能 18 位配置的松散封装版本。
0：禁止松散封装版本
1：使能松散封装版本

位 7:4 保留，必须保持复位值

位 3:0 **COLC**：颜色编码 (Color Coding)
此字段配置 DPI 颜色编码
0000：16 位配置 1
0001：16 位配置 2
0010：16 位配置 3
0011：18 位配置 1
0100：18 位配置 2
0101：24 位
0110-1111：保留

29.15.6 DSI 主机 LTDC 极性配置寄存器 (DSI_LPCR)

DSI Host LTDC Polarity 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
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.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	HSP	VSP	DEP
													rw	rw	rw

位 31:3 保留，必须保持复位值

位 2 **HSP**：HSYNC 极性 (HSYNC Polarity)
此位配置 HSYNC 引脚的极性。
0：HSYNC 引脚高电平有效（默认）。
1：VSYNC 引脚低电平有效。

位 1 **VSP**：VSYNC 极性 (VSYNC Polarity)
此位配置 VSYNC 引脚的极性。
0：关断引脚高电平有效（默认）。
1：关断引脚低电平有效。

位 0 **DEP**：数据使能极性 (Data Enable Polarity)
此位配置数据使能引脚的极性。
0：数据使能引脚高电平有效（默认）。
1：数据使能引脚低电平有效。

29.15.7 DSI 主机低功耗模式配置寄存器 (DSI_LPMCR)

DSI Host Low-Power mode Configuration 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
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LPSIZE[7:0]							
								rw							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	VLPSIZE[7:0]							

位 31:24 保留，必须保持复位值

位 23:16 **LPSIZE**：最大数据包大小 (Largest Packet Size)
此字段用于低功耗模式下的命令发送。它定义在 VSA、VBP 和 VFP 区域内可嵌入行中的最大数据包的大小（以字节为单位）。

位 15:8 保留，必须保持复位值

位 7:0 **VLPSIZE**：VACT 最大数据包大小 (VACT Largest Packet Size)
此字段用于低功耗模式下的命令发送。它定义在 VACT 区域内可嵌入行中的最大数据包的大小（以字节为单位）。

29.15.8 DSI 主机协议配置寄存器 (DSI_PCR)

DSI Host Protocol Configuration Register

偏移地址：0x002C

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.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CRCRXE	ECCRXE	BTAE	ETRXE	ETTXE
											rw	rw	rw	rw	rw

位 31:5 保留，必须保持复位值

位 4 **CRCRXE**：CRC 接收使能 (CRC Reception Enable)
此位使能 CRC 接收和错误报告。
0：禁止 CRC 接收。
1：使能 CRC 接收。

位 3 **ECCRXE**：ECC 接收使能 (ECC Reception Enable)
此位使能 ECC 接收、错误校正和报告。
0：禁止 ECC 接收。
1：使能 ECC 接收。

位 2 **BTAE**：总线转向使能 (Bus Turn Around Enable)
此位使能总线转向 (BTA) 请求。
0：禁止总线转向请求。
1：使能总线转向请求。

位 1 **ETRXE**：EoTp 接收使能 (EoTp Reception Enable)
此位使能 EoTp 接收。
0：禁止 EoTp 接收。
1：使能 EoTp 接收。

位 0 **ETTXE**：EoTp 发送使能 (EoTp Transmission Enable)
此位使能 EoTP 发送。
0：禁止 EoTp 发送。
1：使能 EoTp 发送。

29.15.9 DSI 主机通用 VCID 寄存器 (DSI_GVCIDR)

DSI Host Generic VCID Register

偏移地址：0x0030

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.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	VCID[1:0]	

位 31:2 保留，必须保持复位值

位 1:0 **VCID**：虚拟通道 ID (Virtual Channel ID)
此字段指示通用接口回读虚拟通道标识。

29.15.10 DSI 主机模式配置寄存器 (DSI_MCR)

DSI Host mode Configuration Register

偏移地址：0x0034

复位值：0x0000 0001

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.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CMDM
															rw

位 31:1 保留，必须保持复位值

位 0 **CMDM**：命令模式 (Command mode)
此位将 DSI 主机配置为视频模式或命令模式。
0：DSI 主机配置为视频模式。
1：DSI 主机配置为命令模式。

29.15.11 DSI 主机视频模式配置寄存器 (DSI_VMCR)

DSI Host video mode Configuration Register

偏移地址：0x0038

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.	PGO	Res.	Res.	Res.	PGM	Res.	Res.	Res.	PGE
							rw				rw				rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LPCE	FBTAAE	LPHFE	LPHBPE	LPVAE	LPVFPE	LPVBPE	LPVSAE	Res.	Res.	Res.	Res.	Res.	Res.	VMT[1:0]	
rw	rw	rw	rw	rw	rw	rw	rw							rw	

位 31:25 保留，必须保持复位值

位 24 **PGO**：模式发生器方向 (Pattern Generator Orientation)
此位配置彩条方向。
0：垂直彩条。
1：水平彩条。

位 23:21 保留，必须保持复位值

位 20 **PGM**：模式发生器模式 (Pattern Generator mode)
此位用于配置模式发生器模式。
0：彩条（水平或垂直）。
1：BER 模式（仅垂直）。

位 19:17 保留，必须保持复位值

位 16 **PGE**：模式发生器使能 (Pattern Generator Enable)
此位使能视频模式测试图发生器。
0：禁止模式发生器。
1：使能模式发生器。

位 15 **LPCE**：低功耗命令使能 (Low-Power Command Enable)
此位使能仅在低功耗模式下发送命令。
0：禁止在低功耗模式下发送命令。
1：使能在低功耗模式下发送命令。

位 14 **FBTAAE**：帧总线转向应答使能 (Frame Bus-Turn-Around Acknowledge Enable)
此位使能帧结束时的应答响应请求。
0：禁止帧结束时响应应答。
1：使能帧结束时响应应答。

位 13 **LPHFPE**：低功耗水平前沿使能 (Low-Power Horizontal Front-Porch Enable)
时序允许时，此位使能在水平前沿 (HFP) 周期内返回到低功耗状态。
0：禁止在 HFP 周期内返回到低功耗状态。
1：使能在 HFP 周期内返回到低功耗状态。

位 12 **LPHBPE**：低功耗水平后沿使能 (Low-Power Horizontal Back-Porch Enable)
时序允许时，此位使能在水平后沿 (HBP) 周期内返回到低功耗状态。
0：禁止在 HBP 周期内返回到低功耗状态。
1：使能在 HBP 周期内返回到低功耗状态。

位 11 **LPVAE**：低功耗垂直活动使能 (Low-Power Vertical Active Enable)
时序允许时，此位使能在垂直活动 (VACT) 周期内返回到低功耗状态。
0：禁止在 VACT 周期内返回到低功耗状态。
1：使能在 VACT 周期内返回到低功耗状态。

位 10 **LPVFPE**：低功耗垂直前沿使能 (Low-power Vertical Front-porch Enable)
时序允许时，此位使能在垂直前沿 (VFP) 周期内返回到低功耗状态。
0：禁止在 VFP 周期内返回到低功耗状态。
1：使能在 VFP 周期内返回到低功耗状态。

位 9 **LPVBPE**：低功耗垂直后沿使能 (Low-power Vertical Back-Porch Enable)
时序允许时，此位使能在垂直后沿 (VBP) 周期内返回到低功耗状态。
0：禁止在 VBP 周期内返回到低功耗状态。
1：使能在 VBP 周期内返回到低功耗状态。

位 8 **LPVSAE**：低功耗垂直同步活动使能 (Low-Power Vertical Sync Active Enable)
时序允许时，此位使能在垂直同步时间 (VSA) 周期内返回到低功耗状态。
0：禁止在 VSA 周期内返回到低功耗状态。
1：使能在 VSA 周期内返回到低功耗状态。

位 7:2 保留，必须保持复位值

位 1:0 **VMT**：视频模式类型 (video mode Type)
此字段配置视频模式传输类型：
00：使用同步脉冲的非突发模式。
01：使用同步事件的非突发模式。
1x：突发模式

29.15.12 DSI 主机视频数据包配置寄存器 (DSI_VPCR)

DSI Host video Packet Configuration Register

偏移地址：0x003C

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.	VPSIZE[13:0]													
		rw													

位 31:14 保留，必须保持复位值

位 13:0 **VPSIZE**：视频数据包大小 (video Packet Size)
此字段配置单个视频数据包中的像素数。
对于 18 位非松散封装数据类型，像素数必须是 4 的倍数。
对于 YCbCr 数据类型，像素数必须是 2 的倍数，如 DSI 规范中所述。

29.15.13 DSI 主机视频块配置寄存器 (DSI_VCCR)

DSI Host video Chunks Configuration Register

偏移地址：0x0040

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.	Res.	NUMC[12:0]												
			rw												

位 31:13 保留，必须保持复位值

位 12:0 **NUMC**：块数 (Number of Chunks)

此寄存器配置行周期内要发送的块数（一个块包含一个视频数据包和一个空包）。

如果此位设置为 0 或 1，则视频行以单个数据包的形式发送。

如果此位设置为 1，则数据包是块的一部分，如果 NPSIZE > 0，则后跟一个空包。否则，将使用多个块发送每个视频行。

29.15.14 DSI 主机视频空包配置寄存器 (DSI_VNPCR)

DSI Host Video Null Packet Configuration Register

偏移地址：0x0044

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.	Res.	NPSIZE[12:0]												
			rw												

位 31:13 保留，必须保持复位值

位 12:0 **NPSIZE**：空包大小 (Null Packet Size)

此字段配置空包内的字节数。

此位设置为 0 时，将禁止空包。

29.15.15 DSI 主机视频 HSA 配置寄存器 (DSI_VHSACR)

DSI Host Video HSA Configuration Register

偏移地址：0x0048

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.	Res.	Res.	HSA[11:0]											
				rw											

位 31:12 保留，必须保持复位值

位 11:0 **HSA**：水平同步活动持续时间 (Horizontal Synchronism Active duration)
此字段配置以通道字节时钟周期为单位的水平同步活动时间。

29.15.16 DSI 主机视频 HBP 配置寄存器 (DSI_VHBPCR)

DSI Host Video HBP Configuration Register

偏移地址：0x004C

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.	Res.	Res.	HBP[11:0]											
				rw											

位 31:12 保留，必须保持复位值

位 11:0 **HBP**：水平后沿持续时间 (Horizontal Back-Porch duration)
此字段配置以通道字节时钟周期为单位的水平后沿时间。

29.15.17 DSI 主机视频行配置寄存器 (DSI_VLCR)

DSI Host Video Line Configuration Register

偏移地址：0x0050

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.	HLINE[14:0]														
	rw														

位 31:15 保留，必须保持复位值

位 14:0 **HLINE**：水平行持续时间 (Horizontal Line duration)
此字段配置以通道字节时钟周期为单位的水平行总时间 (HSA+HBP+HACT+HFP)。

29.15.18 DSI 主机视频 VSA 配置寄存器 (DSI_VVSACR)

DSI Host Video VSA Configuration Register

偏移地址：0x0054

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.	Res.	Res.	Res.	Res.	VSA[9:0]									
						rw									

位 31:10 保留，必须保持复位值

位 9:0 **VSA**：垂直同步活动持续时间 (Vertical Synchronism Active duration)
此字段配置垂直同步活动周期（以水平行数为单位）。

29.15.19 DSI 主机视频 VBP 配置寄存器 (DSI_VVBPCR)

DSI Host Video VBP Configuration Register

偏移地址：0x0058

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.	Res.	Res.	Res.	Res.	VBP[9:0]									
						rw									

位 31:10 保留，必须保持复位值

位 9:0 **VBP**：垂直后沿持续时间 (Vertical Back-Porch duration)
此字段配置垂直后沿周期（以水平行数为单位）。

29.15.20 DSI 主机视频 VFP 配置寄存器 (DSI_VVFPCR)

DSI Host Video VFP Configuration Register

偏移地址：0x005C

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.	Res.	Res.	Res.	Res.	VFP[9:0]									
						rw									

位 31:10 保留，必须保持复位值

位 9:0 **VFP**：垂直前沿持续时间 (Vertical Front-Porch duration)
此字段配置垂直前沿周期（以水平行数为单位）。

29.15.21 DSI 主机视频 VA 配置寄存器 (DSI_VVACR)

DSI Host Video VA Configuration Register

偏移地址：0x0060

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.	VA[13:0]													
		rw													

位 31:14 保留，必须保持复位值

位 13:0 **VA**：垂直活动持续时间 (Vertical Active duration)
此字段配置垂直活动周期（以水平行数为单位）。

29.15.22 DSI 主机 LTDC 命令配置寄存器 (DSI_LCCR)

DSI Host LTDC Command Configuration Register

偏移地址：0x0064

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
CMDSIZE[15:0]															
rw															

位 31:16 保留，必须保持复位值

位 15:0 **CMDSIZE**：命令大小 (Command Size)
此段配置 LTDC 写入存储器命令的最大允许大小（以像素为单位）。从 LTDC 获取的数据自动分区永久使能。

29.15.23 DSI 主机命令模式配置寄存器 (DSI_CMCR)

DSI Host Command mode Configuration Register

偏移地址：0x0068

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.	MRDPS	Res.	Res.	Res.	Res.	DLWTX	DSR0TX	DSW1TX	DSW0TX
							rw					rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	GLWTX	GSR 2TX	GSR 1TX	GSR 0TX	GSW 2TX	GSW 1TX	GSW 0TX	Res.	Res.	Res.	Res.	Res.	Res.	ARE	TEARE
	rw	rw	rw	rw	rw	rw	rw							rw	rw

位 31:25 保留，必须保持复位值

位 24 **MRDPS**：最大读取数据包大小 (Maximum Read Packet Size)
此位配置最大读取数据包大小命令发送类型：
0：高速。
1：低功耗。

位 23:20 保留，必须保持复位值

位 19 **DLWTX**：DCS 长写发送 (DCS Long Write Transmission)
此位配置 DCS 长写数据包命令发送类型：
0：高速。
1：低功耗。

位 18 **DSR0TX**：DCS 短读 0 参数发送 (DCS Short Read Zero parameter Transmission)
此位配置 DCS 短读数据包和 0 参数命令发送类型：
0：高速。
1：低功耗。

位 17 **DSW1TX**：DCS 短读 1 参数发送 (DCS Short Read One parameter Transmission)
此位配置 DCS 短读数据包和 1 参数命令发送类型：
0：高速。
1：低功耗。

位 16 **DSW0TX**：DCS 短写 0 参数发送 (DCS Short Write Zero parameter Transmission)
此位配置 DCS 短写数据包和 0 参数命令发送类型：
0：高速。
1：低功耗。

位 15 保留，必须保持复位值

位 14 **GLWTX**：通用长写传输 (Generic Long Write Transmission)
此位配置通用长写数据包命令发送类型：
0：高速。
1：低功耗。

位 13 **GSR2TX**：通用短读 2 参数发送 (Generic Short Read Two parameters Transmission)
此位配置通用短读数据包和 2 参数命令发送类型：
0：高速。
1：低功耗。

位 12 **GSR1TX**：通用短读 1 参数发送 (Generic Short Read One parameters Transmission)
此位配置通用短读数据包和 1 参数命令发送类型：
0：高速。
1：低功耗。

位 11 **GSR0TX**：通用短读 0 参数发送 (Generic Short Read Zero parameters Transmission)
此位配置通用短读数据包和 0 参数命令发送类型：
0：高速。
1：低功耗。

位 10 **GSW2TX**：通用短写 2 参数发送 (Generic Short Write Two parameters Transmission)
此位配置通用短写数据包和 2 参数命令发送类型：
0：高速。
1：低功耗。

位 9 **GSW1TX**：通用短写 1 参数发送 (Generic Short Write One parameters Transmission)
此位配置通用短写数据包和 1 参数命令发送类型：
0：高速。
1：低功耗。

位 8 **GSW0TX**：通用短写 0 参数发送 (Generic Short Write Zero parameters Transmission)
此位配置通用短写数据包和 0 参数命令发送类型：
0：高速。
1：低功耗。

位 7:2 保留，必须保持复位值

位 1 **ARE**：应答请求使能 (Acknowledge Request Enable)
此位在每次数据包发送后使能应答请求：
0：禁止应答请求。
1：使能应答请求。

位 0 **TEARE**：撕裂效应应答请求使能 (Tearing Effect Acknowledge Request Enable)
此位使能撕裂效应应答请求：
0：禁止撕裂效应应答请求。
1：使能撕裂效应应答请求。

29.15.24 DSI 主机通用头配置寄存器 (DSI_GHCR)

DSI Host Generic Header Configuration Register

偏移地址：0x006C

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.	WCMSB[7:0]							
								rw							

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

位 31:24 保留，必须保持复位值

位 23:16 **WCMSB**：字计数 MSB (WordCount MSB)
此字段为长数据包配置报头数据包字的最高有效字节，或为短数据包配置数据 1。

位 15:8 **WCLSB**：字计数 LSB (WordCount LSB)
此字段为长数据包配置报头数据包字的最高有效字节，或为短数据包配置数据 1。

位 7:6 **VCID**：虚拟通道 ID
此字段配置报头数据包的虚拟通道 ID。

位 5:0 **DT**：数据类型
此字段配置报头数据包的数据包数据类型。

29.15.25 DSI 主机通用有效负载数据寄存器 (DSI_GPDR)

DSI Host Generic Payload Data Register

偏移地址：0x0070

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DATA4[7:0]								DATA3[7:0]							
rw								rw							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA2[7:0]								DATA1[7:0]							
rw								rw							

位 31:24 **DATA4**：有效负载字节 4 (Payload Byte 4)
此字段指示数据包有效负载的字节 4。

位 23:16 **DATA3**：有效负载字节 3 (Payload Byte 3)
此字段指示数据包有效负载的字节 3。

位 15:8 **DATA2**：有效负载字节 2 (Payload Byte 2)
此字段指示数据包有效负载的字节 2。

位 7:0 **DATA1**：有效负载字节 1 (Payload Byte 1)
此字段指示数据包有效负载的字节 1。

29.15.26 DSI 主机通用数据包状态寄存器 (DSI_GPSR)

DSI Host Generic Packet Status Register

偏移地址：0x0074

复位值：0x0000 0015

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.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RCB	PRDFF	PRDFE	PWRFF	PWRFE	CMDFF	CMDFE
									ro	ro	ro	ro	ro	ro	ro

位 31:7 保留，必须保持复位值

位 6 **RCB**：读取命令忙 (Read Command Busy)
发出读取命令时该位置 1，整个响应存储在 FIFO 中时该位清零：
0：无正在执行的读取命令。
1：正在执行读取命令。

位 5 **PRDFF**：有效负载读取 FIFO 已满 (Payload Read FIFO Full)
此位指示通用读取有效负载 FIFO 已满的状态。
0：读取有效负载 FIFO 未满。
1：读取有效负载 FIFO 已满。

位 4 **PRDFE**：有效负载读取 FIFO 为空 (Payload Read FIFO Empty)
此位指示通用读取有效负载 FIFO 为空的状态。
0：读取有效负载 FIFO 非空。
1：读取有效负载 FIFO 为空。

位 3 **PWRFF**：有效负载写入 FIFO 已满 (Payload Write FIFO Full)
此位指示通用写入有效负载 FIFO 已满的状态。
0：写入有效负载 FIFO 未满。
1：写入有效负载 FIFO 已满。

位 2 **PWRFE**：有效负载写入 FIFO 为空 (Payload Write FIFO Empty)
此位指示通用写入有效负载 FIFO 为空的状态。
0：写入有效负载 FIFO 非空。
1：写入有效负载 FIFO 为空。

位 1 **CMDFF**：命令 FIFO 已满 (Command FIFO Full)
此位指示通用命令 FIFO 已满的状态。
0：写入有效负载 FIFO 未满。
1：写入有效负载 FIFO 已满。

位 0 **CMDFE**：命令 FIFO 为空 (Command FIFO Empty)
此位指示通用命令 FIFO 为空的状态。
0：写入有效负载 FIFO 非空。
1：写入有效负载 FIFO 为空。

29.15.27 DSI 主机超时计数器配置寄存器 0 (DSI_TCCR0)

DSI Host Timeout Counter Configuration Register 0

偏移地址：0x0078

复位值：0x0000 0000

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

位 31:16 **HSTX_TOCNT**：高速发送超时计数器 (High-Speed Transmission Timeout Counter)
此字段配置触发高速发送超时争用检测的超时计数器（以 TOCKDIV 周期为单位）。
如果使用非突发模式，而且没有足够的时间从高速模式切换为低功耗模式并在从一行数据结束到下一行同步开始的时间段内返回，则 DSI 链路每帧返回一次低功耗状态，此时应配置 TOCKDIV 和 HSTX_TOCNT 以符合：
HSTX_TOCNT * lanebyteclkperiod * TOCKDIV >= 一次帧数据发送的时间 * (1 + 10%)
在突发模式下，RGB 像素数据包经过时间压缩，可在扫描行期间留出更多时间。因此，如果在突发模式下有足够的时间从高速模式切换为低功耗模式并在从一行数据结束到下一行同步开始的时间段内返回，则 DSI 链路可在此时间段内返回低功耗状态以节省电能。因此，配置 TOCKDIV 和 HSTX_TOCNT 以符合：
HSTX_TOCNT * lanebyteclkperiod * TOCKDIV >= 一次行数据发送的时间 * (1 + 10%)

位 15:0 **LPRX_TOCNT**：低功耗接收超时计数器 (Low-power Reception Timeout Counter)
此字段配置触发低功耗接收超时争用检测的超时计数器（以 TOCKDIV 周期为单位）。

29.15.28 DSI 主机超时计数器配置寄存器 1 (DSI_TCCR1)

DSI Host Timeout Counter Configuration Register 1

偏移地址：0x007C

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
HSRD_TOCNT[15:0]															
rw															

位 31:16 保留，必须保持复位值

位 15:0 **HSRD_TOCNT**：高速读取超时计数器 (High-Speed Read Timeout Counter)
此字段设置一个时间段，在发送高速读取操作后，DSI 主机在此时间段内使链路保持静止状态。此时间段以 lanebyteclk 周期为单位。从 D-PHY 进入停止状态且不产生中断时开始计算。

29.15.29 DSI 主机超时计数器配置寄存器 2 (DSI_TCCR2)

DSI Host Timeout Counter Configuration Register 2

偏移地址：0x0080

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
LPRD_TOCNT[15:0]															
rw															

位 31:16 保留，必须保持复位值

位 15:0 **LPRD_TOCNT[15:0]**：低功耗读取超时计数器 (Low-Power Read Timeout Counter)
此字段设置一个时间段，在发送低功耗读取操作后，DSI 主机在此时间段内使链路保持静止状态。此时间段以 lanebyteclk 周期为单位。从 D-PHY 进入停止状态且不产生中断时开始计算。

29.15.30 DSI 主机超时计数器配置寄存器 3 (DSI_TCCR3)

DSI Host Timeout Counter Configuration Register 3

偏移地址：0x0084

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.	PM	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
							rw								

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

位 31:25 保留，必须保持复位值

位 24 **PM**：Presp 模式 (Presp mode)

此位置 1 时，确保在满足以下两个条件时，由 HSWR_TOCNT 产生的外设响应超时仅对命令模式下的每个 LTDC 帧使用一次：

– dpivsync_edpiwms 已上升并下降。

– 已在命令模式下发送源自 LTDC 的数据包，且其 FIFO 再次为空。

在这种情况下，即使有来自通用接口的通信准备好发送，也不会将非 LTDC 命令请求发送到 D-PHY，同时会使 D-PHY 返回到停止状态。将非 LTDC 命令请求发送到 D-PHY 时，会激活 PRESP_TO 计数器，仅当此发送完成时，控制器才会发送其他准备就绪的通信。

位 23:16 保留，必须保持复位值

位 15:0 **HSWR_TOCNT**：高速写入超时计数器 (High-Speed Write Timeout Counter)

此字段设置一个时间段，在发送高速写入操作后，DSI 主机在此时间段使链路保持无效状态。此时间段以 lanebyteclk 周期为单位。从 D-PHY 进入停止状态且不产生中断时开始计算。

29.15.31 DSI 主机超时计数器配置寄存器 4 (DSI_TCCR4)

DSI Host Timeout Counter Configuration Register 4

偏移地址：0x0088

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
LPWR_TOCNT[15:0]															
rw															

位 31:16 保留，必须保持复位值

位 15:0 **LSWR_TOCNT**：低功耗写入超时计数器 (Low-Power Write Timeout Counter)
此字段设置一个时间段，在发送低功耗写入操作后，DSI 主机在此时间段内使链路保持静止状态。此时间段以 lanebyteclk 周期为单位。从 D-PHY 进入停止状态且不产生中断时开始计算。

29.15.32 DSI 主机超时计数器配置寄存器 5 (DSI_TCCR5)

DSI Host Timeout Counter Configuration Register 5

偏移地址：0x008C

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
BTA_TOCNT[15:0]															
rw															

位 31:16 保留，必须保持复位值

位 15:0 **BTA_TOCNT**：总线转向超时计数器 (Bus-Turn-Around Timeout Counter)
此字段设置一个时间段，在完成总线转向后，DSI 主机在此时间段内使链路保持静止状态。此时间段以 lanebyteclk 周期为单位。从 D-PHY 进入停止状态且不产生中断时开始计算。

29.15.33 DSI 主机时钟通道配置寄存器 (DSI_CLCR)

DSI Host Clock Lane Configuration Register

偏移地址：0x0094

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.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ACR	DPCC
														rw	rw

位 31:2 保留，必须保持复位值

位 1 **ACR**：自动时钟通道控制 (Automatic Clock lane Control)
时间允许时，此位使能自动机制，以停止在时钟通道中提供时钟。
0：禁止自动时钟通道控制
1：使能自动时钟通道控制

位 0 **DPCC**：D-PHY 时钟控制 (D-PHY Clock Control)
此位控制 D-PHY 时钟状态：
0：时钟通道处于低功耗模式
1：时钟通道在高速模式下运行

29.15.34 DSI 主机时钟通道计时器配置寄存器 (DSI_CLTCR)

DSI Host Clock Lane Timer Configuration Register

偏移地址：0x0098

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.	HS2LP_TIME[9:0]									
						rw									

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	LP2HS_TIME[9:0]									
						rw									

位 31:26 保留，必须保持复位值

位 25:16 **HS2LP_TIME**：高速到低功耗时间 (High-Speed to Low-Power Time)
此字段配置 D-PHY 时钟通道从高速转换为低功耗模式所需的最长时间（以通道字节周期为单位）。

位 15:10 保留，必须保持复位值

位 9:0 **LP2HS_TIME**：低功耗到高速时间 (Low-Power to High-Speed Time)
此字段配置 D-PHY 时钟通道从低功耗转换为高速发送模式所需的最长时间（以通道字节周期为单位）。

29.15.35 DSI 主机数据通道计时器配置寄存器 (DSI_DLTCR)

DSI Host Data Lane Timer Configuration Register

偏移地址：0x009C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
HS2LP_TIME[7:0]								LP2HS_TIME[7:0]							
rw								rw							

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

位 31:24 **HS2LP_TIME**：高速到低功耗时间 (High-Speed To Low-Power Time)
此字段配置 D-PHY 数据通道从高速转换为低功耗发送模式所需的最长时间（以通道字节时钟周期为单位）。

位 23:16 **LP2HS_TIME**：低功耗到高速时间 (Low-Power To High-Speed Time)
此字段配置 D-PHY 数据通道从低功耗到高速模式转换所需的最长时间（以通道字节时钟周期为单位）。

位 15 保留，必须保持复位值

位 14:0 **MRD_TIME**：最长读取时间 (Maximum Read Time)
此字段配置执行读取命令所需的最长时间（以通道字节周期为单位）。只有不执行读取命令时才能修改此寄存器。

29.15.36 DSI 主机 PHY 控制寄存器 (DSI_PCTLR)

DSI Host PHY Control Register

偏移地址：0x00A0

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.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CKE	DEN	Res.
													rw	rw	

位 31:3 保留，必须保持复位值

位 2 **CKE**：时钟使能 (Clock Enable)
此位使能 D-PHY 时钟通道模块：
0：禁止 D-PHY 时钟通道模块。
1：使能 D-PHY 时钟通道模块。

位 1 **DEN**：数字使能 (Digital Enable)
此位设置为 0 时，D-PHY 的数字部分处于复位状态
0：D-PHY 的数字部分处于复位状态。
1：使能 D-PHY 的数字部分。

位 0 保留

29.15.37 DSI 主机 PHY 配置寄存器 (DSI_PCONFR)

DSI Host PHY Configuration Register

偏移地址：0x00A4

复位值：0x0000 0001

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
SW_TIME[7:0]								Res.	Res.	Res.	Res.	Res.	Res.	NL[1:0]	
rw														rw	

位 31:16 保留，必须保持复位值

位 15:8 **SW_TIME**：停止等待时间 (Stop Wait Time)
此字段配置停止状态后请求高速发送的最小等待周期。

位 7:2 保留，必须保持复位值

位 1:0 **NL**：通道数 (Number of Lanes)
此字段配置有效数据通道数。
00：一个数据通道（通道 0）
01：两个数据通道（通道 0 和 1）——复位值
10：保留
11：保留

29.15.38 DSI 主机 PHY ULPS 控制寄存器 (DSI_PUCR)

DSI Host PHY ULPS Control Register

偏移地址：0x00A8

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.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	UEDL	URDL	UECL	URCL
												rw	rw	rw	rw

位 31:4 保留，必须保持复位值

位 3 **UEDL**：在数据通道上退出 ULPS (ULPS Exit on Data Lane)
在全部有效数据通道上退出 ULPS 模式。
0：无退出请求
1：在全部有效数据通道 URDL 上退出 ULPS 模式

位 2 **URDL**：在数据通道上请求 ULPS (ULPS Request on Data Lane)
在全部有效数据通道上请求 ULPS 模式。
0：无 ULPS 请求
1：在全部有效数据通道 UECL 上请求 ULPS 模式

位 1 **UECL**：在时钟通道上退出 ULPS (ULPS Exit on Clock Lane)
在时钟通道上退出 ULPS 模式。
0：无退出请求
1：在时钟通道上退出 ULPS 模式

位 0 **URCL**：在时钟通道上请求 ULPS (ULPS Request on Clock Lane)
在时钟通道上请求 ULPS 模式。
0：无 ULPS 请求
1：在时钟通道上请求 ULPS 模式

29.15.39 DSI 主机 PHY TX 触发配置寄存器 (DSI_PTTCR)

DSI Host PHY TX Triggers Configuration Register

偏移地址：0x00AC

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.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TX_TRIG[3:0]			
												rw			

位 31:4 保留，必须保持复位值

位 3:0 **TX_TRIG**：发送触发 (Transmission Trigger)
逸出模式发送触发 0-3。
在任何给定时间，只有一个 TX_TRIG 位置 1。

29.15.40 DSI 主机 PHY 状态寄存器 (DSI_PSR)

DSI Host PHY Status Register

偏移地址：0x00B0

复位值：0x0000 1528

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.	Res.	Res.	Res.	Res.	Res.	UAN1	PSS1	RUE0	UAN0	PSS0	UANC	PSSC	PD	Res.
							ro	ro	ro	ro	ro	ro	ro	ro	

位 31:9 保留，必须保持复位值

位 8 **UAN1**：ULPS 未激活通道 1 (ULPS Active Not lane 1)
此位指示 ulpsactivenot1lane D-PHY 信号的状态。

位 7 **PSS1**：PHY 停止状态通道 1 (PHY Stop State lane 1)
此位指示 phystopstate1lane D-PHY 信号的状态。

位 6 **RUE0**：RX ULPS 逸出通道 0 (RX ULPS Escape lane 0)
此位指示 rxulpsesc0lane D-PHY 信号的状态。

位 5 **UAN0**：ULPS 未激活通道 1 (ULPS Active Not lane 1)
此位指示 ulpsactivenot0lane D-PHY 信号的状态。

位 4 **PSS0**：PHY 停止状态通道 0 (PHY Stop State lane 1)
此位指示 phystopstate0lane D-PHY 信号的状态。

位 3 **UANC**：ULPS 未激活时钟通道 (ULPS Active Not Clock lane)
此位指示 ulpsactivenotclklane D-PHY 信号的状态。

位 2 **PSSC**：PHY 停止状态时钟通道 (PHY Stop State Clock lane)
此位指示 phystopstateclklane D-PHY 信号的状态。

位 1 **PD**：PHY 方向 (PHY Direction)
此位指示 phydirection D-PHY 信号的状态。

位 0 保留

29.15.41 DSI 主机中断和状态寄存器 0 (DSI_ISR0)

DSI Host Interrupt & Status Register 0

偏移地址：0x00BC

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.	PE4	PE3	PE2	PE1	PE0
											r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AE15	AE14	AE13	AE12	AE11	AE10	AE9	AE8	AE7	AE6	AE5	AE4	AE3	AE2	AE1	AE0
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:21 保留，必须保持复位值

位 20 **PE4**：PHY 错误 4 (PHY Error 4)
此位指示通道 0 的 LP1 争用错误 ErrContentionLP1。

位 19 **PE3**：PHY 错误 3 (PHY Error 3)
此位指示通道 0 的 LP0 争用错误 ErrContentionLP0。

位 18 **PE2**：PHY 错误 2 (PHY Error 2)
此位指示通道 0 的 ErrControl 错误。

位 17 **PE1**：PHY 错误 1 (PHY Error 1)
此位指示通道 0 的 ErrSyncEsc 低功耗传输同步错误。

位 16 **PE0**：PHY 错误 0 (PHY Error 0)
此位指示通道 0 的 ErrEsc 逸出进入错误。

位 15 **AE15**：应答错误 15 (Acknowledge Error 15)
此位从应答错误报告中检索 DSI 协议违例。

位 14 **AE14**：应答错误 14 (Acknowledge Error 14)
此位从应答错误报告中检索保留行为（特定于设备）。

位 13 **AE13**：应答错误 13 (Acknowledge Error 13)
此位从应答错误报告中检索无效发送长度。

位 12 **AE12**：应答错误 12 (Acknowledge Error 12)
此位从应答错误报告中检索 DSI VC ID 无效。

位 11 **AE11**：应答错误 11 (Acknowledge Error 11)
此位从应答错误报告中检索未识别的 DSI 数据类型。

位 10 **AE10**：应答错误 10 (Acknowledge Error 10)
此位从应答错误报告中检索校验和错误（仅长数据包）。

位 9 **AE9**：应答错误 9 (Acknowledge Error 9)
此位从应答错误报告中检索多比特 ECC 错误（已检测但未纠正）。

位 8 **AE8**：应答错误 8 (Acknowledge Error 8)
此位从应答错误报告中检索单比特 ECC 错误（已检测且已纠正）。

位 7 **AE7**：应答错误 7 (Acknowledge Error 7)
此位从应答错误报告中检索保留行为（特定于设备）。

位 6 **AE6**：应答错误 6 (Acknowledge Error 6)
此位从应答错误报告中检索故障控制错误。

位 5 **AE5**：应答错误 5 (Acknowledge Error 5)
此位从应答错误报告中检索外设超时错误。

位 4 **AE4**：应答错误 4 (Acknowledge Error 4)
此位从应答错误报告中检索 LP 同步传输错误。

位 3 **AE3**：应答错误 3 (Acknowledge Error 3)
此位从应答错误报告中检索逸出模式进入命令错误。

位 2 **AE2**：应答错误 2 (Acknowledge Error 2)
此位从应答错误报告中检索 EoT 同步错误。

位 1 **AE1**：应答错误 1 (Acknowledge Error 1)
此位从应答错误报告中检索 SoT 同步错误。

位 0 **AE0**：应答错误 0 (Acknowledge Error 0)
此位从应答错误报告中检索 SoT 错误。

29.15.42 DSI 主机中断和状态寄存器 1 (DSI_ISR1)

DSI Host Interrupt & Status Register 1

偏移地址：0x00C0

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.	Res.	GPRXE	GPRDE	GPTXE	GPWRE	GCWRE	LPWRE	EOTPE	PSE	CRCE	ECCME	ECCSE	TOLPRX	TOHSTX
			r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:13 保留，必须保持复位值

位 12 **GPRXE**：通用有效负载接收错误 (Generic Payload Receive Error)
此位指示在通用接口数据包读回期间，有效负载 FIFO 变满，且接收到的数据损坏。

位 11 **GPRDE**：通用有效负载读取错误 (Generic Payload Read Error)
此位指示 DCS 读取数据期间，有效负载 FIFO 变空，且发送到接口的数据损坏。

位 10 **GPTXE**：通用有效负载发送错误 (Generic Payload Transmit Error)
此位指示在通用接口数据包构建期间，有效负载 FIFO 变空，且发送损坏的数据。

位 9 **GPWRE**：通用有效负载写入错误 (Generic Payload Write Error)
此位指示系统尝试通过通用接口写入有效负载数据，且 FIFO 已满。因此，有效负载未写入。

位 8 **GCWRE**：通用命令写入错误 (Generic Command Write Error)
此位指示系统尝试通过通用接口写入命令，且 FIFO 已满。因此，命令未写入。

位 7 **LPWRE**：LTDC 有效负载写入错误 (LTDC Payload Write Error)
此位指示在 DPI 像素行存储期间，有效负载 FIFO 变满，且存储的数据损坏。

位 6 **EOTPE**：EoTp 错误 (EoTp Error)
此位指示传入外设发送结束时未接收到 EoTp 数据包。

位 5 **PSE**：数据包大小错误 (Packet Size Error)
此位指示数据包接收期间检测到数据包大小错误。

位 4 **CRCE**：CRC 错误 (CRC Error)
此位指示在接收的数据包有效负载中检测到 CRC 错误。

位 3 **ECCME**：ECC 多比特错误 (ECC Multi-bit Error)
此位指示在接收的数据包中检测到 ECC 多比特错误。

位 2 **ECCSE**：ECC 单比特错误 (ECC Single-bit Error)
此位指示在接收的数据包中检测到 ECC 单比特错误并加以纠正。

位 1 **TOLPRX**：超时低功耗接收 (Timeout Low-Power Reception)
此位指示低功耗接收超时计数器结束并检测到争用。

位 0 **TOHSTX**：超时高速发送 (Timeout High-Speed Transmission)
此位指示高速发送超时计数器结束并检测到争用。

29.15.43 DSI 主机中断使能寄存器 0 (DSI_IER0)

DSI Host Interrupt Enable Register 0

偏移地址：0x00C4

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.	PE4IE	PE3IE	PE2IE	PE1IE	PE0IE
											rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AE15IE	AE14IE	AE13IE	AE12IE	AE11IE	AE10IE	AE9IE	AE8IE	AE7IE	AE6IE	AE5IE	AE4IE	AE3IE	AE2IE	AE1IE	AE0IE
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:21 保留，必须保持复位值

位 20 **PE4IE**：PHY 错误 4 中断使能 (PHY Error 4 Interrupt Enable)
此位使能发生 PHY 错误 4 时产生中断。
0：禁止发生 PHY 错误 4 时产生中断。
1：使能发生 PHY 错误 4 时产生中断。

位 19 **PE3IE**：PHY 错误 3 中断使能 (PHY Error 3 Interrupt Enable)
此位使能发生 PHY 错误 3 时产生中断。
0：禁止发生 PHY 错误 3 时产生中断。
1：使能发生 PHY 错误 3 时产生中断。

位 18 **PE2IE**：PHY 错误 2 中断使能 (PHY Error 2 Interrupt Enable)
此位使能发生 PHY 错误 2 时产生中断。
0：禁止发生 PHY 错误 2 时产生中断。
1：使能发生 PHY 错误 2 时产生中断。

位 17 **PE1IE**：PHY 错误 1 中断使能 (PHY Error 1 Interrupt Enable)
此位使能发生 PHY 错误 1 时产生中断。
0：禁止发生 PHY 错误 1 时产生中断。
1：使能发生 PHY 错误 1 时产生中断。

位 16 **PE0IE**：PHY 错误 0 中断使能 (PHY Error 0 Interrupt Enable)
此位使能发生 PHY 错误 0 时产生中断。
0：禁止发生 PHY 错误 0 时产生中断。
1：使能发生 PHY 错误 0 时产生中断。

位 15 **AE15IE**：应答错误 15 中断使能 (Acknowledge Error 15 Interrupt Enable)
此位使能发生应答错误 15 时产生中断。
0：禁止发生应答错误 15 时产生中断。
1：使能发生应答错误 15 时产生中断。

位 14 **AE14IE**：应答错误 14 中断使能 (Acknowledge Error 14 Interrupt Enable)
此位使能发生应答错误 14 时产生中断。
0：禁止发生应答错误 14 时产生中断。
1：使能发生应答错误 14 时产生中断。

位 13 **AE13IE**：应答错误 13 中断使能 (Acknowledge Error 13 Interrupt Enable)
此位使能发生应答错误 13 时产生中断。
0：禁止发生应答错误 13 时产生中断。
1：使能发生应答错误 13 时产生中断。

位 12 **AE12IE**：应答错误 12 中断使能 (Acknowledge Error 12 Interrupt Enable)
此位使能发生应答错误 12 时产生中断。
0：禁止发生应答错误 12 时产生中断。
1：使能发生应答错误 12 时产生中断。

位 11 **AE11IE**：应答错误 11 中断使能 (Acknowledge Error 11 Interrupt Enable)
此位使能发生应答错误 11 时产生中断。
0：禁止发生应答错误 11 时产生中断。
1：使能发生应答错误 11 时产生中断。

位 10 **AE10IE**：应答错误 10 中断使能 (Acknowledge Error 10 Interrupt Enable)
此位使能发生应答错误 10 时产生中断。
0：禁止发生应答错误 10 时产生中断。
1：使能发生应答错误 10 时产生中断。

位 9 **AE9IE**：应答错误 9 中断使能 (Acknowledge Error 9 Interrupt Enable)
此位使能发生应答错误 9 时产生中断。
0：禁止发生应答错误 9 时产生中断。
1：使能发生应答错误 9 时产生中断。

位 8 **AE8IE**：应答错误 8 中断使能 (Acknowledge Error 8 Interrupt Enable)
此位使能发生应答错误 8 时产生中断。
0：禁止发生应答错误 8 时产生中断。
1：使能发生应答错误 8 时产生中断。

位 7 **AE7IE**：应答错误 7 中断使能 (Acknowledge Error 7 Interrupt Enable)
此位使能发生应答错误 7 时产生中断。
0：禁止发生应答错误 7 时产生中断。
1：使能发生应答错误 7 时产生中断。

位 6 **AE6IE**：应答错误 6 中断使能 (Acknowledge Error 6 Interrupt Enable)
此位使能发生应答错误 6 时产生中断。
0：禁止发生应答错误 6 时产生中断。
1：使能发生应答错误 6 时产生中断。

位 5 **AE5IE**：应答错误 5 中断使能 (Acknowledge Error 5 Interrupt Enable)
此位使能发生应答错误 5 时产生中断。
0：禁止发生应答错误 5 时产生中断。
1：使能发生应答错误 5 时产生中断。

位 4 **AE4IE**：应答错误 4 中断使能 (Acknowledge Error 4 Interrupt Enable)
此位使能发生应答错误 4 时产生中断。
0：禁止发生应答错误 4 时产生中断。
1：使能发生应答错误 4 时产生中断。

位 3 **AE3IE**：应答错误 3 中断使能 (Acknowledge Error 3 Interrupt Enable)
此位使能发生应答错误 3 时产生中断。
0：禁止发生应答错误 3 时产生中断。
1：使能发生应答错误 3 时产生中断。

位 2 **AE2IE**：应答错误 2 中断使能 (Acknowledge Error 2 Interrupt Enable)
此位使能发生应答错误 2 时产生中断。
0：禁止发生应答错误 2 时产生中断。
1：使能发生应答错误 2 时产生中断。

位 1 **AE1IE**：应答错误 1 中断使能 (Acknowledge Error 1 Interrupt Enable)
此位使能发生应答错误 1 时产生中断。
0：禁止发生应答错误 1 时产生中断。
1：使能发生应答错误 1 时产生中断。

位 0 **AE0IE**：应答错误 0 中断使能 (Acknowledge Error 0 Interrupt Enable)
此位使能发生应答错误 0 时产生中断。
0：禁止发生应答错误 0 时产生中断。
1：使能发生应答错误 0 时产生中断。

29.15.44 DSI 主机中断使能寄存器 1 (DSI_IER1)

DSI Host Interrupt Enable Register 1

偏移地址：0x00C8

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.	Res.	GPRXEIE	GPRDEIE	GPTXEIE	GPWREIE	GCWREIE	LPWREIE	EOTPEIE	PSEIE	CRCEIE	ECCMEIE	ECCSEIE	TOLPRXIE	TOHSTXIE
			rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:13 保留，必须保持复位值

位 12 **GPRXEIE**：通用有效负载接收错误中断使能 (Generic Payload Receive Error Interrupt Enable)
此位使能发生通用有效负载接收错误时产生中断。
0：禁止发生通用有效负载接收错误时产生中断。
1：使能发生通用有效负载接收错误时产生中断。

位 11 **GPRDEIE**：通用有效负载读取错误中断使能 (Generic Payload Read Error Interrupt Enable)
此位使能发生通用有效负载读取错误时产生中断。
0：禁止发生通用有效负载读取错误时产生中断。
1：使能发生通用有效负载读取错误时产生中断。

位 10 **GPTXEIE**：通用有效负载发送错误中断使能 (Generic Payload Transmit Error Interrupt Enable)
此位使能发生通用有效负载发送错误时产生中断。
0：禁止发生通用有效负载发送错误时产生中断。
1：使能发生通用有效负载发送错误时产生中断。

位 9 **GPWREIE**：通用有效负载写入错误中断使能 (Generic Payload Write Error Interrupt Enable)
此位使能发生通用有效负载写入错误时产生中断。
0：禁止发生通用有效负载写入错误时产生中断。
1：使能发生通用有效负载写入错误时产生中断。

位 8 **GCWREIE**：通用命令写入错误中断使能 (Generic Command Write Error Interrupt Enable)
此位使能发生通用命令写入错误时产生中断。
0：禁止发生通用命令写入错误时产生中断。
1：使能发生通用命令写入错误时产生中断。

位 7 **LPWREIE**：LTDC 有效负载写入错误中断使能 (LTDC Payload Write Error Interrupt Enable)
此位使能发生 LTDC 有效负载写入错误时产生中断。
0：禁止发生 LTDC 有效负载写入错误时产生中断。
1：使能发生 LTDC 有效负载写入错误时产生中断。

位 6 **EOTPEIE**：EoTp 错误中断使能 (EoTp Error Interrupt Enable)
此位使能发生 EoTp 错误时产生中断。
0：禁止发生 EoTp 错误时产生中断。
1：使能发生 EoTp 错误时产生中断。

位 5 **PSEIE**：数据包大小错误中断使能 (Packet Size Error Interrupt Enable)
此位使能发生数据包大小错误时产生中断。
0：禁止发生数据包大小错误时产生中断。
1：使能发生数据包大小错误时产生中断。

位 4 **CRCEIE**：CRC 错误中断使能 (CRC Error Interrupt Enable)
此位使能发生 CRC 错误时产生中断。
0：禁止发生 CRC 错误时产生中断。
1：使能发生 CRC 错误时产生中断。

位 3 **ECCMEIE**：ECC 多比特错误中断使能 (ECC Multi-bit Error Interrupt Enable)
此位使能发生 ECC 多比特错误时产生中断。
0：禁止发生 ECC 多比特错误时产生中断。
1：使能发生 ECC 多比特错误时产生中断。

位 2 **ECCSEIE**：ECC 单比特错误中断使能 (ECC Single-bit Error Interrupt Enable)
此位使能发生 ECC 单比特错误时产生中断。
0：禁止发生 ECC 单比特错误时产生中断。
1：使能发生 ECC 单比特错误时产生中断。

位 1 **TOLPRXIE**：超时低功耗接收中断使能 (Timeout Low-Power Reception Interrupt Enable)
此位使能在超时低功耗接收时产生中断。
0：禁止在超时低功耗接收时产生中断。
1：使能在超时低功耗接收时产生中断。

位 0 **TOHSTXIE**：超时高速传输中断使能 (Timeout High-Speed Transmission Interrupt Enable)
此位使能在超时高速传输时产生中断。
0：禁止在超时高速传输时产生中断。
1：使能在超时高速传输时产生中断。

29.15.45 DSI 主机强制中断寄存器 0 (DSI_FIR0)

DSI Host Force Interrupt Register 0

偏移地址：0x00D8

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.	FPE4	FPE3	FPE2	FPE1	FPE0
											wo	wo	wo	wo	wo
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FAE15	FAE14	FAE13	FAE12	FAE11	FAE10	FAE9	FAE8	FAE7	FAE6	FAE5	FAE4	FAE3	FAE2	FAE1	FAE0
wo	wo	wo	wo	wo	wo	wo	wo	wo	wo	wo	wo	wo	wo	wo	wo

位 31:21 保留，必须保持复位值

位 20 **FPE4**：强制 PHY 错误 4 (Force PHY Error 4)
向此位写入 1 会强制发生 PHY 错误 4。

位 19 **FPE3**：强制 PHY 错误 3 (Force PHY Error 3)
向此位写入 1 会强制发生 PHY 错误 3。

位 18 **FPE2**：强制 PHY 错误 2 (Force PHY Error 2)
向此位写入 1 会强制发生 PHY 错误 2。

位 17 **FPE1**：强制 PHY 错误 1 (Force PHY Error 1)
向此位写入 1 会强制发生 PHY 错误 1。

位 16 **FPE0**：强制 PHY 错误 0 (Force PHY Error 0)
向此位写入 1 会强制发生 PHY 错误 0。

位 15 **FAE15**：强制应答错误 15 (Force Acknowledge Error 15)
向此位写入 1 会强制发生应答错误 15。

位 14 **FAE14**：强制应答错误 14 (Force Acknowledge Error 14)
向此位写入 1 会强制发生应答错误 14。

位 13 **FAE13**：强制应答错误 13 (Force Acknowledge Error 13)
向此位写入 1 会强制发生应答错误 13。

位 12 **FAE12**：强制应答错误 12 (Force Acknowledge Error 12)
向此位写入 1 会强制发生应答错误 12。

位 11 **FAE11**：强制应答错误 11 (Force Acknowledge Error 11)
向此位写入 1 会强制发生应答错误 11。

位 10 **FAE10**：强制应答错误 10 (Force Acknowledge Error 10)
向此位写入 1 会强制发生应答错误 10。

位 9 **FAE9**：强制应答错误 9 (Force Acknowledge Error 9)
向此位写入 1 会强制发生应答错误 9。

位 8 **FAE8**：强制应答错误 8 (Force Acknowledge Error 8)
向此位写入 1 会强制发生应答错误 8。

位 7 **FAE7**：强制应答错误 7 (Force Acknowledge Error 7)
向此位写入 1 会强制发生应答错误 7。

位 6 **FAE6**：强制应答错误 6 (Force Acknowledge Error 6)
向此位写入 1 会强制发生应答错误 6。

位 5 **FAE5**：强制应答错误 5 (Force Acknowledge Error 5)
向此位写入 1 会强制发生应答错误 5。

位 4 **FAE4**：强制应答错误 4 (Force Acknowledge Error 4)
向此位写入 1 会强制发生应答错误 4。

位 3 **FAE3**：强制应答错误 3 (Force Acknowledge Error 3)
向此位写入 1 会强制发生应答错误 3。

位 2 **FAE2**：强制应答错误 2 (Force Acknowledge Error 2)
向此位写入 1 会强制发生应答错误 2。

位 1 **FAE1**：强制应答错误 1 (Force Acknowledge Error 1)
向此位写入 1 会强制发生应答错误 1。

位 0 **FAE0**：强制应答错误 0 (Force Acknowledge Error 0)
向此位写入 1 会强制发生应答错误 0。

29.15.46 DSI 主机强制中断寄存器 1 (DSI_FIR1)

DSI Host Force Interrupt Register 1

偏移地址：0x00DC

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.	Res.	FGP RXE	FGP RDE	FGP TXE	FGP WRE	FGC WRE	FLP WRE	FE OTPE	FPSE	FCRCE	FECC ME	FECC SE	FTOLP RX	FTOHS TX
			wo	wo	wo	wo	wo	wo	wo	wo	wo	wo	wo	wo	wo

位 31:13 保留，必须保持复位值

位 12 **FGPRXE**：强制通用有效负载接收错误 (Force Generic Payload Receive Error)
向此位写入 1 会强制发生通用有效负载接收错误。

位 11 **FGPRDE**：强制通用有效负载读取错误 (Force Generic Payload Read Error)
向此位写入 1 会强制发生通用有效负载读取错误。

位 10 **FGPTXE**：强制通用有效负载发送错误 (Force Generic Payload Transmit Error)
向此位写入 1 会强制发生通用有效负载发送错误。

位 9 **FGPWRE**：强制通用有效负载写入错误 (Force Generic Payload Write Error)
向此位写入 1 会强制发生通用有效负载写入错误。

位 8 **FGCWRE**：强制通用命令写入错误 (Force Generic Command Write Error)
向此位写入 1 会强制发生通用命令写入错误。

位 7 **FLPWRE**：强制 LTDC 有效负载写入错误 (Force LTDC Payload Write Error)
向此位写入 1 会强制发生 LTDC 有效负载写入错误。

位 6 **FEOTPE**：强制 EoTp 错误 (Force EoTp Error)
向此位写入 1 会强制发生 EoTp 错误。

位 5 **FPSE**：强制数据包大小错误 (Force Packet Size Error)
向此位写入 1 会强制发生数据包大小错误。

位 4 **FCRCE**：强制 CRC 错误 (Force CRC Error)
向此位写入 1 会强制发生 CRC 错误。

位 3 **FECCME**：强制 ECC 多比特错误 (Force ECC Multi-bit Error)
向此位写入 1 会强制发生 ECC 多比特错误。

位 2 **FECCSE**：强制 ECC 单比特错误 (Force ECC Single-bit Error)
向此位写入 1 会强制发生 ECC 单比特错误。

位 1 **FTOLPRX**：强制超时低功耗接收 (Force Timeout Low-Power Reception)
向此位写入 1 会强制发生超时低功耗接收。

位 0 **FTOHSTX**：强制超时高速发送 (Force Timeout High-Speed Transmission)
向此位写入 1 会强制发生超时高速发送。

29.15.47 DSI 主机视频影子控制寄存器 (DSI_VSCR)

DSI Host Video Shadow Control Register

偏移地址：0x0100

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.	Res.	Res.	Res.	Res.	Res.	UR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	EN
							rw								rw

位 31:9 保留，必须保持复位值

位 8 **UR**：更新寄存器 (Update Register)
此位置 1 时，LTDC 寄存器会复制到辅助寄存器。复制后，此位自动清零。
0：未请求更新。
1：已请求寄存器更新。

位 7:1 保留，必须保持复位值

位 0 **EN**：使能 (Enable)
此位置 1 时，DSI 主机 LTDC 接口从辅助寄存器接收有效配置。
此位与 UR 位同时置 1 时，辅助寄存器自动更新。
0：禁止寄存器更新。
1：使能寄存器更新。

29.15.48 DSI 主机 LTDC 当前 VCID 寄存器 (DSI_LCVCIDR)

DSI Host LTDC Current VCID Register

偏移地址：0x010C

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.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	VCID[1:0]	
														ro	

位 31:2 保留，必须保持复位值

位 1:0 **VCID**：虚拟通道 ID (Virtual Channel ID)
此位返回 LTDC 接口的虚拟通道 ID。

29.15.49 DSI 主机 LTDC 当前颜色编码寄存器 (DSI_LCCCR)

DSI Host LTDC Current Color Coding Register

偏移地址：0x0110

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.	Res.	Res.	Res.	Res.	Res.	LPE	Res.	Res.	Res.	Res.	COLC[3:0]			
							ro					ro			

位 31:9 保留，必须保持复位值

位 8 **LPE**：松散封装使能 (Loosely Packed Enable)
此位返回 18 位配置的松散封装版本的当前状态。
0：禁止松散封装版本。
1：使能松散封装版本。

位 7:4 保留，必须保持复位值

位 3:0 **COLC**：颜色编码 (Color Coding)
此字段返回当前 LTDC 接口颜色编码
0000：16 位配置 1
0001：16 位配置 2
0010：16 位配置 3
0011：18 位配置 1
0100：18 位配置 2
0101：24 位
0110 - 1111：保留
如果选择命令模式下的 LTDC 接口且当前在命令模式下工作 (CMDM=1)，则 0110-1111：24 位

29.15.50 DSI 主机低功耗模式当前配置寄存器 (DSI_LPMCCR)

DSI Host Low-Power mode Current Configuration Register

偏移地址：0x0118

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

位 31:24 保留，必须保持复位值

位 23:16 **LPSIZE**：最大数据包大小 (Largest Packet Size)
此字段返回可在 VSA、VBP 和 VFP 区域嵌入行中的最大数据包的当前大小（以字节为单位），以便在低功耗模式下发送命令。

位 15:8 保留，必须保持复位值

位 7:0 **VLPSIZE**：VACT 最大数据包大小 (VACT Largest Packet Size)
此字段返回可在 VACT 区域嵌入行中的最大数据包的当前大小（以字节为单位），以便在低功耗模式下发送命令。

29.15.51 DSI 主机视频模式当前配置寄存器 (DSI_VMCCR)

DSI Host Video mode Current Configuration Register

偏移地址：0x0138

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.	Res.	Res.	Res.	Res.	LPCE	FBTAAE	LPHFE	LPHBPE	LPVAE	LPVFPE	LPVBPE	LPVSAE	VMT[1:0]	
ro	ro	ro	ro	ro	ro	ro	ro							ro	

位 31:10 保留，必须保持复位值

位 9 **LPCE**：低功耗命令使能 (Low-Power Command Enable)
此位在低功耗模式下返回当前命令发送状态。
0：禁止在低功耗模式下发送命令。
1：使能在低功耗模式下发送命令。

位 8 **FBTAAE**：帧 BTA 应答使能 (Frame BTA Acknowledge Enable)
此位在帧结束时返回应答响应请求的当前状态。
0：禁止帧结束时响应应答。
1：使能帧结束时响应应答。

位 7 **LPHFE**：低功耗水平前沿使能 (Low-Power Horizontal Front-Porch Enable)
时序允许时，此位返回水平前沿 (HFP) 周期内返回到低功耗的当前状态。
0：禁止在 HFP 周期内返回到低功耗状态。
1：使能在 HFP 周期内返回到低功耗状态。

位 6 **LPHBPE**：低功耗水平后沿使能 (Low-power Horizontal Back-Porch Enable)
时序允许时，此位返回水平后沿 (HBP) 周期内返回到低功耗的当前状态。
0：禁止在 HBP 周期内返回到低功耗状态。
1：使能在 HBP 周期内返回到低功耗状态。

位 5 **LPVAE**：低功耗垂直活动使能 (Low-Power Vertical Active Enable)
时序允许时，此位返回垂直活动 (VACT) 周期内返回到低功耗的当前状态。
0：禁止在 VACT 周期内返回到低功耗状态。
1：使能在 VACT 周期内返回到低功耗状态。

位 4 **LPVFPE**：低功耗垂直前沿使能 (Low-power Vertical Front-Porch Enable)
时序允许时，此位返回垂直前沿 (VFP) 周期内返回到低功耗的当前状态。
0：禁止在 VFP 周期内返回到低功耗状态。
1：使能在 VFP 周期内返回到低功耗状态。

位 3 **LPVBPE**：低功耗垂直后沿使能 (Low-power Vertical Back-Porch Enable)
时序允许时，此位返回垂直后沿 (VBP) 周期内返回到低功耗的当前状态。
0：禁止在 VBP 周期内返回到低功耗状态。
1：使能在 VBP 周期内返回到低功耗状态。

位 2 **LPVSAE**：低功耗垂直同步时间使能 (Low-Power Vertical Sync time Enable)
时序允许时，此位返回垂直同步时间 (VSA) 周期内返回到低功耗的当前状态。
0：禁止在 VSA 周期内返回到低功耗状态。
1：使能在 VSA 周期内返回到低功耗状态。

位 1:0 **VMT**：视频模式类型 (Video mode Type)
此字段返回当前视频模式发送类型：
00：使用同步脉冲的非突发模式。
01：使用同步事件的非突发模式。
1x：突发模式

29.15.52 DSI 主机视频数据包当前配置寄存器 (DSI_VPCCR)

DSI Host Video Packet Current Configuration Register

偏移地址：0x013C

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.	VPSIZE[13:0]													
		ro													

位 31:14 保留，必须保持复位值

位 13:0 **VPSIZE**：视频数据包大小 (Video Packet Size)
此字段返回单个视频数据包中的像素数。

29.15.53 DSI 主机视频块当前配置寄存器 (DSI_VCCCR)

DSI Host Video Chunks Current Configuration Register

偏移地址：0x0140

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.	Res.	NUMC[12:0]												
			ro												

位 31:13 保留，必须保持复位值

位 12:0 **NUMC**：块数 (Number of Chunks)
此字段返回行周期内发送的块数。

29.15.54 DSI 主机视频空包当前配置寄存器 (DSI_VNPCCR)

DSI Host Video Null Packet Current Configuration Register

偏移地址：0x0144

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.	Res.	NPSIZE[12:0]												
			ro												

位 31:13 保留，必须保持复位值

位 12:0 **NPSIZE**：空包大小 (Null Packet Size)
此字段返回空包内的字节数。

29.15.55 DSI 主机视频 HSA 当前配置寄存器 (DSI_VHSACCR)

DSI Host Video HSA Current Configuration Register

偏移地址：0x0148

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.	Res.	Res.	HSA[11:0]											
				ro											

位 31:12 保留，必须保持复位值

位 11:0 **HSA**：水平同步活动持续时间 (Horizontal Synchronism Active duration)
此字段返回以通道字节时钟周期为单位的水平同步活动时间。

29.15.56 DSI 主机视频 HBP 当前配置寄存器 (DSI_VHBPCCR)

DSI Host Video HBP Current Configuration Register

偏移地址：0x014C

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.	Res.	Res.	HBP[11:0]											
				ro											

位 31:12 保留，必须保持复位值

位 11:0 **HBP**：水平后沿持续时间 (Horizontal Back-Porch duration)
此字段返回以通道字节时钟周期为单位的水平后沿时间。

29.15.57 DSI 主机视频行当前配置寄存器 (DSI_VLCCR)

DSI Host Video Line Current Configuration Register

偏移地址：0x0150

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.	HLINE[14:0]														
	ro														

位 31:15 保留，必须保持复位值

位 14:0 **HLINE**：水平行持续时间 (Horizontal Line duration)
此字段返回以通道字节时钟周期为单位的水平行总时间 (HSA+HBP+HACT+HFP)。

29.15.58 DSI 主机视频 VSA 当前配置寄存器 (DSI_VVSACCR)

DSI Host Video VSA Current Configuration Register

偏移地址：0x0154

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.	Res.	Res.	Res.	Res.	VSA[9:0]									
						ro									

位 31:10 保留，必须保持复位值

位 9:0 **VSA**：垂直同步活动持续时间 (Vertical Synchronism Active duration)
此字段返回以水平行数为单位的当前垂直同步活动周期。

29.15.59 DSI 主机视频 VBP 当前配置寄存器 (DSI_VVBPCCR)

DSI Host Video VBP Current Configuration Register

偏移地址：0x0158

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.	Res.	Res.	Res.	Res.	VBP[9:0]									
						ro									

位 31:10 保留，必须保持复位值

位 9:0 **VBP**：垂直后沿持续时间 (Vertical Back-Porch duration)
此字段返回以水平行数为单位的当前垂直后沿周期。

29.15.60 DSI 主机视频 VFP 当前配置寄存器 (DSI_VVFPCCR)

DSI Host Video VFP Current Configuration Register

偏移地址：0x015C

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.	Res.	Res.	Res.	Res.	VFP[9:0]									
						ro									

位 31:10 保留，必须保持复位值

位 9:0 **VFP**：垂直前沿持续时间 (Vertical Front-Porch duration)
此字段返回以水平行数为单位的当前垂直前沿周期。

29.15.61 DSI 主机视频 VA 当前配置寄存器 (DSI_VVACCR)

DSI Host Video VA Current Configuration Register

偏移地址：0x0160

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.	VA[13:0]													
		ro													

位 31:14 保留，必须保持复位值

位 13:0 **VA**：垂直活动持续时间 (Vertical Active duration)
此字段返回以水平行数为单位的当前垂直活动周期。

29.16 DSI 封装器寄存器

29.16.1 DSI 封装器配置寄存器 (DSI_WCFGR)

DSI Wrapper Configuration Register

偏移地址：0x0400

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.	Res.	Res.	Res.	Res.	Res.	Res.	VSPOL	AR	TEPOL	TESRC	COLMUX[2:0]			DSIM
								rw	rw	rw	rw	rw			rw

位 31:8 保留

位 7 **VSPOL**：VSync 极性 (VSync Polarity)
选择 LTDC 停止的 VSync 边沿。
0：LTDC 在下降沿停止。
1：LTDC 在上升沿停止。
此位仅在 DSI 停止时更改（CR.DSIEN = 0 且 CR.EN = 0）。

位 6 **AR**：自动刷新 (Automatic Refresh)
选择 DBI 模式下的刷新模式
0：禁止自动刷新模式。
1：使能自动刷新模式。
此位仅在 DSI 主机停止时更改 (CR.EN = 0)。

位 5 **TEPOL**：TE 极性 (TE Polarity)
选择外部引脚撕裂效应 (TE) 源的极性
0：上升沿。
1：下降沿。
此位仅在 DSI 主机停止时更改 (CR.EN = 0)。

位 4 **TESRC**：TE 源 (TE Source)
选择撕裂效应 (TE) 源
0：DSI 链路。
1：外部引脚。
此位仅在 DSI 主机停止时更改 (CR.EN = 0)。

位 3:1 **COLMUX**：颜色复用 (Color Multiplexing)
选择 DSI 主机使用的颜色复用
000：16 位配置 1
001：16 位配置2
010：16 位配置 3
011：18 位配置 1
100：18 位配置 2
101：24 位
此字段仅在 DSI 停止时更改（CR.DSIEN = 0 且 CR.EN = 0）。

位 0 **DSIM**：DSI 模式 (DSI mode)
选择视频发送的模式
0：视频模式。
1：适配命令模式。
此位仅在 DSI 主机停止时更改 (CR.EN = 0)。

29.16.2 DSI 封装器控制寄存器 (DSI_WCR)

DSI Wrapper Control Register

偏移地址：0x0404

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.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DSIEN	LTDCEN	SHTDN	COLM
												rw	w1o	rw	rw

位 31:4 保留

位 3 **DSIEN**：DSI 使能 (DSI Enable)
使能 DSI 封装器
0：禁止 DSI。
1：使能 DSI。

位 2 **LTDCEN**：LTDC 使能 (LTDC Enable)
使能适配命令模式下帧发送的 LTDC
0：禁止 LTDC。
1：使能 LTDC。

位 1 **SHTDN**：关断 (Shutdown)
控制视频模式下的显示屏关断：
0：显示屏开启。
1：显示屏关闭。

位 0 **COLM**：颜色模式 (Color mode)
控制视频模式下的显示屏颜色模式：
0：全色模式。
1：八色模式。

29.16.3 DSI 封装器中断使能寄存器 (DSI_WIER)

DSI Wrapper Interrupt Enable Register

偏移地址：0x0408

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.	RRIE	Res.	Res.	PLLUIE	PLLLIE	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ERIE	TEIE
		rw			rw	rw								rw	rw

位 31:14 保留，必须保持复位值

位 13 **RRIE**：稳压器就绪中断使能 (Regulator Ready Interrupt Enable)
使能稳压器就绪中断
0：禁止稳压器就绪中断。
1：使能稳压器就绪中断。

位 12:11 保留，必须保持复位值

位 10 **PLLUIE**：PLL 解锁中断使能 (PLL Unlock Interrupt Enable)
使能 PLL 解锁中断
0：禁止 PLL 解锁中断。
1：使能 PLL 解锁中断。

位 9 **PLLLIE**：PLL 锁定中断使能 (PLL Lock Interrupt Enable)
使能 PLL 锁定中断
0：禁止 PLL 锁定中断。
1：使能 PLL 锁定中断。

位 8:2 保留，必须保持复位值

位 1 **ERIE**：刷新结束中断使能 (End of Refresh Interrupt Enable)
使能刷新结束中断
0：禁止刷新结束中断。
1：使能刷新结束中断。

位 0 **TEIE**：撕裂效应中断使能 (Tearing Effect Interrupt Enable)
使能撕裂效应中断
0：禁止撕裂效应中断。
1：使能撕裂效应中断。

29.16.4 DSI 封装器中断和状态寄存器 (DSI_WISR)

DSI Wrapper Interrupt & Status Register

偏移地址：0x040C

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.	RRIF	RRS	Res.	PLLUIF	PLLLIF	PLLLS	Res.	Res.	Res.	Res.	Res.	BUSY	ERIF	TEIF
		r	r		r	r	r						r	r	r

位 31:14 保留，必须保持复位值

位 13 **RRIF**：稳压器就绪中断标志 (Regulator Ready Interrupt Flag)
当稳压器就绪时，此位会被置 1。
0：未发生稳压器就绪事件。
1：发生稳压器就绪事件。

位 12 **RRS**：稳压器就绪状态 (Regulator Ready Status)
此位指示稳压器的状态：
0：稳压器未就绪。
1：稳压器就绪。

位 11 保留，必须保持复位值

位 10 **PLLUIF**：PLL 解锁中断标志 (PLL Unlock Interrupt Flag)
当 PLL 解锁时，此位会被置 1。
0：未发生 PLL 解锁事件。
1：发生 PLL 解锁事件。

位 9 **PLLLIF**：PLL 锁定中断标志 (PLL Lock Interrupt Flag)
当 PLL 锁定时，此位会被置 1。
0：未发生 PLL 锁定事件。
1：发生 PLL 锁定事件。

位 8 **PLLLS**：PLL 锁定状态 (PLL Lock Status)
当 PLL 锁定或未锁定时清零后，此位会被置 1。
0：PLL 未锁定。
1：PLL 已锁定。

位 7:3 保留，必须保持复位值

位 2 **BUSY**：忙标志 (Busy Flag)
在适配命令模式下进行帧传输时，此位会被置 1：
0：无正在进行的传输
1：正在进行传输

位 1 **ERIF**：刷新结束中断标志 (End of Refresh Interrupt Flag)
在适配命令模式下完成帧传输时，此位会被置 1：
0：未发生刷新结束事件。
1：发生刷新结束事件。

位 0 **TEIF**：撕裂效应中断标志 (Tearing Effect Interrupt Flag)
发生撕裂效应事件时，此位会被置 1
0：未发生撕裂效应事件。
1：发生撕裂效应事件。

29.16.5 DSI 封装器中断标志清零寄存器 (DSI_WIFCR)

DSI Wrapper Interrupt Flag Clear Register

偏移地址：0x0410

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.	CRRIF	Res.	Res.	CPLLUIF	CPLLLIF	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CERIF	CTEIF
		w1o			w1o	w1o								w1o	w1o

位 31:14 保留，必须保持复位值

位 13 **CRRIF**：稳压器就绪中断清零标志 (Clear Regulator Ready Interrupt Flag)
写入 1 可将 DSI_WSR 寄存器中的 RRIF 标志清零

位 12:11 保留，必须保持复位值

位 10 **CPLLUIF**：清零 PLL 解锁中断标志 (Clear PLL Unlock Interrupt Flag)
写入 1 可将 DSI_WSR 寄存器中的 PLLUIF 标志清零

位 9 **CPLLLIF**：清零 PLL 锁定中断标志 (Clear PLL Lock Interrupt Flag)
写入 1 可将 DSI_WSR 寄存器中的 PLLLIF 标志清零

位 8:2 保留，必须保持复位值

位 1 **CERIF**：清零刷新结束中断标志 (Clear End of Refresh Interrupt Flag)
写入 1 可将 DSI_WSR 寄存器中的 ERIF 标志清零

位 0 **CTEIF**：清零撕裂效应中断标志 (Clear Tearing Effect Interrupt Flag)
写入 1 可将 DSI_WSR 寄存器中的 TEIF 标志清零

29.16.6 DSI 封装器 PHY 配置寄存器 0 (DSI_WPCR0)

DSI Wrapper PHY Configuration Register 0

偏移地址：0x0418

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.	TCLKPOSTEN	TLPXCEN	THSEXITEN	TLPXDEN	THSZEROEN	THSTRAILEN	THSPREPEN	TCLKZEROEN	TCLKPREPEN	PDEN	Res.	TDDL
				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.	CDOFFDL	FTXSMDL	FTXSMCL	HSIDL1	HSIDL0	HSICL	SWDL1	SWDL0	SWCL	UIX4[5:0]					
	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw					

位 31:28 保留

位 27 **TCLKPOSTEN**：$t_{CLK-POST}$ 的自定义时间使能 (custom time for $t_{CLK-POST}$ Enable)
此位使能在 D-PHY 中手动编程 $t_{CLK-POST}$ 持续时间。必须在 DSI_WPCR4 寄存器的 TCLKPOST 字段中编程所需值。
0：对于 $t_{CLKPOST}$，使用默认值
1：对于 $t_{CLKPOST}$，使用编程值

位 26 **TLPXCEN**：时钟通道的 t_{LPX} 自定义时间使能 (custom time for t_{LPX} for Clock lane Enable)
此位使能在 D-PHY 中手动编程时钟通道的 t_{LPX} 持续时间。必须在 DSI_WPCR3 寄存器的 TLPXC 字段中编程所需值。
0：对于时钟通道的 t_{LPX}，使用默认值。
1：对于时钟通道的 t_{LPX}，使用编程值。

位 25 **THSEXITEN**：$t_{HS-EXIT}$ 的自定义时间使能 (custom time for $t_{HS-EXIT}$ Enable)
此位使能在 D-PHY 中手动编程 $t_{HS-EXIT}$ 持续时间。必须在 DSI_WPCR3 寄存器的 THSEXIT 字段中编程所需值。
0：为 $t_{HS-EXIT}$ 使用默认值
1：为 $t_{HS-EXIT}$ 使用可编程值

位 24 **TLPXDEN**：数据通道的 t_{LPX} 自定义时间使能 (custom time for t_{LPX} for Data lanes Enable)
此位使能在 D-PHY 中手动编程数据通道的 T_{LPX} 持续时间。必须在 DSI_WPCR3 寄存器的 TLPXD 字段中编程所需值。
0：为数据通道的 T_{LPX} 使用默认值。
1：为数据通道的 T_{LPX} 使用可编程值。

位 23 **THSZEROEN**：$t_{HS-ZERO}$ 的自定义时间使能 (custom time for $t_{HS-ZERO}$ Enable)
此位使能在 D-PHY 中手动编程 $t_{HS-ZERO}$ 持续时间。必须在 DSI_WPCR3 寄存器的 THSZERO 字段中编程所需值。
0：为 $t_{HS-ZERO}$ 使用默认值
1：为 $t_{HS-ZERO}$ 使用可编程值

位 22 **THSTRAILEN**：$t_{HS-TRAIL}$ 的自定义时间使能 (custom time for $t_{HS-TRAIL}$ Enable)
此位使能在 D-PHY 中手动编程 $T_{HS-TRAIL}$ 持续时间。必须在 DSI_WPCR2 寄存器的 THSRAIL 字段中编程所需值。
0：为 $T_{HS-TRAIL}$ 使用默认值
1：为 $T_{HS-TRAIL}$ 使用可编程值

位 21 **THSPREPEN**：$t_{HS-PREPARE}$ 的自定义时间使能 (custom time for $t_{HS-PREPARE}$ Enable)
此位使能在 D-PHY 中手动编程 $t_{HS-PREPARE}$ 持续时间。必须在 DSI_WPCR2 寄存器的 THSPREP 字段中编程所需值。
0：为 $t_{HS-PREPARE}$ 使用默认值
1：为 $t_{HS-PREPARE}$ 使用可编程值

位 20 **TCLKZEROEN**：$t_{CLK-ZERO}$ 的自定义时间使能 (custom time for $t_{CLK-ZERO}$ Enable)
此位使能在 D-PHY 中手动编程 $t_{CLK-ZERO}$ 持续时间。必须在 DSI_WPCR2 寄存器的 TCLKZERO 字段中编程所需值。
0：为 $t_{CLK-ZERO}$ 使用默认值
1：为 $t_{CLK-ZERO}$ 使用可编程值

位 19 **TCLKPREPEN**：$t_{CLK-PREPARE}$ 的自定义时间使能 (custom time for $t_{CLK-PREPARE}$ Enable)
此位使能在 D-PHY 中手动编程 $t_{CLK-PREPARE}$ 持续时间。必须在 DSI_WPCR2 寄存器的 TLKCPREP 字段中编程所需值。
0：为 $t_{CLK-PREPARE}$ 使用默认值
1：为 $t_{CLK-PREPARE}$ 使用可编程值

位 18 **PDEN**：下拉使能 (Pull-Down Enable)
此位使能通道上的下拉以防止未使用时出现浮空状态：
0：禁止通道上的下拉
1：使能通道上的下拉

位 17 保留

位 16 **TDDL**：开启禁止数据通道 (Turn Disable Data Lanes)
此位强制数据通道在接收到来自另一端的总线转向请求时保持在 RX 事件中：
0：无影响。
1：BTA 后强制数据通道处于 RX 模式。

位 15 保留

位 14 **CDOFFDL**：数据通道上的争用检测关闭 (Contention Detection OFF on Data Lanes)
仅使用正向逸出模式时，可以将此信号置为高电平来关闭争用检测器并降低静态功耗：
0：数据通道上的争用检测开启。
1：数据通道上的争用检测关闭。

位 13 **FTXSMDL**：强制数据通道进入 TX 停止模式 (Force in TX Stop mode the Data Lanes)
此位强制数据通道进入 TX 停止模式。此位用于在发送模式下初始化通道模块。它会使通道模块立即跳转到发送控制模式并开始发送停止状态 (LP-11)。此位可用于错误 BTA 序列后返回到 TX 模式：
0：无影响。
1：强制数据通道进入 TX 停止模式。

位 12 **FTXSMCL**：强制时钟通道进入 TX 停止模式 (Force in TX Stop mode the Clock Lane)
此位强制时钟通道进入 TX 停止模式。此位用于在发送模式下初始化通道模块。它会使通道模块立即跳转到发送控制模式并开始发送停止状态 (LP-11)。此位可用于错误 BTA 序列后返回到 TX 模式：
0：无影响。
1：强制时钟通道进入 TX 停止模式。

位 11 **HSIDL1**：反转数据通道 1 上的高速数据信号 (Invert the High-Speed data signal on Data Lane 1)
此位反转数据通道 1 上的高速数据信号：
0：正常数据信号配置。
1：反转数据信号配置。

位 10 **HSIDL0**：反转数据通道 0 上的高速数据信号 (Invert the High-Speed data signal on Data Lane 0)

此位反转时钟通道的高速数据信号：

0：正常数据信号配置。

1：反转数据信号配置。

位 9 **HSICL**：反转时钟通道上的高速数据信号 (Invert High-Speed data signal on Clock Lane)

此位反转时钟通道的高速数据信号：

0：正常数据配置。

1：反转数据配置。

位 8 **SWDL1**：交换数据通道 1 引脚 (Swap Data Lane 1 pins)

此位交换时钟通道上的引脚

0：常规时钟通道引脚配置。

1：交换时钟通道引脚。

位 7 **SWDL0**：交换数据通道 0 引脚 (Swap Data Lane 0 pins)

此位交换数据通道 0 上的引脚：

0：常规时钟通道引脚配置。

1：交换时钟通道引脚。

位 6 **SWCL**：交换时钟通道引脚 (Swap Clock Lane pins)

此位交换时钟通道上的引脚：

0：常规时钟通道引脚配置。

1：交换时钟通道引脚。

位 5:0 **UIX4**：单位间隔乘以 4 (Unit Interval multiplied by 4)

此字段以 0.25 ns 为单位定义高速模式下的位周期。

例如，如果单位间隔为 3 ns，此输入的值应为 12 (0x0C)。此值用于生成延迟。如果该周期不是 0.25 ns 的倍数，则应将驱动值向下舍入。例如，600 Mb/s 的链路使用的单位间隔为 1.667 ns。乘以 4 的结果为 6.667。在这种情况下，ui_x4 输入的值应当为 6（而不是 7）。

29.16.7 DSI 封装器 PHY 配置寄存器 1 (DSI_WPCR1)

DSI Wrapper PHY Configuration Register 1

偏移地址：0x041C

复位值：0x0000 0000

注： *此寄存器应仅在 DSI 停止（CR. DSIEN=0 且 CR.EN = 0）时编程。*

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	SDDC	Res.	Res.	LPSRDL[1:0]		LPSRCL[1:0]		Res.	Res.	HSTXDDL[1:0]		HSTXDCL[1:0]	
			rw			rw		rw				rw		rw	

位 31:27 保留

位 26:25 **LPRXFT**：低功耗 RX 低通滤波调谐 (Low-Power RX low-pass Filtering Tuning)

此信号可用于调整 LPRX 输入端的低通滤波器的截止频率。

位 24:23 保留

位 22 **FLPRXLPM**：强制 LP 接收器进入低功耗模式 (Forces LP Receiver in Low-Power mode)

此位使能 LP 接收器 (LPRX) 的低功耗模式。此位置 1 时，LPRX 始终在低功耗模式下运行（此位未激活时，LPRX 仅在 ULPS 期间以低功耗模式运行）：

0：无影响。

1：强制 LPRX 进入低功耗模式。

位 21:20 保留

位 19:18 **HSTXSRCDL**：数据通道上的高速发送压摆率控制 (High-Speed Transmission Slew Rate Control on Data Lanes)

高速发送输出的压摆率控制。可用于更改数据通道 HS 转换的压摆率。

默认值应为“00”。

位 17:16 **HSTXSRCCL**：时钟通道上的高速发送压摆率控制 (High-Speed Transmission Slew Rate Control on Clock Lane)

高速发送输出的压摆率控制。可用于更改时钟通道 HS 转换的压摆率。

默认值应为“00”。

位 15:13 保留

位 12 **SDDC**：SDD 控制 (SDD Control)

打开附加电流路径，以在时钟和数据通道上满足 MIPI® D-PHY 规范定义的 SDDTx 参数。

0：无影响。

1：在所有通道上激活附加电流路径。

位 11:10 保留

位 9:8 **LPSRCDL**：数据通道上的低功耗发送压摆率补偿 (Low-Power transmission Slew Rate Compensation on Data Lanes)

可用于更改数据通道 LP 转换的压摆率。

默认值应为“00”。

位 7:6 **LPSRCCL**：时钟通道上的低功耗发送压摆率补偿 (Low-Power transmission Slew Rate Compensation on Clock Lane)

可用于更改时钟通道 LP 转换的压摆率。

默认值应为“00”。

位 5:4 保留

位 3:2 **HSTXDDL**：数据通道上高速发送延迟 (High-Speed Transmission Delay on Data Lanes)

延迟调谐器控制以改变数据路径中的延迟（直到 DP/DN）。可用于根据 DP/DN 上的时钟边沿更改数据边沿转换位置。

默认值应为“00”。

位 1:0 **HSTXDCL**：时钟通道上高速发送延迟 (High-Speed Transmission Delay on Clock Lane)

延迟调谐器控制以改变时钟路径中的延迟（直到 DP/DN）。可用于根据 DP/DN 上数据位转换更改时钟边沿位置。

默认值应为“00”。

29.16.8 DSI 封装器 PHY 配置寄存器 2 (DSI_WPCR2)

DSI Wrapper PHY Configuration Register 2

偏移地址：0x0420

复位值：0x0000 0000

注： *此寄存器应仅在 DSI 停止（CR. DSIEN=0 且 CR.EN = 0）时编程。*

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

位 31:24 **THSTRAIL**：$t_{HSTRAIL}$

此字段定义 MIPI® D-PHY 规范中指定的 $t_{HS-TRAIL}$。DSI_WPCR0 的 THSTRAILEN 位置 1 时，D-PHY 将使用该值。

THSTRAIL = 2 x $t_{HS-TRAIL}$（以 ns 表示）。当 DSI_WPCR0 的 THSTRAILEN 位复位时，D-PHY 使用的默认值为 140，即 70 ns + 8*UI。

位 23:16 **THSPREP**：$t_{HS-PREPARE}$

此字段定义 MIPI® D-PHY 规范中指定的 $t_{HS-PREPARE}$。DSI_WPCR0 的 THSPREPEN 位置 1 时，D-PHY 将使用该值。

THSPREP = 2 x $t_{HS-PREPARE}$（以 ns 表示）。当 DSI_WPCR0 的 THSPREPEN 位复位时，D-PHY 使用的默认值为 126，即 63 ns + 12*UI。

位 15:8 **TCLKZERO**：$t_{CLK-ZERO}$

此字段定义 MIPI® D-PHY 规范中指定的 $t_{CLK-ZERO}$。DSI_WPCR0 的 TCLKZEROEN 位置 1 时，D-PHY 将使用该值。

TCLKZERO = $t_{CLK-ZERO}$ / 2（以 ns 表示）。当 DSI_WPCR0 的 TCLKZEROEN 位复位时，D-PHY 使用的默认值为 195，即 390 ns。

位 7:0 **TCLKPREP**：$t_{CLK-PREPARE}$

此字段定义 MIPI® D-PHY 规范中指定的 $t_{CLK-PREPARE}$。DSI_WPCR0 的 TCLKPREPEN 位置 1 时，D-PHY 将使用该值。

TCLKPREP = 2 x $t_{CLK-PREPARE}$（以 ns 表示）。当 DSI_WPCR0 的 TCLKPREPEN 位复位时，D-PHY 使用的默认值为 120，即 60 ns + 20*UI。

29.16.9 DSI 封装器 PHY 配置寄存器 3 (DSI_WPCR4)

DSI Wrapper PHY Configuration Register 3

偏移地址：0x0424

复位值：0x0000 0000

注： *此寄存器应仅在 DSI 停止（CR. DSIEN=0 且 DSI_CR.EN = 0）时编程。*

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

位 31:24 **TLPXC**：数时钟通道的 t_{LPXC} (t_{LPXC} for Clock lane)

此字段定义 MIPI® D-PHY 规范中指定的时钟通道的 t_{LPX}。DSI_WPCR1 的 TLPXCEN 位置 1 时，D-PHY 将使用该值。

TLPXC = 2 x t_{LPX}（以 ns 表示）。当 DSI_WPCR1 的 TLPXCEN 位复位时，D-PHY 使用的默认值为 100，即 50 ns。

位 23:16 **THSEXIT**：t_{HSEXIT}

此字段定义 MIPI® D-PHY 规范中指定的 $t_{HS-EXIT}$。DSI_WPCR1 的 THSEXITEN 位置 1 时，D-PHY 将使用该值。

THSEXIT = $t_{HS-ZERO}$（以 ns 表示）。当 DSI_WPCR1 的 THSEXITEN 位复位时，D-PHY 使用的默认值为 100，即 100 ns。

位 15:8 **TLPXD**：数据通道的 t_{LPX} (t_{LPX} for Data lanes)

此字段定义 MIPI® D-PHY 规范中指定的数据通道的 t_{LPX}。DSI_WPCR1 的 TLPXDEN 位置 1 时，D-PHY 将使用该值。

TLPXD = 2 x t_{LPX}（以 ns 表示）。当 DSI_WPCR1 的 TLPXDEN 位复位时，D-PHY 使用的默认值为 100，即 50 ns。

位 7:0 **THSZERO**：$t_{HS-ZERO}$

此字段定义 MIPI® D-PHY 规范中指定的 $t_{HS-ZERO}$。DSI_WPCR1 的 THSZEROEN 位置 1 时，D-PHY 将使用该值。

THSZERO = $t_{HS-ZERO}$（以 ns 表示）。当 DSI_WPCR1 的 THSZEROEN 位复位时，D-PHY 使用的默认值为 175，即 175 ns。

29.16.10 DSI 封装器 PHY 配置寄存器 4 (DSI_WPCR4)

DSI Wrapper PHY Configuration Register 4

偏移地址：0x0428

复位值：0x0000 0000

注： *此寄存器应仅在 DSI 停止（CR. DSIEN=0 且 DSI_CR.EN = 0）时编程。*

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.	Res.	Res.	Res.	Res.	Res.	Res.	TCLKPOST[7:0]							
								rw							

位 31:8 保留

位 7:0 **TCLKPOST**：$t_{CLK-POST}$
此字段定义 MIPI® D-PHY 规范中指定的 $t_{CLK-POST}$。DSI_WPCR0 的 TCLKPOSTEN 位置 1 时，D-PHY 将使用该值。
TCLKPOST = 2 x $t_{CLK-POST}$（以 ns 表示）。当 DSI_WPCR0 的 TCLKPOSTEN 位复位时，D-PHY 使用的默认值为 200，即 100 ns + 120*UI。

29.16.11 DSI 封装器稳压器和 PLL 控制寄存器 (DSI_WRPCR)

DSI Wrapper Regulator and PLL Control Register

偏移地址：0x0430

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.	REGEN	Res.	Res.	Res.	Res.	Res.	Res.	ODF[1:0]	
							rw							rw	

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

位 31:25 保留

位 24 **REGEN**：使能稳压器 (Regulator Enable)
此位使能 DPHY 稳压器：
0：禁止稳压器
1：使能稳压器

位 23:18 保留

位 17:16 **ODF**：PLL 输出分频系数 (PLL Output Division Factor)
此字段配置 PLL 输出分频系数：
00：PLL 输出 1 分频。
01：PLL 输出 2 分频。
10：PLL 输出 4 分频。
11：PLL 输出 8 分频。

位 15 保留

位 14:11 **IDF**：PLL 输入分频系数 (PLL Input Division Factor)
此字段配置 PLL 输入分频系数：
000：PLL 输入 1 分频。
001：PLL 输入 1 分频。
010：PLL 输入 2 分频。
011：PLL 输入 3 分频。
100：PLL 输入 4 分频。
101：PLL 输入 5 分频。
110：PLL 输入 6 分频。
111：PLL 输入 7 分频。

位 10:9 保留

位 8:2 **NDIV**：PLL 环路分频系数 (PLL Loop Division Factor)
此字段配置 PLL 环路分频系数：
0 到 9：保留。
10 到 125：允许的环路分频系数值。
126 到 127：保留。

位 1 保留

位 0 **PLLEN**：PLL 使能 (PLL Enable)
此位使能 D-PHY PLL：
0：禁用 PLL。
1：使能 PLL。

29.17 DSI 主机寄存器映射

下表对 DSI 主机寄存器进行了汇总。有关基本地址，请参见寄存器边界地址表。

表 185. DSIHOST 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
0x0000	**DSI_VR**	VERSION[31:0]																															
	Reset value	0	0	1	1	0	0	0	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	0	0	0	0	1	0	1	1	0	0
0x0004	**DSI_CR**	Res.																															EN
	Reset value																																0
0x0008	**DSIHSOT_CCR**	Res.																TOCKDIV[7:0]								TXECKDIV[7:0]							
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x000C	**DSI_LVCIDR**	Res.																														VCID[1:0]	
	Reset value																															0	0
0x0010	**DSI_LCOLCR**	Res.																							LPE	Res.				COLC[3:0]			
	Reset value																								0					0	0	0	0
0x0014	**DSI_LPCR**	Res.																													HSP	VSP	DEP
	Reset value																														0	0	0
0x0018	**DSI_LPMCR**	Res.								LPSIZE[7:0]																VLPSIZE[7:0]							
	Reset value									0	0	0	0	0	0	0	0									0	0	0	0	0	0	0	0
0x001C-0x0028		Res.																															
0x002C	**DSI_PCR**	Res.																											CRCRXE	ECCRXE	BTAE	ETRXE	ETTXE
	Reset value																												0	0	0	0	0
0x0030	**DSI_GVCIDR**	Res.																														VCID[1:0]	
	Reset value																															0	0
0x0034	**DSI_MCR**	Res.																															CMDM
	Reset value																																1
0x0038	**DSI_VMCR**	Res.							PGO	Res.			PGM	Res.			PGE	LPCE	FBTAAE	LPHFE	LPHBPE	LVAE	LPVFPE	LPVBPE	LPVSAE	Res.						VMT[1:0]	
	Reset value								0				0				0	0	0	0	0	0	0	0	0							0	0
0x003C	**DSI_VPCR**	Res.																VPSIZE[13:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0040	**DSI_VCCR**	Res.																	NUMC[12:0]														
	Reset value																		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0044	**DSI_VNPCR**	Res.																	NPSIZE[12:0]														
	Reset value																		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0048	**DSI_VHSACR**	Res.																		HSA[11:0]													
	Reset value																			0	0	0	0	0	0	0	0	0	0	0	0	0	0

表 185. DSIHOST 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
0x004C	**DSI_VHBPCR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	HBP[11:0]												
	Reset value																				0	0	0	0	0	0	0	0	0	0	0	0	0
0x0050	**DSI_VLCR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	HLINE[14:0]															
	Reset value																		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0054	**DSI_VVSACR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	VSA[9:0]									
	Reset value																							0	0	0	0	0	0	0	0	0	0
0x0058	**DSI_VVBPCR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	VBP[9:0]									
	Reset value																							0	0	0	0	0	0	0	0	0	0
0x005C	**DSI_VVFPCR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	VFP[9:0]									
	Reset value																							0	0	0	0	0	0	0	0	0	0
0x0060	**DSI_VVACR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	VA[13:0]													
	Reset value																			0	0	0		0	0	0	0	0	0	0	0	0	0
0x0064	**DSI_LCCR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CMDSIZE[15:0]															
	Reset value																	0	0	0		0	0	0	0	0	0	0	0	0	0	0	0
0x0068	**DSI_CMCR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MRDPS	Res.	Res.	Res.	Res.	DLWTX	DSR0TX	DSW1TX	DSW0TX	Res.	GLWTX	GSR2TX	GSR1TX	GSR0TX	GSW2TX	GSW1TX	GSW0TX	Res.	Res.	Res.	Res.	Res.	Res.	ARE	TEARE
	Reset value								0					0	0	0	0		0	0	0	0	0	0	0							0	0
0x006C	**DSI_GHCR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WCMSB[7:0]								WCLSB[7:0]								VCID		DT[5:0]					
	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
0x0070	**DSI_GPDR**	DATA4[7:0]								DATA3[7:0]								DATA2[7:0]								DATA1[7:0]							
	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
0x0074	**DSI_CMCR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RCB	PRDFF	PRDFE	PWRFF	PWRFE	CMDFF	CMDFE
	Reset value																										0	0	0	0	0	0	0
0x0078	**DSI_TCCR0**	HSTX_TOCNT[15:0]																LPRX_TOCNT[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
0x007C	**DSI_TCCR1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	HSRD_TOCNT[15:0]															
	Reset value																	0	0	0		0	0	0	0	0	0	0	0	0	0	0	0
0x0080	**DSI_TCCR2**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LPRD_TOCNT[15:0]															
	Reset value																	0	0	0		0	0	0	0	0	0	0	0	0	0	0	0
0x0084	**DSI_TCCR3**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PM	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	HSWR_TOCNT[15:0]															
	Reset value								0									0	0	0		0	0	0	0	0	0	0	0	0	0	0	0
0x0088	**DSI_TCCR4**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LSWR_TOCNT[15:0]															
	Reset value																	0	0	0		0	0	0	0	0	0	0	0	0	0	0	0
0x008C	**DSI_TCCR5**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BTA_TOCNT[15:0]															
	Reset value																	0	0	0		0	0	0	0	0	0	0	0	0	0	0	0
0x0094	**DSI_CLCR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ACR	DPCC
	Reset value																															0	0
0x0098	**DSI_CLTCR**	Res.	Res.	Res.	Res.	Res.	Res.	HS2LP_TIME[9:0]										Res.	Res.	Res.	Res.	Res.	Res.	LP2HS_TIME[9:0]									
	Reset value							0	0	0	0	0	0	0	0	0	0							0	0	0	0	0	0	0	0	0	0

表 185. DSIHOST 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
0x009C	**DSI_DLTCR**	HS2LP_TIME[7:0]								LP2HS_TIME[7:0]								Res.	MRD_TIME[14:0]														
	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
0x00A0	**DSI_PCTLR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CKE	DEN	Res.
	Reset value																														0	0	
0x00A4	**DSI_PCCONFR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SW_TIME[7:0]								Res.	Res.	Res.	Res.	Res.	Res.	NL[1:0]	
	Reset value																	0	0	0	0	0	0	0	0							0	0
0x00A8	**DSI_PUCR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	UEDL	URDL	UECL	URCL
	Reset value																													0	0	0	0
0x00AC	**DSI_PTTCR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TX_TRIG[3:0]			
	Reset value																													0	0	0	0
0x00B0	**DSI_PSR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	UAN1	PSS1	RUE0	UAN0	PSS0	UANC	PSSC	PD	Res.
	Reset value																								0	0	0	0	0	0	0	0	
0x00B4 - 0x00B8		Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
0x00BC	**DSI_ISR0**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PE4	PE3	PE2	PE1	PE0	AE15	AE14	AE13	AE12	AE11	AE10	AE9	AE8	AE7	AE6	AE5	AE4	AE3	AE2	AE1	AE0
	Reset value												0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x00C0	**DSI_ISR1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	GPRXE	GPRDE	GPTXE	GPWRE	GCWRE	LPWRE	EOTPE	PSE	CRCE	ECCME	ECCSE	TOLPRX	TOHSTX
	Reset value																				0	0	0	0	0	0	0	0	0	0	0	0	0
0x00C4	**DSI_IER0**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PE4IE	PE3IE	PE2IE	PE1IE	PE0IE	AE15IE	AE14IE	AE13IE	AE12IE	AE11IE	AE10IE	AE9IE	AE8IE	AE7IE	AE6IE	AE5IE	AE4IE	AE3IE	AE2IE	AE1IE	AE0IE
	Reset value												0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x00C8	**DSI_IER1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	GPRXEIE	GPRDEIE	GPTXEIE	GPWREIE	GCWREIE	LPWREIE	EOTPEIE	PSEIE	CRCEIE	ECCMEIE	ECCSEIE	TOLPRXIE	TOHSTXIE
	Reset value																				0	0	0	0	0	0	0	0	0	0	0	0	0
0x00CC- 0x00D4		Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
0x00D8	**DSI_FIR0**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FPE4	FPE3	FPE2	FPE1	FPE0	FAE15	FAE14	FAE13	FAE12	FAE11	FAE10	FAE9	FAE8	FAE7	FAE6	FAE5	FAE4	FAE3	FAE2	FAE1	FAE0
	Reset value												0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x00DC	**DSI_FIR1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FGPRXE	FGPRDE	FGPTXE	FGPWRE	FGCWRE	FLPWRE	FEOTPE	FPSE	FCRCE	FECCME	FECCSE	FTOLPRX	FTOHSTX
	Reset value																				0	0	0	0	0	0	0	0	0	0	0	0	0
0x00E0 - 0x00FF		Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
0x0100	**DSI_VSCR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	UR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	EN
	Reset value																								0								0

表 185. DSIHOST 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
0x0104 - 0x0108		Res.																															
0x010C	**DSI_LCVCIDR**	Res.																														VCID[1:0]	
	Reset value																															0	0
0x0110	**DSI_LCCCR**	Res.																							LPE					COLC[3:0]			
	Reset value																								0					0	0	0	0
0x0114		Res.																															
0x0118	**DSI_LPMCCR**	Res.								LPSIZE[7:0]																VLPSIZE[7:0]							
	Reset value									0	0	0	0	0	0	0	0									0	0	0	0	0	0	0	0
0x0120- 0x0138		Res.																															
0x0138	**DSI_VMCCR**	Res.																						LPCE	FBTAAE	LPHFE	LPHBPE	LVAE	LPVFPE	LPVBPE	LPVSAE	VMT[1:0]	
	Reset value																	0	0	0	0	0	0	0	0							0	0
0x013C	**DSI_VPCCR**	Res.																	VPSIZE[13:0]														
	Reset value																		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0140	**DSI_VCCCR**	Res.																		NUMC[12:0]													
	Reset value																			0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0144	**DSI_VNPCCR**	Res.																		NPSIZE[12:0]													
	Reset value																			0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0148	**DSI_VHSACCR**	Res.																			HSA[11:0]												
	Reset value																				0	0	0	0	0	0	0	0	0	0	0	0	0
0x014C	**DSI_VHBPCCR**	Res.																			HBP[11:0]												
	Reset value																				0	0	0	0	0	0	0	0	0	0	0	0	0
0x0150	**DSI_VLCCR**	Res.																		HLINE[14:0]													
	Reset value																			0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0154	**DSI_VVSACCR**	Res.																						VSA[9:0]									
	Reset value																							0	0	0	0	0	0	0	0	0	0
0x0158	**DSI_VVBPCCR**	Res.																						VBP[9:0]									
	Reset value																							0	0	0	0	0	0	0	0	0	0
0x015C	**DSI_VVFPCCR**	Res.																						VFP[9:0]									
	Reset value																							0	0	0	0	0	0	0	0	0	0
0x0160	**DSI_VVACCR**	Res.																		VA[13:0]													
	Reset value																			0	0	0		0	0	0	0	0	0	0	0	0	0
0x0164- 0x0190		Res.																															
0x0194- 0x03FC		Res.																															

表 185. DSIHOST 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
0x0400	**DSI_WCFGR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	VSPOL	AR	TEPOL	TESRC	COLMUX[2:0]			DSIM
	Reset value																									0	0	0	0	0	0	0	0
0x0404	**DSI_WCR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DSIEN	LTDCEN	SHTDN	COLM
	Reset value																													0	0	0	0
0x0408	**DSI_WIER**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RRIE	Res.	Res.	PLLUIE	PLLLIE	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ERIE	TEIE
	Reset value																			0			0	0								0	0
0x040C	**DSI_WISR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RRIF	RRS	Res.	PLLUIF	PLLLIF	PLLLS	Res.	Res.	Res.	Res.	Res.	BUSY	ERIF	TEIF
	Reset value																			0	0		0	0	0						0	0	0
0x0410	**DSI_WIFCR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CRRIF	Res.	Res.	CPLLUIF	CPLLLIF	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CERIF	CTEIF
	Reset value																			0			0	0								0	0
0x0414		Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
0x0418	**DSI_WPCR0**	Res.	Res.	Res.	Res.	TCLKPOSTEN	TLPXCEN	THSEXITEN	TLPXDEN	THSZEROEN	THSTRAILEN	THSPREPEN	TCLKZEROEN	TCLKPREPEN	PDEN	Res.	TDDL	Res.	CDOFFDL	FTXSMDL	FTXSMCL	HSIDL1	HSIDL0	HSICL	SWDL1	SWDL0	SWCL	UIX4[: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
0x041C	**DSI_WPCR1**	Res.	Res.	Res.	Res.	Res.	LPRXFT[1:0]		Res.	Res.	FLPRXLPM	Res.	Res.	HSTXSRCDL[1:0]		HSTXSRCCL[1:0]		Res.	Res.	Res.	SDCC	Res.	Res.	LPSRDL[1:0]		LPSRCL[1:0]		Res.	Res.	HSTXDLL[1:0]		HSTXDCL[1:0]	
	Reset value						0	0			0			0	0	0	0				0			0	0	0	0			0	0	0	0
0x0420	**DSI_WPCR2**	THSTRAIL[7:0]								THSPREP[7:0]								TCLKZEO[7:0]								TCLKPREP[7:0]							
	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
0x0424	**DSI_WPCR3**	TLPXC[7:0]								THSEXIT[7:0]								TLPXD[7:0]								THSZERO[7:0]							
	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
0x0428	**DSI_WPCR4**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	THSZERO[7:0]							
	Reset value																									0	0	0	0	0	0	0	0
0x042C		Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
0x0430	**DSI_WRPCR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REGEN	Res.	Res.	Res.	Res.	Res.	Res.	ODF[1:0]		Res.	IDF[3:0]				Res.	Res.	NDIV[6:0]						Res.		PLLEN
	Reset value								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*第 78 页的第 2.2.2 节*。

30 触摸感应控制器 (TSC)

30.1 简介

触摸感应控制器提供了一种简单的解决方案，可用于向任何应用添加电容感应功能。电容感应技术能够检测电极附近的手指，该电极受绝缘体（例如玻璃、塑料）保护以防直接接触。由手指（或任何导电物质）产生的电容变化可基于表面电荷转移采集原理，使用已证实有效的实现方法进行测量。

触摸感应控制器受 STMTouch 触摸感应固件库完全支持，该固件库可供免费使用，用于在最终应用中可靠地实现触摸感应功能。

30.2 TSC 主要特性

触摸感应控制器具有以下主要特性：

- 采用经验证的稳定的表面电荷转移采集原理
- 支持多达 24 个电容感应通道
- 可以并行采集多达 8 个电容感应通道，因此具备极佳的响应时间
- 具有扩频功能，可以提高系统在噪声环境中的稳定性
- 对电荷转移采集序列完全采用硬件管理
- 可编程电荷转移频率
- 可编程采样电容 I/O 引脚
- 可编程通道 I/O 引脚
- 可编程最大计数值，从而避免通道出现故障时采集时间过长
- 具有中断功能的专用采集结束和最大计数错误标志
- 一个采样电容可用于多达 3 个电容感应通道，从而可减少系统组件
- 与接近、触键、线性和旋转触摸传感器实现方案兼容
- 设计为与 STMTouch 触摸感应固件库配合使用

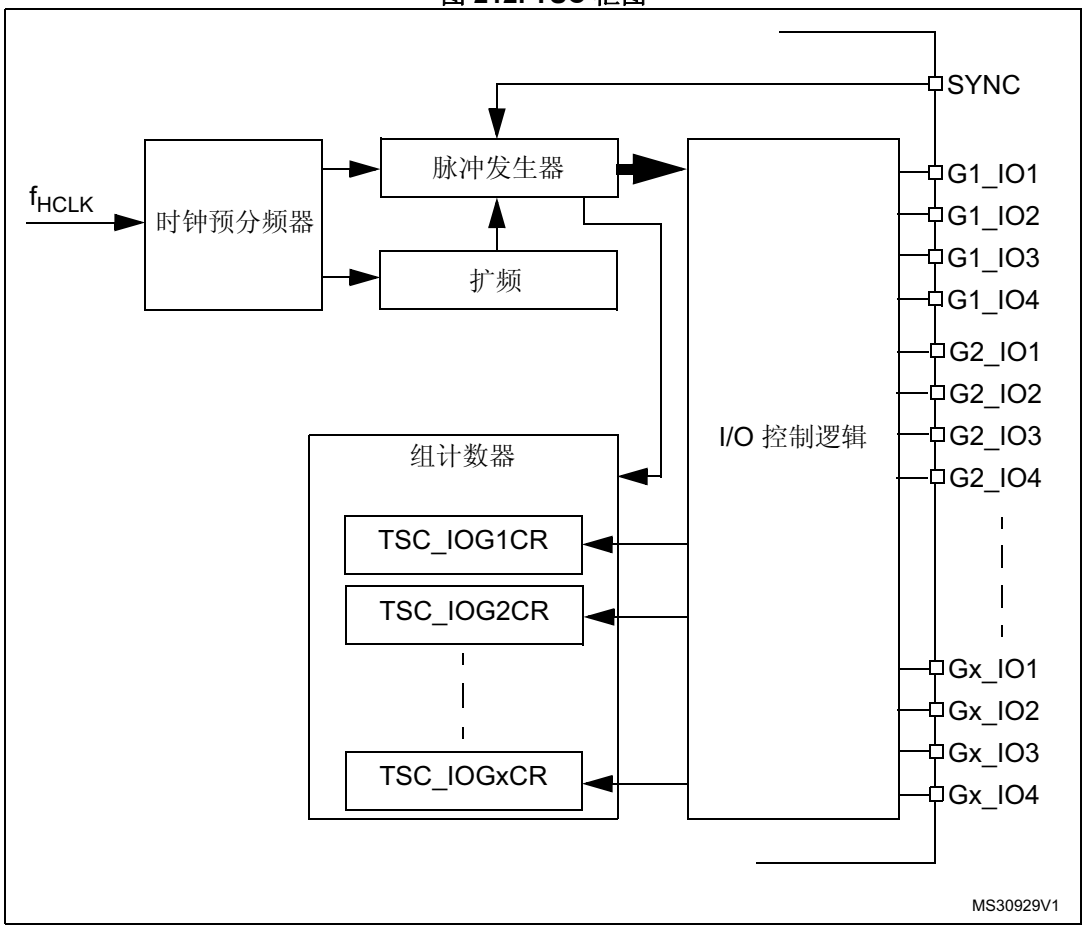
注： *电容感应通道的数量取决于封装尺寸并受 I/O 可用性的限制。*

30.3 TSC 功能说明

30.3.1 TSC 框图

触摸感应控制器的框图如*图 212：TSC 框图*所示。

图 212. TSC 框图



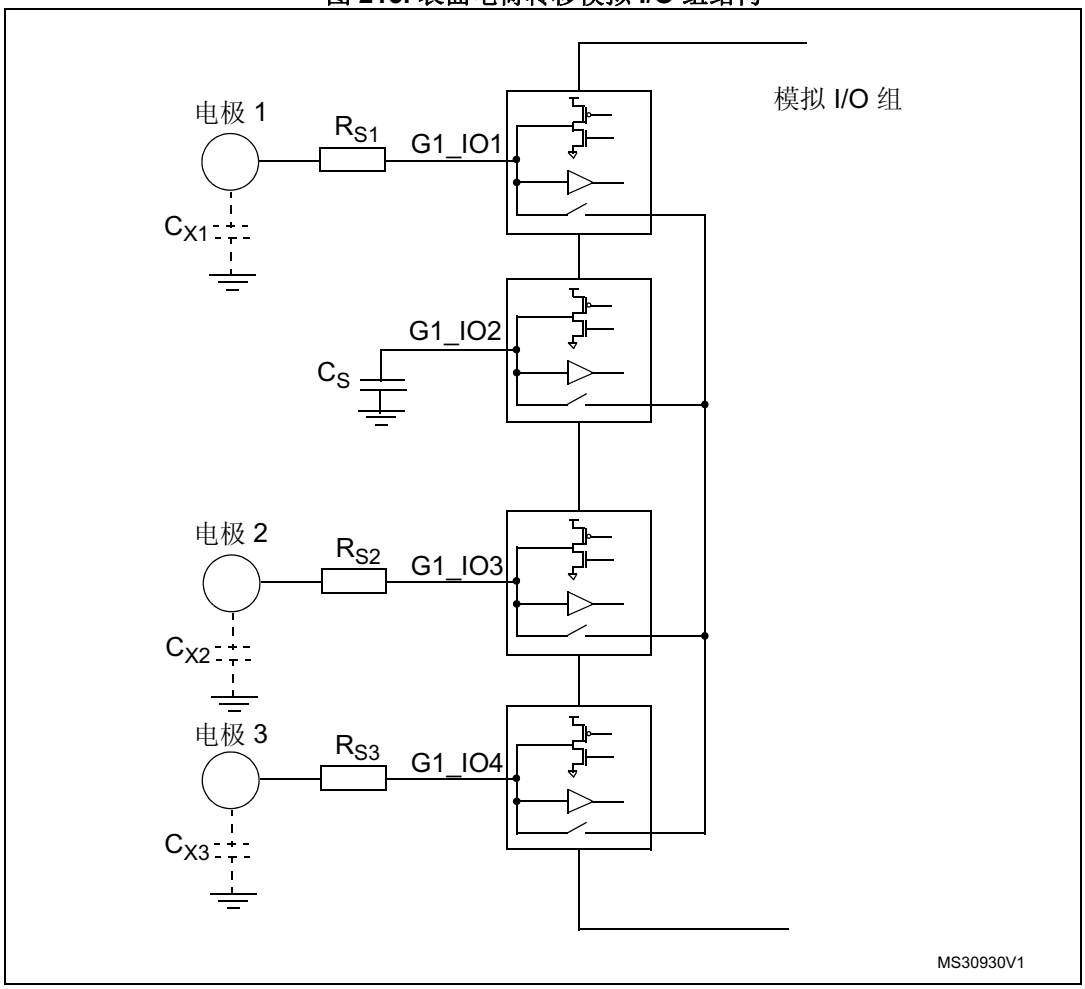
30.3.2 表面电荷转移采集概述

表面电荷转移采集是用于测量电容的一种成熟、稳定且有效的方式。它使用最少数量的外部组件对单端电极类型进行操作。该采集围绕模拟 I/O 组设计，后者由四个 GPIO 组成。（请参见*图 213*）。提供多个模拟 I/O 组，以便可以同时采集多个电容感应通道并支持更多的电容感应通道。在同一个模拟 I/O 组内，电容感应通道的采集操作是连续的。

其中一个 GPIO 专门用于采样电容 C_S。每个模拟 I/O 组在同一时间必须仅使能一个采样电容 I/O。

其余的 GPIO 专用于电极，通常称为通道。对于一些特定需要（如接近检测），每个模拟 I/O 组也可以同时使能多个通道。

图 213. 表面电荷转移模拟 I/O 组结构



注： *Gx_IOy，其中 x 是模拟 I/O 组编号，y 是所选组内的 GPIO 编号。*

表面电荷转移采集原理包括对电极电容 (C_X) 进行充电并将一部分累积电荷转移到采样电容 (C_S)。重复该过程，直到 C_S 两端的电压达到给定的阈值（在本例中为 V_{IH}）。达到阈值所需的电荷转移次数直接表示电极电容的大小。

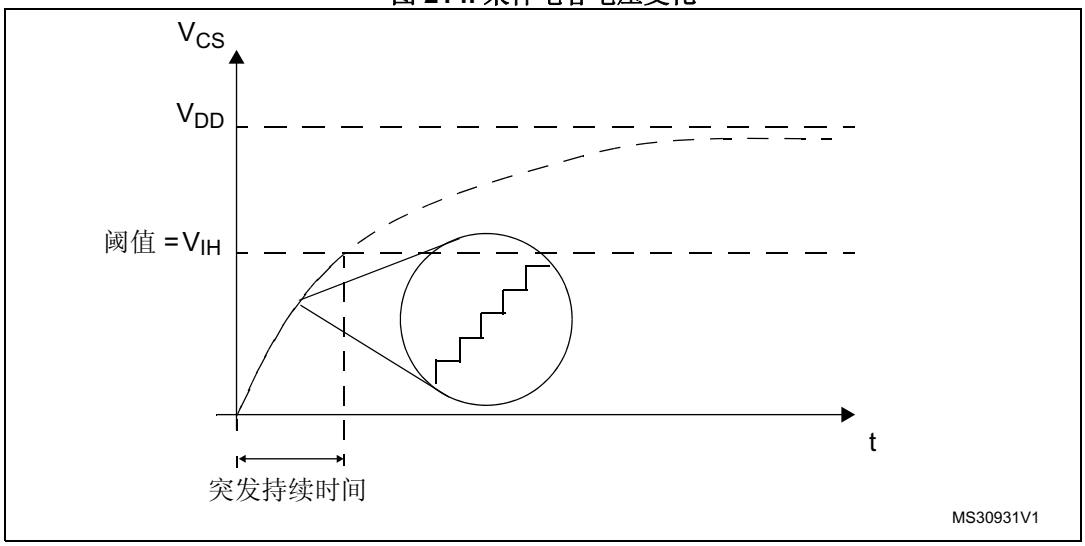
表 186 详细说明了电容感应通道 1 的电荷转移采集序列。重复状态 3 到 7，直到 C_S 两端的电压达到给定的阈值。同一序列适用于其他通道的采集。电极串联电阻 R_S 提高了解决方案的 ESD 抗扰性。

表 186. 采集序列汇总

状态	G1_IO1（通道）	G1_IO2（采样）	G1_IO3（通道）	G1_IO4（通道）	状态说明
#1	输入浮空且模拟开关闭合	开漏输出处于低电平，且模拟开关闭合	输入浮空且模拟开关闭合		将所有 C_X 和 C_S 放电
#2	输入浮空				死区时间
#3	推挽输出处于高电平	输入浮空			将 C_{X1} 充电
#4	输入浮空				死区时间
#5	输入浮空且模拟开关闭合		输入浮空		将电荷从 C_{X1} 转移到 C_S
#6	输入浮空				死区时间
#7	输入浮空				测量 C_S 电压

下图详细描述了采样电容 C_S 的电压随时间的变化情况：

图 214. 采样电容电压变化



30.3.3 复位和时钟

TSC 时钟源是 AHB 时钟 (HCLK)。两个可编程预分频器用于产生脉冲发生器和扩频内部时钟：

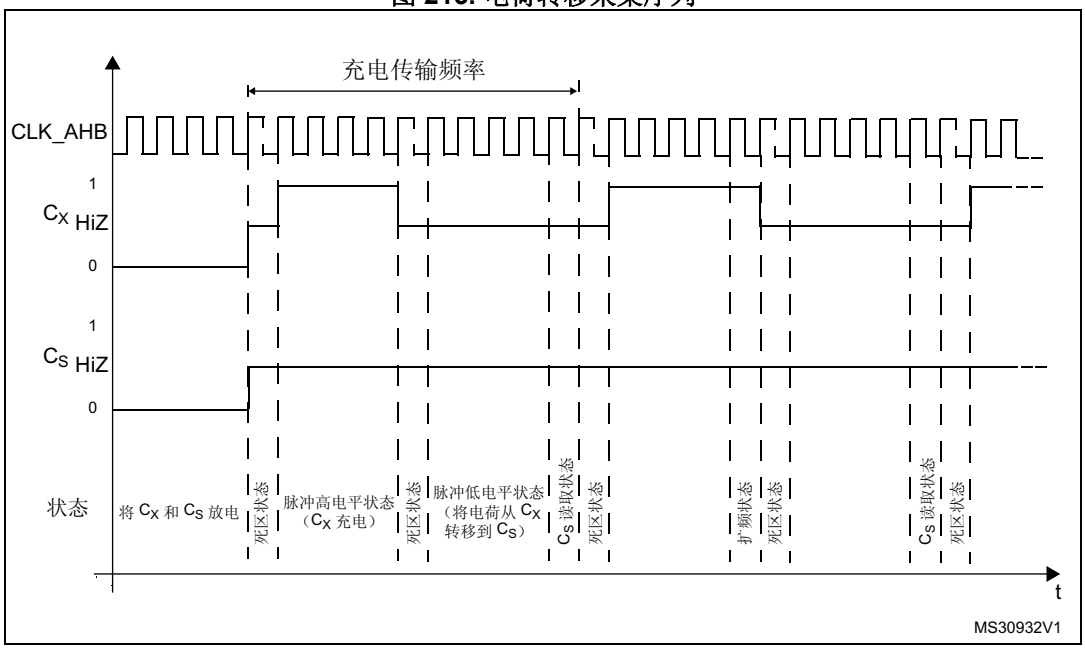
- 脉冲发生器时钟 (PGCLK) 使用 TSC_CR 寄存器的 PGPSC[2:0] 位进行定义
- 扩频时钟 (SSCLK) 使用 TSC_CR 寄存器的 SSPSC 位进行定义

复位和时钟控制器 (RCC) 提供用于使能触摸感应控制器时钟及复位该外设的专用位。有关详细信息，请参见*第 6 节：复位和时钟控制 (RCC)*。

30.3.4 电荷转移采集序列

图 215 详细描述了电荷转移采集序列的示例。

图 215. 电荷转移采集序列



为提高灵活性，可完全配置电荷转移频率。可以使用 TSC_CR 寄存器中的 CTPH[3:0] 和 CTPL[3:0] 位定义脉冲高电平状态（为 C_X 充电）和脉冲低电平状态（将电荷从 C_X 转移到 C_S）的持续时间。脉冲高电平和低电平状态持续时间的标准范围为 500 ns 到 2 µs。为了确保正确测量电极电容，必须设置脉冲高电平状态持续时间以确保 C_X 始终充分充电。

在脉冲高电平和低电平状态之间插入死区时间（在此期间，采样电容 I/O 和通道 I/O 处于输入浮空状态），以确保最优的电荷转移采集序列。死区时间状态的持续时间是 HCLK 的 1 个周期。

如果已启用扩频功能，在脉冲高电平状态结束时，将增加可变数量的 SSCLK 时钟周期。

在脉冲低电平状态的结束时，会读取采样电容 I/O，确定 C_S 两端的电压是否达到给定阈值，读取操作的持续时间是 HCLK 的一个周期。

注： *禁止采用以下 TSC 控制寄存器配置：*

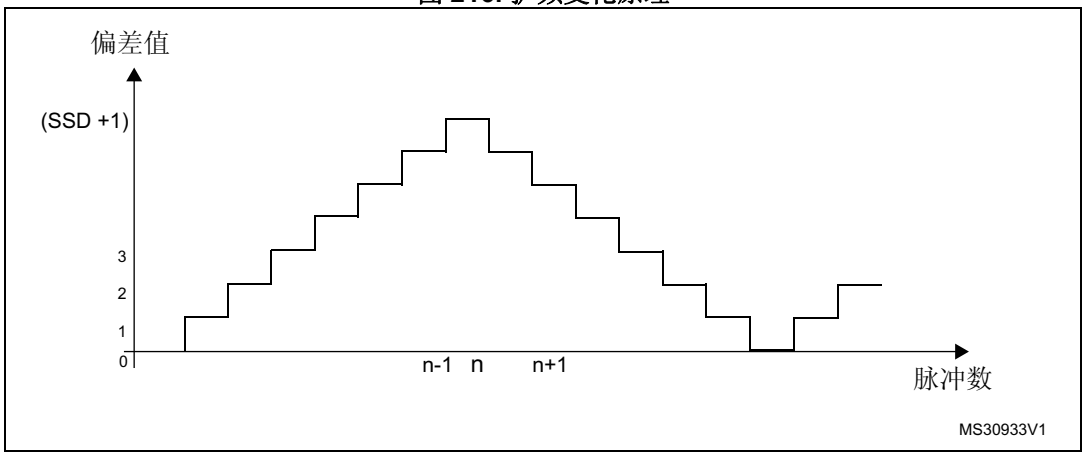
- *PGPSC 位设置为 “000” 且 CTPL 位设置为 “0000”*
- *PGPSC 位设置为 “000” 且 CTPL 位设置为 “0001”*
- *PGPSC 位设置为 “001” 且 CTPL 位设置为 “0000”*

30.3.5 扩频功能

扩频功能用于使电荷转移频率发生变化。这样做是为了改善噪声环境下电荷转移采集的稳定性，并减少由此产生的辐射。最大频率变化范围为标称电荷转移周期的 10% 到 50%。例如，对于 250 kHz (4 µs) 的标称电荷转移频率，典型的扩频偏差为 10% (400 ns)，这将导致出现 227 kHz 的最低电荷转移频率。

实际上，扩频包括使用以下所示的原理将可变数量的 SSCLK 周期添加到脉冲高电平状态：

图 216. 扩频变化原理



下表详细列出了不同 HCLK 设置对应的最大频率偏差：

表 187. 扩频偏差与 AHB 时钟频率之间的关系

f_{HCLK}	扩频步长	最大扩频偏差
24 MHz	41.6 ns	10666.6 ns
48 MHz	20.8 ns	5333.3 ns
80 MHz	12.5 ns	3205.1 ns
120 MHz	8.3 ns	2128.2 ns

可以使用 TSC_CR 寄存器中的 SSE 位禁止/使用扩频功能。也可以根据器件 HCLK 时钟频率和所选电荷转移频率使用 TSC_CR 寄存器中的 SSPSC 和 SSD[6:0] 配置频率偏差。

30.3.6 最大计数错误

最大计数错误可防止因电容感应通道发生故障而导致采集时间较长。它包括为模拟 I/O 组计数器指定最大计数值。最大计数值可使用 TSC_CR 寄存器中的 MCV[2:0] 位进行指定。采集组计数器达到此最大值后，正在进行的采集操作将停止，并且采集结束标志（EOAF 位）和最大计数错误标志（MCEF 位）都会置 1。如果相应的采集结束（EOAIE 位）或/和最大计数错误（MCEIE 位）中断使能位都置 1，还可以产生中断。

30.3.7 采样电容 I/O 和通道 I/O 模式选择

要实现由触摸感应控制器控制 GPIO，必须通过标准 GPIO 寄存器和 GPIOxAFR 寄存器使能相应的复用功能。

GPIO 模式由 TSC 控制，并使用 TSC_IOSCR 和 TSC_IOCCR 寄存器进行定义。

当未执行任何采集操作时，由触摸感应控制器控制的所有 I/O 都处于默认状态。当执行采集操作时，仅未使用的 I/O（未定义为采样电容 I/O 或通道 I/O）处于默认状态。TSC_CR 寄存器中的 IODEF 位定义处于默认状态的 I/O 的配置。下表根据 I/O 模式对其配置进行了汇总。

表 188. I/O 状态（取决于其模式和 IODEF 位值）

IODEF 位	采集状态	未使用的 I/O 模式	通道 I/O 模式	采样电容 I/O 模式
0 （推挽输出处于低电平）	未执行	推挽输出 处于低电平	推挽输出 处于低电平	推挽输出 处于低电平
0 （推挽输出处于低电平）	正在执行	推挽输出 处于低电平	-	-
1 （输入浮空）	未执行	输入浮空	输入浮空	输入浮空
1 （输入浮空）	正在执行	输入浮空	-	-

未使用的 I/O 模式

未使用的 I/O 对应于由 TSC 外设控制的 GPIO，但未定义为电极 I/O 或采样电容 I/O。

采样电容 I/O 模式

要实现由 TSC 外设控制采样电容 I/O，必须首先将相应的 GPIO 设置为复用输出开漏模式，然后必须将 TSC_IOSCR 寄存器中相应的 Gx_IOy 位置 1。

每个模拟 I/O 组在同一时间必须仅使能一个采样电容。

通道 I/O 模式

要实现由 TSC 外设控制通道 I/O，必须首先将相应的 GPIO 设置为复用输出推挽模式，然后必须将 TSC_IOCCR 寄存器中的相应 Gx_IOy 位置 1。

对于要求更高等效电极表面的接近检测或为了加快采集过程，可以使能属于同一模拟 I/O 组的多个通道并同时进行采集。

注： *在采集阶段，即使未使能 TSC 外设复用功能，只要 TSC_IOSCR 或 TSC_IOCCR 位置 1，相应的 GPIO 模拟开关就由触摸感应控制器自动进行控制。*

30.3.8 采集模式

触摸感应控制器具有两种采集模式：

- 正常采集模式：TSC_CR 寄存器中的 START 位置 1 后，采集会立即开始。
- 同步采集模式：通过将 TSC_CR 寄存器中的 START 位置 1 来使能采集，但仅当检测到下降沿或上升沿并且 SYNC 输入引脚处于高电平时，采集才会开始。此模式可用于将电容感应通道采集与外部信号同步，而不会额外增加 CPU 负载。

TSC_IOGCSR 寄存器中的 GxE 位用于指定使能哪个模拟 I/O 组（相对计数器正在计数）。已禁止模拟 I/O 组的 C_S 电压不会受到监视，而且该组不会参与触发采集结束标志。但是，如果已禁止模拟 I/O 组包含一些通道，这些通道将被脉冲化。

已使能的模拟 I/O 组的 C_S 电压达到给定阈值时，TSC_IOGCSR 寄存器中相应的 GxS 位置 1。当已使能的所有模拟 I/O 组的采集完成时（已使能的所有模拟 I/O 组的所有 GxS 位都置 1），TSC_ISR 寄存器中的 EOAF 标志会置 1。如果 TSC_IER 寄存器中的 EOAIE 位置 1，则会产生中断请求。

如果检测到最大计数错误，正在进行的采集将会停止，并且 TSC_ISR 寄存器中的 EOAF 和 MCEF 标志都置 1。如果相应位（TSCIER 寄存器的 EOAIE 和 MCEIE 位）置 1，可以针对这两种事件产生中断请求。注意，当检测到最大计数错误时，已使能模拟 I/O 组中的其余 GxS 位不会置 1。

要清除中断标志，TSC_ICR 寄存器中相应的 EOAIC 和 MCEIC 位必须置 1。

当开始新的采集时，模拟 I/O 组计数器清零。采集完成时，这些计数器会更新为在相应通道上产生的电荷转移周期数。

30.3.9 I/O 滞后和模拟开关控制

为提高灵活性，触摸感应控制器还可用于控制每个 Gx_IOy 的施密特触发器滞后和模拟开关。在触摸感应控制器使能后，无论 I/O 控制模式是由标准 GPIO 寄存器还是由其他外设进行控制，此控制均可用。该控制还可用于执行不同的采集序列或用于其他用途。

为提高系统的抗扰性，必须通过复位 TSC_IOHCR 寄存器中相应的 Gx_IOy 位来禁止由 TSC 控制的 GPIO 施密特触发器滞后。

30.4 TSC 低功耗模式

表 189. 低功耗模式对 TSC 的作用

模式	说明
睡眠	无影响。外设中断可使器件退出睡眠模式。
低功耗运行	无影响。
低功耗睡眠	无影响。外设中断可使器件退出低功耗睡眠模式。
停止 0/停止 1	外设寄存器内容仍被保持。
停止 2	
待机	下电。在退出待机或关断模式后，必须重新初始化外设。
关断	

30.5 TSC 中断

表 190. 中断控制位

中断事件	使能控制位	事件标志	标志清零位	退出 睡眠模式	退出 停止模式	退出 待机模式
采集结束	EOAIE	EOAIF	EOAIC	是	否	否
最大计数错误	MCEIE	MCEIF	MCEIC	是	否	否

30.6 TSC 寄存器

有关寄存器说明中使用的缩写，请参见参考手册中的*第 72 页的第 1.1 节*。

外设寄存器可按字（32 位）进行访问。

30.6.1 TSC 控制寄存器 (TSC_CR)

TSC control 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
CTPH[3:0]				CTPL[3:0]				SSD[6:0]							SSE
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
SSPSC	PGPSC[2:0]			Res.	Res.	Res.	Res.	MCV[2:0]			IODEF	SYNC POL	AM	START	TSCE
rw	rw	rw	rw					rw	rw	rw	rw	rw	rw	rw	rw

位 31:28 **CTPH[3:0]**：电荷转移脉冲处于高电平 (Charge transfer pulse high)

这些位将由软件置 1 和清零。它们定义电荷转移脉冲处于高电平状态的持续时间（C_X 充电）。

0000：1x t_{PGCLK}
0001：2x t_{PGCLK}
...
1111：16x t_{PGCLK}

注： 正在进行采集时不得修改这些位。

位 27:24 **CTPL[3:0]**：电荷转移脉冲处于低电平 (Charge transfer pulse low)

这些位将由软件置 1 和清零。它们定义电荷转移脉冲处于低电平状态的持续时间（将电荷从 C_X 转移到 C_S）。

0000：1x t_{PGCLK}
0001：2x t_{PGCLK}
...
1111：16x t_{PGCLK}

注： 正在进行采集时不得修改这些位。

注： 有些配置禁止采用。有关详细信息，请参见第 30.3.4 节：电荷转移采集序列。

位 23:17 **SSD[6:0]**：扩频偏差 (Spread spectrum deviation)

这些位将由软件置 1 和清零。它们定义扩频偏差，扩频偏差是由将可变数量的 SSCLK 时钟周期添加到电荷转移脉冲高电平状态产生的。

0000000：1x t_{SSCLK}
0000001：2x t_{SSCLK}
...
1111111：128x t_{SSCLK}

注： 正在进行采集时不得修改这些位。

位 16 **SSE**：扩频使能 (Spread spectrum enable)

此位由软件置 1 和清零，用于使能/禁止扩频功能。

0：禁止扩频

1：使能扩频

注： 正在进行采集时不得修改该位。

位 15 **SSPSC**：扩频预分频器 (Spread spectrum prescaler)

此位由软件置 1 和清零。它选择用于生成扩频时钟 (SSCLK) 的 AHB 时钟分频器。

0：f_{HCLK}

1：f_{HCLK} /2

注： 正在进行采集时不得修改该位。

位 14:12 **PGPSC[2:0]**：脉冲发生器预分频器 (pulse generator prescaler)

这些位由软件置 1 和清零。它们选择用于生成脉冲发生器时钟 (PGCLK) 的 AHB 时钟分频器。

000：f_{HCLK}

001：f_{HCLK} /2

010：f_{HCLK} /4

011：f_{HCLK} /8

100：f_{HCLK} /16

101：f_{HCLK} /32

110：f_{HCLK} /64

111：f_{HCLK} /128

注： 正在进行采集时不得修改这些位。

注： 有些配置禁止采用。有关详细信息，请参见第 30.3.4 节：电荷转移采集序列。

位 11:8 保留，必须保持复位值。

位 7:5 **MCV[2:0]**：最大计数值 (Max count value)

这些位将由软件置 1 和清零。它们定义在产生最大计数错误之前可以生成的电荷转移脉冲的最大数量。

000：255

001：511

010：1023

011：2047

100：4095

101：8191

110：16383

111：保留

注： 正在进行采集时不得修改这些位。

位 4 **IODEF**：I/O 默认模式 (I/O Default mode)

此位由软件置 1 和清零。它定义当未执行任何采集时所有 TSC I/O 的配置。当存在正在执行的采集时，它定义所有未使用 I/O（未定义为采样电容 I/O 或通道 I/O）的配置。

0：强制 I/O 处于推挽输出低电平状态

1：I/O 处于输入浮空状态

注： 正在进行采集时不得修改该位。

位 3 **SYNCPOL**：同步引脚极性 (Synchronization pin polarity)

此位由软件置 1 和清零，以选择同步输入引脚的极性。

0：仅下降沿

1：上升沿和高电平

位 2 **AM**：采集模式 (Acquisition mode)

此位由软件置 1 和清零，用以选择采集模式。

0：正常采集模式（START 位置 1 后，采集立即开始）

1：同步采集模式（如果 START 位置 1，并在 SYNC 输入引脚上检测到选定的信号，则采集开始）

注： 正在进行采集时不得修改该位。

位 1 **START**：启动新的采集 (Start a new acquisition)

此位由软件置 1，用于启动新的采集。采集完成后，此位会由硬件立即清零，也可由软件将此位清零以取消正在执行的采集。

0：不启动采集

1：启动新的采集

位 0 **TSCE**：触摸感应控制器使能 (Touch sensing controller enable)

此位由软件置 1 和清零，用于使能/禁止触摸感应控制器。

0：禁止触摸感应控制器

1：使能触摸感应控制器

注： 禁止触摸感应控制器后，TSC 寄存器设置不起任何作用。

30.6.2 TSC 中断使能寄存器 (TSC_IER)

TSC interrupt enable 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
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.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MCEIE	EOAIE
														rw	rw

位 31:2 保留，必须保持复位值。

位 1 **MCEIE**：最大计数错误中断使能 (Max count error interrupt enable)

该位由软件置 1 和清零，用于使能/禁止最大计数错误中断。

0：禁止最大计数错误中断

1：使能最大计数错误中断

位 0 **EOAIE**：采集结束中断使能 (End of acquisition interrupt enable)

该位由软件置 1 和清零，用于使能/禁止采集结束中断。

0：禁止采集结束中断

1：使能采集结束中断

30.6.3 TSC 中断清零寄存器 (TSC_ICR)

TSC interrupt 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
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.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MCEIC	EOAIC
														rw	rw

位 31:2 保留，必须保持复位值。

位 1 **MCEIC**：最大计数错误中断清零 (Max count error interrupt clear)

该位由软件置 1，用于清除最大计数错误标志。该标志复位后，该位由硬件清零。写入“0”则不会带来任何影响。

0：无影响

1：将 TSC_ISR 寄存器中相应的 MCEF 清零

位 0 **EOAIC**：采集结束中断清零 (End of acquisition interrupt clear)

该位由软件置 1，用于清除采集结束标志。该标志复位后，该位由硬件清零。写入“0”则不会带来任何影响。

0：无影响

1：将 TSC_ISR 寄存器的相应 EOAF 清零

30.6.4 TSC 中断状态寄存器 (TSC_ISR)

TSC interrupt status 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
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.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MCEF	EOAF
														r	r

位 31:2 保留，必须保持复位值。

位 1 **MCEF**：最大计数错误标志 (Max count error flag)

模拟 I/O 组计数器达到指定的最大计数值时，该位由硬件置 1。由软件向 TSC_ICR 寄存器的 MCEIC 位写入 1，可将此位清零。

0：未检测到最大计数错误 (MCE)

1：检测到最大计数错误 (MCE)

位 0 **EOAF**：采集结束标志 (End of acquisition flag)

当已使能的所有组的采集完成时（已使能的所有模拟 I/O 组的所有 GxS 位都置 1，或检测到最大计数错误），该位由硬件置 1。由软件向 TSC_ICR 寄存器的 EOAIC 位写入 1，可将此位清零。

0：采集正在执行或未启动

1：采集完成

30.6.5 TSC I/O 滞后控制寄存器 (TSC_IOHCR)

TSC I/O hysteresis control register

偏移地址：0x10

复位值：0xFFFF FFFF

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
G8_IO4	G8_IO3	G8_IO2	G8_IO1	G7_IO4	G7_IO3	G7_IO2	G7_IO1	G6_IO4	G6_IO3	G6_IO2	G6_IO1	G5_IO4	G5_IO3	G5_IO2	G5_IO1
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
G4_IO4	G4_IO3	G4_IO2	G4_IO1	G3_IO4	G3_IO3	G3_IO2	G3_IO1	G2_IO4	G2_IO3	G2_IO2	G2_IO1	G1_IO4	G1_IO3	G1_IO2	G1_IO1
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:0 **Gx_IOy**：Gx_IOy 施密特触发器滞后模式 (Gx_IOy Schmitt trigger hysteresis mode)

这些位由软件置 1 和清零，用于使能/禁止 Gx_IOy 施密特触发器滞后。

0：禁止 Gx_IOy 施密特触发器滞后

1：使能 Gx_IOy 施密特触发器滞后

注：　这些位控制 I/O 施密特触发器滞后，无论 I/O 控制模式如何（即使由标准 GPIO 寄存器进行控制也是如此）。

30.6.6 TSC I/O 模拟开关控制寄存器 (TSC_IOASCR)

TSC I/O analog switch control 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
G8_IO4	G8_IO3	G8_IO2	G8_IO1	G7_IO4	G7_IO3	G7_IO2	G7_IO1	G6_IO4	G6_IO3	G6_IO2	G6_IO1	G5_IO4	G5_IO3	G5_IO2	G5_IO1
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
G4_IO4	G4_IO3	G4_IO2	G4_IO1	G3_IO4	G3_IO3	G3_IO2	G3_IO1	G2_IO4	G2_IO3	G2_IO2	G2_IO1	G1_IO4	G1_IO3	G1_IO2	G1_IO1
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:0 **Gx_IOy**：Gx_IOy 模拟开关使能 (Gx_IOy analog switch enable)

这些位由软件置 1 和清零，用于使能/禁止 Gx_IOy 模拟开关。

0：禁止 Gx_IOy 模拟开关（断开）

1：使能 Gx_IOy 模拟开关（闭合）

注：　这些位控制 I/O 模拟开关，无论 I/O 控制模式如何（即使由标准 GPIO 寄存器进行控制也是如此）。

30.6.7 TSC I/O 采样控制寄存器 (TSC_IOSCR)

TSC I/O sampling control 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
G8_IO4	G8_IO3	G8_IO2	G8_IO1	G7_IO4	G7_IO3	G7_IO2	G7_IO1	G6_IO4	G6_IO3	G6_IO2	G6_IO1	G5_IO4	G5_IO3	G5_IO2	G5_IO1
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
G4_IO4	G4_IO3	G4_IO2	G4_IO1	G3_IO4	G3_IO3	G3_IO2	G3_IO1	G2_IO4	G2_IO3	G2_IO2	G2_IO1	G1_IO4	G1_IO3	G1_IO2	G1_IO1
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:0 **Gx_IOy**：Gx_IOy 采样模式 (Gx_IOy sampling mode)

这些位由软件置 1 和清零，用于将 Gx_IOy 配置为采样电容 I/O。每个模拟 I/O 组必须仅有一个 I/O 定义为采样电容。

0：Gx_IOy 未使用

1：Gx_IOy 用作采样电容

注：　正在进行采集时不得修改这些位。

在采集阶段，即使未使能 TSC 外设复用功能，只要 TSC_IOSCR 位置 1，相应的 GPIO 模拟开关就由触摸感应控制器自动进行控制。

30.6.8 TSC I/O 通道控制寄存器 (TSC_IOCCR)

TSC I/O channel control register

偏移地址：0x28

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
G8_IO4	G8_IO3	G8_IO2	G8_IO1	G7_IO4	G7_IO3	G7_IO2	G7_IO1	G6_IO4	G6_IO3	G6_IO2	G6_IO1	G5_IO4	G5_IO3	G5_IO2	G5_IO1
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
G4_IO4	G4_IO3	G4_IO2	G4_IO1	G3_IO4	G3_IO3	G3_IO2	G3_IO1	G2_IO4	G2_IO3	G2_IO2	G2_IO1	G1_IO4	G1_IO3	G1_IO2	G1_IO1
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:0 **Gx_IOy**：Gx_IOy 通道模式 (Gx_IOy channel mode)

这些位由软件置 1 和清零，用于将 Gx_IOy 配置为通道 I/O。

0：Gx_IOy 未使用

1：Gx_IOy 用作通道

注： 正在进行采集时不得修改这些位。

在采集阶段，即使未使能 TSC 外设复用功能，只要 TSC_IOCCR 位置 1，相应的 GPIO 模拟开关就由触摸感应控制器自动进行控制。

30.6.9 TSC I/O 组控制状态寄存器 (TSC_IOGCSR)

TSC I/O group control status register

偏移地址：0x30

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	G8S	G7S	G6S	G5S	G4S	G3S	G2S	G1S
								r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	G8E	G7E	G6E	G5E	G4E	G3E	G2E	G1E
								rw	rw	rw	rw	rw	rw	rw	rw

位 31:24 保留，必须保持复位值。

位 23:16 **GxS**：模拟 I/O 组 x 状态 (Analog I/O group x status)

当相应已使能模拟 I/O 组 x 上的采集完成时，这些位由硬件置 1。当新的采集开始时，这些位由硬件清零。

0：模拟 I/O 组 x 上的采集正在执行或未启动

1：模拟 I/O 组 x 上的采集已完成

注： 检测到最大计数错误后，已使能模拟 I/O 组的其余 GxS 位不置 1。

位 15:8 保留，必须保持复位值。

位 7:0 **GxE**：模拟 I/O 组 x 使能 (Analog I/O group x enable)

这些位由软件置 1 和清零，用于使能/禁止相应模拟 I/O 组 x 上的采集（计数器正在计数）。

0：禁止模拟 I/O 组 x 上的采集

1：使能模拟 I/O 组 x 上的采集

30.6.10 TSC I/O 组 x 计数器寄存器 (TSC_IOGxCR)

TSC I/O group x counter register

x 表示模拟 I/O 组编号。

偏移地址：0x30 + 0x04 * x，(x = 1..8)

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.	CNT[13:0]													
		r	r	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 13:0 **CNT[13:0]**：计数器值 (Counter value)

这些位表示为完成采集（C_S 两端电压已达到阈值）在模拟 I/O 组 x 上产生的电荷转移周期数。

30.6.11 TSC 寄存器映射

表 191. TSC 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
0x0000	**TSC_CR**	CTPH[3:0]				CTPL[3:0]				SSD[6:0]							SSE	SSPSC	PGPSC[2:0]			Res.	Res.	Res.	Res.	MCV [2:0]			IODEF	SYNCPOL	AM	START	TSCE
	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
0x0004	**TSC_IER**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MCEIE	EOAIE
	Reset value																															0	0
0x0008	**TSC_ICR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MCEIC	EOAIC
	Reset value																															0	0
0x000C	**TSC_ISR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MCEF	EOAF
	Reset value																															0	0
0x0010	**TSC_IOHCR**	G8_IO4	G8_IO3	G8_IO2	G8_IO1	G7_IO4	G7_IO3	G7_IO2	G7_IO1	G6_IO4	G6_IO3	G6_IO2	G6_IO1	G5_IO4	G5_IO3	G5_IO2	G5_IO1	G4_IO4	G4_IO3	G4_IO2	G4_IO1	G3_IO4	G3_IO3	G3_IO2	G3_IO1	G2_IO4	G2_IO3	G2_IO2	G2_IO1	G1_IO4	G1_IO3	G1_IO2	G1_IO1
	Reset value	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0x0014	Reserved																																
0x0018	**TSC_IOASCR**	G8_IO4	G8_IO3	G8_IO2	G8_IO1	G7_IO4	G7_IO3	G7_IO2	G7_IO1	G6_IO4	G6_IO3	G6_IO2	G6_IO1	G5_IO4	G5_IO3	G5_IO2	G5_IO1	G4_IO4	G4_IO3	G4_IO2	G4_IO1	G3_IO4	G3_IO3	G3_IO2	G3_IO1	G2_IO4	G2_IO3	G2_IO2	G2_IO1	G1_IO4	G1_IO3	G1_IO2	G1_IO1
	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
0x001C	Reserved																																
0x0020	**TSC_IOSCR**	G8_IO4	G8_IO3	G8_IO2	G8_IO1	G7_IO4	G7_IO3	G7_IO2	G7_IO1	G6_IO4	G6_IO3	G6_IO2	G6_IO1	G5_IO4	G5_IO3	G5_IO2	G5_IO1	G4_IO4	G4_IO3	G4_IO2	G4_IO1	G3_IO4	G3_IO3	G3_IO2	G3_IO1	G2_IO4	G2_IO3	G2_IO2	G2_IO1	G1_IO4	G1_IO3	G1_IO2	G1_IO1
	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
0x0024	Reserved																																
0x0028	**TSC_IOCCR**	G8_IO4	G8_IO3	G8_IO2	G8_IO1	G7_IO4	G7_IO3	G7_IO2	G7_IO1	G6_IO4	G6_IO3	G6_IO2	G6_IO1	G5_IO4	G5_IO3	G5_IO2	G5_IO1	G4_IO4	G4_IO3	G4_IO2	G4_IO1	G3_IO4	G3_IO3	G3_IO2	G3_IO1	G2_IO4	G2_IO3	G2_IO2	G2_IO1	G1_IO4	G1_IO3	G1_IO2	G1_IO1
	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
0x002C	Reserved																																
0x0030	**TSC_IOGCSR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	G8S	G7S	G6S	G5S	G4S	G3S	G2S	G1S	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	G8E	G7E	G6E	G5E	G4E	G3E	G2E	G1E
	Reset value									0	0	0	0	0	0	0	0									0	0	0	0	0	0	0	0
0x0034	**TSC_IOG1CR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CNT[13:0]													
	Reset value																			0	0	0	0	0	0	0	0	0	0	0	0	0	0

表 191. TSC 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
0x0038	**TSC_IOG2CR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CNT[13:0]													
	Reset value																			0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x003C	**TSC_IOG3CR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CNT[13:0]													
	Reset value																			0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0040	**TSC_IOG4CR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CNT[13:0]													
	Reset value																			0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0044	**TSC_IOG5CR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CNT[13:0]													
	Reset value																			0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0048	**TSC_IOG6CR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CNT[13:0]													
	Reset value																			0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x004C	**TSC_IOG7CR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CNT[13:0]													
	Reset value																			0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0050	**TSC_IOG8CR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CNT[13:0]													
	Reset value																			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*第 78 页的第 2.2.2 节*。

31 真随机数发生器 (RNG)

31.1 简介

RNG 是一个真随机数发生器，可向应用程序提供作为 32 位采样的全熵输出，它由一个实时熵源（模拟）和一个内部调节组件构成。

RNG 可作为一个实时熵源用来构建符合 NIST 要求的确定性随机位发生器 (DRBG)。

RNG 真随机数发生器已按照德国 AIS-31 标准进行了验证。

31.2 RNG 主要特性

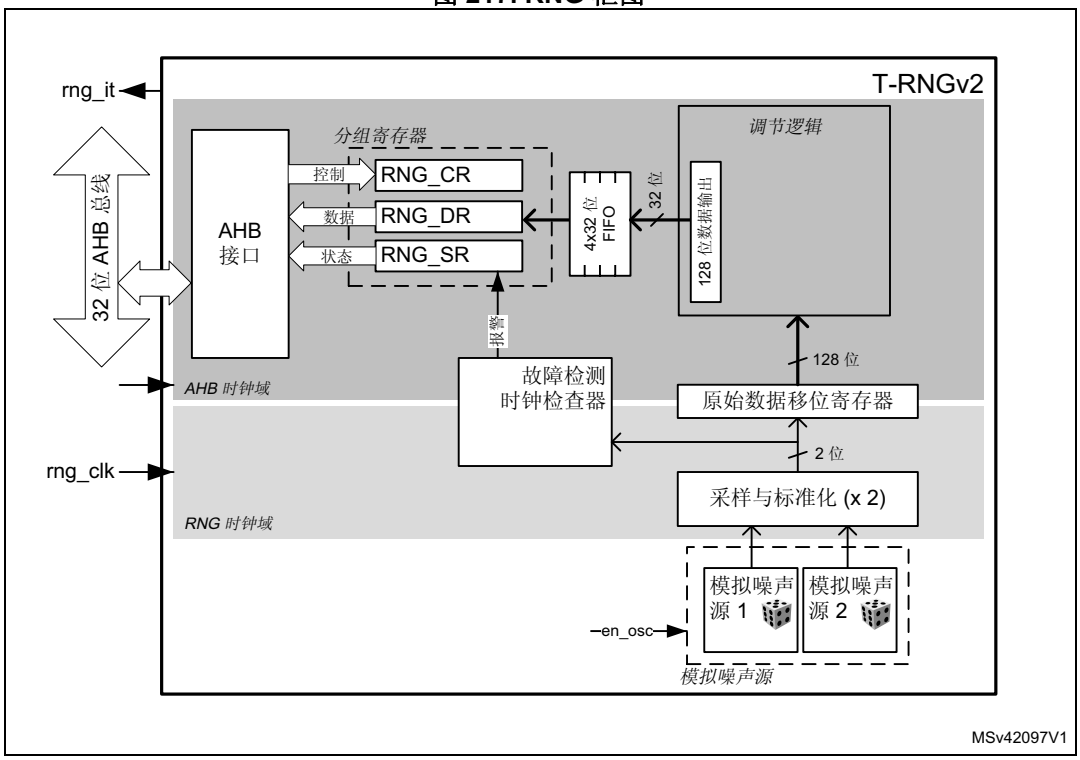
- RNG 提供的 32 位真随机数由模拟熵源生成并通过高质量调节级进行处理。
- RNG 按照 AIS-31 预定义类 PTG.2 评估方法进行验证，该评估方法属于德国通用标准 (CC) 方案的组成部分。
- 如果值大于 213 个周期（否则为 213 个周期），则每 16x $\frac{f_{AHB}}{f_{RNG}}$ AHB 个时钟周期生成四个 32 位随机采样。
- 支持内置连续基本运行状态测试及相关错误管理。
  - 包括过低采样时钟检测和重复计数测试。
- 可被禁止以降低功耗。
- 具有 AMBA AHB 从外设，只能通过 32 位字进行单次访问（否则会针对写访问产生 AHB 总线错误），请务必注意！任何不等于 32 位的写操作都可能破坏寄存器内容。

31.3 RNG 功能说明

31.3.1 RNG 框图

图 217 显示了 RNG 框图。

图 217. RNG 框图



31.3.2 RNG 内部信号

表 192 中列出了 RNG 级而非 STM32 产品级（焊盘上）所提供的内部信号，对这些信号有所了解会很有用。

表 192. RNG 内部输入/输出信号

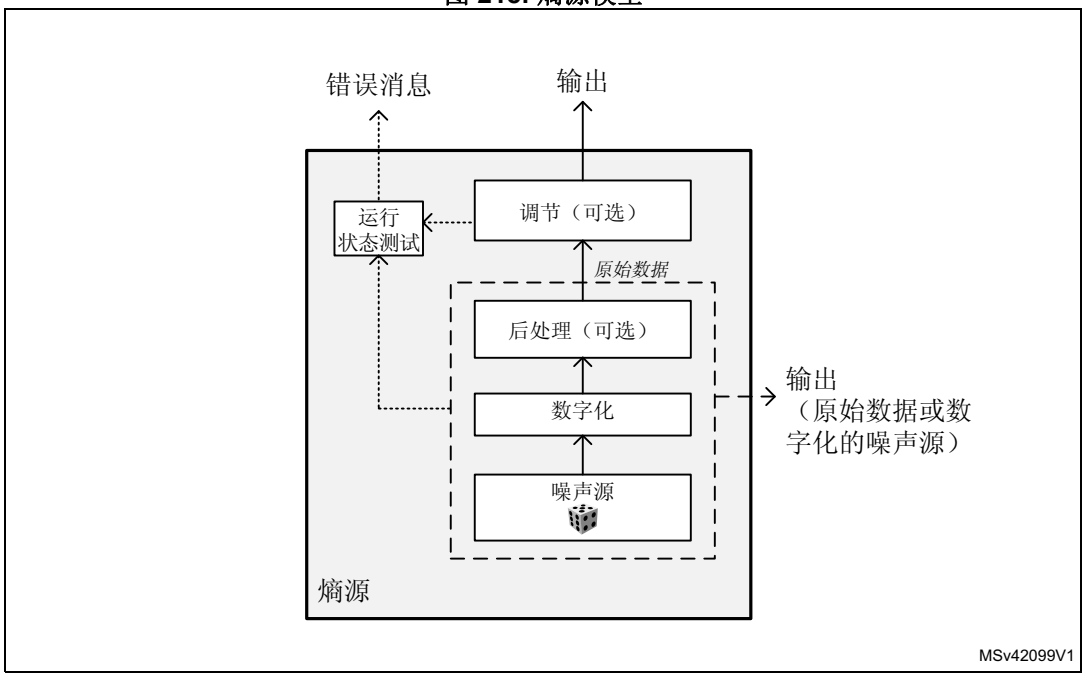
信号名称	信号类型	说明
rng_it	数字输出	RNG 全局中断请求
rng_hclk2	数字输入	AHB2 时钟
RNG_CLK	数字输入	RNG 专用时钟，与 rng_hclk2 异步

31.3.3 随机数生成

真随机数发生器 (RNG) 按确定的间隔通过其 AHB 接口提供真随机数。RNG 可实现*图 218*所示的熵源模型，并为应用程序提供三个主要功能：

- 采集熵源的位串输出
- 获取噪声源采样，用于验证目的
- 采集连续运行状态测试得到的错误消息

图 218. 熵源模型



RNG 的主要组件包括：

- 物理随机源（模拟噪声源）
- 模拟噪声源的数字化处理级
- 模拟噪声源的加密处理级
- 输出缓冲区，用于缓冲熵源输出和噪声源采样
- 运行状态监视模块，用于对整个熵源执行测试

下文对所有组件进行了详细介绍。

噪声源：

噪声源组件包含不确定性的、能够提供熵值的活动，其最终决定了由熵源产生的输出位串的不确定性。它包括：

- 两个模拟噪声源，每个噪声源由三个自由运行的环形振荡器输出异或 (XOR) 而成。可以禁止这些模拟振荡器以实现节能，如*第 31.4 节：RNG 低功耗使用*所述。
- 这些输出的采样级由专用时钟输入 (**rng_clk**) 提供时钟信号，输出 2 位原始数据。

另外，该噪声源采样与 AHB 接口时钟频率 (**rng_hclk**) 无关。

注： *第 31.7 节：熵源验证中提供了建议使用的 RNG 时钟频率。*

后期处理

从真随机噪声源获取的采样值由 2 位的位串组成。由于该噪声源输出有偏差，因此 RNG 会利用后期处理组件将偏差降至可接受的水平。

更具体来讲，对于每个噪声源产生的 2 位数值，一位是由 RNG 取自采样噪声源，另一位是对取自采样噪声源的值取反。因此，如果噪声源生成的“1”多于“0”（或者相反），则噪声源会被过滤掉。

调节

RNG 中的调节组件是确定性函数，可以提高生成固定长度位串输出（128 位）的熵率。

注： *第 31.6 节：RNG 处理时间中介绍了 RNG 初始化过程中的延时。*

另外请注意，如果已经接收了 32 或更多位原始数据，并且输出 FIFO 需要重新填充，则会触发后期处理计算。因此，当 RNG 128 位 FIFO 在 64 个 RNG 时钟周期后被应用程序清空时，RNG 输出熵为最大值。

*第 31.6 节：RNG 处理时间*中给出了生成两个随机数之间和 RNG 初始化之间所需的时间，以及第一个采样的可用性的说明。

调节组件由更快的 AHB 时钟提供时钟信号。

输出缓冲区

数据输出缓冲区最多可存储四个从调节组件输出的 32 位字。如果已通过 RNG_DR 寄存器从输出 FIFO 读取了四个字，则 128 位调节输出寄存器的内容会被推送到输出 FIFO 中，同时自动启动新的一轮调节。213 个 AHB 时钟周期后，会将四个新字添加到调节输出寄存器。

当通过 RNG_DR 寄存器可获取随机数时，DRDY 标志就从“0”变为“1”。该标志会保持在置 1 状态，直至从 RNG_DR 寄存器读取四个字后输出缓冲区变空。

注： *如果使能了中断，则当该数据就绪标志从“0”变为“1”时，会产生中断。随后，中断会自动由 RNG 清零，如上所述。*

运行状态检查

该组件可确保整个熵源（包括其噪声源）正常启动并按预期运行，能够快速准确地定位故障，从而提供了高可靠性保证。

RNG 可实现以下运行状态检查功能：

1. 行为测试，可在熵源*运行时*进行
   - 重复计数测试，在以下情况时会指示发生错误：
   a) 其中一个噪声源持续不变地输出了 64 位或以上的“0”或“1”，或者 32 个或以上的“01”或“10”。
   b) 两个噪声源都持续不变地输出了 32 位或以上的“0”或“1”，或者 16 个或以上的“01”或“10”。
2. 厂商专用连续测试
   - 实时“过慢”采样时钟检测器，如果一个 RNG 时钟周期小于 AHB 时钟周期的 32 分之一，则该检测器会指示错误。

RNG_SR 寄存器中的 CECS 和 SECS 状态位会在检测到错误条件时进行指示，*第 31.3.7 节：错误管理*对此进行了详细说明。

注： *检测到错误时生成中断。*

31.3.4 RNG 初始化

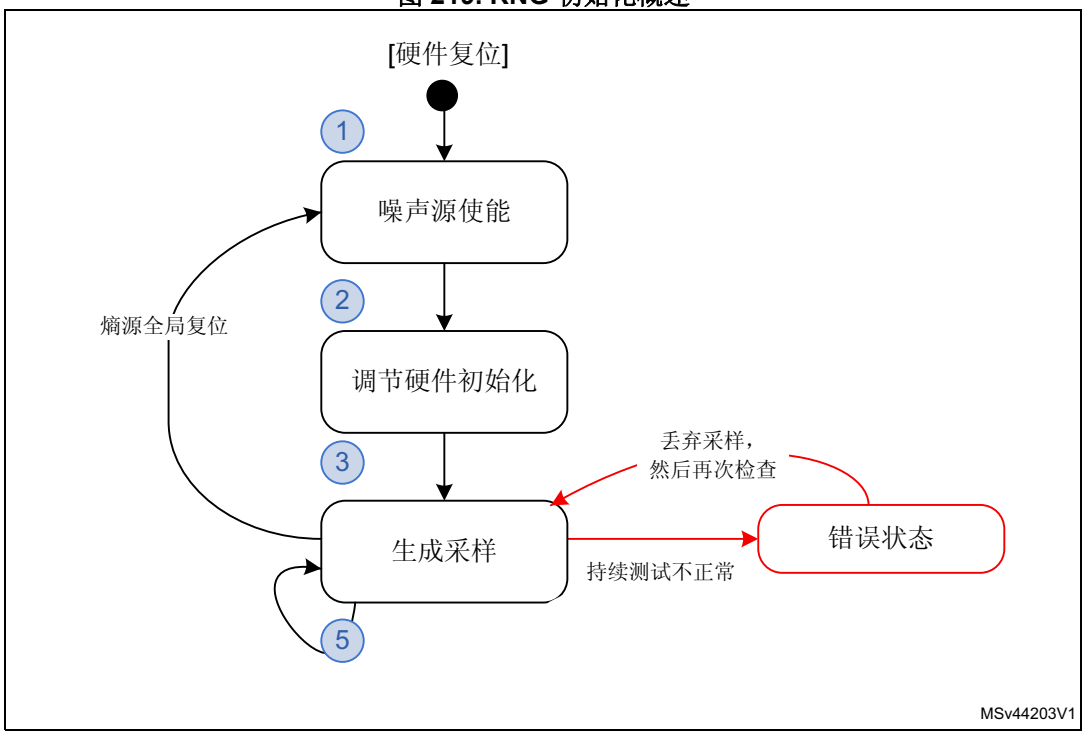
RNG 简化状态机如*图 219* 所示。

如果发生硬件复位，则会发生以下一系列事件：

1. 使能模拟噪声源，逻辑系统立即开始采样模拟输出并填充 128 位调节移位寄存器。
2. 使能调节逻辑并且使用两个 128 噪声源位初始化后期处理上下文。
3. 再次用 128 位数据填充调节级内部输入数据缓冲区，并执行一轮调节。然后使用后期处理结果填充输出缓冲区。
4. 输出缓冲区会按照 RNG 使用情况自动重填。

相关初始化时间请参见*第 31.6 节：RNG 处理时间*。

图 219. RNG 初始化概述



31.3.5 RNG 操作

正常工作

要使用中断运行 RNG，建议执行以下步骤：

1. 通过将 RNG_CR 寄存器中的 IE 位置 1 来使能中断。同时通过将 RNGEN 位置 1 来使能 RNG。
2. 这以后，准备好随机数时或出现错误时就产生中断。因此，每次发生中断时，应检查：
   - 未发生错误。RNG_SR 寄存器中的 SEIS 和 CEIS 位应设为“0”。
   - 随机数准备就绪。RNG_SR 寄存器中的 DRDY 位必须置“1”。
   - 如果满足以上两个条件，最多可连续四次读取 RNG_DR 寄存器的内容。如果调节输出缓冲区中存在有效数据，则应用程序可以额外读取四个字（此时 DRDY 位仍保持为 1）。如果上述条件中有一个或两个都无法满足，则不得读取 RNG_DR 寄存器。如果发生错误，则应使用*第 31.3.7 节*中介绍的错误恢复序列。

要在轮询模式下运行 RNG，建议执行以下步骤：

1. 通过将 RNG_CR 寄存器中的 RNGEN 位置“1”使能随机数生成。
2. 读取 RNG_SR 寄存器并检查：
   – 未发生错误（SEIS 和 CEIS 位应设为“0”）
   – 随机数准备就绪（DRDY 位应置“1”）
3. 如果满足以上两个条件，最多可连续四次读取 RNG_DR 寄存器的内容。如果调节输出缓冲区中存在有效数据，则应用程序可以额外读取四个字（此时 DRDY 位仍保持为 1）。如果上述条件中有一个或两个都无法满足，则不得读取 RNG_DR 寄存器。如果发生错误，则应使用*第 31.3.7 节*中介绍的错误恢复序列。

注： *如果数据未就绪（DRDY = “0”），则 RNG_DR 会返回 0。*

低功耗运行

如果应用程序比较关注功耗，则可以使用低功耗策略，如*第 960 页的第 31.4 节：RNG 低功耗使用*所述。

软件后期处理

无需进行特定的软件后期处理 / 调节即可满足 AIS-31 认证。如果需要安全强度为 128 位、符合 NIST 要求的 DRBG，则必须基于 RNG 真随机数发生器构建符合要求的随机数发生器软件。

*第 31.3.3 节：随机数生成*介绍了内置的运行状态检查功能。

31.3.6 RNG 时钟

RNG 在两个不同的时钟上运行：AHB 总线时钟和专用 RNG 时钟。

AHB 时钟用于为 AHB 分组寄存器和调节组件提供时钟信号。RNG 时钟用于噪声源采样。*第 31.7 节：熵源验证*中详细介绍了建议使用的时钟配置。

注意： 如果 RNG_CR 寄存器中的 CED 位置“0”，那么 RNG 时钟频率**必须大于** AHB 时钟频率的 32 分之一，否则时钟检查器会指示时钟错误（RNG_SR 寄存器中的 CECS 或 CEIS），RNG 将停止生成随机数。

详细信息参见*第 31.3.1 节：RNG 框图*（AHB 和 RNG 时钟域）。

31.3.7 错误管理

运行状态检查模块在生成随机数的同时还验证噪声源行为以及 RNG 源时钟频率是正常的，具体说明见本部分。另外，还描述了相关的错误状态。

时钟错误检测

如果时钟错误检测已使能 (CED = 0) 且 RNG 时钟频率过低，RNG 会停止生成随机数，并会将 **CEIS** 和 **CECS** 位都置“1”，指示发生了时钟错误。在这种情况下，应用程序应检查 RNG 时钟是否配置正确（参见*第 31.3.6 节：RNG 时钟*），随后必须将中断标志 CEIS 位清零。RNG 时钟正常运行后，CECS 将立即自动清零。

仅当 CECS 位置“0”时，RNG 才运行。但需要注意的是，时钟错误对之前生成的随机数没有影响，因此 RNG_DR 寄存器内容仍可使用。

噪声源错误检测

如果发生噪声源（或种子）错误，RNG 会停止生成随机数，并会将 **SEIS** 和 **SECS** 位都置“1”，指示发生了种子错误。如果 RNG_DR 寄存器中有可用值，不能使用该值，因为它可能没有足够的熵。如果在初始化阶段检测到错误，则 RNG 将自动重启整个初始化序列。

要从 RNG 初始化后的种子错误中完全恢复，应使用以下序列：

1. 向 SEIS 位写入“0”，将该位清零。
2. 从 RNG_DR 寄存器读取 12 个字，并将每个字丢弃，将管道清空。
3. 确认 SEIS 仍处于清零状态。随机数生成恢复正常。

31.4 RNG 低功耗使用

考虑到功耗问题，可在 DRDY 位置“1”后立即禁止 RNG，具体方法为将 RNG_CR 寄存器中的 RNGEN 位设为“0”。由于 RNGEN =“0”时后期处理逻辑和输出缓冲区仍处于工作状态，因此软件可使用以下功能：

- 如果输出缓冲区中存在有效字，仍可从 RNG_DR 寄存器中读取四个随机数。
- 如果调节输出内部寄存器中存在有效位，仍可从 RNG_DR 寄存器中额外读取四个随机数。如果不属于以上两种情况，则必须由应用程序重新使能 RNG，直至从噪声源采集了至少 32 个新的位并且完成了一轮完整的调节。这对应于 16 个 RNG 时钟周期来采样新的位，216 个 AHB 时钟周期运行一轮调节。

禁止 RNG 时，用户会禁用所有模拟种子发生器，各发生器的功耗列于器件手册中的电气特性部分。用户还会对所有由 RNG 时钟提供时钟信号的逻辑系统进行门控。请注意，由于 RNG 需要一定的时间进行初始化，因此该策略会在一段延时后才可在 RNG_DR 寄存器中提供随机采样。

如果 RNG 模块在初始化期间（也就是刚好在 DRDY 位首次置 1 之前）被禁止，则初始化序列将从 RNGEN 位置“1”时停止的位置恢复运行。

31.5 RNG 中断

在 RNG 中，发生以下事件时会产生中断：

- 数据就绪标志
- 种子错误，请参见*第 31.3.7 节：错误管理*
- 时钟错误，请参见*第 31.3.7 节：错误管理*

可以使用专用的中断使能控制位，如*表 193* 所示。

表 193. RNG 中断请求

中断事件	事件标志	使能控制位
数据就绪标志	DRDY	IE
种子错误标志	SEIS	IE
时钟错误标志	CEIS	IE

用户可以通过更改 RNG_CR 寄存器中的屏蔽位或通用中断控制位 IE 的方式分别使能或禁止上述中断源。可以从 RNG_SR 寄存器中读取各个中断源的状态。

注： *仅当 RNG 已使能时，才会产生中断。*

31.6 RNG 处理时间

如果值大于 213 个周期（否则为 213 个周期），则每 16x $\frac{f_{AHB}}{f_{RNG}}$ 个时钟周期，调节级可以生成四个 32 位随机数。

设备退出复位状态后需要更长的时间才能生成第一组随机数（请参见*第 31.3.4 节：RNG 初始化*）。事实上，首次使能 RNG 后，会在以下时间后首次提供随机数据：

- 128 RNG 个时钟周期 + 426 个 AHB 周期（$f_{AHB} < f_{threshold}$ 时）
- 192 RNG 个时钟周期 + 213 个 AHB 周期（$f_{AHB} \geq f_{threshold}$ 时）

其中 $f_{threshold} = (213 \times f_{RNG})/ 64$

31.7 熵源验证

31.7.1 简介

为了评估 RNG 可提供的熵大小，意法半导体按照 AIS-31 PTG.2 测试集对外设进行了测试。测试结果可根据需要提供，客户也可以使用 AIS 参考软件重现测量结果。客户还可以按照较早的 NIST SP800-22 测试集对 RNG 进行测试。

31.7.2 验证条件

意法半导体在以下条件下对 RNG 真随机数发生器进行了验证：

- RNG 时钟 rng_clk= 48 MHz（RNG_CR 寄存器中的 CED 位 =“0”）且 rng_clk = 400 kHz（RNG_CR 寄存器中的 CED 位 =“1”）。
- AHB 时钟 rng_hclk = 216 MHz

31.7.3 数据采集

为了运行统计测试，需要对熵源的原始数据和输出进行采样。

如果需要为产品检索上述采样，请联系意法半导体。

31.8 RNG 寄存器

RNG 与控制寄存器、数据寄存器和状态寄存器相关联。

31.8.1 RNG 控制寄存器 (RNG_CR)

RNG control register

偏移地址：0x000

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.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CED	Res.	IE	RNGEN	Res.	Res.
										rw		rw	rw		

位 31:6 保留，必须保持复位值。

位 5 **CED**：时钟错误检测 (Clock error detection)

0：使能时钟错误检测

1：禁止时钟错误检测

当使能 RNG 时，不能实时使能或禁止时钟错误检测，即必须禁止 RNG 才能使能或禁止 CED。

位 4 保留，必须保持复位值。

位 3 **IE**：中断使能 (Interrupt enable)

0：禁止 RNG 中断

1：使能 RNG 中断。如果 RNG_SR 寄存器中 DRDY = “1”、SEIS = “1”或 CEIS = “1”，则中断处于挂起状态。

位 2 **RNGEN**：真随机数发生器使能 (True random number generator enable)

0：禁止真随机数发生器。模拟噪声源关闭，并会对由 RNG 时钟提供时钟信号的逻辑系统进行门控。

1：使能真随机数发生器。

位 1:0 保留，必须保持复位值。

31.8.2 RNG 状态寄存器 (RNG_SR)

RNG status register

偏移地址：0x004

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.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SEIS	CEIS	Res.	Res.	SECS	CECS	DRDY
									rc_w0	rc_w0			r	r	r

位 31:7 保留，必须保持复位值。

位 6 **SEIS**：种子错误中断状态 (Seed error interrupt status)
此位与 SECS 同时置 1，通过向其写入“0”来清零。
0：未检测到错误序列
1：至少检测到一个错误序列。有关详细信息，请参见 **SECS** 位说明。
如果 RNG_CR 寄存器中 IE =“1”，则中断处于挂起状态。

位 5 **CEIS**：时钟错误中断状态 (Clock error interrupt status)
此位与 CECS 同时置 1，通过向其写入“0”来清零。
0：RNG 时钟正确 (fRNGCLK > fHCLK/32)
1：已检测到 RNG 时钟过慢 (fRNGCLK < fHCLK/32)
如果 RNG_CR 寄存器中 IE =“1”，则中断处于挂起状态。

位 4:3 保留，必须保持复位值。

位 2 **SECS**：当前状态种子错误 (Seed error current status)
0：目前未检测到错误序列。如果 SEIS 位置 1，则意味着已检测到错误序列并已恢复正常。
1：至少检测到一个以下错误序列：
– 其中一个噪声源持续不变地输出了 64 位或以上的“0”或“1”，或者 32 个或以上的“01”或“10”。
– 两个噪声源都持续不变地输出了 32 位或以上的“0”或“1”，或者 16 个或以上的“01”或“10”。

位 1 **CECS**：当前状态时钟错误 (Clock error current status)
0：RNG 时钟正确 (fRNGCLK > fHCLK/32)。如果 CEIS 位置 1，则意味着已检测到时钟过慢并已恢复正常。
1：RNG 时钟过慢 (fRNGCLK < fHCLK/32)。
注： 只有 RNG_CR 寄存器中的 CED 位置“0”时，CECS 位才有效。

位 0 **DRDY**：数据就绪 (Data ready)
0：RNG_DR 寄存器尚未有效，无可用随机数据
1：RNG_DR 寄存器包含有效随机数据
在输出缓冲区变空之后（读取 RNG_DR 寄存器之后），在生成新的随机值之前，该位返回“0”。
注： 当禁用外设时（RNG_CR 寄存器中的 RNGEN =“0”），可以将 DRDY 位置 1。
如果 RNG_CR 寄存器中的 IE =“1”，则在 DRDY =“1”时生成中断。

31.8.3 RNG 数据寄存器 (RNG_DR)

RNG data register

偏移地址：0x008

复位值：0x0000 0000

RNG_DR 寄存器是只读寄存器，在读取时提供 32 位随机数值。读取该寄存器后，如果输出 FIFO 为空，则该寄存器将在 216 个 AHB 时钟周期之后提供一个新的随机值。

当 DRDY = “1” 时，即使 RNGEN = “0”，该寄存器的内容也是有效的。

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

位 31:0 **RNDATA[31:0]**：随机数据 (Random data)

DRDY = “1” 时有效的 32 位随机数据。DRDY = “0” 时，RNDATA 值为 0。

31.8.4 RNG 寄存器映射

表 194 给出了 RNG 寄存器映射和复位值。

表 194. RNG 寄存器映射和复位值

偏移	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
0x000	RNG_CR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CED	Res.	IE	RNGEN	Res.	Res.
	Reset value																											0		0	0		
0x004	RNG_SR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SEIS	CEIS	Res.	Res.	SECS	CECS	DRDY
	Reset value																										0	0			0	0	0
0x008	RNG_DR	RNDATA[31:0]																															
	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

32 AES 硬件加速器 (AES)

32.1 简介

AES 硬件加速器 (AES) 使用完全符合联邦信息处理标准 (FIPS) 出版物 197 中规定的高级加密标准 (AES) 的算法来对数据进行加密或解密。

支持多种链接模式（ECB、CBC、CTR、GCM、GMAC、CCM），支持 128 或 256 位密钥大小。

AES 加速器为 32 位 AHB 外设。它支持对传入和传出数据进行 DMA 单次传输（需要两个 DMA 通道）。

AES 外设可为 STM32 加密库所实现的 AES 加密算法提供硬件加速。

AES 为 AMBA AHB 从外设，仅支持 32 位字单次访问（否则，会产生 AHB 总线错误）。32 位以外的写入可能损坏寄存器内容。

32.2 AES 主要特性

- 符合自 2001 年 11 月起的 NIST *“高级加密标准 (AES)，FIPS 出版物 197”*
- 128 位数据块处理
- 支持 128 位和 256 位密钥长度
- 基于多种链接模式的加密和解密：
  - 电子密码本 (ECB) 模式
  - 密码分组链接 (CBC) 模式
  - 计数器 (CTR) 模式
  - Galois 计数器模式 (GCM)
  - Galois 消息认证码 (GMAC) 模式
  - CBC-MAC 计数器 (CCM) 模式
- ECB 模式处理 128 位数据块的时候会有延迟，使用 128 位密钥或 256 位密钥分别延迟 51 或 75 个时钟周期
- 集成有密钥调度器，包括密钥分散阶段（仅限 ECB 或 CBC 解密）
- AMBA AHB 从外设，仅支持 32 位字单次访问
- 256 位寄存器，用于存储加密密钥（8 个 32 位寄存器）
- 128 位寄存器，用于存储初始化向量（4 个 32 位寄存器）
- 32 位缓冲器，用于数据输入和输出
- 采用自动数据流控制，支持单次传输直接存储器访问 (DMA)（使用两个通道，分别用于传入数据和已处理数据）
- 采用数据交换逻辑，支持 1 位、8 位、16 位或 32 位数据
- 在 AES 需要处理优先级更高的消息时，可通过软件将当前消息挂起，然后恢复原始消息

32.3 AES 实现

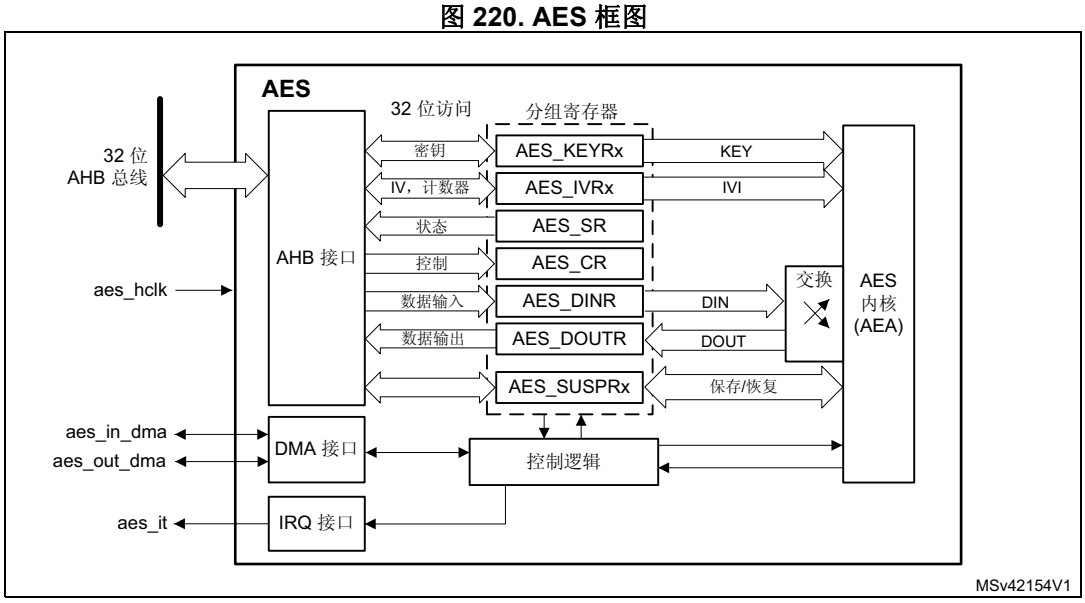
该器件具有单个AES外设实例。

32.4 AES 功能描述

32.4.1 AES 框图

图 220 显示了 AES 的框图。

图 220. AES 框图



32.4.2 AES 内部信号

表 195 描述了连接 AES 外设的用户相关内部信号。

表 195. AES 内部输入/输出信号

信号名称	信号类型	说明
aes_hclk	数字输入	AHB 总线时钟
aes_it	数字输出	AES中断请求
aes_in_dma	数字输入/输出	输入 DMA 单次请求/确认
aes_out_dma	数字输入/输出	输出 DMA 单次请求/确认

32.4.3 AES 加密内核

概述

AES 加密内核由以下部分组成：

- AES 算法 (AEA)
- 基于二元 Galois 域的乘法器 (GF2mul)
- 密钥输入
- 初始化向量 (IV) 输入
- 链接算法逻辑（XOR、反馈/计数器、屏蔽）

AES 内核适用于密钥长度为 128 位或 256 位的 128 位数据块（四字）。根据链接模式的不同，AES 可能需要一个 96 位初始化向量 IV（和一个 32 位计数器字段），也可能不需要。

AES 具有以下工作模式：

- **模式 1：**
  使用存储在 AES_KEYRx 寄存器中的密钥实现明文加密。
- **模式 2：**
  ECB 或 CBC 解密密钥准备。在选择 ECB 或 CBC 链接模式的模式 3 之前，必须先使用此模式。准备解密的密钥自动存储在 AES_KEYRx 寄存器中。现在，AES 外设准备好切换到模式 3 以执行数据解密。
- **模式 3：**
  使用存储在 AES_KEYRx 寄存器中的密钥实现密文解密。当选择 ECB 和 CBC 链接模式时，必须通过模式 2 预先准备好密钥。
- **模式 4：**
  使用存储在 AES_KEYRx 寄存器中的密钥实现 ECB 或 CBC 密文单次解密（初始密钥自动分散）。

注： *仅在执行 ECB 和 CBC 解密时使用模式 2 和模式 4。*

当选择模式 4 时，只能进行一次解密，因此推荐使用模式 2 和模式 3。

通过对 AES_CR 寄存器中的 MODE[1:0] 位域进行编程来选择工作模式。只能在 AES 外设已禁止时进行。

典型数据处理

AES 的典型用法如*第 973 页的第 32.4.4 节：执行密码操作的 AES 过程*中所述。

注： *中间 AEA 阶段的输出始终不会在加密边界外暴露，IVI 位域除外。*

链接模式

AES 支持以下链接模式，可通过 AES_CR 寄存器的 CHMOD[2:0] 位域选择：

- 电子密码本 (ECB)
- 密码分组链接 (CBC)
- 计数器 (CTR)
- Galois 计数器模式 (GCM)
- Galois 消息认证码 (GMAC)
- CBC-MAC 计数器模式 (CCM)

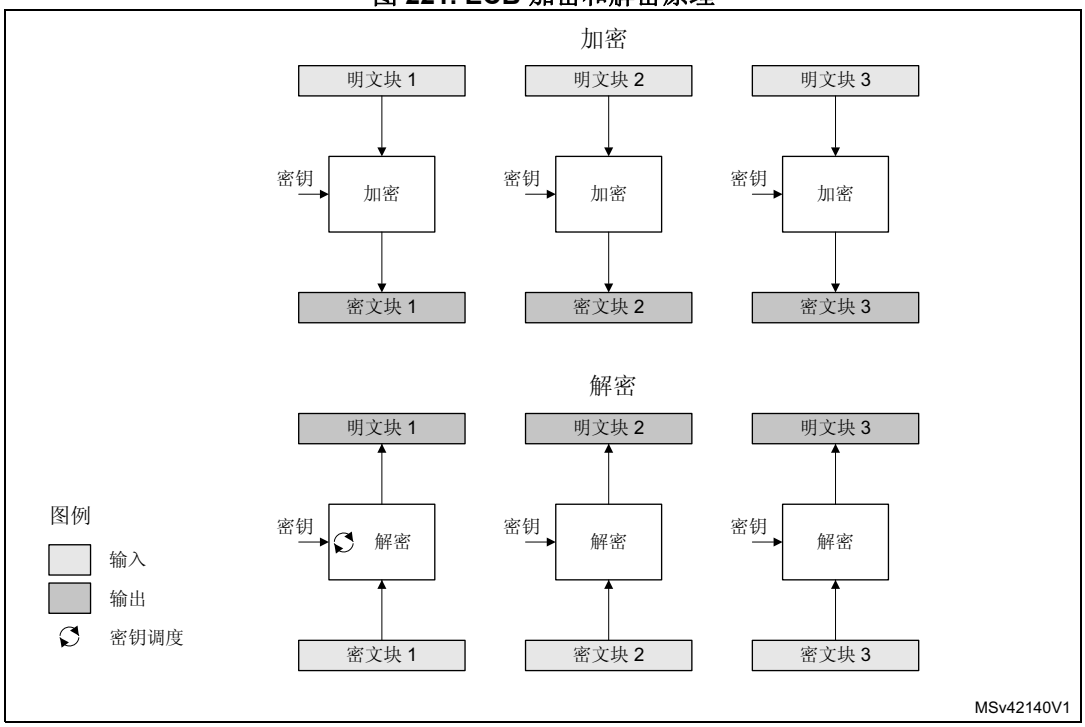
注： *只有在 AES 禁止（AES_CR 寄存器的位 EN 清零）时，才能更改链接模式。*

后续子章节对每个 AES 链接模式的原理进行了介绍。

详细信息请参见从*第 32.4.8 节：AES 基本链接模式（ECB 和 CBC）*开始的专用章节。

电子密码本 (ECB) 模式

图 221. ECB 加密和解密原理

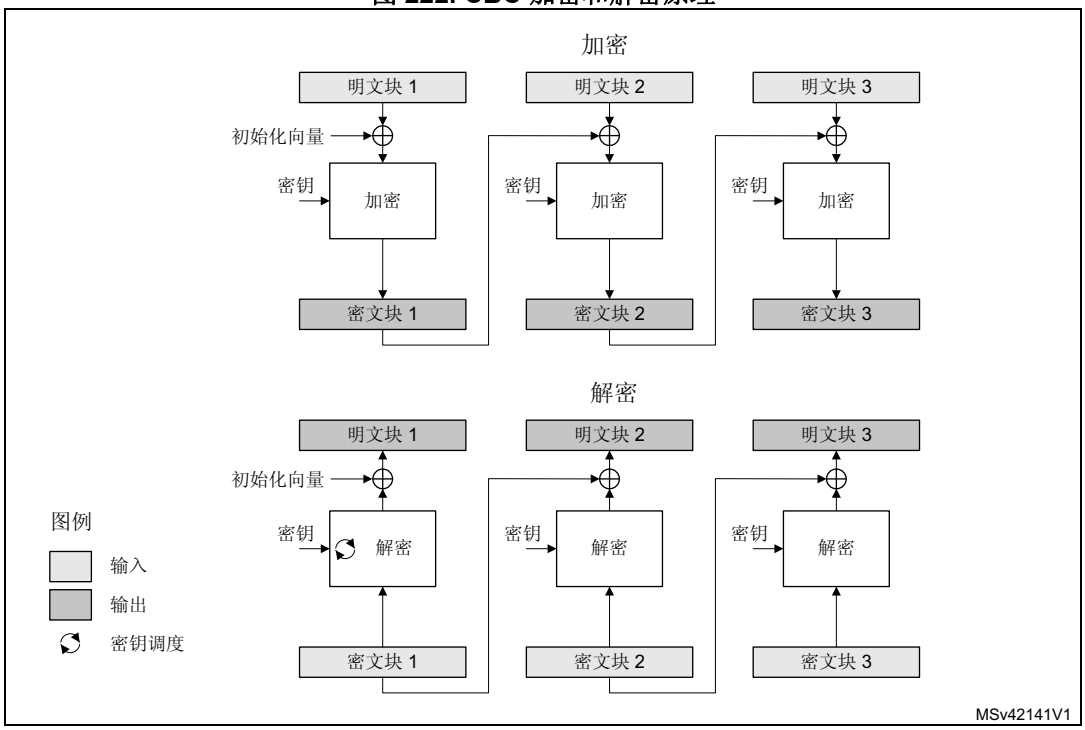


ECB 是最简单的工作模式。无链接操作，且无特殊初始化阶段。消息会划分到各个块中，并对各个块分别进行加密或解密。

注： *对于解密，在处理第一个块之前需要特殊的密钥调度。*

密码分组链接 (CBC) 模式

图 222. CBC 加密和解密原理

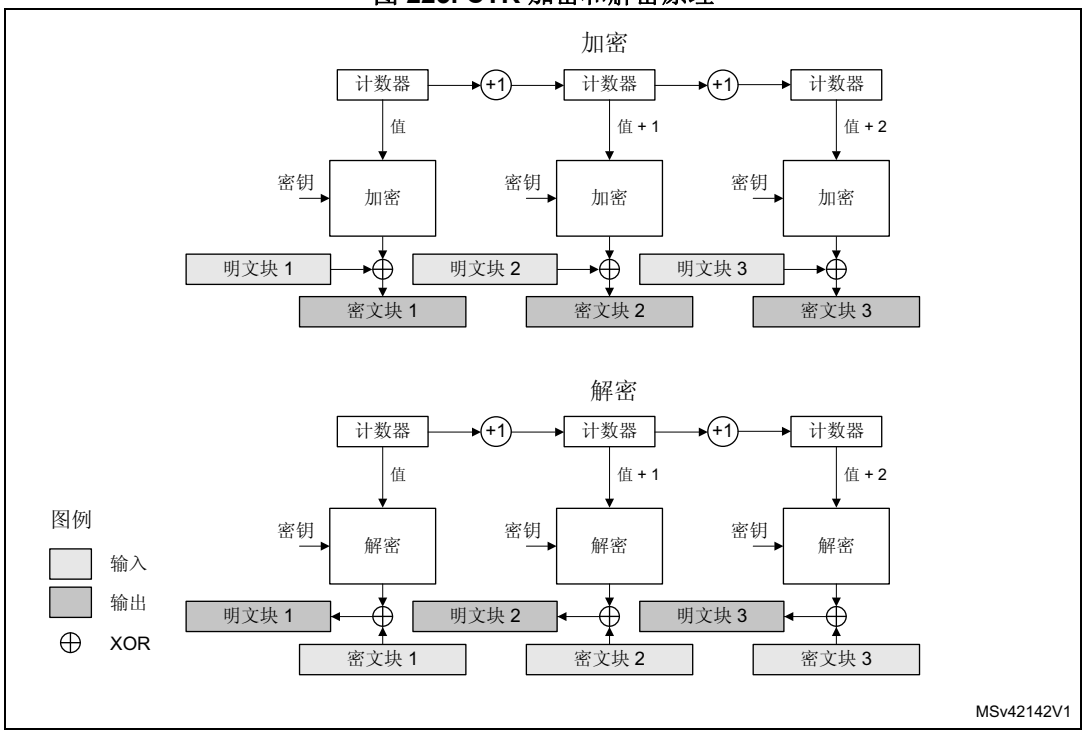


在 CBC 模式下，每个块的输出均与下一个块的输入相连。为了确保每条消息都是唯一的，会在第一个块处理过程中使用初始化向量。

注： *对于解密，在处理第一个块之前需要特殊的密钥调度。*

计数器 (CTR) 模式

图 223. CTR 加密和解密原理

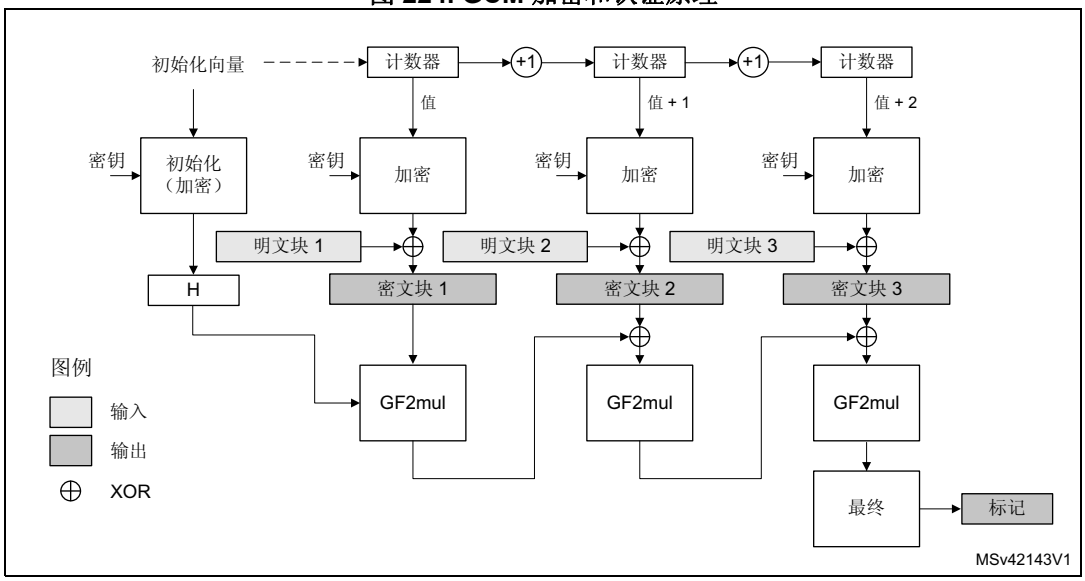


CTR 模式使用 AES 内核生成密匙流。这些密钥将与明文进行逻辑异或运算，以获得密文（如 NIST *特别出版物 800-38A，块密码工作模式的建议*中所述）。

注： *与 ECB 和 CBC 模式不同，CTR 解密无需密钥调度，因为在该链接方案中，AES 内核始终在加密模式下用于生成密钥流或计数器块。*

Galois/计数器模式 (GCM)

图 224. GCM 加密和认证原理

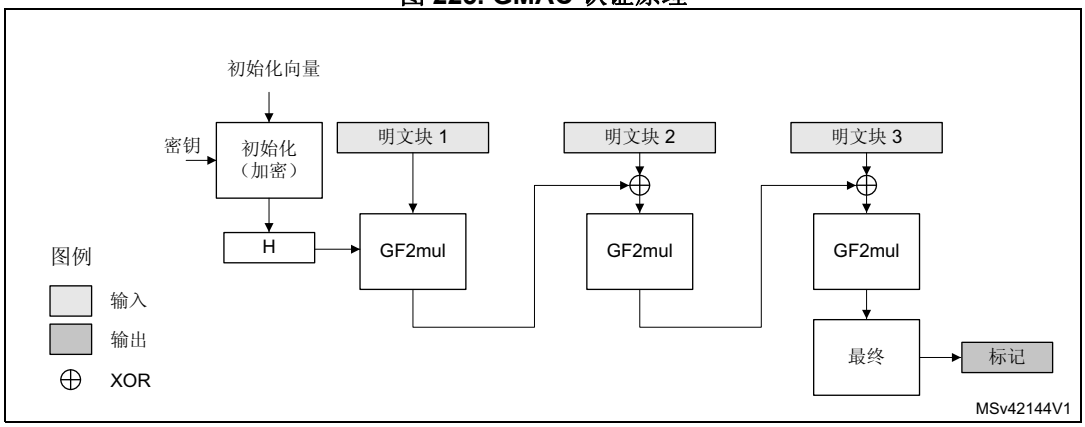


在 Galois/计数器模式 (GCM) 下，将对明文消息进行加密，同时会对消息认证码 (MAC) 进行计算，从而生成相应的密文及其 MAC（也称为认证标记）。NIST *特别出版物 800-38D，块密码工作模式的建议 - Galois/计数器模式 (GCM) 和 GMAC* 中对此进行了相关定义。

GCM 模式基于 AES 计数器模式，可实现机密性。它使用固定有限域乘法器来计算消息认证码。消息末尾处需要一个初始值和一个特定的 128 位块。

Galois 消息认证码 (GMAC) 原理

图 225. GMAC 认证原理

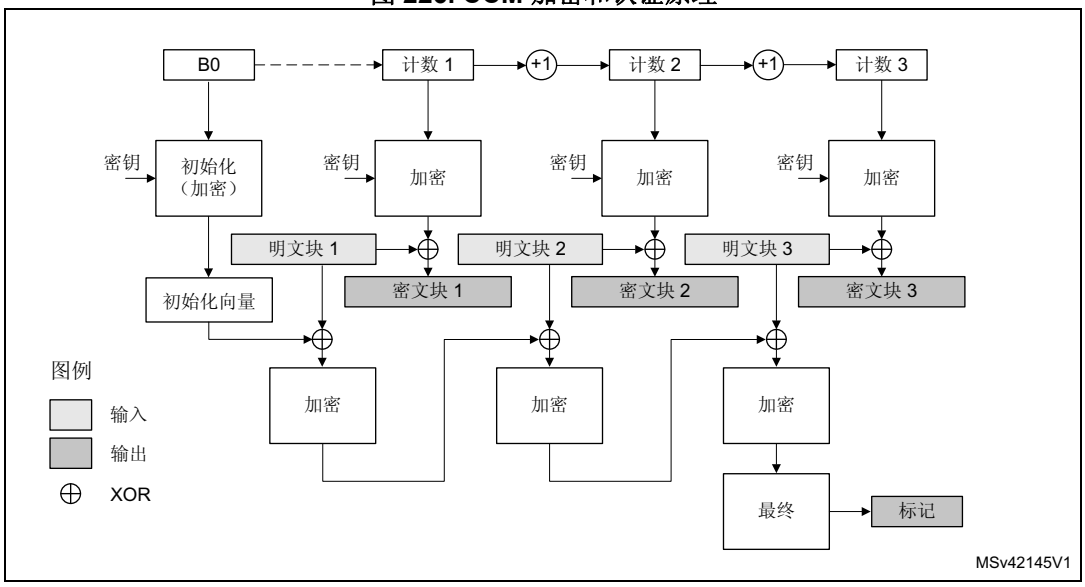


Galois 消息认证码 (GMAC) 允许认证消息并生成相应的消息认证码 (MAC)。NIST *特别出版物 800-38D，块密码工作模式的建议 - Galois/计数器模式 (GCM) 和 GMAC* 中对此进行了相关定义。

GMAC 与 GCM 类似，只不过它应用于仅由明文认证数据组成的消息（即只有标头，无有效负载）。

CBC-MAC 计数器 (CCM) 原理

图 226. CCM 加密和认证原理



在密码块链接-消息认证码计数器 (CCM) 模式下，将对明文消息进行加密，同时会对消息认证码 (MAC) 进行计算，从而生成相应的密文以及相应的 MAC（也称为标记）。NIST *特别出版物 800-38C，块密码工作模式的建议 - CCM 模式实现认证和机密性*中对此进行了介绍。

CCM 模式基于 AES 计数器模式，可实现机密性，并且使用 CBC 来计算消息认证码。它需要一个初始值。

与 GCM 类似，CCM 链接模式可应用于仅由明文认证数据组成的消息（即只有标头，无有效负载）。请注意，这种 CCM 使用方式不被称为 CMAC（它并非类似于 GCM/GMAC），NIST 不推荐这种用途。

32.4.4 执行密码操作的 AES 过程

简介

下面对典型密码操作进行了介绍。详细信息请参见从*第 32.4.8 节：AES 基本链接模式（ECB 和 CBC）*开始的章节。

图 227 和*图 228* 中的流程图介绍了 STM32 加密库如何实现 AES 算法。AES 可在 ECB、CBC、CTR、CCM 和 GCM 工作模式下加速执行 AES-128 和 AES-256 加密算法。

注： *有关加密库的更多详细信息，请参见 UM1924 用户手册“STM32 加密库”（可从 www.st.com 获取）。*

图 227. STM32 加密库 AES 流程图示例

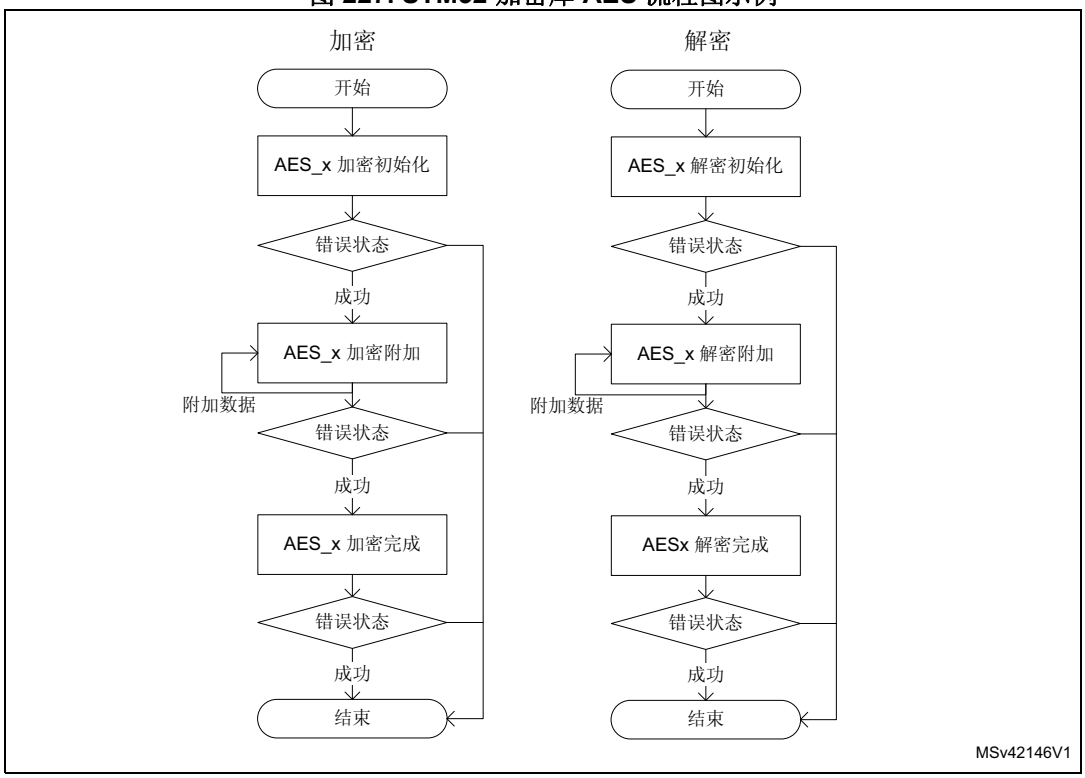
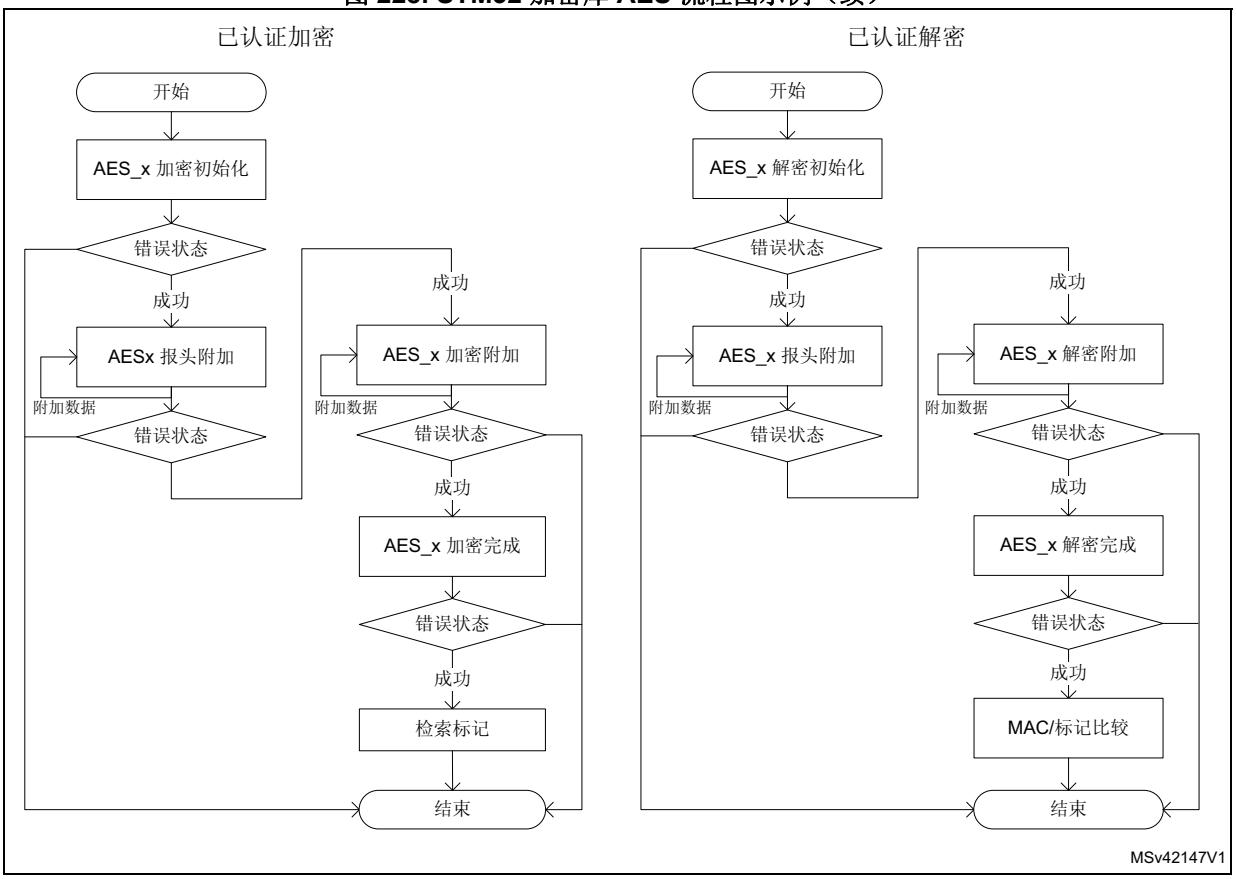


图 228. STM32 加密库 AES 流程图示例（续）



初始化 AES

要初始化 AES，首先通过将 AES_CR 寄存器中的 EN 位清零来禁止它。然后以任何顺序执行以下步骤：

- 通过对 AES_CR 寄存器中的 MODE[1:0] 位域进行编程来配置 AES 模式。
  - 对于加密，必须选择模式 1 (MODE[1:0] = 00)。
  - 对于解密，必须选择模式 3 (MODE[1:0] = 10)，除非使用 ECB 或 CBC 链接模式。在后一种情况下，必须执行加密密钥的初始密钥分散，如*第 32.4.5 节：AES 解密密钥准备*中所述。
- 通过对 AES_CR 寄存器中的 CHMOD[2:0] 位域进行编程来选择链接模式。
- 使用 AES_CR 寄存器中的 KEYSIZE 位域配置密钥大小（128 位或 256 位）。
- 将对称密钥写入 AES_KEYRx 寄存器（4 或 8 个寄存器，具体取决于密钥大小）。
- 使用 AES_CR 寄存器中的 DATATYPE[1:0] 位域配置数据类型（1 位、8 位、16 位或 32 位）。
- 必要时（例如，CBC 或 CTR 链接模式），向 AES_IVRx 寄存器中写入初始化向量。

附加数据

本节介绍附加数据进行处理的不同方式，其中要处理数据的大小不是 128 位的倍数。

有关 ECB 或 CBC 模式，请参见*第 32.4.6 节：AES 密文窃取和数据填充*。这些情况下的最后一个块管理比本节描述序列下的更复杂。

通过轮询附加数据

此方法按照以下序列使用标志轮询来控制附加数据：

1. 通过将 AES_CR 寄存器中的 EN 位置 1 来使能 AES 外设。
2. 重复以下子序列，直到完全处理有效负载：
   a) 将四个输入数据字写入 AES_DINR 寄存器。
   b) AES_SR 中的状态标志 CCF 置 1 后，从 AES_DOUTR 寄存器读取四个数据字。
   c) 通过将 AES_CR 寄存器中的 CCFC 位置 1，将 CCF 标志清零。
   d) 如果刚刚处理的数据块是消息的倒数第二个块，并且要处理的最后一个块中的有效数据不足 128 位，则用零填充最后一个块的剩余部分，并且在 GCM 有效负载加密或 CCM 有效负载解密的情况下，使用 AES_CR 寄存器的 NPBLB 位域指定无效字节的数量，以便 AES 计算正确的标记。
3. 丢弃不属于有效负载的数据，然后通过将 AES_CR 寄存器中的 EN 位清零来禁止 AES 外设。

注： *在对 AES_DINR 寄存器的两次连续写入之间自动插入最多三个等待周期，以便将密钥发送到 AES 处理器。*

NPBLB 不用于 GCM、GMAC 和 CCM 链接模式的标头阶段。

使用中断附加数据

此方法通过以下序列，使用 AES 外设中断来控制附加数据：

1. 通过将 AES_CR 寄存器中的 CCFIE 位置 1 来使能 AES 中断。
2. 通过将 AES_CR 寄存器中的 EN 位置 1 来使能 AES 外设。
3. 将前四个输入数据字写入 AES_DINR 寄存器。
4. 中断时处理 AES 中断服务程序中的数据：
   a) 从 AES_DOUTR 寄存器读取四个输出数据字。
   b) 通过将 AES_CR 寄存器中的 CCFC 位置 1，将 CCF 标志和挂起中断清零。
   c) 如果刚刚处理的数据块是消息的倒数第二个块，并且要处理的最后一个块中的有效数据不足 128 位，则用零填充最后一个块的其余部分，并且在 GCM 有效负载加密或 CCM 有效负载解密的情况下，使用 AES_CR 寄存器的 NPBLB 位域指定无效字节的数量，以便 AES 计算正确的标记。然后继续进行 4e)。
   d) 如果刚刚处理的数据块是消息的最后一块，则丢弃不属于有效负载的数据，然后通过将 AES_CR 寄存器中的 EN 位清零来禁止 AES 外设，并退出中断服务程序。
   e) 将接下来的四个输入数据字写入 AES_DINR 寄存器，并退出中断服务程序。

注： *AES 允许连续读取或写入操作之间存在延迟，例如在两次 AES 计算之间要响应的另一个外设的中断。*

NPBLB 不用于 GCM、GMAC 和 CCM 链接模式的标头阶段。

使用 DMA 附加数据

借助此方法，所有传输和处理过程均由 DMA 和 AES 管理。要使用此方法，请按照以下步骤操作：

1. 用零填充块的其余部分来准备最后四个字的数据块（如果要处理的数据没有完全填充它）。
2. 将 DMA 控制器配置为将来自存储器的要处理的数据传输到 AES 外设输入以及将来自 AES 外设输出的已处理数据传输到存储器，如*第 32.4.16 节：AES DMA 接口*所述。配置 DMA 控制器，以在传输完成时产生中断。在 GCM 有效负载加密或 CCM 有效负载解密的情况下，DMA **不得**包括最后一个四字块（如果块用零填充）。必修为此最后一个块必须使用*通过轮询附加数据*中描述的序列，因为处理块之前必须设置 NPBLB 位，以便 AES 计算正确的标记。
3. 通过将 AES_CR 寄存器中的 EN 位置 1 来使能 AES 外设。
4. 将 AES_CR 寄存器中的 DMAINEN 和 DMAOUTEN 位置 1，以使能 DMA 请求。
5. 当 DMA 中断指示传输完成时，从存储器中获取 AES 处理过的数据。

注： *CCF 标志在这种方法中没有作用，因为读取 AES_DOUTR 寄存器是通过 DMA 自动管理的，在计算过程结束时没有任何软件操作。*

NPBLB 不用于 GCM、GMAC 和 CCM 链接模式的标头阶段。

32.4.5 AES 解密密钥准备

对于 ECB 或 CBC 解密，第一轮解密的密钥必须从最后一轮加密的密钥中分散。因此，在执行解密之前需要完整的加密密钥计划。除 ECB 或 CBC 模式外的 AES 解密**不需要**密钥准备。

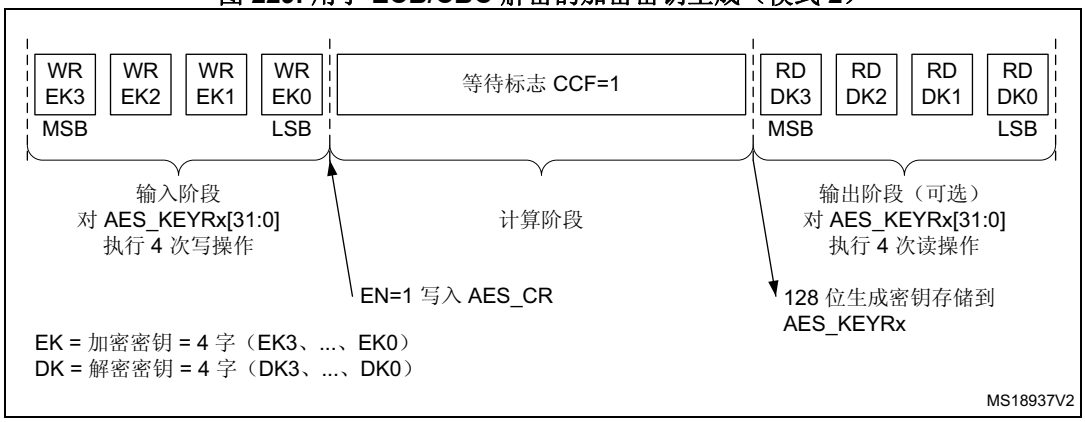
推荐方法是通过将 AES_CR 的 MODE[1:0] 位域设置为 01 来选择模式 2（仅密钥处理），然后通过将 MODE[1:0] 设置为 10 来继续解密（模式 3，仅解密）。模式 2 的使用说明如下：

1. 通过将 AES_CR 寄存器中的 EN 位清零来禁止 AES 外设。
2. 通过将 AES_CR 的 MODE[1:0] 位域设置为 01 来选择模式 2。由于此密钥生成模式独立于所选的链接算法，因此 CHMOD[2:0] 位域在这种情况下并不重要。
3. 通过 AES_CR 寄存器的 KEYSIZE 位，将密钥长度设置为 128 或 256 位。
4. 将加密密钥写入 AES_KEYRx 寄存器（128 或 256 位），如*图 229* 所示。对 AES_IVRx 寄存器执行写入操作不起作用。
5. 通过将 AES_CR 寄存器中的 EN 位置 1 来使能 AES 外设。
6. 等待 AES_SR 寄存器中的 CCF 标志置 1。
7. 生成的密钥在 AES 内核中，可以用于解密。如果需要，应用程序还可以读取 AES_KEYRx 寄存器以获得生成的密钥，如*图 229* 所示（已处理密钥自动加载到 AES_KEYRx 寄存器中）。

注： *分散密钥可用时，将由硬件禁止 AES。*

要重新开始计算生成密钥，请重复步骤 4、5、6 和 7。

图 229. 用于 ECB/CBC 解密的加密密钥生成（模式 2）



如果软件存储了为解密准备的初始密钥，则对于要用给定密钥解密的所有数据，仅执行一次密钥调度操作就已足够。

注： *密钥准备操作持续 59 或 82 个时钟周期，具体取决于密钥大小（128 或 256 位）。*

注： *备用密钥准备是通过将 AES_CR 寄存器中的 MODE[1:0] 位域设置为 11 来选择模式 4。这种情况下，模式 3 不可用。*

32.4.6 AES 密文窃取和数据填充

在 ECB 或 CBC 模式下使用 AES 来管理不是块大小（128 位）倍数的消息时，请使用密文窃取技术，如 NIST *特别出版物 800-38A，块密码工作模式的建议：CBC 模式密文窃取的三种变型*中介绍的窃取技术。由于 AES 外设不支持此类技术，应用程序必须使用倒数第二个块中的数据完成输入数据的最后一个块。

注： *本参考手册中未对密文窃取技术进行介绍。*

类似地，在除 ECB 或 CBC 之外的模式下使用 AES 时，必须先用零填充不完整的输入数据块（即，输入数据短于 128 位的块），然后再加密（即，必须在数据串的尾端附加额外的位）。解密后必须丢弃额外填充的位。由于 AES 不会对**最后一个块**执行自动数据填充操作，因此应用程序必须遵循*第 973 页的第 32.4.4 节：执行密码操作的 AES 过程*中给出的建议来管理大小不是 128 位倍数的消息。

注： *填充数据根据 AES_CR 寄存器中的 DATATYPE[1:0] 字段，以类似于正常数据的方式进行交换（有关详细信息，请参见第 997 页的第 32.4.13 节：AES 数据寄存器和数据交换）。*

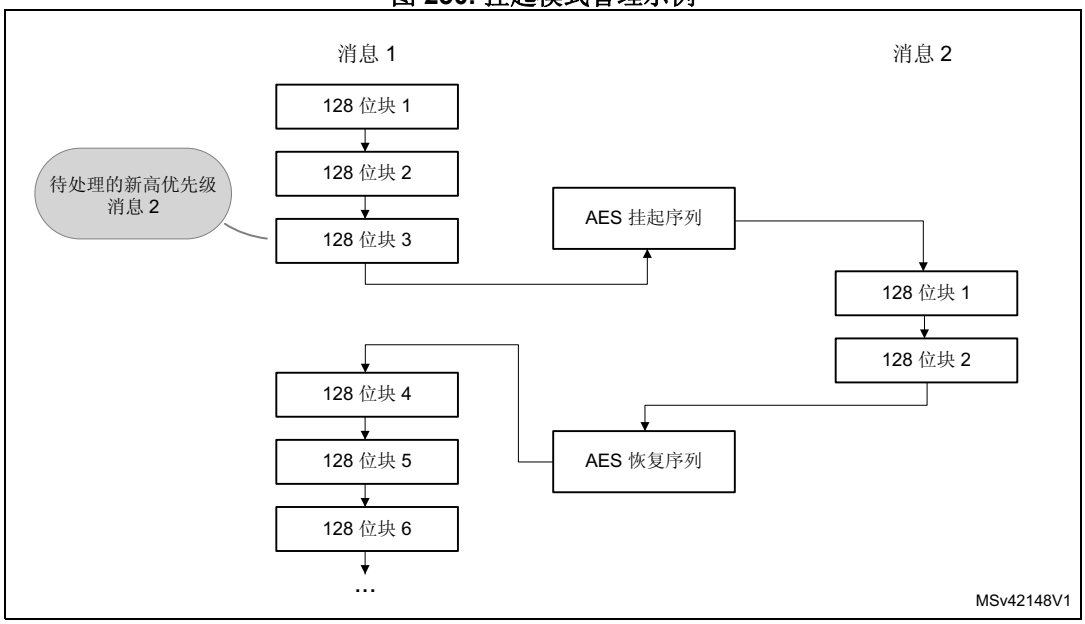
32.4.7 AES 任务挂起和恢复

如果必须处理另一条优先级较高的消息，则可将消息挂起。该最高优先级消息完成发送后，可在加密或解密模式下恢复挂起的消息。

挂起/恢复操作不会破坏链接运算，并且再次使能 AES 以接收下一个数据块时，可立即恢复消息处理。

图 230 所示为挂起/恢复操作示例：为发送长度更短、优先级更高的消息 2 而将消息 1 挂起。

图 230. 挂起模式管理示例



有关挂起/恢复操作的详细说明，请参见每个 AES 模式的相应章节。

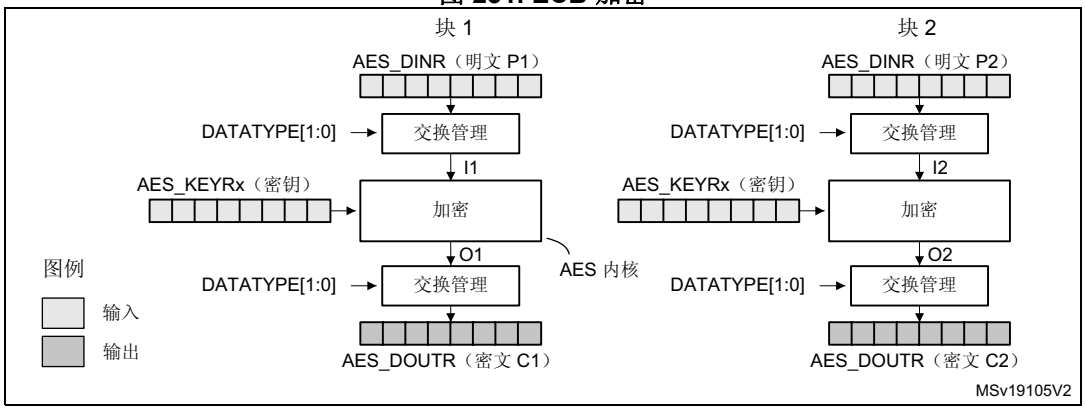
32.4.8 AES 基本链接模式（ECB 和 CBC）

概述

本节简要介绍 AES 计算内核提供的四种基本工作模式：ECB 加密、ECB 解密、CBC 加密和 CBC 解密。有关详细信息，请参见 2001 年 11 月 26 日的 FIPS 出版物 197。

图 231 介绍了电子密码本 (ECB) 加密。

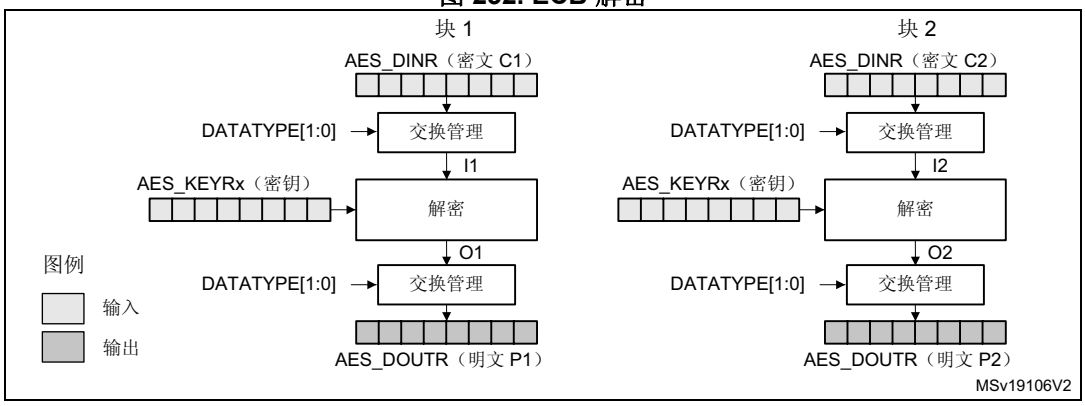
图 231. ECB 加密



在 ECB 加密模式下，AES_DINR 寄存器中的 128 位明文输入数据块 Px 首先进行位/字节/半字交换。使用 128 位或 256 位密钥，通过在 AES 加密模式下设置的 AES 内核对交换结果 Ix 进行处理。加密结果 Ox 经过位/字节/半字交换，然后作为 128 位密文输出数据块 Cx 存储在 AES_DOUTR 寄存器中。ECB 加密以这种方式继续，直到最后一个完整的明文块被加密。

图 232 介绍了电子密码本 (ECB) 解密。

图 232. ECB 解密

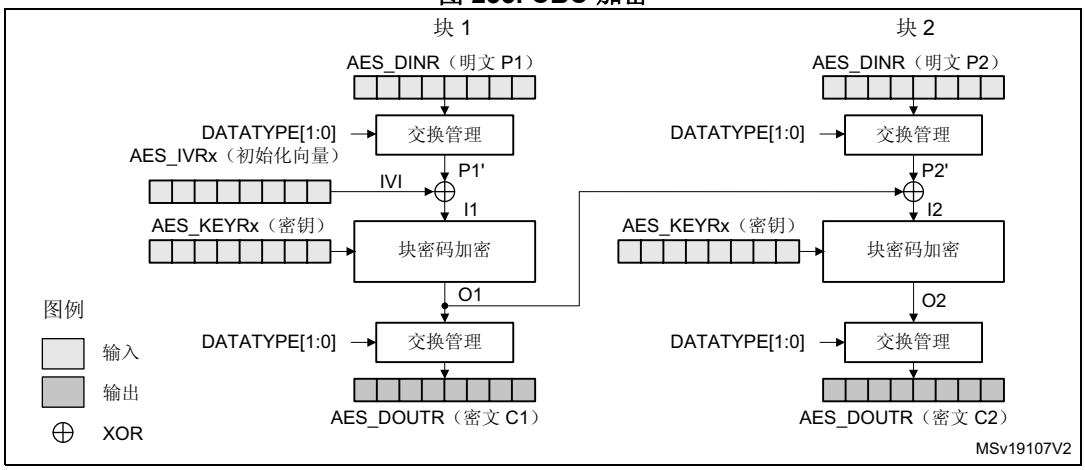


要在 ECB 模式下执行 AES 解密，需要通过收集最后一轮加密密钥准备密钥（需要针对加密首先执行完整的密钥计划），并将其用作解密密文的第一个轮密钥。该准备由 AES 内核提供支持。

在 ECB 解密模式下，AES_DINR 寄存器中的 128 位密文输入数据块 C1 首先进行位/字节/半字交换。该密钥序列与 ECB 加密中的密钥序列相反。使用先前准备的解密密钥，通过在解密模式下设置的 AES 内核对交换结果 I1 进行处理。解密结果经过位/字节/半字交换，然后作为 128 位明文输出数据块 P1 存储在 AES_DOUTR 寄存器中。ECB 解密以这种方式继续，直到最后一个完整的密文块被解密。

图 233 说明了密码块链接 (CBC) 加密模式。

图 233. CBC 加密

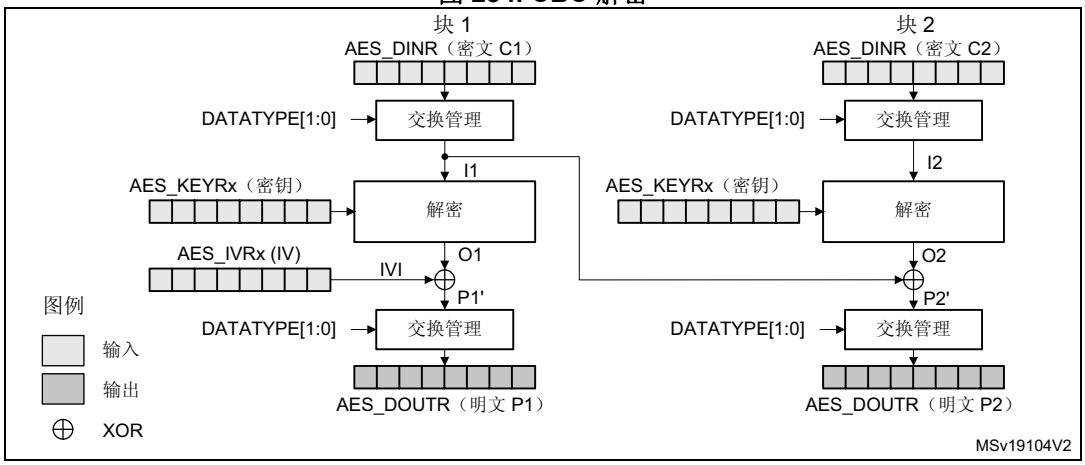


在 CBC 加密模式下，执行位/字节/半字交换 (P1') 后的第一个明文输入块与一个 128 位 IVI 位域（初始化向量和计数器）进行逻辑异或运算，从而使用 128 位或 256 位密钥通过 AES 内核生成用于加密的 I1 输入数据。在交换操作之后，得到的 128 位输出块 O1 用作密文 C1。然后将 O1 数据与第二块明文数据 P2' 进行逻辑异或运算，生成用于 AES 内核的 I2 输入数据以生成第二块密文数据。数据块的链接以这种方式继续，直到消息中的最后一个明文块被加密。

如果消息大小不是 128 位的倍数，则将按照*第 32.4.6 节：AES 密文窃取和数据填充*中说明的方式对最后的不完整数据块进行加密。

图 234 说明了密码块链接 (CBC) 解密。

图 234. CBC 解密



在 CBC 解密模式下（如 ECB 解密模式），必须准备密钥才可以执行 AES 解密。

完成密钥准备过程后，按下述方式执行解密：使用 128 位或 256 位密钥将第一个 128 位密文块（在交换操作之后）直接用作用于解密操作的 AES 核心输入块 I1。其输出 O1 与 128 位 IVI 字段（必须与加密期间使用的字段相同）进行逻辑异或运算以生成第一个明文块 P1。

除了使用来自第一个块的 I1 数据代替初始化向量外，第二个密文块的处理方式与第一个块相同。

解密以这种方式继续，直到最后一个完整的密文块被解密。

如果消息大小不是 128 位的倍数，则将按照*第 32.4.6 节：AES 密文窃取和数据填充*中说明的方式对最后的不完整数据块进行解密。

有关数据交换的更多信息，请参见*第 32.4.13 节：AES 数据寄存器和数据交换*。

ECB/CBC 加密序列

执行 ECB/CBC 加密的事件序列（详见*第 32.4.4 节*）：

1. 通过将 AES_CR 寄存器中的 EN 位清零来禁止 AES 外设。
2. 通过将 AES_CR 寄存器的 MODE[1:0] 位域设置为 00 来选择模式 1，通过将 AES_CR 寄存器的 CHMOD[2:0] 位域设置为 000 或 001 来分别选择 ECB 或 CBC 链接模式。也可以使用 DATATYPE[1:0] 位域定义数据类型。
3. 通过 AES_CR 寄存器中的 KEYSIZE 位选择 128 位或 256 位密钥长度。
4. 将加密密钥写入 AES_KEYRx 寄存器（128 或 256 位）。如果选择了 CBC 模式，则用初始化向量数据填充 AES_IVRx 寄存器。
5. 通过将 AES_CR 寄存器中的 EN 位置 1 来使能 AES 外设。
6. 对 AES_DINR 寄存器执行四次写入操作，以输入明文（MSB 优先），如*图 235* 所示。
7. 等待 AES_SR 寄存器中的 CCF 标志置 1。
8. 对 AES_DOUTR 寄存器进行四次读取操作，以获取密文（MSB 优先），如*图 235* 所示。然后通过将 AES_CR 寄存器中的 CCFC 位置 1，将 CCF 标志清零。
9. 重复执行步骤 *6*-*7*-*8*，使用相同加密密钥处理所有块。

图 235. ECB/CBC 加密（模式 1）



ECB/CBC 解密序列

执行 AES ECB/CBC 解密的事件序列如下（详见*第 32.4.4 节*）：

1. 按照*第 976 页的第 32.4.5 节：AES 解密密钥准备*中所述的步骤，准备 AES 内核中的解密密钥。
2. 通过将 AES_CR 寄存器的 MODE[1:0] 位域设置为 10 来选择模式 3，并通过将 AES_CR 寄存器的 CHMOD[2:0] 位域设置为 000 或 001 来分别选择 ECB 或 CBC 链接模式。也可以使用 DATATYPE[1:0] 位域定义数据类型。
3. 通过 AES_CR 寄存器的 KEYSIZE 位域，选择 128 或 256 位密钥长度。
4. 将初始化向量写入 AES_IVRx 寄存器（仅在 CBC 模式下需要此步骤）。
5. 通过将 AES_CR 寄存器中的 EN 位置 1 来使能 AES。
6. 对 AES_DINR 寄存器执行四次写入操作，以输入密文（MSB 优先），如*图 236* 所示。
7. 等待 AES_SR 寄存器中的 CCF 标志置 1。
8. 对 AES_DOUTR 寄存器进行四次读取操作，以获取明文（MSB 优先），如*图 236* 所示。然后通过将 AES_CR 寄存器中的 CCFC 位置 1，将 CCF 标志清零。
9. 重复执行步骤 *6*-*7*-*8*，使用相同密钥解密所有块。

图 236. ECB/CBC 解密（模式 3）



ECB/CBC 模式下的挂起/恢复操作

要挂起消息处理，请执行以下操作：

1. 如果使用 DMA，则通过将 AES_CR 寄存器中的 DMAINEN 位清零来停止 AES DMA 对 IN FIFO 的数据传输。
2. 如果未使用 DMA，则读取四次 AES_DOUTR 寄存器以保存最后处理的块。如果使用 DMA，等待 AES_SR 寄存器中的 CCF 标志置 1，然后通过将 AES_CR 寄存器中的 DMAOUTEN 输出位清零来停止 DMA 对 OUT FIFO 的数据传输。
3. 如果未使用 DMA，轮询 AES_SR 寄存器的 CCF 标志，直到它变为 1（计算完成）。
4. 通过将 AES_CR 寄存器中的 CCFC 位置 1，将 CCF 标志清零。
5. 保存初始化向量寄存器（仅在 CBC 模式下需要此步骤，因为 AES_IVRx 寄存器在数据处理过程中已发生更改）。
6. 通过将 AES_CR 寄存器中的 EN 位清零来禁止 AES 外设。
7. 将当前 AES 配置保存到存储器中（AES 初始化向量值除外）。
8. 如果使用 DMA，则保存 DMA 控制器状态（指向 IN 和 OUT 数据传输的指针以及剩余字节数等）。

注： *在第 7 步中，如果中断的进程是解密，则可以选择将存储在 AES_KEYRx 寄存器中的已分散密钥信息保存在存储器中。否则无需保存这些寄存器，因为应用程序已知初始密钥值。*

要恢复消息处理，请执行以下操作：

1. 如果使用 DMA，则配置 DMA 控制器以完成 FIFO IN 和 FIFO OUT 传输的其余部分。
2. 确保 AES 禁止（AES_CR 的 EN 位必须为 0）。
3. 使用已保存配置的值恢复 AES_CR 和 AES_KEYRx 寄存器设置。在解密的情况下，可将生成的密钥信息写入 AES_KEYRx 寄存器，而不是原始密钥值。
4. 按*第 32.4.5 节：AES 解密密钥准备*所述准备解密密钥（仅 ECB 或 CBC 解密需要此步骤）。如果已将生成的密钥信息加载到 AES_KEYRx 寄存器中，则此步骤不是必需的。
5. 使用保存的配置恢复 AES_IVRx 寄存器（仅在 CBC 模式下需要此步骤）。
6. 通过将 AES_CR 寄存器中的 EN 位置 1 来使能 AES 外设。
7. 如果使用 DMA，将 AES_CR 寄存器中的 DMAINEN 和 DMAOUTEN 位置 1，以使能 AES DMA 传输。

使用模式 4 执行备用单次 ECB/CBC 解密

使用模式 4 执行单次 ECB/CBC 解密的事件序列为：

1. 通过将 AES_CR 寄存器中的 EN 位清零来禁止 AES 外设。
2. 通过将 AES_CR 寄存器的 MODE[1:0] 位域设置为 11 来选择模式 4，通过将 AES_CR 寄存器的 CHMOD[2:0] 位域设置为 000 或 001 来分别选择 ECB 或 CBC 链接模式。
3. 通过 AES_CR 寄存器的 KEYSIZE 位域，选择 128 或 256 位密钥长度。
4. 将加密密钥写入 AES_KEYRx 寄存器。如果选择 CBC 模式，则写入 AES_IVRx 寄存器。
5. 通过将 AES_CR 寄存器中的 EN 位置 1 来使能 AES 外设。
6. 对 AES_DINR 寄存器执行四次写入操作，以输入密文（MSB 优先）。
7. 等待 AES_SR 寄存器中的 CCF 标志置 1。
8. 对 AES_DOUTR 寄存器进行四次读取操作，以获取明文（MSB 优先）。然后通过将 AES_CR 寄存器中的 CCFC 位置 1，将 CCF 标志清零。

注： *选择模式 4 时，不能使用模式 3。*

在模式 4 下，AES_KEYRx 寄存器包含所有处理阶段的加密密钥。生成密钥不会存储在这些寄存器中。它会存储在 AES 内部。

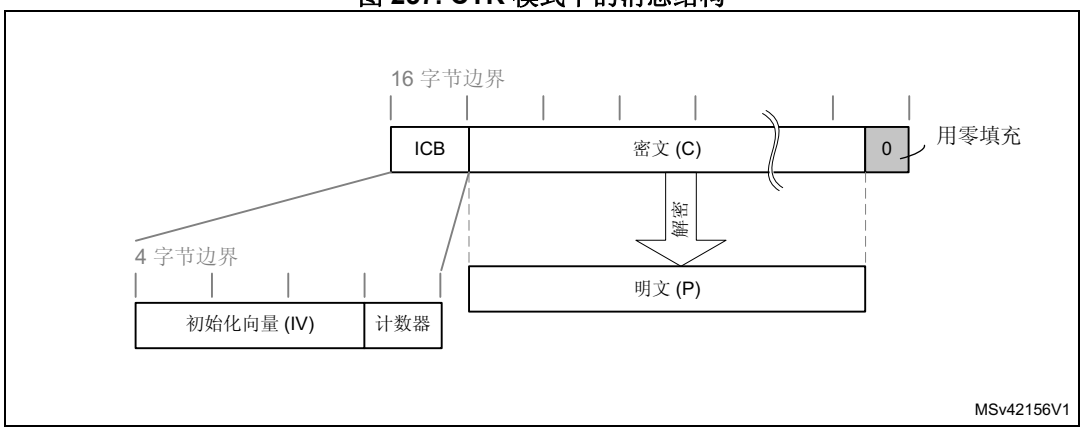
32.4.9 AES 计数器 (CTR) 模式

概述

计数器模式 (CTR) 使用 AES 作为密钥流生成器。然后，生成的密钥与明文进行逻辑异或运算来获得密文。

NIST *特别出版物 800-38A，块密码工作模式的建议*对 CTR 链接进行了相关定义。*图 237* 给出了 CTR 模式下的典型消息结构。

图 237. CTR 模式下的消息结构



该消息的结构为：

- 16 字节的初始计数器块 (ICB)，它由两个不同的字段组成：
  - **初始化向量** (IV)：96 位值，对于给定密钥下每个加密周期，该值都必须唯一。
  - **计数器**：32 位大端模式的整数，随着块的处理次数而递增。应将计数器的初始值设为 1。
- 明文 P 被加密为密文 C（长度已知）。该长度可以不是 16 字节的倍数，在这种情况下需要明文填充。

CTR 加密和解密

图 238 和*图 239* 分别描述了 AES 外设中实现的 CTR 加密和解密过程。通过向 AES_CR 寄存器中的 CHMOD[2:0] 位域写入 010 来选择 CTR 模式。

图 238. CTR 加密

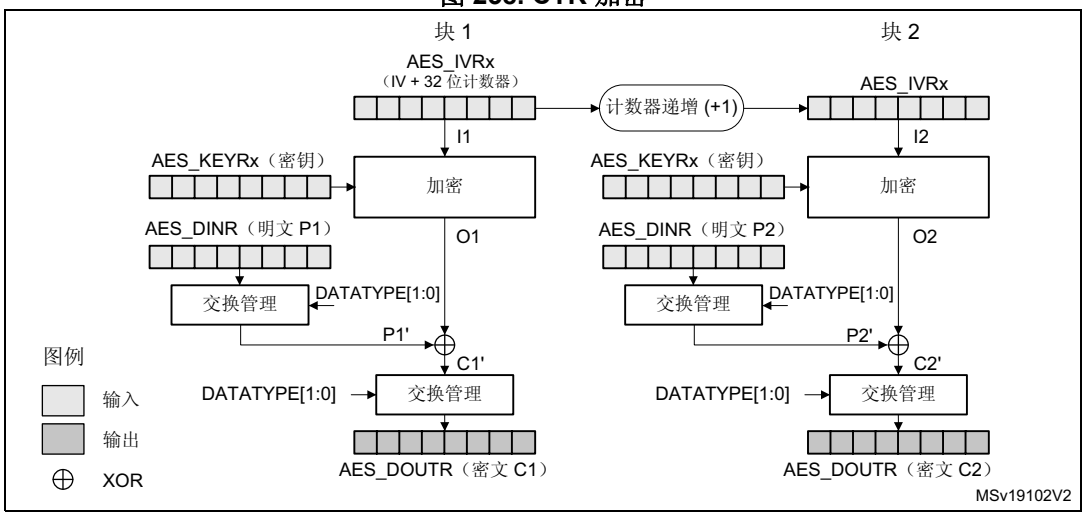
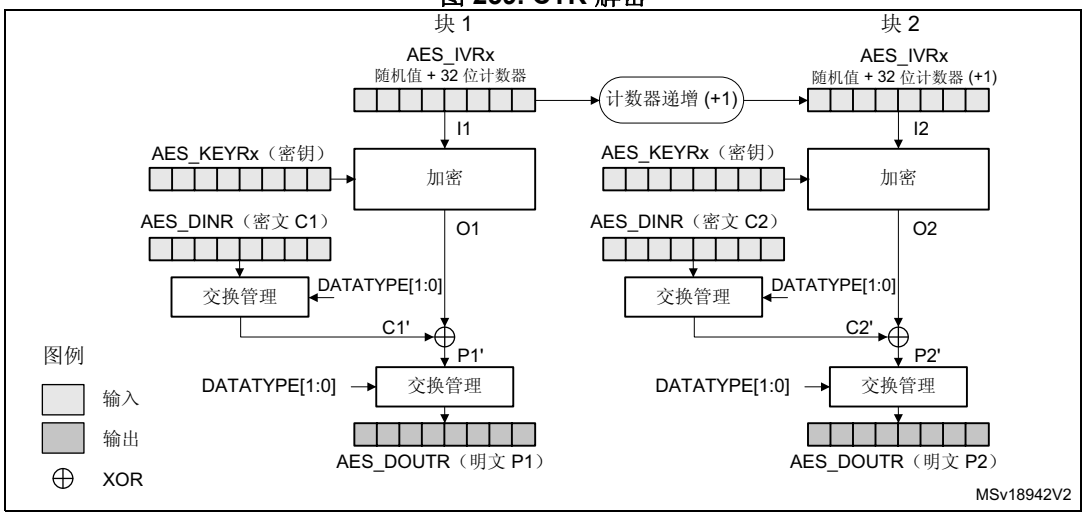


图 239. CTR 解密



在 CTR 模式中，将加密核心输出（也称为密钥流）Ox 与相关输入块（用于加密的 Px'，用于解密的 Cx'）进行逻辑异或运算，以生成正确的输出块（用于加密的 Cx'，用于解密的 Px'）。AES 中的初始化向量必须按*表 196* 所示进行初始化。

表 196. CTR 模式初始化矢量定义

AES_IVR3[31:0]	AES_IVR2[31:0]	AES_IVR1[31:0]	AES_IVR0[31:0]
Nonce[31:0]	Nonce[63:32]	Nonce[95:64]	32 位计数器 = 0x0001

在 CBC 模式下，处理第一个数据块时仅使用 AES_IVRx 寄存器一次，而在 CTR 模式下则有所不同，处理每个数据块都使用 AES_IVRx 寄存器，并且 AES 外设会使初始化向量的计数器位递增（随机值位保持不变）。

CTR 解密与 CTR 加密并无不同，因为内核始终会对当前的计数器块加密以产生密钥数据流，该密钥数据流与明文（CTR 加密）或密文（CTR 解密）输入进行异或运算。在 CTR 模式下，MODE[1:0] 位域设置 11、10 或 00 将全部默认为加密模式，同时禁止设置 01（密钥生成）。

在 CTR 链接模式下执行加密或解密的事件序列：

1. 确保 AES 禁止（AES_CR 的 EN 位必须为 0）。
2. 通过将 AES_CR 寄存器中的 CHMOD[2:0] 位域设置为 010 来选择 CTR 链接模式。将 MODE[1:0] 位域设置为除 01 之外的任何值。
3. 初始化 AES_KEYRx 寄存器，并按*表 196* 所述加载 AES_IVRx 寄存器。
4. 将 AES_CR 寄存器的 EN 位置 1，开始加密当前计数器（计算完成时 EN 自动复位）。
5. 如果是最后一个块，可在需要时用零填充数据以获得完整的块。
6. 在 AES 中附加数据并读取结果。*第 32.4.4 节：执行密码操作的 AES 过程*介绍了三种可能的情况。
7. 重复上一步，直到处理完倒数第二个块。对于最后一个块，执行前两步并将不属于有效负载的一部分的位丢弃（如果最后一个输入块中有效数据的大小小于 16 字节）。

CTR 模式下的挂起/恢复操作

该模式与 CBC 模式相似，可以中断消息、发送优先级更高的消息，并可恢复已中断的消息。有关 CBC 挂起/恢复序列的详细信息，请参见*第 32.4.8 节：AES 基本链接模式（ECB 和 CBC）*。

注： *与 CBC 模式类似，恢复操作期间必须重新加载 AES_IVRx 寄存器。*

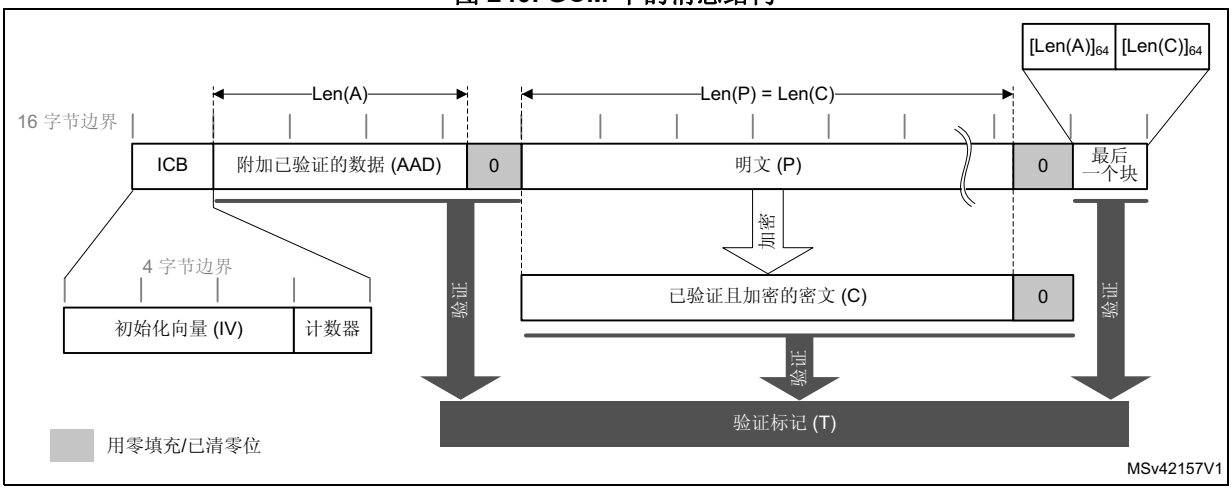
32.4.10 AES Galois/计数器模式 (GCM)

概述

AES Galois/计数器模式 (GCM) 允许将明文消息加密和验证为对应的密文和标记（也称为消息认证码）。为实现保密性，GCM 算法基于 AES 计数器模式。它使用固定有限域乘法器来生成标记。

NIST *特别出版物 800-38D，块密码工作模式的建议 - Galois/计数器模式 (GCM) 和 GMAC* 中对 GCM 链接进行了相关定义。*图 240* 所示为 GCM 模式下的典型消息结构。

图 240. GCM 中的消息结构



消息具有以下结构：

- **16 字节的初始计数器块 (ICB)**，它由两个不同的字段组成：
  - **初始化向量 (IV)**：96 位值，对于给定密钥下每个加密周期，该值都必须唯一。请注意，GCM 标准支持短于 96 位的 IV，在这种情况下应遵循严格的规则。
  - **计数器**：32 位大端模式的整数，随着块的处理次数而递增。根据 NIST 规范，处理有效负载的第一个块时，计数器值为 0x2。
- **认证的标头 AAD**（也称为附加认证数据）具有已知长度 Len(A)，此长度值可以不是 16 字节的倍数，但不能超过 $2^{64} – 1$ 位。消息的这一部分仅经过认证，未被加密。
- **明文消息 P** 将经过认证且被加密为密文 C，其具有已知长度 Len(P)，该长度值可以不是 16 字节的倍数，但不能超过 $2^{32} - 2$ 个 128 位块。
- **最后一个块**包含 AAD 标头长度（位 [32:63]）和有效负载长度（位 [96:127]）信息，如 *表 197* 所示。

GCM 标准规定密文 C 的位长度与明文 P 的位长度相同。

当消息某一部分（AAD 或 P）的长度不是 16 字节的倍数时，需要采取特殊填充方案。

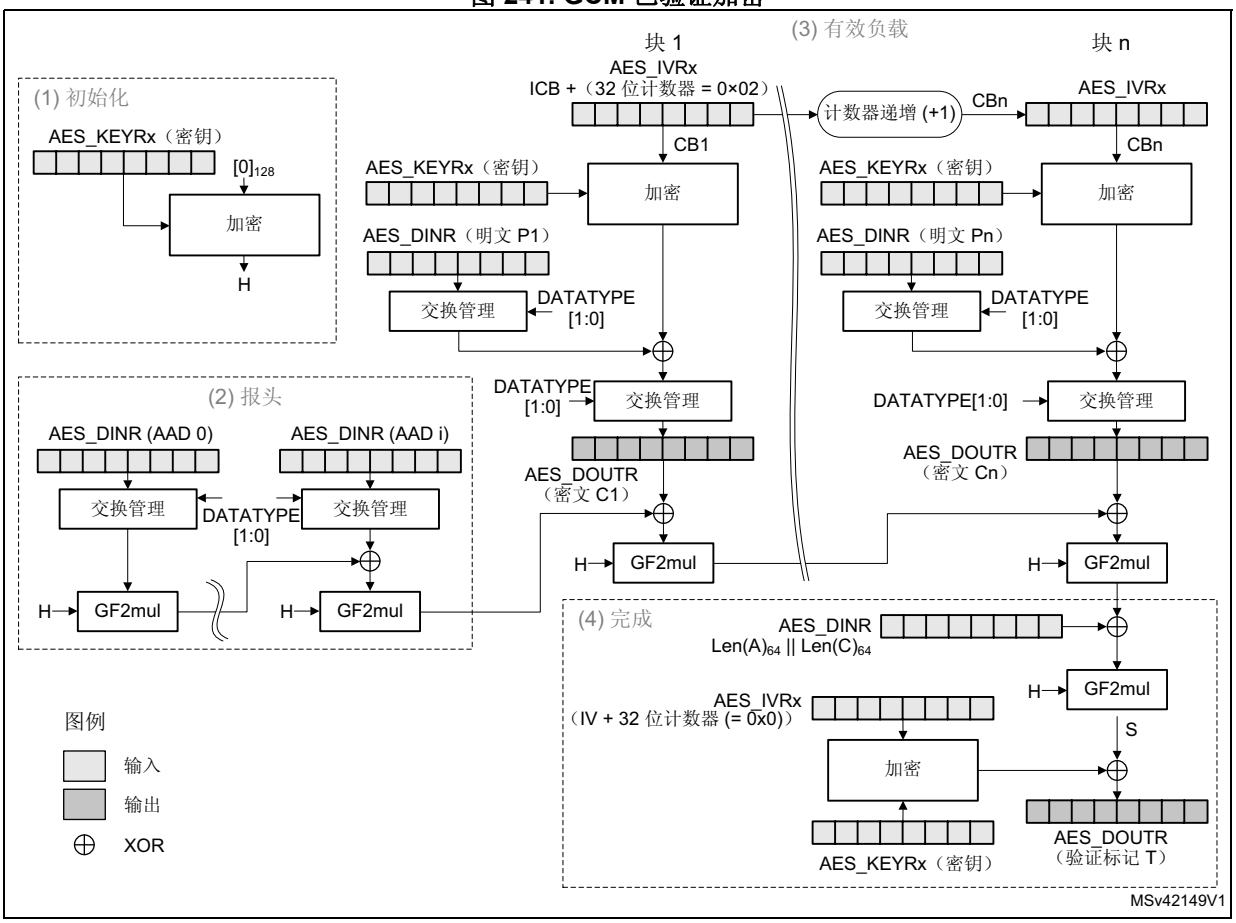
表 197. GCM 最后一个块定义

字节序	位 [0] ---------- 位 [31]	位 [32]---------- 位 [63]	位 [64] -------- 位 [95]	位 [96] --------- 位 [127]
输入数据	0x0	AAD 长度 [31:0]	0x0	有效负载长度 [31:0]

GCM 处理

图 241 列出了 AES 外设中的 GCM 实现。通过向 AES_CR 寄存器中的 CHMOD[2:0] 位域写入 011 来选择 GCM。

图 241. GCM 已验证加密



GCM 模式下明文的保密性机制与计数器模式下的类似，其具有特定递增函数（表示为 32 位递增），可生成输入计数器块序列。

保持数据的**计数器块**的 AES_IVRx 寄存器用于处理每个数据块。AES 外设自动递增计数器 [31:0] 位域。第一个计数器块 (CB1) 由应用软件从初始计数器块 ICB 分散（请参见*表 198*）。

表 198. GCM 模式 IVI 位域初始化

寄存器	AES_IVR3[31:0]	AES_IVR2[31:0]	AES_IVR1[31:0]	AES_IVR0[31:0]
输入数据	ICB[31:0]	ICB[63:32]	ICB[95:64]	Counter[31:0] = 0x2

注： *在 GCM 模式下，禁止将 MODE[1:0] 位域设置为 01 和 11。*

GCM 模式下的认证机制基于散列函数（称为 **GF2mul**），其在二元 Galois 域内执行与固定参数（称为散列子密钥 (H)）的乘法。

GCM 消息通过以下阶段进行处理（将在下一小节中对此进行进一步介绍）：

- **初始化阶段**：AES准备 GCM 散列子密钥 (H)。
- **标头阶段**：AES 处理附加认证数据 (AAD)（仅使用散列计算）。
- **有效负载阶段**：AES 基于散列计算、计数器块加密和数据异或运算处理明文 (P)。其操作方式与密文 (C) 类似。
- **最后阶段**：AES 使用消息的最后一个块生成认证标记 (T)。

GCM 初始化阶段

在此第一步中，将在内部计算和保存 GCM 散列子密钥 (H)，供处理所有块使用。建议序列为：

1. 确保 AES 禁止（AES_CR 的 EN 位必须为 0）。
2. 通过将 AES_CR 寄存器的 CHMOD[2:0] 位域设置为 011 选择 GCM 链接模式，并可选择将 DATATYPE[1:0] 位域置 1。
3. 通过将 AES_CR 寄存器中的 GCMPH[1:0] 位域设置为 00 来指示初始化阶段。
4. 将 AES_CR 寄存器的 MODE[1:0] 位域设置为 00 或 10。
5. 使用密钥初始化 AES_KEYRx 寄存器，使用*表 198* 中定义的信息初始化 AES_IVRx 寄存器。
6. 开始通过将 AES_CR 寄存器的 EN 位设置为 1 来计算散列密钥（计算完成时 EN 自动复位）。
7. 等待计算完成，通过将 AES_SR 的 CCF 标志转换为 1 进行指示。或者，使用相应的中断。
8. 通过将 AES_CR 寄存器中的 CCFC 位设置为 1，将 AES_SR 寄存器中的 CCF 标志清零。

GCM 标头阶段

GCM 初始化阶段之后的这个阶段必须在有效负载阶段之前完成。要执行的序列为（加密与解密完全相同）：

1. 通过将 AES_CR 寄存器中的 GCMPH[1:0] 位域设置为 01 来指示标头阶段。请勿修改初始化阶段中设置的 MODE[1:0] 位域。
2. 通过将 AES_CR 寄存器中的 EN 位置 1 来使能 AES 外设。
3. 如果它是最后一个块，并且块中的 AAD 大小低于 128 位，则用零填充块的其余部分。然后按*第 973 页的第 32.4.4 节：执行密码操作的 AES 过程*中所述的其中一种方式将数据块附加到 AES。
4. 重复步骤 *3*，直到处理完最后一个附加认证数据块。

注： *如无 AAD（即 Len(A) = 0），则可以跳过标头阶段。*

GCM 有效负载阶段

在 GCM 标头阶段之后执行此阶段（加密和解密相同）。在此阶段，加密/解密的有效负载将存储在 AES_DOUTR 寄存器中。要执行的序列为：

1. 如果跳过了标头阶段，则通过将 AES_CR 寄存器中的 EN 位置 1 来使能 AES 外设。
2. 通过将 AES_CR 寄存器中的 GCMPH[1:0] 位域设置为 10 来指示有效负载阶段。请勿修改初始化阶段中设置的 MODE[1:0] 位域。
3. 如果它是最后一个块，并且块中的明文（加密）或密文（解密）大小低于 128 位，则用零填充块的其余部分。
4. 按*第 973 页的第 32.4.4 节：执行密码操作的 AES 过程*中所述的其中一种方式将数据块附加到 AES，并读取结果。
5. 重复上一步，直到倒数第二个明文块被加密或直到最后一个密文块被解密。对于最后一个明文块（仅限加密），执行前两步。对于最后一个块，当其大小小于 16 字节时，将不属于有效负载的一部分的位丢弃。

注： *如无有效负载数据（即 Len(C) = 0），则可以跳过有效负载阶段（请参见 GMAC 模式）。*

GCM 最后阶段

在此最后一个阶段中，AES 外设会生成 GCM 认证标记并将其存储在 AES_DOUTR 寄存器中。要执行的序列为：

1. 通过将 AES_CR 寄存器中的 GCMPH[1:0] 位域设置为 11 来指示最后阶段。通过将 AES_CR 寄存器中的 MODE[1:0] 位域设置为 00 来选择加密模式。
2. 通过连接 AAD 位长度和有效负载位长度来组成块的数据，如*表 197* 中所示。将块写入 AES_DINR 寄存器。
3. 等待计算完成，通过将 AES_SR 的 CCF 标志转换为 1 进行指示。
4. 对 AES_DOUTR 寄存器进行四次读取操作，以获取 GCM 认证标记。
5. 通过将 AES_CR 寄存器中的 CCFC 位设置为 1，将 AES_SR 寄存器中的 CCF 标志清零。
6. 通过将 AES_CR 寄存器中的 EN 位清零来禁止 AES 外设。如果它是经验证的解密，请将生成的标记与通过消息传递的预期标记进行比较。

注： *在最后阶段，必须正常插入数据（无交换）。*

当从报头或有效负载阶段过渡到最后阶段时，不得禁止 AES 外设，否则结果会是错误的。

GCM 模式下的挂起/恢复操作

要挂起消息处理，请执行以下操作：

1. 如果使用 DMA，则通过将 AES_CR 寄存器中的 DMAINEN 位清零来停止 AES DMA 对 IN FIFO 的数据传输。如果未使用 DMA，确保当前计算已完成，完成由 AES_SR 寄存器的 CCF 标志设置为 1 指示。
2. 在有效负载阶段，如果未使用 DMA，则读取四次 AES_DOUTR 寄存器以保存最后处理的块。如果使用 DMA，等待 AES_SR 寄存器中的 CCF 标志置 1，然后通过将 AES_CR 寄存器中的 DMAOUTEN 输出位清零来停止 DMA 对 OUT FIFO 的数据传输。
3. 通过将 AES_CR 寄存器中的 CCFC 位设置为 1，将 AES_SR 寄存器中的 CCF 标志清零。
4. 将 AES_SUSPxR 寄存器保存到存储器中，其中 x 等于 0 到 7。
5. 在有效负载阶段，保存 AES_IVRx 寄存器，因为在数据处理过程中，它们相较于初始值发生更改。在标头阶段，则不需要此步骤。

6. 通过将 AES_CR 寄存器中的 EN 位清零来禁止 AES 外设。
7. 将当前 AES 配置保存到存储器中（初始化向量寄存器 AES_IVRx 除外）。无需保存密钥寄存器，因为应用程序已知初始密钥值。
8. 如果使用 DMA，则保存 DMA 控制器状态 （指向 IN 数据传输的指针以及剩余字节数等）。在有效负载阶段，也必须保存指向 OUT 数据传输的指针。

要恢复消息处理，请执行以下操作：

1. 如果使用 DMA，则重新配置 DMA 控制器以完成 FIFO IN 传输的其余部分。在有效负载阶段，也必须在 DMA 控制器中配置 FIFO OUT 的其余部分。
2. 确保 AES 外设已禁止（AES_CR 寄存器的 EN 位必须为 0）。
3. 将先前保存在存储器中的挂起寄存器值回写到其相应的 AES_SUSPxR 寄存器中，其中 x 等于 0 到 7。
4. 在有效负载阶段，将先前保存在存储器中的初始化向量寄存器值写回到其相应的 AES_IVRx 寄存器中。在标头阶段，将初始化设置值写回到 AES_IVRx 寄存器中。
5. 恢复 AES_CR 和 AES_KEYRx 寄存器中的初始化设置值。
6. 通过将 AES_CR 寄存器中的 EN 位置 1 来使能 AES 外设。
7. 如果使用 DMA，将 AES_CR 寄存器中的 DMAINEN 位（和 DMAOUTEN 位，如果在有效负载阶段）置 1，以使能 AES DMA 请求。

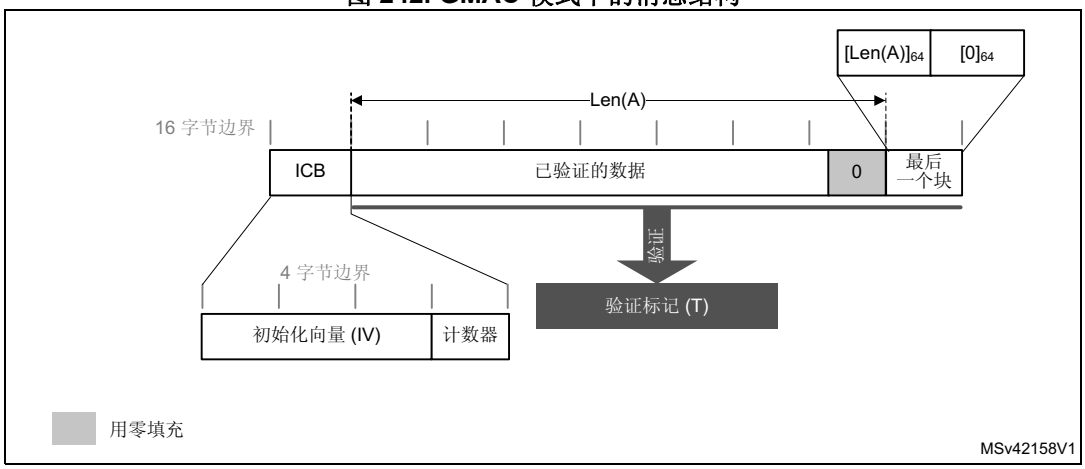
32.4.11 AESGalois 消息认证码 (GMAC)

概述

Galois 消息认证码 (GMAC) 支持认证明文，生成相应的标记信息（也称为消息认证码）。它基于 GCM 算法，NIST *特别出版物 800-38D，块密码工作模式的建议 - Galois/ 计数器模式 (GCM) 和 GMAC* 中对此进行了相关定义。

图 242 给出了 GMAC 下的典型消息结构。

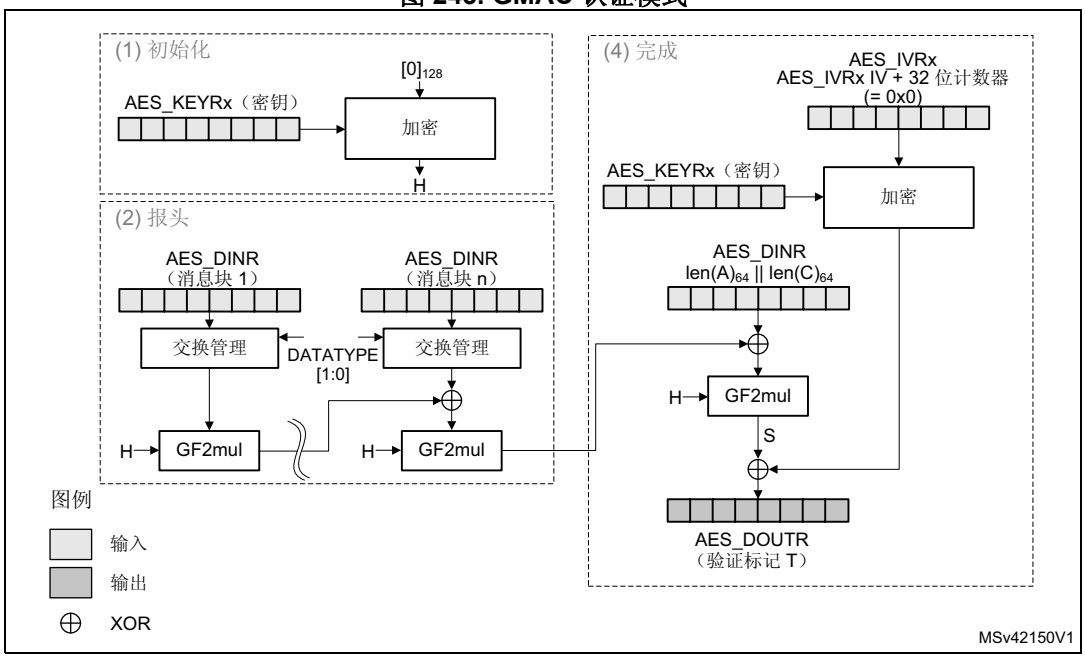
图 242. GMAC 模式下的消息结构



AES GMAC 处理

图 243 列出了 AES 外设中的 GMAC 模式实现。通过向 AES_CR 寄存器中的 CHMOD[2:0] 位域写入 011 来选择此模式。

图 243. GMAC 认证模式



GMAC 算法相当于应用在仅包含标头的消息上的 GCM 算法。因此，除了忽略有效负载阶段之外，所有步骤和设置均与 GCM 相同。

GMAC 模式下的挂起/恢复操作

在 GMAC 模式下，除了只能中断标头阶段外，针对 GCM 描述的序列适用。

32.4.12 AES CBC-MAC 计数器模式 (CCM)

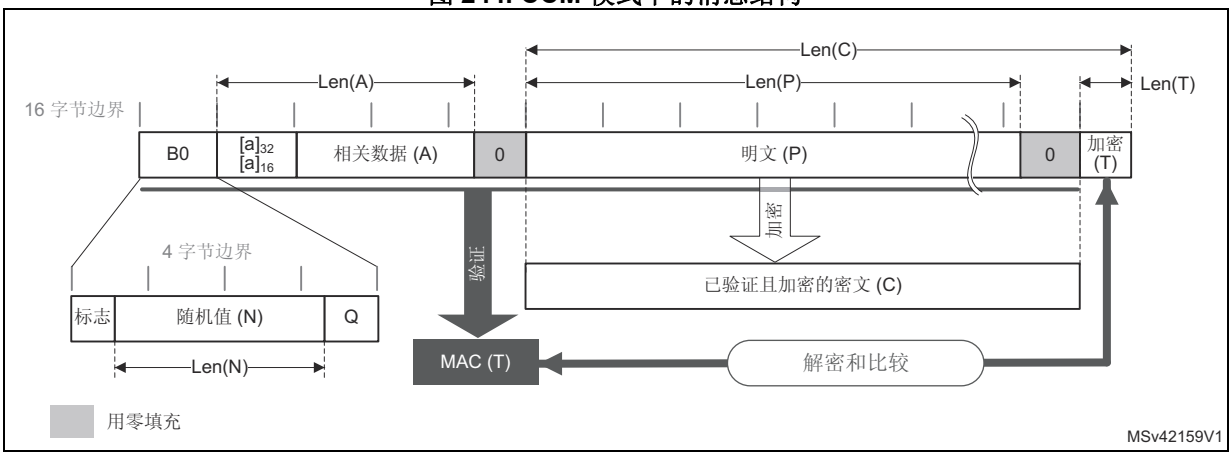
概述

AES **密码块链接-消息认证码计数器模式** (CCM) 算法可用于加密和认证明文，生成对应的密文和标记（也称为消息认证码）。为实现保密性，CCM 算法基于 AES 计数器模式。它使用密码块链接技术来生成消息认证码。这通常被称为 CBC-MAC。

注： *NIST 不允许将该 CBC-MAC 用作 CCM 规范上下文之外的认证模式。*

NIST *特别出版物 800-38C，块密码工作模式的建议 - CCM 模式实现认证和保密性*中对 CCM 链接进行了介绍。*图 244* 给出了 CCM 下的典型消息结构。

图 244. CCM 模式下的消息结构



消息的结构为：

- **16 字节的首次认证块 (B0)**，它由三个不同的字段组成：
  - **Q**：位字符串，表示 P 的八位字节长度 (Len(P))
  - **随机值 (N)**：大小为 Len(N) 的一次性值（即，应为每一次新的通信分配新的随机值）。Len(N) + Len(P) 的和必须等于 15 个字节。
  - **标志**：最高有效八位字节，包含四个控制信息标志（根据标准规定）。它包含两个 3 位字符串，用于编码值 **t**（MAC 长度，以字节表示）和 **Q**（明文长度，例如 Len(P) < 2^{8q} 个字节）。与 **Q** 相关的计数器块范围为 2^{8Q-4}，即，如果 **Q** 的最大值为 8，则密码中使用的计数器块应为 60 位。
- 与相关数据 (A) 关联的 **16 字节块 (B)**。
  消息的这一部分仅经过认证，未被加密。这部分具有已知长度 Len(A)，此长度值可以不是 16 字节的倍数（请参见*图 244*）。标准还具有以下规定：对于第一个消息块 (B1) 的 MSB 位，以字节表示的相关数据长度 (a) 必须按如下所述进行编码：
  - 如果 $0 < a < 2^{16} - 2^8$，则将其编码为 $[a]_{16}$，即两个字节。
  - 如果 $2^{16} - 2^8 < a < 2^{32}$，则将其编码为 0xff || 0xfe || $[a]_{32}$，即六个字节。
  - 如果 $2^{32} < a < 2^{64}$，则将其编码为 0xff || 0xff || $[a]_{64}$，即十个字节。
- 与明文消息 P 相关的 **16 字节块 (B)**，此明文消息将经过认证并被加密为密文 C（具有已知长度 Len(P)）。此长度可以不是 16 字节的倍数（请参见*图 244*）。
- 长度为 Len(T) 的**加密 MAC (T)** 被附加到总长度为 Len(C) 的密文 C。

当消息某一部分（A 或 P）的长度不是 16 字节的倍数时，需要采取特殊填充方案。

注： *CCM 链接模式同样只能与相关数据搭配使用（即，无有效负载）。*

例如，NIST 特别出版物 800-38C 的 C.1 部分给出了以下值（十六进制数）：

N: 10111213 141516（Len(N)= 56 位或 7 字节）

A: 00010203 04050607（Len(A)= 64 位或 8 字节）

P: 20212223（Len(P)= 32 位或 4 字节）

T: 6084341B（Len(T)= 32 位或 t = 4）

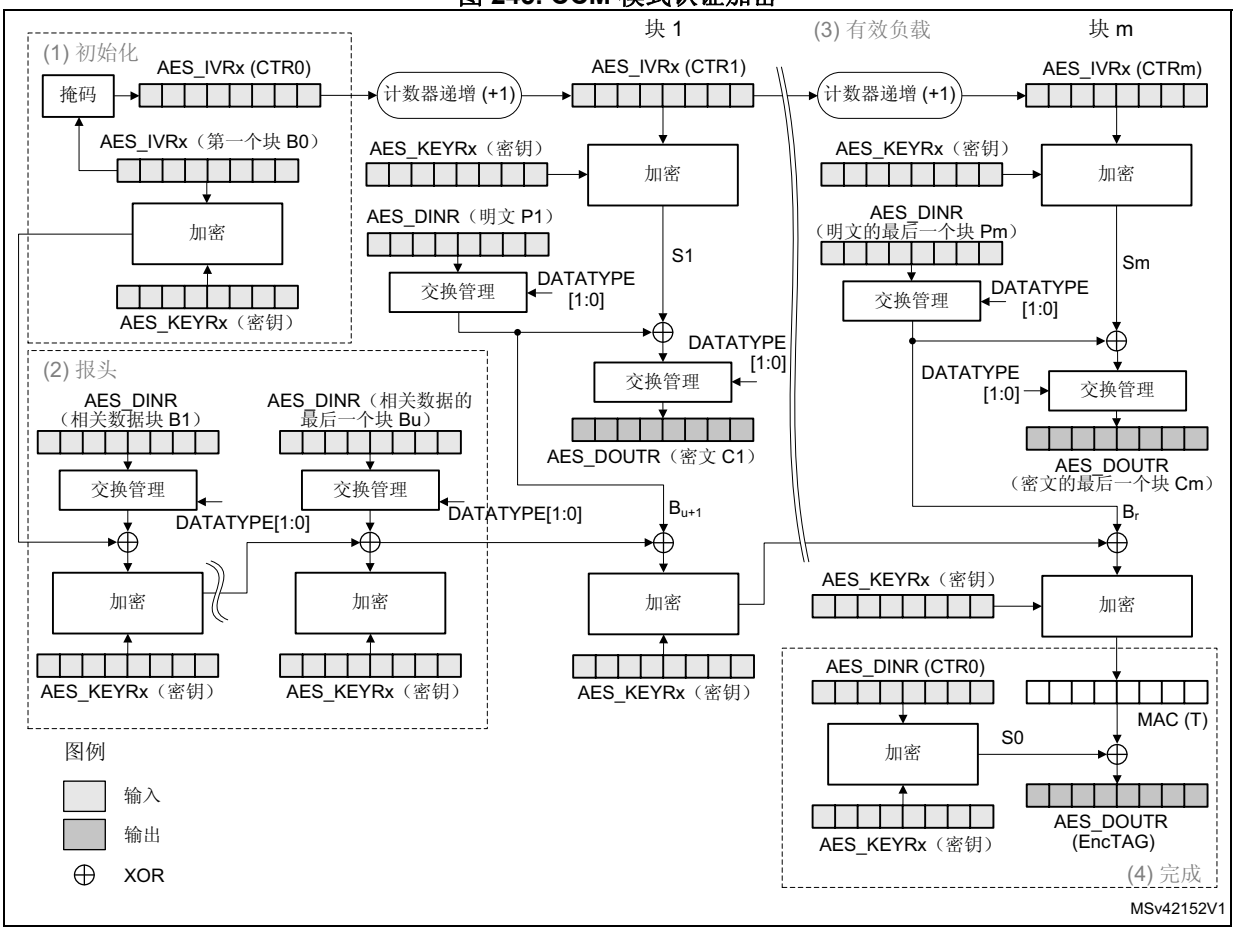
```
B0: 4F101112 13141516 00000000 00000004
B1: 00080001 02030405 06070000 00000000
B2: 20212223 00000000 00000000 00000000
CTR0: 0710111213 141516 00000000 00000000
CTR1: 0710111213 141516 00000000 00000001
```

格式化输入数据块 Bx（特别是 B0 和 B1）的生成必须由应用程序管理。

CCM 处理

图 245 列出了 AES 外设中的 CCM 实现（加密示例）。通过向 AES_CR 寄存器中的 CHMOD[2:0] 位域写入 100 来选择此模式。

图 245. CCM 模式认证加密



生成-加密过程的数据输入包括一个有效随机值、一个有效的有效负载字符串和一个有效的相关数据字符串，这些数据均经过正确格式化。对经过格式化的明文数据应用 CBC 链接机制可生成长度已知的 MAC。计数器模式加密需要足够长的计数器块序列作为输入，其将应用于有效负载字符串并单独应用于 MAC。得到的密文 C 是明文 P 的生成-加密过程的输出。

AES_IVRx 寄存器用于处理每个数据块，AES 以第一个块 B0 定义的位长度自动使 CTR 计数器递增。*表 199* 显示了应用程序必须如何加载 B0 数据。

注： *根据 NIST 规定，CCM 模式下的 AES 外设最多支持 64 位计数器。*

表 199. 在 CCM 模式下初始化 AES_IVRx 寄存器

寄存器	AES_IVR3[31:0]	AES_IVR2[31:0]	AES_IVR1[31:0]	AES_IVR0[31:0]
输入数据	B0[31:0]	B0[63:32]	B0[95:64]	B0[127:96]

注： *在 CCM 模式下，禁止将 MODE[1:0] 位域设置为 01 和 11（密钥分散）。*

CCM 消息通过以下阶段进行处理（将在下一小节中对此进行进一步介绍）：

- **初始化阶段**：AES 处理第一个块并准备第一个计数器块。
- **标头阶段**：AES 仅使用标记计算处理相关数据 (A)。
- **有效负载阶段**：IP 基于标记计算、计数器块加密和数据异或运算处理明文 (P)。其操作方式与密文 (C) 类似。
- **最后阶段**：AES 生成消息认证码 (MAC)。

CCM 初始化阶段

在此阶段中，将 CCM 消息的第一个块 B0 写入 AES_IVRx 寄存器中。AES_DOUTR 寄存器不包含任何输出数据。建议序列为：

1. 确保 AES 外设已禁止（AES_CR 的 EN 位必须为 0）。
2. 通过将 AES_CR 寄存器的 CHMOD[2:0] 位域设置为 100 选择 CCM 链接模式，并可选择将 DATATYPE[1:0] 位域置 1。
3. 通过将 AES_CR 寄存器中的 GCMPH[1:0] 位域设置为 00 来指示初始化阶段。
4. 将 AES_CR 寄存器的 MODE[1:0] 位域设置为 00 或 10。
5. 使用密钥初始化 AES_KEYRx 寄存器，并如*表 199* 所述使用 B0 数据初始化 AES_IVRx 寄存器。
6. 通过将 AES_CR 寄存器的 EN 位设置为 1 来启动计数器计算（计算完成时 EN 自动复位）。
7. 等待计算完成，通过将 AES_SR 的 CCF 标志转换为 1 进行指示。或者，使用相应的中断。
8. 通过将 AES_CR 寄存器中的 CCFC 位设置为 1，将 AES_SR 寄存器中的 CCF 标志清零。

CCM 标头阶段

GCM 初始化阶段之后的这个阶段必须在有效负载阶段之前完成。在此阶段期间，AES_DOUTR 寄存器不包含任何输出数据。

要执行的序列为（加密与解密完全相同）：

1. 通过将 AES_CR 寄存器中的 GCMPH[1:0] 位域设置为 01 来指示标头阶段。请勿修改初始化阶段中设置的 MODE[1:0] 位域。
2. 通过将 AES_CR 寄存器中的 EN 位置 1 来使能 AES 外设。
3. 如果它是最后一个块，并且块中的 AAD 大小低于 128 位，则用零填充块的其余部分。然后按*第 973 页的第 32.4.4 节：执行密码操作的 AES 过程*中所述的其中一种方式将数据块附加到 AES。
4. 重复步骤 *3*，直到处理完最后一个附加认证数据块。

注： *如无相关数据（即 Len(A) = 0），则可以跳过标头阶段。*

必须由软件使用相关数据长度格式化相关数据的首个块 (B1)。

CCM 有效负载阶段（加密或解密）

在 CCM 标头阶段之后执行此阶段（加密和解密相同）。在此阶段，加密/解密的有效负载将存储在 AES_DOUTR 寄存器中。要执行的序列为：

1. 通过将 AES_CR 寄存器中的 GCMPH[1:0] 位域设置为 10 来指示有效负载阶段。请勿修改初始化阶段中设置的 MODE[1:0] 位域。
2. 通过将 AES_CR 寄存器中的 EN 位置 1 来使能 AES 外设。
3. 如果它是最后一个待加密的数据块，并且块中明文大小低于 128 位，则用零填充块的其余部分。
4. 按*第 973 页的第 32.4.4 节：执行密码操作的 AES 过程*中所述的其中一种方式将数据块附加到 AES，并读取结果。
5. 重复上一步，直到倒数第二个明文块被加密或直到最后一个密文块被解密。对于最后一个明文块（仅限加密），应用两步。对于最后一个块，当其大小小于 16 字节时，将不属于有效负载的一部分的位丢弃。

注： *如果没有有效负载数据，即当 Len(P) = 0 或 Len(P) = Len(T) 时，可以跳过有效负载阶段。*

解密密文 C 时，移除 $LSB_{Len(T)}(C)$ 加密标记信息。

CCM 最后阶段

在此最后一个阶段中，AES 外设会生成 GCM 认证标记并将其存储在 AES_DOUTR 寄存器中。要执行的序列为：

1. 通过将 AES_CR 寄存器中的 GCMPH[1:0] 位域设置为 11 来指示最后阶段。通过将 AES_CR 寄存器中的 MODE[1:0] 位域保持为 00 来选择加密模式。
2. 等待 AES_SR 寄存器的计算完成标志 CCF 置 1。
3. 读取 AES_DOUTR 寄存器四次：此输出即为 CCM 认证标记。
4. 通过将 AES_CR 寄存器中的 CCFC 位设置为 1，将 AES_SR 寄存器中的 CCF 标志清零。
5. 通过将 AES_CR 寄存器中的 EN 位清零来禁止 AES 外设。
6. 对于认证解密，将生成的加密标记与密文中填充的加密标记进行比较。

注： *在此最后阶段，正常读取数据（无交换）。*

当从报头阶段过渡到最后阶段时，不得禁止 AES 外设，否则结果会是错误的。

应用程序必须使用标记长度屏蔽认证标记输出以获取有效标记。

CCM 模式下的挂起/恢复操作

要在标头或有效负载阶段挂起消息处理，请执行以下操作：

1. 如果使用 DMA，则通过将 AES_CR 寄存器中的 DMAINEN 位清零来停止 AES DMA 对 IN FIFO 的数据传输。如果未使用 DMA，确保当前计算已完成，完成由 AES_SR 寄存器的 CCF 标志设置为 1 指示。
2. 在有效负载阶段，如果未使用 DMA，则读取四次 AES_DOUTR 寄存器以保存最后处理的块。如果使用 DMA，等待 AES_SR 寄存器中的 CCF 标志置 1，然后通过将 AES_CR 寄存器中的 DMAOUTEN 输出位清零来停止 DMA 对 OUT FIFO 的数据传输。
3. 通过将 AES_CR 寄存器中的 CCFC 位设置为 1，将 AES_SR 寄存器中的 CCF 标志清零。
4. 将 AES_SUSPxR 寄存器（其中 x 等于 0 到 7）保存到存储器中。
5. 保存 AES_IVRx 寄存器，因为在数据处理过程中，它们相较于初始值发生更改。
6. 通过将 AES_CR 寄存器中的 EN 位清零来禁止 AES 外设。
7. 将当前 AES 配置保存到存储器中（初始化向量寄存器 AES_IVRx 除外）。无需保存密钥寄存器，因为应用程序已知初始密钥值。
8. 如果使用 DMA，则保存 DMA 控制器状态 （指向 IN 数据传输的指针以及剩余字节数等）。在有效负载阶段，也必须保存指向 OUT 数据传输的指针。

要恢复消息处理，请执行以下操作：

1. 如果使用 DMA，则重新配置 DMA 控制器以完成 FIFO IN 传输的其余部分。在有效负载阶段，也必须在 DMA 控制器中配置 FIFO OUT 的其余部分。
2. 确保 AES 外设已禁止（AES_CR 寄存器的 EN 位必须为 0）。
3. 将先前保存在存储器中的挂起寄存器值回写到其相应的 AES_SUSPxR 寄存器中（其中 x 等于 0 到 7）。
4. 将先前保存在存储器中的初始化向量寄存器值写回到其相应的 AES_IVRx 寄存器中。
5. 恢复 AES_CR 和 AES_KEYRx 寄存器中的初始化设置值。
6. 通过将 AES_CR 寄存器中的 EN 位置 1 来使能 AES 外设。
7. 如果使用 DMA，将 AES_CR 寄存器中的 DMAINEN 位置 1（如果在有效负载阶段，还要将 DMAOUTEN 位置 1），以使能 AES DMA 请求。

32.4.13 AES 数据寄存器和数据交换

数据输入和输出

通过将四个连续的 32 位字写入 AES_DINR 寄存器（位域 DIN[127:0]），将 128 位数据块输入到 AES 外设，先最高有效字（位 [127:96]），后最低有效字（位 [31:0]）。

通过从 AES_DOUTR 寄存器（位域 DOUT[127:0]）读取四个连续的 32 位字，从 AES 外设检索 128 位数据块，先最高有效字（位 [127:96]），后最低有效字（位 [31:0]）。

写入 AES_DINR 寄存器或从 AES_DOUTR 寄存器读取的 32 位数据字使用大端模式进行组织，即：

- 要写入 AES_DINR 的字的最高有效字节必须存储在保留要写入的字的四个相邻存储单元的最低地址中，或
- 从 AES_DOUTR 读取的字的最高有效字节存储在接收字的四个相邻存储单元的最低地址中

要使用 DMA 将输入数据块写入 AES，输入块的四个字必须以大端模式连续存储在存储器中，即最高有效字存储在最低地址中。请参见*第 32.4.16 节：AES DMA 接口*。

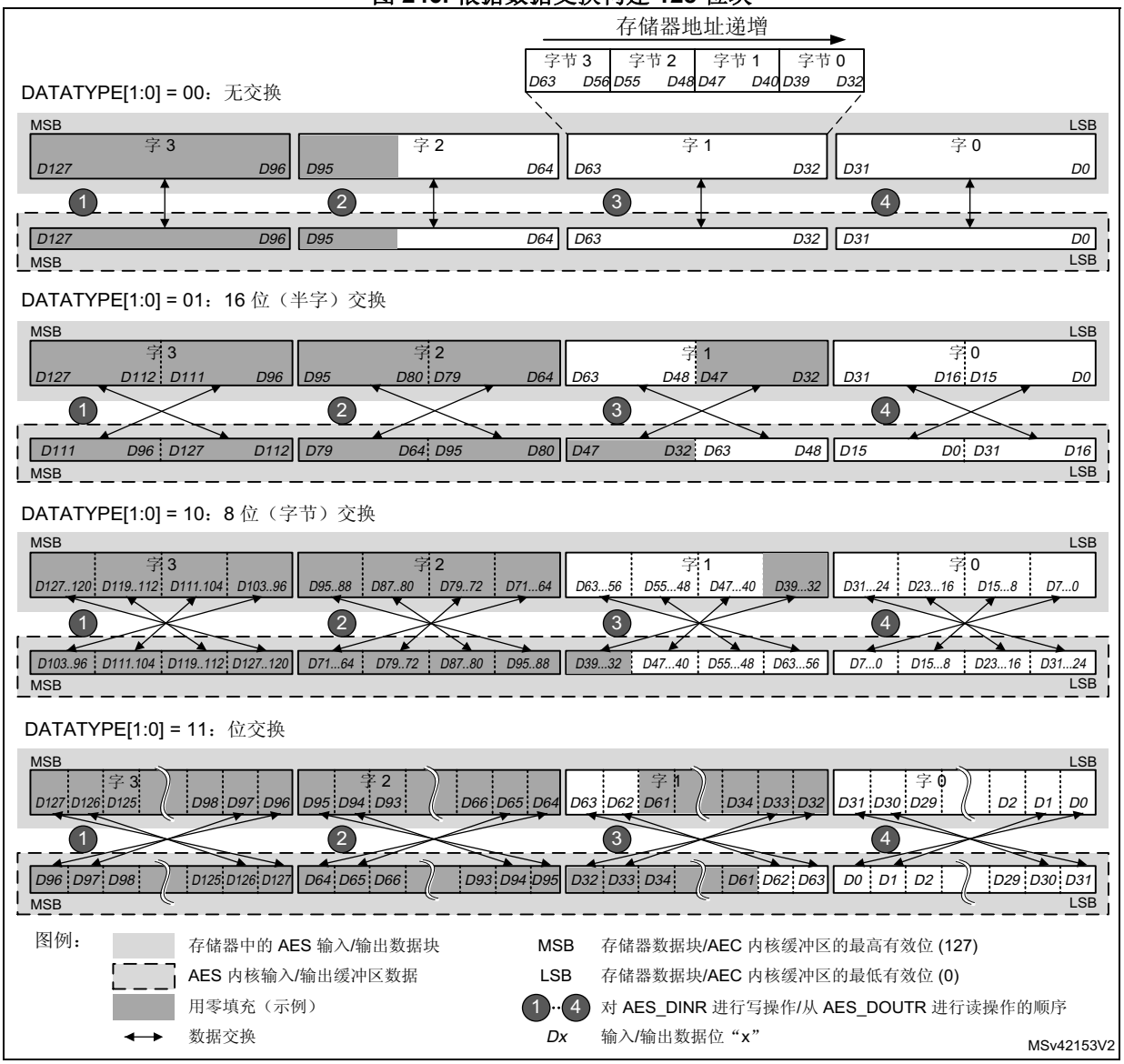
数据交换

可将 AES 外设配置为在将其加载到 AES 处理内核前对 AES_DINR 寄存器中的输入数据字，以及在将其发送到 AES_DOUTR 寄存器前对 AES 处理内核的输出数据，执行位、字节、半字或无交换。选择取决于数据类型。例如，对 ASCII 文本流使用字节交换。

通过 AES_CR 寄存器中的 DATATYPE[1:0] 位域来选择数据交换类型。该选择适用于 AES 内核的输入和输出。

对于不同的数据交换类型，*图 246* 显示了通过 AES_DINR 寄存器输入的输入数据的 AES 处理内核输入缓冲区数据 P127..0 的结构，或通过 AES_DOUTR 寄存器的输出数据的 AES 处理内核输出缓冲区数据 P127..0 的结构。

图 246. 根据数据交换构建 128 位块



注： *AES 密钥寄存器 (AES_KEYRx) 和初始化寄存器 (AES_IVRx) 中的数据对所选交换模式不敏感。*

数据填充

图 246 还给出了一个用零填充存储器数据块的示例，这样数据交换后的补零位在 AES 内核输入缓冲区的 MSB 端形成一个连续区域。该示例显示了输入数据块的填充，包含：

- 48 个消息位，DATATYPE[1:0] = 01
- 56 个消息位，DATATYPE[1:0] = 10
- 34 个消息位，DATATYPE[1:0] = 11

32.4.14 AES 密钥寄存器

AES_KEYRx 寄存器存储加密或解密密钥位域 KEY[127:0] 或 KEY[255:0]。要写入每个寄存器或从中读取的数据以小端模式组织在存储器中，即最高有效字节存储在最高地址中。

密钥分布在八个寄存器中，如*表 200* 中所示。

表 200. AES_KEYRx 寄存器中的密钥字节序（128 位或 256 位密钥长度）

AES_KEYR7 [31:0]	AES_KEYR6 [31:0]	AES_KEYR5 [31:0]	AES_KEYR4 [31:0]	AES_KEYR3 [31:0]	AES_KEYR2 [31:0]	AES_KEYR1 [31:0]	AES_KEYR0 [31:0]
-	-	-	-	KEY[127:96]	KEY[95:64]	KEY[63:32]	KEY[31:0]
KEY[255:224]	KEY[223:192]	KEY[191:160]	KEY[159:128]	KEY[127:96]	KEY[95:64]	KEY[63:32]	KEY[31:0]

AES 外设禁止时，加密或解密的密钥可以写入这些寄存器。

密钥寄存器不受由 AES_CR 寄存器的 DATATYPE[1:0] 位域控制的数据交换影响。

32.4.15 AES 初始化向量寄存器

四个 AES_IVRx 寄存器保持初始化向量输入位域 IVI[127:0]。要写入每个寄存器或从中读取的数据以小端模式组织在存储器中，即最高有效字节存储在最高地址中。寄存器也以最低地址 (AES_IVR0) 到最高地址 (AES_IVR3) 的顺序排序。

位域中数据的有效性取决于选择的链接模式。使用时，位域在 AES 内核的每个计算周期内都会更新。

AES 外设使能时，对 AES_IVRx 寄存器的写入操作对寄存器内容没有影响。要修改 AES_IVRx 寄存器的内容，必须首先清零 AES_CR 寄存器的 EN 位。

读取 AES_IVRx 寄存器会返回最新计数器值（用于管理挂起模式）。

AES_IVRx 寄存器不受由 AES_CR 寄存器的 DATATYPE[1:0] 位域控制的数据交换功能的影响。

32.4.16 AES DMA 接口

AES 外设使用一个接口连接 DMA（直接存储器访问）控制器。DMA 操作通过 AES_CR 寄存器进行控制。

使用 DMA 的数据输入

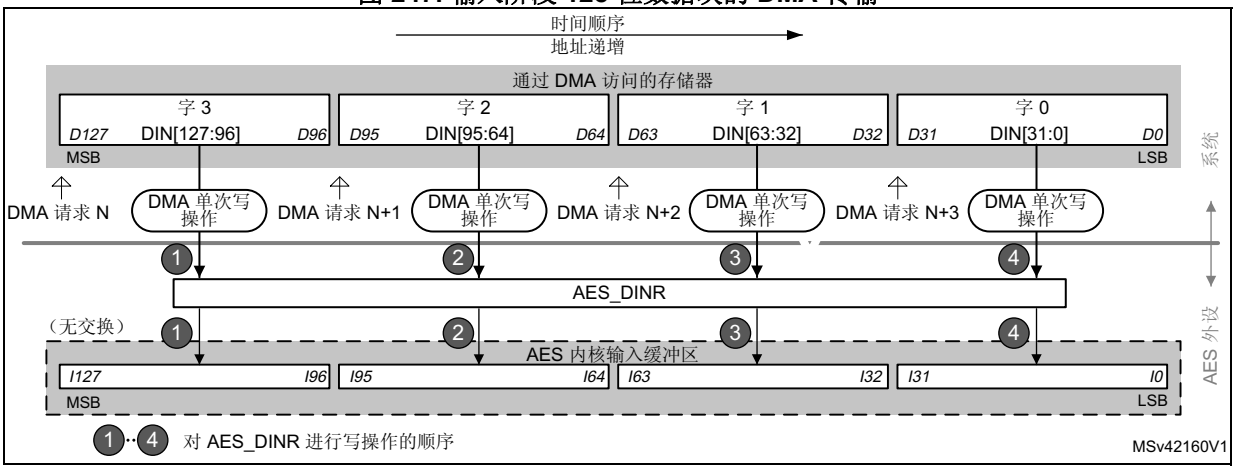
通过将 AES_CR 寄存器的 DMAINEN 位置 1，可使能 DMA 写入 AES。然后，AES 外设每次需要向 AES_DINR 寄存器中写入字时，都会在输入阶段发出 DMA 请求。它使能四个 DMA 请求，传输存储器的一个 128 位（四字）输入数据块，如*图 247* 中所示。

有关推荐的 DMA 配置，请参见*表 201*。

表 201. 用于存储器到 AES 数据传输的 DMA 通道配置

DMA 通道控制寄存器字段	推荐配置
传输数据量大小	消息长度：128 位的倍数。 根据所选算法和模式，可能需要采用特殊填充/密文窃取技术。例如在 AES GCM 加密或 AES CCM 解密情况下，DMA 传输不得包括最后一个块。有关详细信息，请参见*第 32.4.4 节：执行密码操作的 AES 过程*。
源突发大小（存储器）	单
目标突发大小（外设）	单
DMA FIFO 大小	AES FIFO_size = 4 个字节
源传输宽度（存储器）	32 位字
目标传输宽度（外设）	32 位字
源地址递增（存储器）	是，每次 32 位传输后
目标地址递增（外设）	AES_DINR 的固定地址（无递增）

图 247. 输入阶段 128 位数据块的 DMA 传输



使用 DMA 的数据输出

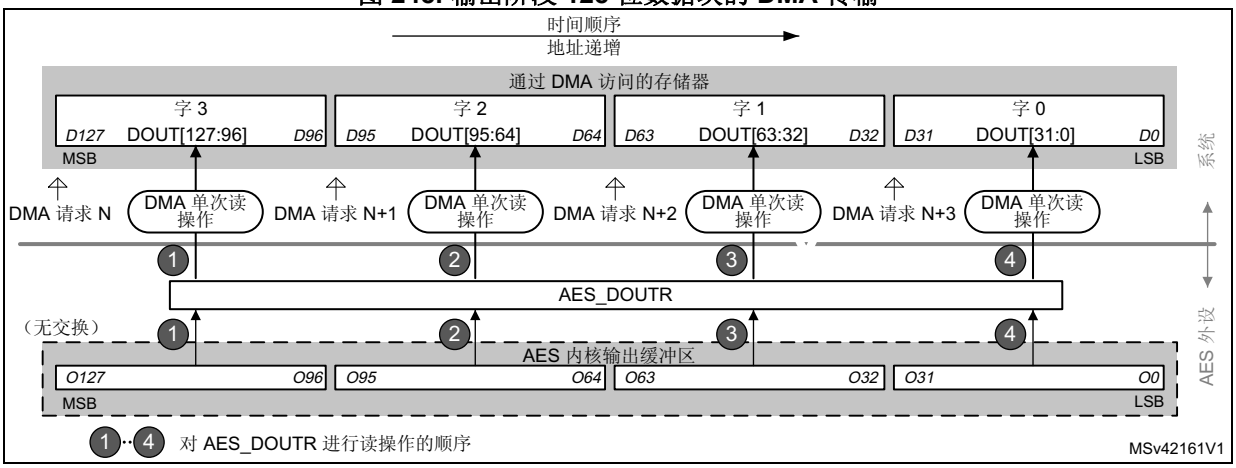
通过将 AES_CR 寄存器的 DMAOUTEN 位置 1，可使能 DMA 从 AES 读取。然后，AES 外设每次需要从 AES_DOUTR 寄存器中读取字时，都会在输出阶段发出 DMA 请求。它使能四个 DMA 请求，将一个 128 位（四字）输出数据块传输到存储器，如*图 248* 中所示。

有关推荐的 DMA 配置，请参见*表 202*。

表 202. 用于 AES 到存储器数据传输的 DMA 通道配置

DMA 通道控制寄存器字段	推荐配置
传输数据量大小	消息长度，AES 块大小（4 个字）的倍数。根据情况，需要丢弃额外的字节
源突发大小（外设）	单
目标突发大小（存储器）	单
DMA FIFIO 大小	AES FIFO_size = 4 个字节
源传输宽度（外设）	32 位字
目标传输宽度（存储器）	32 位字
源地址递增（外设）	AES_DINR 的固定地址（无递增）
目标地址递增（存储器）	是，每次 32 位传输后

图 248. 输出阶段 128 位数据块的 DMA 传输



不同工作模式下的 DMA 操作

当通过 AES_CR 寄存器的 MODE[1:0] 位域选择模式 1（加密）或模式 3（解密）时，DMA 操作可用。在模式 2（密钥生成）下，AES_KEYRx 寄存器必须通过软件写入，通过 AES_CR 寄存器的 DMAINEN 和 DMAOUTEN 位使能 DMA 传输在该模式下无效。

AES 会一直产生 DMA 单次请求，直到被禁用。因此，128 位数据块处理结束时的数据输出阶段之后，AES 会自动切换到下一个数据块的新数据输入阶段（如果存在）。

当 AES 和存储器之间的数据传输由 DMA 管理时，CCF 标志无意义，可通过软件忽略（置 1）。只有在转换回由软件管理的数据传输时才能清零此标志。请参见*第 32.4.8 节：AES 基本链接模式（ECB 和 CBC）*中的 *ECB/CBC 模式下的挂起/恢复操作*作为示例。

32.4.17 AES 错误管理

如果检测到未预期的读取或写入操作，则会将 AES_SR 寄存器中的读取错误标志 (RDERR) 和写入错误标志 (WRERR) 位置 1。如果 AES_CR 寄存器中的错误中断使能 (ERRIE) 位置 1，则会产生中断。有关详细信息，请参见*第 32.5 节：AES 中断*。

注： *检测到错误后，AES 不会禁止，而是会继续处理。*

可随时通过将 AES_CR 寄存器中的 EN 位清零再置 1 来重新初始化 AES。

读取错误标志 (RDERR)

如果在计算阶段或输入阶段检测到未预期的读取操作时，则会将 AES_SR 寄存器中的 AES 读取错误标志 (RDERR) 置 1。如果 AES_CR 寄存器中的 ERRIE 位置 1，则会产生中断。

通过将 AES_CR 寄存器中的相应 ERRC 位置 1 来清零 RDERR 标志。

写入错误标志 (WDERR)

如果在计算阶段或输出阶段检测到未预期的写入操作时，则会将 AES_SR 寄存器中的 AES 写入错误标志 (WRERR) 置 1。如果 AES_CR 寄存器中的 ERRIE 位置 1，则会产生中断。

通过将 AES_CR 寄存器中的相应 ERRC 位置 1 来清零 WDERR 标志。

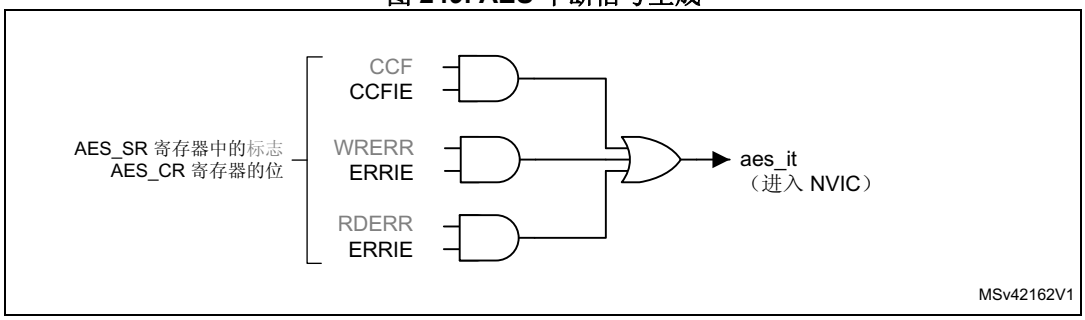
32.5 AES 中断

AES 外设可生成三个独立的可屏蔽中断源，用以通知发生以下事件：

- 计算完成
- 读取错误，请参见*第 32.4.17 节*
- 写入错误，请参见*第 32.4.17 节*

这三个中断源合并为一个连接到 NVIC（嵌套向量中断控制器）的通用中断信号 aes_it。

图 249. AES 中断信号生成



可通过将 AES_CR 寄存器的相应使能位置 1/清零，分别使能/禁止每个 AES 中断源。请参见 *图 249*。

可以从 AES_SR 寄存器中读取各个可屏蔽中断源的状态。

表 203 对中断源及其事件标志和使能位进行了总结。

表 203. AES 中断请求

AES 中断事件	事件标志	使能位
计算完成标志	CCF	CCFIE
读取错误标志	RDERR	ERRIE
写入错误标志	WRERR	ERRIE

32.6 AES 处理延迟

下表列出了每种工作模式下处理 128 位块的延迟。

表 204. ECB、CBC 和 CTR 模式下的处理延迟（以时钟周期计）

密钥大小	工作模式	算法	输入阶段 + FSM 集	计算阶段	输出阶段	总计
128 位	模式 1：加密	ECB、CBC、CTR	9	38	4	**51**
	模式 2：密钥生成	-	-	59	-	**59**
	模式 3：解密	ECB、CBC、CTR	9	38	4	**51**
	模式 4：密钥分散和解密	ECB、CBC	9	93	4	**106**
256 位	模式 1：加密	ECB、CBC、CTR	13	58	4	**75**
	模式 2：密钥分散	-	-	82	-	**82**
	模式 3：解密	ECB、CBC、CTR	13	58	4	**75**
	模式 4：密钥生成和解密	ECB、CBC	13	128	4	**145**

表 205. GCM 和 CCM 模式下的处理延迟（以时钟周期计）

密钥大小	工作模式	算法	初始化阶段	标头阶段	有效负载阶段	标记阶段
128 位	模式 1：加密/ 模式 3：解密	GCM	64	35	51	59
		CCM	63	55	114	58
256 位	模式 1：加密/ 模式 3：解密	GCM	88	35	75	75
		CCM	87	79	162	82

注： *数据插入可以包括 AES 在 AHB 总线上强制的等待周期（最多 3 个周期，通常为 1 个周期）。这适用于所有标头/有效负载/标记阶段。*

32.7 AES 寄存器

32.7.1 AES 控制寄存器 (AES_CR)

AES control 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
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	NPBLB[3:0]				Res.	KEYSIZE	Res.	CHMOD[2]
								rw					rw		rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	GCMPH[1:0]		DMAOUTEN	DMAINEN	ERRIE	CCFIE	ERRC	CCFC	CHMOD[1:0]		MODE[1:0]		DATATYPE[1:0]		EN
	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:24 保留，必须保持为零

位 23:20 **NPBLB[3:0]**：最后一个块中填充字节的数量 (Number of padding bytes in last block)

该位域用于设置有效负载的最后一个块中填充字节的数量：
0000：所有字节均有效（无填充）
0001：填充最后一个块的一个最低有效字节
...
1111：填充最后一个块的 15 个最低有效字节

位 19 保留，必须保持为零。

位 18 **KEYSIZE**：密钥大小选择 (Key size selection)

此位域定义了 AES 加密内核中使用的密钥长度（以位为单位）。
0: 128
1: 256

仅当 AES 禁止时才允许更改位值，以避免不可预测的行为。

位 17 保留，必须保持为零

位 16 **CHMOD[2]**：链接模式选择，位 [2] (Chaining mode selection, bit [2])

有关 CHMOD[2:0] 位域的说明，请参见寄存器的位 [5:6]

位 15 保留，必须保持为零

位 14:13 **GCMPH[1:0]**：GCM 或 CCM 阶段选择 (GCM or CCM phase selection)

该位域选择 GCM、GMAC 或 CCM 算法的阶段：
00：初始化阶段
01：标头阶段
10：有效负载阶段
11：最后阶段

如果选择 GCM、GMAC 或 CCM 以外的算法（通过 ALGOMODE 位域），则该位域不起作用。

位 12 **DMAOUTEN**：DMA 输出使能 (DMA output enable)

该位在输出阶段使能/禁止使用 DMA 进行数据传输：

0：禁止

1：使能

该位置 1 时，AES 会在输出数据阶段自动生成 DMA 请求。该功能仅在通过 MODE[1:0] 位域选择了模式 1 或模式 3 时才有效。它不适用于模式 2（密钥生成）。

不建议在模式 4 下（单次解密）使用 DMA。

位 11 **DMAINEN**：DMA 输入使能 (DMA input enable)

该位在输入阶段使能/禁止使用 DMA 进行数据传输：

0：禁止

1：使能

该位置 1 时，AES 会在输入数据阶段自动生成 DMA 请求。该功能仅在通过 MODE[1:0] 位域选择了模式 1 或模式 3 时才有效。它不适用于模式 2（密钥生成）。

不建议在模式 4 下（单次解密）使用 DMA。

位 10 **ERRIE**：错误中断使能 (Error interrupt enable)

当 RDERR 和/或 WRERR 置 1 时，该位使能或禁止（屏蔽）AES 中断：

0：禁止（屏蔽）

1：使能

位 9 **CCFIE**：CCF 中断使能 (CCF interrupt enable)

当 CCF（计算完成标志）置 1 时，该位使能或禁止（屏蔽）AES 中断：

0：禁止（屏蔽）

1：使能

位 8 **ERRC**：错误标志清零 (Error flag clear)

写入 1 后，该位将 AES_SR 寄存器中的 RDERR 和 WRERR 错误标志清零：

0：不起作用

1：清零 RDERR 和 WRERR 标志

读取标志始终返回零。

位 7 **CCFC**：计算完成标志清零 (Computation complete flag clear)

写入 1 后，该位将 AES_SR 寄存器中的计算完成标志 (CCF) 清零：

0：不起作用

1：清零 CCF

读取标志始终返回零。

位 6:5 **CHMOD[1:0]**：链接模式选择，位 [1:0](Chaining mode selection, bits [1:0])
这些位与位 CHMOD[2]（请参见此寄存器的位 16）一起构成 CHMOD[2:0] 位域，用于选择 AES 链接模式：
000：电子密码本 (ECB)
001：密码块链接 (CBC)
010：计数器模式 (CTR)
011：Galois 计数器模式 (GCM) 和 Galois 消息认证码 (GMAC)
100：CBC-MAC 计数器模式 (CCM)
>100：保留

仅当 AES 禁止时才允许更改位域值，以避免不可预测的行为。

位 4:3 **MODE[1:0]**：AES 工作模式 (AES operating mode)
此位域选择 AES 工作模式：
00：模式 1：加密
01：模式 2：密钥生成（或针对 ECB/CBC 解密准备密钥）
10：模式 3：解密
11：模式 4：密钥生成和单次解密

仅当 AES 禁止时才允许更改位域值，以避免不可预测的行为。在未选择 ECB 或 CBC 链接模式时选择模式 4 的任何尝试，都将默认为有效选择模式 3。在模式 4 之后不能选择模式 3。

位 2:1 **DATATYPE[1:0]**：数据类型选择 (Data type selection)
此位域通过选择数据交换模式，定义了写入 AES_DINR 寄存器或从 AES_DOUTR 寄存器中读取的数据的格式：
00：无
01：半字（16 位）
10：字节（8 位）
11：位

有关详细信息，请参见*第 32.4.13 节：AES 数据寄存器和数据交换*。
仅当 AES 禁止时才允许更改位域值，以避免不可预测的行为。

位 0 **EN**：AES 使能 (AES enable)
该位用于使能/禁止 AES 外设：
0：禁止
1：使能

可随时通过将该位清零再置 1 来重新初始化 AES 外设。
当密钥准备过程结束时（模式 2），该位由硬件自动清零。

32.7.2 AES 状态寄存器 (AES_SR)

AES 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
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BUSY	WRERR	RDERR	CCF
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:4 保留，必须保持为零

位 3 **BUSY**：忙 (Busy)

该标志指示 AES 在 GCM 有效负载**加密**阶段是空闲还是繁忙：
0：空闲
1：忙

该标志由硬件控制。当标志指示“空闲”时，可以挂起当前消息处理以处理更高优先级的消息。
该标志仅在 GCM 有效负载加密阶段有效。在其他链接模式下，或在除有效负载加密外的 GCM 阶段，必须忽略该标志。

位 2 **WRERR**：写入错误 (Write error)

此标志指示检测到 AES_DINR 寄存器中发生了未预期的写入操作（计算阶段或数据输出阶段）：
0：未检测到
1：已检测到

该标志由硬件置 1。此位用软件清零，方法是将 AES_CR 寄存器中的 ERRC 位置 1。
标志置 1 时，如果通过 AES_CR 寄存器的 ERRIE 位使能，则会产生中断。
标志置 1 对 AES 操作没有影响。
当选择密钥生成模式或 GCM/CCM 初始化阶段时，该标志无效。

位 1 **RDERR**：读取错误标志 (Read error flag)

此标志指示检测到 AES_DOUTR 寄存器中发生了未预期的读取操作（计算阶段或数据输入阶段）：
0：未检测到
1：已检测到

该标志由硬件置 1。此位用软件清零，方法是将 AES_CR 寄存器中的 ERRC 位置 1。
标志置 1 时，如果通过 AES_CR 寄存器的 ERRIE 位使能，则会产生中断。
标志置 1 对 AES 操作没有影响。
当选择密钥生成模式或 GCM/CCM 初始化/标头阶段时，该标志无效。

位 0 **CCF**：计算完成标志 (Computation completed flag)

该标志指示计算是否完成：
0：未完成
1：已完成

计算完成后，该标志由硬件置 1。此位用软件清零，方法是将 AES_CR 寄存器中的 CCFC 位置 1。
标志置 1 时，如果通过 AES_CR 寄存器的 CCFIE 位使能，则会产生中断。
仅当 DMAOUTEN 位为 0 时，该标志才有效。DMA_EN 为 1 时，该位可能保持为 1。

32.7.3 AES 数据输入寄存器 (AES_DINR)

AES data input register

偏移地址：0x08

复位值：0x0000 0000

仅支持 32 位访问类型。

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

位 31:0 **DIN[x+31:x]**：写入外设的 128 位输入数据块的四个 32 位字之一

该位域响应一个 32 位输入缓冲区。在输入阶段，对该位域进行 4 倍连续写入，几乎可将一个完整的 128 位输入数据块写入 AES 外设。在每次写入时，数据交换块根据 DATATYPE[1:0] 位域对输入缓冲区的数据进行处理，然后写入 AES 内核 128 位输入缓冲区。

第一次到第四次写入操作分别将“x”替换为：96、64、32 和 0。也就是说，第一次到第四次写入操作的数据为：DIN[127:96]、DIN[95:64]、DIN[63:32] 和 DIN[31:0]。

输入数据块的数据有效性取决于 AES 工作模式：

- **模式 1**（加密）：明文
- **模式 2**（密钥生成）：未使用该位域（将 AES_KEYRx 寄存器用于输入）
- **模式 3**（解密）和**模式 4**（密钥生成和单次解密）：密文

*第 997 页的第 32.4.13 节：AES 数据寄存器和数据交换*对数据交换操作进行了介绍。

32.7.4 AES 数据输出寄存器 (AES_DOUTR)

AES data output register

偏移地址：0x0C

复位值：0x0000 0000

仅支持 32 位访问类型。

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

位 31:0 **DOUT[x+31:x]**：从外设读取的 128 位输出数据块的四个 32 位字之一

该只读位域获取一个 32 位输出缓冲区。计算完成后（CCF 置 1），对该位域进行 4 倍连续读取，几乎可从 AES 外设读取一个完整的 128 位输出数据块。在到达输出缓冲区之前，数据交换块根据 DATATYPE[1:0] 位域对 AES 内核产生的数据进行处理。

第一次到第四次读取操作分别将 DOUT[x+31:x] 替换为：96、64、32 和 0。也就是说，第一次到第四次读取操作的数据为：DOUT[127:96]、DOUT[95:64]、DOUT[63:32] 和 DOUT[31:0]。

输出数据块的数据有效性取决于 AES 工作模式：

- **模式 1**（加密）：密文
- **模式 2**（密钥生成）：未使用该位域（将 AES_KEYRx 寄存器用于输出）
- **模式 3**（解密）和**模式 4**（密钥生成和单次解密）：明文

*第 997 页的第 32.4.13 节：AES 数据寄存器和数据交换*对数据交换操作进行了介绍。

32.7.5 AES 密钥寄存器 0 (AES_KEYR0)

AES key register 0

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
KEY[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
KEY[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:0 **KEY[31:0]**：加密密钥 (Cryptographic key)，位 [31:0]

该位域包含 AES 加密或解密密钥的位 [31:0]，具体取决于工作模式：

- **模式 1**（加密）、**模式 2**（密钥分散）和**模式 4**（密钥生成和单次解密）：写入位域的值是加密密钥。
- **模式 3**（解密）：写入位域的值是在用于解密之前生成的加密密钥。将加密密钥写入位域后，在使能 AES 之前的读取返回相同的值。使能 AES 后且将 CCF 标志置 1 后的读取返回从加密密钥生成的解密密钥。

注： 在模式 4（密钥分散和单次解密）中，位域总是包含加密密钥。

只有在禁止 AES 外设时，才能向 AES_KEYRx 寄存器执行写操作。

更多详细信息，请参见*第 999 页的第 32.4.14 节：AES 密钥寄存器*。

32.7.6 AES 密钥寄存器 1 (AES_KEYR1)

AES key register 1

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
KEY[63:48]															
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
KEY[47:32]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:0 **KEY[63:32]**：加密密钥 (Cryptographic key)，位 [63:32]
有关 KEY[255:0] 位域的说明，请参见 AES_KEYR0 寄存器。

32.7.7 AES 密钥寄存器 2 (AES_KEYR2)

AES key register 2

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
KEY[95:80]															
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
KEY[79:64]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:0 **KEY[95:64]**：加密密钥 (Cryptographic key)，位 [95:64]
有关 KEY[255:0] 位域的说明，请参见 AES_KEYR0 寄存器。

32.7.8 AES 密钥寄存器 3 (AES_KEYR3)

AES key register 3

偏移地址：0x1C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
KEY[127:112]															
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
KEY[111:96]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:0 **KEY[127:96]**：加密密钥 (Cryptographic key)，位 [127:96]
有关 KEY[255:0] 位域的说明，请参见 AES_KEYR0 寄存器。

32.7.9 AES 初始化向量寄存器 0 (AES_IVR0)

AES initialization vector register 0

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
IVI[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
IVI[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:0 **IVI[31:0]**：初始化向量输入 (Initialization vector input)，位 [31:0]
有关 IVI[127:0] 位域的说明，请参见*第 999 页的第 32.4.15 节：AES 初始化向量寄存器*。
初始化向量仅用于除 ECB 以外的链接模式。
只有在禁止 AES 外设时，才能向初始化向量执行写操作。

32.7.10 AES 初始化向量寄存器 1 (AES_IVR1)

AES initialization vector register 1

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
IVI[63:48]															
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
IVI[47:32]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:0 **IVI[63:32]**：初始化向量输入 (Initialization vector input)，位 [63:32]
有关 IVI[128:0] 位域的说明，请参见 AES_IVR0 寄存器。

32.7.11 AES 初始化向量寄存器 2 (AES_IVR2)

AES initialization vector register 2

偏移地址：0x28

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IVI[95:80]															
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
IVI[79:64]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:0 **IVI[95:64]**：初始化向量输入 (Initialization vector input)，位 [95:64]
有关 IVI[128:0] 位域的说明，请参见 AES_IVR0 寄存器。

32.7.12 AES 初始化向量寄存器 3 (AES_IVR3)

AES initialization vector register 3

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
IVI[127:112]															
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
IVI[111:96]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:0 **IVI[127:96]**：初始化向量输入 (Initialization vector input)，位 [127:96]
有关 IVI[128:0] 位域的说明，请参见 AES_IVR0 寄存器。

32.7.13 AES 密钥寄存器 4 (AES_KEYR4)

AES key register 4

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
KEY[159:144]															
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
KEY[143:128]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:0 **KEY[159:128]**：加密密钥 (Cryptographic key)，位 [159:128]
有关 KEY[255:0] 位域的说明，请参见 AES_KEYR0 寄存器。

32.7.14 AES 密钥寄存器 5 (AES_KEYR5)

AES key register 5

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
KEY[191:176]															
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
KEY[175:160]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:0 **KEY[191:160]**：加密密钥 (Cryptographic key)，位 [191:160]
有关 KEY[255:0] 位域的说明，请参见 AES_KEYR0 寄存器。

32.7.15 AES 密钥寄存器 6 (AES_KEYR6)

AES key register 6

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
KEY[223:208]															
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
KEY[207:192]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:0 **KEY[223:192]**：加密密钥 (Cryptographic key)，位 [223:192]
有关 KEY[255:0] 位域的说明，请参见 AES_KEYR0 寄存器。

32.7.16 AES 密钥寄存器 7 (AES_KEYR7)

AES key register 7

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
KEY[255:240]															
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
KEY[239:224]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:0 **KEY[255:224]**：加密密钥 (Cryptographic key)，位 [255:224]
有关 KEY[255:0] 位域的说明，请参见 AES_KEYR0 寄存器。

注： *仅当选择 256 位密钥长度时，才使用密钥寄存器 4 到密钥寄存器 7。当选择 128 位密钥长度时，它们不起作用（此时仅使用密钥寄存器 0 到密钥寄存器 3）。*

32.7.17 AES 挂起寄存器 (AES_SUSPxR)

AES suspend registers

偏移地址：0x040 + x * 0x4，（x = 0 到 7）

复位值：0x0000 0000

当前任务的 AES 处理被挂起以处理更高优先级的任务时，这些寄存器包含 AES 处理器的完整内部寄存器状态。
挂起后，软件在将 AES 处理器用于更高优先级的任务前，读取 AES_SUSPxR 寄存器内容（其中 x 等于 0 到 7）并将其保存到存储器中。完成后，软件在恢复原始任务前将保存的内容恢复到相应的挂起寄存器中。

注： *这些寄存器仅在选择了 GCM、GMAC 或 CCM 链接模式时才可使用。*

仅当 AES 使能时才能读取这些寄存器。当 AES 禁止时读取这些寄存器将返回 0x0000 0000。

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

位 31:0 **SUSPx**：AES 挂起 (AES suspend)
挂起操作后，每个 AES_SUSPxR 寄存器的此位域取内部 AES 寄存器之一的值。

32.7.18 AES 寄存器映射

表 206. AES 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
0x0000	AES_CR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	NPBLB				Res.	KEYSIZE	Res.	CHMOD[2]	Res.	GCMPH[1:0]		DMAOUTEN	DMAINEN	ERRIE	CCFIE	ERRC	CCFC	CHMOD[1:0]		MODE[1:0]		DATATYPE[1:0]		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
0x0004	AES_SR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BUSY	WRERR	RDERR	CCF
	Reset value																													0	0	0	0
0x0008	AES_DINR x=96,64,32,0	DIN[x+31: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
0x000C	AES_DOUTR x=96,64,32,0	DOUT[x+31: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
0x0010	AES_KEYR0	KEY[31:0]																															
	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
0x0014	AES_KEYR1	KEY[63:32]																															
	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
0x0018	AES_KEYR2	KEY[95:64]																															
	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
0x001C	AES_KEYR3	KEY[127:96]																															
	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
0x0020	AES_IVR0	IVI[31:0]																															
	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
0x0024	AES_IVR1	IVI[63:32]																															
	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
0x0028	AES_IVR2	IVI[95:64]																															
	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
0x002C	AES_IVR3	IVI[127:96]																															
	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
0x0030	AES_KEYR4	KEY[159:128]																															
	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
0x0034	AES_KEYR5	KEY[191:160]																															
	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
0x0038	AES_KEYR6	KEY[223:192]																															
	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
0x003C	AES_KEYR7	KEY[255:224]																															
	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
0x0040	AES_SUSP0R	SUSP0[31:0]																															
	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
0x0044	AES_SUSP1R	SUSP1[31:0]																															
	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

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
0x0048	AES_SUSP2R	SUSP2[31:0]																															
	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
0x004C	AES_SUSP3R	SUSP3[31:0]																															
	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
0x0050	AES_SUSP4R	SUSP4[31:0]																															
	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
0x0054	AES_SUSP5R	SUSP5[31:0]																															
	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
0x0058	AES_SUSP6R	SUSP6[31:0]																															
	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
0x005C	AES_SUSP7R	SUSP7[31:0]																															
	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

33 散列处理器 (HASH)

33.1 简介

散列处理器完全兼容安全散列算法（SHA-1、SHA-224 和 SHA-256）、MD5（消息摘要算法 5）散列算法和适合多种应用的 HMAC（密钥散列消息认证码）散列算法。HMAC 适用于需要进行消息验证的应用程序。

散列处理器为最多 (261 - 1) 位的消息计算经 FIPS（联邦信息处理标准）认可的 160、224 和 256 位长度的摘要。此外，它还可以为 MD5 算法计算 128 位摘要。

33.2 散列主要特性

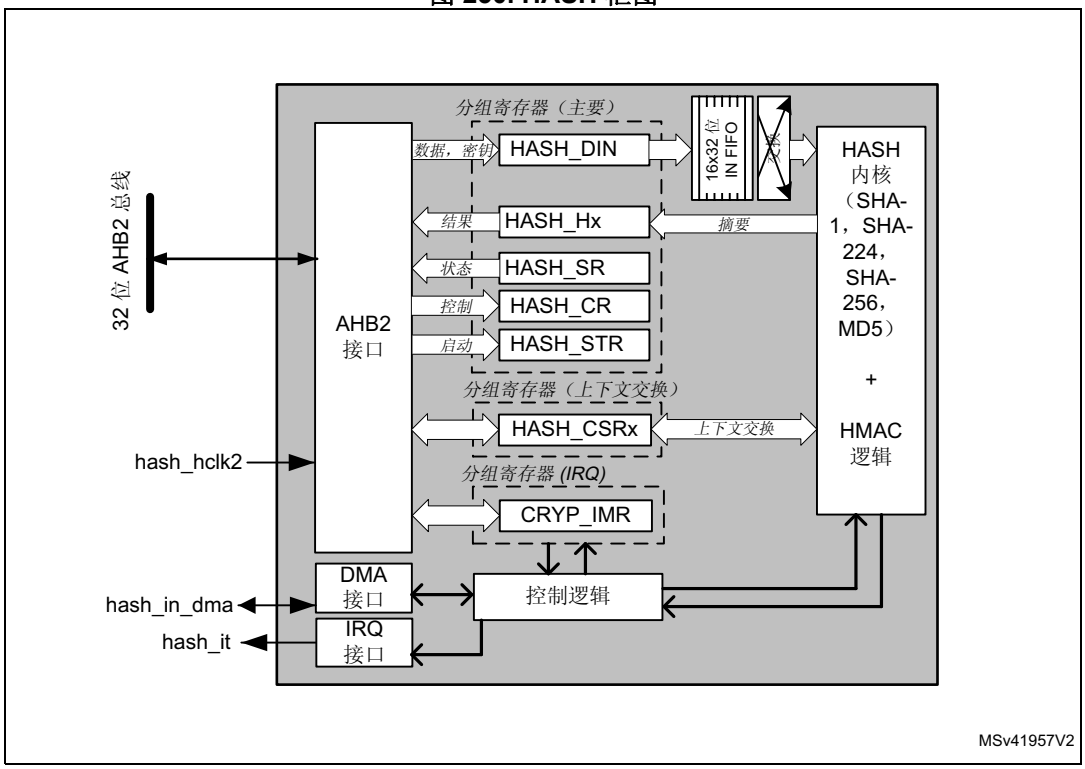
- 适合于数据验证应用，符合以下标准：
  - 联邦信息处理标准出版物 FIPS PUB 180-4 *安全散列标准*（SHA-1 和 SHA-2 系列）
  - 互联网工程任务组 (IETF) 征求意见 RFC 1321 *MD5 消息摘要算法*
  - 互联网工程任务组 (IETF) 征求意见 RFC 2104 *HMAC：密钥散列消息认证*和联邦信息处理标准出版物 FIPS PUB 198-1 *密钥散列消息认证码 (HMAC)*
- 连续消息块中摘要的对应 32 位字添加到彼此之中，以构成整个消息的摘要
  - 可自动进行 32 位字交换，以兼容输入位串的内部小端模式表示法
  - 支持的字交换：位、字节、半字和 32 位字
- 可自动填充来补足输入位串，从而适应 512 位（16×32 位）的摘要最小块大小
- 单个 32 位输入寄存器，它与十六个 32 位字的内部输入 FIFO 相关联，对应于一个块的大小
- 快速计算 SHA-1、SHA-224、SHA-256 以及 MD5
  - 使用 SHA-1（或 SHA-256）算法处理一个 512 位数据块需要 82（或 66）个时钟周期
  - 使用 MD5 算法处理一个 512 位数据块需要 66 个时钟周期
- AHB 从外设，仅支持 32 位字访问（否则会产生 AHB 错误）
- 8×32 位字（H0 到 H7）用于输出消息摘要
- 数据流自动控制，支持使用一个通道进行直接存储器访问 (DMA)。支持固定 4 个突发
- 每 32 位字可中断一次消息摘要计算
  - 可重载摘要寄存器
  - 散列计算挂起/恢复机制，包括使用 DMA

33.3 散列功能说明

33.3.1 HASH 框图

图 250 显示了散列处理器的框图。

图 250. HASH 框图



33.3.2 HASH 内部信号

表 207 列出了 HASH 级而非产品级（焊盘上）所提供的内部信号，对这些信号有所了解会很有用。

表 207. HASH 内部输入/输出信号

信号名称	信号类型	说明
hash_hclk2	数字输入	AHB2 总线时钟
hash_it	数字输出	散列处理器全局中断请求
hash_in_dma	数字输入/输出	DMA 突发请求/确认

33.3.3 关于安全散列算法

散列处理器完全兼容由 FIPS PUB 180-4 标准和 IETF RFC1321 出版物 (MD5) 定义的安全散列算法。

对于每个算法，HASH 计算消息或数据文件的压缩表示。更具体来讲，输入中提供的任何消息长度低于 2^{64} 位时，SHA-1、SHA-224、SHA-256 和 MD5 处理内核将分别生成一个 160 位、224 位、256 位和 128 位的输出位串，称为消息摘要。然后可以通过数字签名算法来处理此消息摘要，以便生成或验证消息的签名。

对消息摘要而不是对消息签名通常可提高处理的效率，因为消息摘要通常比消息要小得多。数字签名的验证程序和数字签名的创建程序必须使用相同散列算法。

散列处理器支持的 SHA-2 功能安全可靠，因为要找出与某个给定消息摘要对应的消息（自 2017 年 2 月起 SHA-1 不再安全可靠），或找出两个生成相同消息摘要的不同消息，在计算层面无法实现。对传输中的消息进行任何更改都极有可能产生不同的消息摘要，从而导致签名验证失败。

33.3.4 消息数据馈送

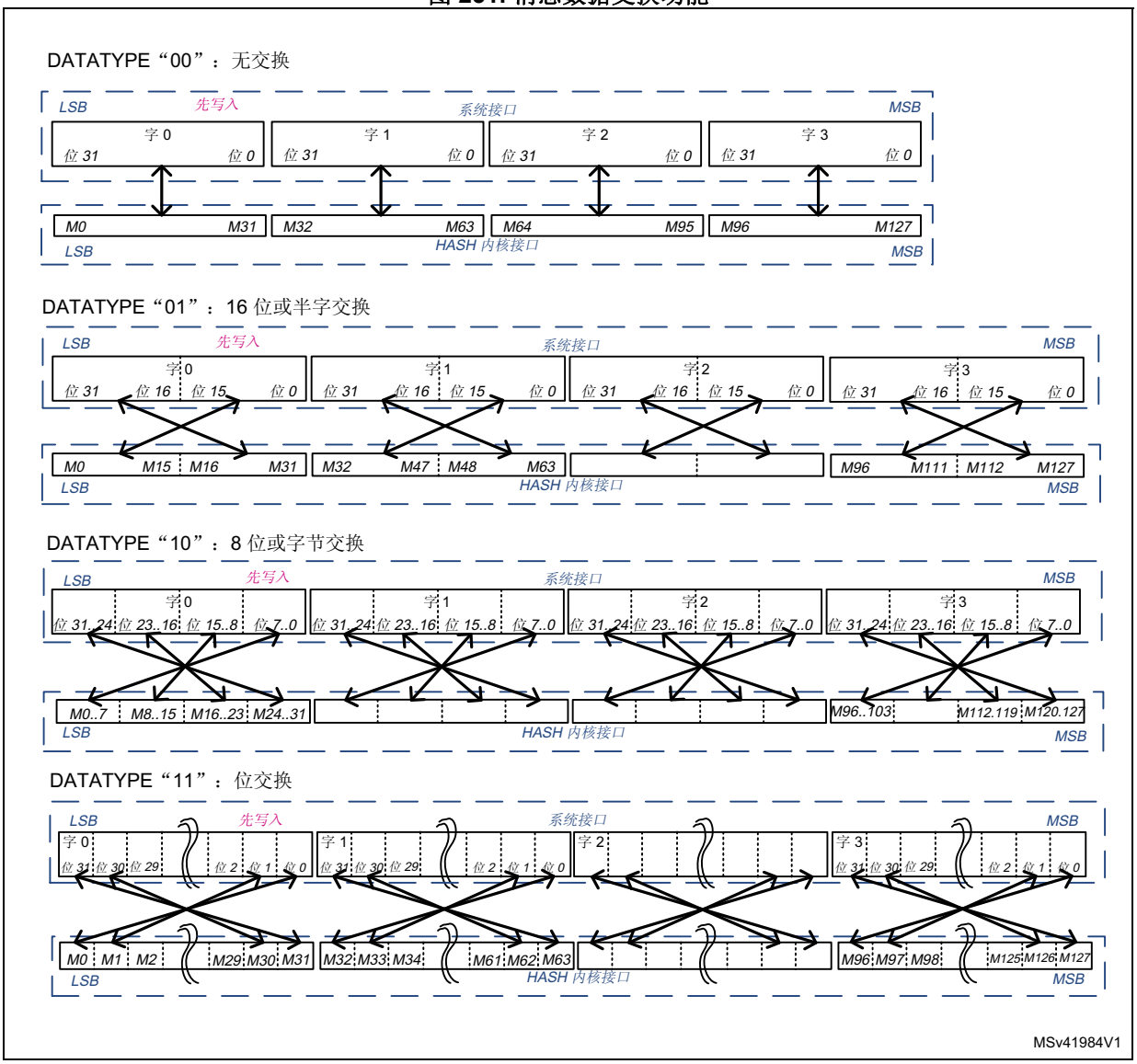
要由 HASH 处理的消息（或数据文件）应视为位串。根据 FIPS PUB 180-1 和 180-2 标准，该消息位串从左到右增长（采用十六进制字“大端”约定来表示），从而使得在每个字内，最高有效位存储在最左侧位的位置上。以位串表示为“01100001 01100010 01100011”的消息串“abc”为例，其 32 位字表示为 0x00636261，8 位字表示为 0x61626300。

通过将数据写入到 HASH_DIN 寄存器，数据将一次性输入到 HASH 的一个 32 位字中。每次在寄存器中写入新数据时，HASH_DIN 寄存器的当前内容将传输到 16 字输入 FIFO (IN FIFO)。因此，HASH_DIN 和输入 FIFO 构成一个长度为十七个 32 位字的 FIFO（名为 IN 缓冲器）。

根据要处理的数据类型（例如，当数据是 ASCII 文本流时的字节交换），必须先对从输入 FIFO 读取的数据进行位、字节、半字或无交换操作，然后再将其输入小端散列处理内核。*图 251* 显示了如何根据 HASH 控制寄存器 (HASH_CR) 中的 DATATYPE 位域，使用由驱动程序弹出到 IN FIFO 中的一个 32 位字构建散列处理内核 32 位数据块 M0...31。

当禁用位交换时（DATATYPE=“00”），HASH_DIN 数据字节序如下所述：消息的最低有效位必须在输入到散列处理器的第一个字中的 MSB 位置，位串的第 32 位必须在输入到散列处理器的第二个字中的 MSB 位置，依此类推。

图 251. 消息数据交换功能



33.3.5 消息摘要计算

当计算消息摘要时，散列处理器会按顺序处理 512 位块。这样，DMA 或 CPU 每次将 16 x 32 位字（= 512 位）写入到散列处理器时，散列处理器将自动开始计算消息摘要。此操作称为“部分摘要计算”。

如*第 33.3.4 节：消息数据馈送*中所述，待处理的消息将一次性输入到 HASH 32 位字中，从而写入 HASH_DIN 寄存器以填充输入 FIFO。应用程序应使用以下序列对该数据执行散列计算。

1. 使用 HASH_CR 寄存器初始化散列处理器：
   - 使用 ALGO 字段选择正确的算法。如果需要，可使用 HASH_CR 中的 DATATYPE 位域为消息输入字编程正确的交换操作。
   - 如果已选择 HMAC 模式，则设置 MODE = 1 并使用 LKEY 选择密钥长度。
   - 如果最后一个字中的有效位数不是 32 位，则更新 NBLW 以定义最后一个字中的有效位数。在这种情况下，HASH 可以应用自动填充。
2. 通过将 HASH_CR 中的 INIT 位置 1 来完成初始化。如果通过 DMA 传输数据，也将 DMAE 位置 1。

注意： 编程步骤 2 时，务必事先或同时设置正确的配置值（ALGO、DATATYPE、HMAC 模式、密钥长度、NBLW）。

3. 开始通过写入 HASH_DIN 寄存器来填充数据（自动通过 DMA 传输数据的情况除外）。请注意，只有当块的最后一个值输入 IN FIFO 后，才能开始处理块。部分或最终摘要计算的处理方式取决于数据馈送到处理器的方式：

   a) 通过软件填充数据时：
   - 当软件将额外的一个字（实际上是下一个块的第一个字）写入到 HASH_DIN 寄存器时，将触发部分摘要计算。一旦处理器再次准备就绪（HASH_SR 中的 DINIS = 1），软件即可将新数据写入 HASH_DIN。这种机制可防止 HASH 引入等待状态。
   - 当输入最后一个块并且软件向 DCAL 位写入 1 时，将触发最终摘要计算。如果消息长度不是 512 位的整数倍，则必须在写入 DCAL 位之前写入 HASH_STR 寄存器中的 NBLW 字段（有关详细信息，请参见*第 33.3.6 节*）。

   b) 通过 DMA 以单次 DMA 传输（MDMAT 位 =“0”）填充数据时：
   - 每次 FIFO 已满时，将自动触发部分摘要计算。
   - 当最后一个块传输到 HASH_DIN 寄存器后，将自动触发最终摘要计算（硬件将 DCAL 位置 1）。如果消息长度不是 512 位的整数倍，则必须在使能 DMA 之前写入 HASH_STR 寄存器中的 NBLW 字段（有关详细信息，请参见*第 33.3.6 节*）。

   c) 当使用多次 DMA 传输（MDMAT 位=“1”）填充数据时：
   - 可针对单次 DMA 传输触发部分摘要计算。但是，当最后一个块传输到 HASH_DIN 寄存器后，不会自动触发最终摘要计算（硬件不将 DCAL 位置 1）。这使得散列处理器能够接收新的 DMA 传输（作为此摘要计算的一部分）。为了启动最终摘要计算，软件必须在最后一次 DMA 传输之前将 MDMAT 位置 0，以便如同针对单次 DMA 传输（请参见之前的说明）一样触发最终摘要计算。

4. 计算完成后，可以按照*表 208* 中所述从输出寄存器读取摘要。

表 208. 散列处理器输出

算法	有效输出寄存器	最高有效位	摘要大小（位）
MD5	HASH_H0 到 HASH_H3	HASH_H0[31]	128
SHA-1	HASH_H0 到 HASH_H4	HASH_H0[31]	160
SHA-224	HASH_H0 到 HASH_H6	HASH_H0[31]	224
SHA-256	HASH_H0 到 HASH_H7	HASH_H0[31]	256

关于 HMAC 详细指令的更多信息，请参见*第 33.3.7 节：HMAC 运算*。

33.3.6 消息填充

概述

当计算消息的压缩表示时，循环执行将数据馈送到散列处理器（每 512 位块执行一次自动部分摘要计算）的过程，直到原始消息的最后几位被写入 HASH_DIN 寄存器。

因为消息的长度（位数）可以是任何整数值，因此写入到散列处理器的最后一个字的有效位数在 1 到 32 之间。这个最后一个字的有效位数必须写到 HASH_STR 突寄存器的 NBLW 位段中，以便消息填充能够在最终消息摘要计算之前得到正确处理。

填充处理

*第 33.3.5 节：消息摘要计算*中介绍了使能或禁止 DMA 时的详细填充序列。

填充示例

根据联邦信息处理标准 PUB 180-1 和 PUB 180-2 规定，消息填充包括添加一个“1”，后跟 *k* 个“0”以及一个 64 位整数（其数值等于消息长度 *L*（位数））。经过这三个填充操作后生成的消息长度为 *L* + 1 + *k* + 64（512 位的整数倍）。

对于散列处理器，“1”将添加到 HASH_DIN 寄存器中写入的最后一个字，位位置由 NBLW 位域定义，而其余更高的位将被清零（“0”）。

FIPS PUB180-2 提供的示例

我们假定原始消息是采用 ASCII 二进制编码格式的“abc”，长度 L=24：

```
字节 0    字节 1    字节 2    字节 3
01100001 01100010 01100011 UUUUUUUU
<-- 写入 HASH_DIN 的第一个字 -->
```

NBLW 必须以值 24 加载：“1”在位串中的位 24 处追加（在以上位串中从左到右开始计数），这对应于 HASH_DIN 寄存器中的位 31（小端模式约定）：

```
01100001 01100010 01100011 1UUUUUUU
```

由于 L = 24，因此以上位串中的位数是 25，并将追加 423 个“0”位，现在为 448 位。

这将生成十六进制（大端模式的字节字）：

```
61626380 00000000 00000000 00000000
00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000
00000000 00000000
```

将追加双字格式的消息长度值 L，即 00000000 00000018。因此，最后填充的消息为十六进制（大端模式的字节字）：

```
61626380 00000000 00000000 00000000
00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000018
```

如果散列处理器被编程为在 HASH_DIN 输入寄存器（HASH_CR 中的 DATATYPE = 10）中交换字节，则必须按照以下序列输入上述消息：

1. 0xUU636261 写入到 HASH_DIN 寄存器（其中的“U”意味着无关）。
2. 将 0x18 写入到 HASH_STR 寄存器（写入到 HASH_DIN 寄存器的最后一个字中的有效位数是 24，因为原始消息长度为 24 位）。
3. 将 0x10 写入到 HASH_STR 寄存器以启动消息填充（如上文所述）以及随后执行摘要计算。
4. 完成散列计算后，HASH_HRx 寄存器 (x = 0...4) 中的消息摘要可供 SHA-1 算法使用。对于此 FIPS 示例，预期值如下：

```
HASH_H0 = 0xA9993E36
HASH_H1 = 0x4706816A
HASH_H2 = 0xBA3E2571
HASH_H3 = 0x7850C26C
HASH_H4 = 0x9CD0D89D
```

33.3.7 HMAC 运算

概述

根据互联网工程任务组 *RFC2104，HMAC：密钥散列消息认证*规定，HMAC 算法以不可逆方式将正在处理的消息与用户所选的密钥进行绑定，从而用于消息验证。该算法由两个嵌套的散列运算组成：

$$HMAC(message) = Hash((key \mid pad) \text{ XOR } [0x5C]_n \mid Hash((key \mid pad) \text{ XOR } [0x36]_n \mid message))$$

其中：

- $[X]_n$表示 X 重复 *n* 次，其中的 *n* 等于底层散列函数数据块的大小（对于 SHA-1、SHA224、SHA-256、 MD5 散列算法是 512 位，即 n = 64）。
- pad 是将密钥扩展到上面定义的长度 *n* 所需的零序列。如果密钥长度大于 *n*，应用程序应首先使用 Hash() 函数对密钥进行散列处理，然后将得到的字节串用作 HMAC 的实际密钥。
- | 表示串联运算符

HMAC 处理

计算 HMAC 需要下列四个步骤：

1. 当 MODE 位为 1 且 ALGO 位已根据所需算法设好对应的值时，通过向 INIT 位写入 1 对块进行初始化。如果正在使用的密钥的长度超过 64 字节，则还必须将 LKEY 位置 1。在这种情况下，根据 HMAC 规范的要求，散列处理器将使用密钥的散列代替真实密钥。
2. 必须将用于内部散列函数的密钥提供给散列处理器：

   密钥加载操作与消息位串加载操作遵循相同的机制，即将密钥数据写入 HASH_DIN 并通过写入 HASH_STR 寄存器来完成传输。

注： *有关字节序的详细信息，可参见第 33.3.4 节：消息数据馈送。*

3. 输入了最后一个密钥字并且计算开始后，散列处理器便开始生成内部密钥材料。一旦完成此操作，即可接受消息位字符串，如*第 33.3.4 节：消息数据馈送*中所述。
4. 在最后一轮散列之后，散列处理器将返回“就绪”以表明其可以接收用于外部散列函数的密钥（通常，此密钥与用于内部散列函数的密钥相同）。在输入了密钥的最后一个字并且计算开始后，HMAC 结果存储于 HASH_H0...HASH_H7 寄存器。

注： *HMAC 本原的计算延迟取决于密钥和消息的长度，如第 33.5 节：HASH 处理时间中所述。*

HMAC 示例

下面是一个由 NIST 指定的 HMAC SHA-1 算法示例（HASH_CR 中的 ALGO = “00”且 MODE = “1”）。

我们假定原始消息是采用 ASCII 二进制编码格式的“Sample message for keylen=blocklen”（keylen=blocklen 时的示例消息），长度 L = 34 字节。如果 HASH 编程为无交换模式（HASH_CR 中的 DATATYPE = 00），则必须将以下数据按顺序加载至 HASH_DIN 寄存器：

1. **由 NIST 指定的内部散列密钥**输入（长度 = 64，即不填充）。由于密钥长度 = 64，HASH_CR 寄存器中的 LKEY 位置 0

```
00010203 04050607 08090A0B 0C0D0E0F 10111213 14151617
18191A1B 1C1D1E1F 20212223 24252627 28292A2B 2C2D2E2F
30313233 34353637 38393A3B 3C3D3E3F
```

2. **消息**输入（长度 = 34，即需要填充）。必须将 HASH_STR 设置为 0x20，以开始消息填充和内部散列计算（将“U”视为无关）

```
53616D70 6C65206D 65737361 67652066 6F72206B 65796C65
6E3D626C 6F636B6C 656EUUUU
```

3. **外部散列密钥**输入（长度 = 64，即不填充）。在此输入与内部散列密钥相同的密钥。
4. **最终外部散列计算**由 HASH 执行。HMAC SHA-1 摘要结果存储于 HASH_HRx 寄存器 (x = 0...4) 中，如下所示：

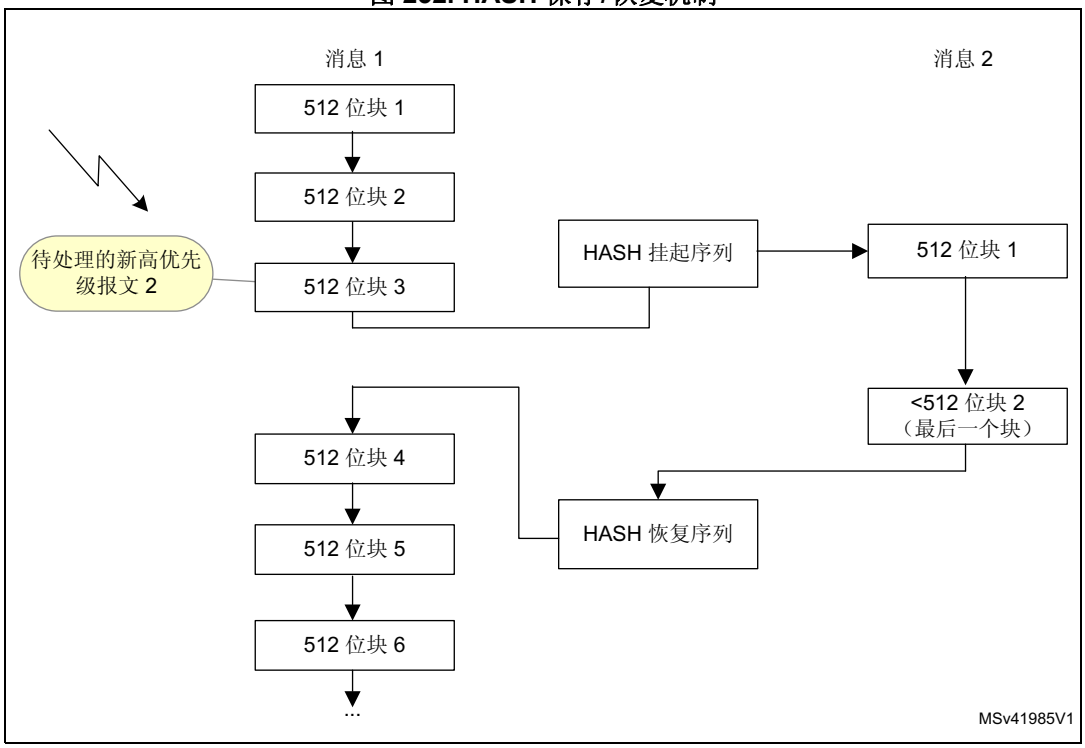
```
HASH_H0 = 0x5FD596EE
HASH_H1 = 0x78D5553C
HASH_H2 = 0x8FF4E72D
HASH_H3 = 0x266DFD19
HASH_H4 = 0x2366DA29
```

33.3.8 上下文交换

概述

可以中断散列/HMAC 操作以执行另一个优先级较高的处理。处理完优先级较高的任务后，可以继续完成之前中断的过程，如*图 252* 所示。

图 252. HASH 保存/恢复机制



要这样做，必须将被中断的任务的上下文从散列寄存器保存到存储器，然后从存储器恢复到散列寄存器。

以下说明由软件或 DMA 控制数据流的过程。

通过软件加载数据

如果不使用 DMA 将消息加载到散列处理器，则只有在当前未处理任何块时才能保存上下文。这意味着用户应用程序必须等待至 DINIS = 1（最后一个块已处理且输入 FIFO 为空）或 NBW ≠ 0（FIFO 未满且当前未进行任何处理）。详细程序如下所述。

- **当前上下文保存**

  在中断当前消息摘要计算之前，应用程序必须将以下寄存器的内容存储到存储器中：
  - HASH_IMR
  - HASH_STR
  - HASH_CR
  - HASH_CSR0 到 HASH_CSR53

- **当前上下文恢复**

  要恢复处理中断的消息，应用程序必须遵循以下步骤：

  a) 将存储器中保存的值写入以下寄存器中：HASH_IMR、HASH_STR 和 HASH_CR

  b) 通过将 HASH_CR 寄存器中的 INIT 位置 1 来初始化散列处理器

  c) 将存储器中保存的值写入 HASH_CSR0 到 HASH_CSR53 寄存器中

  d) 从之前的中断点重新开始处理

通过 DMA 加载数据

如果使用 DMA 将消息加载到散列处理器，则无法预测是否正在进行 DMA 传输。因此，用户应用程序必须停止 DMA 传输，然后等待至散列处理器准备就绪后再中断当前消息摘要计算。详细程序如下所述。

- **当前上下文保存**

  在使用 DMA 中断当前消息摘要计算之前，应用程序必须遵循以下步骤：

  a) 将 DMAE 位清零以禁止 DMA 接口。

  b) 等待直至当前 DMA 传输完成（等待 HASH_SR 寄存器中的 DMAS = 0）。请注意，块不一定已完全传输到散列。

  c) 在 DMA 控制器中禁止对应的通道。

  d) 等到散列处理器就绪（未在处理任何块），也就是等待 DINIS = 1。

- **当前上下文恢复**

  要恢复处理使用 DMA 中断的消息，应用程序必须遵循以下步骤：

  a) 重新配置 DMA 控制器，以便其在没有再次被中断的情况下继续传输消息直到结束。

  b) 通过将 DMAE 位置 1，从之前的中断点重新开始处理。

注： *如果上下文交换不涉及 HMAC 操作，则不需要保存和恢复 HASH_CSR38 到 HASH_CSR53 寄存器。*

如果在两个块之间进行上下文交换（最后一个块已完全处理并且下一个块尚未推入到 IN FIFO，HASH_CR 寄存器中 NBW = 000），则不需要保存和恢复 HASH_CSR22 到 HASH_CSR37 寄存器。

33.3.9 HASH DMA 接口

散列处理器可使用一个接口连接 DMA 控制器。通过将 HASH_CR 寄存器中的 DMAE 位置 1，可以使用此 DMA 向 HASH 中写入数据。当该位置 1 时，如果 FIFO 中有足够的空闲字以支持四个字的突发，HASH 会向 DMA 控制器发送突发请求信号。

一旦接收到四个 32 位字，HASH 便会将自动重新启动此过程，检查 FIFO 大小，并在 FIFO 状态允许突发接收时产生新的请求。有关详细信息，请参见*第 33.3.5 节：消息摘要计算*。

在开始 DMA 传输之前，软件必须对将要复制到 HASH_DIN 寄存器的最后一个字中的有效位数进行编程。这可通过向 HASH_STR 寄存器中写入以下值来完成：

```
NBLW = Len(Message)% 32
```

其中，“x%32”表示 x 除以 32 的余数。

HASH_SR 寄存器中的 DMAS 位提供有关 DMA 接口活动的信息。该位与 DMAE 一起设置，并且在 DMAE 清零且未在进行任何 DMA 传输时将被清零。

注： *没有任何中断与 DMAS 位相关联。*

33.3.10 HASH 错误管理

HASH 硬件不生成错误标志。

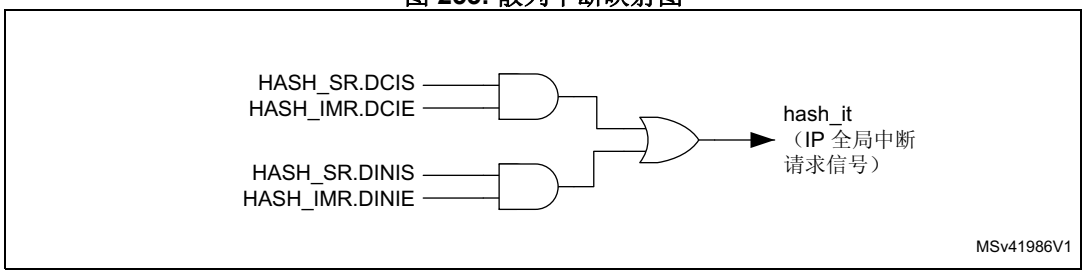
33.4 HASH 中断

散列处理器可生成两个单独的可屏蔽中断源，用以通知发生以下事件：

- 摘要计算完成 (DCIS)
- 数据输入缓冲器就绪 (DINIS)

这两个中断源连接到同一全局中断请求信号，如*图 253* 所示。

图 253. 散列中断映射图



通过更改 HASH_IMR 寄存器中的掩码位，可以单独使能或禁止以上中断源。将相应的屏蔽位置 1 以使能中断。

可以从 HASH_SR 寄存器中读取各个中断事件的状态。*表 209* 对可用功能进行了总结。

表 209. HASH 中断请求

中断事件	事件标志	使能控制位
摘要计算完成标志	DCIS	DCIE
数据输入缓冲器准备好获取新的块标志	DINIS	DINIE

33.5 HASH 处理时间

表 210 汇总了每种工作模式下处理 512 位中间块需要的时间。

表 210. 处理时间（时钟周期数）

工作模式	FIFO 负荷(1)	计算阶段	总计
MD5	16	50	**66**
SHA-1	16	66	**82**
SHA-224	16	50	**66**
SHA-256			

1. 必须将向处理器中加载 16 字的块所需的时间添加到该值中。

处理消息的最后一个块（或者 HMAC 中某个密钥的最后一个块）所需要的时间可能会更长。此时间取决于最后一个块的长度以及密钥的大小（在 HMAC 模式中）。

与处理中间块相比，此时间可能按以下系数增加：

- 对于散列消息，该系数为 **1 到 2.5**
- 对于 HMAC 输入密钥，该系数在 **2.5 左右**
- 对于 HMAC 消息，该系数为 **1 到 2.5**
- 对于 HMAC 输出密钥，如果是短密钥，则该系数在 **2.5 左右**
- 对于 HMAC 输出密钥，如果是长密钥，则该系数为 **3.5 到 5**

33.6 HASH 寄存器

HASH 内核与多个控制和状态寄存器以及五个消息摘要寄存器相关联。所有这些寄存器都只能通过 32 位字来访问，否则将生成 AHB2 错误。

33.6.1 HASH 控制寄存器 (HASH_CR)

HASH control 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
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ALGO[1]	Res.	LKEY
													rw		rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	MDMAT	DINNE	NBW[3:0]				ALGO[0]	MODE	DATATYPE		DMAE	INIT	Res.	Res.
		rw	r	r	r	r	r	rw	rw	rw	rw	rw	w		

位 31:19 保留，必须保持复位值。

位 18 **ALGO[1]**：请参见位 7 说明

位 17 保留，必须保持复位值。

位 16 **LKEY**：长密钥选择 (Long key selection)

在 HMAC 模式中，该位在短密钥（≤ 64 字节）或长密钥（> 64 字节）之间进行选择。

0：短密钥（≤ 64 字节）

1：长密钥（> 64 字节）

注： 仅当 INIT 位置 1 并且 MODE = 1 时此选择才有效。在计算期间更改该位不起作用。

位 15 保留，必须保持复位值。

位 14 保留，必须保持复位值。

位 13 **MDMAT**：多个 DMA 传输 (Multiple DMA Transfers)

如果在需要多个 DMA 传输时对大型文件进行散列运算，则将此位置 1。

0：在 DMA 传输结束时自动将 DCAL 置 1。

1：在 DMA 传输结束时不自动将 DCAL 置 1。

位 12 **DINNE**：DIN 非空 (DIN not empty)

当 HASH_DIN 寄存器包含有效数据时（也就是在至少写入一次之后），该位将置 1。当 INIT 位（初始化）或 DCAL 位（上一个消息处理已完成）写入到 1 时，会将此位清零。

0：数据输入缓冲器中不存在任何数据

1：输入缓冲器至少包含一个字的数据

位 11:8 **NBW[3:0]**：已推入的字数 (Number of words already pushed)

该位域反映了消息中已推入到 IN FIFO 的字数。如果在 DINNE = 1 时对 HASH_DIN 寄存器执行写访问，则 NBW 递增 (+1)。

当 INIT 位写入 1 或者当摘要计算启动时（DCAL 写入 1 或者 DMA 传输结束），NBW 将变为 0000。

如果未使用 DMA：

0000 且 DINNE=0：没有任何字推入到 DIN 缓冲器，即 HASH_DIN 寄存器和 IN FIFO 均为空。

0000 且 DINNE=1：有一个字已推入到 DIN 缓冲器，即 HASH_DIN 寄存器包含一个字而 IN FIFO 为空。

0001：有两个字已推入到 DIN 缓冲器，即 HASH_DIN 寄存器和 IN FIFO 分别包含一个字。

...

1111：16 个字已推入到 DIN 缓冲器

如果使用了 DMA：

则 NBW 刚好是通过 DMA 已推入到 IN FIFO 的字数。

位 18 和位 7 **ALGO[1:0]**：算法选择 (Algorithm selection)

这些位用于选择 SHA-1、SHA-224、SHA-256 或 MD5 算法：

00：选择 SHA-1 算法

01：选择 MD5 算法

10：选择 SHA-224 算法

11：选择 SHA-256 算法

注：　仅当 INIT 位置 1 时此选择才有效。在计算期间更改该位不起作用。

位 6 **MODE**：模式选择 (Mode selection)

该位针对所选算法选择散列或 HMAC 模式：

0：选择散列模式

1：选择 HMAC 模式。如果正在使用的密钥的长度超过 64 字节，则必须 LKEY 置 1。

注：　仅当 INIT 位置 1 时此选择才有效。在计算期间更改该位不起作用。

位 5:4 **DATATYPE**：数据类型选择 (Data type selection)

这些位定义在 HASH_DIN 寄存器中输入的数据格式：

00：32 位数据。写入到 HASH_DIN 的数据将由散列处理直接使用，不会重新排序。

01：16 位数据或半字。写入到 HASH_DIN 的数据被视为两个半字，并且在交换之后由散列处理使用。

10：8 位数据或字节。写入到 HASH_DIN 的数据被视为四个字节，并且在交换之后由散列处理使用。

11：位数据或位串。写入到 HASH_DIN 的数据被视为 32 位（位串的第一位处于位置 0），并且在交换之后由散列处理使用（位串的第一位处于位置 31）。

位 3 **DMAE**：DMA 使能 (DMA enable)

0：禁止 DMA 传输

1：使能 DMA 传输。散列内核可以接收数据时，便发送 DMA 请求。

该位置 1 后，将在消息的最后一个数据被写入散列处理器时由硬件清零。

当正在进行 DMA 传输时，将此位置 0 不会中止当前传输。HASH 的 DMA 接口在内部继续保持使能状态，直到传输完成或 INIT 被写入 1。

将 INIT 位置 1 不会清零 DMAE 位。

位 2 **INIT**：初始化消息摘要计算 (Initialize message digest calculation)
将该位写入到 1 将会复位散列处理器内核，以使散列可以计算新消息的消息摘要。
若在此位中写入 0 则不会产生影响。读取该位将始终返回 0。

位 1:0 保留，必须保持复位值。

33.6.2 HASH 数据输入寄存器 (HASH_DIN)

HASH data input register

偏移地址：0x04

复位值：0x0000 0000

HASH_DIN 是数据输入寄存器。其宽度为 32 位。该寄存器用于以 512 位的块来输入消息。编程 HASH_DIN 寄存器时，在 AHB 数据总线上呈现的值将被“推入”散列内核，并且寄存器获取 AHB 数据总线上呈现的新值。要获得正确的消息格式，必须先对 HASH_CR 寄存器中的 DATATYPE 位进行配置。

16 字的块写入到 HASH_DIN 寄存器后，将启动中间摘要计算。

- 如果未使用 DMA，将新数据写入到 HASH_DIN 寄存器来启动，即写入下一个块的第一个字（中间摘要计算）
- 如果使用了 DMA，则自动启动

最后一个块写入到 HASH_DIN 寄存器之后，将启动最终摘要计算（包括填充）。

- 通过在 HASH_STR 寄存器中将 DCAL 位写入 1 来启动（最终摘要计算）。
- 如果使用了 DMA 并且 MDMAT 位设置为 0，则自动启动。

当摘要计算（中间或最终）正在进行时，如果对 HASH_DIN 寄存器执行新的写访问，则将在 AHB2 总线上插入等待状态，直至散列计算完成。

读取 HASH_DIN 寄存器时，将访问该位置中写入的最后一个字（复位后为零）。

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

位 31:0 **DATAIN[31:0]**：数据输入 (Data input)
读取该寄存器会返回当前寄存器内容。
写入该寄存器会将当前寄存器内容推入到 IN FIFO，并且寄存器获取在 AHB 数据总线上呈现的新值。

33.6.3 HASH 启动寄存器 (HASH_STR)

HASH start register

偏移地址：0x08

复位值：0x0000 0000

HASH_STR 寄存器有两个功能：

- 用于定义散列处理器中输入的消息的最后一个字的有效位数（也就是写入到 HASH_DIN 寄存器的最后一个数据中有效的最低有效位的数量）
- 通过将 DCAL 位写入到 1，启动对消息中最后一个块的处理

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.	Res.	Res.	Res.	Res.	Res.	DCAL	Res.	Res.	Res.	NBLW[4:0]				
							w				rw	rw	rw	rw	rw

位 31:9 保留，必须保持复位值。

位 8 **DCAL**：摘要计算 (Digest calculation)

将该位写 1 将开始使用先前写入的 NBLW 值来填充消息，并且，由于 INIT 位已被写 1，因此将使用写入到 IN FIFO 的所有数据字开始计算最终消息摘要。

读取该位将返回 0。

位 7:5 保留，必须保持复位值。

位 4:0 **NBLW[4:0]**：最后一个字中的有效位数 (Number of valid bits in the last word)

当消息位串的最后一个字被写入 HASH_DIN 寄存器并且经过内部数据交换后，散列处理器会只接收如下指定的有效位：

0x00：写入的最后一个数据的全部 32 位均是有效消息位，即 M[31:0]

0x01：写入的最后一个数据中只有一位（在交换后）是有效消息位，即 M[0]

0x02：写入的最后一个数据中只有两位（在交换后）是有效消息位，即 M[1:0]

0x03：写入的最后一个数据中只有三位（在交换后）是有效消息位，即 M[2:0]

...

0x1F：写入的最后一个数据中只有31位（在交换后）是有效消息位，即 M[30:0]

上述机制仅在 DCAL = 0 时有效。如果在 DCAL 置 1 时写入 NBLW 位，则 NBLW 位域保持不变。换句话说，配置 NBLW 和设置 DCAL 不能同时进行。

读取 NBLW 位将返回写入到 NBLW 的最后一个值。

33.6.4 HASH 摘要寄存器 (HASH_HR0..7)

HASH digest registers

这些寄存器包含消息摘要结果，其命名方式如下：

- HASH_HR0、HASH_HR1、HASH_HR2、HASH_HR3 和 HASH_HR4 寄存器返回 SHA-1 摘要结果。

  在这种情况下，不使用 HASH_HR5 到 HASH_HR7 寄存器，并且其读取值为零。
- HASH_HR0、HASH_HR1、HASH_HR2 和 HASH_HR3 寄存器分别返回 MD5 中定义的 A、B、C、D 的值。

  在这种情况下，不使用 HASH_HR4 到 HASH_HR7 寄存器，并且其读取值为零。
- HASH_HR0 到 HASH_HR6 寄存器返回 SHA-224 摘要结果。

  在这种情况下，不使用 HASH_HR7 并且其读取值为零。
- HASH_HR0 到 HASH_HR7 寄存器返回 SHA-256 摘要结果。

在所有情况下，摘要的最高有效位均存储在 HASH_HR0[31] 中，并且不使用。

如果当散列内核在计算中间摘要或最终消息摘要时（也就是在 DCAL 位已经写入到 1 时），对这些寄存器之一进行读访问，则读取操作将被拖延直到散列计算完成。

注： *将 HASH_HR0、HASH_HR1、HASH_HR2、HASH_HR3 和 HASH_HR4 映射复制到两个存储区域中。*

HASH_HR0

偏移地址：0x0C 和 0x310

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
H0															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
H0															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

HASH_HR1

偏移地址：0x10 和 0x314

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
H1															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
H1															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

HASH_HR2

偏移地址：0x14 和 0x318

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
H2															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
H2															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

HASH_HR3

偏移地址：0x18 和 0x31C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
H3															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
H3															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

HASH_HR4

偏移地址：0x1C 和 0x320

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
H4															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
H4															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

HASH_HR5

偏移地址：0x324

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
H5															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
H5															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

HASH_HR6

偏移地址：0x328

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
H6															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
H6															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

HASH_HR7

偏移地址：0x32C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
H7															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
H7															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

注：　当启动新位流的摘要计算时（通过向 INIT 位写入 1），这些寄存器将被强制设为其复位值。

33.6.5 HASH 中断使能寄存器 (HASH_IMR)

HASH interrupt enable 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
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.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DCIE	DINIE
														rw	rw

位 31:2 保留，必须保持复位值。

位 1 **DCIE**：摘要计算完成中断使能 (Digest calculation completion interrupt enable)

0：禁止摘要计算完成中断

1：使能摘要计算完成中断

位 0 **DINIE**：数据输入中断使能 (Data input interrupt enable)

0：禁止数据输入中断

1：使能数据输入中断

33.6.6 HASH 状态寄存器 (HASH_SR)

HASH status register

偏移地址：0x24

复位值：0x0000 0001

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.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BUSY	DMAS	DCIS	DINIS
												r	r	rc_w0	rc_w0

位 31:4 保留，必须保持复位值。

位 3 **BUSY**：忙碌位 (Busy bit)

0：当前未在处理任何块
1：散列内核正在处理某个数据块

位 2 **DMAS**：DMA 状态 (DMA Status)

该位提供有关 DMA 接口活动的信息。该位与 DMAE 一起设置，并且在 DMAE=0 且未在进行任何 DMA 传输时将被清零。没有任何中断与该位相关联。

0：禁止 DMA 接口 (DMAE=0) 并且未在进行任何传输
1：使能 DMA 接口 (DMAE=1) 或者某个传输在进行

位 1 **DCIS**：摘要计算完成中断状态 (Digest calculation completion interrupt status)

该位是在摘要就绪时（也就是整个消息已处理完毕时）由硬件设置。要将该位清零可写入 0，或者向 HASH_CR 寄存器中的 INIT 位写入 1。

0：HASH_HRx 寄存器中无任何摘要
1：摘要计算已完成，摘要存储于 HASH_HRx 寄存器中。如果 HASH_IMR 寄存器中将 DCIE 位置 1，则产生中断。

位 0 **DINIS**：数据输入中断状态 (Data input interrupt status)

该位是在输入缓冲器准备好获取新块时（16 个位置空闲）由硬件置 1。要将该位清零写入 0，或者写 HASH_DIN 寄存器。

0：输入缓冲器中的空闲位置少于 16 个
1：可以将一个新块输入到输入缓冲器中。如果 HASH_IMR 寄存器中将 DINIE 位置 1，则产生中断。

33.6.7 HASH 上下文交换寄存器 (HASH_CSRx)

HASH context swap registers

这些寄存器包含散列处理器的完整内部寄存器状态。如果高优先级的任务需要使用散列处理器，而该处理器正在执行其他任务，则必须使用上述寄存器进行上下文交换。

发生此类事件时，必须读取 HASH_CSRx 寄存器，并且读取值必须保存到系统存储器空间中。然后有优先权的任务便可以使用散列处理器，在散列计算完成时，可以从存储器中读取保存的上下文并回写到 HASH_CSRx 寄存器中。

HASH_CSR0

偏移地址：0x0F8

复位值：0x0000 0002

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CS0															
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
CS0															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

HASH_CSRx（x = 1 至 53）

偏移地址：0x0F8 + x * 0x4

复位值：0x0000 0000

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

33.6.8 HASH 寄存器映射

表 211 汇总了 HASH 寄存器映射和复位值。

表 211. HASH 寄存器映射和复位值

偏移	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	HASH_CR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ALGO[1]	Res.	LKEY	Res.	Res.	MDMAT	DINNE	NBW				ALGO[0]	MODE	DATATYPE		DMAE	INIT	Res.	Res.
	Reset value														0		0			0	0	0	0	0	0	0	0	0	0	0	0		
0x04	HASH_DIN	DATAIN																															
	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	HASH_STR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DCAL	Res.	Res.	Res.	NBLW				
	Reset value																								0				0	0	0	0	0
0x0C	HASH_HR0	H0																															
	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	HASH_HR1	H1																															
	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
0x14	HASH_HR2	H2																															
	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
0x18	HASH_HR3	H3																															
	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	HASH_HR4	H4																															
	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
0x20	HASH_IMR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DCIE	DINIE
	Reset value																															0	0
0x24	HASH_SR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BUSY	DMAS	DCIS	DINIS
	Reset value																													0	0	0	1
0xF8	HASH_CSR0	CSR0																															
	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	1	0
0xFC	HASH_CSR1	CSR1																															
	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
...																																	
0x1CC	HASH_CSR53	CSR53																															
	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
Reserved																																	
0x310	HASH_HR0	H0																															
	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
0x314	HASH_HR1	H1																															
	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
0x318	HASH_HR2	H2																															
	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
0x31C	HASH_HR3	H3																															
	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
0x320	HASH_HR4	H4																															
	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
0x324	HASH_HR5	H5																															
	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
0x328	HASH_HR6	H6																															
	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
0x32C	HASH_HR7	H7																															
	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*第 78 页的第 2.2.2 节*。

34 高级控制定时器 (TIM1/TIM8)

34.1 TIM1/TIM8 简介

高级控制定时器 (TIM1/TIM8) 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 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。

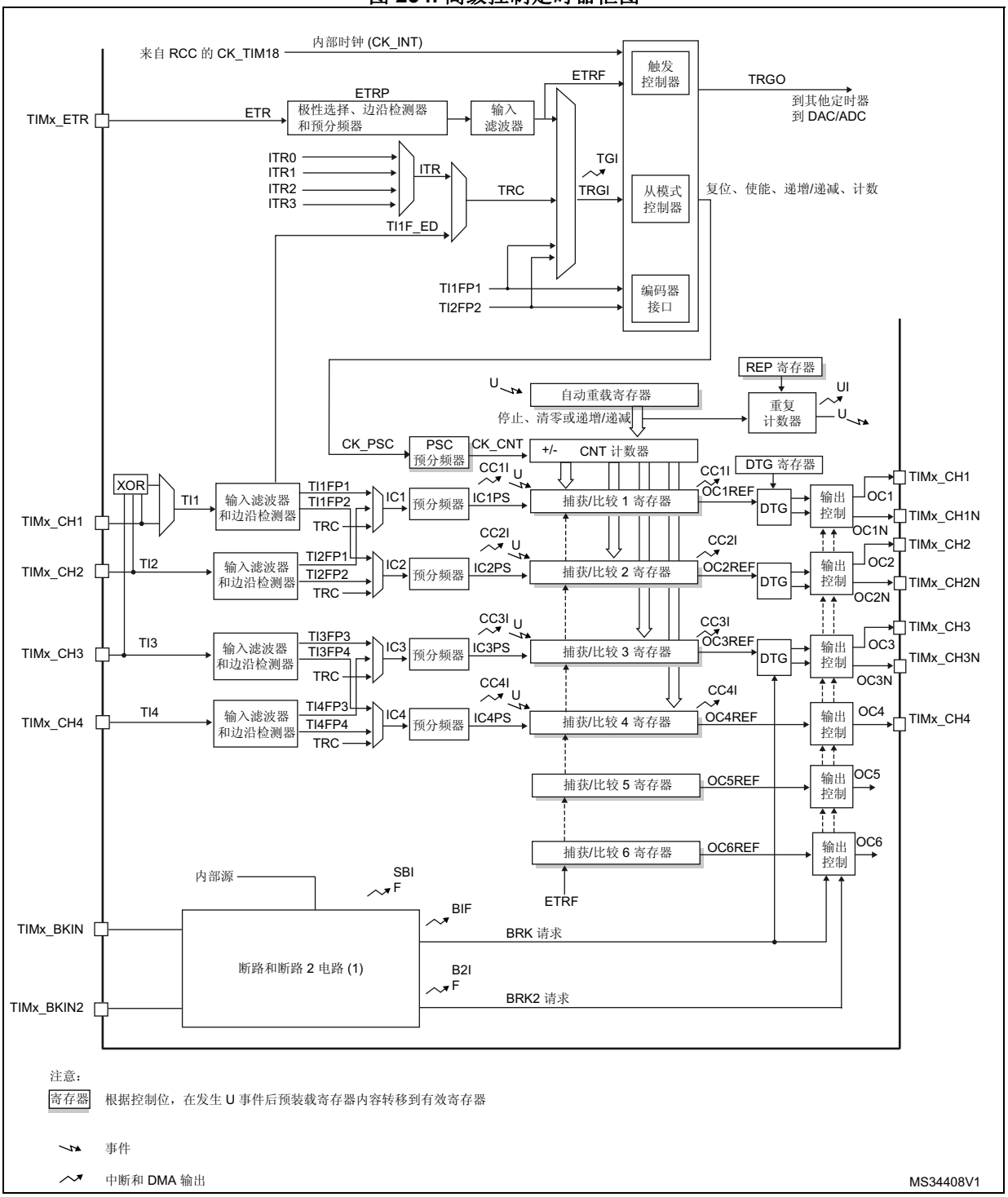
高级控制定时器 (TIM1/TIM8) 和通用定时器 (TIMy) 彼此完全独立，不共享任何资源。如*第 34.3.26 节：定时器同步*中所述，它们可以一起同步操作。

34.2 TIM1/TIM8 主要特性

TIM1/TIM8 定时器主要特性：

- 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（可在运行时修改），分频系数介于 1 到 65536 之间。
- 多达 6 个独立通道，可用于：
  - 输入捕获（但通道 5 和通道 6 除外）
  - 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。
- 2 个断路输入，用于将定时器的输出信号置于用户可选的安全配置中。
- 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。
- 触发输入用作外部时钟或逐周期电流管理。

图 254. 高级控制定时器框图



1. 详细信息，请参见*图 298：断路和断路 2 电路概述*。

34.3 TIM1/TIM8 功能描述

34.3.1 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 (TIMx_CNT)
- 预分频器寄存器 (TIMx_PSC)
- 自动重载寄存器 (TIMx_ARR)
- 重复计数器寄存器 (TIMx_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立即传送到影子寄存器，也可以在每次发生更新事件 (UEV) 时传送到影子寄存器，这取决于 TIMx_CR1 寄存器中的自动重载预装载使能位 (ARPE)。当计数器达到上溢值（或者在递减计数时达到下溢值）并且 TIMx_CR1 寄存器中的 UDIS 位为 0 时，将发送更新事件。该更新事件也可由软件产生。下文将针对各配置的更新事件的产生进行详细介绍。

计数器由预分频器输出 CK_CNT 提供时钟，仅当 TIMx_CR1 寄存器中的计数器启动位 (CEN) 置 1 时，才会启动计数器（有关计数器使能的更多详细信息，另请参见从模式控制器的相关说明）。

注意，计数器将在 TIMx_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 16 位寄存器 TIMx_PSC 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 255 和*图 256* 以一些示例说明在预分频比实时变化时计数器的行为：

图 255. 预分频器分频由 1 变为 2 时的计数器时序图

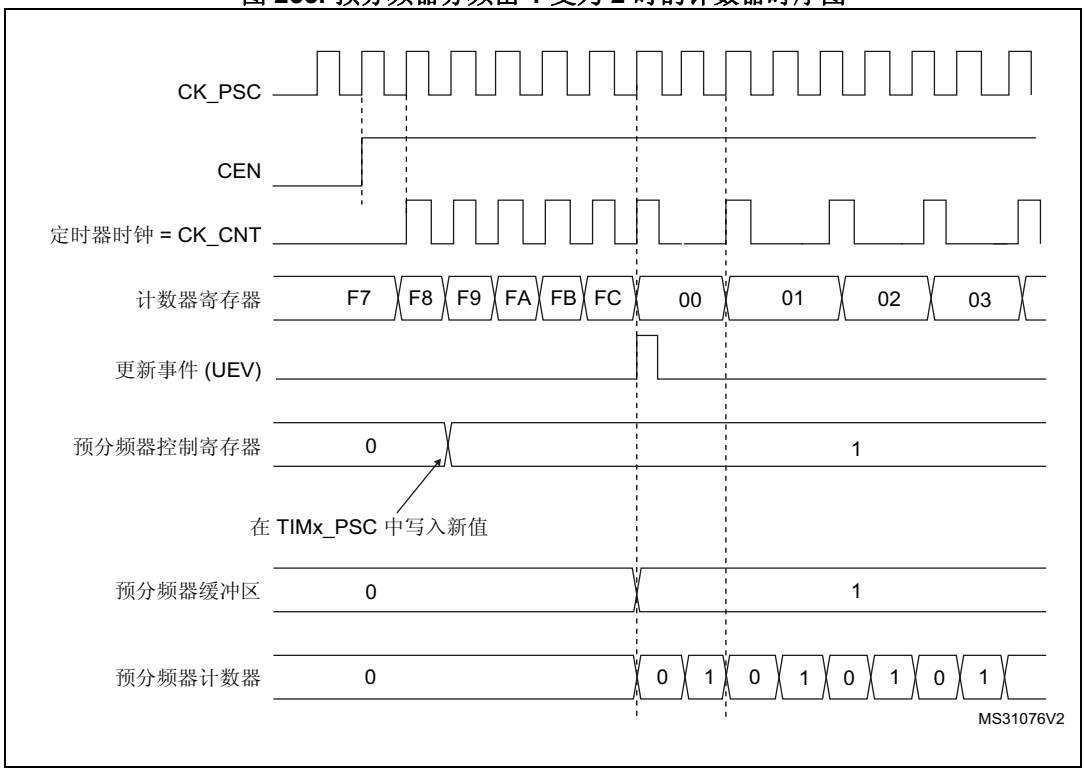
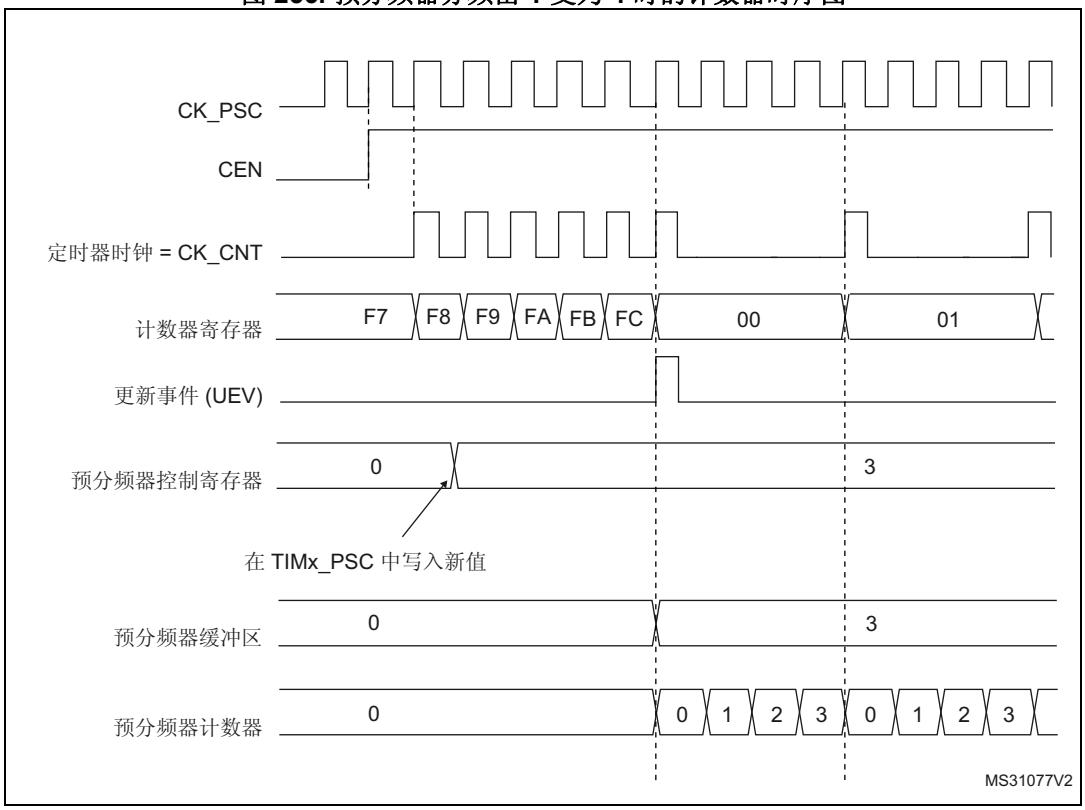


图 256. 预分频器分频由 1 变为 4 时的计数器时序图



34.3.2 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（TIMx_ARR 寄存器的内容），然后重新从 0 开始计数并生成计数器上溢事件。

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设定的次数加一次（(TIMx_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 TIMx_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 TIMx_CR1 寄存器中的 UDIS 位置 1 可禁止 UEV 事件。这可避免向预装载寄存器写入新值时更新影子寄存器。在 UDIS 位写入 0 之前不会产生任何更新事件。不过，计数器和预分频器计数器都会重新从 0 开始计数（而预分频比保持不变）。此外，如果 TIMx_CR1 寄存器中的 URS 位（更新请求选择）已置 1，则将 UG 位置 1 会生成更新事件 UEV，但不会将 UIF 标志置 1（因此，不会发送任何中断或 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（TIMx_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 TIMx_RCR 寄存器的内容。
- 使用预装载值 (TIMx_ARR) 更新自动重载影子寄存器。
- 预分频器的缓冲区中将重新装载预装载值（TIMx_PSC 寄存器的内容）。

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

图 257. 计数器时序图，1 分频内部时钟

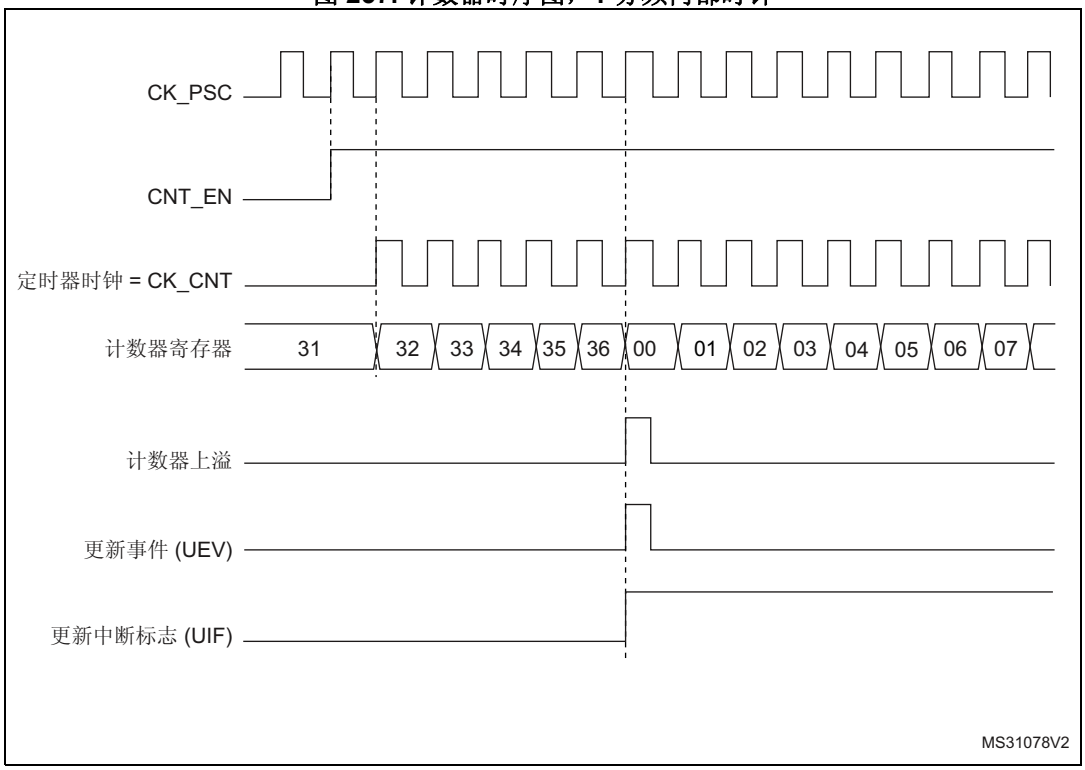


图 258. 计数器时序图，2 分频内部时钟

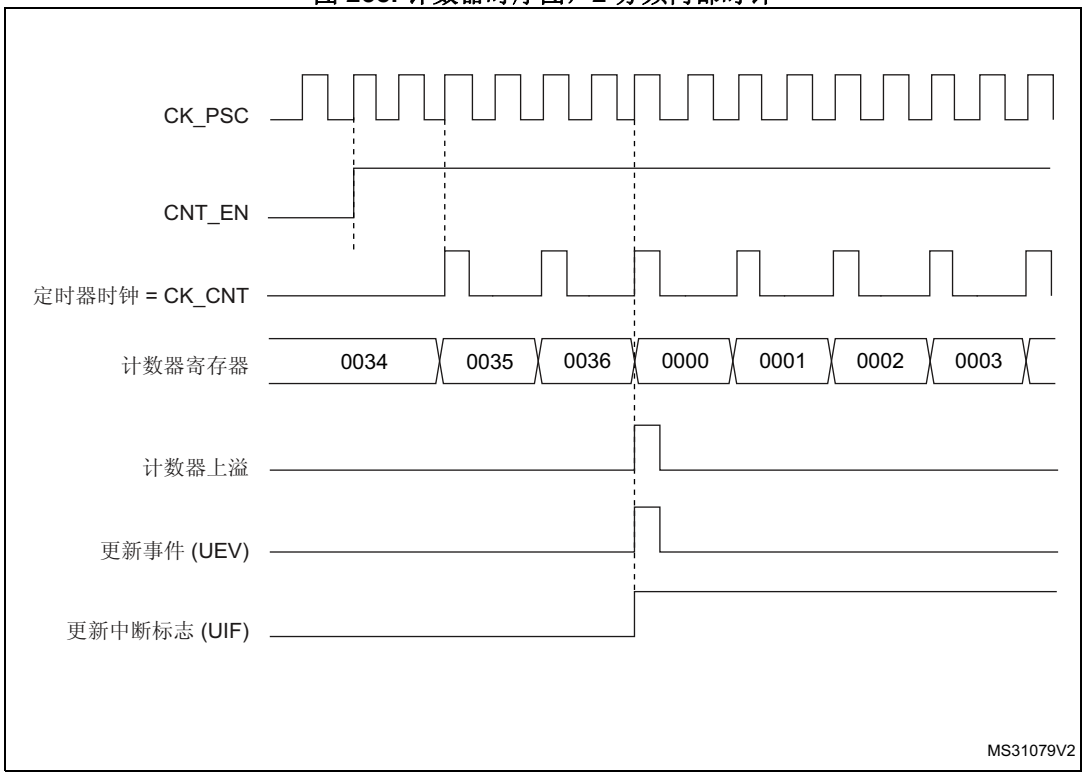


图 259. 计数器时序图，4 分频内部时钟

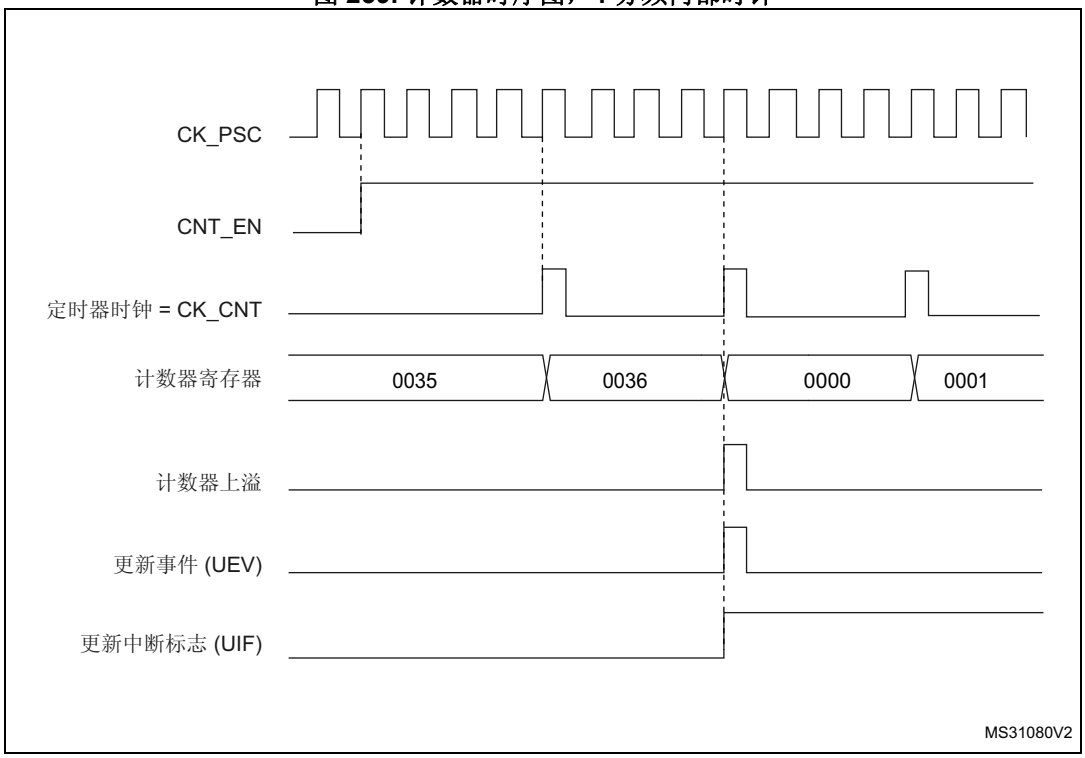


图 260. 计数器时序图，N 分频内部时钟

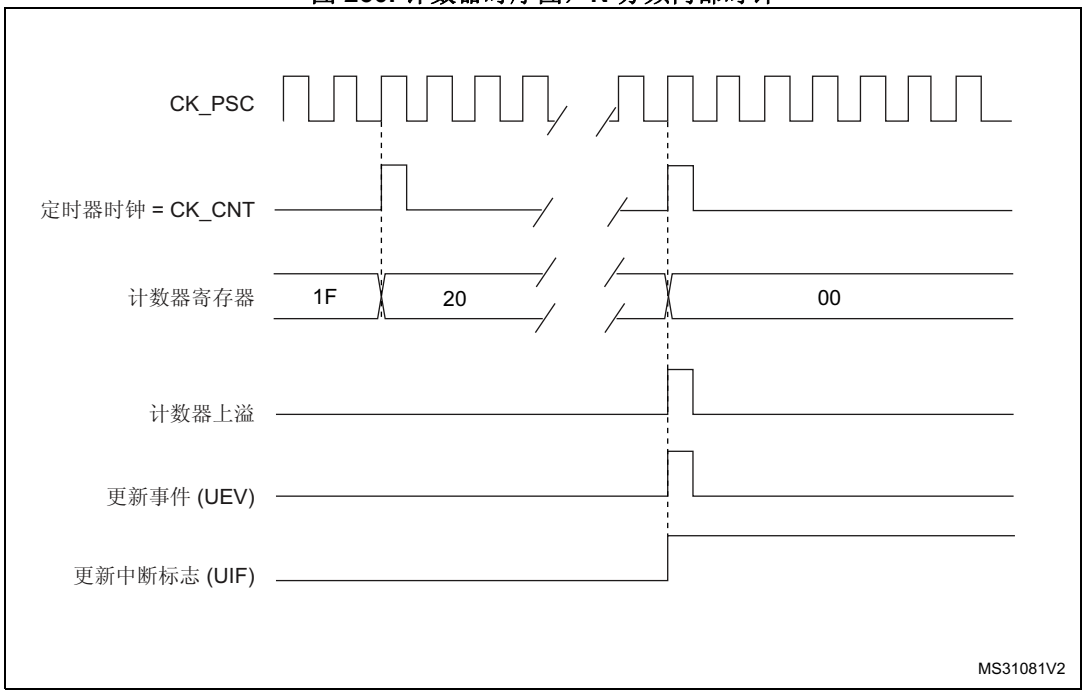


图 261. 计数器时序图，ARPE=0 时更新事件（TIMx_ARR 未预装载）

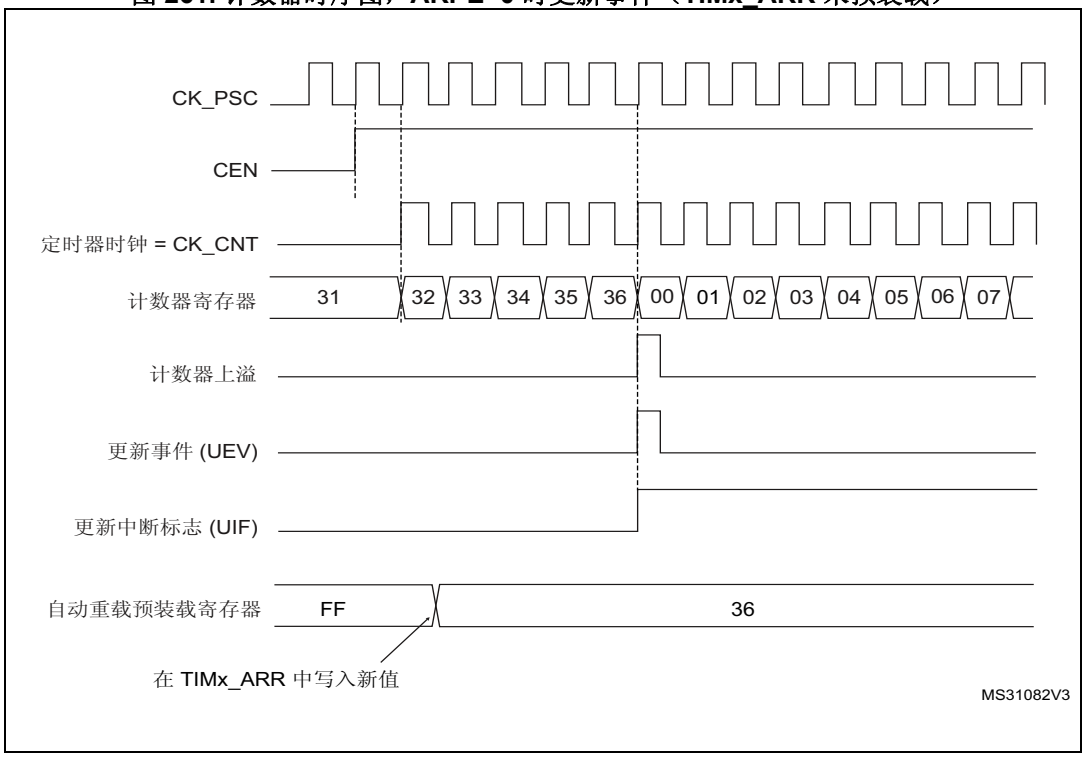
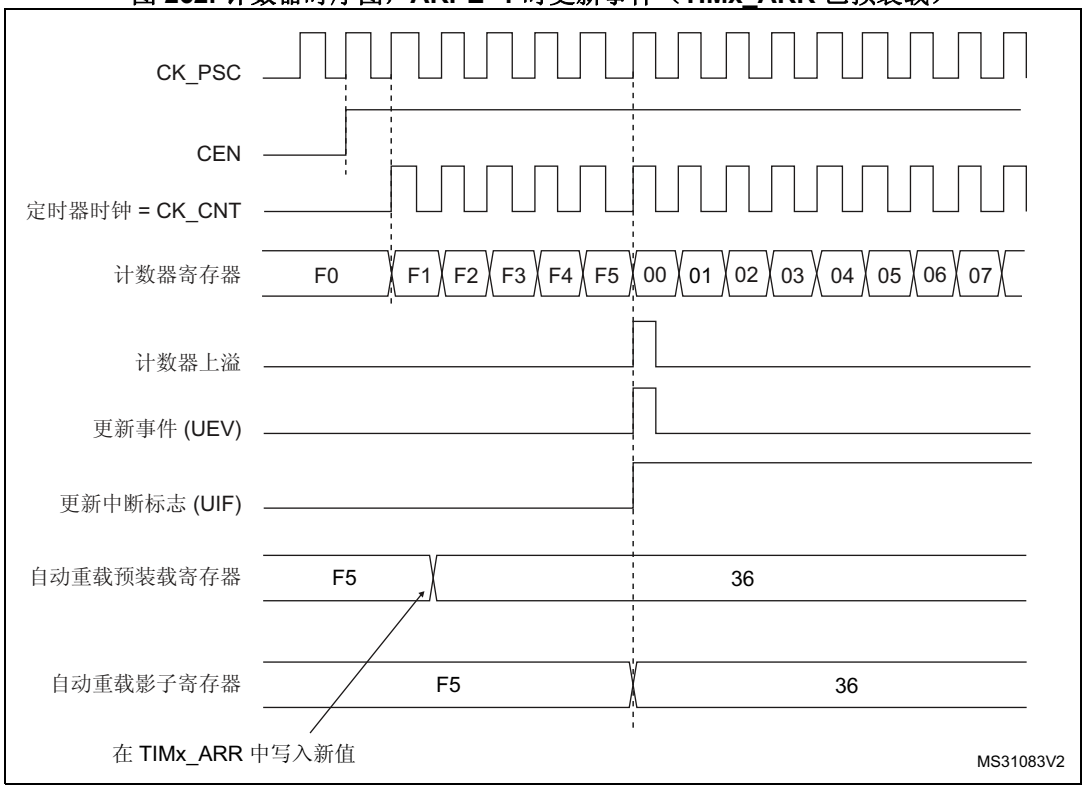


图 262. 计数器时序图，ARPE=1 时更新事件（TIMx_ARR 已预装载）



递减计数模式

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

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（(TIMx_RCR) + 1）后，将产生更新事件 (UEV)。否则，将在每次计数器下溢时产生更新事件。

将 TIMx_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 TIMx_CR1 寄存器中的 UDIS 位置 1 可禁止 UEV 更新事件。这可避免向预装载寄存器写入新值时更新影子寄存器。在 UDIS 位写入 0 之前不会产生任何更新事件。不过，计数器会重新从当前自动重载值开始计数，而预分频器计数器则重新从 0 开始计数（但预分频比保持不变）。

此外，如果 TIMx_CR1 寄存器中的 URS 位（更新请求选择）已置 1，则将 UG 位置 1 会生成更新事件 UEV，但不会将 UIF 标志置 1（因此，不会发送任何中断或 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（TIMx_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 TIMx_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（TIMx_PSC 寄存器的内容）。
- 自动重载有效寄存器将以预装载值（TIMx_ARR 寄存器的内容）进行更新。注意，ARR 寄存器更新在计数器重载之前被更新，因此下一个周期就是预期的值。

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

图 263. 计数器时序图，1 分频内部时钟

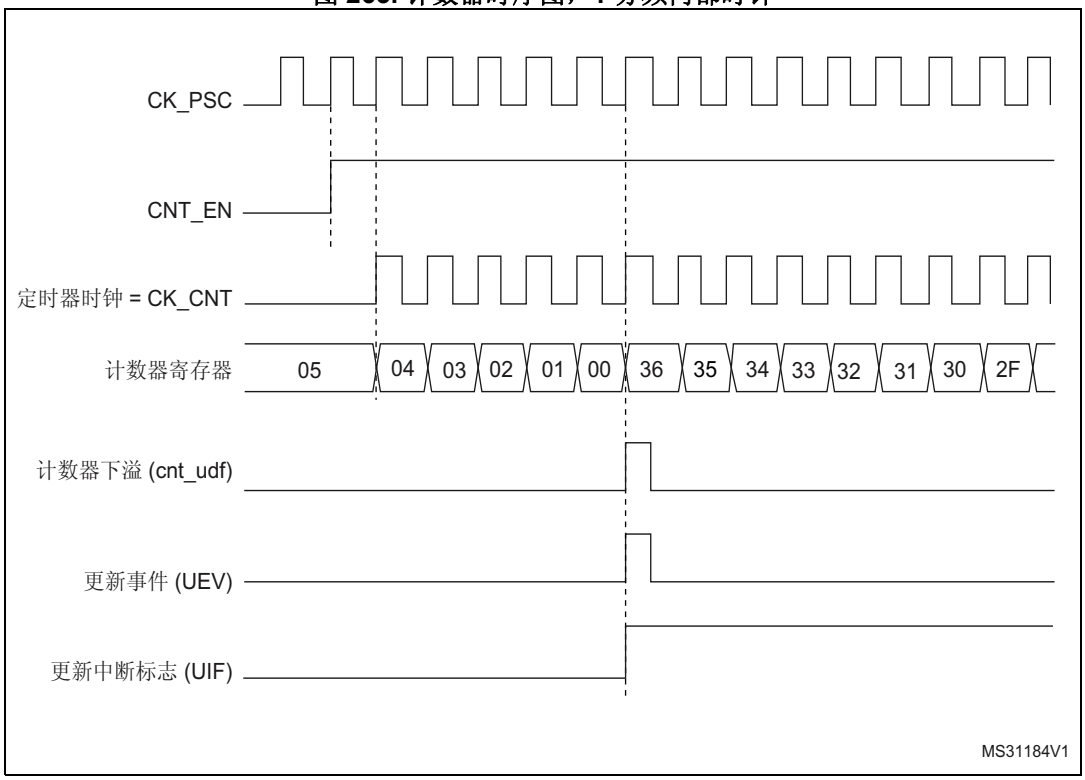


图 264. 计数器时序图，2 分频内部时钟

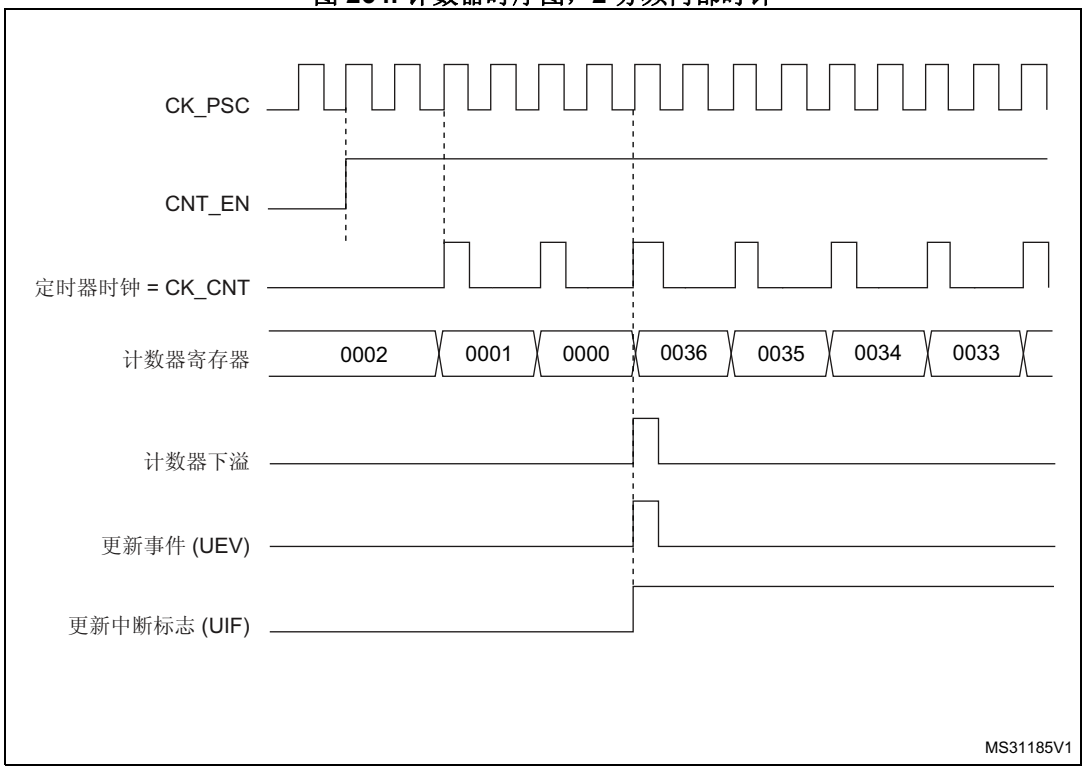


图 265. 计数器时序图，4 分频内部时钟

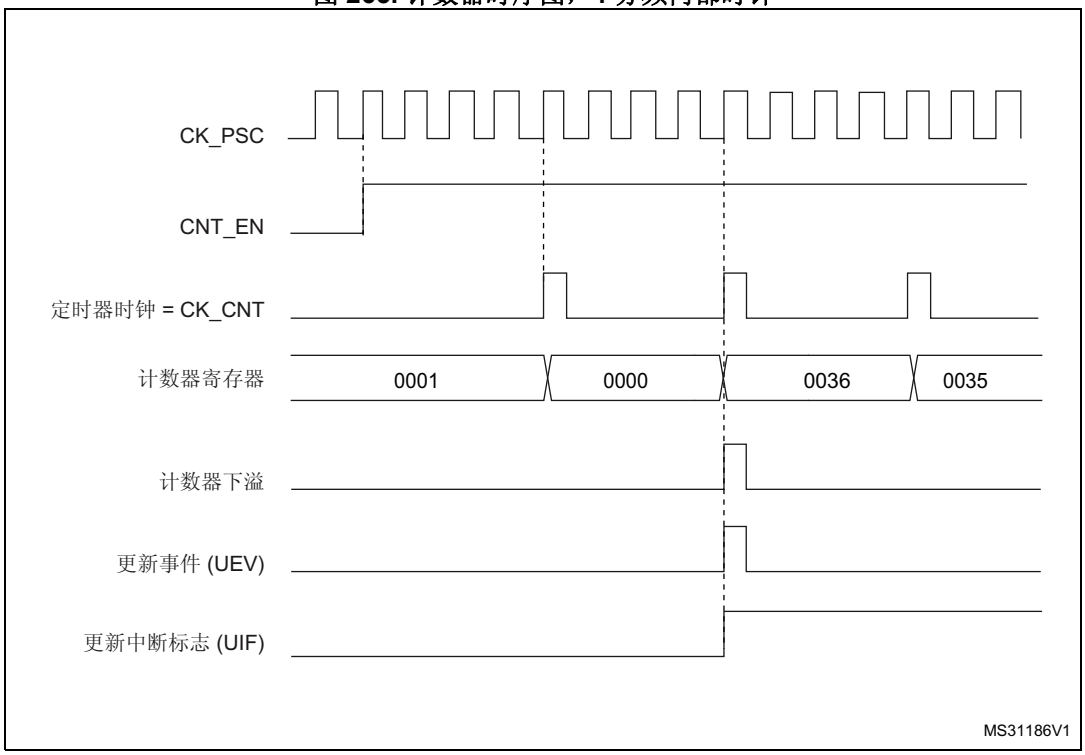


图 266. 计数器时序图，N 分频内部时钟

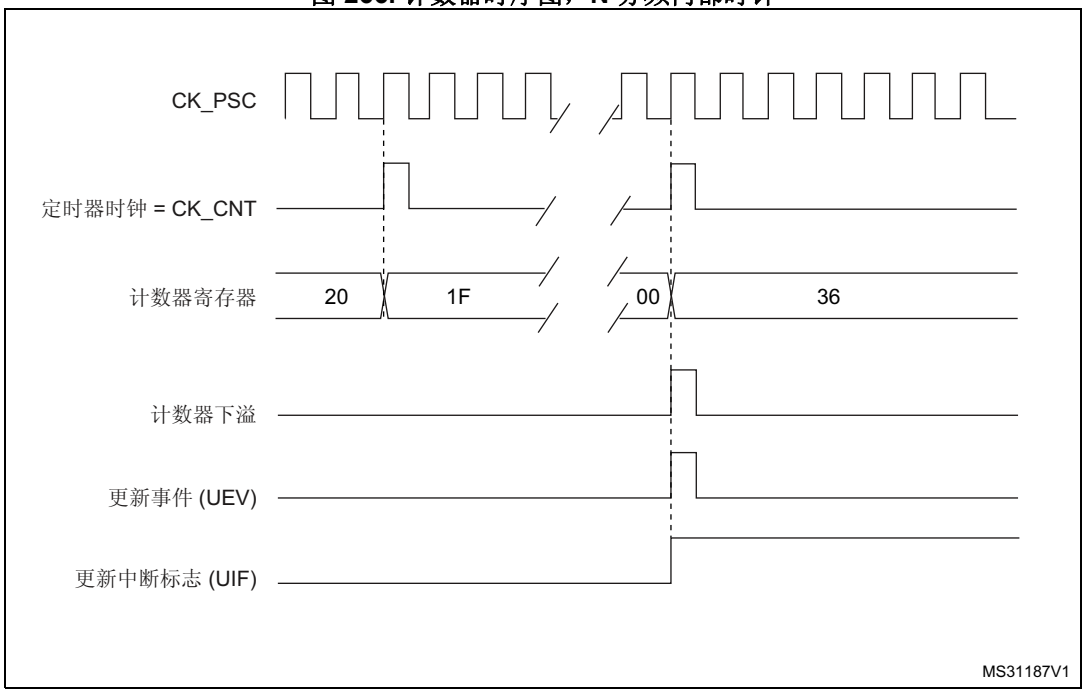
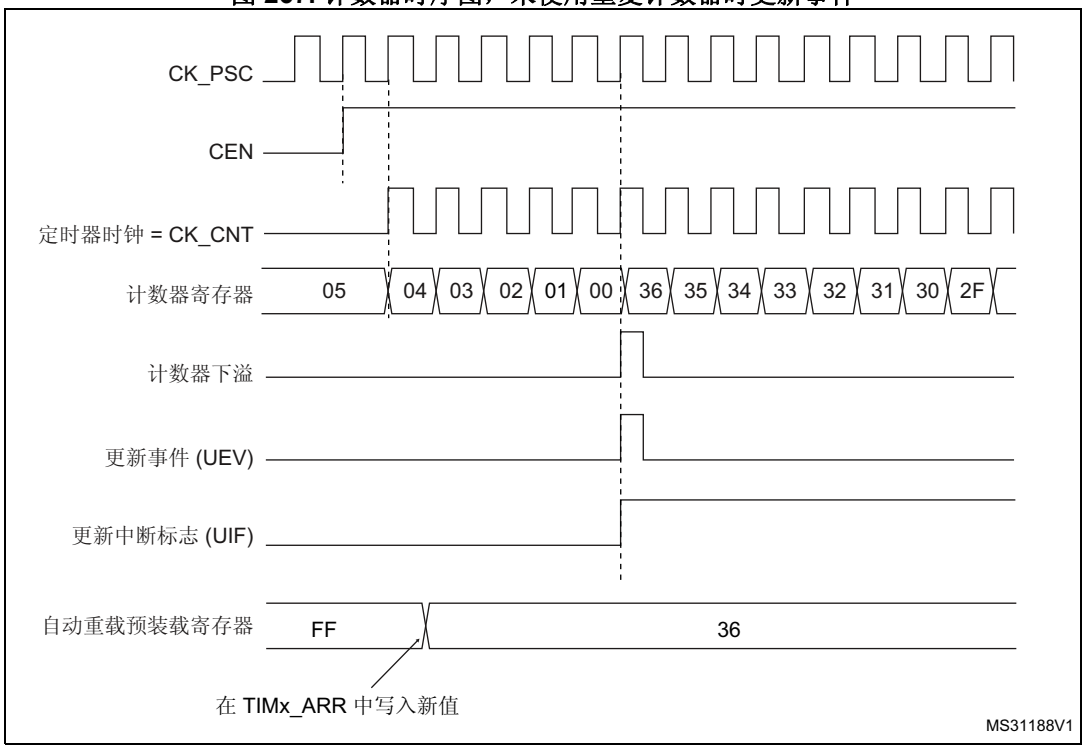


图 267. 计数器时序图，未使用重复计数器时更新事件



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（TIMx_ARR 寄存器的内容）– 1，生成计数器上溢事件；然后从自动重载值开始向下计数到 1 并生成计数器下溢事件。之后从 0 开始重新计数。

当 TIMx_CR1 寄存器中的 CMS 位不为“00”时，中心对齐模式有效。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，TIMx_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 TIMx_EGR 寄存器中的 UG 位置 1（通过软件或使用从模式控制器）也可以生成更新事件。这种情况下，计数器以及预分频器计数器将重新从 0 开始计数。

UEV 更新事件可通过软件禁止，只需将 TIMx_CR1 寄存器中的 UDIS 位置 1。这可避免向预装载寄存器写入新值时更新影子寄存器。在 UDIS 位写入 0 之前不会产生任何更新事件。不过，计数器仍会根据当前自动重载值进行递增和递减计数。

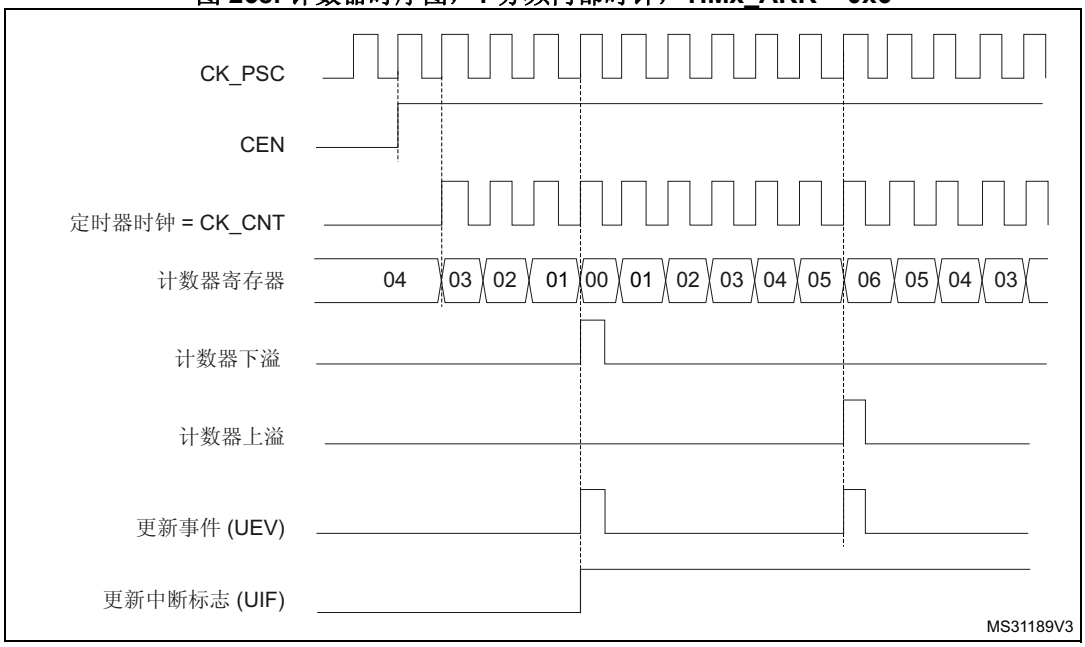
此外，如果 TIMx_CR1 寄存器中的 URS 位（更新请求选择）已置 1，则将 UG 位置 1 会产生 UEV 更新事件，但不会将 UIF 标志置 1（因此，不会发送任何中断或 DMA 请求）。这样一来，如果在发生捕获事件时将计数器清零，将不会同时产生更新中断和捕获中断。

发生更新事件时，将更新所有寄存器且将更新标志（TIMx_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 TIMx_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（TIMx_PSC 寄存器的内容）。
- 自动重载有效寄存器将以预装载值（TIMx_ARR 寄存器的内容）进行更新。注意，如果更新操作是由计数器上溢触发的，则 ARR 寄存器在计数器重载之前更新，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 268. 计数器时序图，1 分频内部时钟，TIMx_ARR = 0x6



1. 此处使用中心对齐模式 1（有关详细信息，请参见*第 34.4 节：TIM1/TIM8 寄存器*）。

图 269. 计数器时序图，2 分频内部时钟

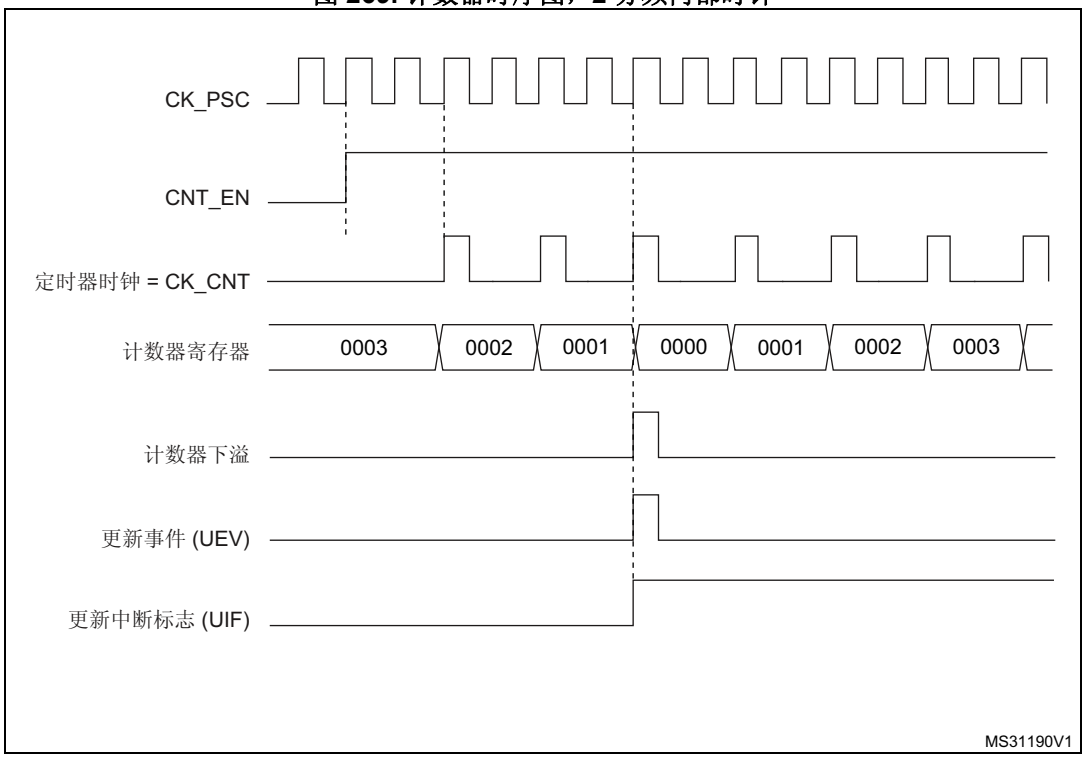


图 270. 计数器时序图，4 分频内部时钟，TIMx_ARR=0x36

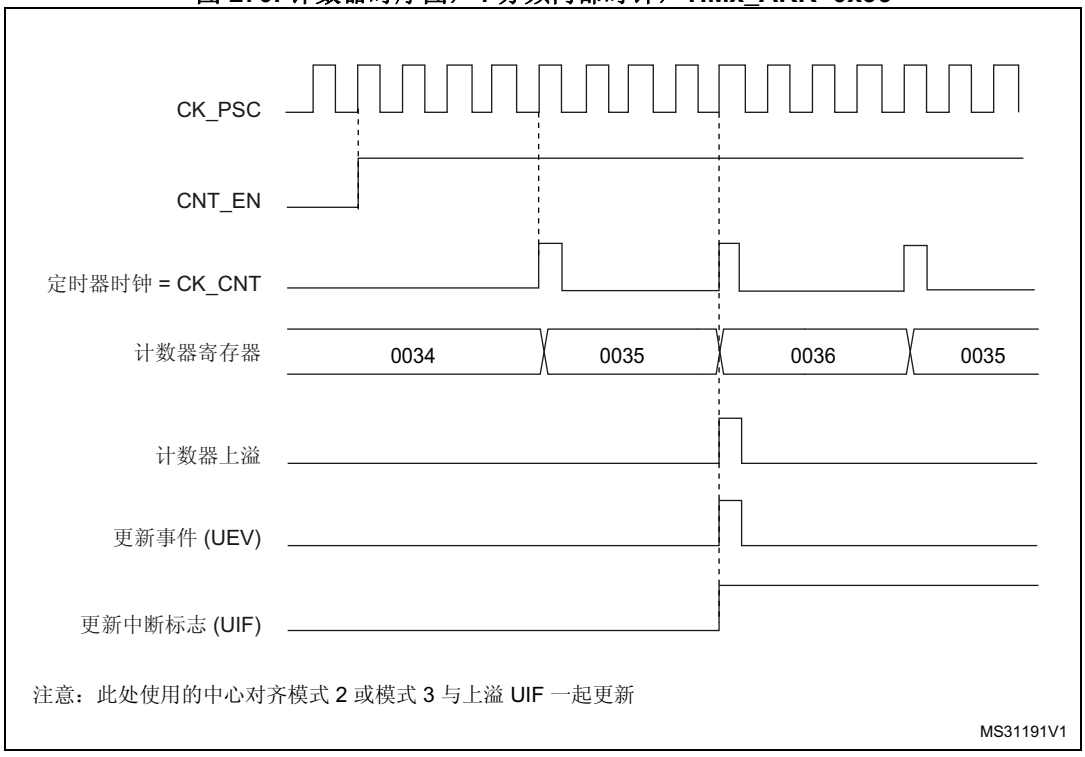


图 271. 计数器时序图，N 分频内部时钟

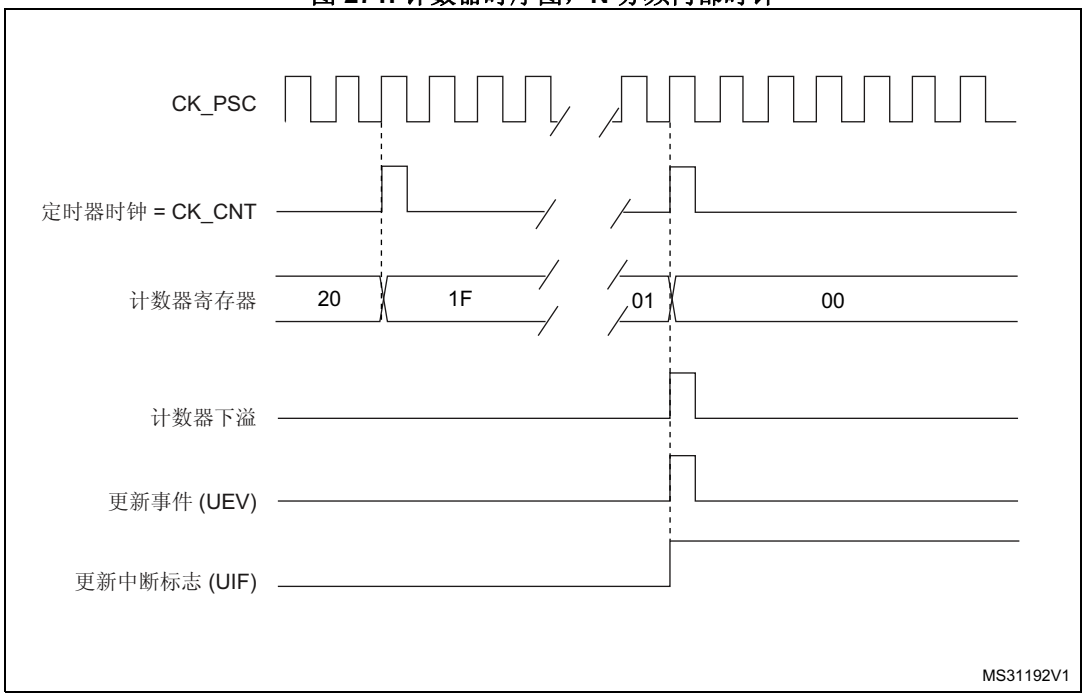


图 272. 计数器时序图，ARPE=1 时的更新事件（计数器下溢）

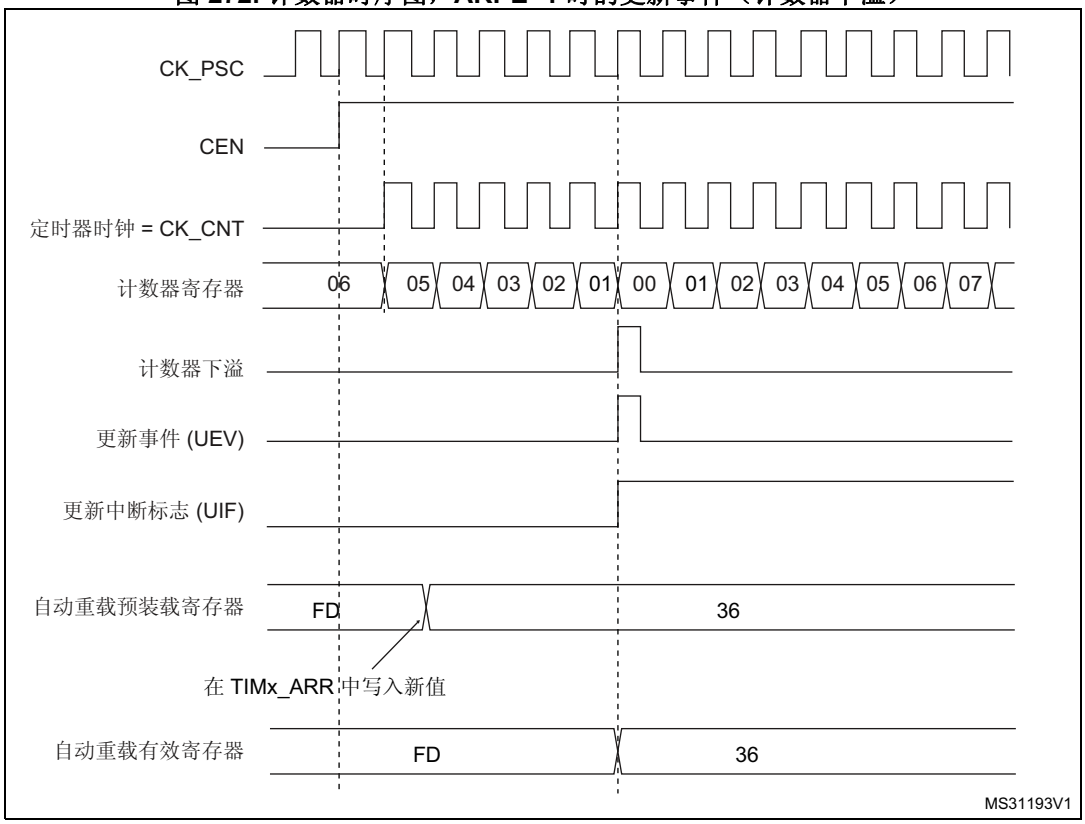
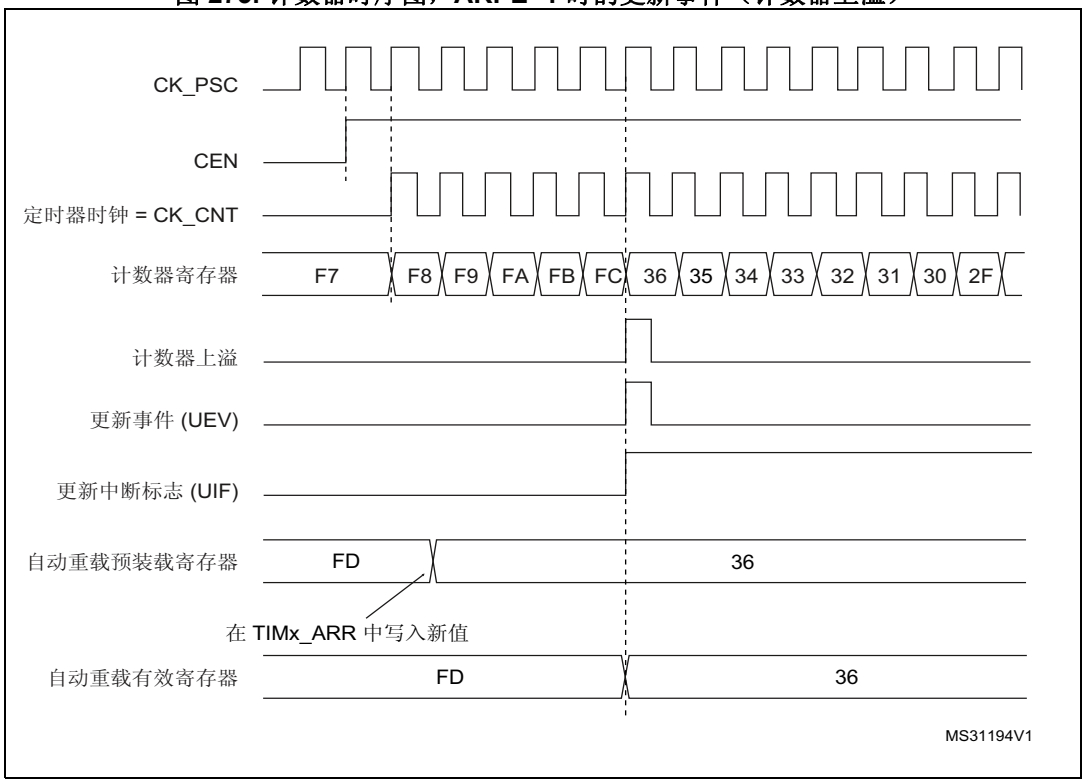


图 273. 计数器时序图，ARPE=1 时的更新事件（计数器上溢）



34.3.3 重复计数器

*第 34.3.1 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 TIMx_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（TIMx_ARR 自动重载寄存器、TIMx_PSC 预分频器寄存器以及比较模式下的 TIMx_CCRx 捕获/比较寄存器）。

重复计数器在下列情况下递减：

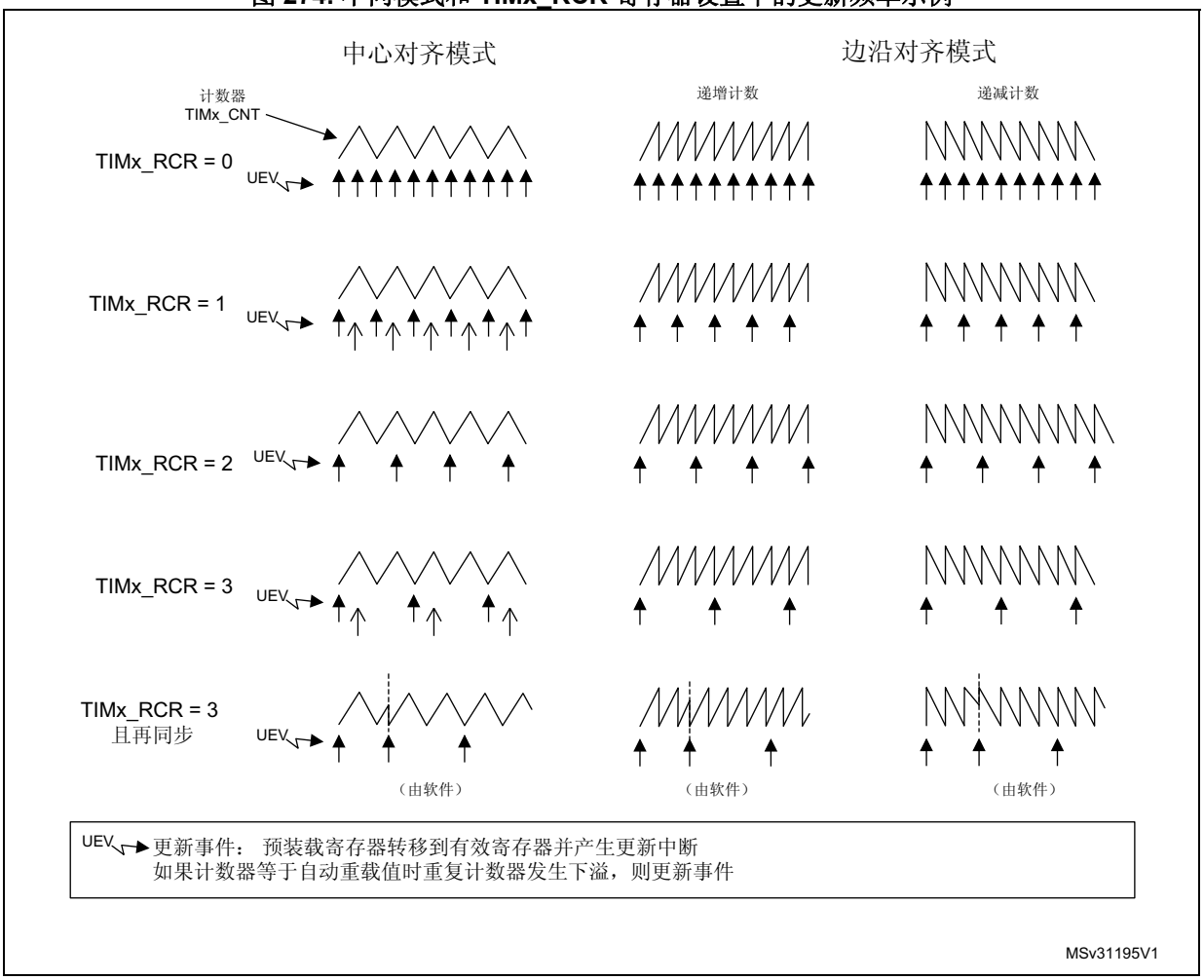
- 递增计数模式下的每个计数器上溢。
- 递减计数模式下的每个计数器下溢。
- 中心对齐模式下每个计数器上溢和计数器下溢。尽管这使得最大重复次数不超过 32768 个 PWM 周期，但在每个 PWM 周期内可更新占空比两次。当在中心对齐模式下，每个 PWM 周期仅刷新一次比较寄存器时，由于模式的对称性，最大分辨率为 $2xT_{ck}$。

重复计数器是自动重载类型；其重复率为 TIMx_RCR 寄存器所定义的值（请参见*图 274*）。当更新事件由软件（通过将 TIMx_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 TIMx_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产生（取决于何时写入 RCR）。

图 274. 不同模式和 TIMx_RCR 寄存器设置下的更新频率示例



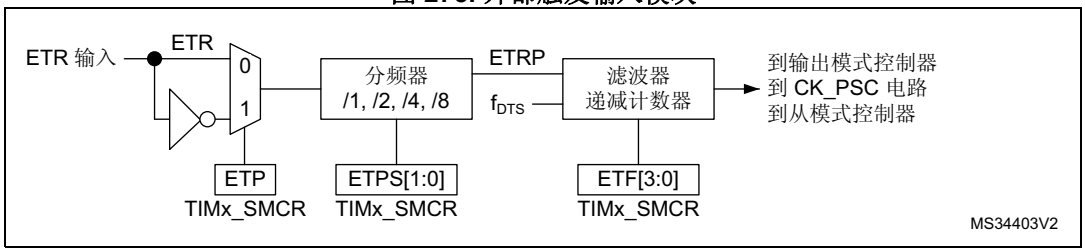
34.3.4 外部触发输入

定时器具有一个外部触发输入 ETR，它可用作：

- 外部时钟（外部时钟模式 2，请参见*第 34.3.5 节*）
- 用于从模式的触发信号（请参见*第 34.3.26 节*）
- 用于逐周期电流调节的 PWM 复位输入（请参见*第 34.3.7 节*）

下面的*图 275* 介绍了 ETR 输入的调节过程。输入极性通过 TIMx_SMCR 寄存器中的 ETP 位定义。触发信号可通过 ETPS[1:0] 位域编程的分频比进行预分频，然后通过 ETF[3:0] 位域进行数字滤波。

图 275. 外部触发输入模块



ETR 输入来自多个源：输入引脚（默认配置）、比较器输出和模拟看门狗。使用以下位域进行选择：

- TIMx_OR2 寄存器中的 ETRSEL[2:0] 位域
- TIMx_OR1[1:0] 寄存器中的 ETR_ADC1_RMP 位域

图 276. TIM1 ETR 输入电路

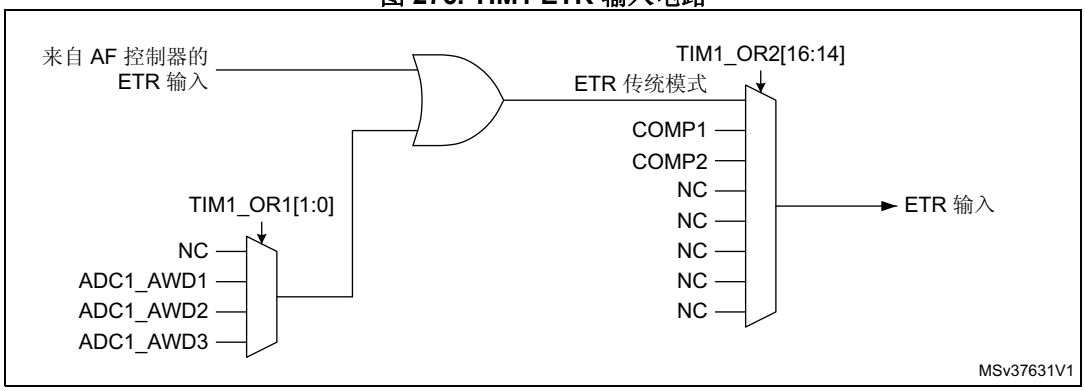
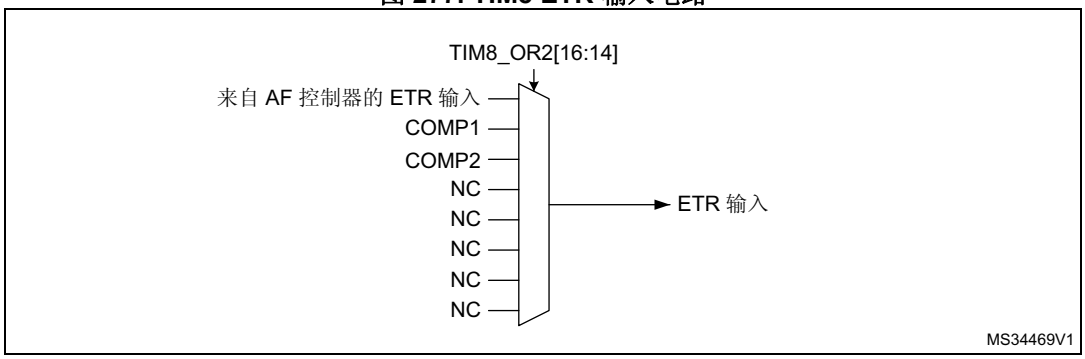


图 277. TIM8 ETR 输入电路



34.3.5 时钟选择

计数器时钟可由下列时钟源提供：

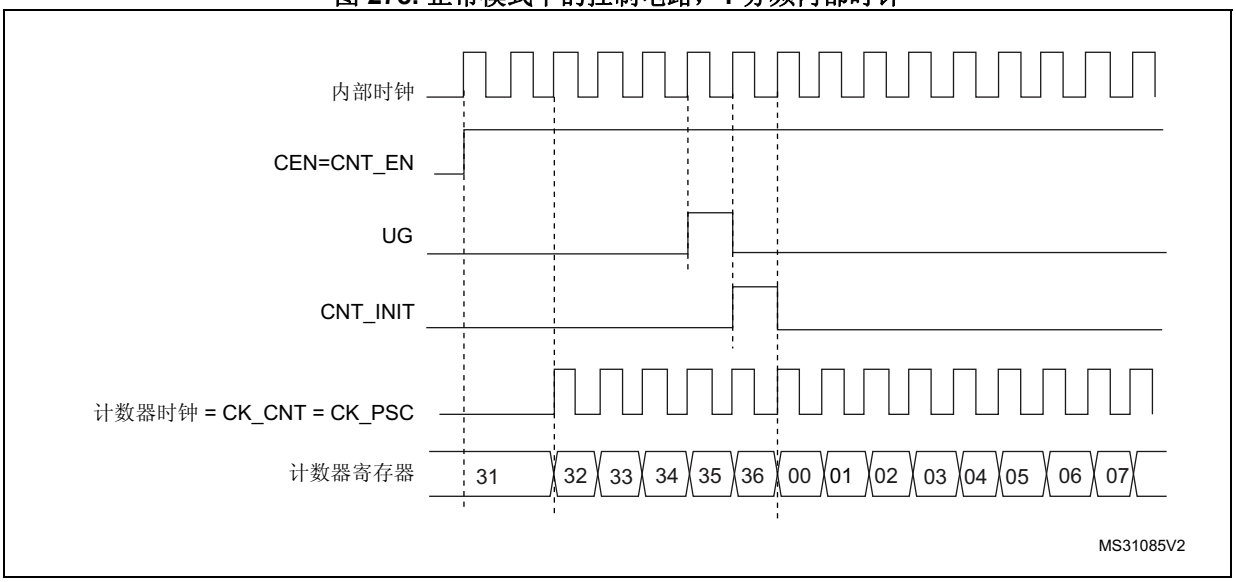
- 内部时钟 (CK_INT)
- 外部时钟模式 1：外部输入引脚
- 外部时钟模式 2：外部触发输入 ETR
- 编码器模式

内部时钟源 (CK_INT)

如果禁止从模式控制器 (SMS=000)，则 CEN 位、DIR 位（TIMx_CR1 寄存器中）和 UG 位（TIMx_EGR 寄存器中）为实际控制位，并且只能通过软件进行更改（UG 除外，仍保持自动清零）。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 278 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）。

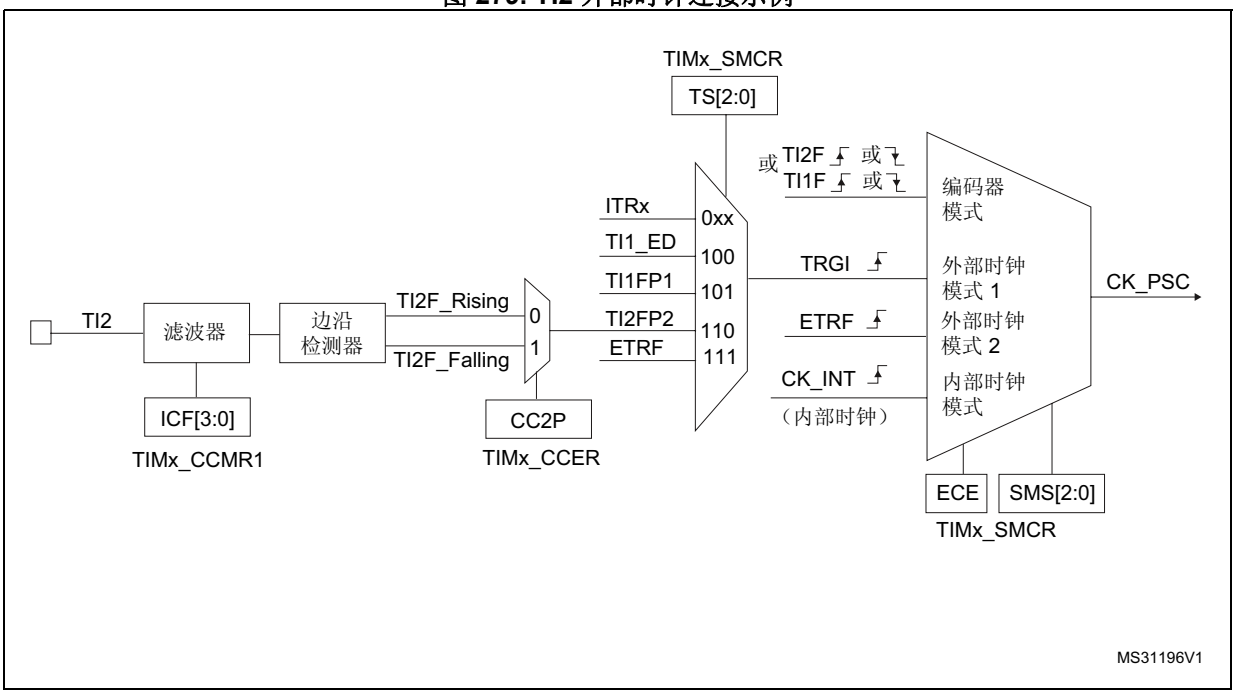
图 278. 正常模式下的控制电路，1 分频内部时钟



外部时钟源模式 1

当 TIMx_SMCR 寄存器中的 SMS=111 时，可选择此模式。计数器可在选定的输入信号上出现上升沿或下降沿时计数。

图 279. 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：

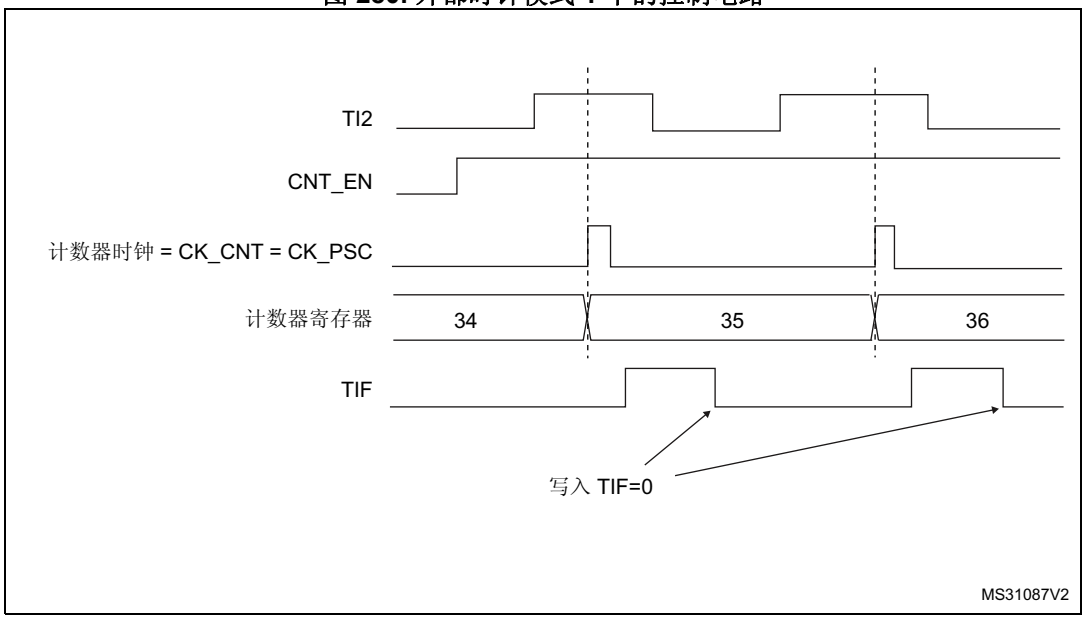
1. 通过在 TIMx_CCMR1 寄存器中写入 CC2S=“01”来配置通道 2，使其能够检测 TI2 输入的上升沿。
2. 通过在 TIMx_CCMR1 寄存器中写入 IC2F[3:0] 位来配置输入滤波带宽（如果不需要任何滤波器，请保持 IC2F=0000）。
3. 通过在 TIMx_CCER 寄存器中写入 CC2P=0 和 CC2NP=0 来选择上升沿极性。
4. 通过在 TIMx_SMCR 寄存器中写入 SMS=111，使定时器在外部时钟模式 1 下工作。
5. 通过在 TIMx_SMCR 寄存器中写入 TS=110 来选择 TI2 作为触发输入源。
6. 通过在 TIMx_CR1 寄存器中写入 CEN=1 来使能计数器。

注： *由于捕获预分频器不用于触发操作，因此用户无需对其进行配置。*

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 280. 外部时钟模式 1 下的控制电路



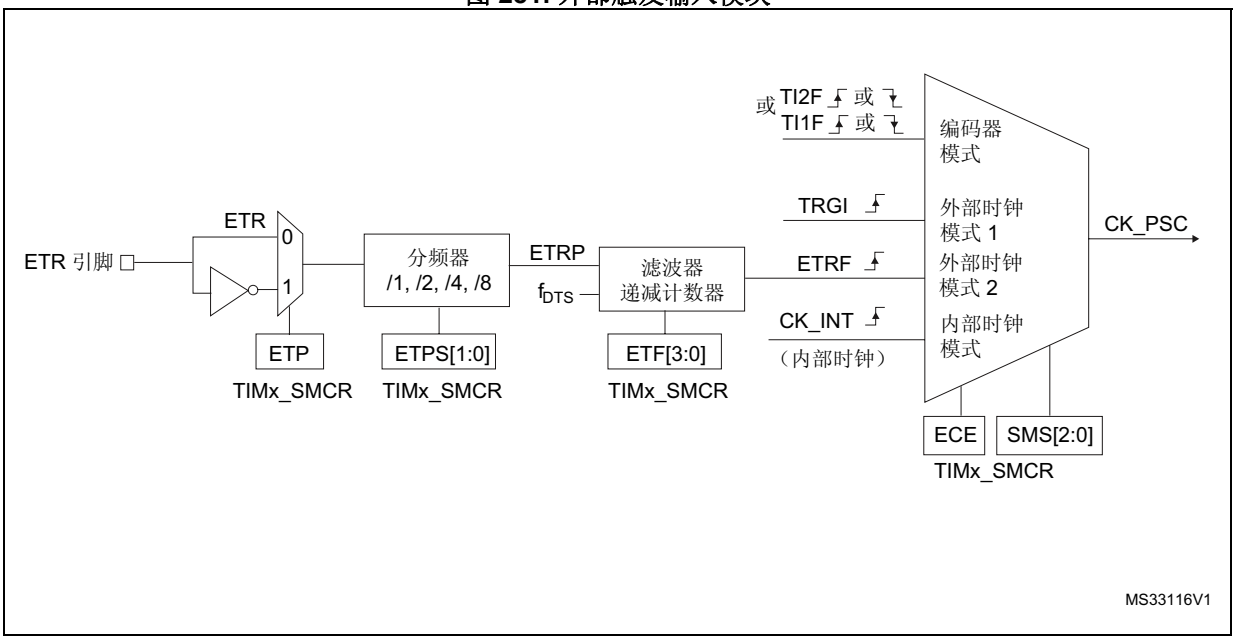
外部时钟源模式 2

通过在 TIMx_SMCR 寄存器中写入 ECE=1 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 281 简要介绍了外部触发输入模块。

图 281. 外部触发输入模块



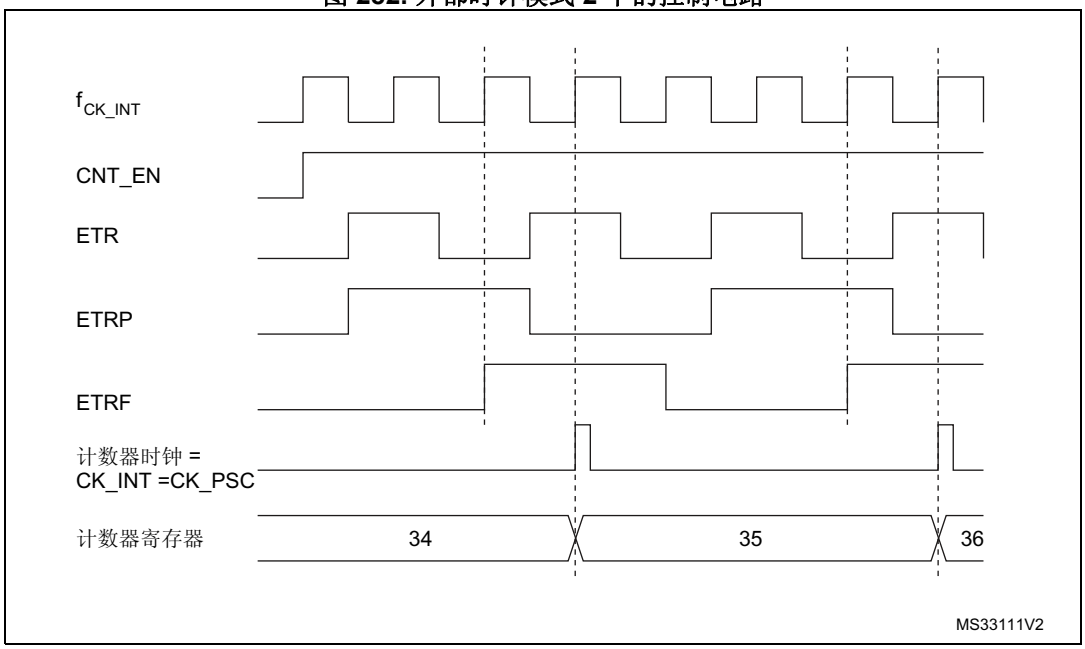
例如，要使递增计数器在 ETR 每出现 2 个上升沿时计数，请执行以下步骤：

1. 由于此例中不需滤波器，因此在 TIMx_SMCR 寄存器中写入 ETF[3:0]=0000。
2. 通过在 TIMx_SMCR 寄存器中写入 ETPS[1:0]=01 来设置预分频器。
3. 通过在 TIMx_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 TIMx_SMCR 寄存器中写入 ECE=1 来使能外部时钟模式 2。
5. 通过在 TIMx_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 282. 外部时钟模式 2 下的控制电路



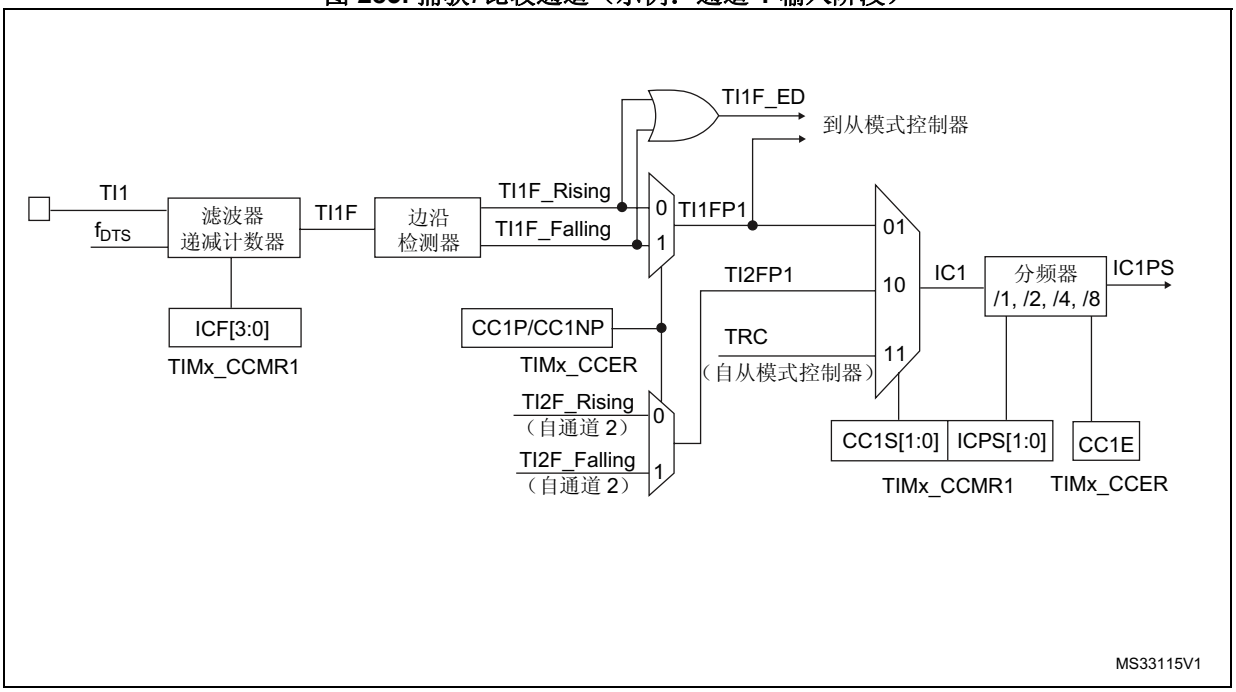
34.3.6 捕获/比较通道

每个捕获/比较通道均围绕一个捕获/比较寄存器（包括一个影子寄存器）、一个捕获输入阶段（数字滤波、多路复用和预分频器，通道 5 和通道 6 除外）和一个输出阶段（比较器和输出控制）构建而成。

图 283 到*图 286* 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 283. 捕获/比较通道（示例：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）。链的末端决定最终输出信号的极性。

图 284. 捕获/比较通道 1 主电路

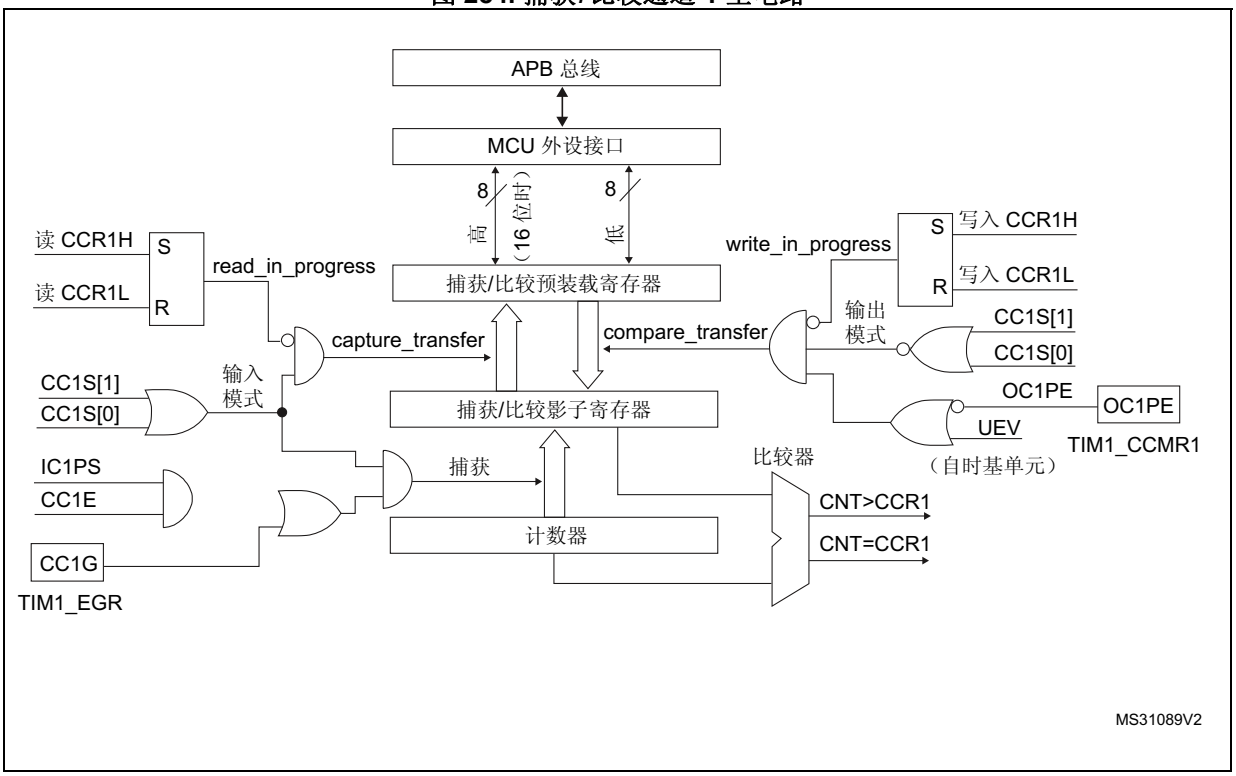
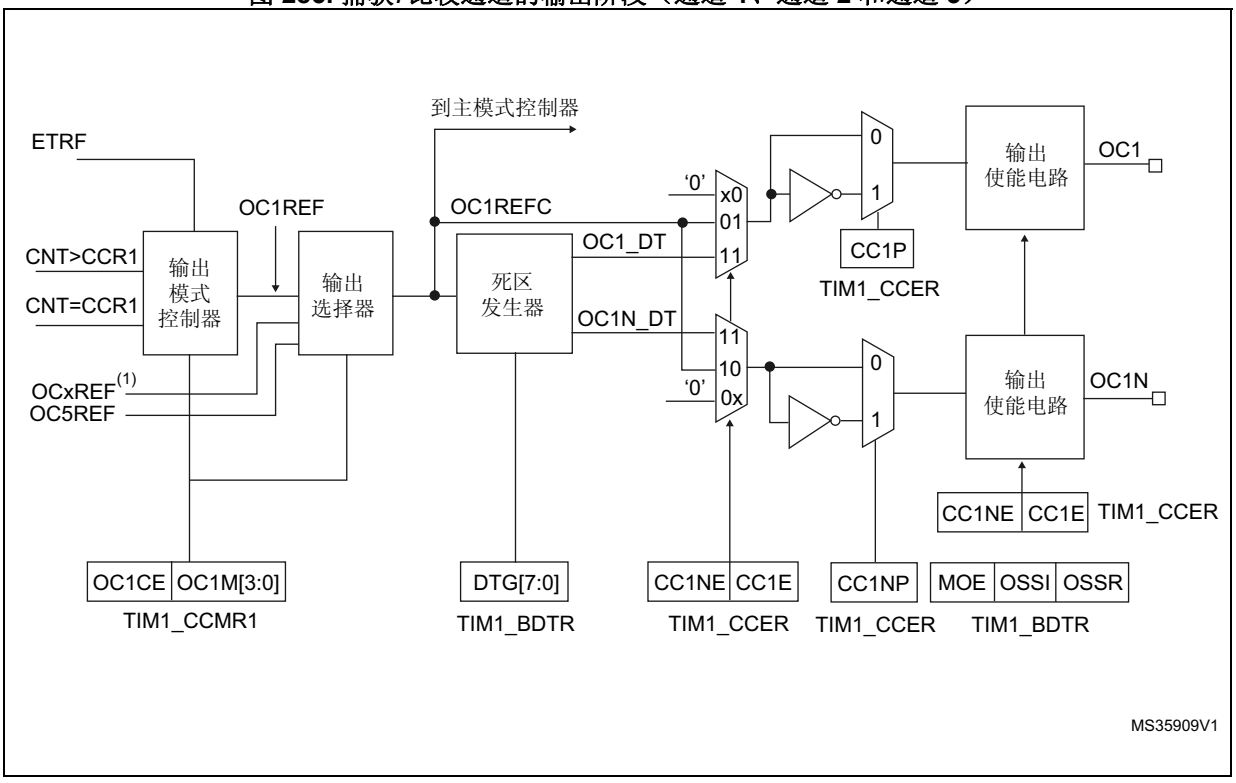


图 285. 捕获/比较通道的输出阶段（通道 1、通道 2 和通道 3）



1. OCxREF，其中 x 为互补通道的序号

图 286. 捕获/比较通道的输出阶段（通道 4）

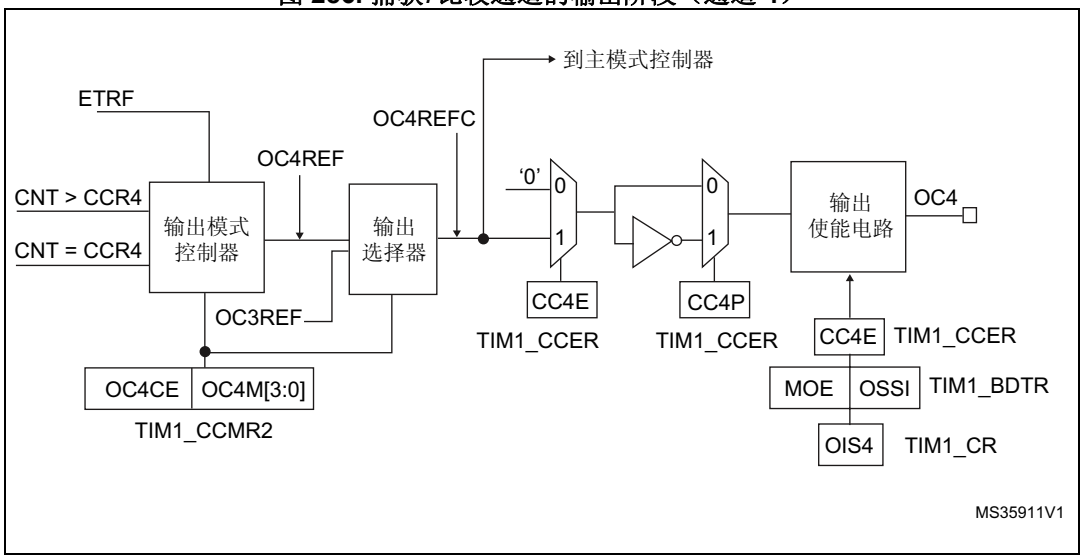
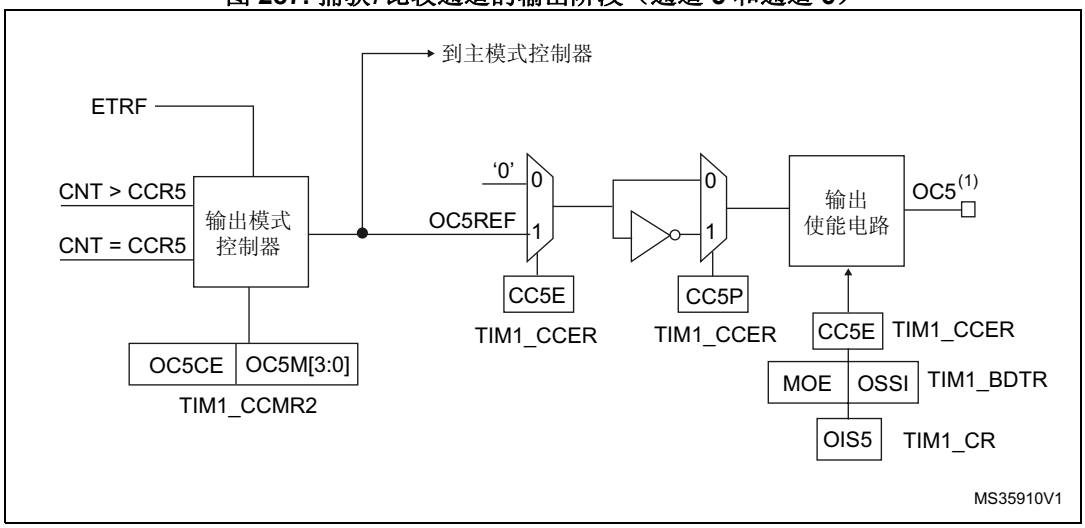


图 287. 捕获/比较通道的输出阶段（通道 5 和通道 6）



1. 不适用于外部。

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。

34.3.7 输入捕获模式

在输入捕获模式下，当相应的 ICx 信号检测到跳变沿后，将使用捕获/比较寄存器 (TIMx_CCRx) 来锁存计数器的值。发生捕获事件时，会将相应的 CCxIF 标志（TIMx_SR 寄存器）置 1，并可发送中断或 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（TIMx_SR 寄存器）置 1。可通过软件将 CCxIF 清零，方法是：向 CCxIF 写入“0”，或读取存储在 TIMx_CCRx 寄存器中的已捕获数据。向 CCxOF 写入“0”后会将其清零。

以下示例说明了如何在 TI1 输入出现上升沿时将计数器的值捕获到 TIMx_CCR1 中。具体操作步骤如下：

1. 选择有效输入：TIMx_CCR1 必须连接到 TI1 输入，因此向 TIMx_CCMR1 寄存器中的 CC1S 位写入 01。只要 CC1S 不等于 00，就会将通道配置为输入模式，并且 TIMx_CCR1 寄存器将处于只读状态。
2. 根据连接到定时器的信号，对所需的输入滤波带宽进行编程（如果输入为 TIx 之一，则对 TIMx_CCMRx 寄存器中的 ICxF 位进行编程）。假设信号边沿变化时，输入信号最多在 5 个内部时钟周期内发生抖动。因此，我们必须将滤波带宽设置为大于 5 个内部时钟周期。在检测到 8 个具有新电平的连续采样（以 f_{DTS} 频率采样）后，可以确认 TI1 上的跳变沿。然后向 TIMx_CCMR1 寄存器中的 IC1F 位写入 0011。
3. 通过在 TIMx_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）。
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 TIMx_CCMR1 寄存器中的 IC1PS 位写入“00”）。
5. 通过将 TIMx_CCER 寄存器中的 CC1E 位置 1，允许将计数器的值捕获到捕获寄存器中。
6. 如果需要，可通过将 TIMx_DIER 寄存器中的 CC1IE 位置 1 来使能相关中断请求，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，TIMx_CCR1 寄存器会获取计数器的值。
- 将 CC1IF 标志置 1（中断标志）。如果至少发生了两次连续捕获，但 CC1IF 标志未被清零，这样 CC1OF 捕获溢出标志会被置 1。
- 根据 CC1IE 位生成中断。
- 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： *通过软件将 TIMx_EGR 寄存器中的相应 CCxG 位置 1 可生成 IC 中断和/或 DMA 请求。*

34.3.8 PWM 输入模式

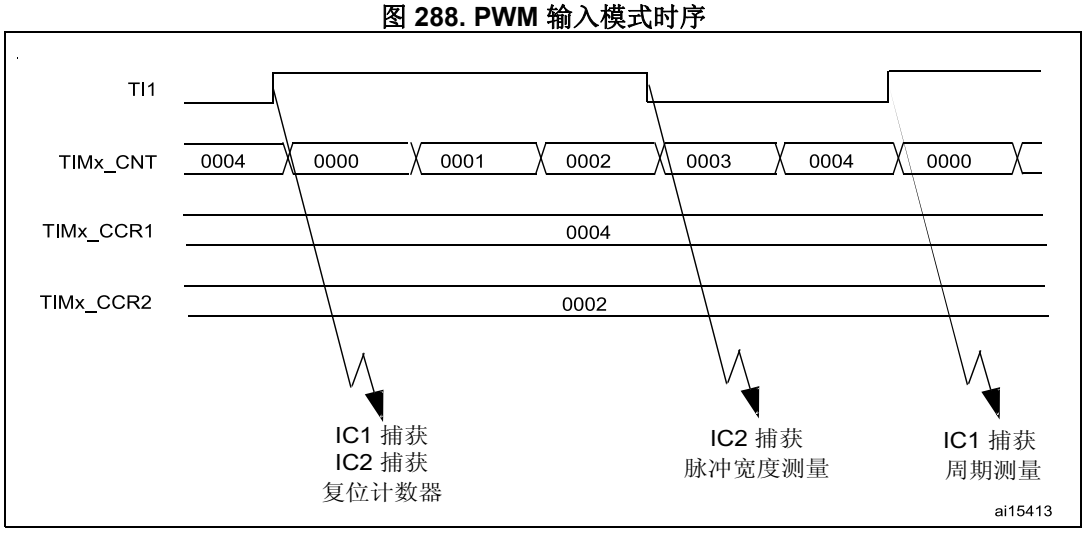
此模式是输入捕获模式的一个特例。其实现步骤与输入捕获模式基本相同，仅存在以下不同之处：

- 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，用户可通过以下步骤对应用于 TI1 的 PWM 的周期（位于 TIMx_CCR1 寄存器中）和占空比（位于 TIMx_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 TIMx_CCR1 的有效输入：向 TIMx_CCMR1 寄存器中的 CC1S 位写入 01（选择 TI1）。
2. 选择 TI1FP1 的有效极性（用于在 TIMx_CCR1 中捕获和计数器清零）：向 CC1P 位和 CC1NP 位写入“0”（上升沿有效）。
3. 选择 TIMx_CCR2 的有效输入：向 TIMx_CCMR1 寄存器中的 CC2S 写入 10 （选择 TI1）。
4. 选择 TI1FP2 的有效极性（用于在 TIMx_CCR2 中捕获）：向 CC2P 位和 CC2NP 位写入 CC2P/CC2NP=“10”（下降沿有效）。
5. 选择有效触发输入：向 TIMx_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 TIMx_SMCR 寄存器中的 SMS 位写入 0100。
7. 使能捕获：向 TIMx_CCER 寄存器中的 CC1E 位和 CC2E 位写入“1”。

图 288. PWM 输入模式时序



34.3.9 强制输出模式

在输出模式（TIMx_CCMRx 寄存器中的 CCxS 位 = 00）下，可直接由软件将每个输出比较信号（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，用户只需向相应 TIMx_CCMRx 寄存器中的 OCxM 位写入 0101。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 TIMx_CCMRx 寄存器中的 OCxM 位写入 0100，可将 OCxREF 信号强制设置为低电平。

无论如何，TIMx_CCRx 影子寄存器与计数器之间的比较仍会执行，而且允许将标志置 1。因此可发送相应的中断和 DMA 请求。下面的输出比较模式一节对此进行了介绍。

34.3.10 输出比较模式

此功能用于控制输出波形，或指示已经过某一时间段。通道 1 到通道 4 可用作输出，而通道 5 和通道 6 只能在单片机内部使用（例如，用于产生混合波形或触发 ADC）。

当捕获/比较寄存器与计数器之间相匹配时，输出比较功能：

- 将为相应的输出引脚分配一个可编程值，该值由输出比较模式（TIMx_CCMRx 寄存器中的 OCxM 位）和输出极性（TIMx_CCER 寄存器中的 CCxP 位）定义。匹配时，输出引脚既可保持其电平 (OCxM=0000)，也可设置为有效电平 (OCxM=0001)、无效电平 (OCxM=0010) 或进行翻转 (OCxM=0011)。
- 将中断状态寄存器中的标志置 1（TIMx_SR 寄存器中的 CCxIF 位）。
- 如果相应中断使能位（TIMx_DIER 寄存器中的 CCXIE 位）置 1，将生成中断。
- 如果相应使能位（TIMx_DIER 寄存器的 CCxDE 位，TIMx_CR2 寄存器的 CCDS 位，用来选择 DMA 请求）置 1，将发送 DMA 请求。

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

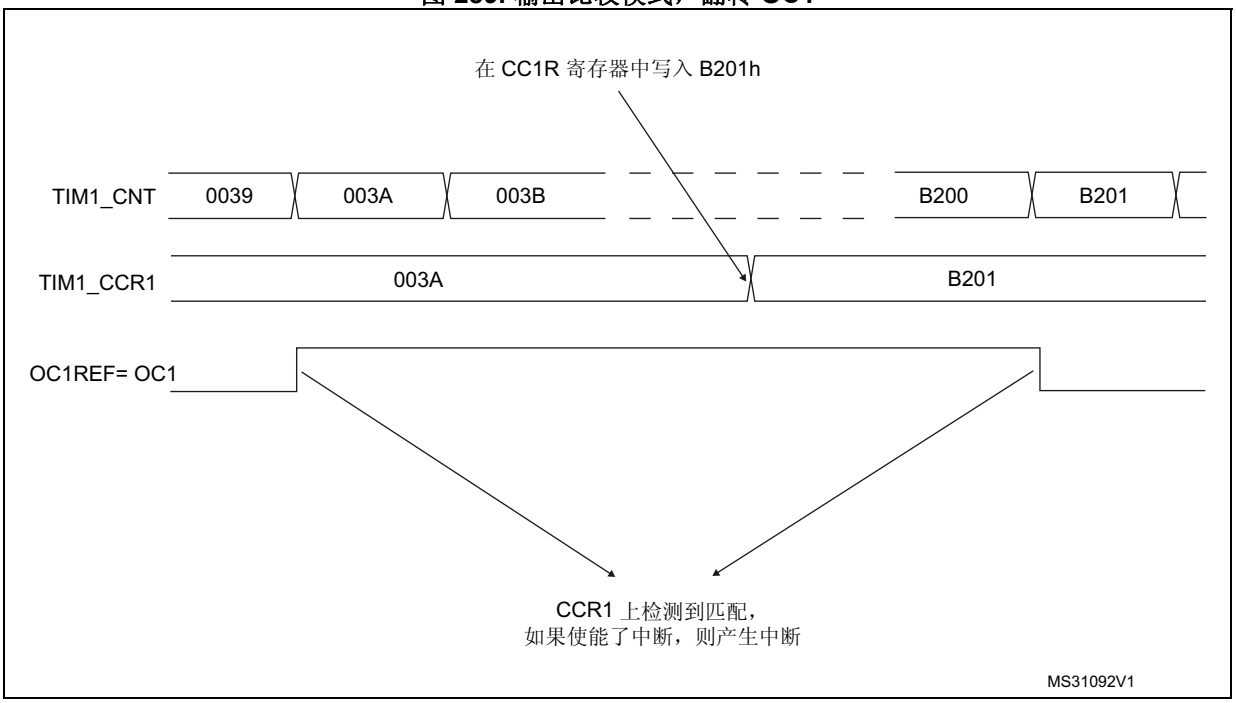
在输出比较模式下，更新事件 UEV 对 OCxREF 和 OCx 输出毫无影响。同步的精度可以达到计数器的一个计数周期。输出比较模式也可用于输出单脉冲（在单脉冲模式下）。

步骤

1. 选择计数器时钟（内部、外部、预分频器）。
2. 在 TIMx_ARR 和 TIMx_CCRx 寄存器中写入所需数据。
3. 如果要生成中断请求，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=0011 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 TIMx_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 TIMx_CCRx 寄存器以控制输出波形，前提是未使能预装载寄存器（OCxPE=“0”，否则 TIMx_CCRx 影子寄存器仅在下一更新事件 UEV 发生时进行更新）。*图 289* 给出了一个示例。

图 289. 输出比较模式，翻转 OC1



34.3.11 PWM 模式

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

各通道可以独立选择 PWM 模式（每个 OCx 输出对应一个 PWM），只需向 TIMx_CCMRx 寄存器的 OCxM 位写入“0110”（PWM 模式 1）或“0111”（PWM 模式 2）。必须通过将 TIMx_CCMRx 寄存器中的 OCxPE 位置 1 使能相应预装载寄存器，最后通过将 TIMx_CR1 寄存器中的 ARPE 位置 1 使能自动重载预装载寄存器（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 TIMx_EGR 寄存器中的 UG 位置 1 来初始化所有寄存器。

OCx 极性可通过软件来编程（使用 TIMx_CCER 寄存器的 CCxP 位）。可将其编程为高电平有效或低电平有效。通过 CCxE、CCxNE、MOE、OSSI 和 OSSR 位（TIMx_CCER 和 TIMx_BDTR 寄存器）的组合使能 OCx 输出。有关详细信息，请参见 TIMx_CCER 寄存器说明。

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

根据 TIMx_CR1 寄存器中的 CMS 位状态，定时器能够产生边沿对齐模式或中心对齐模式的 PWM 信号。

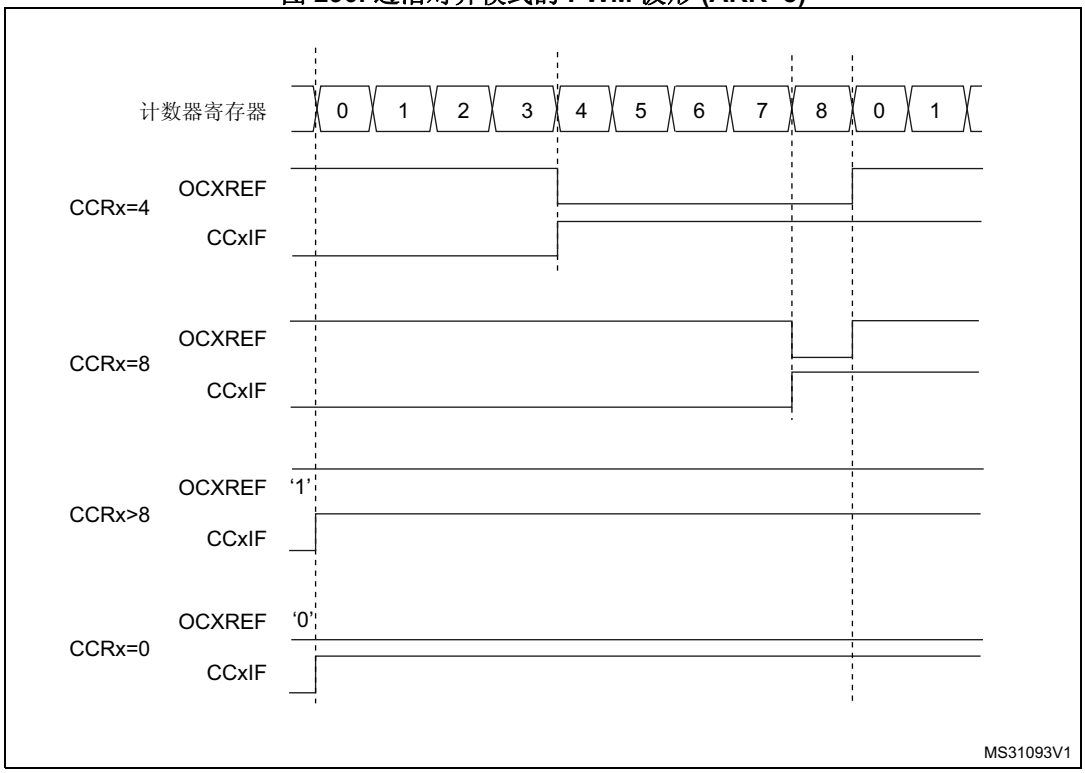
PWM 边沿对齐模式

- 递增计数配置

  当 TIMx_CR1 寄存器中的 DIR 位为低时执行递增计数。请参见*第 1044 页的递增计数模式*。

  以下以 PWM 模式 1 为例。只要 TIMx_CNT < TIMx_CCRx，PWM 参考信号 OCxREF 便为高电平，否则为低电平。如果 TIMx_CCRx 中的比较值大于自动重载值（TIMx_ARR 中），则 OCxREF 保持为“1”。如果比较值为 0，则 OCxRef 保持为“0”。*图 290* 举例介绍边沿对齐模式的一些 PWM 波形 (TIMx_ARR=8)。

图 290. 边沿对齐模式的 PWM 波形 (ARR=8)



- 递减计数配置

  当 TIMx_CR1 寄存器中的 DIR 位为高时执行递减计数。请参见*第 1048 页的递减计数模式*。

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

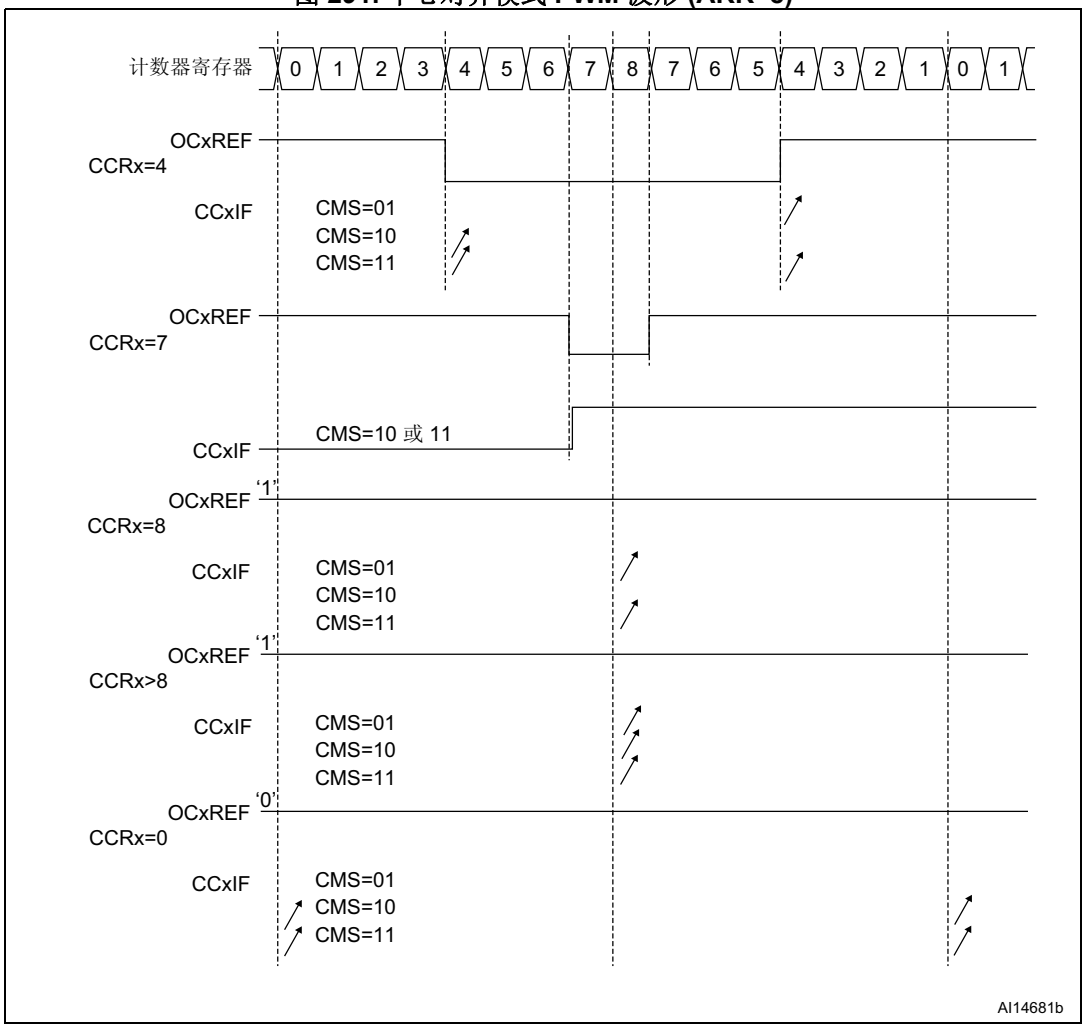
PWM 中心对齐模式

当 TIMx_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。TIMx_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*第 1051 页的中心对齐模式（递增/递减计数）*。

图 291 显示了中心对齐模式的 PWM 波形，在此例中：

- TIMx_ARR=8。
- PWM 模式为 PWM 模式 1。
- 在根据 TIMx_CR1 寄存器中 CMS=01 而选择的中心对齐模式 1 下，当计数器递减计数时，比较标志置 1。

图 291. 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 TIMx_CR1 寄存器中 DIR 位的值进行递增或递减计数。此外，不得同时通过软件修改 DIR 和 CMS 位。
- 不建议在运行中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 (TIMx_CNT>TIMx_ARR)，计数方向不会更新。例如，如果计数器之前递增计数，则继续递增计数。
  - 如果向计数器写入 0 或 TIMx_ARR 的值，计数方向会更新，但不生成更新事件 UEV。
- 使用中心对齐模式最为保险的方法是：在启动计数器前通过软件生成更新（将 TIMx_EGR 寄存器中的 UG 位置 1），并且不要在计数器运行过程中对其执行写操作。

34.3.12 不对称 PWM 模式

在不对称模式下，生成的两个中心对齐 PWM 信号间允许存在可编程相移。频率由 TIMx_ARR 寄存器的值确定，而占空比和相移则由一对 TIMx_CCRx 寄存器确定。两个寄存器分别控制递增计数和递减计数期间的 PWM，这样每半个 PWM 周期便会调节一次 PWM：

- OC1REFC（或 OC2REFC）由 TIMx_CCR1 和 TIMx_CCR2 控制
- OC3REFC（或 OC4REFC）由 TIMx_CCR3 和 TIMx_CCR4 控制

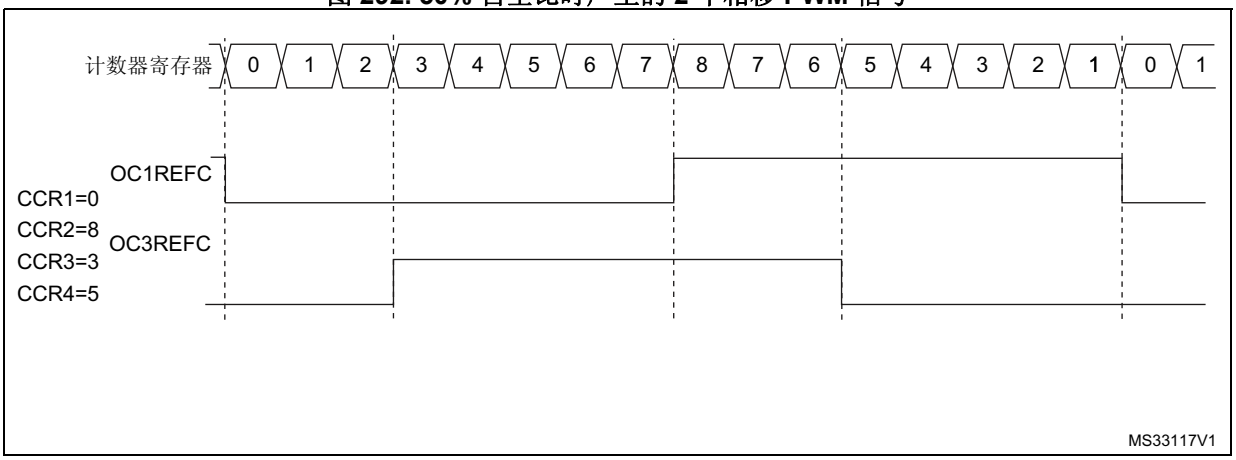
两个通道可以独立选择不对称 PWM 模式（每对 CCR 寄存器一个 OCx 输出），只需向 TIMx_CCMRx 寄存器的 OCxM 位写入“1110”（不对称 PWM 模式 1）或“1111”（不对称 PWM 模式 2）。

注： *出于兼容性原因，OCxM[3:0] 位域分为两部分，最高有效位与最低有效的 3 位不相邻。*

给定通道用作不对称 PWM 通道时，也可使用其互补通道。例如，如果通道 1 上产生 OC1REFC 信号（不对称 PWM 模式 1），则由于不对称 PWM 模式 1 的原因，通道 2 上可输出 OC2REF 信号或 OC2REFC 信号。

图 292 显示了不对称 PWM 模式下可以产生的信号示例（通道 1 到通道 4 在不对称 PWM 模式 1 下配置）。与死区发生器配合使用时，这可控制相移全桥直流到直流转换器。

图 292. 50% 占空比时产生的 2 个相移 PWM 信号



34.3.13 组合 PWM 模式

在组合 PWM 模式下，生成的两个边沿或中心对齐 PWM 信号的各个脉冲间允许存在可编程延时和相移。频率由 TIMx_ARR 寄存器的值确定，而占空比和延时则由两个 TIMx_CCRx 寄存器确定。产生的信号 OCxREFC 由两个参考 PWM 的逻辑或运算或者逻辑与运算组合组成。

- OC1REFC（或 OC2REFC）由 TIMx_CCR1 和 TIMx_CCR2 控制
- OC3REFC（或 OC4REFC）由 TIMx_CCR3 和 TIMx_CCR4 控制

两个通道可以独立选择组合 PWM 模式（每对 CCR 寄存器一个 OCx 输出），只需向 TIMx_CCMRx 寄存器的 OCxM 位写入“1100”（组合 PWM 模式 1）或“1101”（组合 PWM 模式 2）。

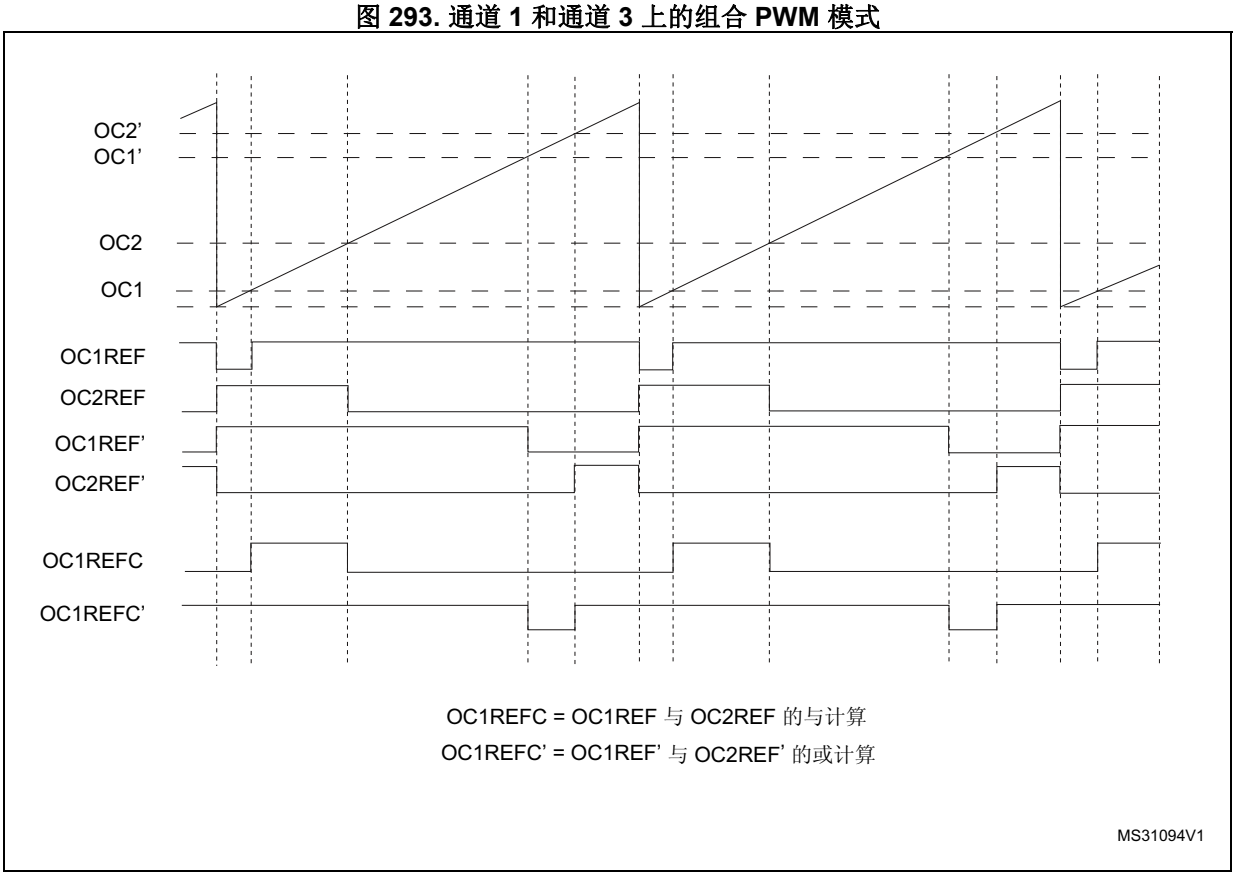
当给定通道用作组合 PWM 通道时，其互补通道必须在相反的 PWM 模式下配置（例如，一个通道在组合 PWM 模式 1 下配置，另一个通道在组合 PWM 模式 2 下配置）。

注： *出于兼容性原因，OCxM[3:0] 位域分为两部分，最高有效位与最低有效的 3 位不相邻。*

图 293 显示了不对称 PWM 模式下可以产生的信号示例，通过以下配置可获得这些信号：

- 通道 1 在组合 PWM 模式 2 下配置。
- 通道 2 在 PWM 模式 1 下配置。
- 通道 3 在组合 PWM 模式 2 下配置。
- 通道 4 在 PWM 模式 1 下配置。

图 293. 通道 1 和通道 3 上的组合 PWM 模式



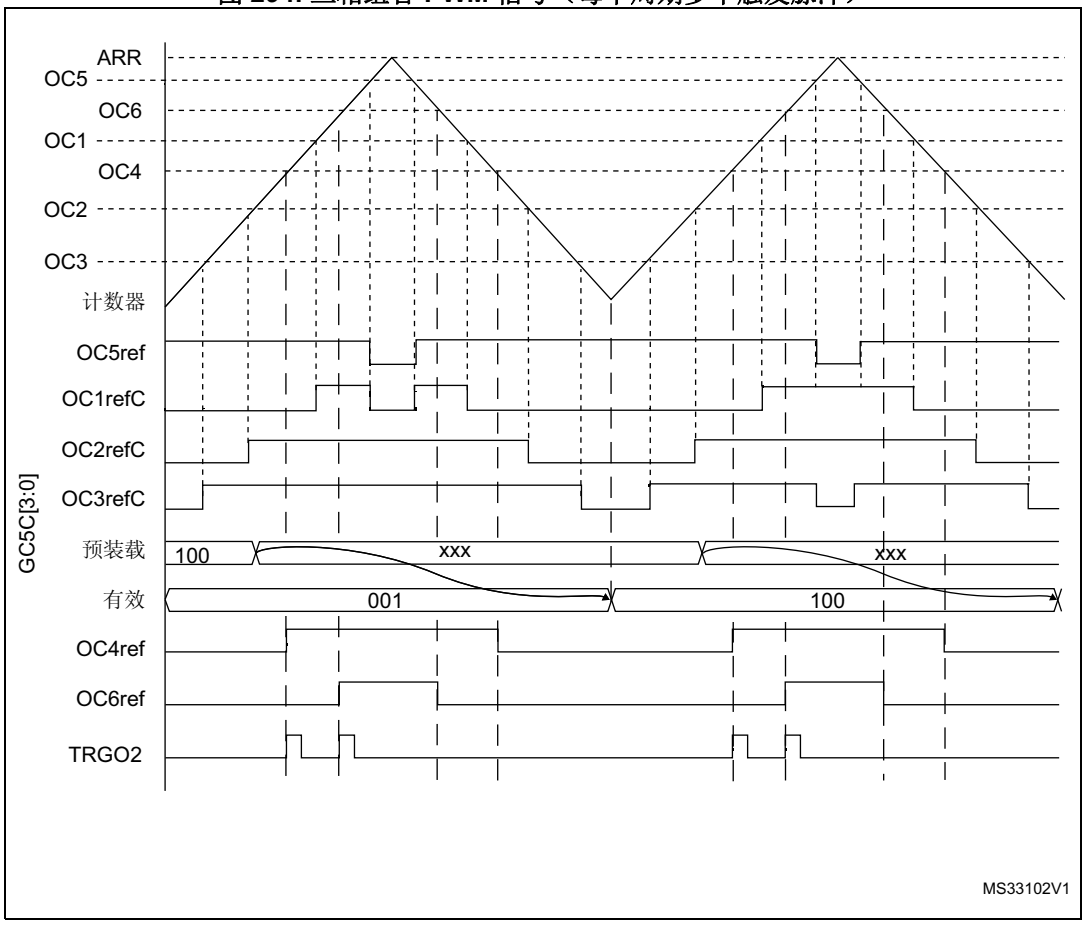
34.3.14 组合三相 PWM 模式

在组合三相 PWM 模式下，产生的一至三个中心对齐 PWM 信号与一个可编程信号间允许在脉冲中间进行逻辑与运算。OC5REF 信号用于定义产生的组合信号。凭借 TIMx_CCR5 中的 3 位 GC5C[3:1]，可以选择 OC5REF 与哪个参考信号组合。产生的信号 OCxREFC 由两个参考 PWM 的逻辑与运算组合组成。

- 如果 GC5C1 置 1，则 OC1REFC 由 TIMx_CCR1 和 TIMx_CCR5 控制
- 如果 GC5C2 置 1，则 OC2REFC 由 TIMx_CCR2 和 TIMx_CCR5 控制
- 如果 GC5C3 置 1，则 OC3REFC 由 TIMx_CCR3 和 TIMx_CCR5 控制

通道 1 到通道 3 可独立选择组合三相 PWM 模式，只需将 3 位 GC5C[3:1] 中的至少一位置 1。

图 294. 三相组合 PWM 信号（每个周期多个触发脉冲）



TRGO2 波形说明了如何根据给定的三相 PWM 信号同步 ADC。更多详细信息，请参见*第 34.3.27 节：ADC 同步*。

34.3.15 互补输出和死区插入

高级控制定时器 (TIM1/TIM8) 可以输出两路互补信号，并管理输出的关断与接通瞬间。

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

每路输出可以独立选择输出极性（主输出 OCx 或互补输出 OCxN）。可通过对 TIMx_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：TIMx_CCER 寄存器中的 CCxE 和 CCxNE 位以及 TIMx_BDTR 和 TIMx_CR2 寄存器中的 MOE、OISx、OISxN、OSSI 和 OSSR 位。更多详细信息，请参见*第 1116 页的表 216：具有断路功能的互补通道 OCx 和 OCxN 的输出控制位*。应当注意，切换至空闲状态（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。每个通道有一个 10 位死区发生器。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 295. 带死区插入的互补输出

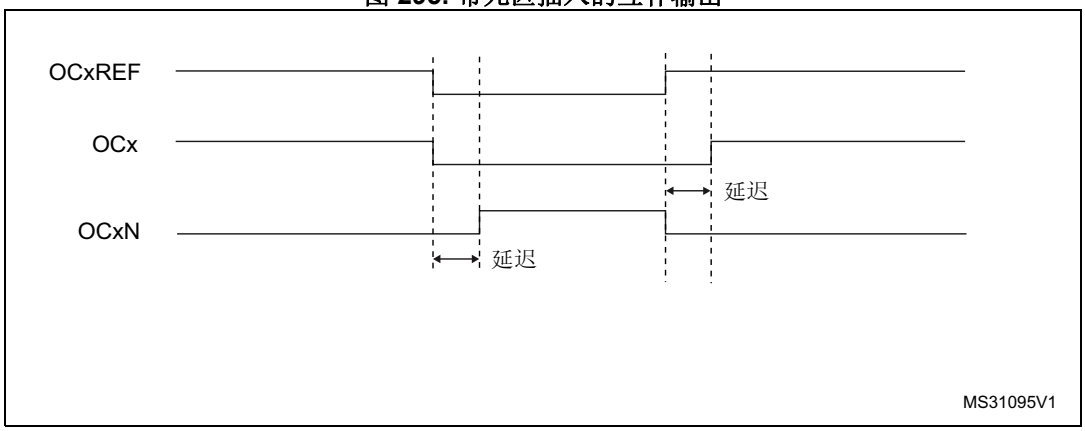


图 296. 延迟时间大于负脉冲宽度的死区波形

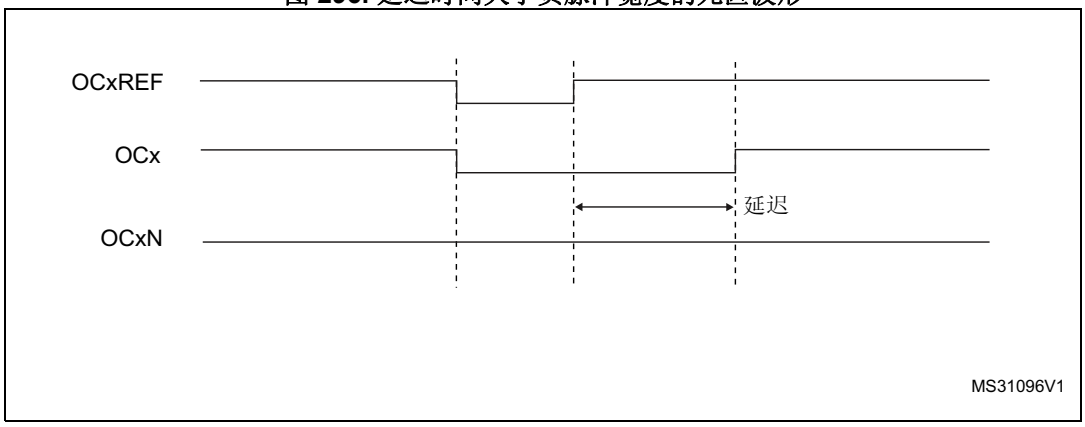
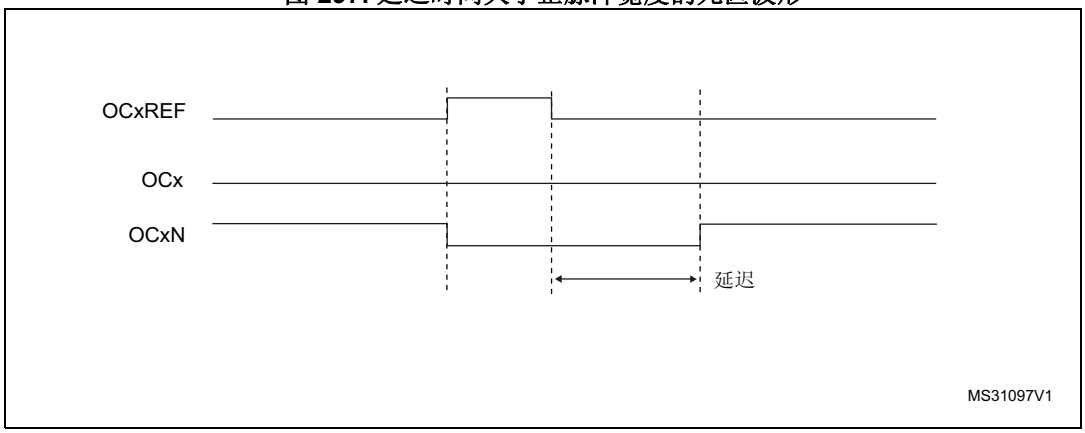


图 297. 延迟时间大于正脉冲宽度的死区波形



死区延迟对于所有通道均相同，可通过 TIMx_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*第 34.4.18 节：TIM1/TIM8 断路和死区寄存器 (TIMx_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 TIMx_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： *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。*

34.3.16 使用断路功能

断路功能的目的是保护由 TIM1 和 TIM8 定时器产生的 PWM 信号所驱动的功率开关。两个断路输入通常连接到功率级和三相逆变器的故障输出。激活时，断路电路会关闭 PWM 输出，并将其强制为预定义的安全状态。也可选择一些内部 MCU 事件来触发输出关断。

有两个断路通道。一个断路通道收集系统级故障（时钟失效和奇偶校验错误等）和应用故障（来自输入引脚和内置比较器），可以在死区持续时间后将输出强制为预定义的电平（有效或无效）。断路 2 通道只包括应用故障，能够将输出强制为无效状态。

断路期间的输出使能信号和输出电平取决于多个控制位：

- TIMx_BDTR 寄存器中的 MOE 位，允许通过软件使能/禁止输出，在发生断路和断路 2 事件时复位。
- TIMx_BDTR 寄存器中的 OSSI 位，定义定时器将输出控制在无效状态下，还是释放控制权给 GPIO 控制器（通常使其处于高阻态模式）
- TIMx_CR2 寄存器中的 OISx 和 OISxN 位，将输出设置为关断电平（有效或无效）。无论 OISx 和 OISxN 的值为何，均无法在给定时间将 OCx 和 OCxN 输出同时设置为有效电平。更多详细信息，请参见*第 1116 页的表 216：具有断路功能的互补通道 OCx 和 OCxN 的输出控制位*。

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 TIMx_BDTR 寄存器中的 BKE 位和 BKE2 位置 1，可使能断路功能。可通过配置同一寄存器中的 BKP 位和 BKP2 位来选择断路输入的极性。BKEx 和 BKPx 位可同时修改。对 BKEx 和 BKPx 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 TIMx_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。

可以使用 TIMx_OR2 和 TIMx_OR3 寄存器从多个源产生断路，这些源可以单独使能并使用可编程边沿有效性。

断路 (BRK) 通道的源为：

- 连接到 BKIN 引脚的外部源（由 AFIO 控制器设定），具有极性选择和可选的数字滤波
- 内部源：
  - Cortex®-M4 LOCKUP 输出
  - PVD 输出
  - SRAM 奇偶校验错误信号
  - Flash ECC 错误
  - CSS 检测器产生时钟故障事件
  - 比较器的输出，具有极性选择和可选的数字滤波
  - DFSDM1 外设的模拟看门狗输出

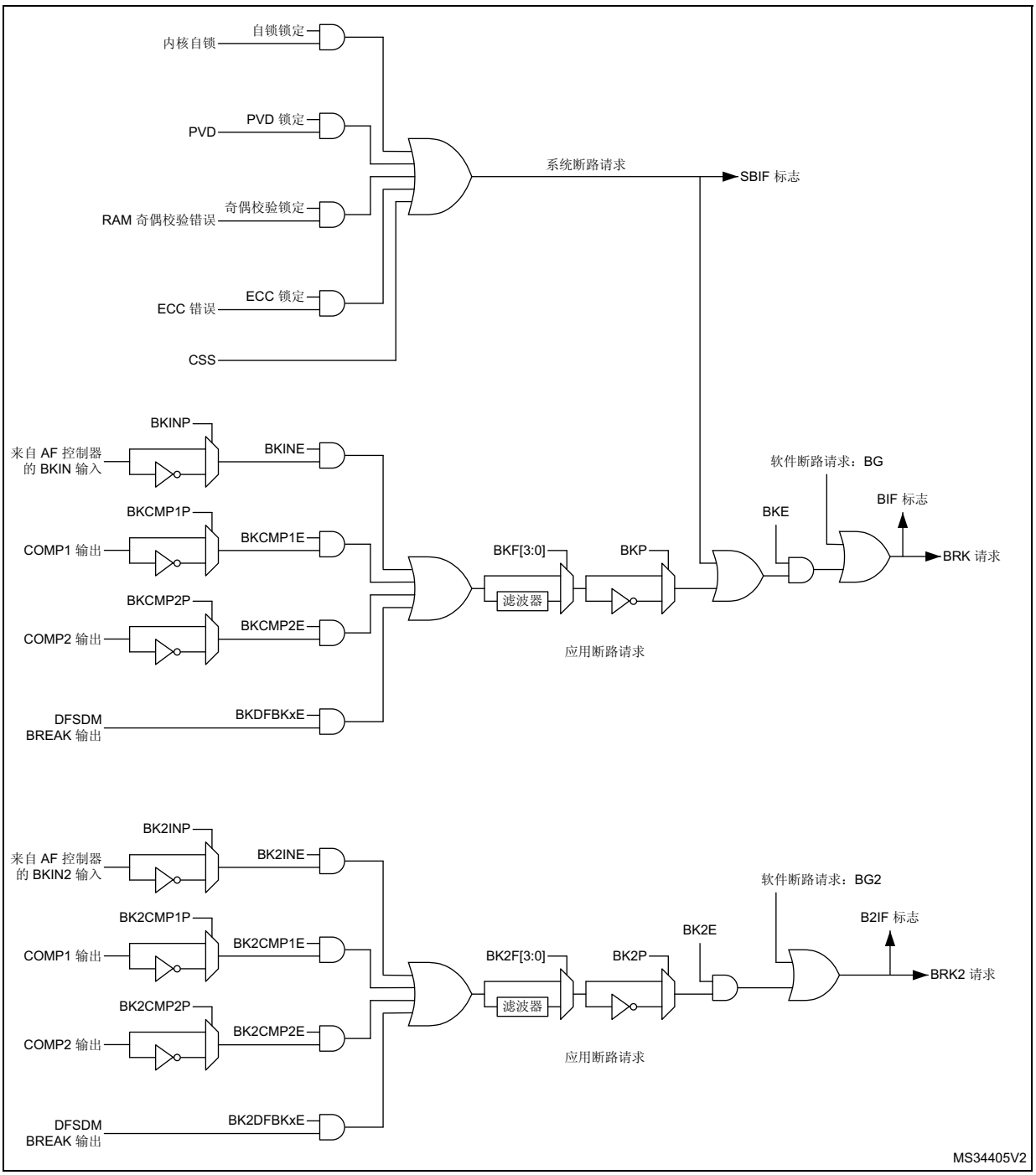
断路 2 (BRK2) 的源：

- 连接到 BKIN2 引脚的外部源（由 AFIO 控制器设定），具有极性选择和可选的数字滤波
- 来自比较器输出的内部源

也可由软件通过 TIMx_EGR 寄存器中的 BG 和 B2G 位产生断路事件。无论 BKE 和 BKE2 使能位的值如何，都可以使用 BG 和 BG2 通过软件生成断路。

在所有源进入定时器 BRK 或 BRK2 输入之前，对其进行 OR 运算，如以下*图 298* 所示。

图 298. 断路和断路 2 电路概述



注： *只有禁止可编程滤波器时才能保证异步（无时钟）操作。如果使能可编程滤波器，必须使用故障安全时钟模式（例如，使用内部 PLL 和/或 CSS）来保证能够处理断路事件。*

发生断路之一（其中一个断路输入上出现所选电平）时：

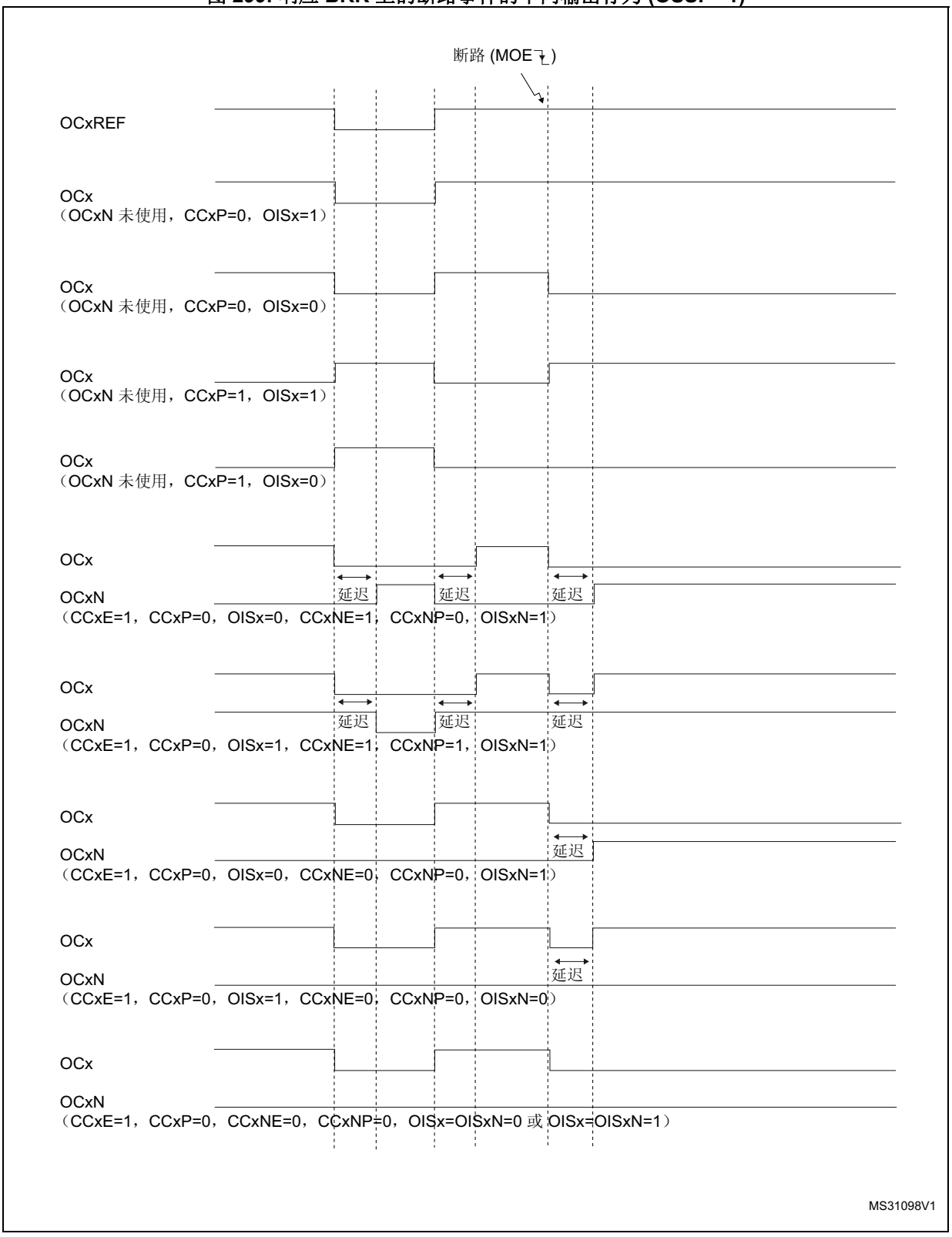
- MOE 位异步清零，使输出处于无效状态、空闲状态甚至释放控制权给 GPIO 控制器（通过 OSSI 位进行选择）。即使 MCU 振荡器关闭，该功能仍然使能。
- MOE=0 时，将以 TIMx_CR2 寄存器 OISx 位中编程的电平驱动每个输出通道。如果 OSSI=0，定时器将释放输出控制（由 GPIO 控制器接管），否则使能输出保持高电平。
- 使用互补输出时：
  - 输出首先置于无效状态（取决于极性）。这是异步操作，因此即使没有为定时器提供时钟，该操作仍有效。
  - 如果定时器时钟仍存在，则将重新激活死区发生器，进而在死区后以 OISx 和 OISxN 位中编程的电平驱动输出。即使在这种情况下，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稍长一些（约 2 个 ck_tim 时钟周期）。
  - 如果 OSSI=0，定时器将释放输出控制（由强制高阻态的 GPIO 控制器接管），否则使能输出将保持高电平或在 CCxE 或 CCxNE 位之一为高电平时立即变为高电平。
- 将断路状态标志（TIMx_SR 寄存器中的 SBIF、BIF 和 B2IF 位）置 1。如果 TIMx_DIER 寄存器中的 BIE 位置 1，则会产生中断。如果 TIMx_DIER 寄存器中的 BDE 位置 1，可发送 DMA 请求。
- 如果 TIMx_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。例如，这可用于执行调节。否则，MOE 将始终保持低电平，直到应用将其再次置 1。这种情况下，这一特性可用于确保安全。可以将断路输入连接到功率驱动器的警报、温度传感器或任何安全元件。

注： *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 和 B2IF 清零。*

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）。应用可以通过 TIMx_BDTR 寄存器中的 LOCK 位，从 3 种保护级别中进行选择。请参见*第 34.4.18 节：TIM1/TIM8 断路和死区寄存器 (TIMx_BDTR)*。MCU 复位后只能对 LOCK 位执行一次写操作。

图 299 所示为输出对断路响应行为的示例。

图 299. 响应 BRK 上的断路事件的不同输出行为 (OSSI = 1)



两个断路输入针对定时器输出具有不同的行为：

- BRK 输入可禁止（无效状态）PWM 输出，也可将 PWM 输出强制为预定义的安全状态。
- BRK2 只能禁止（无效状态）PWM 输出。

BRK 输入的优先级高于 BRK2 输入，如*表 212* 所示。

注： *BRK2 必须只在 OSSR = OSSI = 1 时使用。*

表 212. 定时器输出行为与 BRK/BRK2 输入

BRK	BRK2	定时器输出状态	典型用例	
			OCxN 输出（下桥臂开关）	OCx 输出（上桥臂开关）
有效	X	– 无效，之后强制为输出状态（死区后） – 如果 OSSI = 0，则禁止输出（由 GPIO 逻辑接管控制）	死区插入后开启	关闭
无效	有效	无效	关闭	关闭

图 300 给出了 BRK 和 BRK2 输入上出现有效信号时 OCx 和 OCxN 输出行为示例。在这种情况下，两个输出的极性均为高电平有效（TIMx_CCER 寄存器中的 CCxP = CCxNP = 0）。

图 300. BRK 和 BRK2 引脚使能后的 PWM 输出状态 (OSSI=1)

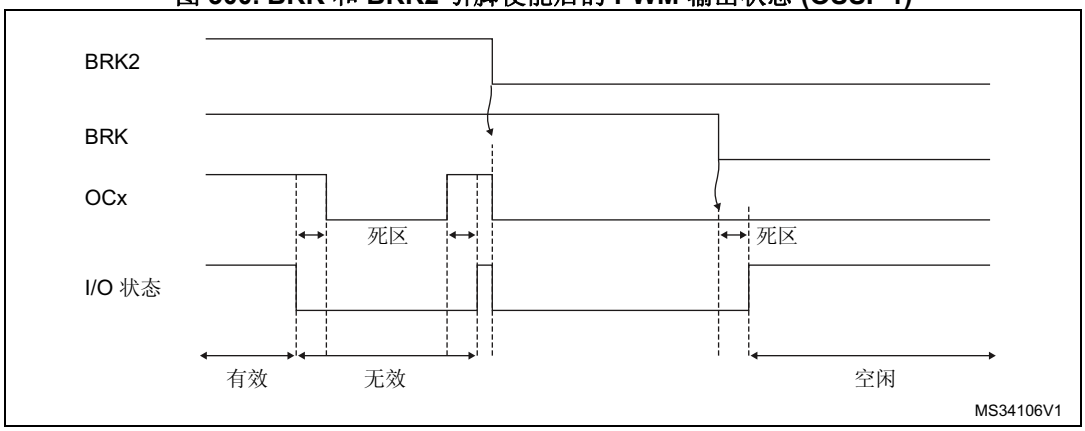
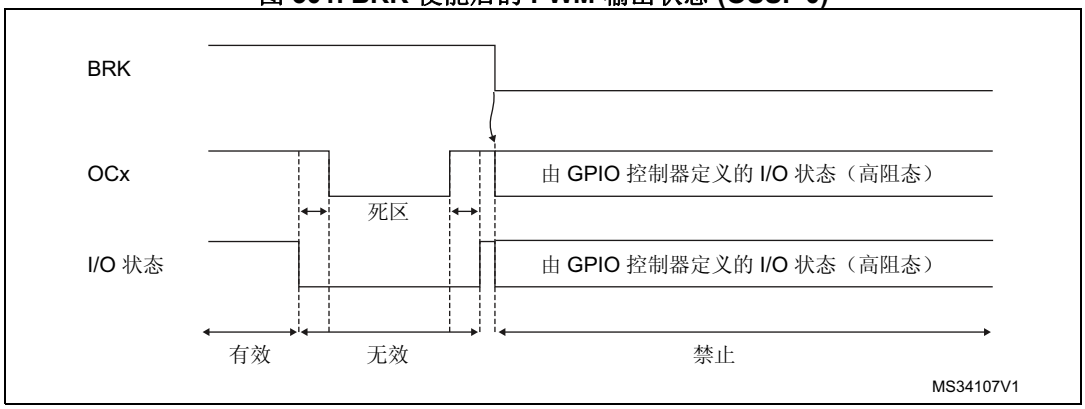


图 301. BRK 使能后的 PWM 输出状态 (OSSI=0)



34.3.17 双向断路输入

TIM1/TIM8 具有双向断路 I/O，如*图 302* 所示。

它们可以：

- 板级全局断路信号，用于向外部 MCU 或栅极驱动器发送故障信号，唯一的引脚作为输入和输出状态引脚。
- 在必须将多个内部和外部断路输入合并时，将内部断路源和多个外部开漏比较器输出“或”连接在一起，触发唯一断路事件。

使用 TIMxBDTR 寄存器的 BKBID 和 BK2BID 位，将断路和断路 2 输入配置为双向模式。可以使用 TIMxBDTR 寄存器中的 LOCK 位，将 BKBID 编程位锁定在只读模式（锁定级别 1 或更高）。

双向模式可以用于断路和断路 2 输入，需要将 I/O 配置为低电平有效极性的开漏模式（使用 BKINP、BKP、BK2INP 和 BK2P 位进行配置）。任何来自系统（例如 CSS）、片上外设或断路输入的断路请求都会强制将断路输入置为低电平，以通知发生了故障事件。如果未正确设置极性位（高电平有效极性），则出于安全目的禁止双向模式。

软件断路事件（BG 和 B2G）也会导致断路 I/O 被强制置为“0”，从而向外部组件指示定时器已进入断路状态。但是仅在使能断路 (BK(2)E = 1) 时有效。当生成软件断路事件，并且 BK(2)E = 0 时，输出会置于安全状态，并将断路标志置 1，但对断路 (2) I/O 无影响。

安全解除机制可防止系统被明确锁定（断路输入上的低电平会触发断路，进而将相同输入强制置为低电平）。

当 BKDSRM (BK2DSRM) 位置 1 时，会释放断路输出以清除故障信号，从而使系统能够重新获得保护。

在任何情况下都不能禁止断路保护电路：

- 断路输入路径始终有效：即使 BKDSRM (BK2DSRM) 位置 1 且释放开漏控制，断路事件也仍然有效。这样可以在发生断路期间防止 PWM 输出重新启动。
- 使能输出（MOE 位置 1）后，BK(2)DSRM 位不能解除断路保护（请参见*表 213*）。

表 213. 断路保护解除条件

MOE	BKDIR (BK2DIR)	BKDSRM (BK2DSRM)	断路保护状态
0	0	X	保护
0	1	0	保护
0	1	1	解除
1	X	X	保护

启动和重新启动断路电路

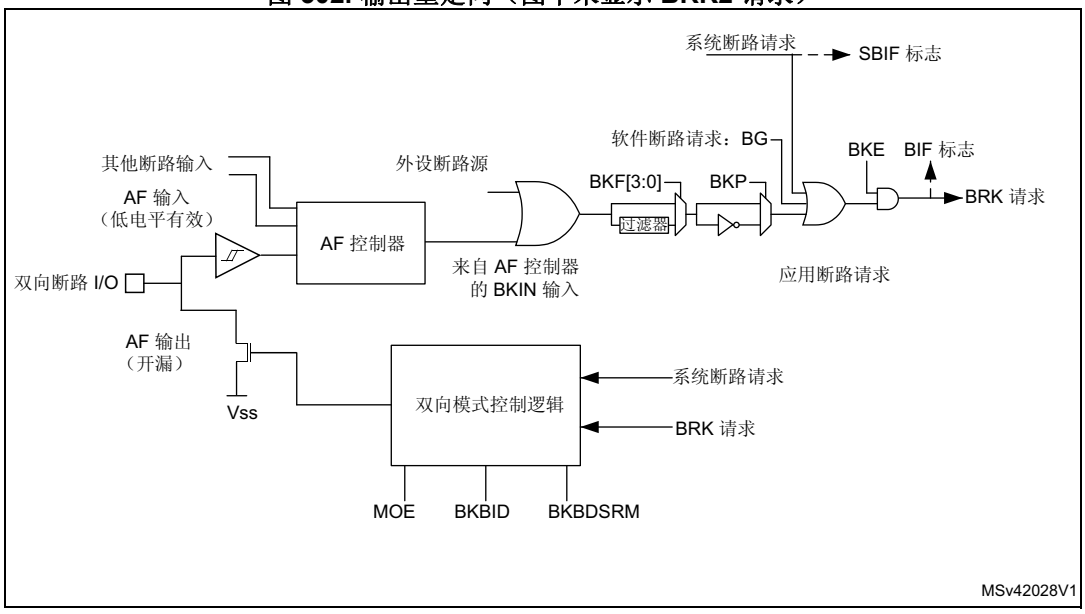
默认情况下（外设复位配置）会启动断路电路（在输入或双向模式下）。

发生断路（断路 2）事件后，必须按照以下步骤重新启动保护：

- 必须将 BKDSRM (BK2DSRM) 位置 1，以释放输出控制
- 软件必须等待系统断路条件消失（如果有），并清零 SBIF 状态标志（或在重新启动前由系统清零）
- 软件必须轮询 BKDSRM (BK2DSRM) 位，直到该位由硬件清零（当应用程序断路条件消失时）

此后，断路电路即启动并激活，可以通过将 MOE 位置 1 来重新使能 PWM 输出。

图 302. 输出重定向（图中未显示 BRK2 请求）



34.3.18 发生外部事件时清除 OCxREF 信号

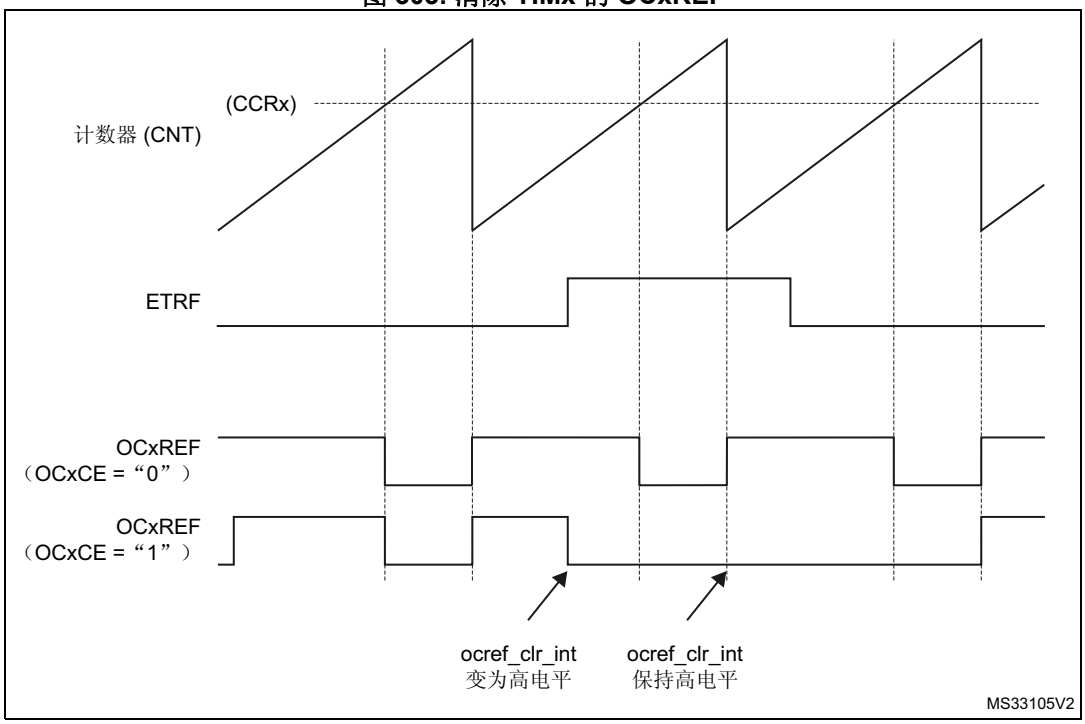
对于给定通道，在 ETRF 输入施加高电平（相应 TIMx_CCMRx 寄存器中的 OCxCE 使能位置 1），可将 OCxREF 信号复位。OCxREF 信号将保持低电平，直到发生下一更新事件 (UEV)。该功能只能在输出比较模式和 PWM 模式下使用。在强制模式下不起作用。例如，OCxREF 信号可以连接到比较器的输出，用于控制电流。

选择 ETRF 时，ETR 必须配置如下：

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

图 303 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 TIMx 编程为 PWM 模式。

图 303. 清除 TIMx 的 OCxREF



注： *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。*

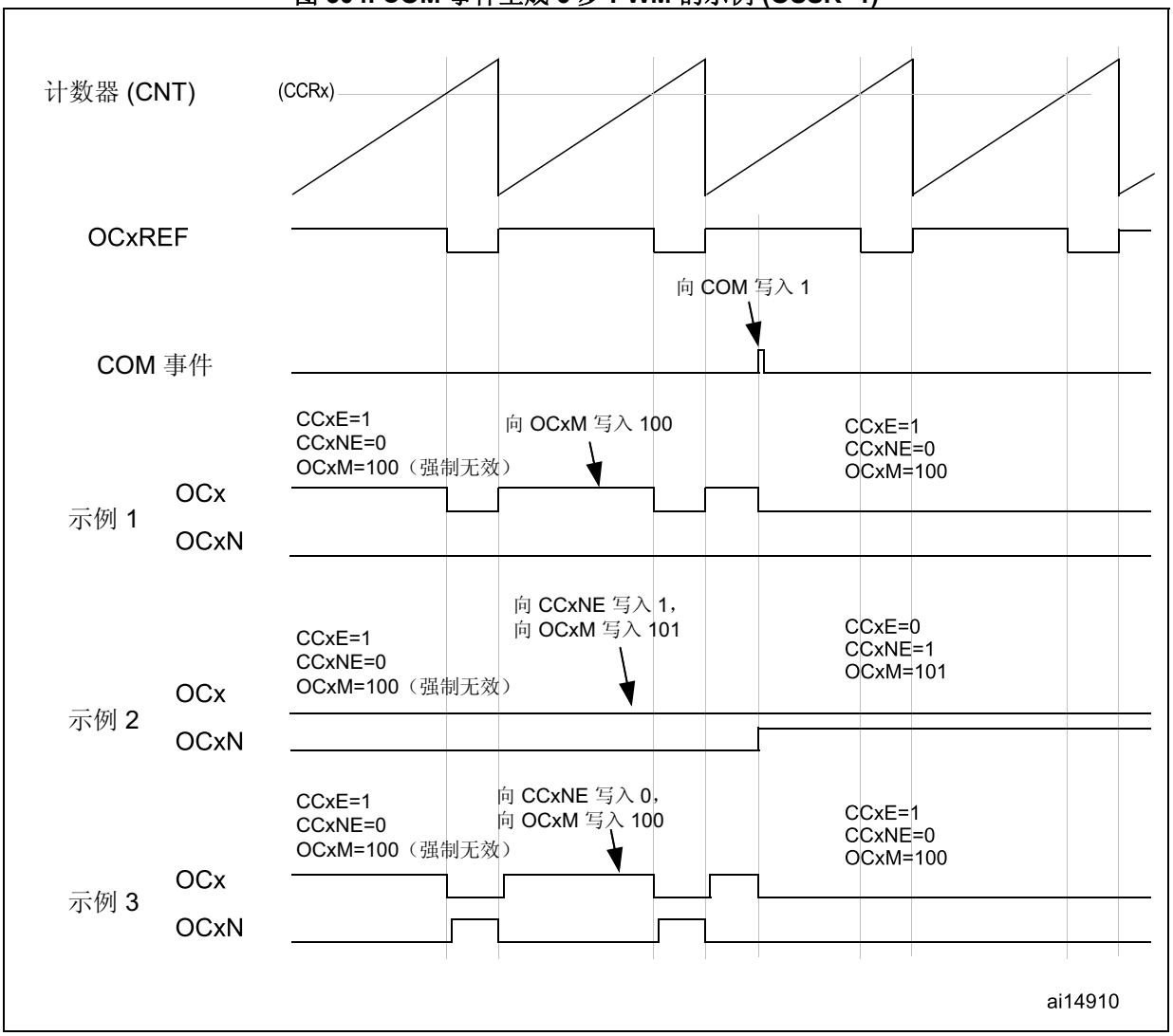
34.3.19 生成 6 步 PWM

当通道使用互补输出时，OCxM、CCxE 和 CCxNE 位上提供预装载位。发生 COM 换向事件时，这些预装载位将传输到影子位。因此，用户可以预先编程下一步骤的配置，并同时更改所有通道的配置。COM 可由软件通过将 TIMx_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（TIMx_SR 寄存器中的 COMIF 位）将会置 1。这时，如果 TIMx_DIER 寄存器中的 COMIE 位置 1，将产生中断；如果 TIMx_DIER 寄存器中的 COMDE 位置 1，则将产生 DMA 请求。

图 304 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 304. COM 事件生成 6 步 PWM 的示例 (OSSR=1)



34.3.20 单脉冲模式

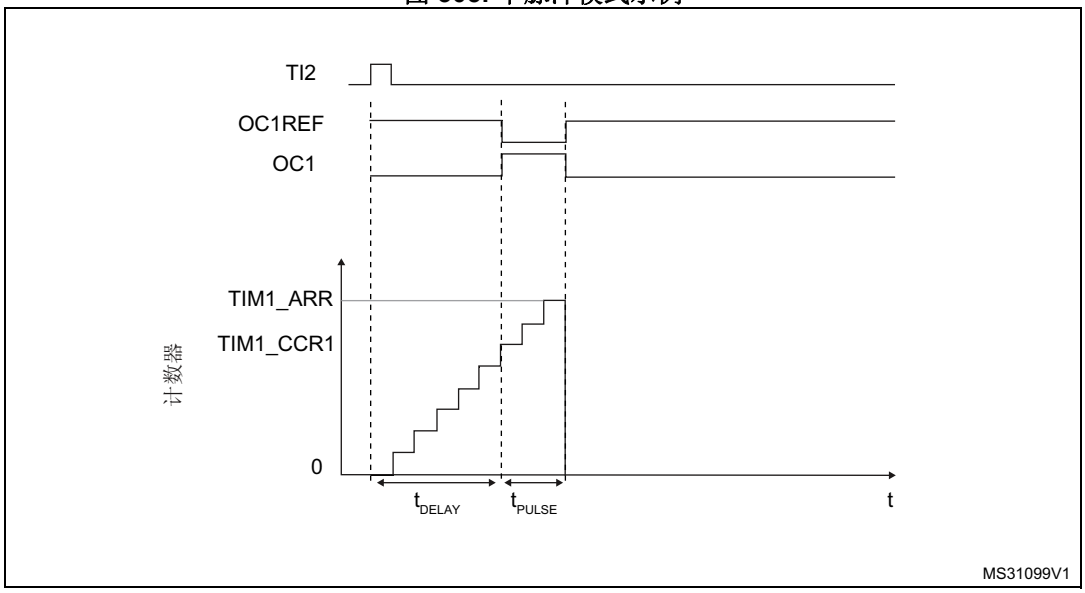
单脉冲模式 (OPM) 是上述模式的一个特例。在这种模式下，计数器可以在一个激励信号的触发下启动，并可在一段可编程的延时后产生一个脉宽可编程的脉冲。

可以通过从模式控制器启动计数器。可以在输出比较模式或 PWM 模式下生成波形。将 TIMx_CR1 寄存器中的 OPM 位置 1，即可选择单脉冲模式。这样，发生下一更新事件 UEV 时，计数器将自动停止。

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

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

图 305. 单脉冲模式示例



例如，用户希望达到这样的效果：在 TI2 输入引脚检测到上升沿时，经过 t_{DELAY} 的延迟，在 OC1 上产生一个长度为 t_{PULSE} 的正脉冲。

使用 TI2FP2 作为触发 1：

1. 在 TIMx_CCMR1 寄存器中写入 CC2S=“01”，以将 TI2FP2 映射到 TI2。
2. 在 TIMx_CCER 寄存器中写入 CC2P=“0”和 CC2NP=“0”，使 TI2FP2 能够检测上升沿。
3. 在 TIMx_SMCR 寄存器中写入 TS=110，将 TI2FP2 配置为从模式控制器的触发 (TRGI)。
4. 在 TIMx_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）。

- t_{DELAY} 由写入 TIMx_CCR1 寄存器的值定义。
- t_{PULSE} 由自动重载值与比较值之差 (TIMx_ARR - TIMx_CCR1) 来定义。
- 假设希望产生这样的波形：信号在发生比较匹配时从“0”变为“1”，在计数器达到自动重载值时由“1”变为“0”。为此，应在 TIMx_CCMR1 寄存器中写入 OC1M=111，以使能 PWM 模式 2。如果需要，可选择在 TIMx_CCMR1 寄存器的 OC1PE 和 TIMx_CR1 寄存器的 ARPE 中写入“1”，以使能预装载寄存器。这种情况下，必须在 TIMx_CCR1 寄存器中写入比较值并在 TIMx_ARR 寄存器中写入自动重载值，通过将 UG 位置 1 来产生更新，然后等待 TI2 上的外部触发事件。本例中，CC1P 的值为“0”。

在本例中，TIMx_CR1 寄存器中的 DIR 和 CMS 位应为低。

由于仅需要 1 个脉冲（单脉冲模式），因此应向 TIMx_CR1 寄存器的 OPM 位写入“1”，以便在发生下一更新事件（计数器从自动重载值返回到 0）时使计数器停止计数。TIMx_CR1 寄存器中的 OPM 位置“0”时，即选择重复模式。

特殊情况：OCx 快速使能：

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

如果要输出延迟时间最短的波形，可以将 TIMx_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。

34.3.21 可再触发单脉冲模式 (OPM)

该模式允许计数器可以在一个激励信号的触发下启动，并且能产生长度可编程的脉冲，但与不可再触发单脉冲模式间存在以下差别，如*第 34.3.20 节*所述：

- 发生触发时，脉冲立即产生（无可编程延时）
- 如果在上一个触发完成前发生新的触发，脉冲将延长

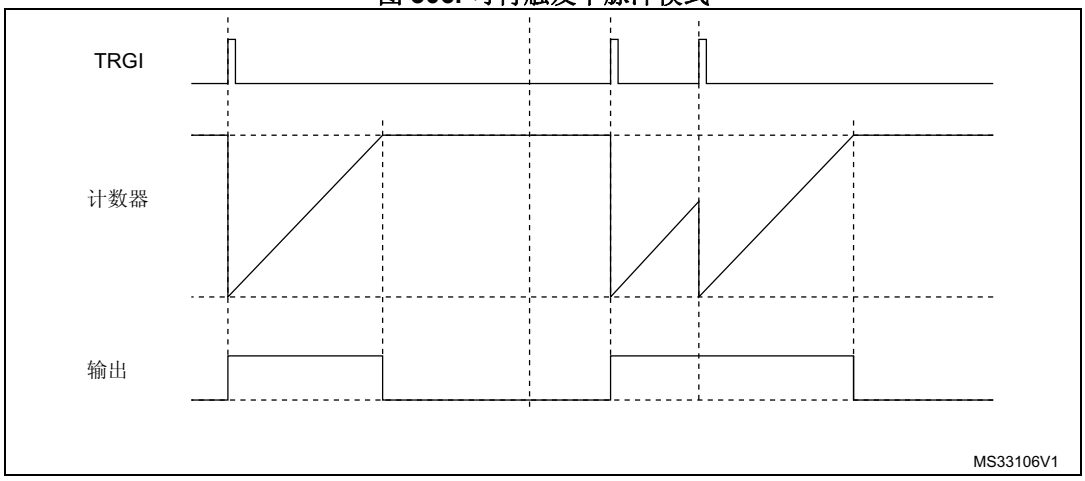
定时器必须处于从模式，TIMx_SMCR 寄存器中的位 SMS[3:0] =“1000”（组合复位 + 触发模式），针对可再触发 OPM 模式 1 或模式 2 将 OCxM[3:0] 位设置为“1000”或“1001”。

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

注： *出于兼容性原因，OCxM[3:0] 和 SMS[3:0] 位域分为两部分，最高有效位与最低有效的 3 位不相邻。*

此模式不能与中心对齐 PWM 模式一起使用。在 TIMx_CR1 中必须设置 CMS[1:0] = 00。

图 306. 可再触发单脉冲模式



34.3.22 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 TIMx_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 TIMx_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 两个输入用于连接正交编码器。请参见 *表 214*。如果使能计数器（在 TIMx_CR1 寄存器的 CEN 位中写入“1”），则计数器的时钟由 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 TIMx_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 TIMx_ARR 寄存器中的自动重载值之间进行连续计数（根据具体方向，从 0 递增计数到 ARR，或从 ARR 递减计数到 0）。因此，在启动前必须先配置 TIMx_ARR。同样，捕获、比较、重复计数器和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。

注： *使能编码器模式时，预分频器必须设置为零*

在此模式下，计数器会根据正交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 214. 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

正交编码器可直接与 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 307 以计数器工作为例，说明了计数信号的产生和方向控制。同时也说明了选择双边沿时如何对输入抖动进行补偿。将传感器靠近其中一个切换点放置时可能出现这种情况。本例中假设配置如下：

- CC1S=“01”（TIMx_CCMR1 寄存器，TI1FP1 映射到 TI1 上）。
- CC2S=“01”（TIMx_CCMR2 寄存器，TI1FP2 映射到 TI2 上）。
- CC1P=“0”，CC1NP=“0”（TIMx_CCER 寄存器，TI1FP1 未反相，TI1FP1=TI1）。
- CC2P=“0”，CC2NP=“0”（TIMx_CCER 寄存器，TI1FP2 未反相，TI1FP2= TI2）。
- SMS=“011”（TIMx_SMCR 寄存器，两个输入在上升沿和下降沿均有效）。
- CEN=“1”（TIMx_CR1 寄存器，使能计数器）。

图 307. 编码器接口模式下的计数器工作示例

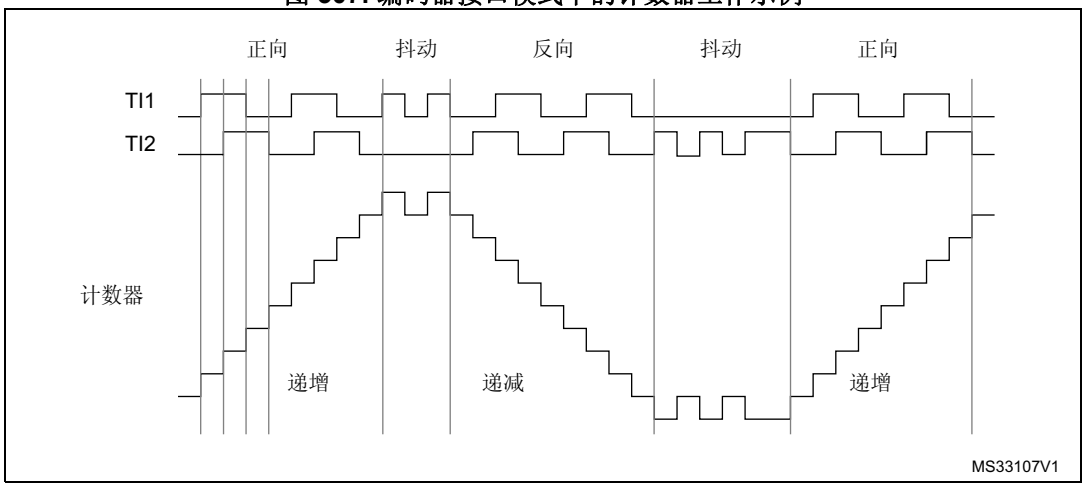
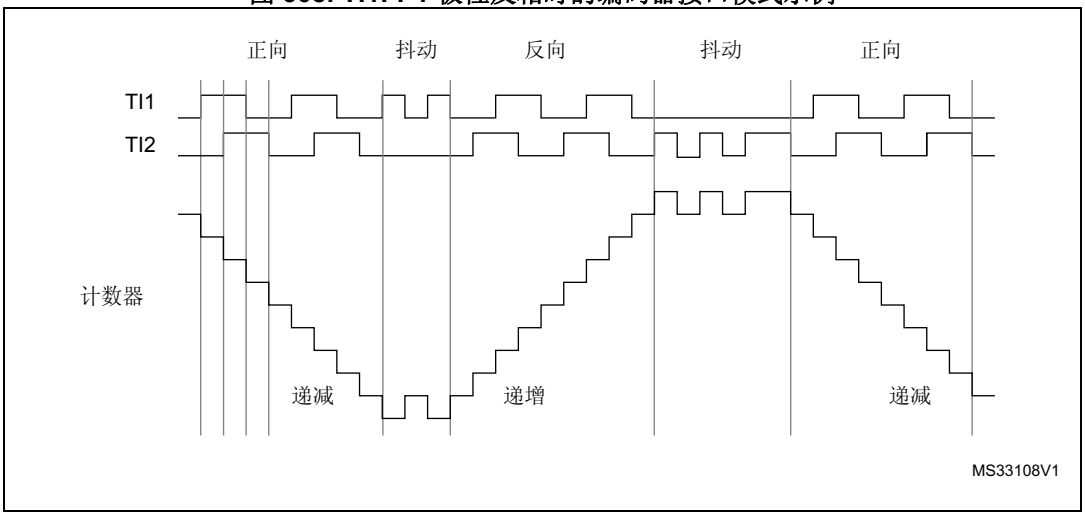


图 308 举例说明 TI1FP1 极性反相时计数器的行为（除 CC1P=“1”外，其他配置与上例相同）。

图 308. 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 DMA 请求读取计数器值。

在 32 位定时器实现中，当 UIFREMAP 位置 1 时，计数器的位 31 在读访问时由 UIFCPY 标志覆盖（计数器的最高有效位只能在写模式下访问）。

34.3.23 UIF 位重映射

TIMx_CR1 寄存器中的 UIFREMAP 位强制将更新中断标志 UIF 连续复制到定时计数器寄存器的位 31 (TIMxCNT[31]) 中。这样便可自动读取计数器值以及由 UIFCPY 标志发出的电位翻转条件。在特定情况下，这可避免在后台任务（计数器读）和中断（更新中断）之间共享处理时产生竞争条件，从而简化计算。

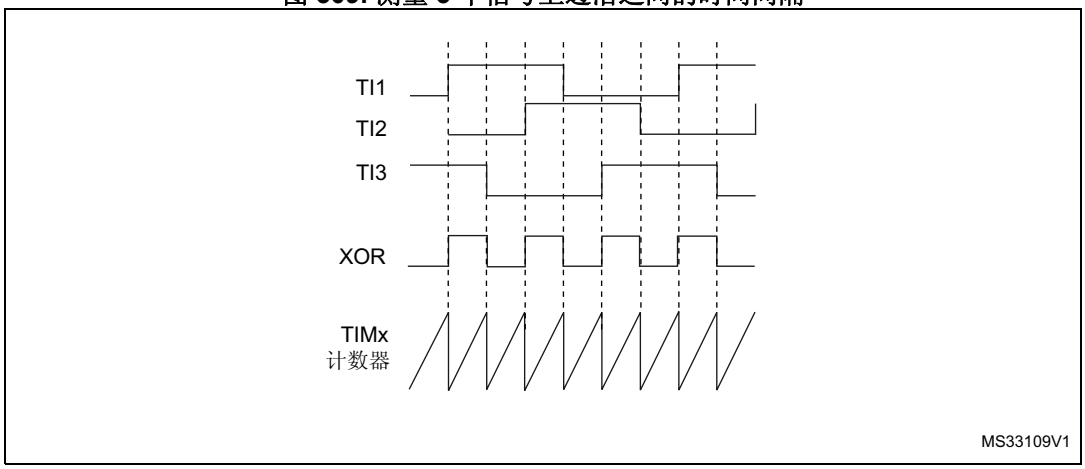
UIF 和 UIFCPY 标志使能之间没有延迟。

34.3.24 定时器输入异或功能

通过 TIMx_CR2 寄存器中的 TI1S 位，可将通道 1 的输入滤波器连接到异或门的输出，从而将 TIMx_CH1 到 TIMx_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。这样便于测量两个输入信号上边沿之间的间隔（如下面的*图 309* 所示）。

图 309. 测量 3 个信号上边沿之间的时间间隔



34.3.25 连接霍尔传感器

可通过用于产生电机驱动 PWM 信号的高级控制定时器（TIM1 或 TIM8）以及*图 310* 中称为“接口定时器”的另一个定时器 TIMx（TIM2、TIM3 和 TIM4），实现与霍尔传感器的连接。3 个定时器输入引脚（CC1、CC2 和 CC3）通过异或门连接到 TI1 输入通道（通过将 TIMx_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*第 1062 页上的图 283：捕获/比较通道（示例：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（TIM1 或 TIM8）各个通道的配置。TIM1 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（TIM1 或 TIM8）。

示例：霍尔输入与一个 TIMx 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 TIM1 的 PWM 配置。

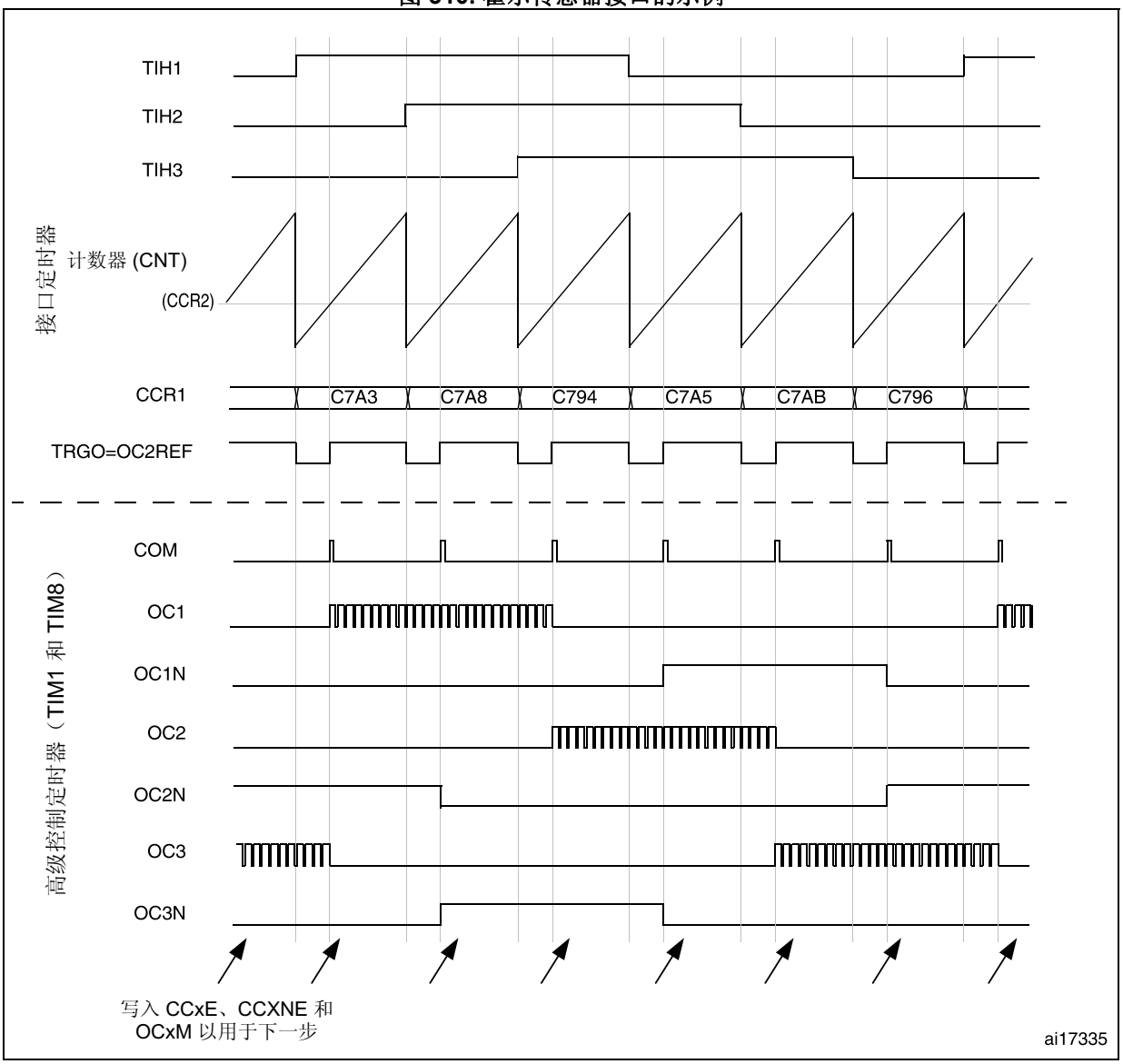
- 向 TIMx_CR2 寄存器的 TI1S 位写入“1”，使 3 个定时器输入经过异或运算后进入 TI1 输入通道。
- 时基编程：向 TIMx_ARR 写入其最大值（计数器必须通过 TI1 的变化清零）。设置预分频器，以得到最大计数器周期，该周期长于传感器上两次变化的间隔时间。
- 将通道 1 编程为捕获模式（选择 TRC）：向 TIMx_CCMR1 寄存器的 CC1S 位写入“01”。如果需要，还可以编程数字滤波器。

- 将通道 2 编程为 PWM 2 模式，并具有所需延迟：向 TIMx_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 TIMx_CR2 寄存器的 MMS 位写入“101”。

在高级控制定时器 TIM1 中，必须选择正确的 ITR 输入作为触发输入，定时器编程为可产生 PWM 信号，捕获/比较控制信号进行预装载（TIMx_CR2 寄存器的 CCPC=1），并且 COM 事件由触发输入控制（TIMx_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 310 为本示例的示意图。

图 310. 霍尔传感器接口的示例



34.3.26 定时器同步

TIMx 定时器从内部连接在一起，以实现定时器同步或链接。它们可在以下几种模式下实现同步：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 TIMx_CR1 寄存器中的 URS 位为 0，则会生成更新事件 UEV。然后，所有预装载寄存器（TIMx_ARR 和 TIMx_CCRx）都将更新。

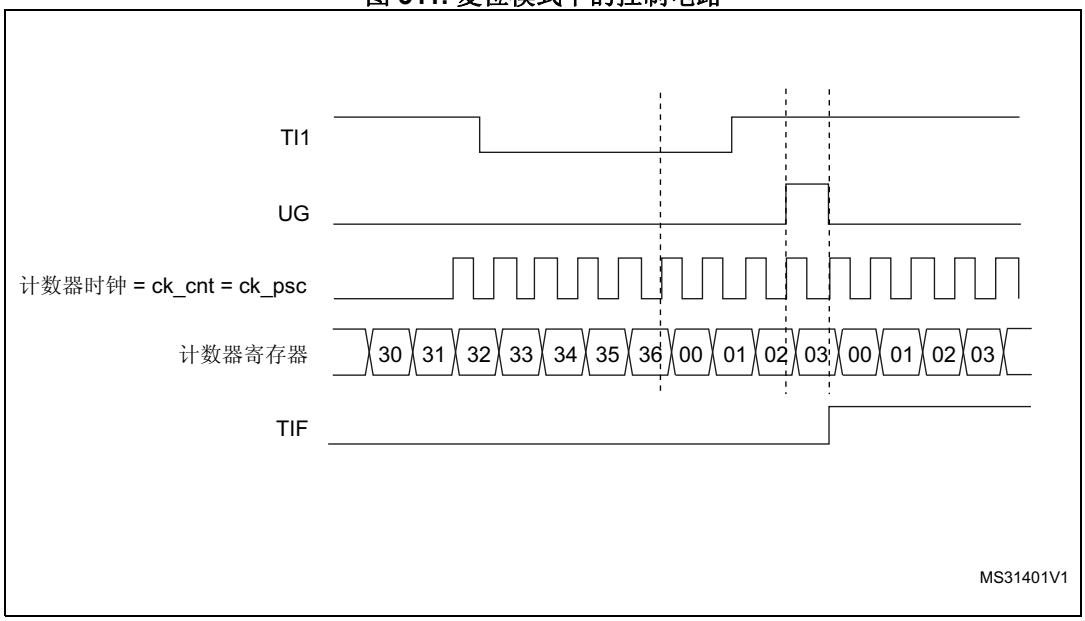
在以下示例中，TI1 输入上出现上升沿时，递增计数器清零：

- 将通道 1 配置为检测 TI1 的上升沿。配置输入滤波带宽（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 TIMx_CCMR1 寄存器中的 CC1S = 01。在 TIMx_CCER 寄存器中写入 CC1P=0 和 CC1NP=“0”，验证极性（仅检测上升沿）。
- 在 TIMx_SMCR 寄存器中写入 SMS=100，将定时器配置为复位模式。在 TIMx_SMCR 寄存器中写入 TS=101，选择 TI1 作为输入源。
- 在 TIMx_CR1 寄存器中写入 CEN=1，启动计数器。

计数器开始根据内部时钟计数，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（TIMx_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 TIMx_DIER 寄存器中的 TIE 和 TDE 位）。

下图显示了自动重载寄存器 TIMx_ARR=0x36 时的相关行为。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 311. 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。

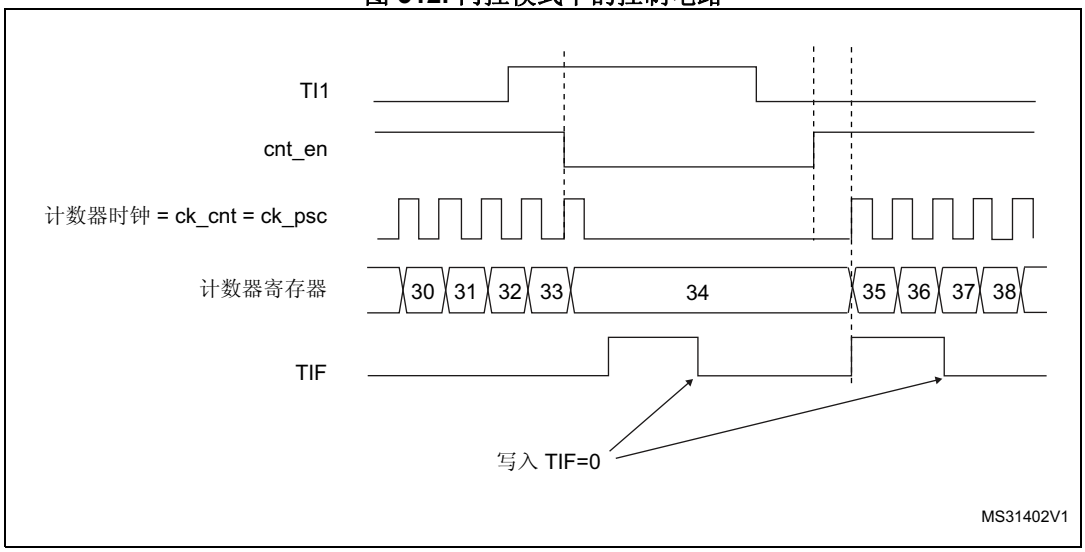
在以下示例中，递增计数器仅在 TI1 输入为低电平时计数：

- 将通道 1 配置为检测 TI1 上的低电平。配置输入滤波带宽（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 TIMx_CCMR1 寄存器中的 CC1S=01。在 TIMx_CCER 寄存器中写入 CC1P=1 和 CC1NP=“0”，以确定极性（仅检测低电平）。
- 在 TIMx_SMCR 寄存器中写入 SMS=101，将定时器配置为门控模式。在 TIMx_SMCR 寄存器中写入 TS=101，选择 TI1 作为输入源。
- 在 TIMx_CR1 寄存器中写入 CEN=1，使能计数器（在门控模式下，如果 CEN=0，则无论触发输入电平如何，计数器都不启动）。

只要 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，TIMx_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 312. 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。

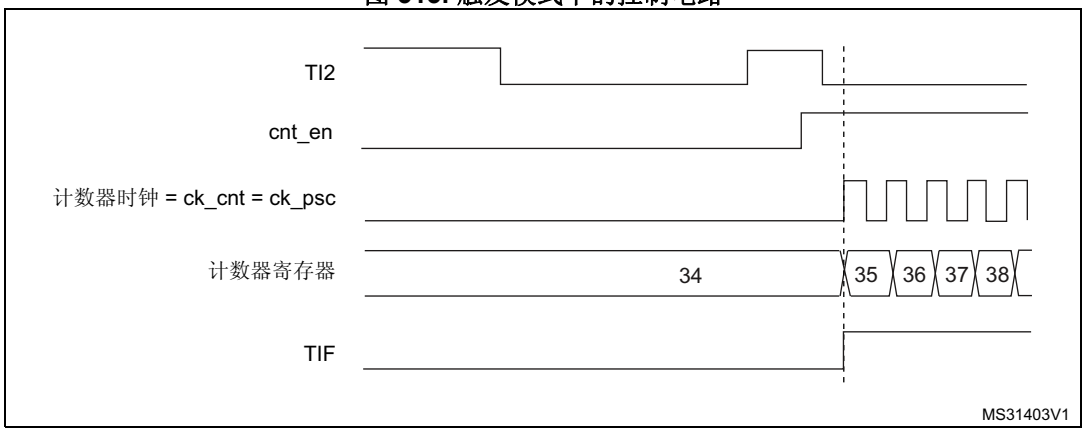
在以下示例中，TI2 输入上出现上升沿时，递增计数器启动：

- 将通道 2 配置为检测 TI2 上的上升沿。配置输入滤波带宽（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 TIMx_CCMR1 寄存器中的 CC2S=01。在 TIMx_CCER 寄存器中写入 CC2P=1 和 CC2NP=0，以确定极性（仅检测低电平）。
- 在 TIMx_SMCR 寄存器中写入 SMS=110，将定时器配置为触发模式。在 TIMx_SMCR 寄存器中写入 TS=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 313. 触发模式下的控制电路



从模式：组合复位 + 触发模式

在这种情况下，在出现所选触发输入 (TRGI) 上升沿时，重新初始化计数器，生成一个寄存器更新事件，并启动计数器。

该模式用于单脉冲模式。

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 TIMx_SMCR 寄存器中的 TS 位来选择 ETR 作为 TRGI。

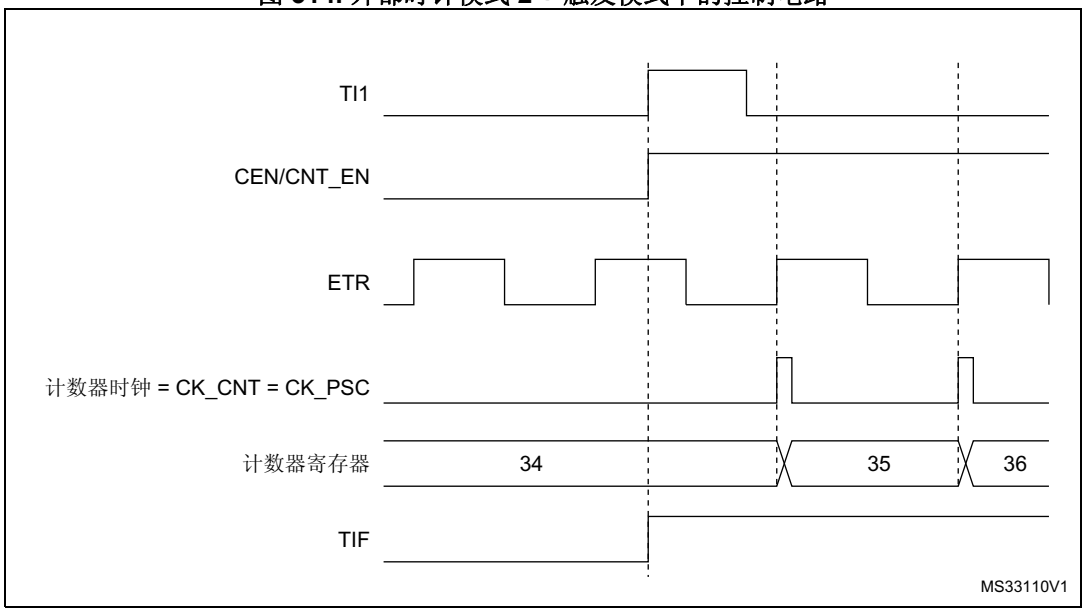
在以下示例中，只要 TI1 出现上升沿，递增计数器即会在 ETR 信号的每个上升沿处递增：

1. 通过对 TIMx_SMCR 寄存器进行如下编程，配置外部触发输入电路：
   - ETF=0000：无滤波器。
   - ETPS=00：禁止预分频器。
   - ETP=0：检测 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。
   - TIMx_CCMR1 寄存器中 CC1S=01，只选择输入捕获源。
   - TIMx_CCER 寄存器中 CC1P=0 且 CC1NP=“0”，以确定极性（仅检测上升沿）。
3. 在 TIMx_SMCR 寄存器中写入 SMS=110，将定时器配置为触发模式。在 TIMx_SMCR 寄存器中写入 TS=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 314. 外部时钟模式 2 + 触发模式下的控制电路



注： *必须先使能接收 TRGO 或 TRGO2 信号的从外设（定时器、ADC 等）的时钟，才能从主定时器接收事件；并且从主定时器接收触发信号时，不得实时更改时钟频率（预分频器）。*

34.3.27 ADC 同步

定时器可通过多种内部信号产生 ADC 触发事件，例如复位、使能或比较事件。也可生成由内部边沿检测器发出的脉冲，例如：

- OC4ref 的上升沿和下降沿
- OC5ref 上的上升沿或 OC6ref 上的下降沿

在重定向到 ADC 的 TRGO2 内部线路上发出触发信号。共有 16 个可能的事件，它们可通过 TIMx_CR2 寄存器中的 MMS2[3:0] 位选择。

第 1073 页的图 294 给出了三相电机驱动的应用示例。

注： *必须先使能接收 TRGO 或 TRGO2 信号的从外设（定时器、ADC 等）的时钟，才能从主定时器接收事件；并且从主定时器接收触发信号时，不得实时更改时钟频率（预分频器）。*

注： *必须先使能 ADC 时钟，才能从主定时器接收事件；从定时器接收触发信号时，不得实时更改 ADC 时钟。*

34.3.28 DMA 连续传送模式

TIMx 定时器能够根据一个事件生成多个 DMA 请求。主要目的是能够对定时器的一部分多次重新编程而无需软件开销，但也可用于定期读取一行中的多个寄存器。

DMA 控制器目标唯一，必须指向虚拟寄存器 TIMx_DMAR。发生给定的定时器事件时，定时器会启动 DMA 请求序列（突发）。每次写入 TIMx_DMAR 寄存器都会重定向到其中一个定时器寄存器。

TIMx_DCR 寄存器中的 DBL[4:0] 位设置 DMA 连续传送长度。当对 TIMx_DMAR 地址进行读或写访问时，定时器进行一次连续传送，即传送次数（按半字或字节）。

TIMx_DCR 寄存器中的 DBA[4:0] 位定义 DMA 传送的 DMA 基址（通过 TIMx_DMAR 地址执行读/写访问时）。DBA 定义为从 TIMx_CR1 寄存器地址开始计算的偏移量：

示例：

00000：TIMx_CR1

00001：TIMx_CR2

00010：TIMx_SMCR

例如，定时器 DMA 连续传送功能用于在发生更新事件后将 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 TIMx 更新 DMA 请求（DIER 寄存器中的 UDE 位置 1）。
4. 使能 TIMx。
5. 使能 DMA 通道。

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。

注： *可以将空值写入保留的寄存器中。*

34.3.29 调试模式

当微控制器进入调试模式（Cortex®-M4 内核停止）时，TIMx 计数器会根据 DBG 模块中的 DBG_TIMx_STOP 配置位选择继续正常工作或者停止工作。

为了安全起见，当计数器停止 (DBG_TIMx_STOP = 1) 时，输出被禁止（就像 MOE 位被复位一样）。可以将输出强制变为未激活状态（OSSI 位 = 1），或者通过 GPIO 控制器（OSSI 位 = 0）来控制输出，通常将其强制为高阻态。

有关详细信息，请参见*第 51.16.2 节：对定时器、RTC、看门狗、bxCAN 和 I2C 的调试支持*。

34.4 TIM1/TIM8 寄存器

有关寄存器说明中使用的缩写，请参见相应列表。

34.4.1 TIM1/TIM8 控制寄存器 1 (TIMx_CR1)

TIM1/TIM8 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
Res.	Res.	Res.	Res.	UIFRE MAP	Res.	CKD[1:0]		ARPE	CMS[1:0]		DIR	OPM	URS	UDIS	CEN
				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 **UIFREMAP**：UIF 状态位重映射 (UIF status bit remapping)
0：无重映射。UIF 状态位不复制到 TIMx_CNT 寄存器的位 31。
1：使能重映射。UIF 状态位复制到 TIMx_CNT 寄存器的位 31。

位 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 (t_{DTS}) 之间的分频比，
00：$t_{DTS}=t_{CK_INT}$
01：$t_{DTS}=2*t_{CK_INT}$
10：$t_{DTS}=4*t_{CK_INT}$
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：TIMx_ARR 寄存器不进行缓冲
1：TIMx_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（TIMx_CCMRx 寄存器中的 CCxS=00）的输出比较中断标志才置 1。

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

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

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

位 4 **DIR**：方向 (Direction)

0：计数器递增计数

1：计数器递减计数

注： 当定时器配置为中心对齐模式或编码器模式时，该位为只读状态。

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，所有以下事件都会产生更新中断或 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： 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。

34.4.2 TIM1/TIM8 控制寄存器 2 (TIMx_CR2)

TIM1/TIM8 control register 2

偏移地址：0x04

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MMS2[3:0]				Res.	OIS6	Res.	OIS5
								rw	rw	rw	rw		rw		rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	OIS4	OIS3N	OIS3	OIS2N	OIS2	OIS1N	OIS1	TI1S	MMS[2:0]			CCDS	CCUS	Res.	CCPC
	rw	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 **MMS2[3:0]**：主模式选择 2 (Master mode selection 2)

这些位可选择将发送到 ADC 以实现同步的信息 (TRGO2)。这些位的组合如下：

0000：**复位**——TIMx_EGR 寄存器中的 UG 位用作触发输出 (TRGO2)。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 TRGO2 上的信号相比实际复位会有延迟。

0001：**使能**——计数器使能信号 CNT_EN 用作触发输出 (TRGO2)。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，TRGO2 上会存在延迟，选择主/从模式时除外（请参见 TIMx_SMCR 寄存器中 MSM 位的说明）。

0010：**更新**——选择更新事件作为触发输出 (TRGO2)。例如，主定时器可用作从定时器的预分频器。

0011：**比较脉冲**——CC1IF 标志置 1 时（即使已为高），只要发生捕获或比较匹配，触发输出 (TRGO2) 都会发送一个正脉冲。

0100：**比较**——OC1REF 信号用作触发输出 (TRGO2)
0101：**比较**——OC2REF 信号用作触发输出 (TRGO2)
0110：**比较**——OC3REF 信号用作触发输出 (TRGO2)
0111：**比较**——OC4REF 信号用作触发输出 (TRGO2)
1000：**比较**——OC5REF 信号用作触发输出 (TRGO2)
1001：**比较**——OC6REF 信号用作触发输出 (TRGO2)
1010：**比较脉冲**——OC4REF 上升沿或下降沿时，TRGO2 上生成脉冲
1011：**比较脉冲**——OC6REF 上升沿或下降沿时，TRGO2 上生成脉冲
1100：**比较脉冲**——OC4REF 或 OC6REF 上升沿时，TRGO2 上生成脉冲
1101：**比较脉冲**——OC4REF 上升沿或 OC6REF 下降沿时，TRGO2 上生成脉冲
1110：**比较脉冲**——OC5REF 或 OC6REF 上升沿时，TRGO2 上生成脉冲
1111：**比较脉冲**——OC5REF 上升沿或 OC6REF 下降沿时，TRGO2 上生成脉冲

注：　必须先使能从定时器或 ADC 的时钟，才能从主定时器接收事件；并且从主定时器接收触发信号时，不得实时更改从定时器或 ADC 的时钟。

位 19 保留，必须保持复位值。

位 18 **OIS6**：输出空闲状态 6（OC6 输出）(Output Idle state 6 (OC6 output))
请参见 OIS1 位

位 17 保留，必须保持复位值。

位 16 **OIS5**：输出空闲状态 5（OC5 输出）(Output Idle state 5 (OC5 output))
请参见 OIS1 位

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))
请参见 OIS1 位

位 13 **OIS3N**：输出空闲状态 3（OC3N 输出）(Output Idle state 3 (OC3N output))
请参见 OIS1N 位

位 12 **OIS3**：输出空闲状态 3（OC3 输出）(Output Idle state 3 (OC3 output))
请参见 OIS1 位

位 11 **OIS2N**：输出空闲状态 2（OC2N 输出）(Output Idle state 2 (OC2N output))
请参见 OIS1N 位

位 10 **OIS2**：输出空闲状态 2（OC2 输出）(Output Idle state 2 (OC2 output))
请参见 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：　只要编程了 LOCK（TIMx_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：　只要编程了 LOCK（TIMx_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：TIMx_CH1 引脚连接到 TI1 输入
1：TIMx_CH1、CH2 和 CH3 引脚连接到 TI1 输入（异或组合）

位 6:4 **MMS[2:0]**：主模式选择 (Master mode selection)
这些位可选择主模式下将要发送到从定时器以实现同步的信息 (TRGO)。这些位的组合如下：
000：**复位**——TIMx_EGR 寄存器中的 UG 位用作触发输出 (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 TIMx_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 标志被置 1 时（即使已为高），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)
注：　必须先使能从定时器或 ADC 的时钟，才能从主定时器接收事件；并且从主定时器接收触发信号时，不得实时更改从定时器或 ADC 的时钟。

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/比较控制位进行预装载 (CCPC=1)，仅通过将 COMG 位置 1 来对这些位进行更新
1：如果捕获/比较控制位进行预装载 (CCPC=1)，可通过将 COMG 位置 1 或 TRGI 的上升沿对这些位进行更新。
注： 此位仅对具有互补输出的通道有效。

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： 此位仅对具有互补输出的通道有效。

34.4.3 TIM1/TIM8 从模式控制寄存器 (TIMx_SMCR)

TIM1/TIM8 slave mode control 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
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SMS[3]
															rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ETP	ECE	ETPS[1:0]		ETF[3:0]				MSM	TS[2:0]			Res.	SMS[2:0]		
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw		rw	rw	rw

位 31:17 保留，必须保持复位值。

位 16 **SMS[3]**：从模式选择——位 3 (Slave mode selection - bit 3)
请参见 SMS 说明——位 2:0

位 15 **ETP**：外部触发极性 (External trigger polarity)
此位可选择将 ETR 还是 $\overline{ETR}$ 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： **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 TIMxCLK 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 的数字滤波器带宽。数字滤波器由事件计数器组成，每 N 个连续事件才视为一个有效输出边沿：

0000：无滤波器，按 f_{DTS} 频率进行采样
0001：$f_{SAMPLING}=f_{CK_INT}$，N=2
0010：$f_{SAMPLING}=f_{CK_INT}$，N=4
0011：$f_{SAMPLING}=f_{CK_INT}$，N=8
0100：$f_{SAMPLING}=f_{DTS}/2$，N=6
0101：$f_{SAMPLING}=f_{DTS}/2$，N=8
0110：$f_{SAMPLING}=f_{DTS}/4$，N=6
0111：$f_{SAMPLING}=f_{DTS}/4$，N=8
1000：$f_{SAMPLING}=f_{DTS}/8$，N=6
1001：$f_{SAMPLING}=f_{DTS}/8$，N=8
1010：$f_{SAMPLING}=f_{DTS}/16$，N=5
1011：$f_{SAMPLING}=f_{DTS}/16$，N=6
1100：$f_{SAMPLING}=f_{DTS}/16$，N=8
1101：$f_{SAMPLING}=f_{DTS}/32$，N=5
1110：$f_{SAMPLING}=f_{DTS}/32$，N=6
1111：$f_{SAMPLING}=f_{DTS}/32$，N=8

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*第 1103 页的表 215：TIMx 内部触发连接*。

注：　这些位只能在未使用的情况下（例如，SMS=000 时）进行更改，以避免转换时出现错误的边沿检测。

注：　其他位位于同一寄存器的位置 16

位 3 保留，必须保持复位值。

位 16、2、1、0 **SMS[3:0]**：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）。

0000：禁止从模式——如果 CEN=“1”，预分频器时钟直接由内部时钟提供。
0001：编码器模式 1——计数器根据 TI2FP2 电平在 TI1FP1 边沿递增/递减计数。
0010：编码器模式 2——计数器根据 TI1FP1 电平在 TI2FP2 边沿递增/递减计数。
0011：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输入的电平。
0100：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。
0101：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被控制。
0110：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。
0111：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。
1000：组合复位 + 触发模式——在出现所选触发输入 (TRGI) 上升沿时，重新初始化计数器，生成一个寄存器更新事件并启动计数器。
1000 以上的代码：保留。

注：　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。

注：　必须先使能接收 TRGO 或 TRGO2 信号的从外设（定时器、ADC 等）的时钟，才能从主定时器接收事件；并且从主定时器接收触发信号时，不得实时更改时钟频率（预分频器）。

表 215. TIMx 内部触发连接

从 TIM	ITR0 (TS = 000)	ITR1 (TS = 001)	ITR2 (TS = 010)	ITR3 (TS = 011)
TIM1	TIM15	TIM2	TIM3	TIM4
TIM8	TIM1	TIM2	TIM4	TIM5

34.4.4 TIM1/TIM8 DMA/中断使能寄存器 (TIMx_DIER)

TIM1/TIM8 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：禁止触发 DMA 请求
1：使能触发 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 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 3 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 2 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 **BIE**：断路中断使能 (Break interrupt enable)
0：禁止断路中断
1：使能断路中断

位 6 **TIE**：触发中断使能 (Trigger interrupt enable)
0：禁止触发中断
1：使能触发中断

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

34.4.5 TIM1/TIM8 状态寄存器 (TIMx_SR)

TIM1/TIM8 status 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
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CC6IF	CC5IF
														rc_w0	rc_w0
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	SBIF	CC4OF	CC3OF	CC2OF	CC1OF	B2IF	BIF	TIF	COMIF	CC4IF	CC3IF	CC2IF	CC1IF	UIF
		rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0

位 31:18 保留，必须保持复位值。

位 17 **CC6IF**：比较 6 中断标志 (Compare 6 interrupt flag)
请参见 CC1IF 说明（注意：通道 6 只能配置为输出）

位 16 **CC5IF**：比较 5 中断标志 (Compare 5 interrupt flag)
请参见 CC1IF 说明（注意：通道 5 只能配置为输出）

位 15:14 保留，必须保持复位值。

位 13 **SBIF**：系统断路中断标志 (System Break interrupt flag)
只要系统断路输入变为有效状态，此标志便由硬件置 1。系统断路输入无效后可通过软件对其清零。
此标志必须复位以使 PWM 重新开始工作。
0：未发生断路事件。
1：在系统断路输入上检测到有效电平。如果 TIMx_DIER 寄存器中 BIE=1，则会生成中断。

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：TIMx_CCR1 寄存器中已捕获到计数器值且 CC1IF 标志已置 1。

位 8 **B2IF**：断路 2 中断标志 (Break 2 interrupt flag)
只要断路 2 输入变为有效状态，此标志便由硬件置 1。断路 2 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 2 输入上检测到有效电平。如果 TIMx_DIER 寄存器中 BIE=1，则会生成中断。

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。如果 TIMx_DIER 寄存器中 BIE=1，则会生成中断。

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 TIMx_CR1 寄存器中的 CMS 位说明）。但需要通过软件清零。

0：不匹配。

1：TIMx_CNT 计数器的值与 TIMx_CCR1 寄存器的值匹配。当 TIMx_CCR1 的值大于 TIMx_ARR 的值时，CC1IF 位将在计数器发生上溢（递增计数模式和增减计数模式下）或下溢（递减计数模式下）时变为高。

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 TIMx_CCR1 寄存器将该位清零。

0：未发生输入捕获事件

1：TIMx_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：

– TIMx_CR1 寄存器中的 UDIS=0，并且重复计数器值上溢或下溢时（重复计数器 = 0 时更新）。

– TIMx_CR1 寄存器中的 URS=0 且 UDIS=0，并且由软件使用 TIMx_EGR 寄存器中的 UG 位重新初始化 CNT 时。

– TIMx_CR1 寄存器中的 URS=0 且 UDIS=0，并且 CNT 由触发事件重新初始化时（请参见 *第 34.4.3 节：TIM1/TIM8 从模式控制寄存器 (TIMx_SMCR)*）。

34.4.6 TIM1/TIM8 事件生成寄存器 (TIMx_EGR)

TIM1/TIM8 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
Res.	Res.	Res.	Res.	Res.	Res.	Res.	B2G	BG	TG	COMG	CC4G	CC3G	CC2G	CC1G	UG
							w	w	w	w	w	w	w	w	w

位 15:9 保留，必须保持复位值。

位 8 **B2G**：断路 2 生成 (Break 2 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 2 事件。MOE 位清零且 B2IF 标志置 1。使能后可发生相关中断。

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**：触发生成 (Trigger generation)
此位由软件置 1 以生成事件，并由硬件自动清零。
0：不执行任何操作。
1：TIMx_SR 寄存器中的 TIF 标志置 1。使能后可发生相关中断或 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： 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：
TIMx_CCR1 寄存器中将捕获到计数器当前值。使能时，CC1IF 标志置 1 并发送相应的中断或 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 (TIMx_ARR)。

34.4.7 TIM1/TIM8 捕获/比较模式寄存器 1 (TIMx_CCMR1)

TIM1/TIM8 capture/compare mode register 1

偏移地址：0x18

复位值：0x0000 0000

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。此寄存器的所有其他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。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC2M[3]	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC1M[3]
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
							rw								rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OC2 CE	OC2M[2:0]			OC2 PE	OC2 FE	CC2S[1:0]		OC1 CE	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 31:25 保留，必须保持复位值。

位 24 **OC2M[3]**：输出比较 2 模式——位 3 (Output Compare 2 mode - bit 3)

请参见 OC2M 说明——位 14:12

位 23:17 保留，必须保持复位值。

位 16 **OC1M[3]**：输出比较 1 模式——位 3 (Output Compare 1 mode - bit 3)

请参见 OC1M 说明——位 6:4

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（TIMx_SMCR 寄存器）选择内部触发输入时有效

注： 仅当通道关闭时（TIMx_CCER 中的 CC2E = "0"），才可向 CC2S 位写入数据。

位 7 **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。

0000：冻结——输出比较寄存器 TIMx_CCR1 与计数器 TIMx_CNT 进行比较不会对输出造成任何影响。（该模式用于生成时基）。

0001：将通道 1 设置为匹配时输出有效电平。当计数器 TIMx_CNT 与捕获/比较寄存器 1 (TIMx_CCR1) 匹配时，OC1REF 信号强制变为高电平。

0010：将通道 1 设置为匹配时输出无效电平。当计数器 TIMx_CNT 与捕获/比较寄存器 1 (TIMx_CCR1) 匹配时，OC1REF 信号强制变为低电平。

0011：翻转——TIMx_CNT=TIMx_CCR1 时，OC1REF 发生翻转。

0100：强制变为无效电平——OC1REF 强制变为低电平。

0101：强制变为有效电平——OC1REF 强制变为高电平。

0110：PWM 模式 1——在递增计数模式下，只要 TIMx_CNT<TIMx_CCR1，通道 1 便为有效状态，否则为无效状态。在递减计数模式下，只要 TIMx_CNT>TIMx_CCR1，通道 1 便为无效状态（OC1REF="0"），否则为有效状态（OC1REF="1"）。

0111：PWM 模式 2——在递增计数模式下，只要 TIMx_CNT<TIMx_CCR1，通道 1 便为无效状态，否则为有效状态。在递减计数模式下，只要 TIMx_CNT>TIMx_CCR1，通道 1 便为有效状态，否则为无效状态。

1000：可再触发 OPM 模式 1——在递增计数模式下，通道为有效状态，直至（在 TRGI 信号上）检测到触发事件。然后，在 PWM 模式 1 下进行比较，通道会在下一次更新时再次变为有效状态。在递减计数模式下，通道为无效状态，直至（在 TRGI 信号上）检测到触发事件。然后，在 PWM 模式 1 下进行比较，通道会在下一次更新时再次变为无效状态。

1001：可再触发 OPM 模式 2——在递增计数模式下，通道为无效状态，直至（在 TRGI 信号上）检测到触发事件。然后，在 PWM 模式 2 下进行比较，通道会在下一次更新时再次变为无效状态。在递减计数模式下，通道为有效状态，直至（在 TRGI 信号上）检测到触发事件。然后，在 PWM 模式 2 下进行比较，通道会在下一次更新时再次变为有效状态。

1010：保留。

1011：保留。

1100：组合 PWM 模式 1——OC1REF 与在 PWM 模式 1 下的行为相同。OC1REFC 是 OC1REF 和 OC2REF 的逻辑或运算结果。

1101：组合 PWM 模式 2——OC1REF 与在 PWM 模式 2 下的行为相同。OC1REFC 是 OC1REF 和 OC2REF 的逻辑与运算结果。

1110：不对称 PWM 模式 1——OC1REF 与在 PWM 模式 1 下的行为相同。计数器递增计数时，OC1REFC 输出 OC1REF；计数器递减计数时，OC1REFC 输出 OC2REF。

1111：不对称 PWM 模式 2——OC1REF 与在 PWM 模式 2 下的行为相同。计数器递增计数时，OC1REFC 输出 OC1REF；计数器递减计数时，OC1REFC 输出 OC2REF。

注： 只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 3 且 CC1S="00"（通道配置为输出），这些位即无法修改。

注： 在 PWM 模式下，仅当比较结果发生改变或输出比较模式由"冻结"模式切换到"PWM"模式时， OCREF 电平才会发生更改。

注： 此位域将在具有互补输出的通道上进行预装载。如果 TIMx_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 TIMx_CCR1 相关的预装载寄存器。可随时向 TIMx_CCR1 写入数据，写入后将立即使用新值。

1：使能与 TIMx_CCR1 相关的预装载寄存器。可读/写访问预装载寄存器。TIMx_CCR1 预装载值在每次生成更新事件时都会装载到有效寄存器中。

*注： **1**：只要编程了 LOCK（TIMx_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（TIMx_CR1 寄存器中的 OPM 位置 1）。其他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 输出的最短延迟时间为 5 个时钟周期。

1：触发输入上出现有效边沿相当于 CC1 输出上的比较匹配。随后，无论比较结果如何，OC 都设置为比较电平。采样触发输入和激活 CC1 输出的延迟时间缩短为 3 个时钟周期。仅当通道配置为 PWM1 或 PWM2 模式时，OCFE 才会起作用。

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（TIMx_SMCR 寄存器）选择内部触发输入时有效

注： 仅当通道关闭时（TIMx_CCER 中的 CC1E = “0”），才可向 CC1S 位写入数据。

输入捕获模式

位 31:16 保留，必须保持复位值。

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（TIMx_SMCR 寄存器）选择内部触发输入时有效

注： 仅当通道关闭时（TIMx_CCER 中的 CC2E = “0”），才可向 CC2S 位写入数据。

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 的数字滤波器的采样长度。数字滤波器由事件计数器组成，每 N 个连续事件才视为一个有效输出边沿：

0000：无滤波器，按 f_{DTS} 频率进行采样
0001：$f_{SAMPLING}=f_{CK_INT}$，N=2
0010：$f_{SAMPLING}=f_{CK_INT}$，N=4
0011：$f_{SAMPLING}=f_{CK_INT}$，N=8
0100：$f_{SAMPLING}=f_{DTS}/2$，N=6
0101：$f_{SAMPLING}=f_{DTS}/2$，N=8
0110：$f_{SAMPLING}=f_{DTS}/4$，N=6
0111：$f_{SAMPLING}=f_{DTS}/4$，N=8
1000：$f_{SAMPLING}=f_{DTS}/8$，N=6
1001：$f_{SAMPLING}=f_{DTS}/8$，N=8
1010：$f_{SAMPLING}=f_{DTS}/16$，N=5
1011：$f_{SAMPLING}=f_{DTS}/16$，N=6
1100：$f_{SAMPLING}=f_{DTS}/16$，N=8
1101：$f_{SAMPLING}=f_{DTS}/32$，N=5
1110：$f_{SAMPLING}=f_{DTS}/32$，N=6
1111：$f_{SAMPLING}=f_{DTS}/32$，N=8

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”（TIMx_CCER 寄存器），预分频器便立即复位。

00：无预分频器，捕获输入上每检测到一个边沿便执行捕获
01：每发生 2 个事件便执行一次捕获
10：每发生 4 个事件便执行一次捕获
11：每发生 8 个事件便执行一次捕获

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（TIMx_SMCR 寄存器）选择内部触发输入时有效

注： 仅当通道关闭时（TIMx_CCER 中的 CC1E = “0”），才可向 CC1S 位写入数据。

34.4.8 TIM1/TIM8 捕获/比较模式寄存器 2 (TIMx_CCMR2)

TIM1/TIM8 capture/compare mode register 2

偏移地址：0x1C

复位值：0x0000 0000

请参见上述 CCMR1 寄存器说明。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC4M[3]	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC3M[3]
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
							rw								rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OC4 CE	OC4M[2:0]			OC4 PE	OC4 FE	CC4S[1:0]		OC3 CE.	OC3M[2:0]			OC3 PE	OC3 FE	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 31:25 保留，必须保持复位值。

位 24 **OC4M[3]**：输出比较 4 模式——位 3 (Output Compare 4 mode - bit 3)

位 23:17 保留，必须保持复位值。

位 16 **OC3M[3]**：输出比较 3 模式——位 3 (Output Compare 3 mode - bit 3)

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（TIMx_SMCR 寄存器）选择内部触发输入时有效

注： 仅当通道关闭时（TIMx_CCER 中的 CC4E = “0”），才可向 CC4S 位写入数据。

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（TIMx_SMCR 寄存器）选择内部触发输入时有效

注： 仅当通道关闭时（TIMx_CCER 中的 CC3E = “0”），才可向 CC3S 位写入数据。

输入捕获模式

位 31:16 保留，必须保持复位值。

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（TIMx_SMCR 寄存器）选择内部触发输入时有效

注： 仅当通道关闭时（TIMx_CCER 中的 CC4E = “0”），才可向 CC4S 位写入数据。

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（TIMx_SMCR 寄存器）选择内部触发输入时有效

注： 仅当通道关闭时（TIMx_CCER 中的 CC3E = “0”），才可向 CC3S 位写入数据。

34.4.9 TIM1/TIM8 捕获/比较使能寄存器 (TIMx_CCER)

TIM1/TIM8 capture/compare enable 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
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CC6P	CC6E	Res.	Res.	CC5P	CC5E
										rw	rw			rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CC4NP	Res.	CC4P	CC4E	CC3NP	CC3NE	CC3P	CC3E	CC2NP	CC2NE	CC2P	CC2E	CC1NP	CC1NE	CC1P	CC1E
rw		rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:22 保留，必须保持复位值。

位 21 **CC6P**：捕获/比较 6 输出极性 (Capture/Compare 6 output polarity)

请参见 CC1P 说明

位 20 **CC6E**：捕获/比较 6 输出使能 (Capture/Compare 6 output enable)

请参见 CC1E 说明

位 19:18 保留，必须保持复位值。

位 17 **CC5P**：捕获/比较 5 输出极性 (Capture/Compare 5 output polarity)

请参见 CC1P 说明

位 16 **CC5E**：捕获/比较 5 输出使能 (Capture/Compare 5 output enable)

请参见 CC1E 说明

位 15 **CC4NP**：捕获/比较 4 互补输出极性 (Capture/Compare 4 complementary 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 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 CC1P 说明。
注： 只要编程了 LOCK（TIMx_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。
此位将在具有互补输出的通道上进行预装载。如果 TIMx_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。

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。
此位将在具有互补输出的通道上进行预装载。如果 TIMx_CR2 寄存器中的 CCPC 位置 1，则仅当生成换向事件时，CC1NE 有效位才会从预装载位获取新值。

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：非反相/上升沿和下降沿均触发。电路对 TIxFP1 上升沿和下降沿都敏感（在复位模式、外部时钟模式或触发模式下执行捕获或触发操作），TIxFP1 未反相（在门控模式下执行触发操作）。编码器模式下不得使用此配置。

注：　只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 2 或 3，此位立即变为不可写状态。

此位将在具有互补输出的通道上进行预装载。如果 TIMx_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。

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 信号输出到相应的输出引脚上，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 (TIMx_CCR1) 中。

0：禁止捕获。

1：使能捕获。

此位将在具有互补输出的通道上进行预装载。如果 TIMx_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 216. 具有断路功能的互补通道 OCx 和 OCxN 的输出控制位

控制位					输出状态(1)	
MOE 位	OSSI 位	OSSR 位	CCxE 位	CCxNE 位	OCx 输出状态	OCxN 输出状态
1	X	X	0	0	禁止输出（不由定时器驱动：高阻态） OCx=0、OCxN=0	
		0	0	1	禁止输出（不由定时器驱动：高阻态） OCx=0	OCxREF + 极性 OCxN = OCxREF 异或 CCxNP
		0	1	0	OCxREF + 极性 OCx=OCxREF 异或 CCxP	禁止输出 （不由定时器驱动：高阻态） OCxN=0
		X	1	1	OCREF + 极性 + 死区	OCREF 互补项（对 OCREF 进行“非”运算）+ 极性 + 死区
		1	0	1	关闭状态 （输出使能为无效状态） OCx=CCxP	OCxREF + 极性 OCxN = OCxREF 异或 CCxNP
		1	1	0	OCxREF + 极性 OCx=OCxREF 异或 CCxP	关闭状态 （输出使能为无效状态） OCxN=CCxNP
0	0	X	X	X	禁止输出（不再由定时器驱动）。输出状态由 GPIO 控制器定义，可以是高电平、低电平或高阻态。	
	1		0	0		
			0	1	关闭状态（输出使能为无效状态） 异步：OCx=CCxP、OCxN=CCxNP（如果触发 BRK 或 BRK2）。 随后（仅当触发 BRK 时才有效），如果存在时钟：在死区后 OCx=OISx 且 OCxN=OISxN，假定 OISx 和 OISxN 并没有都设置成 OCx 及 OCxN 的有效电平（否则在半桥配置下驱动开关时可能导致短路）。 **注意**：BRK2 只能在 OSSI = OSSR = 1 时使用。	
			1	0		
			1	1		

1. 如果一个通道的两个输出均未使用（由 GPIO 接管控制），则 OISx、OISxN、CCxP 和 CCxNP 位必须保持清零状态。

注： *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。*

34.4.10 TIM1/TIM8 计数器 (TIMx_CNT)

TIM1/TIM8 counter

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

位 31 **UIFCPY**：UIF 副本 (UIF copy)

该位是 TIMx_ISR 寄存器中 UIF 位的只读副本。如果 TIMxCR1 中的 UIFREMAP 位复位，则位 31 保留，读为 0。

位 30:16 保留，必须保持复位值。

位 15:0 **CNT[15:0]**：计数器值 (Counter value)

34.4.11 TIM1/TIM8 预分频器 (TIMx_PSC)

TIM1/TIM8 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 f_{CK_PSC} / (PSC[15:0] + 1)。

PSC 包含每次发生更新事件（包括计数器通过 TIMx_EGR 寄存器中的 UG 位清零时，或在配置为“复位模式”时通过触发控制器清零时）时要装载到有效预分频器寄存器的值。

34.4.12 TIM1/TIM8 自动重载寄存器 (TIMx_ARR)

TIM1/TIM8 auto-reload register

偏移地址：0x2C

复位值：0xFFFF

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*第 1042 页的第 34.3.1 节：时基单元*。

当自动重载值为空时，计数器不工作。

34.4.13 TIM1/TIM8 重复计数器寄存器 (TIMx_RCR)

TIM1/TIM8 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
REP[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 15:0 **REP[15:0]**：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，从预装载寄存器向有效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 TIMx_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。

34.4.14 TIM1/TIM8 捕获/比较寄存器 1 (TIMx_CCR1)

TIM1/TIM8 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 为要装载到有效捕获/比较 1 寄存器的值（预装载值）。如果没有通过 TIMx_CCMR1 寄存器中的 OC1PE 位来使能预装载功能，则该值立刻生效；否则只在发生更新事件时生效（拷贝到有效的捕获/比较寄存器 1）。
有效捕获/比较寄存器中包含要与计数器 TIMx_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：CR1 为上一个输入捕获 1 事件 (IC1) 发生时的计数器值。只能读取 TIMx_CCR1 寄存器，无法对其进行编程。

34.4.15 TIM1/TIM8 捕获/比较寄存器 2 (TIMx_CCR2)

TIM1/TIM8 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 为要装载到有效捕获/比较 2 寄存器的值（预装载值）。如果没有通过 TIMx_CCMR1 寄存器中的 OC2PE 位来使能预装载功能，则该值立刻生效；否则只在发生更新事件时生效（拷贝到有效的捕获/比较寄存器 2）。
有效捕获/比较寄存器中包含要与计数器 TIMx_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。只能读取 TIMx_CCR2 寄存器，无法对其进行编程。

34.4.16 TIM1/TIM8 捕获/比较寄存器 3 (TIMx_CCR3)

TIM1/TIM8 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 为要装载到有效捕获/比较 3 寄存器的值（预装载值）。
如果没有通过 TIMx_CCMR2 寄存器中的 OC3PE 位来使能预装载功能，则该值立刻生效；否则只在发生更新事件时生效（拷贝到有效的捕获/比较寄存器 3）。
有效捕获/比较寄存器中包含要与计数器 TIMx_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。只能读取 TIMx_CCR3 寄存器，无法对其进行编程。

34.4.17 TIM1/TIM8 捕获/比较寄存器 4 (TIMx_CCR4)

TIM1/TIM8 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 为要装载到有效捕获/比较 4 寄存器的值（预装载值）。
如果没有通过 TIMx_CCMR2 寄存器中的 OC4PE 位来使能预装载功能，则该值立刻生效；否则只在发生更新事件时生效（拷贝到有效的捕获/比较寄存器 4）。
有效捕获/比较寄存器中包含要与计数器 TIMx_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。只能读取 TIMx_CCR4 寄存器，无法对其进行编程。

34.4.18 TIM1/TIM8 断路和死区寄存器 (TIMx_BDTR)

TIM1/TIM8 break and dead-time register

偏移地址：0x44

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	BK2BID	BKBID	BK2 DSRM	BK DSRM	BK2P	BK2E	BK2F[3:0]				BKF[3:0]			
		rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MOE	AOE	BKP	BKE	OSSR	OSSI	LOCK[1:0]		DTG[7:0]							
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

注： *由于可以根据 LOCK 配置锁定位 BK2BID、BKBID、BK2DSRM、BKDSRM、BK2P、BK2E、BK2F[3:0]、BKF[3:0]、AOE、BKP、BKE、OSSI、OSSR 和 DTG[7:0] 的写操作，因此必须在第一次对 TIMx_BDTR 寄存器执行写访问时对这些位进行配置。*

位 31:30 保留，必须保持复位值。

位 29 **BK2BID**：断路 2 双向 (Break2 bidirectional)

请参见 BKBID 说明

位 28 **BKBID**：断路双向 (Break Bidirectional)

0：断路输入 BRK 为输入模式

1：断路输入 BRK 为双向模式

在双向模式下（BKBID 位置 1），断路输入配置为输入模式和开漏输出模式。任何激活的断路事件都将使断路输入上呈逻辑低电平，以向外部器件指示发生了内部断路事件。

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

注： *对该位执行任何写操作后，都需要经过 1 个 APB 时钟周期的延迟才生效。*

位 27 **BK2DSRM**：断路 2 解除 (Break2 Disarm)

请参见 BKDSRM 说明

位 26 **BKDSRM**：断路解除 (Break Disarm)

0：启动断路输入 BRK

1：解除断路输入 BRK

当没有断路源被激活时，此位由硬件清零。

必须通过软件将 BKDSRM 位置 1 以释放双向输出控制（开漏输出处于高阻态），然后不断轮询该位，直到其由硬件复位，指示故障条件已消失。

注： *对该位执行任何写操作后，都需要经过 1 个 APB 时钟周期的延迟才生效。*

位 25 **BK2P**：断路 2 极性 (Break 2 polarity)

0：断路输入 BRK2 为低电平有效

1：断路输入 BRK2 为高电平有效

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

注： *对该位执行任何写操作后，都需要经过 1 个 APB 时钟周期的延迟才生效。*

位 24 **BK2E**：断路 2 使能 (Break 2 enable)

该位可使能完整的断路 2 保护（包括连接到 bk_acth 的所有源和相应的 BKIN 源，如*图 298：断路和断路 2 电路概述*部分所示）。

0：禁止断路 2 功能

1：使能断路 2 功能

注： BRKIN2 必须只在 OSSR = OSSI = 1 时使用。

注： 编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1 后，此位即无法修改。

注： 对该位执行任何写操作后，都需要经过 1 个 APB 时钟周期的延迟才生效。

位 23:20 **BK2F[3:0]**：断路 2 滤波器 (Break 2 filter)

此位域可定义 BRK2 输入的采样频率和适用于 BRK2 的数字滤波器带宽。数字滤波器由事件计数器组成，每 N 个连续事件才视为一个有效输出边沿：

0000：无滤波器，BRK2 异步工作

0001：$f_{SAMPLING}=f_{CK_INT}$，N=2

0010：$f_{SAMPLING}=f_{CK_INT}$，N=4

0011：$f_{SAMPLING}=f_{CK_INT}$，N=8

0100：$f_{SAMPLING}=f_{DTS}/2$，N=6

0101：$f_{SAMPLING}=f_{DTS}/2$，N=8

0110：$f_{SAMPLING}=f_{DTS}/4$，N=6

0111：$f_{SAMPLING}=f_{DTS}/4$，N=8

1000：$f_{SAMPLING}=f_{DTS}/8$，N=6

1001：$f_{SAMPLING}=f_{DTS}/8$，N=8

1010：$f_{SAMPLING}=f_{DTS}/16$，N=5

1011：$f_{SAMPLING}=f_{DTS}/16$，N=6

1100：$f_{SAMPLING}=f_{DTS}/16$，N=8

1101：$f_{SAMPLING}=f_{DTS}/32$，N=5

1110：$f_{SAMPLING}=f_{DTS}/32$，N=6

1111：$f_{SAMPLING}=f_{DTS}/32$，N=8

注： 编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1 后，此位即无法修改。

位 19:16 **BKF[3:0]**：断路滤波器 (Break filter)

此位域可定义 BRK 输入的采样频率和适用于 BRK 的数字滤波器带宽。数字滤波器由事件计数器组成，每 N 个连续事件才视为一个有效输出边沿：

0000：无滤波器，BRK 异步工作

0001：$f_{SAMPLING}=f_{CK_INT}$，N=2

0010：$f_{SAMPLING}=f_{CK_INT}$，N=4

0011：$f_{SAMPLING}=f_{CK_INT}$，N=8

0100：$f_{SAMPLING}=f_{DTS}/2$，N=6

0101：$f_{SAMPLING}=f_{DTS}/2$，N=8

0110：$f_{SAMPLING}=f_{DTS}/4$，N=6

0111：$f_{SAMPLING}=f_{DTS}/4$，N=8

1000：$f_{SAMPLING}=f_{DTS}/8$，N=6

1001：$f_{SAMPLING}=f_{DTS}/8$，N=8

1010：$f_{SAMPLING}=f_{DTS}/16$，N=5

1011：$f_{SAMPLING}=f_{DTS}/16$，N=6

1100：$f_{SAMPLING}=f_{DTS}/16$，N=8

1101：$f_{SAMPLING}=f_{DTS}/32$，N=5

1110：$f_{SAMPLING}=f_{DTS}/32$，N=6

1111：$f_{SAMPLING}=f_{DTS}/32$，N=8

注： 编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1 后，此位即无法修改。

位 15 **MOE**：主输出使能 (Main output enable)

只要断路输入（BRK 或 BRK2）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：响应断路事件（2 个）。禁止 OC 和 OCN 输出

响应断路事件或向 MOE 写入 0 时：OC 和 OCN 输出被禁止或被强制为空闲状态，具体取决于 OSSI 位。

1：如果 OC 和 OCN 输出的相应使能位（TIMx_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 34.4.9 节：TIM1/TIM8 捕获/比较使能寄存器 (TIMx_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 BRK 和 BRK2 均无效）

注：　只要编程了 LOCK（TIMx_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：　只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1，此位即无法修改。

注：　对该位执行任何写操作后，都需要经过 1 个 APB 时钟周期的延迟才生效。

位 12 **BKE**：断路使能 (Break enable)

该位可使能完整的断路保护（包括连接到 bk_acth 的所有源和相应的 BKIN 源，如*图 298：断路和断路 2 电路概述*部分所示）。

0：禁止断路功能

1：使能断路功能

注：　编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1 后，此位即无法修改。

注：　对该位执行任何写操作后，都需要经过 1 个 APB 时钟周期的延迟才生效。

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 34.4.9 节：TIM1/TIM8 捕获/比较使能寄存器 (TIMx_CCER)*）。

0：处于无效状态时，禁止 OC/OCN 输出（定时器释放输出控制，由强制高阻态的 GPIO 逻辑接管）。

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（输出仍由定时器控制）。

注：　编程了 LOCK（TIMx_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)

当由于断路事件或软件写操作而使 MOE=0 时，此位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 34.4.9 节：TIM1/TIM8 捕获/比较使能寄存器 (TIMx_CCER)*）。

0：处于无效状态时，禁止 OC/OCN 输出（定时器释放输出控制，由强制高阻态的 GPIO 逻辑接管）。

1：处于无效状态时，首先将 OC/OCN 输出强制为其无效电平，然后在死区后将其强制为空闲电平。定时器始终控制输出。

注： 编程了 LOCK（TIMx_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 TIMx_BDTR 寄存器中的 DTG 位、TIMx_CR2 寄存器中的 OISx 和 OISxN 位以及 TIMx_BDTR 寄存器中的 *BK2BID*、*BKBID*、*BK2DSRM*、*BKDSRM*、*BK2P*、*BK2E*、*BK2F[3:0]*、*BKF[3:0]*、*AOE*、*BKP*、*BKE*、*OSSI*、*OSSR* 和 *DTG[7:0]* 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（TIMx_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（TIMx_CCMRx 寄存器中的 OCxM 和 OCxPE 位，只要通过 CCxS 位将相关通道配置为输出）执行写操作。

注： 复位后只能对 LOCK 位执行一次写操作。对 TIMx_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-time generator setup)

此位域定义插入到互补输出之间的死区持续时间。DT 与该持续时间相对应。

DTG[7:5]=0xx => DT=DTG[7:0]x t_{dtg}，其中 $t_{dtg}=t_{DTS}$。

DTG[7:5]=10x => DT=(64+DTG[5:0])xt_{dtg}，其中 $T_{dtg}=2xt_{DTS}$。

DTG[7:5]=110 => DT=(32+DTG[4:0])xt_{dtg}，其中 $T_{dtg}=8xt_{DTS}$。

DTG[7:5]=111 => DT=(32+DTG[4:0])xt_{dtg}，其中 $T_{dtg}=16xt_{DTS}$。

示例：如果 T_{DTS}=125 ns (8 MHz)，则可能的死区值为：

0 到 15875 ns（步长为 125 ns），

16 µs 到 31750 ns（步长为 250 ns），

32 µs 到 63 µs（步长为 1 µs），

64 µs 到 126 µs（步长为 2 µs）

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

34.4.19 TIM1/TIM8 DMA 控制寄存器 (TIMx_DCR)

TIM1/TIM8 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
Res.	Res.	Res.	DBL[4:0]					Res.	Res.	Res.	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 位向量定义了 DMA 的传送长度（当对 TIMx_DMAR 地址进行读或写访问时，定时器进行一次连续传送），即传送次数。可按半字或字节进行传送（请参见下面的示例）。

00000：1 次传送
00001：2 次传送
00010：3 次传送
...
10001：18 次传送

示例：以下面的传送为例：DBL = 7 字节且 DBA = TIM2_CR1。

– 如果 DBL = 7 字节且 DBA = TIM2_CR1 表示待传送字节的地址，应通过以下公式给出传送的地址：

(TIMx_CR1 地址) + DBA + (DMA 索引)，其中 DMA 索引 = DBL

在本例中，将为 (TIMx_CR1 地址) + DBA 加上 7 个字节，得到将要复制数据的源/目标地址。

在这种情况下，将向自以下地址开始的 7 个寄存器传送数据：(TIMx_CR1 地址) + DBA

根据 DMA 数据大小的配置，可能发生下面几种情况：

– 如果按半字配置 DMA 数据大小，则将向 7 个寄存器中的每一个传送 16 位数据。
– 如果按字节配置 DMA 数据大小，也将向 7 个寄存器传送数据：第一个寄存器包含第一个 MSB 字节，第二个寄存器包含第一个 LSB 字节，依此类推。因此，使用传送定时器时，还必须指定 DMA 传送的数据大小。

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 TIMx_DMAR 地址进行读/写访问时）。DBA 定义为从 TIMx_CR1 寄存器地址开始计算的偏移量。

示例：

00000：TIMx_CR1，
00001：TIMx_CR2，
00010：TIMx_SMCR，
...

34.4.20 TIM1/TIM8 全传输 DMA 地址 (TIMx_DMAR)

TIM1/TIM8 DMA address for full transfer

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
DMAB[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
DMAB[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:0 **DMAB[31:0]**：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：(TIMx_CR1 地址) + (DBA + DMA 索引) x 4

其中 TIMx_CR1 地址为控制寄存器 1 的地址，DBA 为 TIMx_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（TIMx_DCR 寄存器中配置的 DBL）之间。

34.4.21 TIM1 选项寄存器 1 (TIM1_OR1)

TIM1 option register 1

偏移地址：0x50

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	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.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TI1_RMP	Res.	Res.	ETR_ADC1_RMP	
											rw			rw	rw

位 31:5 保留，必须保持复位值。

位 4 **TI1_RMP**：输入捕获 1 重映射 (Input Capture 1 remap)
0：TIM1 输入捕获 1 连接到 I/O。
1：TIM1 输入捕获 1 连接到 COMP1 输出。

位 3:2 保留，必须保持复位值。

位 1:0 **ETR_ADC1_RMP**：ADC1 模拟看门狗的外部触发重映射 (External trigger remap on ADC1 analog watchdog)
00：TIM1_ETR 不连接到 ADC1 AWDx。当 ETR 来自于 I/O 时，必须选择该配置。
01：TIM1_ETR 连接到 ADC1 AWD1。
10：TIM1_ETR 连接到 ADC1 AWD2。
11：TIM1_ETR 连接到 ADC1 AWD3。

注： *ADC1 AWDx 源与 TIM1_ETR 输入信号进行逻辑“或”运算。使用 ADC1 AWDx 时，需要确保复用功能控制器中未使能相应的 TIM1_ETR 输入引脚。请参见图 276：TIM1 ETR 输入电路。*

34.4.22 TIM8 选项寄存器 1 (TIM8_OR1)

TIM8 option register 1

偏移地址：0x50

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	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.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TI1_RMP	Res.	Res.	Res.	Res.
											rw				

位 31:5 保留，必须保持复位值。

位 4 **TI1_RMP**：输入捕获 1 重映射 (Input Capture 1 remap)
0：TIM8 输入捕获 1 连接到 I/O。
1：TIM8 输入捕获 1 连接到 COMP2 输出。

位 3:0 保留，必须保持复位值。

34.4.23 TIM1/TIM8 捕获/比较模式寄存器 3 (TIMx_CCMR3)

TIM1/TIM8 capture/compare mode register 3

偏移地址：0x54

复位值：0x0000 0000

请参见上述 CCMR1 寄存器说明。通道 5 和通道 6 只能配置为输出。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC6M[3]	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC5M[3]
							rw								rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OC6 CE	OC6M[2:0]			OC6 PE	OC6FE	Res.	Res.	OC5 CE	OC5M[2:0]			OC5PE	OC5FE	Res.	Res.
rw	rw	rw	rw	rw	rw			rw	rw	rw	rw	rw	rw		

输出比较模式

位 31:25 保留，必须保持复位值。

位 23:17 保留，必须保持复位值。

位 15 **OC6CE**：输出比较 6 清零使能 (Output compare 6 clear enable)

位 24、14、13、12 **OC6M[3:0]**：输出比较 6 模式 (Output Compare 6 mode)

位 11 **OC6PE**：输出比较 6 预装载使能 (Output compare 6 preload enable)

位 10 **OC6FE**：输出比较 6 快速使能 (Output compare 6 fast enable)

位 9:8 保留，必须保持复位值。

位 7 **OC5CE**：输出比较 5 清零使能 (Output compare 5 clear enable)

位 16、6、5、4 **OC5M[3:0]**：输出比较 5 模式 (Output Compare 5 mode)

位 3 **OC5PE**：输出比较 5 预装载使能 (Output Compare 5 preload enable)

位 2 **OC5FE**：输出比较 5 快速使能 (Output compare 5 fast enable)

位 1:0 保留，必须保持复位值。

34.4.24 TIM1/TIM8 捕获/比较寄存器 5 (TIMx_CCR5)

TIM1/TIM8 capture/compare register 5

偏移地址：0x58

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GC5C3	GC5C2	GC5C1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
rw	rw	rw													
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR5[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31 **GC5C3**：通道 5 和通道 3 组 (Group Channel 5 and Channel 3)

通道 3 输出上失真：

0：OC5REF 对 OC3REFC 无影响

1：OC3REFC 是 OC3REFC 和 OC5REF 的逻辑与运算结果

该位可以立即生效，也可预装载并在更新事件后执行（如果在 TIMxCCMR2 中选择了预装载功能）。

注： 也可在组合 PWM 信号上应用此失真。

位 30 **GC5C2**：通道 5 和通道 2 组 (Group Channel 5 and Channel 2)

通道 2 输出上失真：

0：OC5REF 对 OC2REFC 无影响

1：OC2REFC 是 OC2REFC 和 OC5REF 的逻辑与运算结果

该位可以立即生效，也可预装载并在更新事件后执行（如果在 TIMxCCMR1 中选择了预装载功能）。

注： 也可在组合 PWM 信号上应用此失真。

位 29 **GC5C1**：通道 5 和通道 1 组 (Group Channel 5 and Channel 1)

通道 1 输出上失真：

0：OC5REF 对 OC1REFC5 无影响

1：OC1REFC 是 OC1REFC 和 OC5REF 的逻辑与运算结果

该位可以立即生效，也可预装载并在更新事件后执行（如果在 TIMxCCMR1 中选择了预装载功能）。

注： 也可在组合 PWM 信号上应用此失真。

位 28:16 保留，必须保持复位值。

位 15:0 **CCR5[15:0]**：捕获/比较 5 值 (Capture/Compare 5 value)

CCR5 是捕获/比较寄存器 5 的预装载值。

如果没有通过 TIMx_CCMR3 寄存器中的 OC5PE 位来使能预装载功能，则该值立刻生效；否则只在发生更新事件时生效（拷贝到有效的捕获/比较寄存器 5）。

有效捕获/比较寄存器中包含要与计数器 TIMx_CNT 进行比较并在 OC5 输出上发出信号的值。

34.4.25 TIM1/TIM8 捕获/比较寄存器 6 (TIMx_CCR6)

TIM1/TIM8 capture/compare register 6

偏移地址：0x5C

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR6[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 15:0 **CCR6[15:0]**：捕获/比较 6 值 (Capture/Compare 6 value)

CCR6 是捕获/比较寄存器 6 的预装载值。

如果没有通过 TIMx_CCMR3 寄存器中的 OC6PE 位来使能预装载功能，则该值立刻生效；否则只在发生更新事件时生效（拷贝到有效的捕获/比较寄存器 6）。

有效捕获/比较寄存器中包含要与计数器 TIMx_CNT 进行比较并在 OC6 输出上发出信号的值。

34.4.26 TIM1 选项寄存器 2 (TIM1_OR2)

TIM1 option register 2

偏移地址：0x60

复位值：0x0000 0001

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ETRSEL[1:0]		Res.	Res.	BKCMP2P	BKCMP1P	BKINP	BKDF1BK0E	Res.	Res.	Res.	Res.	Res.	BKCMP2E	BKCMP1E	BKINE
rw	rw			rw	rw	rw	rw						rw	rw	rw

位 31:17 保留，必须保持复位值。

位 16:14 **ETRSEL[2:0]**：ETR 源选择 (ETR source selection)

这些位选择 ETR 输入源。

000：ETR 输入连接到 I/O

001：COMP1 输出

010：COMP2 输出

011：ADC1 AWD1

100：ADC1 AWD2

101：ADC1 AWD3

其他值：保留

注： 只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1，这些位即无法修改。

位 13:12 保留，必须保持复位值。

位 11 **BKCMP2P**：BRK COMP2 输入极性 (BRK COMP2 input polarity)
此位选择 COMP2 输入灵敏度，必须与 BKP 极性位一起编程。
0：COMP2 输入为高电平有效
1：COMP2 输入为低电平有效
注： 只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1，此位即无法修改。

位 10 **BKCMP1P**：BRK COMP1 输入极性 (BRK COMP1 input polarity)
此位选择 COMP1 输入灵敏度，必须与 BKP 极性位一起编程。
0：COMP1 输入为高电平有效
1：COMP1 输入为低电平有效
注： 只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1，此位即无法修改。

位 9 **BKINP**：BRK BKIN 输入极性 (BRK BKIN input polarity)
此位选择 BKIN 复用功能输入灵敏度，必须与 BKP 极性位一起编程。
0：BKIN 输入为高电平有效
1：BKIN 输入为低电平有效
注： 只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1，此位即无法修改。

位 8 **BKDF1BK0E**：BRK dfsdm1_break[0] 使能 (BRK dfsdm1_break[0] enable)
此位使能定时器 BRK 输入的 dfsdm1_break[0]。dfsdm1_break[0] 输出与其他 BRK 源进行“或”运算。
0：禁止 dfsdm1_break[0] 输入
1：使能 dfsdm1_break[0] 输入
注： 只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1，此位即无法修改。

位 7:3 保留，必须保持复位值。

位 2 **BKCMP2E**：BRK COMP2 使能 (BRK COMP2 enable)
此位使能定时器 BRK 输入的 COMP2。COMP2 输出与其他 BRK 源进行“或”运算。
0：禁止 COMP2 输入
1：使能 COMP2 输入
注： 只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1，此位即无法修改。

位 1 **BKCMP1E**：BRK COMP1 使能 (BRK COMP1 enable)
此位使能定时器 BRK 输入的 COMP1。COMP1 输出与其他 BRK 源进行“或”运算。
0：禁止 COMP1 输入
1：使能 COMP1 输入
注： 只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1，此位即无法修改。

位 0 **BKINE**：BRK BKIN 输入使能 (BRK BKIN input enable)
此位使能定时器 BRK 输入的 BKIN 复用功能。BKIN 输入与其他 BRK 源进行“或”运算。
0：禁止 BKIN 输入
1：使能 BKIN 输入
注： 只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1，此位即无法修改。

注： 请参见图 276：TIM1 ETR 输入电路和图 298：断路和断路 2 电路概述。

34.4.27 TIM1 选项寄存器 3 (TIM1_OR3)

TIM1 option register 3

偏移地址：0x64

复位值：0x0000 0001

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.	Res.	Res.	BK2 CMP2 P	BK2 CMP1 P	BK2 INP	BK2DF1 BK1E	Res.	Res.	Res.	Res.	Res.	BK2 CMP2E	BK2 CMP1E	BK2INE
				rw	rw	rw	rw						rw	rw	rw

位 31:12 保留，必须保持复位值。

位 11 **BK2CMP2P**：BRK2 COMP2 输入极性 (BRK2 COMP2 input polarity)

此位选择 COMP2 输入灵敏度，必须与 BKP2 极性位一起编程。

0：COMP2 输入为低电平有效

1：COMP2 输入为高电平有效

注： 只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1，此位即无法修改。

位 10 **BK2CMP1P**：BRK2 COMP1 输入极性 (BRK2 COMP1 input polarity)

此位选择 COMP1 输入灵敏度，必须与 BKP2 极性位一起编程。

0：COMP1 输入为低电平有效

1：COMP1 输入为高电平有效

注： 只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1，此位即无法修改。

位 9 **BK2INP**：BRK2 BKIN2 输入极性 (BRK2 BKIN2 input polarity)

此位选择 BKIN2 复用功能输入灵敏度，必须与 BKP2 极性位一起编程。

0：BKIN2 输入为低电平有效

1：BKIN2 输入为高电平有效

注： 只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1，此位即无法修改。

位 8 **BK2DF1BK1E**：BRK2 dfsdm1_break[1] 使能 (BRK2 dfsdm1_break[1] enable)

此位使能定时器 BRK2 输入的 dfsdm1_break[1]。dfsdm1_break[1] 输出与其他 BRK2 源进行“或”运算。

0：禁止 dfsdm1_break[1] 输入

1：使能 dfsdm1_break[1] 输入

注： 只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1，此位即无法修改。

位 7:3 保留，必须保持复位值。

位 2 **BK2CMP2E**：BRK2 COMP2 使能 (BRK2 COMP2 enable)

此位使能定时器 BRK2 输入的 COMP2。COMP2 输出与其他 BRK2 源进行“或”运算。

0：禁止 COMP2 输入

1：使能 COMP2 输入

注： 只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1，此位即无法修改。

位 1 **BK2CMP1E**：BRK2 COMP1 使能 (BRK2 COMP1 enable)

此位使能定时器 BRK2 输入的 COMP1。COMP1 输出与其他 BRK2 源进行“或”运算。

0：禁止 COMP1 输入

1：使能 COMP1 输入

注： 只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1，此位即无法修改。

位 0 **BK2INE**：BRK2 BKIN 输入使能 (BRK2 BKIN input enable)
此位使能定时器 BRK2 输入的 BKIN2 复用功能。BKIN2 输入与其他 BRK2 源进行“或”运算。
0：禁止 BKIN2 输入
1：使能 BKIN2 输入
注： 只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1，此位即无法修改。

注： 请参见图 298：断路和断路 2 电路概述。

34.4.28 TIM8 选项寄存器 2 (TIM8_OR2)

TIM8 option register 2

偏移地址：0x60

复位值：0x0000 0001

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ETRSEL[1:0]		Res.	Res.	BK CMP2 P	BK CMP1 P	BKINP	BKDF1 BK2E	Res.	Res.	Res.	Res.	Res.	BK CMP2E	BK CMP1E	BKINE
rw	rw			rw	rw	rw	rw						rw	rw	rw

位 31:17 保留，必须保持复位值。

位 16:14 **ETRSEL[2:0]**：ETR 源选择 (ETR source selection)
这些位选择 ETR 输入源。
000：ETR 输入连接到 I/O
001：COMP1 输出
010：COMP2 输出
其他值：保留
注： 只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1，这些位即无法修改。

位 13:12 保留，必须保持复位值。

位 11 **BKCMP2P**：BRK COMP2 输入极性 (BRK COMP2 input polarity)
此位选择 COMP2 输入灵敏度，必须与 BKP 极性位一起编程。
0：COMP2 输入为高电平有效
1：COMP2 输入为低电平有效
注： 只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1，此位即无法修改。

位 10 **BKCMP1P**：BRK COMP1 输入极性 (BRK COMP1 input polarity)
此位选择 COMP1 输入灵敏度，必须与 BKP 极性位一起编程。
0：COMP1 输入为高电平有效
1：COMP1 输入为低电平有效
注： 只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1，此位即无法修改。

位 9 **BKINP**：BRK BKIN 输入极性 (BRK BKIN input polarity)
此位选择 BKIN 复用功能输入灵敏度，必须与 BKP 极性位一起编程。
0：BKIN 输入为高电平有效
1：BKIN 输入为低电平有效
注： 只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1，此位即无法修改。

位 8 **BKDF1BK2E**：BRK dfsdm1_break[2] 使能 (BRK dfsdm1_break[2] enable)
此位使能定时器 BRK 输入的 dfsdm1_break[2]。dfsdm1_break[2] 输出与其他 BRK 源进行“或”运算。
0：禁止 dfsdm1_break[2] 输入
1：使能 dfsdm1_break[2] 输入
注： 只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1，此位即无法修改。

位 7:3 保留，必须保持复位值。

位 2 **BKCMP2E**：BRK COMP2 使能 (BRK COMP2 enable)
此位使能定时器 BRK 输入的 COMP2。COMP2 输出与其他 BRK 源进行“或”运算。
0：禁止 COMP2 输入
1：使能 COMP2 输入
注： 只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1，此位即无法修改。

位 1 **BKCMP1E**：BRK COMP1 使能 (BRK COMP1 enable)
此位使能定时器 BRK 输入的 COMP1。COMP1 输出与其他 BRK 源进行“或”运算。
0：禁止 COMP1 输入
1：使能 COMP1 输入
注： 只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1，此位即无法修改。

位 0 **BKINE**：BRK BKIN 输入使能 (BRK BKIN input enable)
此位使能定时器 BRK 输入的 BKIN 复用功能。BKIN 输入与其他 BRK 源进行“或”运算。
0：禁止 BKIN 输入
1：使能 BKIN 输入
注： 只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1，此位即无法修改。

注： 请参见图 277：TIM8 ETR 输入电路和图 298：断路和断路 2 电路概述。

34.4.29 TIM8 选项寄存器 3 (TIM8_OR3)

TIM8 option register 3

偏移地址：0x64

复位值：0x0000 0001

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.	Res.	Res.	BK2 CMP2 P	BK2 CMP1 P	BK2 INP	BK2DF1 BK3E	Res.	Res.	Res.	Res.	Res.	BK2 CMP2E	BK2 CMP1E	BK2INE
				rw	rw	rw	rw						rw	rw	rw

位 31:12 保留，必须保持复位值。

位 11 **BK2CMP2P**：BRK2 COMP2 输入极性 (BRK2 COMP2 input polarity)
此位选择 COMP2 输入灵敏度，必须与 BKP2 极性位一起编程。
0：COMP2 输入为低电平有效
1：COMP2 输入为高电平有效
注： 只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1，此位即无法修改。

位 10 **BK2CMP1P**：BRK2 COMP1 输入极性 (BRK2 COMP1 input polarity)
此位选择 COMP1 输入灵敏度，必须与 BKP2 极性位一起编程。
0：COMP1 输入为低电平有效
1：COMP1 输入为高电平有效
注： 只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1，此位即无法修改。

位 9 **BK2INP**：BRK2 BKIN2 输入极性 (BRK2 BKIN2 input polarity)
此位选择 BKIN2 复用功能输入灵敏度，必须与 BKP2 极性位一起编程。
0：BKIN2 输入为低电平有效
1：BKIN2 输入为高电平有效
注： 只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1，此位即无法修改。

位 8 **BK2DF1BK3E**：BRK2 dfsdm1_break[3] 使能 (BRK2 dfsdm1_break[3] enable)
此位使能定时器 BRK2 输入的 dfsdm1_break[3]。dfsdm1_break[3] 输出与其他 BRK2 源进行“或”运算。
0：禁止 dfsdm1_break[3] 输入
1：使能 dfsdm1_break[3] 输入
注： 只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1，此位即无法修改。

位 7:3 保留，必须保持复位值。

位 2 **BK2CMP2E**：BRK2 COMP2 使能 (BRK2 COMP2 enable)
此位使能定时器 BRK2 输入的 COMP2。COMP2 输出与其他 BRK2 源进行“或”运算。
0：禁止 COMP2 输入
1：使能 COMP2 输入
注： 只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1，此位即无法修改。

位 1 **BK2CMP1E**：BRK2 COMP1 使能 (BRK2 COMP1 enable)
此位使能定时器 BRK2 输入的 COMP1。COMP1 输出与其他 BRK2 源进行“或”运算。
0：禁止 COMP1 输入
1：使能 COMP1 输入
注： 只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1，此位即无法修改。

位 0 **BK2INE**：BRK2 BKIN 输入使能 (BRK2 BKIN input enable)
此位使能定时器 BRK2 输入的 BKIN2 复用功能。BKIN2 输入与其他 BRK2 源进行“或”运算。
0：禁止 BKIN2 输入
1：使能 BKIN2 输入
注： 只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1，此位即无法修改。

注： 请参见图 298：断路和断路 2 电路概述。

34.4.30 TIM1 寄存器映射

TIM1 寄存器可映射为 16 位可寻址寄存器，如下表所述：

表 217. TIM1 寄存器映射和复位值

偏移	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	**TIM1_CR1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	UIFREMAP	Res.	CKD [1:0]		ARPE	CMS [1:0]		DIR	OPM	URS	UDIS	CEN
	Reset value																					0		0	0	0	0	0	0	0	0	0	0
0x04	**TIM1_CR2**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MMS2[3:0]				Res.	OIS6	Res.	OIS5	Res.	OIS4	OIS3N	OIS3	OIS2N	OIS2	OIS1N	OIS1	TI1S	MMS [2:0]			CCDS	CCUS	Res.	CCPC
	Reset value									0	0	0	0		0		0		0	0	0	0	0	0	0	0	0	0	0	0	0		0
0x08	**TIM1_SMCR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SMS[3]	ETP	ECE	ETPS [1:0]		ETF[3:0]				MSM	TS[2:0]			Res.	SMS[2:0]		
	Reset value																0	0	0	0	0	0	0	0	0	0	0	0	0		0	0	0
0x0C	**TIM1_DIER**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	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	**TIM1_SR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CC6IF	CC5IF	Res.	Res.	SBIF	CC4OF	CC3OF	CC2OF	CC1OF	B2IF	BIF	TIF	COMIF	CC4IF	CC3IF	CC2IF	CC1IF	UIF
	Reset value															0	0			0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x14	**TIM1_EGR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	B2G	BG	TG	COM	CC4G	CC3G	CC2G	CC1G	UG
	Reset value																								0	0	0	0	0	0	0	0	0
0x18	**TIM1_CCMR1** Output Compare mode	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC2M[3]	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC1M[3]	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	0	0
	TIM1_CCMR1 Input Capture mode	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	IC2F[3:0]				IC2PSC [1:0]		CC2S [1:0]		IC1F[3:0]				IC1PSC [1:0]		CC1S [1:0]	
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x1C	**TIM1_CCMR2** Output Compare mode	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC4M[3]	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC3M[3]	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	0	0
	TIM1_CCMR2 Input Capture mode	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	IC4F[3:0]				IC4PSC [1:0]		CC4S [1:0]		IC3F[3:0]				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	**TIM1_CCER**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CC6P	CC6E	Res.	Res.	CC5P	CC5E	Res.	Res.	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	0	0	0	0

表 217. TIM1 寄存器映射和复位值（续）

偏移	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
0x24	**TIM1_CNT**	UIFCPY	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CNT[15:0]															
	Reset value	0																0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x28	**TIM1_PSC**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PSC[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x2C	**TIM1_ARR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ARR[15:0]															
	Reset value																	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0x30	**TIM1_RCR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REP[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x34	**TIM1_CCR1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CCR1[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x38	**TIM1_CCR2**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CCR2[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x3C	**TIM1_CCR3**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CCR3[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x40	**TIM1_CCR4**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CCR4[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x44	**TIM1_BDTR**	Res.	Res.	BK2BID	BKBID	BK2DSRM	BKDSRM	BK2P	BK2E	BK2F[3:0]				BKF[3:0]				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	0	0	0	0	0	0	0	0	0	0	0	0	0
0x48	**TIM1_DCR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DBL[4:0]					Res.	Res.	Res.	DBA[4:0]				
	Reset value																				0	0	0	0	0				0	0	0	0	0
0x4C	**TIM1_DMAR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DMAB[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x50	**TIM1_OR1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TI1_RMP	Res.	Res.	ETR_ADC1_RMP	
	Reset value																												0			0	0
0x54	**TIM1_CCMR3** Output Compare mode	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC6M[3]	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC5M[3]	OC6CE	OC6M [2:0]			OC6PE	OC6FE	Res.	Res.	OC5CE	OC5M [2:0]			OC5PE	OC5FE	Res.	Res.
	Reset value								0								0	0	0	0	0	0	0			0	0	0	0	0	0		

表 217. TIM1 寄存器映射和复位值（续）

偏移	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
0x58	**TIM1_CCR5**	GC5C3	GC5C2	GC5C1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CCR5[15:0]															
	Reset value	0	0	0														0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x5C	**TIM1_CCR6**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CCR6[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x60	**TIM1_OR2**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ETRSEL [2:0]			Res.	Res.	BKCMP2P	BKCMP1P	BKINP	BKDF1BK0E	Res.	Res.	Res.	Res.	Res.	BKCMP2E	BKCMP1E	BKINE
	Reset value																0	0	0			0	0	0	0						0	0	1
0x64	**TIM1_OR3**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BK2CMP2P	BK2CMP1P	BK2INP	BK2DF1BK1E	Res.	Res.	Res.	Res.	Res.	BK2CMP2E	BK2CMP1E	BK2INE
	Reset value																					0	0	0	0						0	0	1

有关寄存器边界地址的信息，请参见*第 2.2.2 节：存储器映射和寄存器边界地址*。

34.4.31 TIM8 寄存器映射

TIM8 寄存器可映射为 16 位可寻址寄存器，如下表所述：

表 218. TIM8 寄存器映射和复位值

偏移	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	**TIM8_CR1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	UIFREMAP	Res.	CKD [1:0]		ARPE	CMS [1:0]		DIR	OPM	URS	UDIS	CEN
	Reset value																					0		0	0	0	0	0	0	0	0	0	0
0x04	**TIM8_CR2**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MMS2[3:0]				Res.	OIS6	Res.	OIS5	Res.	OIS4	OIS3N	OIS3	OIS2N	OIS2	OIS1N	OIS1	TI1S	MMS [2:0]			CCDS	CCUS	Res.	CCPC
	Reset value									0	0	0	0		0		0		0	0	0	0	0	0	0	0	0	0	0	0	0		0
0x08	**TIM8_SMCR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SMS[3]	ETP	ECE	ETPS [1:0]		ETF[3:0]				MSM	TS[2:0]			Res.	SMS[2:0]		
	Reset value																0	0	0	0	0	0	0	0	0	0	0	0	0		0	0	0
0x0C	**TIM8_DIER**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	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	**TIM8_SR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CC6IF	CC5IF	Res.	Res.	SBIF	CC4OF	CC3OF	CC2OF	CC1OF	B2IF	BIF	TIF	COMIF	CC4IF	CC3IF	CC2IF	CC1IF	UIF
	Reset value															0	0			0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x14	**TIM8_EGR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	B2G	BG	TG	COM	CC4G	CC3G	CC2G	CC1G	UG
	Reset value																								0	0	0	0	0	0	0	0	0
0x18	**TIM8_CCMR1** Output Compare mode	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC2M[3]	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC1M[3]	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	0	0
	TIM8_CCMR1 Input Capture mode	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	IC2F[3:0]				IC2PSC [1:0]		CC2S [1:0]		IC1F[3:0]				IC1PSC [1:0]		CC1S [1:0]	
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x1C	**TIM8_CCMR2** Output Compare mode	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC4M[3]	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC3M[3]	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	0	0
	TIM8_CCMR2 Input Capture mode	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	IC4F[3:0]				IC4PSC [1:0]		CC4S [1:0]		IC3F[3:0]				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	**TIM8_CCER**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CC6P	CC6E	Res.	Res.	CC5P	CC5E	Res.	Res.	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	0	0	0	0
0x24	**TIM8_CNT**	UIFCPY	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CNT[15:0]															
	Reset value	0																0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

表 218. TIM8 寄存器映射和复位值（续）

偏移	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
0x28	**TIM8_PSC**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PSC[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x2C	**TIM8_ARR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ARR[15:0]															
	Reset value																	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0x30	**TIM8_RCR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REP[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x34	**TIMx_CCR1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CCR1[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x38	**TIM8_CCR2**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CCR2[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x3C	**TIM8_CCR3**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CCR3[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x40	**TIM8_CCR4**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CCR4[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x44	**TIM8_BDTR**	Res.	Res.	BK2BID	BKBID	BK2DSRM	BKDSRM	BK2P	BK2E	BK2F[3:0]				BKF[3:0]				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	0	0	0	0	0	0	0	0	0	0	0	0	0
0x48	**TIM8_DCR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DBL[4:0]			Res.	Res.	Res.	DBA[4:0]						
	Reset value																				0	0	0	0	0				0	0	0	0	0
0x4C	**TIM8_DMAR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DMAB[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x54	**TIM8_CCMR3** Output Compare mode	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC6M[3]	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC5M[3]	OC6CE	OC6M [2:0]			OC6PE	OC6FE	Res.	Res.	OC5CE	OC5M [2:0]			OC5PE	OC5FE	Res.	Res.
	Reset value								0								0	0	0	0	0	0	0			0	0	0	0	0	0		
0x58	**TIM8_CCR5**	GC5C3	GC5C2	GC5C1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CCR5[15:0]															
	Reset value	0	0	0														0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

表 218. TIM8 寄存器映射和复位值（续）

偏移	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
0x5C	**TIM8_CCR6**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CCR6[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x60	**TIM8_OR2**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ETRSEL [2:0]			Res.	Res.	BKCMP2P	BKCMP1P	BKINP	BKDF1BK2E	Res.	Res.	Res.	Res.	Res.	BKCMP2E	BKCMP1E	BKINE
	Reset value																0	0	0			0	0	0	0						0	0	1
0x64	**TIM8_OR3**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BK2CMP2P	BK2CMP1P	BK2INP	BK2DF1BK3E	Res.	Res.	Res.	Res.	Res.	BK2CMP2E	BK2CMP1E	BK2INE
	Reset value																					0	0	0	0						0	0	1

有关寄存器边界地址的信息，请参见*第 78 页的第 2.2.2 节*。

35 通用定时器 (TIM2/TIM3/TIM4/TIM5)

35.1 TIM2/TIM3/TIM4/TIM5 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。

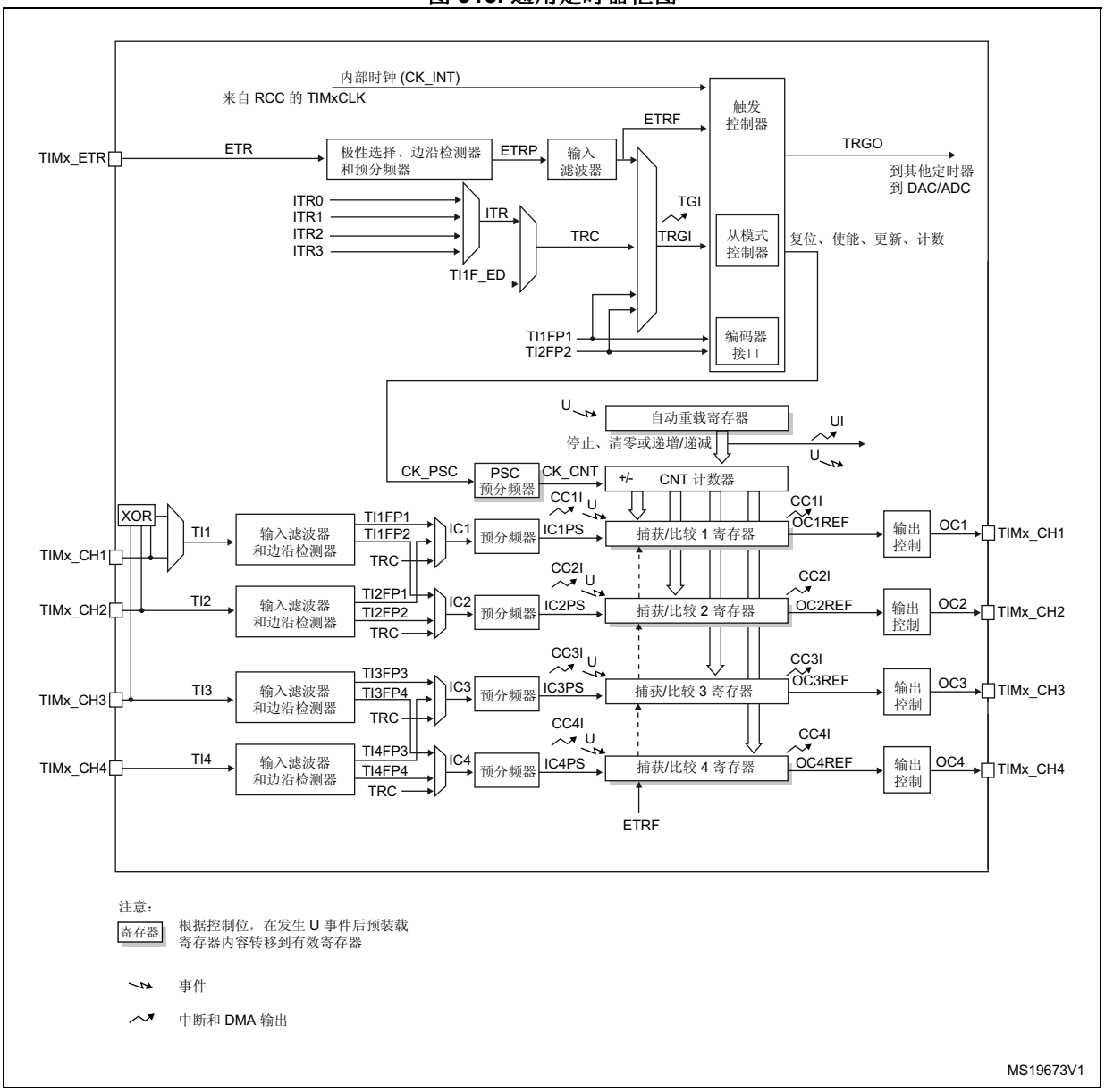
这些定时器彼此完全独立，不共享任何资源。如*第 35.3.19 节：定时器同步*中所述，它们可以一起同步操作。

35.2 TIM2/TIM3/TIM4/TIM5 主要特性

通用 TIMx 定时器具有以下特性：

- 16 位（TIM3 和 TIM4）或 32 位（TIM2 和 TIM5）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（可在运行时修改），分频系数介于 1 到 65535 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。
- 触发输入用作外部时钟或逐周期电流管理。

图 315. 通用定时器框图



35.3 TIM2/TIM3/TIM4/TIM5 功能描述

35.3.1 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。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 (TIMx_CNT)
- 预分频器寄存器 (TIMx_PSC)
- 自动重载寄存器 (TIMx_ARR)

自动重载寄存器是预装载的。对自动重载寄存器执行写入或读取操作时会访问预装载寄存器。预装载寄存器的内容既可以直接传送到影子寄存器，也可以在每次发生更新事件 (UEV) 时传送到影子寄存器，这取决于 TIMx_CR1 寄存器中的自动重载预装载使能位 (ARPE)。当计数器达到上溢值（或者在递减计数时达到下溢值）并且 TIMx_CR1 寄存器中的 UDIS 位为 0 时，将发送更新事件。该更新事件也可由软件产生。下文将针对各配置的更新事件的产生进行详细介绍。

计数器由预分频器输出 CK_CNT 提供时钟，仅当 TIMx_CR1 寄存器中的计数器启动位 (CEN) 置 1 时，才会启动计数器（有关计数器使能的更多详细信息，另请参见从模式控制器的相关说明）。

请注意，实际的计数器使能信号 CNT_EN 在 CEN 置 1 的一个时钟周期后被置 1。

预分频器说明

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 16 位/32 位寄存器（TIMx_PSC 寄存器）所控制的 16 位计数器。由于该控制寄存器具有缓冲功能，因此预分频器可实现实时更改。而新的预分频比将在下一更新事件发生时被采用。

图 316 和*图 317* 以一些示例说明在预分频比实时变化时计数器的行为：

图 316. 预分频器分频由 1 变为 2 时的计数器时序图

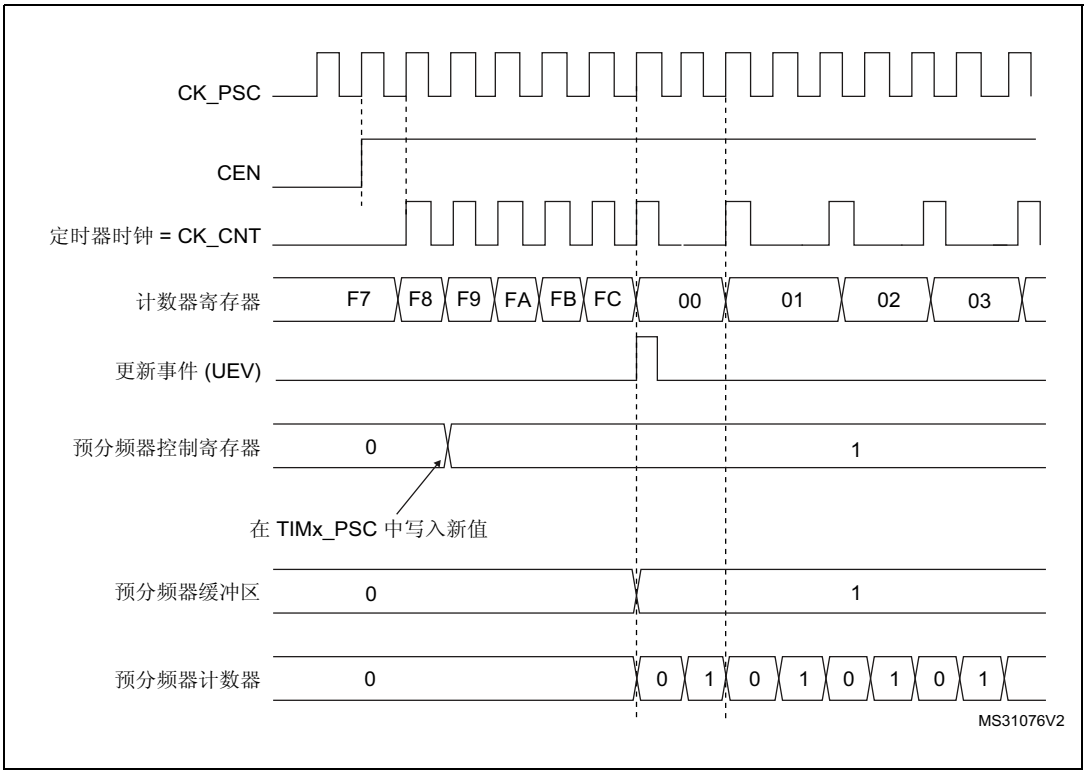
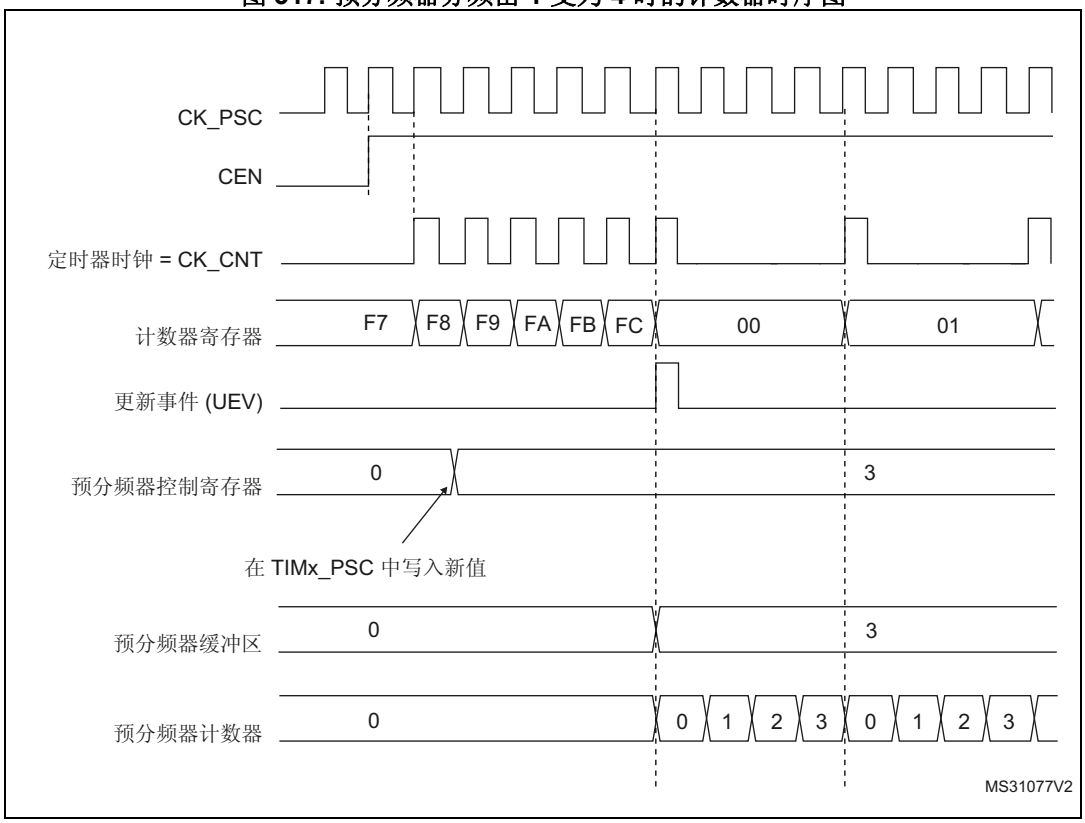


图 317. 预分频器分频由 1 变为 4 时的计数器时序图



35.3.2 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（TIMx_ARR 寄存器的内容），然后重新从 0 开始计数并生成计数器上溢事件。

每次发生计数器上溢时会生成更新事件，或将 TIMx_EGR 寄存器中的 UG 位置 1（通过软件或使用从模式控制器）也可以生成更新事件。

通过软件将 TIMx_CR1 寄存器中的 UDIS 位置 1 可禁止 UEV 事件。这可避免向预装载寄存器写入新值时更新影子寄存器。在 UDIS 位写入 0 之前不会产生任何更新事件。不过，计数器和预分频器计数器都会重新从 0 开始计数（而预分频比保持不变）。此外，如果 TIMx_CR1 寄存器中的 URS 位（更新请求选择）已置 1，则将 UG 位置 1 会生成更新事件 UEV，但不会将 UIF 标志置 1（因此，不会发送任何中断或 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（TIMx_SR 寄存器中的 UIF 位）置 1（取决于 URS 位）：

- 预分频器的缓冲区中将重新装载预装载值（TIMx_PSC 寄存器的内容）
- 使用预装载值 (TIMx_ARR) 更新自动重载影子寄存器

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

图 318. 计数器时序图，1 分频内部时钟

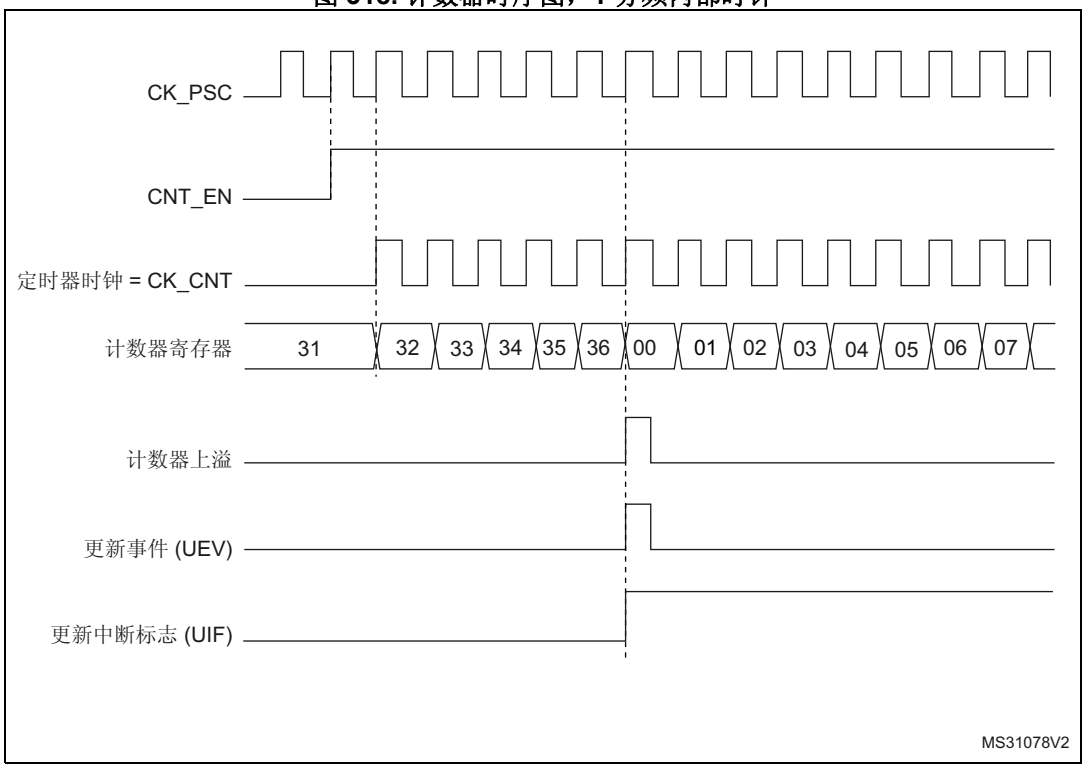


图 319. 计数器时序图，2 分频内部时钟

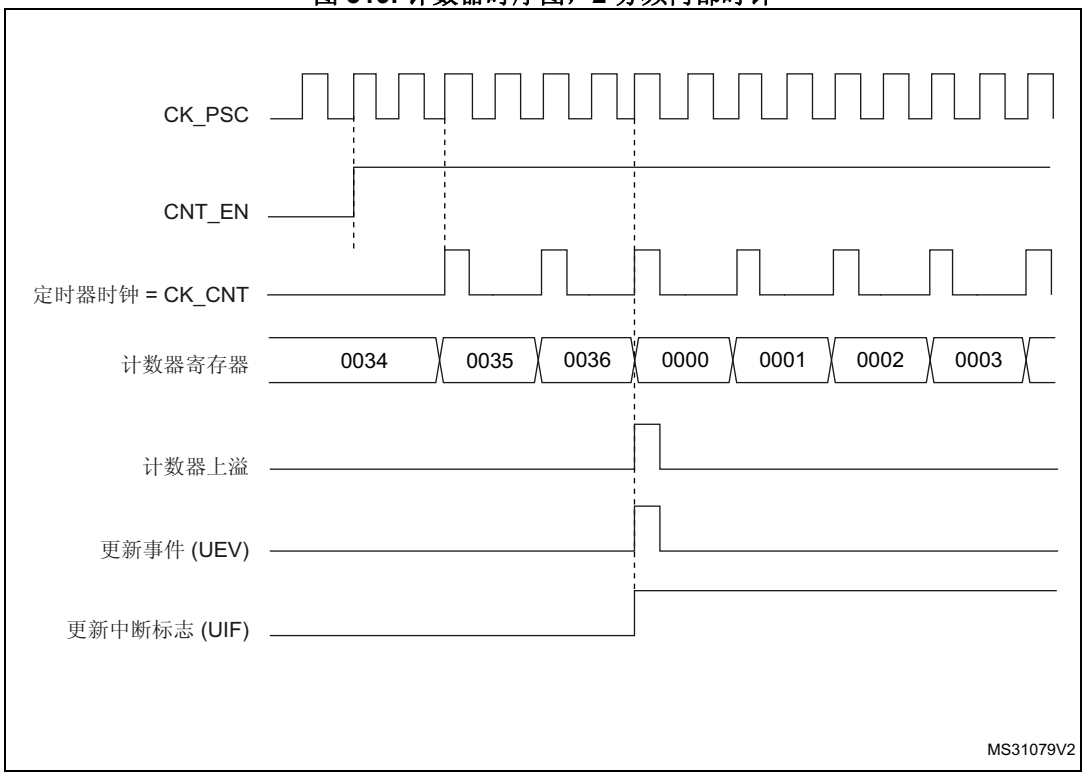


图 320. 计数器时序图，4 分频内部时钟

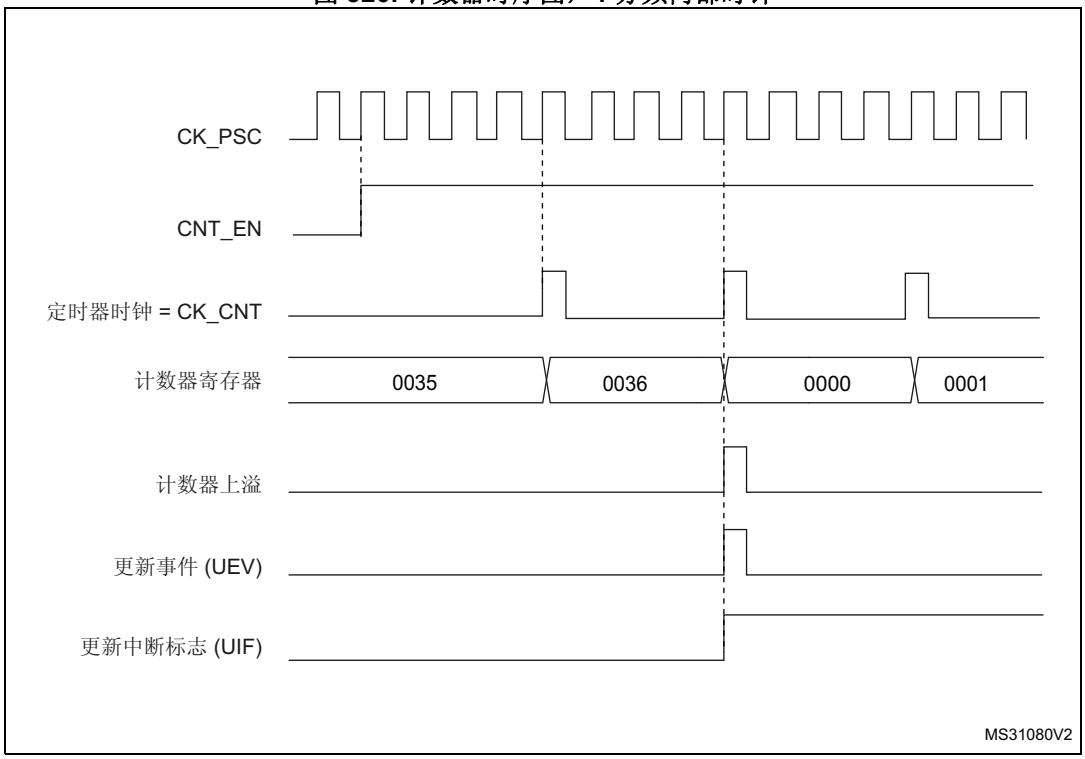


图 321. 计数器时序图，N 分频内部时钟

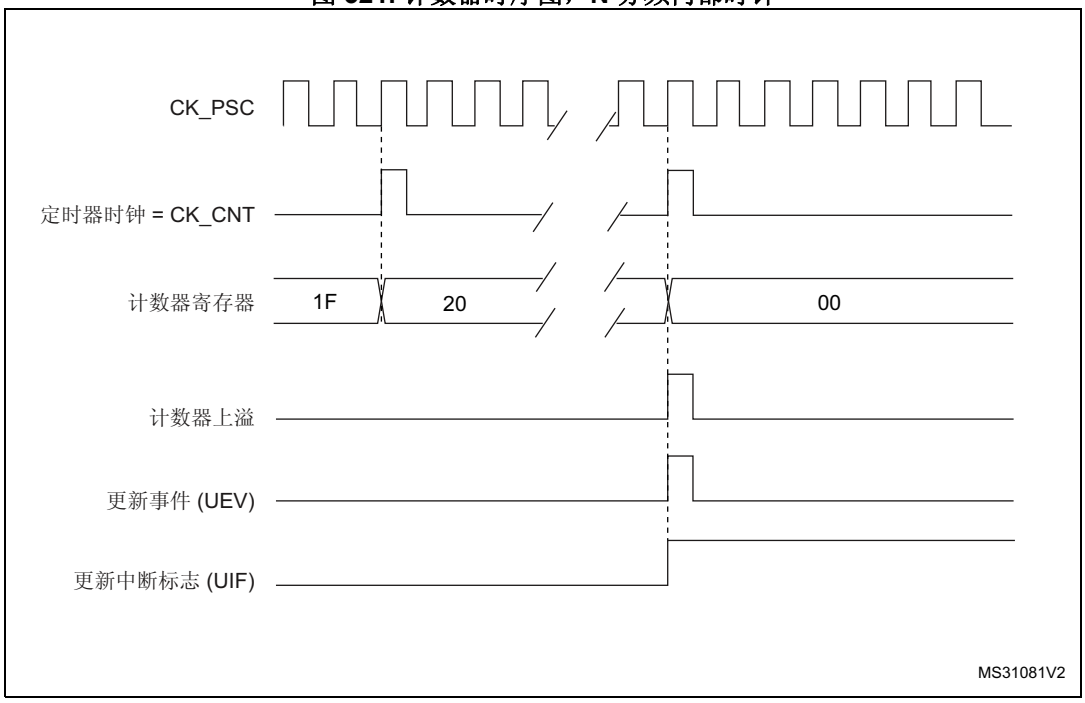


图 322. 计数器时序图，ARPE=0 时更新事件（TIMx_ARR 未预装载）

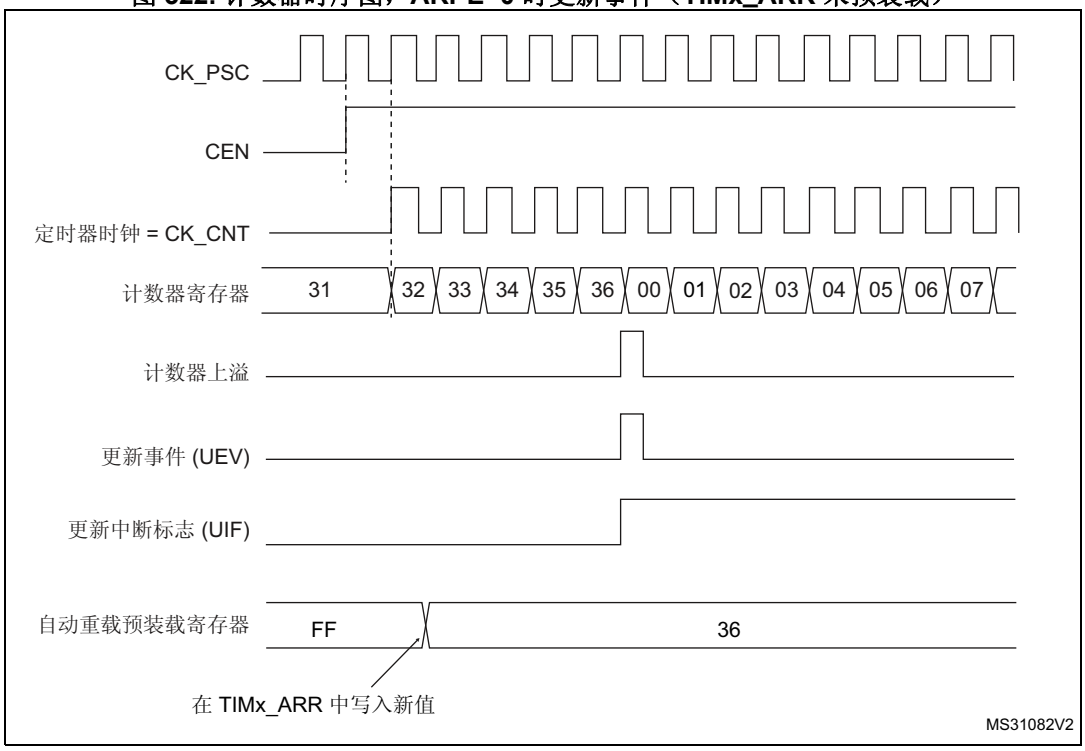
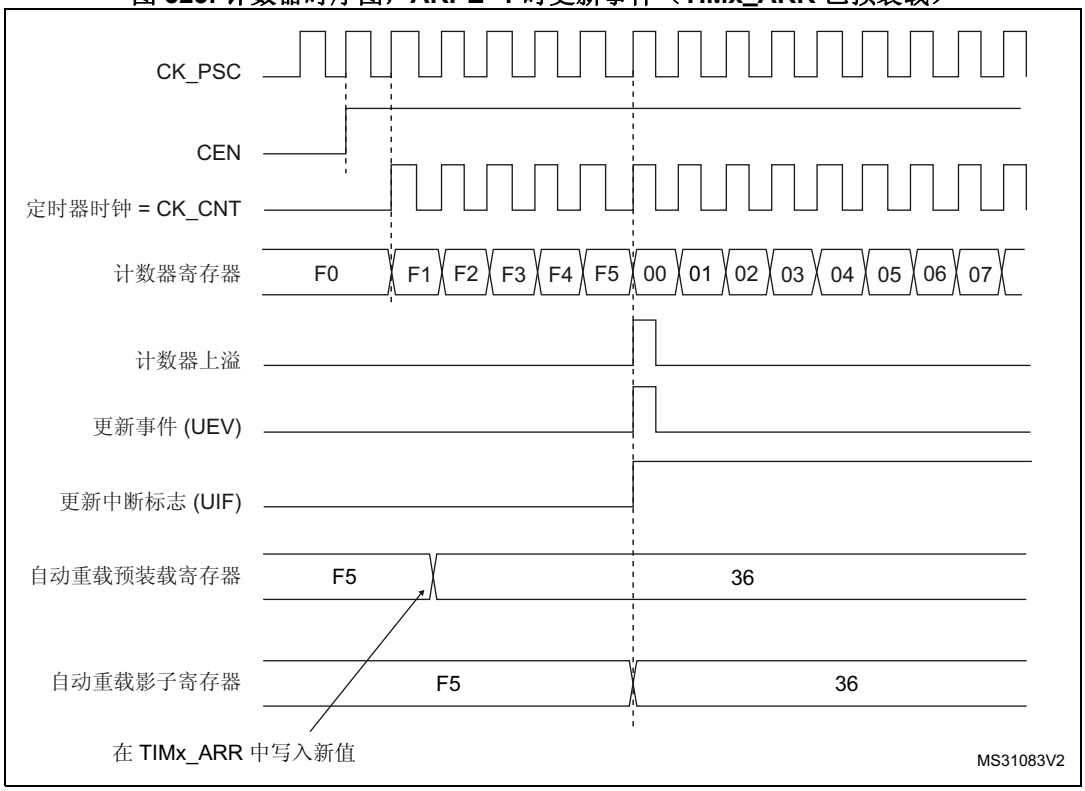


图 323. 计数器时序图，ARPE=1 时更新事件（TIMx_ARR 已预装载）



递减计数模式

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

每次发生计数器下溢时会生成更新事件，或将 TIMx_EGR 寄存器中的 UG 位置 1（通过软件或使用从模式控制器）也可以生成更新事件

通过软件将 TIMx_CR1 寄存器中的 UDIS 位置 1 可禁止 UEV 更新事件。这可避免向预装载寄存器写入新值时更新影子寄存器。在 UDIS 位写入 0 之前不会产生任何更新事件。不过，计数器会重新从当前自动重载值开始计数，而预分频器计数器则重新从 0 开始计数（但预分频比保持不变）。

此外，如果 TIMx_CR1 寄存器中的 URS 位（更新请求选择）已置 1，则将 UG 位置 1 会生成更新事件 UEV，但不会将 UIF 标志置 1（因此，不会发送任何中断或 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（TIMx_SR 寄存器中的 UIF 位）置 1（取决于 URS 位）：

- 预分频器的缓冲区中将重新装载预装载值（TIMx_PSC 寄存器的内容）。
- 自动重载有效寄存器将以预装载值（TIMx_ARR 寄存器的内容）进行更新。注意，ARR 寄存器更新在计数器重载之前被更新，因此下一个周期就是预期的值。

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

图 324. 计数器时序图，1 分频内部时钟

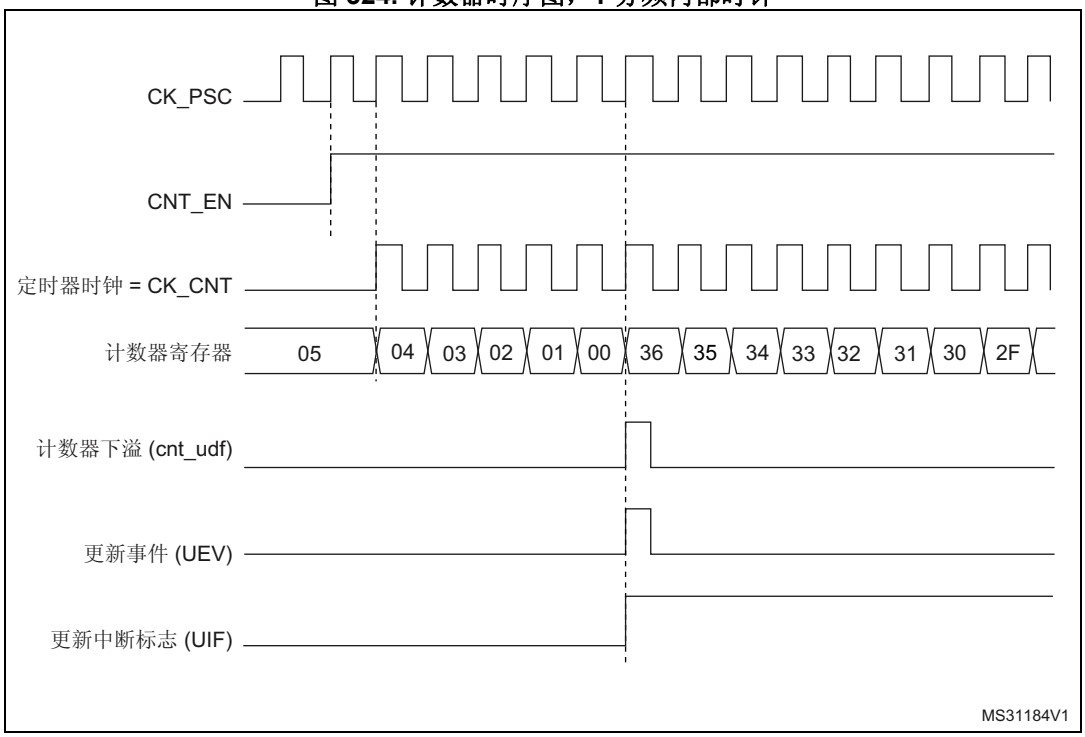


图 325. 计数器时序图，2 分频内部时钟

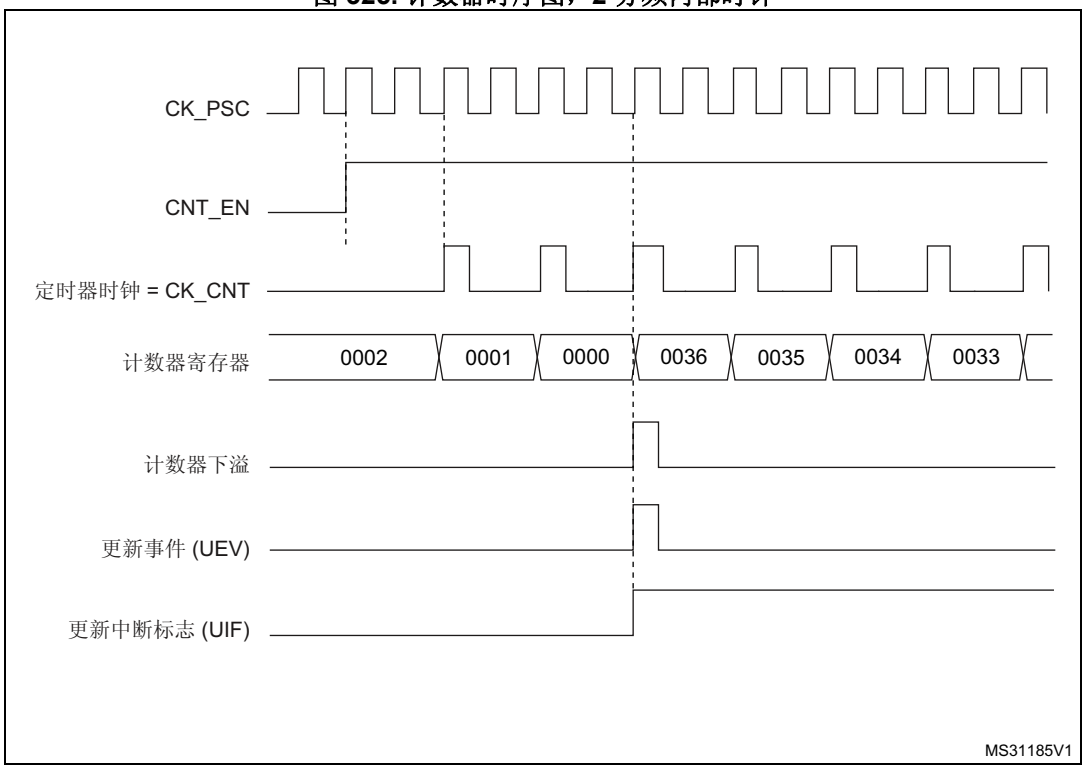


图 326. 计数器时序图，4 分频内部时钟

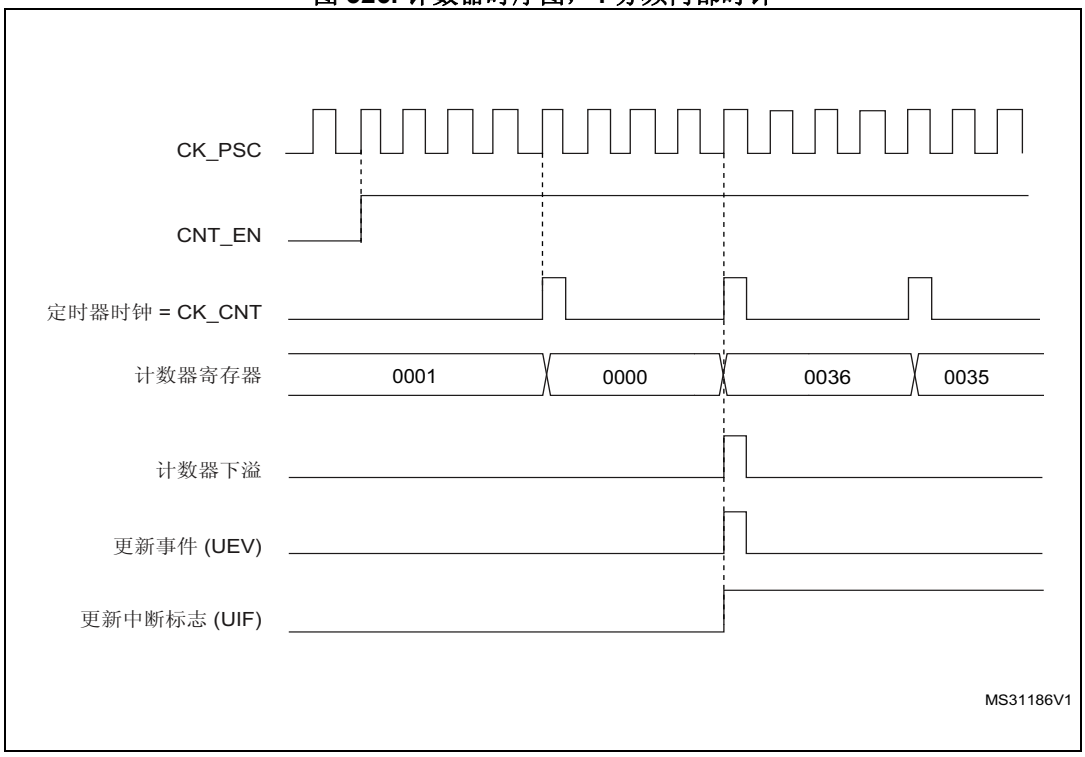


图 327. 计数器时序图，N 分频内部时钟

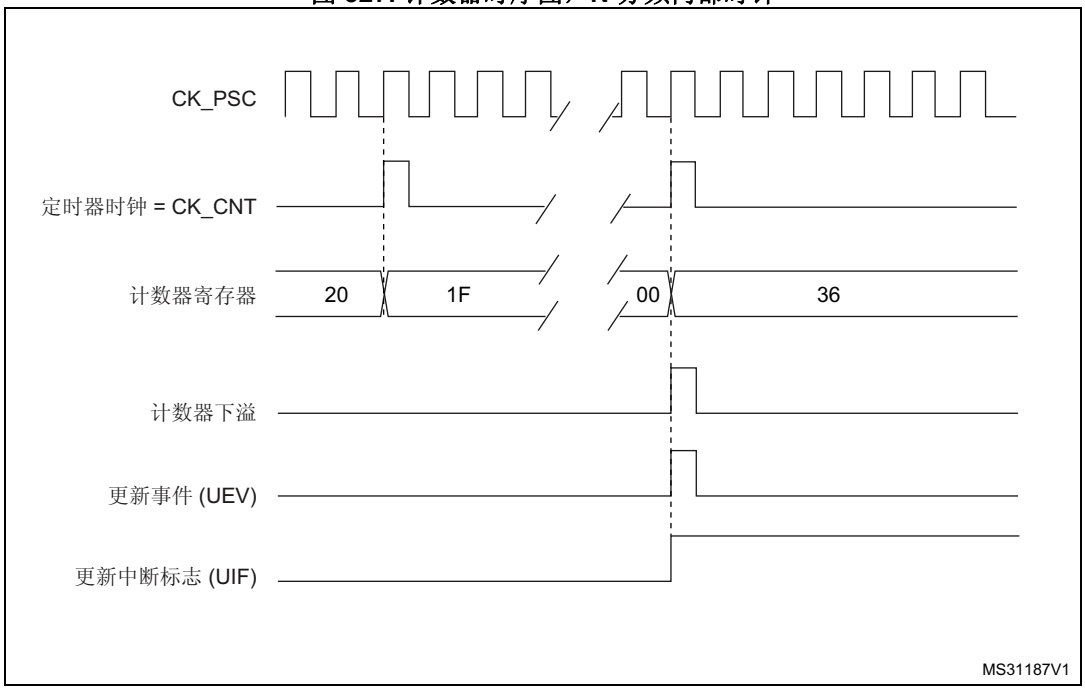
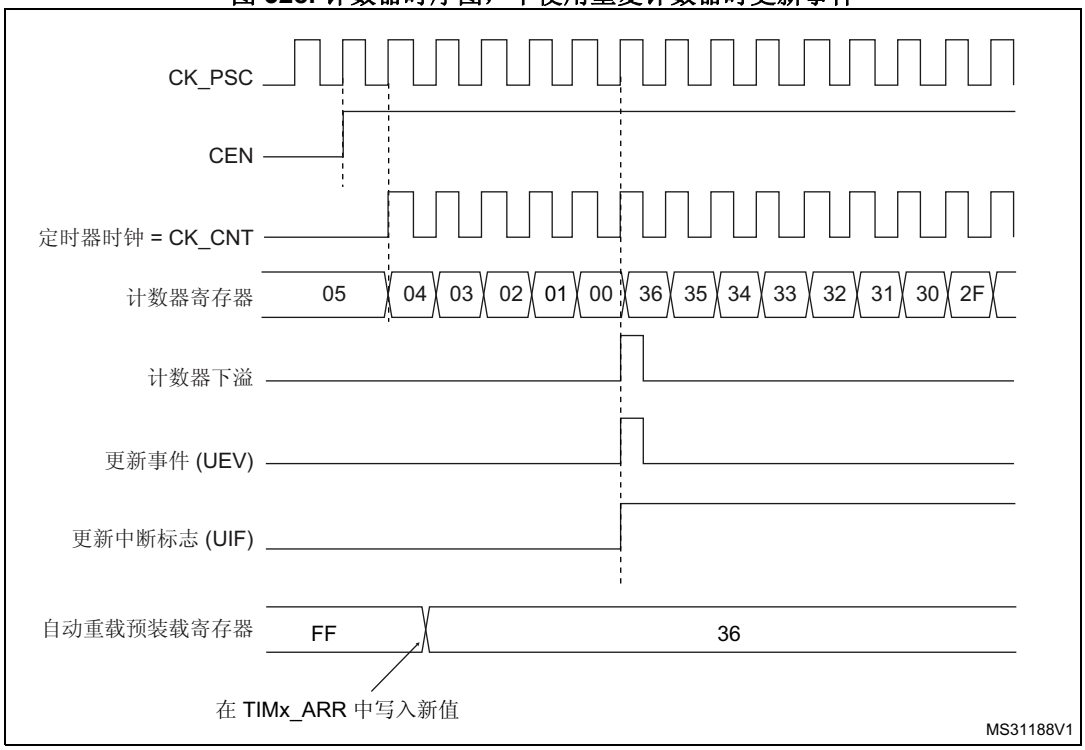


图 328. 计数器时序图，不使用重复计数器时更新事件



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（TIMx_ARR 寄存器的内容）– 1，生成计数器上溢事件；然后从自动重载值开始向下计数到 1 并生成计数器下溢事件。之后从 0 开始重新计数。

当 TIMx_CR1 寄存器中的 CMS 位不为“00”时，中心对齐模式有效。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（TIMx_CR1 寄存器中的 DIR 位）。而是由硬件更新并指示当前计数器方向。

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

通过软件将 TIMx_CR1 寄存器中的 UDIS 位置 1 可禁止 UEV 更新事件。这可避免向预装载寄存器写入新值时更新影子寄存器。在 UDIS 位写入 0 之前不会产生任何更新事件。不过，计数器仍会根据当前自动重载值进行递增和递减计数。

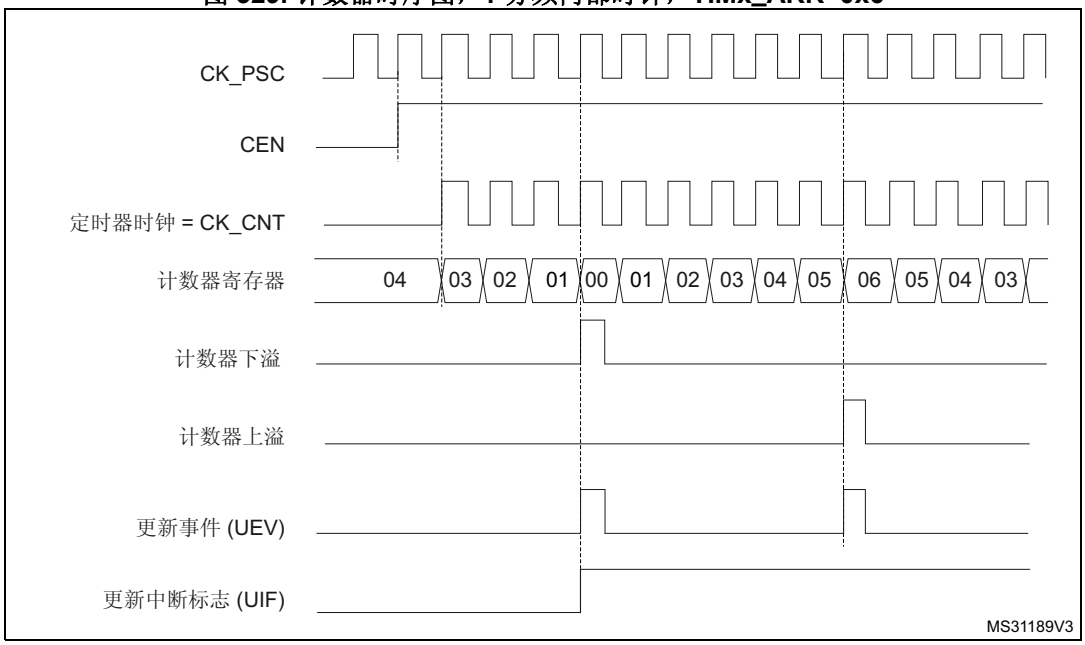
此外，如果 TIMx_CR1 寄存器中的 URS 位（更新请求选择）已置 1，则将 UG 位置 1 会生成更新事件 UEV，但不会将 UIF 标志置 1（因此，不会发送任何中断或 DMA 请求）。这样一来，如果在发生捕获事件时将计数器清零，将不会同时产生更新中断和捕获中断。

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

- 预分频器的缓冲区中将重新装载预装载值（TIMx_PSC 寄存器的内容）。
- 自动重载有效寄存器将以预装载值（TIMx_ARR 寄存器的内容）进行更新。注意，如果更新操作是由计数器上溢触发的，则 ARR 寄存器在计数器重载之前更新，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 329. 计数器时序图，1 分频内部时钟，TIMx_ARR=0x6



1. 此处使用中心对齐模式 1（有关详细信息，请参见*第 1185 页的第 35.4.1 节：TIMx 控制寄存器 1 (TIMx_CR1)*）。

图 330. 计数器时序图，2 分频内部时钟

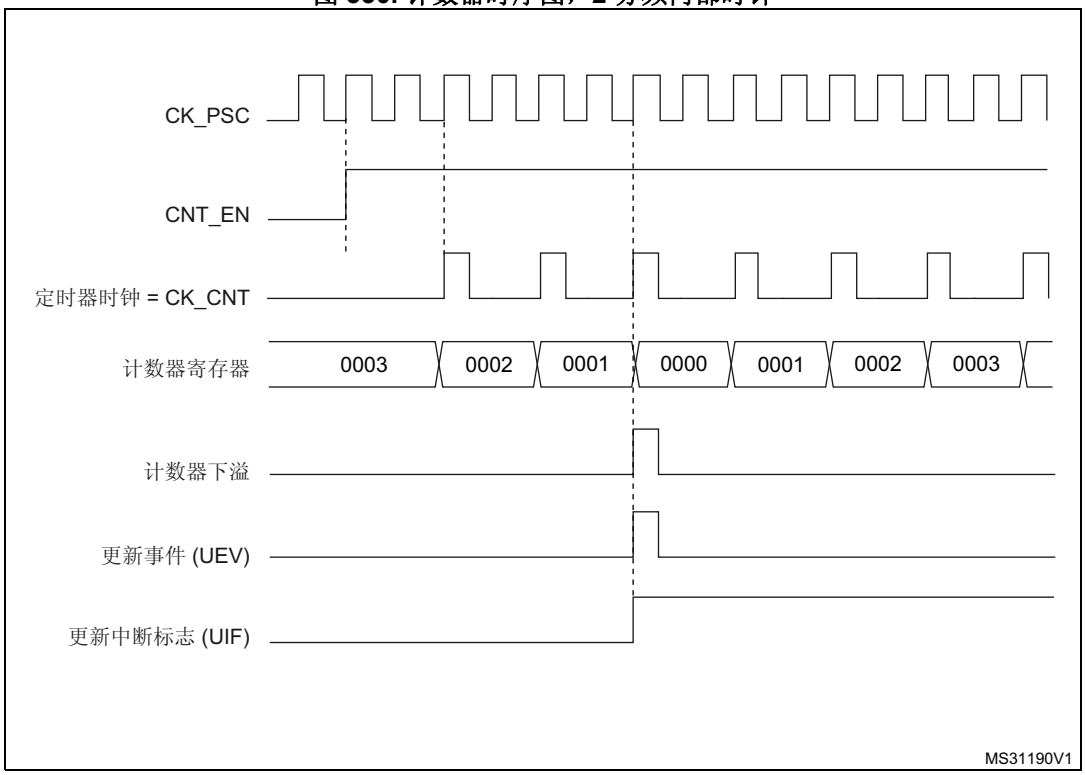
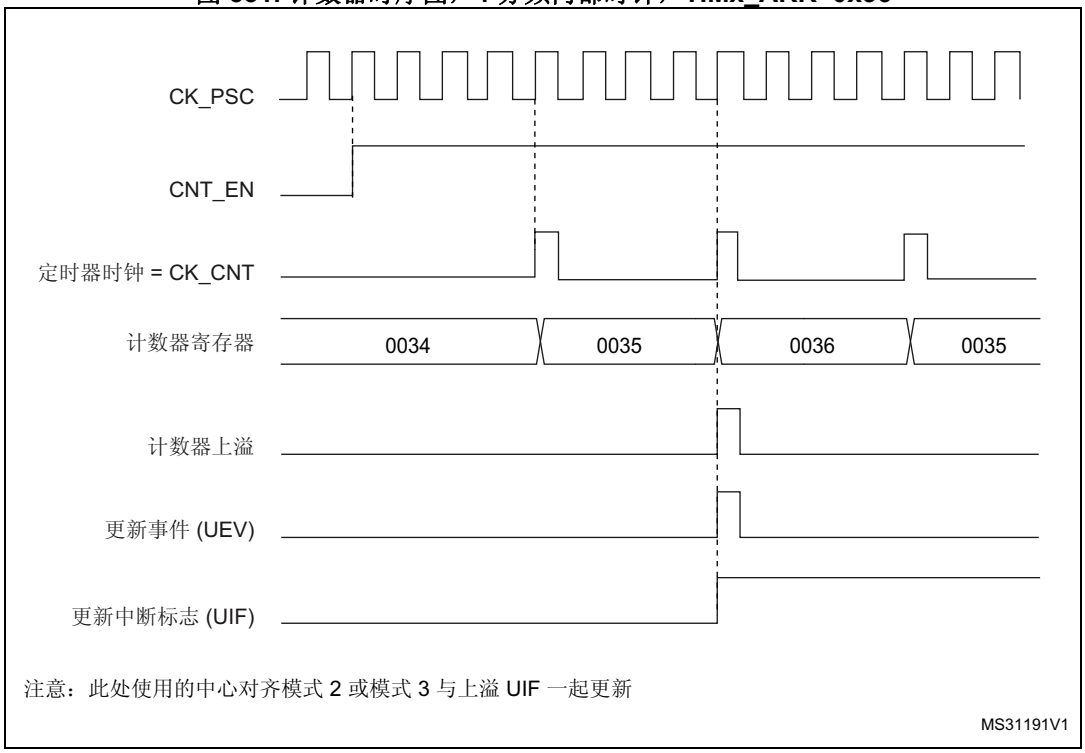


图 331. 计数器时序图，4 分频内部时钟，TIMx_ARR=0x36



1. 中心对齐模式 2 或模式 3 与上溢 UIF 结合使用。

图 332. 计数器时序图，N 分频内部时钟

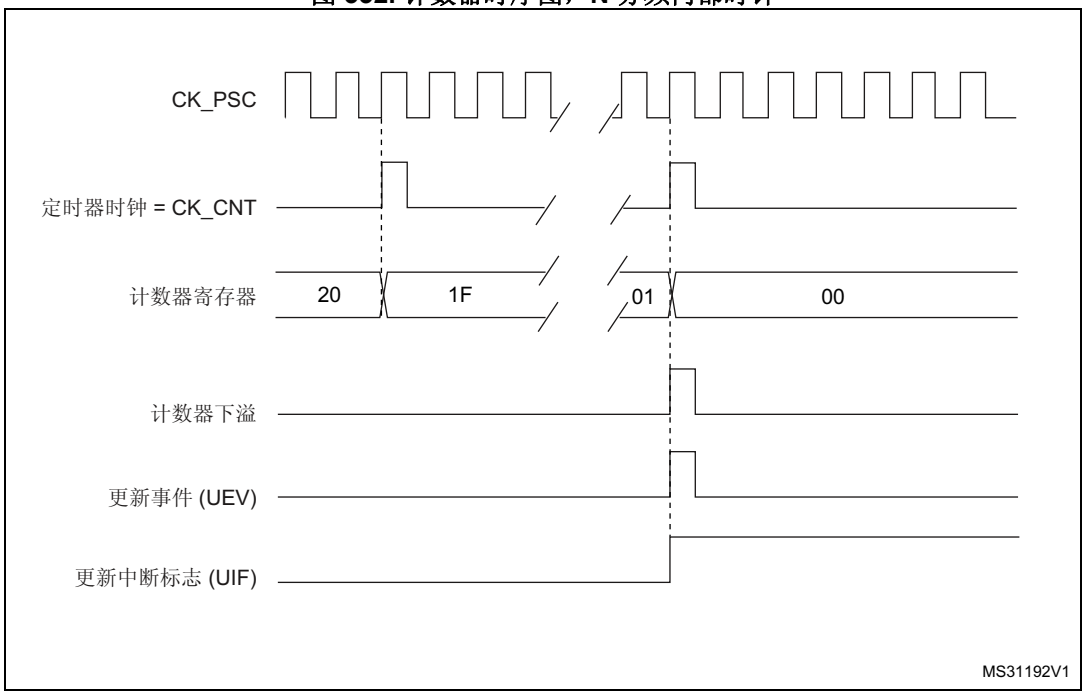


图 333. 计数器时序图，ARPE=1 时的更新事件（计数器下溢）

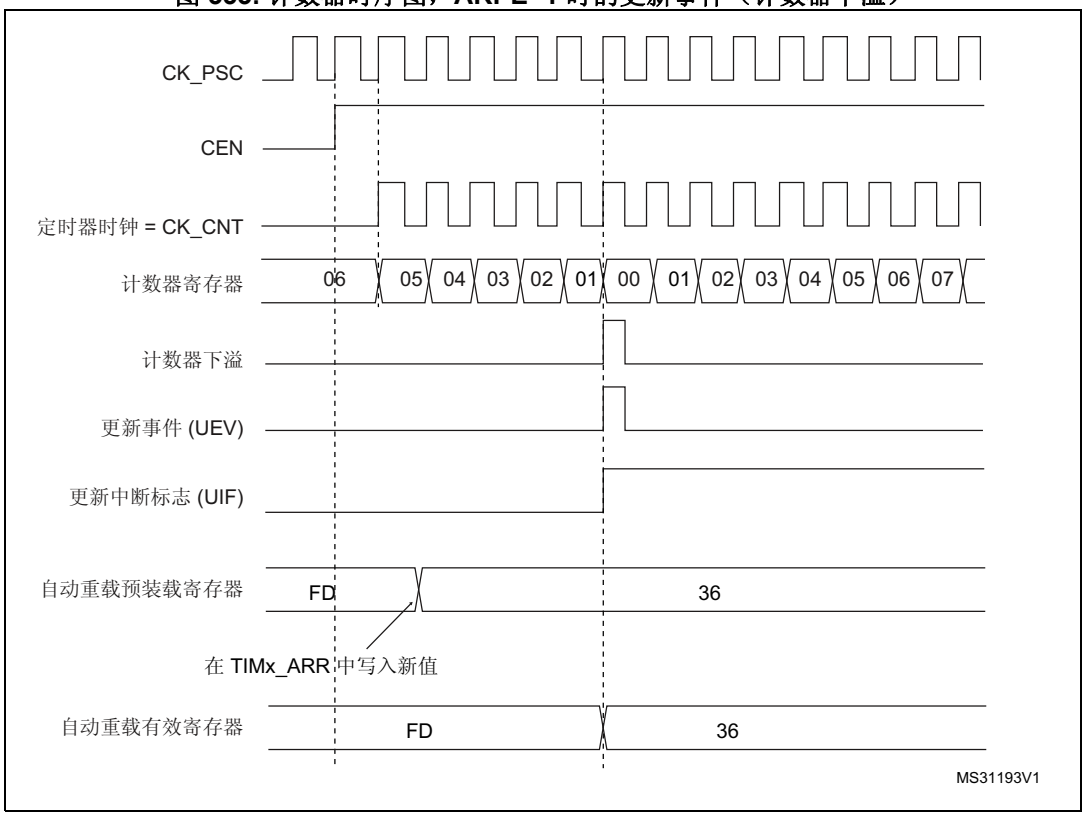
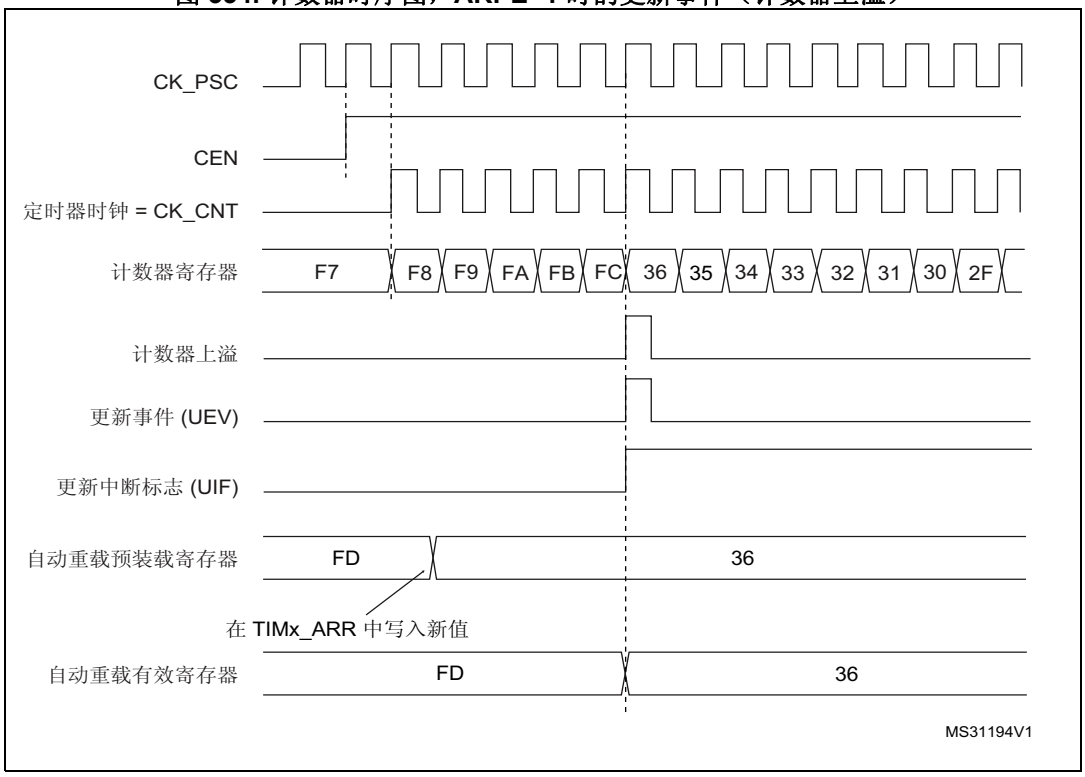


图 334. 计数器时序图，ARPE=1 时的更新事件（计数器上溢）



35.3.3 时钟选择

计数器时钟可由下列时钟源提供：

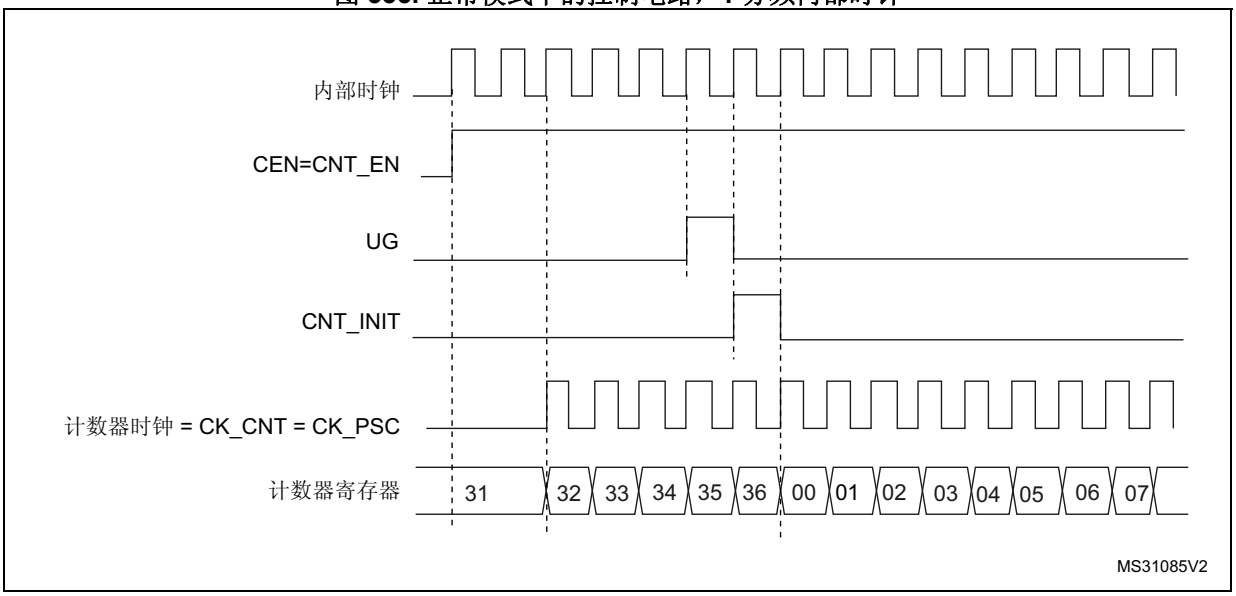
- 内部时钟 (CK_INT)
- 外部时钟模式 1：外部输入引脚 (TIx)
- 外部时钟模式 2：外部触发输入 (ETR)
- 外部触发输入 (ITRx)：使用一个定时器作为另一定时器的预分频器，例如，可将定时器 13 配置为定时器 2 的预分频器。更多详细信息，请参见*第 1179 页的将一个定时器用作另一个定时器的预分频器*。

内部时钟源 (CK_INT)

如果禁止从模式控制器（TIMx_SMCR 寄存器中 SMS=000），则 CEN 位、DIR 位（TIMx_CR1 寄存器中）和 UG 位（TIMx_EGR 寄存器中）为实际控制位，并且只能通过软件进行更改（UG 除外，仍自动清零）。当对 CEN 位写入 1 时，预分频器的时钟就由内部时钟 CK_INT 提供。

图 335 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）。

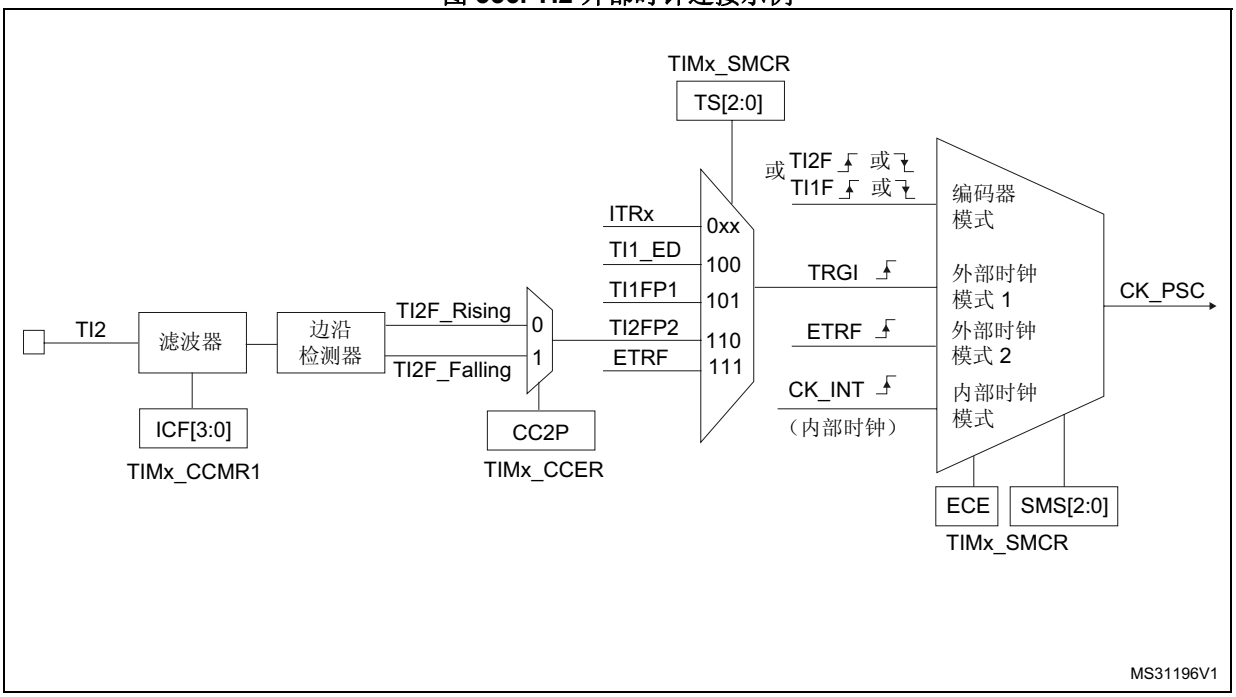
图 335. 正常模式下的控制电路，1 分频内部时钟



外部时钟源模式 1

当 TIMx_SMCR 寄存器中的 SMS=111 时，可选择此模式。计数器可在选定的输入信号上出现上升沿或下降沿时计数。

图 336. 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 TIMx_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 TIMx_CCMR1 寄存器中写入 IC2F[3:0] 位来配置输入滤波带宽（如果不需要任何滤波器，请保持 IC2F=0000）。

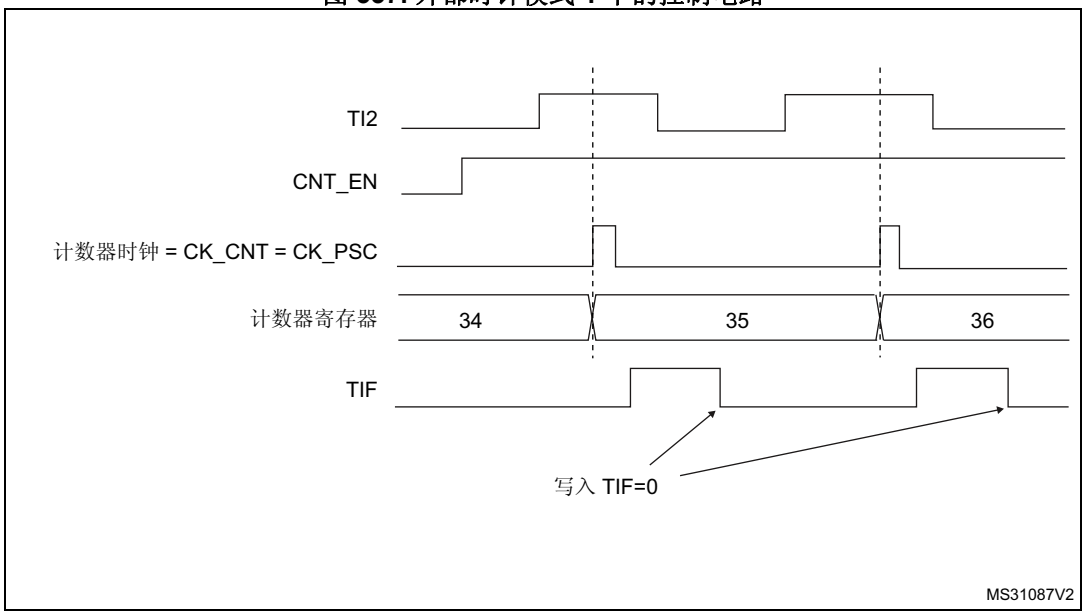
注： *由于捕获预分频器不用于触发操作，因此无需对其进行配置。*

3. 通过在 TIMx_CCER 寄存器中写入 CC2P=0 和 CC2NP=0 来选择上升沿极性。
4. 通过在 TIMx_SMCR 寄存器中写入 SMS=111，使定时器在外部时钟模式 1 下工作。
5. 通过在 TIMx_SMCR 寄存器中写入 TS=110 来选择 TI2 作为输入源。
6. 通过在 TIMx_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 337. 外部时钟模式 1 下的控制电路



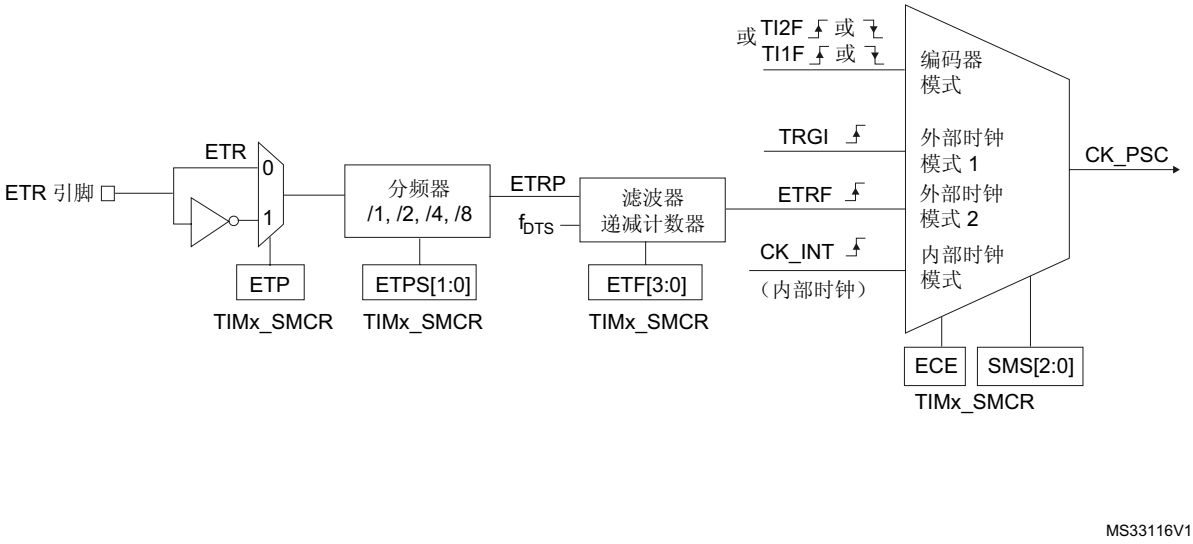
外部时钟源模式 2

通过在 TIMx_SMCR 寄存器中写入 ECE=1 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 338 简要介绍了外部触发输入模块。

图 338. 外部触发输入模块



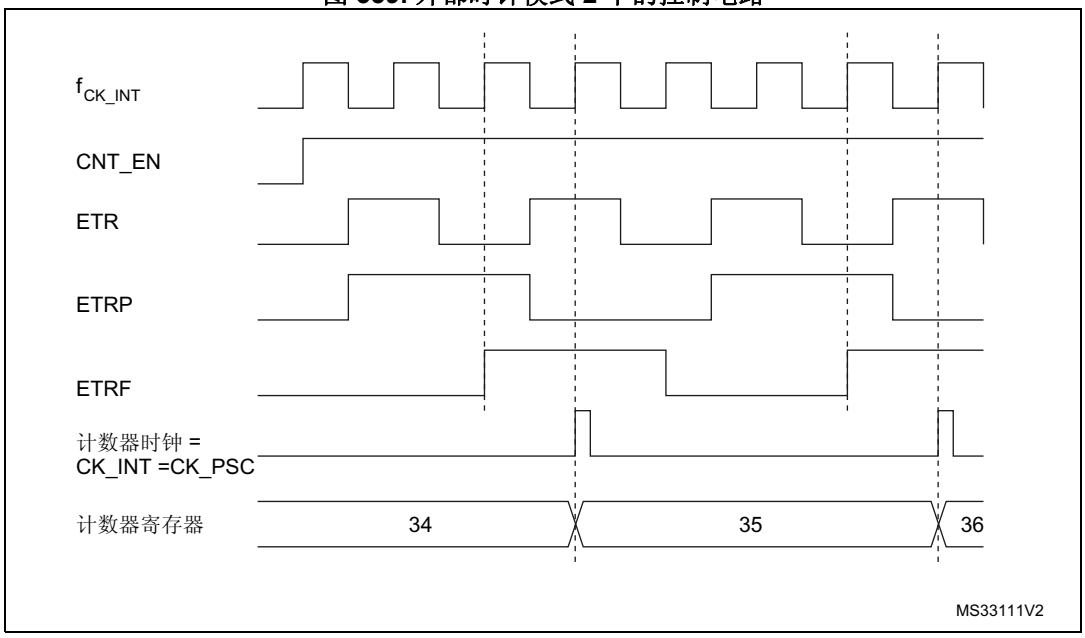
例如，要使递增计数器在 ETR 每出现 2 个上升沿时计数，请执行以下步骤：

1. 由于此例中不需滤波器，因此在 TIMx_SMCR 寄存器中写入 ETF[3:0]=0000。
2. 通过在 TIMx_SMCR 寄存器中写入 ETPS[1:0]=01 来设置预分频器。
3. 通过在 TIMx_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 TIMx_SMCR 寄存器中写入 ECE=1 来使能外部时钟模式 2。
5. 通过在 TIMx_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 339. 外部时钟模式 2 下的控制电路



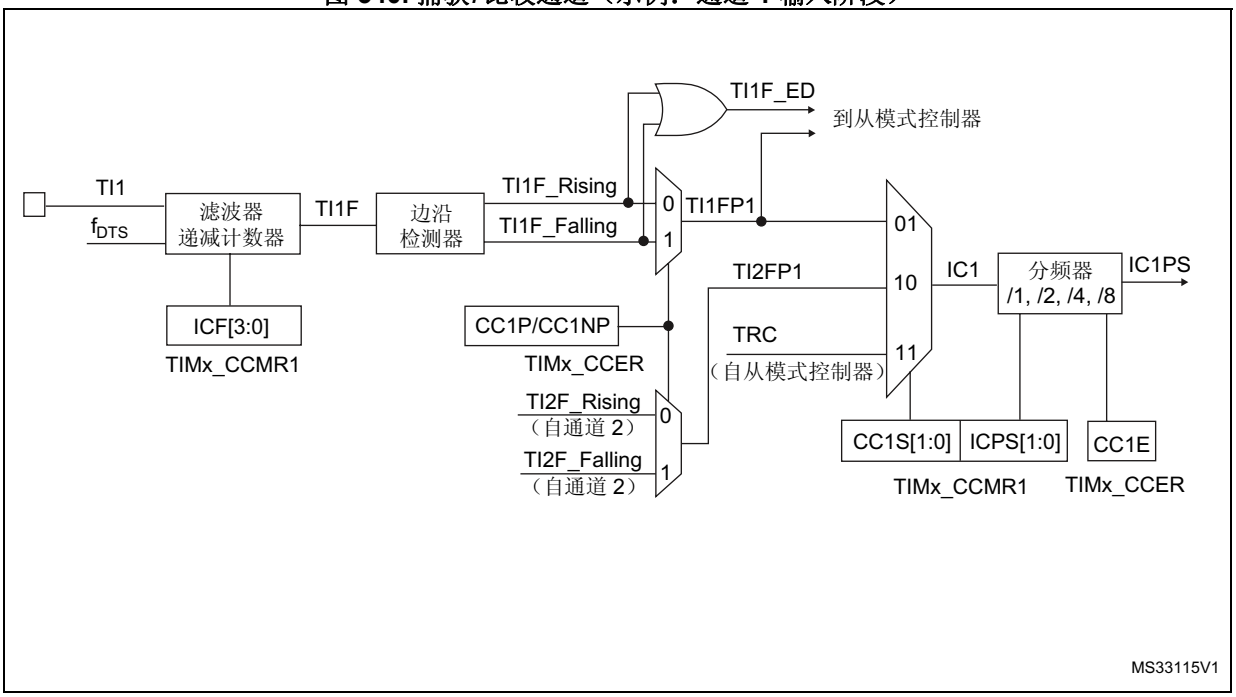
35.3.4 捕获/比较通道

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。

下图简要介绍了一路捕获/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 340. 捕获/比较通道（示例：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）。链的末端决定最终输出信号的极性。

图 341. 捕获/比较通道 1 主电路

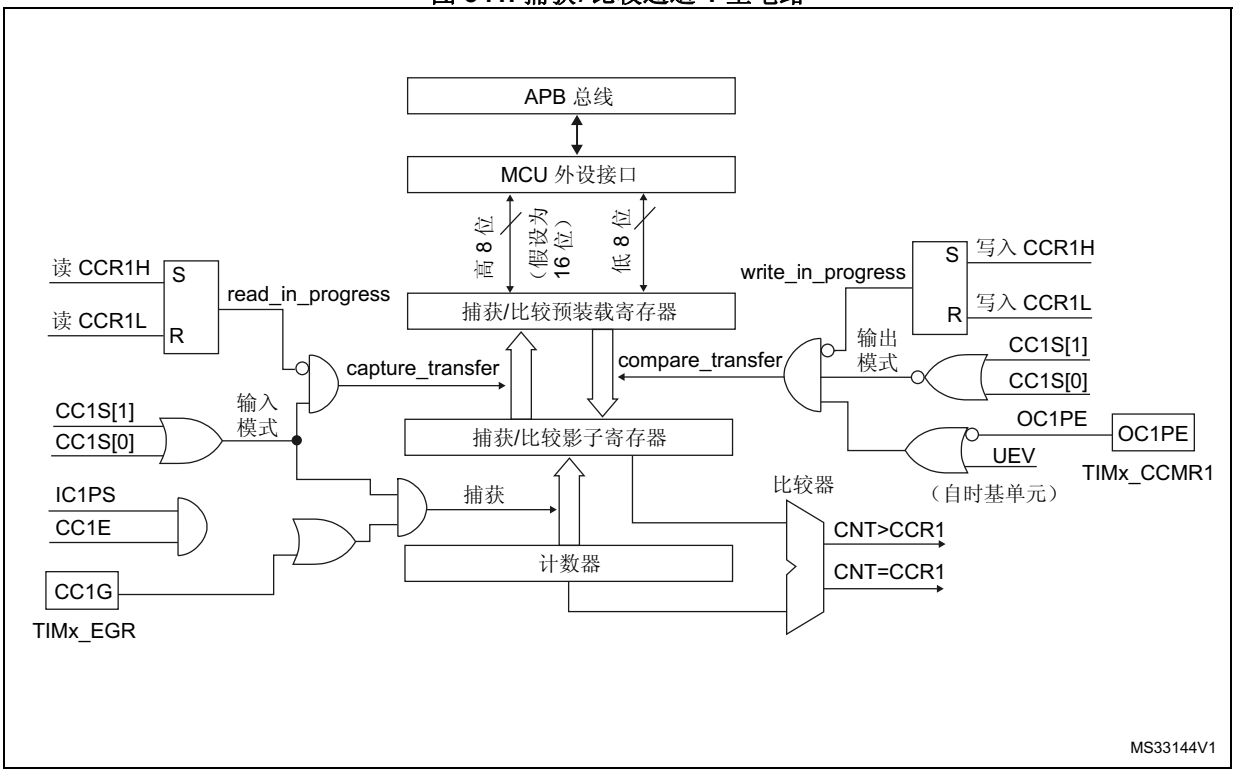
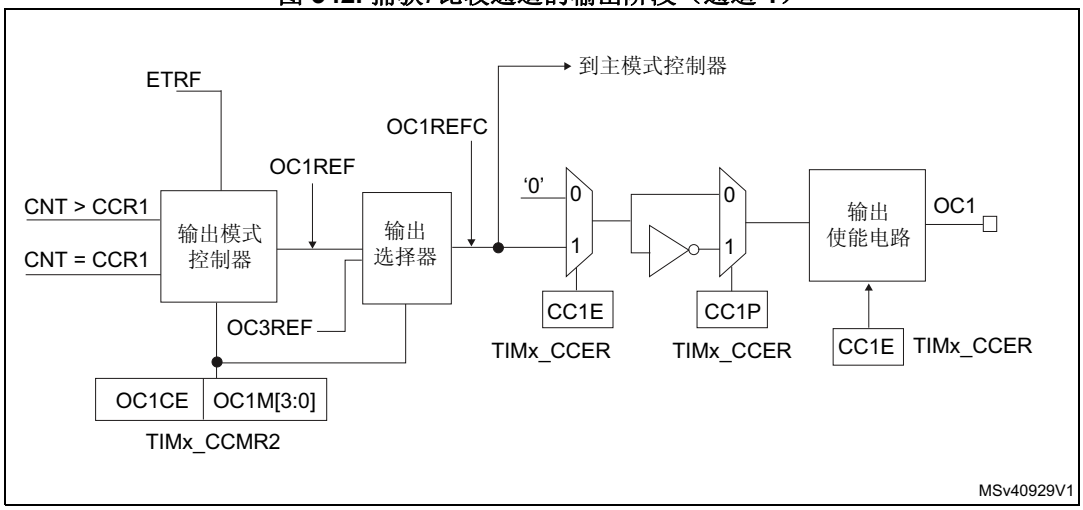


图 342. 捕获/比较通道的输出阶段（通道 1）



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。

35.3.5 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 (TIMx_CCRx) 来锁存计数器的值。发生捕获事件时，会将相应的 CCXIF 标志（TIMx_SR 寄存器）置 1，并可发送中断或 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（TIMx_SR 寄存器）置 1。可通过软件方法向 CCxIF 写入 0 来给 CCxIF 清零，或读取存储在 TIMx_CCRx 寄存器中的已捕获数据。向 CCxOF 写入 0 后会将其清零。

以下示例说明了如何在 TI1 输入出现上升沿时将计数器的值捕获到 TIMx_CCR1 中。具体操作步骤如下：

1. 选择有效输入：TIMx_CCR1 必须连接到 TI1 输入，因此向 TIMx_CCMR1 寄存器中的 CC1S 位写入 01。只要 CC1S 不等于 00，就会将通道配置为输入模式，并且 TIMx_CCR1 寄存器将处于只读状态。
2. 根据连接到定时器的信号，对所需的输入滤波带宽进行编程（如果输入为 TIx 之一，则对 TIMx_CCMRx 寄存器中的 ICxF 位进行编程）。假设信号边沿变化时，输入信号最多在 5 个内部时钟周期内发生抖动。因此，我们必须将滤波带宽设置为大于 5 个内部时钟周期。在检测到 8 个具有新电平的连续采样（以 f_{DTS} 频率采样）后，可以确认 TI1 上的跳变沿。然后向 TIMx_CCMR1 寄存器中的 IC1F 位写入 0011。
3. 通过向 TIMx_CCER 寄存器中的 CC1P 位和 CC1NP 位写入 000，选择 TI1 通道的有效跳变沿（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 TIMx_CCMR1 寄存器中的 IC1PS 位写入 00）。
5. 通过将 TIMx_CCER 寄存器中的 CC1E 位置 1，允许将计数器的值捕获到捕获寄存器中。
6. 如果需要，可通过将 TIMx_DIER 寄存器中的 CC1IE 位置 1 来使能相关中断请求，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，TIMx_CCR1 寄存器会获取计数器的值。
- 将 CC1IF 标志置 1（中断标志）。如果至少发生了两次连续捕获，但 CC1IF 标志未被清零，这样 CC1OF 重复捕获标志会被置 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，建议在读出重复捕获标志之前读取数据。这样可避免丢失在读取重复捕获标志之后与读取数据之前可能出现的重复捕获信息。

注： *通过软件将 TIMx_EGR 寄存器中的相应 CCxG 位置 1 可生成 IC 中断和/或 DMA 请求。*

35.3.6 PWM 输入模式

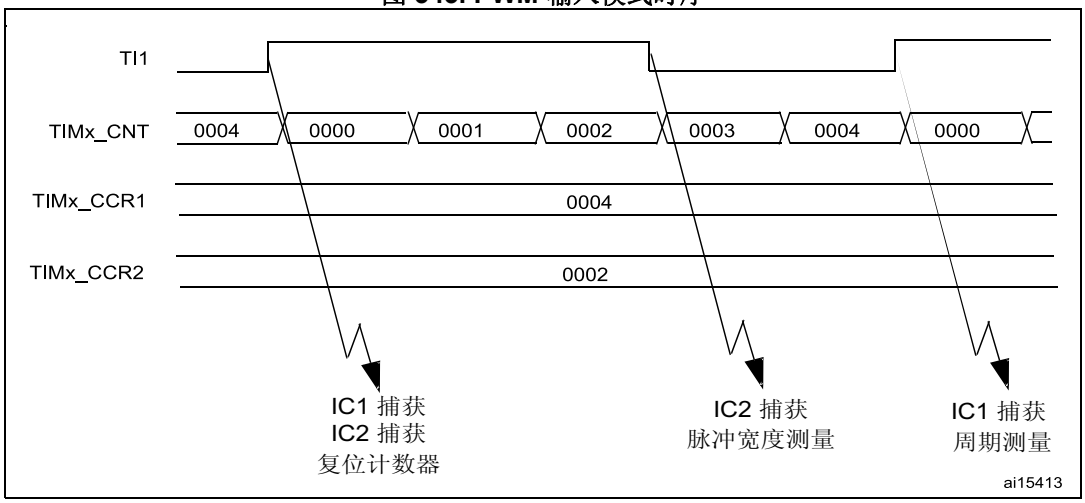
此模式是输入捕获模式的一个特例。其实现步骤与输入捕获模式基本相同，仅存在以下不同之处：

- 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 TIMx_CCR1 寄存器中）和占空比（位于 TIMx_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 TIMx_CCR1 的有效输入：向 TIMx_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 TIMx_CCR1 中的捕获和计数器清零）：向 CC1P 位和 CC1NP 位写入“0”（上升沿有效）。
3. 选择 TIMx_CCR2 的有效输入：向 TIMx_CCMR1 寄存器中的 CC2S 写入 10 （选择 TI1）。
4. 选择 TI1FP2 的有效极性（用于 TIMx_CCR2 中的捕获）：向 CC2P 位写入“1”，向 CC2NP 位写入“0”（下降沿有效）。
5. 选择有效触发输入：向 TIMx_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 TIMx_SMCR 寄存器中的 SMS 位写入 100。
7. 使能捕获：向 TIMx_CCER 寄存器中的 CC1E 位和 CC2E 位写入“1”。

图 343. PWM 输入模式时序



1. PWM 输入模式只能与 TIMx_CH1/TIMx_CH2 信号配合使用，因为只有 TI1FP1 和 TI2FP2 与从模式控制器相连。

35.3.7 强制输出模式

在输出模式（TIMx_CCMRx 寄存器中的 CCxS 位 = 00）下，可直接由软件将每个输出比较信号（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 TIMx_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 TIMx_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，TIMx_CCRx 影子寄存器与计数器之间的比较仍会执行，而且允许将标志置 1。因此可发送相应的中断和 DMA 请求。输出比较模式一节对此进行了介绍。

35.3.8 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（TIMx_CCMRx 寄存器中的 OCxM 位）和输出极性（TIMx_CCER 寄存器中的 CCxP 位）定义。匹配时，输出引脚既可保持其电平 (OCXM=000)，也可设置为有效电平 (OCXM=001)、无效电平 (OCXM=010) 或进行翻转 (OCxM=011)。
- 将中断状态寄存器中的标志置 1（TIMx_SR 寄存器中的 CCxIF 位）。
- 如果相应中断使能位（TIMx_DIER 寄存器中的 CCXIE 位）置 1，将生成中断。
- 如果相应使能位（TIMx_DIER 寄存器的 CCxDE 位，TIMx_CR2 寄存器的 CCDS 位，用来选择 DMA 请求）置 1，将发送 DMA 请求。

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

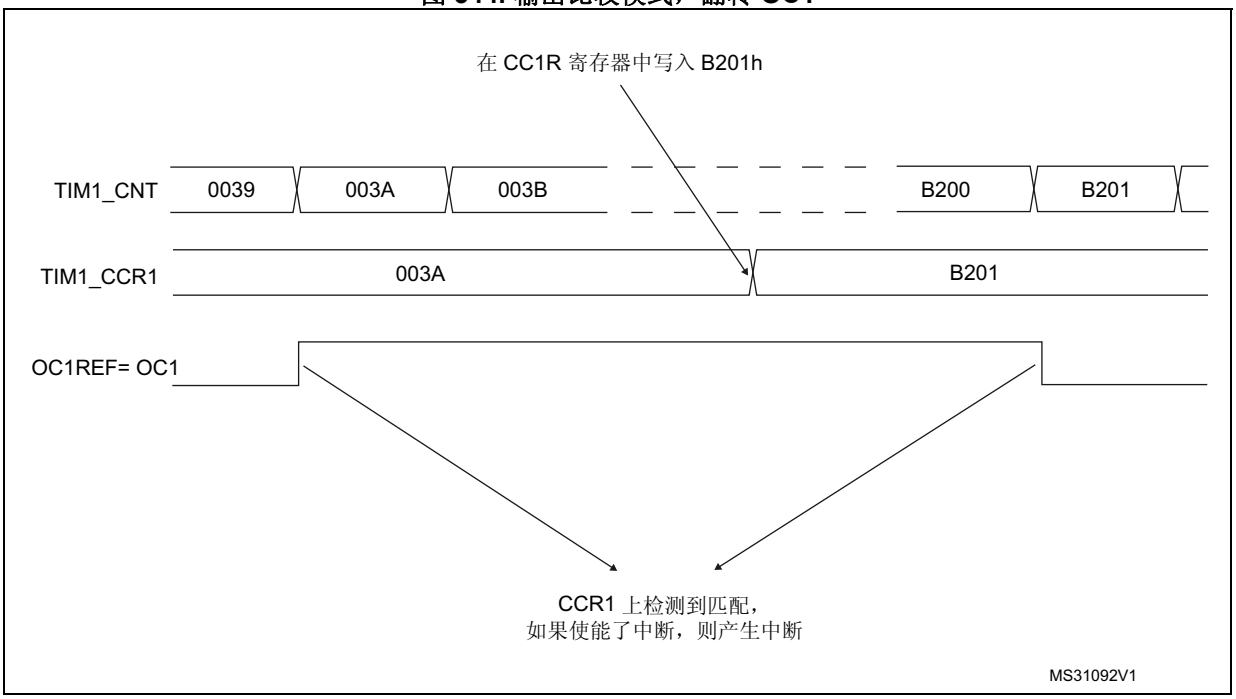
在输出比较模式下，更新事件 UEV 对 ocxref 和 OCx 输出毫无影响。同步的精度可以达到计数器的一个计数周期。输出比较模式也可用于输出单脉冲（在单脉冲模式下）。

步骤

1. 选择计数器时钟（内部、外部、预分频器）。
2. 在 TIMx_ARR 和 TIMx_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 TIMx_CR1 寄存器中的 CEN 位置 1 来使能计数器。

可随时通过软件更新 TIMx_CCRx 寄存器以控制输出波形，前提是未使能预装载寄存器（OCxPE=0，否则仅当发生下一个更新事件 UEV 时，才会更新 TIMx_CCRx 影子寄存器）。*图 344* 给出了一个示例。

图 344. 输出比较模式，翻转 OC1



35.3.9 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 TIMx_ARR 寄存器值决定，其占空比则由 TIMx_CCRx 寄存器值决定。

通过向 TIMx_CCMRx 寄存器中的 OCxM 位写入 110（PWM 模式 1）或 111（PWM 模式 2），可以独立选择各通道（每个 OCx 输出对应一个 PWM）的 PWM 模式。必须通过将 TIMx_CCMRx 寄存器中的 OCxPE 位置 1 使能相应预装载寄存器，最后通过将 TIMx_CR1 寄存器中的 ARPE 位置 1 使能自动重载预装载寄存器（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 TIMx_EGR 寄存器中的 UG 位置 1 来初始化所有寄存器。

OCx 极性可通过软件来编程（使用 TIMx_CCER 寄存器的 CCxP 位）。可将其编程为高电平有效或低电平有效。OCx 输出通过将 TIMx_CCER 寄存器中的 CCxE 位置 1 来使能。有关详细信息，请参见 TIMx_CCERx 寄存器说明。

在 PWM 模式（1 或 2）下，TIMx_CNT 总是与 TIMx_CCRx 进行比较，以确定是 TIMx_CCRx ≤ TIMx_CNT 还是 TIMx_CNT ≤ TIMx_CCRx（取决于计数器计数方向）。不过，为符合 OCREF_CLR 功能（在下一个 PWM 周期之前，ETR 信号上的一个外部事件能够清除 OCREF），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（TIMx_CCMRx 寄存器中的 OCxM 位）从“冻结”配置（不进行比较，OCxM=“000”）切换为任一 PWM 模式（OCxM=“110”或“111”）。

定时器运行期间，可以通过软件强制 PWM 输出。

根据 TIMx_CR1 寄存器中的 CMS 位状态，定时器能够产生边沿对齐模式或中心对齐模式的 PWM 信号。

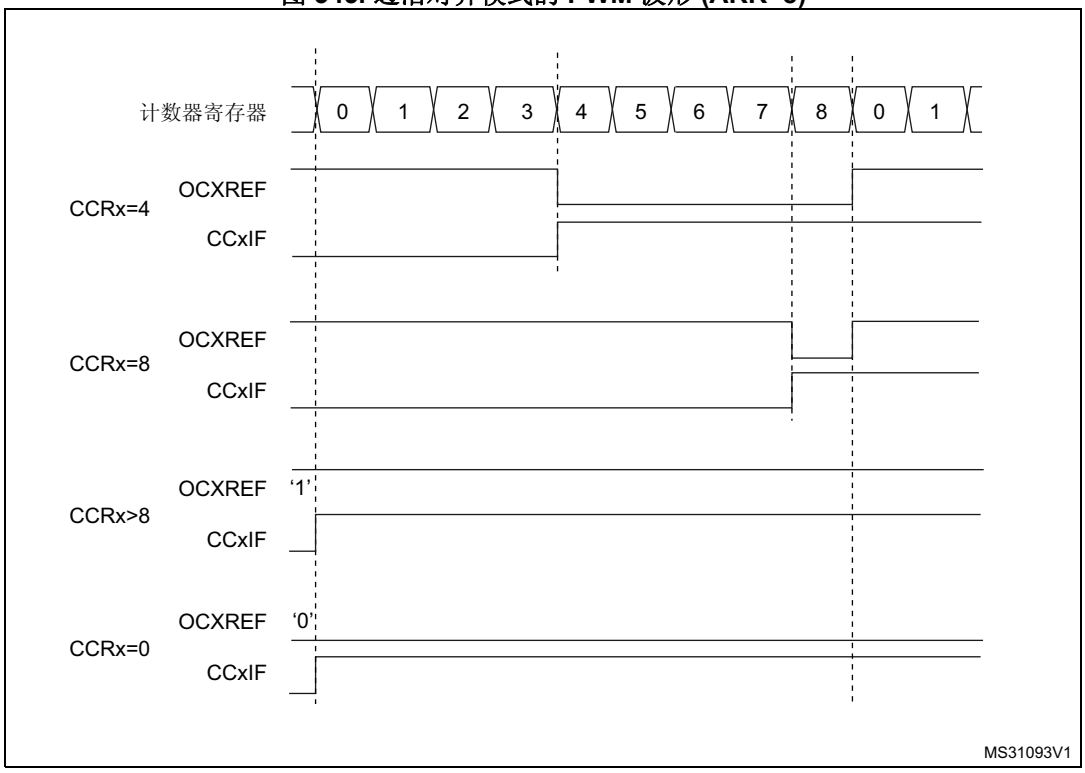
PWM 边沿对齐模式

递增计数配置

当 TIMx_CR1 寄存器中的 DIR 位为低时执行递增计数。请参见*第 1145 页的递增计数模式*。

以下以 PWM 模式 1 为例。只要 TIMx_CNT < TIMx_CCRx，PWM 参考信号 OCxREF 便为高电平，否则为低电平。如果 TIMx_CCRx 中的比较值大于自动重载值（TIMx_ARR 中），则 OCxREF 保持为“1”。如果比较值为 0，则 OCxREF 保持为“0”。*图 345* 举例介绍边沿对齐模式的一些 PWM 波形 (TIMx_ARR=8)。

图 345. 边沿对齐模式的 PWM 波形 (ARR=8)



递减计数配置

当 TIMx_CR1 寄存器中的 DIR 位为高时执行递减计数。请参见*第 1148 页的递减计数模式*。

在 PWM 模式 1 下，只要 TIMx_CNT>TIMx_CCRx，参考信号 ocxref 便为低电平，否则为高电平。如果 TIMx_CCRx 中的比较值大于 TIMx_ARR 中的自动重载值，则 ocxref 保持为 100%。此模式下不可能产生 PWM。

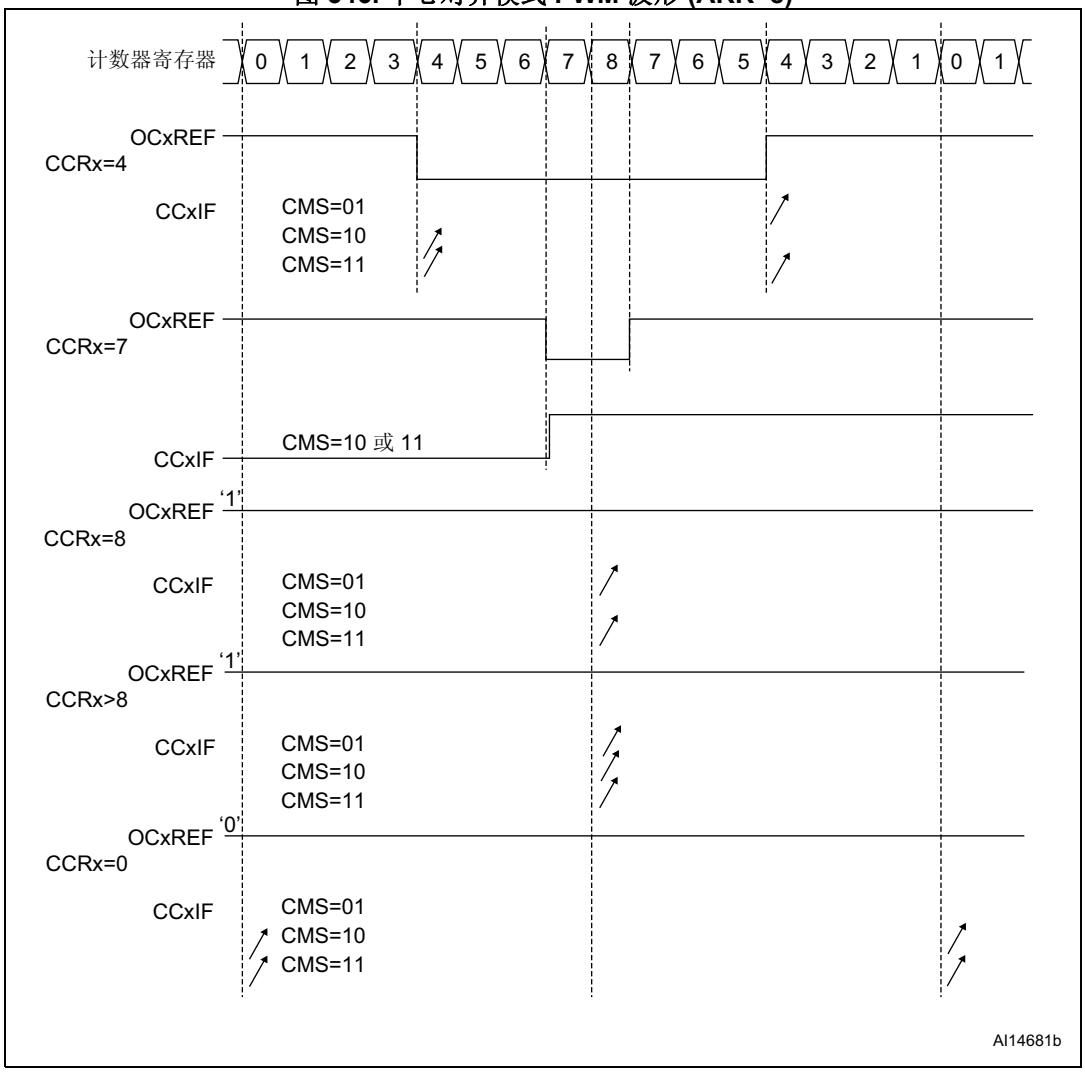
PWM 中心对齐模式

当 TIMx_CR1 寄存器中的 CMS 位不为“00”时（其余所有配置对 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。TIMx_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*第 1151 页的中心对齐模式（递增/递减计数）*。

图 346 显示了中心对齐模式的 PWM 波形，在此例中：

- TIMx_ARR=8
- PWM 模式为 PWM 模式 1
- 在根据 TIMx_CR1 寄存器中 CMS=01 而选择的中心对齐模式 1 下，当计数器递减计数时，比较标志置 1

图 346. 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 TIMx_CR1 寄存器中 DIR 位的值进行递增或递减计数。此外，不得同时通过软件修改 DIR 和 CMS 位。
- 不建议在运行中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 (TIMx_CNT>TIMx_ARR)，计数方向不会更新。例如，如果计数器之前递增计数，则继续递增计数。
  - 如果向计数器写入 0 或 TIMx_ARR 的值，计数方向会更新，但不生成更新事件 UEV。
- 使用中心对齐模式最为保险的方法是：在启动计数器前通过软件生成更新（将 TIMx_EGR 寄存器中的 UG 位置 1），并且不要在计数器运行过程中对其执行写操作。

35.3.10 不对称 PWM 模式

在不对称模式下，生成的两个中心对齐 PWM 信号间允许存在可编程相移。频率由 TIMx_ARR 寄存器的值确定，而占空比和相移则由一对 TIMx_CCRx 寄存器确定。两个寄存器分别控制递增计数和递减计数期间的 PWM，这样每半个 PWM 周期便会调节一次 PWM：

- OC1REFC（或 OC2REFC）由 TIMx_CCR1 和 TIMx_CCR2 控制
- OC3REFC（或 OC4REFC）由 TIMx_CCR3 和 TIMx_CCR4 控制

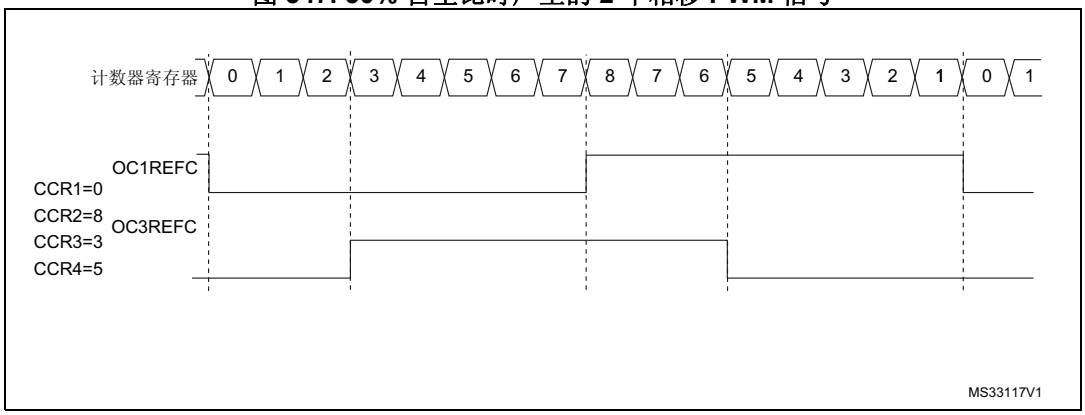
两个通道可以独立选择不对称 PWM 模式（每对 CCR 寄存器一个 OCx 输出），只需向 TIMx_CCMRx 寄存器的 OCxM 位写入“1110”（不对称 PWM 模式 1）或“1111”（不对称 PWM 模式 2）。

注： *出于兼容性原因，OCxM[3:0] 位域分为两部分，最高有效位与最低有效的 3 位不相邻。*

给定通道用作不对称 PWM 通道时，也可使用其辅助通道。例如，如果通道 1 上产生 OC1REFC 信号（不对称 PWM 模式 1），则由于不对称 PWM 模式 2 的原因，通道 2 上可输出 OC2REF 信号或 OC2REFC 信号。

图 347 显示了不对称 PWM 模式下可以产生的信号示例（通道 1 到通道 4 在不对称 PWM 模式 1 下配置）。

图 347. 50% 占空比时产生的 2 个相移 PWM 信号



35.3.11 组合 PWM 模式

在组合 PWM 模式下，生成的两个边沿或中心对齐 PWM 信号的各个脉冲间允许存在可编程延时和相移。频率由 TIMx_ARR 寄存器的值确定，而占空比和延时则由两个 TIMx_CCRx 寄存器确定。产生的信号 OCxREFC 由两个参考 PWM 的逻辑或运算或者逻辑与运算组合组成。

– OC1REFC（或 OC2REFC）由 TIMx_CCR1 和 TIMx_CCR2 控制

– OC3REFC（或 OC4REFC）由 TIMx_CCR3 和 TIMx_CCR4 控制

两个通道可以独立选择组合 PWM 模式（每对 CCR 寄存器一个 OCx 输出），只需向 TIMx_CCMRx 寄存器的 OCxM 位写入“1100”（组合 PWM 模式 1）或“1101”（组合 PWM 模式 2）。

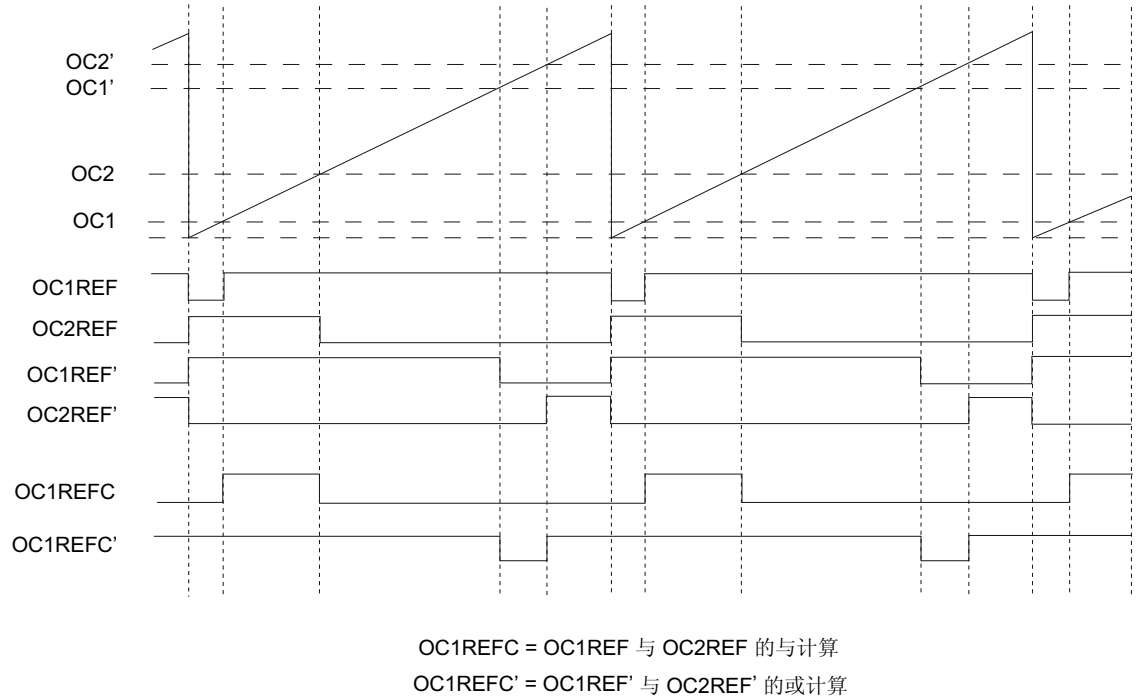
当给定通道用作组合 PWM 通道时，其辅助通道必须在相反的 PWM 模式下配置（例如，一个通道在组合 PWM 模式 1 下配置，另一个通道在组合 PWM 模式 2 下配置）。

注： *出于兼容性原因，OCxM[3:0] 位域分为两部分，最高有效位与最低有效的 3 位不相邻。*

图 348 显示了不对称 PWM 模式下可以产生的信号示例，通过以下配置可获得这些信号：

- 通道 1 在组合 PWM 模式 2 下配置
- 通道 2 在 PWM 模式 1 下配置
- 通道 3 在组合 PWM 模式 2 下配置
- 通道 4 在 PWM 模式 1 下配置

图 348. 通道 1 和通道 3 上的组合 PWM 模式



35.3.12 发生外部事件时清除 OCxREF 信号

对于给定通道，在 ocref_clr_int 输入上施加高电平（相应 TIMx_CCMRx 寄存器中的 OCxCE 使能位置 1），可将 OCxREF 信号清零。OCxREF 信号将保持低电平，直到发生下一更新事件 (UEV)。该功能只能在输出比较模式和 PWM 模式下使用。在强制模式下不起作用。

ocref_clr_int 连接到 ETRF 信号（滤波后的 ETR）。

对于给定通道，在 ETRF 输入施加高电平（相应 TIMx_CCMRx 寄存器中的 OCxCE 使能位置 1），可将 OCxREF 信号复位。OCxREF 信号将保持低电平，直到发生下一更新事件 (UEV)。

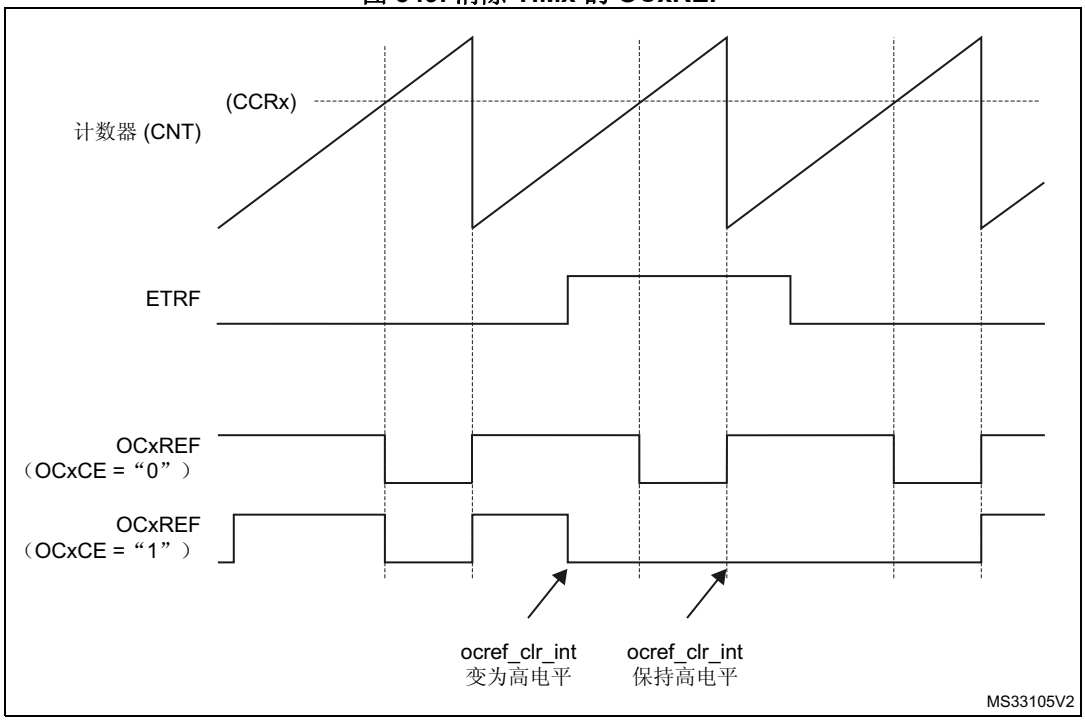
该功能只能在输出比较模式和 PWM 模式下使用。在强制模式下不起作用。

例如，OCxREF 信号可以连接到比较器的输出，用于控制电流。此时，ETR 必须配置如下：

1. 必须关闭外部触发预分频器：TIMx_SMCR 寄存器中的 ETPS[1:0] 位清为 00。
2. 必须禁止外部时钟模式 2：TIM1_SMCR 寄存器中的 ECE 位清为 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 349 对比了不同使能位 OCxCE 值的情况，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 TIMx 编程为 PWM 模式。

图 349. 清除 TIMx 的 OCxREF



注： *如果 PWM 的占空比为 100% (CCRx>ARR)，则下次计数器上溢时会再次使能 OCxREF。*

35.3.13 单脉冲模式

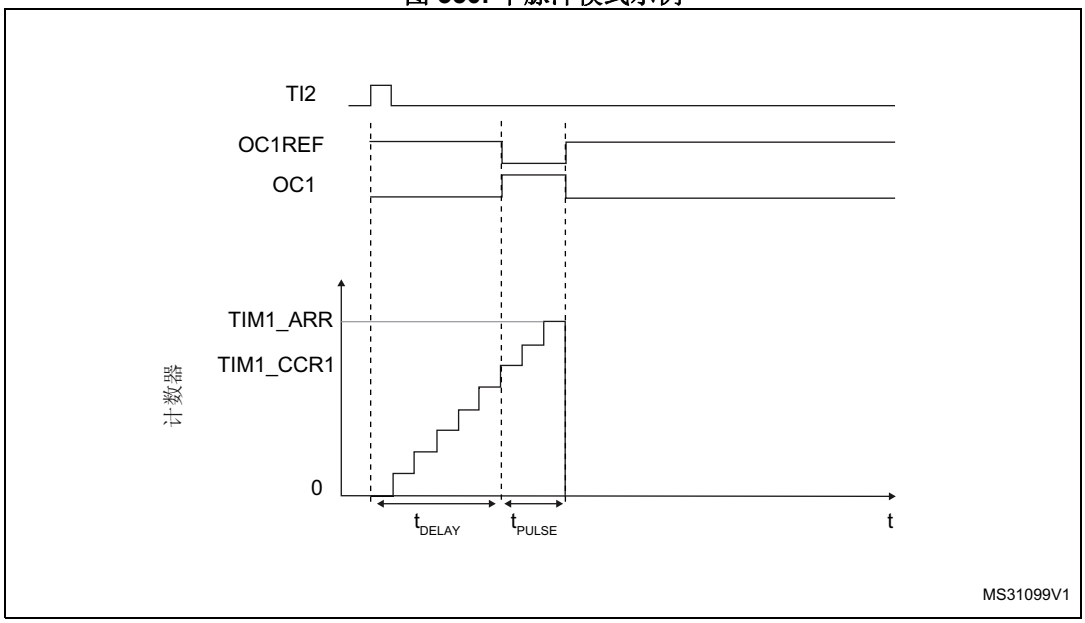
单脉冲模式 (OPM) 是上述模式的一个特例。在这种模式下，计数器可以在一个激励信号的触发下启动，并可在一段可编程的延时后产生一个脉宽可编程的脉冲。

可以通过从模式控制器启动计数器。可以在输出比较模式或 PWM 模式下生成波形。将 TIMx_CR1 寄存器中的 OPM 位置 1，即可选择单脉冲模式。这样，发生下一更新事件 UEV 时，计数器将自动停止。

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

- CNT<CCRx ≤ ARR（特别注意，0 < CCRx）

图 350. 单脉冲模式示例



例如，用户希望达到这样的效果：在 TI2 输入引脚检测到上升沿时，经过 t_{DELAY} 的延迟，在 OC1 上产生一个长度为 t_{PULSE} 的正脉冲。

使用 TI2FP2 作为触发 1：

1. 在 TIMx_CCMR1 寄存器中写入 CC2S = 01，将 TI2FP2 映射到 TI2。
2. 在 TIMx_CCER 寄存器中写入 CC2P=0 和 CC2NP=“0”，使 TI2FP2 能够检测上升沿。
3. 在 TIMx_SMCR 寄存器中写入 TS=110，将 TI2FP2 配置为从模式控制器的触发 (TRGI)。
4. 在 TIMx_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）。

- t_{DELAY} 由写入 TIMx_CCR1 寄存器的值定义。
- t_{PULSE} 由自动重载值与比较值之差 (TIMx_ARR - TIMx_CCR1) 来定义。
- 假设希望产生这样的波形：信号在发生比较匹配时从“0”变为“1”，在计数器达到自动重载值时由“1”变为“0”。为此，应在 TIMx_CCMR1 寄存器中写入 OC1M=111，以使能 PWM 模式 2。如果需要，可选择在 TIMx_CCMR1 寄存器的 OC1PE 和 TIMx_CR1 寄存器的 ARPE 中写入 1，使能预装载寄存器。这种情况下，必须在 TIMx_CCR1 寄存器中写入比较值并在 TIMx_ARR 寄存器中写入自动重载值，通过将 UG 位置 1 来产生更新，然后等待 TI2 上的外部触发事件。本例中，CC1P 的值为“0”。

在本例中，TIMx_CR1 寄存器中的 DIR 和 CMS 位应为低。

由于仅需要 1 个脉冲（单脉冲模式），因此应向 TIMx_CR1 寄存器的 OPM 位写入“1”，以便在发生下一更新事件（计数器从自动重载值返回到 0）时使计数器停止计数。TIMx_CR1 寄存器中的 OPM 位置“0”时，即选择重复模式。

特殊情况：OCx 快速使能：

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

如果要输出延迟时间最短的波形，可以将 TIMx_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。

35.3.14 可再触发单脉冲模式 (OPM)

该模式允许计数器可以在一个激励信号的触发下启动，并且能产生长度可编程的脉冲，但与不可再触发单脉冲模式间存在以下差别，如*第 35.3.13 节*所述：

- 发生触发时，脉冲立即产生（无可编程延时）
- 如果在上一个触发完成前发生新的触发，脉冲将延长

定时器必须处于从模式，TIMx_SMCR 寄存器中的位 SMS[3:0] = “1000”（组合复位 + 触发模式），针对可再触发 OPM 模式 1 或模式 2 将 OCxM[3:0] 位设置为“1000”或“1001”。

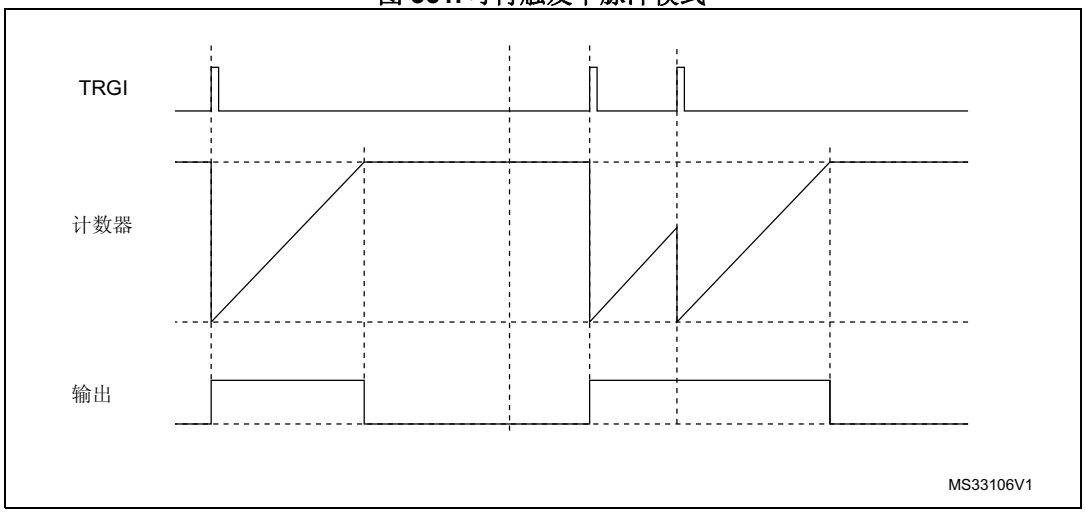
定时器配置为递增计数模式时，相应的 CCRx 必须置 0（ARR 寄存器设置脉冲长度）。如果定时器配置为递减计数模式，CCRx 必须高于或等于 ARR。

注： *在可再触发单脉冲模式下，CCxIF 标志没有意义。*

出于兼容性原因，OCxM[3:0] 和 SMS[3:0] 位域分为两部分，最高有效位与最低有效的 3 位不相邻。

此模式不能与中心对齐 PWM 模式一起使用。在 TIMx_CR1 中必须设置 CMS[1:0] = 00。

图 351. 可再触发单脉冲模式



35.3.15 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 TIMx_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 TIMx_CCER 寄存器的 CC1P 和 CC2P 位，选择 TI1 和 TI2 极性。CC1NP 和 CC2NP 必须保持清零。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 219*。如果使能计数器（在 TIMx_CR1 寄存器的 CEN 位中写入“1”），则计数器的时钟由 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 TIMx_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 TIMx_ARR 寄存器中的自动重载值之间进行连续计数（根据具体方向，从 0 递增计数到 ARR，或从 ARR 递减计数到 0）。因此，在启动前必须先配置 TIMx_ARR。同样，捕获、比较、预分频器、触发输出功能继续正常工作。

在此模式下，计数器会根据正交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 219. 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 352 以计数器工作为例，说明了计数信号的生成和方向控制。同时也说明了选择双边沿时如何对输入抖动进行补偿。将传感器靠近其中一个切换点放置时可能出现这种情况。本例中假设配置如下：

- CC1S=01（TIMx_CCMR1 寄存器，TI1FP1 映射到 TI1 上）
- CC2S=01（TIMx_CCMR2 寄存器，TI2FP2 映射到 TI2 上）
- CC1P 和 CC1NP =“0”（TIMx_CCER 寄存器，TI1FP1 未反相，TI1FP1=TI1）
- CC2P 和 CC2NP =“0”（TIMx_CCER 寄存器，TI2FP2 未反相，TI2FP2= TI2）
- SMS=011（TIMx_SMCR 寄存器，两个输入在上升沿和下降沿均有效）
- CEN = 1（TIMx_CR1 寄存器，使能计数器）

图 352. 编码器接口模式下的计数器工作示例

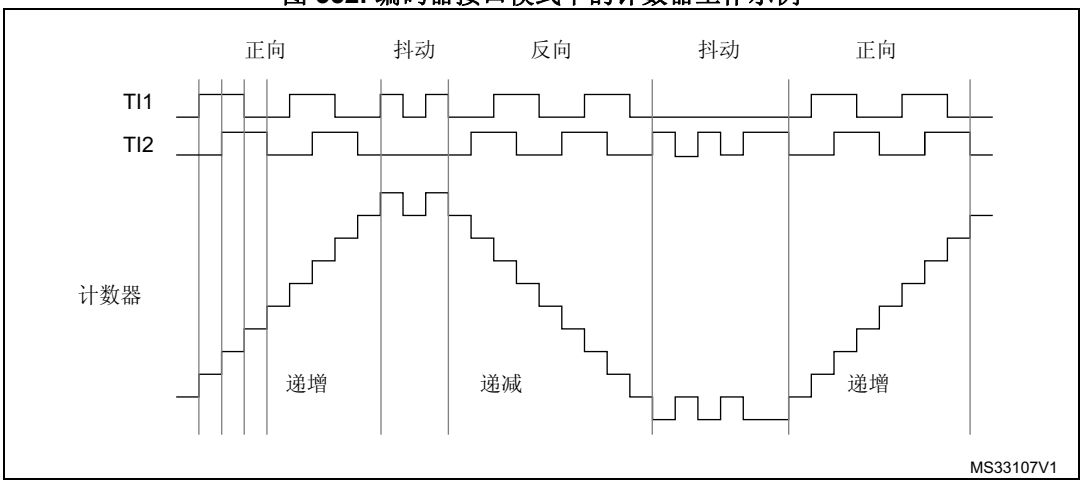
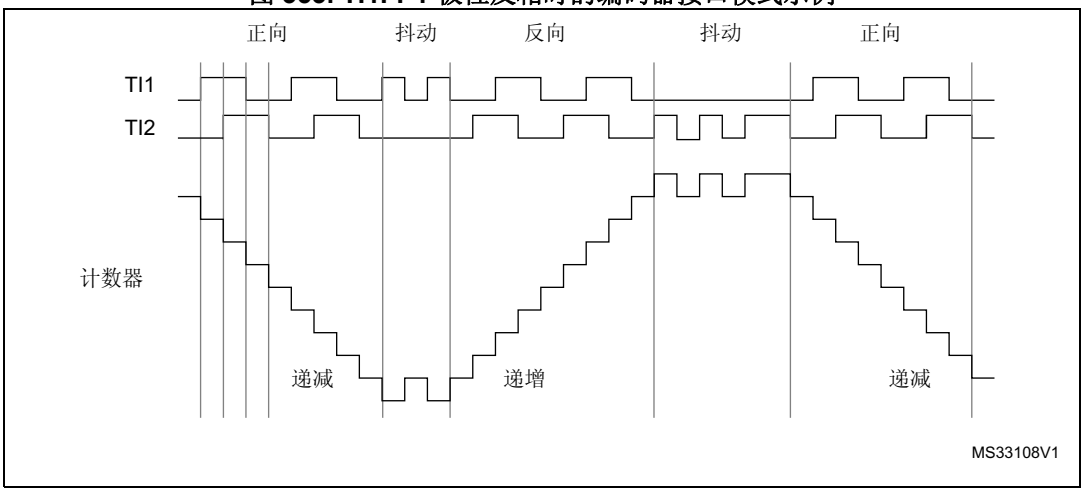


图 353 说明了 TI1FP1 极性反相时的计数器行为示例（除 CC1P=1 外，其他配置与上例相同）。

图 353. 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）；此外，还可以通过实时时钟产生的 DMA 请求读取计数器值。

35.3.16 UIF 位重映射

TIMx_CR1 寄存器中的 IUFREMAP 位强制将更新中断标志 (UIF) 连续复制到定时计数器寄存器的位 31 (TIMxCNT[31]) 中。这样便可自动读取计数器值以及由 UIFCPY 标志发出的电位翻转条件。这可避免在后台任务（计数器读）和中断（更新中断）之间共享处理时产生竞争条件，从而简化角速度的计算。

UIF 和 UIFCPY 标志使能之间没有延迟。

在 32 位定时器实现中，当 IUFREMAP 位置 1 时，计数器的位 31 在读访问时由 UIFCPY 标志覆盖（计数器的最高有效位只能在写模式下访问）。

35.3.17 定时器输入异或功能

借助 TIM1xx_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 TIMx_CH1 到 TIMx_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。

*第 1090 页的第 34.3.25 节：连接霍尔传感器*以连接霍尔传感器为例介绍了此功能。

35.3.18 定时器与外部触发同步

TIMx 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

当触发输入信号发生变化时，计数器及其预分频器可重新初始化。此外，如果 TIMx_CR1 寄存器中的 URS 位为 0，则会生成更新事件 UEV。然后，所有预装载寄存器（TIMx_ARR 和 TIMx_CCRx）都将更新。

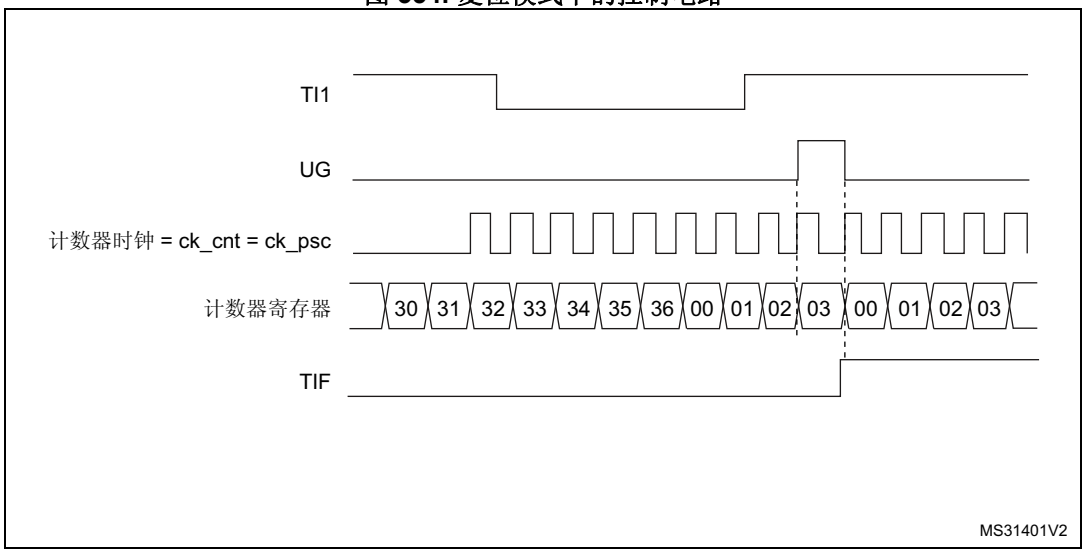
在以下示例中，TI1 输入上出现上升沿时，递增计数器清零：

1. 将通道 1 配置为检测 TI1 的上升沿。配置输入滤波带宽（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 TIMx_CCMR1 寄存器中的 CC1S = 01。在 TIMx_CCER 寄存器中写入 CC1P=0 和 CC1NP=0，验证极性（仅检测上升沿）。
2. 在 TIMx_SMCR 寄存器中写入 SMS=100，将定时器配置为复位模式。在 TIMx_SMCR 寄存器中写入 TS=101，选择 TI1 作为输入源。
3. 在 TIMx_CR1 寄存器中写入 CEN=1，启动计数器。

计数器开始根据内部时钟计数，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（TIMx_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 TIMx_DIER 寄存器中的 TIE 和 TDE 位）。

下图显示了自动重载寄存器 TIMx_ARR=0x36 时的相关行为。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 354. 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。

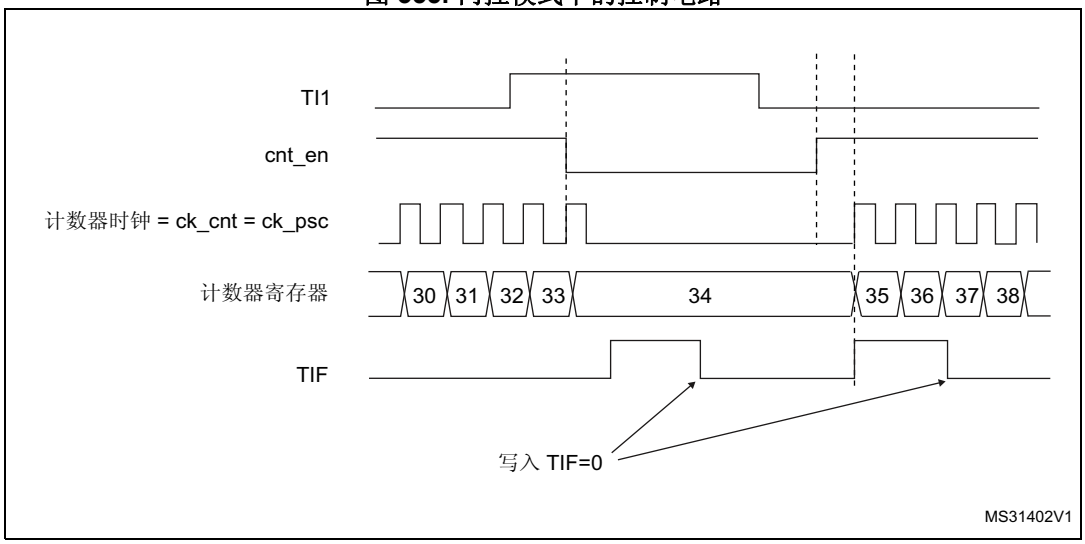
在以下示例中，递增计数器仅在 TI1 输入为低电平时计数：

1. 将通道 1 配置为检测 TI1 上的低电平。配置输入滤波带宽（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 TIMx_CCMR1 寄存器中的 CC1S=01。在 TIMx_CCER 寄存器中写入 CC1P=1 和 CC1NP=0，以确定极性（仅检测低电平）。
2. 在 TIMx_SMCR 寄存器中写入 SMS=101，将定时器配置为门控模式。在 TIMx_SMCR 寄存器中写入 TS=101，选择 TI1 作为输入源。
3. 在 TIMx_CR1 寄存器中写入 CEN=1，使能计数器（在门控模式下，如果 CEN=0，则无论触发输入电平如何，计数器都不启动）。

只要 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，TIMx_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 355. 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。

注： *由于门控模式作用于电平而非边沿，因此在门控模式下，“CCxP=CCxNP=1”（同时检测上升沿和下降沿）不发挥任何作用。*

从模式：触发模式

所选输入上发生某一事件时可以启动计数器。

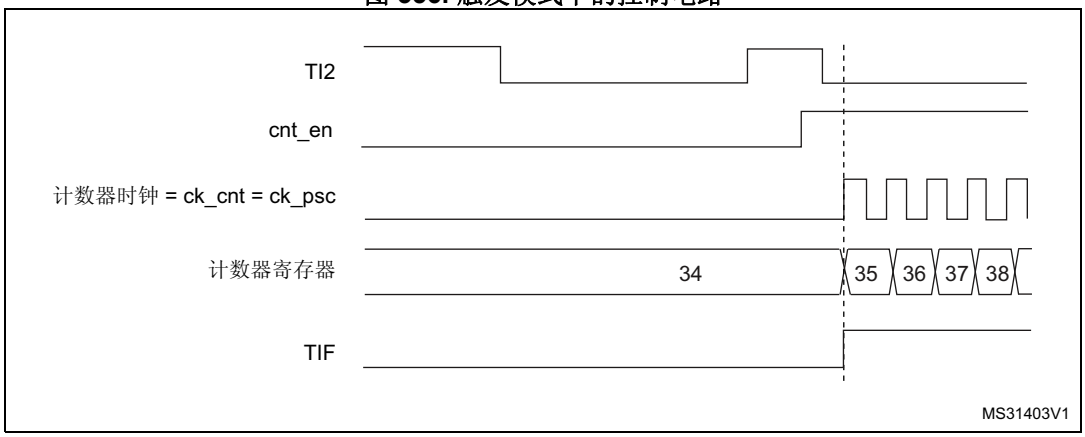
在以下示例中，TI2 输入上出现上升沿时，递增计数器启动：

1. 将通道 2 配置为检测 TI2 上的上升沿。配置输入滤波带宽（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 TIMx_CCMR1 寄存器中的 CC2S=01。在 TIMx_CCER 寄存器中写入 CC2P=1 和 CC2NP=0，以确定极性（仅检测低电平）。
2. 在 TIMx_SMCR 寄存器中写入 SMS=110，将定时器配置为触发模式。在 TIMx_SMCR 寄存器中写入 TS=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 356. 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 TIMx_SMCR 寄存器中的 TS 位来选择 ETR 作为 TRGI。

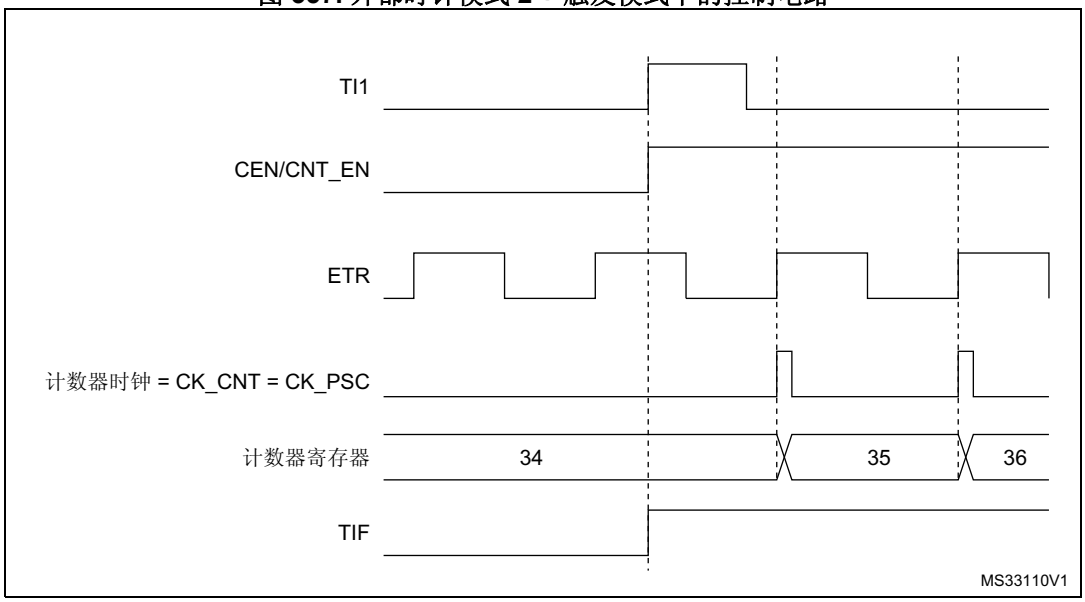
在以下示例中，只要 TI1 出现上升沿，递增计数器即会在 ETR 信号的每个上升沿处递增：

1. 通过对 TIMx_SMCR 寄存器进行如下编程，配置外部触发输入电路：
   - ETF = 0000：无滤波器。
   - ETPS = 00：禁止预分频器。
   - ETP = 0：检测 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。
   - TIMx_CCMR1 寄存器中 CC1S=01，只选择输入捕获源。
   - TIMx_CCER 寄存器中 CC1P=0 且 CC1NP= 0，以确定极性（仅检测上升沿）。
3. 在 TIMx_SMCR 寄存器中写入 SMS=110，将定时器配置为触发模式。在 TIMx_SMCR 寄存器中写入 TS=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 357. 外部时钟模式 2 + 触发模式下的控制电路

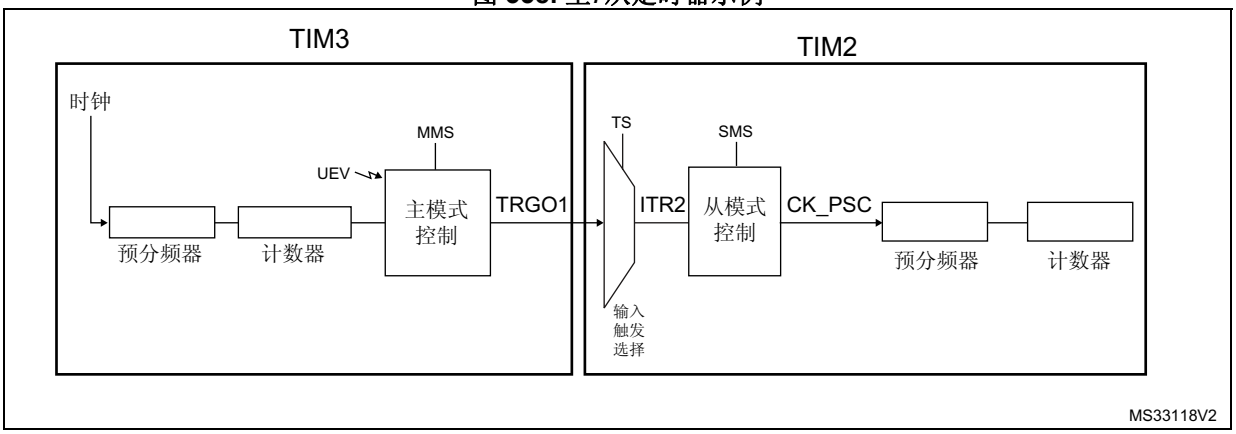


35.3.19 定时器同步

TIMx 定时器从内部连接在一起，以实现定时器同步或链接。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 358：主/从定时器示例*简要介绍了触发选择和主模式选择框图。

图 358. 主/从定时器示例



将一个定时器用作另一个定时器的预分频器

例如，可以将 TIM3 配置为 TIM2 的预分频器。请参见*图 358*。为此：

1. 将 TIM3 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 TIM3_CR2 寄存器中写入 MMS=010，则每次生成更新事件时，TRGO 都会输出一个上升沿。
2. 要将 TIM3 的 TRGO 输出连接到 TIM2，必须将 TIM2 配置为从模式，使用 ITR2 作为内部触发。通过 TIM2_SMCR 寄存器中的 TS 位（写入 TS=010）可对此进行选择。
3. 然后将从模式控制器设为外部时钟模式 1（在 TIM2_SMCR 寄存器中写入 SMS=111）。这样一来，TIM2 的时钟将由 TIM3 周期性触发信号的上升沿（与 TIM3 的计数器上溢对应）提供。
4. 最后必须通过将这两个定时器的相应 CEN 位（TIMx_CR1 寄存器）置 1 同时使能二者。

注： *如果选择 TIM3 的 OCx 信号作为触发输出 (MMS=1xx)，该信号的上升沿将用于驱动 TIM2 的计数器。*

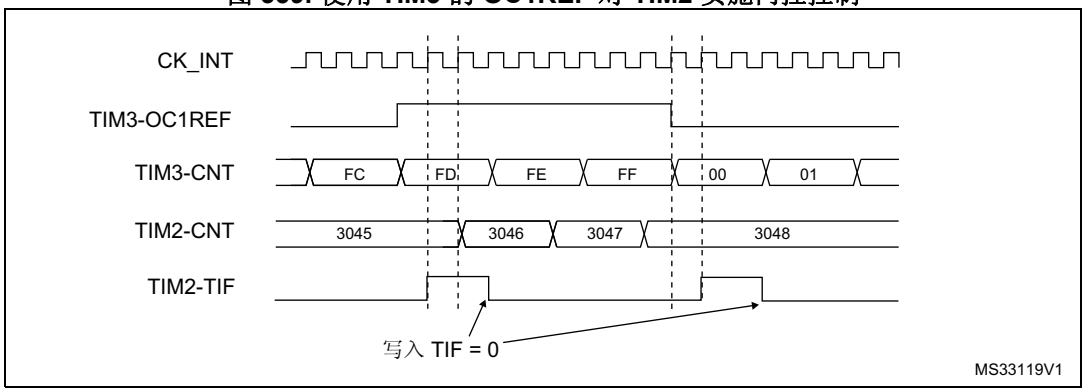
使用一个定时器使能另一个定时器

本例中通过定时器 3 的输出比较 1 来使能 TIM2。相关连接图，请参见*图 358*。仅当 TIM3 的 OC1REF 为高电平时，TIM2 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 ($f_{CK_CNT} = f_{CK_INT}/3$)。

1. 将 TIM3 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（TIM3_CR2 寄存器中的 MMS=100）。
2. 配置 TIM3 的 OC1REF 波形（TIM3_CCMR1 寄存器）。
3. 配置 TIM2 以接收来自 TIM3 的输入触发（TIM2_SMCR 寄存器中的 TS=010）。
4. 将 TIM2 配置为门控模式（TIM2_SMCR 寄存器中的 SMS=101）。
5. 通过向 CEN 位（TIM2_CR1 寄存器）写入“1”使能 TIM2。
6. 通过向 CEN 位（TIM3_CR1 寄存器）写入“1”启动 TIM3。

注： *计数器 2 的时钟与计数器 1 不同步，此模式仅影响 TIM2 的计数器使能信号。*

图 359. 使用 TIM3 的 OC1REF 对 TIM2 实施门控控制

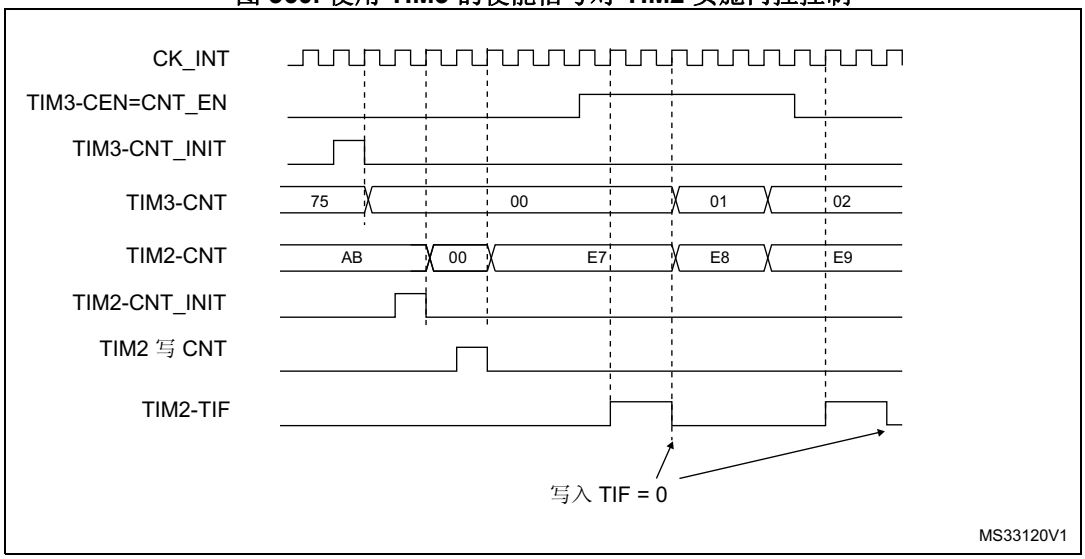


在*图 359* 的示例中，TIM2 的计数器和预分频器在启动前未进行初始化。因此从各自的当前值开始计数。启动定时器 TIM3 之前，通过复位这两个定时器可以从指定值开始计数。这样便可以在定时器计数器中写入所需的任意值。两个定时器都可通过软件使用 TIMx_EGR 寄存器中的 UG 位轻松复位。

在下一示例中（请参见*图 360*），TIM3 与 TIM2 同步。TIM3 为主模式，从 0 开始计数。TIM2 为从模式，从 0xE7 开始计数。两个定时器的预分频比相同。通过向 TIM3_CR1 寄存器中的 CEN 位写入“0”来禁止 TIM3 时，TIM2 将停止：

1. 将 TIM3 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（TIM3_CR2 寄存器中的 MMS=100）。
2. 配置 TIM3 的 OC1REF 波形（TIM3_CCMR1 寄存器）。
3. 配置 TIM2 以接收来自 TIM3 的输入触发（TIM2_SMCR 寄存器中的 TS=010）。
4. 将 TIM2 配置为门控模式（TIM2_SMCR 寄存器中的 SMS=101）。
5. 通过向 UG 位（TIM3_EGR 寄存器）写入“1”复位 TIM3。
6. 通过向 UG 位（TIM2_EGR 寄存器）写入“1”复位 TIM2。
7. 通过在 TIM2 的计数器 (TIM2_CNTL) 中写入“0xE7”使 TIM2 初始化为 0xE7。
8. 通过向 CEN 位（TIM2_CR1 寄存器）写入“1”使能 TIM2。
9. 通过向 CEN 位（TIM3_CR1 寄存器）写入“1”启动 TIM3。
10. 通过向 CEN 位（TIM3_CR1 寄存器）写入“0”停止 TIM3。

图 360. 使用 TIM3 的使能信号对 TIM2 实施门控控制

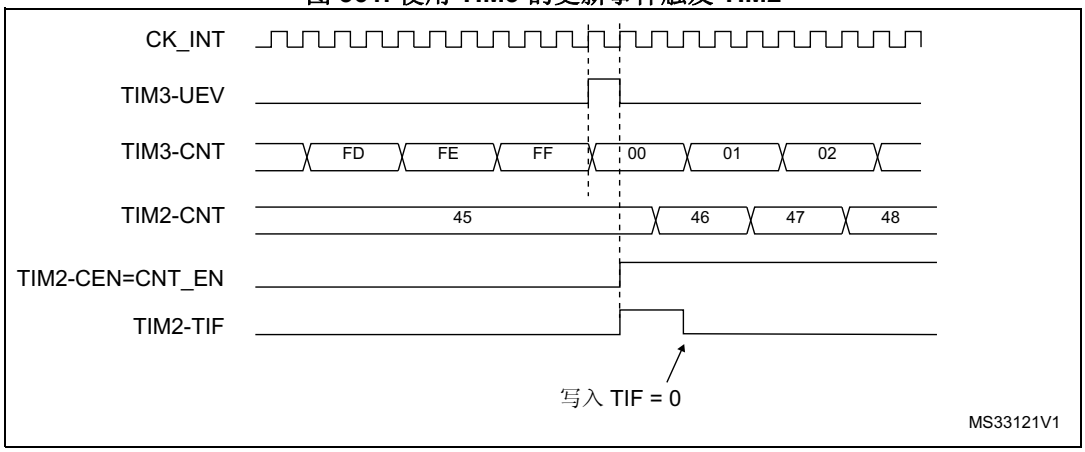


使用一个定时器启动另一个定时器

本例中，使用定时器 3 的更新事件使能定时器 2。相关连接图，请参见*图 358*。只要定时器 3 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 TIM2_CR1 寄存器的 CEN 位写入“0”后停止计数。两个计数器的时钟频率都基于 CK_INT 通过预分频器执行 3 分频 ($f_{CK_CNT} = f_{CK_INT}/3$)。

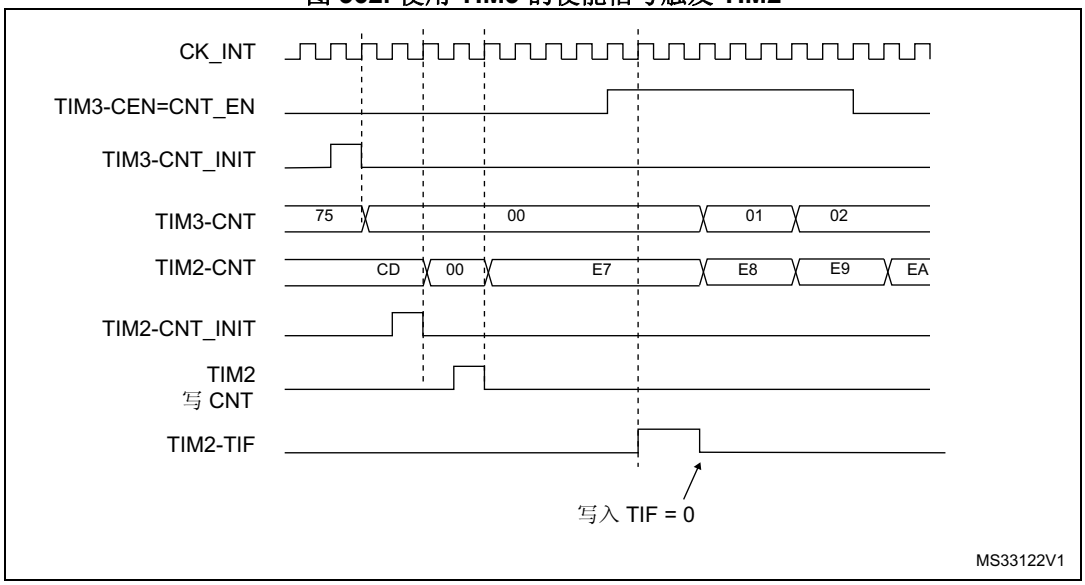
1. 将 TIM3 配置为主模式，发送其更新事件 (UEV) 作为触发输出（TIM3_CR2 寄存器中的 MMS=010）。
2. 配置 TIM3 的周期（TIM3_ARR 寄存器）。
3. 配置 TIM2 以接收来自 TIM3 的输入触发（TIM2_SMCR 寄存器中的 TS=010）。
4. 将 TIM2 配置为触发模式（TIM2_SMCR 寄存器中的 SMS=110）。
5. 通过向 CEN 位（TIM3_CR1 寄存器）写入“1”启动 TIM3。

图 361. 使用 TIM3 的更新事件触发 TIM2



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 362* 显示了与*图 361* 具有相同配置，只不过处于触发模式（TIM2_SMCR 寄存器中的 SMS=110）而非门控模式的计数行为。

图 362. 使用 TIM3 的使能信号触发 TIM2



使用一个外部触发同步启动 2 个定时器

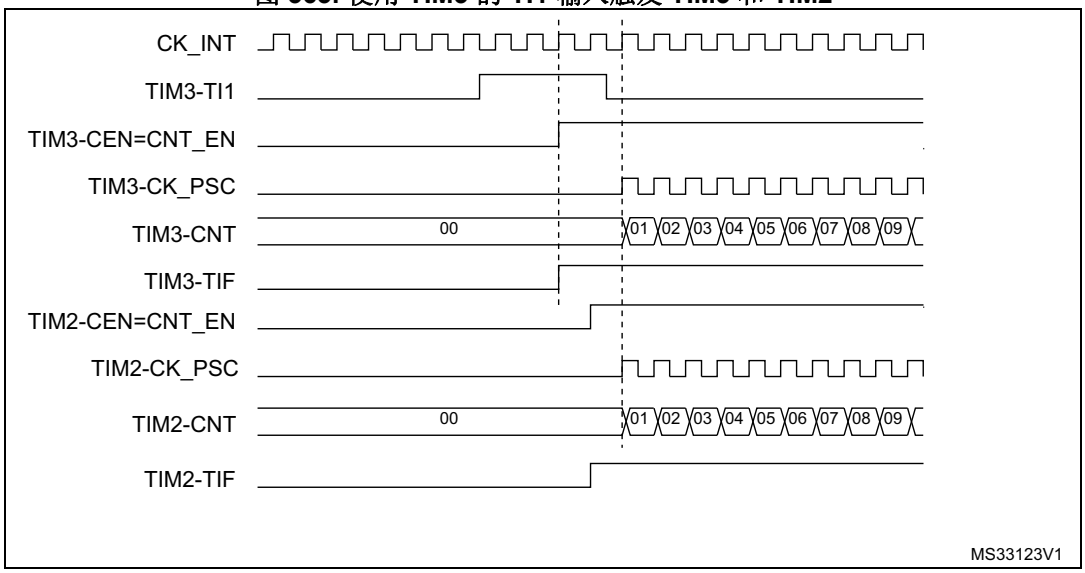
本例中，TIM3 的 TI1 输入出现上升沿时使能 TIM3，使能 TIM3 的同时使能 TIM2。相关连接图，请参见*图 358*。要确保计数器对齐，TIM3 必须配置为主/从模式（对于 TI1 而言，TIM3 为从；对于 TIM2 而言，TIM3 为主）：

1. 将 TIM3 配置为主模式，发送其使能信号作为触发输出（TIM3_CR2 寄存器中的 MMS=001）。
2. 将 TIM3 配置为从模式以接收来自 TI1 的输入触发（TIM3_SMCR 寄存器中的 TS=100）。
3. 将 TIM3 配置为触发模式（TIM3_SMCR 寄存器中的 SMS=110）。
4. 通过写入 MSM=1（TIM3_SMCR 寄存器）将 TIM3 配置为主/从模式。
5. 配置 TIM2 以接收来自 TIM3 的输入触发（TIM2_SMCR 寄存器中的 TS=000）。
6. 将 TIM2 配置为触发模式（TIM2_SMCR 寄存器中的 SMS=110）。

当 TI1 (TIM3) 出现上升沿时，两个计数器开始根据内部时钟同步计数，并且两个 TIF 标志都置 1。

注： *本例中，两个定时器都在启动之前进行了初始化（通过将各自的 UG 位置 1）。两个计数器都从 0 开始计数，但可以通过对任意一个计数器寄存器 (TIMx_CNT) 进行写操作，在二者之间轻松插入一个偏移量。可注意到主/从模式在 TIM3 的 CNT_EN 与 CK_PSC 之间产生了延迟。*

图 363. 使用 TIM3 的 TI1 输入触发 TIM3 和 TIM2



注： *必须先使能接收 TRGO 或 TRGO2 信号的从外设（定时器、ADC 等）的时钟，才能从主定时器接收事件；并且从主定时器接收触发信号时，不得实时更改时钟频率（预分频器）。*

35.3.20 DMA 连续传送模式

TIMx 定时器能够根据一个事件生成多个 DMA 请求。主要目的是能够对定时器的一部分多次重新编程而无需软件开销，但也可用于定期读取一行中的多个寄存器。

DMA 控制器目标唯一，必须指向虚拟寄存器 TIMx_DMAR。发生给定的定时器事件时，定时器会启动 DMA 请求序列（突发）。每次写入 TIMx_DMAR 寄存器都会重定向到其中一个定时器寄存器。

TIMx_DCR 寄存器中的 DBL[4:0] 位设置 DMA 连续传送长度。当对 TIMx_DMAR 地址进行读或写访问时，定时器进行一次连续传送，即传送次数（按半字或字节）。

TIMx_DCR 寄存器中的 DBA[4:0] 位定义 DMA 传送的 DMA 基址（通过 TIMx_DMAR 地址执行读/写访问时）。DBA 定义为从 TIMx_CR1 寄存器地址开始计算的偏移量：

示例：

00000：TIMx_CR1

00001：TIMx_CR2

00010：TIMx_SMCR

例如，定时器 DMA 连续传送功能用于在发生更新事件后将 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 TIMx 更新 DMA 请求（DIER 寄存器中的 UDE 位置 1）。
4. 使能 TIMx。
5. 使能 DMA 通道。

本例适用于每个 CCRx 寄存器必须更新一次的情况。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。

注： *可以将空值写入保留的寄存器中。*

35.3.21 调试模式

当微控制器进入调试模式时（Cortex®-M4 内核停止），TIMx 计数器会根据 DBGMCU 模块中的 DBG_TIMx_STOP 配置位选择继续正常工作或者停止工作。有关详细信息，请参见 *第 51.16.2 节：对定时器、RTC、看门狗、bxCAN 和 I2C 的调试支持*。

35.4 TIM2/TIM3/TIM4/TIM5 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。

35.4.1 TIMx 控制寄存器 1 (TIMx_CR1)

TIMx 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
Res.	Res.	Res.	Res.	UIFREMAP	Res.	CKD[1:0]		ARPE	CMS		DIR	OPM	URS	UDIS	CEN
				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 **UIFREMAP**：UIF 状态位重映射 (UIF status bit remapping)
0：无重映射。UIF 状态位不复制到 TIMx_CNT 寄存器的位 31。
1：使能重映射。UIF 状态位复制到 TIMx_CNT 寄存器的位 31。

位 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) 频率与数字滤波器所使用的采样时钟（ETR、TIx）之间的分频比，
00：$t_{DTS} = t_{CK_INT}$
01：$t_{DTS} = 2 \times t_{CK_INT}$
10：$t_{DTS} = 4 \times t_{CK_INT}$
11：保留

位 7 **ARPE**：自动重载预装载使能 (Auto-reload preload enable)
0：TIMx_ARR 寄存器不进行缓冲
1：TIMx_ARR 寄存器进行缓冲

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（TIMx_CCMRx 寄存器中的 CCxS=00）的输出比较中断标志才置 1。
10：中心对齐模式 2。计数器交替进行递增计数和递减计数。仅当计数器递增计数时，配置为输出的通道（TIMx_CCMRx 寄存器中的 CCxS=00）的输出比较中断标志才置 1。
11：中心对齐模式 3。计数器交替进行递增计数和递减计数。当计数器递增计数或递减计数时，配置为输出的通道（TIMx_CCMRx 寄存器中的 CCxS=00）的输出比较中断标志都会置 1。
注：　只要计数器处于使能状态 (CEN=1)，就不得从边沿对齐模式切换为中心对齐模式。

位 4 **DIR**：方向 (Direction)

0：计数器递增计数

1：计数器递减计数

注： 当定时器配置为中心对齐模式或编码器模式时，该位为只读状态。

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，所有以下事件都会产生更新中断或 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： 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。

35.4.2 TIMx 控制寄存器 2 (TIMx_CR2)

TIMx 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.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TI1S	MMS[2:0]			CCDS	Res.	Res.	Res.
								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：TIMx_CH1 引脚连接到 TI1 输入

1：TIMx_CH1、CH2 和 CH3 引脚连接到 TI1 输入（异或组合）。另请参见*第 1090 页的第 34.3.25 节：连接霍尔传感器*

位 6:4 **MMS[2:0]**：主模式选择 (Master mode selection)

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

000：**复位**——TIMx_EGR 寄存器中的 UG 位用作触发输出 (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 TIMx_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 标志被置 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)

注：　必须先使能从定时器或 ADC 的时钟，才能从主定时器接收事件；并且从主定时器接收触发信号时，不得实时更改从定时器或 ADC 的时钟。

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。

35.4.3 TIMx 从模式控制寄存器 (TIMx_SMCR)

TIMx slave mode control 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
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SMS[3]
															rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ETP	ECE	ETPS[1:0]		ETF[3:0]				MSM	TS[2:0]			Res.	SMS[2:0]		
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw		rw	rw	rw

位 31:17 保留，必须保持复位值。

位 16 **SMS[3]**：从模式选择——位 3 (Slave mode selection - bit 3)
请参见 SMS 说明——位 2:0

位 15 **ETP**：外部触发极性 (External trigger polarity)
此位可选择将 ETR 还是 $\overline{ETR}$ 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[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 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 的数字滤波器带宽。数字滤波器由事件计数器组成，每 N 个连续事件才视为一个有效输出边沿：

0000：无滤波器，按 f_{DTS} 频率进行采样
0001：$f_{SAMPLING}=f_{CK_INT}$，N=2
0010：$f_{SAMPLING}=f_{CK_INT}$，N=4
0011：$f_{SAMPLING}=f_{CK_INT}$，N=8
0100：$f_{SAMPLING}=f_{DTS}/2$，N=6
0101：$f_{SAMPLING}=f_{DTS}/2$，N=8
0110：$f_{SAMPLING}=f_{DTS}/4$，N=6
0111：$f_{SAMPLING}=f_{DTS}/4$，N=8
1000：$f_{SAMPLING}=f_{DTS}/8$，N=6
1001：$f_{SAMPLING}=f_{DTS}/8$，N=8
1010：$f_{SAMPLING}=f_{DTS}/16$，N=5
1011：$f_{SAMPLING}=f_{DTS}/16$，N=6
1100：$f_{SAMPLING}=f_{DTS}/16$，N=8
1101：$f_{SAMPLING}=f_{DTS}/32$，N=5
1110：$f_{SAMPLING}=f_{DTS}/32$，N=6
1111：$f_{SAMPLING}=f_{DTS}/32$，N=8

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**：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*第 1190 页的表 220：TIMx 内部触发连接*。

注：　这些位只能在未使用的情况下（例如，SMS=000 时）进行更改，以避免转换时出现错误的边沿检测。

位 3 保留，必须保持复位值。

位 16、2、1、0 **SMS[3:0]**：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）。

0000：禁止从模式——如果 CEN = “1”，预分频器时钟直接由内部时钟提供。
0001：编码器模式 1——计数器根据 TI2FP2 电平在 TI1FP1 边沿递增/递减计数。
0010：编码器模式 2——计数器根据 TI1FP1 电平在 TI2FP2 边沿递增/递减计数。
0011：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输入的电平。
0100：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。
0101：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被控制。
0110：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。
0111：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。
1000：组合复位 + 触发模式——在出现所选触发输入 (TRGI) 上升沿时，重新初始化计数器，生成一个寄存器更新事件并启动计数器。

注：　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。

注：　必须先使能接收 TRGO 或 TRGO2 信号的从外设（定时器、ADC 等）的时钟，才能从主定时器接收事件；并且从主定时器接收触发信号时，不得实时更改时钟频率（预分频器）。

表 220. TIMx 内部触发连接

从 TIM	ITR0 (TS = 000)	ITR1 (TS = 001)	ITR2 (TS = 010)	ITR3 (TS = 011)
TIM2	TIM1	TIM8/OTG_FS_SOF(1)	TIM3	TIM4
TIM3	TIM1	TIM2	TIM5	TIM4
TIM4	TIM1	TIM2	TIM3	TIM8
TIM5	TIM2	TIM3	TIM4	TIM8

1. 取决于 TIM2_OR1 寄存器中的 ITR1_RMP 位。

35.4.4 TIMx DMA/中断使能寄存器 (TIMx_DIER)

TIMx 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，必须保持复位值。

位 12 **CC4DE**：捕获/比较 4 DMA 请求使能 (Capture/Compare 4 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 3 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 2 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**：触发中断使能 (Trigger interrupt enable)
0：禁止触发中断。
1：使能触发中断。

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 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。

35.4.5 TIMx 状态寄存器 (TIMx_SR)

TIMx 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
Res.	Res.	Res.	CC4OF	CC3OF	CC2OF	CC1OF	Res.	Res.	TIF	Res.	CC4IF	CC3IF	CC2IF	CC1IF	UIF
			rc_w0	rc_w0	rc_w0	rc_w0			rc_w0		rc_w0	rc_w0	rc_w0	rc_w0	rc_w0

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：TIMx_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 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

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（请参见 TIMx_CR1 寄存器中的 CMS 位说明）和可再触发单脉冲模式下除外。但需要通过软件清零。
0：不匹配。
1：TIMx_CNT 计数器的内容与 TIMx_CCR1 寄存器的内容匹配。
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 TIMx_CCR1 寄存器将该位清零。
0：未发生输入捕获事件。
1：TIMx_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 TIM2 到 TIM4）以及当 TIMx_CR1 寄存器中的 UDIS = 0 时。
TIMx_CR1 寄存器中的 URS = 0 且 UDIS = 0，并且由软件使用 TIMx_EGR 寄存器中的 UG 位重新初始化 CNT 时。
TIMx_CR1 寄存器中的 URS=0 且 UDIS=0，并且 CNT 由触发事件重新初始化时（参见同步控制寄存器说明）。

35.4.6 TIMx 事件生成寄存器 (TIMx_EGR)

TIMx 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
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	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**：触发生成 (Trigger generation)
此位由软件置 1 以生成事件，并由硬件自动清零。
0：不执行任何操作。
1：TIMx_SR 寄存器中的 TIF 标志置 1。使能后可发生相关中断或 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 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：
TIMx_CCR1 寄存器中将捕获到计数器当前值。使能时，CC1IF 标志置 1 并发送相应的中断或 DMA 请求。如果 CC1IF 标志已为高，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 (TIMx_ARR)。

35.4.7 TIMx 捕获/比较模式寄存器 1 (TIMx_CCMR1)

TIMx 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。此寄存器的所有其他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。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC2M [3]	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC1M [3]
							Res.								Res.
							rw								rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OC2CE	OC2M[2:0]			OC2PE	OC2FE	CC2S[1:0]		OC1CE	OC1M[2:0]			OC1PE	OC1FE	CC1S[1:0]	
IC2F[3:0]				IC2PSC[1:0]				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 31:25 保留，始终读为 0。

位 24 **OC2M[3]**：输出比较 2 模式——位 3 (Output Compare 2 mode - bit 3)

位 23:17 保留，始终读为 0。

位 16 **OC1M[3]**：输出比较 1 模式——位 3 (Output Compare 1 mode - bit 3)

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)
请参见 OC1M 说明——位 6:4

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（TIMx_SMCR 寄存器）选择内部触发输入时有效
注： 仅当通道关闭时（TIMx_CCER 中的 CC2E = 0），才可向 CC2S 位写入数据。

位 7 **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。

0000：冻结——输出比较寄存器 TIMx_CCR1 与计数器 TIMx_CNT 进行比较不会对输出造成任何影响。（该模式用于生成时基）。

0001：将通道 1 设置为匹配时输出有效电平。当计数器 TIMx_CNT 与捕获/比较寄存器 1 (TIMx_CCR1) 匹配时，OC1REF 信号强制变为高电平。

0010：将通道 1 设置为匹配时输出无效电平。当计数器 TIMx_CNT 与捕获/比较寄存器 1 (TIMx_CCR1) 匹配时，OC1REF 信号强制变为低电平。

0011：翻转——TIMx_CNT=TIMx_CCR1 时，OC1REF 发生翻转。

0100：强制变为无效电平——OC1REF 强制变为低电平。

0101：强制变为有效电平——OC1REF 强制变为高电平。

0110：PWM 模式 1——在递增计数模式下，只要 TIMx_CNT<TIMx_CCR1，通道 1 便为有效状态，否则为无效状态。在递减计数模式下，只要 TIMx_CNT>TIMx_CCR1，通道 1 便为无效状态 (OC1REF=0)，否则为有效状态 (OC1REF=1)。

0111：PWM 模式 2——在递增计数模式下，只要 TIMx_CNT<TIMx_CCR1，通道 1 便为无效状态，否则为有效状态。在递减计数模式下，只要 TIMx_CNT>TIMx_CCR1，通道 1 便为有效状态，否则为无效状态。

1000：可再触发 OPM 模式 1——在递增计数模式下，通道为有效状态，直至（在 TRGI 信号上）检测到触发事件。然后，在 PWM 模式 1 下进行比较，通道会在下一次更新时再次变为无效状态。在递减计数模式下，通道为无效状态，直至（在 TRGI 信号上）检测到触发事件。然后，在 PWM 模式 1 下进行比较，通道会在下一次更新时再次变为无效状态。

1001：可再触发 OPM 模式 2——在递增计数模式下，通道为无效状态，直至（在 TRGI 信号上）检测到触发事件。然后，在 PWM 模式 2 下进行比较，通道会在下一次更新时再次变为无效状态。在递减计数模式下，通道为有效状态，直至（在 TRGI 信号上）检测到触发事件。然后，在 PWM 模式 2 下进行比较，通道会在下一次更新时再次变为有效状态。

1010：保留。

1011：保留。

1100：组合 PWM 模式 1——OC1REF 与在 PWM 模式 1 下的行为相同。OC1REFC 是 OC1REF 和 OC2REF 的逻辑或运算结果。

1101：组合 PWM 模式 2——OC1REF 与在 PWM 模式 2 下的行为相同。OC1REFC 是 OC1REF 和 OC2REF 的逻辑与运算结果。

1110：不对称 PWM 模式 1——OC1REF 与在 PWM 模式 1 下的行为相同。计数器递增计数时，OC1REFC 输出 OC1REF；计数器递减计数时，OC1REFC 输出 OC2REF。

1111：不对称 PWM 模式 2——OC1REF 与在 PWM 模式 2 下的行为相同。计数器递增计数时，OC1REFC 输出 OC1REF；计数器递减计数时，OC1REFC 输出 OC2REF。

注：　在 PWM 模式下，仅当比较结果发生改变或输出比较模式由“冻结”模式切换到“PWM”模式时，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 TIMx_CCR1 相关的预装载寄存器。可随时向 TIMx_CCR1 写入数据，写入后将立即使用新值。

1：使能与 TIMx_CCR1 相关的预装载寄存器。可读/写访问预装载寄存器。TIMx_CCR1 预装载值在每次生成更新事件时都会装载到有效寄存器中。

注： 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（TIMx_CR1 寄存器中的 OPM 位置 1）。其他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 输出的最短延迟时间为 5 个时钟周期。

1：触发输入上出现有效边沿相当于 CC1 输出上的比较匹配。随后，无论比较结果如何，OC 都设置为比较电平。采样触发输入和激活 CC1 输出的延迟时间缩短为 3 个时钟周期。仅当通道配置为 PWM1 或 PWM2 模式时，OCFE 才会起作用。

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（TIMx_SMCR 寄存器）选择内部触发输入时有效

注： 仅当通道关闭时（TIMx_CCER 中的 CC1E = 0），才可向 CC1S 位写入数据。

输入捕获模式

位 31:16 保留，始终读为 0。

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（TIMx_SMCR 寄存器）选择内部触发输入时有效。

注： 仅当通道关闭时（TIMx_CCER 中的 CC2E = 0），才可向 CC2S 位写入数据。

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 的数字滤波器的采样长度。数字滤波器由事件计数器组成，每 N 个连续事件才视为一个有效输出边沿：

0000：无滤波器，按 f_{DTS} 频率进行采样
0001：$f_{SAMPLING}=f_{CK_INT}$，N=2
0010：$f_{SAMPLING}=f_{CK_INT}$，N=4
0011：$f_{SAMPLING}=f_{CK_INT}$，N=8
0100：$f_{SAMPLING}=f_{DTS}/2$，N=6
0101：$f_{SAMPLING}=f_{DTS}/2$，N=8
0110：$f_{SAMPLING}=f_{DTS}/4$，N=6
0111：$f_{SAMPLING}=f_{DTS}/4$，N=8
1000：$f_{SAMPLING}=f_{DTS}/8$，N=6
1001：$f_{SAMPLING}=f_{DTS}/8$，N=8
1010：$f_{SAMPLING}=f_{DTS}/16$，N=5
1011：$f_{SAMPLING}=f_{DTS}/16$，N=6
1100：$f_{SAMPLING}=f_{DTS}/16$，N=8
1101：$f_{SAMPLING}=f_{DTS}/32$，N=5
1110：$f_{SAMPLING}=f_{DTS}/32$，N=6
1111：$f_{SAMPLING}=f_{DTS}/32$，N=8

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（TIMx_CCER 寄存器），预分频器便立即复位。

00：无预分频器，捕获输入上每检测到一个边沿便执行捕获
01：每发生 2 个事件便执行一次捕获
10：每发生 4 个事件便执行一次捕获
11：每发生 8 个事件便执行一次捕获

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（TIMx_SMCR 寄存器）选择内部触发输入时有效

注：　仅当通道关闭时（TIMx_CCER 中的 CC1E = 0），才可向 CC1S 位写入数据。

35.4.8 TIMx 捕获/比较模式寄存器 2 (TIMx_CCMR2)

TIMx 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。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC4M [3]	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC3M [3]
							Res.								Res.
							rw								rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OC4CE	OC4M[2:0]			OC4PE	OC4FE	CC4S[1:0]		OC3CE	OC3M[2:0]			OC3PE	OC3FE	CC3S[1:0]	
IC4F[3:0]				IC4PSC[1:0]				IC3F[3:0]				IC3PSC[1:0]			
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

输出比较模式

位 31:25 保留，始终读为 0。

位 24 **OC4M[3]**：输出比较 2 模式——位 3 (Output Compare 2 mode - bit 3)

位 23:17 保留，始终读为 0。

位 16 **OC3M[3]**：输出比较 1 模式——位 3 (Output Compare 1 mode - bit 3)

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)
请参见 OC1M 说明（TIMx_CCMR1 寄存器中的位 6:4）

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（TIMx_SMCR 寄存器）选择内部触发输入时有效。
注： 仅当通道关闭时（TIMx_CCER 中的 CC4E = 0），才可向 CC4S 位写入数据。

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)
请参见 OC1M 说明（TIMx_CCMR1 寄存器中的位 6:4）

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（TIMx_SMCR 寄存器）选择内部触发输入时有效。
注： 仅当通道关闭时（TIMx_CCER 中的 CC3E = 0），才可向 CC3S 位写入数据。

输入捕获模式

位 31:16 保留，始终读为 0。

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（TIMx_SMCR 寄存器）选择内部触发输入时有效
注： 仅当通道关闭时（TIMx_CCER 中的 CC4E = 0），才可向 CC4S 位写入数据。

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（TIMx_SMCR 寄存器）选择内部触发输入时有效。

注： 仅当通道关闭时（TIMx_CCER 中的 CC3E = 0），才可向 CC3S 位写入数据。

35.4.9 TIMx 捕获/比较使能寄存器 (TIMx_CCER)

TIMx 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	CC3NP	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 互补输出极性 (Capture/Compare 4 complementary 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 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 保留，必须保持复位值。

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 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：在这种情况下，CC1NP 必须保持清零。
CC1 通道配置为输入：该位与 CC1P 配合使用可定义 TI1FP1/TI2FP1 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。

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 无效

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 (TIMx_CCR1) 中。

0：禁止捕获

1：使能捕获

表 221. 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： *与标准 OCx 通道相连的外部 IO 引脚的状态取决于 OCx 通道的状态以及 GPIO 和 AFIO 寄存器。*

35.4.10 TIMx 计数器 (TIMx_CNT)

TIMx counter

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
CNT[31] 或 UIFCPY	CNT[30:16]														
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CNT[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31 值取决于 TIMx_CR1 中的 IUFREMAP。

如果 UIFREMAP = 0

CNT[31]：计数器值的最高有效位（对于 TIM2 和 TIM5）(Most significant bit of counter value (on TIM2 and TIM5))

其他定时器上保留

如果 UIFREMAP = 1

UIFCPY：UIF 副本 (UIF Copy)

该位是 TIMx_ISR 寄存器中 UIF 位的只读副本

位 30:16 **CNT[30:16]**：计数器值的最高有效部分（对于 TIM2 和 TIM5）(Most significant part counter value (on TIM2 and TIM5))

位 15:0 **CNT[15:0]**：计数器值的最低有效部分 (Least significant part of counter value)

35.4.11 TIMx 预分频器 (TIMx_PSC)

TIMx 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 f_{CK_PSC} / (PSC[15:0] + 1)。

PSC 包含每次发生更新事件（包括计数器通过 TIMx_EGR 寄存器中的 UG 位清零时，或在配置为“复位模式”时通过触发控制器清零时）时要装载到有效预分频器寄存器的值。

35.4.12 TIMx 自动重载寄存器 (TIMx_ARR)

TIMx auto-reload register

偏移地址：0x2C

复位值：0xFFFF FFFF

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

位 31:16 **ARR[31:16]**：自动重载值的高 16 位（对于 TIM2 和 TIM5）(High auto-reload value (on TIM2 and TIM5))

位 15:0 **ARR[15:0]**：自动重载值的低 16 位 (Low 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*第 1143 页的第 35.3.1 节：时基单元*。

当自动重载值为空时，计数器不工作。

35.4.13 TIMx 捕获/比较寄存器 1 (TIMx_CCR1)

TIMx 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]（取决于定时器）															
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 值的高 16 位（对于 TIM2 和 TIM5）(High Capture/Compare 1 value (on TIM2 and TIM5))

位 15:0 **CCR1[15:0]**：捕获/比较 1 值的低 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 TIMx_CCMR1 寄存器中的 OC1PE 位来使能预装载功能，则该值立刻生效；否则只在发生更新事件时生效（拷贝到有效的捕获/比较寄存器 1）。

有效捕获/比较寄存器中包含要与计数器 TIMx_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。只能读取 TIMx_CCR1 寄存器，无法对其进行编程。

35.4.14 TIMx 捕获/比较寄存器 2 (TIMx_CCR2)

TIMx capture/compare register 2

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
CCR2[31:16]（取决于定时器）															
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 31:16 **CCR2[31:16]**：捕获/比较 2 值的高 16 位（对于 TIM2 和 TIM5）(High Capture/Compare 2 value (on TIM2 and TIM5))

位 15:0 **CCR2[15:0]**：捕获/比较 2 值的低 16 位 (Low Capture/Compare 2 value)

如果通道 CC2 配置为输出：

CCR2 是捕获/比较寄存器 2 的预装载值。

如果没有通过 TIMx_CCMR1 寄存器中的 OC2PE 位来使能预装载功能，则该值立刻生效；否则只在发生更新事件时生效（拷贝到有效的捕获/比较寄存器 2）。

有效捕获/比较寄存器中包含要与计数器 TIMx_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。只能读取 TIMx_CCR2 寄存器，无法对其进行编程。

35.4.15 TIMx 捕获/比较寄存器 3 (TIMx_CCR3)

TIMx 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]（取决于定时器）															
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 值的高 16 位（对于 TIM2 和 TIM5）(High Capture/Compare 3 value (on TIM2 and TIM5))

位 15:0 **CCR3[15:0]**：捕获/比较 3 值的低 16 位 (Low Capture/Compare 3 value)

如果通道 CC3 配置为输出：

CCR3 是捕获/比较寄存器 3 的预装载值。

如果没有通过 TIMx_CCMR2 寄存器中的 OC3PE 位来使能预装载功能，则该值立刻生效；否则只在发生更新事件时生效（拷贝到有效的捕获/比较寄存器 3）。

有效捕获/比较寄存器中包含要与计数器 TIMx_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。只能读取 TIMx_CCR3 寄存器，无法对其进行编程。

35.4.16 TIMx 捕获/比较寄存器 4 (TIMx_CCR4)

TIMx 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]（取决于定时器）															
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 值的高 16 位（对于 TIM2 和 TIM5）(High Capture/Compare 4 value (on TIM2 and TIM5))

位 15:0 **CCR4[15:0]**：捕获/比较 4 值的低 16 位 (Low Capture/Compare 4 value)

1. 如果 CC4 通道配置为输出（CC4S 位）：
   CCR4 是捕获/比较寄存器 4 的预装载值。
   如果没有通过 TIMx_CCMR2 寄存器中的 OC4PE 位来使能预装载功能，则该值立刻生效；否则只在发生更新事件时生效（拷贝到有效的捕获/比较寄存器 4）。
   有效捕获/比较寄存器中包含要与计数器 TIMx_CNT 进行比较并在 OC4 输出上发出信号的值。
2. 如果 CC4 通道配置为输入（TIMx_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。只能读取 TIMx_CCR4 寄存器，无法对其进行编程。

35.4.17 TIMx DMA 控制寄存器 (TIMx_DCR)

TIMx 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
Res.	Res.	Res.	DBL[4:0]					Res.	Res.	Res.	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 位向量定义了 DMA 的传送次数（当对 TIMx_DMAR 寄存器进行读或写时，定时器进行一次连续传送）。

00000：1 次传送，
00001：2 次传送，
00010：3 次传送，
...
10001：18 次传送。

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 TIMx_DMAR 地址进行读/写访问时）。DBA 定义为从 TIMx_CR1 寄存器地址开始计算的偏移量。

示例：
00000：TIMx_CR1
00001：TIMx_CR2
00010：TIMx_SMCR
...

示例：以下面的传送为例：DBL = 7 次传送且 DBA = TIMx_CR1。这种情况下将向/从自 TIMx_CR1 地址开始的 7 个寄存器传输数据。

35.4.18 TIMx 全传输 DMA 地址 (TIMx_DMAR)

TIMx 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

(TIMx_CR1 地址) + (DBA + DMA 索引) x 4

其中 TIMx_CR1 地址为控制寄存器 1 的地址，DBA 为 TIMx_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（TIMx_DCR 寄存器中配置的 DBL）之间。

35.4.19 TIM2 选项寄存器 1 (TIM2_OR1)

TIM2 option register 1

偏移地址：0x50

复位值：0x0000

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.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TI4_RMP[1:0]		ETR1_RMP	ITR1_RMP
												rw	rw	rw	rw

位 31:4 保留，必须保持复位值。

位 3:2 **TI4_RMP[1:0]**：输入捕获 4 重映射 (Input Capture 4 remap)
- 00：TIM2 输入捕获 4 连接到 I/O
- 01：TIM2 输入捕获 4 连接到 COMP1_OUT
- 10：TIM2 输入捕获 4 连接到 COMP2_OUT
- 11：TIM2 输入捕获 4 连接到 COMP1_OUT 和 COMP2_OUT 的逻辑或运算结果

位 1 **ETR1_RMP**：外部触发重映射 (External trigger remap)
- 0：TIM2_ETR 连接到 I/O
- 1：TIM2_ETR 连接到 LSE

位 0 **ITR1_RMP**：内部触发 1 重映射 (Internal trigger 1 remap)
- 0：TIM2_ITR1 连接到 TIM8_TRGO
- 1：TIM2_ITR1 连接到 OTG_FS_SOF

35.4.20 TIM3 选项寄存器 1 (TIM3_OR1)

TIM3 option register 1

偏移地址：0x50

复位值：0x0000

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.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TI1_RMP[1:0]	
														rw	rw

位 31:2 保留，必须保持复位值。

位 1:0 **TI1_RMP[1:0]**：输入捕获 1 重映射 (Input Capture 1 remap)
- 00：TIM3 输入捕获 1 连接到 I/O
- 01：TIM3 输入捕获 1 连接到 COMP1_OUT
- 10：TIM3 输入捕获 1 连接到 COMP2_OUT
- 11：TIM3 输入捕获 1 连接到 COMP1_OUT 和 COMP2_OUT 的逻辑或运算结果

35.4.21 TIM2 选项寄存器 2 (TIM2_OR2)

TIM2 option register 2

偏移地址：0x60

复位值：0x0000

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.	ETR SEL2
															rw

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

位 31:17 保留，必须保持复位值。

位 16:14 **ETRSEL[2:0]**：ETR 源选择 (ETR source selection)

这些位选择 ETR 输入源。

000：使用 TIM2_OR1 寄存器中的 ETR1_RMP 位域选择 TIM2_ETR 源

001：COMP1 输出连接到 ETR 输入

010：COMP2 输出连接到 ETR 输入

其他值：保留

位 13:0 保留，必须保持复位值。

35.4.22 TIM3 选项寄存器 2 (TIM3_OR2)

TIM3 option register 2

偏移地址：0x60

复位值：0x0000

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.	ETR SEL2
															rw

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

位 31:17 保留，必须保持复位值。

位 16:14 **ETRSEL[2:0]**：ETR 源选择 (ETR source selection)

这些位选择 ETR 输入源。

000：ETR 输入连接到 I/O

001：COMP1 输出连接到 ETR 输入

其他值：保留

位 13:0 保留，必须保持复位值。

35.4.23 TIMx 寄存器映射

TIMx 寄存器按照下表所述方式映射：

表 222. TIM2/TIM3/TIM4/TIM5 寄存器映射和复位值

偏移	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	**TIMx_CR1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	UIFREMAP	Res.	CKD [1:0]		ARPE	CMS [1:0]		DIR	OPM	URS	UDIS	CEN
	Reset value																					0		0	0	0	0	0	0	0	0	0	0
0x04	**TIMx_CR2**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TI1S	MMS[2:0]			CCDS	Res.	Res.	Res.
	Reset value																									0	0	0	0	0			
0x08	**TIMx_SMCR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SMS[3]	ETP	ECE	ETPS [1:0]		ETF[3:0]				MSM	TS[2:0]			Res.	SMS[2:0]		
	Reset value																0	0	0	0	0	0	0	0	0	0	0	0	0		0	0	0
0x0C	**TIMx_DIER**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TDE	Res.	CC4DE	CC3DE	CC2DE	CC1DE	UDE	Res.	TIE	Res.	CC4IE	CC3IE	CC2IE	CC1IE	UIE
	Reset value																		0		0	0	0	0	0		0		0	0	0	0	0
0x10	**TIMx_SR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CC4OF	CC3OF	CC2OF	CC1OF	Res.	Res.	TIF	Res.	CC4IF	CC3IF	CC2IF	CC1IF	UIF
	Reset value																				0	0	0	0			0		0	0	0	0	0
0x14	**TIMx_EGR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TG	Res.	CC4G	CC3G	CC2G	CC1G	UG
	Reset value																										0		0	0	0	0	0
0x18	**TIMx_CCMR1** Output Compare mode	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC2M[3]	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC1M[3]	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	0	0
	TIMx_CCMR1 Input Capture mode	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	IC2F[3:0]				IC2 PSC [1:0]		CC2S [1:0]		IC1F[3:0]				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	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC4M[3]	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC3M[3]	O24CE	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	0	0
	TIMx_CCMR2 Input Capture mode	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	IC4F[3:0]				IC4 PSC [1:0]		CC4S [1:0]		IC3F[3:0]				IC3 PSC [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**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CC4NP	Res.	CC4P	CC4E	CC3NP	Res.	CC3P	CC3E	CC2NP	Res.	CC2P	CC2E	CC1NP	Res.	CC1P	CC1E
	Reset value																	0		0	0	0		0	0	0		0	0	0		0	0

表 222. TIM2/TIM3/TIM4/TIM5 寄存器映射和复位值（续）

偏移	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
0x24	**TIMx_CNT**	CNT[31] or UIFCPY	CNT[30:16]（仅限 TIM2 和 TIM5，其它定时器上保留）															CNT[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
0x28	**TIMx_PSC**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	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]（仅限 TIM2 和 TIM5，其它定时器上保留）																ARR[15:0]															
	Reset value	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0x30	Reserved																																
0x34	**TIMx_CCR1**	CCR1[31:16]（仅限 TIM2 和 TIM5，其它定时器上保留）																CCR1[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
0x38	**TIMx_CCR2**	CCR2[31:16]（仅限 TIM2 和 TIM5，其它定时器上保留）																CCR2[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
0x3C	**TIMx_CCR3**	CCR3[31:16]（仅限 TIM2 和 TIM5，其它定时器上保留）																CCR3[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
0x40	**TIMx_CCR4**	CCR4[31:16]（仅限 TIM2 和 TIM5，其它定时器上保留）																CCR4[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
0x44	Reserved																																
0x48	**TIMx_DCR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DBL[4:0]					Res.	Res.	Res.	DBA[4:0]				
	Reset value																				0	0	0	0	0				0	0	0	0	0
0x4C	**TIMx_DMAR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DMAB[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x50	**TIM2_OR1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TI4_RMP[1:0]		ETR1_RMP	ITR1_RMP
	Reset value																													0	0	0	0

表 222. TIM2/TIM3/TIM4/TIM5 寄存器映射和复位值（续）

偏移	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
0x50	**TIM3_OR1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TI1_RMP[1:0]
	Reset value																																0
0x60	**TIM2_OR2**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ETRSEL [2:0]			Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	Reset value																0	0	0														
0x60	**TIM3_OR2**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ETRSEL [2:0]			Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	Reset value																0	0	0														

有关寄存器边界地址的信息，请参见*第 78 页的第 2.2.2 节*。

36 通用定时器 (TIM15/TIM16/TIM17)

36.1 TIM15/TIM16/TIM17 简介

TIM15/TIM16/TIM17 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。

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。

TIM15/TIM16/TIM17 定时器彼此完全独立，不共享任何资源。如*第 36.4.22 节：定时器同步 (TIM15)* 中所述，TIM15 可以进行同步。

36.2 TIM15 主要特性

TIM15 包括下列特性：

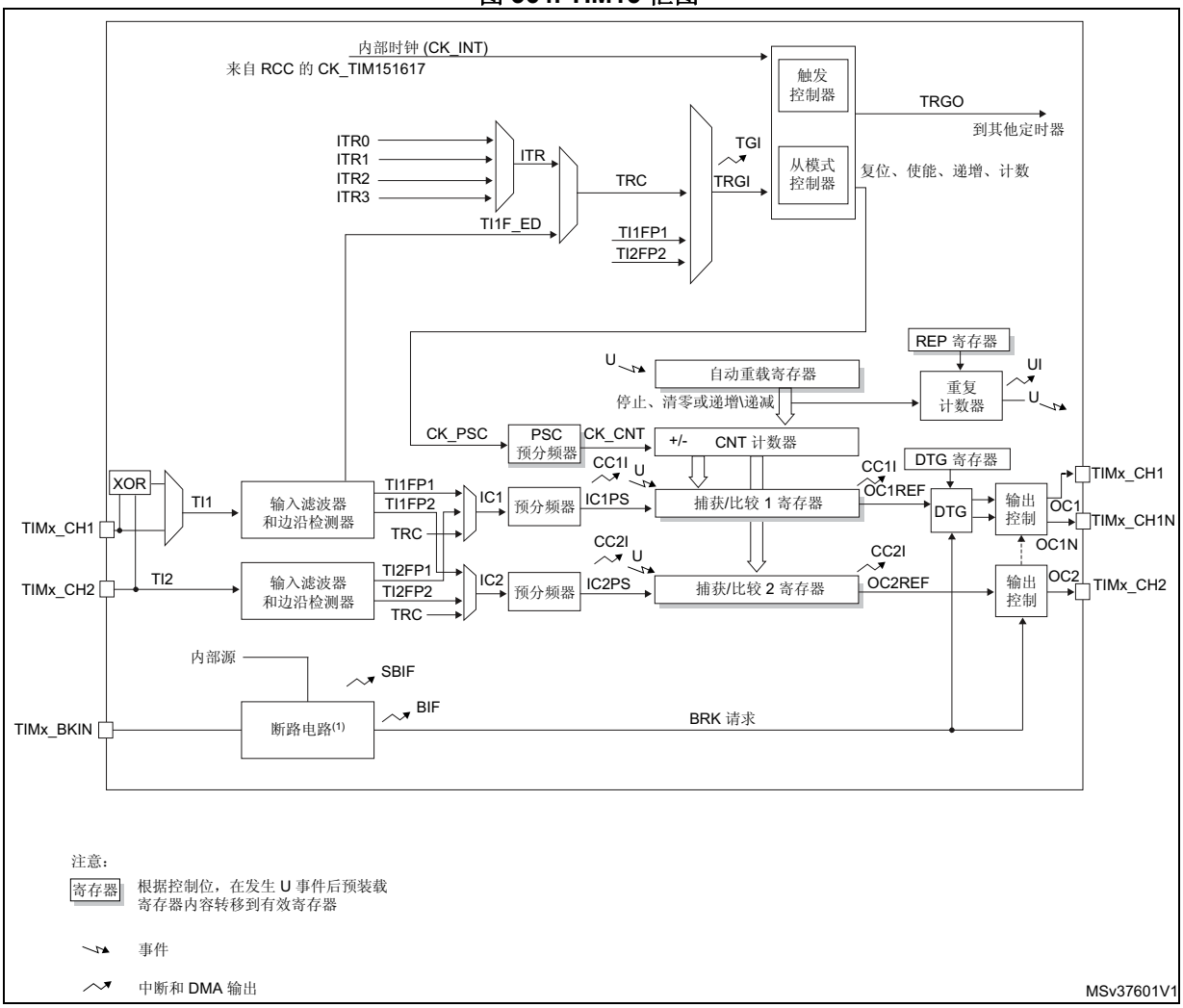
- 16 位自动重载递增计数器
- 16 位可编程预分频器，用于对计数器时钟频率进行分频（可在运行时修改），分频系数介于 1 和 65535 之间
- 多达 2 个独立通道，可用于：
  - 输入捕获
  - 输出比较
  - PWM 生成（边沿模式）
  - 单脉冲模式输出
- 带可编程死区时间的互补输出（仅适用于通道 1）
- 使用外部信号控制定时器且可实现多个定时器互连的同步电路
- 重复计数器，用于仅在给定数目的计数器周期后更新定时器寄存器
- 用于将定时器的输出信号置于复位状态或已知状态的断路输入
- 发生如下事件时生成中断/DMA 请求：
  - 更新：计数器上溢、计数器初始化（通过软件或内部/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（中断请求）

36.3 TIM16/TIM17 主要特性

TIM16/TIM17 定时器包括下列特性：

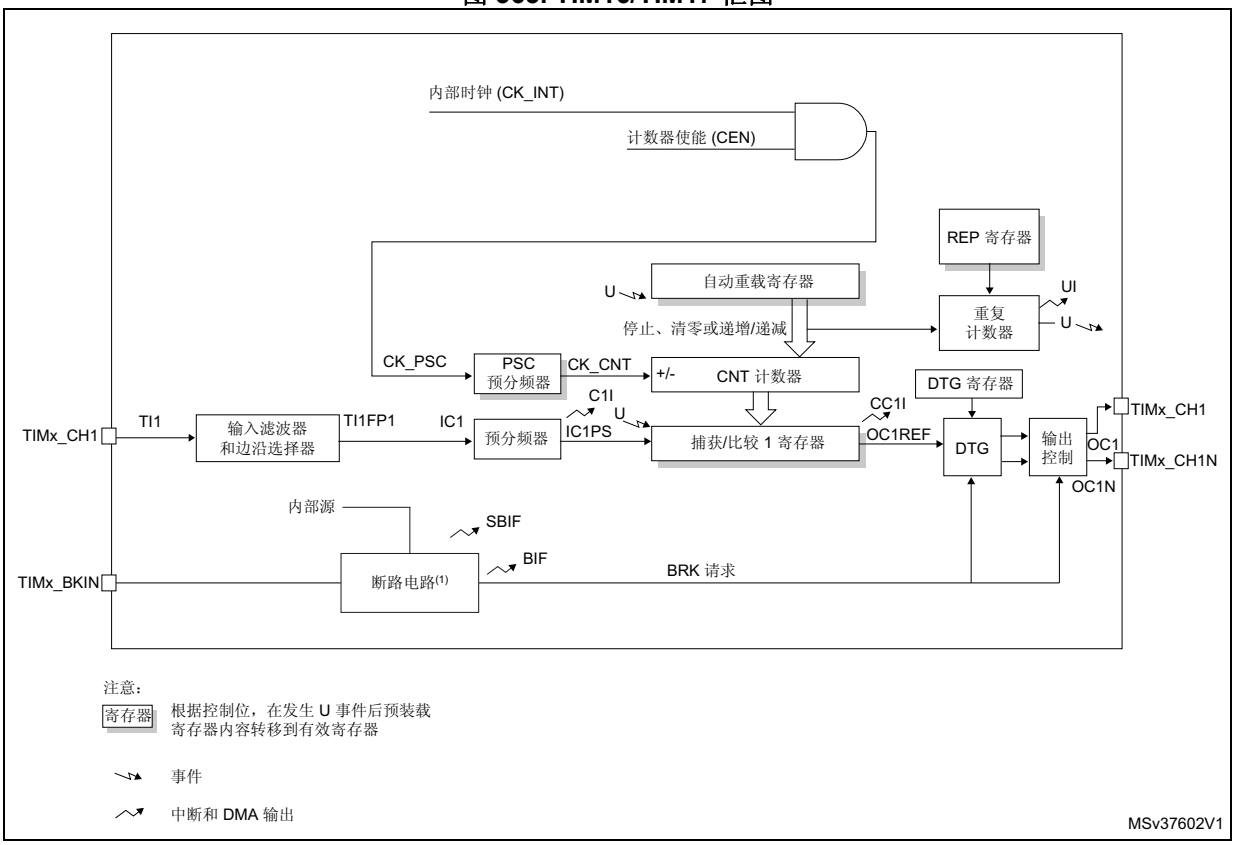
- 16 位自动重载递增计数器
- 16 位可编程预分频器，用于对计数器时钟频率进行分频（可在运行时修改），分频系数介于 1 和 65535 之间
- 一个具有以下用途的通道：
  - 输入捕获
  - 输出比较
  - PWM 生成（边沿对齐模式）
  - 单脉冲模式输出
- 带可编程死区的互补输出
- 重复计数器，用于仅在给定数目的计数器周期后更新定时器寄存器
- 用于将定时器的输出信号置于复位状态或已知状态的断路输入
- 发生如下事件时生成中断/DMA 请求：
  - 更新：计数器上溢
  - 输入捕获
  - 输出比较
  - 断路输入

图 364. TIM15 框图



1. 内部断路事件源可以是：
   - 由 CSS 生成的时钟故障事件。有关 CSS 的更多信息，请参见*第 6.2.10 节：时钟安全系统 (CSS)*
   - PVD 输出
   - SRAM 奇偶校验错误信号
   - Cortex®-M4 LOCKUP (Hardfault) 输出
   - COMP 输出

图 365. TIM16/TIM17 框图



1. 内部断路事件源可以是：
   - 由 CSS 生成的时钟故障事件。有关 CSS 的更多信息，请参见*第 6.2.10 节：时钟安全系统 (CSS)*
   - PVD 输出
   - SRAM 奇偶校验错误信号
   - Cortex®-M4 LOCKUP (Hardfault) 输出
   - COMP 输出

36.4 TIM15/TIM16/TIM17 功能描述

36.4.1 时基单元

可编程高级控制定时器的主要模块是一个 16 位递增计数器及其相关的自动重载寄存器。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 (TIMx_CNT)
- 预分频器寄存器 (TIMx_PSC)
- 自动重载寄存器 (TIMx_ARR)
- 重复计数器寄存器 (TIMx_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 TIMx_CR1 寄存器中的自动重载预装载使能位 (ARPE)。当计数器达到上溢值并且 TIMx_CR1 寄存器中的 UDIS 位为 0 时，将发送更新事件。该更新事件也可由软件产生。下文将针对各配置的更新事件的产生进行详细介绍。

计数器由预分频器输出 CK_CNT 提供时钟，仅当 TIMx_CR1 寄存器中的计数器启动位 (CEN) 置 1 时，才会启动计数器（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 TIMx_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 TIMx_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 366 和*图 367* 以一些示例说明在预分频比实时变化时计数器的行为：

图 366. 预分频器分频由 1 变为 2 时的计数器时序图

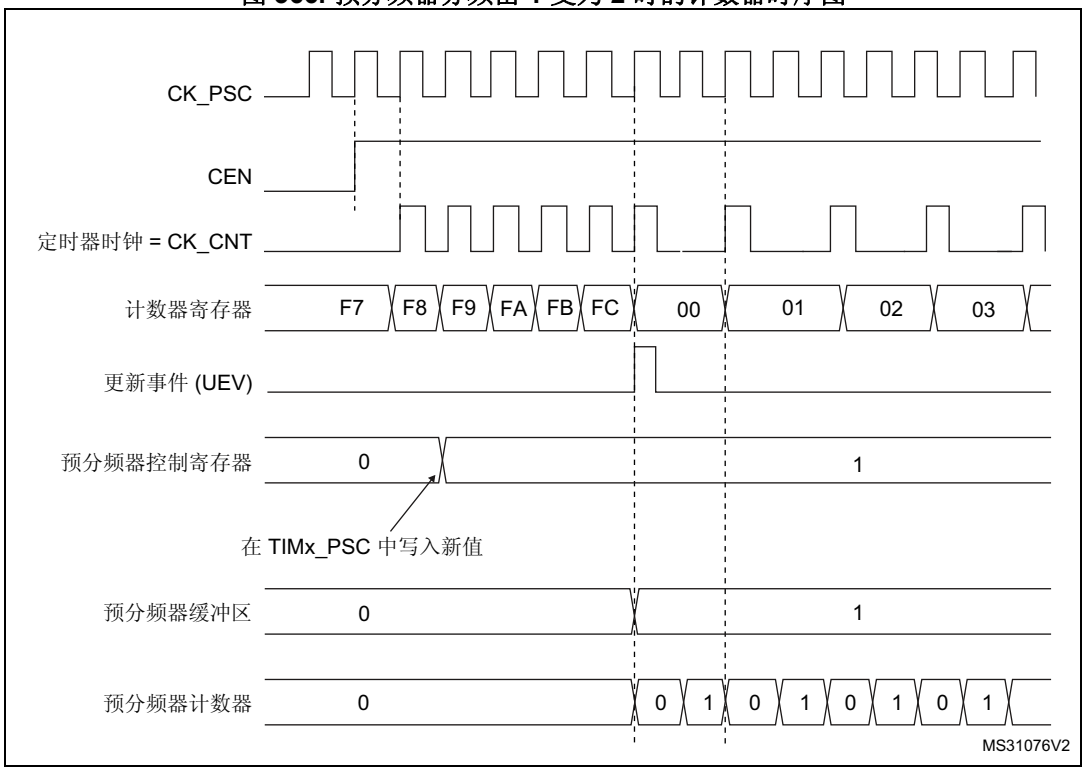
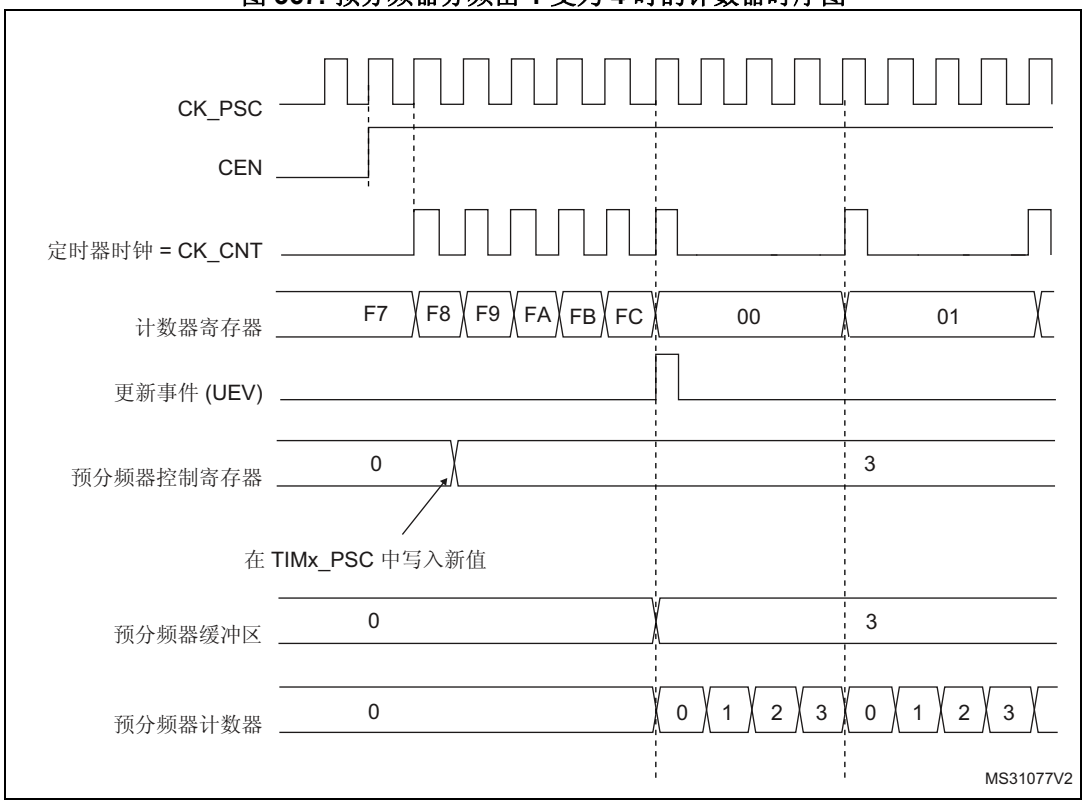


图 367. 预分频器分频由 1 变为 4 时的计数器时序图



36.4.2 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（TIMx_ARR 寄存器的内容），然后重新从 0 开始计数并生成计数器上溢事件。

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 (TIMx_RCR) 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 TIMx_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 TIMx_CR1 寄存器中的 UDIS 位置 1 可禁止 UEV 事件。这可避免向预装载寄存器写入新值时更新影子寄存器。在 UDIS 位写入 0 之前不会产生任何更新事件。不过，计数器和预分频器计数器都会重新从 0 开始计数（而预分频比保持不变）。此外，如果 TIMx_CR1 寄存器中的 URS 位（更新请求选择）已置 1，则将 UG 位置 1 会生成更新事件 UEV，但不会将 UIF 标志置 1（因此，不会发送任何中断或 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（TIMx_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 TIMx_RCR 寄存器的内容。
- 自动重载影子寄存器将以预装载值 (TIMx_ARR) 进行更新。
- 预分频器的缓冲区中将重新装载预装载值（TIMx_PSC 寄存器的内容）。

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

图 368. 计数器时序图，1 分频内部时钟

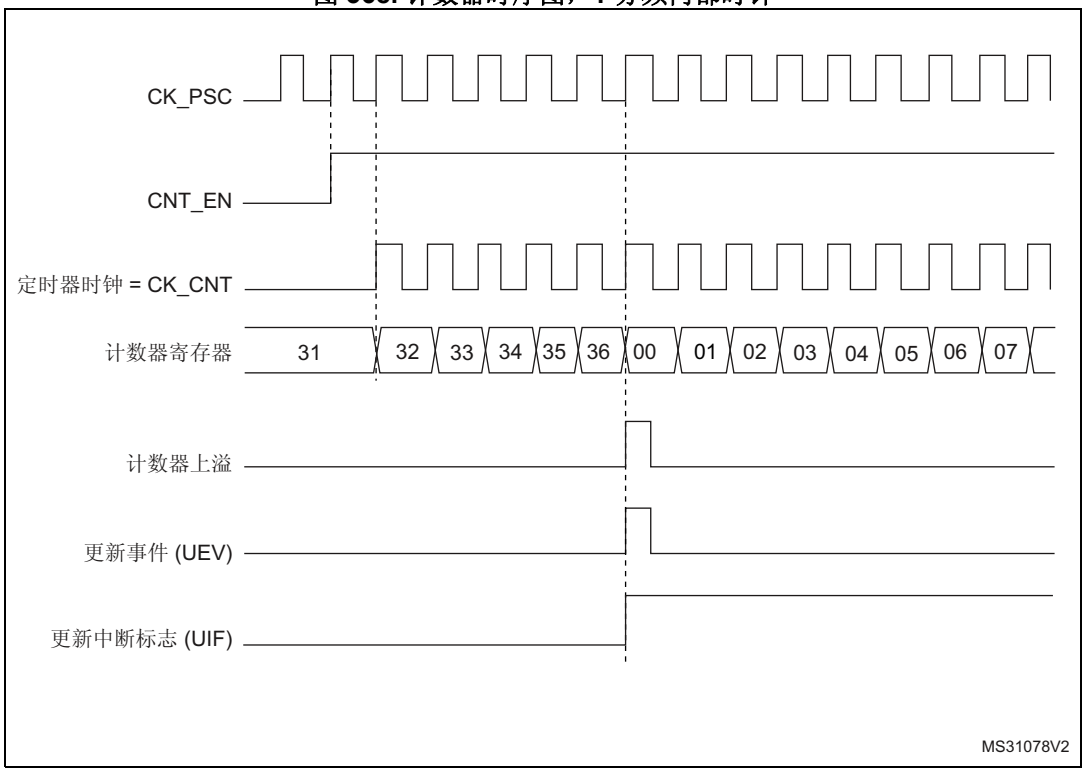


图 369. 计数器时序图，2 分频内部时钟

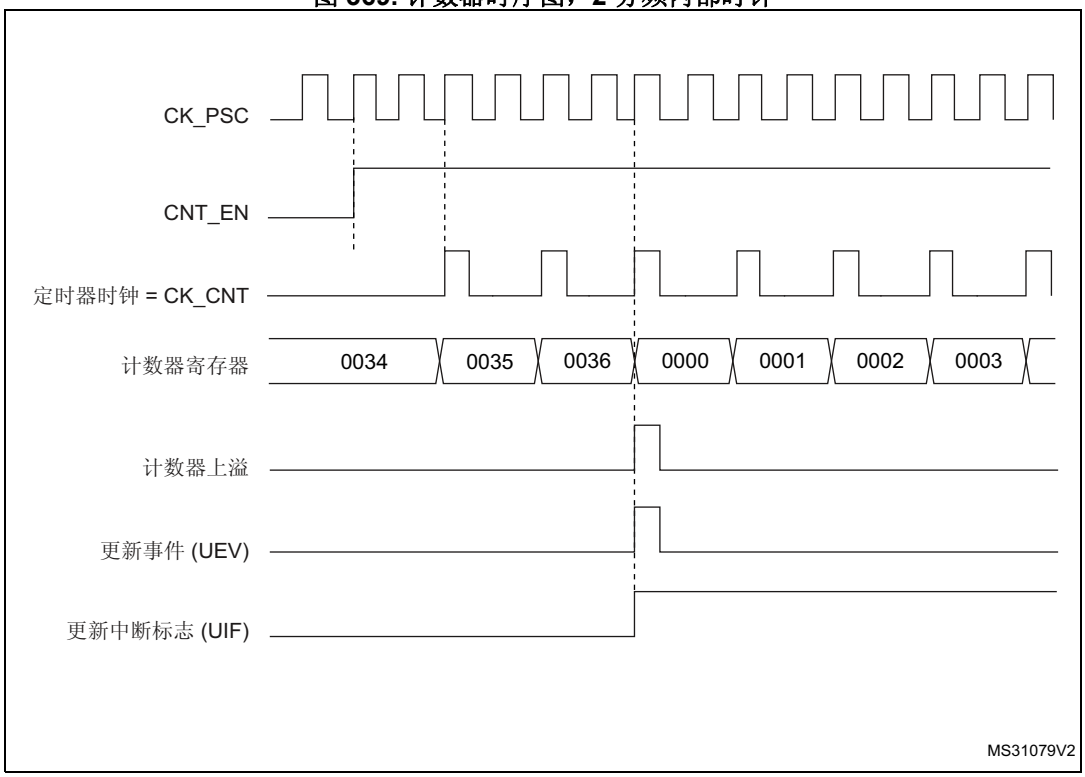


图 370. 计数器时序图，4 分频内部时钟

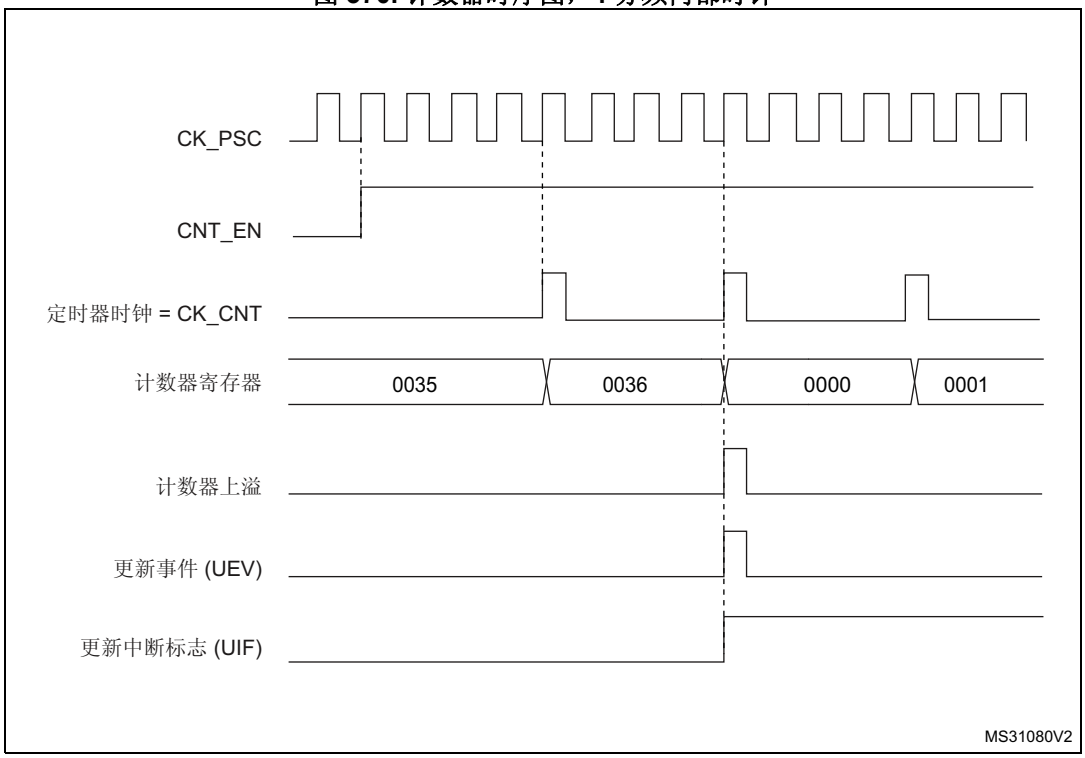


图 371. 计数器时序图，N 分频内部时钟

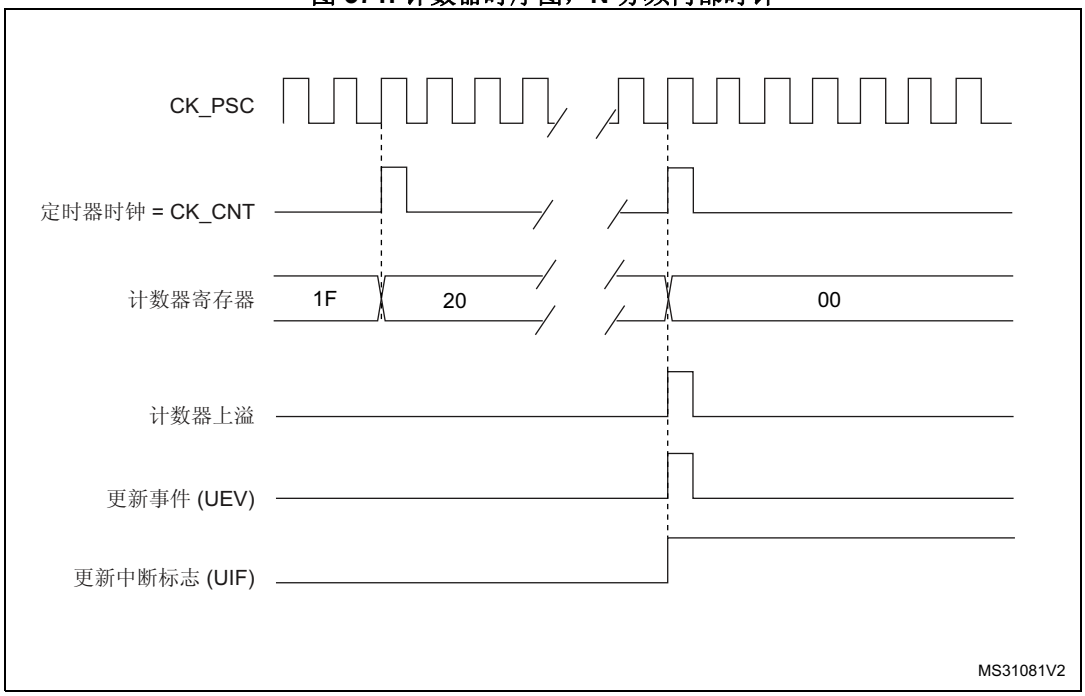


图 372. 计数器时序图，ARPE=0 时更新事件（TIMx_ARR 未预装载）

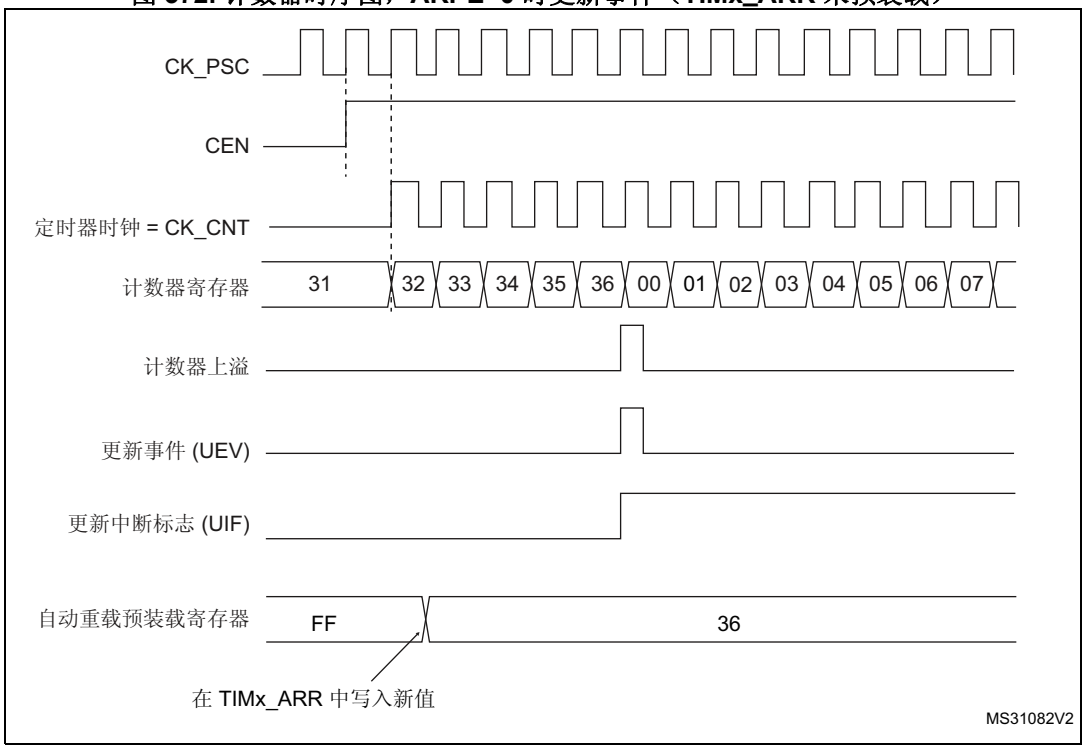
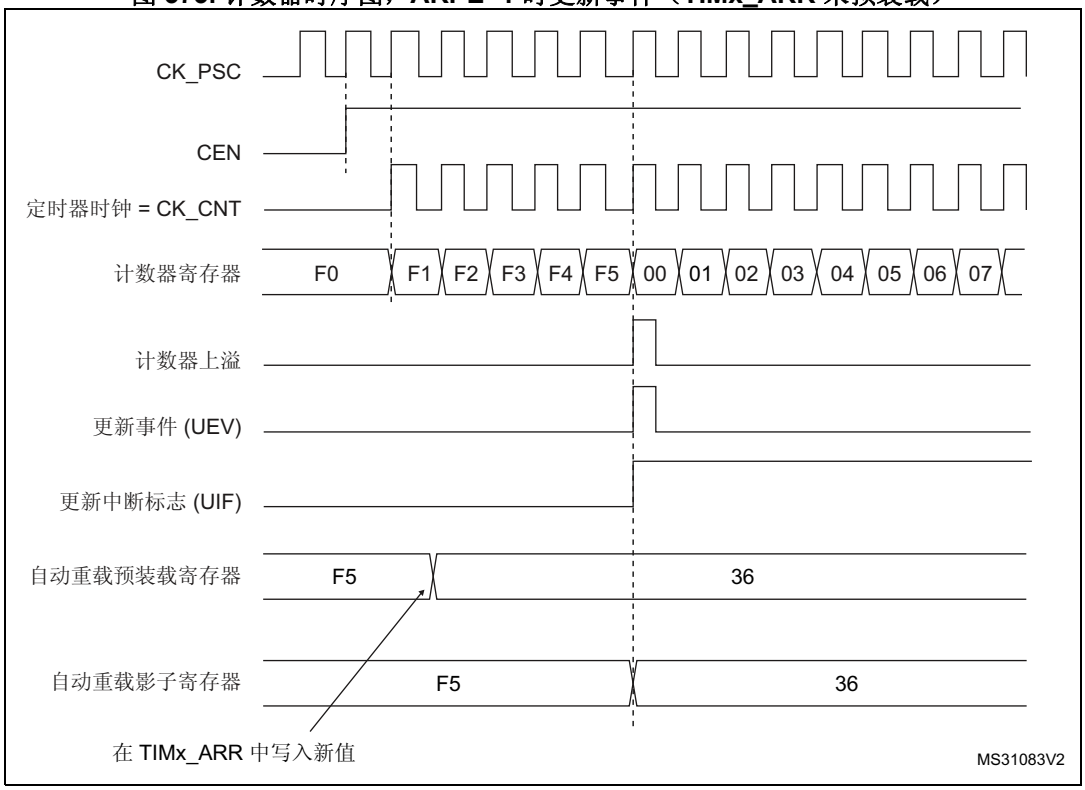


图 373. 计数器时序图，ARPE=1 时更新事件（TIMx_ARR 未预装载）



36.4.3 重复计数器

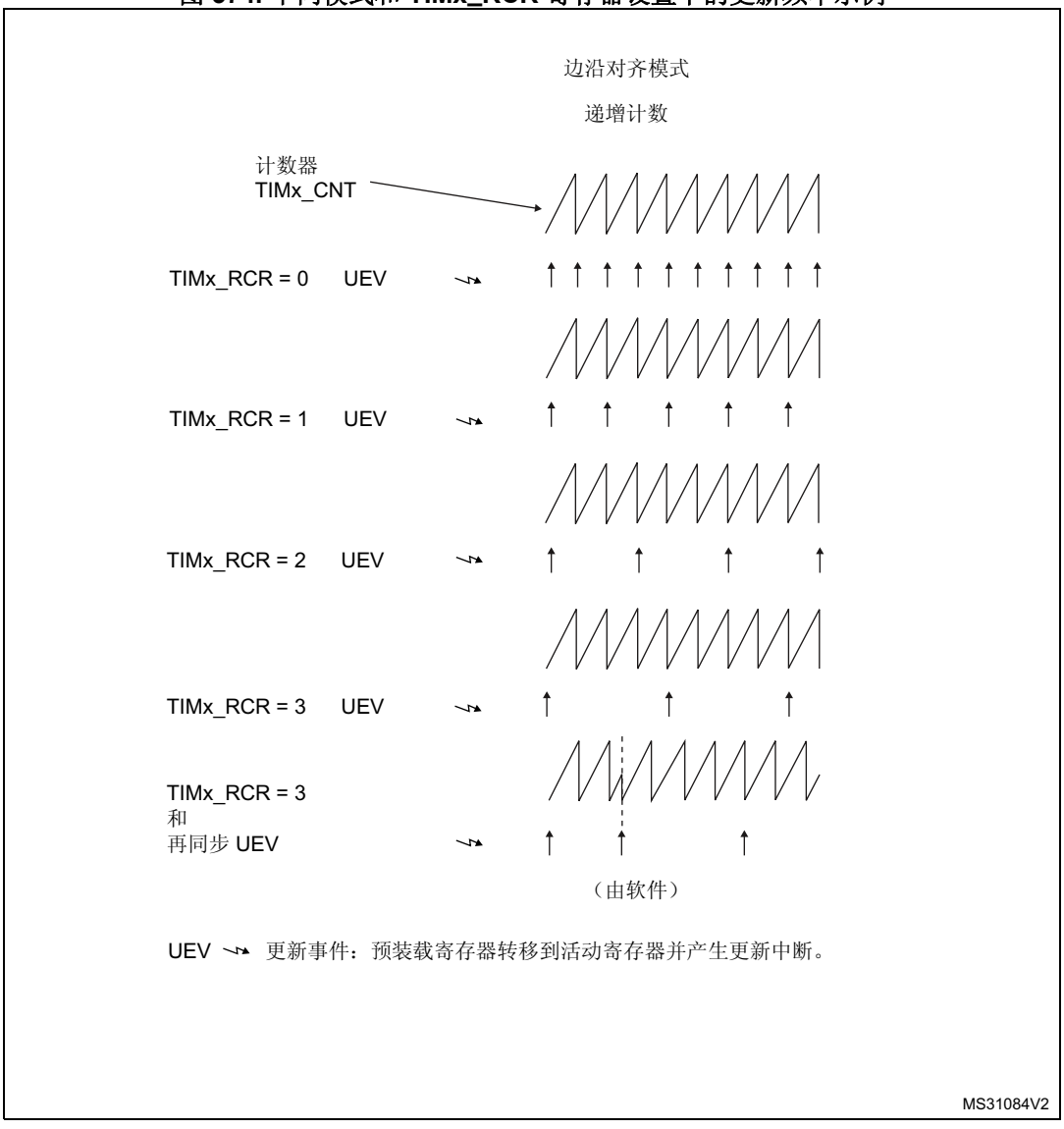
*第 36.4.1 节：时基单元*介绍如何因计数器上溢而生成更新事件 (UEV)。实际上，只有当重复计数器达到零时，才会生成更新事件。这在生成 PWM 信号时很有用。

这意味着，每当发生 N 个计数器上溢（其中，N 是 TIMx_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（TIMx_ARR 自动重载寄存器、TIMx_PSC 预分频器寄存器以及比较模式下的 TIMx_CCRx 捕获/比较寄存器）。

重复计数器在每个计数器上溢时递减。

重复计数器是自动重载类型；其重复率为 TIMx_RCR 寄存器所定义的值（请参见*图 374*）。当更新事件由软件（通过将 TIMx_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 TIMx_RCR 寄存器的内容。

图 374. 不同模式和 TIMx_RCR 寄存器设置下的更新频率示例



36.4.4 时钟选择

计数器时钟可由下列时钟源提供：

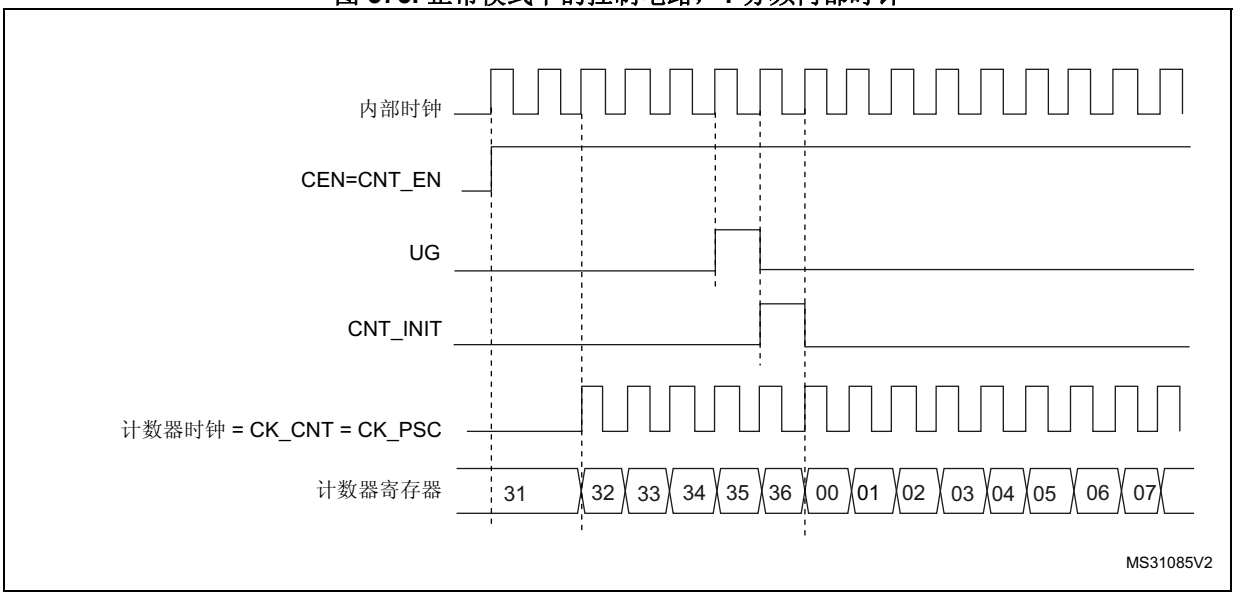
- 内部时钟 (CK_INT)
- 外部时钟模式 1：外部输入引脚
- 内部触发输入 (ITRx)（仅适用于 TIM15）：使用一个定时器作为另一个定时器的预分频器，例如可以将 TIM1 配置为 TIM15 的预分频器。更多详细信息，请参见*第 1179 页的将一个定时器用作另一个定时器的预分频器*。

内部时钟源 (CK_INT)

如果禁止从模式控制器 (SMS=000)，则 CEN 位（TIMx_CR1 寄存器中）和 UG 位（TIMx_EGR 寄存器中）为实际控制位，并且只能通过软件进行更改（UG 除外，仍保持自动清零）。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 375 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）。

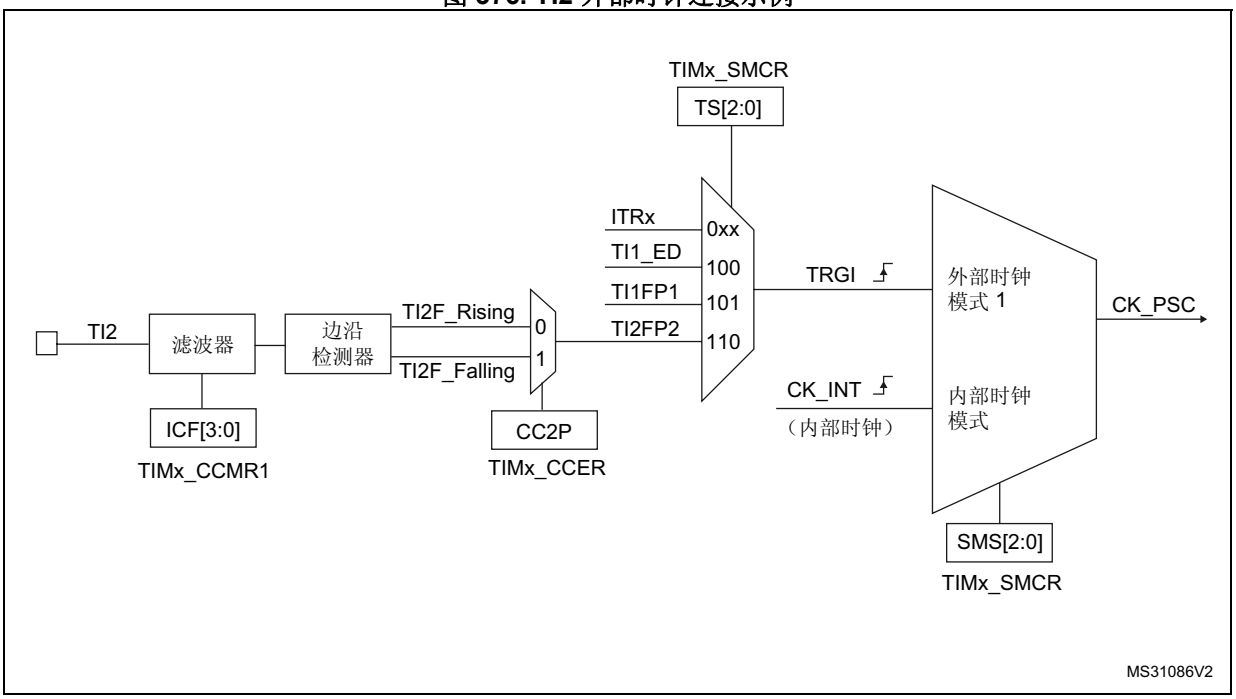
图 375. 正常模式下的控制电路，1 分频内部时钟



外部时钟源模式 1

当 TIMx_SMCR 寄存器中的 SMS=111 时，可选择此模式。计数器可在选定的输入信号上出现上升沿或下降沿时计数。

图 376. 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：

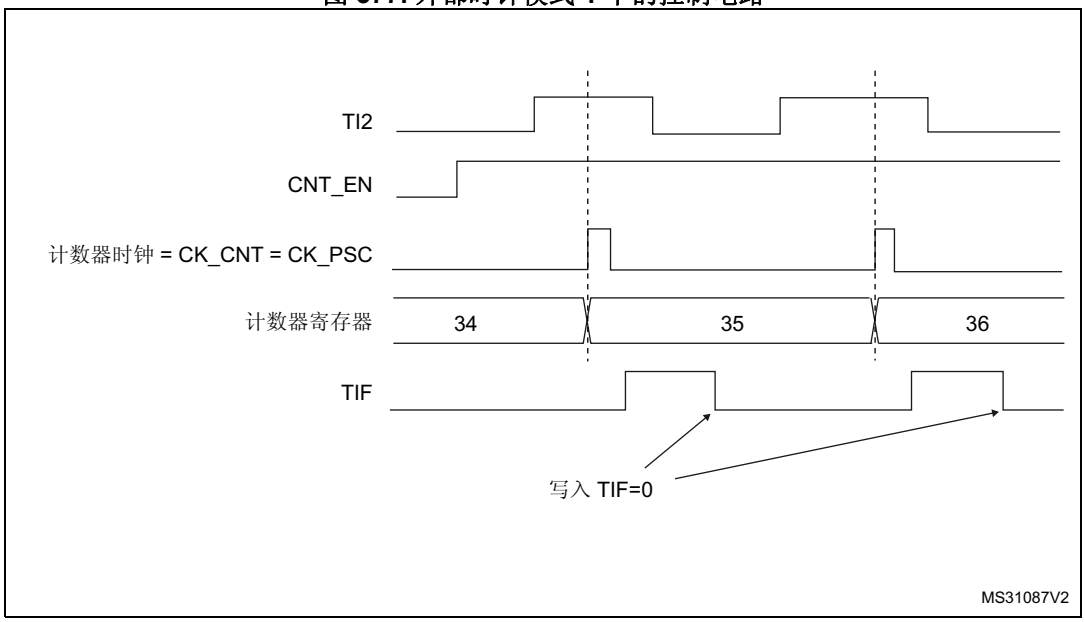
1. 通过在 TIMx_CCMR1 寄存器中写入 CC2S = “01” 来配置通道 2，使其能够检测 TI2 输入的上升沿。
2. 通过在 TIMx_CCMR1 寄存器中写入 IC2F[3:0] 位来配置输入滤波带宽（如果不需要任何滤波器，请保持 IC2F=0000）。
3. 通过在 TIMx_CCER 寄存器中写入 CC2P = 0 来选择上升沿极性。
4. 通过在 TIMx_SMCR 寄存器中写入 SMS=111，使定时器在外部时钟模式 1 下工作。
5. 通过在 TIMx_SMCR 寄存器中写入 TS=110 来选择 TI2 作为触发输入源。
6. 通过在 TIMx_CR1 寄存器中写入 CEN=1 来使能计数器。

注： *由于捕获预分频器不用于触发操作，因此无需对其进行配置。*

当 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 377. 外部时钟模式 1 下的控制电路



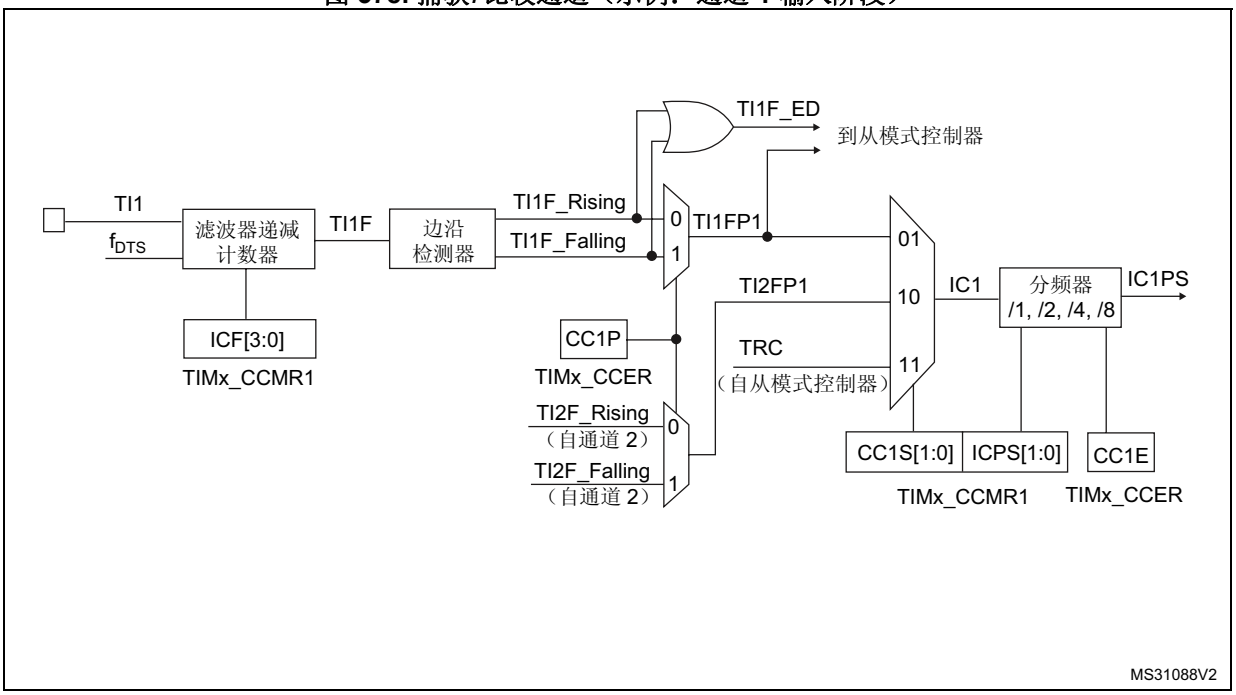
36.4.5 捕获/比较通道

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 378 到*图 381* 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 378. 捕获/比较通道（示例：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）。链的末端决定最终输出信号的极性。

图 379. 捕获/比较通道 1 主电路

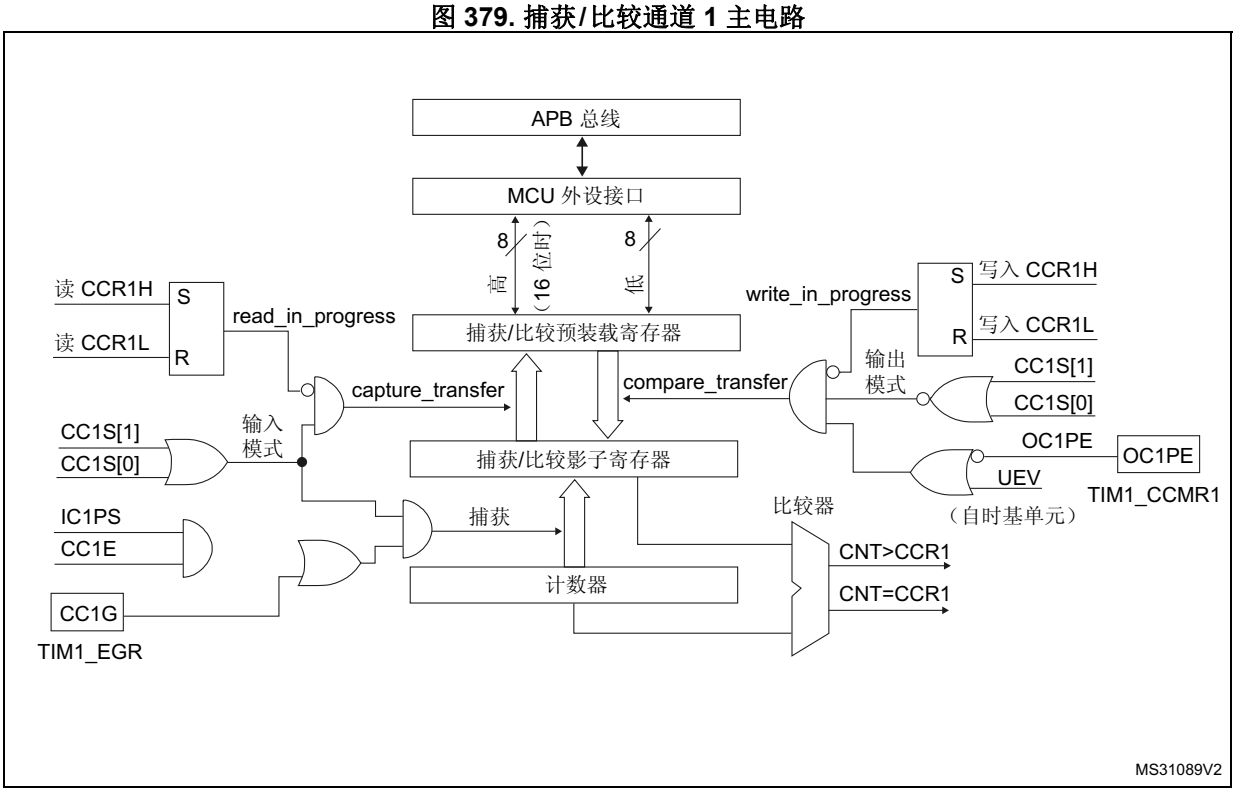


图 380. 捕获/比较通道的输出阶段（通道 1）

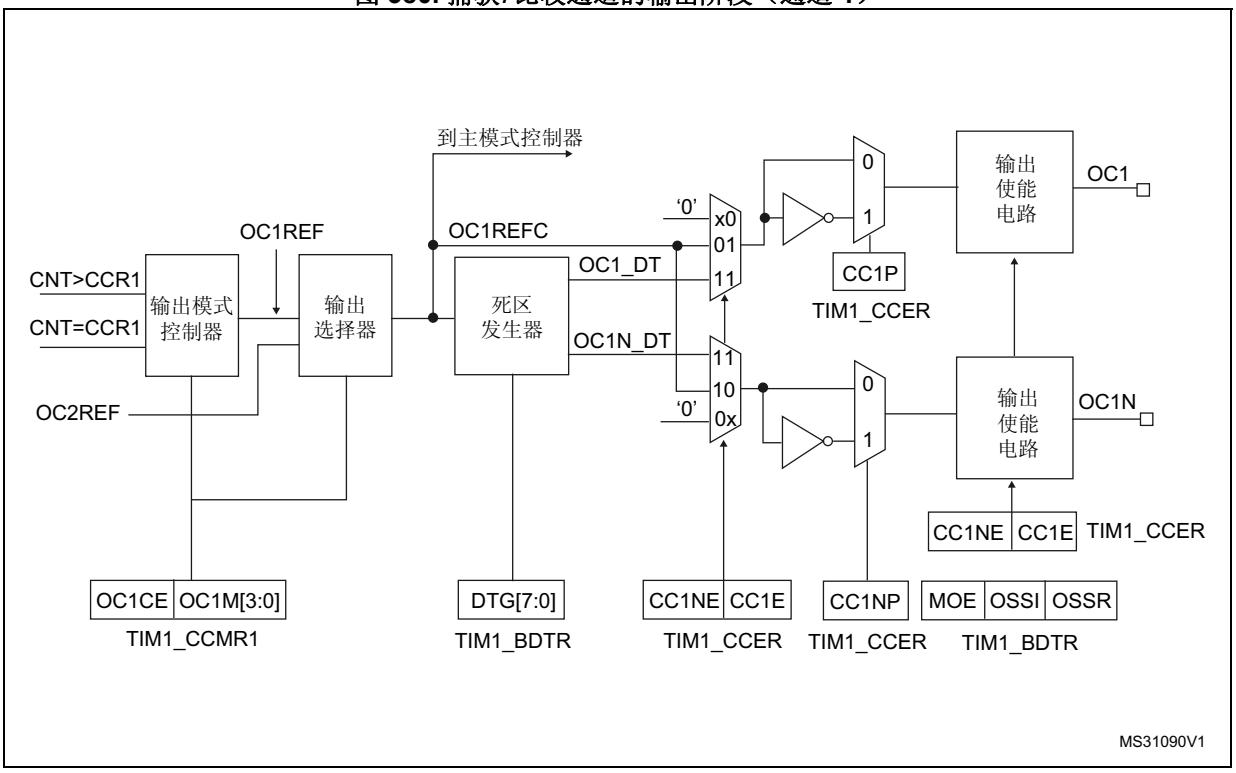
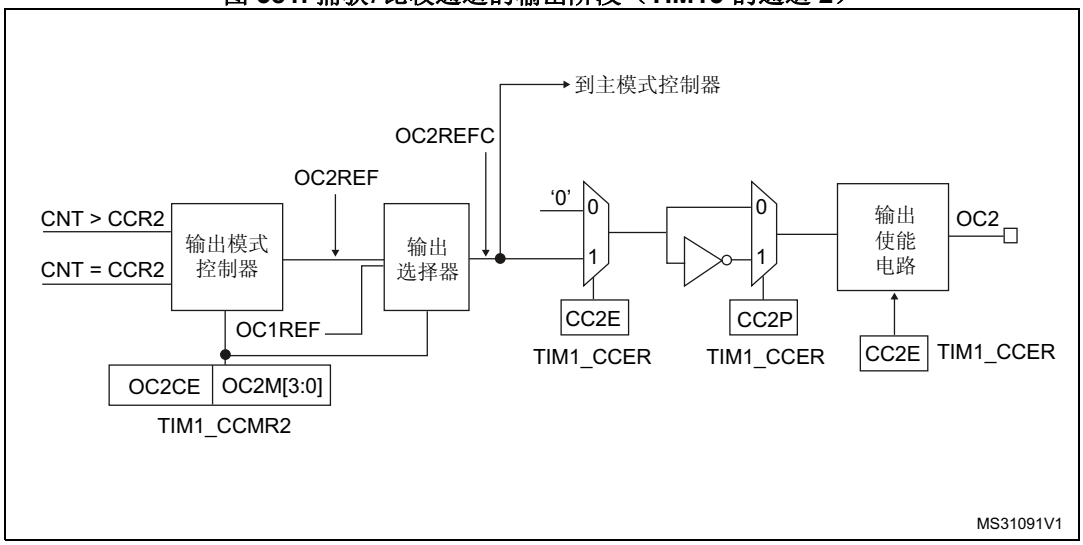


图 381. 捕获/比较通道的输出阶段（TIM15 的通道 2）



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。

36.4.6 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 (TIMx_CCRx) 来锁存计数器的值。发生捕获事件时，会将相应的 CCXIF 标志（TIMx_SR 寄存器）置 1，并可发送中断或 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（TIMx_SR 寄存器）置 1。可通过软件将 CCxIF 清零，方法是：向 CCxIF 写入“0”，或读取存储在 TIMx_CCRx 寄存器中的已捕获数据。向 CCxOF 写入“0”后会将其清零。

以下示例说明了如何在 TI1 输入出现上升沿时将计数器的值捕获到 TIMx_CCR1 中。具体操作步骤如下：

1. 选择有效输入：TIMx_CCR1 必须连接到 TI1 输入，因此向 TIMx_CCMR1 寄存器中的 CC1S 位写入 01。只要 CC1S 不等于 00，就会将通道配置为输入模式，并且 TIMx_CCR1 寄存器将处于只读状态。
2. 根据连接到定时器的信号，对所需的输入滤波带宽进行编程（如果输入为 TIx 之一，则对 TIMx_CCMRx 寄存器中的 ICxF 位进行编程）。假设信号边沿变化时，输入信号最多在 5 个内部时钟周期内发生抖动。因此，我们必须将滤波带宽设置为大于 5 个内部时钟周期。在检测到 8 个具有新电平的连续采样（以 f_{DTS} 频率采样）后，可以确认 TI1 上的跳变沿。然后向 TIMx_CCMR1 寄存器中的 IC1F 位写入 0011。
3. 通过在 TIMx_CCER 寄存器中将 CC1P 位写入 0，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 TIMx_CCMR1 寄存器中的 IC1PS 位写入“00”）。
5. 通过将 TIMx_CCER 寄存器中的 CC1E 位置 1，允许将计数器的值捕获到捕获寄存器中。
6. 如果需要，可通过将 TIMx_DIER 寄存器中的 CC1IE 位置 1 来使能相关中断请求，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，TIMx_CCR1 寄存器会获取计数器的值。
- 将 CC1IF 标志置 1（中断标志）。如果至少发生了两次连续捕获，但 CC1IF 标志未被清零，这样 CC1OF 捕获溢出标志会被置 1。
- 根据 CC1IE 位生成中断。
- 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： *通过软件将 TIMx_EGR 寄存器中的相应 CCxG 位置 1 可生成 IC 中断和/或 DMA 请求。*

36.4.7 PWM 输入模式（仅适用于 TIM15）

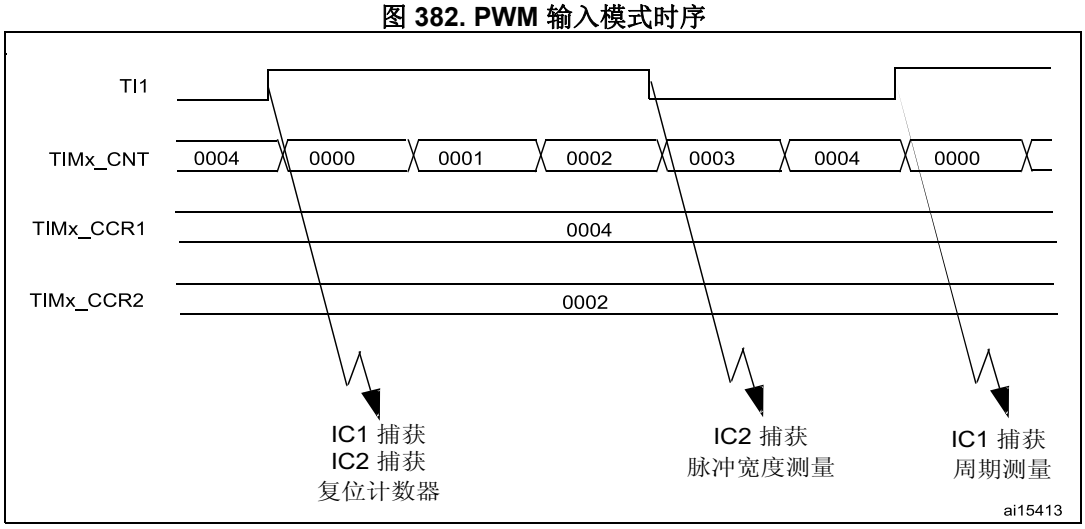
此模式是输入捕获模式的一个特例。其实现步骤与输入捕获模式基本相同，仅存在以下不同之处：

- 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 TIMx_CCR1 寄存器中）和占空比（位于 TIMx_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 TIMx_CCR1 的有效输入：向 TIMx_CCMR1 寄存器中的 CC1S 位写入 01（选择 TI1）。
2. 选择 TI1FP1 的有效极性（用于在 TIMx_CCR1 中捕获和计数器清零）：向 CC1P 位和 CC1NP 位写入“0”（上升沿有效）。
3. 选择 TIMx_CCR2 的有效输入：向 TIMx_CCMR1 寄存器中的 CC2S 写入 10 （选择 TI1）。
4. 选择 TI1FP2 的有效极性（用于在 TIMx_CCR2 中捕获）：向 CC2P 位和 CC2NP 位写入“1”（下降沿有效）。
5. 选择有效触发输入：向 TIMx_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 TIMx_SMCR 寄存器中的 SMS 位写入 100。
7. 使能捕获：向 TIMx_CCER 寄存器中的 CC1E 位和 CC2E 位写入“1”。

图 382. PWM 输入模式时序



1. PWM 输入模式只能与 TIMx_CH1/TIMx_CH2 信号配合使用，因为只有 TI1FP1 和 TI2FP2 与从模式控制器相连。

36.4.8 强制输出模式

在输出模式（TIMx_CCMRx 寄存器中的 CCxS 位 = 00）下，可直接由软件将每个输出比较信号（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 TIMx_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 TIMx_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，TIMx_CCRx 影子寄存器与计数器之间的比较仍会执行，而且允许将标志置 1。因此可发送相应的中断和 DMA 请求。下面的输出比较模式一节对此进行了介绍。

36.4.9 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（TIMx_CCMRx 寄存器中的 OCxM 位）和输出极性（TIMx_CCER 寄存器中的 CCxP 位）定义。匹配时，输出引脚既可保持其电平 (OCXM=000)，也可设置为有效电平 (OCXM=001)、无效电平 (OCXM=010) 或进行翻转 (OCxM=011)。
- 将中断状态寄存器中的标志置 1（TIMx_SR 寄存器中的 CCxIF 位）。
- 如果相应中断使能位（TIMx_DIER 寄存器中的 CCXIE 位）置 1，将生成中断。
- 如果相应使能位（TIMx_DIER 寄存器的 CCxDE 位，TIMx_CR2 寄存器的 CCDS 位，用来选择 DMA 请求）置 1，将发送 DMA 请求。

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

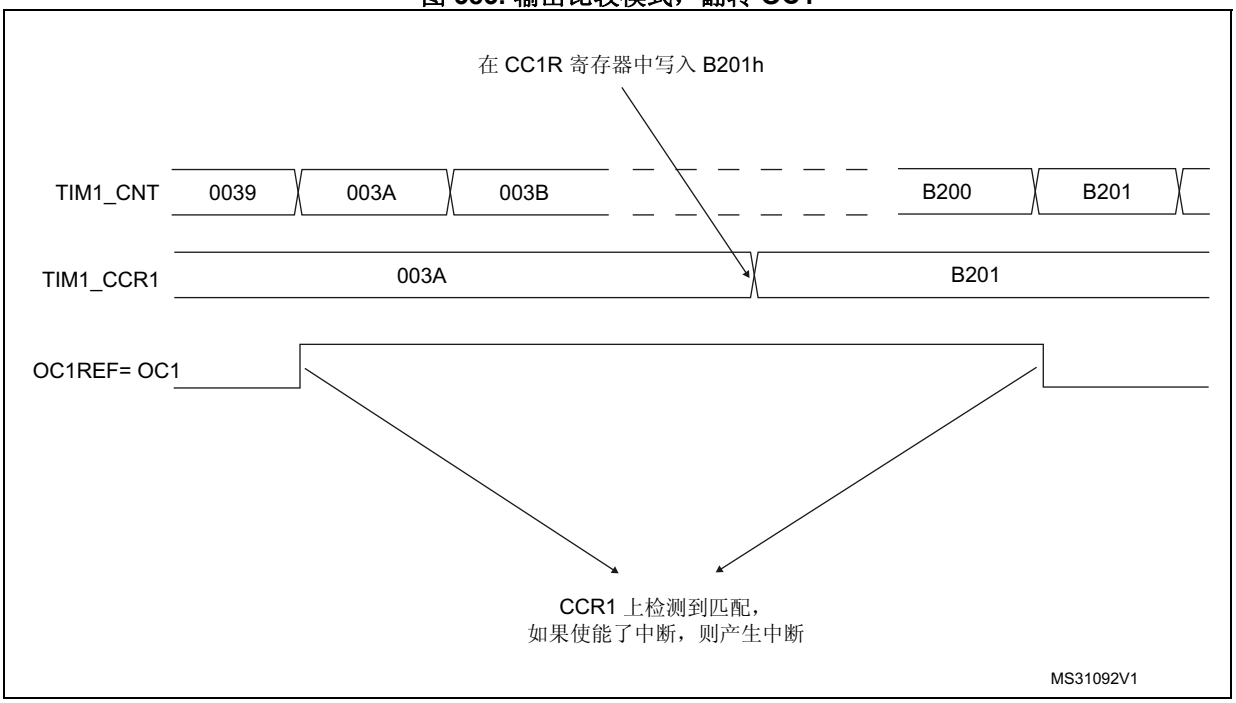
在输出比较模式下，更新事件 UEV 对 OCxREF 和 OCx 输出毫无影响。同步的精度可以达到计数器的一个计数周期。输出比较模式也可用于输出单脉冲（在单脉冲模式下）。

步骤

1. 选择计数器时钟（内部、外部、预分频器）。
2. 在 TIMx_ARR 和 TIMx_CCRx 寄存器中写入所需数据。
3. 如果要生成中断请求，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 TIMx_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 TIMx_CCRx 寄存器以控制输出波形，前提是未使能预装载寄存器（OCxPE=“0”，否则 TIMx_CCRx 影子寄存器仅在下一更新事件 UEV 发生时进行更新）。*图 383* 给出了一个示例。

图 383. 输出比较模式，翻转 OC1



36.4.10 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 TIMx_ARR 寄存器值决定，其占空比则由 TIMx_CCRx 寄存器值决定。

各通道可以独立选择 PWM 模式（每个 OCx 输出对应一个 PWM），只需向 TIMx_CCMRx 寄存器的 OCxM 位写入“110”（PWM 模式 1）或“111”（PWM 模式 2）。必须通过将 TIMx_CCMRx 寄存器中的 OCxPE 位置 1 使能相应预装载寄存器，最后通过将 TIMx_CR1 寄存器中的 ARPE 位置 1 使能自动重载预装载寄存器（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 TIMx_EGR 寄存器中的 UG 位置 1 来初始化所有寄存器。

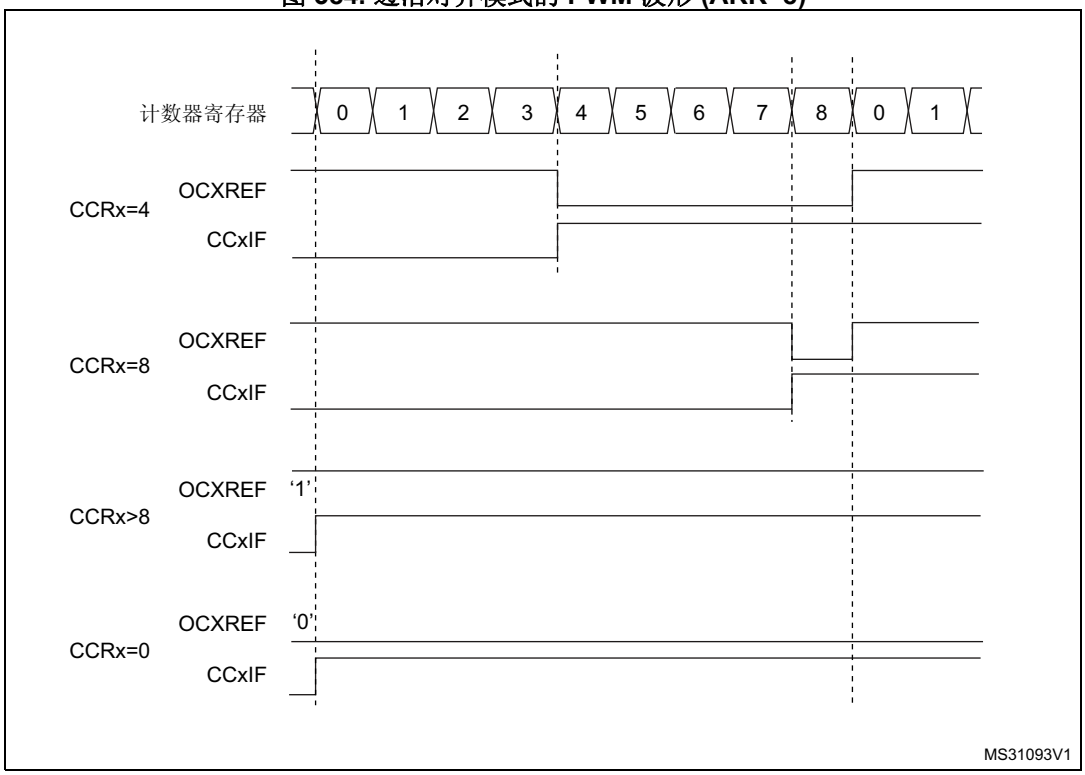
OCx 极性可通过软件来编程（使用 TIMx_CCER 寄存器的 CCxP 位）。可将其编程为高电平有效或低电平有效。通过 CCxE、CCxNE、MOE、OSSI 和 OSSR 位（TIMx_CCER 和 TIMx_BDTR 寄存器）的组合使能 OCx 输出。有关详细信息，请参见 TIMx_CCER 寄存器说明。

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

TIM15/TIM16/TIM17 只能递增计数。请参见*第 1217 页的递增计数模式*。

以下以 PWM 模式 1 为例。只要 TIMx_CNT < TIMx_CCRx，PWM 参考信号 OCxREF 便为高电平，否则为低电平。如果 TIMx_CCRx 中的比较值大于自动重载值（TIMx_ARR 中），则 OCxREF 保持为“1”。如果比较值为 0，则 OCxRef 保持为“0”。*图 384* 举例介绍边沿对齐模式的一些 PWM 波形 (TIMx_ARR=8)。

图 384. 边沿对齐模式的 PWM 波形 (ARR=8)



36.4.11 组合 PWM 模式（仅适用于 TIM15）

在组合 PWM 模式下，生成的两个边沿或中心对齐 PWM 信号的各个脉冲间允许存在可编程延时和相移。频率由 TIMx_ARR 寄存器的值确定，而占空比和延时则由两个 TIMx_CCRx 寄存器确定。产生的信号 OCxREFC 由两个参考 PWM 的逻辑或运算或者逻辑与运算组合组成。

- OC1REFC（或 OC2REFC）由 TIMx_CCR1 和 TIMx_CCR2 寄存器控制

两个通道可以独立选择组合 PWM 模式（每对 CCR 寄存器一个 OCx 输出），只需向 TIMx_CCMRx 寄存器的 OCxM 位写入“1100”（组合 PWM 模式 1）或“1101”（组合 PWM 模式 2）。

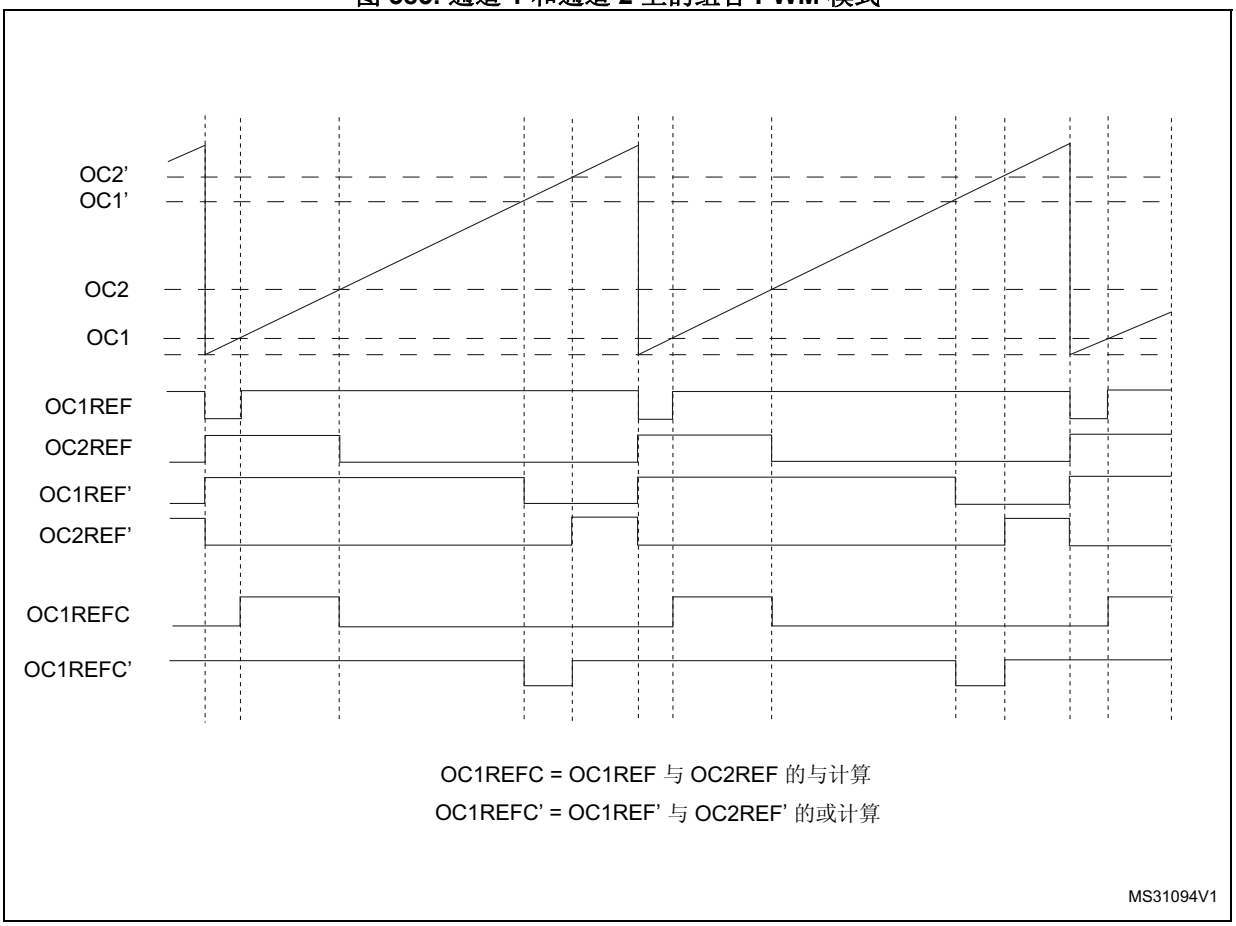
当给定通道用作组合 PWM 通道时，其互补通道必须在相反的 PWM 模式下配置（例如，一个通道在组合 PWM 模式 1 下配置，另一个通道在组合 PWM 模式 2 下配置）。

注： *出于兼容性原因，OCxM[3:0] 位域分为两部分，最高有效位与最低有效的 3 位不相邻。*

图 385 显示了不对称 PWM 模式下可以产生的信号示例，通过以下配置可获得这些信号：

- 通道 1 在组合 PWM 模式 2 下配置
- 通道 2 在 PWM 模式 1 下配置

图 385. 通道 1 和通道 2 上的组合 PWM 模式



36.4.12 互补输出和死区插入

TIM15/TIM16/TIM17 通用定时器可以输出一路互补信号，并管理输出的关断和接通。

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 TIMx_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：TIMx_CCER 寄存器中的 CCxE 和 CCxNE 位以及 TIMx_BDTR 和 TIMx_CR2 寄存器中的 MOE、OISx、OISxN、OSSI 和 OSSR 位。更多详细信息，请参见*第 1283 页的表 227：具有断路功能的互补通道 OCx 和 OCxN 的输出控制位 (TIM16/17)*。应当注意，切换至空闲状态（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。每个通道有一个 10 位死区发生器。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 386. 带死区插入的互补输出

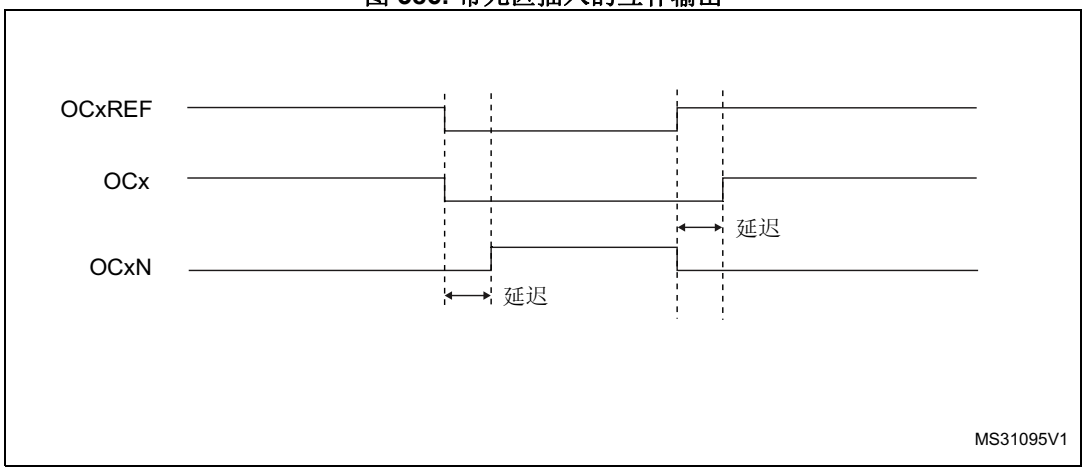


图 387. 延迟时间大于负脉冲宽度的死区波形

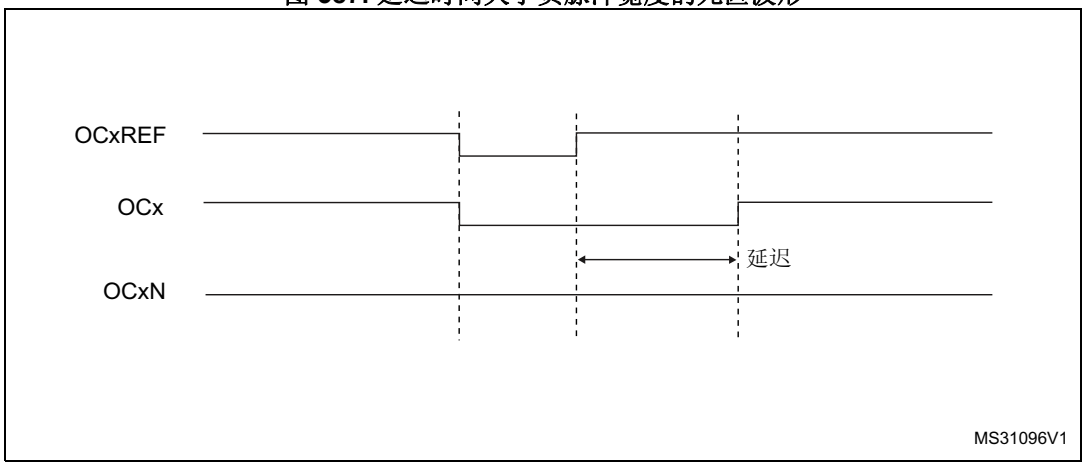
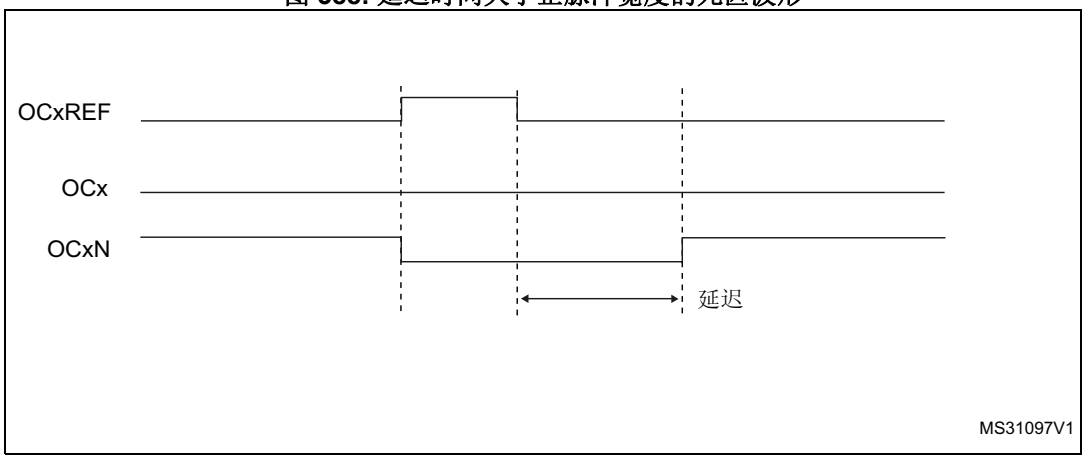


图 388. 延迟时间大于正脉冲宽度的死区波形



死区延迟对于所有通道均相同，可通过 TIMx_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*第 1286 页的第 36.6.13 节：TIM16/TIM17 断路和死区寄存器 (TIMx_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 TIMx_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： *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。*

36.4.13 使用断路功能

断路功能的目的是保护由 TIM15/TIM16/TIM17 定时器生成的 PWM 信号所驱动的电源开关。断路输入通常被连接到功率级和三相逆变器的故障输出。激活时，断路电路会关闭 PWM 输出，并将其强制为预定义的安全状态。

断路通道收集系统级故障（时钟失效和奇偶校验错误等）和应用故障（来自输入引脚和内置比较器），可以在死区持续时间后将输出强制为预定义的电平（有效或无效）。

断路期间的输出使能信号和输出电平取决于多个控制位：

- TIMx_BDTR 寄存器中的 MOE 位，允许通过软件使能/禁止输出，在发生断路和断路 2 事件时复位。
- TIMx_BDTR 寄存器中的 OSSI 位，定义定时器将输出控制在无效状态下，还是释放对 GPIO 控制器的控制（通常使其处于高阻态模式）。
- TIMx_CR2 寄存器中的 OISx 和 OISxN 位，将输出设置为关断电平（有效或无效）。无论 OISx 和 OISxN 的值为何，均无法在给定时间将 OCx 和 OCxN 输出同时设置为有效电平。更多详细信息，请参见 *第 1262 页的表 225：具有断路功能的互补通道 OCx 和 OCxN 的输出控制位 (TIM15)*。

退出复位状态后，断路功能处于禁止状态，MOE 位处于低电平。通过设置 TIMx_BDTR 寄存器中的 BKE 位来使能断路功能。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 TIMx_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。

断路可以从 TIMx_OR2 寄存器的多个源产生，这些源可以单独使能并使用可编程边沿有效性。

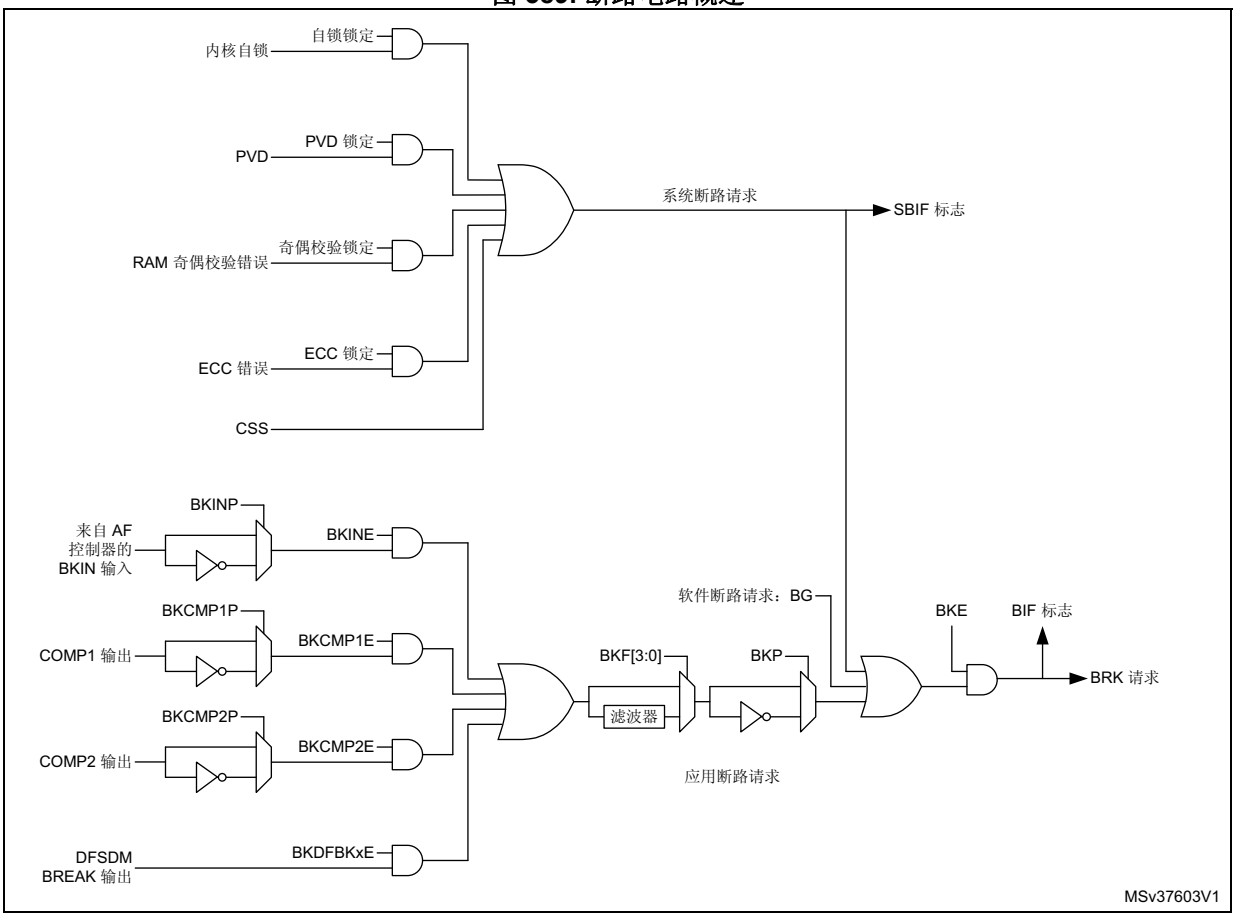
断路 (BRK) 通道的源为：

- 连接到 BKIN 引脚的外部源（由 AFIO 控制器设定），具有极性选择和可选的数字滤波
- 内部源：
  - Cortex®-M4 LOCKUP 输出
  - PVD 输出
  - SRAM 奇偶校验错误信号
  - Flash ECC 错误
  - CSS 检测器产生时钟故障事件
  - 比较器的输出，具有极性选择和可选的数字滤波
  - DFSDM1 外设的模拟看门狗输出

也可由软件通过 TIMx_EGR 寄存器中的 BG 位产生断路事件。

在所有源进入定时器 BRK 输入之前，对其进行 OR 运算，如以下*图 389* 所示。

图 389. 断路电路概述



注： *只有禁止可编程滤波器时才能保证异步（无时钟）操作。如果使能可编程滤波器，必须使用故障安全时钟模式（例如，使用内部 PLL 和/或 CSS）来保证能够处理断路事件。*

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、空闲状态甚至释放对 AFIO 控制器的控制（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 TIMx_CR2 寄存器 OISx 位中编程的电平驱动每个输出通道。如果 OSSI = 0，定时器将释放输出控制（由 AFIO 控制器接管），否则使能输出保持高电平。
- 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_tim 时钟周期）。
  - 如果 OSSI = 0，定时器将释放使能输出（由强制高阻态的 AFIO 控制器接管），否则使能输出将保持高电平或在 CCxE 或 CCxNE 位之一为高电平时立即变为高电平。
- 将断路状态标志（TIMx_SR 寄存器中的 BIF 位）置 1。如果 TIMx_DIER 寄存器中的 BIE 位置 1，可产生中断。如果 TIMx_DIER 寄存器中的 BDE 位置 1，可发送 DMA 请求。
- 如果 TIMx_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。可以将断路输入连接到功率驱动器的警报、温度传感器或任何安全元件。

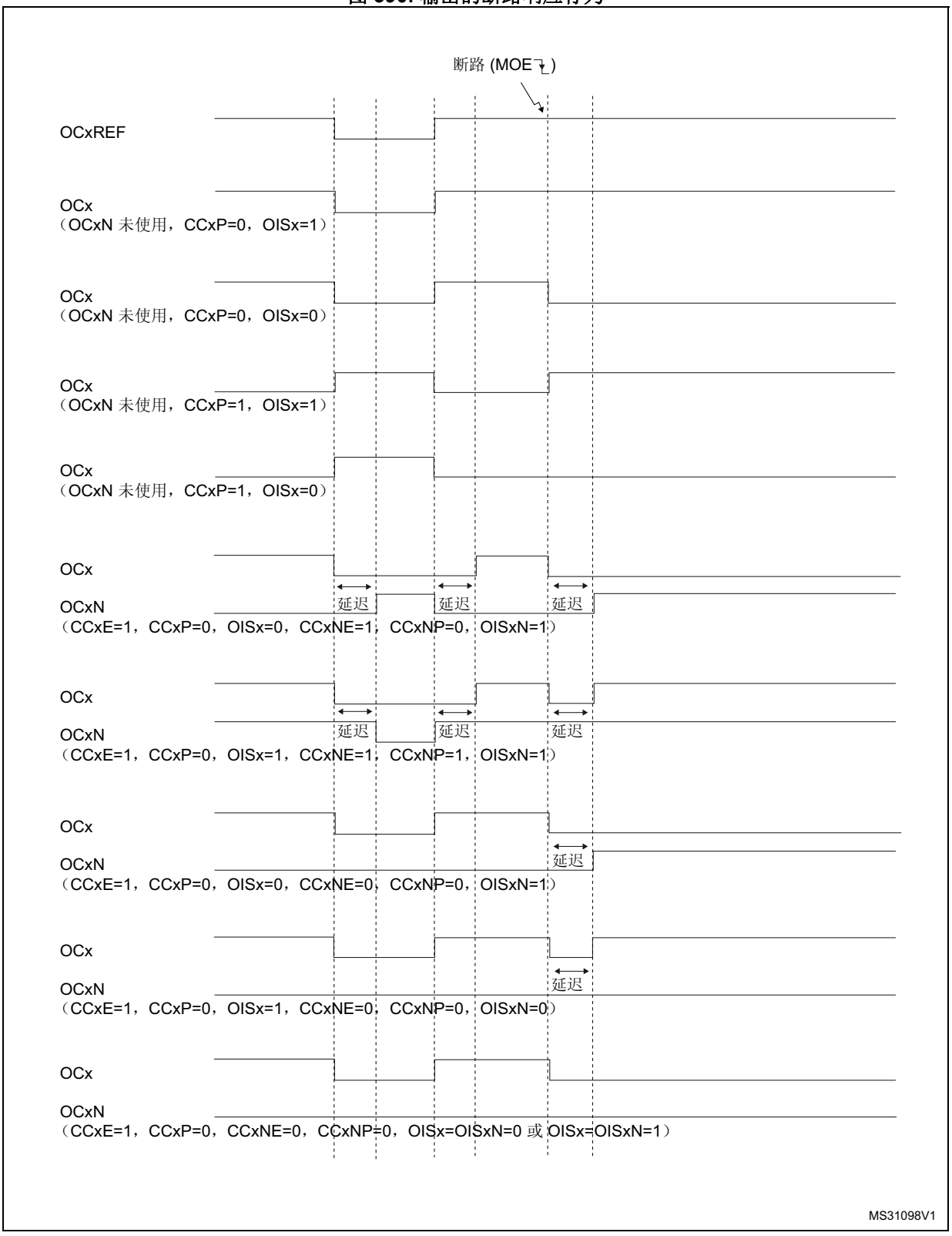
注： *断路输入为电平有效。因此，当断路输入有效电平时，不能将 MOE 位置 1（自动或通过软件）。同时，不能将状态标志 BIF 清零。*

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 TIMx_BDTR 寄存器中。

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 TIMx_BDTR 寄存器中的 LOCK 位，从 3 种保护级别中进行选择。请参见*第 1286 页的第 36.6.13 节：TIM16/TIM17 断路和死区寄存器 (TIMx_BDTR)*。MCU 复位后只能对 LOCK 位执行一次写操作。

图 390 所示为输出对断路响应行为的示例。

图 390. 输出的断路响应行为



36.4.14 双向断路输入

TIM15/TIM16/TIM17 具有双向断路 I/O，如*图 391* 所示。

它们可以：

- 将板级全局断路信号用于向外部 MCU 或栅极驱动器发送故障信号，唯一的引脚作为输入和输出状态引脚。
- 在必须将多个内部和外部断路输入合并时，将内部断路源和多个外部开漏比较器输出“或”连接在一起，触发唯一断路事件。

使用 TIMxBDTR 寄存器的 BKBID 位将断路输入配置为双向模式。可以使用 TIMxBDTR 寄存器中的 LOCK 位，将 BKBID 编程位锁定在只读模式（处于锁定级别 1 或更高级别）。

双向模式需要将 I/O 配置为开漏模式且使极性低电平有效（使用 BKINP 和 BKP 位）。任何来自系统（例如 CSS）、片上外设或断路输入的断路请求都会强制将断路输入置为低电平，以通知发生了故障事件。如果未正确设置极性位（高电平有效极性），则出于安全目的禁止双向模式。

软件断路事件 (BG) 也会导致断路 I/O 被强制为“0”，从而向外部组件指示定时器已进入断路状态。但是仅在断路使能时 (BKE = 1) 有效。当生成软件断路事件且 BKE = 0 时，输出将被置于安全状态，并且断路标志置 1，但对断路 I/O 无影响。

安全解除机制可防止系统被最终锁定（断路输入上的低电平会触发断路，进而将相同输入强制置为低电平）。

当 BKDSRM 位置 1 时，会释放断路输出以清除故障信号，从而使系统能够重新启动。

在任何情况下都不能禁止断路保护电路：

- 断路输入路径始终有效：即使 BKDSRM 位置 1 且释放开漏控制，断路事件也仍然有效。这样可以在发生断路期间防止 PWM 输出重新启动。
- 使能输出（MOE 位置 1）后，BKDSRM 位不能解除断路保护（请参见*表 223*）。

表 223. 断路保护解除条件

MOE	BKDIR	BKDSRM	断路保护状态
0	0	X	启动
0	1	0	启动
0	1	1	解除
1	X	X	启动

启动和重新启动断路电路

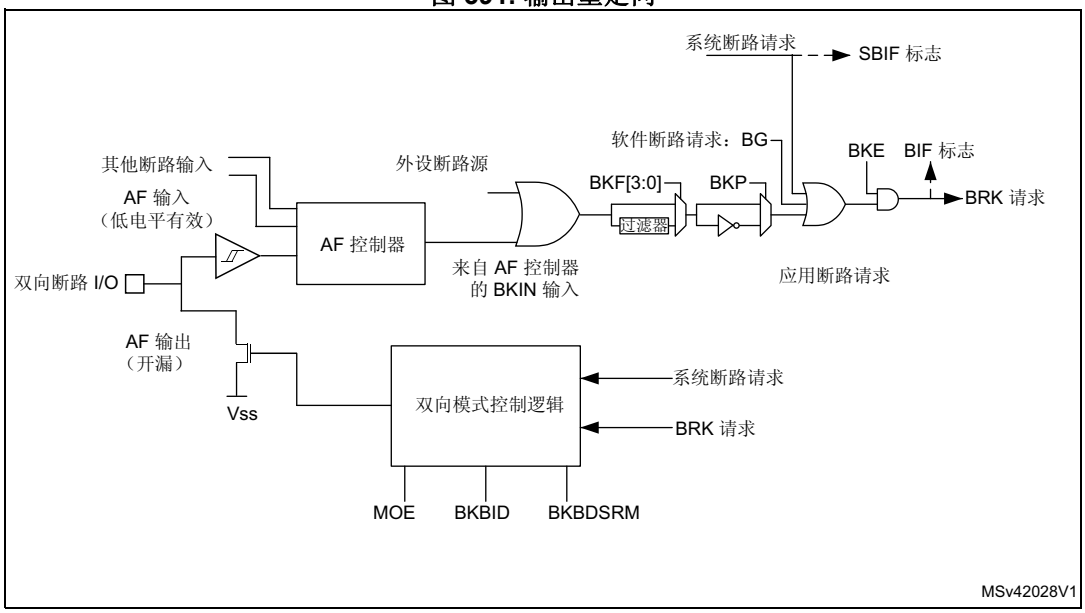
默认情况下（外设复位配置）会启动断路电路（在输入或双向模式下）。

发生断路事件后，必须按照以下步骤重新启动保护：

- 必须将 BKDSRM 位置 1，以释放输出控制
- 软件必须等待系统断路条件消失（如果有），并清零 SBIF 状态标志（或在重新启动前由系统清零）
- 软件必须轮询 BKDSRM 位，直到该位由硬件清零（当应用断路条件消失时）

此后，断路电路即启动并激活，可以通过将 MOE 位置 1 来重新使能 PWM 输出。

图 391. 输出重定向



36.4.15 单脉冲模式

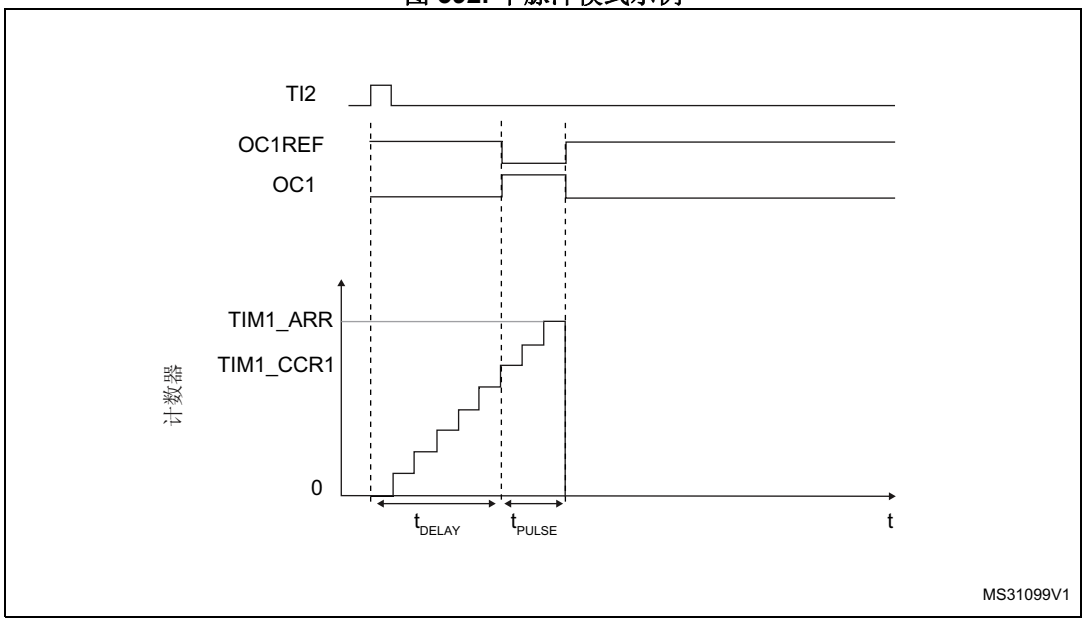
单脉冲模式 (OPM) 是上述模式的一个特例。在这种模式下，计数器可以在一个激励信号的触发下启动，并可在一段可编程的延时后产生一个脉宽可编程的脉冲。

可以通过从模式控制器启动计数器。可以在输出比较模式或 PWM 模式下生成波形。将 TIMx_CR1 寄存器中的 OPM 位置 1，即可选择单脉冲模式。这样，发生下一更新事件 UEV 时，计数器将自动停止。

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

- CNT < CCRx ≤ ARR（特别注意，0 < CCRx）

图 392. 单脉冲模式示例



例如，用户希望达到这样的效果：在 TI2 输入引脚检测到上升沿时，经过 t_{DELAY} 的延迟，在 OC1 上产生一个长度为 t_{PULSE} 的正脉冲。

使用 TI2FP2 作为触发 1：

1. 在 TIMx_CCMR1 寄存器中写入 CC2S=“01”，以将 TI2FP2 映射到 TI2。
2. 在 TIMx_CCER 寄存器中写入 CC2P=“0”和 CC2NP=“0”，使 TI2FP2 能够检测上升沿。
3. 在 TIMx_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 TIMx_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）。

- t_{DELAY} 由写入 TIMx_CCR1 寄存器的值定义。
- t_{PULSE} 由自动重载值与比较值之差 (TIMx_ARR - TIMx_CCR1) 来定义。
- 假设希望产生这样的波形：信号在发生比较匹配时从“0”变为“1”，在计数器达到自动重载值时由“1”变为“0”。为此，应在 TIMx_CCMR1 寄存器中写入 OC1M=111，以使能 PWM 模式 2。如果需要，可选择在 TIMx_CCMR1 寄存器的 OC1PE 和 TIMx_CR1 寄存器的 ARPE 中写入“1”，以使能预装载寄存器。这种情况下，必须在 TIMx_CCR1 寄存器中写入比较值并在 TIMx_ARR 寄存器中写入自动重载值，通过将 UG 位置 1 来产生更新，然后等待 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，因此应向 TIMx_CR1 寄存器的 OPM 位写入“1”，以便在发生下一更新事件（计数器翻转从自动重载值返回到 0）时使计数器停止计数。

特殊情况：OCx 快速使能

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

如果要输出延迟时间最短的波形，可以将 TIMx_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。

36.4.16 可再触发单脉冲模式 (OPM)（仅限 TIM15）

该模式允许计数器可以在一个激励信号的触发下启动，并且能产生长度可编程的脉冲，但与不可再触发单脉冲模式间存在以下差别，如*第 36.4.15 节*所述：

- 发生触发时，脉冲立即产生（无可编程延时）
- 如果在上一个触发完成前发生新的触发，脉冲将延长

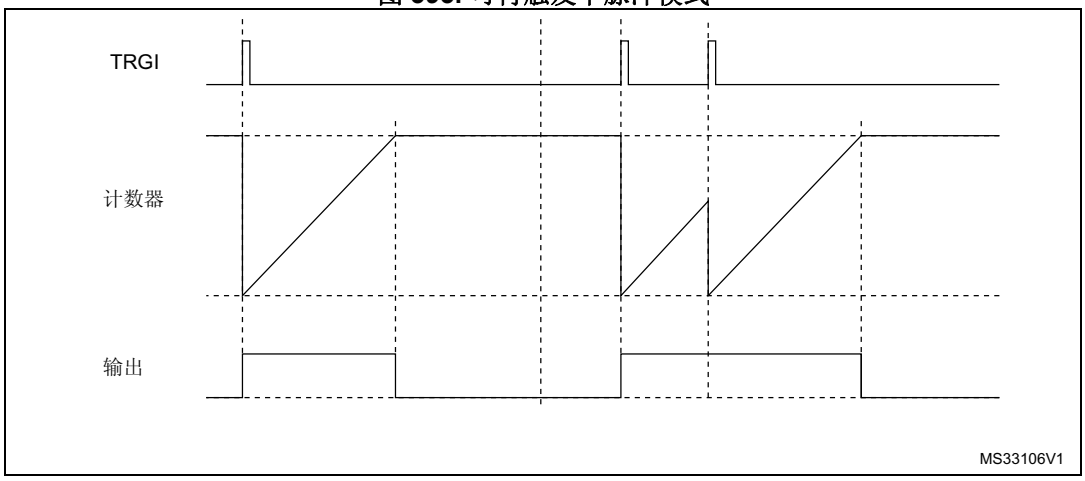
定时器必须处于从模式，TIMx_SMCR 寄存器中的位 SMS[3:0] = “1000”（组合复位 + 触发模式），针对可再触发 OPM 模式 1 或模式 2 将 OCxM[3:0] 位设置为“1000”或“1001”。

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

注： *出于兼容性原因，OCxM[3:0] 和 SMS[3:0] 位域分为两部分，最高有效位与最低有效的 3 位不相邻。*

此模式不能与中心对齐 PWM 模式一起使用。在 TIMx_CR1 中必须设置 CMS[1:0] = 00。

图 393. 可再触发单脉冲模式



36.4.17 UIF 位重映射

TIMx_CR1 寄存器中的 IUFREMAP 位强制将更新中断标志 UIF 连续复制到定时计数器寄存器的位 31 (TIMxCNT[31]) 中。这样便可自动读取计数器值以及由 UIFCPY 标志发出的电位翻转条件。在特定情况下，这可避免在后台任务（计数器读）和中断（更新中断）之间共享处理时产生竞争条件，从而简化计算。

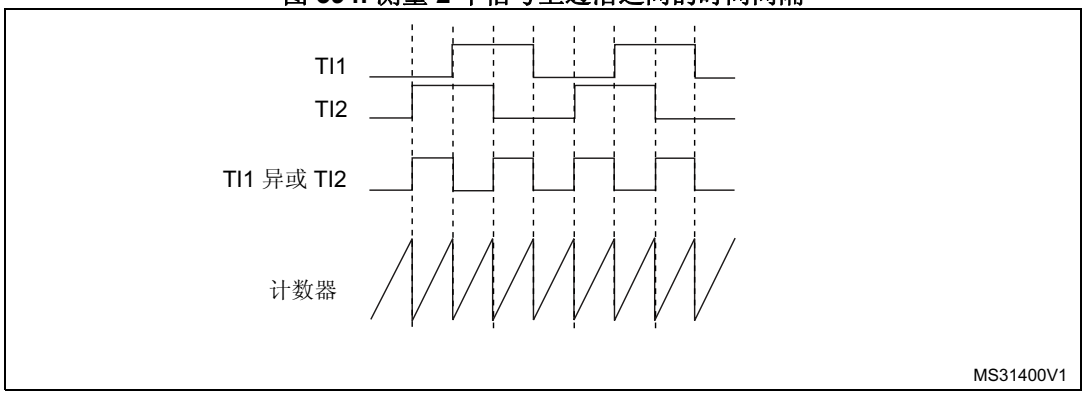
UIF 和 UIFCPY 标志使能之间没有延迟。

36.4.18 定时器输入异或功能（仅适用于 TIM15）

通过 TIMx_CR2 寄存器中的 TI1S 位，可将通道 1 的输入滤波器连接到异或门的输出，从而将 TIMx_CH1 和 TIMx_CH2 这两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。这样可用于测量两个输入信号上边沿之间的间隔，如下面的*图 394* 所示。

图 394. 测量 2 个信号上边沿之间的时间间隔



36.4.19 外部触发同步（仅适用于 TIM15）

TIM 定时器从内部链接在一起，以实现定时器同步或级联。

TIM15 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

当触发输入信号发生变化时，计数器及其预分频器可重新初始化。此外，如果 TIMx_CR1 寄存器中的 URS 位处于低电平，则会生成更新事件 UEV。然后，所有预装载寄存器（TIMx_ARR 和 TIMx_CCRx）都将更新。

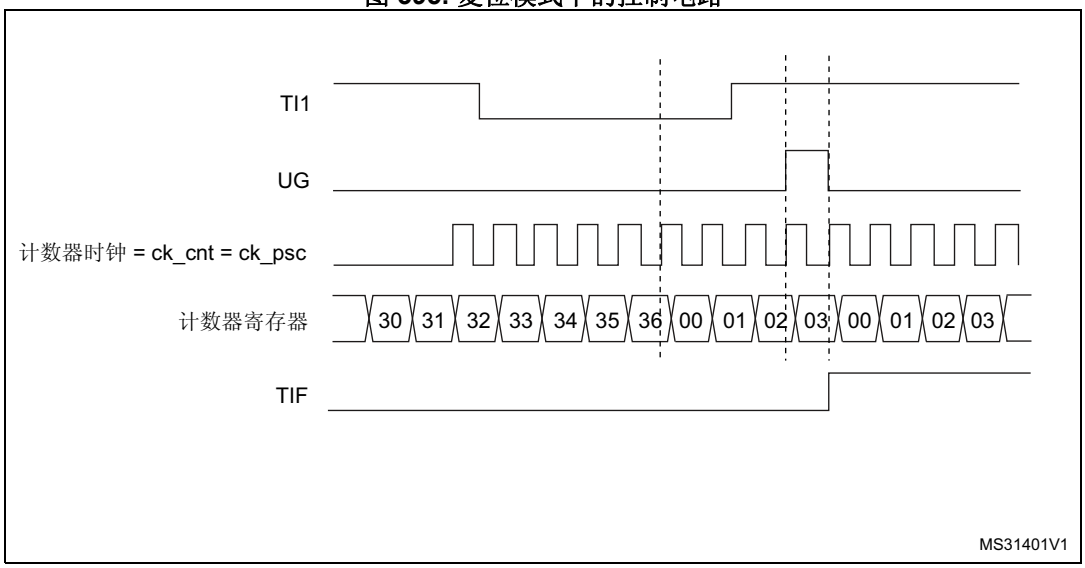
在以下示例中，TI1 输入上出现上升沿时，递增计数器清零：

1. 将通道 1 配置为检测 TI1 的上升沿。配置输入滤波带宽（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 TIMx_CCMR1 寄存器中的 CC1S = 01。在 TIMx_CCER 寄存器中写入 CC1P =“0”和 CC1NP =“0”，使极性有效（仅检测上升沿）。
2. 在 TIMx_SMCR 寄存器中写入 SMS=100，将定时器配置为复位模式。在 TIMx_SMCR 寄存器中写入 TS=101，选择 TI1 作为输入源。
3. 在 TIMx_CR1 寄存器中写入 CEN=1，启动计数器。

计数器开始根据内部时钟计数，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（TIMx_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 TIMx_DIER 寄存器中的 TIE 和 TDE 位）。

下图显示了自动重载寄存器 TIMx_ARR=0x36 时的相关行为。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 395. 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。

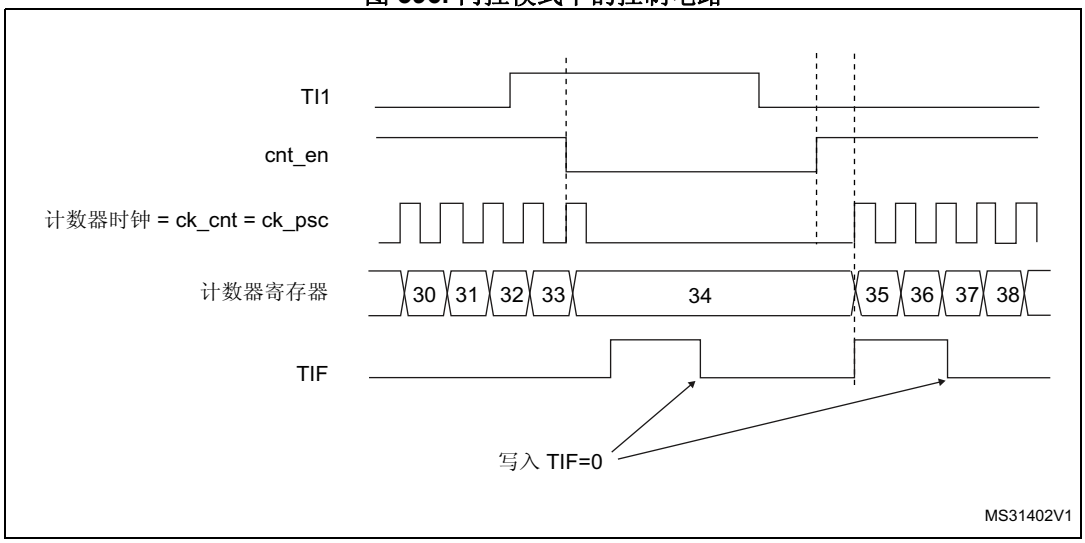
在以下示例中，递增计数器仅在 TI1 输入为低电平时计数：

1. 将通道 1 配置为检测 TI1 上的低电平。配置输入滤波带宽（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 TIMx_CCMR1 寄存器中的 CC1S=01。在 TIMx_CCER 寄存器中写入 CC1P = 1 和 CC1NP = “0”，以确定极性（仅检测低电平）。
2. 在 TIMx_SMCR 寄存器中写入 SMS=101，将定时器配置为门控模式。在 TIMx_SMCR 寄存器中写入 TS=101，选择 TI1 作为输入源。
3. 在 TIMx_CR1 寄存器中写入 CEN=1，使能计数器（在门控模式下，如果 CEN=0，则无论触发输入电平如何，计数器都不启动）。

只要 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，TIMx_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 396. 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。

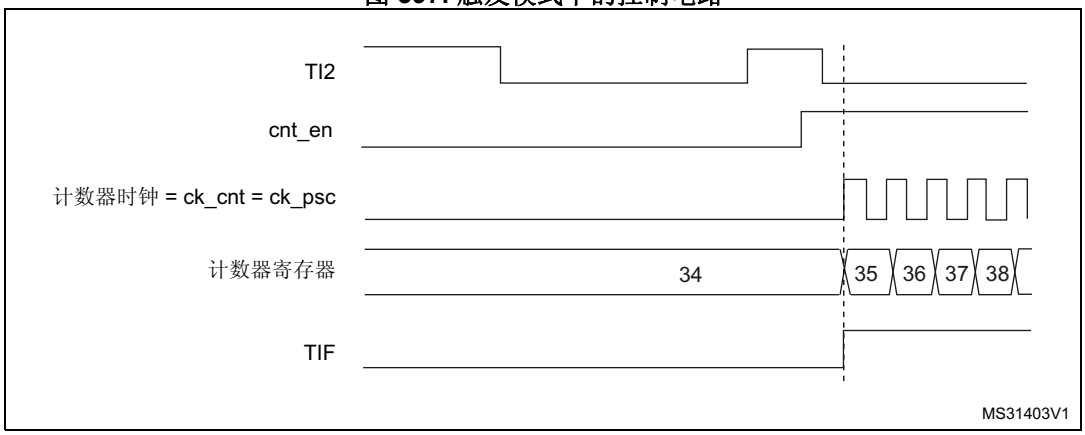
在以下示例中，TI2 输入上出现上升沿时，递增计数器启动：

1. 将通道 2 配置为检测 TI2 上的上升沿。配置输入滤波带宽（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 TIMx_CCMR1 寄存器中的 CC2S=01。在 TIMx_CCER 寄存器中写入 CC2P =“1”和 CC2NP =“0”，以确定极性（仅检测低电平）。
2. 在 TIMx_SMCR 寄存器中写入 SMS = 110，将定时器配置为触发模式。通过在 TIMx_SMCR 寄存器中写入 TS=110 来选择 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 397. 触发模式下的控制电路



36.4.20 从模式——组合复位 + 触发模式

在这种情况下，在出现所选触发输入 (TRGI) 上升沿时，重新初始化计数器，生成一个寄存器更新事件，并启动计数器。

该模式用于单脉冲模式。

36.4.21 DMA 连续传送模式

TIMx 定时器能够根据一个事件生成多个 DMA 请求。主要目的是能够对几个定时器寄存器多次重新编程而无需软件开销，但也可用于定期读取一行中的多个寄存器。

DMA 控制器目标唯一，必须指向虚拟寄存器 TIMx_DMAR。发生给定的定时器事件时，定时器会启动 DMA 请求序列（突发）。每次写入 TIMx_DMAR 寄存器都会重定向到其中一个定时器寄存器。

TIMx_DCR 寄存器中的 DBL[4:0] 位设置 DMA 连续传送长度。当对 TIMx_DMAR 地址进行读或写访问时，定时器进行一次连续传送，即传送次数（按半字或字节）。

TIMx_DCR 寄存器中的 DBA[4:0] 位定义 DMA 传送的 DMA 基址（通过 TIMx_DMAR 地址执行读/写访问时）。DBA 定义为从 TIMx_CR1 寄存器地址开始计算的偏移量。

示例：

00000：TIMx_CR1

00001：TIMx_CR2

00010：TIMx_SMCR

例如，定时器 DMA 连续传送功能用于在发生更新事件后将 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 TIMx 更新 DMA 请求（DIER 寄存器中的 UDE 位置 1）。
4. 使能 TIMx。
5. 使能 DMA 通道。

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。

注： *可以将空值写入保留的寄存器中。*

36.4.22 定时器同步 (TIM15)

TIMx 定时器从内部连接在一起，以实现定时器同步或链接。有关详细信息，请参见*第 35.3.19 节：定时器同步*。

注： *必须先使能接收 TRGO 或 TRGO2 信号的从外设（定时器、ADC 等）的时钟，才能从主定时器接收事件；并且从主定时器接收触发信号时，不得实时更改时钟频率（预分频器）。*

36.4.23 调试模式

当微控制器进入调试模式（Cortex®-M4 内核停止）时，TIMx 计数器会根据 DBG 模块中的 DBG_TIMx_STOP 配置位选择继续正常工作或者停止工作。有关详细信息，请参见*第 51.16.2 节：对定时器、RTC、看门狗、bxCAN 和 I2C 的调试支持*。

为了安全起见，当计数器停止 (DBG_TIMx_STOP = 1) 时，输出被禁止（就像 MOE 位被复位一样）。可以将输出强制变为未激活状态（OSSI 位 = 1），或者通过 GPIO 控制器（OSSI 位 = 0）来控制输出，以将其强制为高阻态。

36.5 TIM15 寄存器

有关寄存器说明中使用的缩写，请参见*第 1.1 节*。

36.5.1 TIM15 控制寄存器 1 (TIM15_CR1)

TIM15 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
Res.	Res.	Res.	Res.	UIFREMAP	Res.	CKD[1:0]		ARPE	Res.	Res.	Res.	OPM	URS	UDIS	CEN
				rw		rw	rw	rw				rw	rw	rw	rw

位 15:12 保留，必须保持复位值。

位 11 **UIFREMAP**：UIF 状态位重映射 (UIF status bit remapping)
0：无重映射。UIF 状态位不复制到 TIMx_CNT 寄存器的位 31。
1：使能重映射。UIF 状态位复制到 TIMx_CNT 寄存器的位 31。

位 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 (TIx) 所使用的死区及采样时钟 (t_{DTS}) 之间的分频比，
00：$t_{DTS} = t_{CK_INT}$
01：$t_{DTS} = 2*t_{CK_INT}$
10：$t_{DTS} = 4*t_{CK_INT}$
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：TIMx_ARR 寄存器不进行缓冲
1：TIMx_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：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：使能后，只有计数器上溢/下溢会生成更新中断。

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 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： 只有事先通过软件将 CEN 位置 1，才可以使用外部时钟和门控模式。而触发模式可通过硬件自动将 CEN 位置 1。

36.5.2 TIM15 控制寄存器 2 (TIM15_CR2)

TIM15 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.	Res.	Res.	Res.	Res.	OIS2	OIS1N	OIS1	TI1S	MMS[2:0]			CCDS	CCUS	Res.	CCPC
					rw	rw	rw	rw	rw	rw	rw	rw	rw		rw

位 15:11 保留，必须保持复位值。

位 10 **OIS2**：输出空闲状态 2（OC2 输出）(Output Idle state 2 (OC2 output))

0：当 MOE = 0 时，OC2 = 0

1：当 MOE = 0 时，OC2 = 1

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

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： 只要编程了 LOCK（TIMx_BKR 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=1 时，（如果 OC1N 有效，则经过死区时间之后）OC1=0

注： 只要编程了 LOCK（TIMx_BKR 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：TIMx_CH1 引脚连接到 TI1 输入

1：TIMx_CH1、CH2 引脚连接到 TI1 输入（异或组合）

位 6:4 **MMS[2:0]**：主模式选择 (Master mode selection)

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

000：**复位**——TIMx_EGR 寄存器中的 UG 位用作触发输出 (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 TIMx_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 标志被置 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)。

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/比较控制位进行预装载 (CCPC=1)，仅通过将 COMG 位置 1 来对这些位进行更新。

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

注： 此位仅对具有互补输出的通道有效。

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： 此位仅对具有互补输出的通道有效。

36.5.3 TIM15 从模式控制寄存器 (TIM15_SMCR)

TIM15 slave mode control 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
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SMS[3]
															rw

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

位 31:17 保留，必须保持复位值。

位 16 **SMS[3]**：从模式选择——位 3 (Slave mode selection - bit 3)
请参见 SMS 说明——位 2:0

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：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)
有关各定时器 ITRx 含义的详细信息，请参见*第 1252 页的表 224：TIMx 内部触发连接*。

注：　这些位只能在未使用的情况下（例如，SMS=000 时）进行更改，以避免转换时出现错误的边沿检测。

位 3 保留，必须保持复位值。

位 16、2、1、0 **SMS[3:0]**：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）。

0000：禁止从模式——如果 CEN = “1”，预分频器时钟直接由内部时钟提供。

0001：保留

0010：保留

0011：保留

0100：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。

0101：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被控制。

0110：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。

0111：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。

1000：组合复位 + 触发模式——在出现所选触发输入 (TRGI) 上升沿时，重新初始化计数器，生成一个寄存器更新事件并启动计数器。

其他代码：保留。

注： 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。

注： 必须先使能接收 TRGO 或 TRGO2 信号的从外设（定时器、ADC 等）的时钟，才能从主定时器接收事件；并且从主定时器接收触发信号时，不得实时更改时钟频率（预分频器）。

表 224. TIMx 内部触发连接

从 TIM	ITR0 (TS = 000)	ITR1 (TS = 001)	ITR2 (TS = 010)	ITR3 (TS = 011)
TIM15	TIM1	TIM3	TIM16 OC1	TIM17 OC1

36.5.4 TIM15 DMA/中断使能寄存器 (TIM15_DIER)

TIM15 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	Res.	Res.	CC2DE	CC1DE	UDE	BIE	TIE	COMIE	Res.	Res.	CC2IE	CC1IE	UIE
	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 **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 12:11 保留，必须保持复位值。

位 10 **CC2DE**：捕获/比较 2 DMA 请求使能 (Capture/Compare 2 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 **BIE**：断路中断使能 (Break interrupt enable)
0：禁止断路中断
1：使能断路中断

位 6 **TIE**：触发中断使能 (Trigger interrupt enable)
0：禁止触发中断
1：使能触发中断

位 5 **COMIE**：COM 中断使能 (COM interrupt enable)
0：禁止 COM 中断
1：使能 COM 中断

位 4:3 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

36.5.5 TIM15 状态寄存器 (TIM15_SR)

TIM15 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
Res.	Res.	Res.	Res.	Res.	CC2OF	CC1OF	Res.	BIF	TIF	COMIF	Res.	Res.	CC2IF	CC1IF	UIF
					rc_w0	rc_w0		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：TIMx_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)
该标志在发生触发事件时由硬件置 1（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，在门控模式情况下选择两个边沿）。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)
发生一个 COM 事件时，会由硬件将该标志置 1（一旦捕获/比较控制位——CCxE、CCxNE、OCxM——已更新）。但需要通过软件清零。
0：未发生 COM 事件。
1：COM 中断挂起。

位 4:3 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：TIMx_CNT 计数器的值与 TIMx_CCR1 寄存器的值匹配。当 TIMx_CCR1 的值大于 TIMx_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 TIMx_CCR1 寄存器将该位清零。
0：未发生输入捕获事件
1：TIMx_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：
– TIMx_CR1 寄存器中的 UDIS = 0，并且重复计数器值上溢时（重复计数器 = 0 时更新）。
– TIMx_CR1 寄存器中的 URS = 0 且 UDIS = 0，并且由软件使用 TIMx_EGR 寄存器中的 UG 位重新初始化 CNT 时。
– TIMx_CR1 寄存器中的 URS = 0 且 UDIS = 0，并且 CNT 由触发事件重新初始化时（请参见 *第 36.5.3 节：TIM15 从模式控制寄存器 (TIM15_SMCR)*）。

36.5.6 TIM15 事件产生寄存器 (TIM15_EGR)

TIM15 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
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BG	TG	COMG	Res.	Res.	CC2G	CC1G	UG
								w	w	rw			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**：触发生成 (Trigger generation)
此位由软件置 1 以生成事件，并由硬件自动清零。
0：不执行任何操作。
1：TIMx_SR 寄存器中的 TIF 标志置 1。使能后可发生相关中断或 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： 此位仅对具有互补输出的通道有效。

位 4:3 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：
TIMx_CCR1 寄存器中将捕获到计数器当前值。使能时，CC1IF 标志置 1 并发送相应的中断或 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）。

36.5.7 TIM15 捕获/比较模式寄存器 1 (TIM15_CCMR1)

TIM15 capture/compare mode register 1

偏移地址：0x18

复位值：0x0000 0000

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。此寄存器的所有其他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。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC2M [3]	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC1M [3]
							Res.								Res.
							rw								rw

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 31:25 保留，始终读为 0

位 24 **OC2M[3]**：输出比较 2 模式——位 3 (Output Compare 2 mode - bit 3)

位 23:17 保留，始终读为 0

位 16 **OC1M[3]**：输出比较 1 模式——位 3 (Output Compare 1 mode - bit 3)
请参见 OC1M 说明——位 6:4

位 15 保留，始终读为 0

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（TIMx_SMCR 寄存器）选择内部触发输入时有效。
注： 仅当通道关闭时（TIMx_CCER 中的 CC2E = “0”），才可向 CC2S 位写入数据。

位 7 保留，始终读为 0

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。

0000：冻结——输出比较寄存器 TIMx_CCR1 与计数器 TIMx_CNT 进行比较不会对输出造成任何影响。

0001：将通道 1 设置为匹配时输出有效电平。当计数器 TIMx_CNT 与捕获/比较寄存器 1 (TIMx_CCR1) 匹配时，OC1REF 信号强制变为高电平。

0010：将通道 1 设置为匹配时输出无效电平。当计数器 TIMx_CNT 与捕获/比较寄存器 1 (TIMx_CCR1) 匹配时，OC1REF 信号强制变为低电平。

0011：翻转——TIMx_CNT=TIMx_CCR1 时，OC1REF 发生翻转。

0100：强制变为无效电平——OC1REF 强制变为低电平。

0101：强制变为有效电平——OC1REF 强制变为高电平。

0110：PWM 模式 1——只要 TIMx_CNT<TIMx_CCR1，通道 1 便为有效状态，否则为无效状态。

0111：PWM 模式 2——只要 TIMx_CNT<TIMx_CCR1，通道 1 便为无效状态，否则为有效状态。

1000：可再触发 OPM 模式 1——在递增计数模式下，通道为有效状态，直至（在 TRGI 信号上）检测到触发事件。然后，在 PWM 模式 1 下进行比较，通道会在下一次更新时再次变为有效状态。在递减计数模式下，通道为无效状态，直至（在 TRGI 信号上）检测到触发事件。然后，在 PWM 模式 1 下进行比较，通道会在下一次更新时再次变为无效状态。

1001：可再触发 OPM 模式 2——在递增计数模式下，通道为无效状态，直至（在 TRGI 信号上）检测到触发事件。然后，在 PWM 模式 2 下进行比较，通道会在下一次更新时再次变为无效状态。在递减计数模式下，通道为有效状态，直至（在 TRGI 信号上）检测到触发事件。然后，在 PWM 模式 1 下进行比较，通道会在下一次更新时再次变为有效状态。

1010：保留

1011：保留

1100：组合 PWM 模式 1——OC1REF 与在 PWM 模式 1 下的行为相同。OC1REFC 是 OC1REF 和 OC2REF 的逻辑或运算结果。

1101：组合 PWM 模式 2——OC1REF 与在 PWM 模式 2 下的行为相同。OC1REFC 是 OC1REF 和 OC2REF 的逻辑与运算结果。

1110：保留

1111：保留

注： ***1：****只要编程了 LOCK（TIMx_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 TIMx_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 TIMx_CCR1 相关的预装载寄存器。可随时向 TIMx_CCR1 写入数据，写入后将立即使用新值。

1：使能与 TIMx_CCR1 相关的预装载寄存器。可读/写访问预装载寄存器。TIMx_CCR1 预装载值在每次生成更新事件时都会装载到活动寄存器中。

注： ***1：****只要编程了 LOCK（TIMx_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（TIMx_CR1 寄存器中的 OPM 位置 1）。其他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 输出的最短延迟时间为 5 个时钟周期。

1：触发输入上出现有效边沿相当于 CC1 输出上的比较匹配。随后，无论比较结果如何，OC 都设置为比较电平。采样触发输入和激活 CC1 输出的延迟时间缩短为 3 个时钟周期。仅当通道配置为 PWM1 或 PWM2 模式时，OCFE 才会起作用。

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（TIMx_SMCR 寄存器）选择内部触发输入时有效。

注：　仅当通道关闭时（TIMx_CCER 中的 CC1E =“0”），才可向 CC1S 位写入数据。

输入捕获模式

位 31:16 保留，始终读为 0

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（TIMx_SMCR 寄存器）选择内部触发输入时有效。

注：　仅当通道关闭时（TIMx_CCER 中的 CC2E =“0”），才可向 CC2S 位写入数据。

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 的数字滤波器的采样长度。数字滤波器由事件计数器组成，每 N 个连续事件才视为一个有效输出边沿：

0000：无滤波器，按 f_{DTS} 频率进行采样
0001：$f_{SAMPLING}=f_{CK_INT}$，N=2
0010：$f_{SAMPLING}=f_{CK_INT}$，N=4
0011：$f_{SAMPLING}=f_{CK_INT}$，N=8
0100：$f_{SAMPLING}=f_{DTS}/2$，N=6
0101：$f_{SAMPLING}=f_{DTS}/2$，N=8
0110：$f_{SAMPLING}=f_{DTS}/4$，N=6
0111：$f_{SAMPLING}=f_{DTS}/4$，N=8
1000：$f_{SAMPLING}=f_{DTS}/8$，N=6
1001：$f_{SAMPLING}=f_{DTS}/8$，N=8
1010：$f_{SAMPLING}=f_{DTS}/16$，N=5
1011：$f_{SAMPLING}=f_{DTS}/16$，N=6
1100：$f_{SAMPLING}=f_{DTS}/16$，N=8
1101：$f_{SAMPLING}=f_{DTS}/32$，N=5
1110：$f_{SAMPLING}=f_{DTS}/32$，N=6
1111：$f_{SAMPLING}=f_{DTS}/32$，N=8

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”（TIMx_CCER 寄存器），预分频器便立即复位。

00：无预分频器，捕获输入上每检测到一个边沿便执行捕获
01：每发生 2 个事件便执行一次捕获
10：每发生 4 个事件便执行一次捕获
11：每发生 8 个事件便执行一次捕获

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（TIMx_SMCR 寄存器）选择内部触发输入时有效

注： 仅当通道关闭时（TIMx_CCER 中的 CC1E = “0”），才可向 CC1S 位写入数据。

36.5.8 TIM15 捕获/比较使能寄存器 (TIM15_CCER)

TIM15 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
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CC2NP	Res.	CC2P	CC2E	CC1NP	CC1NE	CC1P	CC1E
								rw		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 互补输出极性 (Capture/Compare 2 complementary 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：
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 CC1P 说明。

注： *1. 只要编程了 LOCK（TIMx_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。*

2. 此位将在具有互补输出的通道上进行预装载。如果 TIMx_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。

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。

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）。

注： *1. 只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 2 或 3，此位立即变为不可写状态。*

2. 此位将在具有互补输出的通道上进行预装载。如果 TIMx_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。

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 信号输出到相应的输出引脚上，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 (TIMx_CCR1) 中。
0：禁止捕获
1：使能捕获

表 225. 具有断路功能的互补通道 OCx 和 OCxN 的输出控制位 (TIM15)

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	X	0	0	禁止输出（不由定时器驱动：高阻态） OCx=0 OCxN=0、OCxN_EN=0	
		0	0	1	禁止输出（不由定时器驱动：高阻态） OCx=0	OCxREF + 极性 OCxN = OCxREF 异或 CCxNP
		0	1	0	OCxREF + 极性 OCx=OCxREF 异或 CCxP	禁止输出（不由定时器驱动：高阻态） OCxN=0
		X	1	1	OCREF + 极性 + 死区	OCREF 互补项（对 OCREF 进行“非”运算）+ 极性 + 死区
		1	0	1	关闭状态（输出使能为无效状态） OCx=CCxP	OCxREF + 极性 OCxN = OCxREF 异或 CCxNP
		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
0	0	X	X	X	禁止输出（不再由定时器驱动）。输出状态由 GPIO 控制器定义，可以是高电平、低电平或高阻态。	
	1		0	0		
			0	1	关闭状态（输出使能为无效状态） 异步：OCx = CCxP、OCxN = CCxNP 那么如果时钟存在：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	0		
			1	1		

1. 如果一个通道的两个输出均未使用（由 GPIO 控制器接管控制），则 OISx、OISxN、CCxP 和 CCxNP 位必须保持清零状态。

注： *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 AFIO 寄存器。*

36.5.9 TIM15 计数器 (TIM15_CNT)

TIM15 counter

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

位 31 **UIFCPY**：UIF 副本 (UIF Copy)
该位是 TIMx_ISR 寄存器中 UIF 位的只读副本。

位 30:16 保留，必须保持复位值。

位 15:0 **CNT[15:0]**：计数器值 (Counter value)

36.5.10 TIM15 预分频器 (TIM15_PSC)

TIM15 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 f_{CK_PSC} / (PSC[15:0] + 1)。
PSC 包含每次发生更新事件（包括计数器通过 TIMx_EGR 寄存器中的 UG 位清零时，或在配置为“复位模式”时通过触发控制器清零时）时要装载到活动预分频器寄存器的值。

36.5.11 TIM15 自动重载寄存器 (TIM15_ARR)

TIM15 auto-reload register

偏移地址：0x2C

复位值：0xFFFF

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*第 1215 页的第 36.4.1 节：时基单元*。
当自动重载值为空时，计数器不工作。

36.5.12 TIM15 重复计数器寄存器 (TIM15_RCR)

TIM15 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
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	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 TIMx_RCR 寄存器写入何值都无影响。

这意味着在 PWM 模式 (REP+1)下 对应于边沿对齐模式的 PWM 周期数。

36.5.13 TIM15 捕获/比较寄存器 1 (TIM15_CCR1)

TIM15 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 TIMx_CCMR1 寄存器中的 OC1PE 位来使能预装载功能，则该值立刻生效；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 TIMx_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。

36.5.14 TIM15 捕获/比较寄存器 2 (TIM15_CCR2)

TIM15 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 是捕获/比较寄存器 2 的预装载值。

如果没有通过 TIMx_CCMR2 寄存器中的 OC2PE 位来使能预装载功能，则该值立刻生效；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 2）。

实际捕获/比较寄存器中包含要与计数器 TIMx_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。

36.5.15 TIM15 断路和死区寄存器 (TIM15_BDTR)

TIM15 break and dead-time register

偏移地址：0x44

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	BKBID	Res.	BK DSRM	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
			rw		rw										
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MOE	AOE	BKP	BKE	OSSR	OSSI	LOCK[1:0]		DTG[7:0]							
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

注： *由于 Lock 配置决定了 BKBID、BKDSRM、AOE、BKP、OSSI、OSSR 和 DTG[7:0] 的位可能被写锁定，因此必须在第一次对 TIMx_BDTR 寄存器执行写访问时对这些位进行配置。*

位 31:29 保留，必须保持复位值。

位 28 **BKBID**：断路双向 (Break Bidirectional)

0：断路输入 BRK 为输入模式

1：断路输入 BRK 为双向模式

在双向模式下（BKBID 位置 1），断路输入配置为输入模式和开漏输出模式。任何激活的断路事件都将使断路输入上呈逻辑低电平，以向外部器件指示发生了内部断路事件。

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

注： *对该位执行任何写操作后，都需要经过 1 个 APB 时钟周期的延迟才生效。*

位 27 保留，必须保持复位值。

位 26 **BKDSRM**：断路解除 (Break Disarm)

0：启动断路输入 BRK
1：解除断路输入 BRK
当断路源激活后，此位由硬件清零。
必须通过软件将 BKDSRM 位置 1 以释放双向输出控制（开漏输出处于高阻态），然后不断轮询该位，直到其由硬件复位，指示故障条件已消失。

注： 对该位执行任何写操作后，都需要经过 1 个 APB 时钟周期的延迟才生效。

位 25:20 保留，必须保持复位值。

位 19:16 保留，必须保持复位值。

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 输出被禁止或被强制为空闲状态，具体取决于 OSSI 位。
1：如果 OC 和 OCN 输出的相应使能位（TIMx_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 1260 页的第 36.5.8 节：TIM15 捕获/比较使能寄存器 (TIM15_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： 只要编程了 LOCK（TIMx_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： **1**：只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1，此位即无法修改。*
__2__：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 12 **BKE**：断路使能 (Break enable)

0：禁止断路输入（BRK 和 CCS 时钟故障事件）
1：使能断路输入（BRK 和 CCS 时钟故障事件）
编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1 后，此位即无法修改。

注： 对该位执行任何写操作后，都需要经过 1 个 APB 时钟周期的延迟才生效。

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 1260 页的第 36.5.8 节：TIM15 捕获/比较使能寄存器 (TIM15_CCER)*）。

0：处于无效状态时，禁止 OC/OCN 输出（定时器释放输出控制，由强制高阻态的 AFIO 逻辑接管）。

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（输出仍由定时器控制）。

注：　编程了 LOCK（TIMx_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 1260 页的第 36.5.8 节：TIM15 捕获/比较使能寄存器 (TIM15_CCER)*）。

0：处于无效状态时，禁止 OC/OCN 输出（OC/OCN 使能输出信号 = 0）

1：处于无效状态时，一旦 CCxE = 1 或 CCxNE = 1，便将 OC/OCN 输出首先强制为其空闲电平。然后设置 OC/OCN 使能输出信号=1

注：　编程了 LOCK（TIMx_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 TIMx_BDTR 寄存器中的 DTG 位、TIMx_CR2 寄存器中的 OISx 和 OISxN 位以及 TIMx_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（TIMx_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（TIMx_CCMRx 寄存器中的 OCxM 和 OCxPE 位，只要通过 CCxS 位将相关通道配置为输出）执行写操作。

注：　复位后只能对 LOCK 位执行一次写操作。对 TIMx_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-time generator setup)

此位域定义插入到互补输出之间的死区持续时间。DT 与该持续时间相对应。

DTG[7:5]=0xx => DT=DTG[7:0]x t_{dtg}，其中 $t_{dtg}=t_{DTS}$。

DTG[7:5]=10x => DT=(64+DTG[5:0])xt_{dtg}，其中 $T_{dtg}=2xt_{DTS}$。

DTG[7:5]=110 => DT=(32+DTG[4:0])xt_{dtg}，其中 $T_{dtg}=8xt_{DTS}$。

DTG[7:5]=111 => DT=(32+DTG[4:0])xt_{dtg}，其中 $T_{dtg}=16xt_{DTS}$。

示例：如果 T_{DTS}=125 ns (8 MHz)，则可能的死区值为：

0 到 15875 ns（步长为 125 ns），

16 µs 到 31750 ns（步长为 250 ns），

32 µs 到 63 µs（步长为 1 µs），

64 µs 到 126 µs（步长为 2 µs）。

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

36.5.16 TIM15 DMA 控制寄存器 (TIM15_DCR)

TIM15 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
Res.	Res.	Res.	DBL[4:0]					Res.	Res.	Res.	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 位域定义了 DMA 的传送长度（当对 TIMx_DMAR 寄存器进行读或写时，定时器进行一次连续传送）。

00000：1 次传送
00001：2 次传送
00010：3 次传送
...
10001：18 次传送

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 位域定义 DMA 传输的基址（通过 TIMx_DMAR 地址进行读/写访问时）。DBA 定义为从 TIMx_CR1 寄存器地址开始计算的偏移量。

示例：
00000：TIMx_CR1
00001：TIMx_CR2
00010：TIMx_SMCR
...

36.5.17 TIM15 全传输 DMA 地址 (TIM15_DMAR)

TIM15 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

(TIMx_CR1 地址) + (DBA + DMA 索引) x 4

其中 TIMx_CR1 地址为控制寄存器 1 的地址，DBA 为 TIMx_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（TIMx_DCR 寄存器中配置的 DBL）之间。

36.5.18 TIM15 选项寄存器 1 (TIM15_OR1)

TIM15 option register 1

偏移地址：0x50

复位值：0x0000

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.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ENCODER_ MODE[1:0]		TI1_ RMP
													rw	rw	rw

位 31:3 保留，必须保持复位值。

位 2:1 **ENCODER_MODE[1:0]**：编码器模式 (Encoder mode)

00：无重定向

01：TIM2 IC1 和 TIM2 IC2 分别连接到 TIM15 IC1 和 TIM15 IC2

10：TIM3 IC1 和 TIM3 IC2 分别连接到 TIM15 IC1 和 TIM15 IC2

11：TIM4 IC1 和 TIM4 IC2 分别连接到 TIM15 IC1 和 TIM15 IC2

位 0 **TI1_RMP**：输入捕获 1 重映射 (Input capture 1 remap)

0：TIM15 输入捕获 1 连接到 I/O

1：TIM15 输入捕获 1 连接到 LSE

36.5.19 TIM15 选项寄存器 2 (TIM15_OR2)

TIM15 option register 2

偏移地址：0x60

复位值：0x0000 0001

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
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.	Res.	Res.	BKCM P2P	BKCM P1P	BKINP	BKDF1 BK0E	Res.	Res.	Res.	Res.	Res.	BKCM P2E	BKCM P1E	BKINE
				rw	rw	rw	rw						rw	rw	rw

位 31:12 保留，必须保持复位值。

位 11 **BKCMP2P**：BRK COMP2 输入极性 (BRK COMP2 input polarity)

此位选择 COMP2 输入灵敏度，必须与 BKP 极性位一起编程。

0：COMP2 输入为低电平有效

1：COMP2 输入为高电平有效

注：　只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1，此位即无法修改。

位 10 **BKCMP1P**：BRK COMP1 输入极性 (BRK COMP1 input polarity)

此位选择 COMP1 输入灵敏度，必须与 BKP 极性位一起编程。

0：COMP1 输入为低电平有效

1：COMP1 输入为高电平有效

注：　只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1，此位即无法修改。

位 9 **BKINP**：BRK BKIN 输入极性 (BRK BKIN input polarity)
此位选择 BKIN 复用功能输入灵敏度，必须与 BKP 极性位一起编程。
0：BKIN 输入为低电平有效
1：BKIN 输入为高电平有效
注： 只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1，此位即无法修改。

位 8 **BKDF1BK0E**：BRK dfsdm1_break[0] 使能 (BRK dfsdm1_break[0] enable)
此位使能定时器 BRK 输入的 dfsdm1_break[0]。dfsdm1_break[0] 输出与其他 BRK 源进行“或”运算。
0：禁止 dfsdm1_break[0] 输入
1：使能 dfsdm1_break[0] 输入
注： 只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1，此位即无法修改。

位 7:3 保留，必须保持复位值。

位 2 **BKCMP2E**：BRK COMP2 使能 (BRK COMP2 enable)
此位使能定时器 BRK 输入的 COMP2。COMP2 输出与其他 BRK 源进行“或”运算。
0：禁止 COMP2 输入
1：使能 COMP2 输入
注： 只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1，此位即无法修改。

位 1 **BKCMP1E**：BRK COMP1 使能 (BRK COMP1 enable)
此位使能定时器 BRK 输入的 COMP1。COMP1 输出与其他 BRK 源进行“或”运算。
0：禁止 COMP1 输入
1：使能 COMP1 输入
注： 只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1，此位即无法修改。

位 0 **BKINE**：BRK BKIN 输入使能 (BRK BKIN input enable)
此位使能定时器 BRK 输入的 BKIN 复用功能。BKIN 输入与其他 BRK 源进行“或”运算。
0：禁止 BKIN 输入
1：使能 BKIN 输入
注： 只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1，此位即无法修改。

36.5.20 TIM15寄存器映射

TIM15 寄存器可映射为 16 位可寻址寄存器，如下表所述：

表 226. TIM15 寄存器映射和复位值

偏移	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	**TIM15_CR1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	UIFREMAP	Res.	CKD [1:0]		ARPE	Res.	Res.	Res.	OPM	URS	UDIS	CEN
	Reset value																					0		0	0	0				0	0	0	0
0x04	**TIM15_CR2**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OIS2	OIS1N	OIS1	TI1S	MMS[2:0]			CCDS	CCUS	Res.	CCPC
	Reset value																						0	0	0	0	0	0	0	0	0		0
0x08	**TIM15_SMCR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SMS[3]	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MSM	TS[2:0]			Res.	SMS[2:0]		
	Reset value																0									0	0	0	0		0	0	0
0x0C	**TIM15_DIER**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TDE	COMDE	Res.	Res.	CC2DE	CC1DE	UDE	BIE	TIE	COMIE	Res.	Res.	CC2IE	CC1IE	UIE
	Reset value																		0	0			0	0	0	0	0	0			0	0	0
0x10	**TIM15_SR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CC2OF	CC1OF	Res.	BIF	TIF	COMIF	Res.	Res.	CC2IF	CC1IF	UIF
	Reset value																						0	0		0	0	0			0	0	0
0x14	**TIM15_EGR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BG	TG	COMG	Res.	Res.	CC2G	CC1G	UG
	Reset value																									0	0	0			0	0	0
0x18	**TIM15_CCMR1** Output Compare mode	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC2M[3]	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC1M[3]	Res.	OC2M [2:0]			OC2PE	OC2FE	CC2S [1:0]		Res.	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
	TIM15_CCMR1 Input Capture mode	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	IC2F[3:0]				IC2 PSC [1:0]		CC2S [1:0]		IC1F[3:0]				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
0x20	**TIM15_CCER**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CC2NP	Res.	CC2P	CC2E	CC1NP	CC1NE	CC1P	CC1E
	Reset value																									0		0	0	0	0	0	0

表 226. TIM15 寄存器映射和复位值（续）

偏移	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
0x24	**TIM15_CNT**	UIFCPY or Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CNT[15:0]															
	Reset value	0																0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x28	**TIM15_PSC**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PSC[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x2C	**TIM15_ARR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ARR[15:0]															
	Reset value																	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0x30	**TIM15_RCR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REP[7:0]							
	Reset value																									0	0	0	0	0	0	0	0
0x34	**TIM15_CCR1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CCR1[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x38	**TIM15_CCR2**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CCR2[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x44	**TIM15_BDTR**	Res.	Res.	Res.	BKBID	Res.	BKDSRM	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	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	0
0x48	**TIM15_DCR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DBL[4:0]					Res.	Res.	Res.	DBA[4:0]				
	Reset value																				0	0	0	0	0				0	0	0	0	0
0x4C	**TIM15_DMAR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DMAB[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x50	**TIM15_OR1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ENCODER_MODE[1:0]		TI1_RMP
	Reset value																														0	0	0

表 226. TIM15 寄存器映射和复位值（续）

偏移	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
0x60	**TIM15_OR2**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BKCMP2P	BKCMP1P	BKINP	BKDF1BK0E	Res.	Res.	Res.	Res.	Res.	BKCMP2E	BKCMP1E	BKINE
	Reset value																					0	0	0	0						0	0	1

有关寄存器边界地址的信息，请参见*第 78 页的第 2.2.2 节*。

36.6 TIM16/TIM17 寄存器

有关寄存器说明中使用的缩写，请参见*第 1.1 节*。

36.6.1 TIM16/TIM17 控制寄存器 1 (TIMx_CR1)

TIM16/TIM17 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
Res.	Res.	Res.	Res.	UIFREMAP	Res.	CKD[1:0]		ARPE	Res.	Res.	Res.	OPM	URS	UDIS	CEN
				rw		rw	rw	rw				rw	rw	rw	rw

位 15:12 保留，必须保持复位值。

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

0：无重映射。UIF 状态位不复制到 TIMx_CNT 寄存器的位 31。
1：使能重映射。UIF 状态位复制到 TIMx_CNT 寄存器的位 31。

位 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 (TIx) 所使用的死区及采样时钟 (t_{DTS}) 之间的分频比。

00：$t_{DTS}=t_{CK_INT}$
01：$t_{DTS}=2*t_{CK_INT}$
10：$t_{DTS}=4*t_{CK_INT}$
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：TIMx_ARR 寄存器不进行缓冲
1：TIMx_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：使能时，所有以下事件都会产生更新中断或 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： 只有事先通过软件将 CEN 位置 1，才可以使用外部时钟和门控模式。而触发模式可通过硬件自动将 CEN 位置 1。

36.6.2 TIM16/TIM17 控制寄存器 2 (TIMx_CR2)

TIM16/TIM17 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.	Res.	Res.	Res.	Res.	Res.	OIS1N	OIS1	Res.	Res.	Res.	Res.	CCDS	CCUS	Res.	CCPC
						rw	rw					rw	rw		rw

位 15:10 保留，必须保持复位值。

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： 只要编程了 LOCK（TIMx_BKR 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=1 时，（如果 OC1N 有效，则经过死区时间之后）OC1=0

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

位 7:4 保留，必须保持复位值。

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

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

注： 此位仅对具有互补输出的通道有效。

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 位在写入后被预装载，只有当 COM 位置 1 时才进行更新。
注： 此位仅对具有互补输出的通道有效。

36.6.3 TIM16/TIM17 DMA/中断使能寄存器 (TIMx_DIER)

TIM16/TIM17 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.	Res.	COMDE	Res.	Res.	Res.	CC1DE	UDE	BIE	Res.	COMIE	Res.	Res.	Res.	CC1IE	UIE
		rw				rw	rw	rw		rw				rw	rw

位 15:14 保留，必须保持复位值。

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 12:10 保留，必须保持复位值。

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
1：使能断路中断

位 6 保留，必须保持复位值。

位 5 **COMIE**：COM 中断使能 (COM interrupt enable)
0：禁止 COM 中断
1：使能 COM 中断

位 4:2 保留，必须保持复位值。

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

36.6.4 TIM16/TIM17 状态寄存器 (TIMx_SR)

TIM16/TIM17 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
Res.	Res.	Res.	Res.	Res.	Res.	CC1OF	Res.	BIF	Res.	COMIF	Res.	Res.	Res.	CC1IF	UIF
						rc_w0		rc_w0		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：TIMx_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 保留，必须保持复位值。

位 5 **COMIF**：COM 中断标志 (COM interrupt flag)

发生一个 COM 事件时，会由硬件将该标志置 1（一旦捕获/比较控制位——CCxE、CCxNE、OCxM——已更新）。但需要通过软件清零。

0：未发生 COM 事件。

1：COM 中断挂起。

位 4:2 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：TIMx_CNT 计数器的值与 TIMx_CCR1 寄存器的值匹配。当 TIMx_CCR1 的值大于 TIMx_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 TIMx_CCR1 寄存器将该位清零。

0：未发生输入捕获事件

1：TIMx_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：

– TIMx_CR1 寄存器中的 UDIS = 0，并且重复计数器值上溢时（重复计数器 = 0 时更新）。

– TIMx_CR1 寄存器中的 URS = 0 且 UDIS = 0，并且由软件使用 TIMx_EGR 寄存器中的 UG 位重新初始化 CNT 时。

36.6.5 TIM16/TIM17 事件生成寄存器 (TIMx_EGR)

TIM16/TIM17 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
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BG	Res.	COMG	Res.	Res.	Res.	CC1G	UG
								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 保留，必须保持复位值。

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： 此位仅对具有互补输出的通道有效。

位 4:2 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：

TIMx_CCR1 寄存器中将捕获到计数器当前值。使能时，CC1IF 标志置 1 并发送相应的中断或 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）。

36.6.6 TIM16/TIM17 捕获/比较模式寄存器 1 (TIMx_CCMR1)

TIM16/TIM17 capture/compare mode register 1

偏移地址：0x18

复位值：0x0000 0000

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。此寄存器的所有其他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。

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.	OC1M[3]
															Res.
															rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC1M[2:0]			OC1PE	OC1FE	CC1S[1:0]	
								IC1F[3:0]				IC1PSC[1:0]			
								rw	rw	rw	rw	rw	rw	rw	rw

输出比较模式：

位 31:17 保留，始终读为 0

位 16 **OC1M[3]**：输出比较 1 模式（位 3）(Output Compare 1 mode (bit 3))

位 15:7 保留

位 6:4 **OC1M[2:0]**：输出比较 1 模式（位 2 到 0）(Output Compare 1 mode (bits 2 to 0))

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。

0000：冻结——输出比较寄存器 TIMx_CCR1 与计数器 TIMx_CNT 进行比较不会对输出造成任何影响。

0001：将通道 1 设置为匹配时输出有效电平。当计数器 TIMx_CNT 与捕获/比较寄存器 1 (TIMx_CCR1) 匹配时，OC1REF 信号强制变为高电平。

0010：将通道 1 设置为匹配时输出无效电平。当计数器 TIMx_CNT 与捕获/比较寄存器 1 (TIMx_CCR1) 匹配时，OC1REF 信号强制变为低电平。

0011：翻转——TIMx_CNT=TIMx_CCR1 时，OC1REF 发生翻转。

0100：强制变为无效电平——OC1REF 强制变为低电平。

0101：强制变为有效电平——OC1REF 强制变为高电平。

0110：PWM 模式 1——只要 TIMx_CNT<TIMx_CCR1，通道 1 便为有效状态，否则为无效状态。

0111：PWM 模式 2——只要 TIMx_CNT<TIMx_CCR1，通道 1 便为无效状态，否则为有效状态。

所有其他值：保留

注： ***1：****只要编程了 LOCK（TIMx_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。*

位 3 **OC1PE**：输出比较 1 预装载使能 (Output Compare 1 preload enable)

0：禁止与 TIMx_CCR1 相关的预装载寄存器。可随时向 TIMx_CCR1 写入数据，写入后将立即使用新值。

1：使能与 TIMx_CCR1 相关的预装载寄存器。可读/写访问预装载寄存器。TIMx_CCR1 预装载值在每次生成更新事件时都会装载到活动寄存器中。

*注： **1**：只要编程了 LOCK（TIMx_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（TIMx_CR1 寄存器中的 OPM 位置 1）。其他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 输出的最短延迟时间为 5 个时钟周期。

1：触发输入上出现有效边沿相当于 CC1 输出上的比较匹配。随后，无论比较结果如何，OC 都设置为比较电平。采样触发输入和激活 CC1 输出的延迟时间缩短为 3 个时钟周期。仅当通道配置为 PWM1 或 PWM2 模式时，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（TIMx_SMCR 寄存器）选择内部触发输入时有效

注： 仅当通道关闭时（TIMx_CCER 中的 CC1E = “0”），才可向 CC1S 位写入数据。

输入捕获模式

位 31:8 保留，必须保持复位值。

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 的数字滤波器的采样长度。数字滤波器由事件计数器组成，每 N 个连续事件才视为一个有效输出边沿：

0000：无滤波器，按 f_{DTS} 频率进行采样

0001：$f_{SAMPLING}=f_{CK_INT}$，N=2

0010：$f_{SAMPLING}=f_{CK_INT}$，N=4

0011：$f_{SAMPLING}=f_{CK_INT}$，N=8

0100：$f_{SAMPLING}=f_{DTS}/2$，N=6

0101：$f_{SAMPLING}=f_{DTS}/2$，N=8

0110：$f_{SAMPLING}=f_{DTS}/4$，N=6

0111：$f_{SAMPLING}=f_{DTS}/4$，N=8

1000：$f_{SAMPLING}=f_{DTS}/8$，N=6

1001：$f_{SAMPLING}=f_{DTS}/8$，N=8

1010：$f_{SAMPLING}=f_{DTS}/16$，N=5

1011：$f_{SAMPLING}=f_{DTS}/16$，N=6

1100：$f_{SAMPLING}=f_{DTS}/16$，N=8

1101：$f_{SAMPLING}=f_{DTS}/32$，N=5

1110：$f_{SAMPLING}=f_{DTS}/32$，N=6

1111：$f_{SAMPLING}=f_{DTS}/32$，N=8

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”（TIMx_CCER 寄存器），预分频器便立即复位。

00：无预分频器，捕获输入上每检测到一个边沿便执行捕获

01：每发生 2 个事件便执行一次捕获

10：每发生 4 个事件便执行一次捕获

11：每发生 8 个事件便执行一次捕获

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（TIMx_SMCR 寄存器）选择内部触发输入时有效

注： 仅当通道关闭时（TIMx_CCER 中的 CC1E = “0”），才可向 CC1S 位写入数据。

36.6.7 TIM16/TIM17 捕获/比较使能寄存器 (TIMx_CCER)

TIM16/TIM17 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
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CC1NP	CC1NE	CC1P	CC1E
												rw	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：

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 CC1P 说明。

注： ***1.*** *只要编程了 LOCK（TIMx_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。*

2. *此位将在具有互补输出的通道上进行预装载。如果 TIMx_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。*

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。

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）。

注： ***1.*** *只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 2 或 3，此位立即变为不可写状态。*

2. *此位将在具有互补输出的通道上进行预装载。如果 TIMx_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。*

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 信号输出到相应的输出引脚上，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 (TIMx_CCR1) 中。

0：禁止捕获

1：使能捕获

表 227. 具有断路功能的互补通道 OCx 和 OCxN 的输出控制位 (TIM16/17)

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	X	0	0	禁止输出（不由定时器驱动：高阻态） OCx=0 OCxN=0、OCxN_EN=0	
		0	0	1	禁止输出（不由定时器驱动：高阻态） OCx=0	OCxREF + 极性 OCxN = OCxREF 异或 CCxNP
		0	1	0	OCxREF + 极性 OCx=OCxREF 异或 CCxP	禁止输出（不由定时器驱动：高阻态） OCxN=0
		X	1	1	OCREF + 极性 + 死区	OCREF 互补项（对 OCREF 进行“非”运算）+ 极性 + 死区
		1	0	1	关闭状态（输出使能为无效状态） OCx=CCxP	OCxREF + 极性 OCxN = OCxREF 异或 CCxNP
		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
0	0	X	X	X	禁止输出（不再由定时器驱动）。输出状态由 GPIO 控制器定义，可以是高电平、低电平或高阻态。	
	1		0	0		
			0	1	关闭状态（输出使能为无效状态） 异步：OCx = CCxP、OCxN = CCxNP 那么如果时钟存在：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	0		
			1	1		

1. 如果一个通道的两个输出均未使用（由 GPIO 控制器接管控制），则 OISx、OISxN、CCxP 和 CCxNP 位必须保持清零状态。

注： *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 AFIO 寄存器。*

36.6.8 TIM16/TIM17 计数器 (TIMx_CNT)

TIM16/TIM17 counter

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
UIF CPY	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
r															

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

位 31 **UIFCPY**：UIF 副本 (UIF Copy)

该位是 TIMx_ISR 寄存器中 UIF 位的只读副本。如果 TIMx_CR1 中的 UIFREMAP 位复位，则位 31 保留，读为 0。

位 30:16 保留，必须保持复位值。

位 15:0 **CNT[15:0]**：计数器值 (Counter value)

36.6.9 TIM16/TIM17 预分频器 (TIMx_PSC)

TIM16/TIM17 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 f_{CK_PSC} / (PSC[15:0] + 1)。

PSC 包含每次发生更新事件（包括计数器通过 TIMx_EGR 寄存器中的 UG 位清零时，或在配置为“复位模式”时通过触发控制器清零时）时要装载到活动预分频器寄存器的值。

36.6.10 TIM16/TIM17 自动重载寄存器 (TIMx_ARR)

TIM16/TIM17 auto-reload register

偏移地址：0x2C

复位值：0xFFFF

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*第 1215 页的第 36.4.1 节：时基单元*。

当自动重载值为空时，计数器不工作。

36.6.11 TIM16/TIM17 重复计数器寄存器 (TIMx_RCR)

TIM16/TIM17 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
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	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 TIMx_RCR 寄存器写入何值都无影响。

这意味着在 PWM 模式 (REP+1)下 对应于边沿对齐模式的 PWM 周期数。

36.6.12 TIM16/TIM17 捕获/比较寄存器 1 (TIMx_CCR1)

TIM16/TIM17 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 TIMx_CCMR1 寄存器中的 OC1PE 位来使能预装载功能，则该值立刻生效；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 TIMx_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。

36.6.13 TIM16/TIM17 断路和死区寄存器 (TIMx_BDTR)

TIM16/TIM17 break and dead-time register

偏移地址：0x44

复位值：0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MOE	AOE	BKP	BKE	OSSR	OSSI	LOCK[1:0]		DTG[7:0]							
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

注： *由于 LOCK 配置决定了 BKBID、BKDSRM、AOE、BKP、OSSI、OSSR 和 DTG[7:0] 的位可能被写锁定，因此必须在第一次对 TIMx_BDTR 寄存器执行写访问时对这些位进行配置。*

位 31:29 保留，必须保持复位值。

位 28 **BKBID**：断路双向 (Break Bidirectional)

0：断路输入 BRK 为输入模式
1：断路输入 BRK 为双向模式
在双向模式下（BKBID 位置 1），断路输入配置为输入模式和开漏输出模式。任何激活的断路事件都将使断路输入上呈逻辑低电平，以向外部器件指示发生了内部断路事件。

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

注： *对该位执行任何写操作后，都需要经过 1 个 APB 时钟周期的延迟才生效。*

位 27 保留，必须保持复位值。

位 26 **BKDSRM**：断路解除 (Break Disarm)

0：启动断路输入 BRK
1：解除断路输入 BRK
当断路源激活后，此位由硬件清零。
必须通过软件将 BKDSRM 位置 1 以释放双向输出控制（开漏输出处于高阻态），然后不断轮询该位，直到其由硬件复位，指示故障条件已消失。

注： *对该位执行任何写操作后，都需要经过 1 个 APB 时钟周期的延迟才生效。*

位 25:20 保留，必须保持复位值。

位 19:16 保留，必须保持复位值。

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 输出被禁止或被强制为空闲状态，具体取决于 OSSI 位。
1：如果 OC 和 OCN 输出的相应使能位（TIMx_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 1281 页的第 36.6.7 节：TIM16/TIM17 捕获/比较使能寄存器 (TIMx_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：　只要编程了 LOCK（TIMx_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：　**1.** 只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1，此位即无法修改。*

*　　**2.** 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 12 **BKE**：断路使能 (Break enable)

0：禁止断路输入（BRK 和 CCS 时钟故障事件）

1：使能断路输入（BRK 和 CCS 时钟故障事件）

*注：　**1.** 编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1 后，此位即无法修改。*

*　　**2.** 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 1281 页的第 36.6.7 节：TIM16/TIM17 捕获/比较使能寄存器 (TIMx_CCER)*）。

0：处于无效状态时，禁止 OC/OCN 输出（定时器释放输出控制，由强制高阻态的 AFIO 逻辑接管）。

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（输出仍由定时器控制）。

注：　编程了 LOCK（TIMx_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 1281 页的第 36.6.7 节：TIM16/TIM17 捕获/比较使能寄存器 (TIMx_CCER)*）。

0：处于无效状态时，禁止 OC/OCN 输出（OC/OCN 使能输出信号 = 0）

1：处于无效状态时，一旦 CCxE = 1 或 CCxNE = 1，便将 OC/OCN 输出首先强制为其空闲电平。然后设置 OC/OCN 使能输出信号=1

注：　编程了 LOCK（TIMx_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 TIMx_BDTR 寄存器中的 DTG 位、TIMx_CR2 寄存器中的 OISx 和 OISxN 位以及 TIMx_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（TIMx_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（TIMx_CCMRx 寄存器中的 OCxM 和 OCxPE 位，只要通过 CCxS 位将相关通道配置为输出）执行写操作。

注：　复位后只能对 LOCK 位执行一次写操作。对 TIMx_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-time generator setup)

此位域定义插入到互补输出之间的死区持续时间。DT 与该持续时间相对应。
DTG[7:5]=0xx => DT=DTG[7:0]x t_{dtg}，其中 $t_{dtg}=t_{DTS}$。
DTG[7:5]=10x => DT=(64+DTG[5:0])xt_{dtg}，其中 $T_{dtg}=2xt_{DTS}$。
DTG[7:5]=110 => DT=(32+DTG[4:0])xt_{dtg}，其中 $T_{dtg}=8xt_{DTS}$。
DTG[7:5]=111 => DT=(32+DTG[4:0])xt_{dtg}，其中 $T_{dtg}=16xt_{DTS}$。
示例：如果 T_{DTS}=125 ns (8 MHz)，则可能的死区值为：
0 到 15875 ns（步长为 125 ns）
16 µs 到 31750 ns（步长为 250 ns）
32 µs 到 63 µs（步长为 1 µs）
64 µs 到 126 µs（步长为 2 µs）

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

36.6.14 TIM16/TIM17 DMA 控制寄存器 (TIMx_DCR)

TIM16/TIM17 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
Res.	Res.	Res.	DBL[4:0]					Res.	Res.	Res.	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 位向量定义了 DMA 的传送长度（当对 TIMx_DMAR 地址进行读或写访问时，定时器进行一次连续传送），即一个 DMA 要连续传送的次数。可按半字或字节进行传送（请参见下面的示例）。
00000：1 次传送
00001：2 次传送
00010：3 次传送
...
10001：18 次传送

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 位域定义 DMA 传输的基址（通过 TIMx_DMAR 地址进行读/写访问时）。DBA 定义为从 TIMx_CR1 寄存器地址开始计算的偏移量。
示例：
00000：TIMx_CR1
00001：TIMx_CR2
00010：TIMx_SMCR
...

示例：以下面的传送为例：DBL = 7 次传送且 DBA = TIMx_CR1。这种情况下将向/从自 TIMx_CR1 地址开始的 7 个寄存器传输数据。

36.6.15 TIM16/TIM17 全传输 DMA 地址 (TIMx_DMAR)

TIM16/TIM17 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

(TIMx_CR1 地址) + (DBA + DMA 索引) x 4

其中 TIMx_CR1 地址为控制寄存器 1 的地址，DBA 为 TIMx_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（TIMx_DCR 寄存器中配置的 DBL）之间。

36.6.16 TIM16 选项寄存器 1 (TIM16_OR1)

TIM16 option register 1

偏移地址：0x50

复位值：0x0000

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.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TI1_RMP[1:0]	
														rw	rw

位 31:2 保留，必须保持复位值。

位 1:0 **TI1_RMP[1:0]**：输入捕获 1 重映射 (Input capture 1 remap)

00：TIM16 输入捕获 1 连接到 I/O

01：TIM16 输入捕获 1 连接到 LSI

10：TIM16 输入捕获 1 连接到 LSE

11：TIM16 输入捕获 1 连接到 RTC 唤醒中断

36.6.17 TIM16 选项寄存器 2 (TIM16_OR2)

TIM16 option register 2

偏移地址：0x60

复位值：0x0000 0001

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
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.	Res.	Res.	BKCMP2P	BKCMP1P	BKINP	BKDF1BK1E	Res.	Res.	Res.	Res.	Res.	BKCMP2E	BKCMP1E	BKINE
				rw	rw	rw	rw						rw	rw	rw

位 31:12 保留，必须保持复位值。

位 11 **BKCMP2P**：BRK COMP2 输入极性 (BRK COMP2 input polarity)
此位选择 COMP2 输入灵敏度，必须与 BKP 极性位一起编程。
0：COMP2 输入为低电平有效
1：COMP2 输入为高电平有效
注： 只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1，此位即无法修改。

位 10 **BKCMP1P**：BRK COMP1 输入极性 (BRK COMP1 input polarity)
此位选择 COMP1 输入灵敏度，必须与 BKP 极性位一起编程。
0：COMP1 输入为低电平有效
1：COMP1 输入为高电平有效
注： 只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1，此位即无法修改。

位 9 **BKINP**：BRK BKIN 输入极性 (BRK BKIN input polarity)
此位选择 BKIN 复用功能输入灵敏度，必须与 BKP 极性位一起编程。
0：BKIN 输入为低电平有效
1：BKIN 输入为高电平有效
注： 只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1，此位即无法修改。

位 8 **BKDF1BK1E**：BRK dfsdm1_break[1] 使能 (BRK dfsdm1_break[2] enable)
此位使能定时器 BRK 输入的 dfsdm1_break[1]。dfsdm1_break[1] 输出与其他 BRK 源进行“或”运算。
0：禁止 dfsdm1_break[1] 输入
1：使能 dfsdm1_break[1] 输入
注： 只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1，此位即无法修改。

位 7:3 保留，必须保持复位值。

位 2 **BKCMP2E**：BRK COMP2 使能 (BRK COMP2 enable)
此位使能定时器 BRK 输入的 COMP2。COMP2 输出与其他 BRK 源进行“或”运算。
0：禁止 COMP2 输入
1：使能 COMP2 输入
注： 只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1，此位即无法修改。

位 1 **BKCMP1E**：BRK COMP1 使能 (BRK COMP1 enable)
此位使能定时器 BRK 输入的 COMP1。COMP1 输出与其他 BRK 源进行“或”运算。
0：禁止 COMP1 输入
1：使能 COMP1 输入
注： 只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1，此位即无法修改。

位 0 **BKINE**：BRK BKIN 输入使能 (BRK BKIN input enable)
此位使能定时器 BRK 输入的 BKIN 复用功能。BKIN 输入与其他 BRK 源进行“或”运算。
0：禁止 BKIN 输入
1：使能 BKIN 输入
注： 只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1，此位即无法修改。

36.6.18 TIM17 选项寄存器 1 (TIM17_OR1)

TIM17 option register 1

偏移地址：0x50

复位值：0x0000

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.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TI1_RMP[1:0]	
														rw	rw

位 31:2 保留，必须保持复位值。

位 1:0 **TI1_RMP[1:0]**：输入捕获 1 重映射 (Input capture 1 remap)
00：TIM17 输入捕获 1 连接到 I/O
01：TIM17 输入捕获 1 连接到 MSI
10：TIM17 输入捕获 1 连接到 HSE/32
11：TIM17 输入捕获 1 连接到 MCO

36.6.19 TIM17 选项寄存器 2 (TIM17_OR2)

TIM17 option register 2

偏移地址：0x60

复位值：0x0000 0001

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
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.	Res.	Res.	BKCMP2P	BKCMP1P	BKINP	BKDF1BK2E	Res.	Res.	Res.	Res.	Res.	BKCMP2E	BKCMP1E	BKINE
				rw	rw	rw	rw						rw	rw	rw

位 31:12 保留，必须保持复位值。

位 11 **BKCMP2P**：BRK COMP2 输入极性 (BRK COMP2 input polarity)
此位选择 COMP2 输入灵敏度，必须与 BKP 极性位一起编程。
0：COMP2 输入为低电平有效
1：COMP2 输入为高电平有效
注： 只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1，此位即无法修改。

位 10 **BKCMP1P**：BRK COMP1 输入极性 (BRK COMP1 input polarity)
此位选择 COMP1 输入灵敏度，必须与 BKP 极性位一起编程。
0：COMP1 输入为低电平有效
1：COMP1 输入为高电平有效
注： 只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1，此位即无法修改。

位 9 **BKINP**：BRK BKIN 输入极性 (BRK BKIN input polarity)
此位选择 BKIN 复用功能输入灵敏度，必须与 BKP 极性位一起编程。
0：BKIN 输入为低电平有效
1：BKIN 输入为高电平有效
注： 只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1，此位即无法修改。

位 8 **BKDF1BK2E**：BRK dfsdm1_break[2] 使能 (BRK dfsdm1_break[2] enable)
此位使能定时器 BRK 输入的 dfsdm1_break[2]。dfsdm1_break[2] 输出与其他 BRK 源进行“或”运算。
0：禁止 dfsdm1_break[2] 输入
1：使能 dfsdm1_break[2] 输入
注： 只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1，此位即无法修改。

位 7:3 保留，必须保持复位值。

位 2 **BKCMP2E**：BRK COMP2 使能 (BRK COMP2 enable)
此位使能定时器 BRK 输入的 COMP2。COMP2 输出与其他 BRK 源进行“或”运算。
0：禁止 COMP2 输入
1：使能 COMP2 输入
注： 只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1，此位即无法修改。

位 1 **BKCMP1E**：BRK COMP1 使能 (BRK COMP1 enable)
此位使能定时器 BRK 输入的 COMP1。COMP1 输出与其他 BRK 源进行“或”运算。
0：禁止 COMP1 输入
1：使能 COMP1 输入
注： 只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1，此位即无法修改。

位 0 **BKINE**：BRK BKIN 输入使能 (BRK BKIN input enable)
此位使能定时器 BRK 输入的 BKIN 复用功能。BKIN 输入与其他 BRK 源进行“或”运算。
0：禁止 BKIN 输入
1：使能 BKIN 输入
注： 只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1，此位即无法修改。

36.6.20 TIM16/TIM17 寄存器映射

寄存器TIM16/TIM17可映射为 16 位可寻址寄存器，如下表所述：

表 228. TIM16/TIM17 寄存器映射和复位值

偏移	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	**TIMx_CR1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	UIFREMAP	Res.	CKD [1:0]		ARPE	Res.	Res.	Res.	OPM	URS	UDIS	CEN
	Reset value																					0		0	0	0				0	0	0	0
0x04	**TIMx_CR2**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OIS1N	OIS1	Res.	Res.	Res.	Res.	CCDS	CCUS	Res.	CCPC
	Reset value																							0	0					0	0		0
0x0C	**TIMx_DIER**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	COMDE	Res.	Res.	Res.	CC1DE	UDE	BIE	Res.	COMIE	Res.	Res.	Res.	CC1IE	UIE
	Reset value																			0				0	0	0		0				0	0
0x10	**TIMx_SR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CC1OF	Res.	BIF	Res.	COMIF	Res.	Res.	Res.	CC1IF	UIF
	Reset value																							0		0		0				0	0
0x14	**TIMx_EGR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BG	Res.	COMG	Res.	Res.	Res.	CC1G	UG
	Reset value																									0		0				0	0
0x18	**TIMx_CCMR1** Output Compare mode	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC1M[3]	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC1M [2:0]			OC1PE	OC1FE	CC1S [1:0]	
	Reset value																0										0	0	0	0	0	0	0
	TIMx_CCMR1 Input Capture mode	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	IC1F[3:0]				IC1PSC [1:0]		CC1S [1:0]	
	Reset value																									0	0	0	0	0	0	0	0
0x20	**TIMx_CCER**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CC1NP	CC1NE	CC1P	CC1E
	Reset value																													0	0	0	0
0x24	**TIMx_CNT**	UIFCPY or Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	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**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PSC[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x2C	**TIMx_ARR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ARR[15:0]															
	Reset value																	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

表 228. TIM16/TIM17 寄存器映射和复位值（续）

偏移	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
0x30	**TIMx_RCR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REP[7:0]							
	Reset value																									0	0	0	0	0	0	0	0
0x34	**TIMx_CCR1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CCR1[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x44	**TIMx_BDTR**	Res.	Res.	Res.	BKBID	Res.	BKDSRM	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	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	0
0x48	**TIMx_DCR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DBL[4:0]					Res.	Res.	Res.	DBA[4:0]				
	Reset value																				0	0	0	0	0				0	0	0	0	0
0x4C	**TIMx_DMAR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DMAB[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x50	**TIM16_OR1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TI1_RMP [1:0]	
	Reset value																															0	0
0x60	**TIM16_OR2**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BKCMP2P	BKCMP1P	BKINP	BKDF1BK1E	Res.	Res.	Res.	Res.	Res.	BKCMP2E	BKCMP1E	BKINE
	Reset value																					0	0	0	0						0	0	1
0x50	**TIM17_OR1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TI1_RMP [1:0]	
	Reset value																															0	0
0x60	**TIM17_OR2**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BKCMP2P	BKCMP1P	BKINP	BKDF1BK2E	Res.	Res.	Res.	Res.	Res.	BKCMP2E	BKCMP1E	BKINE
	Reset value																					0	0	0	0						0	0	1

有关寄存器边界地址的信息，请参见*第 78 页的第 2.2.2 节*。

37 基本定时器 (TIM6/TIM7)

37.1 TIM6/TIM7 简介

基本定时器 TIM6 和 TIM7 包含一个 16 位自动重载计数器，该计数器由可编程预分频器驱动。

此类定时器不仅可用作通用定时器以生成时基，还可以专门用于驱动数模转换器 (DAC)。实际上，此类定时器内部连接到 DAC 并能够通过其触发输出驱动 DAC。

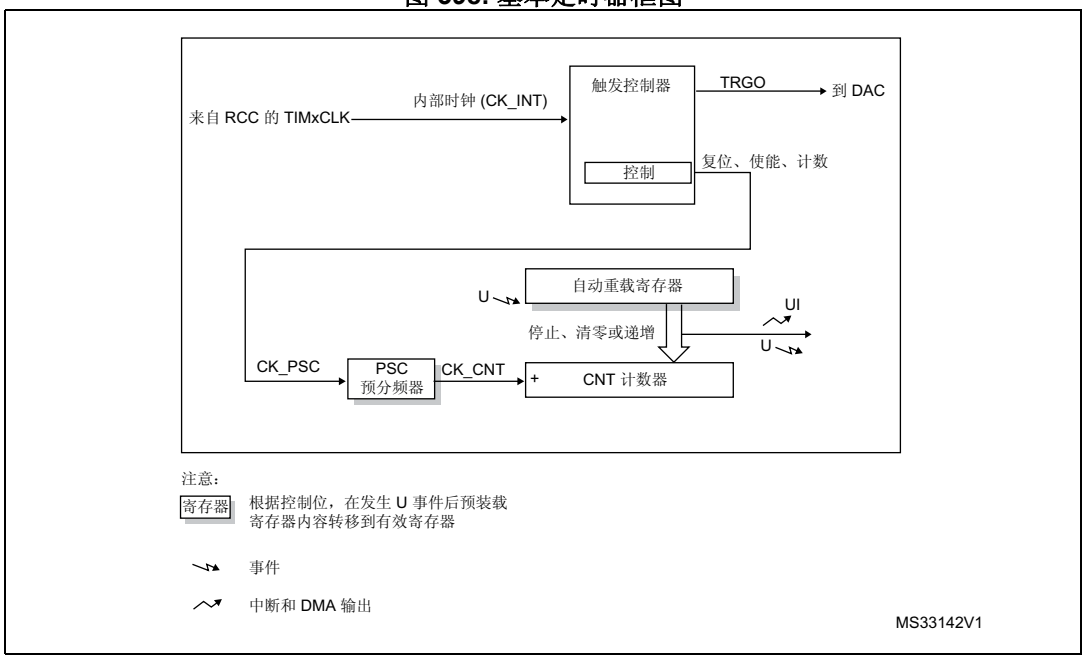
这些定时器彼此完全独立，不共享任何资源。

37.2 TIM6/TIM7 主要特性

基本定时器 (TIM6/TIM7) 的特性包括：

- 16 位自动重载递增计数器
- 16 位可编程预分频器，用于对计数器时钟频率进行分频（可在运行时修改），分频系数介于 1 和 65535 之间
- 用于触发 DAC 的同步电路
- 发生如下更新事件时会生成中断/DMA 请求：计数器上溢

图 398. 基本定时器框图



37.3 TIM6/TIM7 功能说明

37.3.1 时基单元

可编程定时器的主要模块由一个 16 位递增计数器及其相关的自动重载寄存器组成。计数器的时钟可通过预分频器进行分频。

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

时基单元包括：

- 计数器寄存器 (TIMx_CNT)
- 预分频器寄存器 (TIMx_PSC)
- 自动重载寄存器 (TIMx_ARR)

自动重载寄存器是预装载的。每次尝试对自动重载寄存器执行读写操作时，都会访问预装载寄存器。预装载寄存器的内容既可以立即传送到影子寄存器，也可以在每次发生更新事件 UEV 时传送到影子寄存器，这取决于 TIMx_CR1 寄存器中的自动重载预装载使能位 (ARPE)。当计数器达到上溢值并且 TIMx_CR1 寄存器中的 UDIS 位为 0 时，将发送更新事件。该更新事件也可由软件产生。下文将针对各配置的更新事件的产生进行详细介绍。

计数器由预分频器输出 CK_CNT 提供时钟，仅当 TIMx_CR1 寄存器中的计数器启动位 (CEN) 置 1 时，才会启动计数器。

请注意，实际的计数器使能信号 CNT_EN 在 CEN 置 1 的一个时钟周期后被置 1。

预分频器说明

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 16 位寄存器 TIMx_PSC 所控制的 16 位计数器。由于 TIMx_PSC 控制寄存器有缓冲，因此可对预分频器进行实时更改。而新的预分频比将在下一更新事件发生时被采用。

图 399 和 *图 400* 以一些示例说明在预分频比实时变化时计数器的行为。

图 399. 预分频器分频由 1 变为 2 时的计数器时序图

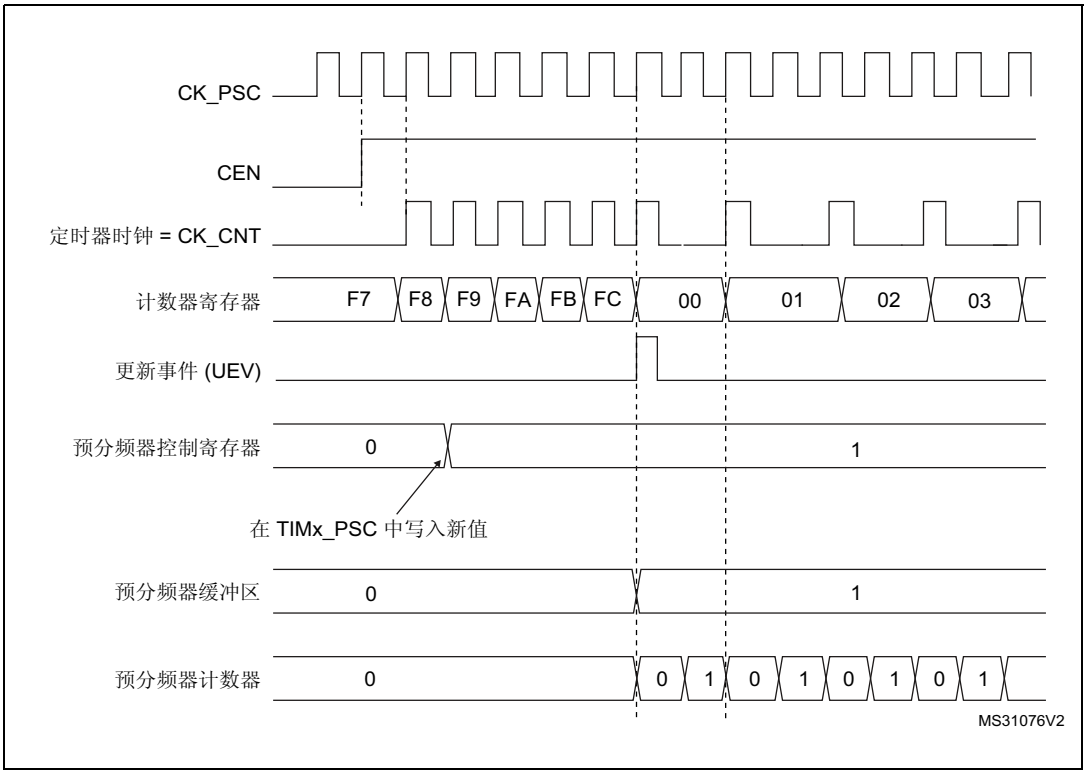
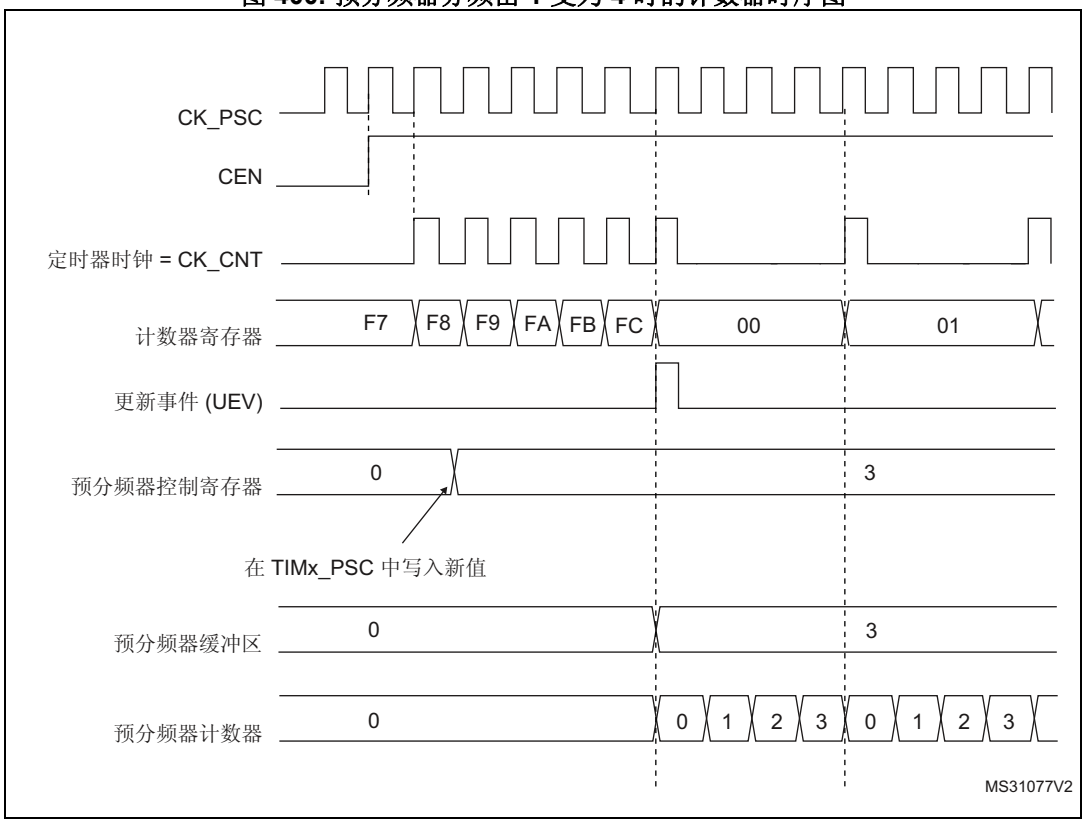


图 400. 预分频器分频由 1 变为 4 时的计数器时序图



37.3.2 计数模式

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

每次发生计数器上溢时会生成更新事件，或将 TIMx_EGR 寄存器中的 UG 位置 1（通过软件或使用从模式控制器）也可以产生更新事件。

通过软件将 TIMx_CR1 寄存器中的 UDIS 位置 1 可禁止 UEV 事件。这可避免向预装载寄存器写入新值时更新影子寄存器。这样，直到 UDIS 位中写入 0 前便不会生成任何更新事件，但计数器和预分频器计数器都会重新从 0 开始计数（而预分频比保持不变）。此外，如果 TIMx_CR1 寄存器中的 URS 位（更新请求选择）已置 1，则将 UG 位置 1 会生成更新事件 UEV，但不会将 UIF 标志置 1（因此，不会发送任何中断或 DMA 请求）。

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

- 预分频器的缓冲区中将重新装载预装载值（TIMx_PSC 寄存器的内容）
- 使用预装载值 (TIMx_ARR) 更新自动重载影子寄存器

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

图 401. 计数器时序图，1 分频内部时钟

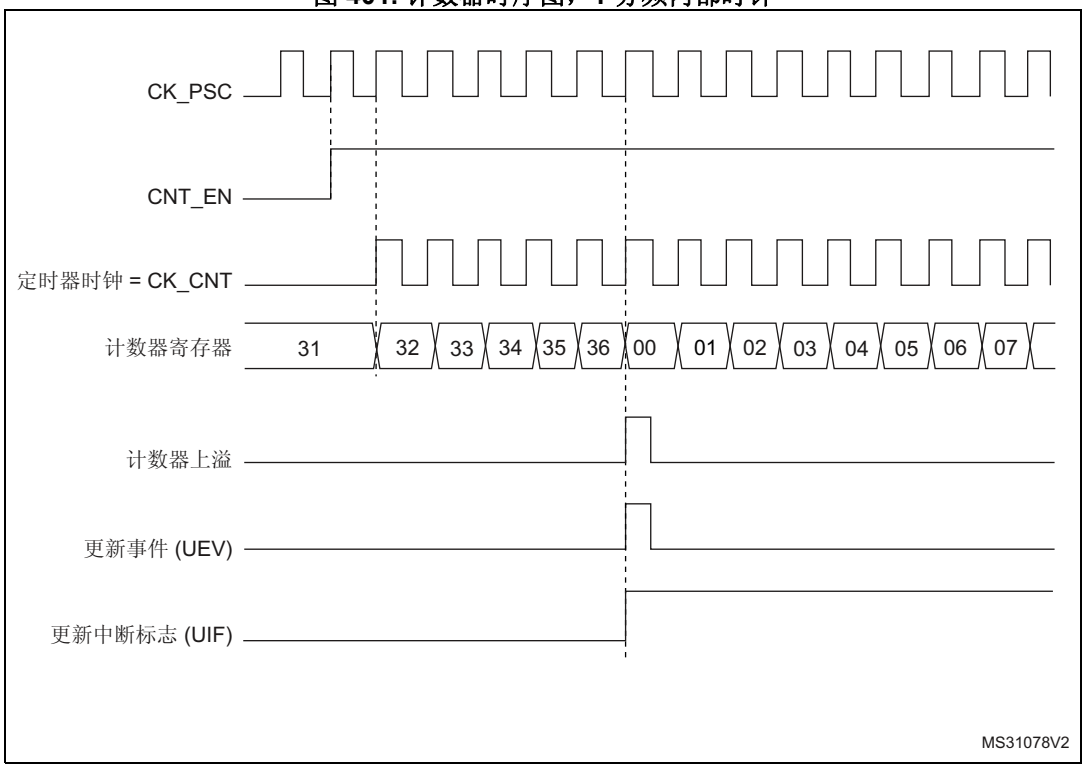


图 402. 计数器时序图，2 分频内部时钟

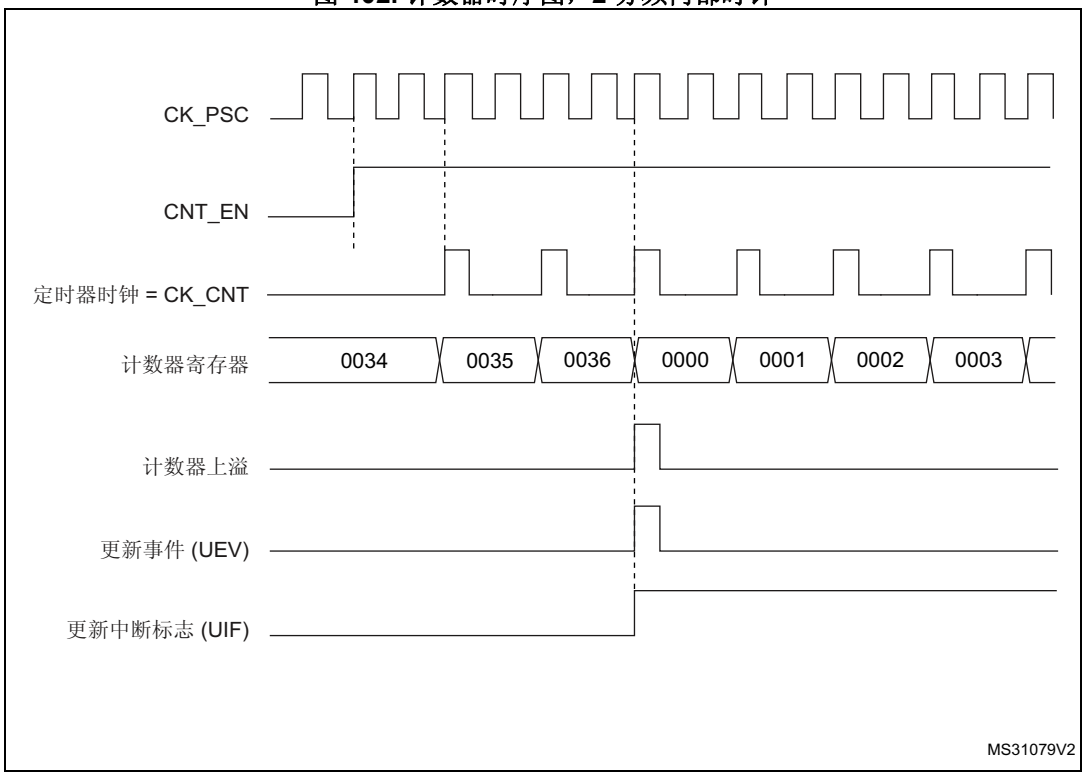


图 403. 计数器时序图，4 分频内部时钟

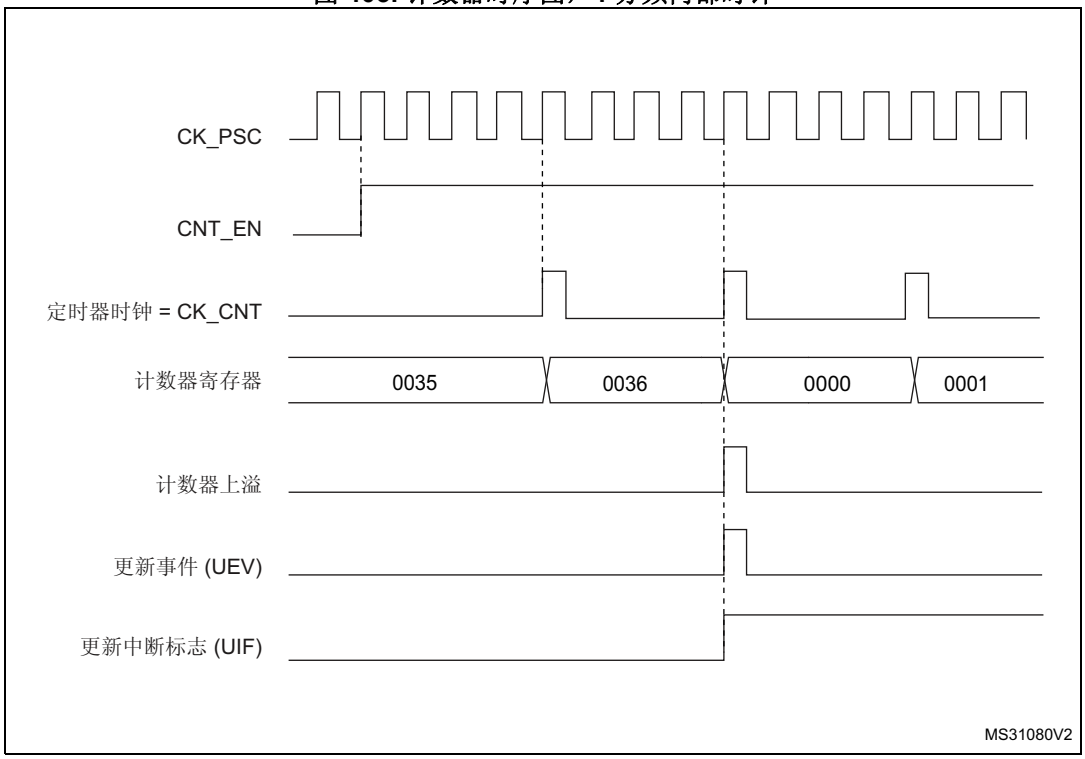


图 404. 计数器时序图，N 分频内部时钟

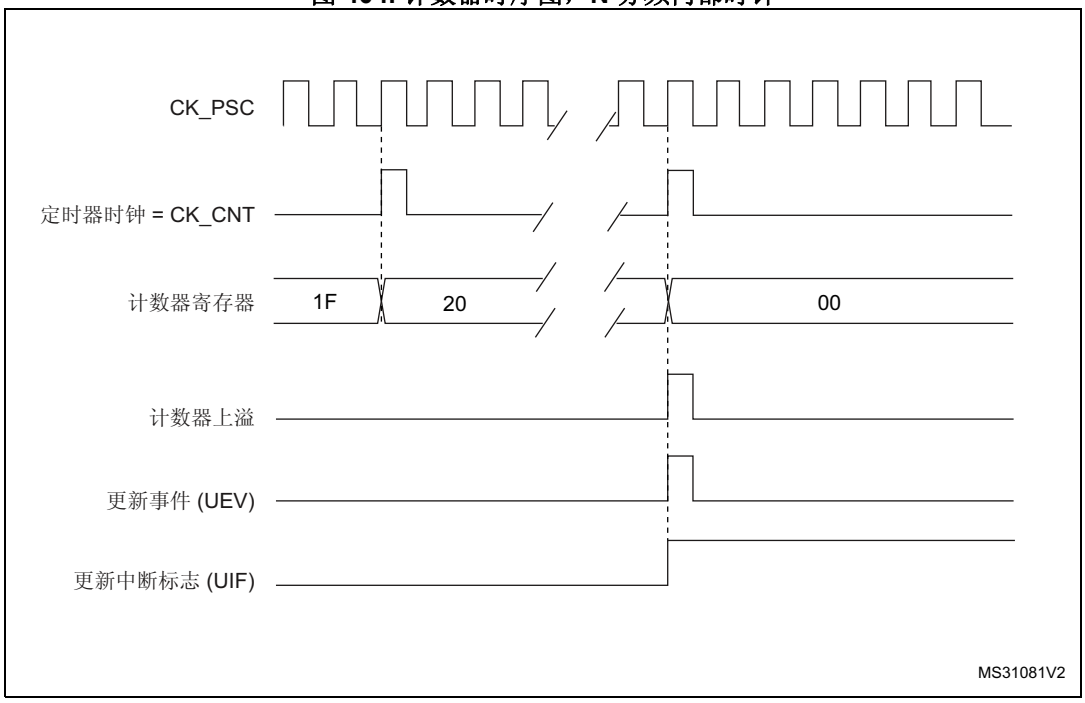


图 405. 计数器时序图，ARPE = 0 时更新事件（TIMx_ARR 未预装载）

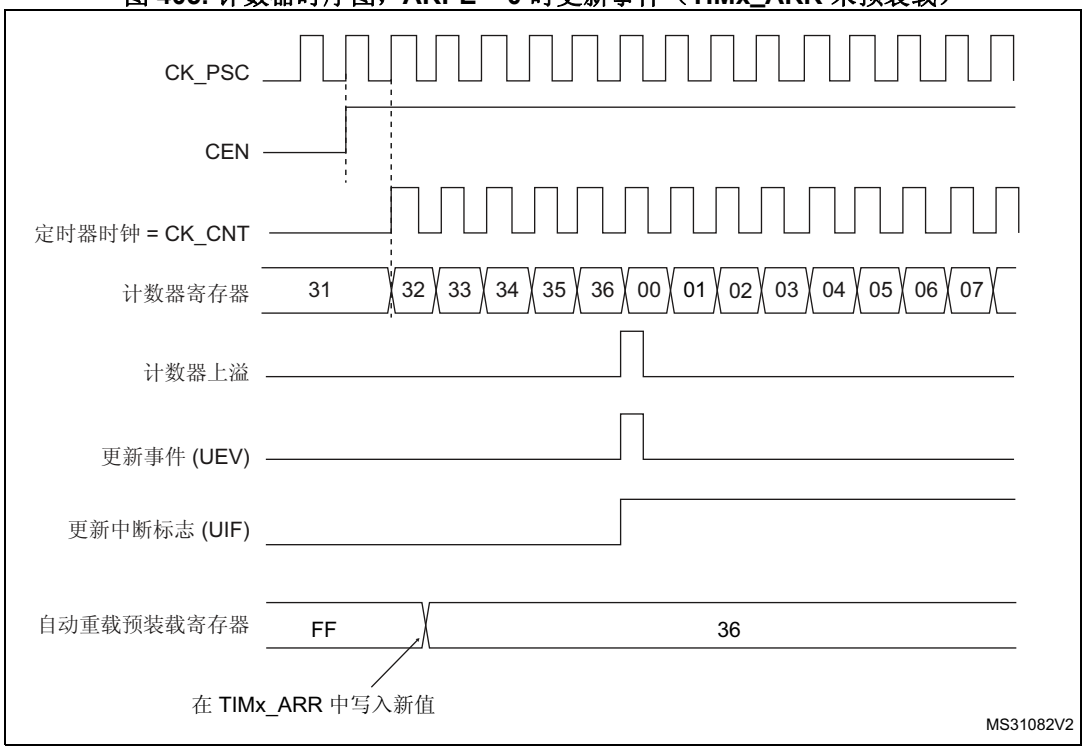
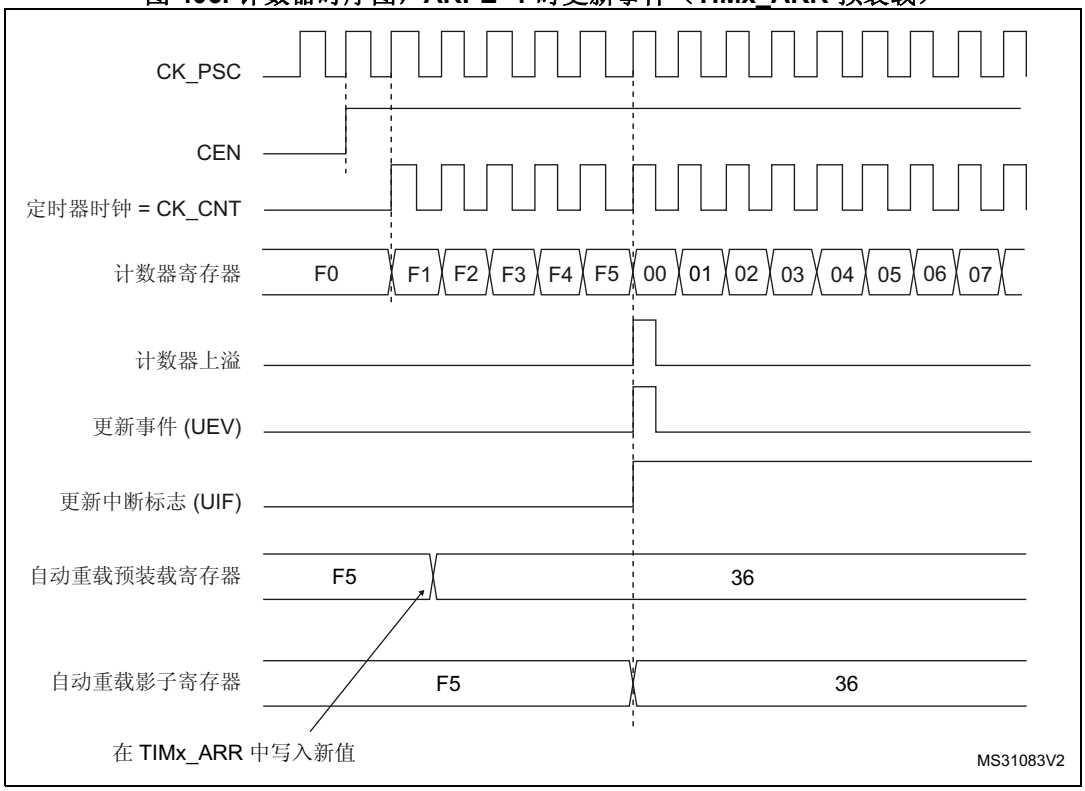


图 406. 计数器时序图，ARPE=1 时更新事件（TIMx_ARR 预装载）



37.3.3 UIF 位重映射

TIMx_CR1 寄存器中的 IUFREMAP 位强制将更新中断标志 UIF 连续复制到定时器计数器寄存器的位 31 (TIMxCNT[31]) 中。这样便可自动读取计数器值以及由 UIFCPY 标志发出的电位翻转条件。在特定情况下，这可避免在后台任务（计数器读）和中断（更新中断）之间共享处理时产生竞争条件，从而简化计算。

UIF 和 UIFCPY 标志使能之间没有延迟。

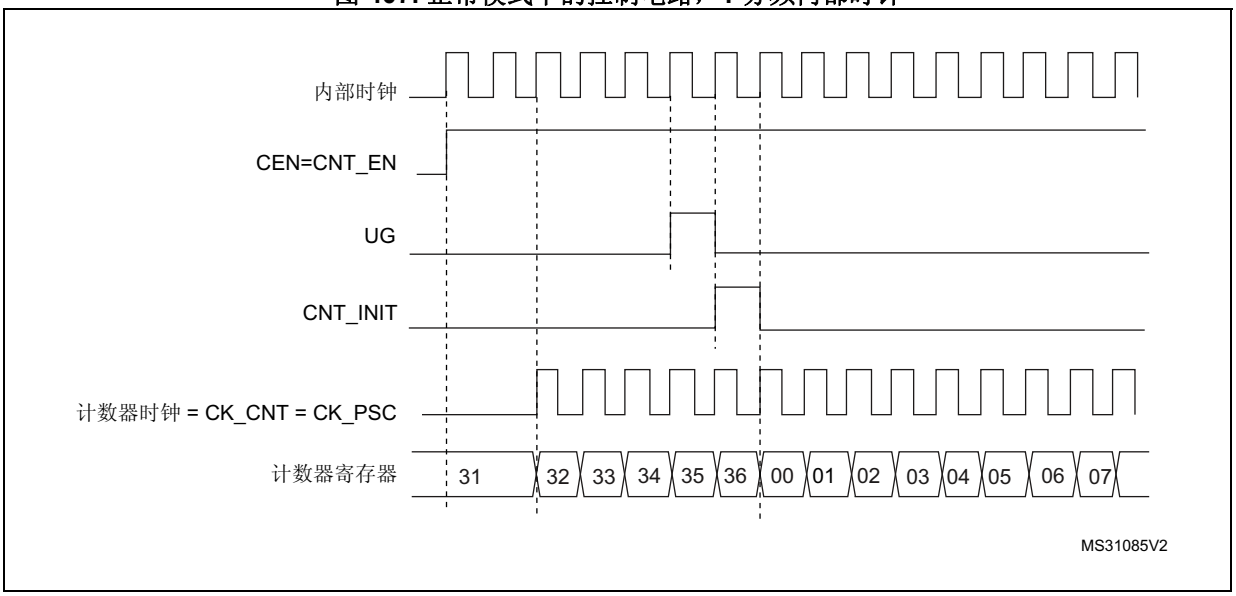
37.3.4 时钟源

计数器时钟由内部时钟 (CK_INT) 源提供。

CEN（TIMx_CR1 寄存器中）和 UG 位（TIMx_EGR 寄存器中）为实际控制位，并且只能通过软件进行更改（除了 UG 位被自动清零外）。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 407 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 407. 正常模式下的控制电路，1 分频内部时钟



37.3.5 调试模式

当微控制器进入调试模式时（Cortex®-M4 内核停止），TIMx 计数器会根据 DBG 模块中的 DBG_TIMx_STOP 配置位选择继续正常工作或者停止工作。有关详细信息，请参见*第 51.16.2 节：对定时器、RTC、看门狗、bxCAN 和 I2C 的调试支持*。

37.4 TIM6/TIM7 寄存器

有关寄存器说明中使用的缩写，请参见*第 72 页的第 1.1 节*。

外设寄存器可支持半字（16 位）或字（32 位）访问。

37.4.1 TIM6/TIM7 控制寄存器 1 (TIMx_CR1)

TIM6/TIM7 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
Res.	Res.	Res.	Res.	UIFREMAP	Res.	Res.	Res.	ARPE	Res.	Res.	Res.	OPM	URS	UDIS	CEN
				rw				rw				rw	rw	rw	rw

位 15:12 保留，必须保持复位值。

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

0：无重映射。UIF 状态位不复制到 TIMx_CNT 寄存器的位 31。

1：使能重映射。UIF 状态位复制到 TIMx_CNT 寄存器的位 31。

位 10:8 保留，必须保持复位值。

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

0：TIMx_ARR 寄存器不进行缓冲。

1：TIMx_ARR 寄存器进行缓冲。

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：使能时，所有以下事件都会产生更新中断或 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

然后带缓冲的寄存器被加载为预加载数值。

1：禁止 UEV。不会生成更新事件，各影子寄存器的值（ARR 和 PSC）保持不变。但如果将 UG 位置 1，或者从从模式控制器接收到硬件复位，则会重新初始化计数器和预分频器。

位 0 **CEN**：计数器使能 (Counter enable)

0：禁止计数器

1：使能计数器

注： 只有事先通过软件将 CEN 位置 1，才可以使用门控模式。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。

37.4.2 TIM6/TIM7 控制寄存器 2 (TIMx_CR2)

TIM6/TIM7 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.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MMS[2:0]			Res.	Res.	Res.	Res.
									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**——TIMx_EGR 寄存器中的 UG 位用作触发输出 (TRGO)。如果复位由触发输入产生（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 TIMx_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。

注： 必须先使能从定时器或 ADC 的时钟，才能从主定时器接收事件；并且从主定时器接收触发信号时，不得实时更改从定时器或 ADC 的时钟。

位 3:0 保留，必须保持复位值。

37.4.3 TIM6/TIM7 DMA/中断使能寄存器 (TIMx_DIER)

TIM6/TIM7 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.	Res.	Res.	Res.	Res.	Res.	Res.	UDE	Res.	Res.	Res.	Res.	Res.	Res.	Res.	UIE
							rw								rw

位 15:9 保留，必须保持复位值。

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 7:1 保留，必须保持复位值。

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。

37.4.4 TIM6/TIM7 状态寄存器 (TIMx_SR)

TIM6/TIM7 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
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	UIF
															rc_w0

位 15:1 保留，必须保持复位值。

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：

– 重复计数器值上溢或下溢并且 TIMx_CR1 寄存器中的 UDIS=0 时。

– 如果 TIMx_CR1 寄存器中 URS = 0 且 UDIS = 0，通过软件使用 TIMx_EGR 寄存器中的 UG 位重新初始化 CNT 时。

37.4.5 TIM6/TIM7 事件产生寄存器 (TIMx_EGR)

TIM6/TIM7 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
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	UG
															w

位 15:1 保留，必须保持复位值。

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：重新初始化定时器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）。

37.4.6 TIM6/TIM7 计数器 (TIMx_CNT)

TIM6/TIM7 counter

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

位 31 **UIFCPY**：UIF 副本 (UIF Copy)
该位是 TIMx_ISR 寄存器中 UIF 位的只读副本。如果 TIMx_CR1 中的 UIFREMAP 位复位，则位 31 保留，读为 0。

位 30:16 保留，必须保持复位值。

位 15:0 **CNT[15:0]**：计数器值 (Counter value)

37.4.7 TIM6/TIM7 预分频器 (TIMx_PSC)

TIM6/TIM7 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 f_{CK_PSC} / (PSC[15:0] + 1)。
PSC 包含在每次发生更新事件时要装载到有效预分频器寄存器的值。
（包括当计数器通过 TIMx_EGR 寄存器的 UG 位或在“复位模式”下通过触发控制器清零时）。

37.4.8 TIM6/TIM7 自动重载寄存器 (TIMx_ARR)

TIM6/TIM7 auto-reload register

偏移地址：0x2C

复位值：0xFFFF

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]**：预分频器值 (Prescaler 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*第 1296 页的第 37.3.1 节：时基单元*。
当自动重载值为空时，计数器不工作。

37.4.9 TIM6/TIM7 寄存器映射

TIMx 寄存器可映射为 16 位可寻址寄存器，如下表所述：

表 229. TIM6/TIM7 寄存器映射和复位值

偏移	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	**TIMx_CR1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	UIFREMAP	Res.	Res.	Res.	ARPE	Res.	Res.	Res.	OPM	URS	UDIS	CEN
	Reset value																					0				0				0	0	0	0
0x04	**TIMx_CR2**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MMS [2:0]			Res.	Res.	Res.	Res.
	Reset value																										0	0	0				
0x08	Reserved																																
0x0C	**TIMx_DIER**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	UDE	Res.	Res.	Res.	Res.	Res.	Res.	Res.	UIE
	Reset value																								0								0
0x10	**TIMx_SR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	UIF
	Reset value																																0
0x14	**TIMx_EGR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	UG
	Reset value																																0
0x18-0x20	Reserved																																
0x24	**TIMx_CNT**	UIFCPY or Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	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**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PSC[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x2C	**TIMx_ARR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ARR[15:0]															
	Reset value																	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

有关寄存器边界地址的信息，请参见*第 2.2.2 节：存储器映射和寄存器边界地址*。

38 低功耗定时器 (LPTIM)

38.1 简介

LPTIM 是一个 16 位定时器，可从降低功耗的最终发展中受益。由于 LPTIM 的时钟源具有多样性，因此 LPTIM 能够在所有电源模式（待机模式除外）下保持运行状态。即使没有内部时钟源，LPTIM 也能运行，鉴于这一点，可将其用作“脉冲计数器”，这种脉冲计数器在某些应用中十分有用。此外，LPTIM 还能将系统从低功耗模式唤醒，因此非常适合实现“超时功能”，在这种功能模式下系统功耗极低。

LPTIM 引入了一个灵活的时钟方案，该方案能够提供所需的功能和性能，同时还能最大程度地降低功耗。

38.2 LPTIM 主要特性

- 16 位递增计数器
- 3 位预分频器，可采用 8 种分频系数（1、2、4、8、16、32、64 和 128）
- 可选时钟
  - 内部时钟源：LSE、LSI、HSI16 或 APB 时钟
  - LPTIM 输入的外部时钟源（在没有 LP 振荡器运行的情况下工作，可在使用脉冲计数器应用场景中使用）
- 16 位 ARR 自动重载寄存器
- 16 位比较寄存器
- 连续/单触发模式
- 可选软件/硬件输入触发
- 可编程数字防抖动干扰滤波器
- 可配置输出：脉冲和 PWM
- 可配置 I/O 极性
- 编码器模式

38.3 LPTIM 实现

表 230 介绍了 STM32L4Rxxx 和 STM32L4Sxxx 器件上的 LPTIM 实现：LPTIM1 支持所有特性。LPTIM2 支持的特性略少，但在其余方面与 LPTIM1 完全相同。

表 230. STM32L4Rxxx 和 STM32L4Sxxx LPTIM 特性

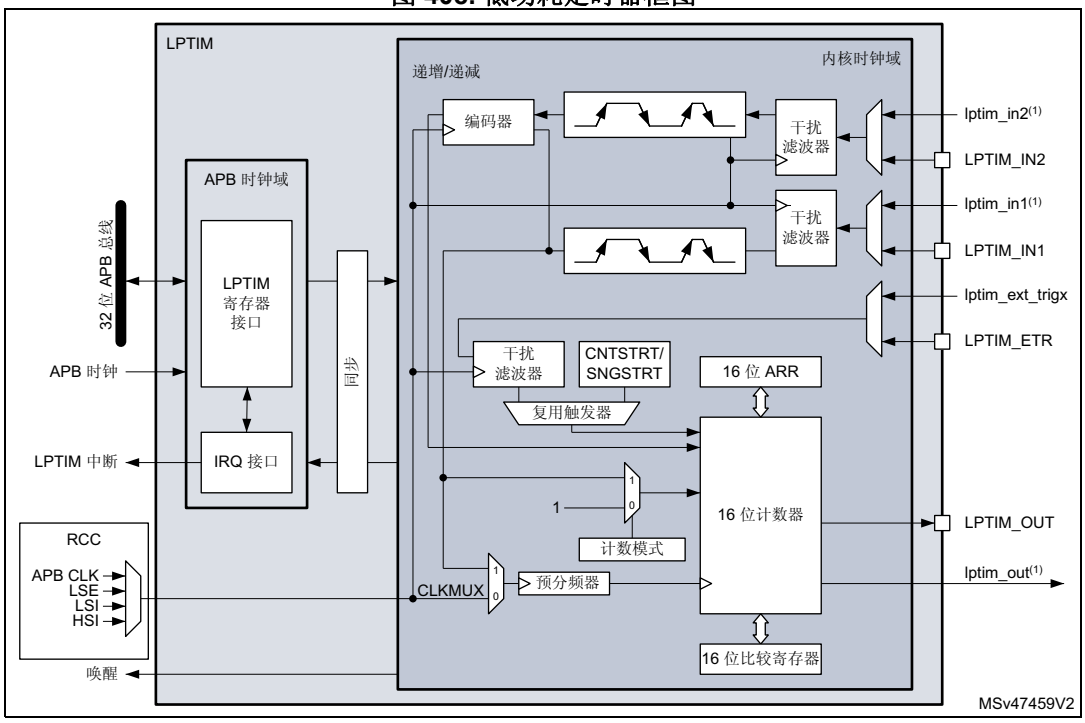
LPTIM 模式/特性(1)	LPTIM1	LPTIM2
编码器模式	X	-

1. X = 支持。

38.4 LPTIM 功能说明

38.4.1 LPTIM 框图

图 408. 低功耗定时器框图



1. lptim_in1 和 lptim_in2 分别为内部 LPTIM 输入 1 和输入 2 的信号，可以连接到内部外设。lptim_out 是内部 LPTIM 输出信号，可以连接到内部外设。

38.4.2 LPTIM 触发映射

下面详细介绍了 LPTIM 外部触发连接：

表 231. LPTIM1 外部触发连接

TRIGSEL	外部触发信号
lptim_ext_trig0	GPIO
lptim_ext_trig1	RTC 闹钟 A
lptim_ext_trig2	RTC 闹钟 B
lptim_ext_trig3	RTC_TAMP1 输入检测
lptim_ext_trig4	RTC_TAMP2 输入检测
lptim_ext_trig5	RTC_TAMP3 输入检测
lptim_ext_trig6	COMP1_OUT
lptim_ext_trig7	COMP2_OUT

表 232. LPTIM2 外部触发连接

TRIGSEL	外部触发信号
lptim_ext_trig0	GPIO
lptim_ext_trig1	RTC 闹钟 A
lptim_ext_trig2	RTC 闹钟 B
lptim_ext_trig3	RTC_TAMP1 输入检测
lptim_ext_trig4	RTC_TAMP2 输入检测
lptim_ext_trig5	RTC_TAMP3 输入检测
lptim_ext_trig6	COMP1_OUT
lptim_ext_trig7	COMP2_OUT

38.4.3 LPTIM 复位和时钟

LPTIM 可通过多个时钟源提供时钟。它可以由内部时钟信号提供时钟，内部时钟信号可通过复位和时钟控制器 (RCC) 在 APB、LSI、LSE 或 HSI16 时钟源中进行选择。此外，LPTIM 还可通过注入到其外部 Input1 上的外部时钟信号提供时钟。当通过外部时钟源提供时钟时，LPTIM 可以在下述两种可能配置中的其中一种配置下运行：

- 第一种配置是，LPTIM 通过外部信号提供时钟，但同时通过 APB 或 LSE、LSI 和 HSI16 等任何其他内置振荡器为 LPTIM 提供内部时钟信号。
- 第二种配置是，LPTIM 仅由外部时钟源通过外部 Input1 提供时钟。此配置可在进入低功耗模式后所有内置振荡器关闭时，用于实现超时功能或脉冲计数器功能。

对 CKSEL 和 COUNTMODE 位进行编程，可控制 LPTIM 使用外部时钟源还是内部时钟源。

当使用外部时钟源时，可使用 CKPOL 位选择外部时钟信号的有效边沿。如果上升沿和下降沿均为有效边沿，则还应提供内部时钟信号（第一种配置）。在这种情况下，内部时钟信号频率应至少为外部时钟信号频率的五倍。

38.4.4 干扰滤波器

外部（映射到微控制器 GPIO）或内部（映射到芯片级或其他嵌入式外设，例如嵌入式比较器）LPTIM 输入由数字滤波器保护，避免任何毛刺和噪声干扰在 LPTIM 内部传播，从而防止产生意外计数或触发。

在激活数字滤波器之前，首先应向 LPTIM 提供内部时钟源，这是保证滤波器正常工作的必要条件。

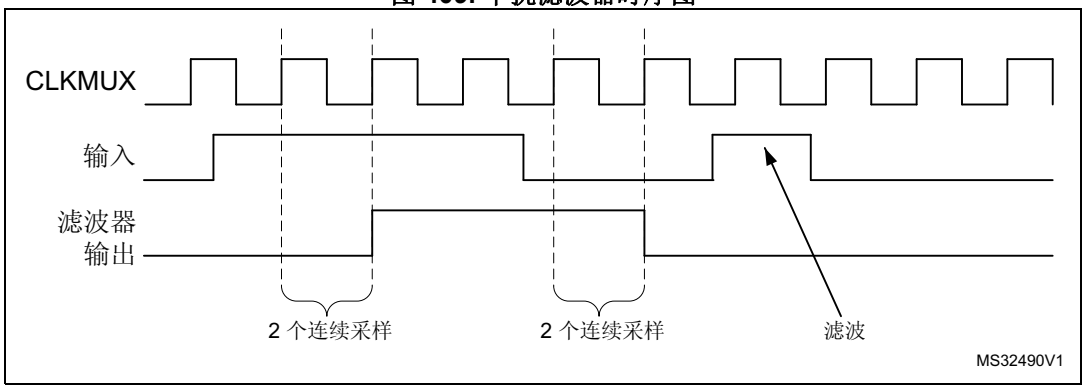
数字滤波器分为两组：

- 第一组数字滤波器保护 LPTIM 外部输入。数字滤波器的敏感性由 CKFLT 位控制。
- 第二组数字滤波器保护 LPTIM 内部触发输入。数字滤波器的敏感性由 TRGFLT 位控制。

注： *数字滤波器的敏感性以组为单位进行控制。无法单独配置同一组内各个数字滤波器的敏感性。*

滤波器的敏感性会影响相同的连续采样的数量，在其中一个 LPTIM 输入上检测到此类连续采样时，才能将某信号电平变化视为有效跳变。*图 409* 给出了编程 2 个连续采样时，干扰滤波器行为的示例。

图 409. 干扰滤波器时序图



注： *不提供内部时钟信号时，必须通过将 CKFLT 和 TRGFLT 位设为 0 来停用数字滤波器。在这种情况下，可使用外部模拟滤波器来防止 LPTIM 外部输入产生干扰。*

38.4.5 预分频器

LPTIM 16 位计数器前面要有一个可配置的 2 次幂预分频器。预分频器的分频比由 PRESC[2:0] 3 位域进行控制。下表列出了所有可能的分频比：

表 233. 预分频器的分频比

编程	分频系数
000	/1
001	/2
010	/4
011	/8
100	/16
101	/32
110	/64
111	/128

38.4.6 触发多路复用器

LPTIM 计数器可通过软件启动，也可以在 8 个触发输入之一上检测到有效边沿后启动。

TRIGEN[1:0] 用于确定 LPTIM 触发源：

- TRIGEN[1:0] 等于“00”时，LPTIM 计数器会在通过软件将 CNTSTRT 位或 SNGSTRT 位其中之一置 1 后立即启动。
- TRIGEN[1:0] 的其余三个可能的值用于配置触发输入使用的有效边沿。LPTIM 计数器会在检测到有效边沿后立即启动。

TRIGEN[1:0] 不等于“00”时，TRIGSEL[2:0] 用于选择使用 8 个触发输入中的哪一个来启动计数器。

外部触发信号视为 LPTIM 的异步信号。因此，检测到触发信号后，由于同步问题，需要延迟两个计数器时钟周期，定时器才能开始运行。

如果在定时器已启动时发生新的触发事件，则此事件将被忽略（除非已使能超时功能）。

注： *必须使能定时器，才能将 SNGSTRT/CNTSTRT 位置 1。当定时器禁止时，对这些位执行的任何写操作都将被硬件丢弃。*

38.4.7 工作模式

LPTIM 支持以下两种工作模式：

- 连续模式：定时器自由运行，由触发事件启动并且直到被禁止才会停止。
- 单触发模式：定时器由触发事件启动，当达到 ARR 值时停止。

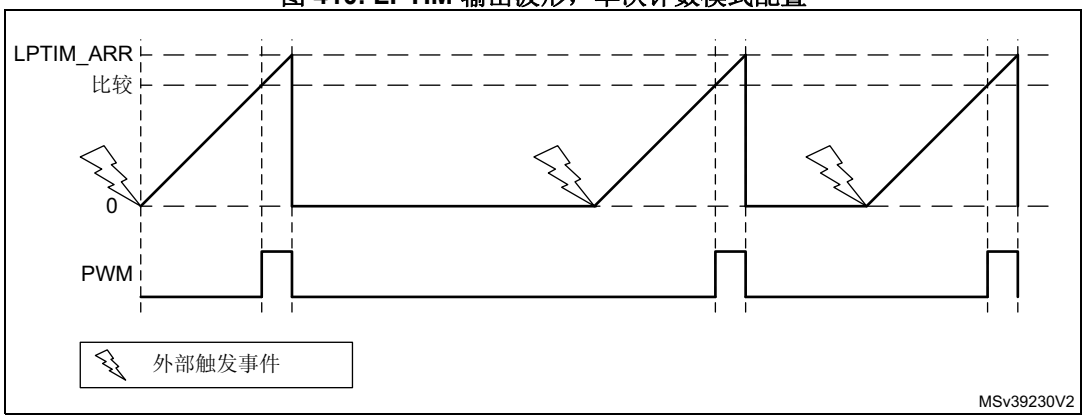
单触发模式

要使能单触发计数，必须将 SNGSTRT 位置 1。

新的触发事件将重新启动定时器。从计数器启动到计数器达到 ARR，这段时间内发生的任何触发事件均将被丢弃。

选择外部触发时，在 SNGSTRT 位置 1 后以及计数器寄存器停止后（包含零值）到达的每个外部触发事件都将为计数器启动新的单触发计数周期，如*图 410* 所示。

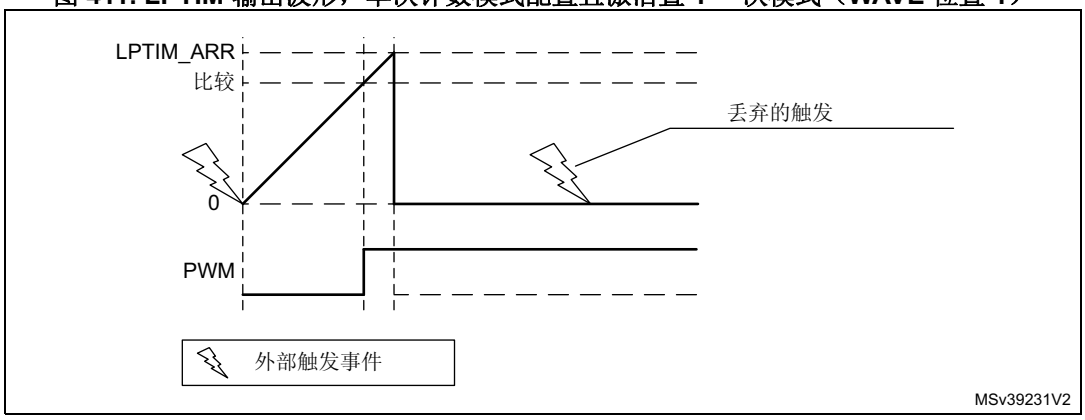
图 410. LPTIM 输出波形，单次计数模式配置



- 置 1 一次模式激活：

应注意，当 LPTIM_CFGR 寄存器中的 WAVE 位域置 1 时，将激活置 1 一次模式。在这种情况下，计数器仅会在第一个触发事件后启动一次，任何后续触发事件都将被丢弃，如*图 411* 所示。

图 411. LPTIM 输出波形，单次计数模式配置且激活置 1 一次模式（WAVE 位置 1）



若通过软件启动（TRIGEN[1:0] =“00”），将 SNGSTRT 置 1 会使计数器进行单触发计数。

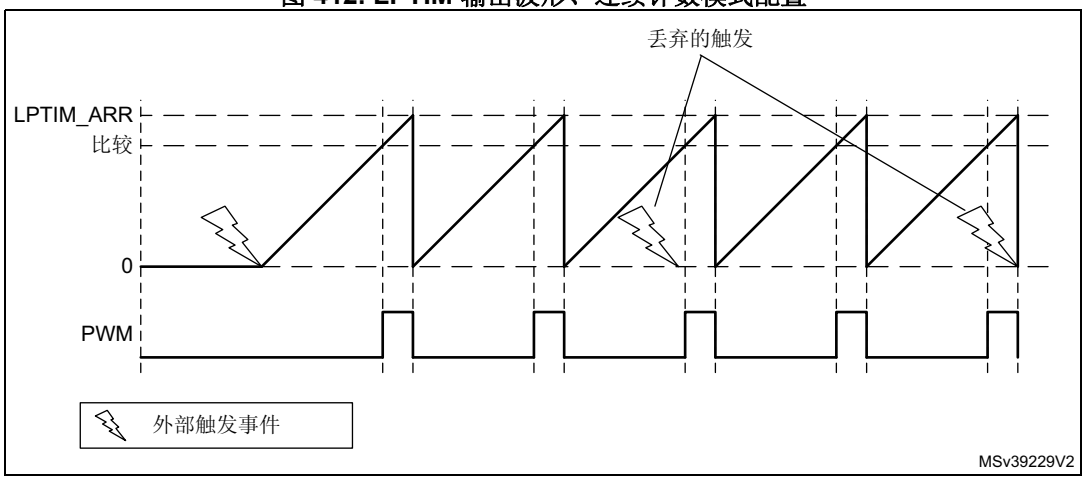
连续模式

要使能连续计数，必须将 CNTSTRT 位置 1。

若选择外部触发，则在 CNTSTRT 置 1 后到达的外部触发事件将启动计数器进行连续计数。任何后续的外部触发事件都将被丢弃，如*图 412* 所示。

若通过软件启动（TRIGEN[1:0] =“00”），将 CNTSTRT 置 1 会使计数器开始连续计数。

图 412. LPTIM 输出波形、连续计数模式配置



SNGSTRT 和 CNTSTRT 位只能在定时器使能时（ENABLE 位置 1）置 1。可以“实时”从单触发模式切换为连续模式。

如果之前选择的是连续模式，则将 SNGSTRT 置 1 会使 LPTIM 切换为单触发模式。计数器（激活时）将在达到 ARR 后立即停止。

如果之前选择的是单触发模式，则将 CNTSTRT 置 1 会使 LPTIM 切换为连续模式。计数器（激活时）将在达到 ARR 后立即重新启动。

38.4.8 超时功能

若在一个选定的触发输入上检测到有效边沿，则可用于复位 LPTIM 计数器。该功能通过 TIMOUT 位进行控制。

第一个触发事件将启动定时器，任何后续的触发事件将复位计数器，且定时器将重新启动。

可实现低功耗超时功能。超时值对应于比较值；如果在预期的时间帧内未发生触发事件，MCU 将由比较匹配事件唤醒。

38.4.9 生成波形

两个 16 位寄存器，LPTIM_ARR（自动重载寄存器）和 LPTIM_CMP（比较寄存器）用于在 LPTIM 输出上生成多个不同的波形

定时器可生成以下波形：

- PWM 模式：若 LPTIM_CMP 寄存器与 LPTIM_CNT 寄存器匹配，则会立即将 LPTIM 输出置 1。若 LPTIM_ARR 寄存器与 LPTIM_CNT 寄存器匹配，则会立即将 LPTIM 输出复位。
- 单脉冲模式：对于第一个脉冲，输出波形与 PWM 模式输出波形类似，随后输出将永久复位。
- 置 1 一次模式：除输出保持最后一个信号电平外（取决于配置的输出极性），输出波形与单脉冲模式输出波形类似。

上述模式要求 LPTIM_ARR 寄存器的值严格大于 LPTIM_CMP 寄存器的值。

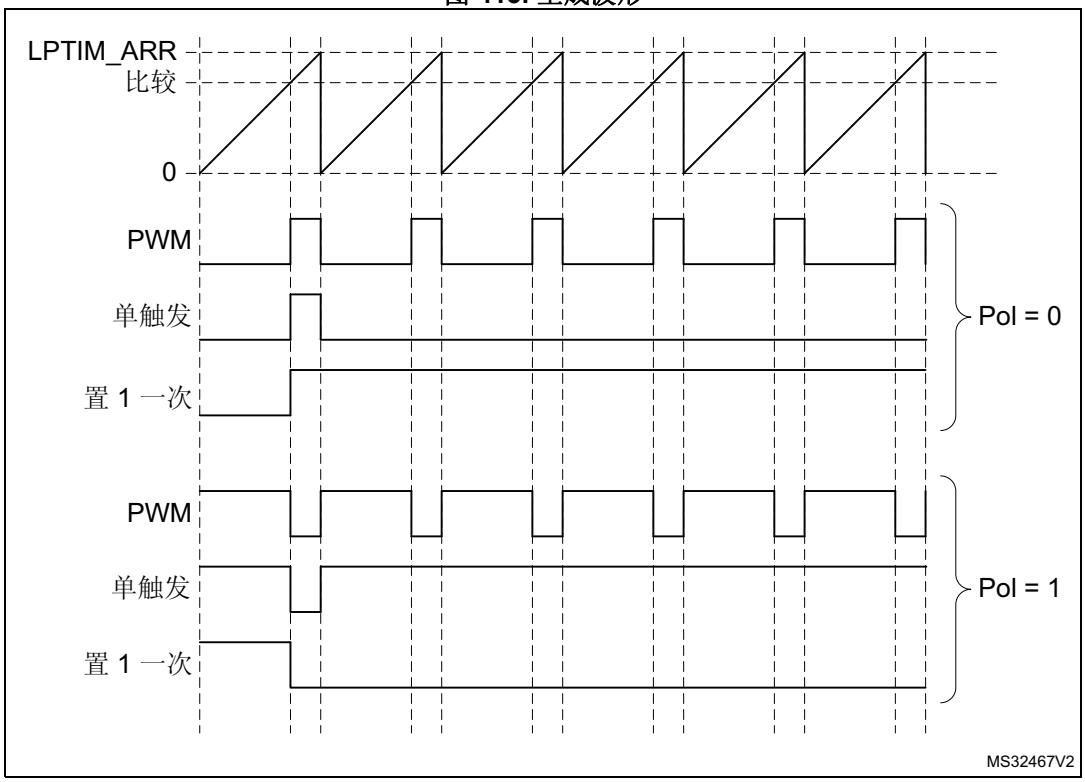
LPTIM 输出波形可通过 WAVE 位配置，具体如下：

- 若将 WAVE 位复位为 0，则会强制 LPTIM 生成 PWM 波形或单脉冲波形，具体取决于将哪个位（CNTSTRT 或 SNGSTRT）置 1。
- 若将 WAVE 位置 1，则会强制 LPTIM 生成置 1 一次模式波形。

WAVPOL 位控制 LPTIM 输出极性。更改立即生效，因此输出默认值将在极性重新配置后立即更改，甚至会在定时器使能前进行更改。

生成的信号的频率高达 LPTIM 时钟频率 2 分频。*图 413* 给出了可能在 LPTIM 输出上生成的三种波形。此外，此图还显示了通过 WAVPOL 位更改极性所产生的效果。

图 413. 生成波形



38.4.10 寄存器更新

LPTIM_ARR 寄存器和 LPTIM_CMP 寄存器在 APB 总线写操作后会立即更新，若定时器已启动，也可在当前周期结束时更新。

PRELOAD 位控制 LPTIM_ARR 寄存器和 LPTIM_CMP 寄存器的更新方式：

- 当 PRELOAD 位复位为 0 时，LPTIM_ARR 寄存器和 LPTIM_CMP 寄存器会在写访问后立即更新。
- 当 PRELOAD 位置 1 时，若定时器已启动，LPTIM_ARR 寄存器和 LPTIM_CMP 寄存器会在当前周期结束时更新。

LPTIM APB 接口和 LPTIM 内核逻辑使用的时钟不同，因此在 APB 写操作后，需要经过一定的延迟，写入值才能用于计数器比较器。在此延迟期间，必须避免向这些寄存器执行其他写操作。

LPTIM_ISR 寄存器中的 ARROK 标志和 CMPOK 标志分别指示 LPTIM_ARR 寄存器和 LPTIM_CMP 寄存器的写操作已完成。

向 LPTIM_ARR 寄存器和 LPTIM_CMP 寄存器执行写操作后，只有在前一次写操作完成后，才能对同一寄存器执行新的写操作。在 ARROK 标志或 CMPOK 标志置 1 前执行连续的写操作将造成无法预知的结果。

38.4.11 计数器模式

LPTIM 计数器可用于对 LPTIM Input1 上的外部事件进行计数，也可用于对内部时钟周期进行计数。CKSEL 位和 COUNTMODE 位用于控制将使用哪些源更新计数器。

若使用 LPTIM 对 Input1 上的外部事件进行计数，计数器可在上升沿、下降沿或两种边沿进行更新，具体取决于写入 CKPOL[1:0] 位的值。

根据 CKSEL 和 COUNTMODE 值，可选择以下计数模式：

- CKSEL = 0：LPTIM 由内部时钟源提供时钟
  - COUNTMODE = 0

    LPTIM 配置为由内部时钟源提供时钟，LPTIM 计数器配置为在每个内部时钟脉冲后进行更新。
  - COUNTMODE = 1

    LPTIM 外部 Input1 通过提供给 LPTIM 的内部时钟进行采样。

    因此，为了不丢失任何事件，外部 Input1 信号变化的频率决不应超过提供给 LPTIM 的内部时钟的频率。此外，不得对提供给 LPTIM 的内部时钟进行预分频 (PRESC[2:0] = 000)。
- CKSEL = 1：LPTIM 由外部时钟源提供时钟

  COUNTMODE 值不相关。

  在这种配置下，LPTIM 无需内部时钟源（已使能干扰滤波器时除外）。注入到 LPTIM 外部 Input1 的信号用作 LPTIM 的系统时钟。此配置适合未使能任何内置振荡器的工作模式。

  对于这种配置，LPTIM 计数器可以在 input1 时钟信号的上升沿或下降沿进行更新，但不可在上升沿和下降沿均更新。

  由于注入到 LPTIM 外部 Input1 的信号也可用于为 LPTIM 内核逻辑提供时钟，计数器递增计数前存在一些初始延时（使能 LPTIM 后）。更确切地说，LPTIM 外部 Input1 的前五个有效边沿将丢失（使能 PTIM 后）。

38.4.12 定时器使能

LPTIM_CR 寄存器中的 ENABLE 位用于使能/禁止 LPTIM 内核逻辑。将 ENABLE 位置 1 后，需要延迟两个计数器时钟周期，才能真正使能 LPTIM。

LPTIM_CFGR 和 LPTIM_IER 寄存器必须在禁止 LPTIM 后才能修改。

38.4.13 定时器计数器复位

为了将 LPTIM_CNT 寄存器的内容复位为零，实现了两种复位机制：

- 同步复位机制：同步复位由 LPTIM_CR 寄存器中的 COUNTRST 位控制。在将 COUNTRST 位域置 1 后，复位信号在 LPTIM 内核时钟域中传播。因此，重要的是要注意，要在经历几个 LPTIM 内核逻辑时钟脉冲之后再考虑复位。这将使 LPTIM 计数器在复位触发和生效之间额外计数几个脉冲。由于 COUNTRST 位位于 APB 时钟域中，并且 LPTIM 计数器位于 LPTIM 内核时钟域中，因此当将 1 写入到 COUNTRST 位时，内核时钟需要 3 个时钟周期的延迟用以同步由 APB 时钟域发出的复位信号。
- 异步复位机制：异步复位由位于 LPTIM_CR 寄存器中的 RSTARE 位控制。当该位置 1 时，对 LPTIM_CNT 寄存器的任何读访问都会将其内容复位为零。应在不提供 LPTIM 内核时钟的时间范围内触发异步复位。例如，当 LPTIM Input1 管脚为外部时钟输入管脚时，只有当足够保证 LPTIM Input1 不会发生反转时，才应用异步复位。
  应注意的是，为了实现可靠的 LPTIM_CNT 寄存器内容读取，必须执行两次连续的读访问并进行比较。当两次读访问的值相等时，可认为读访问可靠。然而，当使能了异步复位时，不可能两次读取 LPTIM_CNT 寄存器。

警告： LPTIM 内没有防止两个复位机制同时使用的机制。所以开发人员应该确保这两个机制是排斥地使用。

38.4.14 编码器模式

此模式用于处理来自正交编码器的信号，此正交编码器用于检测旋转元件的角度位置。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 LPTIM_ARR 寄存器中编程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 LPTIM_ARR。通过两个外部输入信号 Input1 和 Input2 生成时钟信号作为 LPTIM 计数器时钟。这两个信号间的相位确定计数方向。

仅当 LPTIM 由内部时钟源提供时钟时才可使用编码器模式。Input1 和 Input2 输入上的信号频率不得超过 LPTIM 内部时钟频率 4 分频。必须满足此条件才能确保 LPTIM 正常工作。

方向变化由 LPTIM_ISR 寄存器中的两个递减和递增标志指示。此外，如果通过 LPTIM_IER 寄存器使能，还可为两种方向变化事件产生中断。

要激活编码器模式，必须将 ENC 位置 1。LPTIM 必须首先配置为连续模式。

当编码器模式激活时，LPTIM 计数器按照增量编码器的速度和方向自动修改。因此，其内容始终代表编码器的位置。计数方向由递增和递减标志指示，对应于编码器转子的旋转方向。

根据使用 CKPOL[1:0] 位配置的边沿敏感性，可得几种不同的计数方案。下表汇总了可能的组合（假设 Input1 和 Input2 不同时切换）。

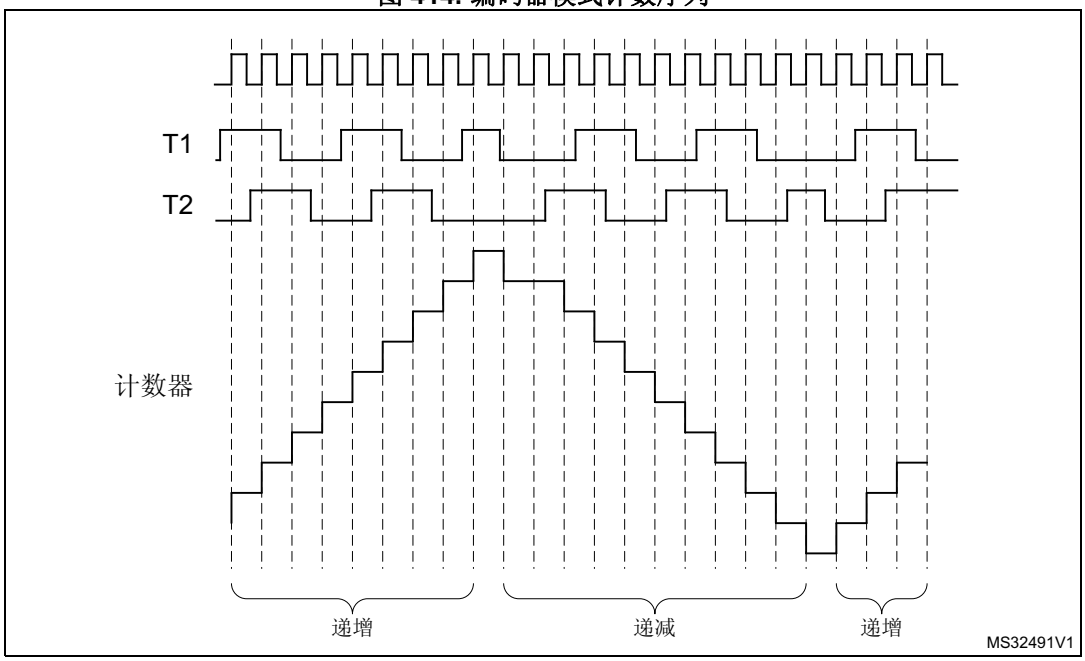
表 234. 编码器计数方案

有效边沿	相反信号的电平（Input1 对应 Input2，Input2 对应 Input1）	Input1 信号		Input2 信号	
		上升	下降	上升	下降
上升沿	高	递减	不计数	递增	不计数
	低	递增	不计数	递减	不计数
下降沿	高	不计数	递增	不计数	递减
	低	不计数	递减	不计数	递增
两种边沿	高	递减	递增	递增	递减
	低	递增	递减	递减	递增

下图所示为编码器模式下配置了两种边沿敏感性的计数序列。

注意： 在此模式下，LPTIM 必须由内部时钟源提供时钟，因此 CKSEL 位必须保持其复位值 0。另外，预分频器分频比必须等于其复位值 1（PRESC[2:0] 位必须为“000”）。

图 414. 编码器模式计数序列



38.5 LPTIM 低功耗模式

表 235. 低功耗模式对 LPTIM 的影响

模式	说明
睡眠	无影响。LPTIM 中断可使器件退出睡眠模式。
低功耗运行	无影响。
低功耗睡眠	无影响。LPTIM 中断可使器件退出低功耗睡眠模式。
停止 0/停止 1	当 LSE 或 LSI 锁定 LPTIM 时无影响。LPTIM 中断会导致器件退出停止 0 和停止 1。
停止 2	当 LSE 或 LSI 锁定 LPTIM1 时对 LPTIM1 无影响。LPTIM1 中断会导致器件退出停止 2。LPTIM2 寄存器的内容会保留，但是在进入停止 2 之前必须禁止 LPTIM2。
待机	LPTIM 外设掉电，退出待机模式或关断模式后必须重新初始化。
关断	

38.6 LPTIM 中断

若以下事件通过 LPTIM_IER 寄存器使能，则这些事件会生成中断/唤醒事件：

- 比较匹配
- 自动重载匹配（编码器模式下无论哪种方向）
- 外部触发事件
- 自动重载寄存器写操作完成
- 比较寄存器写操作完成
- 方向变化（编码器模式），可编程（递增/递减/同时递增和递减）

注： *只要 LPTIM_IER 寄存器（中断使能寄存器）中的位在 LPTIM_ISR 寄存器（状态寄存器）中相应标志置 1 后置 1，就不会触发中断*

表 236. 中断事件

中断事件	说明
比较匹配	当计数器寄存器 (LPTIM_CNT) 的内容与比较寄存器 (LPTIM_CMP) 的内容匹配时，生成中断标志。
自动重载匹配	当计数器寄存器 (LPTIM_CNT) 的内容与自动重载寄存器 (LPTIM_ARR) 的内容匹配时，生成中断标志。
外部触发事件	当检测到外部触发事件时，会生成中断标志。
自动重载寄存器更新成功	当对 LPTIM_ARR 寄存器的写操作完成时，生成中断标志。
比较寄存器更新成功	当对 LPTIM_CMP 寄存器的写操作完成时，生成中断标志。
方向更改	用于编码器模式。嵌入两个中断标志以发出方向更改的信号： – UP 标志发出递增计数方向更改的信号。 – DOWN 标志发出递减计数方向更改的信号。

38.7 LPTIM 寄存器

38.7.1 LPTIM 中断和状态寄存器 (LPTIM_ISR)

LPTIM interrupt and 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
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.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DOWN	UP	ARROK	CMPOK	EXTTRIG	ARRM	CMPM
									r	r	r	r	r	r	r

位 31:9 保留，必须保持复位值。

位 8:7 保留，必须保持复位值。

位 6 **DOWN**：计数方向从递增变为递减 (Counter direction change up to down)
在编码器模式下，由硬件将 DOWN 位置 1 时，会通知应用计数方向由递增变为递减。

位 5 **UP**：计数方向从递减变为递增 (Counter direction change down to up)
在编码器模式下，由硬件将 UP 位置 1 时，会通知应用计数方向由递减变为递增。

位 4 **ARROK**：自动重载寄存器更新成功 (Autoreload register update OK)
由硬件将 ARROK 置 1 时，会通知应用 LPTIM_ARR 寄存器的 APB 总线写操作已成功完成。

位 3 **CMPOK**：比较寄存器更新成功 (Compare register update OK)
由硬件将 CMPOK 置 1 时，会通知应用 LPTIM_CMP 寄存器的 APB 总线写操作已成功完成。

位 2 **EXTTRIG**：外部触发边沿事件 (External trigger edge event)
由硬件将 EXTTRIG 置 1 时，会通知应用所选的外部触发输入上产生有效边沿。如果由于定时器已启动而忽略触发事件，则不会将此标志置 1。

位 1 **ARRM**：自动重载匹配 (Autoreload match)
由硬件将 ARRM 置 1 时，会通知应用 LPTIM_CNT 寄存器的值已达到 LPTIM_ARR 寄存器的值。

位 0 **CMPM**：比较匹配 (Compare match)
由硬件将 CMPM 位置 1 时，会通知应用 LPTIM_CNT 寄存器的值已达到 LPTIM_CMP 寄存器的值。

38.7.2 LPTIM 中断清零寄存器 (LPTIM_ICR)

LPTIM interrupt clear 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
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.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DOWNCF	UPCF	ARROKCF	CMPOKCF	EXTTRIGCF	ARRMCF	CMPMCF
									w	w	w	w	w	w	w

位 31:9 保留，必须保持复位值。

位 8:7 保留，必须保持复位值。

位 6 **DOWNCF**：方向变为递减清零标志 (Direction change to down Clear Flag)
将 1 写入此位时，LPT_ISR 寄存器中的 DOWN 标志将清零。

位 5 **UPCF**：方向变为递增清零标志 (Direction change to UP Clear Flag)
将 1 写入此位时，LPT_ISR 寄存器中的 UP 标志将清零。

位 4 **ARROKCF**：自动重载寄存器更新成功清零标志 (Autoreload register update OK Clear Flag)
将 1 写入此位时，LPT_ISR 寄存器中的 ARROK 标志将清零。

位 3 **CMPOKCF**：比较寄存器更新成功清零标志 (Compare register update OK Clear Flag)
将 1 写入此位时，LPT_ISR 寄存器中的 CMPOK 标志将清零。

位 2 **EXTTRIGCF**：外部触发有效边沿清零标志 (External trigger valid edge Clear Flag)
将 1 写入此位时，LPT_ISR 寄存器中的 EXTTRIG 标志将清零。

位 1 **ARRMCF**：自动重载匹配清零标志 (Autoreload match Clear Flag)
将 1 写入此位时，LPT_ISR 寄存器中的 ARRM 标志将清零。

位 0 **CMPMCF**：比较匹配清零标志 (compare match Clear Flag)
将 1 写入此位时，LPT_ISR 寄存器中的 CMP 标志将清零。

38.7.3 LPTIM 中断使能寄存器 (LPTIM_IER)

LPTIM interrupt enable 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
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.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DOWNIE	UPIE	ARROKIE	CMPOKIE	EXTTRIGIE	ARRMIE	CMPMIE
									rw	rw	rw	rw	rw	rw	rw

位 31:9 保留，必须保持复位值。

位 8:7 保留，必须保持复位值。

位 6 **DOWNIE**：方向变为递减中断使能 (Direction change to down Interrupt Enable)
0：禁止 DOWN 中断
1：使能 DOWN 中断

位 5 **UPIE**：方向变为递增中断使能 (Direction change to UP Interrupt Enable)
0：禁止 UP 中断
1：使能 UP 中断

位 4 **ARROKIE**：自动重载寄存器更新成功中断使能 (Autoreload register update OK Interrupt Enable)
0：禁止 ARROK 中断
1：使能 ARROK 中断

位 3 **CMPOKIE**：比较寄存器更新成功中断使能 (Compare register update OK Interrupt Enable)
0：禁止 CMPOK 中断
1：使能 CMPOK 中断

位 2 **EXTTRIGIE**：外部触发有效边沿中断使能 (External trigger valid edge Interrupt Enable)
0：禁止 EXTTRIG 中断
1：使能 EXTTRIG 中断

位 1 **ARRMIE**：自动重载匹配中断使能 (Autoreload match Interrupt Enable)
0：禁止 ARRM 中断
1：使能 ARRM 中断

位 0 **CMPMIE**：比较匹配中断使能 (Compare match Interrupt Enable)
0：禁止 CMPM 中断
1：使能 CMPM 中断

注意： 必须在 LPTIM 已时禁止（ENABLE 位复位为 0）才能修改 LPTIM_IER 寄存器

38.7.4 LPTIM 配置寄存器 (LPTIM_CFGR)

LPTIM configuration 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
Res.	Res.	Res.	Res.	Res.	Res.	Res.	ENC	COUNT MODE	PRELOAD	WAVPOL	WAVE	TIMOUT	TRIGEN[1:0]		Res.
							rw	rw	rw	rw	rw	rw	rw	rw	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TRIGSEL[2:0]			Res.	PRESC[2:0]			Res.	TRGFLT[1:0]		Res.	CKFLT[1:0]		CKPOL[1:0]		CKSEL
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 保留，必须保持复位值。

位 28:25 保留，必须保持复位值。

位 24 **ENC**：编码器模式使能 (Encoder mode enable)

ENC 位控制编码器模式

0：禁止编码器模式

1：使能编码器模式

位 23 **COUNTMODE**：计数器模式使能 (counter mode enabled)

COUNTMODE 位用于选择 LPTIM 使用哪个时钟源来为计数器提供时钟：

0：计数器在每个内部时钟脉冲后递增

1：计数器在 LPTIM 外部 Input1 上的每个有效时钟脉冲后递增

位 22 **PRELOAD**：寄存器更新模式 (Registers update mode)

PRELOAD 位控制 LPTIM_ARR 寄存器和 LPTIM_CMP 寄存器的更新方式

0：寄存器在每次 APB 总线写访问后更新

1：寄存器在当前 LPTIM 周期结束时更新

位 21 **WAVPOL**：波形极性 (Waveform shape polarity)

WAVEPOL 位控制输出极性

0：LPTIM 输出反映 LPTIM_ARR 寄存器和 LPTIM_CMP 寄存器之间的比较结果

1：LPTIM 输出反映与 LPTIM_ARR 寄存器和 LPTIM_CMP 寄存器之间的比较结果相反的值

位 20 **WAVE**：波形 (Waveform shape)

WAVE 位控制输出波形

0： 停用置 1 一次模式和 PWM/单脉冲波形（具体取决于 OPMODE 位）

1： 激活置 1 一次模式

位 19 **TIMOUT**：超时使能 (Timeout enable)

TIMOUT 位控制超时功能

0：定时器已启动时到达的触发事件将被忽略

1：定时器已启动时到达的触发事件将复位并重新启动计数器

位 18:17 **TRIGEN[1:0]**：触发使能和极性 (Trigger enable and polarity)

TRIGEN 位控制 LPTIM 计数器是否由外部触发信号启动。如果已选择由外部触发信号启动，触发有效边沿的配置有以下三种：

00：软件触发（由软件启动计数）

01：上升沿为有效边沿

10：下降沿为有效边沿

11：上升沿和下降沿均为有效边沿

位 16 保留，必须保持复位值。

位 15:13 **TRIGSEL[2:0]**：触发源选择器 (Trigger selection)

TRIGSEL 位用于选择作为 LPTIM 触发事件的触发源，可用触发源包括以下 8 种：

000：lptim_ext_trig0

001：lptim_ext_trig1

010：lptim_ext_trig2

011：lptim_ext_trig3

100：lptim_ext_trig4

101：lptim_ext_trig5

110：lptim_ext_trig6

111：lptim_ext_trig7

详细信息，请参见*第 38.4.2 节：LPTIM 触发映射*。

位 12 保留，必须保持复位值。

位 11:9 **PRESC[2:0]**：时钟预分频器 (Clock prescaler)

PRESC 位配置预分频器的分频系数。分频系数可从以下分频系数中选择：

000：/1

001：/2

010：/4

011：/8

100：/16

101：/32

110：/64

111：/128

位 8 保留，必须保持复位值。

位 7:6 **TRGFLT[1:0]**：触发信号的可配置数字滤波器 (Configurable digital filter for trigger)

TRGFLT 值用于设置连续相同采样的数量，若在内部触发信号电平发生变化时检测到此类连续采样，才会将此电平变化视为有效电平切换。必须存在内部时钟源才能使用此功能。

00：任何触发信号有效电平变化均视为有效触发。

01：触发信号有效电平变化必须至少稳定 2 个时钟周期，才能将其视为有效触发。

10：触发信号有效电平变化必须至少稳定 4 个时钟周期，才能将其视为有效触发。

11：触发信号有效电平变化必须至少稳定 8 个时钟周期，才能将其视为有效触发。

位 5 保留，必须保持复位值。

位 4:3 **CKFLT[1:0]**：外部时钟的可配置数字滤波器 (Configurable digital filter for external clock)

CKFLT 值用于设置连续相同采样的数量，若在外部时钟信号电平发生变化时检测到此类连续采样，才会将此电平变化视为有效电平切换。必须存在内部时钟源才能使用此功能。

00：任何外部时钟信号电平变化均视为有效切换。

01：外部时钟信号电平变化必须至少稳定 2 个时钟周期，才能将其视为有效切换。

10：外部时钟信号电平变化必须至少稳定 4 个时钟周期，才能将其视为有效切换。

11：外部时钟信号电平变化必须至少稳定 8 个时钟周期，才能将其视为有效切换。

位 2:1 **CKPOL[1:0]**：时钟极性 (Clock Polarity)

如果 LPTIM 由外部时钟源提供时钟：

当 LPTIM 由外部时钟源提供时钟时，CKPOL 位用于配置计数所使用的有效边沿：

00：上升沿为用于计数的有效边沿。

01：下降沿为用于计数的有效边沿。

10：上升沿和下降沿均为有效边沿。当外部时钟信号的上升沿和下降沿均视为有效边沿时，LPTIM 还必须由内部时钟源提供时钟，且内部时钟源频率至少等于外部时钟频率的四倍。

11：不允许

如果将 LPTIM 配置为编码器模式（ENC 位置 1）：

00：编码器子模式 1 激活。

01：编码器子模式 2 激活。

10：编码器子模式 3 激活。

有关编码器模式子模式的更多详细信息，请参见*第 38.4.14 节：编码器模式*。

位 0 **CKSEL**：时钟选择器 (Clock selector)

CKSEL 位选择 LPTIM 将使用的时钟源：

0：LPTIM 由内部时钟源（APB 时钟或任意内置振荡器）提供时钟。

1：LPTIM 由外部时钟源通过 LPTIM 外部 Input1 提供时钟。

注意： 必须在 LPTIM 已禁止时（ENABLE 位复位为 0）才能修改 LPTIM_CFGR 寄存器。

38.7.5 LPTIM 控制寄存器 (LPTIM_CR)

LPTIM control 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
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.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RSTARE	COUNTRST	CNTSTRT	SNGSTRT	ENABLE
											w	rs	rw	rw	rw

位 31:5 保留，必须保持复位值。

位 4 **RSTARE**：读操作后复位使能 (Reset after read enable)

此位由软件置 1 和清零。当 RSTARE 置 1 时，对 LPTIM_CNT 寄存器的任何读取访问都将异步复位 LPTIM_CNT 寄存器的内容。

注意： 只能对该位域执行写操作。这意味着该位不能被读回以验证已写入的值。例如，如果该位设置为 1，即使使能了“读操作后复位”功能（由于该位域以前已写入 1），尝试读取它也将返回 0。要关闭“读操作后复位”或确保已经关闭，应（通过将其编程为 0）将该位复位，即使其已经为 0。

位 3 **COUNTRST**：计数器复位 (Counter reset)

此位通过软件置 1，通过硬件清零。当设置为 1 时，该位将触发 LPTIM_CNT 计数器寄存器的同步复位。由于该复位的同步性质，它仅在 3 个 LPTimer 内核时钟周期（LPTimer 内核时钟可能不同于 APB 时钟）的同步延迟之后发生。

注意： COUNTRST 在已由硬件清零之前，不得由软件置“1”。因此，软件应在尝试将 COUNTRST 位置“1”之前检查其是否已清零。

位 2 **CNTSTRT**：定时器以连续模式启动 (Timer start in Continuous mode)

此位通过软件置 1，通过硬件清零。

若通过软件启动（TRIGEN[1:0] =“00”），将此位置 1 会使 LPTIM 以连续模式启动。

如果禁止软件启动（TRIGEN[1:0] 不等于“00”），将此位置 1 会使定时器在检测到外部触发信号后立即以连续模式启动。

如果在进行单脉冲模式计数时将此位置 1，则在 LPTIM_ARR 寄存器和 LPTIM_CNT 寄存器下一次匹配时定时器不会停止，LPTIM 计数器将继续以连续模式计数。

只有在使能 LPTIM 时才能将此位置 1。此位由硬件自动复位。

位 1 **SNGSTRT**：LPTIM 以单脉冲模式启动 (LPTIM start in Single mode)

此位通过软件置 1，通过硬件清零。

若通过软件启动（TRIGEN[1:0] =“00”），将此位置 1 会使 LPTIM 以单脉冲模式启动。

如果禁止软件启动（TRIGEN[1:0] 不等于“00”），将此位置 1 会使 LPTIM 在检测到外部触发信号后立即以单脉冲模式启动。

如果在 LPTIM 处于连续计数模式时将此位置 1，LPTIM 将在 LPTIM_ARR 寄存器和 LPTIM_CNT 寄存器下一次匹配时停止。

只有在使能 LPTIM 时才能将此位置 1。此位由硬件自动复位。

位 0 **ENABLE**：LPTIM 使能 (LPTIM enable)

ENABLE 位由软件置 1 和清零。

0：禁止 LPTIM

1：使能 LPTIM

38.7.6 LPTIM 比较寄存器 (LPTIM_CMP)

LPTIM compare 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
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
CMP[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 **CMP[15:0]**：比较值 (Compare value)

CMP 为 LPTIM 所使用的比较值。

注意： 必须在 LPTIM 已使能时（ENABLE 位置 1）才能修改 LPTIM_CMP 寄存器。

38.7.7 LPTIM 自动重载寄存器 (LPTIM_ARR)

LPTIM autoreload register

偏移地址：0x18

复位值：0x0000 0001

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	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
ARR[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 **ARR[15:0]**：自动重载值 (Auto reload value)

ARR 为 LPTIM 的自动重载值。

此值必须严格大于 CMP[15:0] 的值。

注意： 必须在 LPTIM 已使能时（ENABLE 位置 1）才能修改 LPTIM_ARR 寄存器。

38.7.8 LPTIM 计数器寄存器 (LPTIM_CNT)

LPTIM counter register

偏移地址：0x1C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	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
CNT[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 **CNT[15:0]**：计数器值 (Counter value)

当 LPTIM 通过异步时钟运行时，读取 LPTIM_CNT 寄存器会返回不可靠的值。因此在这种情况下，必须连续执行读访问两次，并验证两次返回的值是否相同。

应注意的是，为了实现可靠的 LPTIM_CNT 寄存器读访问，必须执行两次连续的读访问并进行比较。当两次连续读访问的值相等时，可认为读访问可靠。

38.7.9 LPTIM1 选项寄存器 (LPTIM1_OR)

LPTIM1 option 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
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.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OR_1	OR_0
														rw	rw

位 31:2 保留，必须保持复位值。

位 1 **OR_1**：选项寄存器位 1 (Option register bit 1)
0： LPTIM1 输入 2 连接到 I/O
1： LPTIM1 输入 2 连接到 COMP2_OUT

位 0 **OR_0**：选择寄存器位 0 (Option register bit 0)
0： LPTIM1 输入 1 连接到 I/O
1： LPTIM1 输入 1 连接到 COMP1_OUT

38.7.10 LPTIM2 选项寄存器 (LPTIM2_OR)

LPTIM2 option 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
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.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OR_1	OR_0
														rw	rw

位 31:2 保留，必须保持复位值。

位 1 **OR_1**：选项寄存器位 1 (Option register bit 1)
0： LPTIM2 输入 1 连接到 I/O
1： LPTIM2 输入 1 连接到 COMP2_OUT

位 0 **OR_0**：选择寄存器位 0 (Option register bit 0)
0： LPTIM2 输入 1 连接到 I/O
1： LPTIM2 输入 1 连接到 COMP1_OUT

注： *如果 OR_1 和 OR_0 均置 1，则 LPTIM2 输入 1 连接到 (COMP1_OUT OR COMP2_OUT)。*

38.7.11 LPTIM 寄存器映射

下表对 LPTIM 寄存器进行了汇总。

表 237. LPTIM 寄存器映射和复位值

偏移	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	LPTIM_ISR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DOWN	UP	ARROK	CMPOK	EXTTRIG	ARRM	CMPM
	Reset value																										0	0	0	0	0	0	0
0x04	LPTIM_ICR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DOWNCF	UPCF	ARROKCF	CMPOKCF	EXTTRIGCF	ARRMCF	CMPMCF
	Reset value																										0	0	0	0	0	0	0
0x08	LPTIM_IER	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DOWNIE	UPIE	ARROKIE	CMPOKIE	EXTTRIGIE	ARRMIE	CMPMIE
	Reset value																										0	0	0	0	0	0	0
0x0C	LPTIM_CFGR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ENC	COUNTMODE	PRELOAD	WAVPOL	WAVE	TIMOUT	TRIGEN	Res.	TRIGSEL[2:0]			Res.	PRESC			Res.	TRGFLT		Res.	CKFLT		CKPOL		CKSEL	
	Reset value								0	0	0	0	0	0	0	0	0	0	0		0	0	0		0	0		0	0	0	0	0	0
0x10	LPTIM_CR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	RSTARE	COUNTRST	CNTSTRT	SNGSTRT	ENABLE
	Reset value																												0	0	0	0	0
0x14	LPTIM_CMP	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CMP[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x18	LPTIM_ARR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ARR[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
0x1C	LPTIM_CNT	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CNT[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x20	LPTIM_OR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OR_1	OR_0
	Reset value																															0	0

有关寄存器边界地址的信息，请参见*第 78 页的第 2.2.2 节*。

39 红外接口 (IRTIM)

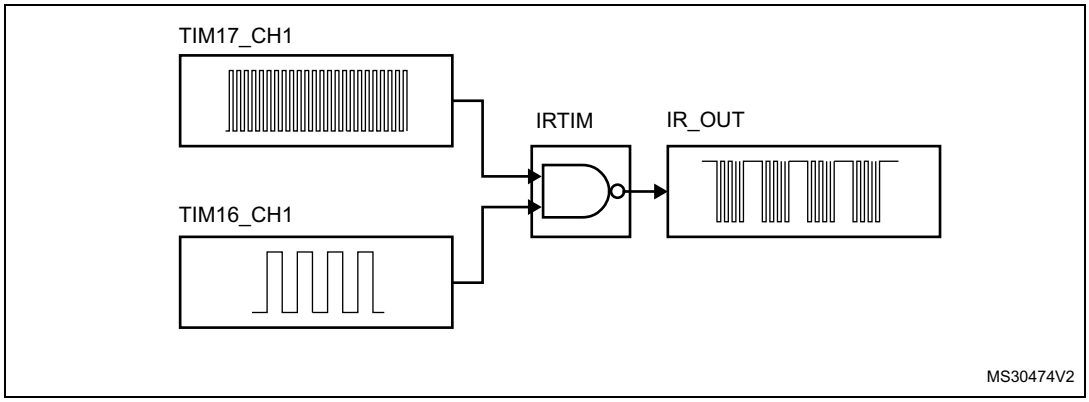
器件具有用于遥控的红外接口 (IRTIM)。此接口与红外 LED 一起用于实现遥控功能。

此接口内部连接到 TIM16 和 TIM17，如*图 415* 所示。

要生成红外遥控信号，必须使能 IR 接口并正确配置 TIM16 通道 1 (TIM16_OC1) 和 TIM17 通道 1 (TIM17_OC1)，以生成正确波形。

通过基本输入捕获模式可以轻松实现红外接收器。

图 415. IRTIM 与 TIM16 和 TIM17 的内部硬件连接



所有标准 IR 脉冲调制模式都可通过编程两个定时器输出比较通道获得。

TIM17 用于生成高频载波信号，而 TIM16 生成调制包络。

在 IR_OUT 引脚上输出红外功能。通过 GPIOx_AFRx 寄存器使能相关复用功能位来激活此功能。

高灌电流 LED 驱动能力（仅在 PB9 引脚上可用）可以通过 SYSCFG_CFGR1 寄存器中的 I2C_PB9_FMP 位激活，并用于吸收直接控制红外 LED 所需的高电流。

40 独立看门狗 (IWDG)

40.1 前言

此器件具有一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。此独立看门狗外设可检测并解决由软件错误导致的故障，并在计数器达到给定的超时值时触发系统复位。

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。

IWDG 最适合应用于需要看门狗作为一个在主程序之外，能够完全独立工作，并且对时间精度要求较低的应用。有关窗口看门狗的详细信息，请参见*第 1340 页的第 41 节*。

40.2 IWDG 主要特性

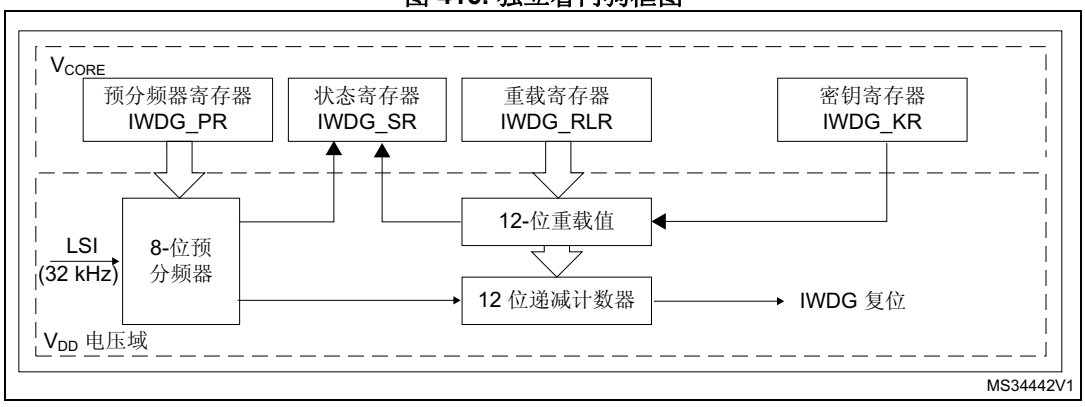
- 自由运行递减计数器
- 时钟由独立 RC 振荡器提供（可在待机和停止模式下运行）
- 复位条件
  - 当递减计数器值小于 0x000 时复位（如果看门狗已激活）
  - 在窗口之外重载递减计数器时复位（如果看门狗已激活）

40.3 IWDG 功能说明

40.3.1 IWDG 框图

图 416 给出了独立看门狗模块的功能框图。

图 416. 独立看门狗框图



1. 看门狗功能由 V_{CORE} 电压域供电，在停止模式和待机模式下仍能工作。

通过向*关键字寄存器 (IWDG_KR)* 中写入值 0x0000 CCCC 来启动独立看门狗时，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）。

任何时候将键值 0x0000 AAAA 写到*关键字寄存器 (IWDG_KR)* 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。

40.3.2 窗口选项

通过在*窗口寄存器 (IWDG_WINR)* 中设置合适的窗口，IWDG 也可以用作窗口看门狗。

当计数器值大于*窗口寄存器 (IWDG_WINR)* 中存储的值时，如果执行重载操作，则会产生复位。

窗口寄存器 (IWDG_WINR) 的默认值为 0x0000 0FFF，因此，如果不更新此默认值，将禁止窗口选项。

窗口值一经更改，便执行重载操作，以便将递减计数器复位为*重载寄存器 (IWDG_RLR)* 值，并方便计算周期数以生成下一次重载。

使能窗口选项时配置 IWDG

1. 通过在*关键字寄存器 (IWDG_KR)* 中写入 0x0000 CCCC 来使能 IWDG。
2. 通过在*关键字寄存器 (IWDG_KR)* 中写入 0x0000 5555 来使能寄存器访问。
3. 通过将*预分频器寄存器 (IWDG_PR)* 编程为 0~7 中的数值来配置 IWDG 预分频器。
4. 对*重载寄存器 (IWDG_RLR)* 进行写操作。
5. 等待寄存器更新 (IWDG_SR = 0x0000 0000)。
6. 对*窗口寄存器 (IWDG_WINR)* 进行写操作。这会自动刷新*重载寄存器 (IWDG_RLR)* 中的计数器值。

注： *当状态寄存器 (IWDG_SR) 设置为 0x0000 0000 时，写入窗口值允许刷新计数器值为 RLR 的值。*

禁止窗口选项时配置 IWDG

不使用窗口选项时，可按以下步骤配置 IWDG：

1. 通过在*关键字寄存器 (IWDG_KR)* 中写入 0x0000 CCCC 来使能 IWDG。
2. 通过在*关键字寄存器 (IWDG_KR)* 中写入 0x0000 5555 来使能寄存器访问。
3. 通过将*预分频器寄存器 (IWDG_PR)* 编程为 0~7 中的数值来配置预分频器。
4. 对*重载寄存器 (IWDG_RLR)* 进行写操作。
5. 等待寄存器更新 (IWDG_SR = 0x0000 0000)。
6. 刷新计数器值为 IWDG_RLR 的值 (IWDG_KR = 0x0000 AAAA)。

40.3.3 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 (IWDG_KR)* 写入相应的值，或者在窗口内部重载了递减计数器，则系统会产生复位。

40.3.4 低功耗冻结

根据 IWDG_STOP 和 IWDG_STBY 选项配置，IWDG 可分别在停止模式和待机模式期间继续计数或停止计数。如果停止模式或待机模式期间 IWDG 保持运行，它可从此模式唤醒器件。更多详细信息，请参见*用户和读保护选项字节*一节。

40.3.5 停止和待机模式下的行为

一旦运行，IWDG 便无法停止。

40.3.6 寄存器访问保护

预分频器寄存器 (IWDG_PR)、*重载寄存器 (IWDG_RLR)* 和 *窗口寄存器 (IWDG_WINR)* 具有写访问保护。若要对其进行修改，用户必须首先对 *关键字寄存器 (IWDG_KR)* 写入代码 0x0000 5555。而写入其他值则会破坏该序列，从而使寄存器访问保护再次生效。这表示重载操作（即写入 0x0000 AAAA）也会启动写保护功能。

状态寄存器指示预分频值、递减计数器重载值或窗口值是否正在被更新。

40.3.7 调试模式

当微控制器进入调试模式时（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。

40.4 IWDG 寄存器

有关寄存器说明中使用的缩写，请参见*第 72 页的第 1.1 节*。

外设寄存器可支持半字（16 位）或字（32 位）访问。

40.4.1 关键字寄存器 (IWDG_KR)

Key register

偏移地址：0x00

复位值：0x0000 0000（待机模式时复位）

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
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)（只能写，读为 0x0000）

必须每隔一段时间便通过软件对这些位写入键值 0xAAAA，否则当计数器计数到 0 时，看门狗会产生复位。

写入键值 0x5555 可使能对 IWDG_PR、IWDG_RLR 和 IWDG_WINR 寄存器的访问（请参见*第 40.3.6 节：寄存器访问保护*）。

写入键值 0xCCCC 可启动看门狗（选中硬件看门狗选项的情况除外）。

40.4.2 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
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.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PR[2:0]		
													rw	rw	rw

位 31:3 保留，必须保持复位值。

位 2:0 **PR[2:0]**：预分频系数 (Prescaler divider)

这些位受写访问保护，请参见*第 40.3.6 节：寄存器访问保护*。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，*状态寄存器 (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：读取该寄存器会返回 V_{DD} 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，只有在状态寄存器 (IWDG_SR) 中的 PVU 位为 0 时，从寄存器读取的值才有效。

40.4.3 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）

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.	Res.	Res.	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*寄存器访问保护*。这个值由软件设置，每次对*关键字寄存器 (IWDG_KR)* 写入值 0xAAAA 时，这个值就会重装载到看门狗计数器中。之后，看门狗计数器便从该装载的值开始递减计数。超时周期由该值和时钟预分频器共同决定。有关超时信息，请参见数据手册。

若要更改重载值，*状态寄存器 (IWDG_SR)* 中的 RVU 位必须为 0。

注： *读取该寄存器会返回 V_{DD} 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，只有在状态寄存器 (IWDG_SR) 中的 RVU 位为 0 时，从寄存器读取的值才有效。*

40.4.4 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
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.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WVU	RVU	PVU
													r	r	r

位 31:3 保留，必须保持复位值。

位 2 **WVU**：看门狗计数器窗口值更新 (Watchdog counter window value update)
可通过硬件将该位置 1 以指示窗口值正在更新。当在 V_{DD} 电压域下完成重载值更新操作后（需要多达 5 个 LSI/预分频器时钟），会通过硬件将该位复位。
窗口值只有在 WVU 位为 0 时才可更新。
此位只有在通用“窗口”= 1 时才生成。

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 LSI/预分频时钟），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 LSI/预分频器时钟），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： *如果应用使用多个重载值、预分频器值或窗口值，则必须等到 RVU 位被复位后才能更改重载值，等到 PVU 位被复位后才能更改预分频器值，而且必须等到 WVU 位被复位后才能更改窗口值。但是，在更新预分频器和/或重载/窗口值之后，则无需等到 RVU、PVU 或 WVU 复位后再继续执行代码（进入低功耗模式时除外）。*

40.4.5 窗口寄存器 (IWDG_WINR)

Window register

偏移地址：0x10

复位值：0x0000 0FFF（待机模式时复位）

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
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.	Res.	Res.	WIN[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 **WIN[11:0]**：看门狗计数器窗口值 (Watchdog counter window value)

这些位受写访问保护，请参见*第 40.3.6 节*，它们包含用于与递减计数器进行比较的窗口值上限。

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 0x0 时必须重载。

若要更改重载值，*状态寄存器 (IWDG_SR)* 中的 WVU 位必须为 0。

注： *读取该寄存器会返回 V_{DD} 电压域的重载值。如果正在对该寄存器执行写操作，则读取的值可能无效。因此，只有在状态寄存器 (IWDG_SR) 中的 WVU 位为 0 时，从寄存器读取的值才有效。*

40.4.6 IWDG 寄存器映射

下表提供了 IWDG 寄存器映射和复位值。

表 238. IWDG 寄存器映射和复位值

偏移	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	**IWDG_KR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	KEY[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x04	**IWDG_PR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PR[2:0]		
	Reset value																														0	0	0
0x08	**IWDG_RLR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RL[11:0]											
	Reset value																					1	1	1	1	1	1	1	1	1	1	1	1
0x0C	**IWDG_SR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WVU	RVU	PVU
	Reset value																														0	0	0
0x10	**IWDG_WINR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WIN[11:0]											
	Reset value																					1	1	1	1	1	1	1	1	1	1	1	1

有关寄存器边界地址的信息，请参见*第 2.2.2 节：存储器映射和寄存器边界地址*。

41 系统窗口看门狗 (WWDG)

41.1 简介

系统窗口看门狗 (WWDG) 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。除非程序在 T6 位变成 0 前刷新递减计数器的值，否则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。

WWDG 时钟由 APB 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。

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。

41.2 WWDG 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 418*）
- 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）

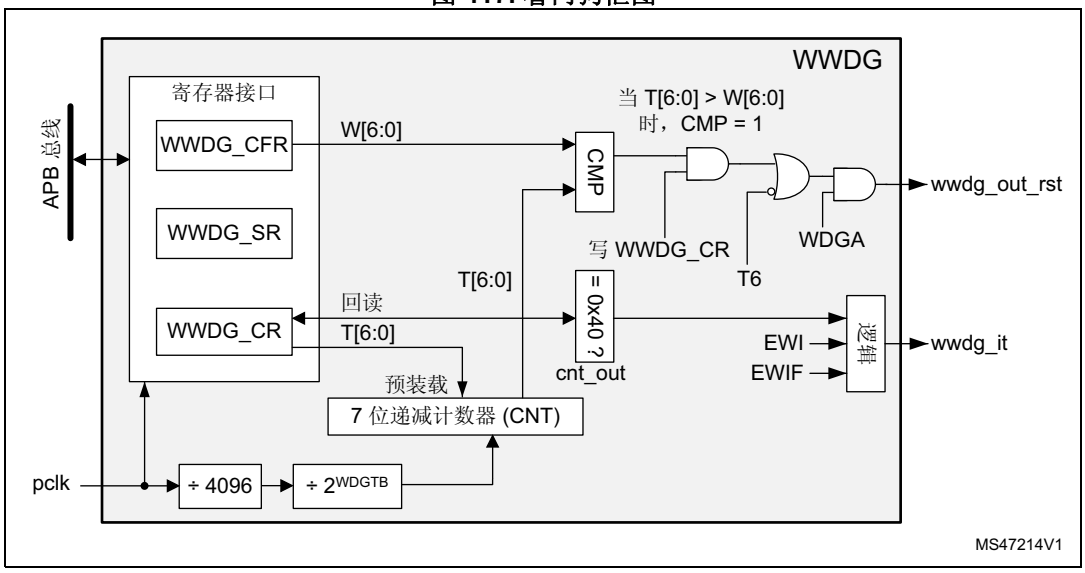
41.3 WWDG 功能说明

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 递减到 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。

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且高于 0x3F 时，才能执行此操作。要存储到 WWDG_CR 寄存器中的值必须介于 0xFF 和 0xC0 之间。

请参见*图 417* 了解 WWDG 框图。

图 417. 看门狗框图



41.3.1 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。

41.3.2 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，预分频器的状态是未知的（请参见 *图 418*）。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 418* 介绍了窗口看门狗的工作过程。

注： *可使用 T6 位产生软件复位（将 WDGA 位置 1 并将 T6 位清零）。*

41.3.3 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。

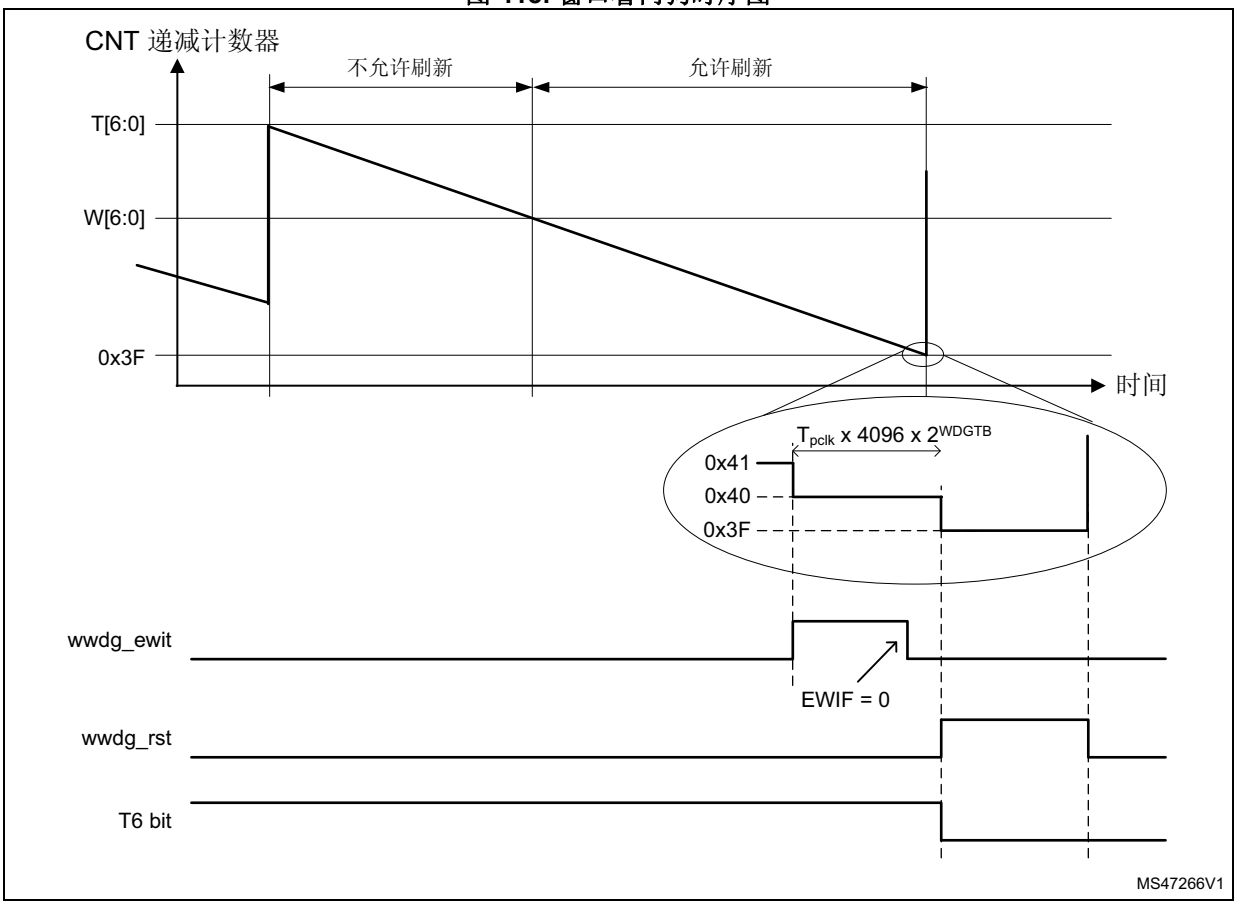
注： *当由于在更高优先级任务中有系统锁定而无法使用 EWI 中断时，最终会产生 WWDG 复位。*

41.3.4 如何设置看门狗超时

使用*图 418* 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 418. 窗口看门狗时序图



超时值的计算公式如下：

$$t_{WWDG} = t_{PCLK} \times 4096 \times 2^{WDGTB[1:0]} \times (T[5:0] + 1) \quad (ms)$$

其中：

t_{WWDG}：WWDG 超时

t_{PCLK}：APB 时钟周期，以 ms 为测量单位

4096：对应于内部分频器的值

例如，假设 APB 频率等于 48 MHz，将 WDGTB[1:0] 设置为 3 并将 T[5:0] 设置为 63：

$$t_{WWDG} = (1/48000) \times 4096 \times 2^{3} \times (63 + 1) = 43.69ms$$

有关 t_{WWDG} 的最小值和最大值，请参见数据手册。

41.3.5 调试模式

当微控制器进入调试模式时（内核停止），WWDG 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。有关详细信息，请参见*第 51.16.2 节：对定时器、RTC、看门狗、bxCAN 和 I2C 的调试支持*。

41.4 WWDG 寄存器

有关寄存器说明中使用的缩写，请参见*第 72 页的第 1.1 节*。

外设寄存器可支持半字（16 位）或字（32 位）访问。

41.4.1 控制寄存器 (WWDG_CR)

Control register

偏移地址：0x000

复位值：0x0000 007F

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.	Res.	Res.	Res.	Res.	Res.	Res.	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）(7-bit counter (MSB to LSB))
这些位用来存储看门狗计数器的值，每隔 (4096 x $2^{WDGTB[1:0]}$) PCLK 个周期递减一次。当它从 0x40 递减到 0x3F（T6 清零）时会产生复位。

41.4.2 配置寄存器 (WWDG_CFR)

Configuration register

偏移地址：0x004

复位值：0x0000 007F

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.	Res.	Res.	Res.	Res.	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 (PCLK div 4096) 分频器 1
01：CK 计数器时钟 (PCLK div 4096) 分频器 2
10：CK 计数器时钟 (PCLK div 4096) 分频器 4
11：CK 计数器时钟 (PCLK 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。

41.4.3 状态寄存器 (WWDG_SR)

Status register

偏移地址：0x008

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.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	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”无影响。如果不使能中断，此位也会被置 1。

41.4.4 WWDG 寄存器映射

下表提供了 WWDG 寄存器映射和复位值。

表 239. 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
0x000	**WWDG_CR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WDGA	T[6:0]						
	Reset value																									0	1	1	1	1	1	1	1
0x004	**WWDG_CFR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	EWI	WDGTB1	WDGTB0	W[6:0]						
	Reset value																							0	0	0	1	1	1	1	1	1	1
0x008	**WWDG_SR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	EWIF
	Reset value																																0

有关寄存器边界地址的信息，请参见*第 2.2.2 节：存储器映射和寄存器边界地址*。

42 实时时钟 (RTC)

42.1 简介

实时时钟 (RTC) 提供用于管理所有低功耗模式的自动唤醒单元。

实时时钟 (RTC) 是一个独立的 BCD 定时器/计数器。RTC 提供具有可编程闹钟中断功能的日历时钟/日历。

RTC 还包含具有中断功能的周期性可编程唤醒标志。

两个 32 位寄存器包含 BCD 格式的秒、分钟、小时（12 或 24 小时制）、星期几、日期、月份和年份。此外，还可提供二进制格式的亚秒值。

系统可以自动将月份的天数补偿为 28、29（闰年）、30 和 31 天。并且还可以进行夏令时补偿。

其他 32 位寄存器还包含可编程的闹钟亚秒、秒、分钟、小时、星期几和日期。

此外，还可以使用数字校准功能对晶振精度的偏差进行补偿。

Backup 域复位后，所有 RTC 寄存器都会受到保护，以防止可能的非正常写访问。

无论器件状态如何（运行模式、低功耗模式或处于复位状态），只要电源电压保持在工作范围内，RTC 便不会停止工作。

42.2 RTC 主要特性

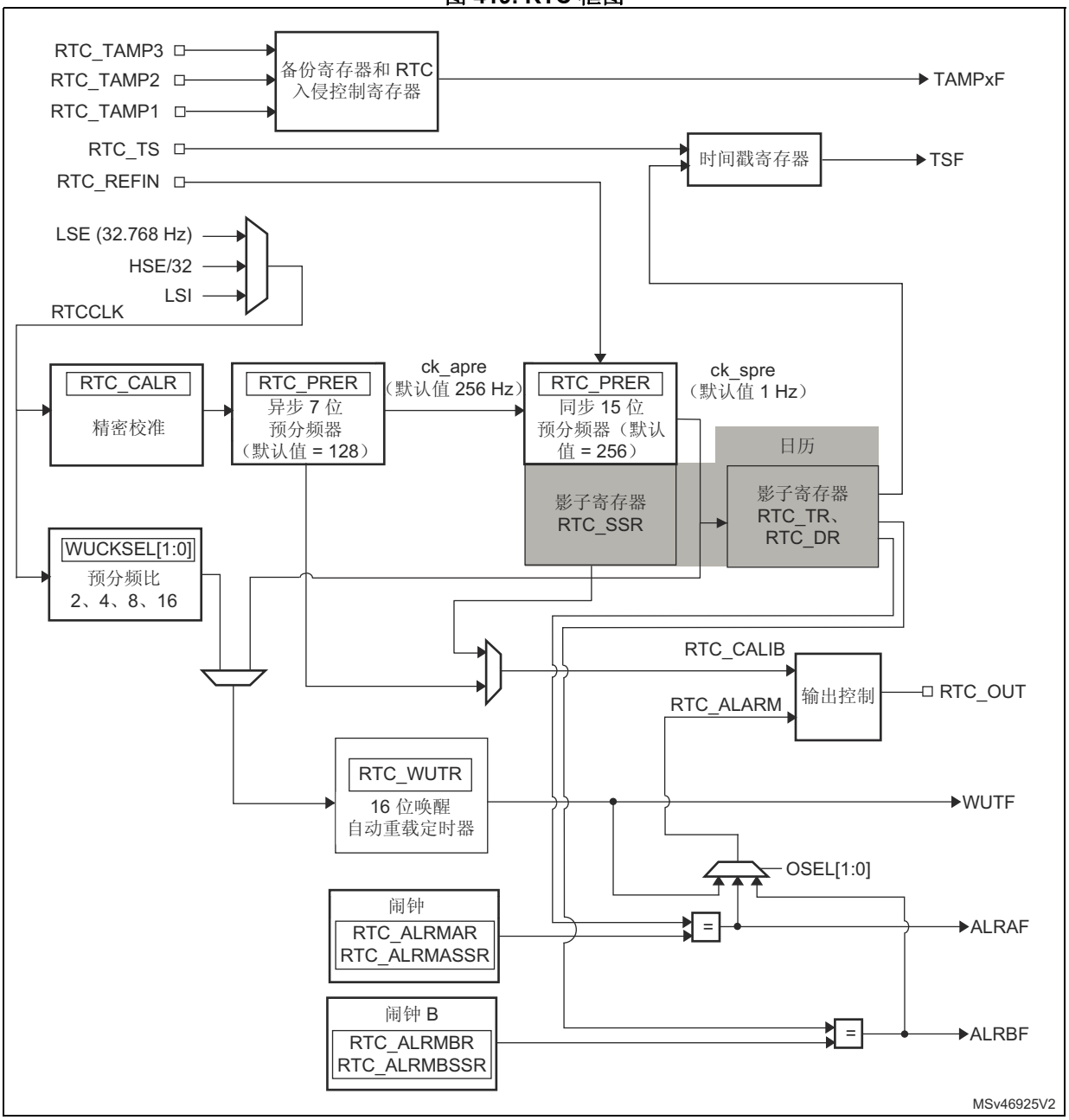
RTC 单元的主要特性如下（参见*图 419：RTC 框图*）：

- 包含亚秒、秒、分钟、小时（12/24 小时制）、星期几、日期、月份和年份的日历。
- 软件可编程的夏令时补偿。
- 具有中断功能的可编程闹钟。可通过任意日历字段的组合触发闹钟。
- 自动唤醒单元，可周期性地生成标志以触发自动唤醒中断。
- 参考时钟检测：可使用更加精确的第二时钟源（50 Hz 或 60 Hz）来提高日历的精确度。
- 利用亚秒级移位特性与外部时钟实现精确同步。
- 数字校准电路（周期性计数器调整）：精度为 0.95 ppm，在数秒钟的校准窗口中获得。
- 用于事件保存的时间戳功能。
- 带可配置过滤器和内部上拉的入侵检测事件。
- 可屏蔽中断/事件：
  - 闹钟 A
  - 闹钟 B
  - 唤醒中断
  - 时间戳
  - 入侵检测
- 32 个备份寄存器。

42.3 RTC 功能说明

42.3.1 RTC 框图

图 419. RTC 框图



RTC 包含：

- 两个闹钟
- 三个来自 I/O 的篡改事件
  - 入侵检测擦除备份寄存器
- 来自 I/O 的一个时间戳事件
- 篡改事件检测可生成一个时间戳事件
- 切换到 V_{BAT} 时会生成时间戳
- 32 个 32 位备份寄存器
  - 备份寄存器 (RTC_BKPxR) 在 RTC 域中实现，可在 VDD 电源关闭时通过 VBAT 保持上电状态。
- 输出功能：RTC_OUT，可选择以下两个输出之一：
  - RTC_CALIB：512 Hz 或 1 Hz 时钟输出（LSE 频率为 32.768 kHz）。可通过将 RTC_CR 寄存器中的 COE 位置 1 来使能此输出。
  - RTC_ALARM：可通过配置 RTC_CR 寄存器中的 OSEL[1:0] 位使能此输出，可选择闹钟 A、闹钟 B 或唤醒输出。
- 输入功能：
  - RTC_TS：时间戳事件
  - RTC_TAMP1：tamper1 事件检测
  - RTC_TAMP2：tamper2 事件检测
  - RTC_TAMP3：tamper3 事件检测
  - RTC_REFIN：50 Hz 或 60 Hz 参考时钟输入

42.3.2 RTC 控制的 GPIO

RTC_OUT、RTC_TS 和 RTC_TAMP1 映射到同一引脚 (PC13)。无论 PC13 的 GPIO 配置如何，PC13 引脚的配置都由 RTC 控制，但 RTC_ALARM 输出开漏模式除外。在此特殊情况下，GPIO 必须如下配置为输入。映射在 PC13 上的 RTC 功能可用于所有低功耗模式和 VBAT 模式。

输出机制遵循*表 240* 中所示的优先级顺序。

表 240. RTC 引脚 PC13 配置(1)

PC13 引脚配置和功能	OSEL[1:0] 位（RTC_ALARM 输出使能）	COE 位（RTC_CALIB 输出使能）	RTC_OUT_RMP 位	RTC_ALARM_TYPE 位	TAMP1E 位（RTC_TAMP1 输入使能）	TSE位（RTC_TS 输入使能）
RTC_ALARM 输出 OD	01 或 10 或 11	无关	0	0	无关	无关
			1			
RTC_ALARM 输出 PP	01 或 10 或 11	无关	0	1	无关	无关
			1			
RTC_CALIB 输出 PP	00	1	0	无关	无关	无关

表 240. RTC 引脚 PC13 配置(1)（续）

PC13 引脚配置和功能	OSEL[1:0] 位（RTC_ALARM 输出使能）	COE 位（RTC_CALIB 输出使能）	RTC_OUT_RMP 位	RTC_ALARM_TYPE 位	TAMP1E 位（RTC_TAMP1 输入使能）	TSE 位（RTC_TS 输入使能）
RTC_TAMP1 输入浮空	00	0	无关	无关	1	0
	00	1	1			
	01 或 10 或 11	0				
RTC_TS 和 RTC_TAMP1 输入浮空	00	0	无关	无关	1	1
	00	1	1			
	01 或 10 或 11	0				
RTC_TS 输入浮空	00	0	无关	无关	0	1
	00	1	1			
	01 或 10 或 11	0				
唤醒引脚或标准 GPIO	00	0	无关	无关	0	0
	00	1	1			
	01 或 10 或 11	0				

1. OD：开漏；PP：推挽。

此外，可借助 RTC_OUT_RMP 位将 RTC_OUT 重映射到 PB2 引脚上。在这种情况下，必须将 PB2 GPIO 寄存器配置为类型正确的复用功能。重映射功能如*表 241* 所示。

表 241. RTC_OUT 映射

OSEL[1:0] 位（RTC_ALARM 输出使能）	COE 位（RTC_CALIB 输出使能）	RTC_OUT_RMP 位	RTC_OUT 映射到 PC13	RTC_OUT 映射到 PB2
00	0	0	-	-
00	1		RTC_CALIB	-
01 或 10 或 11	无关		RTC_Alarm	-
00	0	1	-	-
00	1		-	RTC_CALIB
01 或 10 或 11	0		-	RTC_Alarm
01 或 10 或 11	1		RTC_Alarm	RTC_CALIB

下表汇总了所有模式下的 RTC 引脚和功能。

表 242. 各个模式下的 RTC 功能

引脚	RTC 功能	待机/关断模式以外的所有低功耗模式下的功能	待机/关断模式下的功能性	V_{BAT} 模式下的功能性
PC13	RTC_TAMP1 RTC_TS RTC_OUT	是	是	是
PA0	RTC_TAMP2	是	是	是
PE6	RTC_TAMP3	是	是	是
PB2	RTC_OUT	是	否	否
PB15	RTC_REFIN	是	否	否

42.3.3 时钟和预分频器

RTC 时钟源 (RTCCLK) 通过时钟控制器从 LSE 时钟、LSI 振荡器时钟以及 HSE 时钟三者中选择。有关 RTC 时钟源配置的更多信息，请参见*第 6 节：复位和时钟控制 (RCC)*。

可编程的预分频器阶段可生成 1 Hz 的时钟，用于更新日历。为最大程度地降低功耗，预分频器分为 2 个可编程的预分频器（参见*图 419：RTC 框图*）：

- 一个通过 RTC_PRER 寄存器的 PREDIV_A 位配置的 7 位异步预分频器。
- 一个通过 RTC_PRER 寄存器的 PREDIV_S 位配置的 15 位同步预分频器。

注： *使用两个预分频器时，推荐将异步预分频器配置为较高的值，以最大程度降低功耗。*

要使用频率为 32.768 kHz 的 LSE 获得频率为 1 Hz 的内部时钟 (ck_spre)，需要将异步预分频系数设置为 128，并将同步预分频系数设置为 256。

分频系数的最小值为 1，最大值为 2^{22}。

这对应于约为 4 MHz 的最大输入频率。

f_{ck_apre} 可根据以下公式得出：

$$f_{CK_APRE} = \frac{f_{RTCCLK}}{PREDIV_A + 1}$$

ck_apre 时钟用于为二进制 RTC_SSR 亚秒递减计数器提供时钟。当该计数器计数到 0 时，会使用 PREDIV_S 的内容重载 RTC_SSR。

f_{ck_spre} 可根据以下公式得出：

$$f_{CK_SPRE} = \frac{f_{RTCCLK}}{(PREDIV_S + 1) \times (PREDIV_A + 1)}$$

ck_spre 时钟既可以用于更新日历，也可以用作 16 位唤醒自动重载定时器的时基。为获得较短的超时周期，还可以将 16 位唤醒自动重载定时器与经可编程的 4 位异步预分频器分频的 RTCCLK 一同运行（有关详细信息，请参见*第 42.3.6 节：周期性自动唤醒*）。

42.3.4 实时时钟和日历

RTC 日历时间和日期寄存器可通过与 PCLK（APB 时钟）同步的影子寄存器来访问。这些时间和日期寄存器也可以直接访问，这样可避免等待同步的持续时间。

- RTC_SSR 对应于亚秒
- RTC_TR 对应于时间
- RTC_DR 对应于日期

每隔两个 RTCCLK 周期，便将当前日历值复制到影子寄存器，并将 RTC_ISR 寄存器的 RSF 位置 1（请参见*第 42.6.4 节：RTC 初始化和状态寄存器 (RTC_ISR)*）。在停机和待机模式下不会执行复制操作。退出这两种模式时，影子寄存器会在最长 2 个 RTCCLK 周期后进行更新。

当应用读取日历寄存器时，它会访问影子寄存器的内容。也可以通过将 RTC_CR 寄存器的 BYPSHAD 控制位置 1 来直接访问日历寄存器。默认情况下，该位被清零，用户访问影子寄存器。

在 BYPSHAD=0 模式下读取 RTC_SSR、RTC_TR 或 RTC_DR 寄存器时，APB 时钟频率 (f_{APB}) 必须至少为 RTC 时钟频率 (f_{RTCCLK}) 的 7 倍。

影子寄存器通过系统复位来复位。

42.3.5 可编程闹钟

RTC 单元提供两个可编程闹钟，即闹钟 A 和闹钟 B。以下说明针对闹钟 A，但同样适用于闹钟 B。

可通过 RTC_CR 寄存器中的 ALRAE 位来使能可编程闹钟功能。如果日历亚秒、秒、分钟、小时、日期或日与闹钟寄存器 RTC_ALRMASSR 和 RTC_ALRMAR 中编程的值相匹配，则 ALRAF 标志会被置为 1。可通过 RTC_ALRMAR 寄存器的 MSKx 位以及 RTC_ALRMASSR 寄存器的 MASKSSx 位单独选择各日历字段。可通过 RTC_CR 寄存器中的 ALRAIE 位来使能闹钟中断。

注意： 如果选择秒字段（RTC_ALRMAR 中的 MSK1 位复位），则 RTC_PRER 寄存器中设置的同步预分频器分频系数必须至少为 3，才能确保闹钟正确地运行。

闹钟 A 和闹钟 B（如果已通过 RTC_CR 寄存器中的位 OSEL[1:0] 使能）可连接到 RTC_ALARM 输出。可通过 RTC_CR 寄存器的 POL 位配置 RTC_ALARM 输出极性。

42.3.6 周期性自动唤醒

周期性唤醒标志由 16 位可编程自动重载递减计数器生成。唤醒定时器范围可扩展至 17 位。

可通过 RTC_CR 寄存器中的 WUTE 位来使能此唤醒功能。

唤醒定时器的时钟输入可以是：

- 2、4、8 或 16 分频的 RTC 时钟 (RTCCLK)。
  当 RTCCLK 为 LSE (32.768 kHz) 时，可配置的唤醒中断周期介于 122 µs 和 32 s 之间，且分辨率低至 61 µs。
- ck_spre（通常为 1 Hz 内部时钟）
  当 ck_spre 频率为 1 Hz 时，可得到的唤醒时间为 1s 到 36h 左右，分辨率为 1 秒。这一较大的可编程时间范围分为两部分：
  - WUCKSEL [2:1] = 10 时为 1s 到 18h
  - WUCKSEL [2:1] = 11 时约为 18h 到 36h。在后一种情况下，会将 2^{16} 添加到 16 位计数器当前值。完成初始化序列后（请参见*第 1353 页的编程唤醒定时器*），定时器开始递减计数。在低功耗模式下使能唤醒功能时，递减计数保持有效。此外，当计数器计数到 0 时，RTC_ISR 寄存器的 WUTF 标志会置 1，并且唤醒寄存器会使用其重载值（RTC_WUTR 寄存器值）自动重载。

之后必须用软件清零 WUTF 标志。

通过将 RTC_CR 寄存器中的 WUTIE 位置 1 来使能周期性唤醒中断时，它会使器件退出低功耗模式。

如果已通过 RTC_CR 寄存器的位 OSEL[1:0] 使能周期性唤醒标志，则该标志可连接到 RTC_ALARM 输出。可通过 RTC_CR 寄存器的 POL 位配置 RTC_ALARM 输出极性。

系统复位以及低功耗模式（睡眠、停机和待机）对唤醒定时器没有任何影响。

42.3.7 RTC 初始化和配置

RTC 寄存器访问

RTC 寄存器为 32 位寄存器。除了当 BYPSHAD=0 时对日历影子寄存器执行的读访问之外，APB 接口会在访问 RTC 寄存器时引入 2 个等待周期。

RTC 寄存器写保护

系统复位后，可通过对 PWR_CR1 寄存器中的 DBP 位清零来保护 RTC 寄存器以防止非正常的写访问。必须将 DBP 位置 1 才能使能 RTC 寄存器的写访问。

Backup 域复位后，所有 RTC 寄存器均受到写保护。通过向写保护寄存器 (RTC_WPR) 写入一个密钥来使能对 RTC 寄存器的写操作。

要解锁所有 RTC 寄存器（RTC_TAMPCR、RTC_BKPxR、RTC_OR 和 RTC_ISR[13:8] 除外）的写保护，需要执行以下步骤：

1. 将“0xCA”写入 RTC_WPR 寄存器。
2. 将“0x53”写入 RTC_WPR 寄存器。

写入一个错误的关键字会再次激活写保护。

保护机制不受系统复位影响。

日历初始化和配置

要编程包括时间格式和预分频器配置在内的初始时间和日期日历值，需按照以下顺序操作：

1. 将 RTC_ISR 寄存器中的 INIT 位置 1 以进入初始化模式。在此模式下，日历计数器将停止工作并且其值可更新。
2. 轮询 RTC_ISR 寄存器中的 INITF 位。当 INITF 置 1 时进入初始化阶段模式。大约需要 2 个 RTCCLK 时钟周期（由于时钟同步）。
3. 要为日历计数器生成 1 Hz 时钟，应编程 RTC_PRER 寄存器中的两个预分频系数。
4. 在影子寄存器（RTC_TR 和 RTC_DR）中加载初始时间和日期值，然后通过 RTC_CR 寄存器中的 FMT 位配置时间格式（12 或 24 小时制）。
5. 通过清零 INIT 位退出初始化模式。随后，自动加载实际日历计数器值，在 4 个 RTCCLK 时钟周期后重新开始计数。

当初始化序列完成之后，日历开始计数。

注： *系统复位后，应用可读取 RTC_ISR 寄存器中的 INITS 标志，以检查日历是否已初始化。如果该标志为 0，表明自系统复位以来，日历还尚未初始化过，其年份字段一直还保持着 Backup 域复位默认值 (0x00)。*

要在初始化之后读取日历，必须首先用软件检查 RTC_ISR 寄存器的 RSF 标志是否置 1。

夏令时

可通过 RTC_CR 寄存器的 SUB1H、ADD1H 和 BKP 位管理夏令时。

利用 SUB1H 或 ADD1H，软件只需单次操作便可在日历中减去或增加一个小时，无需执行整个初始化步骤。

此外，软件还可以使用 BKP 位来记录是否曾经执行过此操作。

编程闹钟

要对可编程的闹钟进行编程或更新，必须执行类似的步骤。以下步骤针对闹钟 A，但同样适用于闹钟 B。

1. 将 RTC_CR 中的 ALRAE 位清零以禁止闹钟 A。
2. 编程闹钟 A 寄存器 (RTC_ALRMASSR/RTC_ALRMAR)。
3. 将 RTC_CR 寄存器中的 ALRAE 位置 1 以再次使能闹钟 A。

注： *由于时钟同步缘故，RTC_CR 寄存器的每次更改需要在大约 2 个 RTCCLK 时钟周期后执行。*

编程唤醒定时器

要配置或更改唤醒定时器的自动重载值（RTC_WUTR 中的 WUT[15:0]），需要按照以下顺序操作：

1. 清零 RTC_CR 中的 WUTE 以禁止唤醒定时器。
2. 轮询 RTC_ISR 中的 WUTWF，直到该位置 1，以确保可以访问唤醒自动重载定时器和 WUCKSEL[2:0] 位。大约需要 2 个 RTCCLK 时钟周期（由于时钟同步）。
3. 编程唤醒自动重载值 WUT[15:0]，并选择唤醒时钟（RTC_CR 中的 WUCKSEL[2:0] 位）。将 RTC_CR 寄存器中的 WUTE 位置 1 以再次使能定时器。唤醒定时器重新开始递减计数。由于时钟同步的缘故，在 WUTE 清零后，WUTWF 位也会清零，但需要花费多达 2 个 RTCCLK 时钟周期。

42.3.8 读取日历

当 RTC_CR 寄存器中的 BYPSHAD 控制位清零时

要正确读取 RTC 日历寄存器（RTC_SSR、RTC_TR 和 RTC_DR），APB 时钟频率 (f_{PCLK}) 必须等于或大于 RTC 时钟频率 (f_{RTCCLK}) 的七倍。这可以确保同步机制的安全性。

如果 APB 时钟频率低于 RTC 时钟频率的七倍，则软件必须分两次读取日历时间寄存器和日期寄存器。这样，当两次读取的 RTC_TR 结果相同时，才能确保数据正确。否则必须执行第三次读访问。任何情况下，APB 的时钟频率都不能低于 RTC 的时钟频率。

每次将日历寄存器中的值复制到 RTC_SSR、RTC_TR 和 RTC_DR 影子寄存器时，RTC_ISR 寄存器中的 RSF 位都会置 1。每两个 RTCCLK 周期执行一次复制。为确保这 3 个值来自同一时刻点，读取 RTC_SSR 或 RTC_TR 时会锁定高阶日历影子寄存器中的值，直到读取 RTC_DR。为避免软件对日历执行读访问的时间间隔小于 2 个 RTCCLK 周期：第一次读取日历之后必须通过软件将 RSF 清零，并且软件必须等待到 RSF 置 1 之后才可再次读取 RTC_SSR、RTC_TR 和 RTC_DR 寄存器。

从低功耗模式（停机模式或待机模式）唤醒之后，必须通过软件将 RSF 清零。之后，软件必须等待至 RSF 再次置 1 之后才可以读取 RTC_SSR、RTC_TR 和 RTC_DR 寄存器。

RSF 位必须在唤醒之后而不是进入低功耗模式之前进行清零。

系统复位之后，软件必须等待至 RSF 置 1 之后才可以读取 RTC_SSR、RTC_TR 和 RTC_DR 寄存器。实际上，系统复位会将影子寄存器复位为其默认值。

初始化之后（请参见*第 1353 页的日历初始化和配置*），软件必须等待至 RSF 置 1 之后才可以读取 RTC_SSR、RTC_TR 和 RTC_DR 寄存器。

同步之后（请参见*第 42.3.10 节：RTC 同步*），软件必须等待至 RSF 置 1 之后才可以读取 RTC_SSR、RTC_TR 和 RTC_DR 寄存器。

当 RTC_CR 寄存器中的 BYPSHAD 控制位置 1 时（旁路影子寄存器）

读取日历寄存器时会直接从日历计数器获取值，这样便无需等待至 RSF 位置 1。这对于从低功耗模式（停机模式或待机模式）退出后的情况特别有用，因为影子寄存器在这些模式下不更新。

当 BYPSHAD 位置 1 时，如果在对寄存器的两次读访问之间出现 RTCCLK 沿，则不同寄存器的结果彼此可能不一致。此外，如果在读操作期间出现 RTCCLK 沿，则可能导致其中一个寄存器的值不正确。软件必须分两次读取所有寄存器，然后将两次结果加以比较来确认数据是否一致和正确。此外，软件也可以只比较两次读取日历寄存器得到的结果的最低位。

注： *当 BYPSHAD=1 时，读取日历寄存器的指令需要一个额外的 APB 周期才能完成。*

42.3.9 复位 RTC

日历影子寄存器（RTC_SSR、RTC_TR 和 RTC_DR）以及 RTC 状态寄存器 (RTC_ISR) 的某些位通过所有可用的系统复位源复位为各自的默认值。

相反，以下寄存器则通过 Backup 域复位来复位为各自的默认值并且不受系统复位的影响：RTC 当前日历寄存器、RTC 控制寄存器 (RTC_CR)、预分频器寄存器 (RTC_PRER)、RTC 校准寄存器 (RTC_CALR)、RTC 移位寄存器 (RTC_SHIFTR)、RTC 时间戳寄存器（RTC_TSSSR、RTC_TSTR 和 RTC_TSDR）、RTC 入侵配置寄存器 (RTC_TAMPCR)、RTC 备份寄存器 (RTC_BKPxR)、唤醒定时器寄存器 (RTC_WUTR)、闹钟 A 和闹钟 B 寄存器（RTC_ALRMASSR/RTC_ALRMAR 和 RTC_ALRMBSSR/RTC_ALRMBR），以及选项寄存器 (RTC_OR)。

此外，当由 LSE 提供时钟时，如果复位源并非 Backup 域复位源（有关不受系统复位影响的 RTC 时钟源列表的详细信息，请参见“复位和时钟控制器”的“RTC 时钟”部分），则 RTC 将在系统复位时保持运行状态。发生 Backup 域复位时，RTC 会停止工作，并且所有 RTC 寄存器都会设置为各自的复位值。

42.3.10 RTC 同步

RTC 可与高精度的远程时钟同步。在读取亚秒字段后（RTC_SSR 或 RTC_TSSSR），即可计算远程时钟的时间与 RTC 之间的精准偏差。之后，可使用 RTC_SHIFTR 对 RTC 的时钟进行零点几秒的“平移”，经过调整后可消除此偏差。

RTC_SSR 包含同步预分频器计数器的值。这样，便可计算分辨率低至 1/(PREDIV_S + 1) 秒的 RTC 的准确时间。因此，可通过增大同步预分频器的值 (PREDIV_S[14:0]) 来提高分辨率。将 PREDIV_S 设置为 0x7FFF 时，可得到允许的最大分辨率（30.52 µs，时钟频率为 32768 Hz）。

但是，提高 PREDIV_S 意味着必须降低 PREDIV_A 才能将同步预分频器的输出维持在 1 Hz。这样，异步预分频器的输出频率会增大，RTC 的动态功耗也会相应增加。

可以使用 RTC 平移控制寄存器 (RTC_SHIFTR) 对 RTC 进行微调。可以用大小为 1/(PREDIV_S + 1) 秒的分辨率对 RTC_SHIFTR 进行写操作，将时钟平移（延迟或提前）最长 1 秒。在这种平移操作中，会将 SUBFS[14:0] 值加到同步预分频器计数器 SS[15:0] 中：这将使时钟产生延迟。如果同时将 ADD1S 位置 1，则会增加一秒，与此同时减去的时间为零点几秒，因此将使时钟提前。

注意： 初始化平移操作前，用户必须检查确认 SS[15] = 0，以确保不会发生上溢。

对 RTC_SHIFTR 寄存器执行写操作以启动平移操作时，硬件会将 SHPF 标志置 1 以指示平移操作挂起。完成平移操作时，硬件会将该位清零。

注意： 该同步功能与参考时钟检测功能不兼容：当 REFCKON=1 时，固件不能对 RTC_SHIFTR 执行写操作。

42.3.11 RTC 参考时钟检测

RTC 日历更新可与参考时钟 RTC_REFIN（通常为市电频率，50 Hz 或 60 Hz）同步。RTC_REFIN 参考时钟的精度应高于 32.768 kHz LSE 时钟。使能 RTC_REFIN 检测时（将 RTC_CR 的 REFCKON 位置 1），日历仍由 LSE 提供时钟，而 RTC_REFIN 用于补偿不准确的日历更新频率 (1 Hz)。

每个 1 Hz 时钟边沿都与最近的 RTC_REFIN 时钟边沿进行比较（如果在给定的时间窗口内发现一个边沿）。在大多数情况下，两个时钟边沿恰好对齐。当 1 Hz 时钟由于 LSE 时钟不精确而发生偏离时，RTC 会稍微偏移 1 Hz 时钟，以便后续的 1 Hz 时钟边沿能够对齐。利用这种机制，可使日历像参考时钟一样精确。

RTC 使用 32.768 kHz 石英产生的 256 Hz 时钟 (ck_apre) 检测是否存在参考时钟源。大约在日历每次更新时（每 1 秒钟），便会在时间窗口期间执行一次检测。检测到第一个参考时钟边沿时，该窗口等于 7 个 ck_apre 周期。随后的日历更新使用长度为 3 个 ck_apre 周期的较小窗口。

每次在窗口中检测到参考时钟时，都会行强制输出 ck_spre 时钟的同步预分频器进行重载。当参考时钟与 1 Hz 时钟对齐时，此操作不起作用，因为预分频器会在同一时刻重载。当时钟不对齐时，重载操作会微调后续的 1 Hz 时钟边沿，使其与参考时钟对齐。

如果参考时钟停止（在 3 个 ck_apre 窗口内未出现参考时钟边沿），日历将仅根据 LSE 时钟进行连续更新。RTC 随后使用 ck_spre 边沿上居中的大检测窗口（7 个 ck_apre 周期）等待参考时钟。

使能 RTC_REFIN 检测后，必须将 PREDIV_A 和 PREDIV_S 设置为各自的默认值：

- PREDIV_A = 0x007F
- PREVID_S = 0x00FF

注： *RTC_REFIN 时钟检测在待机模式下不可用。*

42.3.12 RTC 精密数字校准

RTC 频率可采用约 0.954 ppm 的分辨率进行数字校准，校准范围为 -487.1 ppm 到 +488.5 ppm。使用一系列微调（增加和/或减少单独的 RTCCLK 脉冲）进行频率校正。这些微调的分布非常均匀，因此 RTC 的校准效果相当好，即使在短时间内持续观察也是如此。

当输入频率为 32768 Hz 时，精密数字校准的周期约为 2^{20} 个 RTCCLK 脉冲或 32 秒。此周期由一个通过 RTCCLK 提供时钟信号的 20 位计数器 cal_cnt[19:0] 维持。

精密数字校准寄存器 (RTC_CALR) 可指定 32 秒周期内要减少的 RTCCLK 时钟周期数：

- 将位 CALM[0] 置 1 时，32 秒周期内将只减少 1 个脉冲。
- 将 CALM[1] 置 1 时，将减少 2 个周期。
- 将 CALM[2] 置 1 时，将减少 4 个周期。
- 依此类推，将 CALM[8] 置 1 时，将减少 256 个时钟。

注： *CALM[8:0] (RTC_CALR) 可指定 32 秒周期内要减少的 RTCCLK 脉冲数。将位 CALM[0] 置 1 时，32 秒周期内将只减少 1 个脉冲（当 cal_cnt[19:0] = 0x80000 时）；将 CALM[1] 置 1 时，将减少 2 个周期（当 cal_cnt = 0x40000 和 0xC0000 时）；将 CALM[2] 置 1 时，将减少 4 个周期（当 cal_cnt = 0x20000/0x60000/0xA0000/0xE0000 时）；依此类推，将 CALM[8] 置 1 时，将减少 256 个时钟（当 cal_cnt = 0xXX800 时）。*

使用适当分辨率时，CALM 可使 RTC 频率减少最多 487.1 ppm，而 CALP 可用于使频率增加 488.5 ppm。将 CALP 置“1”，可每隔 2^{11} 个 RTCCLK 周期有效插入一个额外的 RTCCLK 脉冲，这意味着每 32 秒周期可增加 512 个时钟。

与 CALM 和 CALP 配合使用时，可在 32 秒周期内增加一个范围为 -511 到 +512 RTCCLK 周期的偏差，对应的校准范围为 -487.1 ppm 到 +488.5 ppm，分辨率约为 0.954 ppm。

若输入频率 (F_{RTCCLK}) 已知，可通过以下公式计算有效校准频率 (F_{CAL})：

$$F_{CAL} = F_{RTCCLK} \times [1 + (CALP \times 512 - CALM) / (2^{20} + CALM - CALP \times 512)]$$

PREDIV_A<3 条件下的校准

当异步预分频器值（RTC_PRER 寄存器中的 PREDIV_A 位）小于 3 时，不能将 CALP 位置 1。如果 CALP 已置 1 并且 PREDIV_A 位的值小于 3，则会忽略 CALP，即假定 CALP 等于 0 而执行校准。

要在 PREDIV_A 小于 3 的条件下执行校准，应降低同步预分频器值 (PREDIV_S) 以便每秒内可加速 8 个 RTCCLK 时钟周期，这意味着每 32 秒可增加 256 个时钟周期。因此，仅使用 CALM 位，可在每 32 秒内有效增加 255 到 256 个时钟脉冲（对应的校准范围为 243.3 ppm 到 244.1 ppm）。

在标称 RTCCLK 频率 32768 Hz 下，当 PREDIV_A 等于 1 时（分频系数为 2），应将 PREDIV_S 设置为 16379 而不是 16383（少 4）。唯一相关的其他情况是，当 PREDIV_A 等于 0 时，应将 PREDIV_S 设置为 32759 而不是 32767（少 8）。

如果以这种方式减少 PREDIV_S，则采用以下公式计算校准输入时钟的有效频率：

$$F_{CAL} = F_{RTCCLK} \times [1 + (256 - CALM) / (2^{20} + CALM - 256)]$$

在这种情况下，如果 RTCCLK 恰好为 32768.00 Hz，则当 CALM[7:0] 等于 0x100 时（CALM 范围的中值），说明设置正确。

验证 RTC 校准

通过测量 RTCCLK 的精确频率，计算正确的 CALM 和 CALP 值以确保 RTC 精度。此外，还为应用提供了一个可选的 1 Hz 输出，用来测量和验证 RTC 精度。

如果在有限的间隔内测量 RTC 的精确频率，则会导致测量期间产生最多 2 个 RTCCLK 时钟周期的测量误差，具体取决于数字校准周期与测量周期的对齐方式。

但是，如果测量周期与校准周期的长度相同，则可以消除此测量误差。在这种情况下，观测到的唯一误差是由数字校准的分辨率导致的误差。

- 默认情况下，校准周期为 32 秒。

在此模式下，测量整个 32 秒内 1 Hz 输出的精度，可确保测量误差在 0.477 ppm 内（32 秒内为 0.5 个 RTCCLK 周期，受校准分辨率限制）。

- 可将 RTC_CALR 寄存器的 CALW16 位置 1，以强制 16 秒的校准周期。

此时，可在 16 秒内测量 RTC 精度，产生的最大误差为 0.954 ppm（16 秒内为 0.5 个 RTCCLK 周期）。但是，由于校准分辨率降低，长期的 RTC 精度也会降到 0.954 ppm：将 CALW16 置 1 时，CALM[0] 位将始终保持为 0。

- 可将 RTC_CALR 寄存器的 CALW8 位置 1，以强制 8 秒的校准周期。

此时，可在 8 秒内测量 RTC 精度，产生的最大误差为 1.907 ppm（8 秒内为 0.5 个 RTCCLK 周期）。长期的 RTC 精度也会降到 1.907 ppm：将 CALW8 置 1 时，CALM[1:0] 位将始终保持为 00。

动态重校准

当 RTC_ISR/INITF=0 时，可动态更新校准寄存器 (RTC_CALR)，具体步骤如下：

1. 轮询 RTC_ISR/RECALPF（重新校准挂起标志）。
2. 如果该标志为 0，则可以根据需要向 RTC_CALR 写入新值。随后 RECALPF 位会被自动置为 1。
3. 新校准设置将在对 RTC_CALR 执行写操作之后的三个 ck_apre 周期内生效。

42.3.13 时间戳功能

将 RTC_CR 寄存器的 TSE 位或 ITSE 位置 1 可使能时间戳。

将 TSE 置 1 时：

当在 RTC_TS 引脚上检测到时间戳事件时，日历会保存到时间戳寄存器（RTC_TSSSR、RTC_TSTR 和 RTC_TSDR）中。

将 ITSE 置 1 时：

当检测到内部时间戳事件时，日历会保存到时间戳寄存器（RTC_TSSSR、RTC_TSTR 和 RTC_TSDR）中。切换至 VBAT 电源可生成内部时间戳事件。

由于内部事件或外部事件而发生时间戳事件时，RTC_ISR 寄存器中的时间戳标志位 (TSF) 将置 1。如果是内部事件，RTC_ISR 寄存器中的 ITSF 标志也将置 1。

通过将 RTC_CR 寄存器中的 TSIE 位置 1，可在发生入侵检测事件时生成中断。

如果在时间戳标志 (TSF) 已置 1 的条件下检测到新的时间戳事件，则时间戳上溢标志 (TSOVF) 将置 1，而时间戳寄存器（RTC_TSTR 和 RTC_TSDR）将保持上一事件的结果。

注： *在发生由同步过程引发的时间戳事件后，TSF 在 2 个 ck_apre 周期置为 1。*

TSOVF 的产生中不存在延迟。也就是说如果两个时间戳事件接连发生，TSOVF 可能为“1”而TSF为“0”。因此，建议只在检测到 TSF 为“1”后再轮询 TSOVF。

注意： 如果在 TSF 位清零后紧接着发生时间戳事件，则 TSF 和 TSOVF 位都将置 1。为防止在时间戳事件发生的同时屏蔽该事件，除非已将 TSF 位读取为“1”，否则应用程序不得将“0”写入 TSF 位。

此外，入侵事件可能导致时间戳被记录。有关 TAMPTS 控制位的说明，请参见*第 42.6.16 节：RTC 入侵配置寄存器 (RTC_TAMPCR)*。

42.3.14 入侵检测

RTC_TAMPx 入侵输入事件既可配置为边沿检测，也可配置为带过滤的电平检测。

入侵检测可配置用于以下目的：

- 擦除 RTC 备份寄存器（默认配置）
- 生成中断，能够从停止模式和待机模式唤醒
- 为低功耗定时器生成硬件触发

RTC 备份寄存器

备份寄存器 (RTC_BKPxR) 不会通过系统复位来复位，也不会在器件从待机模式唤醒时复位。

备份寄存器在发生入侵检测事件（请参见*第 42.6.20 节：RTC 备份寄存器 (RTC_BKPxR)* 和*第 1359 页的入侵检测初始化*）时复位，TAMPxNOERASE 位置 1 时或者 RTC_TAMPCR 寄存器中的 TAMPxMF 置 1 时除外。

入侵检测初始化

可通过将 RTC_TAMPCR 寄存器中相应的 TAMPxE 位置 1 来使能各输入。

每个 RTC_TAMPx 入侵检测输入分别与 RTC_ISR 寄存器中的标志 TAMPxF 相关联。

将 TAMPxMF 清零时：

TAMPxF 标志将在引脚上出现入侵事件后使能，并存在下述延迟：

- 当 TAMPFLT 不为 0x0 时（带过滤的电平检测），延迟为 3 个 ck_apre 周期
- 当 TAMPTS=1 时（入侵事件的时间戳），延迟为 3 个 ck_apre 周期
- 当 TAMPFLT=0x0（边沿检测）且 TAMPTS=0 时，无延迟

在此期间，只要 TAMPxF 置 1，就无法检测到同一引脚上出现的新入侵。

将 TAMPxMF 置 1 时：

在上述延迟期间以及在 2.5 个 ck_rtc 附加周期内，无法检测到同一引脚上出现的新入侵。

通过将 RTC_TAMPCR 寄存器中的 TAMPIE 位置 1，可在发生入侵检测事件时（当 TAMPxF 置 1 时）生成中断。当一个或多个 TAMPxMF 置 1 时，不允许将 TAMPIE 置 1。

将 TAMPIE 清零时，可通过将 RTC_TAMPCR 寄存器中相应的 TAMPxIE 位置 1 来分别使能各个入侵引脚事件中断。当相应的 TAMPxMF 置 1 时，不允许将 TAMPxIE 置 1。

出现入侵事件时生成触发输出

低功耗定时器可将入侵事件检测用作触发输入。

将 RTC_TAMPCR 寄存器中的 TAMPxMF 位清零时，必须通过软件将 TAMPxF 标志清零以便在同一引脚上检测新入侵。

将 TAMPxMF 位置 1 时，TAMPxF 标志会被屏蔽，并在 RTC_ISR 寄存器中保持清零。此配置允许在停止模式下自动触发低功耗定时器，无需通过系统唤醒来执行 TAMPxF 清零。在这种情况下，备份寄存器不清零。

入侵事件的时间戳

当 TAMPTS 置“1”时，任何入侵事件都会导致时间戳事件发生。在这种情况下，如同发生正常时间戳事件一样，RTC_ISR 中的 TSF 位或 TSOVF 位会置 1。在 TSF 或 TSOVF 置 1 的同时，受影响的入侵标志寄存器 TAMPxF 也会随之置 1。

对入侵输入的边沿检测

如果 TAMPFLT 位设置为“00”，当相应的 TAMPxTRG 位上出现上升沿或下降沿时，RTC_TAMPx 引脚将生成入侵检测事件。选择边沿检测时，会禁止 RTC_TAMPx 输入上的内部上拉电阻。

注意： 使用边沿检测时，建议在使能入侵检测后（通过读取 GPIO 寄存器）以及向备份寄存器中写入敏感值前立即通过软件检查入侵引脚电平，以确保使能入侵事件检测后才出现有效边沿。当 TAMPFLT = “00”且 TAMPxTRG = 0（上升沿检测）时，如果在使能入侵检测前入侵输入已处于高电平，则可通过硬件来检测入侵事件。

检测到入侵事件并清零后，应当在重新编程备份寄存器 (RTC_BKPxR) 之前禁止 RTC_TAMPx，然后再重新使能（TAMPxE 置 1）。这可防止应用在 RTC_TAMPx 输入值仍指示入侵检测时，对备份寄存器执行写操作。这相当于对 RTC_TAMPx 输入的电平检测。

注： *当 V_{DD} 电源关闭时，入侵检测仍有效。要避免意外复位备份寄存器，应将 RTC_TAMPx 映射到的引脚从外部连接到正确的电平。*

对 RTC_TAMPx 输入的带过滤电平检测

通过将 TAMPFLT 设置为非零值可执行带过滤的电平检测。在 TAMPxTRG 位指定的电平连续出现 2、4 或 8 个（取决于 TAMPFLT 值）采样时生成入侵检测事件。

除非通过将 TAMPPUDIS 置 1 禁止入侵输入，否则在入侵输入的状态被采样前，将通过 I/O 内部上拉电阻对 RTC_TAMPx 输入预充电。预充电的持续时间由 TAMPPRCH 位确定，允许增大 RTC_TAMPx 输入上的电容。

可使用 TAMPFREQ 确定用于电平检测的采样频率，以便使入侵检测延迟与上拉电阻功耗之间达到最佳平衡。

注： *有关上拉电阻的电气特性，请参见数据手册。*

42.3.15 校准时钟输出

将 RTC_CR 寄存器中的 COE 位置 1 时，会在 RTC_CALIB 器件输出上提供一个参考时钟。

如果 RTC_CR 寄存器中的 COSEL 位复位且 PREDIV_A = 0x7F，则 RTC_CALIB 频率为 $f_{RTCCLK/64}$。这相当于 RTCCLK 频率为 32.768 kHz 时，512 Hz 的校准输出。RTC_CALIB 占空比是不规则的：下降沿上存在轻微抖动。因此推荐使用上升沿。

如果 COSEL 置 1 且“PREDIV_S+1”为 256 的非零整数倍（比如：PREDIV_S[7:0] = 0xFF），则 RTC_CALIB 频率为 $f_{RTCCLK}/(256 * (PREDIV_A+1))$。这相当于 RTCCLK 频率为 32.768 kHz 时， 1 Hz 的校准输出，其中预分频器为默认值 （PREDIV_A = Ox7F、 PREDIV_S = 0xFF）。1 Hz 输出会在进行平移操作时受到影响，并且可能在平移操作期间发生翻转 (SHPF = 1)。

注： *选择 RTC_CALIB 或 RTC_ALARM 输出时，RTC_OUT 引脚会自动配置为输出。*

COSEL 位清零时，RTC_CALIB 输出为异步预分频器的第 6 级输出。

COSEL 位置 1 时，RTC_CALIB 输出为同步预分频器的第 8 级输出。

42.3.16 闹钟输出

RTC_CR 寄存器中的 OSEL[1:0] 控制位用于激活闹钟输出 RTC_ALARM，以及选择输出的功能。这些功能可反映 RTC_ISR 寄存器中相应标志的内容。

输出的极性由 RTC_CR 中的 POL 控制位确定，这样当 POL 置 1 时会输出选定标志位的相反值。

闹钟输出

使用 RTC_OR 寄存器中的控制位 RTC_ALARM_TYPE 可将 RTC_ALARM 引脚配置为输出开漏或输出上拉。

注： *使能 RTC_ALARM 输出之后，其优先级高于 RTC_CALIB（与 COE 位无关，该位必须保持清零）。*

选择 RTC_CALIB 或 RTC_ALARM 输出时，RTC_OUT 引脚会自动配置为输出。

42.4 RTC 低功耗模式

表 243. 低功耗模式对 RTC 的作用

模式	说明
睡眠	无影响。 RTC 中断可使器件退出睡眠模式。
低功耗运行	无影响。
低功耗睡眠	无影响。RTC 中断可使器件退出低功耗睡眠模式。
停止 0	外设寄存器内容保持不变。
停止 1	
停止 2	
待机	当 RTC 时钟源为 LSE 或 LSI 时，RTC 保持工作状态。RTC 闹钟、RTC 入侵事件、RTC 时间戳事件和 RTC 唤醒会使器件退出待机模式。
关断	当 RTC 时钟源为 LSE 时，RTC 保持工作状态。RTC 闹钟、RTC 入侵事件、RTC 时间戳事件和 RTC 唤醒会使器件退出关断模式。

42.5 RTC 中断

所有 RTC 中断均与 EXTI 控制器相连。请参见*第 16 节：扩展中断和事件控制器 (EXTI)*。

要使能 RTC 闹钟中断，需按照以下顺序操作：

1. 将对应于 RTC 闹钟事件的 EXTI 线配置为中断模式并将其使能，然后选择上升沿有效。
2. 配置 NVIC 中的 RTC_ALARM IRQ 通道并将其使能。
3. 配置 RTC 以生成 RTC 闹钟。

要使能 RTC 入侵中断，需按照以下顺序操作：

1. 将对应于 RTC 入侵事件的 EXTI 线配置为中断模式并将其使能，然后选择上升沿有效。
2. 配置 NVIC 中的 RTC_TAMP_STAMP IRQ 通道并将其使能。
3. 配置 RTC 以检测 RTC 入侵事件。

要使能 RTC 时间戳中断，需按照以下顺序操作：

1. 将对应于 RTC 时间戳事件的 EXTI 线配置为中断模式并将其使能，然后选择上升沿有效。
2. 配置 NVIC 中的 RTC_TAMP_STAMP IRQ 通道并将其使能。
3. 配置 RTC 以检测 RTC 时间戳事件。

要使能唤醒定时器中断，需按照以下顺序操作：

1. 将对应于唤醒定时器事件的 EXTI 线配置为中断模式并将其使能，然后选择上升沿有效。
2. 配置 NVIC 中的 RTC_WKUP IRQ 通道并将其使能。
3. 配置 RTC 以检测 RTC 唤醒定时器事件。

表 244. 中断控制位

中断事件	事件标志	使能控制位	退出睡眠模式	退出停止模式	退出待机模式
闹钟 A	ALRAF	ALRAIE	是	是(1)	是(1)
闹钟 B	ALRBF	ALRBIE	是	是(1)	是(1)
RTC_TS 输入（时间戳）	TSF	TSIE	是	是(1)	是(1)
RTC_TAMP1 输入检测	TAMP1F	TAMPIE	是	是(1)	是(1)
RTC_TAMP2 输入检测	TAMP2F	TAMPIE	是	是(1)	是(1)
RTC_TAMP3 输入检测	TAMP3F	TAMPIE	是	是(1)	是(1)
唤醒定时器中断	WUTF	WUTIE	是	是(1)	是(1)

1. 仅当 RTC 时钟源为 LSE 或 LSI 时，才能从停机和待机模式唤醒。

42.6 RTC 寄存器

有关寄存器说明中使用的缩写，请参见参考手册中的*第 72 页的第 1.1 节*。

外设寄存器可按字（32 位）进行访问。

42.6.1 RTC 时间寄存器 (RTC_TR)

RTC time register

RTC_TR 是日历时间影子寄存器。只能在初始化模式下对该寄存器执行写操作。请参见*第 1353 页的日历初始化和配置*和*第 1354 页的读取日历*。

此寄存器受写保护。*第 1352 页的RTC 寄存器写保护*中介绍了写访问的过程。

偏移地址：0x00

Backup 域复位值：0x0000 0000

系统复位：当 BYPSHAD = 0 时为 0x0000 0000。当 BYPSHAD = 1 时不受影响。

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.	PM	HT[1:0]		HU[3:0]			
									rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	MNT[2:0]			MNU[3:0]				Res.	ST[2:0]			SU[3:0]			
	rw	rw	rw	rw	rw	rw	rw		rw	rw	rw	rw	rw	rw	rw

位 31-23 保留，必须保持复位值

位 22 **PM**：AM/PM 符号 (AM/PM notation)
0：AM 或 24 小时制
1：PM

位 21:20 **HT[1:0]**：小时的十位（BCD 格式）(Hour tens in BCD format)

位 19:16 **HU[3:0]**：小时的个位（BCD 格式）(Hour units in BCD format)

位 15 保留，必须保持复位值。

位 14:12 **MNT[2:0]**：分钟的十位（BCD 格式）(Minute tens in BCD format)

位 11:8 **MNU[3:0]**：分钟的个位（BCD 格式）(Minute units in BCD format)

位 7 保留，必须保持复位值。

位 6:4 **ST[2:0]**：秒的十位（BCD 格式）(Second tens in BCD format)

位 3:0 **SU[3:0]**：秒的个位（BCD 格式）(Second units in BCD format)

42.6.2 RTC 日期寄存器 (RTC_DR)

RTC date register

RTC_DR 是日历日期影子寄存器。只能在初始化模式下对该寄存器执行写操作。请参见 *第 1353 页的日历初始化和配置*和*第 1354 页的读取日历*。

此寄存器受写保护。*第 1352 页的RTC 寄存器写保护*中介绍了写访问的过程。

偏移地址：0x04

Backup 域复位值：0x0000 2101

系统复位：当 BYPSHAD = 0 时为 0x0000 2101。当 BYPSHAD = 1 时不受影响。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	YT[3:0]				YU[3:0]			
								rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WDU[2:0]			MT	MU[3:0]				Res.	Res.	DT[1:0]		DU[3:0]			
rw	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 **YT[3:0]**：年份的十位（BCD 格式）(Year tens in BCD format)

位 19:16 **YU[3:0]**：年份的个位（BCD 格式）(Year units in BCD format)

位 15:13 **WDU[2:0]**：星期几的个位 (Week day units)

000：禁止
001：星期一
...
111：星期日

位 12 **MT**：月份的十位（BCD 格式）(Month tens in BCD format)

位 11:8 **MU**：月份的个位（BCD 格式）(Month units in BCD format)

位 7:6 保留，必须保持复位值。

位 5:4 **DT[1:0]**：日期的十位（BCD 格式）(Date tens in BCD format)

位 3:0 **DU[3:0]**：日期的个位（BCD 格式）(Date units in BCD format)

42.6.3 RTC 控制寄存器 (RTC_CR)

RTC control register

偏移地址：0x08

Backup 域复位值：0x0000 0000

系统复位：不受影响

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	ITSE	COE	OSEL[1:0]		POL	COSEL	BKP	SUB1H	ADD1H
							rw	rw	rw	rw	rw	rw	rw	w	w

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TSIE	WUTIE	ALRBIE	ALRAIE	TSE	WUTE	ALRBE	ALRAE	Res.	FMT	BYPSHAD	REFCKON	TSEDGE	WUCKSEL[2:0]		
rw	rw	rw	rw	rw	rw	rw	rw		rw	rw	rw	rw	rw	rw	rw

位 31:25 保留，必须保持复位值。

位 24 **ITSE**：内部事件时间戳使能 (timestamp on internal event enable)
0：禁止内部事件时间戳
1：使能内部事件时间戳

位 23 **COE**：校准输出使能 (Calibration output enable)
该位使能 RTC_CALIB 输出
0：禁止校准输出
1：使能校准输出

位 22:21 **OSEL[1:0]**：输出选择 (Output selection)
这些位用于选择要连接到 RTC_ALARM 输出的标志
00：禁止输出
01：使能闹钟 A 输出
10：使能闹钟 B 输出
11：使能唤醒输出

位 20 **POL**：输出极性 (Output polarity)
该位用于配置 RTC_ALARM 输出的极性
0：当 ALRAF/ALRBF/WUTF 置 1 时（取决于 OSEL[1:0]），该引脚为高电平
1：当 ALRAF/ALRBF/WUTF 置 1 时（取决于 OSEL[1:0]），该引脚为低电平

位 19 **COSEL**：校准输出选择 (Calibration output selection)
当 COE=1 时，该位可选择 RTC_CALIB 上输出的信号。
0：校准输出为 512 Hz（采用默认预分频器设置）
1：校准输出为 1 Hz（采用默认预分频器设置）
在 RTCCLK 为 32.768 kHz 且预分频器为其默认值（PREDIV_A=127 且 PREDIV_S=255）的条件下，这些频率有效。请参见*第 42.3.15 节：校准时钟输出*。

位 18 **BKP**：备份 (Backup)
用户可对此位执行写操作以记录是否已对夏令时进行更改。

位 17 **SUB1H**：减少 1 小时（冬季时间更改）(Subtract 1 hour (winter time change))
该位置 1 时，如果当前小时不是 0，则日历时间将减少 1 小时。此位始终读为 0。
当前小时为 0 时，将此位置 1 没有任何作用。
0：无影响。
1：将当前时间减少 1 小时。这可用于冬季时间更改（在初始化模式以外）。

位 16 **ADD1H**：增加 1 小时（夏季时间更改）(Add 1 hour (summer time change))
该位置 1 时，日历时间将增加 1 小时。此位始终读为 0。
0：无影响。
1：将当前时间增加 1 小时。这可用于夏季时间更改（在初始化模式以外）。

位 15 **TSIE**：时间戳中断使能 (Time-stamp interrupt enable)
0：禁止时间戳中断
1：使能时间戳中断

位 14 **WUTIE**：唤醒定时器中断使能 (Wakeup timer interrupt enable)
0：禁止唤醒定时器中断
1：使能唤醒定时器中断

位 13 **ALRBIE**：闹钟 B 中断使能 (Alarm B interrupt enable)
0：禁止闹钟 B 中断
1：使能闹钟 B 中断

位 12 **ALRAIE**：闹钟 A 中断使能 (Alarm A interrupt enable)
0：禁止闹钟 A 中断
1：使能闹钟 A 中断

位 11 **TSE**：时间戳使能 (timestamp enable)
0：禁止时间戳
1：使能时间戳

位 10 **WUTE**：唤醒定时器使能 (Wakeup timer enable)
0：禁止唤醒定时器
1：使能唤醒定时器
注： 唤醒定时器禁止时，需要等到 WUTWF=1 后才能重新使能。

位 9 **ALRBE**：闹钟 B 使能 (Alarm B enable)
0：禁止闹钟 B
1：使能闹钟 B

位 8 **ALRAE**：闹钟 A 使能 (Alarm A enable)
0：禁止闹钟 A
1：使能闹钟 A

位 7 保留，必须保持复位值。

位 6 **FMT**：小时格式 (Hour format)
0：24 小时 / 天格式
1：AM/PM 小时格式

位 5 **BYPSHAD**：旁路影子寄存器 (Bypass the shadow registers)

0：日历值（从 RTC_SSR、RTC_TR 和 RTC_DR 读取时）取自影子寄存器，该影子寄存器每两个 RTCCLK 周期更新一次。

1：日历值（从 RTC_SSR、RTC_TR 和 RTC_DR 读取时）直接取自日历计数器。

注： 如果 APB 时钟的频率低于 7 倍的 RTCCLK 频率，则必须将 BYPSHAD 置“1”。

位 4 **REFCKON**：RTC_REFIN 参考时钟检测使能（50 Hz 或 60 Hz）(RTC_REFIN reference clock detection enable (50 or 60 Hz))

0：禁止 RTC_REFIN 检测

1：使能 RTC_REFIN 检测

注： PREDIV_S 必须为 0x00FF。

位 3 **TSEDGE**：时间戳事件有效边沿 (Timestamp event active edge)

0：RTC_TS 输入上升沿生成时间戳事件

1：RTC_TS 输入下降沿生成时间戳事件

TSEDGE 发生更改时，必须复位 TSE 以避免将 TSF 意外置 1。

位 2:0 **WUCKSEL[2:0]**：唤醒时钟选择 (Wakeup clock selection)

000：选择 RTC/16 时钟

001：选择 RTC/8 时钟

010：选择 RTC/4 时钟

011：选择 RTC/2 时钟

10x：选择 ck_spre 时钟（通常为 1 Hz）

11x：选择 ck_spre 时钟（通常为 1 Hz）并将 WUT 计数器值增加 2^{16}（见下面的注释）

注： *只能在初始化模式下 (RTC_ISR/INITF = 1) 对该寄存器的位 7、6 和 4 执行写操作。*

WUT = 唤醒单元计数器值。当 WUCKSEL[2:1 = 11] 时，WUT = (0x0000 到 0xFFFF) + 0x10000（增加的值）。

只能在 RTC_CR WUTE 位 = 0 且 RTC_ISR WUTWF 位 = 1 时对该寄存器的位 2 到 0 执行写操作。

建议不要在日历小时递增时更改小时，因为这样做会屏蔽日历小时的增量。

ADD1H 和 SUB1H 的更改在下一秒生效。

此寄存器受写保护。第 1352 页的RTC 寄存器写保护中介绍了写访问的过程。

注意： TSEDGE 发生更改时，必须复位 TSE 以避免对 TSF 的误操作。

42.6.4 RTC 初始化和状态寄存器 (RTC_ISR)

RTC initialization and status register

此寄存器受写保护（RTC_ISR[13:8] 位除外）。*第 1352 页的RTC 寄存器写保护*中介绍了写访问的过程。

偏移地址：0x0C

Backup 域复位值：0x0000 0007

系统复位：不受影响（INIT、INITF 和 RSF 位除外，它们在复位时被清零）

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.	ITSF	RECALPF
														rc_w0	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAMP3F	TAMP2F	TAMP1F	TSOVF	TSF	WUTF	ALRBF	ALRAF	INIT	INITF	RSF	INITS	SHPF	WUTWF	ALRBWF	ALRAWF
rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rw	r	rc_w0	r	r	r	r	r

位 31:18 保留，必须保持复位值

位 17 **ITSF**：内部时间戳标志 (Internal time-stamp flag)
发生内部时间戳事件时，由硬件将此标志置 1。
该标志由软件写零清零，且必须通过向两个位写零将此标志与 TSF 位一起清零。

位 16 **RECALPF**：重新校准挂起标志 (Recalibration pending Flag)
当软件对 RTC_CALR 寄存器执行写操作时，RECALPF 状态标志将自动置“1”，指示 RTC_CALR 寄存器已屏蔽。当采用新的校准设置时，该位恢复为 “0”。请参见*动态重校准*。

位 15 **TAMP3F**：RTC_TAMP3 检测标志 (RTC_TAMP3 detection flag)
在 RTC_TAMP3 输入上检测到入侵检测事件时，由硬件将此标志置 1。
该标志由软件写零清除

位 14 **TAMP2F**：RTC_TAMP2 检测标志 (RTC_TAMP2 detection flag)
在 RTC_TAMP2 输入上检测到入侵检测事件时，由硬件将此标志置 1。
该标志由软件写零清除

位 13 **TAMP1F**：RTC_TAMP1 检测标志 (RTC_TAMP1 detection flag)
在 RTC_TAMP1 输入上检测到入侵检测事件时，由硬件将此标志置 1。
该标志由软件写零清除

位 12 **TSOVF**：时间戳溢出标志 (Time-stamp overflow flag)
当在 TSF 已置 1 的情况下发生时间戳事件时，由硬件将此标志置 1。
该标志由软件写零清除。建议仅在 TSF 位清零之后再检查并清零 TSOVF 位。否则，如果时间戳事件恰好在清零 TSF 位之前刚刚发生，则溢出事件可能会被漏掉。

位 11 **TSF**：时间戳标志 (Time-stamp flag)
发生时间戳事件时，由硬件将此标志置 1。
该标志由软件写零清除。如果 ITSF 标志已置 1，必须通过向两个位写零将 TSF 与 ITSF 一起清零。

位 10 **WUTF**：唤醒定时器标志 (Wakeup timer flag)

当唤醒自动重载计数器计数到 0 时，由硬件将此标志置 1。

该标志由软件写零清除。

软件必须在 WUTF 再次置 1 的 1.5 个 RTCCLK 周期之前将该标志清零。

位 9 **ALRBF**：闹钟 B 标志 (Alarm B flag)

当时间/日期寄存器（RTC_TR 和 RTC_DR）与闹钟 B 寄存器 (RTC_ALRMBR) 匹配时，由硬件将该标志置 1。

该标志由软件写零清除。

位 8 **ALRAF**：闹钟 A 标志 (Alarm A flag)

当时间/日期寄存器（RTC_TR 和 RTC_DR）与闹钟 A 寄存器 (RTC_ALRMAR) 匹配时，由硬件将该标志置 1。

该标志由软件写零清除。

位 7 **INIT**：初始化模式 (Initialization mode)

0：自由运行模式

1：初始化模式，用于编程时间和日期寄存器（RTC_TR 和 RTC_DR）以及预分频器寄存器 (RTC_PRER)。计数器停止计数，当 INIT 被复位后，计数器从新值开始计数。

位 6 **INITF**：初始化标志 (Initialization flag)

当此位置 1 时，RTC 处于初始化状态，此时可更新事件、日期和预分频器寄存器。

0：不允许更新日历寄存器

1：允许更新日历寄存器

位 5 **RSF**：寄存器同步标志 (Registers synchronization flag)

每次将日历寄存器的值复制到影子寄存器（RTC_SSRx、RTC_TRx 和 RTC_DRx）时，都会由硬件将此位置 1。在初始化模式下、平移操作挂起时 (SHPF=1) 或在旁路影子寄存器模式 (BYPSHAD=1) 下，该位由硬件清零。该位还可由软件清零。

在初始化模式下，该位可由软件或硬件清零。

0：日历影子寄存器尚未同步

1：日历影子寄存器已同步

位 4 **INITS**：初始化状态标志 (Initialization status flag)

当日历年份字段不为 0 时（Backup域复位状态），由硬件将该位置 1。

0：日历尚未初始化

1：日历已经初始化

位 3 **SHPF**：平移操作挂起 (Shift operation pending)

0：没有平移操作挂起

1：某个平移操作挂起

只要通过对 RTC_SHIFTR 寄存器执行写操作来启动平移操作，此标志便由硬件置 1。执行完相应的平移操作后，此标志由硬件清零。对 SHPF 位执行写入操作不起作用。

位 2 **WUTWF**：唤醒定时器写标志 (Wakeup timer write flag)

此位在 RTC_CR 中的 WUTE 位清零后，并在 2 个 RTCCLK 周期后由硬件置 1，在 WUTE 位置 1 后并在 2 个 RTCCLK 周期后清零。当 WUTE 位清零且 WUTWF 位置 1 时，可更改唤醒定时器的值。

0：不允许更新唤醒定时器配置

1：允许更新唤醒定时器配置

位 1 **ALRBWF**：闹钟 B 写标志 (Alarm B write flag)

在 RTC_CR 寄存器中的 ALRBE 位置 0 之后，当闹钟 B 的值可更改时，由硬件将该位置 1。该位在初始化模式下由硬件清零。

0：不允许更新闹钟 B

1：允许更新闹钟 B

位 0 **ALRAWF**：闹钟 A 写标志 (Alarm A write flag)

在 RTC_CR 寄存器中的 ALRAE 位置 0 后，当闹钟 A 的值可更改时，由硬件将该位置 1。该位在初始化模式下由硬件清零。

0：不允许更新闹钟 A

1：允许更新闹钟 A

注： *将 ALRAF、ALRBF、WUTF 和 TSF 位编程为 0 之后 2 个 APB 时钟周期，清零生效。*

42.6.5 RTC 预分频器寄存器 (RTC_PRER)

RTC prescaler register

只能在初始化模式下对该寄存器执行写操作。必须通过两次独立的写访问执行初始化。请参见*第 1353 页的日历初始化和配置*。

此寄存器受写保护。*第 1352 页的 RTC 寄存器写保护*中介绍了写访问的过程。

偏移地址：0x10

Backup 域复位值：0x007F 00FF

系统复位：不受影响

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.	PREDIV_A[6:0]						
									rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	PREDIV_S[14:0]														
	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 22:16 **PREDIV_A[6:0]**：异步预分频系数 (Asynchronous prescaler factor)
下面是异步分频系数的公式：
ck_apre 频率 = RTCCLK 频率/(PREDIV_A+1)

位 15 保留，必须保持复位值。

位 14:0 **PREDIV_S[14:0]**：同步预分频系数 (Synchronous prescaler factor)
下面是同步分频系数的公式：
ck_spre 频率 = ck_apre 频率/(PREDIV_S+1)

42.6.6 RTC 唤醒定时器寄存器 (RTC_WUTR)

RTC wakeup timer register

仅当 RTC_ISR 中的 WUTWF 置 1 时才可对该寄存器执行写操作。

此寄存器受写保护。*第 1352 页的 RTC 寄存器写保护*中介绍了写访问的过程。

偏移地址：0x14

Backup 域复位值：0x0000 FFFF

系统复位：不受影响

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
WUT[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 **WUT[15:0]**：唤醒自动重载值位 (Wakeup auto-reload value bits)

当使能唤醒定时器时（WUTE 置 1），每 (WUT[15:0] + 1) 个 ck_wut 周期将 WUTF 标志置 1 一次。ck_wut 周期通过 RTC_CR 寄存器的 WUCKSEL[2:0] 位进行选择。

当 WUCKSEL[2] = 1 时，唤醒定时器变为 17 位，WUCKSEL[1] 等效为 WUT[16]，即要重载到定时器的最高有效位。

WUTF 第一次置 1 发生在 WUTE 置 1 之后 (WUT+1) 个 ck_wut 周期。禁止在 WUCKSEL[2:0]=011(RTCCLK/2) 时将 WUT[15:0] 设置为 0x0000。

42.6.7 RTC 闹钟 A 寄存器 (RTC_ALRMAR)

RTC alarm A register

仅当 RTC_ISR 中的 ALRAWF 置 1 时或在初始化模式下，才可以对该寄存器执行写操作。

此寄存器受写保护。*第 1352 页的 RTC 寄存器写保护*中介绍了写访问的过程。

偏移地址：0x1C

Backup 域复位值：0x0000 0000

系统复位：不受影响

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MSK4	WDSEL	DT[1:0]		DU[3:0]				MSK3	PM	HT[1:0]		HU[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
MSK2	MNT[2:0]			MNU[3:0]				MSK1	ST[2:0]			SU[3:0]			
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31 **MSK4**：闹钟 A 日期掩码 (Alarm A date mask)
0：如果日期/日匹配，则闹钟 A 置 1
1：在闹钟 A 比较中，日期/日无关

位 30 **WDSEL**：星期几选择 (Week day selection)
0：DU[3:0] 代表日期的个位
1：DU[3:0] 代表星期几 DT[1:0] 为无关位

位 29:28 **DT[1:0]**：日期的十位（BCD 格式）(Date tens in BCD format)

位 27:24 **DU[3:0]**：日期的个位或日（BCD 格式）(Date units or day in BCD format)

位 23 **MSK3**：闹钟 A 小时掩码 (Alarm A hours mask)
0：如果小时匹配，则闹钟 A 置 1
1：在闹钟 A 比较中，小时无关

位 22 **PM**：AM/PM 符号 (AM/PM notation)
0：AM 或 24 小时制
1：PM

位 21:20 **HT[1:0]**：小时的十位（BCD 格式）(Hour tens in BCD format)

位 19:16 **HU[3:0]**：小时的个位（BCD 格式）(Hour units in BCD format)

位 15 **MSK2**：闹钟 A 分钟掩码 (Alarm A minutes mask)
0：如果分钟匹配，则闹钟 A 置 1
1：在闹钟 A 比较中，分钟无关

位 14:12 **MNT[2:0]**：分钟的十位（BCD 格式）(Minute tens in BCD format)

位 11:8 **MNU[3:0]**：分钟的个位（BCD 格式）(Minute units in BCD format)

位 7 **MSK1**：闹钟 A 秒掩码 (Alarm A seconds mask)
0：如果秒匹配，则闹钟 A 置 1
1：在闹钟 A 比较中，秒无关

位 6:4 **ST[2:0]**：秒的十位（BCD 格式）(Second tens in BCD format)

位 3:0 **SU[3:0]**：秒的个位（BCD 格式）(Second units in BCD format)

42.6.8 RTC 闹钟 B 寄存器 (RTC_ALRMBR)

RTC alarm B register

仅当 RTC_ISR 中的 ALRBWF 置 1 时或在初始化模式下，才可以对该寄存器执行写操作。

此寄存器受写保护。*第 1352 页的 RTC 寄存器写保护*中介绍了写访问的过程。

偏移地址：0x20

Backup 域复位值：0x0000 0000

系统复位：不受影响

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MSK4	WDSEL	DT[1:0]		DU[3:0]				MSK3	PM	HT[1:0]		HU[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
MSK2	MNT[2:0]			MNU[3:0]				MSK1	ST[2:0]			SU[3:0]			
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31 **MSK4**：闹钟 B 日期掩码 (Alarm B date mask)
0：如果日期和日匹配，则闹钟 B 置 1
1：在闹钟 B 比较中，日期和日无关

位 30 **WDSEL**：星期几选择 (Week day selection)
0：DU[3:0] 代表日期的个位
1：DU[3:0] 代表星期几 DT[1:0] 为无关位

位 29:28 **DT[1:0]**：日期的十位（BCD 格式）(Date tens in BCD format)

位 27:24 **DU[3:0]**：日期个位或日（BCD 格式）(Date units or day in BCD format)

位 23 **MSK3**：闹钟 B 小时掩码 (Alarm B hours mask)
0：如果小时匹配，则闹钟 B 置 1
1：在闹钟 B 比较中，小时无关

位 22 **PM**：AM/PM 符号 (AM/PM notation)
0：AM 或 24 小时制
1：PM

位 21:20 **HT[1:0]**：小时的十位（BCD 格式）(Hour tens in BCD format)

位 19:16 **HU[3:0]**：小时的个位（BCD 格式）(Hour units in BCD format)

位 15 **MSK2**：闹钟 B 分钟掩码 (Alarm B minutes mask)
0：如果分钟匹配，则闹钟 B 置 1
1：在闹钟 B 比较中，分钟无关

位 14:12 **MNT[2:0]**：分钟的十位（BCD 格式）(Minute tens in BCD format)

位 11:8 **MNU[3:0]**：分钟的个位（BCD 格式）(Minute units in BCD format)

位 7 **MSK1**：闹钟 B 秒掩码 (Alarm B seconds mask)
0：如果秒匹配，则闹钟 B 置 1
1：在闹钟 B 比较中，秒无关

位 6:4 **ST[2:0]**：秒的十位（BCD 格式）(Second tens in BCD format)

位 3:0 **SU[3:0]**：秒的个位（BCD 格式）(Second units in BCD format)

42.6.9 RTC 写保护寄存器 (RTC_WPR)

RTC write protection 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
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.	Res.	Res.	Res.	Res.	Res.	Res.	KEY							
								w	w	w	w	w	w	w	w

位 31:8 保留，必须保持复位值。

位 7:0 **KEY**：写保护关键字 (Write protection key)
可通过软件对该字节执行写操作。
读取该字节时，始终返回 0x00。
有关如何解锁 RTC 寄存器写保护的介绍，请参见 *RTC 寄存器写保护*。

42.6.10 RTC 亚秒寄存器 (RTC_SSR)

RTC sub second register

偏移地址：0x28

Backup 域复位值：0x0000 0000

系统复位：当 BYPSHAD = 0 时为 0x0000 0000。当 BYPSHAD = 1 时不受影响。

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
SS[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 **SS**：亚秒值 (Sub seconds value)
SS[15:0] 是同步预分频器计数器的值。此亚秒值可根据以下公式得出：
亚秒值 = (PREDIV_S - SS) / (PREDIV_S + 1)
注： 仅当执行平移操作之后，SS 才能大于 PREDIV_S。在这种情况下，正确的时间/日期比 RTC_TR/RTC_DR 所指示的时间/日期慢一秒钟。

42.6.11 RTC 平移控制寄存器 (RTC_SHIFTR)

RTC shift control register

此寄存器受写保护。*第 1352 页的 RTC 寄存器写保护*中介绍了写访问的过程。

偏移地址：0x2C

Backup 域复位值：0x0000 0000

系统复位：不受影响

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ADD1S	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
w															

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

位 31 **ADD1S**：增加一秒钟 (Add one second)

0：无影响

1：对时钟/日历增加一秒钟

此位为只写位且始终读为 0。当平移操作挂起（RTC_ISR 中的 SHPF=1）时，对该位执行写操作无作用。

此函数应与 SUBFS 配合使用（请参见下文介绍），以便有效地向原子操作机制的时钟添加亚秒值。

位 30:15 保留，必须保持复位值

位 14:0 **SUBFS**：减少亚秒值 (Subtract a fraction of a second)

此位为只写位且始终读为 0。当平移操作挂起（RTC_ISR 中的 SHPF=1）时，对该位执行写操作无作用。

写入 SUBFS 的值将加到同步预分频器计数器中。由于该计数器递减计数，此操作可有效地从时钟减去（延迟）以下时间：

延迟（秒）= SUBFS / (PREDIV_S + 1)

当 ADD1S 函数与 SUBFS 结合使用时，可有效地将亚秒值增加到时钟（提前时钟），使时钟提前以下时间：

提前（秒）= (1 - (SUBFS / (PREDIV_S + 1)))。

注： *对 SUBFS 执行写操作将使 RSF 清零。软件随后会等待至 RSF=1 以确定影子寄存器已更新为平移后的时间。*

42.6.12 RTC 时间戳时间寄存器 (RTC_TSTR)

RTC timestamp time register

仅当 RTC_ISR 中的 TSF 置 1 时，该寄存器的内容才有效。当 TSF 位复位时，清零该寄存器。

偏移地址：0x30

Backup 域复位值：0x0000 0000

系统复位：不受影响

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.	PM	HT[1:0]		HU[3:0]			
									r	r	r	r	r	r	r

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	MNT[2:0]			MNU[3:0]				Res.	ST[2:0]			SU[3:0]			
	r	r	r	r	r	r	r		r	r	r	r	r	r	r

位 31:23 保留，必须保持复位值

位 22 **PM**：AM/PM 符号 (AM/PM notation)
0：AM 或 24 小时制
1：PM

位 21:20 **HT[1:0]**：小时的十位（BCD 格式）(Hour tens in BCD format)

位 19:16 **HU[3:0]**：小时的个位（BCD 格式）(Hour units in BCD format)

位 15 保留，必须保持复位值

位 14:12 **MNT[2:0]**：分钟的十位（BCD 格式）(Minute tens in BCD format)

位 11:8 **MNU[3:0]**：分钟的个位（BCD 格式）(Minute units in BCD format)

位 7 保留，必须保持复位值

位 6:4 **ST[2:0]**：秒的十位（BCD 格式）(Second tens in BCD format)

位 3:0 **SU[3:0]**：秒的个位（BCD 格式）(Second units in BCD format)

42.6.13 RTC 时间戳日期寄存器 (RTC_TSDR)

RTC timestamp date register

仅当 RTC_ISR 中的 TSF 置 1 时，该寄存器的内容才有效。当 TSF 位复位时，清零该寄存器。

偏移地址：0x34

Backup 域复位值：0x0000 0000

系统复位：不受影响

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
WDU[1:0]			MT	MU[3:0]				Res.	Res.	DT[1:0]		DU[3:0]			
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 15:13 **WDU[1:0]**：星期几的个位 (Week day units)

位 12 **MT**：月份的十位（BCD 格式）(Month tens in BCD format)

位 11:8 **MU[3:0]**：月份的个位（BCD 格式）(Month units in BCD format)

位 7:6 保留，必须保持复位值

位 5:4 **DT[1:0]**：日期的十位（BCD 格式）(Date tens in BCD format)

位 3:0 **DU[3:0]**：日期的个位（BCD 格式）(Date units in BCD format)

42.6.14 RTC 时间戳亚秒寄存器 (RTC_TSSSR)

RTC time-stamp sub second register

仅当 RTC_ISR/TSF 置 1 时，该寄存器的内容才有效。当 RTC_ISR/TSF 位复位时，清零该寄存器。

偏移地址：0x38

Backup 域复位值：0x0000 0000

系统复位：不受影响

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
SS[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 **SS**：亚秒值 (Sub seconds value)

当发生时间戳事件时，SS[15:0] 是同步预分频器计数器的值。

42.6.15 RTC 校准寄存器 (RTC_CALR)

RTC calibration register

此寄存器受写保护。*第 1352 页的 RTC 寄存器写保护*中介绍了写访问的过程。

偏移地址：0x3C

Backup 域复位值：0x0000 0000

系统复位：不受影响

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
CALP	CALW8	CALW16	Res.	Res.	Res.	Res.	CALM[8:0]								
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 **CALP**：将 RTC 的频率增加 488.5 ppm (Increase frequency of RTC by 488.5 ppm)
0：不增加 RTCCLK 脉冲。
1：每 2^{11} 个脉冲有效插入一个 RTCCLK 脉冲（将频率增加 488.5 ppm）。
此功能应与 CALM 结合使用，后者在高分辨率下会降低日历的频率。如果输入频率为 32768 Hz，则在 32 秒窗口中增加的 RTCCLK 脉冲数按如下公式计算：(512 * CALP) - CALM。
请参见*第 42.3.12 节：RTC 精密数字校准*。

位 14 **CALW8**：使用 8 秒校准周期 (Use an 8-second calibration cycle period)
当 CALW8 置“1”时，选择 8 秒校准周期。
注： *CALW8=“1”时，CALM[1:0] 将始终保持为“00”。请参见第 42.3.12 节：RTC 精密数字校准。*

位 13 **CALW16**：使用 16 秒校准周期 (Use a 16-second calibration cycle period)
当 CALW16 置“1”时，选择 16 秒校准周期。如果 CALW8 = 1，则不能将该位置“1”。
注： *当 CALW16=“1”时，CALM[0] 将始终保持为“0”。请参见第 42.3.12 节：RTC 精密数字校准。*

位 12:9 保留，必须保持复位值

位 8:0 **CALM[8:0]**：负校准 (Calibration minus)
在 2^{20} 个 RTCCLK 脉冲内屏蔽 CALM 个脉冲（如果输入频率为 32768 Hz，则为 32 秒）来降低日历的频率。其分辨率为 0.9537 ppm。
要提高日历的频率，则应将此功能与 CALP 结合使用。请参见*第 1356 页的第 42.3.12 节：RTC 精密数字校准*。

42.6.16 RTC 入侵配置寄存器 (RTC_TAMPCR)

RTC tamper configuration register

偏移地址：0x40

Backup 域复位值：0x0000 0000

系统复位：不受影响

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	TAMP3 MF	TAMP3 NO ERASE	TAMP3 IE	TAMP2 MF	TAMP2 NO ERASE	TAMP2 IE	TAMP1 MF	TAMP1 NO ERASE	TAMP1 IE
							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
TAMP PUDIS	TAMPPRCH [1:0]		TAMPFLT[1:0]		TAMPFREQ[2:0]			TAMP TS	TAMP3 TRG	TAMP3 E	TAMP2 TRG	TAMP2 E	TAMPI E	TAMP1 TRG	TAMP1 E
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:25 保留，必须保持复位值。

位 24 **TAMP3MF**：入侵 3 屏蔽标志 (Tamper 3 mask flag)

0：入侵 3 事件生成触发事件，必须通过软件将 TAMP3F 清零来允许下一次入侵事件检测。

1：入侵 3 事件生成触发事件。屏蔽 TAMP3F 并由硬件在内部将其清零。不擦除备份寄存器。

注： TAMP3MF 置 1 时，不得使能入侵 3 中断。

位 23 **TAMP3NOERASE**：入侵 3 不擦除 (Tamper 3 no erase)

0：入侵 3 事件擦除备份寄存器。

1：入侵 3 事件不擦除备份寄存器。

位 22 **TAMP3IE**：入侵 3 中断使能 (Tamper 3 interrupt enable)

0：TAMPIE = 0 时禁止入侵 3 中断。

1：使能入侵 3 中断。

位 21 **TAMP2MF**：入侵 2 屏蔽标志 (Tamper 2 mask flag)

0：入侵 2 事件生成触发事件，如果要允许下一次入侵事件检测，必须通过软件将 TAMP2F 清零。

1：入侵 2 事件生成触发事件。屏蔽 TAMP2F 并由硬件在内部将其清零。不擦除备份寄存器。

注： TAMP2MF 置 1 时，不得使能入侵 2 中断。

位 20 **TAMP2NOERASE**：入侵 2 不擦除 (Tamper 2 no erase)

0：入侵 2 事件擦除备份寄存器。

1：入侵 2 事件不擦除备份寄存器。

位 19 **TAMP2IE**：入侵 2 中断使能 (Tamper 2 interrupt enable)

0：TAMPIE = 0 时禁止入侵 2 中断。

1：使能入侵 2 中断。

位 18 **TAMP1MF**：入侵 1 屏蔽标志 (Tamper 1 mask flag)

0：入侵 1 事件生成触发事件，必须通过软件将 TAMP1F 清零来允许下一次入侵事件检测。

1：入侵 1 事件生成触发事件。屏蔽 TAMP1F 并由硬件在内部将其清零。不擦除备份寄存器。

注： TAMP1MF 置 1 时，不得使能入侵 1 中断。

位 17 **TAMP1NOERASE**：入侵 1 不擦除 (Tamper 1 no erase)

0：入侵 1 事件擦除备份寄存器。

1：入侵 1 事件不擦除备份寄存器。

位 16 **TAMP1IE**：入侵 1 中断使能 (Tamper 1 interrupt enable)

0：TAMPIE = 0 时禁止入侵 1 中断。

1：使能入侵 1 中断。

位 15 **TAMPPUDIS**：RTC_TAMPx 上拉禁止 (RTC_TAMPx pull-up disable)

该位决定在每次采样之前是否对每个 RTC_TAMPx 引脚都进行预充电。

0：采样之前对 RTC_TAMPx 引脚进行预充电（使能内部上拉）。

1：禁止对 RTC_TAMPx 引脚进行预充电。

位 14:13 **TAMPPRCH[1:0]**：RTC_TAMPx 预充电持续时间 (RTC_TAMPx precharge duration)

这些位决定了在每次采样之前激活上拉的持续时间。TAMPPRCH 对每个 RTC_TAMPx 输入都有效。

0x0：1 个 RTCCLK 周期

0x1：2 个 RTCCLK 周期

0x2：4 个 RTCCLK 周期

0x3：8 个 RTCCLK 周期

位 12:11 **TAMPFLT[1:0]**：RTC_TAMPx 过滤器计数 (RTC_TAMPx filter count)

这些位决定了为激活入侵事件所需的指定电平 (TAMP*TRG) 上的连续采样次数。TAMPFLT 对每个 RTC_TAMPx 输入都有效。

0x0：在 RTC_TAMPx 输入转变为有效电平的边沿激活入侵事件（RTC_TAMPx 输入上无内部上拉）。

0x1：在有效电平上连续执行 2 次采样后激活入侵事件。

0x2：在有效电平上连续执行 4 次采样后激活入侵事件。

0x3：在有效电平上连续执行 8 次采样后激活入侵事件。

位 10:8 **TAMPFREQ[2:0]**：入侵采样频率 (Tamper sampling frequency)

决定对每个 RTC_TAMPx 输入进行采样时的频率。

0x0：RTCCLK/32768（RTCCLK = 32768 Hz 时为 1 Hz）

0x1：RTCCLK/16384（RTCCLK = 32768 Hz 时为 2 Hz）

0x2：RTCCLK/8192（RTCCLK = 32768 Hz 时为 4 Hz）

0x3：RTCCLK/4096（RTCCLK = 32768 Hz 时为 8 Hz）

0x4：RTCCLK/2048（RTCCLK = 32768 Hz 时为 16 Hz）

0x5：RTCCLK/1024（RTCCLK = 32768 Hz 时为 32 Hz）

0x6：RTCCLK/512（RTCCLK = 32768 Hz 时为 64 Hz）

0x7：RTCCLK/256（RTCCLK = 32768 Hz 时为 128 Hz）

位 7 **TAMPTS**：发生入侵检测事件时激活时间戳 (Activate timestamp on tamper detection event)

0：发生入侵检测事件时不保存时间戳

1：发生入侵检测事件时保存时间戳

即便 RTC_CR 寄存器中的 TSE=0，TAMPTS 仍有效。

位 6 **TAMP3TRG**：RTC_TAMP3 输入的有效电平 (Active level for RTC_TAMP3 input)

如果 TAMPFLT ≠ 00：

0：RTC_TAMP3 输入保持低电平会触发入侵检测事件。

1：RTC_TAMP3 输入保持高电平会触发入侵检测事件。

如果 TAMPFLT = 00：

0：RTC_TAMP3 输入上升沿会触发入侵检测事件。

1：RTC_TAMP3 输入下降沿会触发入侵检测事件。

位 5 **TAMP3E**：RTC_TAMP3 检测使能 (RTC_TAMP3 detection enable)
0：禁止 RTC_TAMP3 输入检测
1：使能 RTC_TAMP3 输入检测

位 4 **TAMP2TRG**：RTC_TAMP2 输入的有效电平 (Active level for RTC_TAMP2 input)
如果 TAMPFLT != 00：
0：RTC_TAMP2 输入保持低电平会触发入侵检测事件。
1：RTC_TAMP2 输入保持高电平会触发入侵检测事件。
如果 TAMPFLT = 00：
0：RTC_TAMP2 输入上升沿会触发入侵检测事件。
1：RTC_TAMP2 输入下降沿会触发入侵检测事件。

位 3 **TAMP2E**：RTC_TAMP2 输入检测使能 (RTC_TAMP2 input detection enable)
0：禁止 RTC_TAMP2 检测
1：使能 RTC_TAMP2 检测

位 2 **TAMPIE**：入侵中断使能 (Tamper interrupt enable)
0：禁止入侵中断
1：使能入侵中断
注： 该位使能所有入侵引脚事件的中断，无论 TAMPxIE 电平如何。如果将该位清零，可以通过将 TAMPxIE 置 1 分别使能每个入侵事件中断。

位 1 **TAMP1TRG**：RTC_TAMP1 输入的有效电平 (Active level for RTC_TAMP1 input)
如果 TAMPFLT != 00
0：RTC_TAMP1 输入保持低电平会触发入侵检测事件。
1：RTC_TAMP1 输入保持高电平会触发入侵检测事件。
如果 TAMPFLT = 00：
0：RTC_TAMP1 输入上升沿会触发入侵检测事件。
1：RTC_TAMP1 输入下降沿会触发入侵检测事件。

位 0 **TAMP1E**：RTC_TAMP1 输入检测使能 (RTC_TAMP1 input detection enable)
0：禁止 RTC_TAMP1 检测
1：使能 RTC_TAMP1 检测

注意： 如果 TAMPFLT = 0，则更改 TAMPxTRG 时必须复位 TAMPxE，以避免将 TAMPxF 意外置 1。

42.6.17 RTC 闹钟 A 亚秒寄存器 (RTC_ALRMASSR)

RTC alarm A sub second register

仅当 RTC_CR 中的 ALRAE 复位时或在初始化模式下，才可以对该寄存器执行写操作。

此寄存器受写保护。*第 1352 页的 RTC 寄存器写保护*中介绍了写访问的过程。

偏移地址：0x44

Backup 域复位值：0x0000 0000

系统复位：不受影响

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

位 31:28 保留，必须保持复位值。

位 27:24 MASKSS[3:0]：屏蔽从此位开始的最高有效位 (Mask the most-significant bits starting at this bit)

0：不对闹钟 A 的亚秒进行比较。当秒单元递增时设置闹钟（假定其余字段均匹配）。
1：在闹钟 A 比较中，SS[14:1] 为无关位。仅比较 SS[0]。
2：在闹钟 A 比较中，SS[14:2] 为无关位。仅比较 SS[1:0]。
3：在闹钟 A 比较中，SS[14:3] 为无关位。仅比较 SS[2:0]。
...
12：在闹钟 A 比较中，SS[14:12] 为无关位。比较 SS[11:0]。
13：在闹钟 A 比较中，SS[14:13] 为无关位。比较 SS[12:0]。
14：在闹钟 A 比较中，SS[14] 为无关位。比较 SS[13:0]。
15：所有 15 个 SS 位均进行比较，并且必须全部匹配才能激活闹钟。
同步计数器的溢出位（位 15）从不进行比较。仅当执行平移操作之后，此位才不为 0。

位 23:15 保留，必须保持复位值。

位 14:0 SS[14:0]：亚秒值 (Sub seconds value)

该值与同步预分频器计数器的内容进行比较以确定是否要激活闹钟 A。仅比较位 0 到 MASKSS-1。

42.6.18 RTC 闹钟 B 亚秒寄存器 (RTC_ALRMBSSR)

RTC alarm B sub second register

仅当 RTC_CR 中的 ALRBE 复位时或在初始化模式下，才可以对该寄存器执行写操作。

该寄存器受写保护。*RTC 寄存器写保护*一节中介绍了写访问的过程。

偏移地址：0x48

Backup 域复位值：0x0000 0000

系统复位：不受影响

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

位 31:28 保留，必须保持复位值。

位 27:24 **MASKSS[3:0]**：屏蔽从此位开始的最高有效位 (Mask the most-significant bits starting at this bit)

0x0：不对闹钟 B 的亚秒进行比较。当秒单元递增时设置闹钟（假定其余字段均匹配）。
0x1：在闹钟 B 比较中，SS[14:1] 为无关位。仅比较 SS[0]。
0x2：在闹钟 B 比较中，SS[14:2] 为无关位。仅比较 SS[1:0]。
0x3：在闹钟 B 比较中，SS[14:3] 为无关位。仅比较 SS[2:0]。
...
0xC：在闹钟 B 比较中，SS[14:12] 为无关位。比较 SS[11:0]。
0xD：在闹钟 B 比较中，SS[14:13] 为无关位。比较 SS[12:0]。
0xE：在闹钟 B 比较中，SS[14] 为无关位。比较 SS[13:0]。
0xF：所有 15 个 SS 位均进行比较，并且必须全部匹配才能激活闹钟。
同步计数器的溢出位（位 15）从不进行比较。仅当执行平移操作之后，此位才不为 0。

位 23:15 保留，必须保持复位值。

位 14:0 **SS[14:0]**：亚秒值 (Sub seconds value)

该值与同步预分频器计数器的内容进行比较以确定是否要激活闹钟 B。仅比较位 0 到 MASKSS-1。

42.6.19 RTC 选项寄存器 (RTC_OR)

RTC option register

偏移地址：0x4C

Backup 域复位值：0x0000 0000

系统复位：不受影响

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.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RTC_OUT_RMP	RTC_ALARM_TYPE
														rw	rw

位 31:2 保留，必须保持复位值。

位 1 **RTC_OUT_RMP**：RTC_OUT 重映射 (RTC_OUT remap)
将此位置 1 可将 RTC 输出重映射到 PB2，具体如下：
RTC_OUT_RMP = “0”：
如果 OSEL/= “00”：RTC_ALARM 输出到 PC13
如果 OSEL= “00”且 COE = “1”：RTC_CALIB 输出到 PC13
RTC_OUT_RMP = “1”：
如果 OSEL /= “00”且 COE = “0”：RTC_ALARM 输出到 PB2
如果 OSEL = “00”且 COE = “1”：RTC_CALIB 输出到 PB2
如果 OSEL /= “00”且 COE = “1”：RTC_CALIB 输出到 PB2 且 RTC_ALARM 输出到 PC13。

位 0 **RTC_ALARM_TYPE**：PC13 上的 RTC_ALARM 输出类型 (RTC_ALARM output type on PC13)
此位由软件置 1 和清零
0：RTC_ALARM 在映射到 PC13 上时为开漏输出
1：RTC_ALARM 在映射到 PC13 上时为推挽输出

42.6.20 RTC 备份寄存器 (RTC_BKPxR)

RTC backup registers

偏移地址：0x50 到 0xCC

Backup 域复位值：0x0000 0000

系统复位：不受影响

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

位 31:0 BKP[31:0]
应用可向/从这些寄存器写入/读取数据。
当 V_{DD} 关闭时，这些寄存器由 V_{BAT} 供电，因而系统复位时，这些寄存器不会复位，并且当器件在低功耗模式下工作时，寄存器的内容仍然有效。
发生入侵检测事件时该寄存器会被复位，并且只要 TAMPxF=1，该寄存器就一直保持复位。

42.6.21 RTC 寄存器映射

表 245. RTC 寄存器映射和复位值

偏移	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	**RTC_TR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PM	HT[1:0]		HU[3:0]				Res.	MNT[2:0]			MNU[3:0]				Res.	ST[2:0]			SU[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
0x04	**RTC_DR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	YT[3:0]				YU[3:0]				WDU[2:0]			MT	MU[3:0]				Res.	Res.	DT[1:0]		DU[3:0]			
	Reset value									0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1			0	0	0	0	0	1
0x08	**RTC_CR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ITSE	COE	OSEL[1:0]		POL	COSEL	BKP	SUB1H	ADD1H	TSIE	WUTIE	ALRBIE	ALRAIE	TSE	WUTE	ALRBE	ALRAE	Res.	FMT	BYPSHAD	REFCKON	TSEDGE	WUCKSEL[2:0]		
	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	**RTC_ISR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ITSF	RECALPF	TAMP3F	TAMP2F	TAMP1F	TSOVF	TSF	WUTF	ALRBF	ALRAF	INIT	INITF	RSF	INITS	SHPF	WUT WF	ALRBWF	ALRAWF
	Reset value															0	0		0	0	0	0	0	0	0	0	0	0	0	0	1	1	1
0x10	**RTC_PRER**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREDIV_A[6:0]							PREDIV_S[14:0]															
	Reset value										1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
0x14	**RTC_WUTR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WUT[15:0]															
	Reset value																	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0x1C	**RTC_ALRMAR**	MSK4	WDSEL	DT[1:0]		DU[3:0]				MSK3	PM	HT[1:0]		HU[3:0]				MSK2	MNT[2:0]			MNU[3:0]				MSK1	ST[2:0]			SU[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
0x20	**RTC_ALRMBR**	MSK4	WDSEL	DT[1:0]		DU[3:0]				MSK3	PM	HT[1:0]		HU[3:0]				MSK2	MNT[2:0]			MNU[3:0]				MSK2	ST[2:0]			SU[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	**RTC_WPR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	KEY							
	Reset value																									0	0	0	0	0	0	0	0
0x28	**RTC_SSR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SS[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x2C	**RTC_SHIFTR**	ADD1S	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SUBFS[14:0]														
	Reset value	0																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

表 245. RTC 寄存器映射和复位值（续）

偏移	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
0x30	**RTC_TSTR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PM	HT[1:0]		HU[3:0]				Res.	MNT[2:0]			MNU[3:0]				Res.	ST[2:0]			SU[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
0x34	**RTC_TSDR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WDU[1:0]			MT	MU[3:0]				Res.	Res.	DT[1:0]		DU[3:0]			
	Reset value																	0	0	0	0	0	0	0	0			0	0	0	0	0	0
0x38	**RTC_TSSSR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SS[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x3C	**RTC_ CALR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CALP	CALW8	CALW16	Res.	Res.	Res.	Res.	CALM[8:0]								
	Reset value																	0	0	0					0	0	0	0	0	0	0	0	0
0x40	**RTC_TAMPCR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TAMP3MF	TAMP3NOERASE	TAMP3IE	TAMP2MF	TAMP2NOERASE	TAMP2IE	TAMP1MF	TAMP1NOERASE	TAMP1IE	TAMPPUDIS	TAMPPRCH[1:0]		TAMPFLT[1:0]		TAMPFREQ[2:0]			TAMPTS	TAMP3TRG	TAMP3E	TAMP2TRG	TAMP2E	TAMPIE	TAMP1TRG	TAMP1E
	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
0x44	**RTC_ ALRMASSR**	Res.	Res.	Res.	Res.	MASKSS [3:0]				Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SS[14:0]														
	Reset value					0	0	0	0										0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x48	**RTC_ ALRMBSSR**	Res.	Res.	Res.	Res.	MASKSS [3:0]				Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SS[14:0]														
	Reset value					0	0	0	0										0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x4C	**RTC_ OR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RTC_OUT_RMP	RTC_ALARM_TYPE
	Reset value																															0	0
0x50 to 0xCC	**RTC_BKP0R**	BKP[31:0]																															
	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
	to **RTC_BKP31R**	BKP[31:0]																															
	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*第 78 页的第 2.2.2 节*。

43 内部集成电路 (I2C) 接口

43.1 简介

I^2C（内部集成电路）总线接口处理微控制器与串行 I^2C 总线间的通信。它提供多主模式功能，可以控制所有 I^2C 总线特定的序列、协议、仲裁和时序。它支持标准模式 (Sm)、快速模式 (Fm) 和超快速模式 (Fm+)。

它还与 SMBus（系统管理总线）和 PMBus（电源管理总线）兼容。

可使用 DMA 来减轻 CPU 的工作量。

43.2 I2C 主要特性

- 兼容 I^2C 总线规范第 03 版：
  - 从模式和主模式
  - 多主模式功能
  - 标准速度模式（高达 100 kHz）
  - 快速模式（高达 400 kHz）
  - 超快速模式（高达 1 MHz）
  - 7 位和 10 位寻址模式
  - 多个 7 位从地址（2 个从设备地址寄存器，1 个具有可配置的掩码位段）
  - 所有 7 位地址应答模式
  - 广播呼叫
  - 总线上的数据建立和保持时间可软件配置
  - 方便易用的事件管理
  - 可选的时钟延长
  - 软件复位
- 带 DMA 功能的 1 字节缓冲
- 可编程模拟和数字噪声滤波器

还可额外提供以下特性，具体取决于产品实现（请参见*第 43.3 节：I2C 特性实现*）：

- 兼容 SMBus 规范第 3.0 版：
  - 具有 ACK 控制的硬件 PEC（数据包错误校验）生成和验证
  - 命令和数据应答控制
  - 支持地址解析协议 (ARP)
  - 支持主机和从设备
  - SMBus 报警
  - 超时和空闲条件检测
- 兼容 PMBus 第 1.3 版标准
- 独立时钟：选择独立时钟源可使 I2C 通信速度不受 PCLK 时钟频率更改的影响
- 地址匹配时从停止模式唤醒

43.3 I2C 特性实现

本手册介绍了 I2C 外设中实现的所有特性。在 STM32L4Rxxx 和 STM32L4Sxxx 器件中，I2C1、I2C2、I2C3 和 I2C4 实现了下表所示的全部功能，其中只有 I2C3 可以从停止 2 模式唤醒。

表 246. I2C 特性实现

I2C 特性(1)	I2C1	I2C2	I2C3	I2C4
7 位寻址模式	X	X	X	X
10 位寻址模式	X	X	X	X
标准速度模式（高达 100 kbps）	X	X	X	X
快速模式（高达 400 kbps）	X	X	X	X
超快速模式 20mA 输出驱动 I/O（高达1 Mb/s）	X	X	X	X
独立时钟	X	X	X	X
从停止 1 模式唤醒	X	X	X	X
从停止 2 模式唤醒	-	-	X	-
SMBus/PMBus	X	X	X	X

1. X = 支持。

43.4 I2C 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^2C 总线。它可以连接到标准速度（高达 100 kHz）、快速（高达 400 kHz）或超快速（高达 1 MHz）I^2C 总线。

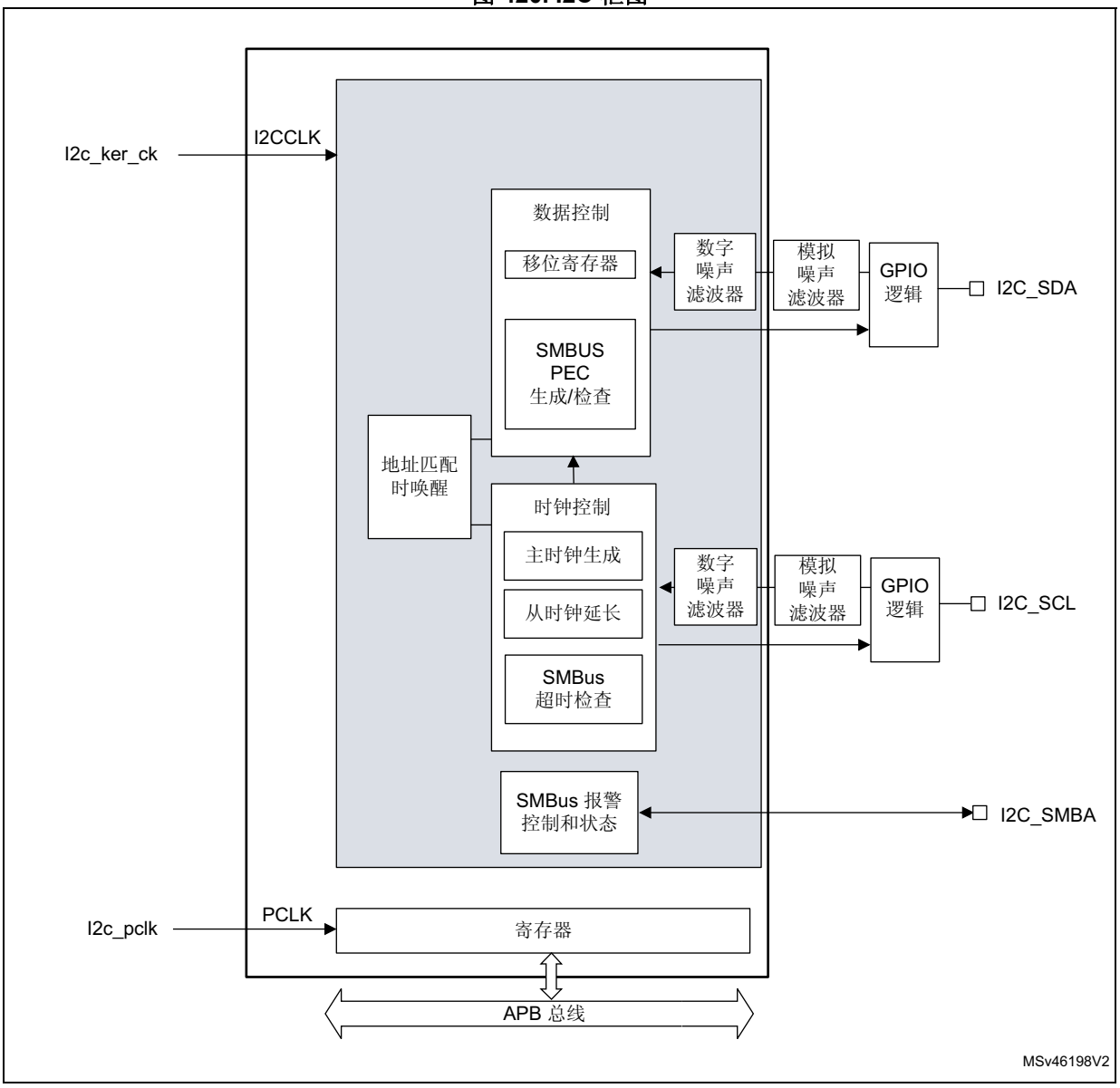
该接口也可通过数据引脚 (SDA) 和时钟引脚 (SCL) 连接到 SMBus。

如果支持 SMBus 功能：还可使用额外的可选 SMBus 报警引脚 (SMBA)。

43.4.1 I2C 框图

I2C 接口如*图 420* 所示。

图 420. I2C 框图



I2C 的时钟由独立时钟源提供，这使得 I2C 能够独立于 PCLK 频率工作。

该独立时钟源可从以下三个时钟源中选择：

- PCLK1：APB1 时钟（默认值）
- HSI16：内部 16 MHz RC 振荡器
- SYSCLK：系统时钟

更多详细信息，请参见*第 6 节：复位和时钟控制 (RCC)*。

对于支持以 20 mA 输出电流驱动超快速模式操作的 I2C I/O，可以通过系统配置控制器 (SYSCFG) 中的控制位使能驱动功能。*请参见第 43.3 节：I2C 特性实现*。

43.4.2 I2C 时钟要求

I2C 内核的时钟由 I2CCLK 提供。

I2CCLK 周期 t_{I2CCLK} 必须遵循以下条件：

$t_{I2CCLK} < (t_{LOW} - t_{filters}) / 4$ 且 $t_{I2CCLK} < t_{HIGH}$

其中：

t_{LOW}：SCL 低电平时间；t_{HIGH}：SCL 高电平时间。

$t_{filters}$：滤波器使能时，该值为模拟滤波器和数字滤波器引入的延时总和。

模拟滤波器延时最大值为 260 ns。数字滤波器延时为 DNF x t_{I2CCLK}。

PCLK 时钟周期 t_{PCLK} 必须遵循以下条件：

$t_{PCLK} < 4/3\ t_{SCL}$

其中，t_{SCL}：SCL 周期

注意： 当 I2C 内核的时钟由 PCLK 提供时，该时钟必须遵循 t_{I2CCLK} 的条件。

43.4.3 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

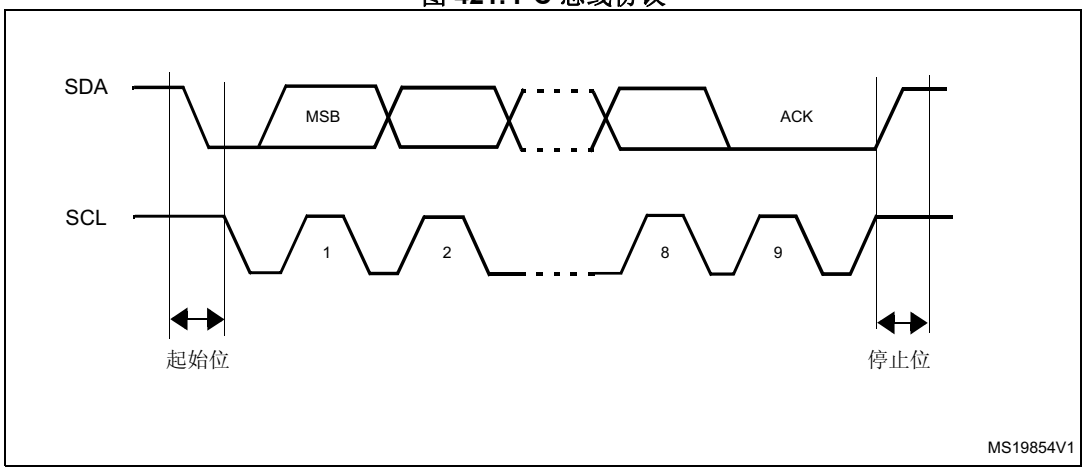
在主模式下，I2C 接口会启动数据传输并生成时钟信号。串行数据传输始终是在出现起始位时开始，在出现停止位时结束。起始位和停止位均在主模式下由软件生成。

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。保留的 SMBus 地址也可由软件使能。

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。请参见下图。

图 421. I²C 总线协议



应答位可由软件使能或禁止。I2C 接口地址可通过软件进行选择。

43.4.4 I2C 初始化

使能和禁止外设

I2C 外设时钟必须在时钟控制器中进行配置和使能（请参见*第 6 节：复位和时钟控制 (RCC)*）。

然后可通过将 I2C_CR1 寄存器中的 PE 位置 1 使能 I2C。

当禁止 I2C (PE=0) 时，I²C 将执行软件复位。更多详细信息，请参见*第 43.4.5 节：软件复位*。

噪声滤波器

通过将 I2C_CR1 寄存器中的 PE 位置 1 来使能 I2C 外设之前，如有必要，用户必须配置噪声滤波器。默认情况下，SDA 和 SCL 输入上集成了模拟噪声滤波器。该模拟滤波器符合 I²C 规范，此规范要求在快速模式和超快速模式下对脉宽在 50ns 以下的脉冲都要抑制。用户可通过将 ANFOFF 位置 1 来禁止该模拟滤波器，通过配置 I2C_CR1 寄存器中的 DNF[3:0] 位来选择数字滤波器。

使能数字滤波器时，SCL 或 SDA 线的电平只有在电平稳定时间超过 DNF x I2CCLK 个周期后才会发生内部变化。从而可抑制的尖峰脉宽在 1 到 15 个 I2CCLK 周期可编程。

表 247. 模拟滤波器与数字滤波器对比

-	模拟滤波器	数字滤波器
抑制的脉冲宽度	≥ 50 ns	从 1 到 15 个 I2C 外设时钟的可编程长度
优点	停止模式中仍可用	– 长度可编程：额外的滤波能力与标准要求 – 稳定长度
缺点	随温度、电压和过程变化会发生变化	当使能数字滤波器后，无法在地址匹配时从停止模式唤醒。

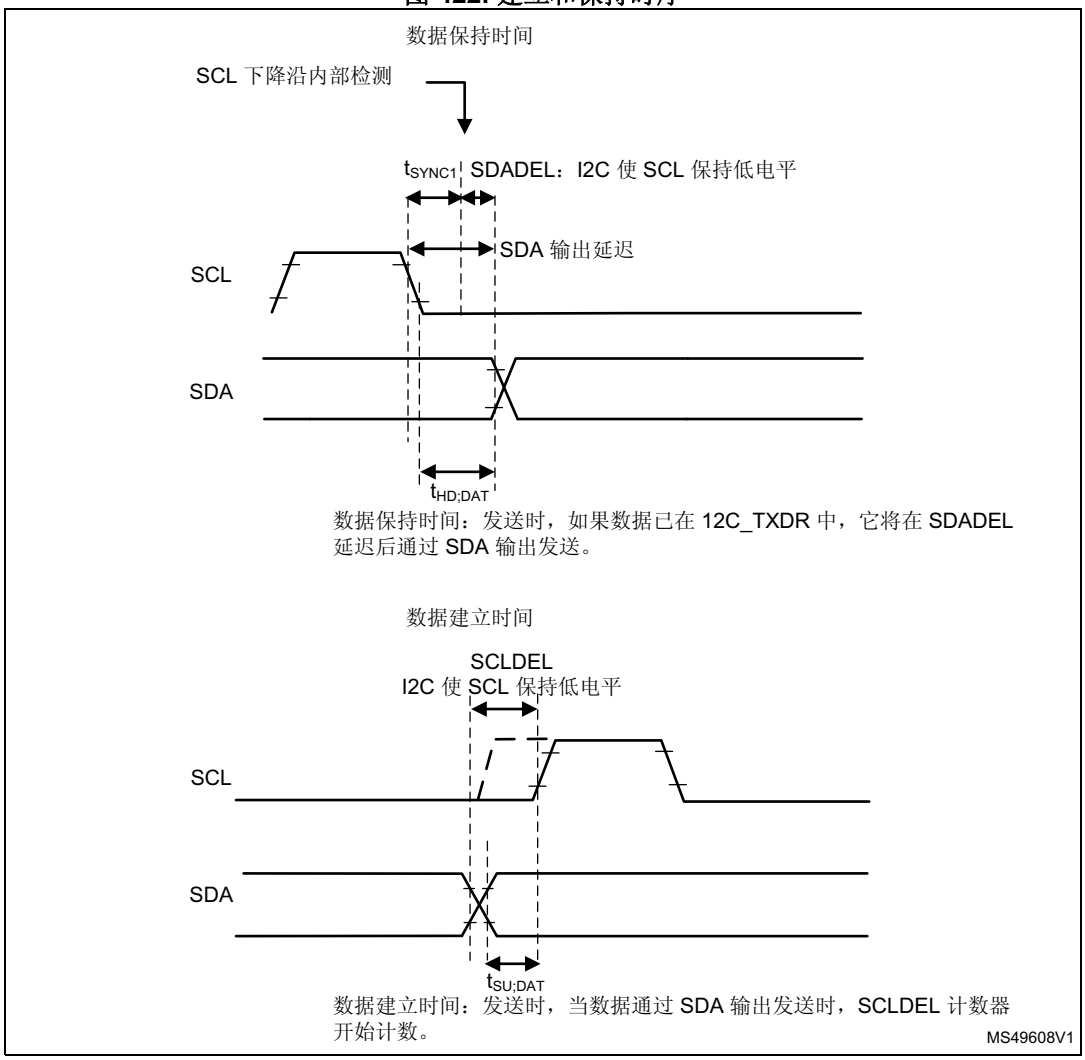
注意： 使能 I2C 时，不允许更改滤波器配置。

I2C 时序

必须配置时序，以便保证主模式和从模式下使用正确的数据保持和建立时间。配置方法是编程 I2C_TIMINGR 寄存器中的 PRESC[3:0]、SCLDEL[3:0] 和 SDADEL[3:0] 位。

STM32CubeMX 工具进行计算并将 I2C_TIMINGR 内容提供在 I2C 配置窗口中。

图 422. 建立和保持时序



- 当内部检测到 SCL 下降沿时，会在发送 SDA 输出之前插入一段延时。该延时为 t_{SDADEL} = SDADEL x t_{PRESC} + t_{I2CCLK}，其中 t_{PRESC} = (PRESC+1) x t_{I2CCLK}。
  T_{SDADEL} 会影响保持时间 $t_{HD;DAT}$。

SDA 总输出延时为：

t_{SYNC1} + {[SDADEL x (PRESC+1) + 1] x t_{I2CCLK} }

t_{SYNC1} 持续时间取决于以下参数：

- SCL 下降斜率
- 模拟滤波器（使能时）引入的输入延时：$t_{AF(min)}$ < t_{AF} < $t_{AF(max)}$ ns
- 数字滤波器（使能时）引入的输入延时：t_{DNF} = DNF x t_{I2CCLK}
- SCL 与 I2CCLK 时钟建立同步而产生的延时（2 到 3 个 I2CCLK 周期）

为了桥接 SCL 下降沿的未定义区域，用户编程 SDADEL 时必须遵循以下条件：

{$t_{f\ (max)}$ +$t_{HD;DAT\ (min)}$ -$t_{AF(min)}$ - [(*DNF* +3) x t_{I2CCLK}]} / {(PRESC +1) x t_{I2CCLK}} ≤ SDADEL

SDADEL ≤ {$t_{HD;DAT\ (max)}$ -$t_{AF(max)}$ - [(*DNF+4) x* t_{I2CCLK}]} / {(PRESC +1) x t_{I2CCLK}}

注： *只有使能模拟滤波器时，公式中才包含 $t_{AF(min)}$ / $t_{AF(max)}$。有关 t_{AF} 值的信息，请参见器件数据手册。*

标准模式、快速模式和超快速模式下的 $t_{HD;DAT}$ 最大值分别可达 3.45 µs、0.9 µs 和 0.45 µs，但必须小于 $t_{VD;DAT}$ 最大值（差值为跳变时间）。只有器件未延长 SCL 信号的低电平周期 (t_{LOW}) 时，才必须满足该最大值条件。如果时钟延长 SCL，数据必须在建立时间内保持有效，之后才能释放时钟。

SDA 上升沿通常为最坏情况，因此在这种情况下，上述公式变成如下形式：

SDADEL ≤ {$t_{VD;DAT\ (max)}$ -$t_{r\ (max)}$ -*260 ns* - [(*DNF+4) x* t_{I2CCLK}]} / {(PRESC +1) x t_{I2CCLK}}。

注： *NOSTRETCH=0 时会违反该条件，这是因为器件会根据 SCLDEL 值来延长 SCL 低电平时间，以保证建立时间。*

有关 t_f、t_r、$t_{HD;DAT}$ 和 $t_{VD;DAT}$ 标准值的信息，请参见*表 248：I2C-SMBUS 规范数据建立和保持时间*。

- 在 t_{SDADEL} 延时后，或在因数据未写入 I2C_TXDR 寄存器而导致从器件必须延长时钟的情况下发送 SDA 输出后，SCL 线会在建立时间内保持低电平。该建立时间为 t_{SCLDEL} = (SCLDEL+1) x t_{PRESC}，其中 t_{PRESC} = (PRESC+1) x t_{I2CCLK}。
  t_{SCLDEL} 会影响建立时间 $t_{SU;DAT}$。

为了桥接 SDA 跳变（上升沿通常为最坏情况）的未定义区域，编程 SCLDEL 时必须遵循以下条件：

{[$t_{r\ (max)}$ + $t_{SU;DAT\ (min)}$] / [(PRESC+1)] x t_{I2CCLK}]} - 1 <= SCLDEL

有关 t_r 和 $t_{SU;DAT}$ 标准值的信息，请参见*表 248：I2C-SMBUS 规范数据建立和保持时间*。

将使用的 SDA 和 SCL 跳变时间值就是应用中的值。使用最大值而非标准值会增加 SDADEL 和 SCLDEL 计算的约束条件，但能够确保任意应用的特性。

注： *在发送和接收模式下，对于每个时钟脉冲，检测到 SCL 下降沿后，I2C 主器件或从器件会至少在 [(SDADEL+SCLDEL+1) x (PRESC+1) + 1] x t_{I2CCLK} 期间内延长 SCL 低电平时间。在发送模式下，如果 SDADEL 计数器计数结束后数据还未写入 I2C_TXDR，则 I2C 会继续延长 SCL 低电平时间，直到写入下一个数据。随后，会将新数据 MSB 发送到 SDA 输出，SCLDEL 计数器将开始计数，同时会继续延长 SCL 低电平时间以确保提供充足的数据建立时间。*

如果从模式下 NOSTRETCH = 1，则 SCL 不会延长。因此，编程 SDADEL 时还必须确保提供充足的建立时间。

表 248. I²C-SMBUS 规范数据建立和保持时间

符号	参数	标准模式 (Sm)		快速模式 (Fm)		超快速模式 (Fm)		SMBus		单位
		最小值	最大值	最小值	最大值	最小值	最大值	最小值	最大值	
$t_{HD;DAT}$	数据保持时间	0	-	0	-	0	-	0.3	-	µs
$t_{VD;DAT}$	数据有效时间	-	3.45	-	0.9	-	0.45	-	-	
$t_{SU;DAT}$	数据建立时间	250	-	100	-	50	-	250	-	ns
t_r	SDA 和 SCL 信号的上升时间	-	1000	-	300	-	120	-	1000	
t_f	SDA 和 SCL 信号的下降时间	-	300	-	300	-	120	-	300	

此外，在主模式下，必须通过编程 I2C_TIMINGR 寄存器中的 PRESC[3:0]、SCLH[7:0] 和 SCLL[7:0] 位来配置 SCL 时钟的高电平和低电平。

- 当内部检测到 SCL 下降沿时，会在释放 SCL 输出之前插入一段延时。该延时为 t_{SCLL} = (SCLL+1) x t_{PRESC}，其中 t_{PRESC} = (PRESC+1) x t_{I2CCLK}。
  t_{SCLL} 会影响 SCL 低电平时间 t_{LOW}。
- 当内部检测到 SCL 上升沿时，会在将 SCL 输出强制为低电平之前插入一段延时。该延时为 t_{SCLH} = (SCLH+1) x t_{PRESC}，其中 t_{PRESC} = (PRESC+1) x t_{I2CCLK}。t_{SCLH} 会影响 SCL 高电平时间 t_{HIGH}。

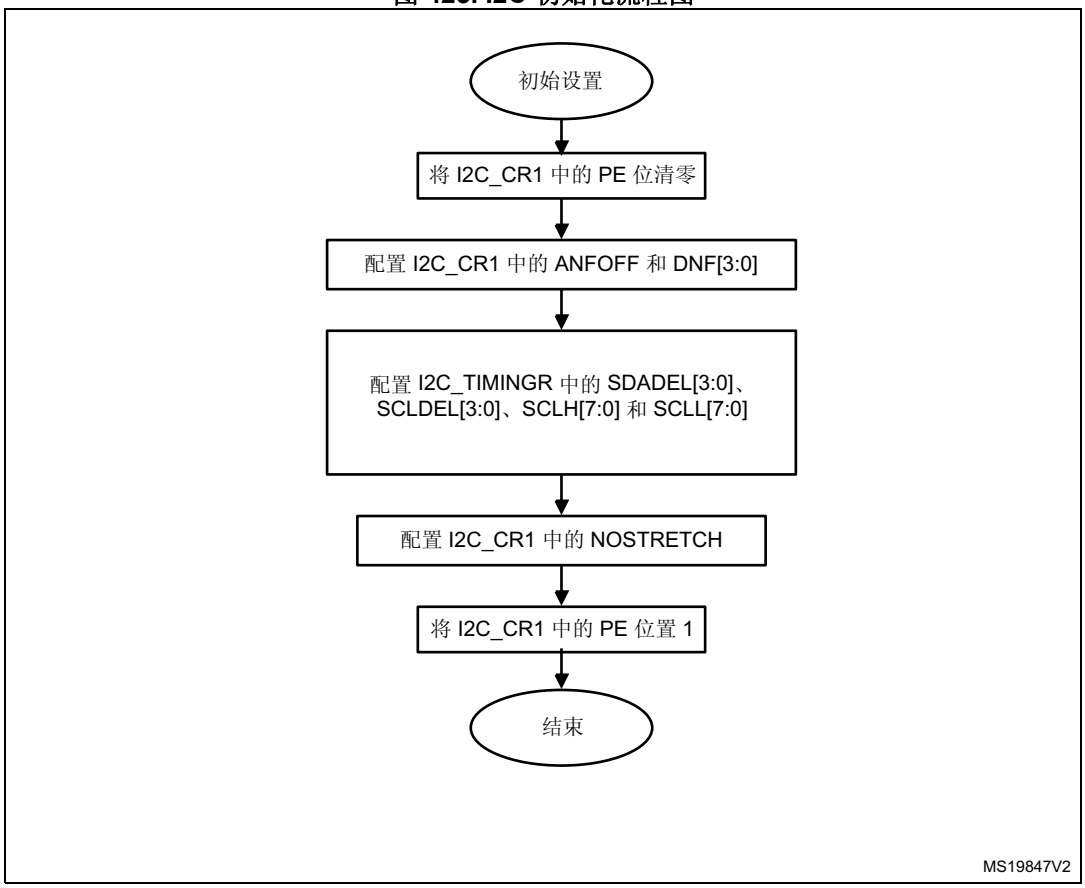
更多详细信息，请参见 *I2C 主模式初始化*。

注意： 使能 I2C 后，不允许更改时序配置。

此外，还必须在使能 I2C 从设备前，对 NOSTRETCH 进行配置。更多详细信息，请参见 *I2C 从模式初始化*。

注意： 使能 I2C 后，不允许更改 NOSTRETCH 配置。

图 423. I2C 初始化流程图



43.4.5 软件复位

可通过将 I2C_CR1 寄存器中的 PE 位清零来执行软件复位。在这种情况下，I2C 线 SCL 和 SDA 被释放。内部状态机复位，通信控制位和状态位恢复为其复位值。配置寄存器不受影响。

下面列出了受影响的寄存器位：

1. I2C_CR2 寄存器：START、STOP 和 NACK
2. I2C_ISR 寄存器：BUSY、TXE、TXIS、RXNE、ADDR、NACKF、TCR、TC、STOPF、BERR、ARLO 和 OVR

支持 SMBus 功能时还会影响到以下寄存器位：

1. I2C_CR2 寄存器：PECBYTE
2. I2C_ISR 寄存器：PECERR、TIMEOUT 和 ALERT

必须使 PE 保持低电平持续至少 3 个 APB 时钟周期，才能成功执行软件复位。使用以下软件写序列可确保这一点：- 写入 PE=0 - 检查 PE=0 - 写入 PE=1。

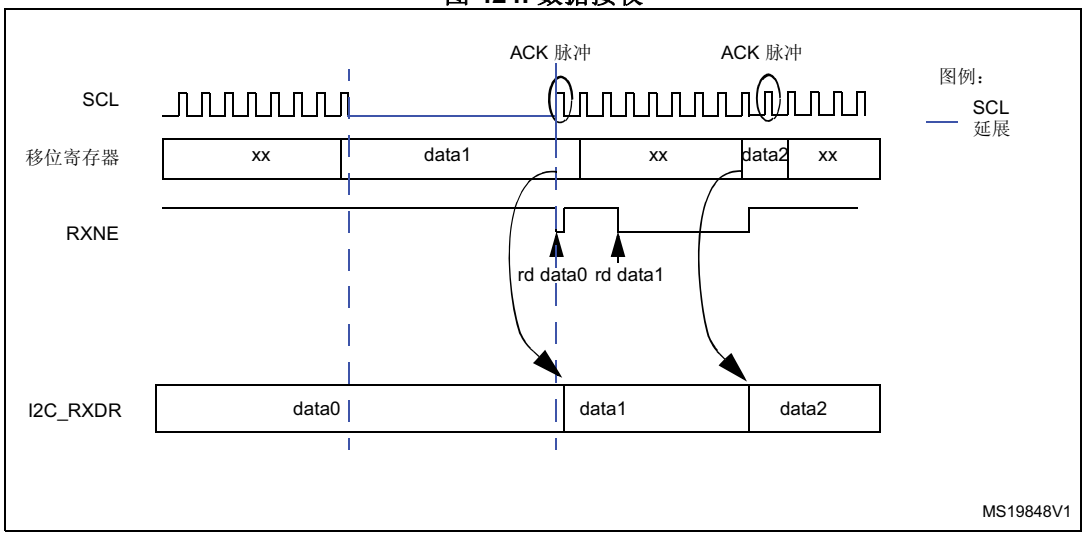
43.4.6 数据传输

数据传输由发送和接收数据寄存器以及移位寄存器来管理。

接收

SDA 输入填充移位寄存器。在第 8 个 SCL 脉冲后（接收到完整的数据字节时），如果 I2C_RXDR 寄存器为空 (RXNE=0)，则移位寄存器的内容会复制到其中。如果 RXNE=1（意味着尚未读取上一次接收到的数据字节），则将延长 SCL 线的低电平时间，直到读取了 I2C_RXDR 为止。在第 8 个和第 9 个 SCL 脉冲之间（应答脉冲之前）插入一段延长的时间。

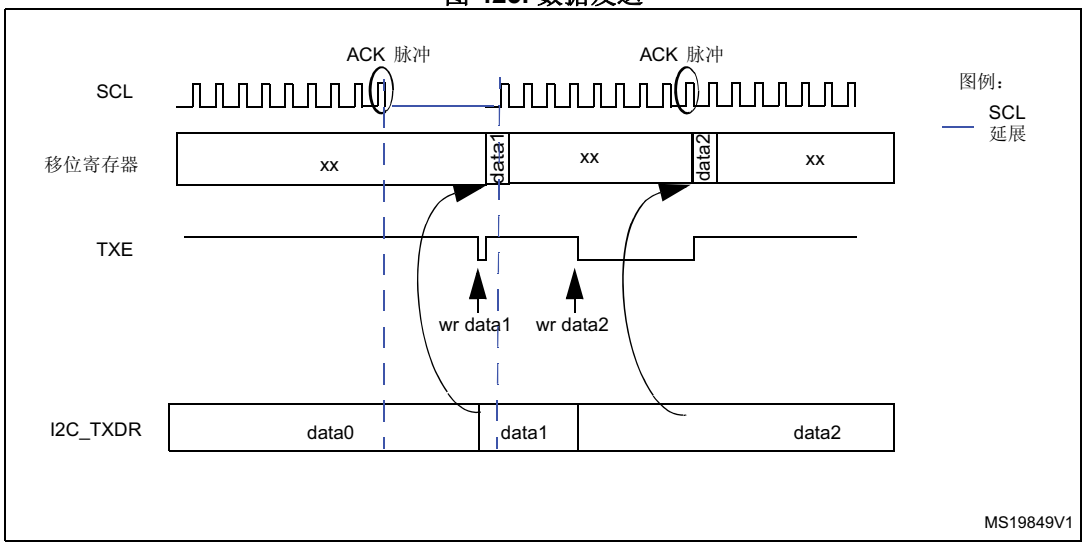
图 424. 数据接收



发送

如果 I2C_TXDR 寄存器不为空 (TXE=0)，则其内容会在第 9 个 SCL 脉冲（应答脉冲）后复制到移位寄存器中。然后移位寄存器的内容会移出到 SDA 线上。如果 TXE=1（意味着 I2C_TXDR 内尚未写入任何数据），则将延长 SCL 线的低电平时间，直到写入了 I2C_TXDR 为止。在第 9 个 SCL 脉冲后进行延长。

图 425. 数据发送



硬件传输管理

I2C 在硬件中内置了字节计数器，以便在下列各种模式下管理字节传输和结束通信：

- 主模式下生成 NACK、STOP 和 ReSTART
- 从接收器模式下控制 ACK 是否发出
- SMBus 模式下生成/校验 PEC

字节计数器通常在主模式下使用。在从模式下，字节计数器默认为禁止状态，但可以通过软件来使能，方法是将 I2C_CR2 寄存器中的 SBC（从字节控制）位置 1。

待传输的字节数在 I2C_CR2 寄存器的 NBYTES[7:0] 位域中进行编程。如果待传输的字节数 (NBYTES) 大于 255，或者接收方希望控制是否对接收到的数据字节进行应答，则必须选择重载模式，方法是将 I2C_CR2 寄存器的 RELOAD 位置 1。在该模式下，完成 NBYTES 中所编程字节数的数据传输之后，TCR 标志将置 1，并且 TCIE 置 1 时将生成中断。只要 TCR 标志置 1，SCL 便会延长。当往 NBYTES 写入一个非零值时，TCR 由软件清零。

在往 NBYTE 中设置最后一次传输的字节数前，必须把 RELOAD 位清零。

当主模式下 RELOAD=0 时，可在以下 2 种模式下使用计数器：

- **自动结束模式**（I2C_CR2 寄存器中的 AUTOEND = “1”）。在该模式下，一旦完成 NBYTES[7:0] 位域中所编程字节数的数据传输，主器件便会自动发送停止位。
- **软件结束模式**（I2C_CR2 寄存器中的 AUTOEND = “0”）。在该模式下，一旦完成 NBYTES[7:0] 位域中所编程字节数的数据传输，TC 标志将置 1，并且 TCIE 置 1 时将生成中断。只要 TC 标志置 1，SCL 信号便会延长，需要软件介入操作。当软件把 I2C_CR2 寄存器中的起始位或停止位置 1 时，TC 标志将被清零。当主器件要发送重复起始位时，必须使用该模式。

注意： 当 RELOAD 位置 1 时，AUTOEND 位将不起作用。

表 249. I2C 配置

功能	SBC 位	RELOAD 位	AUTOEND 位
主 Tx/Rx NBYTES + STOP	x	0	1
主 Tx/Rx + NBYTES + RESTART	x	0	0
从 Tx/Rx 接收的所有字节都要回复应答	0	x	x
具有 ACK 控制的从 Rx	1	1	x

43.4.7 I2C 从模式

I2C 从模式初始化

要在从模式下工作，用户必须至少使能一个从地址。可使用 I2C_OAR1 和 I2C_OAR2 这两个寄存器来编程自身从地址 OA1 和 OA2。

- OA1 既可配置为 7 位寻址模式（默认），也可通过将 I2C_OAR1 寄存器的 OA1MODE 位置 1 配置为 10 位寻址模式。

  通过将 I2C_OAR1 寄存器中的 OA1EN 位置 1 来使能 OA1。
- 如果需要额外的从地址，可配置第 2 个从地址 OA2。将 I2C_OAR2 寄存器的 OA2MSK[2:0] 位置 1 最多可屏蔽 7 个 OA2 LSB。因此，当 OA2MSK 配置为 1 到 6 时，将分别只有 OA2[7:2]、OA2[7:3]、OA2[7:4]、OA2[7:5]、OA2[7:6] 或 OA2[7] 与接收到的地址作比较。只要 OA2MSK 不等于 0， OA2 的地址比较器便会排除 I2C 保留地址 （0000 XXX 和 1111 XXX），这些地址将不会得到应答。如果 OA2MSK=7，接收到的所有 7 位地址（保留地址除外）均得到应答。 OA2 始终为 7 位地址。

  如果这些保留地址在 I2C_OAR1 或 I2C_OAR2 寄存器中进行了编程并且 OA2MSK=0，则它们可以在通过特定使能位使能后得到应答。

  通过将 I2C_OAR2 寄存器中的 OA2EN 位置 1 来使能 OA2。
- 通过将 I2C_CR1 寄存器中的 GCEN 位置 1 来使能广播呼叫地址。

当通过 I2C 的其中一个使能地址来寻址到该 I2C 设备时，ADDR 中断状态标志将置 1，并且 ADDRIE 位置 1 时将生成中断。

默认情况下，从器件使用其时钟延长功能（即必要时延长 SCL 信号的低电平时间）来为软件操作的执行提供时机。如果主器件不支持时钟延长，则必须对 I2C 进行如下配置：将 I2C_CR1 寄存器中 NOSTRETCH 位置 1。

接收到 ADDR 中断后，如果使能多个地址，则用户必须读取 I2C_ISR 寄存器中的 ADDCODE[6:0] 位，以确定是哪个地址匹配。还必须检查 DIR 标志，以获悉传输方向。

带时钟延长的从模式 (NOSTRETCH = 0)

在默认模式下，I2C 从器件会在以下情况下延长 SCL 时钟：

- ADDR 标志置 1 时：接收到的地址与其中一个使能的从地址匹配。通过软件将 ADDRCF 位置 1 以清零 ADRR 标志时，将释放该时钟延展。
- 发送时，前一次数据传输已完成但 I2C_TXDR 寄存器中未写入任何新数据，或者 ADDR 标志清零 (TXE=1) 时未写入第一个数据字节。往 I2C_TXDR 寄存器中写入数据时，将释放该时钟延展。
- 接收时，尚未读取 I2C_RXDR 寄存器但新的数据接收已完成。读取 I2C_RXDR 时，将释放该时钟延展。
- 当从器件字节控制模式和重载模式（SBC=1 且 RELOAD=1）下 TCR = 1 时，这意味着最后一个数据字节已完成传输。通过向 NBYTES[7:0] 字段写入一个非零值以将 TCR 清零时，将释放该时钟延展。
- 在 SCL 下降沿检测之后，I2C 会延长 SCL 的低电平时间（不超过 $[(SDADEL+SCLDEL+1) \times (PRESC+1) + 1] \times t_{I2CCLK}$）。

不带时钟延长的从模式 (NOSTRETCH = 1)

当 I2C_CR1 寄存器中的 NOSTRETCH = 1 时，I2C 从器件不会延长 SCL 信号。

- ADDR 标志置 1 时，不会延长 SCL 时钟。
- 发送时，必须在与发送数据对应的第一个 SCL 脉冲出现之前，向 I2C_TXDR 寄存器写入数据。否则，会发生下溢，I2C_ISR 寄存器中的 OVR 标志将置 1，如果 I2C_CR1 寄存器中的 ERRIE 位置 1，还将生成中断。当第一次数据发送开始而 STOPF 位仍置 1（尚未清零）时，OVR 标志也将置 1。因此，如果写入下一次传输要发送的第一个数据后才清零上一次传输的 STOPF 标志，则会出现 OVR 状态，甚至对于待发送的第一个数据也是如此。
- 接收时，必须在下一个数据字节的第 9 个 SCL 脉冲（ACK 脉冲）出现之前，从 I2C_RXDR 寄存器读取数据。否则，会发生上溢， I2C_ISR 寄存器中的 OVR 标志将置 1，如果 I2C_CR1 寄存器中的 ERRIE 位置 1，还将生成中断。

从器件字节控制模式

要在从接收模式下实现字节 ACK 控制，必须通过将 I2C_CR1 寄存器中的 SBC 位置 1 来使能从器件字节控制模式。这样符合 SMBus 标准。

要在从接收模式下实现字节 ACK 控制，必须选择重载模式 (RELOAD=1)。要控制每个字节，必须在 ADDR 中断子程序中将 NBYTES 初始化为 0x1，并在每接收一个字节后将 NBYTE 重载为 0x1。接收到字节后，TCR 位将置 1，从而延长 SCL 信号的第 8 个和第 9 个脉冲之间的低电平时间。用户可以从 I2C_RXDR 寄存器中读取数据，然后通过配置 I2C_CR2 寄存器中的 ACK 位来决定是否应答。通过将 NBYTES 编程为非零值来释放 SCL 延长：发送应答或不应答信号，然后可继续接收下一个字节。

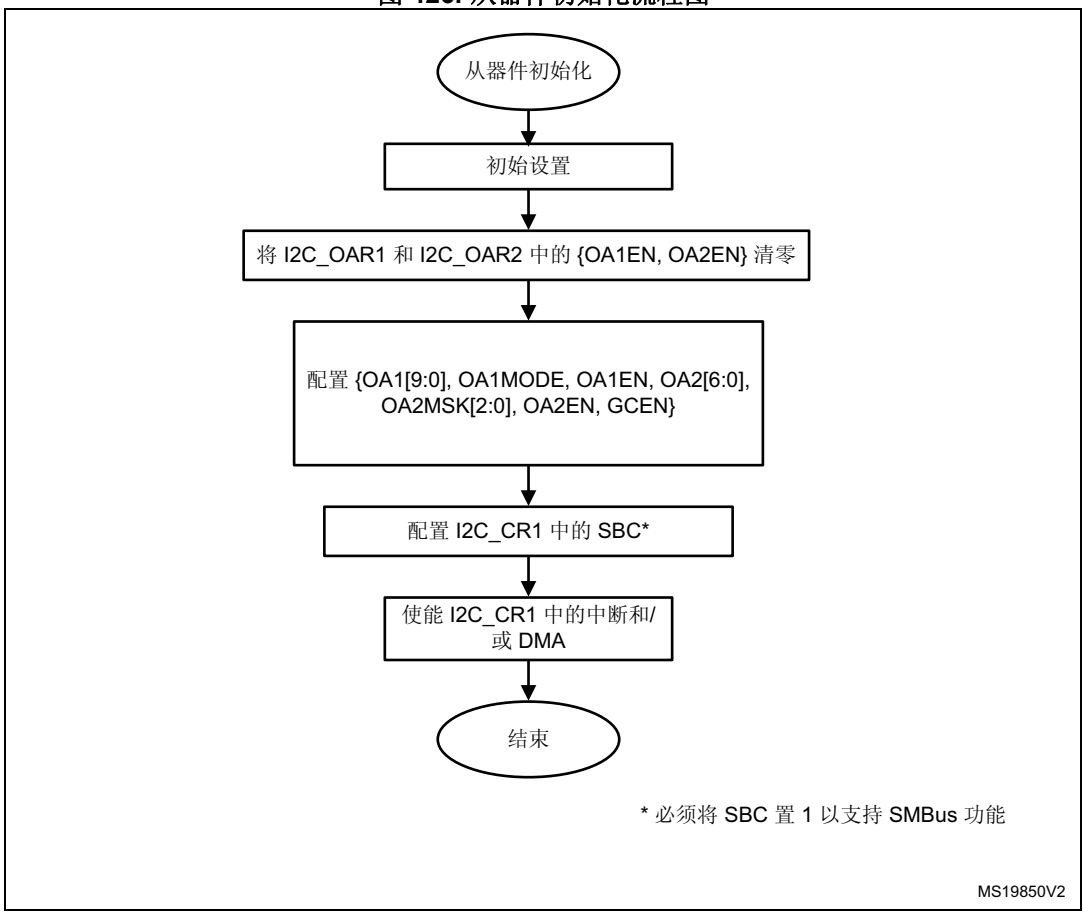
NBYTES 可加载大于 0x1 的值，在这种情况下，接收流在 NBYTES 个数据接收期间是连续的。

注： *SBC 位只能在 I2C 被禁止时、从器件不被寻址时或 ADDR=1 时配置。*

ADDR=1 或 TCR=1 时，可以更改 RELOAD 位的值。

注意： 从器件字节控制模式与 NOSTRETCH 模式不兼容。不允许在 NOSTRETCH=1 时将 SBC 位置 1。

图 426. 从器件初始化流程图



从发送器

当 I2C_TXDR 寄存器为空时，将生成发送中断状态 (TXIS)。如果 I2C_CR1 寄存器中的 TXIE 位置 1，将生成中断。

I2C_TXDR 寄存器中写入待发送的下一个数据字节时，TXIS 位将被清零。

接收到 NACK 时，I2C_ISR 寄存器中的 NACKF 位将置 1，如果 I2C_CR1 寄存器中的 NACKIE 位置 1，还将生成中断。从器件自动释放 SCL 和 SDA 线，以使主器件执行停止或重复起始位的发送。收到 NACK 时，TXIS 位不会置 1。

当接收到停止位且 I2C_CR1 寄存器中的 STOPIE 位置 1 时，I2C_ISR 寄存器中的 STOPF 标志将置 1 并且会生成中断。在大多数应用中，SBC 位通常编程为“0”。在这种情况下，如果接收到从地址 (ADDR=1) 时 TXE = 0，用户可以选择发送 I2C_TXDR 寄存器的内容作为第一个数据字节，也可以选择通过将 TXE 位置 1 来刷新 I2C_TXDR 寄存器以编程新的数据字节。

在从器件字节控制模式 (SBC=1) 下，必须在地址匹配中断子程序 (ADDR=1) 中向 NBYTE 写入待发送数据的个数。在这种情况下，传输期间 TXIS 事件的数量对应于 NBYTES 中编程的值。

注意： 如果 NOSTRETCH = 1，当 ADDR 标志置 1 时不会延长 SCL 时钟，因此用户无法在 ADDR 子程序中刷新 I2C_TXDR 寄存器的内容，从而编程第一个数据字节。必须在 I2C_TXDR 寄存器中预编程待发送的第一个数据字节：

- 该数据可以是前一个传输消息的最后一个 TXIS 事件中写入的数据。
- 如果该数据字节不是待发送的数据字节，可通过将 TXE 位置 1 来刷新 I2C_TXDR 寄存器，从而编程新的数据字节。必须仅在执行完这些操作后再清零 STOPF 位，以确保在地址应答之后第一次数据传输之前执行这些操作。

  如果第一次数据传输开始时 STOPF 仍置 1，则将生成下溢错误（OVR 标志置 1）。

  如果需要 TXIS 事件（发送中断或发送 DMA 请求），用户必须将 TXE 位和 TXIS 位均置 1，以便生成 TXIS 事件。

图 427. I2C 从发送器的传输序列流程图 (NOSTRETCH=0)

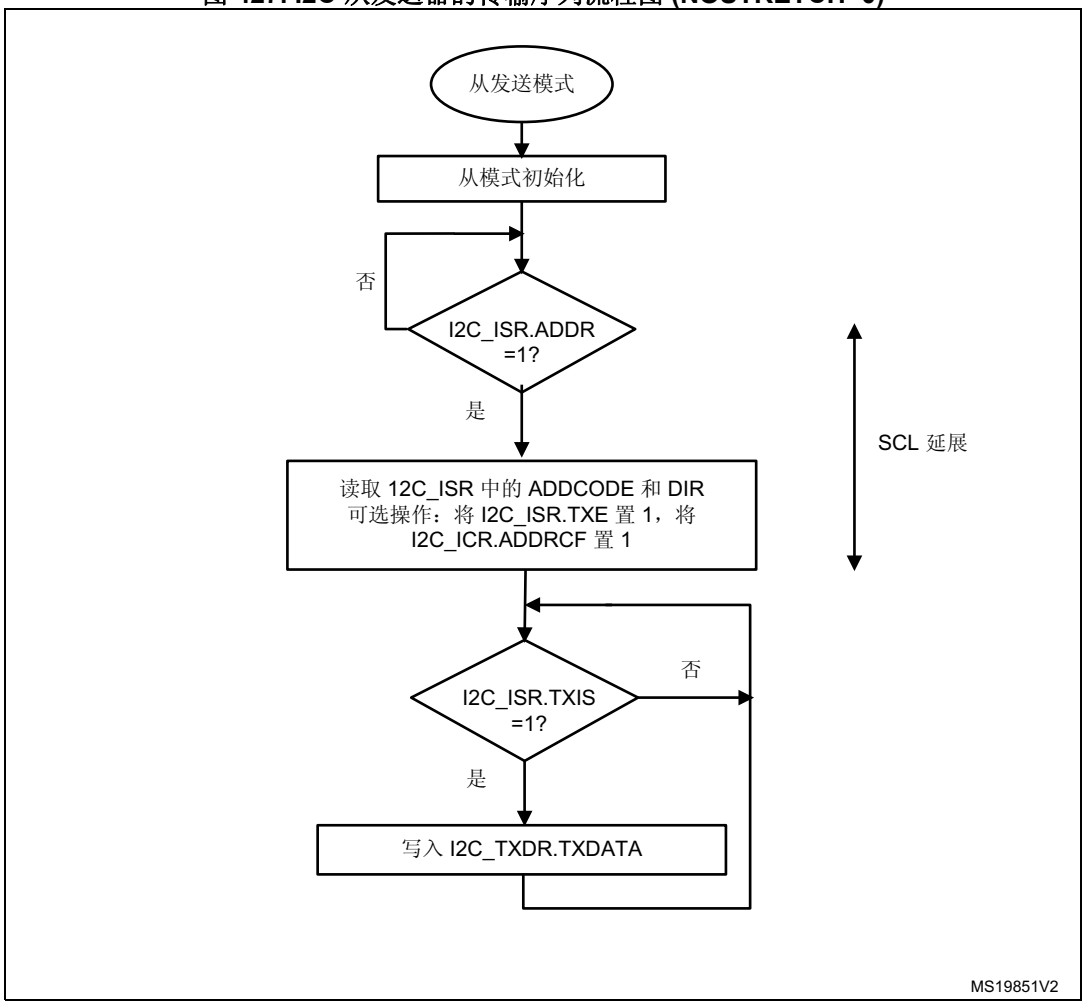


图 428. I2C 从发送器的传输序列流程图 (NOSTRETCH=1)

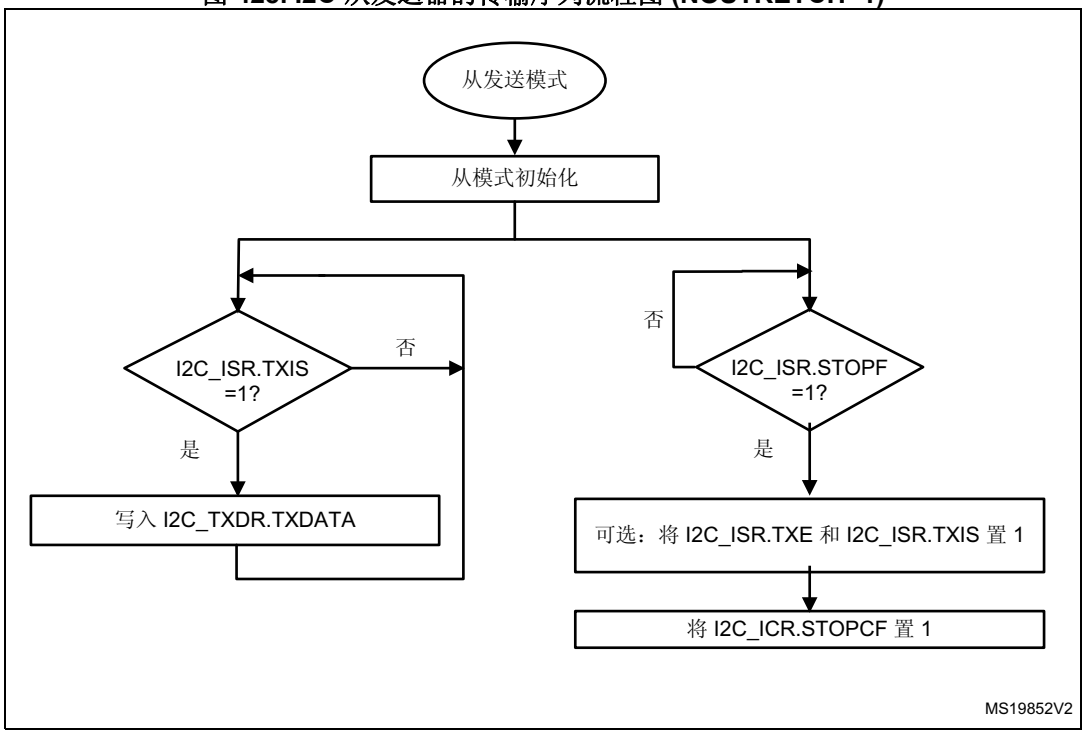
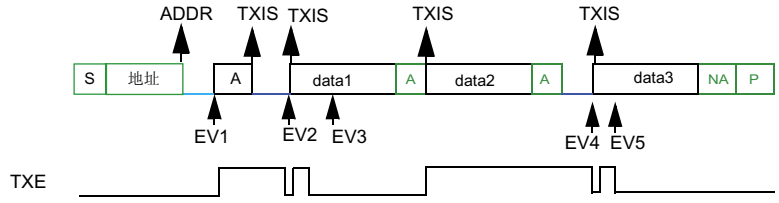


图 429. I2C 从发送器的传输总线图

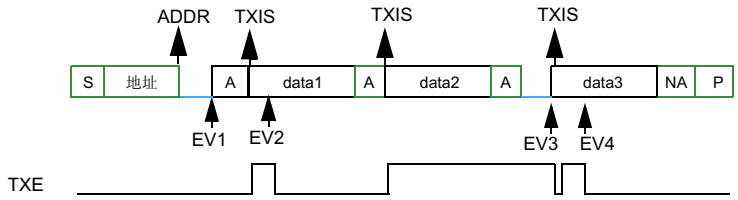
示例：I2C 从器件发送 3 个字节，刷新第 1 个数据，NOSTRETCH=0：



图注：
发送
接收
SCL 延长

EV1：ADDR ISR：检查 ADDCODE 和 DIR，将 TXE 置 1，将 ADDRCF 置 1
EV2：TXIS ISR：写入 data1
EV3：TXIS ISR：写入 data2
EV4：TXIS ISR：写入 data3
EV5：TXIS ISR：写入 data4（不发送）

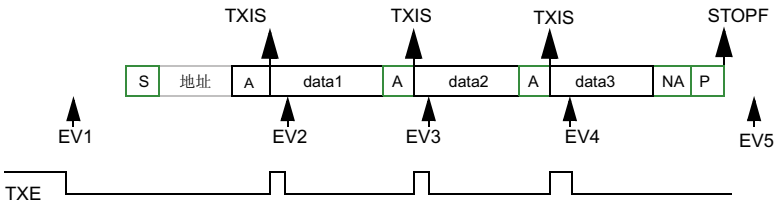
示例：I2C 从器件发送 3 个字节，不刷新第 1 个数据，NOSTRETCH=0：



图注：
发送
接收
SCL 延长

EV1：ADDR ISR：检查 ADDCODE 和 DIR，将 ADDRCF 置 1
EV2：TXIS ISR：写入 data2
EV3：TXIS ISR：写入 data3
EV4：TXIS ISR：写入 data4（不发送）

示例：I2C 从器件发送 3 个字节，NOSTRETCH=1：



图注：
发送
接收
SCL 延长

EV1：写入 data1
EV2：TXIS ISR：写入 data2
EV3：TXIS ISR：写入 data3
EV4：TXIS ISR：写入 data4（不发送）
EV5：STOPF ISR：（可选操作：将 TXE 和 TXIS 置 1），将 STOPCF 置 1

MS19853V1

从接收器

当 I2C_RXDR 满时，I2C_ISR 中的 RXNE 将置 1，如果 I2C_CR1 中的 RXIE 置 1，还将生成中断。读取 I2C_RXDR 时，将清零 RXNE。

接收到停止条件且 I2C_CR1 寄存器中的 STOPIE 置 1 时，I2C_ISR 中的 STOPF 将置 1 并且会生成中断。

图 430. 从接收器的传输序列流程图 (NOSTRETCH=0)

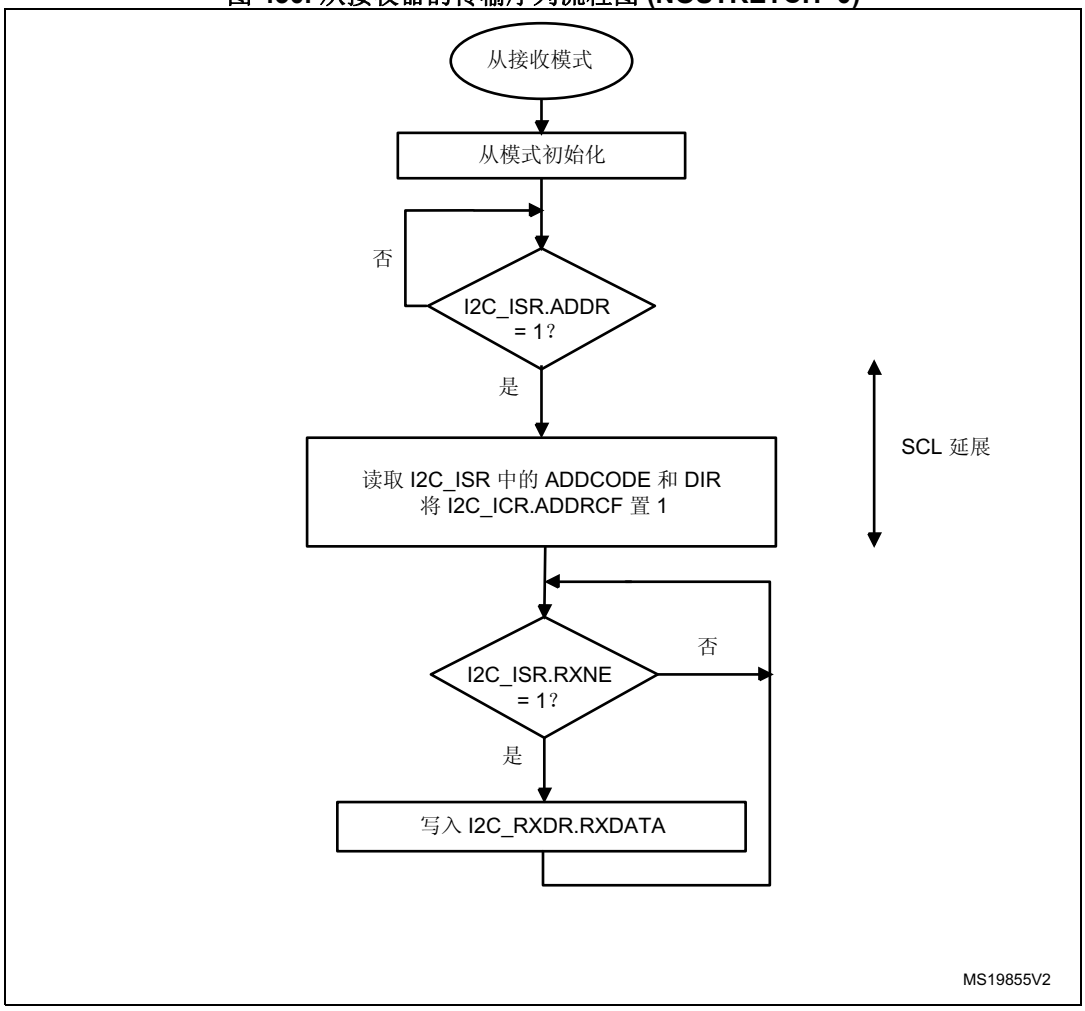


图 431. 从接收器的传输序列流程图 (NOSTRETCH=1)

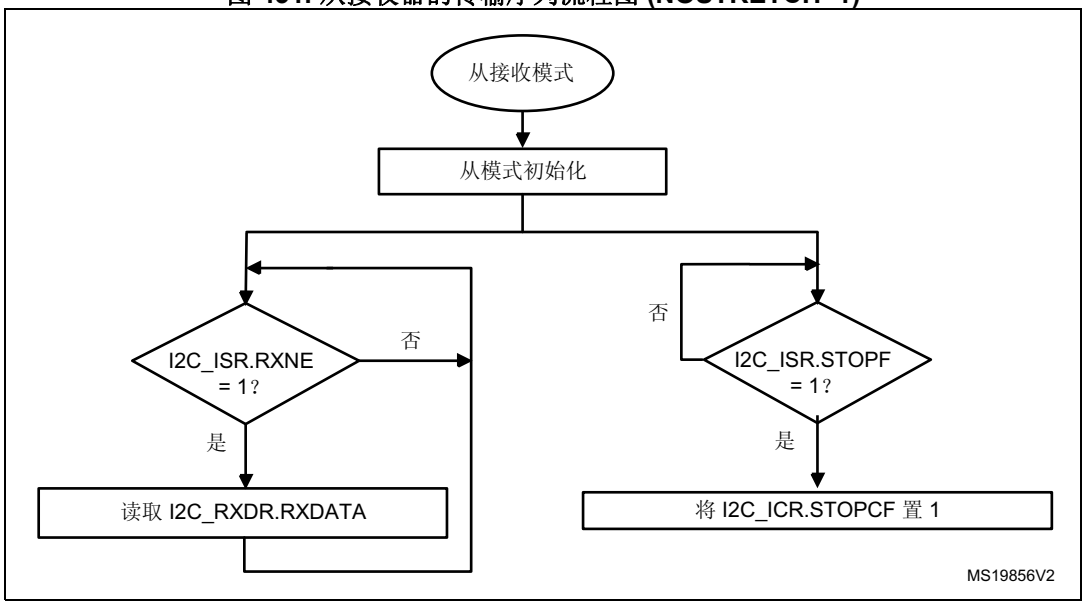
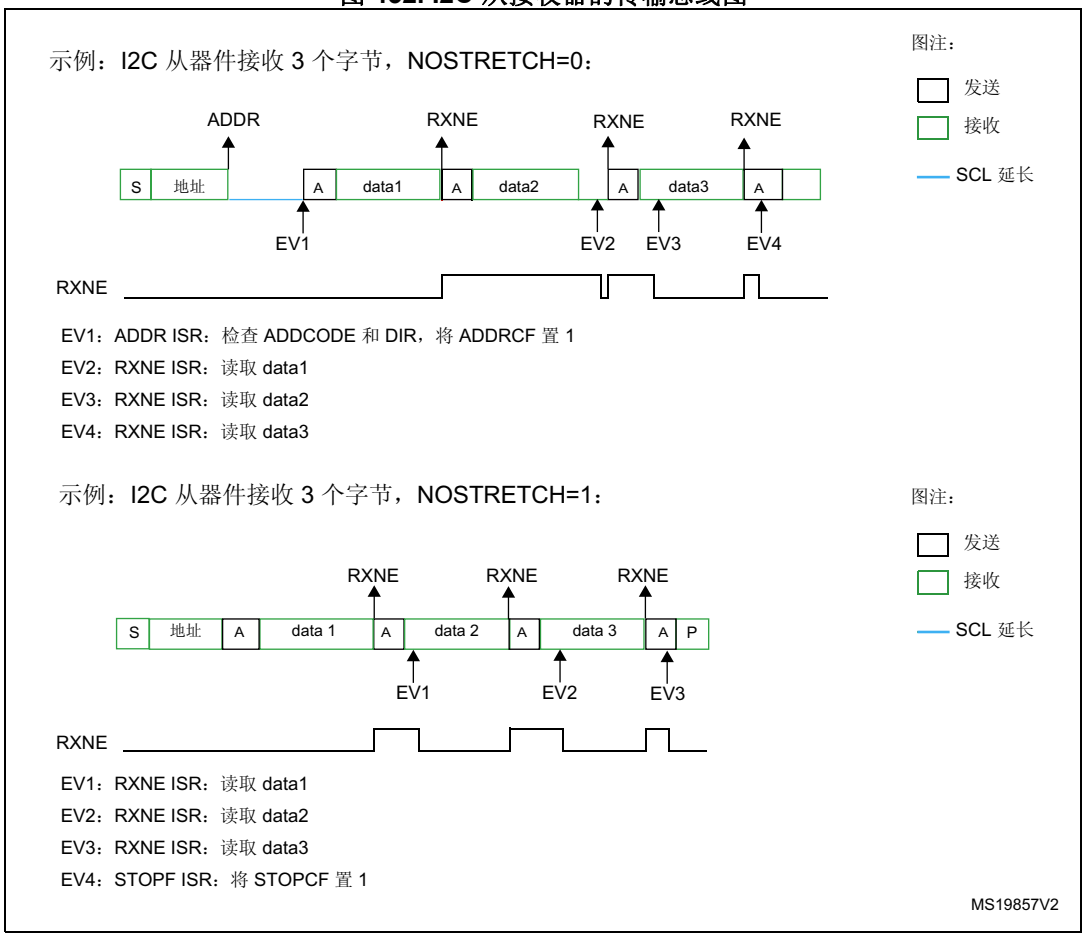


图 432. I2C 从接收器的传输总线图



43.4.8 I2C 主模式

I2C 主模式初始化

使能外设前，必须通过设置 I2C_TIMINGR 寄存器中的 SCLH 和 SCLL 位来配置 I2C 主时钟。

STM32CubeMX 工具进行计算并将 I2C_TIMINGR 内容提供在 I2C 配置窗口中。

为了支持多主环境和从时钟延长，I2C 实现了时钟同步机制。

为了实现时钟同步，需执行以下操作：

- 使用 SCLL 计数器从 SCL 低电平内部检测开始对时钟的低电平进行计数。
- 使用 SCLH 计数器从 SCL 高电平内部检测开始对时钟的高电平进行计数。

I2C 经过 t_{SYNC1} 延时后检测其自身的 SCL 低电平，该延时取决于 SCL 下降沿、SCL 输入噪声滤波器（模拟 + 数字）以及 SCL 与 I2CxCLK 时钟的同步。一旦 SCLL 计数器达到 I2C_TIMINGR 寄存器的 SCLL[7:0] 位中编程的值，I2C 便会将 SCL 释放为高电平。

I2C 经过 t_{SYNC2} 延时后检测其自身的 SCL 高电平，该延时取决于 SCL 上升沿、SCL 输入噪声滤波器（模拟 + 数字）以及 SCL 与 I2CxCLK 时钟的同步。一旦 SCLH 计数器达到 I2C_TIMINGR 寄存器的 SCLH[7:0] 位中编程的值，I2C 便会使 SCL 变为低电平。

因此，主时钟周期为：

$$t_{SCL} = t_{SYNC1} + t_{SYNC2} + \{[(SCLH+1) + (SCLL+1)] \times (PRESC+1) \times t_{I2CCLK}\}$$

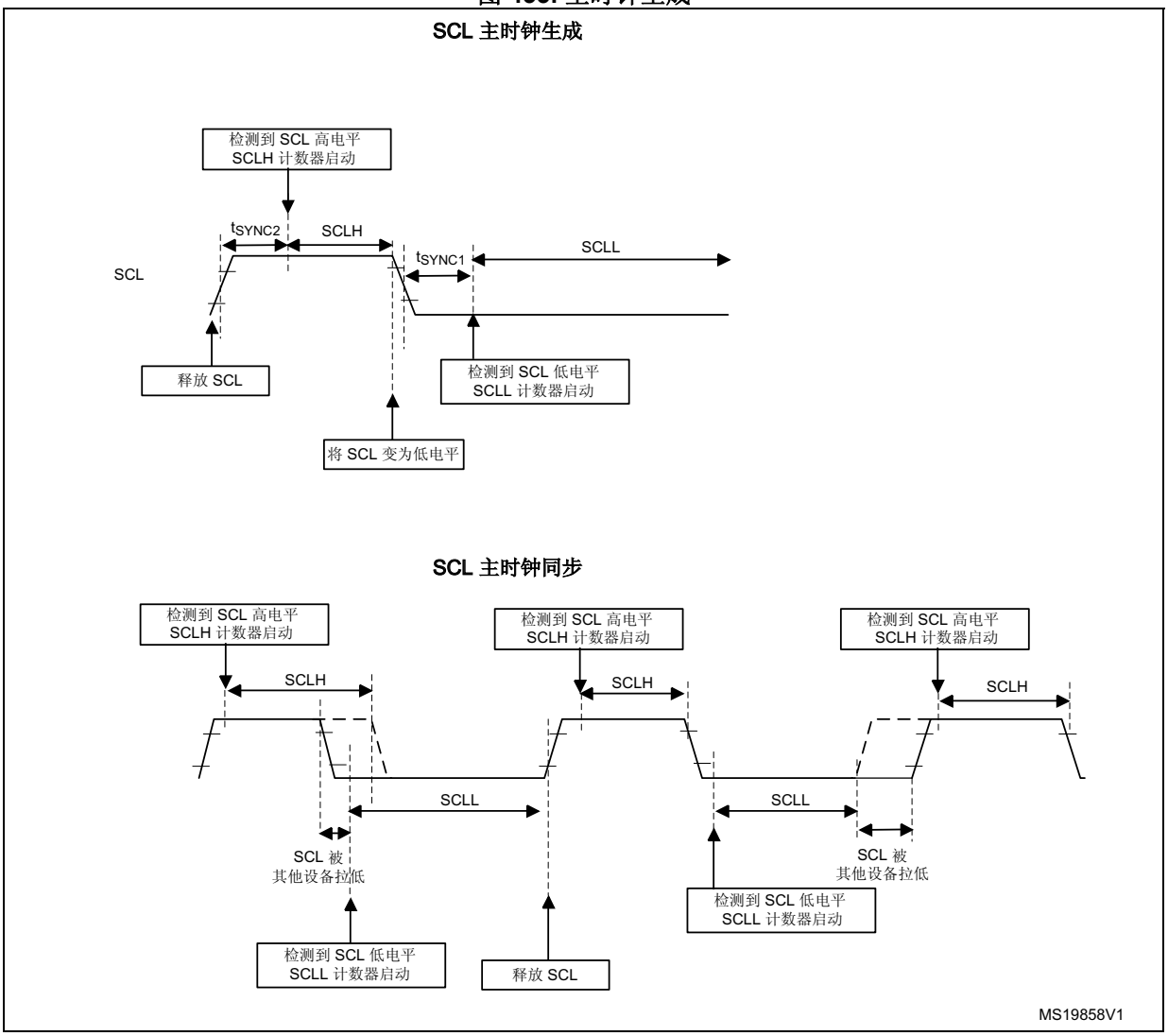
t_{SYNC1} 的持续时间取决于以下参数：

- SCL 下降斜率
- 模拟滤波器（使能时）引入的输入延时
- 数字滤波器（使能时）引入的输入延时：DNF x t_{I2CCLK}
- SCL 与 I2CCLK 时钟建立同步而产生的延时（2 到 3 个 I2CCLK 周期）

t_{SYNC2} 的持续时间取决于以下参数：

- SCL 上升斜率
- 模拟滤波器（使能时）引入的输入延时
- 数字滤波器（使能时）引入的输入延时：DNF x t_{I2CCLK}
- SCL 与 I2CCLK 时钟建立同步而产生的延时（2 到 3 个 I2CCLK 周期）

图 433. 主时钟生成



注意： 为了符合 I^2C 或 SMBus 规范，主时钟必须遵循下表中给出的时序：

表 250. I²C-SMBUS 规范时钟时序

符号	参数	标准模式 (Sm)		快速模式 (Fm)		超快速模式 (Fm)		SMBUS		单位
		最小值	最大值	最小值	最大值	最小值	最大值	最小值	最大值	
f_{SCL}	SCL 时钟频率	-	100	-	400	-	1000	-	100	kHz
$t_{HD:STA}$	（重复）起始条件的保持时间	4.0	-	0.6	-	0.26	-	4.0	-	µs
$t_{SU:STA}$	重复起始条件的建立时间	4.7	-	0.6	-	0.26	-	4.7	-	µs
$t_{SU:STO}$	停止条件的建立时间	4.0	-	0.6	-	0.26	-	4.0	-	µs
t_{BUF}	停止条件和起始条件之间的总线空闲时间	4.7	-	1.3	-	0.5	-	4.7	-	µs
t_{LOW}	SCL 时钟的低电平周期	4.7	-	1.3	-	0.5	-	4.7	-	µs
t_{HIGH}	SCL 时钟的高电平周期	4.0	-	0.6	-	0.26	-	4.0	50	µs
t_r	SDA 和 SCL 信号的上升时间	-	1000	-	300	-	120	-	1000	ns
t_f	SDA 和 SCL 信号的下降时间	-	300	-	300	-	120	-	300	ns

注： *SCLL 还用于生成 t_{BUF} 和 $t_{SU:STA}$ 时序。*

SCLH 还用于生成 $t_{HD:STA}$ 和 $t_{SU:STO}$ 时序。

有关 I2C_TIMINGR 设置与 I2CCLK 频率的示例，请参见*第 43.4.9 节：I2C_TIMINGR 寄存器配置示例*。

主模式通信初始化（地址阶段）

要发起通信，用户必须在 I2C_CR2 寄存器中为寻址的从器件编程以下参数：

- 寻址模式（7 位或 10 位）：ADD10
- 待发送的从地址：SADD[9:0]
- 传输方向：RD_WRN
- 读取 10 位地址时：HEAD10R 位。必须对 HEAD10R 进行相应配置，以指示传输方向变化时必须发送完整的地址序列，还是只发送地址头。
- 待传输的字节数：NBYTES[7:0]。如果字节数等于或大于 255，则初始化时必须将 NBYTES[7:0] 填充为 0xFF。

然后，用户必须将 I2C_CR2 寄存器中的 START 位置 1。START 位置 1 时，不允许更改上述所有位。

之后，当主器件检测到总线空闲 (BUSY = 0) 时，它会在经过 t_{BUF} 的延时后自动发送起始位，随后发出从器件地址。

仲裁丢失时，主器件将自动切换回从模式，如果作为从器件被寻址，还可对其自身地址进行应答。

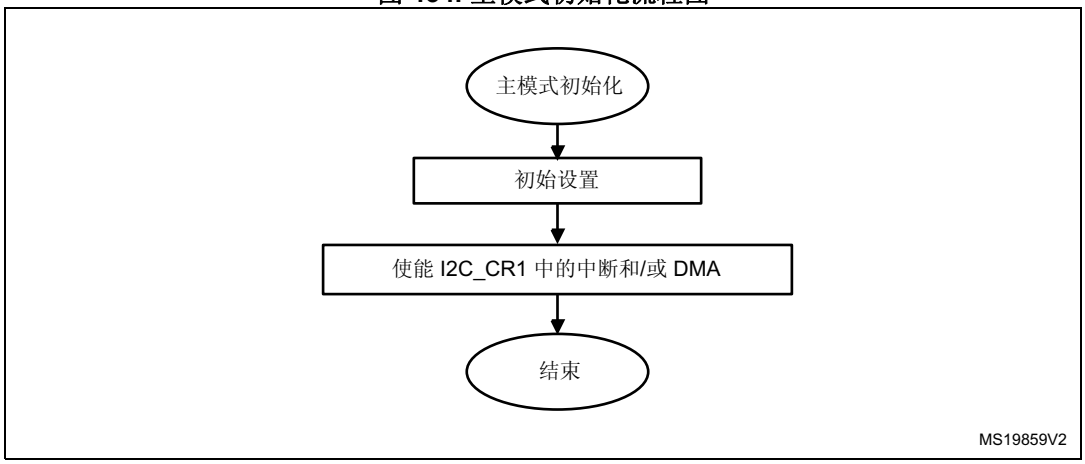
注： *无论接收到的应答值为何，只要已在总线上发送从地址，START 位便会由硬件复位。如果仲裁丢失，START 位也会由硬件复位。*

在 10 位寻址模式下，如果从器件不对从地址的前 7 位进行应答，则主器件将自动重新启动从地址发送，直至接收到 ACK。在这种情况下，如果从从器件接收到 NACK，则必须将 ADDRCF 置 1，以停止发送从地址。

如果当 START 位置 1 时，I2C 作为从器件 (ADDR=1) 被寻址，则 I2C 将切换为从模式，START 位将在 ADDRCF 位置 1 时清零。

注：　该步骤同样适用于重复起始位。在这种情况下，BUSY=1。

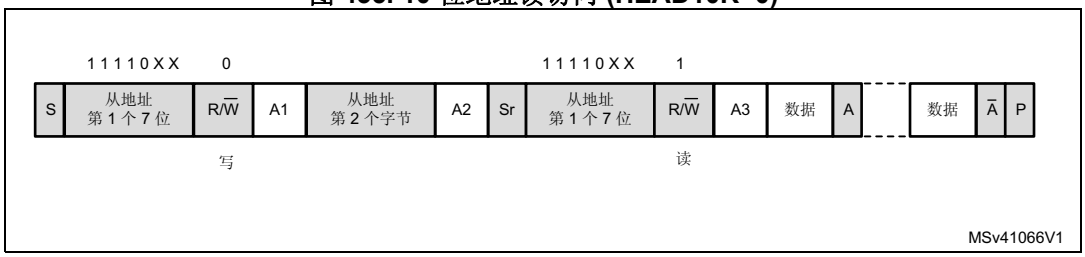
图 434. 主模式初始化流程图



主接收器寻址 10 位地址从器件的初始化过程

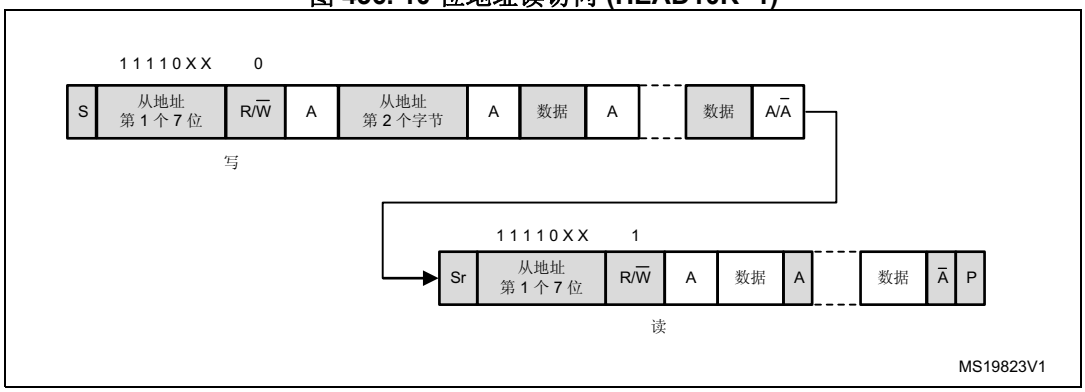
- 如果从地址采用 10 位格式，用户可选择将 I2C_CR2 寄存器中的 HEAD10R 位清零来发送完整的读序列。在这种情况下，主器件会在 START 位置 1 后自动发送以下完整序列：（重复）起始位 + 带写方向的从器件 10 位地址头字节 + 从器件地址第 2 个字节 + 重复起始位 + 带读方向的从器件 10 位地址头字节

图 435. 10 位地址读访问 (HEAD10R=0)



- 如果主器件对 10 位地址从器件进行寻址、向该从器件发送数据、然后再从该从器件读取数据，则必须首先完成主器件发送过程。然后，重复起始位置 1，10 位从地址配置为 HEAD10R=1。在这种情况下，主器件发送以下序列：重复起始位 + 从地址 10 位头读取

图 436. 10 位地址读访问 (HEAD10R=1)



主发送器

写传输时，在发送完每个字节（即第 9 个 SCL 脉冲（接收到 ACK 时））后，TXIS 标志将置 1。

如果 I2C_CR1 寄存器中的 TXIE 位置 1，TXIS 事件将生成中断。当 I2C_TXDR 寄存器中写入待发送的下一个数据字节时，该标志将被清零。

传输期间的 TXIS 事件的数量对应于 NBYTES[7:0] 中编程的值。如果待发送的数据字节总数大于 255，则必须通过将 I2C_CR2 寄存器中的 RELOAD 位置 1 来选择重载模式。在这种情况下，当 NBYTES 数据传输完成时，TCR 标志将置 1，并且 SCL 线的低电平将被延展，直到 NBYTES[7:0] 被写入非零值。

收到 NACK 时，TXIS 标志不会置 1。

- 当 RELOAD=0 且 NBYTES 数据传输完成时：
  - 在自动结束模式 (AUTOEND=1) 下，将自动发送停止位。
  - 在软件结束模式 (AUTOEND=0) 下，TC 标志将置 1 且 SCL 线的低电平将被延展，以便执行以下软件操作：

    可通过将 I2C_CR2 寄存器中的 START 位置 1 并配置适当的从地址和待传输字节数来请求发送重复起始位。将 START 位置 1 会将 TC 标志清零，并在总线上发送起始位。

    可通过将 I2C_CR2 寄存器中的 STOP 位置 1 来请求停止位。将 STOP 位置 1 会将 TC 标志清零，并在总线上发送停止位。
- 如果接收到 NACK：TXIS 标志不会置 1，并且接收到 NACK 后会自动发送停止位。I2C_ISR 寄存器中的 NACKF 标志置 1，如果 NACKIE 位置 1，还将生成中断。

图 437. I2C 主发送器的传输序列流程图（N≤255 字节）

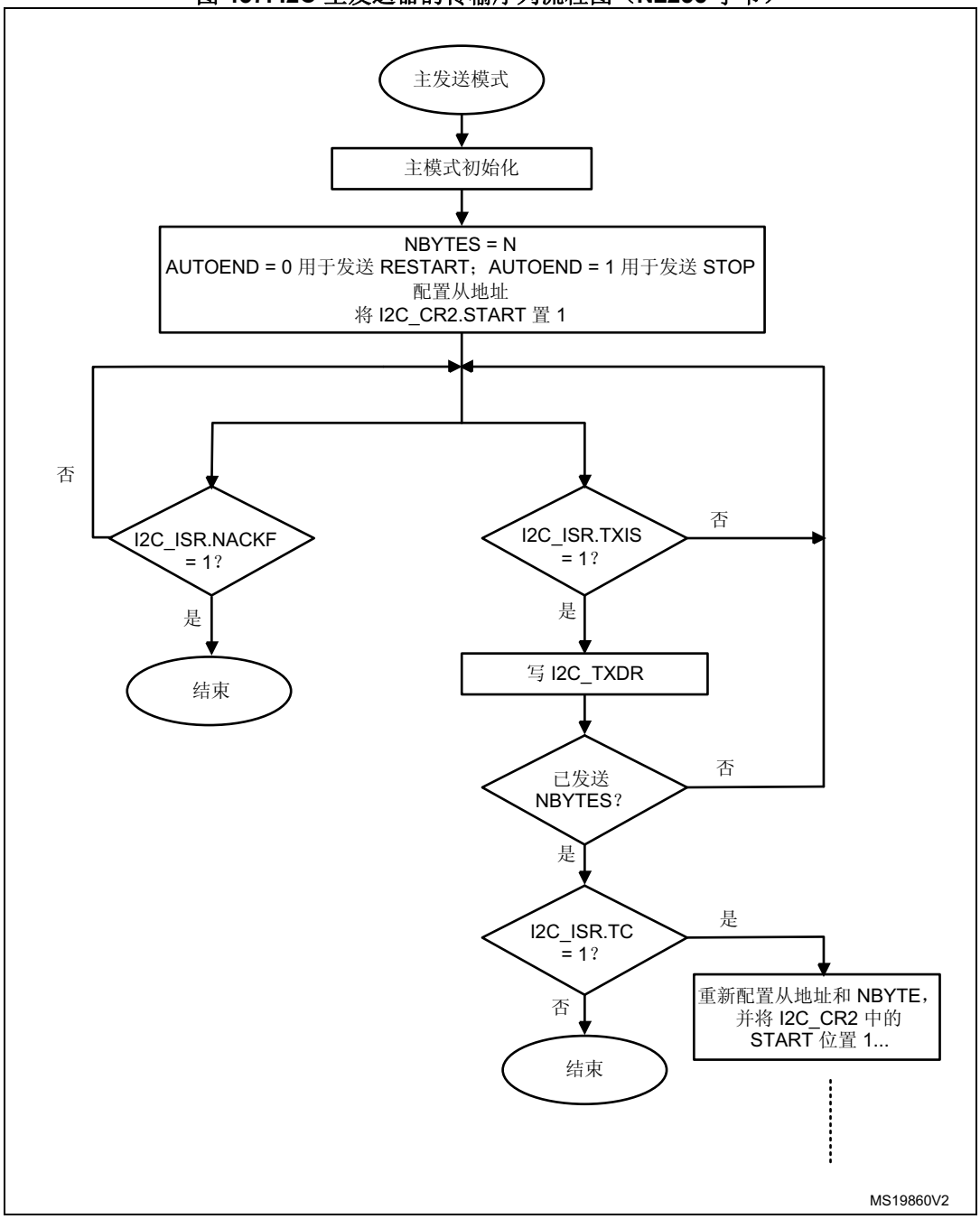


图 438. I2C 主发送器的传输序列流程图（N>255 字节）

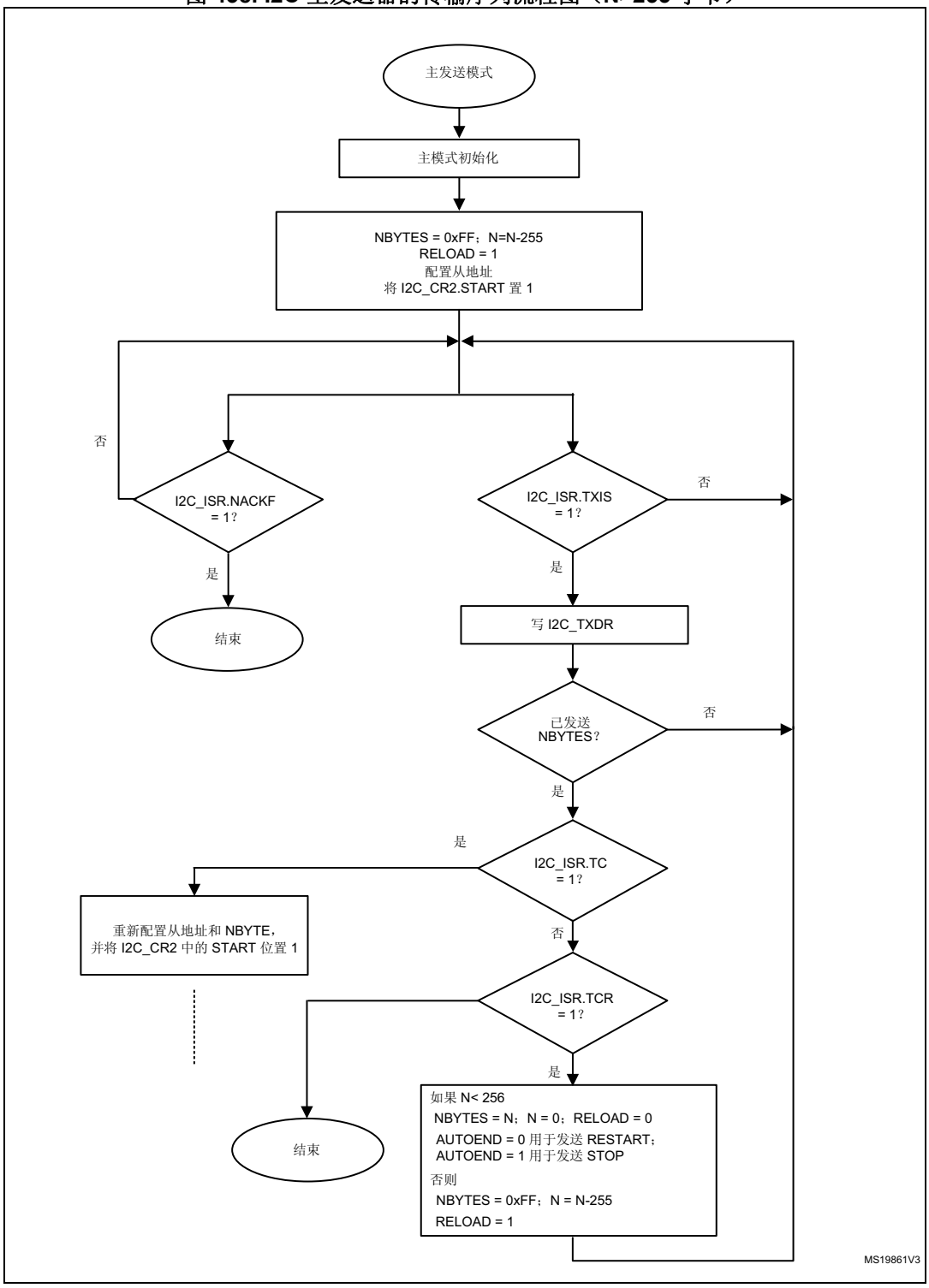
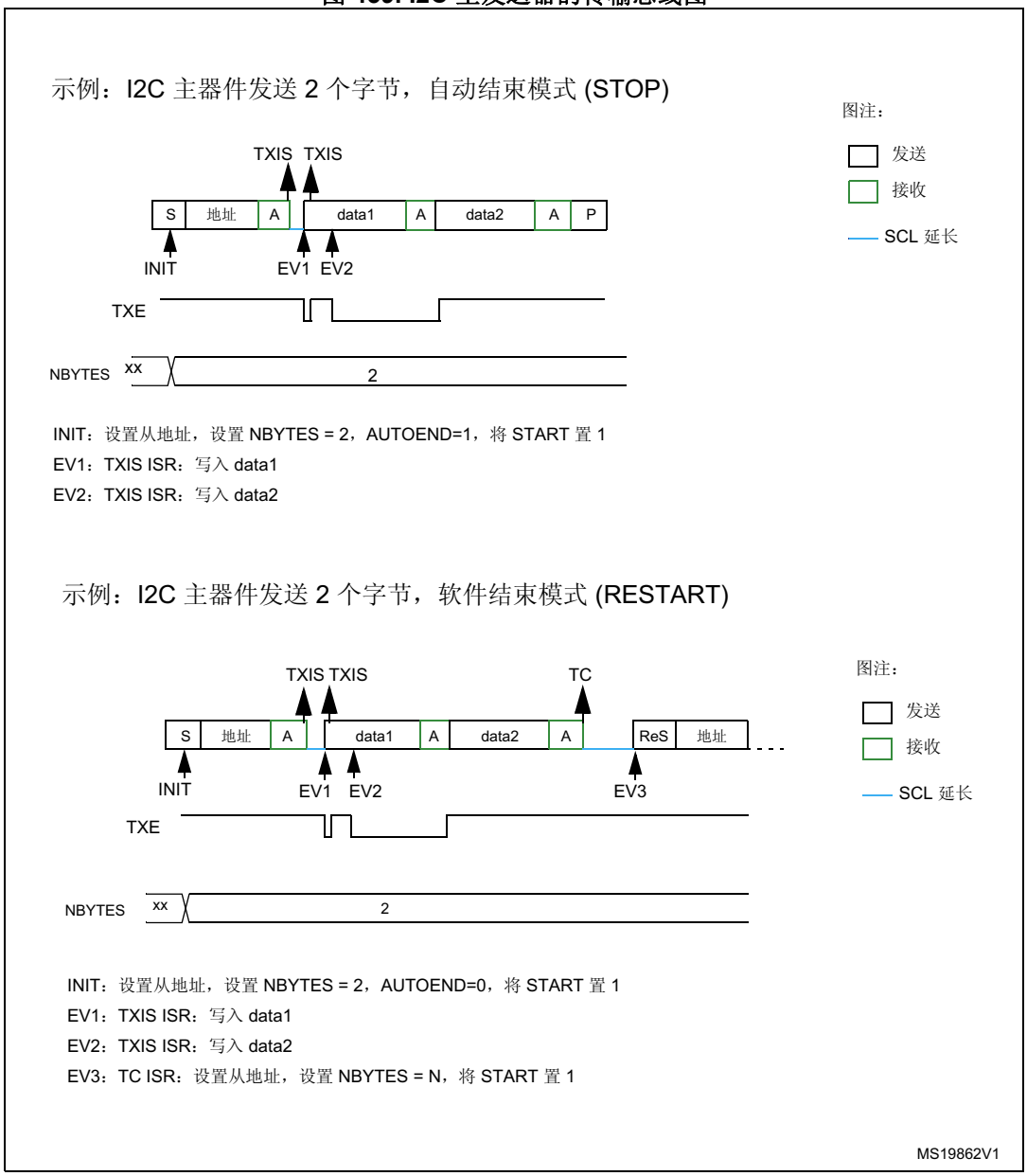


图 439. I2C 主发送器的传输总线图



主接收器

读传输时，在接收到每个字节（即第 8 个 SCL 脉冲）后，RXNE 标志将置 1。如果 I2C_CR1 寄存器中的 RXIE 位置 1，RXNE 事件将生成中断。读取 I2C_RXDR 时，将清零该标志。

如果待接收的数据字节总数大于 255，则必须通过将 I2C_CR2 寄存器中的 RELOAD 位置 1 来选择重载模式。在这种情况下，当 NBYTES[7:0] 数据传输完成时，TCR 标志将置 1，并且 SCL 线的低电平将被延展，直到 NBYTES[7:0] 被写入非零值。

- 当 RELOAD=0 且 NBYTES[7:0] 数据传输完成时：
  - 在自动结束模式 (AUTOEND=1) 下，接收到最后一个字节后，将自动发送 NACK 和停止位。
  - 在软件结束模式 (AUTOEND=0) 下，接收到最后一个字节后，将自动发送 NACK，TC 标志将置 1 且 SCL 线的低电平将被延展，以便执行以下软件操作：

    可通过将 I2C_CR2 寄存器中的 START 位置 1 并配置适当的从地址和待传输字节数来请求发送重复起始位。将 START 位置 1 会将 TC 标志清零，并在总线上发送起始位，后跟从地址。

    可通过将 I2C_CR2 寄存器中的 STOP 位置 1 来请求停止位。将 STOP 位置 1 会将 TC 标志清零，并在总线上发送停止位。

图 440. I2C 主接收器的传输序列流程图（N≤255 字节）

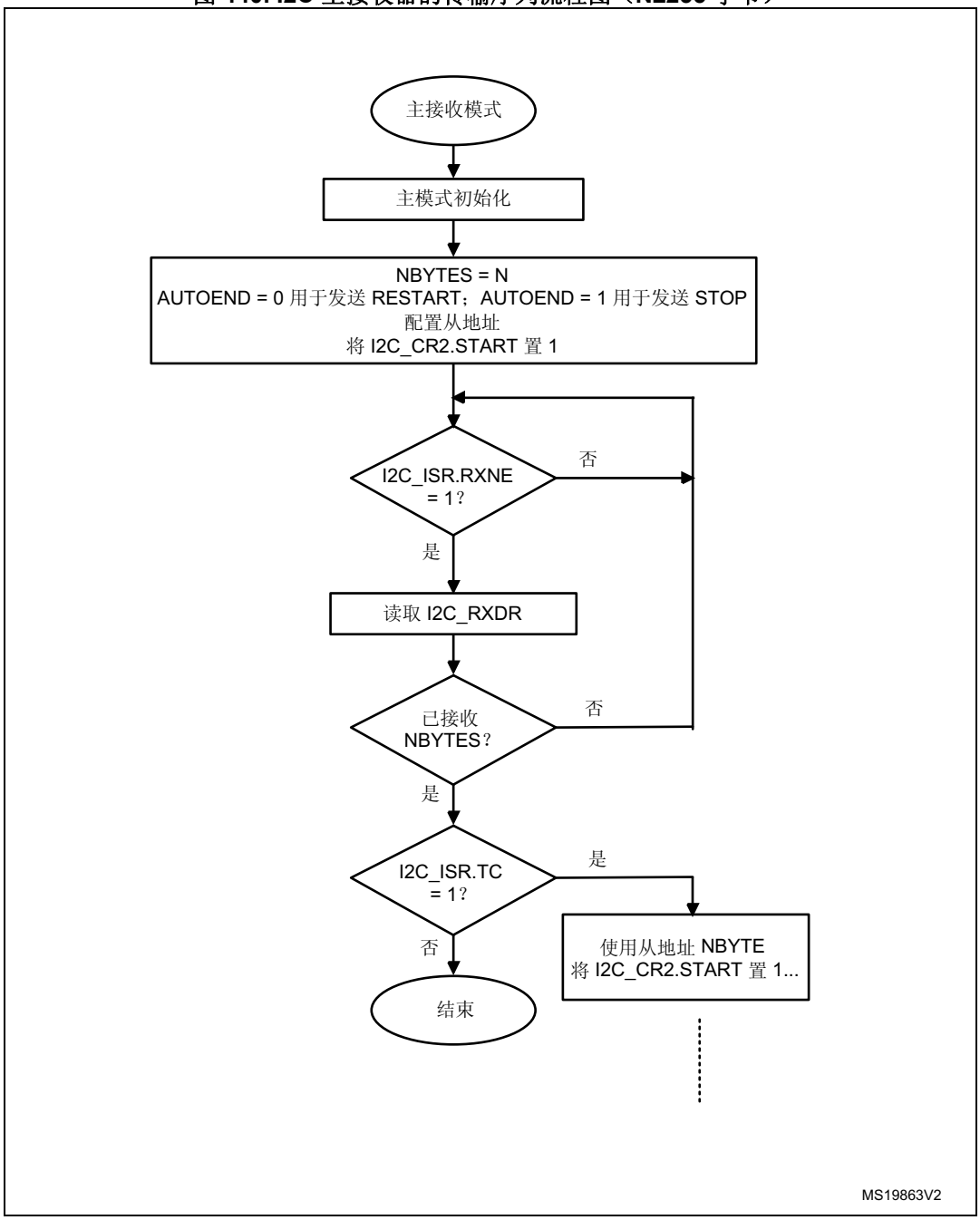


图 441. I2C 主接收器的传输序列流程图（N>255 字节）

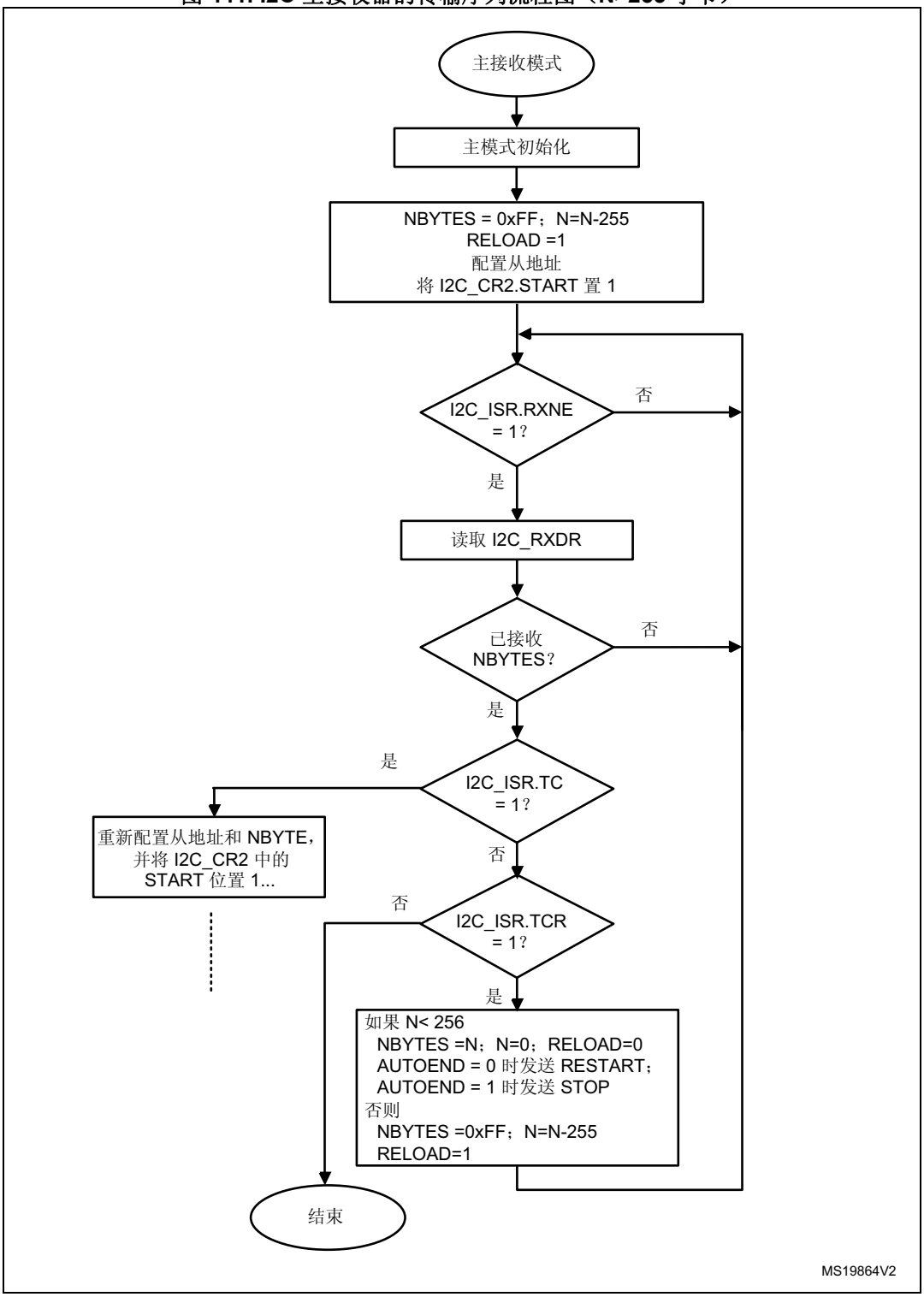
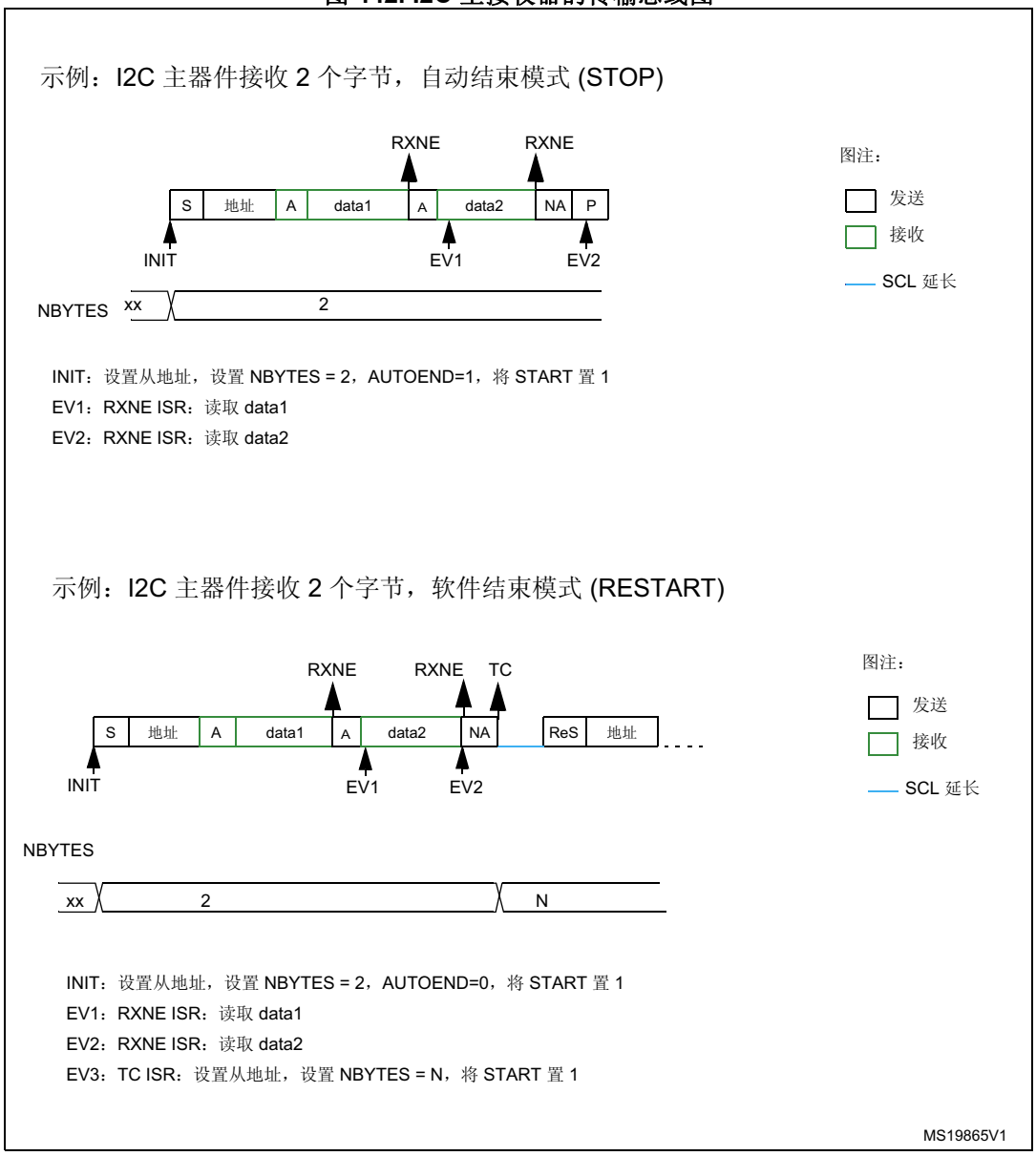


图 442. I2C 主接收器的传输总线图



43.4.9 I2C_TIMINGR 寄存器配置示例

下文各表提供了相应示例，以介绍如何编程 I2C_TIMINGR 才能获得符合 I^2C 规范的时序。要获取更准确的配置值，应使用 STM32CubeMX 工具（I2C 配置窗口）。

表 251. f_{I2CCLK} = 8 MHz 时的时序设置示例

参数	标准模式 (Sm)		快速模式 (Fm)	超快速模式 (Fm+)
	10 kHz	**100 kHz**	**400 kHz**	**500 kHz**
PRESC	1	1	0	0
SCLL	0xC7	0x13	0x9	0x6
t_{SCLL}	200x250 ns = 50 µs	20x250 ns = 5.0 µs	10x125 ns = 1250 ns	7x125 ns = 875 ns
SCLH	0xC3	0xF	0x3	0x3
t_{SCLH}	196x250 ns = 49 µs	16x250 ns = 4.0µs	4x125ns = 500ns	4x125 ns = 500 ns
t_{SCL}(1)	约 100 µs(2)	约 10 µs(2)	约 2500 ns(3)	约 2000 ns(4)
SDADEL	0x2	0x2	0x1	0x0
t_{SDADEL}	2x250 ns = 500 ns	2x250 ns = 500 ns	1x125 ns = 125 ns	0 ns
SCLDEL	0x4	0x4	0x3	0x1
t_{SCLDEL}	5x250 ns = 1250 ns	5x250 ns = 1250 ns	4x125 ns = 500 ns	2x125 ns = 250 ns

1. 由于 SCL 内部检测存在延时，SCL 周期 t_{SCL} 大于 $t_{SCLL} + t_{SCLH}$。为 t_{SCL} 提供的值仅用于举例说明。
2. $t_{SYNC1} + t_{SYNC2}$ 最小值为 4 x t_{I2CCLK} = 500 ns。$t_{SYNC1} + t_{SYNC2}$ = 1000 ns 时的示例。
3. $t_{SYNC1} + t_{SYNC2}$ 最小值为 4 x t_{I2CCLK} = 500 ns。$t_{SYNC1} + t_{SYNC2}$ = 750 ns 时的示例。
4. $t_{SYNC1} + t_{SYNC2}$ 最小值为 4 x t_{I2CCLK} = 500 ns。$t_{SYNC1} + t_{SYNC2}$ = 655 ns 时的示例。

表 252. f_{I2CCLK} = 16 MHz 时的时序设置示例

参数	标准模式 (Sm)		快速模式 (Fm)	超快速模式 (Fm+)
	10 kHz	100 kHz	400 kHz	1000 kHz
PRESC	3	3	1	0
SCLL	0xC7	0x13	0x9	0x4
t_{SCLL}	200 x 250 ns = 50 µs	20 x 250 ns = 5.0 µs	10 x 125 ns = 1250 ns	5 x 62.5 ns = 312.5 ns
SCLH	0xC3	0xF	0x3	0x2
t_{SCLH}	196 x 250 ns = 49 µs	16 x 250 ns = 4.0 µs	4 x 125 ns = 500 ns	3 x 62.5 ns = 187.5 ns
t_{SCL}(1)	约 100 µs(2)	约 10 µs(2)	约 2500 ns(3)	约 1000 ns(4)
SDADEL	0x2	0x2	0x2	0x0
t_{SDADEL}	2 x 250 ns = 500 ns	2 x 250 ns = 500 ns	2 x 125 ns = 250 ns	0 ns
SCLDEL	0x4	0x4	0x3	0x2
t_{SCLDEL}	5 x 250 ns = 1250 ns	5 x 250 ns = 1250 ns	4 x 125 ns = 500 ns	3 x 62.5 ns = 187.5 ns

1. 由于 SCL 内部检测存在延时，SCL 周期 t_{SCL} 大于 t_{SCLL} + t_{SCLH}。为 t_{SCL} 提供的值仅用于举例说明。
2. t_{SYNC1} + t_{SYNC2} 最小值为 4 x t_{I2CCLK} = 250 ns。t_{SYNC1} + t_{SYNC2} = 1000 ns 时的示例。
3. t_{SYNC1} + t_{SYNC2} 最小值为 4 x t_{I2CCLK} = 250 ns。t_{SYNC1} + t_{SYNC2} = 750 ns 时的示例。
4. t_{SYNC1} + t_{SYNC2} 最小值为 4 x t_{I2CCLK} = 250 ns。t_{SYNC1} + t_{SYNC2} = 500 ns 时的示例。

表 253. f_{I2CCLK} = 48 MHz 时的时序设置示例

参数	标准模式 (Sm)		快速模式 (Fm)	超快速模式 (Fm+)
	10 kHz	100 kHz	400 kHz	1000 kHz
PRESC	0xB	0xB	5	5
SCLL	0xC7	0x13	0x9	0x3
t_{SCLL}	200 x 250 ns = 50 µs	20 x 250 ns = 5.0 µs	10 x 125 ns = 1250 ns	4 x 125 ns = 500 ns
SCLH	0xC3	0xF	0x3	0x1
t_{SCLH}	196 x 250 ns = 49 µs	16 x 250 ns = 4.0 µs	4 x 125 ns = 500 ns	2 x 125 ns = 250 ns
t_{SCL}(1)	约 100 µs(2)	约 10 µs(2)	约 2500 ns(3)	约 875 ns(4)
SDADEL	0x2	0x2	0x3	0x0
t_{SDADEL}	2 x 250 ns = 500 ns	2 x 250 ns = 500 ns	3 x 125 ns = 375 ns	0 ns
SCLDEL	0x4	0x4	0x3	0x1
t_{SCLDEL}	5 x 250 ns = 1250 ns	5 x 250 ns = 1250 ns	4 x 125 ns = 500 ns	2 x 125 ns = 250 ns

1. 由于 SCL 内部检测存在延时，SCL 周期 t_{SCL} 大于 t_{SCLL} + t_{SCLH}。为 t_{SCL} 提供的值仅用于举例说明。
2. t_{SYNC1} + t_{SYNC2} 最小值为 4x t_{I2CCLK} = 83.3 ns。t_{SYNC1} + t_{SYNC2} = 1000 ns 时的示例。
3. t_{SYNC1} + t_{SYNC2} 最小值为 4x t_{I2CCLK} = 83.3 ns。t_{SYNC1} + t_{SYNC2} = 750 ns 时的示例。
4. t_{SYNC1} + t_{SYNC2} 最小值为 4x t_{I2CCLK} = 83.3 ns。t_{SYNC1} + t_{SYNC2} = 250 ns 时的示例。

43.4.10 SMBus 特性

仅当支持 SMBus 功能时，才涉及本节内容。请参见*第 43.3 节：I2C 特性实现*。

简介

系统管理总线 (SMBus) 是一个双线制接口，各器件可通过它在彼此之间或者与系统的其余部分进行通信。它以 I^2C 的工作原理为基础。SMBus 可针对系统和电源管理相关的任务提供控制总线。

该外设与 SMBUS 规范兼容 (http://smbus.org)。

系统管理总线规范涉及三类器件。

- 从器件，用于接收或响应命令。
- 主器件，用于发出命令、生成时钟和中止传输。
- 主机，专用的主器件，可提供连接系统 CPU 的主接口。主机必须具有主 - 从器件功能，并且必须支持 SMBus 主机通知协议。系统中只允许存在一个主机。

该外设可配置为主器件或从器件，也可配置为主机。

总线协议

任何给定器件都有十一种可用命令协议。器件既可以在这十一种协议中任选其一，也可以使用全部十一种协议进行通信。这十一种协议分别为快速命令、发送字节、接收字节、写入字节、写入字、读取字节、读取字、过程调用、块读取、块写入以及块写入-块读取过程调用。这些协议应通过用户软件实施。

有关这些协议的详细信息，请参见 SMBus 规范 (http://smbus.org)。

地址解析协议 (ARP)

通过为各个从器件动态分配一个新的唯一地址可解决 SMBus 从地址冲突的问题。为了提供一种机制来针对地址分配隔离各个器件，各器件必须具有唯一的器件标识符 (UDID)。该 128 位数字由软件实现。

该外设支持地址解析协议 (ARP)。通过将 I2C_CR1 寄存器中的 SMBDEN 位置 1 来使能 SMBus 器件默认地址 (0b1100 001)。ARP 命令应通过用户软件实现。

此外，还将在从模式下执行仲裁以支持 ARP。

有关 SMBus 地址解析协议的详细信息，请参见 SMBus 规范 (http://smbus.org)。

接收的命令和数据应答控制

SMBus 接收器必须能够对接收到的每个命令或数据进行否定应答。要在从模式下实现 ACK 控制，必须通过将 I2C_CR1 寄存器中的 SBC 位置 1 来使能从字节控制模式。更多详细信息，请参见*第 1400 页的从器件字节控制模式*。

主机通知协议

该外设通过将 I2C_CR1 寄存器中的 SMBHEN 位置 1 来支持主机通知协议。在这种情况下，主机将应答 SMBus 主机地址 (0b0001 000)。

使用该协议时，器件用作主器件，而主机用作从器件。

SMBus 报警

器件支持 SMBus ALERT 可选信号。只具备从功能的器件可通过 SMBALERT# 引脚向主机发出信号，指示它想要通信。主机会处理该中断并通过报警响应地址 (0b0001 100) 同时访问所有 SMBALERT# 器件。只有那些将 SMBALERT# 拉到低电平的器件会应答报警响应地址。

如果配置为从器件 (SMBHEN=0)，则通过将 I2C_CR1 寄存器中的 ALERTEN 位置 1 来将 SMBA 引脚拉为低电平。这同时还会使能报警响应地址。

如果配置为主机 (SMBHEN=1)，则当 SMBA 引脚上检测到下降沿且 ALERTEN=1 时，I2C_ISR 寄存器中的 ALERT 标志置 1。如果 I2C_CR1 寄存器中的 ERRIE 位置 1，将生成中断。当 ALERTEN=0 时，即使外部 SMBA 引脚为低电平，ALERT 线也将被视为高电平。

如果无需 SMBus ALERT 引脚，则当 ALERTEN=0 时，SMBA 引脚可用作标准 GPIO。

数据包错误校验

SMBus 规范中引入了数据包错误校验机制来提高可靠性和通信稳定性。数据包错误校验的实施方式是在每次消息传输结束时附加数据包错误代码 (PEC)。PEC 的计算方式是对所有消息字节（包括地址和读/写位）使用 CRC-8 多项式 $C(x) = x_8 + x^2 + x + 1$。

外设内置了硬件 PEC 计算器，可在接收到的字节与硬件计算的 PEC 不匹配时自动发送否定应答信号。

超时

该外设内置了硬件定时器，以便符合 SMBus 规范中定义的 3 个超时。

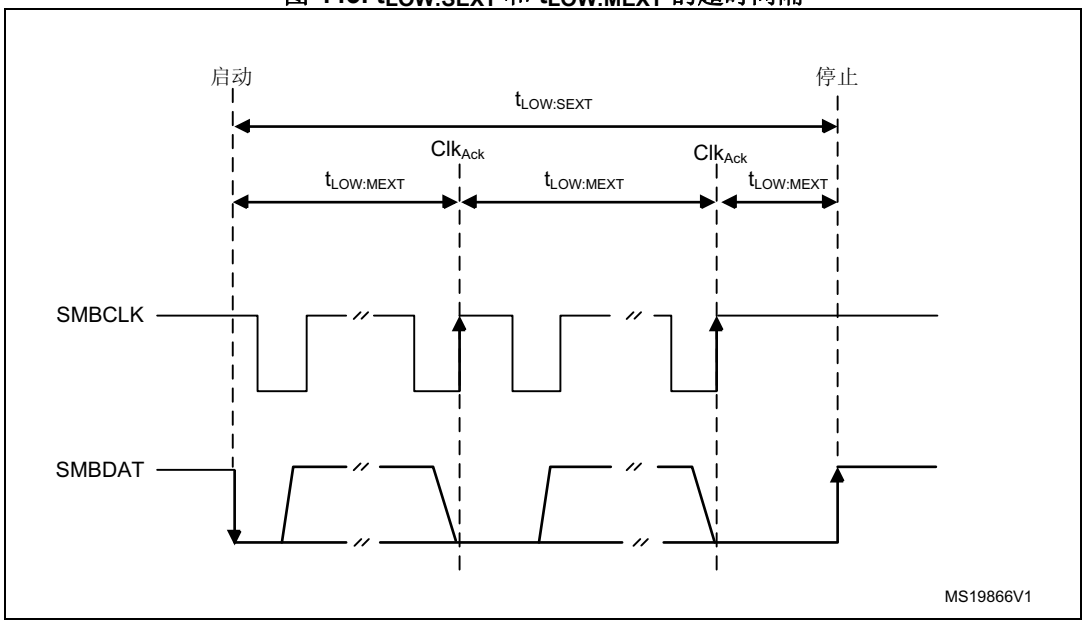
表 254. SMBus 超时规范

符号	参数	限值		单位
		最小值	最大值	
$t_{TIMEOUT}$	检测时钟低电平超时	25	35	ms
$t_{LOW:SEXT}$(1)	累积时钟低电平延长时间（从器件）	-	25	ms
$t_{LOW:MEXT}$(2)	累积时钟低电平延长时间（主器件）	-	10	ms

1. $t_{LOW:SEXT}$ 是一段累积时间，即给定从器件在一条消息的最初起始到停止期间时钟信号可延展的时间。其他从器件或主器件也可能延长时钟，进而导致时钟低电平总延长时间超过 $t_{LOW:SEXT}$。因此，测量该参数时该器件应该是全速主器件寻址的唯一器件。

2. $t_{LOW:MEXT}$ 是一段累积时间，即主器件在消息的每个字节（定义为 START 到 ACK、ACK 到 ACK 或 ACK 到 STOP）内时钟信号可延展的时间。从器件或其他主器件也可能延长时钟，进而导致时钟低电平总时间超过 $t_{LOW:MEXT}$（针对给定字节）。因此，测量该参数时该全速主器件只寻址一个从器件。

图 443. $t_{LOW:SEXT}$ 和 $t_{LOW:MEXT}$ 的超时间隔



总线空闲检测

如果主器件检测到时钟和数据信号的高电平时间已达 t_{IDLE}（超过 $t_{HIGH,MAX}$），则认为总线空闲（请参见*表 248：I2C-SMBUS 规范数据建立和保持时间*）。

该时序参数已考虑如下情况：主器件已动态添加至总线，但可能尚未检测到 SMBCLK 或 SMBDAT 线上的状态转换。在这种情况下，主器件必须等待足够长的时间，以确定当前未进行传输。外设支持硬件总线空闲检测。

43.4.11 SMBus 初始化

仅当支持 SMBus 功能时，才涉及本节内容。请参见*第 43.3 节：I2C 特性实现*。

除了 I2C 初始化之外，还必须进行一些其他的特定初始化，以便执行 SMBus 通信：

接收的命令和数据应答控制（从模式）

SMBus 接收器必须能够对接收到的每个命令或数据进行否定应答。要在从模式下实现 ACK 控制，必须通过将 I2C_CR1 寄存器中的 SBC 位置 1 来使能从字节控制模式。更多详细信息，请参见*第 1400 页的从器件字节控制模式*。

特定地址（从模式）

必要时应使能特定的 SMBus 地址。更多详细信息，请参见*第 1423 页的总线空闲检测*。

- 通过将 I2C_CR1 寄存器中的 SMBDEN 位置 1 来使能 SMBus 器件默认地址 (0b1100 001)。
- 通过将 I2C_CR1 寄存器中的 SMBHEN 位置 1 来使能 SMBus 主机地址 (0b0001 000)。
- 通过将 I2C_CR1 寄存器中的 ALERTEN 位置 1 来使能报警响应地址 (0b0001100)。

数据包错误校验

通过将 I2C_CR1 寄存器中的 PECEN 位置 1 来使能 PEC 的计算。然后，借助硬件字节计数器（I2C_CR2 寄存器中的 NBYTES[7:0]）来管理 PEC 传输。使能 I2C 之前，必须配置 PECEN 位。

PEC 传输由硬件字节计数器来管理，因此在从模式下连接 SMBus 时必须将 SBC 位置 1。当 PECBYTE 位置 1 且 RELOAD 位清零时，传输完 NBYTES-1 字节的数据后会传输 PEC。如果 RELOAD 置 1，PECBYTE 将不起作用。

注意： 使能 I2C 时，不允许更改 PECEN 配置。

表 255. 带 PEC 的 SMBUS 配置

模式	SBC 位	RELOAD 位	AUTOEND 位	PECBYTE 位
主 Tx/Rx NBYTES + PEC+ STOP	x	0	1	1
主 Tx/Rx NBYTES + PEC + ReSTART	x	0	0	1
从 Tx/Rx + PEC	1	0	x	1

超时检测

将 I2C_TIMEOUTR 寄存器中的 TIMOUTEN 和 TEXTEN 位置 1 来使能超时检测。定时器必须按如下方式编程：即在 SMBus 规范规定的时间最大值之前检测出超时情况。

- $t_{TIMEOUT}$ 检查
  要使能 $t_{TIMEOUT}$ 检查，必须将 12 位 TIMEOUTA[11:0] 位编程为定时器重载值，以检查 $t_{TIMEOUT}$ 参数。必须将 TIDLE 位配置为“0”，以检测 SCL 低电平超时。
  然后，通过将 I2C_TIMEOUTR 寄存器中的 TIMOUTEN 位置 1 来使能定时器。
  如果 SCL 的低电平持续时间超过 (TIMEOUTA+1) x 2048 x t_{I2CCLK}，I2C_ISR 寄存器中的 TIMEOUT 标志将置 1。
  请参见*表 256：不同 I2CCLK 频率下的 TIMEOUTA 设置示例（最大 tTIMEOUT = 25 ms）*。

注意： TIMEOUTEN 位置 1 时，不允许更改 TIMEOUTA[11:0] 位和 TIDLE 位的配置。

- $t_{LOW:SEXT}$ 和 $t_{LOW:MEXT}$ 检查
  必须根据外设配置为主器件还是从器件来配置 TIMEOUTB 定时器，以便为从器件校验 $t_{LOW:SEXT}$，为主器件校验 $t_{LOW:MEXT}$。由于标准只规定了最大值，用户可以为这两个参数选择相同的值。
  然后，通过将 I2C_TIMEOUTR 寄存器中的 TEXTEN 位置 1 来使能定时器。
  如果 SMBus 外设延展 SCL 的累积时间超过 (TIMEOUTB+1) x 2048 x t_{I2CCLK}，并且达到*第 1423 页的总线空闲检测*一节给出的超时间隔，则 I2C_ISR 寄存器中的 TIMEOUT 标志将置 1。
  请参见*表 257：不同 I2CCLK 频率下的 TIMEOUTB 设置示例*。

注意： TEXTEN 位置 1 时，不允许更改 TIMEOUTB 配置。

总线空闲检测

要使能 t_{IDLE} 检查，必须将 12 位 TIMEOUTA[11:0] 字段编程为定时器重载值，以获取 t_{IDLE} 参数。必须将 TIDLE 位配置为“1”，以检测 SCL 和 SDA 高电平超时。

然后，通过将 I2C_TIMEOUTR 寄存器中的 TIMOUTEN 位置 1 来使能定时器。

如果 SCL 和 SDA 线的高电平持续时间超过 (TIMEOUTA+1) x 4 x t_{I2CCLK}，I2C_ISR 寄存器中的 TIMEOUT 标志将置 1。

请参见*表 258：不同 I2CCLK 频率下的 TIMEOUTA 设置示例（最大 tIDLE = 50 µs）*。

注意： TIMEOUTEN 置 1 时，不允许更改 TIMEOUTA 和 TIDLE 配置。

43.4.12 SMBus：I2C_TIMEOUTR 寄存器配置示例

仅当支持 SMBus 功能时，才涉及本节内容。请参见*第 43.3 节：I2C 特性实现*。

- 将 $t_{TIMEOUT}$ 的最大持续时间配置为 25 ms：

表 256. 不同 I2CCLK 频率下的 TIMEOUTA 设置示例（最大 $t_{TIMEOUT}$ = 25 ms）

f_{I2CCLK}	TIMEOUTA[11:0] 位	TIDLE 位	TIMEOUTEN 位	$t_{TIMEOUT}$
8 MHz	0x61	0	1	98 x 2048 x 125 ns = 25 ms
16 MHz	0xC3	0	1	196 x 2048 x 62.5 ns = 25 ms
32 MHz	0x186	0	1	391 x 2048 x 31.25 ns = 25 ms
48 MHz	0x249	0	1	586 x 2048 x 20.08 ns = 25 ms

- 将 $t_{LOW:SEXT}$ 和 $t_{LOW:MEXT}$ 的最大持续时间配置为 8 ms：

表 257. 不同 I2CCLK 频率下的 TIMEOUTB 设置示例

f_{I2CCLK}	TIMEOUTB[11:0] 位	TEXTEN位	$t_{LOW:EXT}$
8 MHz	0x1F	1	32 x 2048 x 125 ns = 8 ms
16 MHz	0x3F	1	64 x 2048 x 62.5 ns = 8 ms
48 MHz	0xBB	1	188 x 2048 x 20.08 ns = 8 ms

- 将 t_{IDLE} 的最大持续时间配置为 50 µs

表 258. 不同 I2CCLK 频率下的 TIMEOUTA 设置示例（最大 t_{IDLE} = 50 µs）

f_{I2CCLK}	TIMEOUTA[11:0] 位	TIDLE 位	TIMEOUTEN 位	t_{TIDLE}
8 MHz	0x63	1	1	100 x 4 x 125 ns = 50 µs
16 MHz	0xC7	1	1	200 x 4 x 62.5 ns = 50 µs
48 MHz	0x257	1	1	600 x 4 x 20.08 ns = 50 µs

43.4.13 SMBus 从模式

仅当支持 SMBus 功能时，才涉及本节内容。请参见*第 43.3 节：I2C 特性实现*。

除了 I2C 从模式传输管理（请参见*第 43.4.7 节：I2C 从模式*）之外，还提供了一些额外的软件流程图来支持 SMBus。

SMBus 从发送器

在 SMBus 模式下作为从发送器时，必须将 SBC 编程为“1”，以便在完成已编程数据字节数的传输后进行 PEC 传输。当 PECBYTE 位置 1 时，NBYTES[7:0] 中编程的字节数包含 PEC 传输。在这种情况下，总 TXIS 中断数为 NBYTES-1，如果主器件在完成 NBYTES-1 字节的数据传输后请求传输额外的字节，则将自动发送 I2C_PECR 寄存器的内容。

注意： 当 RELOAD 位置 1 时，PECBYTE 位将不起作用。

图 444. SMBus 从发送器的传输序列流程图（N 字节 + PEC）

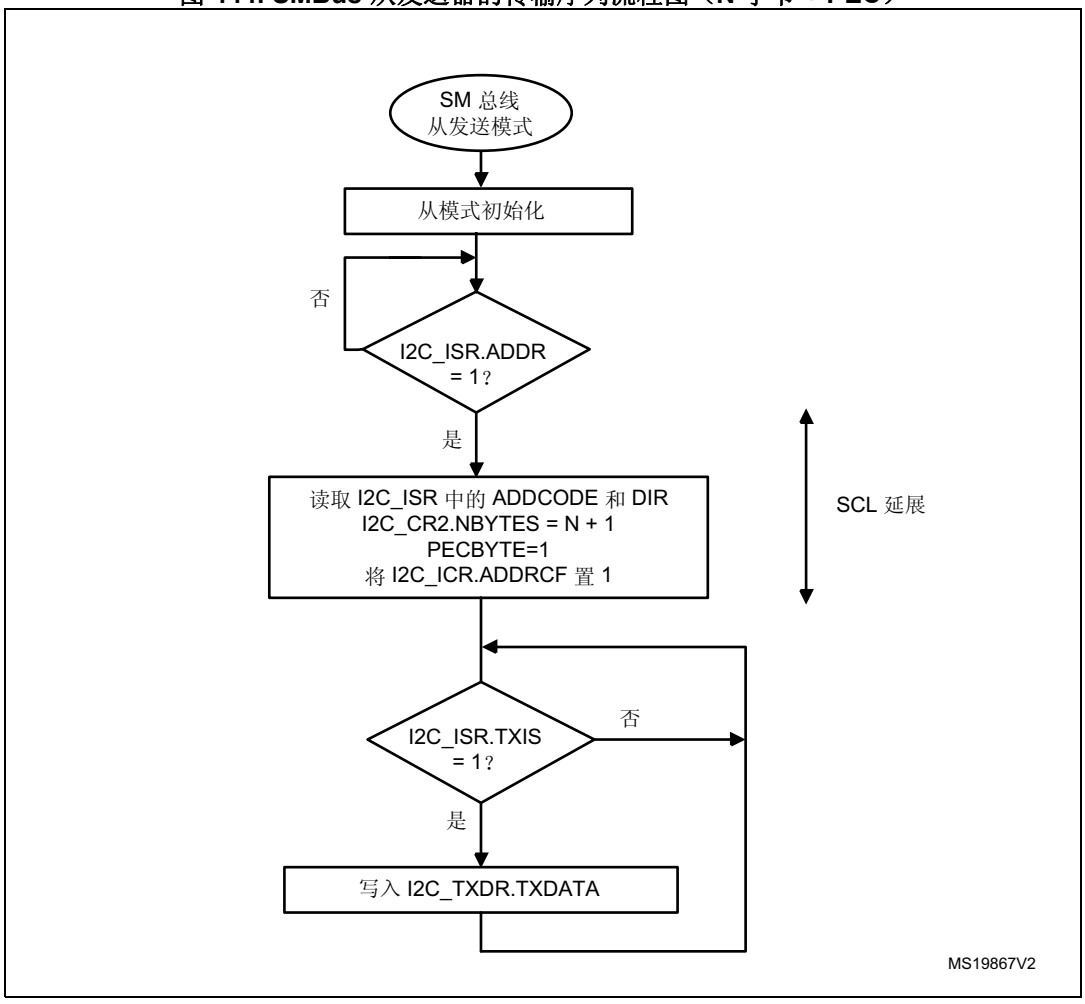
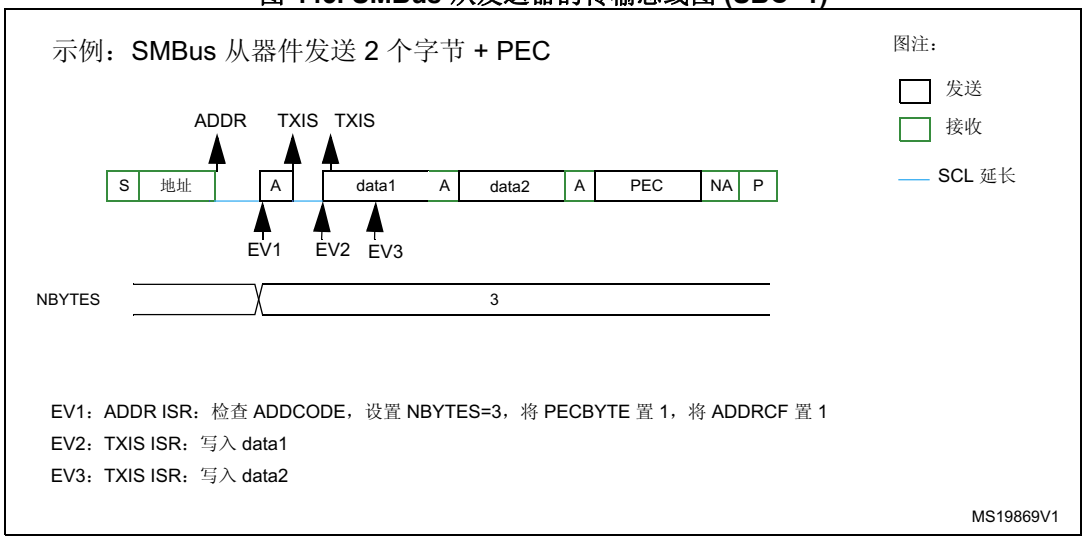


图 445. SMBus 从发送器的传输总线图 (SBC=1)



SMBus 从接收器

在 SMBus 模式下使用 I2C 时，必须将 SBC 编程为“1”，以便在完成已编程数据字节数的传输后进行 PEC 校验。要对每个字节进行 ACK 控制，必须选择重载模式 (RELOAD=1)。更多详细信息，请参见*第 1400 页的从器件字节控制模式*。

要校验 PEC 字节，必须将 RELOAD 位清零并将 PECBYTE 位置 1。在这种情况下，当接收到 NBYTES-1 字节的数据后，接收的下一个字节将与内部 I2C_PECR 寄存器的内容作比较。如果比较不匹配，则将自动生成 NACK 信号；如果比较匹配，则将自动生成 ACK 信号，而与 ACK 位的值无关。PEC 字节一经接收，便会像任何其他数据一样复制到 I2C_RXDR 寄存器中，并且 RXNE 标志将置 1。

当 PEC 不匹配时，PECERR 标志将置 1，如果 I2C_CR1 寄存器中的 ERRIE 位置 1，还将生成中断。

如果无需 ACK 软件控制，用户可编程 PECBYTE=1，在同一写操作下，将 NBYTES 编程为连续接收的字节数。接收到 NBYTES-1 字节的数据后，会将接收的下一个字节视为 PEC 进行校验。

注意： 当 RELOAD 位置 1 时，PECBYTE 位将不起作用。

图 446. SMBus 从接收器的传输序列流程图（N 字节 + PEC）

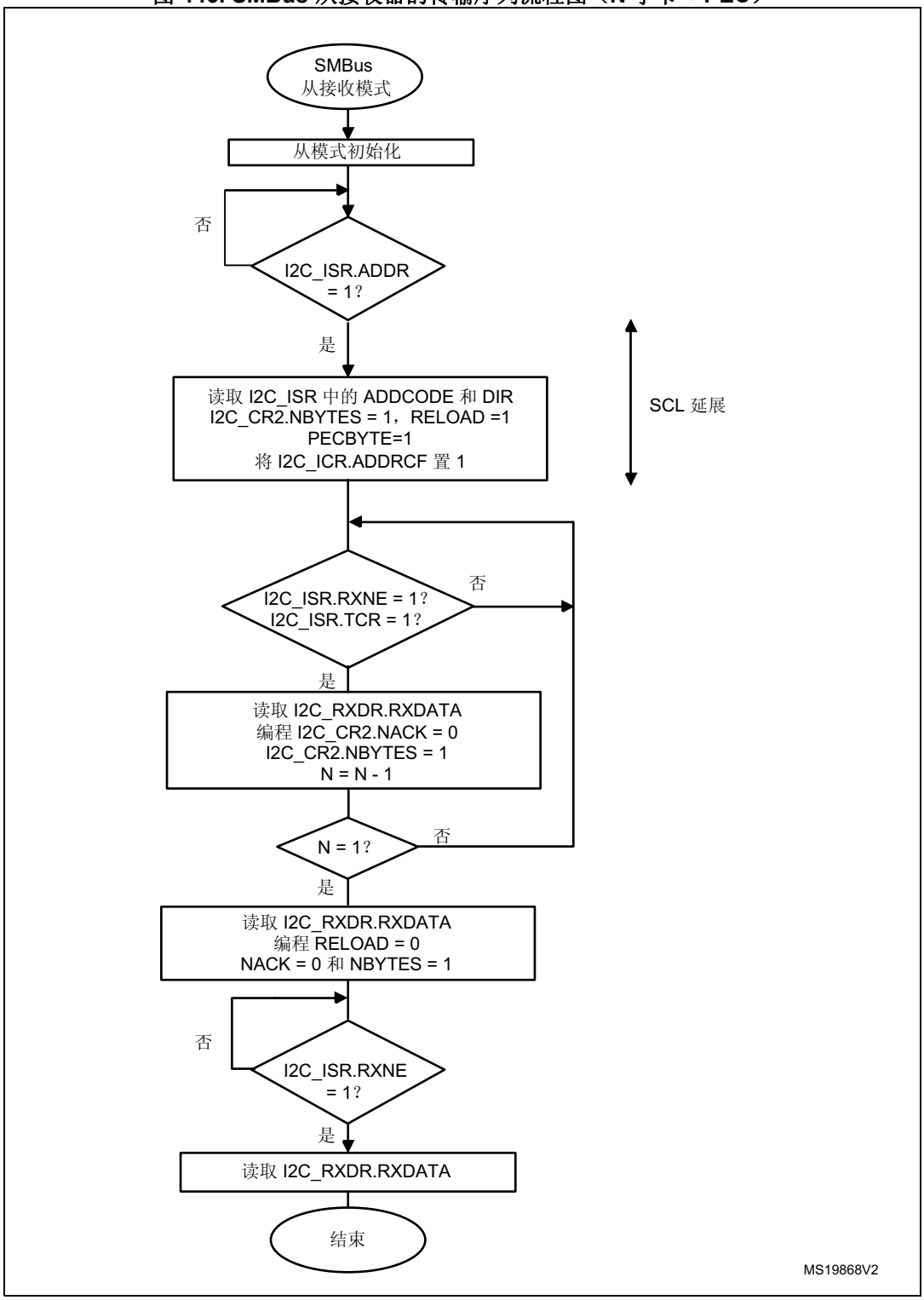
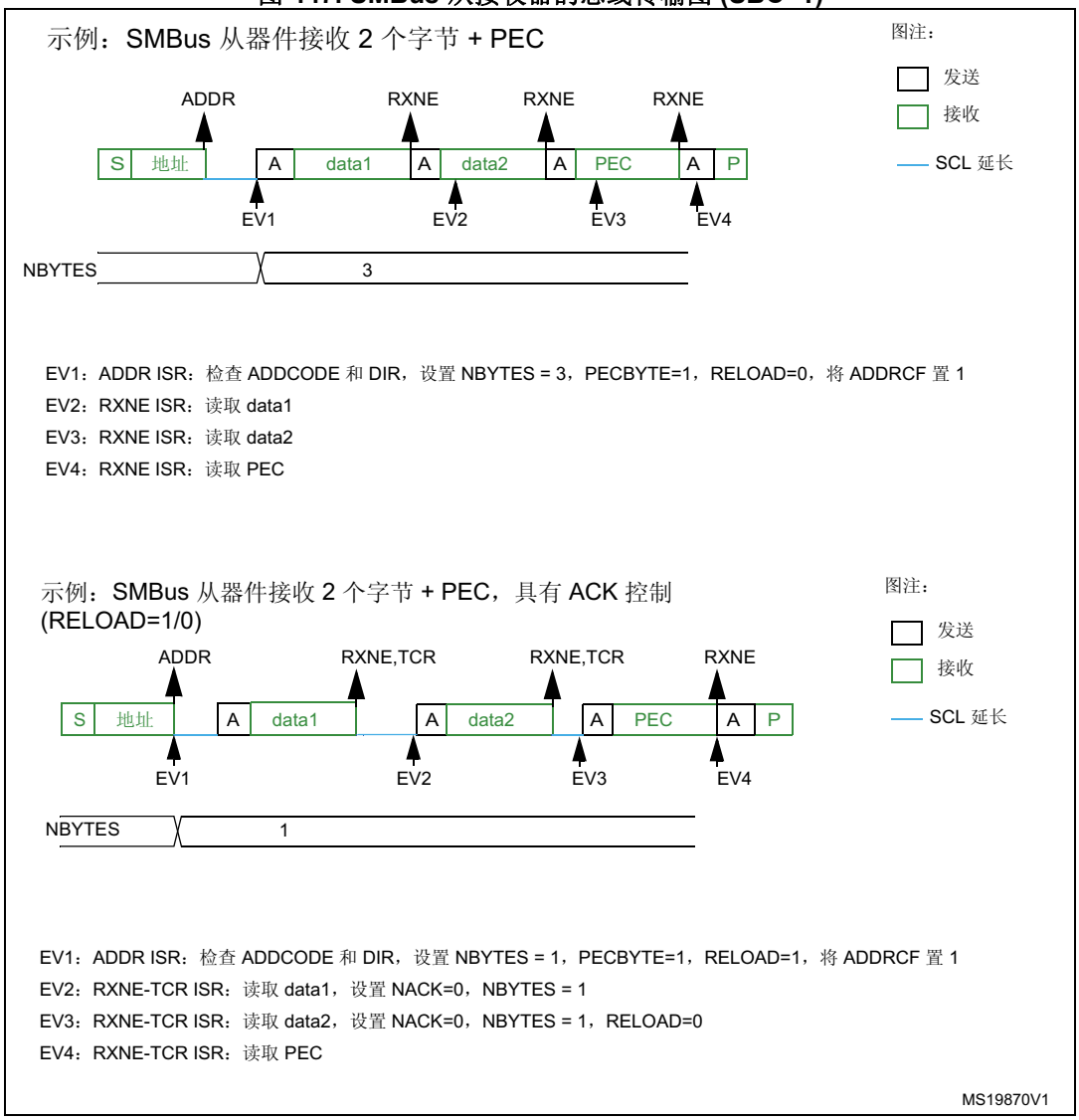


图 447. SMBus 从接收器的总线传输图 (SBC=1)



仅当支持 SMBus 功能时，才涉及本节内容。请参见*第 43.3 节：I2C 特性实现*。

除了 I2C 主模式传输管理（请参见*第 43.4.8 节：I2C 主模式*）之外，还提供了一些额外的软件流程图来支持 SMBus。

SMBus 主发送器

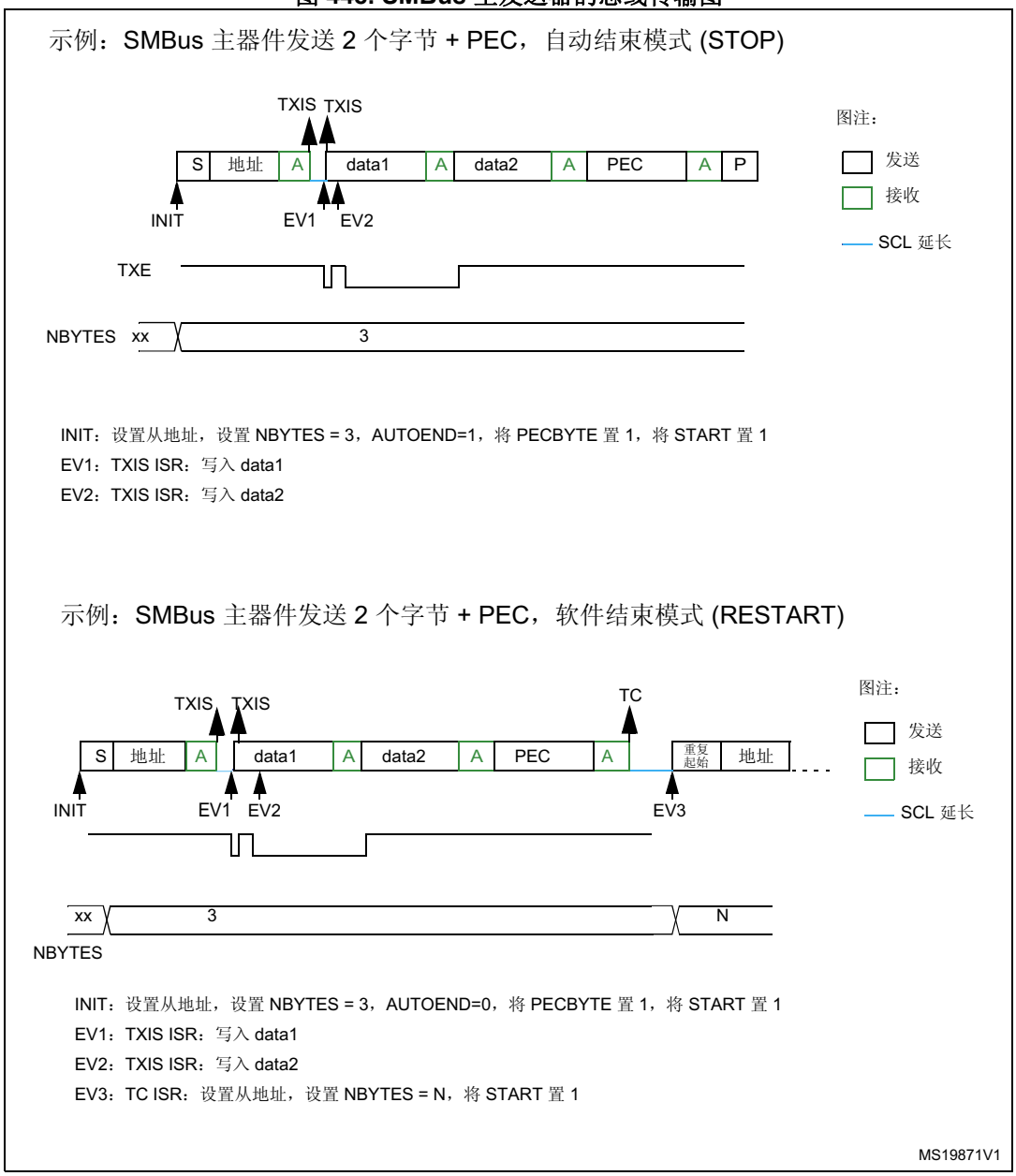
当 SMBus 主器件想要发送 PEC 时，必须在 START 位置 1 前，将 PECBYTE 位置 1 并在 NBYTES[7:0] 字段中设置字节数。在这种情况下，总 TXIS 中断数为 NBYTES-1。因此，如果 PECBYTE 位在 NBYTES=0x1 时置 1，则将自动发送 I2C_PECR 寄存器的内容。

如果 SMBus 主器件想要在 PEC 后发送停止位，则应选择自动结束模式 (AUTOEND=1)。在这种情况下，传输 PEC 后将自动发送停止位。

如果 SMBus 主器件想要在 PEC 后发送重复起始位，则必须选择软件模式 (AUTOEND=0)。在这种情况下，发送 NBYTES-1 字节的数据后，将发送 I2C_PECR 寄存器的内容，TC 标志将在传输完 PEC 之后置 1，SCL 线的低电平时间将延长。必须在 TC 中断子程序中设置重复起始位。

注意： 当 RELOAD 位置 1 时，PECBYTE 位将不起作用。

图 448. SMBus 主发送器的总线传输图



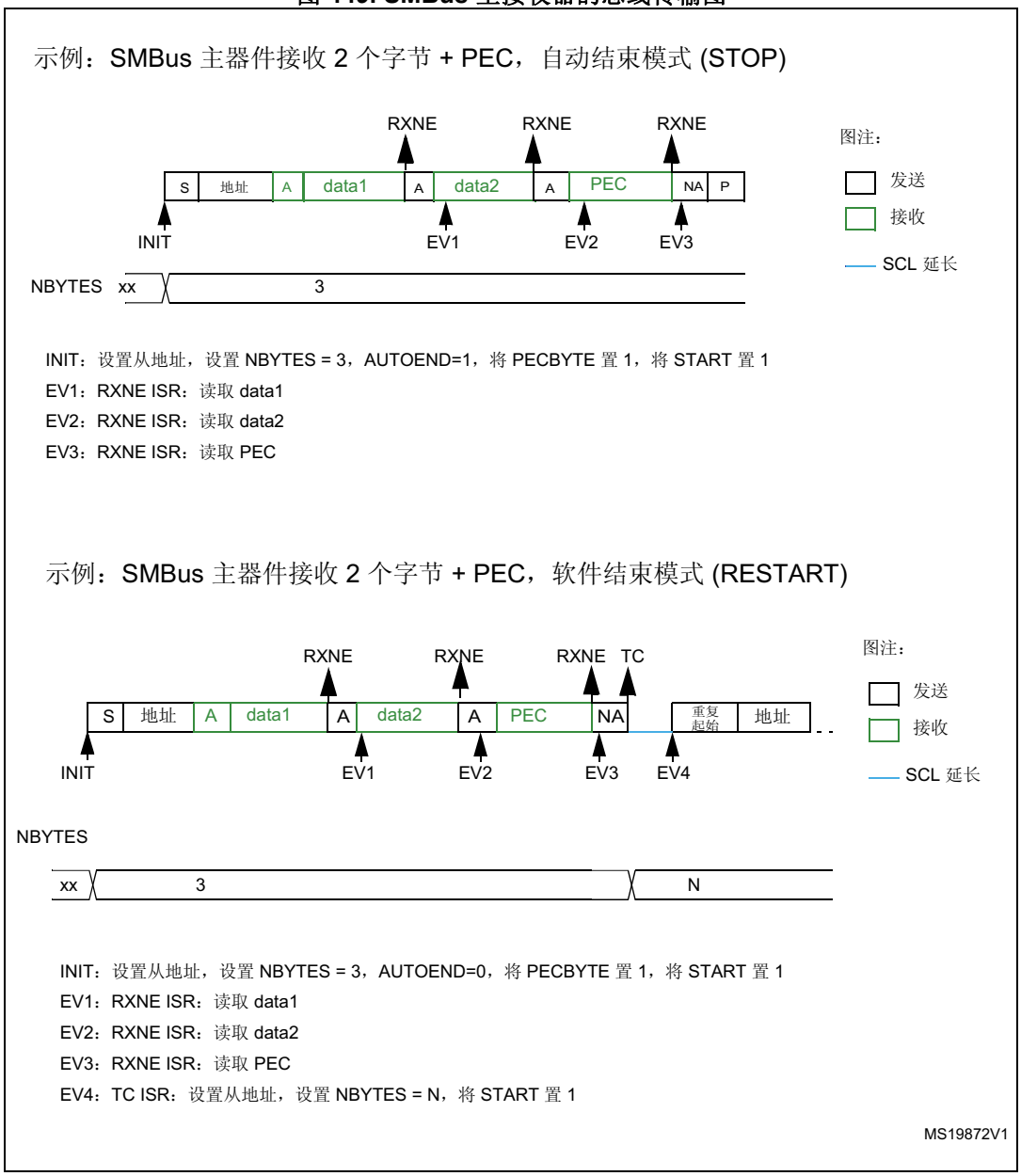
SMBus 主接收器

当 SMBus 主器件想要接收 PEC，并在传输结束后发送 STOP 时，可选择自动结束模式 (AUTOEND=1)。将 START 位置 1 之前，必须将 PECBYTE 位置 1 并设置从地址。在这种情况下，当接收到 NBYTES-1 字节的数据后，将自动使用 I2C_PECR 寄存器的内容对接收的下一个字节进行校验。接收 PEC 字节后，给出 NACK 响应和停止位。

当 SMBus 主接收器想要接收 PEC 字节，并且在传输结束后发送重复起始位时，必须选择软件模式 (AUTOEND=0)。将 START 位置 1 之前，必须将 PECBYTE 位置 1 并设置从地址。在这种情况下，当接收到 NBYTES-1 字节的数据后，将自动使用 I2C_PECR 寄存器的内容对接收的下一个字节进行校验。接收到 PEC 字节后，TC 标志将置 1，SCL 线的低电平时间将延长。可以在 TC 中断子程序中设置重复起始位。

注意： 当 RELOAD 位置 1 时，PECBYTE 位将不起作用。

图 449. SMBus 主接收器的总线传输图



43.4.14 地址匹配时从停止模式唤醒

仅当支持从停止模式唤醒功能时，才涉及本节内容。请参见*第 43.3 节：I2C 特性实现*。

被寻址时，I2C能够从停止模式中唤醒MCU（APB时钟关断）。支持所有寻址模式。

将 I2C_CR1 寄存器中的 WUPEN 位置 1，可以使能从停止模式唤醒功能。对于 I2CCLK，必须选择 HSI16 振荡器作为时钟源，以便从停止模式唤醒。

停止模式下，关闭 HSI16。当检测到 START 时，I2C 接口将 HSI16 接通，并延长 SCL 使其处于低电平直到唤醒 HSI16。

HSI16 随后用来接收地址。

地址匹配的情况下，MCU 唤醒时间内，I2C 延长 SCL 使其处于低电平。当软件清除 ADDR 标志时，此延长被释放，传输正常进行。

如果地址不匹配，HSI16 再次关断，MCU不被唤醒。

注： *如果 I2C 时钟是系统时钟，或者 WUPEN = 0，则接收到 START 后，HSI16 不会接通。*

只有 ADDR 中断能够唤醒 MCU。因此，当 I2C 以主器件身份或在 ADDR 标志置 1 后以被寻址从器件身份执行传输时，不要进入停止模式。通过在 ADDR 中断程序中清除 SLEEPDEEP 位，然后仅在 STOPF 标志置 1 后再将其置 1，来对此进行管理。

注意： 数字滤波器与从停止模式唤醒功能不兼容。如果 DNF 位不等于 0，则将 WUPEN 位置 1 将不起任何作用。

注意： 只有当 I2C 时钟源为 HSI16 振荡器时，该功能才可用。

注意： 必须使能时钟延长 (NOSTRETCH=0) 才能确保从停止模式唤醒功能正常工作。

注意： 如果禁止从停止模式唤醒 (WUPEN=0)，则在进入停止模式前必须禁止 I2C 外设 (PE=0)。

43.4.15 错误条件

以下错误条件可能导致通信失败。

总线错误 (BERR)

当检测到起始位或停止位但不位于第 9N 个 SCL 时钟脉冲之后时，会检测到总线错误。当 SDA 边沿出现且 SCL 为高电平时，会检测到起始或停止位。

只有当 I2C 在传输过程中用作主器件或被寻址为从器件时（即未处于从模式下的地址阶段），才会将总线错误标志置 1。

在从模式下检测到错位的起始位或重复起始位时，I2C 会像接收到正确的起始位一样进入地址识别状态。

检测到总线错误时，I2C_ISR 寄存器中的 BERR 标志将置 1，如果 I2C_CR1 寄存器中的 ERRIE 位置 1，还将生成中断。

仲裁丢失 (ARLO)

当 SDA 线上发送高电平但在 SCL 上升沿却采样到低电平时，会检测到仲裁丢失。

- 在主模式下，将在地址阶段、数据阶段和数据应答阶段检测到仲裁丢失。在这种情况下，SDA 线和 SCL 线被释放，起始控制位由硬件清零，主器件自动切换为从模式。
- 在从模式下，将在数据阶段和数据应答阶段检测到仲裁丢失。在这种情况下，传输停止，SCL 和 SDA 线被释放。

检测到仲裁丢失时，I2C_ISR 寄存器中的 ARLO 标志将置 1，如果 I2C_CR1 寄存器中的 ERRIE 位置 1，还将生成中断。

上溢/下溢错误 (OVR)

当满足 NOSTRETCH=1 和以下条件时，将在从模式下检测到上溢或下溢错误：

- 接收过程中接收到一个新字节但尚未读取 RXDR 寄存器。接收的新字节丢失，自动发送 NACK 来响应新字节。
- 在发送过程中：
  - 当 STOPF=1 且应发送第一个数据字节时。TXE=0 时发送 I2C_TXDR 寄存器的内容，否则发送 0xFF。
  - 应发送一个新字节但尚未向 I2C_TXDR 寄存器写入数据时，将发送 0xFF。

检测到上溢或下溢错误时，I2C_ISR 寄存器中的 OVR 标志将置 1，如果 I2C_CR1 寄存器中的 ERRIE 位置 1，还将生成中断。

数据包错误校验错误 (PECERR)

仅当支持 SMBus 功能时，才涉及本节内容。请参见*第 43.3 节：I2C 特性实现*。

当接收到的 PEC 字节与 I2C_PECR 寄存器的内容不匹配时，将检测到 PEC 错误。接收到错误的 PEC 后，将自动发送 NACK。

检测到 PEC 错误时，I2C_ISR 寄存器中的 PECERR 标志将置 1，如果 I2C_CR1 寄存器中的 ERRIE 位置 1，还将生成中断。

超时错误 (TIMEOUT)

仅当支持 SMBus 功能时，才涉及本节内容。请参见*第 43.3 节：I2C 特性实现*。

满足以下任何条件均会出现超时错误：

- TIDLE=0 且 SCL 的低电平持续时间达到 TIMEOUTA[11:0] 位中定义的时间：这用于检测 SMBus 超时。
- TIDLE=1 且 SDA 和 SCL 的高电平持续时间达到 TIMEOUTA[11:0] 位中定义的时间：这用于检测总线空闲情况。
- 主器件的累积时钟低电平延长时间达到了 TIMEOUTB[11:0] 位中定义的时间（SMBus $t_{LOW:MEXT}$ 参数）
- 从器件的累积时钟低电平延长时间达到了 TIMEOUTB[11:0] 位中定义的时间（SMBus $t_{LOW:SEXT}$ 参数）

当在主模式下检测到超时时，将自动发送停止位。

当在从模式下检测到超时时，将自动释放 SDA 和 SCL 线。

检测到超时错误时，I2C_ISR 寄存器中的 TIMEOUT 标志将置 1，如果 I2C_CR1 寄存器中的 ERRIE 位置 1，还将生成中断。

报警 (ALERT)

仅当支持 SMBus 功能时，才涉及本节内容。请参见*第 43.3 节：I2C 特性实现*。

当 I2C 接口配置为主机 (SMBHEN=1)、使能了报警引脚检测 (ALERTEN=1) 并且在 SMBA 引脚上检测到下降沿时，ALERT 标志将置 1。如果 I2C_CR1 寄存器中的 ERRIE 位置 1，将生成中断。

43.4.16 DMA 请求

使用 DMA 进行发送

将 I2C_CR1 寄存器中的 TXDMAEN 位置 1 可以使能 DMA（直接存储器访问）进行发送。当 TXIS 位置 1 时，数据将从由 DMA 外设配置的 SRAM 区（请参见*第 331 页的第 11 节：直接存储器访问控制器 (DMA)*）装载进 I2C_TXDR 寄存器。

只有数据字节采用 DMA 进行传输。

- 在主模式下：初始化、从地址、方向、字节数和起始位均由软件编程（发送的从地址无法通过 DMA 传输）。当所有数据均通过 DMA 传输时，必须在起始位置 1 之前初始化 DMA。传输结束由 NBYTES 计数器来管理。请参见*第 1411 页的主发送器*。
- 在从模式下：
  - 当 NOSTRETCH=0 时，如果所有数据均通过 DMA 传输，则必须在地址匹配事件之前（或清零 ADDR 之前在 ADDR 中断子程序中）初始化 DMA。
  - 当 NOSTRETCH=1 时，必须在地址匹配事件之前初始化 DMA。
- 支持 SMBus 时：PEC 传输由 NBYTES 计数器管理。请参见*第 1426 页的SMBus 从发送器*和*第 1429 页的SMBus 主发送器*。

注：如果使用 DMA 进行发送，则无需使能 TXIE 位。

使用 DMA 进行接收

将 I2C_CR1 寄存器中的 RXDMAEN 位置 1 可以使能 DMA（直接存储器访问）进行接收。当 RXNE 位置 1 时，数据将从 I2C_RXDR 寄存器装载进由 DMA 外设配置的 SRAM 区（请参见*第 331 页的第 11 节：直接存储器访问控制器 (DMA)*）。只有数据字节（包括 PEC）采用 DMA 进行传输。

- 在主模式下：初始化、从地址、方向、字节数和起始位均由软件编程。当所有数据均通过 DMA 传输时，必须在起始位置 1 之前初始化 DMA。传输结束由 NBYTES 计数器来管理。
- 在从模式下，当 NOSTRETCH=0 时，如果所有数据均通过 DMA 传输，则必须在地址匹配事件之前（或清零 ADDR 标志之前在 ADDR 中断子程序中）初始化 DMA。
- 如果支持 SMBus（请参见*第 43.3 节：I2C 特性实现*）：PEC 传输由 NBYTES 计数器管理。请参见*第 1427 页的SMBus 从接收器*和*第 1431 页的SMBus 主接收器*。

注：如果使用 DMA 进行接收，则无需使能 RXIE 位。

43.4.17 调试模式

当微控制器进入调试模式时（内核停止），SMBus 超时定时器会根据 DBG 模块中的 DBG_I2Cx_SMBUS_TIMEOUT 配置位选择继续正常工作还是停止工作。

43.5 I2C 低功耗模式

表 259. 低功耗模式对 I2C 的影响

模式	说明
睡眠	无影响。I2C 中断可使器件退出睡眠模式。
低功耗运行	无影响。
低功耗睡眠	无影响。I2C 中断可使器件退出低功耗睡眠模式。
停止 0/停止 1	I2C 模块的寄存器内容仍被保持。如果 WUPEN = 1，并且 I2C 的时钟由 HSI16 提供：地址识别功能正常。I2C 地址匹配条件会导致器件退出停止 0 和停止 1 模式。如果 WUPEN=0：必须在进入停止模式之前禁止 I2C。
停止 2	I2C 模块的寄存器内容仍被保持。必须在进入停止 2 模式之前禁止 I2C1、I2C2 和 I2C4。如果 WUPEN = 1，并且 I2C3 的时钟由 HSI16 提供：地址识别功能正常。I2C3 地址匹配条件会导致器件退出停止 2 模式。如果 WUPEN=0：必须在进入停止 2 模式之前禁止 I2C3。
待机	I2C 外设掉电，退出待机模式或关断模式后必须重新初始化。
关断	

43.6 I2C 中断

下表给出了 I2C 中断请求列表。

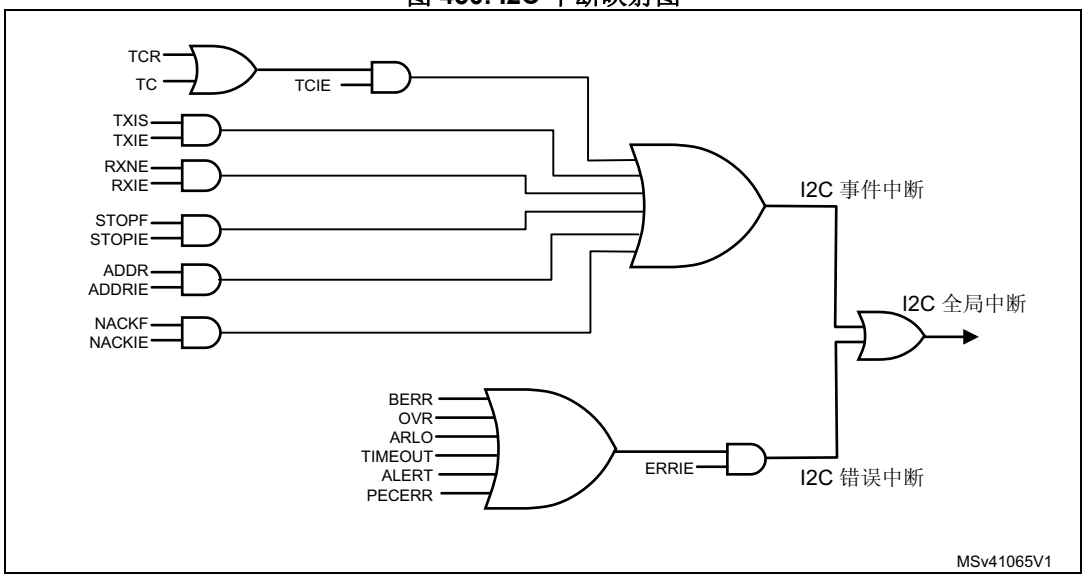
表 260. I2C 中断请求

中断事件	事件标志	事件标志/中断清除方法	中断使能控制位
接收缓冲区非空	RXNE	读取 I2C_RXDR 寄存器	RXIE
发送缓冲区中断状态	TXIS	写入 I2C_TXDR 寄存器	TXIE
停止位检测中断标志	STOPF	写入 STOPCF=1	STOPIE
传输完成等待重载	TCR	写入 I2C_CR2 (NBYTES[7:0] ≠ 0)	TCIE
传输完成	TC	写入 START=1 或 STOP=1	
地址匹配	ADDR	写入 ADDRCF=1	ADDRIE
接收到 NACK 应答	NACKF	写入 NACKCF=1	NACKIE
总线错误	BERR	写入 BERRCF=1	ERRIE
仲裁丢失	ARLO	写入 ARLOCF=1	
上溢/下溢	OVR	写入 OVRCF=1	
PEC 错误	PECERR	写入 PECERRCF=1	
超时/t_{LOW} 错误	TIMEOUT	写入 TIMEOUTCF=1	
SMBus 报警	ALERT	写入 ALERTCF=1	

根据产品实现的不同，上述所有中断既可以共享同一个中断向量（I2C 全局中断），也可以分配到 2 个不同的中断向量上（I2C 事件中断和 I2C 错误中断）。有关详细信息，请参见*表 67：STM32L4Rxxx 和 STM32L4Sxxx 向量表*。

I2C 唤醒事件连接到 EXTI 控制器（请参见*第 16 节：扩展中断和事件控制器 (EXTI)*）。

图 450. I2C 中断映射图



43.7 I2C 寄存器

有关寄存器说明中使用的缩写，请参见*第 72 页的第 1.1 节*。

外设寄存器按字（32 位）进行访问。

43.7.1 控制寄存器 1 (I2C_CR1)

Control register 1

偏移地址：0x00

复位值：0x0000 0000

访问：无等待周期，正在对该寄存器执行写访问时出现另一个写访问的情况除外。在这种情况下，会在第二个写访问中插入等待周期，直到前一个写访问完成为止。第二个写访问的延时最长可达 2 x PCLK1 + 6 x I2CCLK。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PECEN	ALERTEN	SMBDEN	SMBHEN	GCEN	WUPEN	NOSTRETCH	SBC
								rw	rw	rw	rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RXDMAEN	TXDMAEN	Res.	ANFOFF	DNF				ERRIE	TCIE	STOPIE	NACKIE	ADDRIE	RXIE	TXIE	PE
rw	rw		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 **PECEN**：PEC 使能 (PEC enable)

0：禁止 PEC 计算
1：使能 PEC 计算

注： 如果不支持 SMBus 功能，该位保留，并由硬件强制为“0”。请参见第 43.3 节：I2C 特性实现。

位 22 **ALERTEN**：SMBus 报警使能 (SMBus alert enable)

从机模式 (SMBHEN=0)：

0：将 SMBA 引脚释放为高电平并禁止报警响应地址头：0001100x 后跟 NACK。
1：将 SMBA 引脚驱动为低电平并使能报警响应地址头：0001100x 后跟 ACK。

主机模式 (SMBHEN=1)：

0：不支持 SMBus 报警引脚 (SMBA)。
1：支持 SMBus 报警引脚 (SMBA)。

注： 当 ALERTEN=0 时，SMBA 引脚可用作标准 GPIO。
如果不支持 SMBus 功能，该位保留，并由硬件强制为“0”。请参见第 43.3 节：I2C 特性实现。

位 21 **SMBDEN**：SMBus 器件默认地址使能 (SMBus Device Default address enable)

0：禁止器件默认地址。不对地址 0b1100001x 应答。
1：使能器件默认地址。对地址 0b1100001x 应答。

注： 如果不支持 SMBus 功能，该位保留，并由硬件强制为“0”。请参见第 43.3 节：I2C 特性实现。

位 20 **SMBHEN**：SMBus 主机地址使能 (SMBus Host address enable)

0：禁止主机地址。不对地址 0b0001000x 应答。
1：使能主机地址。对地址 0b0001000x 应答。

注： 如果不支持 SMBus 功能，该位保留，并由硬件强制为“0”。请参见第 43.3 节：I2C 特性实现。

位 19 **GCEN**：广播呼叫使能 (General call enable)

0：禁止广播呼叫。不对地址 0b00000000 应答。
1：使能广播呼叫。对地址 0b00000000 应答。

位 18 **WUPEN**：从停止模式唤醒使能 (Wakeup from Stop mode enable)

0：禁止从停止模式唤醒。
1：使能从停止模式唤醒。

注： 如果不支持从停止模式唤醒功能，该位保留并由硬件强制清零。请参见第 43.3 节：I2C 特性实现。

注： 只有当 DNF = “0000”时，WUPEN 才能置 1

位 17 **NOSTRETCH**：时钟延长禁止 (Clock stretching disable)

该位用于在从模式下禁止时钟延长。它在主模式下必须保持清零。

0：使能时钟延长
1：禁止时钟延长

注： 该位只能在 I2C 禁止时 (PE = 0) 编程。

位 16 **SBC**：从设备模式下的字节控制 (Slave byte control)

该位用于在从设备模式下使能硬件字节控制。

0：禁止从设备模式下的字节控制
1：使能从设备模式下的字节控制

位 15 **RXDMAEN**：DMA 接收请求使能 (DMA reception requests enable)

0：禁止 DMA 接收请求
1：使能 DMA 接收请求

位 14 **TXDMAEN**：DMA 发送请求使能 (DMA transmission requests enable)
0：禁止 DMA 发送请求
1：使能 DMA 发送请求

位 13 保留，必须保持复位值。

位 12 **ANFOFF**：模拟噪声滤波器关闭 (Analog noise filter OFF)
0：使能模拟噪声滤波器
1：禁止模拟噪声滤波器
注：　该位只能在 I2C 禁止时 (PE = 0) 编程。

位 11:8 **DNF[3:0]**：数字噪声滤波器 (Digital noise filter)
这些位用于配置 SDA 和 SCL 输入端的数字噪声滤波器。数字滤波器可滤除脉宽 DNF[3:0] * t_{I2CCLK} 以下的尖峰
0000：禁止数字滤波器
0001：使能数字滤波器，可滤除的噪声尖峰脉宽可达 1 t_{I2CCLK}
...
1111：使能数字滤波器，可滤除的噪声尖峰脉宽可达 15 t_{I2CCLK}
注：　如果模拟滤波器也已使能，数字滤波将叠加在模拟滤波之上。
该滤波器只能在 I2C 禁止时 (PE = 0) 编程。

位 7 **ERRIE**：错误中断使能 (Error interrupts enable)
0：禁止错误检测中断
1：使能错误检测中断
注：　以下任一错误均会生成中断：
仲裁丢失 (ARLO)
总线错误检测 (BERR)
上溢/下溢 (OVR)
超时检测 (TIMEOUT)
PEC 错误检测 (PECERR)
报警引脚事件检测 (ALERT)

位 6 **TCIE**：传输完成中断使能 (Transfer complete interrupt enable)
0：禁止传输完成中断
1：使能传输完成中断
注：　以下任一事件均会生成中断：
传输完成 (TC)
传输完成等待重载 (TCR)

位 5 **STOPIE**：停止位检测中断使能 (STOP detection Interrupt enable)
0：禁止停止位检测 (STOPF) 中断
1：使能停止位检测 (STOPF) 中断

位 4 **NACKIE**：接收到否定应答中断使能 (Not acknowledge received Interrupt enable)
0：禁止接收到否定应答 (NACKF) 中断
1：使能接收到否定应答 (NACKF) 中断

位 3 **ADDRIE**：地址匹配中断使能（仅从模式）(Address match Interrupt enable (slave only))
0：禁止地址匹配 (ADDR) 中断
1：使能地址匹配 (ADDR) 中断

位 2 **RXIE**：RX 中断使能 (RX Interrupt enable)
0：禁止接收 (RXNE) 中断
1：使能接收 (RXNE) 中断

位 1 **TXIE**：TX 中断使能 (TX Interrupt enable)

0：禁止发送 (TXIS) 中断

1：使能发送 (TXIS) 中断

位 0 **PE**：外设使能 (Peripheral enable)

0：禁止外设

1：使能外设

注：　当 PE=0 时，将释放 I2C SCL 线和 SDA 线。内部状态机和状态位均恢复为复位值。清零时，PE 必须保持低电平状态至少 3 个 APB 时钟周期。

43.7.2 控制寄存器 2 (I2C_CR2)

Control register 2

偏移地址：0x04

复位值：0x0000 0000

访问：无等待周期，正在对该寄存器执行写访问时出现另一个写访问的情况除外。在这种情况下，会在第二个写访问中插入等待周期，直到前一个写访问完成为止。第二个写访问的延时最长可达 2 x PCLK1 + 6 x I2CCLK。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	PEC BYTE	AUTO END	RE LOAD	NBYTES[7:0]							
					rs	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
NACK	STOP	START	HEAD 10R	ADD10	RD_ WRN	SADD[9:0]									
rs	rs	rs	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 **PECBYTE**：数据包错误校验字节 (Packet error checking byte)

此位由软件置 1，并可在 PEC 传输完成时、接收到停止位或匹配地址时、或者 PE=0 时由硬件清零。

0：不传输 PEC。

1：请求 PEC 发送/接收。

注：　向该位写入“0”不起作用。

当 RELOAD 置 1 时，该位不起作用。

当 SBC=0 时，该位在从模式下不起作用。

如果不支持 SMBus 功能，该位保留，并由硬件强制为“0”。请参见第 43.3 节：I2C 特性实现。

位 25 **AUTOEND**：自动结束模式（主模式）(Automatic end mode (master mode))

此位由软件置 1 和清零。

0：软件结束模式：当 NBYTES 数据传输完成时，TC 标志将置 1，SCL 的低电平时间将延长直到相应软件操作结束。

1：自动结束模式：当 NBYTES 数据传输完成时，将自动发送停止位。

注：　在从模式下，或当 RELOAD 位置 1 时，该位不起作用。

位 24 **RELOAD**：NBYTES 重载模式 (NBYTES reload mode)

此位由软件置 1 和清零。

0：传输 NBYTES 数据（后跟停止位或重复起始位）之后即完成传输。

1：传输 NBYTES 数据之后未完成传输（将重载 NBYTES）。当 NBYTES 数据传输完成时，TCR 标志将置 1，SCL 的低电平时间将延长直到相应软件操作结束。

位 23:16 **NBYTES[7:0]**：字节数 (Number of bytes)

在此设置待发送/接收的字节数。在从模式下，当 SBC=0 时，该字段为无关字段。

注： START 位置 1 时，不允许更改这些位。

位 15 **NACK**：NACK 生成（从模式）(NACK generation (slave mode))

此位由软件置 1，并可在发送 NACK 时、接收到停止位或匹配地址时、或者 PE=0 时由硬件清零。

0：在当前接收的字节后发送 ACK。

1：在当前接收的字节后发送 NACK。

注： 向该位写入“0”不起作用。

该位仅在从模式下使用：在主接收器模式下，无论 NACK 位的值为何，最后一个字节（后跟停止位或重复起始位）后都将自动生成 NACK。

当从接收器 NOSTRETCH 模式下发生上溢时，无论 NACK 位的值为何，都将自动生成 NACK。

使能硬件 PEC 校验时 (PECBYTE=1)，PEC 应答值与 NACK 值无关。

位 14 **STOP**：停止位生成（主模式）(Stop generation (master mode))

该位由软件置 1，并可在检测到停止位时或 PE = 0 时由硬件清零。

在主模式下：

0：不生成停止位。

1：在当前字节传输完成后生成停止位。

注： 向该位写入“0”不起作用。

位 13 **START**：起始位生成 (Start generation)

该位由软件置 1，并可在发送起始位（后跟地址序列）之后、发生仲裁丢失时、出现超时错误时、或者 PE = 0 时由硬件清零。它也可由软件清零，方法是向 I2C_ICR 寄存器中的 ADDRCF 位写入“1”。

0：不生成起始位。

1：生成重复起始/起始位。

– 如果 I2C 已处于主模式下且 AUTOEND = 0，则将该位置 1 会在 NBYTES 传输结束后且 RELOAD=0 的情况下生成重复起始位。

– 否则，将该位置 1 会在总线释放后立即生成起始位。

注： 向该位写入“0”不起作用。

即使总线繁忙或 I2C 处于从模式，也可将 START 位置 1。

当 RELOAD 置 1 时，该位不起作用。在 10 位寻址模式下，如果在地址的第一部分接收到 NACK，则 START 位不会由硬件清零，主器件将重新发送地址序列（除非 START 位被软件清零）。

位 12 **HEAD10R**：读方向传输时，只发送 10 位地址的前 7 位地址头字节（主接收器模式）(10-bit address header only read direction (master receiver mode))

0：主器件发送完整的 10 位从地址读序列：起始位 + 带写方向的 2 字节 10 位地址 + 重复起始位 + 带读方向的 10 位地址的前 7 位。

1：主器件只发送 10 位地址的前 7 位,后跟读方向。

注： START 位置 1 时，不允许更改此位。

位 11 **ADD10**：10 位寻址模式（主模式）(10-bit addressing mode (master mode))
0：主器件工作在 7 位寻址模式下
1：主器件工作在 10 位寻址模式下
注： START 位置 1 时，不允许更改此位。

位 10 **RD_WRN**：传输方向（主模式）(Transfer direction (master mode))
0：主器件请求写传输。
1：主器件请求读传输。
注： START 位置 1 时，不允许更改此位。

位 9:8 **SADD[9:8]**：从地址位 9:8（主模式）(Slave address bit 9:8 (master mode))
在 7 位寻址模式 (ADD10 = 0) 下：
这些位无意义
在 10 位寻址模式 (ADD10 = 1) 下：
这些位应写入待发送从地址的第 8 和第 9 位
注： START 位置 1 时，不允许更改这些位。

位 7:1 **SADD[7:1]**：从地址位 7:1（主模式）(Slave address bit 7:1 (master mode))
在 7 位寻址模式 (ADD10 = 0) 下：
这些位应写入待发送的 7 位从地址
在 10 位寻址模式 (ADD10 = 1) 下：
这些位应写入待发送从地址的第 1 到第 7 位。
注： START 位置 1 时，不允许更改这些位。

位 0 **SADD0**：从地址位 0（主模式）(Slave address bit 0 (master mode))
在 7 位寻址模式 (ADD10 = 0) 下：
该位无意义
在 10 位寻址模式 (ADD10 = 1) 下：
该位应写入待发送从地址的第 0 位
注： START 位置 1 时，不允许更改这些位。

43.7.3 设备自身地址 1 寄存器 (I2C_OAR1)

Own address 1 register

偏移地址：0x08

复位值：0x0000 0000

访问：无等待周期，正在对该寄存器执行写访问时出现另一个写访问的情况除外。在这种情况下，会在第二个写访问中插入等待周期，直到前一个写访问完成为止。第二个写访问的延时最长可达 2 x PCLK1 + 6 x I2CCLK。

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
OA1EN	Res.	Res.	Res.	Res.	OA1 MODE	OA1[9:8]		OA1[7:1]							OA1[0]
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 **OA1EN**：设备自身地址 1 使能 (Own Address 1 enable)
0：禁止设备自身地址 1。不对接收的从地址 OA1 应答。
1：使能设备自身地址 1。对接收的从地址 OA1 应答。

位 14:11 保留，必须保持复位值。

位 10 **OA1MODE**：设备自身地址 1 10 位模式 (Own Address 1 10-bit mode)
0：设备自身地址 1 为 7 位地址。
1：设备自身地址 1 为 10 位地址。
注：　仅可在 OA1EN=0 时写入该位。

位 9:8 **OA1[9:8]**：接口地址 (Interface address)
7 位寻址模式：无关
10 位寻址模式：地址位 9:8
注：　仅可在 OA1EN=0 时写入这些位。

位 7:1 **OA1[7:1]**：接口地址 (Interface address)
7 位寻址模式：7 位地址
10 位寻址模式：10 位地址的位 7:1
注：　仅可在 OA1EN=0 时写入这些位。

位 0 **OA1[0]**：接口地址 (Interface address)
7 位寻址模式：无关
10 位寻址模式：地址位 0
注：　仅可在 OA1EN=0 时写入该位。

43.7.4 设备自身地址 2 寄存器 (I2C_OAR2)

Own address 2 register

偏移地址：0x0C

复位值：0x0000 0000

访问：无等待周期，正在对该寄存器执行写访问时出现另一个写访问的情况除外。在这种情况下，会在第二个写访问中插入等待周期，直到前一个写访问完成为止。第二个写访问的延时最长可达 2 x PCLK1 + 6 x I2CCLK。

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
OA2EN	Res.	Res.	Res.	Res.	OA2MSK[2:0]			OA2[7:1]							Res.
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 **OA2EN**：设备自身地址 2 使能 (Own Address 2 enable)

0：禁止设备自身地址 2。不对接收的从地址 OA2 应答。

1：使能设备自身地址 2。对接收的从地址 OA2 应答。

位 14:11 保留，必须保持复位值。

位 10:8 **OA2MSK[2:0]**：设备自身地址 2 屏蔽位 (Own Address 2 masks)

000：无屏蔽

001：OA2[1] 被屏蔽，为无关位。仅比较 OA2[7:2]。

010：OA2[2:1] 被屏蔽，为无关位。仅比较 OA2[7:3]。

011：OA2[3:1] 被屏蔽，为无关位。仅比较 OA2[7:4]。

100：OA2[4:1] 被屏蔽，为无关位。仅比较 OA2[7:5]。

101：OA2[5:1] 被屏蔽，为无关位。仅比较 OA2[7:6]。

110：OA2[6:1] 被屏蔽，为无关位。仅比较 OA2[7]。

111：OA2[7:1] 被屏蔽，为无关位。不进行比较，对接收到的全部 7 位地址（保留位除外）应答。

注： 仅可在 OA2EN=0 时写入这些位。

只要 OA2MSK 不等于 0，即使比较匹配，也不会对保留的 I2C 地址（0b0000xxx 和 0b1111xxx）应答。

位 7:1 **OA2[7:1]**：接口地址 (Interface address)

7 位寻址模式：7 位地址

注： 仅可在 OA2EN=0 时写入这些位。

位 0 保留，必须保持复位值。

43.7.5 时序寄存器 (I2C_TIMINGR)

Timing register

偏移地址：0x10

复位值：0x0000 0000

访问：无等待周期

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PRESC[3:0]				Res.	Res.	Res.	Res.	SCLDEL[3:0]				SDADEL[3:0]			
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
SCLH[7:0]								SCLL[7:0]							
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:28 **PRESC[3:0]**：时序预分频因子 (Timing prescaler)

该字段用于对 I2CCLK 进行预分频，以生成用于数据建立和保持计数器（请参见*第 1393 页的 I2C 时序*）以及 SCL 高电平和低电平计数器（请参见*第 1407 页的I2C 主模式初始化*）的时钟周期 t_{PRESC}。

$t_{PRESC} = (PRESC+1) \times t_{I2CCLK}$

位 27:24 保留，必须保持复位值。

位 23:20 **SCLDEL[3:0]**：数据建立时间 (Data setup time)

该字段用于在 SDA 边沿和 SCL 上升沿之间生成延时 t_{SCLDEL}。在主模式和从模式下，如果 NOSTRETCH = 0，则 SCL 线的低电平时间将在 t_{SCLDEL} 期间延长。

$t_{SCLDEL} = (SCLDEL+1) \times t_{PRESC}$

注： t_{SCLDEL} 用于生成 $t_{SU:DAT}$ 时序。

位 19:16 **SDADEL[3:0]**：数据保持时间 (Data hold time)

该字段用于在 SCL 下降沿和 SDA 边沿之间生成延时 t_{SDADEL}。在主模式和从模式下，如果 NOSTRETCH = 0，则 SCL 线的低电平时间将在 t_{SDADEL} 期间延长。

$t_{SDADEL} = SDADEL \times t_{PRESC}$

注： SDADEL 用于生成 $t_{HD:DAT}$ 时序。

位 15:8 **SCLH[7:0]**：SCL 高电平周期（主模式）(SCL high period (master mode))

在主模式下，该字段用于生成 SCL 高电平周期。

$t_{SCLH} = (SCLH+1) \times t_{PRESC}$

注： SCLH 还用于生成 $t_{SU:STO}$ 和 $t_{HD:STA}$ 时序。

位 7:0 **SCLL[7:0]**：SCL 低电平周期（主模式）(SCL low period (master mode))

在主模式下，该字段用于生成 SCL 低电平周期。

$t_{SCLL} = (SCLL+1) \times t_{PRESC}$

注： SCLL 还用于生成 t_{BUF} 和 $t_{SU:STA}$ 时序。

注： 该寄存器必须在 I2C 禁止时 (PE = 0) 进行配置。

注： STM32CubeMX 工具进行计算并将 I2C_TIMINGR 内容提供在 I2C 配置窗口中。

43.7.6 超时寄存器 (I2C_TIMEOUTR)

Timeout register

偏移地址：0x14

复位值：0x0000 0000

访问：无等待周期，正在对该寄存器执行写访问时出现另一个写访问的情况除外。在这种情况下，会在第二个写访问中插入等待周期，直到前一个写访问完成为止。第二个写访问的延时最长可达 2 x PCLK1 + 6 x I2CCLK。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TEXTEN	Res.	Res.	Res.	TIMEOUTB[11:0]											
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
TIMOUTEN	Res.	Res.	TIDLE	TIMEOUTA[11:0]											
rw			rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31 **TEXTEN**：时钟信号延展超时使能 (Extended clock timeout enable)

0：禁止时钟信号延展超时检测。

1：使能时钟信号延展超时检测 当 I2C 接口执行 SCL 延展的累积时间超过 $_{LOW:EXT}$ 时，将检测到超时错误 (TIMEOUT=1)。

位 30:28 保留，必须保持复位值。

位 27:16 **TIMEOUTB[11:0]**：总线超时 B (Bus timeout B)

该字段用于配置累积时钟延展超时：

在主模式下，将检测主器件的累积时钟低电平延展时间 ($t_{LOW:MEXT}$)

在从模式下，将检测从器件的累积时钟低电平延展时间 ($t_{LOW:SEXT}$)

$t_{LOW:EXT}$= (TIMEOUTB+1) x 2048 x t_{I2CCLK}

注： 仅可在 TEXTEN=0 时写入这些位。

位 15 **TIMOUTEN**：时钟超时使能 (Clock timeout enable)

0：禁止 SCL 超时检测。

1：使能 SCL 超时检测：当 SCL 的低电平时间超过 $t_{TIMEOUT}$ (TIDLE=0)，或 SCL 的高电平时间超过 t_{IDLE} (TIDLE=1) 时，将检测到超时错误 (TIMEOUT=1)。

位 14:13 保留，必须保持复位值。

位 12 **TIDLE**：空闲时钟超时检测 (Idle clock timeout detection)

0：TIMEOUTA 用于检测 SCL 低电平超时

1：TIMEOUTA 用于检测 SCL 和 SDA 高电平超时（总线空闲条件）

注： 仅可在 TIMOUTEN=0 时写入该位。

位 11:0 **TIMEOUTA[11:0]**：总线超时 A (Bus Timeout A)

该字段用于配置：

– SCL 低电平超时条件 $t_{TIMEOUT}$（当 TIDLE=0 时）

$t_{TIMEOUT}$= (TIMEOUTA+1) x 2048 x t_{I2CCLK}

– 总线空闲条件，即 SCL 和 SDA 高电平（当 TIDLE=1 时）

t_{IDLE}= (TIMEOUTA+1) x 4 x t_{I2CCLK}

注： 仅可在 TIMOUTEN=0 时写入这些位。

注： 如果不支持 SMBus 功能，该寄存器保留，并由硬件强制为“0x00000000”。请参见第 43.3 节：I2C 特性实现。

43.7.7 中断和状态寄存器 (I2C_ISR)

Interrupt and status register

偏移地址：0x18

复位值：0x0000 0001

访问：无等待周期

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ADDCODE[6:0]							DIR
								r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BUSY	Res.	ALERT	TIME OUT	PEC ERR	OVR	ARLO	BERR	TCR	TC	STOPF	NACKF	ADDR	RXNE	TXIS	TXE
r		r	r	r	r	r	r	r	r	r	r	r	r	rs	rs

位 31:24 保留，必须保持复位值。

位 23:17 **ADDCODE[6:0]**：地址匹配代码（从模式）(Address match code (Slave mode))
发生地址匹配事件时 (ADDR = 1)，这些位更新为接收到的地址。
在 10 位地址的情况下，ADDCODE 提供 10 位地址的头字节，后跟地址的 2 个 MSB。

位 16 **DIR**：传输方向（从模式）(Transfer direction (Slave mode))
该标志在发生地址匹配事件时 (ADDR=1) 更新。
0：写传输，从器件进入接收器模式。
1：读传输，从器件进入发送器模式。

位 15 **BUSY**：总线繁忙 (Bus busy)
该标志用于指示总线上正在进行通信。当检测到起始位时，该位由硬件置 1。当检测到停止位或 PE = 0 时，该位由硬件清零。

位 14 保留，必须保持复位值。

位 13 **ALERT**：SMBus 报警 (SMBus alert)
当 SMBHEN=1（SMBus 主机配置）、ALERTEN=1 且在 SMBA 引脚上检测到 SMBALERT 事件（下降沿）时，该标志由硬件置 1。该位由软件清零，方法是将 ALERTCF 位置 1。
注： *当 PE=0 时，该位由硬件清零。*
如果不支持 SMBus 功能，该位保留，并由硬件强制为“0”。请参见第 43.3 节：I2C 特性实现。

位 12 **TIMEOUT**：超时或 t_{LOW} 检测标志 (Timeout or t_{LOW} detection flag)
发生超时或延长时钟超时时，该标志由硬件置 1。该位由软件清零，方法是将 TIMEOUTCF 位置 1。
注： *当 PE=0 时，该位由硬件清零。*
如果不支持 SMBus 功能，该位保留，并由硬件强制为“0”。请参见第 43.3 节：I2C 特性实现。

位 11 **PECERR**：接收期间的 PEC 错误 (PEC Error in reception)

当接收到的 PEC 与 PEC 寄存器的内容不匹配时，该标志由硬件置 1。接收到错误的 PEC 后，将自动发送 NACK。该标志由软件清零，方法是将 PECCF 位置 1。

注：　当 PE=0 时，该位由硬件清零。

如果不支持 SMBus 功能，该位保留，并由硬件强制为“0”。请参见第 43.3 节：I2C 特性实现。

位 10 **OVR**：上溢/下溢（从模式）(Overrun/Underrun (slave mode))

在从模式下且 NOSTRETCH=1 时，如果发生上溢/下溢错误，该标志由硬件置 1。该标志由软件清零，方法是将 OVRCF 位置 1。

注：　当 PE=0 时，该位由硬件清零。

位 9 **ARLO**：仲裁丢失 (Arbitration lost)

发生仲裁丢失时，该标志由硬件置 1。该标志由软件清零，方法是将 ARLOCF 位置 1。

注：　当 PE=0 时，该位由硬件清零。

位 8 **BERR**：总线错误 (Bus error)

当检测到错位的起始位或停止位，而外设也参与传输时，该标志由硬件置 1。在从模式下的地址阶段，该标志不会置 1。该标志由软件清零，方法是将 *BERRCF 位置 1*。

注：　当 PE=0 时，该位由硬件清零。

位 7 **TCR**：传输完成等待重载 (Transfer Complete Reload)

当 RELOAD=1 且 NBYTES 数据传输完成时，该标志由硬件置 1。当 NBYTES 写入一个非零值时，该标志由软件清零。

注：　当 PE=0 时，该位由硬件清零。

该标志单独用于主模式，SBC 位置 1 时单独用于从模式。

位 6 **TC**：传输完成（主模式）(Transfer Complete (master mode))

当 RELOAD=0、AUTOEND=0 且 NBYTES 数据传输完成时，该标志由硬件置 1。当 START 位或 STOP 位置 1 时，该标志由软件清零。

注：　当 PE=0 时，该位由硬件清零。

位 5 **STOPF**：停止位检测标志 (Stop detection flag)

当在总线上检测到停止位，且外设也参与本次传输时，该标志由硬件置 1：

– 外设作为主器件，该位置位的前提是外设已经发出停止位。

– 外设作为从器件，该位置位的前提条件是此次传输的寻址对象就是该外设。

该标志由软件清零，方法是将 STOPCF 位置 1。

注：　当 PE=0 时，该位由硬件清零。

位 4 **NACKF**：接收到否定应答标志 (Not Acknowledge received flag)

传输完字节后接收到 NACK 时，该标志由硬件置 1。该标志由软件清零，方法是将 NACKCF 位置 1。

注：　当 PE=0 时，该位由硬件清零。

位 3 **ADDR**：地址匹配（从模式）(Address matched (slave mode))

接收到的地址与使能的从设备地址之一匹配时，该位由硬件置 1。该位由软件清零，方法是将 *ADDRCF 位置 1*。

注：　当 PE=0 时，该位由硬件清零。

位 2 **RXNE**：接收数据寄存器不为空（接收器）(Receive data register not empty (receivers))
当接收到的数据已复制到 I2C_RXDR 寄存器且准备就绪可供读取时，该位由硬件置 1。读取 I2C_RXDR 时，将清零该位。
注：　当 PE=0 时，该位由硬件清零。

位 1 **TXIS**：发送中断状态（发送器）(Transmit interrupt status (transmitters))
当 I2C_TXDR 寄存器为空时，该位由硬件置 1，待发送的数据必须写入 I2C_TXDR 寄存器。下一个待发送的数据写入 I2C_TXDR 寄存器时，该位被清零。
该位只能在 NOSTRETCH=1 时由软件写入“1”，以生成 TXIS 事件（TXIE=1 时为中断，TXDMAEN=1 时为 DMA 请求）。
注：　当 PE=0 时，该位由硬件清零。

位 0 **TXE**：发送数据寄存器为空（发送器）(Transmit data register empty (transmitters))
当 I2C_TXDR 寄存器为空时，该位由硬件置 1。下一个待发送的数据写入 I2C_TXDR 寄存器时，该位被清零。
该位可由软件写入“1”，以刷新发送数据寄存器 I2C_TXDR。
注：　当 PE=0 时，该位由硬件置 1。

43.7.8 中断清零寄存器 (I2C_ICR)

Interrupt clear register

偏移地址：0x1C

复位值：0x0000 0000

访问：无等待周期

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.	ALERT CF	TIM OUTCF	PECCF	OVRCF	ARLO CF	BERR CF	Res.	Res.	STOP CF	NACK CF	ADDR CF	Res.	Res.	Res.
		w	w	w	w	w	w			w	w	w			

位 31:14 保留，必须保持复位值。

位 13 **ALERTCF**：报警标志清零 (Alert flag clear)
将 1 写入此位时，I2C_ISR 寄存器中的 ALERT 标志将清零。
注：　如果不支持 SMBus 功能，该位保留，并由硬件强制为“0”。请参见第 43.3 节：I2C 特性实现。

位 12 **TIMOUTCF**：超时检测标志清零 (Timeout detection flag clear)
将 1 写入此位时，I2C_ISR 寄存器中的 TIMEOUT 标志将清零。
注：　如果不支持 SMBus 功能，该位保留，并由硬件强制为“0”。请参见第 43.3 节：I2C 特性实现。

位 11 **PECCF**：PEC 错误标志清零 (PEC Error flag clear)
将 1 写入此位时，I2C_ISR 寄存器中的 PECERR 标志将清零。
注：　如果不支持 SMBus 功能，该位保留，并由硬件强制为“0”。请参见第 43.3 节：I2C 特性实现。

位 10 **OVRCF**：上溢/下溢标志清零 (Overrun/Underrun flag clear)
将 1 写入此位时，I2C_ISR 寄存器中的 OVR 标志将清零。

位 9 **ARLOCF**：仲裁丢失标志清零 (Arbitration Lost flag clear)
将 1 写入此位时，I2C_ISR 寄存器中的 ARLO 标志将清零。

位 8 **BERRCF**：总线错误标志清零 (Bus error flag clear)
将 1 写入此位时，I2C_ISR 寄存器中的 BERRF 标志将清零。

位 7:6 保留，必须保持复位值。

位 5 **STOPCF**：停止位检测标志清零 (Stop detection flag clear)
将 1 写入此位时，I2C_ISR 寄存器中的 STOPF 标志将清零。

位 4 **NACKCF**：否定应答标志清零 (Not Acknowledge flag clear)
向此位写入 1 时，I2C_ISR 寄存器中的 NACKF 标志将清零。

位 3 **ADDRCF**：地址匹配标志清零 (Address Matched flag clear)
将 1 写入此位时，I2C_ISR 寄存器中的 ADDR 标志将清零。将 1 写入此位时，I2C_CR2 寄存器中的 START 位也将清零。

位 2:0 保留，必须保持复位值。

43.7.9 PEC 寄存器 (I2C_PECR)

PEC register

偏移地址：0x20

复位值：0x0000 0000

访问：无等待周期

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.	Res.	Res.	Res.	Res.	Res.	Res.	PEC[7:0]							
								r	r	r	r	r	r	r	r

位 31:8 保留，必须保持复位值。

位 7:0 **PEC[7:0]**：数据包错误校验寄存器 (Packet error checking register)
当 PECEN=1 时，此字段包含内部 PEC。
当 PE=0 时，PEC 由硬件清零。

注： *如果不支持 SMBus 功能，该寄存器保留，并由硬件强制为“0x00000000”。请参见第 43.3 节: I2C 特性实现。*

43.7.10 接收数据寄存器 (I2C_RXDR)

Receive data register

偏移地址：0x24

复位值：0x0000 0000

访问：无等待周期

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.	Res.	Res.	Res.	Res.	Res.	Res.	RXDATA[7:0]							
								r	r	r	r	r	r	r	r

位 31:8 保留，必须保持复位值。

位 7:0 **RXDATA[7:0]**：8 位接收数据 (8-bit receive data)
从 I^2C 总线接收的数据字节。

43.7.11 发送数据寄存器 (I2C_TXDR)

Transmit data register

偏移地址：0x28

复位值：0x0000 0000

访问：无等待周期

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.	Res.	Res.	Res.	Res.	Res.	Res.	TXDATA[7:0]							
								rw	rw	rw	rw	rw	rw	rw	rw

位 31:8 保留，必须保持复位值。

位 7:0 **TXDATA[7:0]**：8 位发送数据 (8-bit transmit data)
待发送到 I^2C 总线的数据字节。
注：　仅可在 TXE=1 时写入这些位。

43.7.12 I2C 寄存器映射

下表提供了 I2C 寄存器映射和复位值。

表 261. I2C 寄存器映射和复位值

偏移	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
0x0	**I2C_CR1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PECEN	ALERTEN	SMBDEN	SMBHEN	GCEN	WUPEN	NOSTRETCH	SBC	RXDMAEN	TXDMAEN	Res.	ANFOFF	DNF[3:0]				ERRIE	TCIE	STOPIE	NACKIE	ADDRIE	RXIE	TXIE	PE
	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
0x4	**I2C_CR2**	Res.	Res.	Res.	Res.	Res.	PECBYTE	AUTOEND	RELOAD	NBYTES[7:0]								NACK	STOP	START	HEAD10R	ADD10	RD_WRN	SADD[9:0]									
	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
0x8	**I2C_OAR1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OA1EN	Res.	Res.	Res.	Res.	OA1MODE	OA1[9:0]									
	Reset value																	0					0	0	0	0	0	0	0	0	0	0	0
0xC	**I2C_OAR2**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OA2EN	Res.	Res.	Res.	Res.	OA2MSK [2:0]	OA2[7:1]									Res.
	Reset value																	0					0	0	0	0	0	0	0	0	0	0	
0x10	**I2C_TIMINGR**	PRESC[3:0]				Res.	Res.	Res.	Res.	SCLDEL[3:0]				SDADEL[3:0]				SCLH[7:0]								SCLL[7:0]							
	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
0x14	**I2C_TIMEOUTR**	TEXTEN	Res.	Res.	Res.	TIMEOUTB[11:0]												TIMOUTEN	Res.		TIDLE	TIMEOUTA[11:0]											
	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
0x18	**I2C_ISR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ADDCODE[6:0]							DIR	BUSY	Res.	ALERT	TIMEOUT	PECERR	OVR	ARLO	BERR	TCR	TC	STOPF	NACKF	ADDR	RXNE	TXIS	TXE
	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
0x1C	**I2C_ICR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ALERTCF	TIMOUTCF	PECCF	OVRCF	ARLOCF	BERRCF	Res.	Res.	STOPCF	NACKCF	ADDRCF	Res.	Res.	Res.
	Reset value																			0	0	0	0	0	0			0	0	0			
0x20	**I2C_PECR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PEC[7:0]							
	Reset value																									0	0	0	0	0	0	0	0
0x24	**I2C_RXDR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RXDATA[7:0]							
	Reset value																									0	0	0	0	0	0	0	0

表 261. I2C 寄存器映射和复位值（续）

偏移	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
0x28	**I2C_TXDR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TXDATA[7:0]							
	Reset value																									0	0	0	0	0	0	0	0

有关寄存器边界地址的信息，请参见*第 78 页的第 2.2.2 节*。

44 通用同步异步收发器 (USART)

本节介绍通用同步/异步收发器 (USART)。

44.1 USART 简介

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实现了多种波特率。

USART 不仅支持同步单向通信和半双工单线通信，以及 LIN（局域互连网络）、智能卡协议、IrDA（红外线数据协会）SIR ENDEC 规范和调制解调器操作 (CTS/RTS)，还支持多处理器通信。

通过配置多个缓冲区使用 DMA（直接存储器访问）可实现高速数据通信。

44.2 USART 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，从而在速度容差与时钟容差之间取得最佳平衡
- 波特率发生器系统
- 两个用于收发数据的内部 FIFO
  每个 FIFO 均可由软件使能/禁止，并且均带有一个状态标志
- 通用可编程收发波特率
- 双时钟域，带有独立于 PCLK 的外设专用内核时钟
- 自动波特率检测
- 数据字长度可编程（7 位、8 位或 9 位）
- 可编程的数据顺序，最先移位 MSB 或 LSB
- 停止位可配置（支持 1 个或 2 个停止位）
- 用于同步通信的同步主/从模式和时钟输出/输入
- SPI 从发送下溢错误标志
- 单线半双工通信
- 使用 DMA 实现连续通信
- 使用中央 DMA 在预留的 SRAM 缓冲区中收/发字节
- 发射器和接收器有单独的使能位
- 发送和接收的单独信号极性控制
- Tx/Rx 引脚配置可交换
- 调制解调器和 RS-485 收发器的硬件流控制
- 通信控制/错误检测标志
- 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
- 具有标志的中断源
- 多处理器通信：从静默模式唤醒（通过空闲线检测或地址标记检测）

44.3 USART 扩展特性

- LIN 主模式同步中断发送功能和 LIN 从模式中断检测功能
  - 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
- 正常模式下支持 3/16 位持续时间的 IrDA SIR 编解码器
- 智能卡模式
  - 对于 ISO/IEC 7816-3 标准中定义的智能卡，支持 T=0 和 T=1 异步协议
  - 智能卡工作模式下，支持 0.5 和 1.5 个停止位
- 支持 ModBus 通信
  - 超时功能
  - CR/LF 字符识别

44.4 USART 实现

表 262 介绍了 STM32L4Rxxx 和 STM32L4Sxxx 器件上的 USART 实现。

表 262. USART/LPUART 功能

USART 模式/特性(1)	USART1/2/3	UART4/5	LPUART
调制解调器的硬件流控制	X	X	X
使用 DMA 进行连续通信	X	X	X
多处理器通信	X	X	X
同步模式（主/从）	X	-	-
智能卡模式	X	-	-
单线半双工通信	X	X	X
IrDA SIR ENDEC 模块	X	X	-
LIN 模式	X	X	-
双时钟域和从低功耗模式唤醒	X	X	X
接收器超时中断	X	X	-
Modbus 通信	X	X	-
自动波特率检测	X	X	-
驱动器使能	X	X	X
USART 数据长度	7 位、8 位和 9 位		
Tx/Rx FIFO	X	X	X
Tx/Rx FIFO 大小	8		

1. X = 支持。

44.5 USART 功能说明

44.5.1 USART 框图

图 451. USART 框图

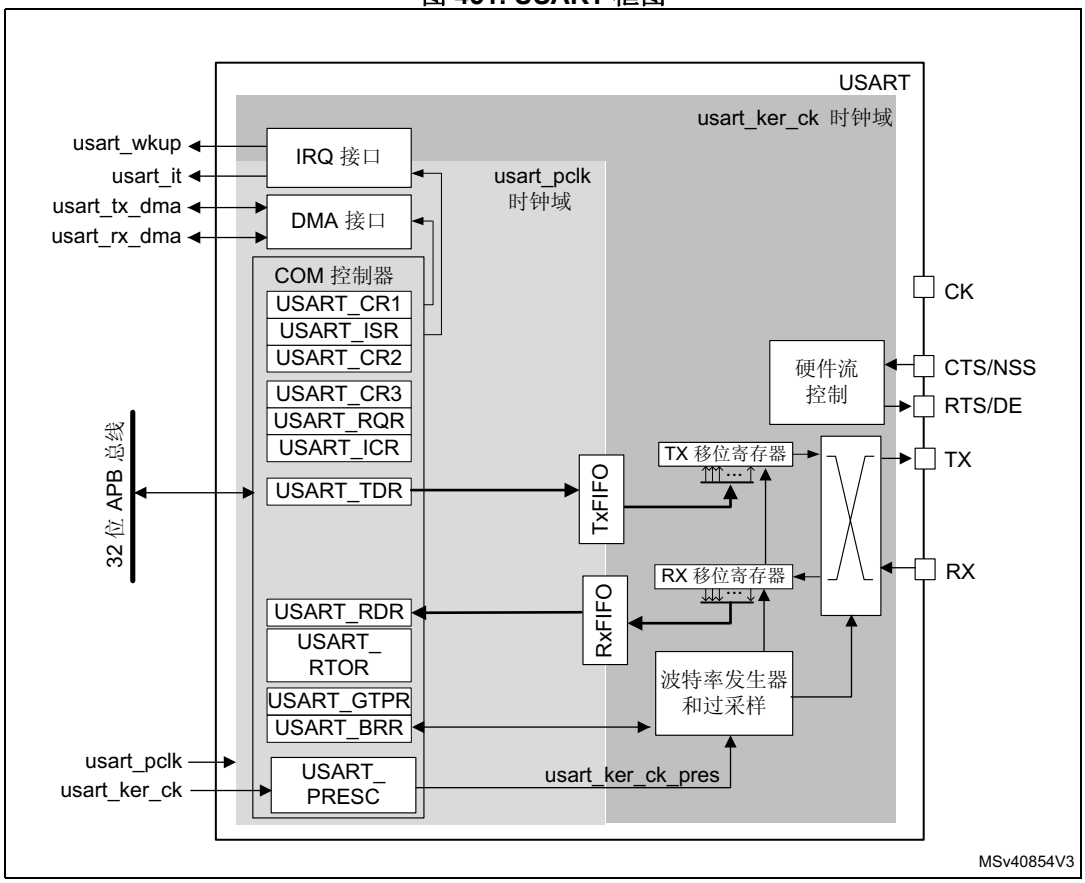


图 451 中的简化框图显示的是两个完全独立的时钟域：

- **usart_pclk** 时钟域。

  usart_pclk 时钟信号馈送外设总线接口。需要访问 USART 寄存器时，该信号必须有效。

- **usart_ker_ck** 内核时钟域。

  usart_ker_ck 是 USART 时钟源。它独立于 **usart_pclk**，由 RCC 提供。因此，即使 **usart_ker_ck** 时钟停止，也可以连续对 USART 寄存器进行读/写操作。

  禁用双时钟域功能时，**usart_ker_ck** 时钟与 **usart_pclk** 时钟相同。

usart_pclk 和 **usart_ker_ck** 之间无任何限制：**usart_ker_ck** 既可快于也可慢于 **usart_pclk**。唯一的限制是软件以足够快的速度管理通信的能力。

USART 工作在 SPI 从器件模式下时，会使用源自外部 SCLK 信号（由外部主 SPI 器件提供）的串行接口时钟来处理数据流。**usart_ker_ck** 时钟的速度必须至少 3 倍于 CK 输入上的时钟。

44.5.2 USART 信号

USART 双向通信

USART 双向通信需要至少两个引脚：接收数据输入引脚 (RX) 和发送数据输出引脚 (TX)：

- **RX**（接收数据输入引脚）

  RX 为串行数据输入引脚。采用过采样技术进行数据恢复，即区分有效输入数据和噪声。

- **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需要发送的数据，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。

RS232 硬件流控制模式

在 RS232 硬件流控制模式下需要以下引脚：

- **CTS**（清除以发送）

  如果驱动为高电平，则该信号用于在当前传输结束时阻止数据发送。

- **RTS**（请求以发送）

  如果为低电平，则该信号用于指示 USART 已准备好接收数据。

RS485 硬件控制模式

在 RS485 硬件控制模式下需要以下引脚：

- **DE**（驱动器使能）

  该信号用于激活外部收发器的发送模式。

注： *DE 和 RTS 共用同一个引脚。*

同步主/从模式和智能卡模式

在同步主/从模式和智能卡模式下需要以下引脚：

- CK

  该引脚在同步主模式和智能卡模式下用作时钟输出。

  它在同步从模式下用作时钟输入。

  在同步主模式下，该引脚用于输出发送器数据时钟，以便按照 SPI 主器件模式进行同步发送（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 引脚上可同步接收并行数据。该机制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，CK 输出向智能卡提供时钟。

- NSS

  该引脚在同步从模式下用作从器件选择输入。

注： *NSS 和 CTS 共用同一个引脚。*

44.5.3 USART 字符说明

可通过对 USART_CR1 寄存器中的 M 位（M0：位 12，M1：位 28）进行编程来将字长设置为 7 位、8 位或 9 位（请参见*图 452*）。

- 7 位字符长度：M[1:0] = “10”
- 8 位字符长度：M[1:0] = “00”
- 9 位字符长度：M[1:0] = “01”

注： *7 位数据长度模式下，不支持智能卡模式、LIN 主模式和自动波特率（0x7F 帧和 0x55 帧检测）。*

在默认情况下，信号（TX 或 RX）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。

通过极性配置控制，可以单独针对每个信号对这些值取反。

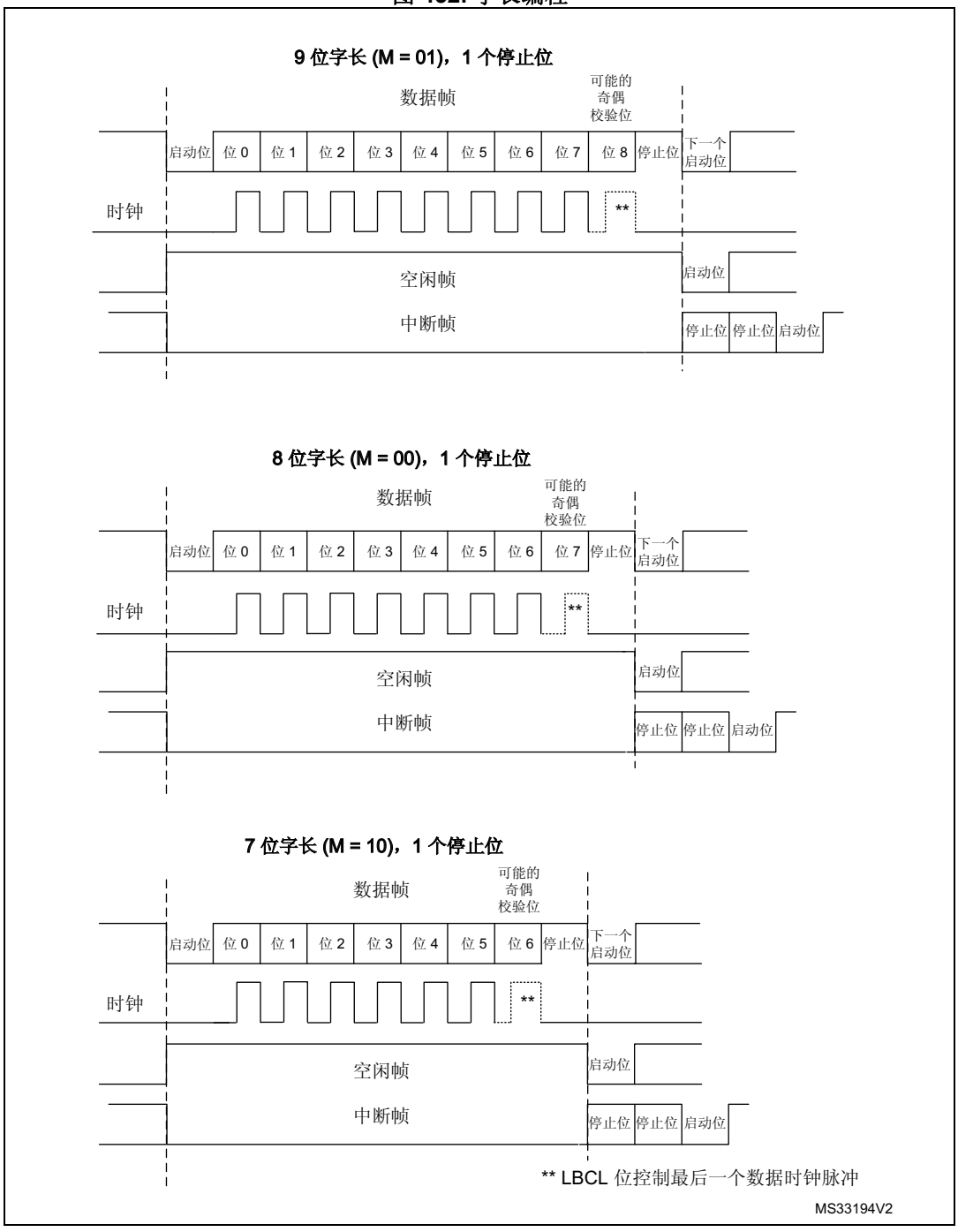
空闲字符可理解为整个帧周期内电平均为“1”（停止位的电平也是“1”）。

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 2 个停止位。

发送和接收操作由通用波特率发生器驱动。当发送器和接收器的使能位置 1 时，将分别生成发送时钟和接收时钟。

下面给出了各个块的详细说明。

图 452. 字长编程


9 位字长 (M = 01)，1 个停止位
数据帧
可能的
奇偶
校验位
启动位
位 0
位 1
位 2
位 3
位 4
位 5
位 6
位 7
位 8
停止位
下一个
启动位
时钟
**
空闲帧
启动位
中断帧
停止位
停止位
启动位
8 位字长 (M = 00)，1 个停止位
数据帧
可能的
奇偶
校验位
启动位
位 0
位 1
位 2
位 3
位 4
位 5
位 6
位 7
停止位
下一个
启动位
时钟
**
空闲帧
启动位
中断帧
停止位
停止位
启动位
7 位字长 (M = 10)，1 个停止位
数据帧
可能的
奇偶
校验位
启动位
位 0
位 1
位 2
位 3
位 4
位 5
位 6
停止位
下一个
启动位
时钟
**
空闲帧
启动位
中断帧
停止位
停止位
启动位
** LBCL 位控制最后一个数据时钟脉冲
MS33194V2

44.5.4 USART FIFO 和阈值

USART 可工作在 FIFO 模式下。

USART 具有一个发送 FIFO (TXFIFO) 和一个接收 FIFO (RXFIFO)。可通过将 USART_CR1 寄存器中的 FIFOEN（位 29）置 1 使能 FIFO 模式。仅 UART、SPI 和智能卡模式下支持该模式。

最大数据字长度为 9 位，因此 TXFIFO 为 9 位宽。不过，RXFIFO 的默认宽度为 12 位。这是因为接收器不仅在 FIFO 中存储数据，而且还存储与每个字符相关的错误标志（奇偶校验错误、噪声错误和帧错误标志）。

注： *接收的数据与相应的标志一起存储在 RXFIFO 中，但读取 RDR 时仅读取数据。*

状态标志位于 USART_ISR 寄存器中。

可以配置触发 Tx 和 RX 中断的 TXFIFO 和 RXFIFO 阈值。这些阈值通过 USART_CR3 控制寄存器中的 RXFTCFG 和 TXFTCFG 位域进行编程。

在这种情况下：

- 当 RXFIFO 中接收到的数据量达到 RXFTCFG 位域中编程的阈值时，会生成 Rx 中断。此时，USART_ISR 寄存器中的 RXFT 标志置 1。这表示已接收到 RXFTCFG 数据：USART_RDR 中有 1 个数据，RXFIFO 中有 (RXFTCFG - 1) 个数据。例如，如果将 RXFTCFG 编程为“101”，则在接收到对应于 FIFO 大小的数据量（RXFIFO 中有（FIFO 大小 -1）个数据，USART_RDR 中有 1 个数据）时，RXFT 标志将置 1。因此，下一个接收到的数据不会将上溢标志置 1。
- 当 TXFIFO 中的空位置数达到在 TXFTCFG 位域中编程的阈值时，会生成 Tx 中断。

44.5.5 USART 发送器

发送器可发送 7 位、8 位或 9 位的数据字，具体取决于 M 位的状态。要激活发送器功能，必须将发送使能位 (TE) 置 1。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（默认配置）。在该模式下，USART_TDR 寄存器的缓冲区 (TDR) 位于内部总线和发送移位寄存器之间。

使能 FIFO 模式时，写入到发送数据寄存器 (USART_TDR) 中的数据会在 TXFIFO 中排队。

每个字符前面都有一个起始位，其对应于一个位周期的逻辑低电平。字符由可配置数量的停止位终止。

停止位的数量可配置为 0.5、1、1.5 或 2。

注： *向 USART_TDR 中写入要发送的数据前，TE 位必须先置 1。*

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，进而损坏 TX 引脚上的数据。当前发送的数据随即丢失。

使能 TE 位时，将发送空闲帧。

可配置的停止位

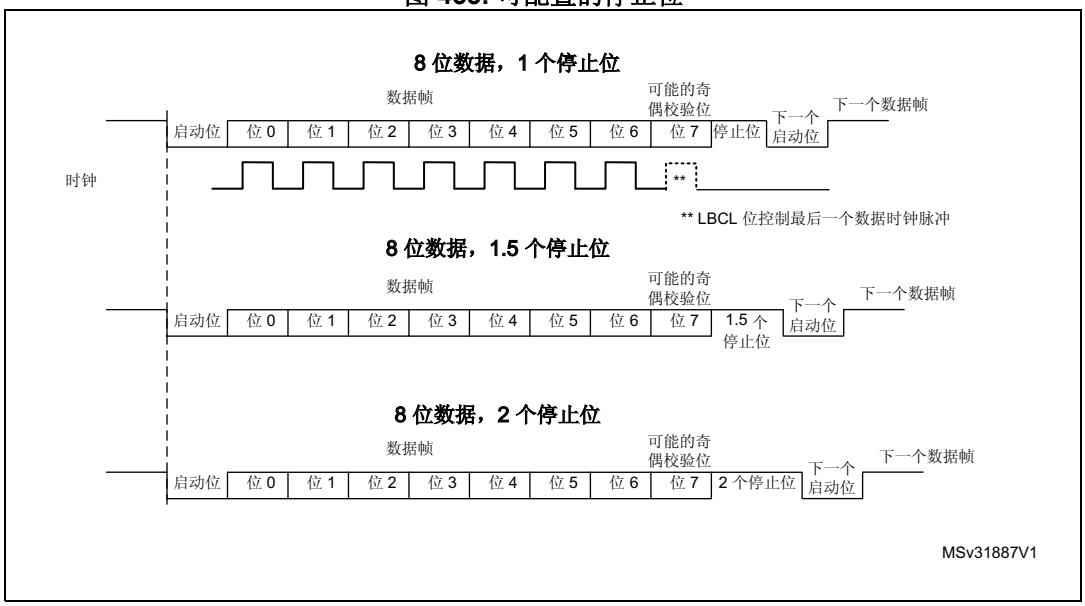
可以在 USART_CR2 的位 13 和位 12 中编程将随各个字符发送的停止位的数量。

- ***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。
- ***1.5 个停止位***：用于智能卡模式。

空闲帧发送将包括停止位。

中断发送是 10 个低电平位（M[1:0] = “00”时）、11 个低电平位（M[1:0] = “01”时）或 9 个低电平位（M[1:0] = “10”时），然后是 2 个停止位（请参见*图 453*）。无法传送长中断（中断长度大于 9/10/11 个低电平位）。

图 453. 可配置的停止位



字符发送步骤

要发送字符，需遵循以下步骤：

1. 对 USART_CR1 中的 M 位进行编程以定义字长。
2. 使用 USART_BRR 寄存器选择所需波特率。
3. 对 USART_CR2 中的停止位数量进行编程。
4. 通过向 USART_CR1 寄存器中的 UE 位写入 1 使能 USART。
5. 如果必须进行多缓冲区通信，请选择 USART_CR3 中的 DMA 使能 (DMAT)。按照*第 44.5.10 节：USART 多处理器通信*中的说明配置 DMA 寄存器。
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_TDR 寄存器中写入要发送的数据。为每个要在单缓冲区模式下发送的数据重复这一步骤。
   - 禁止 FIFO 模式时，向 USART_TDR 写入数据会将 TXE 标志清零。
   - 使能 FIFO 模式时，向 USART_TDR 写入数据会为 TXFIFO 增添一个数据。当 TXFNF 标志置 1 时，会对 USART_TDR 执行写操作。该标志会保持置 1，直到 TXFIFO 已满。
8. 将最后一个数据写入 USART_TDR 寄存器后，等待 TC = 1。
   - 禁止 FIFO 模式时，这表示最后一个帧的发送已完成。
   - 使能 FIFO 模式时，这表示 TXFIFO 和移位寄存器均为空。

   当 USART 被禁止或进入暂停模式时，需要执行此检查来避免损坏最后一次发送。

单字节通信

- 禁止 FIFO 模式时

  对发送数据寄存器进行写操作始终会清零 TXE 位。TXE 标志由硬件置 1。它表示：

  – 数据已从 USART_TDR 寄存器移到移位寄存器中且数据发送已开始；
  – USART_TDR 寄存器为空；
  – USART_TDR 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_TDR 寄存器中的写指令会将数据存储在 TDR 缓冲区中。该数据随后会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_TDR 寄存器的写指令会将数据置于移位寄存器中，数据发送开始时，TXE 位置 1。

- 使能 FIFO 模式时，TXFNF（TXFIFO 未满）标志由硬件置 1，以指示：

  – TXFIFO 未满；
  – USART_TDR 寄存器为空；
  – USART_TDR 寄存器中可写入下一个数据，而不会覆盖前一个数据。发送时，对 USART_TDR 寄存器的写操作会将数据存储在 TXFIFO 中。该数据将在当前发送结束时从 TXFIFO 复制到移位寄存器中。

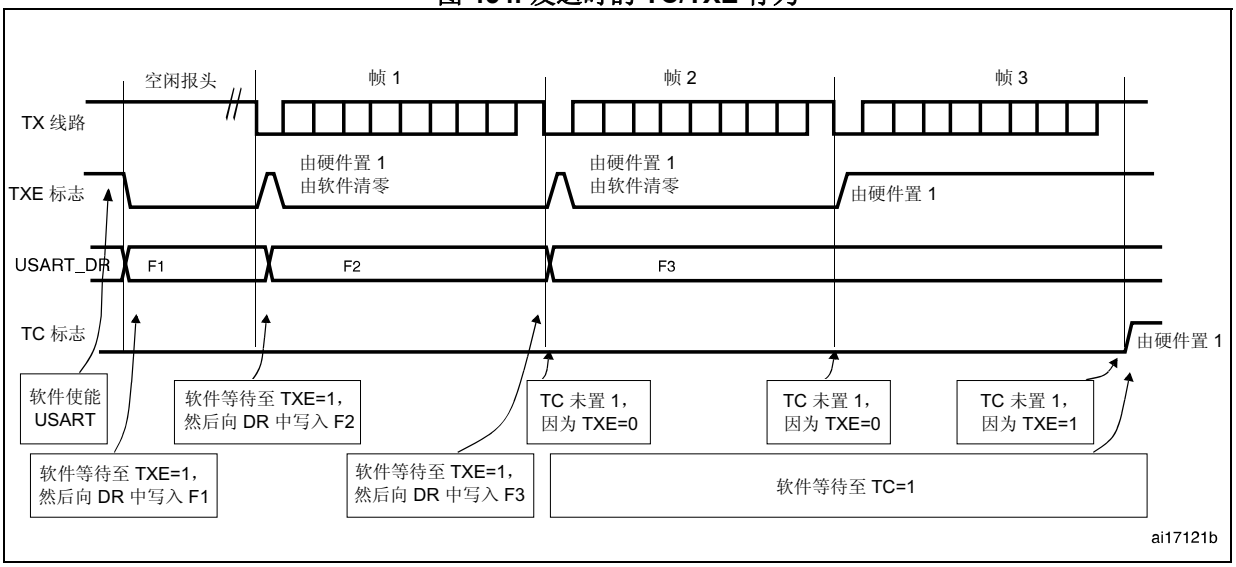
  TXFIFO 未满时，即使在对 USART_TDR 寄存器执行完写操作后，TXFNF 标志也保持为“1”。该标志在 TXFIFO 已满时清零。TXFNFIE 位置 1 时该标志位会生成中断。

  或者，当达到 TXFIFO 阈值时，会生成中断并将数据写入 FIFO。在这种情况下，CPU 可写入由编程的触发阈值定义的数据块。

  如果帧已发送（停止位后）且 TXE 标志（FIFO 模式下的 TXFE）置 1，TC 标志将变为高电平。如果 USART_CR1 寄存器中的 TCIE 位置 1，将生成中断。

向 USART_TDR 寄存器中写入最后一个数据后，必须等待至 TC 置 1，之后才可禁止 USART 或使微控制器进入低功耗模式（请参见*图 454：发送时的 TC/TXE 行为*）。

图 454. 发送时的 TC/TXE 行为



注： *使能 FIFO 管理时，TXFNF 标志将用于数据发送。*

中断字符

将 SBKRQ 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 452*）。

如果将“1”写入 SBKRQ 位，则当前字符发送完成后，将在 TX 线路上发送一个中断字符。通过写操作将 SBKF 位置 1 并在中断字符发送完成时（发送中断字符后的停止位期间），该位由硬件复位。USART 在中断帧末尾的两位持续时间内插入一个逻辑“1”信号 (STOP)，以确保识别下个帧的起始位。

如果 SBKRQ 位置 1，则在当前发送结束时，会发送一个中断字符。

使能 FIFO 模式时，即使 TXFIFO 已满，发送中断字符的优先级也仍高于发送数据的优先级。

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。

44.5.6 USART 接收器

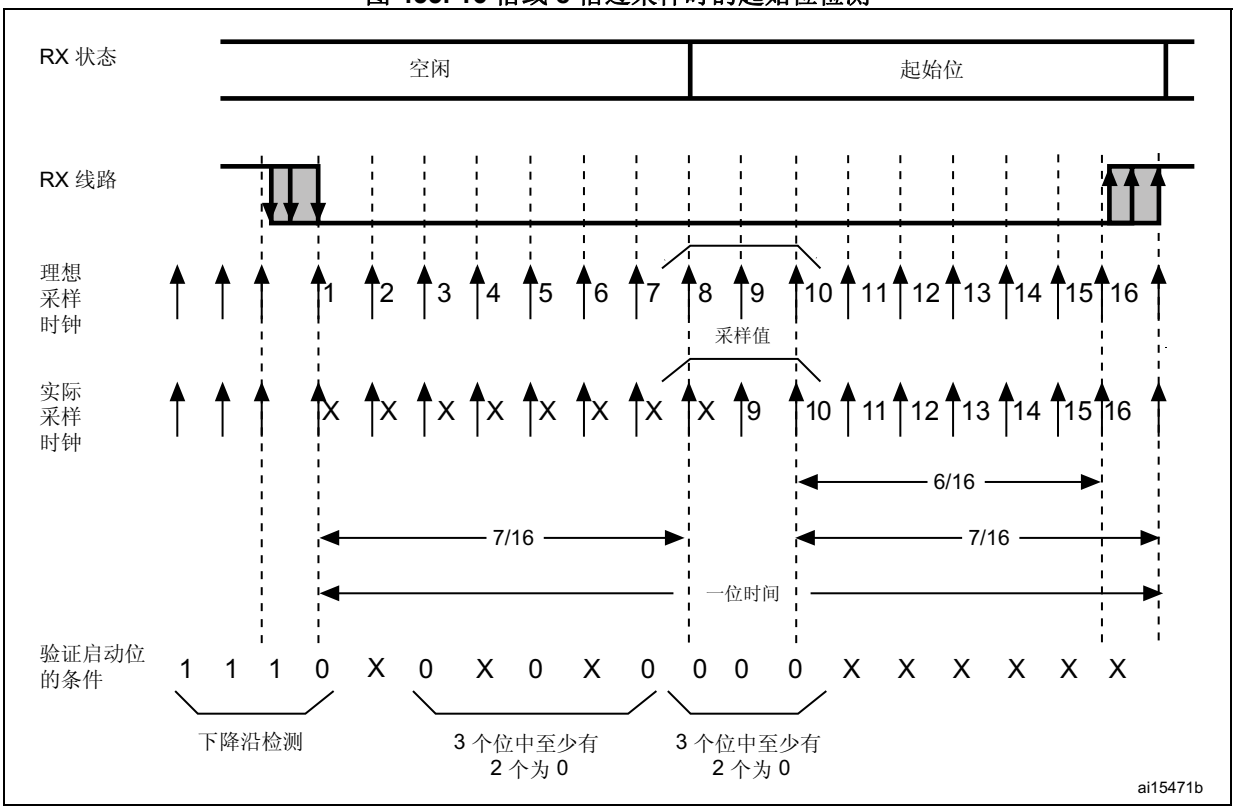
USART 可接收 7 位、8 位或 9 位的数据字，具体取决于 USART_CR1 寄存器中的 M 位。

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。此序列为：1 1 1 0 X 0 X 0X 0X 0 X 0X 0。

图 455. 16 倍或 8 倍过采样时的起始位检测



注： *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仍检测到这 3 位均为“0”），可确认起始位（RXNE 标志置 1 且 RXNEIE = 1 时生成中断；如果使能 FIFO 模式，则 RXFNE 标志置 1 且 RXFNEIE = 1 时生成中断）。

满足以下条件时，可验证起始位但 NE 噪声标志置 1：

a) 对于两次采样，3 个采样位中有 2 位为“0”（针对第 3 位、第 5 位和第 7 位进行采样；针对第 8 位、第 9 位和第 10 位采样）

或

b) 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”。

如果上述条件均不满足，则启动检测中止，接收器返回空闲状态（无标志置 1）。

字符接收

USART 接收期间，首先通过 RX 引脚移出数据的最低有效位（默认配置）。

字符接收步骤

要接收字符，需遵循以下步骤：

1. 对 USART_CR1 中的 M 位进行编程以定义字长。
2. 使用波特率寄存器 USART_BRR 选择所需波特率。
3. 对 USART_CR2 中的停止位数量进行编程。
4. 通过向 USART_CR1 寄存器中的 UE 位写入“1”使能 USART。
5. 如果将进行多缓冲区通信，请选择 USART_CR3 中的 DMA 使能 (DMAR)。按照*第 44.5.10 节：USART 多处理器通信*中的说明配置 DMA 寄存器。
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：

- 如果已禁止 FIFO 模式，则 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（及其相应的错误标志）。
- 如果已使能 FIFO 模式，则 RXFNE 位置 1，这表示 RXFIFO 非空。读取 USART_RDR 会返回输入到 RXFIFO 中的最早数据。接收到数据时，数据以及相应的错误位将一起被存储在 RXFIFO 中。
- 如果 RXNEIE（使能 FIFO 模式时为 RXFNEIE）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、奇偶校验错误或上溢错误，错误标志会置 1。
- 在多缓冲区通信模式下：
  - 如果禁止 FIFO 模式，则 RXNE 标志会在每次接收到字节后置 1。该标志在 DMA 读取接收数据寄存器时被清零。
  - 如果使能 FIFO 模式，则 RXFNE 标志在 RXFIFO 非空时置 1。每次收到 DMA 请求后，都会从 RXFIFO 检索数据。当 RXFIFO 非空时（即，当存在要从 RXFIFO 中读取的数据时），会触发 DMA 请求。
- 在单缓冲区模式下：
  - 如果禁止 FIFO 模式，则通过软件对 USART_RDR 寄存器进行读操作来将 RXNE 标志清零。也可以通过将 USART_RQR 寄存器中的 RXFRQ 位编程为“1”来清零 RXNE 标志。RXNE 标志必须在结束接收下一个字符前清零，以避免发生上溢错误。
  - 如果使能 FIFO 模式，则 RXFNE 在 RXFIFO 非空时置 1。每次对 USART_RDR 执行完读操作后，都会从 RXFIFO 中检索数据。RXFIFO 为空时，RXFNE 标志将清零。也可以通过将 USART_RQR 中的 RXFRQ 位编程为“1”来清零 RXFNE 标志。RXFIFO 已满时，必须在结束接收下一个字符前读取 RXFIFO 中的第一个条目，以避免发生上溢错误。当 RXFNEIE 位置 1 时，RXFNE 标志会生成中断。或者，当达到 RXFIFO 阈值时，会生成中断并从 RXFIFO 中读取数据。在这种情况下，CPU 可读取由编程的阈值定义的数据块。

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，除了在 IDLEIE 位置 1 时会生成中断外，处理步骤与接收到数据字符的情况相同。

上溢错误

- 禁止 FIFO 模式

  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。可通过读取 USART_RDR 寄存器获得之前的数据。
  – 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 或 EIE 位置 1，则会生成中断。

- 使能 FIFO 模式

  移位寄存器准备好传送并且接收 FIFO 已满时，会发生上溢错误。

  在 RXFIFO 中出现一个空闲位置之前，数据无法从移位寄存器传送到 USART_RDR 寄存器。当 RXFIFO 非空时，RXFNE 标志置 1。

  如果 RXFIFO 已满且移位寄存器已准备好传送，则会发生上溢错误。发生上溢错误时：

  – ORE 位置 1。
  – RXFIFO 中的第一个条目不会丢失。通过读取 USART_RDR 寄存器可获得此条目。
  – 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 RXFNEIE 或 EIE 位置 1，则会生成中断。

通过将 USART_ICR 寄存器中的 ORECF 位置 1 来复位 ORE 位。

注： *ORE 位置 1 时表示至少 1 个数据丢失。*

禁止 FIFO 模式时，有以下两种可能

- *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，则最后一个有效数据已被读取，因此 RDR 寄存器中没有要读取的数据会发生这种情况，已读取 RDR 寄存器中的最后一个有效数据的同时接收到新（并且丢失）的数据。*

选择时钟源和合适的过采样方法

通过时钟控制系统选择时钟源（请参见）。在使能 USART 之前，必须通过 UE 位选择时钟源。

必须遵循以下两个条件选择时钟源：

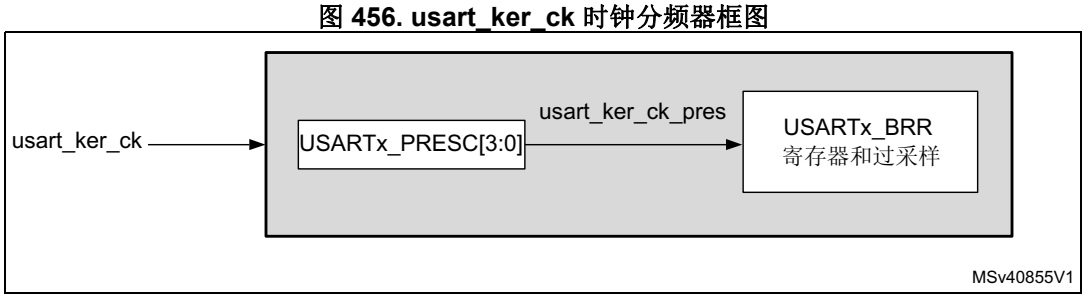
- 可在低功耗模式下使用 USART。
- 通信速度。

时钟源频率为 usart_ker_ck。

如果支持双时钟域和从低功耗模式唤醒功能，则 usart_ker_ck 时钟源可在 RCC 中进行配置。否则，usart_ker_ck 时钟与 usart_pclk 相同。

usart_ker_ck 时钟可根据可编程系数（在 USART_PRESC 寄存器中定义）进行分频。

图 456. usart_ker_ck 时钟分频器框图



某些 usart_ker_ck 时钟源允许 USART 在 MCU 处于低功耗模式时接收数据。需要时，USART 可基于所接收的数据和选择的唤醒模式来唤醒 MCU，以通过用软件读取 USART_RDR 寄存器的方式或通过 DMA 的方式传输已接收数据。

对于其他时钟源，系统必须激活才能进行 USART 通信。

时钟源还决定通信速度范围（尤其是最大通信速度）。

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。这可在最大通信速度与抗噪声/时钟误差性能之间实现最佳平衡。

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 457* 和*图 458*）。

根据应用：

- 选择 8 倍过采样（OVER8 = 1）以获得更高的速度（高达 usart_ker_ck_pres/8）。这种情况下接收器对时钟偏差的最大容差将会降低（请参见*第 1473 页的第 44.5.8 节：USART 接收器对时钟偏差的容差*）
- 选择 16 倍过采样 (OVER8=0) 以增加接收器对时钟偏差的容差。在这种情况下，最大速度被限制为最大 usart_ker_ck_pres/16（其中，usart_ker_ck_pres 为 USART 输入时钟通过预分频器进行分频得到的值）。

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。有两种选择可供使用：

- 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，NE 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 263*），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（请参见*第 1473 页的第 44.5.8 节：USART 接收器对时钟偏差的容差*）。这种情况下 NE 位始终不会置 1。

帧中检测到噪声时：

- 在 RXNE 位（使能 FIFO 模式时为 RXFNE 位）的上升沿时 NE 位置 1。
- 无效数据从移位寄存器传送到 USART_RDR 寄存器。
- 单字节通信时无中断产生。然而，在 RXNE 位（使能 FIFO 模式时为 RXFNE 位）生成中断时，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。

通过将 ICR 寄存器中的 NFCF 位置 1 来复位 NE 位。

注： *SPI 模式不支持噪声错误。*

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 457. 16 倍过采样时的数据采样

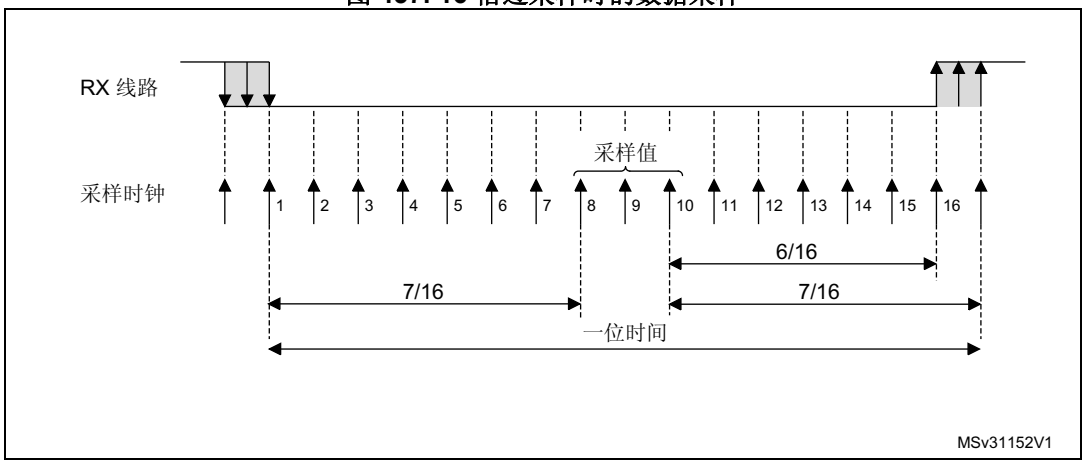


图 458. 8 倍过采样时的数据采样

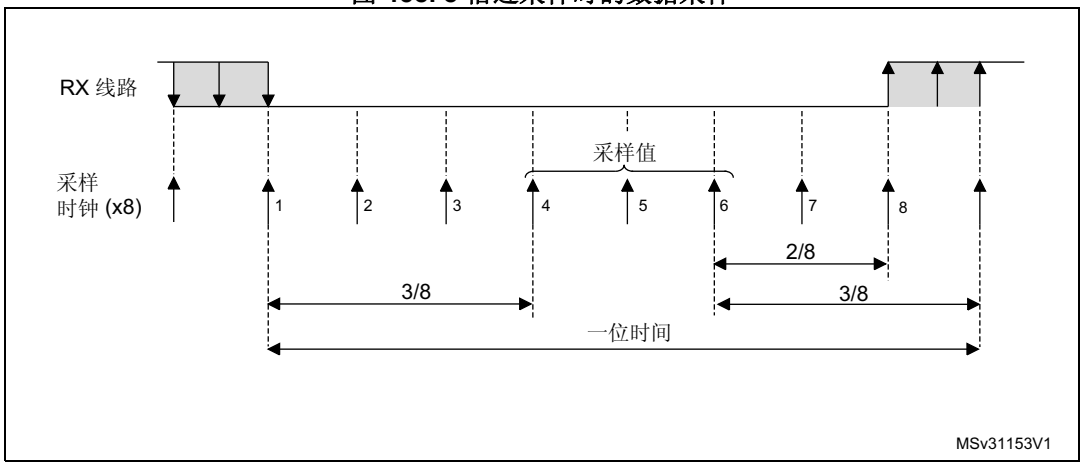


表 263. 通过采样数据进行噪声检测

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

如果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，则会检测到帧错误。

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_RDR 寄存器（使能 FIFO 模式时为 RXFIFO）。
- 单字节通信时无中断产生。然而，在 RXNE 位（使能 FIFO 模式时为 RXFNE 位）生成中断时，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。

通过将“1”写入 USART_ICR 寄存器中的 FECF 位来复位 FE 位。

注： *SPI 模式不支持帧错误。*

接收期间可配置的停止位

可通过 USART_CR 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）。

- **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。
- **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。
- 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 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 标志（使能 FIFO 模式时为 RXFNE）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）（更多详细信息，请参见*第 1485 页的第 44.5.16 节：USART 接收器超时*）。
- 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，则帧错误标志会置 1。

  发生帧错误时不检测第 2 个停止位。RXNE 标志（使能 FIFO 模式时为 RXFNE）将在第一个停止位结束时置 1。

44.5.7 USART 波特率生成

接收器和发送器（Rx 和 Tx）的波特率均设置为 USART_BRR 寄存器中编程的值。

公式 1：适用于标准 USART（包括 SPI 模式）的波特率（OVER8 = “0”或“1”）

在 16 倍过采样的情况下，波特率通过以下公式得出：

$$\text{Tx/Rx 波特} = \frac{\text{usart_ker_ckpres}}{\text{USARTDIV}}$$

在 8 倍过采样的情况下，波特率通过以下公式得出：

$$\text{Tx/Rx 波特} = \frac{2 \times \text{usart_ker_ckpres}}{\text{USARTDIV}}$$

公式 2：智能卡、LIN 和 IrDA 模式下的波特率 (OVER8 = 0)

波特率可根据以下公式得出：

$$\text{Tx/Rx 波特} = \frac{\text{usart_ker_ckpres}}{\text{USARTDIV}}$$

USARTDIV 是一个存放在 USART_BRR 寄存器中的无符号定点数。

- 当 OVER8 = 0 时，BRR = USARTDIV。
- 当 OVER8 = 1 时
  - BRR[2:0] = USARTDIV[3:0]，右移 1 位。
  - BRR[3] 必须保持清零。
  - BRR[15:4] = USARTDIV[15:4]

注： *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。*

16 倍和 8 倍过采样时，USARTDIV 必须大于或等于 16。

如何从 USART_BRR 寄存器中获取 USARTDIV

示例 1

要通过 usart_ker_ck_pres = 8 MHz 获得 9600 波特：

- 16 倍过采样时：
  USARTDIV = 8 000 000/9600
  BRR = USARTDIV = 833d = 0341h
- 8 倍过采样时：
  USARTDIV = 2 * 8 000 000/9600
  USARTDIV = 1666,66 (1667d = 683h)
  BRR[3:0] = 3h >> 1 = 1h
  BRR = 0x681

示例 2

要通过 usart_ker_ck_pres = 48 MHz 获得 921.6 K 波特：

- 16 倍过采样时：
  USARTDIV = 48 000 000/921 600
  BRR = USARTDIV = 52d = 34h
- 8 倍过采样时：
  USARTDIV = 2 * 48 000 000/921 600
  USARTDIV = 104 (104d = 68h)
  BRR[3:0] = USARTDIV[3:0] >> 1 = 8h >> 1 = 4h
  BRR = 0x64

44.5.8 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 + DWU < USART 接收器偏差

其中

DWU 是使用从低功耗模式唤醒时因采样点偏差而产生的误差。

M[1:0] = 01 时：

$$DWU = \frac{t_{WUUSART}}{11 \times Tbit}$$

M[1:0] = 00 时：

$$DWU = \frac{t_{WUUSART}}{10 \times Tbit}$$

M[1:0] = 10 时：

$$DWU = \frac{t_{WUUSART}}{9 \times Tbit}$$

$t_{WUUSART}$ 是检测到起始位下降沿与时钟（由外设请求）就绪、达到外设且调压器就绪之间的时间。

USART 接收器在*表 264* 和*表 265* 中指定的最大容许偏差下可正确接收数据，具体取决于以下设置：

- 由 USART_CR1 寄存器中的 M 位定义的 9 位、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
- USART_BRR 寄存器的 BRR[3:0] 位等于或不等于 0000
- 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 264. BRR [3:0] = 0000 时的 USART 接收器容差

M 位	OVER8 位 = 0		OVER8 位 = 1	
	ONEBIT=0	ONEBIT=1	ONEBIT=0	ONEBIT=1
00	3.75%	4.375%	2.50%	3.75%
01	3.41%	3.97%	2.27%	3.41%
10	4.16%	4.86%	2.77%	4.16%

表 265. BRR[3:0] 不等于 0000 时的 USART 接收器容差

M 位	OVER8 位 = 0		OVER8 位 = 1	
	ONEBIT=0	ONEBIT=1	ONEBIT=0	ONEBIT=1
00	3.33%	3.88%	2%	3%
01	3.03%	3.53%	1.82%	2.73%
10	3.7%	4.31%	2.22%	3.33%

注： *当接收的帧恰好包含 10 个（M 位 = 00）、11 个（M 位 = 01）或 9 个（M 位 = 10）位时间的空闲帧时，*表 264 和表 265 中指定的数据可能与特例中的数据略微不同。

44.5.9 USART 自动波特率检测

USART 可根据接收一个字符检测并自动设置 USART_BRR 寄存器的值。自动波特率检测在以下两种情况下非常有用：

- 事先不知道系统的通信速度。
- 系统正在使用精确度相对较低的时钟源且该机制允许在不测量时钟偏差的情况下获得正确的波特率。

时钟源频率必须与预期通信速度兼容。

- 16 倍过采样时，波特率范围为 usart_ker_ck_pres/65535 到 usart_ker_ck_pres/16。
- 8 倍过采样时，波特率范围为 usart_ker_ck_pres/65535 到 usart_ker_ck_pres/8。

在激活自动波特率检测之前，必须通过 USART_CR2 寄存器中的 ABRMOD[1:0] 字段选择自动波特率检测模式。根据不同的字符模式，存在四种检测模式。在这些自动波特率模式下，波特率在同步接收数据期间被多次测量，每次测量的结果都与前一次进行比较。

这些模式如下：

- **模式 0**：以“1”位开头的任意字符。

  这种情况下，USART 会测量起始位的持续时间（下降沿到上升沿）。
- **模式 1**：以 10xx 位模式开头的任意字符。

  这种情况下，USART 会测量起始位和第一个数据位的持续时间。测量在下降沿到下降沿期间完成，可在信号斜率较小时确保较高的精度。
- **模式 2**：0x7F 字符帧（可以是 LSB 在前模式下的 0x7F 字符，也可以是 MSB 在前模式下的 0xFE 字符）。

  这种情况下，先在起始位结束时更新波特率 (BR)，然后在位 6 结束时更新波特率（根据从下降沿到下降沿执行的测量：BR6）。以 BR 对位 0 到位 6 进行采样，而以 BR6 对字符的其他位进行采样。
- **模式 3**：0x55 字符帧。

  这种情况下，先在起始位结束时更新波特率 (BR)，然后在位 0 结束时更新波特率（根据从下降沿到下降沿执行的测量：BR0），最后在位 6 结束时更新波特率 (BR6)。以 BR 对位 0 进行采样，以 BR0 对位 1 到位 6 进行采样，以 BR6 对字符的其他位进行采样。同时，对 RX 线路的各个中间转换执行其他检查。如果 RX 上的转换与接收器（基于根据位 0 计算的波特率的接收器）未充分同步，则生成错误。

激活自动波特率检测之前，必须先通过向 USART_BRR 寄存器写入非零的波特率值来初始化该寄存器。

通过将 USART_CR2 寄存器中的 ABREN 位置 1 来激活自动波特率检测。之后 USART 将等待 RX 线路上的第一个字符。通过将 USART_ISR 寄存器中的 ABRF 标志置 1 来指示自动波特率操作完成。如果线路繁忙，则无法保证正确的波特率检测。这种情况下，BRR 值可能会损坏，ABRE 错误标志位将置 1。如果通信速度不在自动波特率检测范围（位持续时间不在 16 个和 65536 个时钟周期（16 倍过采样时）之间，也不在 8 个和 65536 个时钟周期（8 倍过采样时）之间）内，也会出现这种情况。

稍后，可通过复位 ABRF 标志（通过写入“0”）重新启动自动波特率检测。

禁止 FIFO 管理且发生自动波特率错误时，ABRE 标志通过 RXNE 和 FE 位置 1。

使能 FIFO 管理且发生自动波特率错误时，ABRE 标志通过 RXFNE 和 FE 位置 1。

如果使能 FIFO 模式，则应使用第一个 RXFIFO 位置上的数据进行自动波特率检测。因此，在启动自动波特率检测之前，请通过检查 USART_ISR 寄存器的 RXFNE 标志来确保 RXFIFO 为空。

注： *如果在自动波特率操作期间禁止 USART (UE=0)，则可能损坏 BRR 值。*

44.5.10 USART 多处理器通信

可以执行 USART 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 输出与其他 USART 的 RX 输入相连；而其他 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。

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。

可通过静默功能将未被寻址的器件置于静默模式下。为了使用静默模式功能，必须将 USART_CR1 寄存器中的 MME 位置 1。

注： *使能 FIFO 管理且 MME 已置 1 时，不得清零 MME 位再快速将其置 1（在两个 usart_ker_ck 周期内），否则静默模式可能保持有效。*

使能静默模式时：

- 不得将接收状态位置 1；
- 禁止任何接收中断；
- USART_ISR 寄存器中的 RWU 位置“1”。在某些情况下，RWU 可以由硬件或软件通过 USART_RQR 寄存器中的 MMRQ 位自动控制。

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：

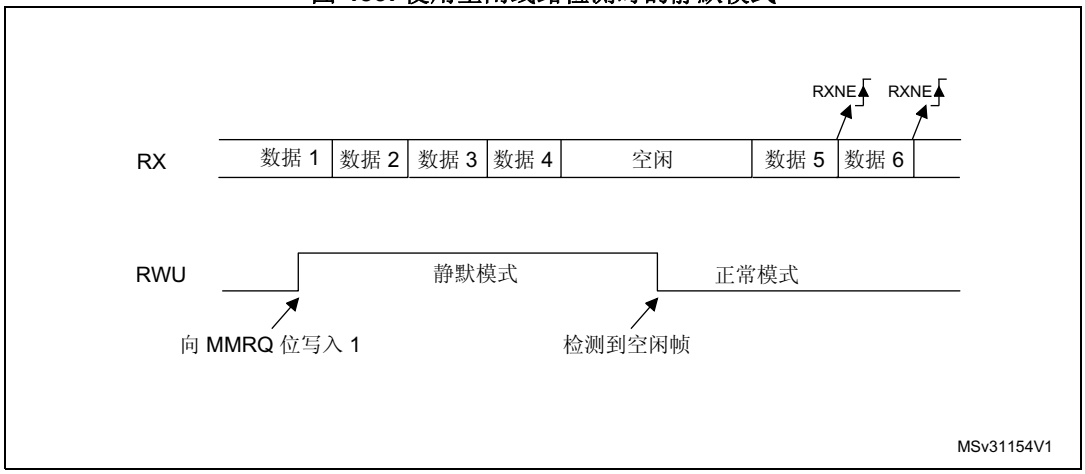
- 如果 WAKE 位被复位，则进行空闲线路检测。
- 如果 WAKE 位置 1，则进行地址标记检测。

空闲线路检测 (WAKE=0)

向 MMRQ 位写入“1”且 RWU 位自动置 1 时，USART 进入静默模式。

检测到空闲帧时，USART 将唤醒。此时 RWU 位会由硬件清零，但 USART_ISR 寄存器中的 IDLE 位不会置 1。*图 459* 中给出了使用空闲线路检测时静默模式行为的示例。

图 459. 使用空闲线路检测时的静默模式



注： *如果在 IDLE 字符已经过去时将 MMRQ 位置 1，将不会进入静默模式（RWU 未置 1）。*

如果在线路处于空闲状态时激活 USART，在一个 IDLE 帧持续时间后（不只在接收一个字符帧后）会检测到空闲状态。

4 位/7 位地址标记检测 (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 个或 7 个 LSB 中。7 位或 4 位地址检测通过 ADDM7 位来选择。接收器会将此 4 位/7 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。

注： *在 7 位和 9 位数据模式下，地址检测分别在 6 位和 8 位地址上完成（ADD[5:0] 和 ADD[7:0]）。*

当接收到与其编程地址不匹配的地址字符时，USART 会进入静默模式。此时，RWU 位将由硬件置 1。USART 进入静默模式后，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。使能 FIFO 管理时，软件应确保在进入静默模式之前 RXFIFO 中至少有一个空位置。

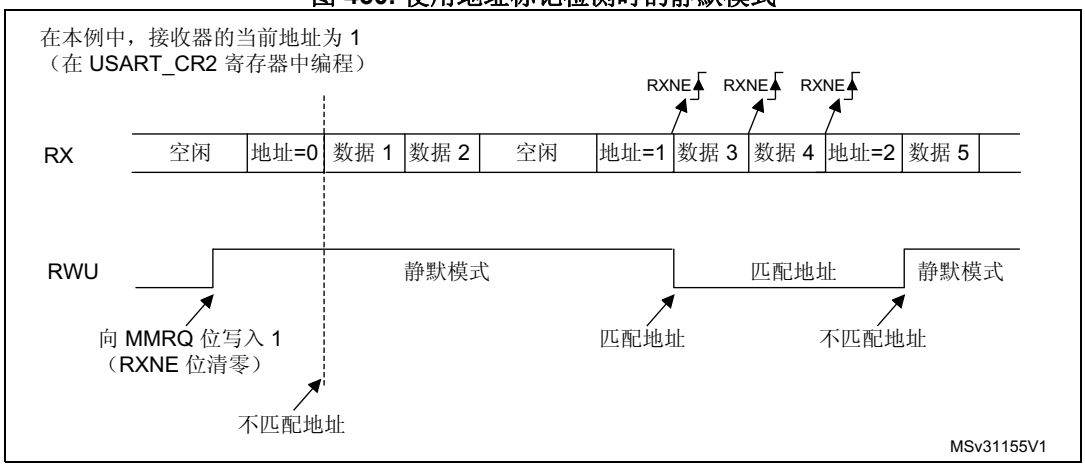
当向 MMRQ 位写入 1 时，USART 也会进入静默模式。这种情况下，RWU 位也自动置 1。

当接收到与编程地址匹配的地址字符时，USART 会退出静默模式。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/RXFNE 位会针对地址字符置 1。

注： *使能 FIFO 管理时，如果在接收器对数据的最后一位进行采样时 MMRQ 置 1，则可在有效进入静默模式之前接收该数据。*

图 460 中给出了使用地址标记检测时静默模式行为的示例。

图 460. 使用地址标记检测时的静默模式



44.5.11 USART Modbus 通信

USART 为 Modbus/RTU 和 Modbus/ASCII 协议的实现提供基本支持。Modbus/RTU 是一个半双工块传输协议。该协议的控制部分（地址识别、块完整性控制和命令解析）必须用软件实现。

USART 为块结束检测提供基本支持，无需软件开销或其他资源。

Modbus/RTU

在此模式下，一个块的结束通过超过 2 个字符时间的“静默”（空闲线路）来识别。此功能通过可编程的超时功能实现。

超时功能和中断必须分别通过 USART_CR2 寄存器中的 RTOEN 位和 USART_CR1 寄存器中的 RTOIE 位激活。与 2 个字符时间（例如 22 个位时间）的超时相对应的值必须在 RTO 寄存器中编程。如果在此期间接收线路空闲，则在接收到最后一个停止位后，将生成中断，同时通知软件当前块接收已完成。

Modbus/ASCII

在此模式下，块结束通过特定 (CR/LF) 字符序列识别。USART 通过字符匹配功能管理此机制。

通过在 ADD[7:0] 字段中编程 LF ASCII 码以及激活字符匹配中断 (CMIE=1)，软件可在接收到 LF 时获得通知并检查 DMA 缓存区中的 CR/LF。

44.5.12 USART 极性控制

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 266* 中列出了可能的 USART 帧格式。

表 266. USART 帧格式

M 位	PCE 位	USART 帧(1)
00	0	| SB | 8 位数据 | STB |
00	1	| SB | 7 位数据 | PB | STB |
01	0	| SB | 9 位数据 | STB |
01	1	| SB | 8 位数据 PB | STB |
10	0	| SB | 7 位数据 | STB |
10	1	| SB | 6 位数据 | PB | STB |

1. 图注：SB：起始位，STB：停止位，P：奇偶校验位。在数据寄存器中，PB 始终位于 MSB 位置（第 8 位或第 7 位，具体取决于 M 位的值）。

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 6 个、7 个或 8 个 LSB 位组成，具体取决于 M 位的值）。

例如，如果数据=00110101 且 4 个位置 1，则在选择偶校验（USART_CR1 寄存器中的 PS 位 = 0）时，校验位为 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 6 个、7 个或 8 个 LSB 位组成，具体取决于 M 位的值）。

例如，如果数据=00110101 且 4 个位置 1，则在选择奇校验（USART_CR1 寄存器中的 PS 位 = 1）时，校验位为 1。

接收时进行奇偶校验检查

如果奇偶校验检查失败，则 USART_ISR 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。通过软件将 1 写入 USART_ICR 寄存器中的 PECF 位来清零 PE 标志。

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）。

44.5.13 USART LIN（局域互连网络）模式

仅在支持 LIN 模式时才与本节相关。请参见*第 1455 页的第 44.4 节：USART 实现*。

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 44.5.4 节*中介绍的步骤，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 SBKRQ 位置 1 会发送 13 个“0”位作为中断字符。然后会发送值为“1”的 2 个位以进行下一启动检测。

LIN 接收

使能 LIN 模式后，将激活中断检测电路。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，只要发生中断即可检测出来。

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中断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 中 LBDL = 0 时）或 11 个（USART_CR2 中 LBDL=1 时）连续位均检测为“0”，且其后跟随分隔符，则 USART_ISR 寄存器中的 LBDF 标志将置 1。如果 LBDIE 位=1，则会生成中断。在验证中断前，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，则中断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。

*第 1480 页的图 461：LIN 模式下的中断检测（11 位中断长度——LBDL 位置 1）*中显示了中断检测器状态机和中断标志的行为。

*第 1481 页的图 462：LIN 模式下的中断检测与帧错误检测*中列出了中断帧的示例。

图 461. LIN 模式下的中断检测（11 位中断长度——LBDL 位置 1）

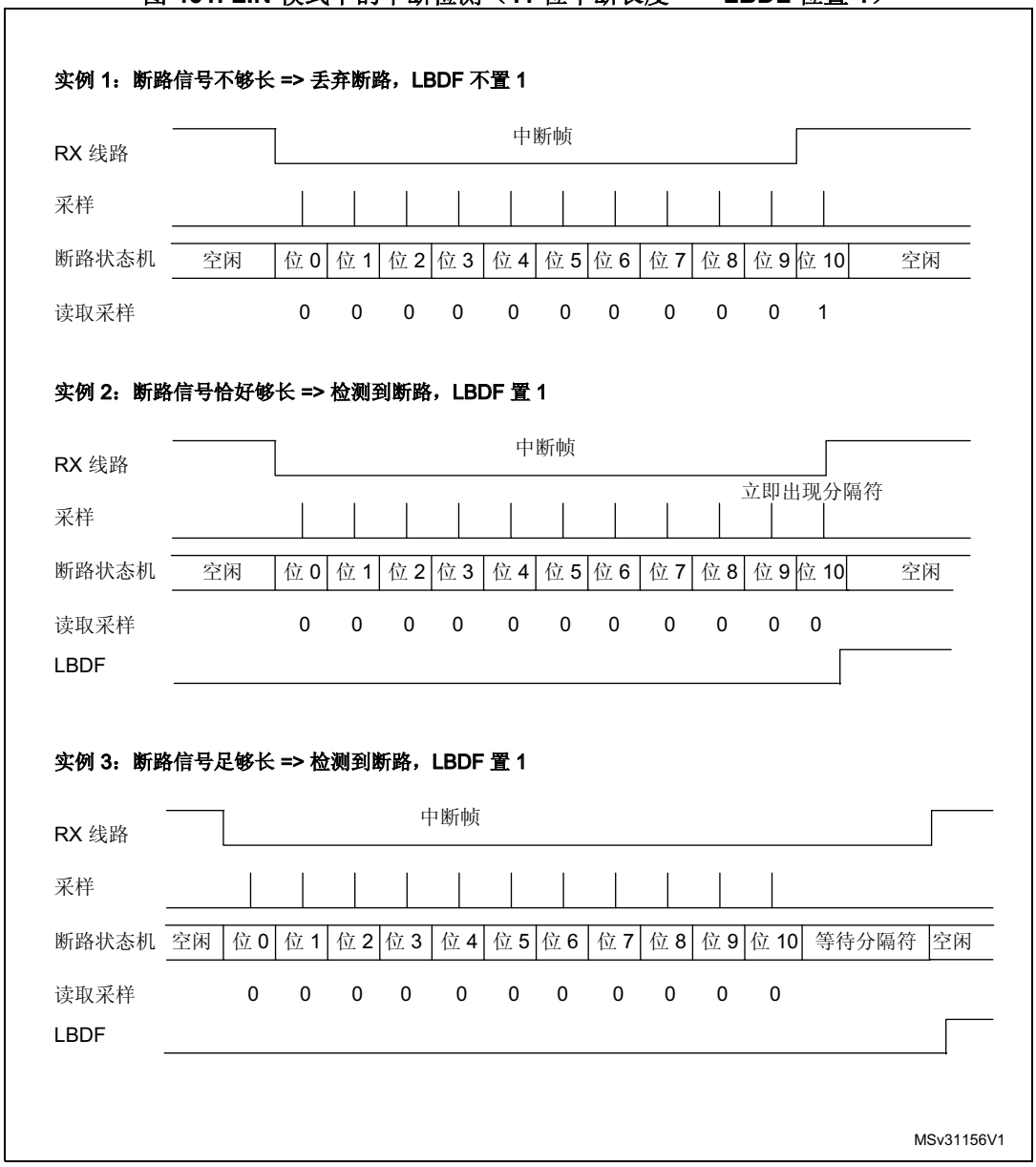
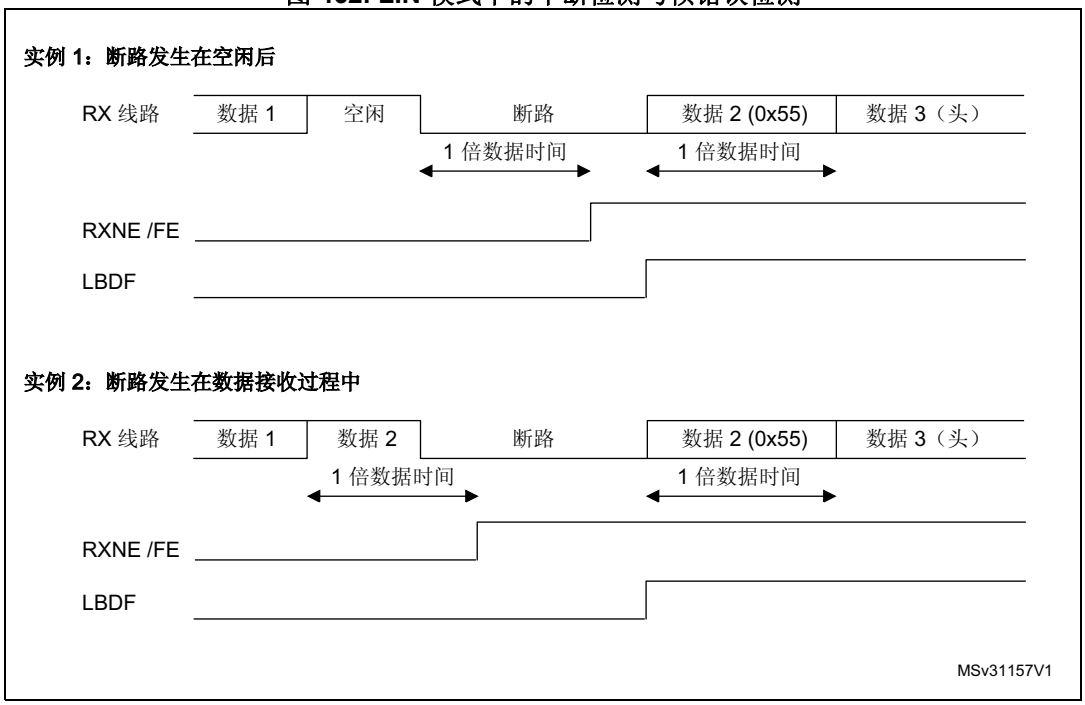


图 462. LIN 模式下的中断检测与帧错误检测



44.5.14 USART 同步模式

主模式

通过将 USART_CR2 寄存器中的 CLKEN 位编程为“1”来选择同步主模式。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。

在此模式下，USART 可用于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，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。USART_CR2 寄存器中的 CPOL 位用于选择时钟极性；USART_CR2 寄存器中的 CPHA 位用于选择外部时钟的相位（请参见*图 463*、*图 464* 和*图 465*）。

在空闲状态、报头模式和发送中断期间，外部 SCLK 时钟处于未激活状态。

在同步主模式下，USART 发送器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。

在同步主模式下，USART 接收器的工作方式与异步模式下不同。如果 RE 置 1，则数据在 SCLK 上采样（上升或下降沿，具体取决于 CPOL 和 CPHA），而不会进行任何过采样。此时必须确保给定建立时间和保持时间（取决于波特率：1/16 位时间）。

注： *在主模式下，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（USART_TDR 数据寄存器已写入），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。*

图 463. USART 同步主发送示例

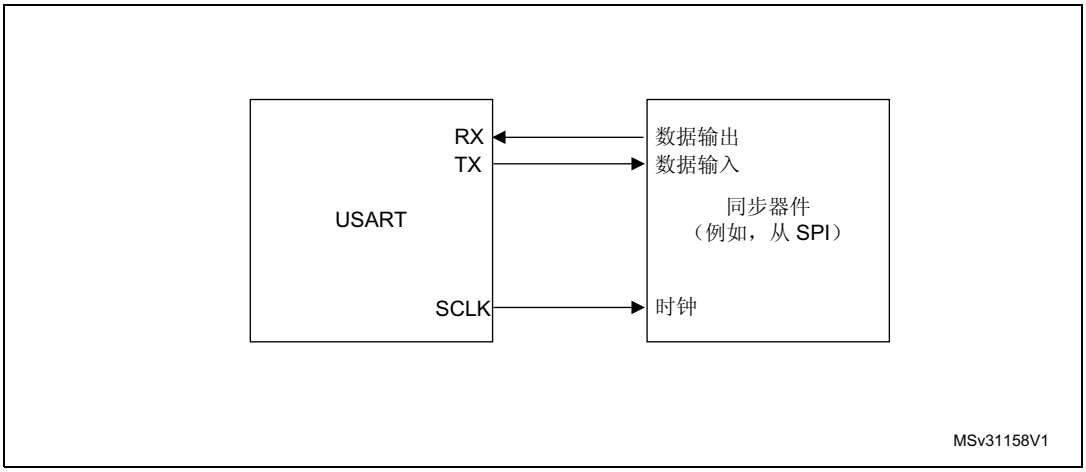


图 464. USART 在同步主模式下的数据时钟时序图（M 位 = 00）

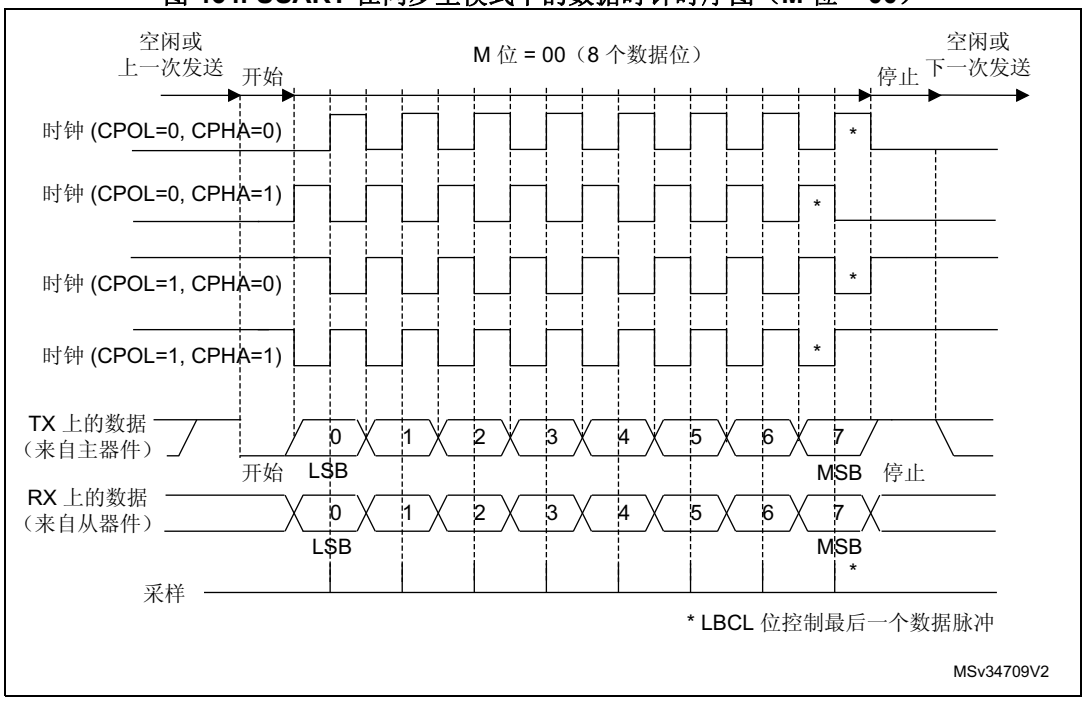
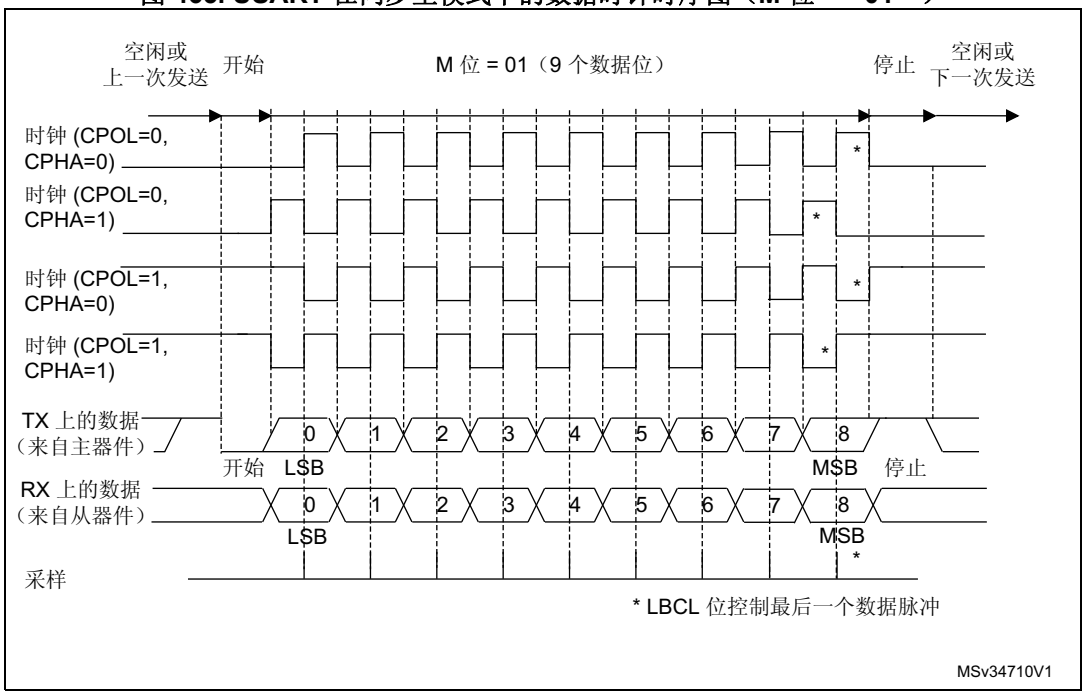


图 465. USART 在同步主模式下的数据时钟时序图（M 位 =“01”）



从模式

通过将 USART_CR2 寄存器中的 SLVEN 位编程为“1”来选择同步从模式。在同步从模式下，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、HDSEL 和 IREN 位。

在此模式下，USART 可用于在从模式下控制双向同步串行通信。在从模式下，SCLK 引脚是 USART 的输入。

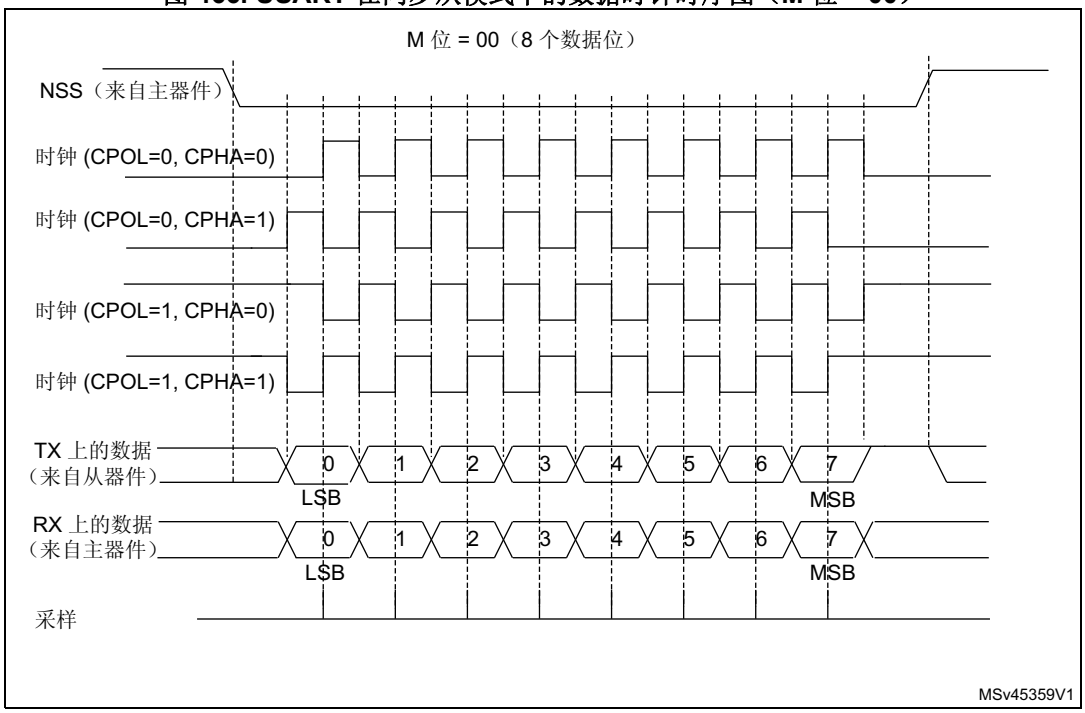
注： *当外设用于 SPI 从器件模式时，外设时钟源 (usart_ker_ck_pres) 的频率必须是 CK 输入频率的 3 倍以上。*

USART_CR2 寄存器中的 CPOL 位和 CPHA 位分别用于选择时钟极性和外部时钟的相位（请参见*图 466*）。

从发送模式支持下溢错误标志。如果在软件尚未将任何值加载到 USART_TDR 之前出现第一个数据发送时钟脉冲，则此标志将置 1。

从器件支持硬件和软件 NSS 管理。

图 466. USART 在同步从模式下的数据时钟时序图（M 位 = 00）



从器件选择 (NSS) 引脚管理

可通过 USART_CR2 寄存器中的 DIS_NSS 位设置硬件或软件从器件选择管理：

- 软件 NSS 管理 (DIS_NSS = 1)
  将始终选择 SPI 从器件并忽略 NSS 输入引脚。
  外部 NSS 引脚空闲，可供其他应用使用。
- 硬件 NSS 管理 (DIS_NSS = 0)
  SPI 从器件选择取决于 NSS 输入引脚。NSS 为低电平时选择从器件，NSS 为高电平时取消选择从器件。

注： *禁止 USART 时 (UE=0)，必须选择 LBCL（仅用于 SPI 主器件模式）、CPOL 和 CPHA 位以确保时钟脉冲正常工作。*

在 SPI 从器件模式下，必须在启动主器件通信之前（或时钟稳定时在相邻帧之间）使能 USART。否则，如果 USART 从器件在主器件处于某个帧中间时使能，则它将与主器件失去同步。在通信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，否则 SPI 从器件会发送零。

SPI 从器件下溢错误

发生下溢错误时，USART_ISR 寄存器中的 UDR 标志会置 1，SPI 从器件继续发送最后一个数据，直到下溢错误标志由软件清零。

下溢标志在帧开始时置 1。如果 USART_CR3 寄存器中的 EIE 位置 1，则会触发下溢错误中断。

通过将 USART_ICR 寄存器中的位 UDRCF 置 1 来清零下溢错误标志。

发生下溢错误时，仍然可以对 TDR 寄存器进行写操作。清除下溢错误将允许发送新数据。

如果发生下溢错误且没有新数据被写入 TDR 中，则 TC 标志在帧结束时置 1。

注： *如果向 USART_TDR 写入数据的时间点过于接近第一个 SCLK 发送边沿，则可能会发生下溢错误。为避免发生此下溢错误，应在第一个 SCLK 边沿的 3 个 usart_ker_ck 周期之前对 USART_TDR 进行写操作。*

44.5.15 USART 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 USART_CR3 寄存器中的控制位 HDSEL 可在半双工通信和全双工通信间进行选择。

向 HDSEL 位写入“1”后：

- 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，以便将 TX 配置为复用功能开漏并外接上拉电阻。

除此之外，通信协议与正常 USART 模式下的通信协议相似。此线路上的任何冲突都必须由软件管理（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。

44.5.16 USART 接收器超时

接收器超时功能可通过将 USART_CR2 控制寄存器中的 RTOEN 位置 1 来使能。

超时间隔通过 USART_RTOR 寄存器中的 RTO 位域进行编程。

接收器超时计数器遵循以下规则开始计数：

- STOP =“00”或 STOP =“11”时从停止位结束时开始计数。
- STOP =“10”时从第二个停止位结束时开始计数。
- STOP =“01”时从停止位开始时开始计数。

经过超时间隔后，USART_ISR 寄存器中的 RTOF 标志置 1。如果 USART_CR1 寄存器中的 RTOIE 位置 1，则会产生超时。

44.5.17 USART 智能卡模式

仅在支持智能卡模式时才涉及本节内容。请参见*第 1455 页的第 44.4 节：USART 实现*。

通过将 USART_CR3 寄存器中的 SCEN 位置 1 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，还可将 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 标准的异步智能卡协议。同时支持 T=0（字符模式）和 T=1（块模式）。

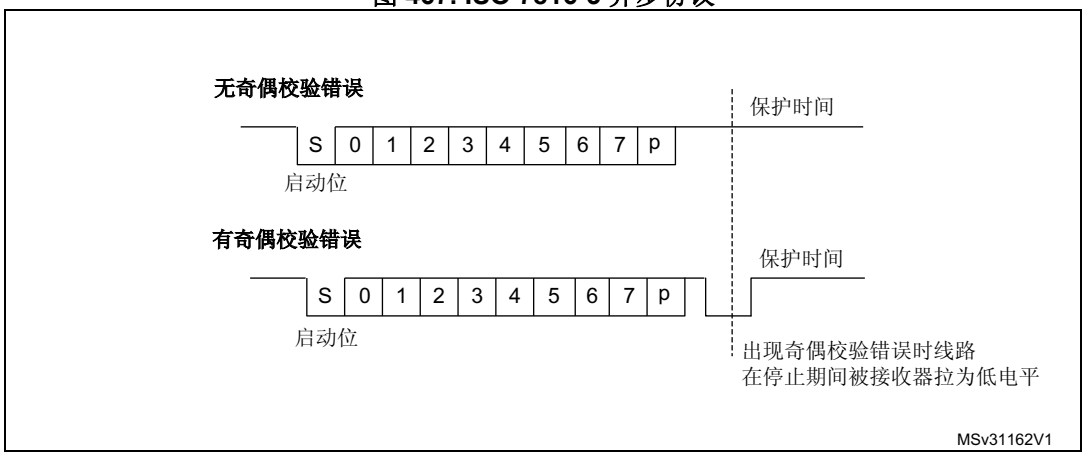
USART 应如下所示进行配置：

- 8 个位加奇偶校验：当 USART_CR1 寄存器中的 M=1 且 PCE=1 时。
- 发送和接收数据时使用 1.5 个停止位：当 USART_CR2 寄存器中的 STOP=“11”时。接收时也可以选择 0.5 个停止位。

在 T=0（字符）模式下，奇偶校验错误在保护时间周期内的每个字符结束时指示。

图 467 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 467. 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。

智能卡模式采用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波特时钟边沿开始移位。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。
- 发送时，如果智能卡检测到奇偶校验错误，它会通过将线路驱动为低电平 (NACK) 告知 USART 此状态。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。USART 可根据协议自动重新发送数据。重试次数在 SCARCNT 位域中编程。如果经过编程次数的重试后，USART 继续收到 NACK，USART 会停止发送并将该错误以帧错误形式发出。TXE 位（使能 FIFO 模式时为 TXFNF 位）可使用 USART_RQR 寄存器中的 TXFRQ 位置 1。

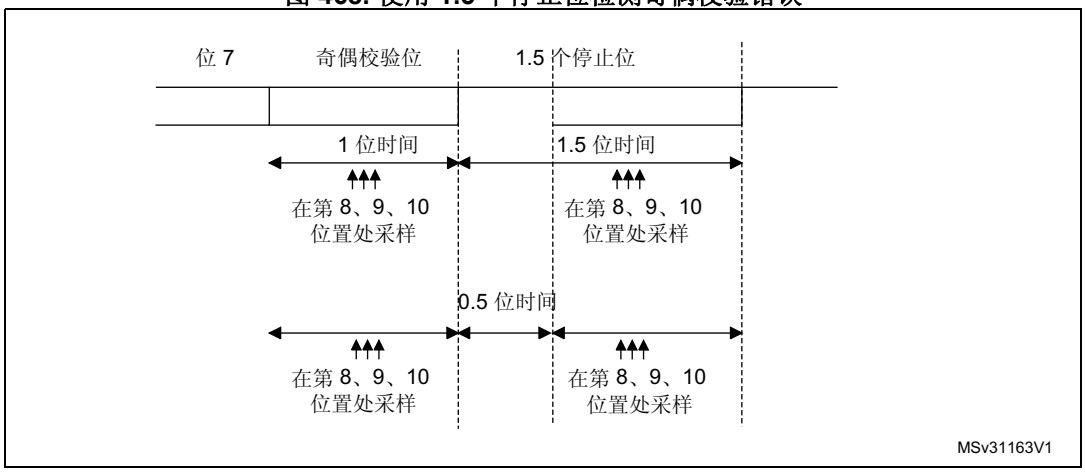
- 发送时智能卡自动重试：在 USART 检测 NACK 与重复字符的起始位之间插入 2.5 个波特率周期的延迟。接收最后一个重复字符后，立即将 TC 位置 1（无保护时间）。如果软件要重复此操作，必须确保标准要求的最短 2 个波特率周期。
- 如果在接收一个使用 1.5 个停止位编程的帧期间检测到奇偶校验错误，则在完成接收帧后，发送线会被拉低一个时钟周期。这是为了向智能卡指出发送到 USART 的数据尚未正确接收。如果 NACK 控制位置 1，接收器会向奇偶校验错误发送“NACK”信号；否则不会发送 NACK 信号（将在 T=1 模式下会使用）。如果接收到的字符错误，则不会激活 RXNE（使能 FIFO 模式时为 RXFNE）/接收 DMA 请求。根据协议规范，智能卡必须重新发送相同的字符。如果经过 SCARCNT 位域中指定的最大重试次数后接收到的字符仍然错误，USART 会停止发送 NACK 信号，并将错误以奇偶校验错误的形式发出。
- 接收时智能卡自动重试：如果 USART 向智能卡发送 NACK 信号，但智能卡不重复字符，则 BUSY 标志将保持置 1。
- 发送时，USART 会在两个连续字符之间插入保护时间（按照保护时间寄存器中编程的值）。由于保护时间在前一个字符的停止位后测量，因此必须将 GT[7:0] 寄存器编程为所需 CGT（字符保护时间，如 7816-3 规范所定义）减去 12（一个字符的持续时间）。
- 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。TCBGT 标志可用于检测数据传输是否结束，而无需等待保护时间结束。该标志在帧发送结束之后且未从智能卡接收到 NACK 时置 1。
- 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： *中断字符在智能卡模式下无效。带有帧错误的 0x00 数据被视为数据，而非中断。*

当翻转 TE 位时，不会发送空闲帧。空闲帧（在其他配置中进行了定义）在 ISO 协议中未进行定义。

图 468 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 468. 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 USART_GTPR 寄存器中进行配置。SCLK 频率可在 usart_ker_ck_pres/2 到 usart_ker_ck_pres/62 之间进行编程，其中 usart_ker_ck_pres 为经过编程的预分频器分频的外设输入时钟。

块模式 (T=1)

在 T=1（块）模式下，通过将 UART_CR3 寄存器中的 NACK 位清零可停用奇偶校验错误发送。

在块模式下请求从智能卡执行读操作时，软件必须将 RTOR 寄存器编程为 BWT（块等待时间）- 值 11。如果在经过此时间段后未从智能卡接收到应答，将生成超时中断。如果该时间段后接收到第一个字符，则通过 RXNE/RXFNE 中断发出信号指示。

注： *即使在使用 DMA 模式下的 USART 从块模式下的智能卡读取时，也必须使能 RXNE/RXFNE 中断。同时，只有在接收到第一个字节后才可使能 DMA。*

接收到第一个字符（RXNE/RXFNE 中断）后，为允许自动检查两个连续字符间的最长等待时间，必须将 RTO 寄存器编程为 CWT（字符等待时间 - 值 11）。此时间以波特率时间单位表示。前一个字符结束后，如果智能卡未在小于 CWT 的时间段内发送新字符，USART 将通过 RTOF 标志和中断（当 RTOIE 位置 1 时）向软件指示此情况。

注： *按照智能卡协议定义，BWT/CWT 的值应从最后一个字符开始（起始位）时定义。必须将 RTO 寄存器分别编程为 BWT -11 或 CWT -11，并考虑最后一个字符本身的长度。*

块长度计数器用于对 USART 接收到的所有字符进行计数。当 USART 进行发送时，此计数器复位。块长度由智能卡在块的第三个字节（起始字段）中传达。必须将此值编程到 USART_RTOR 寄存器中的 BLEN 字段。使用 DMA 模式时，在块开始之前，必须将此寄存器字段编程为最小值 (0x0)。对于该值，在接收到第四个字符后生成中断。软件必须读取 LEN 字段（第三个字节），其值必须从接收缓冲区中读取。

在中断驱动接收模式下，块长度可以由软件或者通过编程 BLEN 值来检查。但是在块开始前，可以编程 BLEN 的最大值 (0xFF)。接收到第三个字符后，将编程实际值。

如果块使用 LRC（纵向冗余校验，1 个结尾字节），则 BLEN=LEN。如果块使用 CRC 机制（2 个结尾字节），则必须编程 BLEN=LEN+1。块总长度（包括起始字段、结尾字段和信息字段）等于 BLEN+4。块结束的信号通过 EOBF 标志和中断（EOBIE 位置 1 时）发送给软件。

如果块长度出现错误，则通过 RTO 中断（字符等待时间上溢）发送块结束信号。

注： *错误检查代码 (LRC/CRC) 必须通过软件计算/验证。*

正向约定和反向约定

智能卡协议定义了两种约定：正向约定和反向约定。

正向约定定义为：LSB 在前，逻辑位值 1 对应于线路的 H 状态，奇偶校验为偶校验。要使用此约定，必须编程以下控制位：MSBFIRST=0，DATAINV=0（默认值）。

反向约定定义为：MSB 在前，逻辑位值 1 对应于信号线路的 L 状态，奇偶校验为偶校验。要使用此约定，必须编程以下控制位：MSBFIRST=1，DATAINV=1。

注： *将逻辑数据值取反（0=H，1=L）时，奇偶校验位将同样取反。*

为识别智能卡约定，智能卡会将初始字符 TS 作为 ATR（复位应答）的第一个字符发送。TS 支持两种格式：LHHL LLL LLH 和 LHHL HHH LLH。

- (H) LHHL LLL LLH 建立反向约定：状态 L 编码为值 1，时间分量 2 传送最高有效位（MSB 在前）。按反向约定解码时，传送的字节等于“3F”。
- (H) LHHL HHH LLH 建立正向约定：状态 H 编码为值 1，时间分量 2 传送最低有效位（LSB 在前）。按正向约定解码时，传送的字节等于 “3B”。

在 2 到 10 的九个时间分量中，如果有偶数个位设置为 1，则字符的奇偶校验正确。

由于 USART 不了解智能卡使用哪种约定，因此 USART 需要能够识别任意一种模式并相应操作。模式识别不在硬件中完成，而是通过软件序列完成。此外，假设以正向约定配置 USART（默认）而智能卡以反向约定 (TS = LHHL LLL LLH) 应答，则 USART 接收到的字节将为“03”，奇偶校验将为奇校验。

因此，有两种方法可用于识别 TS 模式：

方法 1

将 USART 编程为标准智能卡模式/正向约定。这种情况下，TS 模式接收会向智能卡生成奇偶校验错误中断和错误信号。

- 奇偶校验错误中断通知软件智能卡未以正向约定正确应答。之后，软件重新将 USART 编程为反向约定
- 为响应错误信号，智能卡会重试同一 TS 字符，此时重新编程后的 USART 将正确接收该字符

或者，为应答奇偶校验错误中断，软件可决定重新编程 USART，同时向智能卡生成新的复位命令，然后再次等待 TS。

方法 2

将 USART 编程为 9 位/无奇偶校验模式，无位反向。在此模式下，按如下方式接收两种 TS 模式中的任一种：

(H) LHHL LLL LLH = 0x103 -> 选择反向约定

(H) LHHL HHH LLH = 0x13B -> 选择正向约定

软件根据这两种模式检查接收到的字符，如果其中任意一种匹配，则相应编程 USART 以接收下一个字符。

如果两种都未被识别，则可能复位智能卡以重新开始协商。

44.5.18 USART IrDA SIR ENDEC 模块

仅在支持 IrDA 模式时才涉及本节内容。请参见*第 1455 页的第 44.4 节：USART 实现*。

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 469*）。

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 115.2 Kbps。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（USART 正在向 IrDA 编码器发送数据时），则 IrDA 解码器会忽略 IrDA 接收线上的所有数据；如果接收器忙（USART 正在接收来自 USART 的解码数据时），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 470*）。
- 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 µs。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 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 寄存器中的停止位必须配置为“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。只有当持续时间长于 2 个 IrDA 低功耗波特时钟周期（USART_GTPR 的 PSC 值）时，才是有效低电平。

注： *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 469. IrDA SIR ENDEC 框图

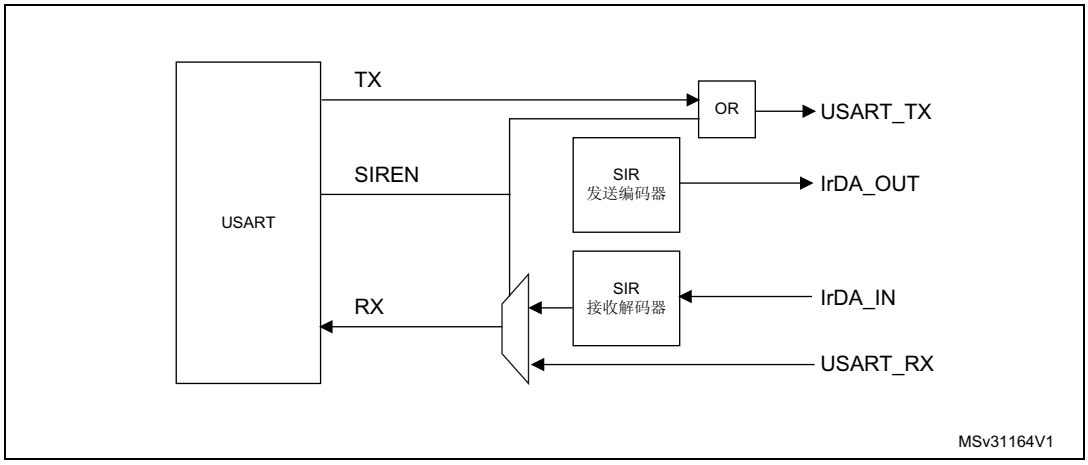
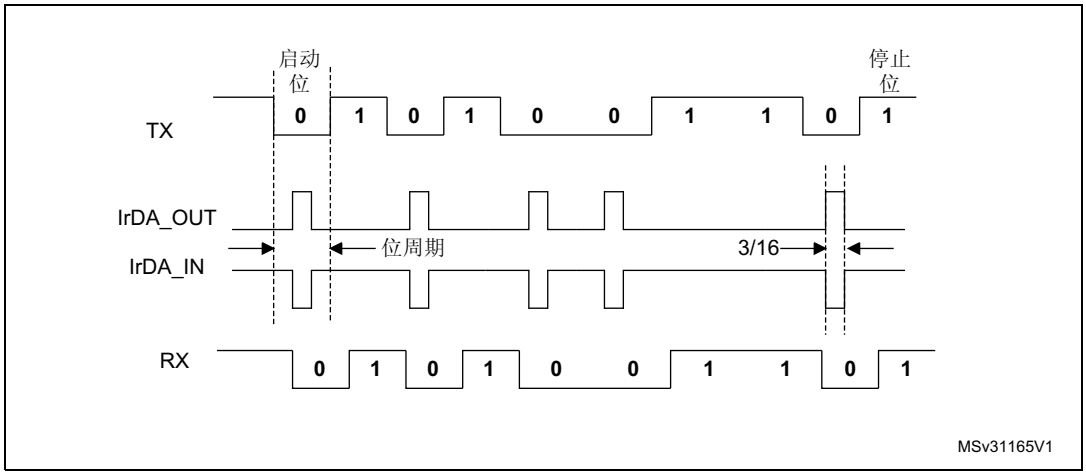


图 470. IrDA 数据调制 (3/16)——正常模式



44.5.19 使用 USART 和 DMA 进行连续通信

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 请求是独立生成的。

注： *要确定是否支持 DMA 模式，请参见第 1455 页的第 44.4 节：USART 实现。如果不支持 DMA 模式，请按照第 44.5.6 节中的说明使用 USART。可以将 USART_ISR 寄存器中的 TXE/RXNE 标志清零，从而在禁止 FIFO 时实现连续通信。*

使用 DMA 进行发送

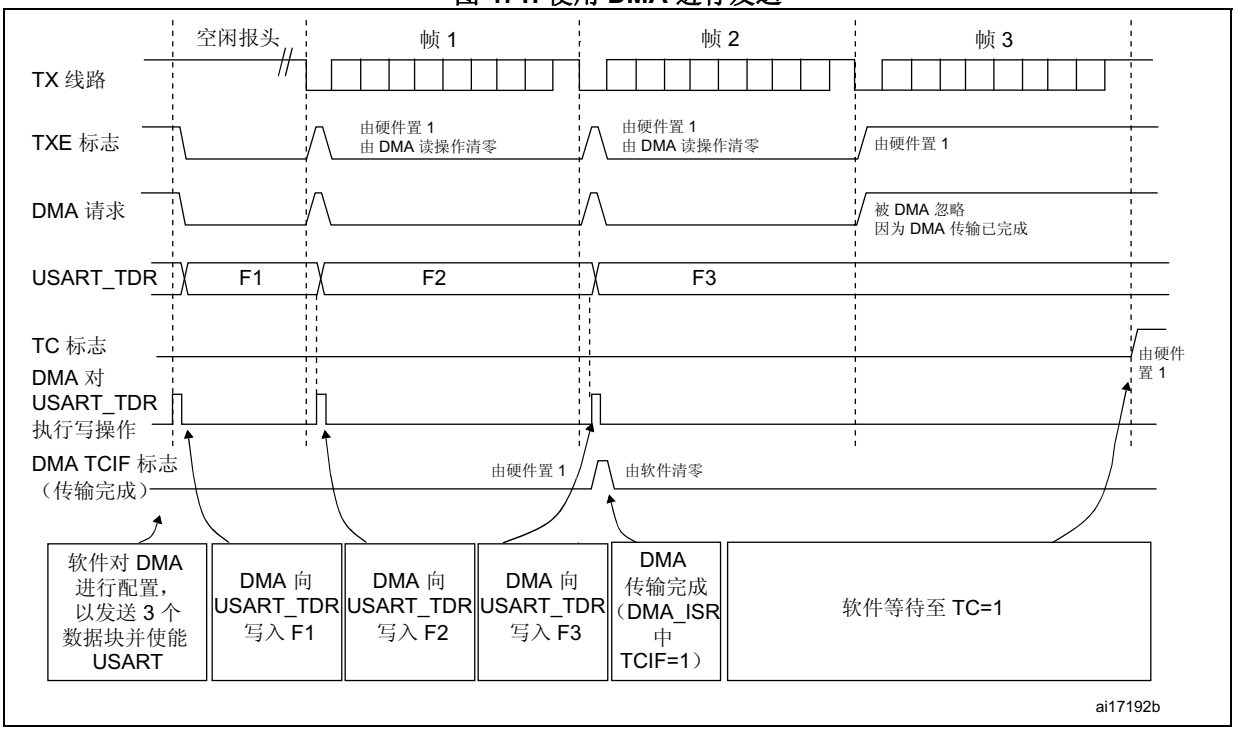
将 USART_CR3 寄存器中的 DMAT 位置 1 可以使能 DMA 模式进行发送。当 TXE 标志（使能 FIFO 模式时为 TXFNF 标志）置 1 时，可将数据从配置的 SRAM 区（参见直接存储访问控制器的第 8 和第 11 节）加载到 USART_TDR 寄存器。要映射一个 DMA 通道以进行 USART 发送，请按以下步骤操作（x 表示通道编号）：

1. 在 DMA 控制寄存器中写入 USART_TDR 寄存器地址，将其配置为传输的目标地址。每次发生 TXE（使能 FIFO 模式时为 TXFNF）事件后，数据都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（使能 FIFO 模式时为 TXFNF）事件后，数据都从该存储区域加载到 USART_TDR 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. 通过将 USART_ICR 寄存器中的 TCCF 位置 1，将 USART_ISR 寄存器中的 TC 标志清零。
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 之前或系统进入低功耗模式之前（禁止外设时钟时）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保持清零状态，然后在最后一帧发送结束时由硬件置 1。

图 471. 使用 DMA 进行发送



注： *使能 FIFO 管理时，DMA 请求由发送 FIFO 未满（即 TXFNF = 1）触发。*

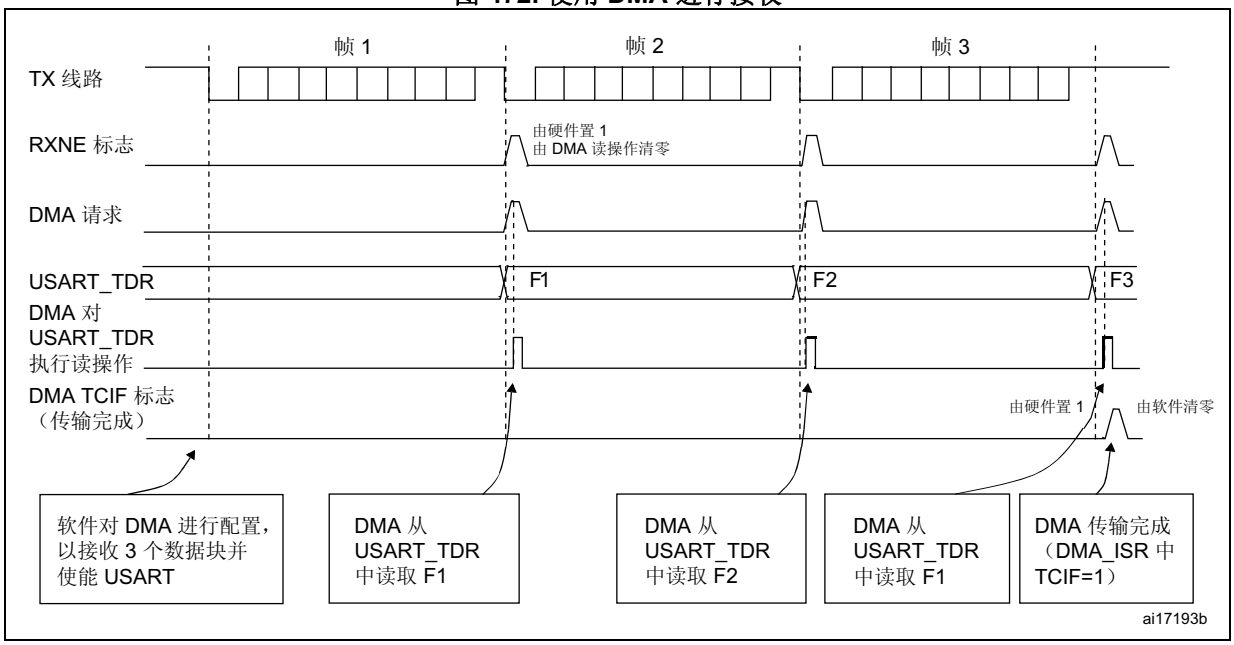
使用 DMA 进行接收

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_RDR 寄存器加载到配置的 SRAM 区域中（参见直接存储访问控制器的第 8 和第 11 节）。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_RDR 寄存器地址，将其配置为传输的源地址。每次发生 RXNE（使能 FIFO 模式时为 RXFNE）事件后，数据都从该地址移动到存储器。
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（使能 FIFO 模式时为 RXFNE）事件后，数据都从 USART_RDR 加载到此存储区域。
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。

图 472. 使用 DMA 进行接收



注： *使能 FIFO 管理时，DMA 请求由接收 FIFO 非空（即 RXFNE = 1）触发。*

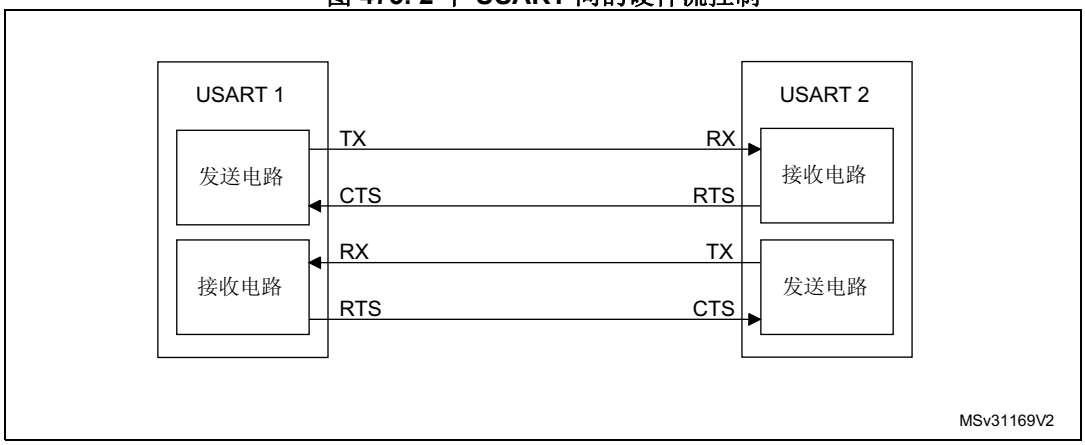
多缓冲区通信中的错误标志和中断生成

在多缓冲区通信模式下，如果事务中发生任何错误，都会在当前字节后将相应错误标志置位。如果中断使能标志置 1，则会产生中断。在单字节接收过程中，与 RXNE（使能 FIFO 模式时为 RXFNE）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，在发生其中任何一个错误时，都会在当前字节后使能中断。

44.5.20 RS232 硬件流控制和 RS485 驱动器使能

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 473* 显示了在这种模式下如何连接 2 个器件：

图 473. 2 个 USART 间的硬件流控制

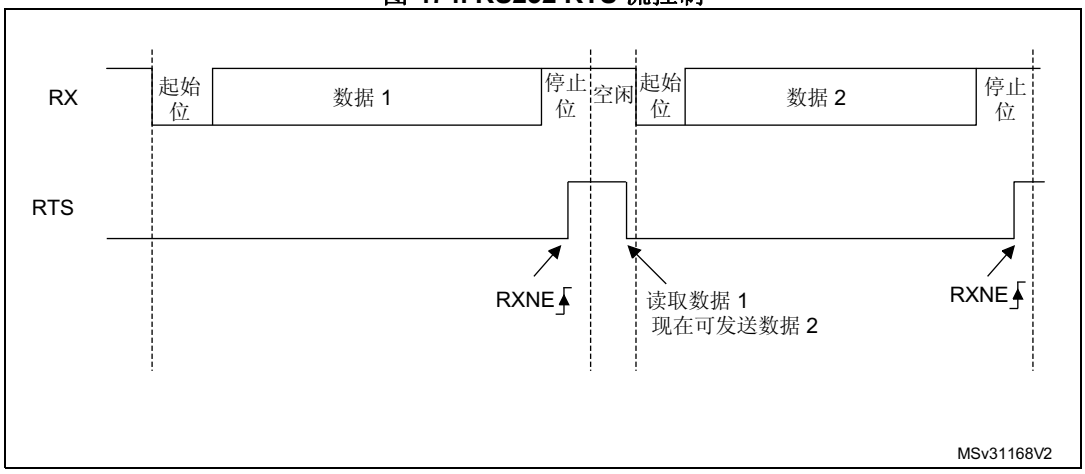


向 USART_CR3 寄存器中的 RTSE 位和 CTSE 位写入“1”，可分别使能 RS232 RTS 和 CTS 流控制。

RS232 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 474* 显示了在使能 RTS 流控制的情况下进行通信的示例。

图 474. RS232 RTS 流控制



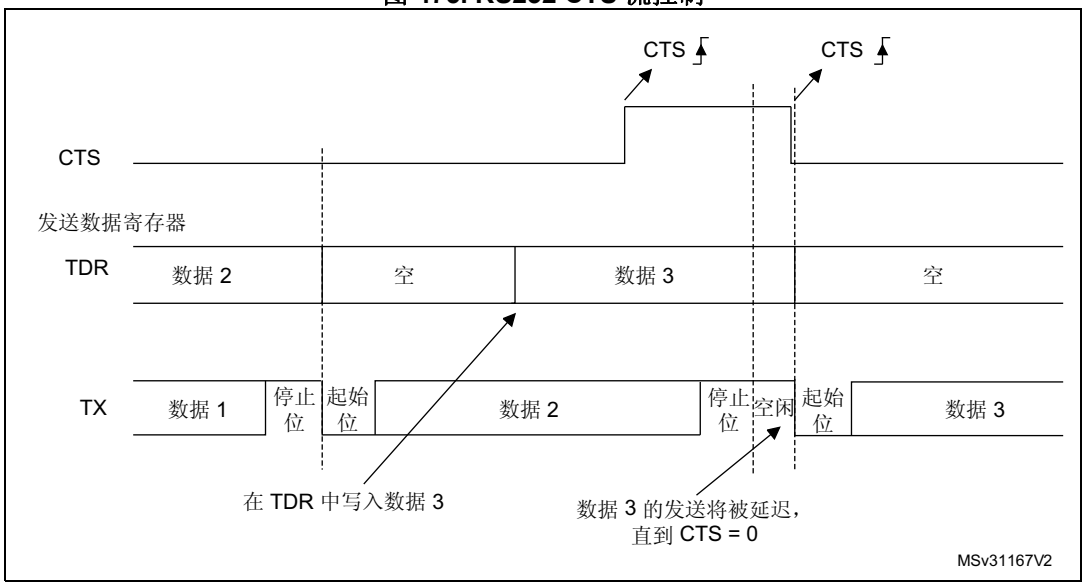
注： *如果使能 FIFO 模式，则仅当 RXFIFO 已满时，nRTS 才会变为无效。*

RS232 CTS 流控制

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/TXFE=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。*图 475* 显示了在使能 CTS 流控制的情况下进行通信的示例。

图 475. RS232 CTS 流控制



注： *为正常运行，必须在当前字符结束前至少 3 个 USART 时钟源周期内使能 nCTS。此外还应注意，当脉冲短于 2 个 PCLK 周期时无法将 CTSCF 标志置 1。*

RS485 驱动器使能

驱动器使能功能可通过将 USART_CR3 控制寄存器中的 DEM 位置 1 使能。这样用户便可通过 DE（驱动器使能）信号激活外部收发器控制。使能时间为激活 DE 信号与 START 位开始间的时间。可以通过 USART_CR1 控制寄存器中的 DEAT [4:0] 位域编程禁止时间。禁止时间为发送的消息中最后一个停止位结束与取消激活 DE 信号间的时间。可以通过 USART_CR1 控制寄存器中的 DEDT [4:0] 位域编程禁止时间。DE 信号的极性可使用 USART_CR3 控制寄存器中的 DEP 位配置。

在 USART 中，DEAT 和 DEDT 以采样时间单位表示（1/8 或 1/16 位时间，具体取决于过采样速率）。

44.5.21 USART 低功耗管理

USART 具有高级低功耗模式功能，即使禁止 usart_pclk 时钟，也能正常传输数据。

UESM 位置 1 时，USART 能够将 MCU 从低功耗模式唤醒。

对 usart_pclk 进行门控时，如果某些特定操作需要激活 **usart_pclk** 时钟，则 USART 会提供唤醒中断 (**usart_wkup**)：

- 禁止 FIFO 模式时

  必须激活 usart_pclk 时钟以清空 USART 数据寄存器。

  在这种情况下，usart_wkup 中断源为 RXNE 置“1”。RXNEIE 位必须在进入低功耗模式之前置 1。

- 使能 FIFO 模式时

  必须激活 usart_pclk 时钟以：

  – 填充 TXFIFO

  – 或清空 RXFIFO

  在这种情况下，usart_wkup 中断源可以为：

  – RXFIFO 非空。此时，RXFNEIE 位必须在进入低功耗模式之前置 1。

  – RxFIFO 已满。此时，RXFFIE 位必须在进入低功耗模式之前置 1，接收到的数据量对应于 RXFIFO 大小，并且 RXFF 标志未置 1。

  – TXFIFO 为空。此时，TXFEIE 位必须在进入低功耗模式之前置 1。

  这允许在低功耗模式下发送/接收 TXFIFO/RXFIFO 中的数据。

  为了避免发生上溢/下溢错误以及在低功耗模式下发送/接收数据，usart_wkup 中断源可以是以下事件之一：

  – 达到 TXFIFO 阈值。此时，TXFTIE 位必须在进入低功耗模式之前置 1。

  – 达到 RXFIFO 阈值。此时，RXFTIE 位必须在进入低功耗模式之前置 1。

  例如，如果唤醒时间短于在线路上接收单个字节所需的时间，则应用可将阈值设为最大 RXFIFO 大小。

  使用 RXFIFO 已满、TXFIFO 为空、RXFIFO 非空和 RXFIFO/TXFIFO 阈值中断将 MCU 从低功耗模式唤醒时，可在低功耗模式下尽可能多地进行 USART 传输，这样有助于优化功耗。

或者，也可通过 WUS 位域选择特定 **usart_wkup** 中断。

检测到唤醒事件后，WUF 标志会由硬件置 1 并在 WUFIE 位置 1 时生成一个 **usart_wkup** 中断。在这种情况下，无需 **usart_wkup** 中断，将 WUF 置 1 便足以将 MCU 从低功耗模式唤醒。

注： *在进入低功耗模式之前，请确保未进行任何 USART 传输。检查 BUSY 标志不能确保在进行数据接收时绝不会进入低功耗模式。*

WUF 标志在检测到唤醒事件时置 1，而与 MCU 处于低功耗模式还是工作模式无关。

如果在初始化和使能接收器后立即进入低功耗模式，则必须检查 REACK 位以确保 USART 确实已使能。

当 DMA 用于接收时，它必须在进入低功耗模式前禁止，并在退出低功耗模式后重新使能。

如果使能 FIFO，则只有在使能静默模式的情况下才能在地址匹配时从低功耗模式唤醒。

使用静默模式和低功耗模式

如果 USART 在进入低功耗模式前处于静默模式：

- 不得使用空闲检测时从静默模式唤醒，因为空闲检测无法在低功耗模式下工作。
- 如果使用地址匹配时从静默模式唤醒，则低功耗模式唤醒源也必须是地址匹配。如果 RXNE 标志在进入低功耗模式时置 1，则接口将在地址匹配时和从低功耗模式唤醒时保持静默模式。

注： *使能 FIFO 管理时，静默模式可与从低功耗模式唤醒搭配使用，而且没有任何限制（即，上述关于静默和低功耗模式的两点仅在 FIFO 管理禁止时有效）。*

USART 内核时钟 (usart_ker_ck) 在低功耗模式下关闭时从低功耗模式唤醒

在低功耗模式下，如果在 USART 接收线上检测到下降沿时 usart_ker_ck 时钟处于关闭状态，则 USART 接口会借助 usart_ker_ck_req 信号请求开启 usart_ker_ck 时钟。usart_ker_ck 随后用来接收帧。

如果唤醒事件通过验证，则 MCU 将从低功耗模式唤醒，并会正常进行数据接收。

如果唤醒事件未通过验证，则 usart_ker_ck 会再次关闭，MCU 不会被唤醒并保持在低功耗模式下，且内核时钟请求被释放。

以下示例显示了将唤醒事件编程为“地址匹配检测”并禁止 FIFO 管理的情况。

图 476 所示为唤醒事件通过验证时的 USART 行为。

图 476. 唤醒事件通过验证（唤醒事件 = 地址匹配，禁止 FIFO）

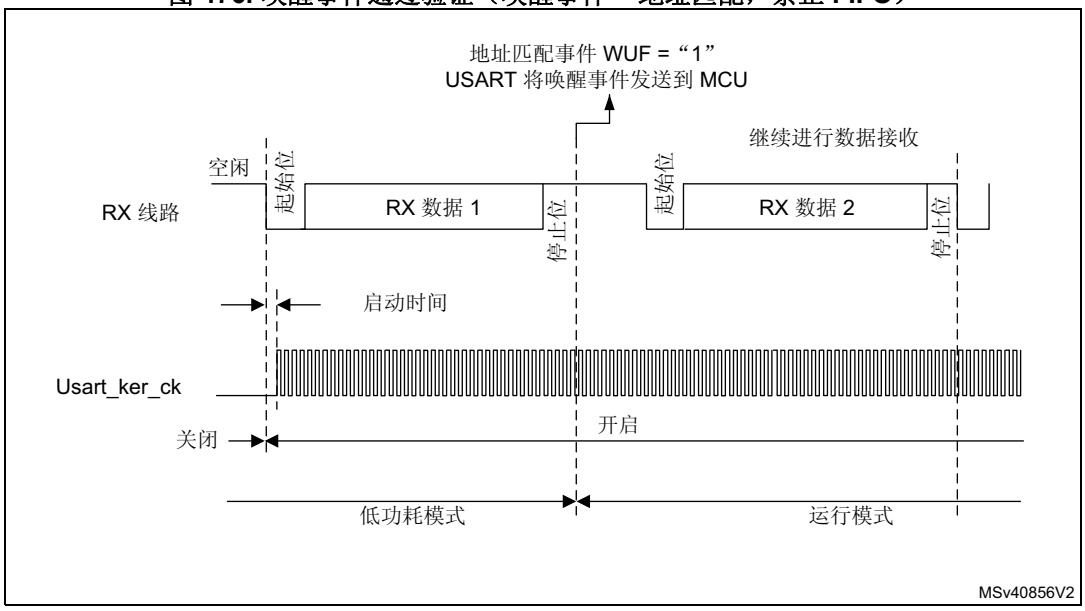
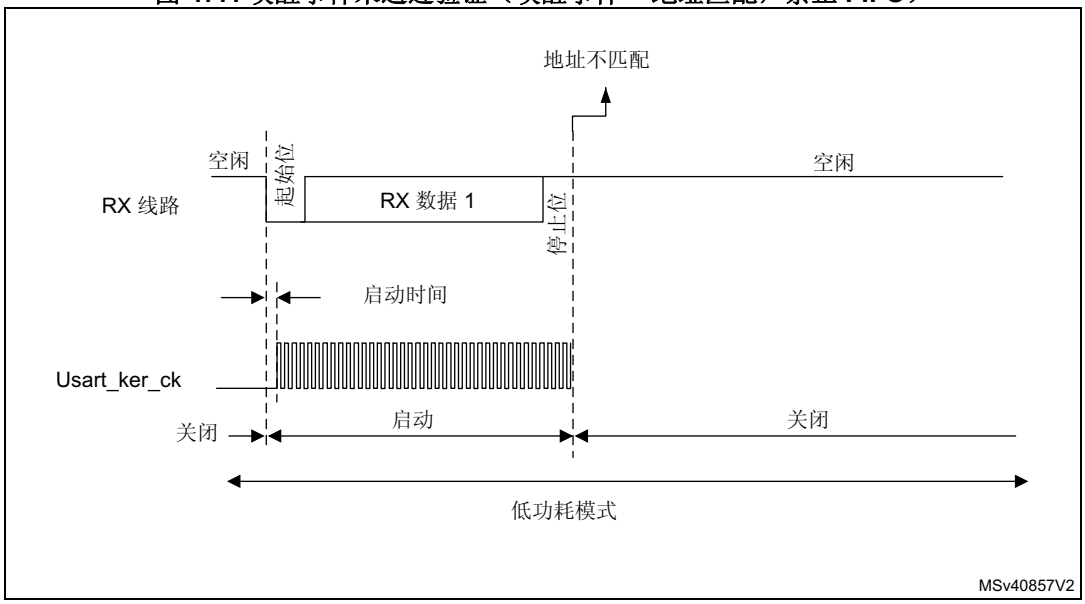


图 477 所示为唤醒事件未通过验证时的 USART 行为。

图 477. 唤醒事件未通过验证（唤醒事件 = 地址匹配，禁止 FIFO）



注： *将地址匹配或任一接收的帧用作唤醒事件时，如上两图适用。如果唤醒事件为起始位检测，则 USART 会在起始位结束时向 MCU 发送唤醒事件。*

确定允许从低功耗模式正确唤醒微控制器的最大 USART 波特率

允许从低功耗模式正确唤醒微控制器的最大波特率取决于唤醒时间参数（请参见器件数据手册）和 USART 接收器容差（请参见*第 44.5.8 节：USART 接收器对时钟偏差的容差*）。

举例来说：OVER8 = 0，M 位 = “01”，ONEBIT = 0，BRR [3:0] = 0000。

在这些条件下，根据*表 264：BRR [3:0] = 0000 时的 USART 接收器容差*，USART 接收器的容差为 3.41%。

DTRA + DQUANT + DREC + DTCL + DWU < USART 接收器的容差

$D_{WUmax} = t_{WUUSART} / (11 \times T_{bit\ Min})$

$T_{bit\ Min} = = t_{WUUSART} / (11 \times D_{WUmax})$

其中 $t_{WUUSART}$ 为从低功耗模式唤醒的时间。

考虑一种理想情况：参数 DTRA、DQUANT、DREC 和 DTCL 为 0%，则 DWU 的最大值为 3.41%。实际上，我们至少需要考虑 usart_ker_ck 不精确的情况。

例如，如果将 HSI 用作 usart_ker_ck，HSI 的不精确度为 1%，则可以得到：

$t_{WUUSART}$ = 3 µs（仅提供数值作为参考；有关正确数值，请参见器件数据手册）。

D_{WUmax} = 3.41% - 1% = 2.41%

$T_{bit\ min}$ = 3 µs/ (11 x 2.41%) = 11.32 µs

结果，允许从低功耗模式正确唤醒的最大波特率为：1/11.32 µs = 88.36 K 波特。

44.6 USART 中断

在 USART 通信过程中，中断 (usart_it) 可由不同事件生成。USART 模块也可生成唤醒中断 (usart_wkup)。

有关所有 USART 中断请求的详细说明，请参见*表 267*。

表 267. USART 中断请求

中断事件	事件标志	使能控制位	中断清除方法	中断已激活	
				usart_it	usart_wkup
发送数据寄存器为空	TXE	TXEIE	TXE 在 TDR 中被写入数据时清零。	是	否
发送 FIFO 未满	TXFNF	TXFNFIE	TXFNF 在 TXFIFO 已满时清零。	是	否
发送 FIFO 为空	TXFE	TXFEIE	TXFE 在 TXFIFO 包含至少一个数据时清零，或通过将 TXFRQ 位置 1 来清零。	是	是
达到发送 FIFO 阈值	TXFT	TXFTIE	TXFT 在 TXFIFO 内容少于编程的阈值时由硬件清零。	是	是
CTS 中断	CTSIF	CTSIE	CTSIF 由软件通过将 CTSCF 位置 1 来清零。	是	否
发送完成	TC	TCIE	TC 在 TDR 中被写入数据时清零，或通过将 TCCF 位置 1 来清零。	是	否
保护时间前发送完成	TCBGT	TCBGTIE	TCBGT 在 TDR 中被写入数据时清零，或通过将 TCBGTCF 位置 1 来清零。	是	否
接收数据寄存器不为空（已准备好读取数据）	RXNE	RXNEIE	RXNE 通过读取 RDR 或通过将 RXFRQ 位置 1 来清零。	是	是
接收 FIFO 非空	RXFNE	RXFNEIE	RXFNE 在 RXFIFO 为空时清零，或通过将 RXFRQ 位置 1 来清零。	是	是
接收 FIFO 已满	RXFF(1)	RXFFIE	RXFF 在 RXFIFO 至少包含一个数据时清零。	是	是
达到接收 FIFO 阈值	RXFT	RXFTIE	RXFT 在 RXFIFO 内容少于编程的阈值时由硬件清零。	是	是
检测到溢出错误	ORE	RXNEIE/ RXFNEIE	ORE 通过将 ORECF 位置 1 来清零。	是	否
检测到空闲线路	IDLE	IDLEIE	IDLE 通过将 IDLECF 位置 1 来清零。	是	否
奇偶校验错误	PE	PEIE	PE 通过将 PECF 位置 1 来清零。	是	否

表 267. USART 中断请求（续）

中断事件	事件标志	使能控制位	中断清除方法	中断已激活	
				usart_it	usart_wkup
LIN 断路	LBDF	LBDIE	LBDF 通过将 LBDCF 位置 1 来清零。	是	否
多缓冲区通信中的噪声标志、溢出错误和帧错误。	NE 或 ORE 或 FE	EIE	NE 通过将 NFCF 位置 1 来清零。 ORE 通过将 ORECF 位置 1 来清零。 FE 标志通过将 FECF 位置 1 来清零。	是	否
字符匹配	CMF	CMIE	CMF 通过将 CMCF 位置 1 来清零。	是	否
接收器超时	RTOF	RTOFIE	RTOF 通过将 RTOCCF 位置 1 来清零。	是	否
块结束	EOBF	EOBIE	EOBF 通过将 EOBCF 位置 1 来清零。	是	否
从低功耗模式唤醒	WUF	WUFIE	WUF 通过将 WUCF 位置 1 来清零。	是	是
SPI 从器件下溢错误	UDR	EIE	UDR 通过将 UDRCF 位置 1 来清零。	是	否
达到发送 FIFO 阈值	TXFT	TXFTIE	TXFT 在 TXFIFO 内容少于编程的阈值时由硬件清零。	是	是
达到接收 FIFO 阈值	RXFT	RXFTIE	RXFT 在 RXFIFO 内容少于编程的阈值时由硬件清零。	是	是

1. 如果 USART 接收 n+1 个数据（n 表示 RXFIFO 大小，RXFIFO 中有 n 个数据，USART_RDR 中有 1 个数据），则 RXFF 标志置 1。在停止模式下，未向 USART_RDR 提供时钟。因此，将不会对该寄存器进行写操作，一旦有 n 个数据被接收并写入到 RXFIFO，RXFF 中断将生效（RXFF 标志未置 1）。

44.7 USART 寄存器

有关寄存器说明中使用的缩写，请参见*第 72 页的第 1.1 节*。

44.7.1 USART 控制寄存器 1 (USART_CR1)

USART control register 1

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
RXFFIE	TXFEIE	FIFOEN	M1	EOBIE	RTOIE	DEAT[4:0]					DEDT[4:0]				
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
OVER8	CMIE	MME	M0	WAKE	PCE	PS	PEIE	TXEIE/TXFNFIE	TCIE	RXNEIE/RXFNEIE	IDLEIE	TE	RE	UESM	UE
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31 **RXFFIE**：RXFIFO 变满时中断使能 (RXFIFO Full interrupt enable)

此位由软件置 1 和清零。

0：禁止中断

1：当 USART_ISR 寄存器中的 RXFF=1 时，生成 USART 中断

注： 禁止 FIFO 模式时，该位保留且必须保持复位值。

位 30 **TXFEIE**：TXFIFO 为空时中断使能 (TXFIFO empty interrupt enable)

此位由软件置 1 和清零。

0：禁止中断

1：当 USART_ISR 寄存器中的 TXFE=1 时，生成 USART 中断

注： 禁止 FIFO 模式时，该位保留且必须保持复位值。

位 29 **FIFOEN**：FIFO 模式使能 (FIFO mode enable)

此位由软件置 1 和清零。

0：禁止 FIFO 模式。

1：使能 FIFO 模式。

只有在禁止 USART (UE=0) 时才能写入此位域。

注： FIFO 模式只能在标准 UART 通信、SPI 主/从器件模式和智能卡模式下使用，不得在 IrDA 和 LIN 模式下使能。

位 28 **M1**：字长 (Word length)

此位必须与位 12 (M0) 搭配使用来确定字长度。该位由软件置 1 或清零。

M[1:0] = “00”：1 个起始位，8 个数据位，n 个停止位

M[1:0] = “01”：1 个起始位，9 个数据位，n 个停止位

M[1:0] = “10”：1 个起始位，7 个数据位，n 个停止位

只有在禁止 USART (UE=0) 时才能写入此位。

注： 在 7 位数据长度模式下，不支持智能卡模式、LIN 主模式和自动波特率（0x7F 帧和 0x55 帧检测）。

位 27 **EOBIE**：块结束中断使能 (End of Block interrupt enable)

此位由软件置 1 和清零。

0：禁止中断

1：USART_ISR 寄存器中的 EOBF 标志置 1 时生成 USART 中断

注：　如果 USART 不支持智能卡模式，该位保留且必须保持复位值。请参见第 1455 页的第 44.4 节：USART 实现。

位 26 **RTOIE**：接收器超时中断使能 (Receiver timeout interrupt enable)

此位由软件置 1 和清零。

0：禁止中断

1：USART_ISR 寄存器中的 RTOF 位置 1 时生成 USART 中断

注：　如果 USART 不支持接收器超时功能，该位保留且必须保持复位值。第 1455 页的第 44.4 节：USART 实现.

位 25:21 **DEAT[4:0]**：驱动器使能使能时间 (Driver Enable assertion time)

该 5 位值用于定义激活 DE（启动器使能）信号与起始位开始间的时间。此时间以采样时间单位表示（1/8 或 1/16 位时间，具体取决于过采样速率）。

只有在禁止 USART (UE=0) 时才能写入此位域。

注：　如果不支持驱动器使能功能，该位保留且必须保持复位值。请参见第 1455 页的第 44.4 节：USART 实现。

位 20:16 **DEDT[4:0]**：驱动器使能禁止时间 (Driver Enable deassertion time)

该 5 位值用于定义发送的消息中最后一个停止位结束与取消激活 DE（驱动器使能）信号间的时间。此时间以采样时间单位表示（1/8 或 1/16 位时间，具体取决于过采样速率）。

如果在 DEDT 时间内对 USART_TDR 寄存器执行写操作，则新数据仅在经过 DEDT 和 DEAT 时间后才会发送。

只有在禁止 USART (UE=0) 时才能写入此位域。

注：　如果不支持驱动器使能功能，该位保留且必须保持复位值。请参见第 1455 页的第 44.4 节：USART 实现。

位 15 **OVER8**：过采样模式 (Oversampling mode)

0：16 倍过采样

1：8 倍过采样

只有在禁止 USART (UE=0) 时才能写入此位。

注：　在 LIN、IrDA 和智能卡模式下，此位必须保持清零。

位 14 **CMIE**：字符匹配中断使能 (Character match interrupt enable)

此位由软件置 1 和清零。

0：禁止中断。

1：USART_ISR 寄存器中的 CMF 位置 1 时生成 USART 中断。

位 13 **MME**：静默模式使能 (Mute mode enable)

此位可使能 USART 静默模式功能。此位置 1 时，USART 会按照 WAKE 位的定义在工作模式与静默模式之间切换。该位由软件置 1 和清零。

0：接收器永久处于活动模式。

1：接收器可在静默模式和活动模式之间切换。

位 12 **M0**：字长 (Word length)

此位与位 28 (M1) 搭配使用来确定字长度。它由软件置 1 或清零（请参见位 28 (M1) 的说明）。

只有在禁止 USART (UE=0) 时才能写入此位。

位 11 **WAKE**：接收器唤醒方法 (Receiver wakeup method)

此位用于确定 USART 静默模式的唤醒方法。该位由软件置 1 或清零。

0：空闲线路

1：地址标记

只有在禁止 USART (UE=0) 时才能写入此位域。

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

只有在禁止 USART (UE=0) 时才能写入此位域。

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

只有在禁止 USART (UE=0) 时才能写入此位域。

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_ISR 寄存器中的 PE=1 时，生成 USART 中断

位 7 **TXEIE/TXFNFIE**：发送数据寄存器为空/TXFIFO 未满中断使能 (Transmit data register empty/ TXFIFO not full interrupt enable)

此位由软件置 1 和清零。

0：禁止中断

1：当 USART_ISR 寄存器中的 TXE/TXFNF=1 时，生成 USART 中断

位 6 **TCIE**：传输完成中断使能 (Transfer complete interrupt enable)

此位由软件置 1 和清零。

0：禁止中断

1：当 USART_ISR 寄存器中的 TC=1 时，生成 USART 中断

位 5 **RXNEIE/RXFNEIE**：接收数据寄存器非空/RXFIFO 非空中断使能 (Receive data register not empty/RXFIFO not empty interrupt enable)

此位由软件置 1 和清零。

0：禁止中断

1：当 USART_ISR 寄存器中的 ORE=1 或 RXNE/RXFNE=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_ISR 寄存器中的 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： *除了在智能卡模式下以外，传送期间 TE 位上的低电平脉冲（“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）。为生成空闲字符，TE 不能立即写入“1”。为确保所需的持续时间，软件可轮询 USART_ISR 寄存器中的 TEACK 位。*

在智能卡模式下，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 **UESM**：低功耗模式下的 USART 使能 (USART enable in low-power mode)

当此位清零时，USART 无法将 MCU 从低功耗模式唤醒。

当此位置 1 时，USART 可将 MCU 从低功耗模式唤醒。

此位由软件置 1 和清零。

0：USART 无法将 MCU 从低功耗模式唤醒。

1：USART 能够将 MCU 从低功耗模式唤醒。

注： *建议在进入低功耗模式前将 UESM 位置 1，并在退出低功耗模式时将其清零。*

如果 USART 不支持从停止模式唤醒功能，该位保留且必须保持复位值。请参见第 1455 页的第 44.4 节：USART 实现。

位 0 **UE**：USART 使能 (USART enable)

此位清零后，USART 预分频器和输出将立即停止，并丢弃所有当前操作。USART 配置会保留，而所有的 USART_ISR 状态标志均会复位。此位由软件置 1 和清零。

0：禁止 USART 预分频器和输出，低功耗模式

1：使能 USART

注： *为进入低功耗模式而不在线路上生成错误，之前必须复位 TE 位，并且软件必须等待 USART_ISR 中的 TC 位置 1 后才能复位 UE 位。*

UE = 0 时也会复位 DMA 请求，因此必须在复位 UE 位前禁止 DMA 通道。

在智能卡模式下 (SCEN = 1)，无论 UE 位值为何，当 CLKEN = 1 时，SCLK 始终可用。

44.7.2 USART 控制寄存器 2 (USART_CR2)

USART control register 2

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
ADD[7:4]				ADD[3:0]				RTOEN	ABRMOD[1:0]		ABREN	MSBFIRST	DATAINV	TXINV	RXINV
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
SWAP	LINEN	STOP[1:0]		CLKEN	CPOL	CPHA	LBCL	Res.	LBDIE	LBDL	ADDM7	DIS_NSS	Res.	Res.	SLVEN
rw	rw	rw	rw	rw	rw	rw	rw		rw	rw	rw	rw			rw

位 31:28 **ADD[7:4]**：USART 节点的地址 (Address of the USART node)

此位域用于指定 USART 节点的地址或要识别的字符代码。

它在多处理器通信时于静默模式或低功耗模式下使用，以通过 7 位地址标记检测唤醒 MCU。发送器发送字符的 MSB 应为 1。此位域还可用于正常接收和静默模式无效时的字符检测（例如，ModBus 协议中的块结束检测）。这种情况下，接收到的整个字符（8 位）将与 ADD[7:0] 值进行比较，如果匹配，CMF 标志将置 1。

仅在禁止接收 (RE = 0) 或禁止 USART (UE=0) 时才能写入该位域。

位 27:24 **ADD[3:0]**：USART 节点的地址 (Address of the USART node)

此位域用于指定 USART 节点的地址或要识别的字符代码。

它在多处理器通信时于静默模式或低功耗模式下使用，以通过地址标记检测进行唤醒。

仅在禁止接收 (RE = 0) 或禁止 USART (UE=0) 时才能写入该位域。

位 23 **RTOEN**：接收器超时使能 (Receiver timeout enable)

此位由软件置 1 和清零。

0：禁止接收器超时功能。

1：使能接收器超时功能。

使能此功能后，如果 RX 线路在 RTOR（接收器超时寄存器）中编程的持续时间内处于空闲状态（无接收），则 USART_ISR 寄存器中的 RTOF 标志置 1。

注： 如果 USART 不支持接收器超时功能，该位保留且必须保持复位值。请参见第 1455 页的第 44.4 节：USART 实现。

位 22:21 **ABRMOD[1:0]**：自动波特率模式 (Auto baud rate mode)

这些位将由软件置 1 和清零。

00：通过测量起始位检测波特率。

01：下降沿到下降沿的测量（接收到的帧必须以一个等于 1 的位开头，即帧 = 10xxxxxx）。

10：0x7F 帧检测。

11：0x55 帧检测。

仅在 ABREN = 0 时或禁止 USART (UE=0) 时才能写入该位域。

注： 如果 DATAINV=1 且/或 MSBFIRST=1，这些模式必须与在线路上时相同，例如 MSBFIRST 的 0xAA。

如果 USART 不支持自动波特率功能，该位保留且必须保持复位值。请参见第 1455 页的第 44.4 节：USART 实现。

位 20 **ABREN**：自动波特率使能 (Auto baud rate enable)

此位由软件置 1 和清零。

0：禁止自动波特率检测。

1：使能自动波特率检测。

注：　如果 USART 不支持自动波特率功能，该位保留且必须保持复位值。请参见第 1455 页的第 44.4 节：USART 实现。

位 19 **MSBFIRST**：最高有效位在前 (Most significant bit first)

此位由软件置 1 和清零。

0：发送/接收数据时位 0 在前，跟在起始位后。

1：发送/接收数据时 MSB（位 7/8）在前，跟在起始位后。

只有在禁止 USART (UE=0) 时才能写入此位域。

位 18 **DATAINV**：二进制数据反向 (Binary data inversion)

此位由软件置 1 和清零。

0：按正/正向逻辑发送/接收数据寄存器中的逻辑数据。（1=H，0=L）

1：按负/反向逻辑发送/接收数据寄存器中的逻辑数据。（1=L，0=H）。奇偶校验位也取反。

只有在禁止 USART (UE=0) 时才能写入此位域。

位 17 **TXINV**：TX 引脚有效电平反向 (TX pin active level inversion)

此位由软件置 1 和清零。

0：TX 引脚信号使用标准逻辑电平（V_{DD} = 1/空闲，Gnd = 0/标记）工作。

1：对 TX 引脚信号值取反。（V_{DD} = 0/标记，Gnd=1/空闲）。

允许在 TX 线路上使用外部反相器。

只有在禁止 USART (UE=0) 时才能写入此位域。

位 16 **RXINV**：RX 引脚有效电平反向 (RX pin active level inversion)

此位由软件置 1 和清零。

0：RX 引脚信号使用标准逻辑电平（V_{DD} = 1/空闲，Gnd = 0/标记）工作。

1：对 RX 引脚信号值取反。（V_{DD} = 0/标记，Gnd=1/空闲）。

允许在 RX 线路上使用外部反相器。

只有在禁止 USART (UE=0) 时才能写入此位域。

位 15 **SWAP**：交换 TX/RX 引脚 (Swap TX/RX pins)

此位由软件置 1 和清零。

0：按标准引脚排列定义使用 TX/RX 引脚。

1：交换 TX 和 RX 引脚功能。允许在与另一个 UART 的交叉连接时工作。

只有在禁止 USART (UE=0) 时才能写入此位域。

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 SBKRQ 位发送 LIN 同步断路（13 个低位），并可检测 LIN 同步断路。

只有在禁止 USART (UE=0) 时才能写入此位域。

注：　如果 USART 不支持 LIN 模式，该位保留且必须保持复位值。请参见第 1455 页的第 44.4 节：USART 实现。

位 13:12 **STOP[1:0]**：停止位 (STOP bits)

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

只有在禁止 USART (UE=0) 时才能写入此位域。

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

只有在禁止 USART (UE=0) 时才能写入此位。

注：　如果既不支持同步模式，也不支持智能卡模式，则该位保留且必须保持复位值。请参见第 1455 页的第 44.4 节：USART 实现。

在智能卡模式下，为了向智能卡正确提供 SCLK 时钟，必须按以下步骤操作：

UE = 0

SCEN = 1

GTPR 配置

CLKEN= 1

UE = 1

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

只有在禁止 USART (UE=0) 时才能写入此位。

注：　如果不支持同步模式，该位保留且必须保持复位值。请参见第 1455 页的第 44.4 节：USART 实现。

位 9 **CPHA**：时钟相位 (Clock phase)

此位用于在同步模式下选择 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 457* 和*图 458*）。

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

只有在禁止 USART (UE=0) 时才能写入此位。

注：　如果不支持同步模式，该位保留且必须保持复位值。请参见第 1455 页的第 44.4 节：USART 实现。

位 8 **LBCL**：最后一个位时钟脉冲 (Last bit clock pulse)

此位用于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：　最后一位为发送的第 7 个、第 8 个或第 9 个数据位，具体取决于 USART_CR1 寄存器中 M 位所选择的 7 位、8 位或 9 位格式。

只有在禁止 USART (UE=0) 时才能写入此位。

注：　如果不支持同步模式，该位保留且必须保持复位值。请参见第 1455 页的第 44.4 节：USART 实现。

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（使用中断分隔符进行中断检测）

0：禁止中断

1：当 USART_ISR 寄存器中 LBDF = 1 时，生成中断

注：　如果不支持 LIN 模式，该位保留且必须保持复位值。请参见第 1455 页的第 44.4 节：USART 实现。

位 5 **LBDL**：LIN 断路检测长度 (LIN break detection length)

该位用于选择 11 位中断检测或 10 位中断检测。

0：10 位中断检测

1：11 位中断检测

只有在禁止 USART (UE=0) 时才能写入此位。

注：　如果不支持 LIN 模式，该位保留且必须保持复位值。请参见第 1455 页的第 44.4 节：USART 实现。

位 4 ADDM7：7 位地址检测/4 位地址检测 (7-bit Address Detection/4-bit Address Detection)

此位用于选择 4 位地址检测或 7 位地址检测。

0：4 位地址检测

1：7 位地址检测（在 8 位数据模式下）

只有在禁止 USART (UE=0) 时才能写入该位。

注：　在 7 位和 9 位数据模式下，地址检测分别在 6 位和 8 位地址上完成（ADD[5:0] 和 ADD[7:0]）。

位 3 **DIS_NSS**：

当 DSI_NSS 位置 1 时，忽略 NSS 引脚输入。

0：SPI 从器件选择取决于 NSS 输入引脚。

1：始终选择 SPI 从器件，忽略 NSS 输入引脚。

注：　不支持 SPI 从器件模式时，该位保留且必须保持复位值。请参见第 1455 页的第 44.4 节：USART 实现。

位 2:1 保留，必须保持复位值。

位 0 **SLVEN**：同步从模式使能 (Synchronous Slave mode enable)

SLVEN 位置 1 时，使能同步从模式。

0：禁止从模式。

1：使能从模式。

注：　不支持 SPI 从器件模式时，该位保留且必须保持复位值。请参见第 1455 页的第 44.4 节：USART 实现。

注：　使能发送器时不应对 CPOL、CPHA 和 LBCL 位进行写操作。

44.7.3 USART 控制寄存器 3 (USART_CR3)

USART control register 3

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
TXFTCFG[2:0]			RXF TIE	RXFTCFG[2:0]			TCBG TIE	TXFTIE	WUFIE	WUS[1:0]		SCARCNT[2:0]			Res.
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
DEP	DEM	DDRE	OVR DIS	ONE BIT	CTSIE	CTSE	RTSE	DMAT	DMAR	SCEN	NACK	HD SEL	IRLP	IREN	EIE
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:29 TXFTCFG[2:0]：TXFIFO 阈值配置 (TXFIFO threshold configuration)

000：TXFIFO 达到其深度的 1/8
001：TXFIFO 达到其深度的 1/4
010：TXFIFO 达到其深度的 1/2
011：TXFIFO 达到其深度的 3/4
100：TXFIFO 达到其深度的 7/8
101：TXFIFO 变空
其余组合：保留

位 28 **RXFTIE**：RXFIFO 阈值中断使能 (RXFIFO threshold interrupt enable)

此位由软件置 1 和清零。
0：禁止中断
1：当接收 FIFO 达到 RXFTCFG 中编程的阈值时，生成 USART 中断

位 27:25 **RXFTCFG[2:0]**：接收 FIFO 阈值配置 (Receive FIFO threshold configuration)

000：接收 FIFO 达到其深度的 1/8
001：接收 FIFO 达到其深度的 1/4
010：接收 FIFO 达到其深度的 1/2
011：接收 FIFO 达到其深度的 3/4
100：接收 FIFO 达到其深度的 7/8
101：接收 FIFO 已满
其余组合：保留

位 24 **TCBGTIE**：保护时间前发送完成中断使能 (Transmission Complete before guard time, interrupt enable)

此位由软件置 1 和清零。
0：禁止中断
1：当 USART_ISR 寄存器中的 TCBGT=1 时，生成 USART 中断

注：　如果 USART 不支持智能卡模式，该位保留且必须保持复位值。
请参见第 1455 页的第 44.4 节：USART 实现。

位 23 **TXFTIE**：TXFIFO 阈值中断使能 (TXFIFO threshold interrupt enable)

此位由软件置 1 和清零。
0：禁止中断
1：当 TXFIFO 达到 TXFTCFG 中编程的阈值时，生成 USART 中断

位 22 **WUFIE**：从低功耗模式唤醒中断使能 (Wakeup from low-power mode interrupt enable)

此位由软件置 1 和清零。

0：禁止中断

1：当 USART_ISR 寄存器中的 WUF=1 时，生成 USART 中断

注： *WUFIE 必须在进入低功耗模式前置 1。*

如果 USART 不支持从停止模式唤醒功能，该位保留且必须保持复位值。请参见第 1455 页的第 44.4 节：USART 实现。

位 21:20 **WUS[1:0]**：从低功耗模式唤醒中断标志选择 (Wakeup from low-power mode interrupt flag selection)

该位域用于指定激活 WUF（从低功耗模式唤醒标志）的事件。

00：WUF 在地址匹配时激活（按 ADD[7:0] 和 ADDM7 所定义）

01：保留

10：WUF 在起始位检测时激活

11：WUF 在 RXNE/RXFNE 时激活

只有在禁止 USART (UE=0) 时才能写入此位域。

如果 USART 不支持从停止模式唤醒功能，该位保留且必须保持复位值。请参见第 1455 页的第 44.4 节：USART 实现。

位 19:17 **SCARCNT[2:0]**：智能卡自动重试计数 (Smartcard auto-retry count)

此位域用于指定智能卡模式下发送和接收的重试次数。

在发送模式下，此位域用于指定生成发送错误（FE 位置 1）前自动重新发送的重试次数。

在接收模式下，此位域用于指定生成接收错误（RXNE/RXFNE 位和 PE 位置 1）前错误接收尝试的次数。

只有在禁止 USART (UE=0) 时才能编程此位域。

使能 USART (UE=1) 时，此位域只能写入 0x0，以停止重新发送。

0x0：禁止重新发送——发送模式下不会自动重新发送。

0x1 到 0x7：自动重新发送的尝试次数（发出错误信号前）

注： *如果不支持智能卡模式，该位保留且必须保持复位值。请参见第 1455 页的第 44.4 节：USART 实现。*

位 16 保留，必须保持复位值。

位 15 **DEP**：驱动器使能极性选择 (Driver enable polarity selection)

0：DE 信号高电平有效。

1：DE 信号低电平有效。

只有在禁止 USART (UE=0) 时才能写入此位。

注： *如果不支持驱动器使能功能，该位保留且必须保持复位值。请参见第 1455 页的第 44.4 节：USART 实现。*

位 14 **DEM**：驱动器使能模式 (Driver enable mode)

此位用于通过 DE 信号激活外部收发器控制。

0：禁止 DE 功能。

1：使能 DE 功能。DE 信号在 RTS 引脚上输出。

只有在禁止 USART (UE=0) 时才能写入此位。

注： *如果不支持驱动器使能功能，该位保留且必须保持复位值。第 1455 页的第 44.4 节：USART 实现.*

位 13 **DDRE**：接收出错时的 DMA 禁止 (DMA Disable on Reception Error)

0：接收出错时不禁止 DMA。相应的错误标志置 1，但 RXNE 保持为 0 以防止上溢。因此，将不使能 DMA 请求，从而不会传送错误数据（无 DMA 请求），但会传送接收到的下一个正确数据。（用于智能卡模式）

1：接收出错后禁止 DMA。相应的错误标志以及 RXNE 均置 1。屏蔽 DMA 请求，直到错误标志清零。这意味着软件必须首先禁止 DMA 请求 (DMAR = 0) 或者将 RXNE（使能 FIFO 模式时为 RXFNE）清零，然后再将错误标志清零。

只有在禁止 USART (UE=0) 时才能写入此位。

注： 接收错误包括：奇偶校验错误、帧错误或噪声错误。

位 12 OVRDIS：上溢禁止 (Overrun Disable)

此位用于禁止接收上溢检测。

0：接收新数据前未读取已接收的数据时，上溢错误标志 ORE 置 1。

1：禁止上溢功能。如果在 RXNE 标志仍置 1 时接收到新数据。

则 ORE 标志不会置 1，且新接收的数据会覆盖 USART_RDR 寄存器之前的内容。使能 FIFO 模式时，RXFIFO 将被旁路，数据将直接写入 USART_RDR 寄存器中。即使在使能 FIFO 管理时，也将使用 RXNE 标志。

只有在禁止 USART (UE=0) 时才能写入此位。

注： 此控制位用于检查通信流而不会读取数据

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 (NE)。

0：三个采样位方法

1：一个采样位方法

只有在禁止 USART (UE=0) 时才能写入此位。

位 10 **CTSIE**：CTS 中断使能 (CTS interrupt enable)

0：禁止中断

1：当 USART_ISR 寄存器中 CTSIF = 1 时，生成中断

注： 如果不支持硬件流控制功能，该位保留且必须保持复位值。请参见第 1455 页的第 44.4 节：USART 实现。

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。

只有在禁止 USART (UE=0) 时才能写入该位。

注： 如果不支持硬件流控制功能，该位保留且必须保持复位值。请参见第 1455 页的第 44.4 节：USART 实现。

位 8 **RTSE**：RTS 使能 (RTS enable)

0：禁止 RTS 硬件流控制

1：使能 RTS 输出，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（拉至 0）。

只有在禁止 USART (UE=0) 时才能写入此位。

注： 如果不支持硬件流控制功能，该位保留且必须保持复位值。请参见第 1455 页的第 44.4 节：USART 实现。

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

只有在禁止 USART (UE=0) 时才能写入此位域。

注：　如果 USART 不支持智能卡模式，该位保留且必须保持复位值。请参见第 1455 页的第 44.4 节：USART 实现。

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

只有在禁止 USART (UE=0) 时才能写入此位域。

注：　如果 USART 不支持智能卡模式，该位保留且必须保持复位值。请参见第 1455 页的第 44.4 节：USART 实现。

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

只有在禁止 USART (UE=0) 时才能写入此位。

位 2 **IRLP**：IrDA 低功耗模式 (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

只有在禁止 USART (UE=0) 时才能写入此位。

注：　如果不支持 IrDA 模式，该位保留且必须保持复位值。请参见第 1455 页的第 44.4 节：USART 实现。

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

只有在禁止 USART (UE=0) 时才能写入此位。

注：　如果不支持 IrDA 模式，该位保留且必须保持复位值。请参见第 1455 页的第 44.4 节：USART 实现。

位 0 **EIE**：错误中断使能 (Error interrupt enable)

如果出现帧错误、上溢错误、噪声标志或 SPI 从器件下溢错误（USART_ISR 寄存器中的 FE=1、ORE=1、NE=1 或 UDR = 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：USART_ISR 寄存器中的 FE=1、ORE=1、NE=1 或 UDR = 1（在 SPI 从器件模式下）时，生成中断。

44.7.4 USART 波特率寄存器 (USART_BRR)

USART baud rate register

只有在禁止 USART (UE=0) 时才能写入此寄存器。在自动波特率检测模式下，该位由硬件自动更新。

偏移地址：0x0C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	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
BRR[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 15:4 **BRR[15:4]**

BRR[15:4] = USARTDIV[15:4]

位 3:0 **BRR[3:0]**

当 OVER8 = 0 时，BRR[3:0] = USARTDIV[3:0]。
当 OVER8 = 1 时：
BRR[2:0] = USARTDIV[3:0]，右移 1 位。
BRR[3] 必须保持清零。

44.7.5 USART 保护时间和预分频器寄存器 (USART_GTPR)

USART guard time and prescaler 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
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
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 time value)

此位域用于编程保护时间值（以波特时钟周期数为单位）。

该位域用于智能卡模式。经过此保护时间后，发送完成标志置 1。

只有在禁止 USART (UE=0) 时才能写入此位域。

注： 如果不支持智能卡模式，该位保留且必须保持复位值。请参见第 1455 页的第 44.4 节：USART 实现。

位 7:0 **PSC[7:0]**：预分频器值 (Prescaler value)

在 IrDA 低功耗和正常的 IrDA 模式下：

PSC[7:0] = IrDA 正常和低功耗波特率

用于编程预分频器，进行 USART 源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

00000001：源时钟 1 分频

00000010：源时钟 2 分频

...

在智能卡模式下：

PSC[4:0]：预分频器值 (Prescaler value)

用于编程预分频器，进行 USART 源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

...

只有在禁止 USART (UE=0) 时才能写入此位域。

注： 如果使用智能卡模式，则位 [7:5] 必须保持清零。

不支持智能卡和 IrDA 模式时，该位域保留并由硬件强制清零。请参见第 1455 页的第 44.4 节：USART 实现。

44.7.6 USART 接收器超时寄存器 (USART_RTOR)

USART receiver timeout 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
BLEN[7:0]								RTO[23:16]							
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
RTO[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:24 **BLEN[7:0]**：块长度 (Block Length)

此位域用于提供智能卡 T=1 接收下的块长度。其值等于信息字符数 + 结尾字段的长度 (1-LEC/2-CRC) - 1。

例如：

BLEN = 0 -> 0 个信息字符 + LEC

BLEN = 1 -> 0 个信息字符 + CRC

BLEN = 255 -> 254 个信息字符 + CRC（总共 256 个字符）

在智能卡模式下，块长度计数器在 TXE=0（使能 FIFO 模式时为 TXFE = 0）时复位。

此位域也可用于其他模式。这种情况下，块长度计数器在 RE=0（禁止接收器）和/或 EOBCF 位写入 1 时复位。

注： 块接收开始后可编程此值（使用起始字段中 LEN 字符中的数据）。每个接收到的块只能对此值编程一次。

位 23:0 **RTO[23:0]**：接收器超时值 (Receiver timeout value)

此位域用于提供接收器的超时值（以波特时钟数为单位）。

在标准模式下，如果在接收到最后一个字符后，在 RTO 值对应的时间内未检测到新的起始位，则 RTOF 标志置 1。

在智能卡模式下，此值用于实施 CWT 和 BWT。有关更多详细信息，请参见智能卡章节。在标准模式下，从接收到的最后一个字符的起始位开始执行 CWT/BWT 测量。

注： 每个接收到的字符只能对此值编程一次。

注： 可以实时写入 RTOR。如果新值小于或等于计数器的值，RTOF 标志置 1。

如果不支持接收器超时功能，此寄存器保留并由硬件强制为“0x00000000”。请参见第 1455 页的第 44.4 节：USART 实现。

44.7.7 USART 请求寄存器 (USART_RQR)

USART request 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
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	
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TXFRQ	RXFRQ	MMRQ	SBKRQ	ABRRQ
											w	w	w	w	w

位 31:5 保留，必须保持复位值。

位 4 **TXFRQ**：发送数据刷新请求 (Transmit data flush request)

禁止 FIFO 模式时，向该位写入“1”会将 TXE 标志置 1。这可丢弃发送数据。由于错误 (NACK) 而未发送数据且 USART_ISR 寄存器中的 FE 标志有效时，只能在智能卡模式下使用此位。如果 USART 不支持智能卡模式，该位保留且必须保持复位值。

使能 FIFO 时，TXFRQ 位置 1 以清空整个 FIFO。这会将 TXFE 标志（发送 FIFO 为空，USART_ISR 寄存器中的位 23）置位。在 UART 模式和智能卡模式下都支持清空发送 FIFO。

注： 在 FIFO 模式下，TXFNF 标志在清空请求期间复位，直到 TxFIFO 为空，以确保数据寄存器中没有写入数据。

位 3 **RXFRQ**：接收数据刷新请求 (Receive data flush request)

向该位写入 1 将清空整个接收 FIFO（即，将 RXFNE 位清零）。

这可以丢弃接收的数据而不对其执行读取操作，并避免发生上溢情况。

位 2 **MMRQ**：静默模式请求 (Mute mode request)

向此位写入 1 可将 USART 置于静默模式，并将 RWU 标志复位。

位 1 **SBKRQ**：发送中断请求 (Send break request)

向此位写入 1 可将 SBKF 标志置 1 并在发送设备可用后立即请求在线路上发送 BREAK。

注： 如果应用需要在之前插入的所有数据（包括尚未发送的数据）后发送中断字符，软件应等到 TXE 标志使能后将 SBKRQ 位置 1。

位 0 **ABRRQ**：自动波特率请求 (Auto baud rate request)

向此位写入 1 可复位 USART_ISR 中的 ABRF 标志，并请求对下一个接收到的数据帧进行自动波特率测量。

注： 如果 USART 不支持自动波特率功能，该位保留且必须保持复位值。请参见第 1455 页的第 44.4 节：USART 实现。

44.7.8 USART 中断和状态寄存器 (USART_ISR)

USART interrupt and status register

偏移地址：0x1C

复位值：0x0XX0 00C0

禁用 FIFO 时，XX = 00

使能 FIFO/智能卡模式时，XX = 28

使能 FIFO 模式且禁止智能卡模式时，XX = 08

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	TXFT	RXFT	TCBGT	RXFF	TXFE	RE ACK	TE ACK	WUF	RWU	SBKF	CMF	BUSY
				r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ABRF	ABRE	UDR	EOBF	RTOF	CTS	CTSIF	LBDF	TXE/TX FNF	TC	RXNE/ RXFNE	IDLE	ORE	NE	FE	PE
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:28 保留，必须保持复位值。

位 27 **TXFT**：TXFIFO 阈值标志 (TXFIFO threshold flag)

当 TXFIFO 达到在 USART_CR3 寄存器的 TXFTCFG 中编程的阈值时（即 TXFIFO 包含 TXFTCFG 个空位置），该位由硬件置 1。如果 USART_CR3 寄存器中的 TXFTIE 位 =1（位 31），则会生成中断。

0：TXFIFO 未达到编程的阈值。

1：TXFIFO 已达到编程的阈值。

位 26 **RXFT**：RXFIFO 阈值标志 (RXFIFO threshold flag)

达到在 USART_CR3 寄存器的 RXFTCFG 中编程的阈值时，该位由硬件置 1。这意味着，接收 FIFO 中有 (RXFTCFG - 1) 个数据，USART_RDR 寄存器中有一个数据。如果 USART_CR3 寄存器中的 RXFTIE 位 = 1（位 27），则会生成中断。

0：接收 FIFO 未达到编程的阈值。

1：接收 FIFO 已达到编程的阈值。

注： *当 RXFTCFG 阈值配置为“101”时，如果存在 16 个数据（即 RXFIFO 中有 15 个数据，USART_RDR 中有 1 个数据），则 RXFT 标志将置 1。因此，接收到的第 17 个数据不会导致上溢错误。接收到第 18 个数据后会发生上溢错误。*

位 25 **TCBGT**：保护时间前发送完成标志 (Transmission complete before guard time flag)

当写入到 USART_TDR 中的最后一个数据已正确从移位寄存器中发出时，该位置 1。

如果包含数据的帧完成发送，并且智能卡未发回任何 NACK，则该位在智能卡模式下由硬件置 1。如果 USART_CR3 寄存器中的 TCBGTIE=1，则生成中断。

此位由软件清零，方法是向 USART_ICR 寄存器中的 TCBGTCF 写入 1 或向 USART_TDR 寄存器执行写操作。

0：发送未完成或发送未成功完成（即，从智能卡接收到 NACK）。

1：发送成功完成（在保护时间结束之前完成，智能卡未发送 NACK）。

注： *如果 USART 不支持智能卡模式，该位保留且保持复位值。如果 USART 支持智能卡模式并使能智能卡模式，则 TCBGT 复位值为“1”。请参见第 1455 页的第 44.4 节：USART 实现。*

位 24 **RXFF**：RXFIFO 已满 (RXFIFO Full)

当接收到的数据量对应于 RXFIFO 大小 + 1（RXFIFO 已满 + USART_RDR 寄存器中的 1 个数据）时，该位由硬件置 1。

如果 USART_CR1 寄存器中 RXFFIE 位 = 1，则会生成中断。

0：RXFIFO 未满。

1：RXFIFO 已满。

位 23 **TXFE**：TXFIFO 为空 (TXFIFO Empty)

当 TXFIFO 为空时，该位由硬件置 1。当 TXFIFO 包含至少一个数据时，该标志被清零。也可以通过向 USART_RQR 寄存器中的位 TXFRQ（位 4）写入 1 将 TXFE 标志置 1。

如果 USART_CR1 寄存器中的 TXFEIE 位 = 1（位 30），则会生成中断。

0：TXFIFO 非空。

1：TXFIFO 为空。

位 22 **REACK**：接收使能确认标志 (Receive enable acknowledge flag)

USART 采用接收使能值时，通过硬件将此位置 1/复位。

此位可用于验证 USART 是否准备好在进入低功耗模式前接收数据。

注： 如果 USART 不支持从停止模式唤醒功能，该位保留且保持复位值。请参见第 1455 页的第 44.4 节：USART 实现。

位 21 **TEACK**：发送使能确认标志 (Transmit enable acknowledge flag)

USART 采用发送使能值时，通过硬件将此位置 1/复位。

通过写入 TE=0 生成空闲帧请求，然后在 USART_CR1 寄存器中写入 TE=1 以遵循 TE=0 最短周期时，可使用此位。

位 20 **WUF**：从低功耗模式唤醒标志 (Wakeup from low-power mode flag)

当检测到唤醒事件时，此位由硬件置 1。事件通过 WUS 位域定义。此位由软件清零，方法是向 USART_ICR 寄存器中的 WUCF 写入 1。

如果 USART_CR3 寄存器中 WUFIE=1，则会生成中断。

注： 当 UESM 清零时，WUF 标志也清零。

如果 USART 不支持从停止模式唤醒功能，该位保留且保持复位值。请参见第 1455 页的第 44.4 节：USART 实现。

位 19 **RWU**：*接收器从静默模式唤醒 (Receiver wakeup from Mute mode)*

该位指示 USART 是否处于静默模式。当识别出唤醒/静默序列时，此位由硬件清零/置 1。静默模式控制序列（地址或 IDLE）通过 USART_CR1 寄存器中的 WAKE 位进行选择。

当选择 IDLE 模式下唤醒时，该位只能通过用软件向 USART_RQR 寄存器中的 MMRQ 位写 1 的方式置 1。

0：接收器处于工作模式

1：接收器处于静默模式

注： 如果 USART 不支持从停止模式唤醒功能，该位保留且保持复位值。请参见第 1455 页的第 44.4 节：USART 实现。

位 18 **SBKF**：发送中断标志 (Send break flag)

此位指示已请求发送中断字符。通过将 1 写入 USART_CR3 寄存器中的 SBKRQ 位，此位由软件置 1。此位在中断发送的停止位期间由硬件自动复位。

0：不发送中断字符

1：将发送中断字符

位 17 **CMF**：字符匹配标志 (Character match flag)

接收到由 ADD[7:0] 定义的字符后由硬件将此位置 1。通过向 USART_ICR 寄存器中的 CMCF 写入 1，此位由软件清零。

如果 USART_CR1 寄存器中 CMIE=1，则会生成中断。

0：未检测到字符匹配

1：检测到字符匹配

位 16 **BUSY**：忙标志 (Busy flag)

此位由硬件置 1 和复位。当 RX 线路上正在进行通信（成功检测到起始位）时有效。在接收结束（成功或失败）时复位。

0：USART 处于空闲状态（无接收）

1：正在接收

位 15 **ABRF**：自动波特率标志 (Auto baud rate flag)

已设置自动波特率（RXNE 也将置 1，并在 RXNEIE = 1 时生成中断），或者自动波特率操作未成功完成时，此位由硬件置 1 (ABRE=1)（此时，ABRE、RXNE 和 FE 也置 1）

为请求新的自动波特率检测，通过向 USART_RQR 寄存器中的 ABRRQ 写入 1，此位由软件清零。

注： 如果 USART 不支持自动波特率功能，该位保留且保持复位值。

位 14 **ABRE**：自动波特率错误 (Auto baud rate error)

如果波特率测量失败（波特率超出范围或字符比较失败），此位由硬件置 1。

通过将 1 写入 USART_CR3 寄存器中的 ABRRQ 位，此位由软件清零。

注： 如果 USART 不支持自动波特率功能，该位保留且保持复位值。

位 13 **UDR**：SPI 从器件下溢错误标志 (SPI slave underrun error flag)

在从发送模式下，如果在软件尚未将任何值加载到 USART_TDR 时出现第一个数据发送时钟脉冲，此标志将置 1。该标志通过将 USART_ICR 寄存器中的 UDRCF 位置 1 来复位。

0：无下溢错误

1：存在下溢错误

注： 如果 USART 不支持 SPI 从器件模式，该位保留且保持复位值。请参见第 1455 页的第 44.4 节：USART 实现。

位 12 **EOBF**：块结束标志 (End of block flag)

接收到完整块后，此位由硬件置 1（例如 T=1 智能卡模式）。接收到的字节数（从块的起始处，包括起始字段）等于或大于 BLEN + 4 时执行检测。

如果 USART_CR2 寄存器中 EOBIE = 1，则会生成中断。

通过向 USART_ICR 寄存器中的 EOBCF 写入 1，此位由软件清零。

0：未达到块结束

1：已达到块结束（字符数）

注： 如果不支持智能卡模式，该位保留且保持复位值。请参见第 1455 页的第 44.4 节：USART 实现。

位 11 **RTOF**：接收器超时 (Receiver timeout)

已经过在 RTOR 寄存器中编程的超时值后，如果无任何通信，此位由硬件置 1。通过向 USART_ICR 寄存器中的 RTOCF 写入 1，此位由软件清零。

如果 USART_CR2 寄存器中 RTOIE=1，则会生成中断。

在智能卡模式下，该超时对应于 CWT 或 BWT 时间。

0：未达到超值值

1：已达到超时值，未接收到任何数据

注： 如果 RTOR 寄存器中编程的时间值将 2 个字符隔开，则 RTOF 不置 1。如果此时间大于该值 + 2 个采样时间（2/16 或 2/8，具体取决于过采样方法），则 RTOF 标志置 1。即使 RE = 0，计数器仍会计数，但 RTOF 仅在 RE = 1 时置 1。如果 RE 置 1 时已经超时，则 RTOF 将置 1。

如果 USART 不支持接收器超时功能，该位保留且保持复位值。

位 10 **CTS**：CTS 标志 (CTS flag)

此位由硬件置 1/复位。此位是对 nCTS 输入引脚的状态取反。

0：nCTS 线置 1

1：nCTS 线复位

注： 如果不支持硬件流控制功能，该位保留且保持复位值。

位 9 **CTSIF**：CTS 中断标志 (CTS interrupt flag)

如果 CTSE 位置 1，当 nCTS 输入切换时，此位由硬件置 1。通过将 1 写入 USART_ICR 寄存器中的 CTSCF 位，此位由软件清零。

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： 如果不支持硬件流控制功能，该位保留且保持复位值。

位 8 **LBDF**：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。通过向 USART_ICR 寄存器中的 LBDCF 写入 1，此位由软件清零。

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： 如果 USART 不支持 LIN 模式，该位保留且保持复位值。请参见第 1455 页的第 44.4 节：USART 实现。

位 7 **TXE/TXFNF**：发送数据寄存器为空/TXFIFO 未满 (Transmit data register empty/TXFIFO not full)

如果禁止 FIFO 模式，则当 USART_TDR 寄存器的内容已传输到移位寄存器时，TXE 由硬件置 1。通过对 USART_TDR 寄存器执行写操作将该位清零。为丢弃数据（仅在智能卡 T=0 模式下出现发送故障时），也可以通过向 USART_RQR 寄存器中的 TXFRQ 写入 1 来将 TXE 标志置 1。

如果使能 FIFO 模式，TXFNF 会在 TXFIFO 未满时由硬件置 1，表示可向 USART_TDR 中写入数据。每次对 USART_TDR 进行写操作都会将数据置于 TXFIFO 中。该标志保持置 1，直到 TXFIFO 已满。当 TXFIFO 已满时，该标志清零，表示不能向 USART_TDR 中写入数据。

如果 USART_CR1 寄存器中 TXEIE/TXFNFIE 位 = 1，则会生成中断。

0：数据寄存器已满/发送 FIFO 已满。

1：数据寄存器/发送 FIFO 未满。

注： 在清空请求期间，TXFNF 保持复位，直到 TXFIFO 为空。发送清空请求（通过将 TXFRQ 位置 1）后，应先检查 TXFNF 标志，然后再写入 TXFIFO（TXFNF 和 TXFE 将同时置 1）。单缓冲区发送期间使用该位。

位 6 **TC**：发送完成 (Transmission complete)

该位指示写入到 USART_TDR 中的最后一个数据已从移位寄存器中发出。

如果已完成对包含数据的帧的发送并且 TXE/TXFE 置 1，则此位由硬件置 1。

如果 USART_CR1 寄存器中 TCIE = 1，则会生成中断。

TC 位由软件清零，方法是向 USART_ICR 寄存器中的 TCCF 写入 1 或向 USART_TDR 寄存器执行写操作。

0：传送未完成

1：传送已完成

注： 如果 TE 位复位且无任何发送正在进行，TC 位会立即置 1。

位 5 **RXNE/RXFNE**：读取数据寄存器非空/RXFIFO 非空 (Read data register not empty/RXFIFO not empty)

当 USART_RDR 移位寄存器的内容已传输到 USART_RDR 寄存器时，RXNE 位由硬件置 1。通过对 USART_RDR 寄存器执行读取操作将该位清零。也可以通过将 USART_RQR 寄存器中的 RXFRQ 位置 1 将 RXNE 标志位清零。

RXFIFO 非空时，RXFNE 位由硬件置 1，这表示可以从 USART_RDR 寄存器读取数据。每次对 USART_RDR 进行读操作都会在 RXFIFO 中释放一个位置。

RXNE/RXFNE 在 RXFIFO 为空时清零。也可以通过将 USART_RQR 寄存器中的 RXFRQ 位置 1 将 RXNE/RXFNE 标志位清零。

如果 USART_CR1 寄存器中 RXNEIE/RXFNEIE = 1，则会生成中断。

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。通过向 USART_ICR 寄存器中的 IDLECF 写入 1，此位由软件清零。

0：未检测到空闲线路

1：检测到空闲线路

注： *直到 RXNE 位已置 1 时（即，当出现新的空闲线路时）IDLE 位才会被再次置 1。*

使能静默模式 (MME=1) 后，如果 USART 未静默 (RWU=0)，则 IDLE 置 1，无论是否通过 WAKE 位选择了静默模式。如果 RWU=1，IDLE 不置 1。

位 3 **ORE**：溢出错误 (Overrun error)

在 RXNE =“1”（使能 FIFO 模式时为 RXFF =“1”）的情况下，当移位寄存器中当前正在接收的数据准备好传输到 USART_RDR 寄存器时，此位由硬件置 1。通过向 USART_ICR 寄存器中的 ORECF 写入 1，此位由软件清零。

如果 USART_CR1 寄存器中的 RXNEIE/RXFNEIE=1 或 EIE = 1，则会生成中断。

0：无溢出错误

1：检测到溢出错误

注： *当此位置 1 时，USART_RDR 寄存器的内容不会丢失，但移位寄存器会被覆盖。EIE 位置 1 后，如果在多缓冲区通信中 ORE 标志置 1，则会生成中断。*

USART_CR3 寄存器中的 OVRDIS 位置 1 时，此位将被永久强制清零（无上溢检测）。

位 2 **NE**：噪声检测标志 (Noise detection flag)

当在接收的帧上检测到噪声时，该位由硬件置 1。通过向 USART_ICR 寄存器中的 NFCF 写入 1，此位由软件清零。

0：未检测到噪声

1：检测到噪声

注： *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/RXFNE 位出现的时间相同。EIE 位置 1 后，如果在多缓冲区通信中 NE 标志置 1，则会生成中断。*

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 NE 标志（请参见第 1473 页的第 44.5.8 节：USART 接收器对时钟偏差的容差）。

在 FIFO 模式下，此错误与 USART_RDR 中的字符相关联。

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。通过向 USART_ICR 寄存器中的 FECF 写入 1，此位由软件清零。

在智能卡模式下发送数据时，如果在达到最大发送尝试次数后仍未成功（智能卡向数据帧发送 NACK 信号），则此位置 1。

如果 USART_CR1 寄存器中 EIE = 1，则会生成中断。

0：未检测到帧错误

1：检测到帧错误或中断字符

注： *在 FIFO 模式下，此错误与 USART_RDR 中的字符相关联。*

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。通过向 USART_ICR 寄存器中的 PECF 写入 1，此位由软件清零。

如果 USART_CR1 寄存器中的 PEIE = 1，则会生成中断。

0：无奇偶校验错误

1：奇偶校验错误

注： *在 FIFO 模式下，此错误与 USART_RDR 中的字符相关联。*

44.7.9 USART 中断标志清零寄存器 (USART_ICR)

USART interrupt flag clear 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
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WUCF	Res.	Res.	CMCF	Res.
											w			w	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	UDRCF	EOBCF	RTOCF	Res.	CTSCF	LBDCF	TCBGTCF	TCCF	TXFECF	IDLECF	ORECF	NECF	FECF	PECF
		w	w	w		w	w	w	w	w	w	w	w	w	w

位 31:21 保留，必须保持复位值。

位 20 **WUCF**：从低功耗模式唤醒清零标志 (Wakeup from low-power mode clear flag)

将 1 写入此位时，USART_ISR 寄存器中 WUF 标志将清零。

注： *如果 USART 不支持从停止模式唤醒功能，该位保留且必须保持复位值。请参见第 1455 页的第 44.4 节：USART 实现。*

位 19:18 保留，必须保持复位值。

位 17 **CMCF**：字符匹配清零标志 (Character match clear flag)
将 1 写入此位时，USART_ISR 寄存器中 CMF 标志将清零。

位 16:14 保留，必须保持复位值。

位 13 **UDRCF**：SPI 从器件下溢清零标志 (SPI slave underrun clear flag)
将 1 写入此位时，USART_ISR 寄存器中 UDRF 标志将清零。
注：　如果 USART 不支持 SPI 从器件模式，该位保留且必须保持复位值。请参见第 1455 页的第 44.4 节：USART 实现。

位 12 **EOBCF**：块结束清零标志 (End of block clear flag)
将 1 写入此位时，USART_ISR 寄存器中 EOBF 标志将清零。
注：　如果 USART 不支持智能卡模式，该位保留且必须保持复位值。请参见第 1455 页的第 44.4 节：USART 实现。

位 11 **RTOCF**：接收器超时清零标志 (Receiver timeout clear flag)
将 1 写入此位时，USART_ISR 寄存器中 RTOF 标志将清零。
注：　如果 USART 不支持接收器超时功能，该位保留且必须保持复位值。请参见第 1455 页的第 44.4 节：USART 实现。

位 10 保留，必须保持复位值。

位 9 **CTSCF**：CTS 清零标志 (CTS clear flag)
将 1 写入此位时，USART_ISR 寄存器中 CTSIF 标志将清零。
注：　如果不支持硬件流控制功能，该位保留且必须保持复位值。请参见第 1455 页的第 44.4 节：USART 实现。

位 8 **LBDCF**：LIN 断路检测清零标志 (LIN break detection clear flag)
将 1 写入此位时，USART_ISR 寄存器中 LBDF 标志将清零。
注：　如果不支持 LIN 模式，该位保留且必须保持复位值。请参见第 1455 页的第 44.4 节：USART 实现。

位 7 **TCBGTCF**：保护时间前发送完成清零标志 (Transmission complete before Guard time clear flag)
将 1 写入此位时，USART_ISR 寄存器中 TCBGT 标志将清零。

位 6 **TCCF**：发送完成清零标志 (Transmission complete clear flag)
将 1 写入此位时，USART_ISR 寄存器中 TC 标志将清零。

位 5 **TXFECF**：TXFIFO 为空清零标志 (TXFIFO empty clear flag)
将 1 写入此位时，USART_ISR 寄存器中 TXFE 标志将清零。

位 4 **IDLECF**：检测到空闲线路清零标志 (Idle line detected clear flag)
将 1 写入此位时，USART_ISR 寄存器中 IDLE 标志将清零。

位 3 **ORECF**：上溢错误清零标志 (Overrun error clear flag)
将 1 写入此位时，USART_ISR 寄存器中 ORE 标志将清零。

位 2 **NECF**：检测到噪声清零标志 (Noise detected clear flag)
将 1 写入此位时，USART_ISR 寄存器中 NE 标志将清零。

位 1 **FECF**：帧错误清零标志 (Framing error clear flag)
将 1 写入此位时，USART_ISR 寄存器中 FE 标志将清零。

位 0 **PECF**：奇偶校验错误清零标志 (Parity error clear flag)
将 1 写入此位时，USART_ISR 寄存器中 PE 标志将清零。

44.7.10 USART 接收数据寄存器 (USART_RDR)

USART receive data register

偏移地址：0x24

复位值：0xXXXX XXXX

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.	Res.	Res.	Res.	Res.	Res.	RDR[8:0]								
							r	r	r	r	r	r	r	r	r

位 31:9 保留，必须保持复位值。

位 8:0 **RDR[8:0]**：接收数据值 (Receive data value)

包含接收到的数据字符。

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（请参见*图 451*）。

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。

44.7.11 USART 发送数据寄存器 (USART_TDR)

USART transmit data register

偏移地址：0x28

复位值：0xXXXX XXXX

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.	Res.	Res.	Res.	Res.	Res.	TDR[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 **TDR[8:0]**：发送数据值 (Transmit data value)

包含要发送的数据字符。

USART_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（请参见*图 451*）。

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。

注：　只能在 TXE/TXFNF=1 时写入此寄存器。

44.7.12 USART 预分频器寄存器 (USART_PRESC)

USART prescaler register

只有在禁止 USART (UE=0) 时才能写入此寄存器。

偏移地址：0x2C

复位值：0x0000 0000

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

位 31:4 保留，必须保持复位值。

位 3:0 **PRESCALER[3:0]**：时钟预分频器 (Clock prescaler)

USART 输入时钟可通过预分频系数进行分频：

0000：输入时钟未分频
0001：输入时钟 2 分频
0010：输入时钟 4 分频
0011：输入时钟 6 分频
0100：输入时钟 8 分频
0101：输入时钟 10 分频
0110：输入时钟 12 分频
0111：输入时钟 16 分频
1000：输入时钟 32 分频
1001：输入时钟 64 分频
1010：输入时钟 128 分频
1011：输入时钟 256 分频
其余组合：保留

注： 为 PRESCALER 编程不允许的值时，编程的预分频值将为“1011”，即输入时钟除以 256。

44.7.13 USART 寄存器映射

下表提供了 USART 寄存器映射和复位值。

表 268. USART 寄存器映射和复位值

偏移	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	USART_CR1	RXFFIE	TXFEIE	FIFOEN	M1	EOBIE	RTOIE	DEAT4	DEAT3	DEAT2	DEAT1	DEAT0	DEDT4	DEDT3	DEDT2	DEDT1	DEDT0	OVER8	CMIE	MME	M	WAKE	PCE	PS	PEIE	TXEIE	TCIE	RXNEIE	IDLEIE	TE	RE	UESM	UE
	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
0x04	USART_CR2	ADD[7:4]				ADD[3:0]				RTOEN	ABRMOD1	ABRMOD0	ABREN	MSBFIRST	DATAINV	TXINV	RXINV	SWAP	LINEN	STOP [1:0]		CLKEN	CPOL	CPHA	LBCL	Res.	LBDIE	LBDL	ADDM7	DIS_NSS	Res.	Res.	SLVEN
	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
0x08	USART_CR3	TXFT CFG			RXFTIE	RXFT CFG			TCBGTIE	TXFTIE	WUFIE	WUS [1:0]		SCAR CNT [2:0]			Res.	DEP	DEM	DDRE	OVRDIS	ONEBIT	CTSIE	CTSE	RTSE	DMAT	DMAR	SCEN	NACK	HDSEL	IRLP	IREN	EIE
	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
0x0C	USART_BRR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BRR[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x10	USART_GTPR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	GT[7:0]								PSC[7:0]							
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x14	USART_RTOR	BLEN[7:0]								RTO[23:0]																							
	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
0x18	USART_RQR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TXFRQ	RXFRQ	MMRQ	SBKRQ	ABRRQ
	Reset value																												0	0	0	0	0
0x1C	USART_ISR	Res.	Res.	Res.	Res.	TXFT	RXFT	TCBGT	RXFF	TXFE	REACK	TEACK	WUF	RWU	SBKF	CMF	BUSY	ABRF	ABRE	UDR	EOBF	RTOF	CTS	CTSIF	LBDF	TXE	TC	RXNE	IDLE	ORE	NE	FE	PE
	Reset value					0	0	X	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
0x20	USART_ICR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WUCF	Res.	Res.	CMCF	Res.	Res.	Res.	UDRCF	EOBCF	RTOCF	Res.	CTSCF	LBDCF	TCBGTCF	TCCF	TXFECF	IDLECF	ORECF	NECF	FECF	PECF
	Reset value												0			0				0	0	0		0	0	0	0	0	0	0	0	0	0
0x24	USART_RDR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RDR[8:0]								
	Reset value																								0	0	0	0	0	0	0	0	0
0x28	USART_TDR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TDR[8:0]								
	Reset value																								0	0	0	0	0	0	0	0	0

表 268. USART 寄存器映射和复位值（续）

偏移	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
0x2C	USART_PRESC	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PRESCALER[3:0]			
	Reset value																													0	0	0	0

有关寄存器边界地址的信息，请参见*第 78 页的第 2.2.2 节*。

45 低功耗通用异步接收器 (LPUART)

本节介绍低功耗通用异步收发器 (LPUART)。

45.1 LPUART 简介

LPUART 是一种 UART，允许在有限功耗下双向 UART 通信。仅需 32.768 kHz LSE 时钟即可进行高达 9600 波特/s 的 UART 通信。当 LPUART 由与 LSE 时钟不同的时钟源计时，可以达到更高的波特率。

即使当微控制器处于低功耗模式，能耗极低时，LPUART 也会等待 UART 帧的到来。LPUART 包含所有必要的硬件支持，使在最小功耗下可以进行异步串行通信。

它支持半双工单线通信和调制解调器操作 (CTS/RTS)，还支持多处理器通信。

DMA（直接存储器访问）可用于数据发送/接收。

45.2 LPUART 主要特性

- 全双工异步通信
- NRZ 标准格式（标记/空格）
- 可编程波特率
- 使用 32.768 kHz 时钟源时波特率为 300 波特/s 到 9600 波特/s
- 使用高频时钟源可实现更高的波特率
- 两个用于收发数据的内部 FIFO
  每个 FIFO 均可由软件使能/禁止，并且均带有用于指示 FIFO 状态的状态标志
- 双时钟域，带有独立于 PCLK 的外设专用内核时钟
- 数据字长度可编程（7 位、8 位或 9 位）
- 可编程的数据顺序，最先移位 MSB 或 LSB
- 停止位可配置（支持 1 个或 2 个停止位）
- 单线半双工通信
- 使用 DMA 实现连续通信
- 使用中央 DMA 在预留的 SRAM 缓冲区中收/发字节
- 发射器和接收器有单独的使能位
- 发送和接收的单独信号极性控制
- Tx/Rx 引脚配置可交换
- 调制解调器和 RS-485 收发器的硬件流控制
- 传输检测标志：
  - 接收缓冲区已满
  - 发送缓冲区为空
  - BUSY 标志和发送结束标志
- 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：
  - 上溢错误
  - 噪声检测
  - 帧错误
  - 奇偶校验错误
- 具有标志的中断源
- 多处理器通信：从静默模式唤醒（通过空闲线检测或地址标记检测）

45.3 LPUART 功能说明

45.3.1 LPUART 框图

图 478. LPUART 框图

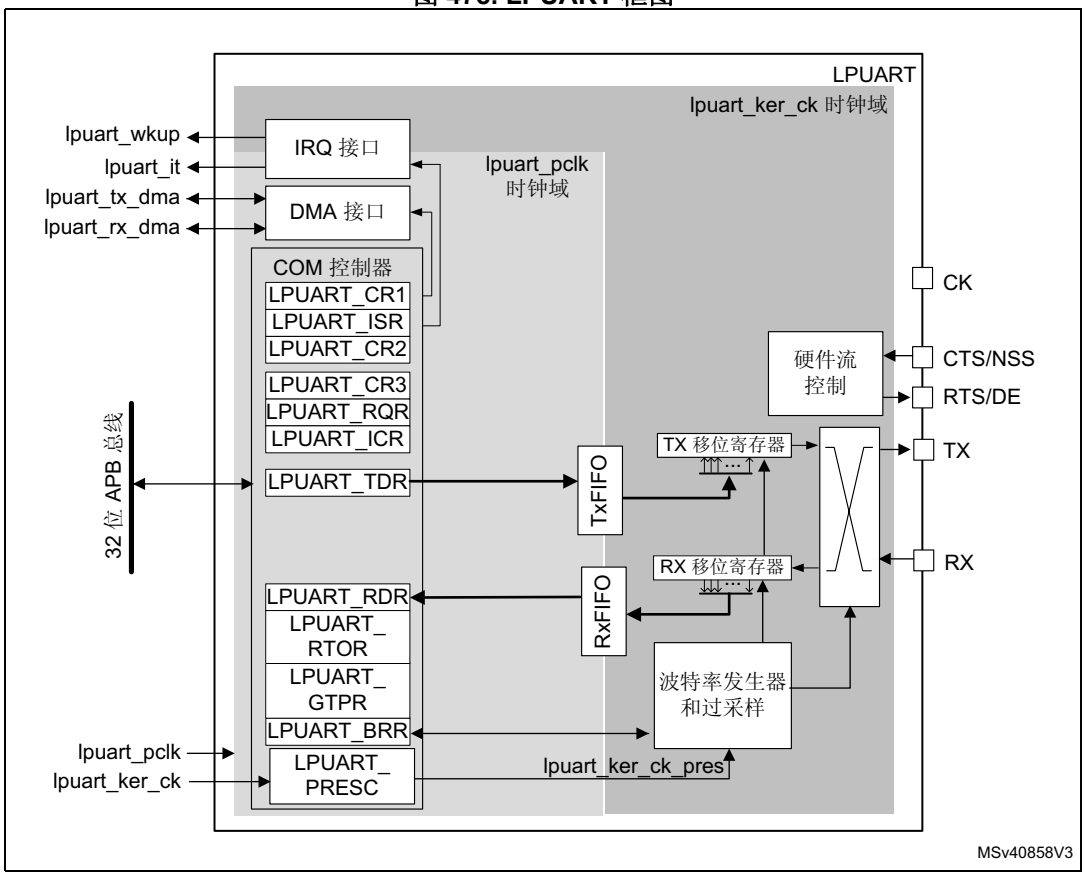


图 478 中的简化框图显示的是两个完全独立的时钟域：

- **lpuart_pclk** 时钟域

  lpuart_pclk 时钟信号馈送外设总线接口。需要访问 LPUART 寄存器时，该信号必须有效。

- **lpuart_ker_ck** 内核时钟域

  lpuart_ker_ck 是 LPUART 时钟源。它独立于 **lpuart_pclk**，由 RCC 提供。因此，即使 **lpuart_ker_ck** 停止，也可以对 LPUART 寄存器进行读/写操作。

  禁用双时钟域功能时，**lpuart_ker_ck** 与 lpuart_pclk 时钟相同。

lpuart_pclk 和 **lpuart_ker_ck** 之间无任何限制：**lpuart_ker_ck** 既可快于也可慢于 **lpuart_pclk**，唯一的限制是软件以足够快的速度管理通信的能力。

45.3.2 LPUART 信号

LPUART 双向通信需要至少两个引脚：接收数据输入引脚 (RX) 和发送数据输出引脚 (TX)：

- **RX**（接收数据输入引脚）
  RX 为串行数据输入引脚。
- **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。在单线模式下，该 I/O 用于发送和接收数据。

RS232 硬件流控制模式

在 RS232 硬件流控制模式下需要以下引脚：

- **CTS**（清除以发送）
  如果驱动为高电平，则该信号用于在当前传输结束时阻止数据发送。
- **RTS**（请求以发送）
  如果为低电平，则该信号用于指示 USART 已准备好接收数据。

RS485 硬件流控制模式

在 RS485 硬件控制模式下需要以下引脚：

- **DE** 驱动器使能
  该信号用于激活外部收发器的发送模式。

注： *DE 和 RTS 共用同一个引脚。*

45.3.3 LPUART 字符说明

可通过对 LPUART_CR1 寄存器中的 M 位（M0：位 12，M1：位 28）进行编程来将字长设置为 7 位、8 位或 9 位（请参见*图 452*）。

- 7 位字符长度：M[1:0] = “10”
- 8 位字符长度：M[1:0] = “00”
- 9 位字符长度：M[1:0] = “01”

在默认情况下，信号（TX 或 RX）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。

通过极性配置控制，可以单独针对每个信号对这些值取反。

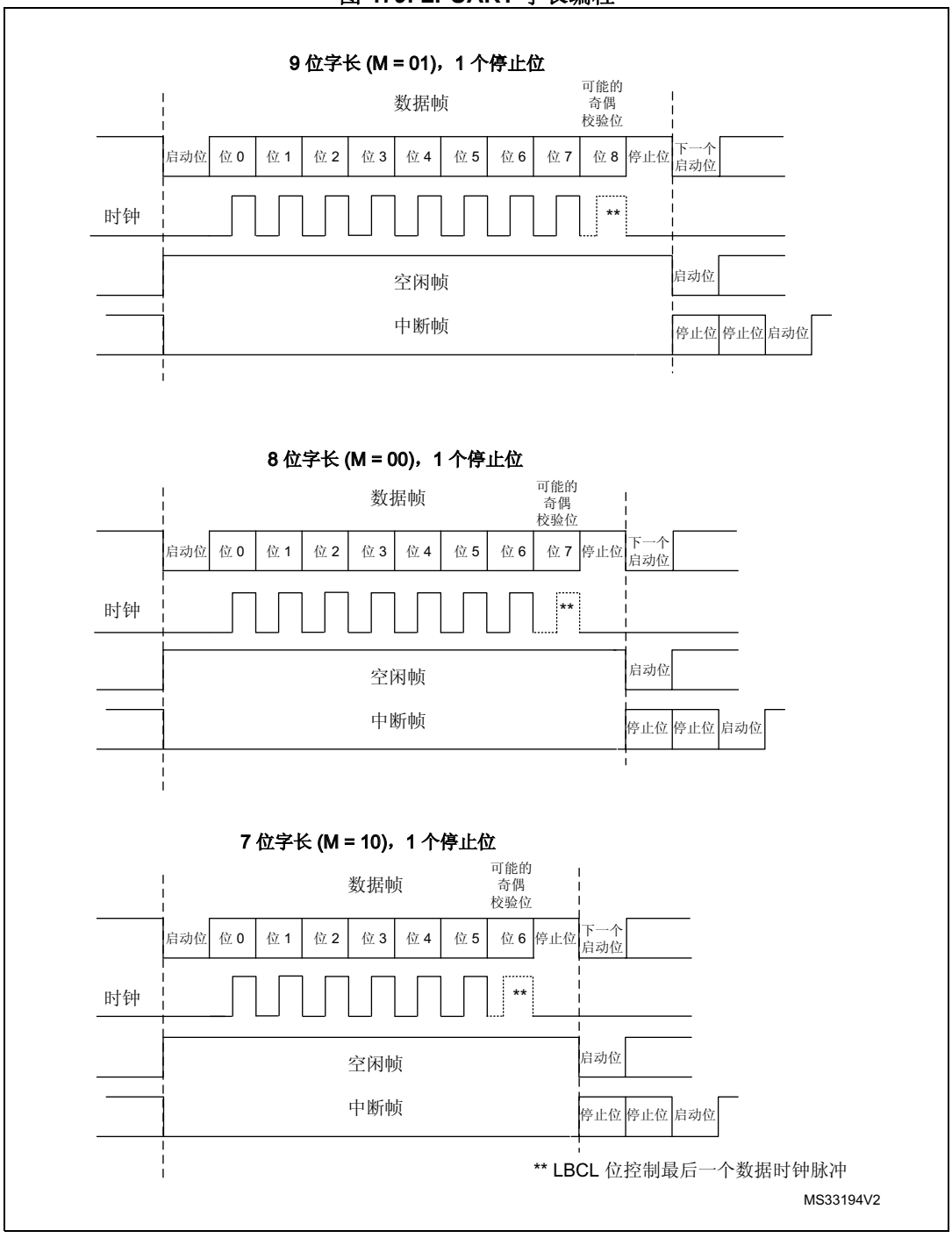
空闲字符可理解为整个帧周期内电平均为“1”。（停止位的电平也是“1”）。

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 2 个停止位。

发送和接收操作由通用波特率发生器驱动。发送器和接收器的使能位置 1 时，将分别生成发送时钟和接收时钟。

下面给出了各个块的详细信息。

图 479. LPUART 字长编程



45.3.4 LPUART FIFO 和阈值

LPUART 可工作在 FIFO 模式下。

LPUART 具有一个发送 FIFO (TXFIFO) 和一个接收 FIFO (RXFIFO)。可通过将 LPUART_CR1 寄存器中的 FIFOEN 位（位 29）置 1 使能 FIFO 模式。

最大数据字长度为 9 位，因此 TXFIFO 为 9 位宽。不过，RXFIFO 的默认宽度为 12 位。这是因为接收器不仅在 FIFO 中存储数据，而且还存储与每个字符相关的错误标志（奇偶校验错误、噪声错误和帧错误标志）。

注： *接收的数据与相应的标志一起存储在 RXFIFO 中，但读取 RDR 时仅读取数据。*

状态标志位于 LPUART_ISR 寄存器中。

可以定义触发 Tx 和 RX 中断的 TXFIFO 和 RXFIFO 阈值。这些阈值通过 LPUART_CR3 控制寄存器中的 RXFTCFG 和 TXFTCFG 位域进行编程。

在这种情况下：

- 当 RXFIFO 中接收到的数据量达到 RXFTCFG 位域中编程的阈值时，会生成 Rx 中断。此时，LPUART_ISR 寄存器中的 RXFT 标志置 1。这表示已接收到 RXFTCFG 数据：LPUART_RDR 中有 1 个数据，RXFIFO 中有 (RXFTCFG - 1) 个数据。例如，如果将 RXFTCFG 编程为“101”，则在接收到对应于 FIFO 大小的数据量（RXFIFO 中有 FIFO 大小 - 1 个数据，LPUART_RDR 中有 1 个数据）时，RXFT 标志将置 1。因此，下一个接收到的数据不会将上溢标志置 1。
- 当 TXFIFO 中的空位置数达到在 TXFTCFG 位域中编程的阈值时，会生成 Tx 中断。

45.3.5 LPUART 发送器

发送器可发送 7 位、8 位或 9 位的数据字，具体取决于 M 位的状态。要激活发送器功能，必须将发送使能位 (TE) 置 1。发送移位寄存器中的数据输出到 TX 引脚。

字符发送

LPUART 发送期间，首先通过 TX 引脚移出数据的最低有效位（默认配置）。该模式下，LPUART_TDR 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 478*）。

使能 FIFO 模式时，写入到 LPUART_TDR 寄存器中的数据会在 TXFIFO 中排队。

每个字符前面都有一个起始位，其对应于一个位周期的逻辑低电平。字符由可配置数量的停止位终止。

停止位的数量可为 1 或 2。

注： *向 LPUART_TDR 中写入要发送的数据前，TE 位必须先置 1。*

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，进而损坏 TX 引脚上的数据。当前发送的数据将会丢失。

使能 TE 位后，将会发送空闲帧。

可配置的停止位

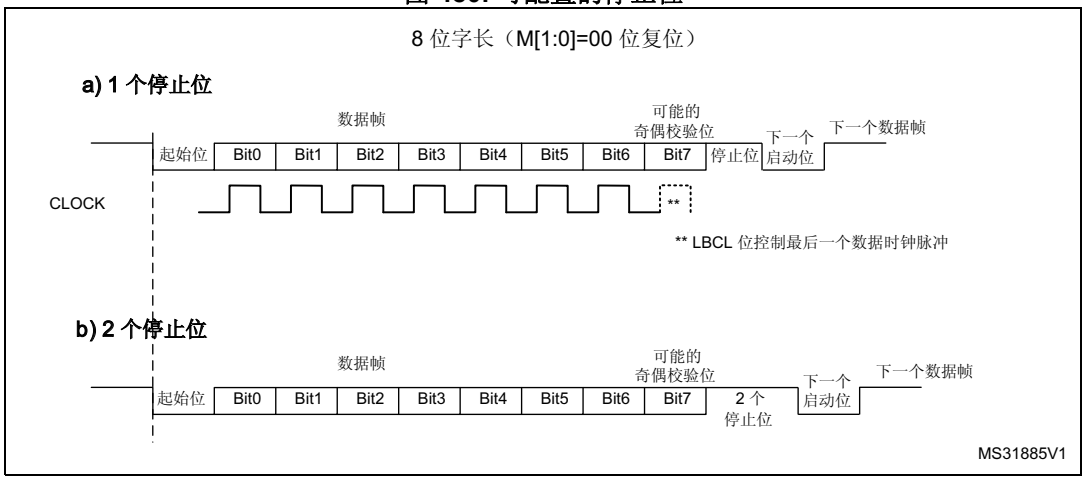
可以在 LPUART_CR2 的位 13 和位 12 中编程将随各个字符发送的停止位的数量。

- ***1 个停止位：***这是停止位数量的默认值。
- ***2 个停止位：***正常 LPUART 模式、单线模式和调制解调器模式支持该值。

空闲帧发送将包括停止位。

中断发送是 10 个低电平位（M[1:0] = “00”时）、11 个低电平位（M[1:0] = “01”时）或 9 个低电平位（M[1:0] = “10”时），然后是 2 个停止位。无法传送长中断（中断长度大于 9/10/11 个低电平位）。

图 480. 可配置的停止位



字符发送步骤

要发送字符，需遵循以下步骤：

1. 对 LPUART_CR1 中的 M 位进行编程以定义字长。
2. 使用 LPUART_BRR 寄存器选择所需波特率。
3. 对 LPUART_CR2 中的停止位数量进行编程。
4. 通过向 LPUART_CR1 寄存器中的 UE 位写入“1”使能 LPUART。
5. 如果将进行多缓冲区通信，请选择 LPUART_CR3 中的 DMA 使能 (DMAT)。按照*第 44.5.10 节：USART 多处理器通信*中的说明配置 DMA 寄存器。
6. 将 LPUART_CR1 中的 TE 位置 1 以便在首次发送时发送一个空闲帧。
7. 在 LPUART_TDR 寄存器中写入要发送的数据。为每个要在单缓冲区模式下发送的数据重复该操作。
   - 禁止 FIFO 模式时，向 LPUART_TDR 写入数据会将 TXE 标志清零。
   - 使能 FIFO 模式时，向 LPUART_TDR 写入数据会为 TXFIFO 增添一个数据。当 TXFNF 标志置 1 时，会对 LPUART_TDR 执行写操作。该标志会保持置 1，直到 TXFIFO 已满。
8. 将最后一个数据写入 LPUART_TDR 寄存器后，等待 TC =1。这表明最后一个帧的传送已完成。
   - 禁止 FIFO 模式时，这表示最后一个帧的发送已完成。
   - 使能 FIFO 模式时，这表示 TXFIFO 和移位寄存器均为空。

   当 LPUART 被禁止或进入暂停模式时，需要执行此检查来避免损坏最后一次发送。

单字节通信

- 禁止 FIFO 模式时：

  对发送数据寄存器进行写操作始终会清零 TXE 位。TXE 标志由硬件置 1 以指示：

  – 数据已从 LPUART_TDR 寄存器移到移位寄存器中且数据发送已开始；
  – LPUART_TDR 寄存器为空；
  – LPUART_TDR 寄存器中可写入下一个数据，而不会覆盖前一个数据。

  当 TXEIE 位置 1 时，TXE 标志会生成中断。

  发送时，要传入 LPUART_TDR 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 LPUART_TDR 寄存器的写指令会将数据置于移位寄存器中，数据发送开始时，TXE 位置 1。

- 使能 FIFO 模式时，TXFNF（TXFIFO 未满）标志由硬件置 1，以指示：
  – TXFIFO 未满；
  – LPUART_TDR 寄存器为空；
  – LPUART_TDR 寄存器中可写入下一个数据，而不会覆盖前一个数据。发送时，对 LPUART_TDR 寄存器的写操作会将数据存储在 TXFIFO 中。该数据将在当前发送结束时从 TXFIFO 复制到移位寄存器中。

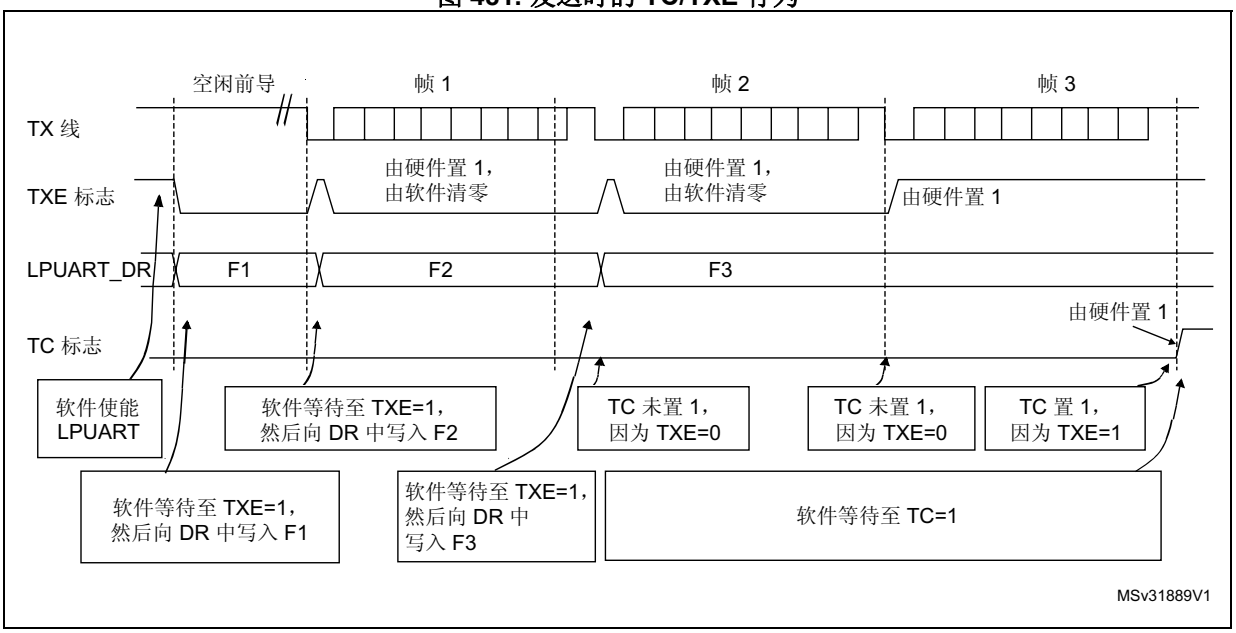
  TXFIFO 未满时，即使在对 LPUART_TDR 执行完写操作后，TXFNF 标志也保持为“1”。该标志在 TXFIFO 已满时清零。TXFNEIE 位置 1 时，该标志会生成中断。

  或者，当达到 TXFIFO 阈值时，会生成中断并将数据写入 TXFIFO。在这种情况下，CPU 可写入由编程的阈值定义的数据块。

  如果帧已发送（停止位后）且 TXE 标志（FIFO 模式下的 TXFE）置 1，TC 位将变为高电平。如果 LPUART_CR1 寄存器中的 TCIE 位置 1，将生成中断。

  向 LPUART_TDR 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 LPUART 或使微控制器进入低功率模式（请参见*图 481：发送时的 TC/TXE 行为*）。

图 481. 发送时的 TC/TXE 行为



注： *使能 FIFO 管理时，TXFNF 标志将用于数据发送。*

中断字符

将 SBKRQ 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 479*）。

如果将“1”写入 SBKRQ 位，则当前字符发送完成后，将在 TX 线路上发送一个中断字符。通过写操作将 SBKF 位置 1 并在中断字符发送完成时（发送中断字符后的停止位期间），该位由硬件复位。LPUART 在中断帧末尾的两位持续时间内插入一个逻辑“1”信号 (STOP)，以确保识别下个帧的起始位。

如果 SBKRQ 位置 1，则在当前发送结束时，会发送一个中断字符。

使能 FIFO 模式时，即使 TXFIFO 已满，发送中断字符的优先级也仍高于发送数据的优先级。

空闲字符

将 TE 位置 1 会驱动 LPUART 在第一个数据帧之前发送一个空闲帧。

45.3.6 LPUART 接收器

LPUART 可接收 7 位、8 位或 9 位的数据字，具体取决于 LPUART_CR1 寄存器中的 M 位。

起始位检测

在 LPUART 中，先在 Rx 线路上出现下降沿时检测起始位，然后在起始位中间采样以确认电平是否仍为“0”。如果起始采样为“1”，则噪声错误标志 (NE) 置 1，起始位将被丢弃，接收器将等待新的起始位。否则，接收器将继续正常采样所有传入位。

字符接收

LPUART 接收期间，首先通过 RX 引脚移入数据的最低有效位（默认配置）。该模式下，LPUART_RDR 寄存器的缓冲区 (RDR) 位于内部总线和接收移位寄存器之间。

字符接收步骤

要接收字符，需遵循以下步骤：

1. 对 LPUART_CR1 中的 M 位进行编程以定义字长。
2. 使用波特率寄存器 LPUART_BRR 选择所需波特率。
3. 对 LPUART_CR2 中的停止位数量进行编程。
4. 通过向 LPUART_CR1 寄存器中的 UE 位写入“1”使能 LPUART。
5. 如果将进行多缓冲区通信，请选择 LPUART_CR3 中的 DMA 使能 (DMAR)。按照*第 44.5.10 节：USART 多处理器通信*中的说明配置 DMA 寄存器。
6. 将 RE 位 LPUART_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

- 禁止 FIFO 模式时，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（及其相应的错误标志）。
- 如果已使能 FIFO 模式，则 RXFNE 位置 1，这表示 RXFIFO 非空。读取 LPUART_RDR 会返回输入到 RXFIFO 中的最早数据。接收到数据时，数据以及相应的错误位将一起被存储在 RXFIFO 中。
- 如果 RXNEIE（FIFO 模式下时为 RXFNEIE）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。

- 在多缓冲区通信模式下：
  - 禁止 FIFO 模式时，每接收到一个字节后 RXNE 标志均置 1，然后通过 DMA 对接收数据寄存器执行读操作清零。
  - 如果使能 FIFO 模式，则 RXFNE 标志在 RXFIFO 非空时置 1。每次收到 DMA 请求后，都会从 RXFIFO 检索数据。在 RXFIFO 非空时（即，当 RXFIFO 中存在要读取的数据时），会触发 DMA 请求。
- 在单缓冲区模式下：
  - 如果禁止 FIFO 模式，则通过软件对 LPUART_RDR 寄存器进行读操作来将 RXNE 标志清零。也可以通过将 LPUART_RQR 寄存器中的 RXFRQ 位置 1 将 RXNE 标志位清零。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。
  - 如果使能 FIFO 模式，则 RXFNE 标志在 RXFIFO 非空时置 1。每次对 LPUART_RDR 寄存器执行完读操作后，都会从 RXFIFO 中检索数据。RXFIFO 为空时，RXFNE 标志将清零。也可以通过将 LPUART_RQR 寄存器中的 RXFRQ 位置 1 将 RXFNE 标志清零。RXFIFO 已满时，必须在结束接收下一个字符前读取 RXFIFO 中的第一个条目，以避免发生上溢错误。当 RXFNEIE 位置 1 时，RXFNE 标志会生成中断。或者，当达到 RXFIFO 阈值时，会生成中断并从 RXFIFO 中读取数据。在这种情况下，CPU 可读取由编程的阈值定义的数据块。

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，除了在 IDLEIE 位置 1 时会生成中断外，处理步骤与接收到数据字符的情况相同。

上溢错误

- 禁止 FIFO 模式

  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 标志位置 1 时，如果在接收到下一个数据或尚未处理上一个 DMA 请求，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 LPUART_RDR 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 位置 1，则会生成中断。
- 使能 FIFO 模式

  移位寄存器准备好传送且接收 FIFO 已满时，会发生上溢错误。

  在 RXFIFO 中存在一个空闲位置之前，数据无法从移位寄存器传送到 LPUART_RDR 寄存器。当 RXFIFO 非空时，RXFNE 标志置 1。

  如果 RXFIFO 已满且移位寄存器已准备好传送，则会发生上溢错误。发生上溢错误时：
  - ORE 位置 1。
  - RXFIFO 中的第一个条目不会丢失。可通过对 LPUART_RDR 执行读操作来获取。
  - 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 RXFNEIE 位置 1 或 EIE 位置 1，则会生成中断。

通过将 ICR 寄存器中的 ORECF 位置 1 来复位 ORE 位。

注： *ORE 位置 1 时表示至少 1 个数据丢失。*

禁止 FIFO 模式时，有以下两种可能

- *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（和丢失）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。*

选择时钟源

时钟源的选择通过时钟控制系统完成（请参见*复位和时钟控制器 (RCC) 部分*）。在使能 LPUART 之前，必须通过 UE 位选择时钟源。

必须遵循以下两个条件选择时钟源：

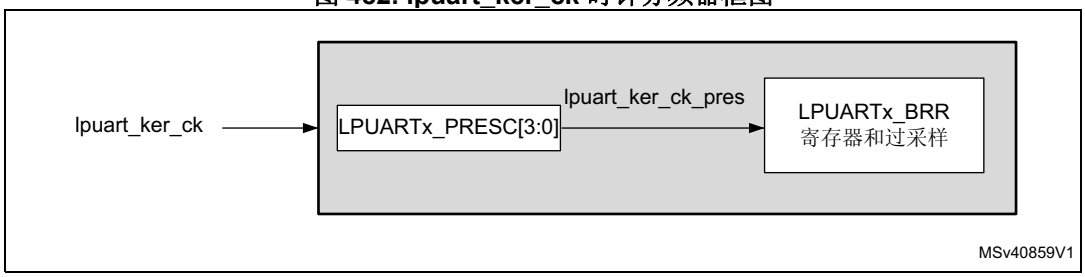
- 可在低功耗模式下使用 LPUART
- 通信速度

时钟源频率为 lpuart_ker_ck。

如果支持双时钟域和从低功耗模式唤醒功能，则 lpuart_ker_ck 时钟源可在 RCC 中进行配置（请参见*复位和时钟控制器 (RCC) 部分*）。否则，lpuart_ker_ck 与 lpuart_pclk 相同。

lpuart_ker_ck 可根据 LPUART_PRESC 寄存器中的可编程系数进行分频。

图 482. lpuart_ker_ck 时钟分频器框图



某些 lpuart_ker_ck 时钟源允许 LPUART 在 MCU 处于低功耗模式时接收数据。必要时，LPUART 可基于所接收的数据和唤醒模式的选择来唤醒 MCU，以通过用软件读取 LPUART_RDR 寄存器的方式或通过 DMA 的方式传输已接收数据。

对于其他时钟源，系统必须激活才能进行 LPUART 通信。

时钟源还决定通信速度范围（尤其是最大通信速度）。

接收器在尽可能靠近波特周期中间的位置采样每个传入波特。每个传入波特仅进行一次采样。

注： *不进行数据噪声检测。*

帧错误

如果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，则会检测到帧错误。

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 LPUART_RDR 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，LPUART_CR3 寄存器中的 EIE 位置 1 时将发出中断。

通过将 1 写入 LPUART_ICR 寄存器中的 FECF 位来复位 FE 位。

接收期间可配置的停止位

可通过 LPUART_CR2 的控制位配置要接收的停止位的数量：可以是 1 个或 2 个（正常模式下）。

- **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。
- **2 个停止位**：将在第二个停止位的中间对 2 个停止位进行采样。RXNE 和 FE 标志在此采样期间（例如，在第二个停止位期间）置 1。发生帧错误时不检测第一个停止位。

45.3.7 LPUART 波特率生成

接收器和发送器（Rx 和 Tx）的波特率均设置为 LPUART_BRR 寄存器中编程的值。

$$\text{Tx/Rx 波特} = \frac{256 \times \text{lpuartckpres}}{\text{LPUARTDIV}}$$

LPUARTDIV 在 LPUART_BRR 寄存器中定义。

注： *对 LPUART_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。*

禁止在 LPUART_BRR 寄存器中写入小于 0x300 的值。

f_{CK} 必须介于 3 x 波特率到 4096 x 波特率之间。

LPUART 时钟源为 LSE 时可达到的最大波特率为 9600 波特。当 LPUART 不使用 LSE 作为时钟源时（如 PCLK，SYSCLK），可以达到更高的波特率。例如，如果 LPUART 时钟源频率为 100 MHz，则可达到的最大波特率约为 33 M 波特。

表 269. lpuart_ker_ck_pres = 32,768 KHz 时编程的波特率的误差计算

波特率		lpuart_ker_ck_pres = 32,768 KHz		
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
1	0.3 KBps	0.3 KBps	0x6D3A	0
2	0.6 Kbps	0.6 Kbps	0x369D	0
3	1200 Bps	1200.087 Bps	0x1B4E	0.007
4	2400 Bps	2400.17 Bps	0xDA7	0.007
5	4800 Bps	4801.72 Bps	0x6D3	0.035
6	9600 KBps	9608.94 Bps	0x369	0.093

表 270. 采用 16 倍过采样 (OVER8 = 0) 时，在 f_{CK} = 100 MHz 下

波特率		f_{CK} = 100 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
1	38400 波特	38400,04 波特	A2C2A	0,0001
2	57600 波特	57600,06 波特	6C81C	0,0001
3	115200 波特	115200,12 波特	3640E	0,0001
4	230400 波特	230400,23 波特	1B207	0,0001
5	460800 波特	460804,61 波特	D903	0,001
6	921600 波特	921625,81 波特	6C81	0,0028
7	4000 K 波特	4000000,00 波特	1900	0
8	10000 K 波特	10000000,00 波特	A00	0
9	20000 K 波特	20000000,00 波特	500	0
10	30000 K 波特	33032258,06 波特	307	0,1

45.3.8 LPUART 接收器对时钟偏差的容差

仅当总时钟系统偏差小于 LPUART 接收器的容差时，LPUART 异步接收器才能正常工作。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 + DWU < LPUART 接收器偏差

其中

DWU 是使用从低功耗模式唤醒时因采样点偏差而产生的误差。

LPUART 接收器在*表 271* 中指定的最大容许偏差下可正确接收数据：

- 通过 LPUART_CR2 寄存器中的 STOP[1:0] 位定义的停止位数。
- LPUART_BRR 寄存器值。

表 271. LPUART 接收器的容差

M 位	768 < BRR < 1024	1024 < BRR < 2048	2048 < BRR < 4096	4096 ≤ BRR
8 位（M=00），1 个停止位	1.82%	2.56%	3.90%	4.42%
9 位（M=01），1 个停止位	1.69%	2.33%	2.53%	4.14%
7 位（M=“10”），1 个停止位	2.08%	2.86%	4.35%	4.42%
8 位（M=00），2 个停止位	2.08%	2.86%	4.35%	4.42%
9 位（M=01），2 个停止位	1.82%	2.56%	3.90%	4.42%
7 位（M=“10”），2 个停止位	2.34%	3.23%	4.92%	4.42%

注： *当接收的帧恰好包含 10 个（M 位 = “00”）、11 个（M 位 = “01”）或 9 个（M 位 = “10”）位时间的空闲帧时，*表 271 中指定的数据可能与特例中的数据略微不同。

45.3.9 LPUART 多处理器通信

可以执行 LPUART 多处理器通信（多个 LPUART 连接在一个网络中）。例如，其中一个 LPUART 可以是主器件，其 TX 输出与其他 LPUART 的 RX 输入相连接。其他 LPUART 为从器件，其各自的 TX 输出在逻辑上通过与运算连在一起，并与主 LPUART 的 RX 输入相连。

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 LPUART 服务开销。

可通过静默功能将未被寻址的器件置于静默模式下。为了使用静默模式功能，必须将 LPUART_CR1 寄存器中的 MME 位置 1。

注： *使能 FIFO 管理且 MME 已置 1 时，不得清零 MME 位再快速将其置 1（在两个 lpuart_ker_ck 周期内），否则静默模式可能保持有效。*

使能静默模式时：

- 不得将接收状态位置 1。
- 禁止任何接收中断。
- LPUART_ISR 寄存器中的 RWU 位置“1”。在某些情况下，RWU 可以由硬件或软件通过 LPUART_RQR 寄存器中的 MMRQ 位自动控制。

根据 LPUART_CR1 寄存器中 WAKE 位的设置，LPUART 可使用以下两种方法进入或退出静默模式：

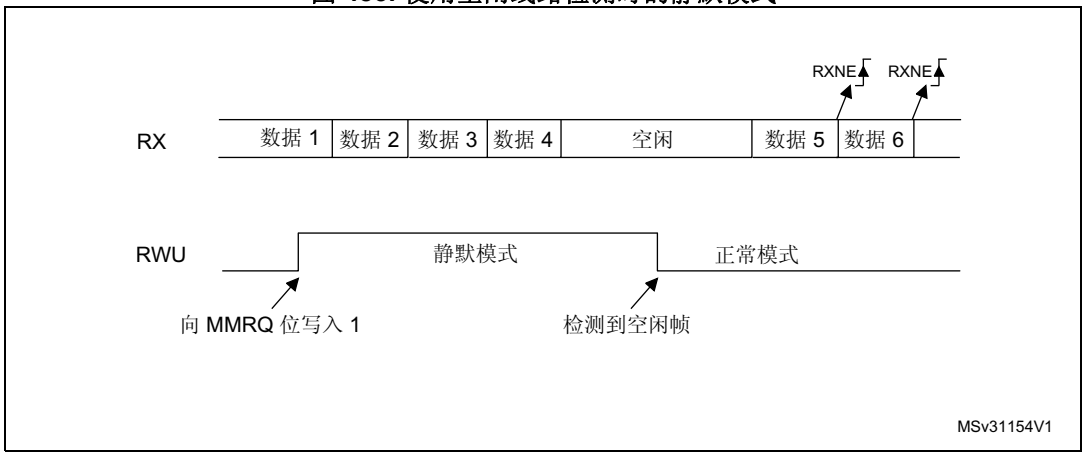
- 如果 WAKE 位被复位，则进行空闲线路检测。
- 如果 WAKE 位置 1，则进行地址标记检测。

空闲线路检测 (WAKE=0)

当向 MMRQ 位写入 1 且 RWU 位自动置 1 时，LPUART 进入静默模式。

检测到空闲帧时，LPUART 将唤醒。此时 RWU 位会由硬件清零，但 LPUART_ISR 寄存器中的 IDLE 位不会置 1。*图 483* 中给出了使用空闲线路检测时静默模式行为的示例。

图 483. 使用空闲线路检测时的静默模式



注： *如果在 IDLE 字符已经过去时将 MMRQ 位置 1，将不会进入静默模式（RWU 未置 1）。*

如果在线路处于空闲状态时激活 LPUART，在一个 IDLE 帧持续时间后（不只在接收一个字符帧后）会检测到空闲状态。

4 位/7 位地址标记检测 (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 个或 7 个 LSB 中。7 位或 4 位地址检测通过 ADDM7 位来选择。接收器会将此 4 位/7 位字与其地址进行比较，该接收器的地址在 LPUART_CR2 寄存器的 ADD 位中进行设置。

注： *在 7 位和 9 位数据模式下，地址检测分别在 6 位和 8 位地址上完成（ADD[5:0] 和 ADD[7:0]）。*

当接收到与其编程地址不匹配的地址字符时，LPUART 会进入静默模式。此时，RWU 位将由硬件置 1。LPUART 进入静默模式后，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。

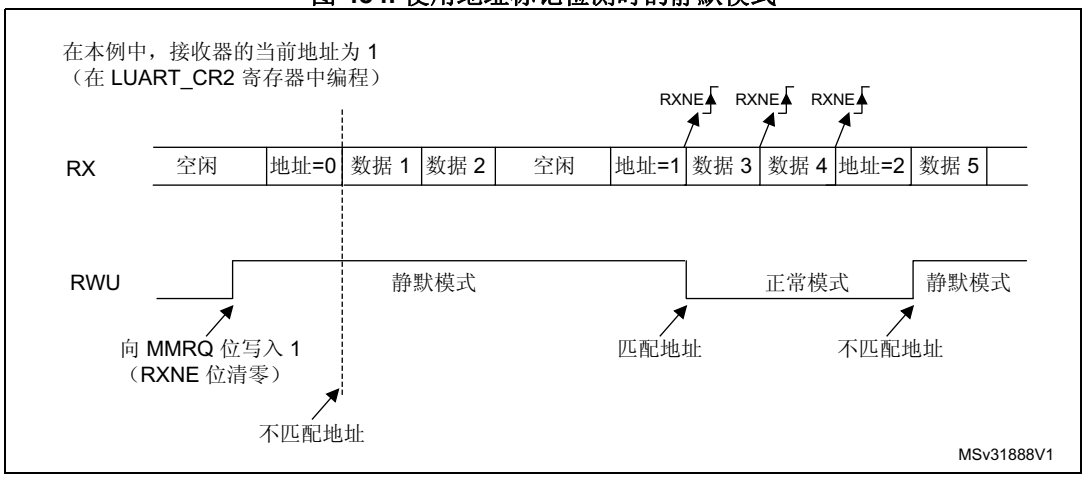
当向 MMRQ 位写入“1”时，LPUART 也会进入静默模式。这种情况下，RWU 位也自动置 1。

当接收到与编程地址匹配的地址字符时，LPUART 会退出静默模式。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/RXFNE 位会针对地址字符置 1。

注： *使能 FIFO 管理时，如果在接收器对数据的最后一位进行采样时 MMRQ 位置 1，则可在有效进入静默模式之前接收该数据。*

图 484 中给出了使用地址标记检测时静默模式行为的示例。

图 484. 使用地址标记检测时的静默模式



45.3.10 LPUART 极性控制

将 LPUART_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 272* 中列出了可能的 LPUART 帧格式。

表 272：LPUART 帧格式

M 位	PCE 位	LPUART 帧(1)
00	0	| SB | 8 位数据 | STB |
00	1	| SB | 7 位数据 | PB | STB |
01	0	| SB | 9 位数据 | STB |
01	1	| SB | 8 位数据 PB | STB |
10	0	| SB | 7 位数据 | STB |
10	1	| SB | 6 位数据 | PB | STB |

1. 图注：SB：起始位，STB：停止位，P：奇偶校验位。
2. 在数据寄存器中，PB 始终位于 MSB 位置（第 8 位或第 7 位，具体取决于 M 位的值）。

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 6 个、7 个或 8 个 LSB 位组成，具体取决于 M 位的值）。

例如，如果数据=00110101 且 4 个位置 1，则在选择偶校验（LPUART_CR1 寄存器中的 PS 位 = 0）时，校验位为 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 6 个、7 个或 8 个 LSB 位组成，具体取决于 M 位的值）。

例如，如果数据=00110101 且 4 个位置 1，则在选择奇校验（LPUART_CR1 寄存器中的 PS 位 = 1）时，校验位为 1。

接收时进行奇偶校验检查

如果奇偶校验检查失败，则 LPUART_ISR 寄存器中的 PE 标志置 1；如果 LPUART_CR1 寄存器中 PEIE 位置 1，则会生成中断。通过软件将 1 写入 LPUART_ICR 寄存器中的 PECF 位来清零 PE 标志。

发送时的奇偶校验生成

如果 LPUART_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）。

45.3.11 LPUART 单线半双工通信

通过将 LPUART_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：

- LPUART_CR2 寄存器中的 LINEN 和 CLKEN 位。
- LPUART_CR3 寄存器中的 SCEN 和 IREN 位。

LPUART 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 LPUART_CR3 寄存器中的控制位 HDSEL 可在半双工通信和全双工通信间进行选择。

向 HDSEL 位写入“1”后：

- 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，以便将 TX 配置为复用功能开漏并外接上拉电阻。

除此之外，通信协议与正常 LPUART 模式下的通信协议相似。此线路上的任何冲突都必须由软件管理（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。

注： *在 LPUART 通信中，当进行 1 个停止位配置时，RXNE 标志在停止位中间置 1。*

45.3.12 使用 DMA 和 LPUART 进行连续通信

LPUART 能够使用 DMA 进行连续通信。接收缓冲区和发送缓冲区的 DMA 请求是独立生成的。

注： *要确定是否支持 DMA 模式，请参见第 1455 页的第 44.4 节：USART 实现。*如果不支持 DMA，请按照*第 44.5.6 节中的说明使用 LPUSRT。当 FIFO 禁止时，可以将 LPUART_ISR 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

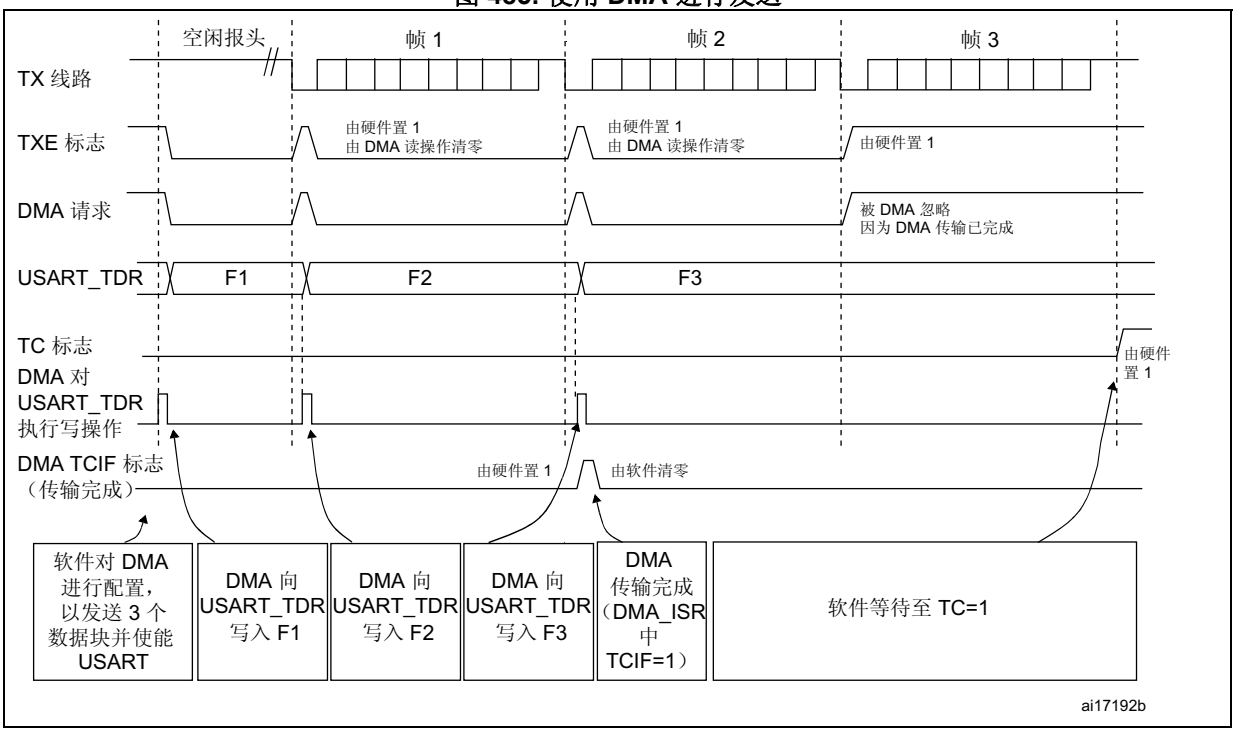
将 LPUART_CR3 寄存器中的 DMAT 位置 1 可以使能 DMA 模式进行发送。当 TXE 标志（使能 FIFO 模式时为 TXFNF 标志）置 1 时，可将数据从配置的 SRAM 区（通过 DMA 外设，请参见*“直接存储器访问控制器”部分*）加载到 LPUART_TDR 寄存器。要映射一个 DMA 通道以进行 LPUART 发送，请按以下步骤操作（x 表示通道编号）：

1. 在 DMA 控制寄存器中写入 LPUART_TDR 寄存器地址，将其配置为传输的目标地址。每次发生 TXE（使能 FIFO 模式时为 TXFNF）事件后，数据都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（使能 FIFO 模式时为 TXFNF）事件后，数据都从该存储区域加载到 LPUART_TDR 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. 通过将 LPUART_ICR 寄存器中的 TCCF 位置 1，将 LPUART_ISR 寄存器中的 TC 标志清零。
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 LPUART 通信已完成。在禁止 LPUART 或进入低功耗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保持清零状态，然后在最后一帧发送结束时由硬件置 1。

图 485. 使用 DMA 进行发送



注： *使能 FIFO 管理时，DMA 请求由发送 FIFO 未满（即 TXFNF = 1）触发。*

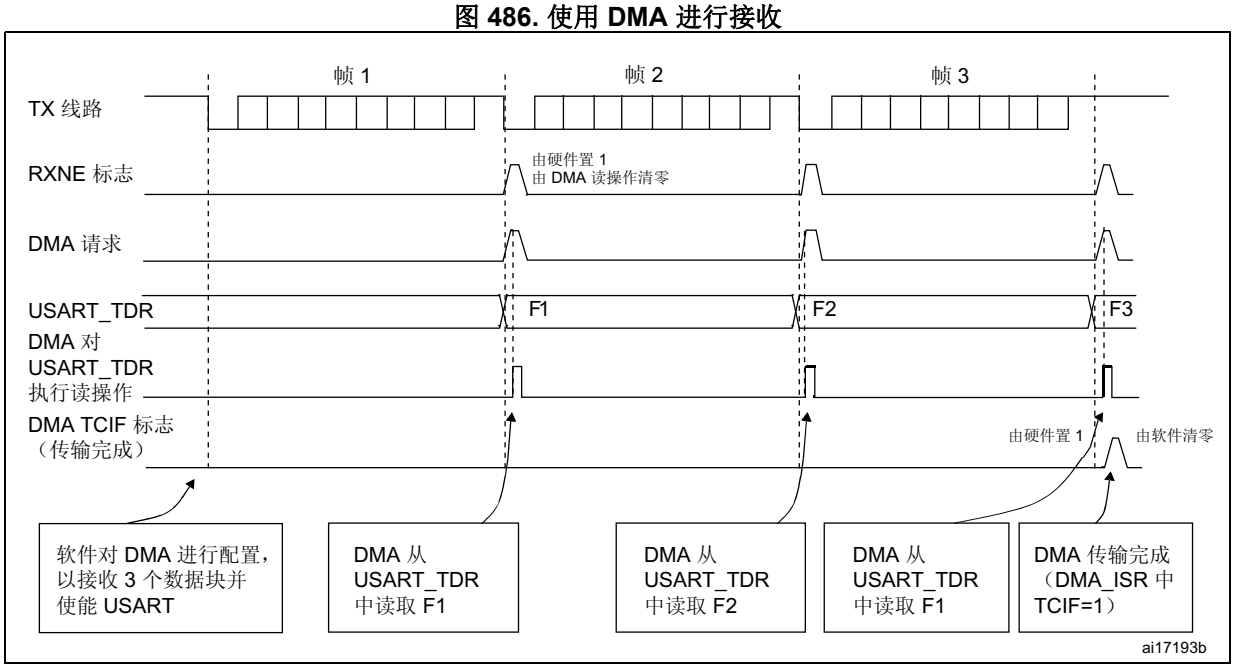
使用 DMA 进行接收

将 LPUART_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，会使用 DMA 将数据从 LPUART_RDR 寄存器加载到 SRAM 区域中（请参见 *“直接存储器访问控制器 (DMA)”部分*）。要映射一个 DMA 通道以进行 LPUART 接收，请按以下步骤操作：

1. 在 DMA 控制寄存器中写入 LPUART_RDR 寄存器地址，将其配置为传输的源地址。每次发生 RXNE（使能 FIFO 模式时为 RXFNE）事件后，数据都从该地址移动到存储器。
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（使能 FIFO 模式时为 RXFNE）事件后，数据都从 LPUART_RDR 加载到此存储区域。
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。

图 486. 使用 DMA 进行接收



注： *使能 FIFO 管理时，DMA 请求由接收 FIFO 非空（即 RXFNE = 1）触发。*

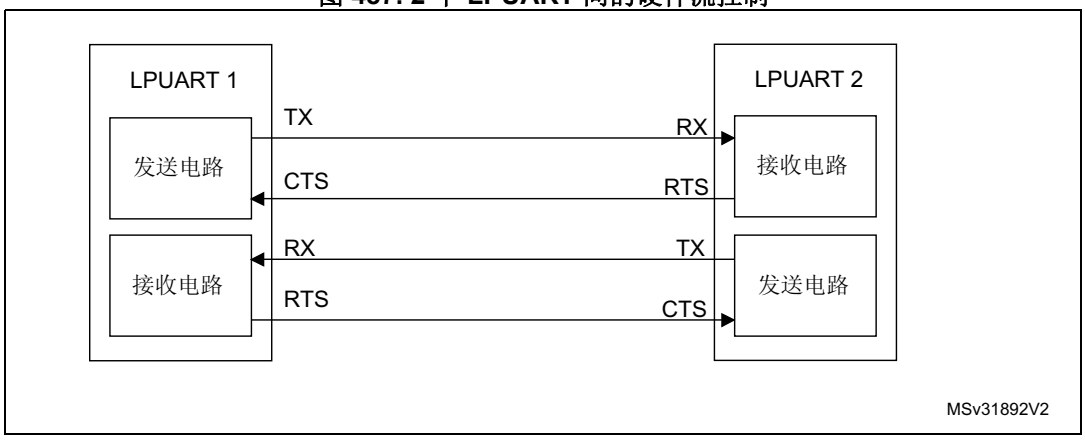
多缓冲区通信中的错误标志和中断生成

在多缓冲区通信模式下，如果事务中发生任何错误，都会在当前字节后将相应错误标志置 1。如果中断使能标志置 1，则会产生中断。在单字节接收过程中，与 RXNE（使能 FIFO 模式时为 RXFNE）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（LPUART_CR3 寄存器中的 EIE 位）；如果该位置 1，在发生其中任何一个错误时，都会在当前字节后使能中断。

45.3.13 RS232 硬件流控制和 RS485 驱动器使能

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 473* 显示了在这种模式下如何连接 2 个器件：

图 487. 2 个 LPUART 间的硬件流控制

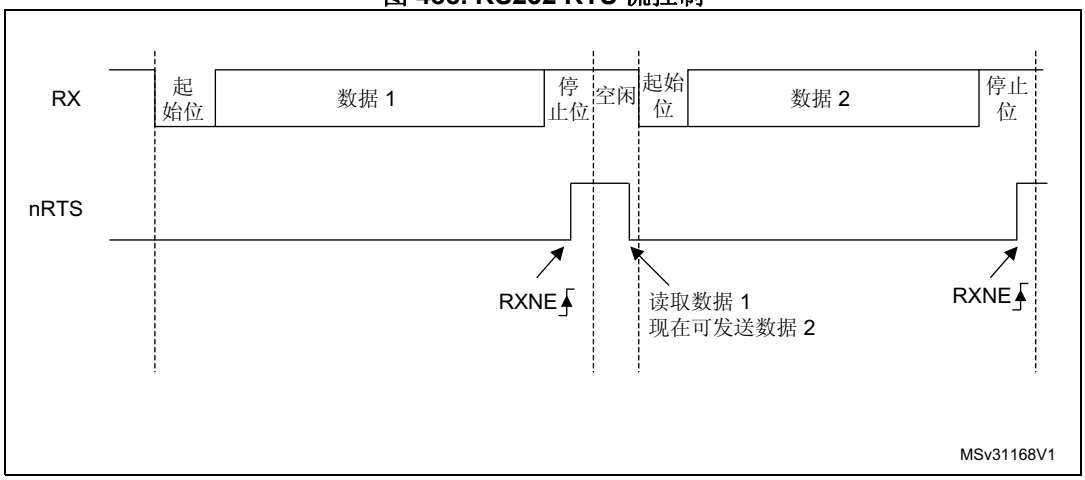


分别向 LPUART_CR3 寄存器中的 RTSE 位和 CTSE 位写入 1，可以分别使能 RS232 RTS 和 CTS 流控制。

RS232 RTS 流控制

如果使能 RTS 流控制 (RTSE=1)，只要 LPUART 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 488* 显示了在使能 RTS 流控制的情况下进行通信的示例。

图 488. RS232 RTS 流控制



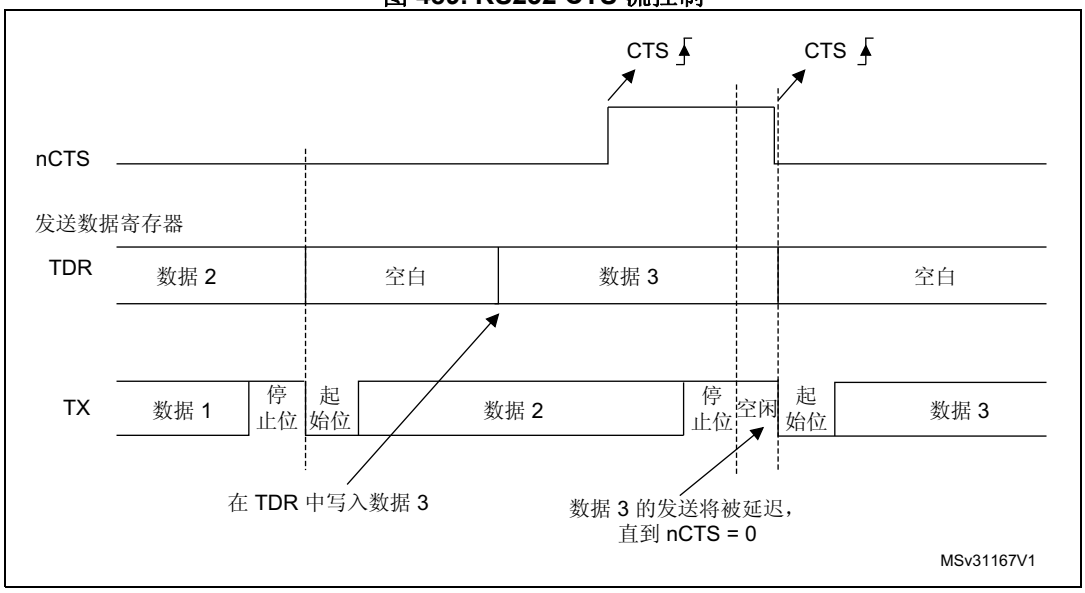
注： *如果使能 FIFO 模式，则仅当 RXFIFO 已满时，nRTS 才会变为无效。*

RS232 CTS 流控制

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/TXFE=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 LPUART_CR3 寄存器中的 CTSIE 位置 1，则会产生中断。*图 489* 显示了在使能 CTS 流控制的情况下进行通信的示例。

图 489. RS232 CTS 流控制



注： *为正常运行，必须在当前字符结束前至少 3 个 LPUART 时钟源周期内使能 nCTS。此外还应注意，当脉冲短于 2 个 PCLK 周期时无法将 CTSCF 标志置 1。*

RS485 驱动器使能

驱动器使能功能可通过将 LPUART_CR3 控制寄存器中的 DEM 位置 1 使能。这样便可通过 DE（驱动器使能）信号激活外部收发器控制。使能时间为激活 DE 信号与 START 位开始间的时间。可以通过 LPUART_CR1 控制寄存器中的 DEAT [4:0] 位域编程禁止时间。禁止时间为发送的消息中最后一个停止位结束与取消激活 DE 信号间的时间。可以通过 LPUART_CR1 控制寄存器中的 DEDT [4:0] 位域编程禁止时间。DE 信号的极性可使用 LPUART_CR3 控制寄存器中的 DEP 位配置。

LPUART 的 DEAT 和 DEDT 用 LPUART 时钟源 (f_{CK}) 周期表示：

- 驱动器使能有效时间等于
  - (1 + (DEAT x P)) x f_{CK}（如果 P # 0）
  - (1 + DEAT) x f_{CK}（如果 P = 0）
- 驱动器使能禁止时间等于
  - (1 + (DEDT x P)) x f_{CK}（如果 P # 0）
  - (1 + DEDT) x f_{CK}（如果 P = 0）

其中 P = BRR[20:11]

45.3.14 LPUART 低功耗管理

LPUART 具有高级低功耗模式功能，即使禁止 lpuart_pclk 时钟，也能正常传输数据。UESM 位置 1 时，LPUART 能够将 MCU 从低功耗模式唤醒。

对 usart_pclk 进行门控时，如果某些特定操作需要激活 **usart_pclk** 时钟，则 LPUART 会提供唤醒中断 (**usart_wkup**)：

- 禁止 FIFO 模式时

  必须激活 luart_pclk 时钟以清空 LPUART 数据寄存器。

  在这种情况下，lpuart_wkup 中断源为 RXNE 置“1”。RXNEIE 位必须在进入低功耗模式之前置 1。

- 使能 FIFO 模式时

  必须激活 luart_pclk 时钟以

  – 填充 TXFIFO

  – 或清空 RXFIFO

  在这种情况下，lpuart_wkup 中断源可以为：

  – RXFIFO 非空。此时，RXFNEIE 位必须在进入低功耗模式之前置 1。

  – RxFIFO 已满。此时，RXFFIE 位必须在进入低功耗模式之前置 1，接收到的数据量对应于 RXFIFO 大小，并且 RXFF 标志未置 1。

  – TXFIFO 为空。此时，TXFEIE 位必须在进入低功耗模式之前置 1。

  这允许在低功耗模式下发送/接收 TXFIFO/RXFIFO 中的数据。

  为了避免发生上溢/下溢错误以及在低功耗模式下发送/接收数据，lpuart_wkup 中断源可以是以下事件之一：

  – 达到 TXFIFO 阈值。此时，TXFTIE 位必须在进入低功耗模式之前置 1。

  – 达到 RXFIFO 阈值。此时，RXFTIE 位必须在进入低功耗模式之前置 1。

  例如，如果唤醒时间短于在线路上接收单个字节所需的时间，则应用可将阈值设为最大 RXFIFO 大小。

  使用 RXFIFO 已满、TXFIFO 为空、RXFIFO 非空和 RXFIFO/TXFIFO 阈值中断将 MCU 从低功耗模式唤醒时，可在低功耗模式下尽可能多地进行 LPUART 传输，这样有助于优化功耗。

或者，也可通过 WUS 位域选择特定 **lpuart_wkup** 中断。

检测到唤醒事件后，WUF 标志会由硬件置 1 并在 WUFIE 位置 1 时生成 **lpuart_wkup** 中断。在这种情况下，无需 **lpuart_wkup** 中断即可唤醒。将 WUF 置 1 便足以将 MCU 从低功耗模式唤醒。

注： *在进入低功耗模式之前，请确保未进行任何 LPUART 传输。检查 BUSY 标志不能确保在进行数据接收时绝不会进入低功耗模式。*

WUF 标志在检测到唤醒事件时置 1，而与 MCU 处于低功耗模式还是工作模式无关。

如果在初始化和使能接收器后立即进入低功耗模式，则必须校验 REACK 位以确保 LPUART 确实已使能。

当 DMA 用于接收时，它必须在进入低功耗模式前禁止，并在退出低功耗模式后重新使能。

如果使能 FIFO，则只有在使能静默模式的情况下才能在地址匹配时从低功耗模式唤醒。

使用静默模式和低功耗模式

如果 LPUART 在进入低功耗模式前处于静默模式：

- 不得使用空闲检测时从静默模式唤醒，因为空闲检测无法在低功耗模式下工作。
- 如果使用地址匹配时从静默模式唤醒，则低功耗模式唤醒源也必须是地址匹配。如果 RXNE 标志在进入低功耗模式时置 1，则接口将在地址匹配时和从低功耗模式唤醒时保持静默模式。

注： *使能 FIFO 管理时，静默模式可与从低功耗模式唤醒搭配使用，而且没有任何限制（即，上述关于静默和低功耗模式的两点仅在 FIFO 管理禁止时有效）。*

LPUART 内核时钟 lpuart_ker_ck 在低功耗模式下关闭时从低功耗模式唤醒

在低功耗模式下，如果在 LPUART 接收线上检测到下降沿时 lpuart_ker_ck 时钟处于关闭状态，则 LPUART 接口会借助 lpuart_ker_ck_req 信号请求开启 lpuart_ker_ck 时钟。lpuart_ker_ck 随后用来接收帧。

如果唤醒事件通过验证，则 MCU 将从低功耗模式唤醒，并会正常进行数据接收。

如果唤醒事件未通过验证，则 usart_ker_ck 会再次关闭，MCU 不会被唤醒并保持在低功耗模式下，且内核时钟请求被释放。

以下示例显示了将唤醒事件编程为“地址匹配检测”并禁止 FIFO 管理的情况。

图 490 所示为唤醒事件通过验证时的行为。

图 490. 唤醒事件通过验证（唤醒事件 = 地址匹配，禁止 FIFO）

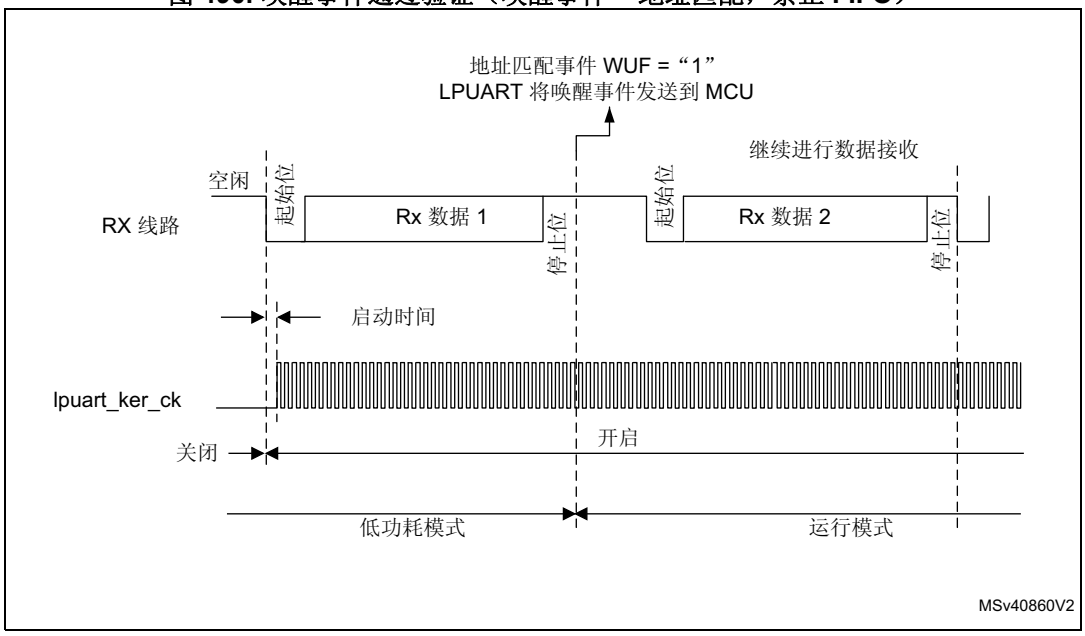
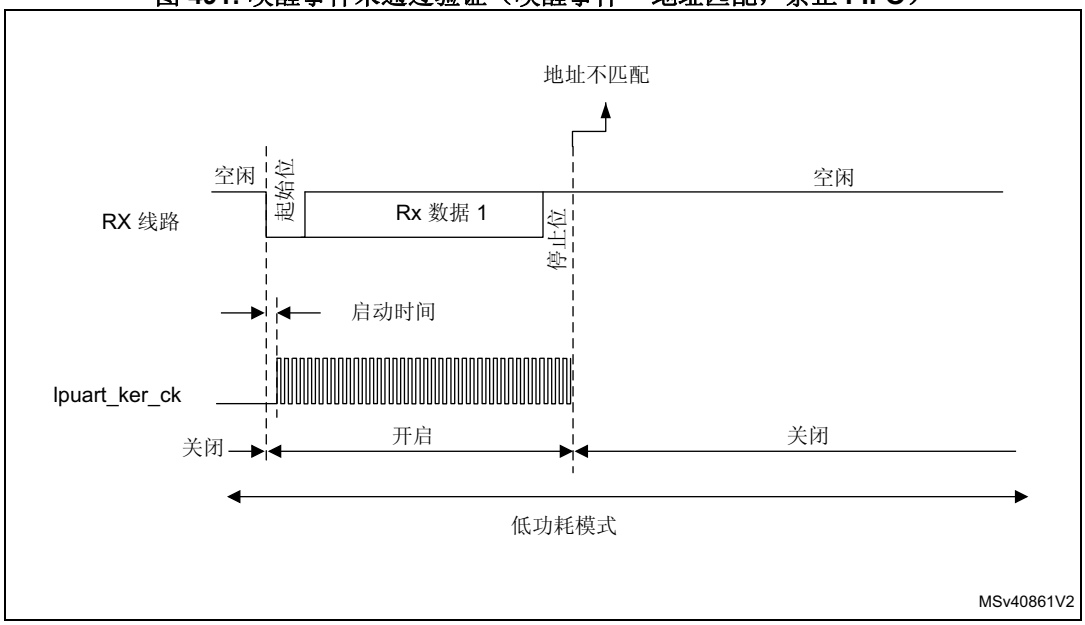


图 491 所示为唤醒事件未通过验证时的行为。

图 491. 唤醒事件未通过验证（唤醒事件 = 地址匹配，禁止 FIFO）



注： *将地址匹配或任一接收的帧用作唤醒事件时，如上两图适用。如果唤醒事件为起始位检测，则 LPUART 会在起始位结束时向 MCU 发送唤醒事件。*

确定允许从低功耗模式正确唤醒 MCU 的最大 LPUART 波特率

允许从低功耗模式正确唤醒 MCU 的最大波特率取决于唤醒时间参数（请参见器件数据手册）和 LPUART 接收器容差（请参见*第 45.3.8 节：LPUART 接收器对时钟偏差的容差*）。

举例来说：OVER8 = 0，M 位 = “01”，ONEBIT = 0，BRR [3:0] = 0000。

在这些条件下，根据*表 271：LPUART 接收器的容差*，LPUART 接收器的容差为 3.41%。

DTRA + DQUANT + DREC + DTCL + DWU < LPUART 接收器的容差

$D_{WUmax} = t_{WULPUART} / (11 \times T_{bit\ Min})$

$T_{bit\ Min} = = t_{WULPUART} / (11 \times D_{WUmax})$

其中 $t_{WULPUART}$ 为从低功耗模式唤醒的时间。

考虑一种理想情况：参数 DTRA、DQUANT、DREC 和 DTCL 为 0%，则 DWU 的最大值为 3.41%。实际上，我们至少需要考虑 lpuart_ker_ck 不精确的情况。

例如，如果将 HSI 用作 lpuart_ker_ck，HSI 的不精确度为 1%，则可以得到：

$t_{WULPUART}$ = 3 µs（仅提供数值作为参考；有关正确数值，请参见器件数据手册）。

D_{WUmax} = 3.41% - 1% = 2.41%。

$T_{bit\ min}$ = 3 µs/ (11 x 2.41%) = 11.32 µs。

结果，允许从低功耗模式正确唤醒的最大波特率为：1/11.32 µs = 88.36 K 波特。

45.4 LPUART 中断

有关所有 LPUART 中断请求的详细说明，请参见*表 267*。

表 273. LPUART 中断请求

中断事件	事件标志	使能控制位	中断清除方法	中断已激活	
				lpuart_it	lpuart_wkup
发送数据寄存器为空	TXE	TXEIE	TXE 在 TDR 中被写入数据时清零。	是	否
发送 FIFO 未满	TXFNF	TXFNFIE	TXFNF 在 TXFIFO 已满时清零。	是	否
发送 FIFO 为空	TXFE	TXFEIE	TXFE 在 TXFIFO 包含至少一个数据时清零，或通过将 TXFRQ 位置 1 来清零。	是	是
达到发送 FIFO 阈值	TXFT	TXFTIE	TXFT 在 TXFIFO 内容少于编程的阈值时由硬件清零。	是	是
CTS 中断	CTSIF	CTSIE	CTSIF 由软件通过将 CTSCF 位置 1 来清零。	是	否
发送完成	TC	TCIE	TC 在 TDR 中被写入数据时清零，或通过将 TCCF 位置 1 来清零。	是	否
接收数据寄存器不为空（已准备好读取数据）	RXNE	RXNEIE	RXNE 通过读取 RDR 或通过将 RXFRQ 位置 1 来清零。	是	是
接收 FIFO 非空	RXFNE	RXFNEIE	RXFNE 在 RXFIFO 为空时清零，或通过将 RXFRQ 位置 1 来清零。	是	是
接收 FIFO 已满	RXFF(1)	RXFFIE	RXFF 在 RXFIFO 至少包含一个数据时清零。	是	是
达到接收 FIFO 阈值	RXFT	RXFTIE	RXFT 在 RXFIFO 内容少于编程的阈值时由硬件清零。	是	是
检测到溢出错误	ORE	RXNEIE/ RXFNEIE	ORE 通过将 ORECF 位置 1 来清零。	是	否
检测到空闲线路	IDLE	IDLEIE	IDLE 通过将 IDLECF 位置 1 来清零。	是	否
奇偶校验错误	PE	PEIE	PE 通过将 PECF 位置 1 来清零。	是	否

表 273. LPUART 中断请求（续）

中断事件	事件标志	使能控制位	中断清除方法	中断已激活	
				lpuart_it	lpuart_wkup
多缓冲区通信中的噪声标志、溢出错误和帧错误。	NE 或 ORE 或 FE	EIE	NE 通过将 NFCF 位置 1 来清零。 ORE 通过将 ORECF 位置 1 来清零。 FE 标志通过将 FECF 位置 1 来清零。	是	否
字符匹配	CMF	CMIE	CMF 通过将 CMCF 位置 1 来清零。	是	否
从低功耗模式唤醒	WUF	WUFIE	WUF 通过将 WUCF 位置 1 来清零。	是	是
达到发送 FIFO 阈值	TXFT	TXFTIE	TXFT 在 TXFIFO 内容少于编程的阈值时由硬件清零。	是	是
达到接收 FIFO 阈值	RXFT	RXFTIE	RXFT 在 RXFIFO 内容少于编程的阈值时由硬件清零。	是	是

1. 如果 LPUART 接收 n+1 个数据（n 表示 RXFIFO 大小，RXFIFO 中有 n 个数据，LPUART_RDR 中有 1 个数据），则 RXFF 标志置 1。在停止模式下，未向 LPUART_RDR 提供时钟。因此，将不会对该寄存器进行写操作，一旦有 n 个数据被接收并写入到 RXFIFO，RXFF 中断将生效（RXFF 标志未置 1）。

45.5 LPUART寄存器

有关寄存器说明中使用的缩写，请参见*第 72 页的第 1.1 节*。

45.5.1 控制寄存器 1 (LPUART_CR1)

Control register 1

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
RXFFIE	TXFEIE	FIFOEN	M1	Res.	Res.	DEAT[4:0]					DEDT[4:0]				
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
Res.	CMIE	MME	M0	WAKE	PCE	PS	PEIE	TXEIE/TXFNFIE	TCIE	RXNEIE/RXFNEIE	IDLEIE	TE	RE	UESM	UE
	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31 **RXFFIE**：RXFIFO 变满时中断使能 (RXFIFO full interrupt enable)
此位由软件置 1 和清零。
0：禁止中断
1：当 LPUART_ISR 寄存器中的 RXFF=1 时，产生相应的 LPUART_ISR 中断
注： 禁止 FIFO 模式时，该位保留且必须保持复位值。

位 30 **TXFEIE**：TXFIFO 为空时中断使能 (TXFIFO empty interrupt enable)
此位由软件置 1 和清零。
0：禁止中断
1：当 LPUART_ISR 寄存器中的 TXFE=1 时，产生相应的 LPUART_ISR 中断
注： 禁止 FIFO 模式时，该位保留且必须保持复位值。

位 29 **FIFOEN**：FIFO 模式使能 (FIFO mode enable)
此位由软件置 1 和清零。
0：禁止 FIFO 模式。
1：使能 FIFO 模式。

位 28 **M1**：字长 (Word length)
此位必须与位 12 (M0) 搭配使用来确定字长度。该位由软件置 1 或清零。
M[1:0] = “00”： 1 个起始位，8 个数据位，n 个停止位
M[1:0] = “01”： 1 个起始位，9 个数据位，n 个停止位
M[1:0] = “10”： 1 个起始位，7 个数据位，n 个停止位
只有在禁止 LPUART (UE=0) 时才能写入此位。
注： 7 位数据长度模式下，不支持智能卡模式、LIN 主模式和自动波特率（0x7F 帧和 0x55 帧检测）。

位 27:26 保留，必须保持复位值。

位 25:21 **DEAT[4:0]**：驱动器使能使能时间 (Driver Enable assertion time)
该 5 位值用于定义激活 DE（启动器使能）信号与起始位开始间的时间。以 lpuart_ker_ck 时钟周期数表示。有关详细信息，请参见*第 44.5.20 节：RS232 硬件流控制和 RS485 驱动器使能*。
只有在禁止 LPUART (UE=0) 时才能写入此位域。

位 20:16 **DEDT[4:0]**：驱动器使能禁止时间 (Driver Enable deassertion time)

该 5 位值用于定义发送的消息中最后一个停止位结束与取消激活 DE（驱动器使能）信号间的时间。以 lpuart_ker_ck 时钟周期数表示。有关详细信息，请参见*第 45.3.13 节：RS232 硬件流控制和 RS485 驱动器使能*。

如果在 DEDT 时间内对 LPUART_TDR 寄存器执行写操作，则新数据仅在经过 DEDT 和 DEAT 时间后才会发送。

只有在禁止 LPUART (UE=0) 时才能写入此位域。

位 15 保留，必须保持复位值。

位 14 **CMIE**：字符匹配中断使能 (Character match interrupt enable)

此位由软件置 1 和清零。

0：禁止中断。

1：如果 LPUART_ISR 寄存器中的 CMF 位置 1，则生成 LPUART 中断。

位 13 **MME**：静默模式使能 (Mute mode enable)

此位用于激活 LPUART 的静默模式功能，此位置 1 时，LPUART 可按 WAKE 位定义的方式在活动模式与静默模式之间切换。该位由软件置 1 和清零。

0：接收器永久处于活动模式。

1：接收器可在静默模式和活动模式之间切换。

位 12 **M0**：字长 (Word length)

此位与位 28 (M1) 搭配使用来确定字长度。它由软件置 1 或清零（请参见位 28 (M1) 的说明）。

只有在禁止 LPUART (UE=0) 时才能写入此位。

位 11 **WAKE**：接收器唤醒方法 (Receiver wakeup method)

此位用于确定 LPUART 静默模式的唤醒方法。该位由软件置 1 或清零。

0：空闲线路

1：地址标记

只有在禁止 LPUART (UE=0) 时才能写入此位域。

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

只有在禁止 LPUART (UE=0) 时才能写入此位域。

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

只有在禁止 LPUART (UE=0) 时才能写入此位域。

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 LPUART_ISR 寄存器中的 PE=1 时，生成 LPUART 中断

位 7 **TXEIE/TXFNFIE**：发送数据寄存器为空/TXFIFO 未满中断使能 (Transmit data register empty/ TXFIFO not full interrupt enable)

此位由软件置 1 和清零。

0：禁止中断

1：当 LPUART_ISR 寄存器中 TXE/TXFNF =1 时，生成 LPUART 中断

位 6 **TCIE**：传输完成中断使能 (Transfer complete interrupt enable)

此位由软件置 1 和清零。

0：禁止中断

1：当 LPUART_ISR 寄存器中的 TC=1 时，生成 LPUART 中断

位 5 **RXNEIE/RXFNEIE**：接收数据寄存器非空/RXFIFO 非空中断使能 (Receive data register not empty/RXFIFO not empty interrupt enable)

此位由软件置 1 和清零。

0：禁止中断

1：当 LPUART_ISR 寄存器中的 ORE=1 或 RXNE/RXFNE=1 时，生成 LPUART 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 LPUART_ISR 寄存器中的 IDLE=1 时，生成 LPUART 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： *传送期间 TE 位上的低电平脉冲（“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）。为生成空闲字符，TE 不能立即写入 1。为确保所需的持续时间，软件可轮询 LPUART_ISR 寄存器中的 TEACK 位。*

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 **UESM**：停止模式下的 LPUART 使能 (LPUART enable in Stop mode)

当此位清零时，LPUART 无法将 MCU 从低功耗模式唤醒。

当此位置 1 时，LPUART 能够将 MCU 从低功耗模式唤醒，前提是 LPUART 时钟选择为 HSI 或 LSE（在 RCC 中）。

此位由软件置 1 和清零。

0：LPUART 无法将 MCU 从低功耗模式唤醒。

1：LPUART 能够将 MCU 从低功耗模式唤醒。当该功能激活时，LPUART 的时钟源必须是 HSI 或 LSE（请参见 RCC 一章）。

注： *建议在进入低功耗模式前将 UESM 位置 1，并在退出低功耗模式时将其清零。*

位 0 **UE**：LPUART 使能 (LPUART enable)

此位清零后，LPUART 预分频器和输出将立即停止，并丢弃当前操作。LPUART 的配置保留，但 LPUART_ISR 中的所有状态标志将被复位。此位由软件置 1 和清零。

0：禁止 LPUART 预分频器和输出，低功耗模式

1：使能 LPUART

注： *为进入低功耗模式而不在线路上生成错误，之前必须复位 TE 位，并且软件必须等待 LPUART_ISR 寄存器中的 TC 位置 1 后才能复位 UE 位。*

UE = 0 时也会复位 DMA 请求，因必须在复位 UE 位前禁止 DMA 通道。

45.5.2 控制寄存器 2 (LPUART_CR2)

Control register 2

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
ADD[7:4]				ADD[3:0]				Res.	Res.	Res.	Res.	MSBFIRST	DATAINV	TXINV	RXINV
rw	rw	rw	rw	rw	rw	rw	rw					rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SWAP	Res.	STOP[1:0]		Res.	Res.	Res.	Res.	Res.	Res.	Res.	ADDM7	Res.	Res.	Res.	Res.
rw		rw	rw								rw				

位 31:28 **ADD[7:4]**：LPUART 节点的地址 (Address of the LPUART node)

此位域用于指定 LPUART 节点的地址或要识别的字符代码。

它在多处理器通信时于静默模式或停止模式下使用，以通过 7 位地址标记检测唤醒 MCU。发送器发送字符的 MSB 应为 1。此位域还可用于正常接收和静默模式无效时的字符检测（例如，ModBus 协议中的块结束检测）。这种情况下，接收到的整个字符（8 位）将与 ADD[7:0] 值进行比较，如果匹配，CMF 标志将置 1。

仅在禁止接收 (RE = 0) 或禁止 LPUART (UE=0) 时才能写入该位域

位 27:24 **ADD[3:0]**：LPUART 节点的地址 (Address of the LPUART node)

此位域用于指定 LPUART 节点的地址或要识别的字符代码。

它在多处理器通信时于静默模式或低功耗模式下使用，以通过地址标记检测进行唤醒。

仅在禁止接收 (RE = 0) 或禁止 LPUART (UE=0) 时才能写入该位域。

位 23:20 保留，必须保持复位值。

位 19 **MSBFIRST**：最高有效位在前 (Most significant bit first)

此位由软件置 1 和清零。

0：发送/接收数据时位 0 在前，后跟起始位。

1：发送/接收数据时 MSB（位 7/8）在前，后跟起始位。

只有在禁止 LPUART (UE=0) 时才能写入此位域。

位 18 **DATAINV**：二进制数据反向 (Binary data inversion)

此位由软件置 1 和清零。

0：按正/正向逻辑发送/接收数据寄存器中的逻辑数据。（1=H，0=L）

1：按负/反向逻辑发送/接收数据寄存器中的逻辑数据。（1=L，0=H）。奇偶校验位也取反。

只有在禁止 LPUART (UE=0) 时才能写入此位域。

位 17 **TXINV**：TX 引脚有效电平反向 (TX pin active level inversion)

此位由软件置 1 和清零。

0：TX 引脚信号使用标准逻辑电平（V_{DD} = 1/空闲，Gnd = 0/标记）工作

1：对 TX 引脚信号值取反。（V_{DD} = 0/标记，Gnd=1/空闲）。

允许在 TX 线路上使用外部反相器。

只有在禁止 LPUART (UE=0) 时才能写入此位域。

位 16 **RXINV**：RX 引脚有效电平反向 (RX pin active level inversion)
此位由软件置 1 和清零。
0：RX 引脚信号使用标准逻辑电平（V_{DD} = 1/空闲，Gnd = 0/标记）工作。
1：对 RX 引脚信号值取反。（V_{DD} = 0/标记，Gnd=1/空闲）。
允许在 RX 线路上使用外部反相器。
只有在禁止 LPUART (UE=0) 时才能写入此位域。

位 15 **SWAP**：交换 TX/RX 引脚 (Swap TX/RX pins)
此位由软件置 1 和清零。
0：按标准引脚排列定义使用 TX/RX 引脚。
1：交换 TX 和 RX 引脚功能。允许在与另一个 UART 的交叉连接时工作。
只有在禁止 LPUART (UE=0) 时才能写入此位域。

位 14 保留，必须保持复位值。

位 13:12 **STOP[1:0]**：停止位 (STOP bits)
这些位用于编程停止位。
00：1 个停止位
01：保留
10：2 个停止位
11：保留
只有在禁止 LPUART (UE=0) 时才能写入此位域。

位 11:5 保留，必须保持复位值。

位 4 **ADDM7**：7 位地址检测/4 位地址检测 (7-bit Address Detection/4-bit Address Detection)
此位用于选择 4 位地址检测或 7 位地址检测。
0：4 位地址检测
1：7 位地址检测（在 8 位数据模式下）
只有在禁止 LPUART (UE=0) 时才能写入该位
注： 在 7 位和 9 位数据模式下，地址检测分别在 6 位和 8 位地址上完成（ADD[5:0] 和 ADD[7:0]）。

位 3:0 保留，必须保持复位值。

45.5.3 控制寄存器 3 (LPUART_CR3)

Control register 3

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
RXFTCFG			RXFTIE	RXFTCFG			Res.	TXFTIE	WUFIE	WUS[2:0]		Res.	Res.	Res.	Res.
rw			rw	rw				rw	rw	rw	rw				
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DEP	DEM	DDRE	OVRDIS	Res.	CTSIE	CTSE	RTSE	DMAT	DMAR	Res.	Res.	HDSEL	Res.	Res.	EIE
rw	rw	rw	rw		rw	rw	rw	rw	rw			rw			rw

位 31:29 **TXFTCFG**：TXFIFO 阈值配置 (TXFIFO threshold configuration)
000：TXFIFO 达到其深度的 1/8。
001：TXFIFO 达到其深度的 1/4。
110：TXFIFO 达到其深度的 1/2。
011：TXFIFO 达到其深度的 3/4。
100：TXFIFO 达到其深度的 7/8。
101：TXFIFO 变空。
其余组合：保留。

位 28 **RXFTIE**：RXFIFO 阈值中断使能 (RXFIFO threshold interrupt enable)。
此位由软件置 1 和清零。
0：禁止中断
1：当接收 FIFO 达到 RXFTCFG 中编程的阈值时，生成 LPUART 中断

位 27:25 **RXFTCFG**：接收 FIFO 阈值配置 (Receive FIFO threshold configuration)
000：接收 FIFO 达到其深度的 1/8。
001：接收 FIFO 达到其深度的 1/4。
110：接收 FIFO 达到其深度的 1/2。
011：接收 FIFO 达到其深度的 3/4。
100：接收 FIFO 达到其深度的 7/8。
101：接收 FIFO 已满。
其余组合：保留。

位 24 保留，必须保持复位值。

位 23 **TXFTIE**：TXFIFO 阈值中断使能 (TXFIFO threshold interrupt enable)
此位由软件置 1 和清零。
0：禁止中断
1：当 TXFIFO 达到 TXFTCFG 中编程的阈值时，生成 LPUART 中断

位 22 **WUFIE**：从低功耗模式唤醒中断使能 (Wakeup from low-power mode interrupt enable)
此位由软件置 1 和清零。
0：禁止中断
1：当 LPUART_ISR 寄存器中的 WUF=1 时，生成 LPUART 中断
注： *WUFIE 必须在进入低功耗模式前置 1。*
如果 LPUART 不支持从停止模式唤醒功能，该位保留且必须保持复位值。请参见第 44.4 节：USART 实现。

位 21:20 **WUS[1:0]**：从低功耗模式唤醒中断标志选择 (Wakeup from low-power mode interrupt flag selection)
该位域用于指定激活 WUF（从低功耗模式唤醒标志）的事件。
00：WUF 在地址匹配时激活（按 ADD[7:0] 和 ADDM7 所定义）
01：保留
10：WUF 在起始位检测时激活
11：WUF 在 RXNE 时激活
只有在禁止 LPUART (UE=0) 时才能写入此位域。
注： *如果 LPUART 不支持从停止模式唤醒功能，该位保留且必须保持复位值。请参见第 44.4 节：USART 实现。*

位 19:16 保留，必须保持复位值。

位 15 **DEP**：驱动器使能极性选择 (Driver enable polarity selection)

0：DE 信号高电平有效。

1：DE 信号低电平有效。

只有在禁止 LPUART (UE=0) 时才能写入此位。

位 14 **DEM**：驱动器使能模式 (Driver enable mode)

此位用于通过 DE 信号激活外部收发器控制。

0：禁止 DE 功能。

1：使能 DE 功能。DE 信号在 RTS 引脚上输出。

只有在禁止 LPUART (UE=0) 时才能写入此位。

位 13 **DDRE**：接收出错时的 DMA 禁止 (DMA Disable on Reception Error)

0：接收出错时不禁止 DMA。相应的错误标志置 1，但 RXNE 保持为 0 以防止上溢。因此，将不使能 DMA 请求，从而不会传送错误数据（无 DMA 请求），但会传送接收到的下一个正确数据。

1：接收出错后禁止 DMA。相应的错误标志以及 RXNE 均置 1。屏蔽 DMA 请求，直到错误标志清零。这意味着软件必须首先禁止 DMA 请求 (DMAR = 0) 或者将 RXNE 清零，然后再将错误标志清零。

只有在禁止 LPUART (UE=0) 时才能写入此位。

注： 接收错误包括：奇偶校验错误、帧错误或噪声错误。

位 12 **OVRDIS**：上溢禁止 (Overrun Disable)

此位用于禁止接收上溢检测。

0：接收新数据前未读取已接收的数据时，上溢错误标志 ORE 置 1。

1：禁止上溢功能。如果在 RXNE 标志仍置 1 时接收到新数据。

则 ORE 标志不会置 1，且新接收的数据会覆盖 LPUART_RDR 寄存器之前的内容。

只有在禁止 LPUART (UE=0) 时才能写入此位。

注： 此控制位用于检查通信流而不会读取数据。

位 11 保留，必须保持复位值。

位 10 **CTSIE**：CTS 中断使能 (CTS interrupt enable)

0：禁止中断

1：当 LPUART_ISR 寄存器中 CTSIF = 1 时，产生中断

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。

只有在禁止 LPUART (UE=0) 时才能写入该位。

位 8 **RTSE**：RTS 使能 (RTS enable)

0：禁止 RTS 硬件流控制。

1：使能 RTS 输出，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（拉至 0）。

只有在禁止 LPUART (UE=0) 时才能写入此位。

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 5:4 保留，必须保持复位值。

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
只有在禁止 LPUART (UE=0) 时才能写入此位。

位 2:1 保留，必须保持复位值。

位 0 **EIE**：错误中断使能 (Error interrupt enable)
如果发生帧错误、上溢错误或出现噪声标志（LPUART_ISR 寄存器中 FE = 1 或 ORE = 1 或 NE = 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
LPUART_ISR 寄存器中的 FE=1 或 ORE=1 或 NE=1 时生成中断。

45.5.4 波特率寄存器 (LPUART_BRR)

Baud rate register

只有在禁止 LPUART (UE=0) 时才能写入此寄存器。在自动波特率检测模式下，该位由硬件自动更新。

偏移地址：0x0C

复位值：0x0000 0000

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

位 31:20 保留，必须保持复位值。

位 19:0 **BRR[19:0]**

注： *禁止在 LPUART_BRR 寄存器中写入小于 0x300 的值。*

如果 LPUART_BRR 必须 ≥ 0x300 且 LPUART_BRR 为 20 位，则使用高 fck 值生成高波特率时应十分谨慎。fck 必须在 [3 x 波特率到 4096 x 波特率] 的范围内。

45.5.5 请求寄存器 (LPUART_RQR)

Request 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
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	
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TXFRQ	RXFRQ	MMRQ	SBKRQ	Res.
											w	w	w	w	

位 31:5 保留，必须保持复位值。

位 4 **TXFRQ**：发送数据刷新请求 (Transmit data flush request)

该位在 FIFO 模式使能时使用。TXFRQ 位置 1 以清空整个 FIFO。这会将标志 TXFE（TXFIFO 为空，LPUART_ISR 寄存器中的位 23）置 1。

注： 在 FIFO 模式下，TXFNF 标志在清空请求期间复位，直到 TxFIFO 为空，以确保数据寄存器中没有写入数据。

位 3 **RXFRQ**：接收数据刷新请求 (Receive data flush request)

向该位写入 1 时会将 RXNE 标志清零。

这可以丢弃接收的数据而不对其执行读取操作，并避免发生上溢。

位 2 **MMRQ**：静默模式请求 (Mute mode request)

向此位写入 1 可将 LPUART 置于静默模式，并将 RWU 标志复位。

位 1 **SBKRQ**：发送中断请求 (Send break request)

向此位写入 1 可将 SBKF 标志置 1 并在发送设备可用后立即请求在线路上发送 BREAK。

注： 如果应用需要在之前插入的所有数据（包括尚未发送的数据）后发送中断字符，软件应等到 TXE 标志使能后将 SBKRQ 位置 1。

位 0 保留，必须保持复位值。

45.5.6 中断和状态寄存器 (LPUART_ISR)

Interrupt and status register

偏移地址：0x1C

复位值：0x0XX0 00C0

禁用 FIFO 时，XX = 00

使能 FIFO 模式且禁止智能卡模式时，XX = 08

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	TXFT	RXFT	Res.	RXFF	TXFE	REACK	TEACK	WUF	RWU	SBKF	CMF	BUSY
				r	r		r	r	r	r	r	r	r	r	r

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	CTS	CTSIF	Res.	TXE/TXFNF	TC	RXNE/RXFNE	IDLE	ORE	NE	FE	PE
					r	r		r	r	r	r	r	r	r	r

位 31:28 保留，必须保持复位值。

位 27 **TXFT**：TXFIFO 阈值标志 (TXFIFO threshold flag)

当 TXFIFO 达到在 LPUART_CR3 寄存器的 TXFTCFG 中编程的阈值时（即 TXFIFO 包含 TXFTCFG 个空位置），该位由硬件置 1。如果 LPUART_CR3 寄存器中的 TXFTIE 位 =1（位 31），则会生成中断。

0：TXFIFO 未达到编程的阈值。

1：TXFIFO 已达到编程的阈值。

位 26 **RXFT**：RXFIFO 阈值标志 (RXFIFO threshold flag)

当 RXFIFO 达到在 LPUART_CR3 寄存器的 RXFTCFG 中编程的阈值时（即接收 FIFO 包含 RXFTCFG 个数据），该位由硬件置 1。如果 LPUART_CR3 寄存器中的 RXFTIE 位 =1（位 27），则会生成中断。

0：接收 FIFO 未达到编程的阈值。

1：接收 FIFO 已达到编程的阈值。

位 25 保留，必须保持复位值。

位 24 **RXFF**：RXFIFO 已满 (RXFIFO Full)

当接收到的数据量对应于 RXFIFO 大小 + 1（RXFIFO 已满 + LPUART_RDR 寄存器中的 1 个数据）时，该位由硬件置 1。

如果 LPUART_CR1 寄存器中 RXFFIE 位 = 1，则会生成中断。

0：RXFIFO 未满。

1：RXFIFO 已满。

位 23 **TXFE**：TXFIFO 为空 (TXFIFO Empty)

当 TXFIFO 为空时，该位由硬件置 1。当 TXFIFO 包含至少一个数据时，该标志被清零。也可以通过向 LPUART_RQR 寄存器中的位 TXFRQ（位 4）写入 1 将 TXFE 标志置 1。

如果 LPUART_CR1 寄存器中的 TXFEIE 位 =1（位 30），则会生成中断。

0：TXFIFO 非空。

1：TXFIFO 为空。

位 22 **REACK**：接收使能确认标志 (Receive enable acknowledge flag)

LPUART 采用接收使能值时，通过硬件将此位置 1/复位。

此位可用于验证 LPUART 是否准备好在进入低功耗模式前接收数据。

注： 如果 LPUART 不支持从停止模式唤醒功能，该位保留且保持复位值。

位 21 **TEACK**：发送使能确认标志 (Transmit enable acknowledge flag)

LPUART 采用发送使能值时，通过硬件将此位置 1/复位。

通过写入 TE=0 生成空闲帧请求，然后在 LPUART_CR1 寄存器中写入 TE=1 以遵循 TE=0 最短周期时，可使用此位。

位 20 **WUF**：从低功耗模式唤醒标志 (Wakeup from low-power mode flag)

当检测到唤醒事件时，此位由硬件置 1。事件通过 WUS 位域定义。通过向 LPUART_ICR 寄存器中的 WUCF 写入 1，此位由软件清零。

如果 LPUART_CR3 寄存器中 WUFIE=1，则会生成中断。

注： 当 UESM 清零时，WUF 标志也清零。

如果 LPUART 不支持从停止模式唤醒功能，该位保留且保持复位值。

位 19 **RWU**：*接收器从静默模式唤醒 (Receiver wakeup from Mute mode)*

此位指示 LPUART 是否处于静默模式。当识别出唤醒/静默序列时，此位由硬件清零/置 1。静默模式控制序列（地址或 IDLE）通过 LPUART_CR1 寄存器中的 WAKE 位选择。

当选择 IDLE 模式下唤醒时，该位只能通过用软件向 LPUART_RQR 寄存器中的 MMRQ 位写 1 的方式置 1。

0：接收器处于工作模式

1：接收器处于静默模式

注： 如果 LPUART 不支持从停止模式唤醒功能，该位保留且保持复位值。

位 18 **SBKF**：发送中断标志 (Send break flag)

此位指示已请求发送中断字符。通过将 1 写入 LPUART_CR3 寄存器中的 SBKRQ 位，此位由软件置 1。此位在中断发送的停止位期间由硬件自动复位。

0：不发送中断字符

1：将发送中断字符

位 17 **CMF**：字符匹配标志 (Character match flag)

接收到由 ADD[7:0] 定义的字符后由硬件将此位置 1。通过向 LPUART_ICR 寄存器中的 CMCF 写入 1，此位由软件清零。

如果 LPUART_CR1 寄存器中 CMIE=1，则会生成中断。

0：未检测到字符匹配

1：检测到字符匹配

位 16 **BUSY**：忙标志 (Busy flag)

此位由硬件置 1 和复位。当 RX 线路上正在进行通信（成功检测到起始位）时有效。在接收结束（成功或失败）时复位。

0：LPUART 处于空闲状态（无接收）

1：正在接收

位 15:11 保留，必须保持复位值。

位 10 **CTS**：CTS 标志 (CTS flag)

此位由硬件置 1/复位。此位是对 nCTS 输入引脚的状态取反。

0：nCTS 线置 1

1：nCTS 线复位

注： 如果不支持硬件流控制功能，该位保留且保持复位值。

位 9 **CTSIF**：CTS 中断标志 (CTS interrupt flag)

如果 CTSE 位置 1，当 nCTS 输入切换时，此位由硬件置 1。通过将 1 写入 LPUART_ICR 寄存器中的 CTSCF 位，此位由软件清零。

如果 LPUART_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： 如果不支持硬件流控制功能，该位保留且保持复位值。

位 8 保留，必须保持复位值。

位 7 **TXE/TXFNF**：发送数据寄存器为空/TXFIFO 未满 (Transmit data register empty/TXFIFO not full)

如果禁止 FIFO 模式，则当 LPUART_TDR 寄存器的内容已传输到移位寄存器时，TXE 由硬件置 1。通过对 LPUART_TDR 寄存器执行写入操作将该位清零。

如果使能 FIFO 模式，TXFNF 在 TXFIFO 未满时由硬件置 1，并因此可向 LPUART_TDR 中写入数据。每次对 LPUART_TDR 进行写操作都会将数据置于 TXFIFO 中。该标志保持置 1，直到 TXFIFO 已满。当 TXFIFO 已满时，该标志清零，表示不能向 LPUART_TDR 写入数据。

注：在清空请求期间，TXFNF 保持复位，直到 TXFIFO 为空。发送清空请求（通过将 TXFRQ 位置 1）后，应先检查 TXFNF 标志，然后再写入 TXFIFO（TXFNF 和 TXFE 将同时置 1）。

如果 LPUART_CR1 寄存器中 TXEIE/TXFNFIE 位 = 1，则会生成中断。

0：数据寄存器已满/发送 FIFO 已满。

1：数据寄存器/发送 FIFO 未满。

注： 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/TXFF 置 1，则该位由硬件置 1。如果 LPUART_CR1 寄存器中 TCIE = 1，则会生成中断。通过向 LPUART_ICR 寄存器中的 TCCF 写入 1 或向 LPUART_TDR 寄存器执行写操作，此位由软件清零。

如果 LPUART_CR1 寄存器中 TCIE = 1，则会生成中断。

0：传送未完成

1：传送已完成

注： 如果 TE 位复位且无任何发送正在进行，TC 位会立即置 1。

位 5 **RXNE/RXFNE**：读取数据寄存器非空/RXFIFO 非空 (Read data register not empty/RXFIFO not empty)

当 LPUART_RDR 移位寄存器的内容已传输到 LPUART_RDR 寄存器时，RXNE 位由硬件置 1。通过对 LPUART_RDR 寄存器执行读入操作将该位清零。

也可以通过向 LPUART_RQR 寄存器中的 RXFRQ 写入 1 来将 RXNE 标志清零。

RXFIFO 非空时，RXFNE 位由硬件置 1，并因此可以从 LPUART_RDR 寄存器读取数据。每次对 LPUART_RDR 进行读操作都会在 RXFIFO 中释放一个位置。它在 RXFIFO 为空时清零。

也可以通过将 LPUART_RQR 寄存器中的 RXFRQ 位置 1 将 RXNE/RXFNE 标志位清零。

如果 LPUART_CR1 寄存器中 RXNEIE/RXFNEIE = 1，则会生成中断。

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 LPUART_CR1 寄存器中 IDLEIE=1，则会生成中断。通过向 LPUART_ICR 寄存器中的 IDLECF 写入 1，此位由软件清零。

0：未检测到空闲线路

1：检测到空闲线路

注： 直到 RXNE 位已置 1 时（即，当出现新的空闲线路时）IDLE 位才会被再次置 1。

使能静默模式 (MME=1) 后，如果 LPUART 未静默 (RWU=0)，则 IDLE 置 1，无论是否通过 WAKE 位选择了静默模式。如果 RWU=1，IDLE 不置 1。

位 3 **ORE**：溢出错误 (Overrun error)

在 RXNE =“1”（使能 FIFO 模式时为 RXFF =“1”）的情况下，当移位寄存器中当前正在接收的数据准备好传输到 LPUART_RDR 寄存器且 RXNE=1 时（如果使能 FIFO 模式则为 RXFF = 1），此位由硬件置 1。通过向 LPUART_ICR 寄存器中的 ORECF 写入 1，此位由软件清零。

如果 LPUART_CR1 寄存器中的 RXNEIE/RXFNEIE=1 或 EIE = 1，则会生成中断。

0：无溢出错误

1：检测到溢出错误

注： *当此位置 1 时，LPUART_RDR 寄存器的内容不会丢失，但移位寄存器会被覆盖。EIE 位置 1 后，如果在多缓冲区通信中 ORE 标志置 1，则会生成中断。*

LPUART_CR3 寄存器中的 OVRDIS 位置 1 时，此位将被永久强制清零（无上溢检测）。

位 2 **NE**：START 位噪声检测标志 (Start bit noise detection flag)

当在接收的帧的起始位上检测到噪声时，此位由硬件置 1。通过向 LPUART_ICR 寄存器中的 NFCF 写入 1，此位由软件清零。

0：未检测到噪声

1：检测到噪声

注： *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/RXFNE 位出现的时间相同。EIE 位置 1 后，如果在多缓冲区通信中 NE 标志置 1，则会生成中断。*

在 FIFO 模式下，此错误与 LPUART_RDR 中的字符相关联。

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。通过向 LPUART_ICR 寄存器中的 FECF 写入 1，此位由软件清零。

在智能卡模式下发送数据时，如果在达到最大发送尝试次数后仍未成功（智能卡向数据帧发送 NACK 信号），则此位置 1。

如果 LPUART_CR1 寄存器中 EIE = 1，则会生成中断。

0：未检测到帧错误

1：检测到帧错误或中断字符

注： *在 FIFO 模式下，此错误与 LPUART_RDR 中的字符相关联。*

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。通过向 LPUART_ICR 寄存器中的 PECF 写入 1，此位由软件清零。

如果 LPUART_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

注： *在 FIFO 模式下，此错误与 LPUART_RDR 中的字符相关联。*

45.5.7 中断标志清零寄存器 (LPUART_ICR)

Interrupt flag clear 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
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WUCF	Res.	Res.	CMCF	Res.
											w			w	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	CTSCF	Res.	Res.	TCCF	Res.	IDLECF	ORECF	NECF	FECF	PECF
						w			w		w	w	w	w	w

位 31:21 保留，必须保持复位值。

位 20 **WUCF**：从低功耗模式唤醒清零标志 (Wakeup from low-power mode clear flag)
将 1 写入此位时，LPUART_ISR 寄存器中 WUF 标志将清零。

注： *如果 LPUART 不支持从停止模式唤醒功能，该位保留且保持复位值。请参见第 44.4 节：USART 实现。*

位 19:18 保留，必须保持复位值。

位 17 **CMCF**：字符匹配清零标志 (Character match clear flag)
将 1 写入此位时，LPUART_ISR 寄存器中 CMF 标志将清零。

位 16:10 保留，必须保持复位值。

位 9 **CTSCF**：CTS 清零标志 (CTS clear flag)
将 1 写入此位时，LPUART_ISR 寄存器中 CTSIF 标志将清零。

位 8:7 保留，必须保持复位值。

位 6 **TCCF**：发送完成清零标志 (Transmission complete clear flag)
将 1 写入此位时，LPUART_ISR 寄存器中 TC 标志将清零。

位 5 保留，必须保持复位值。

位 4 **IDLECF**：检测到空闲线路清零标志 (Idle line detected clear flag)
将 1 写入此位时，LPUART_ISR 寄存器中 IDLE 标志将清零。

位 3 **ORECF**：上溢错误清零标志 (Overrun error clear flag)
将 1 写入此位时，LPUART_ISR 寄存器中 ORE 标志将清零。

位 2 **NECF**：检测到噪声清零标志 (Noise detected clear flag)
将 1 写入此位时，LPUART_ISR 寄存器中 NE 标志将清零。

位 1 **FECF**：帧错误清零标志 (Framing error clear flag)
将 1 写入此位时，LPUART_ISR 寄存器中 FE 标志将清零。

位 0 **PECF**：奇偶校验错误清零标志 (Parity error clear flag)
将 1 写入此位时，LPUART_ISR 寄存器中 PE 标志将清零。

45.5.8 接收数据寄存器 (LPUART_RDR)

Receive data 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
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.	Res.	Res.	Res.	Res.	Res.	RDR[8:0]								
							r	r	r	r	r	r	r	r	r

位 31:9 保留，必须保持复位值。

位 8:0 **RDR[8:0]**：接收数据值 (Receive data value)

包含接收到的数据字符。

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（请参见*图 478*）。

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。

45.5.9 发送数据寄存器 (LPUART_TDR)

Transmit data register

偏移地址：0x28

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	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.	Res.	Res.	Res.	Res.	Res.	TDR[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 **TDR[8:0]**：发送数据值 (Transmit data value)

包含要发送的数据字符。

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（请参见*图 478*）。

在使能奇偶校验位的情况下（LPUART_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。

注： 只能在 TXE/TXFNF=1 时写入此寄存器。

45.5.10 预分频器寄存器 (LPUART_PRESC)

Prescaler register

只有在禁止 LPUART (UE=0) 时才能写入此寄存器。

偏移地址：0x2C

复位值：0x0000 0000

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

位 31:4 保留，必须保持复位值。

位 3:0 **PRESCALER[3:0]**：时钟预分频器 (Clock prescaler)

LPUART 输入时钟可通过预分频器进行分频：

0000：输入时钟未分频
0001：输入时钟 2 分频
0010：输入时钟 4 分频
0011：输入时钟 6 分频
0100：输入时钟 8 分频
0101：输入时钟 10 分频
0110：输入时钟 12 分频
0111：输入时钟 16 分频
1000：输入时钟 32 分频
1001：输入时钟 64 分频
1010：输入时钟 128 分频
1011：输入时钟 256 分频
其余组合：保留。

注： 为 PRESCALER 编程不允许的值时，编程的预分频值将为“1011”，即输入时钟除以 256。

45.5.11 LPUART 寄存器映射

下表提供了 LPUART 寄存器映射和复位值。

表 274. LPUART 寄存器映射和复位值

偏移	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	LPUART_CR1	RXFFIE	TXFEIE	FIFOEN	M1	Res.	Res.	DEAT4	DEAT3	DEAT2	DEAT1	DEAT0	DEDT4	DEDT3	DEDT2	DEDT1	DEDT0	Res.	CMIE	MME	M	WAKE	PCE	PS	PEIE	TXEIE	TCIE	RXNEIE	IDLEIE	TE	RE	UESM	UE
	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
0x04	LPUART_CR2	ADD[7:4]				ADD[3:0]				Res.	Res.	Res.	Res.	MSBFIRST	DATAINV	TXINV	RXINV	SWAP	Res.	STOP [1:0]		Res.	Res.	Res.	Res.	Res.	Res.	Res.	ADDM7	Res.	Res.	Res.	Res.
	Reset value	0	0	0	0	0	0	0	0					0	0	0	0	0		0	0								0				
0x08	LPUART_CR3	TXFT CFG			RXFTIE	RXFT CFG			Res.	TXFTIE	WUFIE	WUS [1:0]		Res.	Res.	Res.	Res.	DEP	DEM	DDRE	OVRDIS	Res.	CTSIE	CTSE	RTSE	DMAT	DMAR	Res.	Res.	HDSEL	Res.	Res.	EIE
	Reset value	0	0	0	0	0	0	0		0	0	0	0					0	0	0	0		0	0	0	0	0			0			0
0x0C	LPUART_BRR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BRR[19:0]																			
	Reset value																		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x10-0x14	Reserved																																
0x18	LPUART_RQR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TXFRQ	RXFRQ	MMRQ	SBKRQ	Res.
	Reset value																												0	0	0	0	
0x1C	LPUART_ISR	Res.	Res.	Res.	Res.	TXFT	RXFT	Res.	RXFF	TXFF	REACK	TEACK	WUF	RWU	SBKF	CMF	BUSY	Res.	Res.	Res.	Res.	Res.	CTS	CTSIF	Res.	TXE/TXFNF	TC	RXNE	IDLE	ORE	NE	FE	PE
	Reset value					0	0		0	0	0	0	0	0	0	0	0						0	0		1	0	0	0	0	0	0	0
0x20	LPUART_ICR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WUCF	Res.	Res.	CMCF	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CTSCF	Res.	Res.	TCCF	Res.	IDLECF	ORECF	NECF	FECF	PECF
	Reset value												0			0								0			0		0	0	0	0	0
0x24	LPUART_RDR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RDR[8:0]								
	Reset value																								0	0	0	0	0	0	0	0	0
0x28	LPUART_TDR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TDR[8:0]								
	Reset value																								0	0	0	0	0	0	0	0	0
0x2C	LPUART_PRESC	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PRESCALER[3:0]			
	Reset value																													0	0	0	0

有关寄存器边界地址的信息，请参见*第 78 页的第 2.2.2 节*。

46 串行外设接口 (SPI)

46.1 简介

SPI 接口可用于使用 SPI 协议或与外部器件进行通信。SPI 模式可通过软件进行选择。器件复位后默认选择 SPI Motorola 模式。

串行外设接口 (SPI) 协议支持与外部器件进行半双工、全双工和单工同步串行通信。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。

46.2 SPI 主要特性

- 主模式或从模式操作
- 基于三条线的全双工同步传输
- 基于双线的半双工同步传输，其中一条可作为双向数据线
- 基于双线的单工同步传输，其中一条可作为单向数据线
- 数据大小可选择 4 位到 16 位
- 多主模式功能
- 8 个主模式波特率预分频器，可达 $f_{PCLK}/2$
- 从模式频率可达 $f_{PCLK}/2$
- 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
- 支持 SPI Motorola 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和上溢标志
- CRC 错误标志
- 具有 DMA 功能的两个 32 位内置 Rx 和 Tx FIFO
- 支持 SPI TI 模式

46.3 SPI 实现

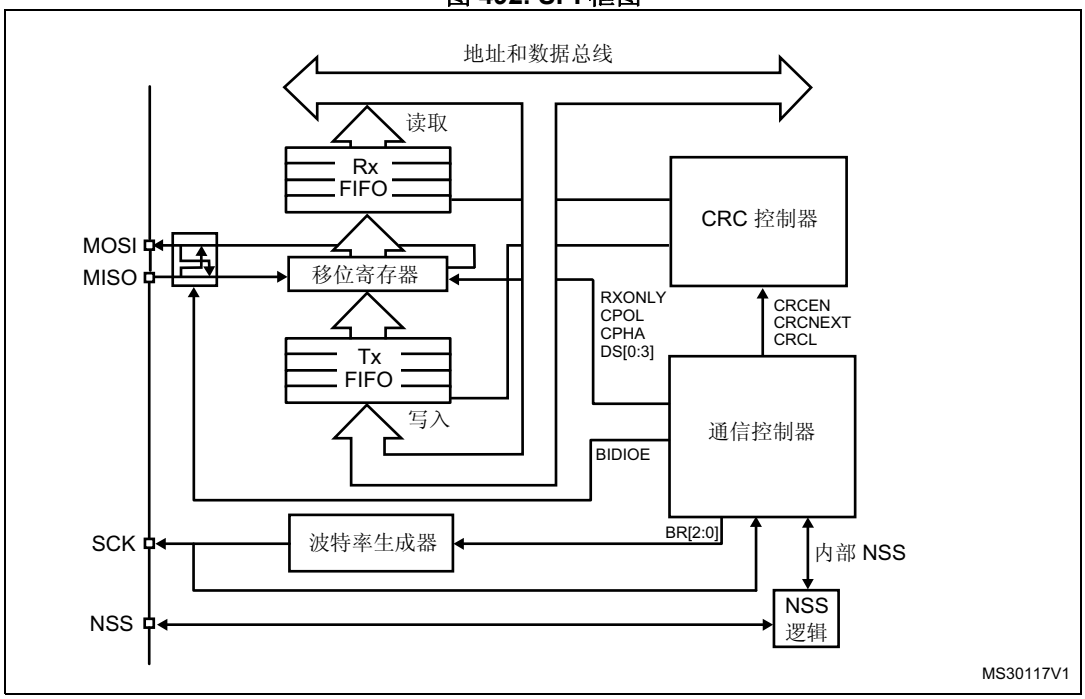
本手册介绍了 SPI/I2S 在器件 STM32L4Rxxx 和 STM32L4Sxxx 中的实现。

46.4 SPI 功能说明

46.4.1 概述

SPI 支持在 MCU 与外部器件之间进行同步串行通信。应用软件可通过轮询状态标志或使用专用 SPI 中断对通信进行管理。SPI 的主要组件及其交互方式如下面的*图 492* 框图所示。

图 492. SPI 框图



四个 I/O 引脚专用于与外部器件进行 SPI 通信。

- **MISO**：主输入/从输出数据。通常情况下，此引脚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。通常情况下，此引脚用于在主模式下发送数据和在从模式下接收数据。
- **SCK**：SPI 主器件的串行时钟输出引脚以及 SPI 从器件的串行时钟输入引脚。
- **NSS**：从器件选择引脚。根据 SPI 和 NSS 设置，该引脚可用于：
  - 选择单个从器件以进行通信
  - 同步数据帧或
  - 检测多个主器件之间是否存在冲突

  详细信息，请参见*第 46.4.5 节：从器件选择 (NSS) 引脚管理*。

SPI 总线支持一个主器件与一个或多个从器件之间进行通信。该总线至少由两条线构成——一条用于时钟信号，另一条用于同步数据传输。其他信号可以根据 SPI 节点间的数据交换及其从器件选择信号管理进行添加。

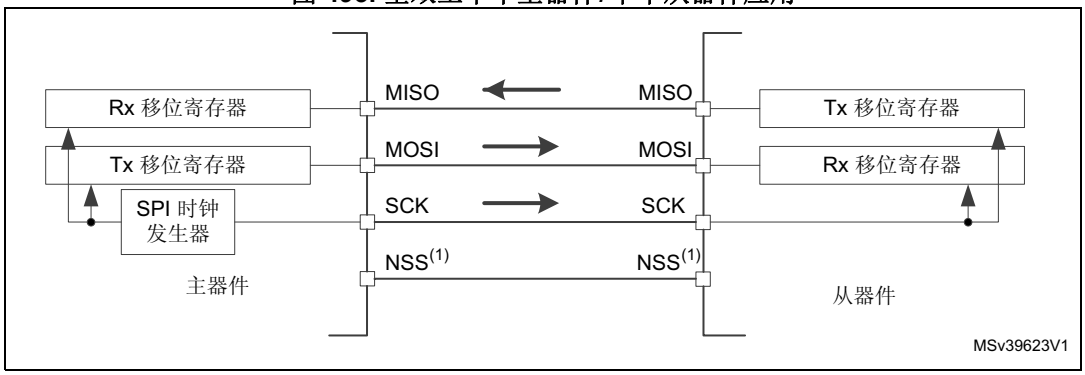
46.4.2 一个主器件和一个从器件之间的通信

SPI 支持 MCU 基于目标器件和应用要求使用不同的配置进行通信。这些配置使用 2 条或 3 条线（通过软件 NSS 管理），也可以使用 3 条或 4 条线（通过硬件 NSS 管理）。通信始终由主器件发起。

全双工通信

默认情况下，SPI 配置为进行全双工通信。在这种配置下，主器件和从器件的移位寄存器通过 MOSI 和 MISO 引脚之间的两条单向线连接。在 SPI 通信过程中，数据随主器件提供的 SCK 时钟边沿同步移位。主器件通过 MOSI 线将待发送的数据发送给从器件，通过 MISO 线从从器件接收数据。当数据帧传输完成时（所有位均移出），主器件和从器件之间即完成信息交换。

图 493. 全双工单个主器件 / 单个从器件应用

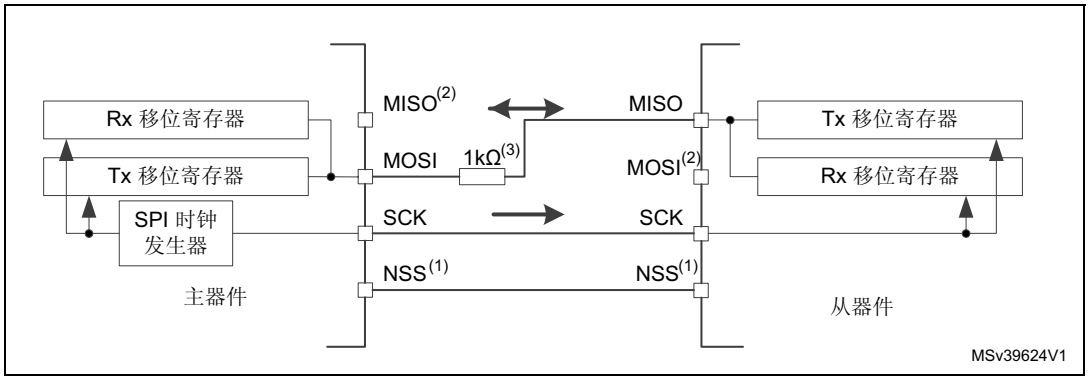


1. NSS 引脚可用于在主器件和从器件之间提供硬件控制流。外设也可选择不使用这些引脚。之后，必须在内部为主器件和从器件处理硬件控制流。有关更多详细信息，请参见*第 46.4.5 节：从器件选择 (NSS) 引脚管理*。

半双工通信

通过将 SPIx_CR1 寄存器的 BIDIMODE 位置 1，SPI 可采用半双工模式进行通信。在这种配置下，使用一条交叉连接线将主器件和从器件的移位寄存器连接起来。在此通信过程中，数据随 SCK 时钟边沿在移位寄存器之间进行移位，传输方向由主器件和从器件通过各自 SPIx_CR1 寄存器中的 BDIOE 位进行选择。在这种配置下，主器件的 MISO 引脚和从器件的 MOSI 引脚空闲，可在其他应用中用作 GPIO。

图 494. 半双工单个主器件/单个从器件应用



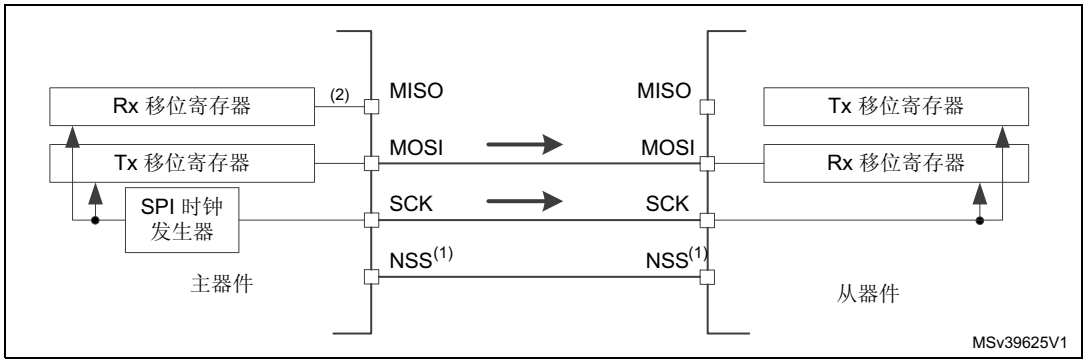
1. NSS 引脚可用于在主器件和从器件之间提供硬件控制流。外设也可选择不使用这些引脚。之后，必须在内部为主器件和从器件处理硬件控制流。有关更多详细信息，请参见*第 46.4.5 节：从器件选择 (NSS) 引脚管理*。
2. 在这种配置下，主器件的 MISO 引脚和从器件的 MOSI 引脚可用作 GPIO。
3. 当以双向模式工作的两个节点间的通信方向不是同步变化时，会出现临界情况，新发送器访问共用数据线，而前一个发送器仍保持线路上的相反值（值取决于 SPI 配置和通信数据）。两个节点会出现冲突，在共用线上短暂提供相反的输出电平，直到下一个节点也相应地改变其方向设置。建议此模式下在 MISO 和 MOSI 引脚之间插入串行电阻以在这种情况下保护输出并限制电流在二者之间流过。

单工通信

通过 SPIx_CR2 寄存器中的 RXONLY 位将 SPI 设置为只发送模式或只接收模式，可使 SPI 以单工模式进行通信。在这种配置下，仅使用一条线在主器件和从器件的移位寄存器之间进行传输。其余 MISO 和 MOSI 引脚对不用于通信，可用作标准 GPIO。

- **只发送模式 (RXONLY=0)**：配置设置与全双工设置相同。应用必须忽略在未使用的输入引脚上捕获的信息。该引脚可以用作标准 GPIO。
- **只接收模式 (RXONLY=1)**：应用可通过将 RXONLY 位置 1 来禁止 SPI 输出功能。在从器件配置下，MISO 输出被禁止，该引脚可用作 GPIO。当从器件选择信号有效时，从器件继续从 MOSI 引脚接收数据（请参见 *46.4.5：从器件选择 (NSS) 引脚管理*）。基于数据缓冲区的配置产生接收数据事件。在主器件配置下，MOSI 输出被禁止，该引脚可用作 GPIO。只要 SPI 处于使能状态，便不断生成时钟信号。停止时钟的唯一方式是将 RXONLY 位或 SPE 位清零，直至来自 MISO 引脚的传入模式结束，然后基于相应配置填充数据缓冲区结构。

图 495. 单工单个主器件/单个从器件应用（主器件为只发送模式/从器件为只接收模式）



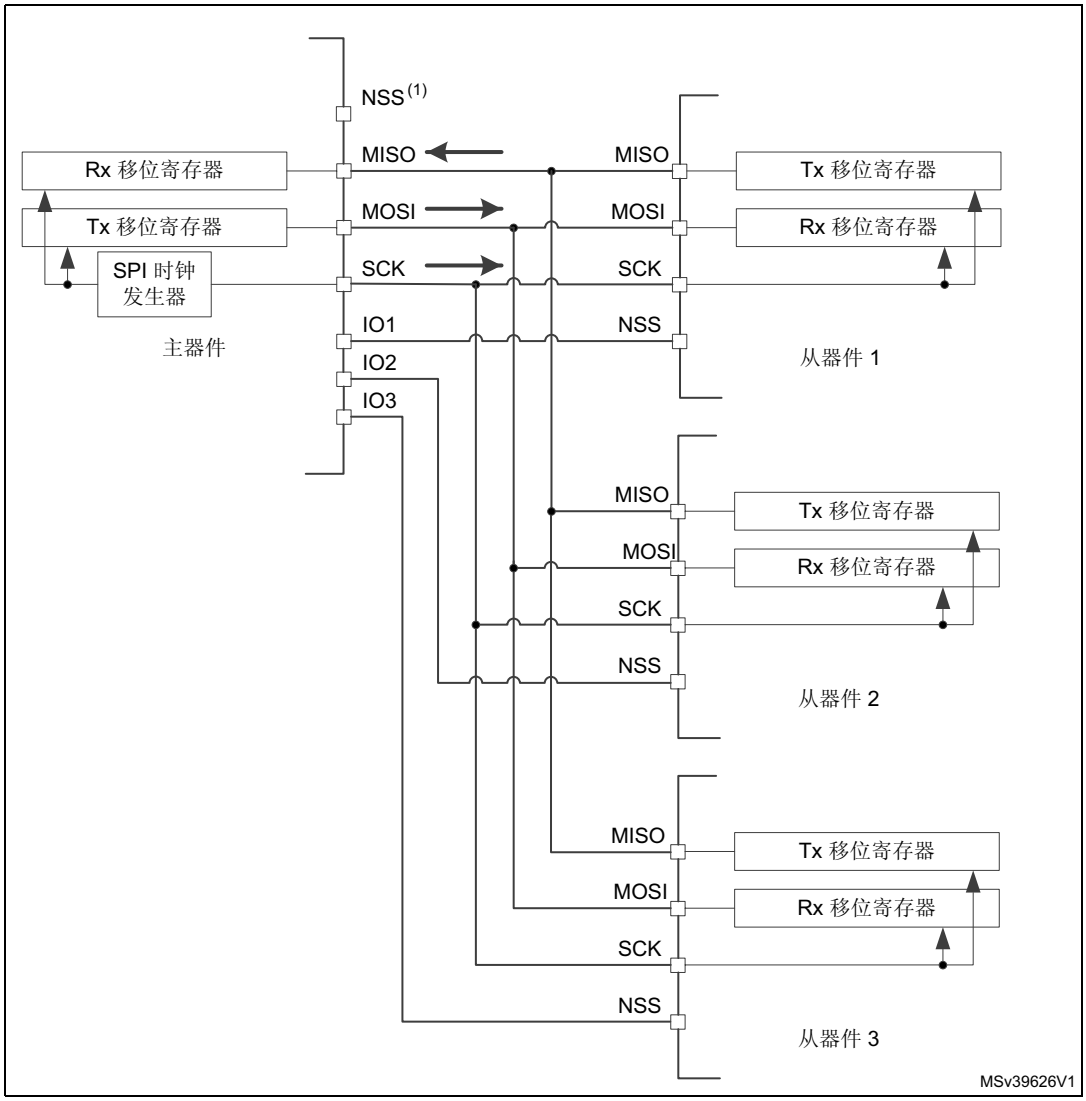
1. NSS 引脚可用于在主器件和从器件之间提供硬件控制流。外设也可选择不使用这些引脚。之后，必须在内部为主器件和从器件处理硬件控制流。有关更多详细信息，请参见*第 46.4.5 节：从器件选择 (NSS) 引脚管理*。
2. 在发送器 Rx 移位寄存器的输入上捕获意外输入信息。标准只发送模式下必须忽略与发送器接收流相关的所有事件（例如 OVF 标志）。
3. 在这种配置下，两个 MISO 引脚均可用作 GPIO。

注： *任何单工通信都可以通过把半双工模式中的方向设置固定来实现，（使能双向模式，同时 BDIO 位保持不变）。*

46.4.3 标准多从器件通信

在具有两个或多个独立从器件的配置下，主器件使用 GPIO 引脚来管理每个从器件的片选线（请参见*图 496*）。主器件必须通过拉低与从器件 NSS 输入相连的 GPIO 的电平来单独选择一个从器件。执行该操作后，便建立了标准主器件与专用从器件之间的通信。

图 496. 主器件和三个独立的从器件



1. 此配置的主器件侧不使用 NSS 引脚。该引脚必须在内部管理 (SSM = 1，SSI = 1) 以避免任何 MODF 错误。
2. 由于从器件的 MISO 引脚连在一起，所有从器件 MISO 引脚的 GPIO 配置必须设置为复用功能开漏（请参见 *第 297 页的第 8.3.7 节：I/O 复用功能输入/输出*）。

46.4.4 多主通信

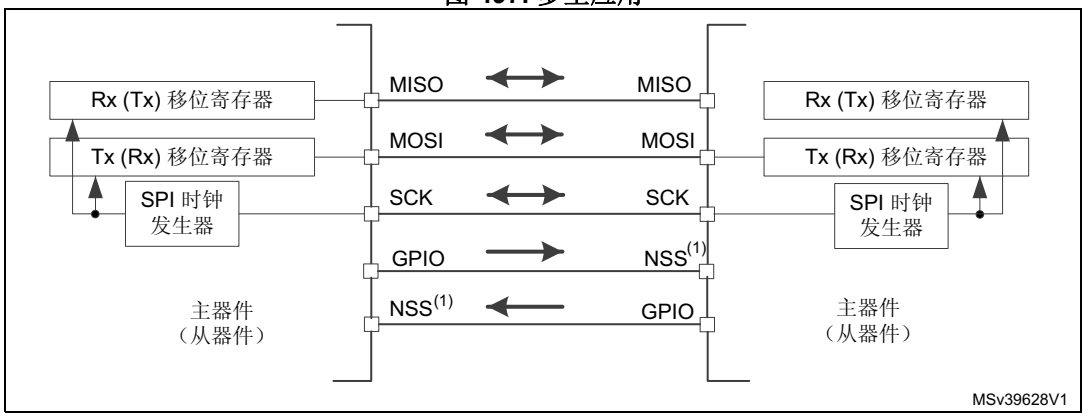
如果 SPI 总线未用于多主功能，用户可使用内置功能来检测尝试同时控制总线的两个节点间是否存在潜在冲突。对于该检测，NSS 引脚配置为硬件输入模式。

由于此时只有一个结点可将其输出施加到公用数据线上，因此无法连接超过两个以此模式工作的 SPI 节点。

当节点无效时，默认情况下均保持从模式。一旦一个节点要接管对总线的控制，它会将自身切换到主模式，然后通过专用 GPIO 引脚向其他节点的从器件选择输入施加有效电平。会话完成后，有效的从器件选择信号将被释放，控制总线的节点会短暂切换回被动从模式，等待下一个会话开始。

如果两个节点同时发出各自的控制请求，则会出现总线冲突（请参见模式故障 MODF 事件）。随后，用户可应用某个简单的仲裁过程（例如，在两个节点上施加不同的预定义超时来推迟下一个尝试）。

图 497. 多主应用



1. 在两个节点上，NSS 引脚配置为硬件输入模式。当无效节点配置为从器件时，其有效电平将使能 MISO 线输出控制。

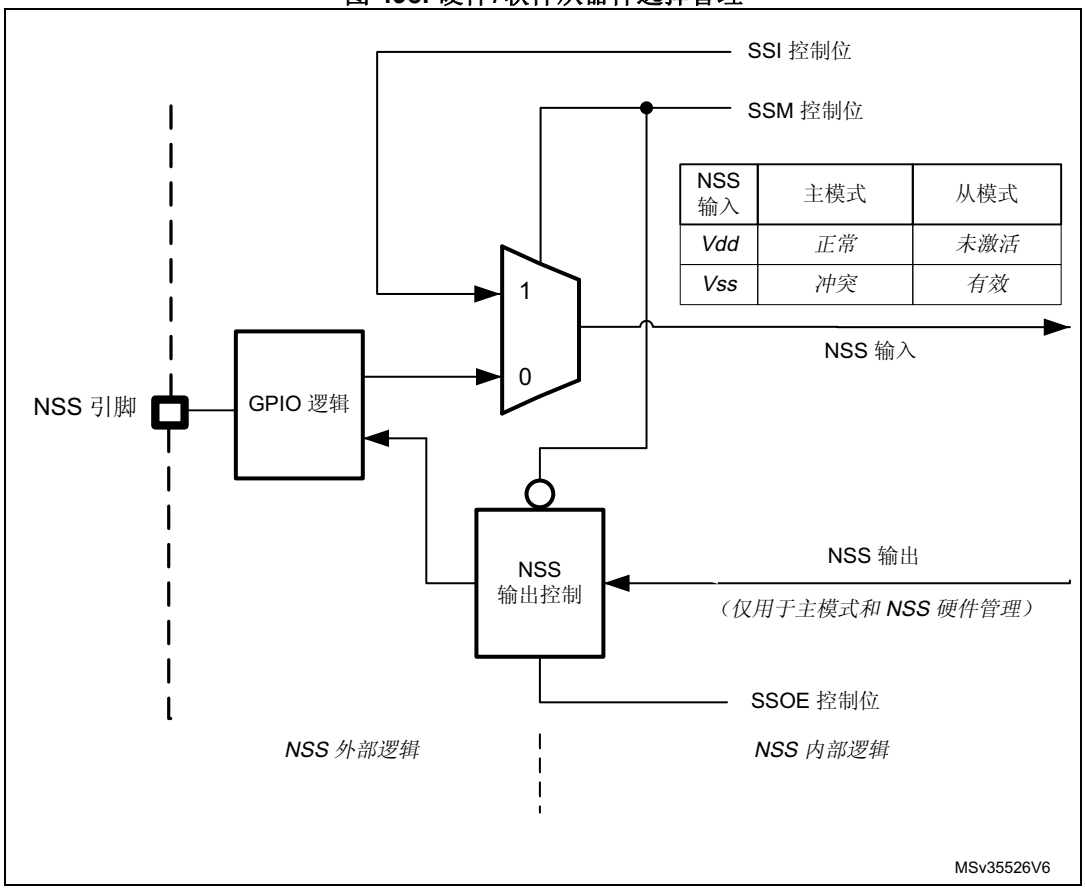
46.4.5 从器件选择 (NSS) 引脚管理

在从模式下，NSS 用作标准的“片选”输入，使从器件与主器件进行通信。在主模式下，NSS 可用作输出或输入。用作输入时，可防止多主模式总线冲突；用作输出时，可驱动单个从器件的从器件选择信号。

可以使用 SPIx_CR1 寄存器中的 SSM 位设置硬件或软件从器件选择管理：

- **软件 NSS 管理 (SSM = 1)**：在这种配置下，由 SPIx_CR1 寄存器中的 SSI 位的值内部驱动从器件选择信息。外部 NSS 引脚空闲，可供其他应用使用。
- **硬件 NSS 管理 (SSM = 0)**：在这种情况下，可行的配置有两种：所用配置取决于 NSS 输出配置（SPIx_CR1 寄存器中的 SSOE 位）。
  - **NSS 输出使能（SSM=0 且 SSOE = 1）**：仅在将 MCU 设置为主器件时才使用该配置。NSS 引脚由硬件管理。只要在主模式下使能 SPI (SPE=1)，NSS 信号便会被驱动为低电平，并且会一直保持低电平状态，直至禁止 SPI (SPE =0)。如果激活 NSS 脉冲模式 (NSSP=1)，连续通信间会生成脉冲。SPI 无法在采用此 NSS 设置的多主模式配置下工作。
  - **NSS 输出禁止（SSM=0 且 SSOE = 0）**：如果微控制器在总线上用作主器件，此配置可实现多主模式功能。如果在该模式下将 NSS 引脚拉至低电平，SPI 将进入主模式故障状态，器件将在从模式下自动进行重新配置。在从模式下，NSS 引脚用作标准的“片选”输入，当 NSS 线为低电平时将选择从器件。

图 498. 硬件/软件从器件选择管理



46.4.6 通信格式

SPI 通信过程中，将同时执行接收和发送操作。串行时钟 (SCK) 对数据线上的信息的移位和采样进行同步。通信格式取决于时钟相位、时钟极性和数据帧格式。为了能够在彼此间进行通信，主器件和从器件必须遵循相同的通信格式。

时钟相位和极性控制

通过 SPIx_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）位控制不传输任何数据时时钟的空闲状态值。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 位置 1，则会在 SCK 引脚的第二个边沿捕获传输的第一个数据位（如果复位 CPOL 位，则为下降沿；如果将 CPOL 位置 1，则为上升沿）。即，在每次出现该时钟边沿时锁存数据。如果将 CPHA 位复位，则会在 SCK 引脚的第一个边沿捕获传输的第一个数据位（如果将 CPOL 位置 1，则为下降沿；如果将 CPOL 位复位，则为上升沿）。即，在每次出现该时钟边沿时锁存数据。

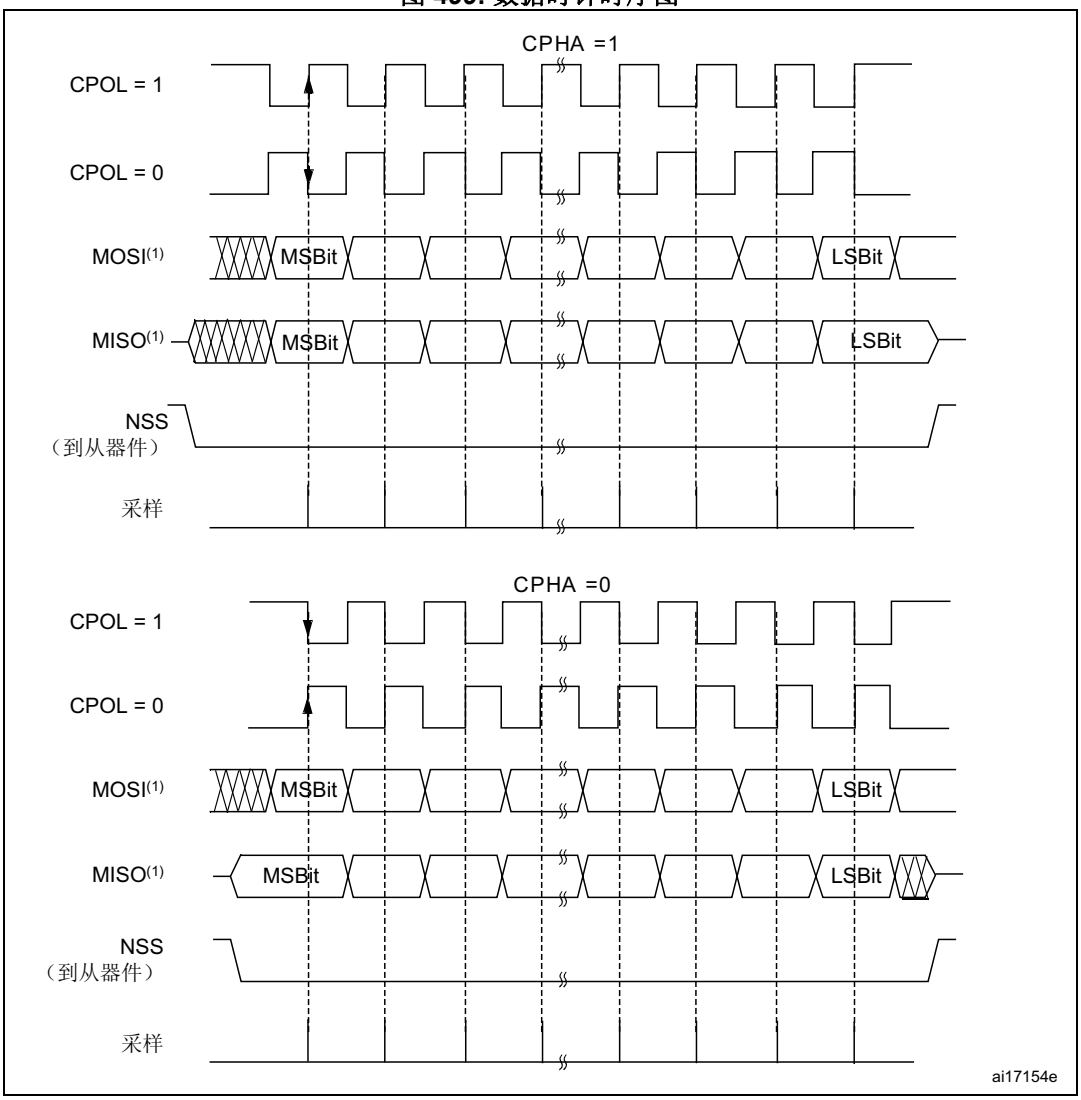
CPOL（时钟极性）和 CPHA（时钟相位）位的组合用于选择数据捕获时钟边沿。

图 499 给出了在 CPHA 和 CPOL 位的四种组合下的 SPI 全双工传输。

注： *在切换 CPOL/CPHA 位之前，必须通过复位 SPE 位来禁止 SPI。*

SCK 的空闲状态必须与 SPIx_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）。

图 499. 数据时钟时序图

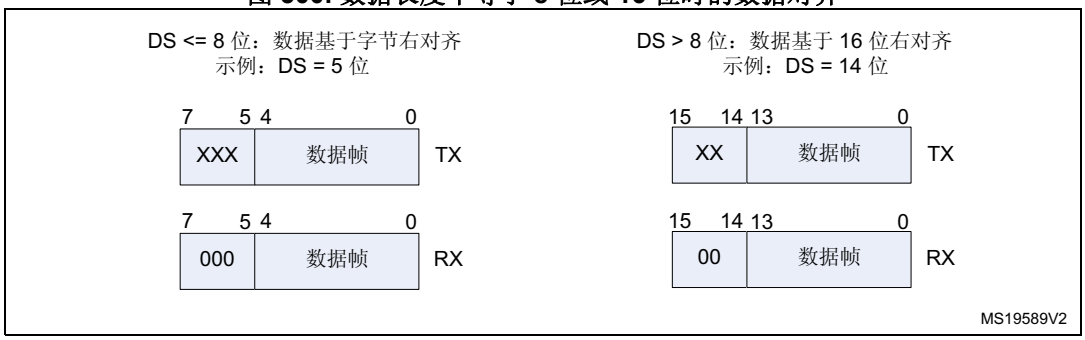


1. 数据位的顺序取决于 LSBFIRST 位的设置。

数据帧格式

SPI 移位寄存器可设置为以 MSB 在前或 LSB 在前的方式移出数据，具体取决于 LSBFIRST 位的值。数据帧的长度通过 DS 位进行选择。数据帧的长度可设置为 4 位到 16 位，且此设置对于发送和接收均适用。无论所选的数据帧长度为何，必须依照 FRXTH 电平对 FIFO 进行读访问。访问 SPIx_DR 寄存器时，数据帧始终按字节（数据不超过一个字节时）或半字进行右对齐（请参见*图 500*）。在通信过程中，只会为数据帧内的位提供时钟并传输这些位。

图 500. 数据长度不等于 8 位或 16 位时的数据对齐



注： *数据长度至少为 4 位。如果所选的数据长度不足 4 位，则会将数据帧长度强制为 8 位。*

46.4.7 SPI 配置

主器件和从器件的配置步骤几乎相同。对于具体的模式设置，请遵从相应部分的内容。若要对标准通信进行初始化，请执行以下步骤：

1. 对适当的 GPIO 寄存器执行写操作：将 MOSI、MISO 和 SCK 引脚配置为 GPIO。
2. 对 SPI_CR1 寄存器执行写操作：
   a) 通过 BR[2:0] 位配置串行时钟波特率（注：4）。
   b) 配置 CPOL 和 CPHA 位的组合，定义数据传输和串行时钟之间的关系（四种关系中的一种）（NSSP 模式下必须将 CPHA 清零）。（注：2——在 TI 模式下使能 CRC 的情况除外）。
   c) 通过配置 RXONLY 或 BIDIMODE 和 BIDIOE 来选择单工或半双工模式（RXONLY 和 BIDIMODE 不可同时置 1）。
   d) 配置 LSBFIRST 位以定义帧格式（注：2）。
   e) 如果需要 CRC，请配置 CRCL 和 CRCEN 位（SCK 时钟信号处于空闲状态时）。
   f) 配置 SSM 和 SSI（注：2 和 3）。
   g) 配置 MSTR 位（在多主模式 NSS 配置下，如果主器件配置为防止发生 MODF 错误，则应避免 NSS 上出现状态冲突）。
3. 对 SPI_CR2 寄存器执行写操作：
   a) 配置 DS[3:0] 位，选择传输的数据长度。
   b) 配置 SSOE（注：1、2 和 3）。
   c) 如果需要使用 TI 协议，请将 FRF 位置 1（TI 模式下将 NSSP 位保持清零状态）。
   d) 如果在两个数据单元之间需要 NSS 脉冲模式，请将 NSSP 位置 1（NSSP 模式下将 CHPA 和 TI 位保持清零状态）。
   e) 配置 FRXTH 位。RXFIFO 阈值必须与 SPIx_DR 寄存器的读访问大小相符。
   f) 如果封装模式下使用 DMA，请初始化 LDMA_TX 和 LDMA_RX 位。
4. 对 SPI_CRCPR 寄存器执行写操作：需要时配置 CRC 多项式。
5. 对相应的 DMA 寄存器执行写操作：如果使用 DMA 数据流，请在 DMA 寄存器中配置 SPI Tx 和 Rx 专用的 DMA 数据流。

注： *(1) 从模式下无需此步骤。*

(2) TI 模式下无需此步骤。

(3) NSSP 模式下无需此步骤。

(4) 从模式下无需此步骤，但从器件在 TI 模式下工作时除外。

46.4.8 使能 SPI 的步骤

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。从器件的数据寄存器必须包含待发送的数据才能开始与主器件通信（在通信时钟的第一个边沿；如果时钟信号连续，则是在正在进行的通信结束前）。使能 SPI 从器件前，SCK 信号必须稳定为所选极性对应的空闲状态电平。

当 SPI 处于使能状态，且 TXFIFO 不为空或者对 TXFIFO 执行下一次写操作时，全双工模式（或任何只发送模式）下的主器件开始通信。

在任何主器件只接收模式（RXONLY=1 或 BIDIMODE=1 且 BIDIOE=0）下，使能 SPI 后，主器件立即开始通信且时钟立即开始运行。

要处理 DMA，请遵从相应部分的内容。

46.4.9 数据发送和接收过程

RXFIFO 和 TXFIFO

所有 SPI 数据交互均经由 32 位内置 FIFO。这使得 SPI 能够以连续流工作，并能防止在数据帧长度较短时发生上溢。每个方向都有其自身的 FIFO，称为 TXFIFO 和 RXFIFO。这些 FIFO 可在所有 SPI 模式下使用，但已使能 CRC 计算的只接收模式（从器件或主器件）除外（请参见*第 46.4.14 节：CRC 计算*）。

FIFO 的处理取决于数据交换模式（双工和单工）、数据帧格式（帧中的位数）、FIFO 数据寄存器中的访问大小（8 位或 16 位）以及访问 FIFO 时是否使用数据封装（请参见*第 46.4.13 节：TI 模式*）。

对 SPIx_DR 寄存器执行读访问时，会返回尚未读取的 RXFIFO 中存储的最早的值。对 SPIx_DR 执行写访问时，会在发送队列结束时将写入的数据存储到 TXFIFO 中。读访问必须始终符合由 SPIx_CR2 寄存器中的 FRXTH 位配置的 RXFIFO 阈值。FTLVL[1:0] 和 FRLVL[1:0] 位用于指示两个 FIFO 当前的占用水平。

SPIx_DR 寄存器的读访问必须通过 RXNE 事件进行管理。当数据存储到 RXFIFO 中且达到阈值（由 FRXTH 位定义）时会触发该事件。当 RXNE 清零时，RXFIFO 被视为空。同样地，待发送数据帧的写访问通过 TXE 事件进行管理。当 TXFIFO 占用水平小于或等于其容量的一半时会触发该事件。否则，TXE 清零，TXFIFO 被视为已满。通过这种方式，当数据帧格式不超过 8 位时，RXFIFO 最多可存储四个数据帧，而 TXFIFO 最多只能存储三个数据帧。当软件尝试向 TXFIFO 中写入更多 16 位模式的数据时，这种差异能够防止 TXFIFO 中已存储的 3 个 8 位数据帧出现损坏的情况。TXE 和 RXNE 事件可通过中断来轮询或处理。请参见*图 502* 到*图 505*。

另一种管理数据交换的方式是使用 DMA（请参见*使用 DMA（直接存储器寻址）进行通信*）。

如果在 RXFIFO 已满时收到下一个数据，将发生上溢事件（请参见*第 46.4.10 节：SPI 状态标志*中的 OVR 标志说明）。上溢事件可通过中断来轮询或处理。

BSY 位被置 1 表示正在处理当前数据帧交互。当时钟信号连续运行时，在主器件中，BSY 标志在数据帧之间保持置 1 状态，但在从器件中，BSY 标志在数据帧传输之间变为低电平并持续最短的一段时间（一个 SPI 时钟）。

序列处理

可通过一个序列传送一些数据帧从而完成一条消息。使能发送后，序列即开始，只要主器件的 TXFIFO 中存在数据便一直继续。时钟信号由主器件持续提供，直至 TXFIFO 变为空，之后时钟信号停止，等待其他数据。

在只接收模式、半双工模式（BIDIMODE=1 且 BIDIOE=0）或单工模式（BIDIMODE=0 且 RXONLY=1）下，当使能 SPI 并激活只接收模式后，主器件将立即启动序列。时钟信号由主器件提供，且直至主器件禁止 SPI 或只接收模式才会停止。在此之前，主器件会连续接收数据帧。

当主器件能够以连续模式（SCK 信号连续）提供所有交互时，任何时候都必须根据从器件功能来处理数据流及其内容。必要时，主器件必须降低通信速度，提供较慢的时钟或带有足够延时的单独帧或数据段。请注意，SPI 模式下不存在主器件或从器件的下溢错误信号，来自从器件的数据始终由主器件处理，即使从器件无法及时正确地准备数据也是如此。从器件最好使用 DMA，尤其是数据帧较短而总线速率较高时。

在多从器件系统中，每个序列必须通过 NSS 脉冲进行控制，从而只选择其中一个从器件进行通信。在单个从器件系统中，无需通过 NSS 来控制从器件，但此时提供此脉冲通常会更好，以在每个数据序列开始时同步从器件。NSS 可通过软件和硬件进行管理（请参见*第 46.4.5 节：从器件选择 (NSS) 引脚管理*）。

当 BSY 位置 1 时，表示正在处理数据帧事务。当所进行的帧交互完成时，RXNE 标志将置 1。最后一位采样后，整个数据帧会存储到 RXFIFO 中。

禁止 SPI 的步骤

当禁止 SPI 时，必须按照本段中介绍的禁止步骤进行操作。当外设时钟停止时，在系统进入低功耗模式前做到这一点是十分重要的。否则会损坏正在进行的交互。在某些模式下，禁止步骤是停止所进行的连续通信的唯一方式。

当处于全双工或只发送模式下的主器件停止提供待发送的数据时，可结束任何事务。在这种情况下，时钟在最后一个数据传输后停止。当处理奇数数量的数据帧时，必须特别注意封装模式以防止交换一些空字节（请参见*数据封装*部分）。在这些模式下禁止 SPI 之前，用户必须按照标准的禁止步骤进行操作。SPI 在主模式传输时，如果在数据帧处理的过程中，或者 TXFIFO 中有待传输的数据收，把 SPI 禁止，SPI 的此时的状态是不可预测的。所以，必须按照标准的禁止流程进行操作。

只要主器件处于只接收模式，停止连续时钟的唯一方式就是通过 SPE=0 来禁止外设。这必须在最后一个数据帧传输内的特定时间段，即第一位采样与最后一位传输开始之间完成（以便接收全部数量的预期数据帧并防止在最后一个有效数据帧后读取任何其他的“空”数据）。在该模式下禁止 SPI 时必须遵循特定步骤。

禁止 SPI 后，已接收但未读取的数据始终存储在 RXFIFO 中，这些数据必须在下次使能 SPI 后进行处理，然后才能启动新序列。为防止存在未读取的数据，需确保禁止 SPI 时 RXFIFO 为空，可通过正确的禁止步骤来禁止 SPI，也可以通过控制外设复位专用的特定寄存器以软件复位的方式来初始化所有 SPI 寄存器从而禁止 SPI（请参见 RCC_APBiRSTR 寄存器中的 SPIiRST 位）。

标准禁止步骤通过轮询 BSY 状态以及 FTLVL[1:0] 来检查发送会话是否完全结束。还可以在必须识别正在处理的传输是否结束的特定情况下完成这种检查，例如：

- 当 NSS 信号由软件管理且主器件必须为从器件提供 NSS 脉冲结束时，或者
- 最后一个数据帧或 CRC 帧传输仍在外设总线中处理时，来自 DMA 或 FIFO 的传输数据流完成。

正确的禁止步骤如下（使用只接收模式时除外）：

1. 等待至 FTLVL[1:0] = 00（无需发送更多数据）。
2. 等待至 BSY=0（最后一个数据帧已处理完）。
3. 禁止 SPI (SPE = 0)。
4. 读取数据，直至 FRLVL[1:0] = 00（读取接收的所有数据）。

某些只接收模式的正确禁止步骤如下：

1. 当最后一个数据帧正在处理时，通过在特定时间窗口内禁止 SPI (SPE=0) 来中断接收流。
2. 等待至 BSY=0（最后一个数据帧已处理完）。
3. 读取数据，直至 FRLVL[1:0] = 00（读取接收的所有数据）。

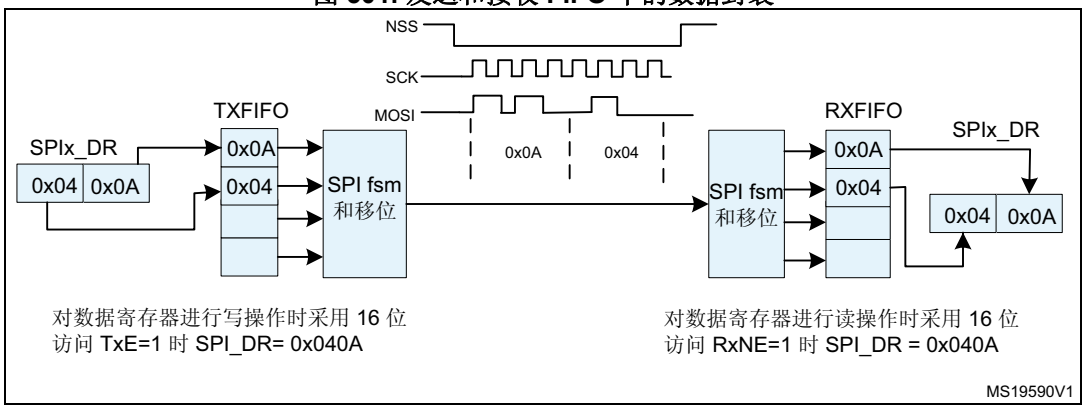
注： *如果使用封装模式并且必须接收奇数数量的数据帧且数据帧格式为小于或等于 8 位（不超过一个字节），则 FRXTH 必须在 FRLVL[1:0] = 01 时置 1，以便生成 RXNE 事件从而读取最后的奇数编号数据帧并且使 FIFO 指针保持正确对齐。*

数据封装

若数据帧长度不足一个字节（小于或等于 8 位），当 SPIx_DR 寄存器上执行任何 16 位读写访问时将自动使用数据封装。在这种情况下将并行处理双数据帧模式。最初，SPI 以所访问字的 LSB 中存储的模式工作，然后以 MSB 中存储的另一种模式来工作。*图 501* 给出了数据封装模式序列处理的示例。在对发送器的 SPIx_DR 寄存器执行单次 16 位访问后发送两个数据帧。如果 RXFIFO 阈值设置为 16 位 (FRXTH=0)，该序列只会在接收器中生成一个 RXNE 事件。接收器随后必须通过对 SPIx_DR 执行单次 16 位读访问来访问这两个数据帧，从而响应该单个 RXNE 事件。RxFIFO 阈值设置和后续读访问必须始终与接收器侧保持一致，因为若不一致，则会丢失数据。

如果必须处理奇数数量的此类“不超过一个字节”的数据帧，则会出现特定问题。在发送器侧，只需将任意奇序列的最后一个数据帧以 8 位访问的方式写入 SPIx_DR 即可。接收器必须针对所接收的奇序列中的最后一个数据帧更改 Rx_FIFO 阈值大小，以生成 RXNE 事件。

图 501. 发送和接收 FIFO 中的数据封装



使用 DMA（直接存储器寻址）进行通信

为了以最大速度工作并且方便避免上溢所需的数据寄存器读/写过程，SPI 提供了 DMA 功能，该功能采用了简单的请求/应答协议。

将 SPIx_CR2 寄存器中的使能位 TXE 或 RXNE 置 1 时，将请求 DMA 访问。必须向发送缓冲区和接收缓冲区发出单独的请求。

- 在发送过程中，每次 TXE 位置 1 都会发出 DMA 请求。然后，DMA 将对 SPIx_DR 寄存器执行写操作。
- 在接收过程中，每次 RXNE 位置 1 都会发出 DMA 请求。然后，DMA 将对 SPIx_DR 寄存器执行读操作。

请参见*图 502* 到*图 505*。

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，OVR 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 写入所有要发送的数据（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。在禁止 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发送。软件必须首先等待 FTLVL[1:0]=00，再等待 BSY=0。

通过 DMA 开始通信时，为防止 DMA 通道管理引发错误事件，必须按顺序执行以下步骤：

1. 如果使用 DMA Rx，通过 SPI_CR2 寄存器中的 RXDMAEN 位来使能 DMA 接收缓冲区。
2. 如果使用数据流，通过 DMA 寄存器来使能 Tx 和 Rx 的 DMA 数据流。
3. 如果使用 DMA Tx，通过 SPI_CR2 寄存器中的 TXDMAEN 位来使能 DMA 发送缓冲区。
4. 通过将 SPE 位置 1 使能 SPI。

要关闭通信，必须按顺序执行以下步骤：

1. 如果使能了 DMA，通过 DMA 寄存器来禁止 Tx 和 Rx 的 DMA 数据流。
2. 通过后续 SPI 禁止步骤来禁止 SPI。
3. 如果使用 DMA Tx 和/或 DMA Rx，通过将 SPI_CR2 寄存器中的 TXDMAEN 和 RXDMAEN 位清零来禁止 DMA 发送缓冲区和接收缓冲区。

使用 DMA 时的数据封装

如果由 DMA 管理传输（SPIx_CR2 寄存器中的 TXDMAEN 和 RXDMAEN 位置 1），则将根据为 SPI TX 和 SPI RX 的 DMA 通道配置的 PSIZE 值自动使能/禁止封装模式。如果 DMA 通道的 PSIZE 值等于 16 位，且 SPI 数据大小小于或等于 8 位，则将使能封装模式。然后，DMA 将自动管理对 SPIx_DR 寄存器的写操作。

如果使用数据封装模式，且要传输的数据数量不是 2 的倍数，则 LDMA_TX/LDMA_RX 位必须置 1。然后，SPI 会认为最后一次 DMA 传输时只发送或接收一个数据（有关详细信息，请参见*第 1586 页的数据封装*）。

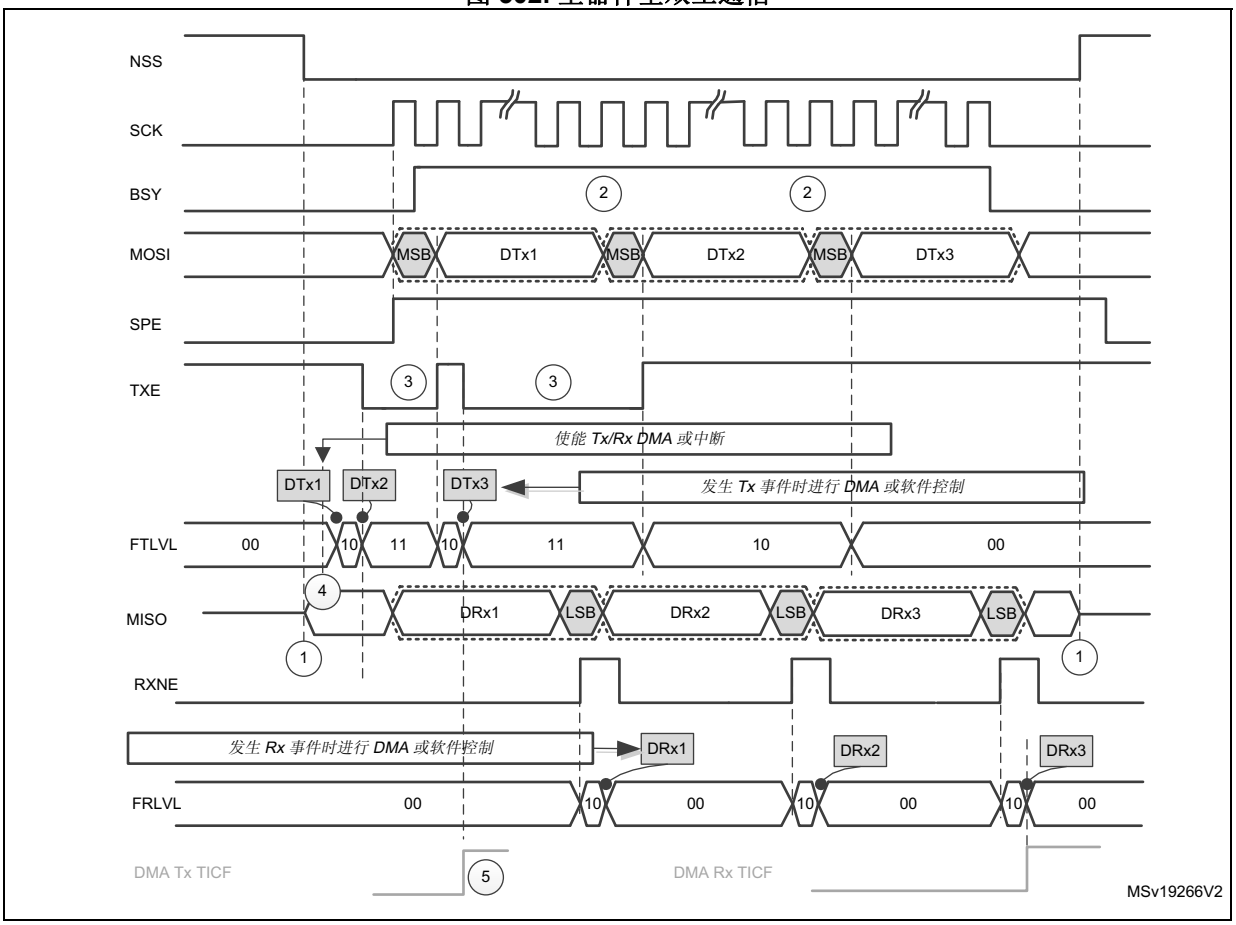
通信图

本部分将介绍一些典型的时序图。无论 SPI 事件是通过轮询、中断还是 DMA 进行处理，这些时序图均有效。为简单起见，此处均假设 LSBFIRST=0、CPOL=0 和 CPHA=1。不提供 DMA 数据流的完整配置。

以下带编号的注释对*第 1589 页的图 502* 到*第 1592 页的图 505* 均适用。

1. 激活 NSS 并使能 SPI 后，从器件开始控制 MISO 线，而当其中一个条件不成立时，从器件将与 MISO 线断开。必须为从器件提供充足的时间，以便在传输开始前准备好主器件专用的数据。
   在主器件上，只有使能 SPI 后，SPI 外设才会控制 MOSI 和 SCK 信号（偶尔还会控制 NSS 信号）。如果禁止 SPI，SPI 外设会与 GPIO 逻辑断开，因此这些线上的电平只取决于 GPIO 设置。
2. 在主器件上，如果通信（时钟信号）连续，BSY 在数据帧之间保持有效。在从器件上，BSY 信号在数据帧之间始终会保持至少一个时钟周期的低电平状态。
3. 只有 TXFIFO 已满时，TXE 信号才会清零。
4. DMA 仲裁过程在 TXDMAEN 位置 1 后立即开始。TXE 中断在 TXEIE 置 1 后立即生成。TXE 信号处于有效电平时，开始向 TxFIFO 传输数据，直至 TxFIFO 已满或 DMA 传输完成。
5. 如果要发送的所有数据可装入 TxFIFO，则 DMA Tx TCIF 标志甚至会在 SPI 总线上的通信开始前置 1。SPI 传输完成前，该标志始终为高电平状态。
6. 封装的 CRC 值在 SPIx_TxCRCR 和 SPIx_RxCRCR 寄存器中逐帧连续进行计算。完成整个数据封装后，CRC 信息可通过 DMA 自动处理（Tx 通道必须设置为要处理的数据帧数），也可由软件处理（用户必须在处理最后一个数据帧的过程中处理 CRCNEXT 位）。SPIx_TxCRCR 中计算的 CRC 值仅由发送器发出时，接收的 CRC 信息将加载到 RxFIFO 中，然后与 SPIx_RxCRCR 寄存器的内容进行比较（如果存在任何差异，CRC 错误标志会置 1）。因此用户必须注意刷新 FIFO 中的相关信息，可以通过软件读出 RxFIFO 中存储的所有内容的方式来实现，若已针对 Rx 通道预设置了适当数量的数据帧（数据帧数 + CRC 帧数），也可通过 DMA 的方式来实现（请参见示例假设中的设置）。
7. 在数据封装模式下，TxE 和 RxNE 事件成对出现，对 FIFO 的每次读/写访问都为 16 位宽，直至数据帧数为偶数。如果 TxFIFO 处于 ¾ 满状态，则 FTLVL 将保持 FIFO 满时对应的状态。因此在 TxFIFO 变为 ½ 满状态前，无法存储最后一个奇数编号的数据帧。该帧可通过软件或自动由 DMA（LDMA_TX 控制置 1 时）对其进行 8 位访问的方式存储在 TxFIFO 中。
8. 要在封装模式下接收最后一个奇数编号的数据帧，必须在处理最后一个数据帧后，将 Rx 阈值更改为 8 位，这可通过软件（设置为 FRXTH=1）实现或由 DMA 内部信号在 LDMA_RX 置 1 时自动实现。

图 502. 主器件全双工通信



主器件全双工通信示例的假设条件如下：

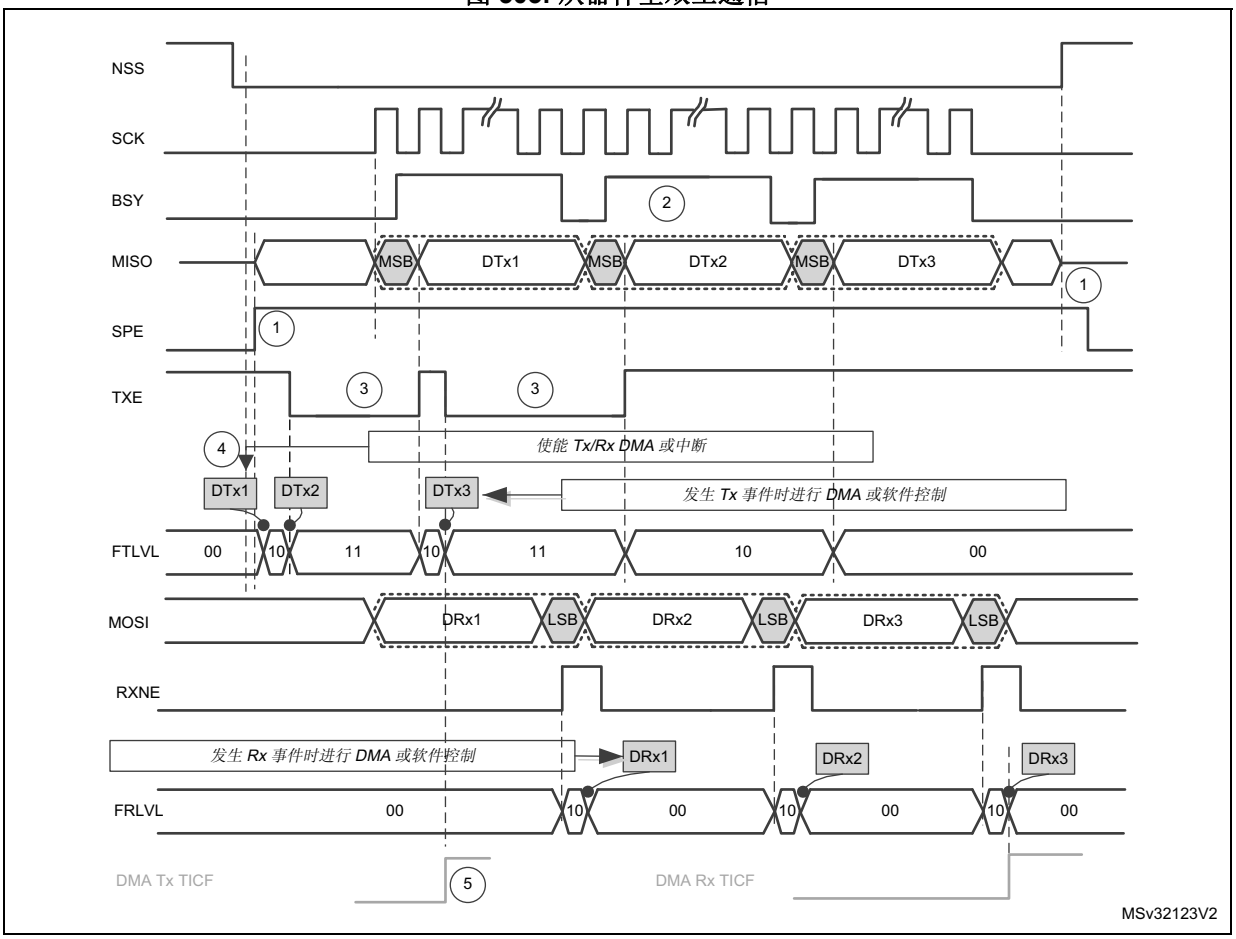
- 数据大小 > 8 位

如果使用 DMA：

- 由 DMA 处理的 Tx 帧的数量设置为 3
- 由 DMA 处理的 Rx 帧的数量设置为 3

有关通用假设条件和注释的详细信息，另请参见*第 1588 页的通信图*。

图 503. 从器件全双工通信



从器件全双工通信示例的假设条件如下：

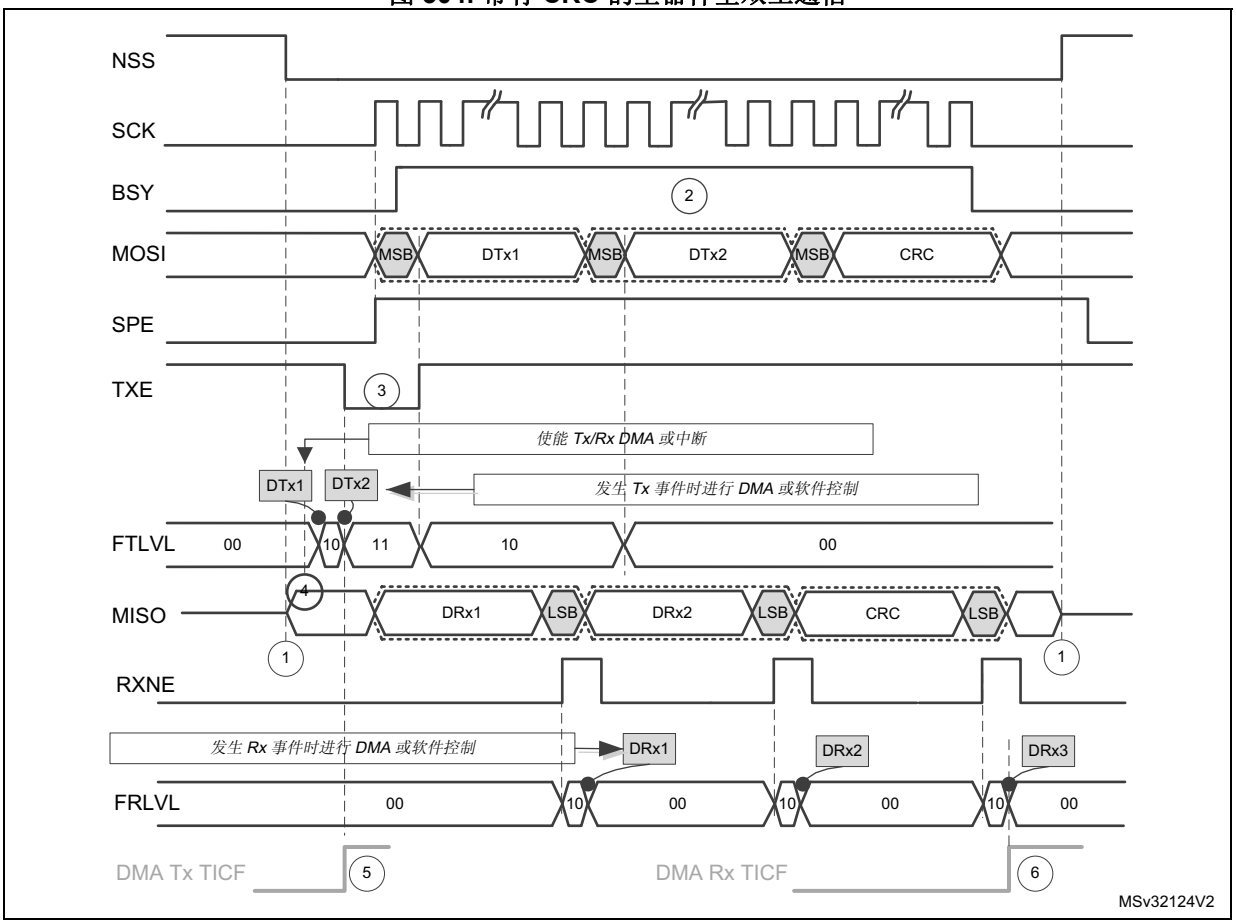
- 数据大小 > 8 位

如果使用 DMA：

- 由 DMA 处理的 Tx 帧的数量设置为 3
- 由 DMA 处理的 Rx 帧的数量设置为 3

有关通用假设条件和注释的详细信息，另请参见*第 1588 页的通信图*。

图 504. 带有 CRC 的主器件全双工通信



带有 CRC 的主器件全双工通信的假设条件如下：

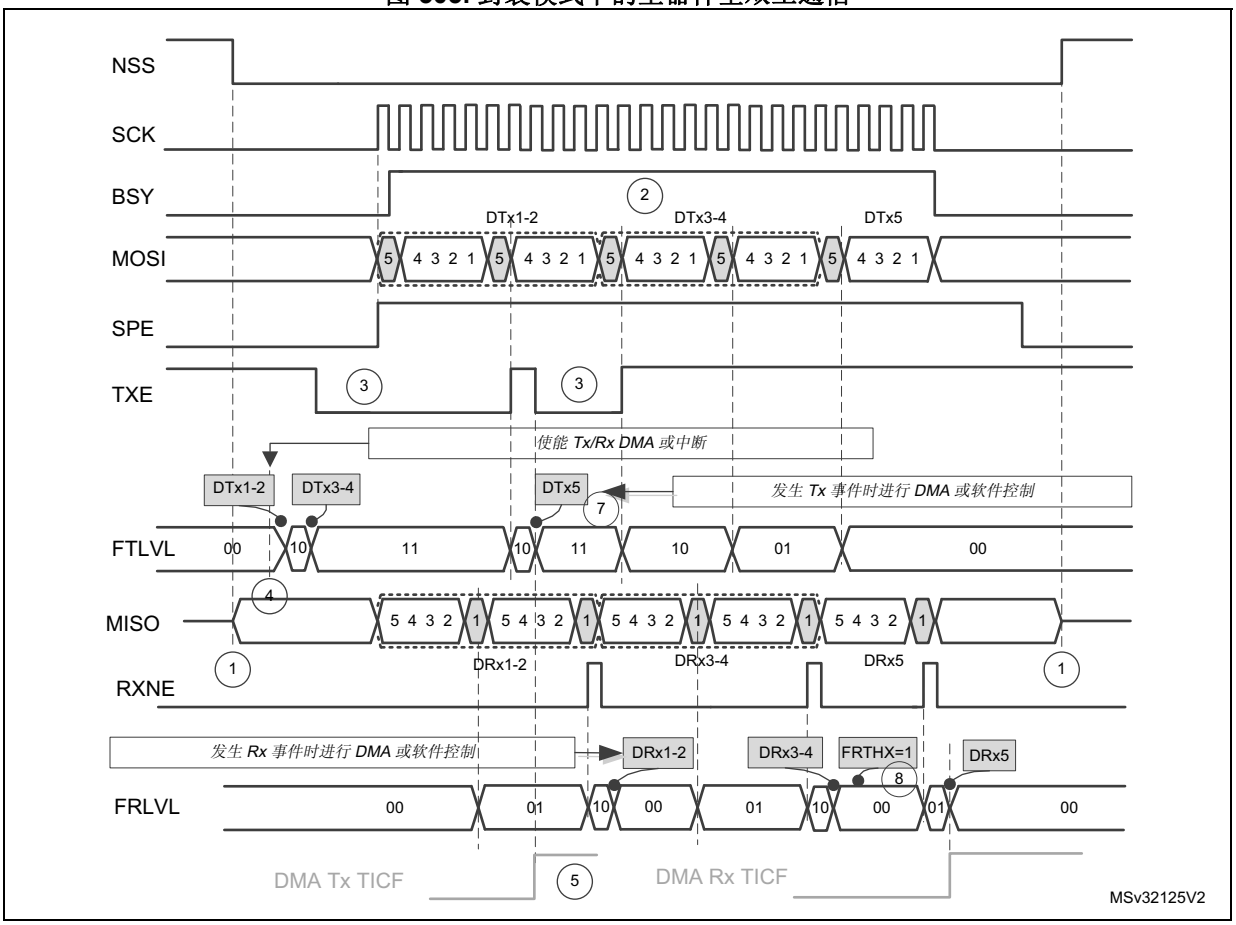
- 数据大小 = 16 位
- CRC 已使能

如果使用 DMA：

- 由 DMA 处理的 Tx 帧的数量设置为 2
- 由 DMA 处理的 Rx 帧的数量设置为 3

有关通用假设条件和注释的详细信息，另请参见*第 1588 页的通信图*。

图 505. 封装模式下的主器件全双工通信



封装模式下的主器件全双工通信示例的假设条件如下：

- 数据大小 = 5 位
- 主要通过 16 位访问的方式来读/写 FIFO
- FRXTH=0

如果使用 DMA：

- 要由 DMA 处理的 Tx 帧的数量设置为 3
- 要由 DMA 处理的 Rx 帧的数量设置为 3
- Tx 和 Rx 的 DMA 通道 的 PSIZE 均设置为 16 位
- LDMA_TX=1 且 LDMA_RX=1

有关通用假设条件和注释的详细信息，另请参见*第 1588 页的通信图*。

46.4.10 SPI 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)

当发送 TXFIFO 有足够的空间来存储要发送的数据时，TXE 标志将置 1。TXE 标志与 TXFIFO 占用水平相关。该标志变为高电平后将一直保持高电平状态，直至 TXFIFO 占用水平小于或等于 FIFO 深度的 1/2。如果 SPIx_CR2 寄存器中的 TXEIE 位置 1，可产生中断。当 TXFIFO 占用水平超过 1/2 时，该位将自动清零。

接收缓冲区非空 (RXNE)

RXNE 标志根据 SPIx_CR2 寄存器中 FRXTH 位的值进行设置：

- 如果 FRXTH 置 1，RXNE 将变为高电平并一直保持高电平状态，直至 RXFIFO 占用水平大于或等于 1/4（8 位）。
- 如果 FRXTH 清零，RXNE 将变为高电平并一直保持高电平状态，直至 RXFIFO 占用水平大于或等于 1/2（16 位）。

如果 SPIx_CR2 寄存器中的 RXNEIE 位置 1，可产生中断。

当上述条件不再为真时，RXNE 将由硬件自动清零。

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）。

当 BSY 置 1 时，表示 SPI 上正在进行数据传输（SPI 总线繁忙）。

某些模式下可以使用 BSY 标志来检测传输是否结束，以便软件在进入低功耗模式（该模式下不提供外设时钟）前禁止 SPI 或其外设时钟。这可避免破坏最后一个数据的传输。

BSY 标志还可用于避免在多主模式系统中发生写冲突。

在以下任意一种条件下，BSY 标志将清零：

- 正确禁止 SPI 时
- 在主模式下检测到故障时（MODF 位置 1）
- 在主模式下，完成了数据发送并且不准备发送任何新数据时
- 在从模式下，BSY 标志在各传输之间的至少一个 SPI 时钟周期内置为“0”时

注： *当主器件可以立即处理下一次发送时（例如，如果主器件处于只接收模式或其发送 FIFO 不为空），在主器件侧的传输之间，通信连续且 BSY 标志始终置“1”。尽管从器件并非如此，但建议始终使用 TXE 和 RXNE 标志（而非 BSY 标志）来处理数据发送或接收操作。*

46.4.11 SPI 错误标志

如果以下其中一个错误标志置 1 且已通过将 ERRIE 位置 1 使能了中断，则将生成 SPI 中断。

上溢标志 (OVR)

当主器件或从器件接收了数据但 RXFIFO 没有足够的空间来存储接收的数据时，将出现上溢的情况。如果软件或 DMA 没有足够的时间来读取之前接收的数据（存储在 RXFIFO 中）或数据存储空间受限（例如在只接收模式下使能 CRC 时 RXFIFO 不可用，在这种情况下，接收缓冲区便限制为一个数据帧缓冲区），会发生这种情况（请参见*第 46.4.14 节：CRC 计算*）。

当出现上溢的情况时，新接收的值不会覆盖 RXFIFO 中之前的值。新接收的值将被丢弃，之后发送的所有数据都将丢失。要将 OVR 位清零，应首先对 SPI_DR 寄存器执行读访问，然后再对 SPI_SR 寄存器执行读访问。

模式故障 (MODF)

当主器件的内部 NSS 信号（NSS 硬件模式下为 NSS 引脚，NSS 软件模式下为 SSI 位）被拉低时，将发生模式故障。这会自动将 MODF 位置 1。主模式故障会在以下几方面影响 SPI 接口：

- 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，并禁止 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 SPIx_SR 寄存器执行读或写访问。
2. 然后，对 SPIx_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。安全起见，硬件不允许在 MODF 位置 1 时将 SPE 和 MSTR 位置 1。在从器件中，MODF 位不可置 1，但由前一次多主模式冲突引起时除外。

CRC 错误 (CRCERR)

当 SPIx_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 SPIx_RXCRCR 的值不匹配，SPIx_SR 寄存器中的 CRCERR 标志将置 1。该标志由软件清零。

TI 模式帧格式错误 (FRE)

如果 SPI 在从模式下工作，并配置为符合 TI 模式协议，则在通信进行期间出现 NSS 脉冲时，将检测到 TI 模式帧格式错误。出现此错误时，SPIx_SR 寄存器中的 FRE 标志将置 1。发生错误时不会禁止 SPI，但会忽略 NSS 脉冲，并且 SPI 会等待下一个 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。

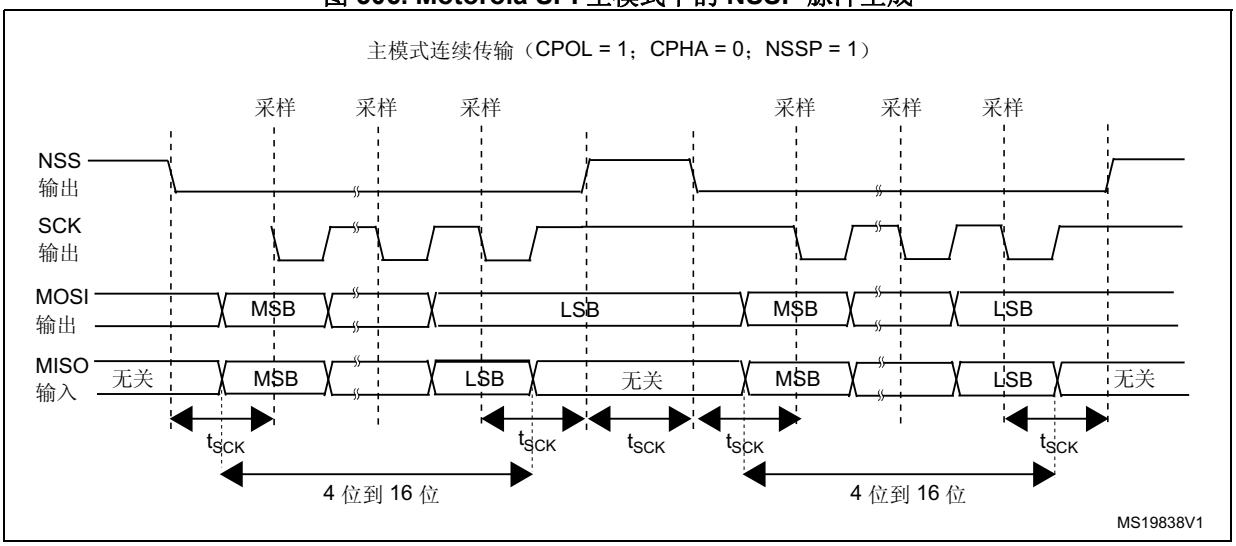
读取 SPIx_SR 寄存器时，将清零 FRE 标志。如果 ERRIE 位置 1，则检测到 NSS 错误时将生成中断。在这种情况下，由于无法保证数据的一致性，应禁止 SPI，并在重新使能从 SPI 后，由主器件重新发起通信。

46.4.12 NSS 脉冲模式

该模式通过 SPIx_CR2 寄存器中的 NSSP 位来激活，只有将 SPI 接口配置为 Motorola SPI 主模式 (FRF=0) 且在第一个边沿捕捉时，该模式才起作用（SPIx_CR1 CPHA = 0，CPOL 设置忽略）。激活后，当 NSS 至少保持一个时钟周期的高电平状态时，两个连续的数据帧传输间将生成 NSS 脉冲。该模式下，从器件可以锁存数据。NSSP 脉冲模式旨在用于具有一个主器件-从器件对的应用。

图 506 给出了使能 NSSP 脉冲模式后的 NSS 引脚管理情况。

图 506. Motorola SPI 主模式下的 NSSP 脉冲生成



注： 当 CPOL = 0 时会出现类似行为。在这种情况下，采样边沿为 SCK 的*上升*沿，NSS 的使能和禁止均参考该采样边沿。

46.4.13 TI 模式

主模式下的 TI 协议

SPI 接口与 TI 协议兼容。可以使用 SPIx_CR2 寄存器的 FRF 位来配置 SPI，以兼容此协议。

时钟极性和相位都被强制为遵循 TI 协议，和 SPIx_CR1 中的设置无关。NSS 管理也特定于 TI 协议，在这种情况下，无法通过 SPIx_CR1 和 SPIx_CR2 寄存器（SSM、SSI 和 SSOE）来对 NSS 管理进行配置。

在从模式下，SPI 波特率预分频器用于控制在当前传输完成时 MISO 引脚切换为高阻态的时刻（请参见*图 507*）。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 信号变为高阻态的延时 ($t_{release}$) 取决于内部重新同步以及通过 SPIx_CR1 寄存器的 BR[2:0] 位设置的波特率值。具体公式如下：

$$\frac{t_{baud_rate}}{2} + 4 \times t_{pclk} < t_{release} < \frac{t_{baud_rate}}{2} + 6 \times t_{pclk}$$

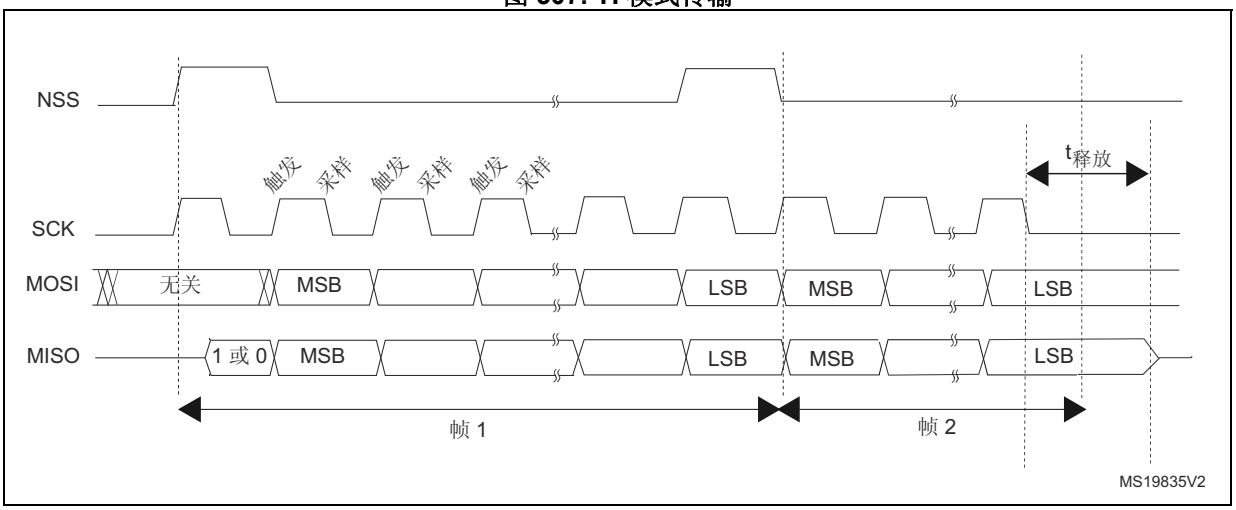
如果从器件在数据帧传输期间检测到错位的 NSS 脉冲，TIFRE 标志将置 1。

如果数据大小等于 4 位或 5 位，处于全双工模式或只发送模式的主器件将使用在 LSB 后再添加一个空数据位的协议。每个周期内，将在该空位（而非 LSB）时钟周期生成 TI NSS 脉冲。

此特性不适用于 Motorola SPI 通信（FRF 位为 0）。

*图 507：TI 模式传输*给出了选择 TI 模式时的 SPI 通信波形。

图 507. TI 模式传输



46.4.14 CRC 计算

为检查发送数据和接收数据的可靠性，使用两个独立的 CRC 计算器。SPI 可提供 CRC8 或 CRC16 计算，而与固定为 8 位或 16 位的数据帧长度无关。对于所有其他的数据帧长度，CRC 均不适用。

CRC 原理

在使能 SPI (SPE = 1) 前，通过将 SPIx_CR1 寄存器中的 CRCEN 位置 1 来使能 CRC 计算。使用值为奇数的可编程多项式对每个位计算 CRC 值。在由 SPIx_CR1 寄存器中的 CPHA 位和 CPOL 位定义的采样时钟边沿进行计算。所计算的 CRC 值在数据块末尾自动进行校验，以及针对由 CPU 或 DMA 管理的传输进行校验。当检测到所接收数据内部计算的 CRC 与发送器发送的 CRC 不匹配时，CRCERR 标志将置 1 以指示数据损坏错误。CRC 计算的正确处理步骤取决于 SPI 配置和所选的传输管理。

注： *多项式值只应为奇数。不支持任何偶数值。*

CPU 管理的 CRC 传输

通信开始后将一直持续到必须发送或接收 SPIx_DR 寄存器中的最后一个数据帧时。之后，SPIx_CR1 寄存器中的 CRCNEXT 位必须置 1，以指示当前处理的数据帧传输后将处理 CRC 帧传输。CRCNEXT 位必须在最后一个数据帧传输结束前置 1。在 CRC 传输期间，CRC 计算将冻结。

所接收的 CRC 以数据字节或数据字的形式存储在 RXFIFO 中。因此仅在 CRC 模式下，接收缓冲区才必须被视为一个 16 位缓冲区且一次仅接收一个数据帧。

CRC 帧的传输通常在数据序列结束时再传输一个数据帧。然而，在设置通过 16 位 CRC 校验的 8 位数据帧时，还需要另外两个帧才能发送完整的 CRC。

接收最后一个 CRC 数据后，将执行自动校验，将接收的值与 SPIx_RXCRC 寄存器中的值进行比较。软件必须校验 SPIx_SR 寄存器中的 CRCERR 标志，以确定数据传输是否损坏。软件通过向 CRCERR 标志写入“0”来将其清零。

接收 CRC 后，CRC 值存储到 RXFIFO 中，且必须在 SPIx_DR 寄存器中进行读取，以将 RXNE 标志清零。

DMA 管理的 CRC 传输

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（在只接收模式下读取 CRC 数据时除外）。CRCNEXT 位不是一定要通过软件来处理。SPI 发送 DMA 通道计数器必须设置为要发送的数据帧数，其中不包括 CRC 帧。在接收器侧，接收的 CRC 值在传输结束时通过 DMA 自动处理，但用户必须注意刷新 RXFIFO 中接收的 CRC 信息（因为该信息始终加载到其中）。在全双工模式下，接收 DMA 通道计数器可设置为要接收的数据帧数，其中包括 CRC，这意味着在通过 16 位 CRC 校验的 8 位数据帧的特殊情况下：

DMA_RX = Numb_of_data + 2

在只接收模式下，DMA 接收通道计数器应仅包含已传输的数据量，而不包括 CRC 计算。之后，基于通过 DMA 实现的完整传输，必须通过软件从 FIFO 中读回所有 CRC 值，因为 FIFO 在该模式下用作单个缓冲区。

如果传输过程中出现损坏，则在数据和 CRC 传输结束时，SPIx_SR 寄存器中的 CRCERR 标志将置 1。

如果使用封装模式，则 LDMA_RX 位需要管理数据数是否为奇数。

复位 SPIx_TXCRC 和 SPIx_RXCRC 值

在 CRC 阶段后对新数据进行采样时，SPIx_TXCRC 和 SPIx_RXCRC 值将自动清零。借此，可使用 DMA 循环模式（只接收模式下不适用）从而不间断地传输数据（中间 CRC 校验阶段会涉及一些数据块）。

如果在通信过程中禁止 SPI，必须按以下顺序进行操作：

1. 禁止 SPI
2. 将 CRCEN 位清零
3. 使能 CRCEN 位
4. 使能 SPI

注： *当 SPI 接口配置为从模式时，一旦 CRCNEXT 信号被释放，NSS 内部信号需要在 CRC 阶段事务期间保持低电平。这就是当 NSS 硬件模式正常应用于从器件时，不能在 NSS 脉冲模式下使用 CRC 计算的原因（有关更多信息，请参见产品勘误表）。*

在 TI 模式下，尽管时钟相位和时钟极性设置固定并且与 SPIx_CR1 寄存器无关，但如果应用 CRC，则 SPIx_CR1 寄存器中必须保持相应的设置（CPOL = 0，CPHA = 1）。此外，CRC 计算必须通过 SPI 禁止序列在会话之间复位，方法是在主器件和从器件两侧重新使能上述 CRCEN 位，否则 CRC 计算可能在该特定模式下损坏。

46.5 SPI 中断

在 SPI 通信过程中，中断可由以下事件产生：

- 发送 TXFIFO 准备就绪，可以装载数据
- 接收 RXFIFO 中接收了数据
- 主模式故障
- 上溢错误
- TI 帧格式错误
- CRC 协议错误

中断可分别进行使能和禁止。

表 275. SPI 中断请求

中断事件	事件标志	使能控制位
发送 TXFIFO 准备就绪，可以装载数据	TXE	TXEIE
RXFIFO 中接收了数据	RXNE	RXNEIE
主模式故障	MODF	ERRIE
上溢错误	OVR	
TI 帧格式错误	FRE	
CRC 协议错误	CRCERR	

46.6 SPI 寄存器

外设寄存器可支持半字（16 位）或字（32 位）访问。此外，SPI_DR 可支持 8 位访问。

46.6.1 SPI 控制寄存器 1 (SPIx_CR1)

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	CRCL	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)。该位通过一条共用的双向数据线来支持半双工通信。双向模式激活时，使 RXONLY 位保持清零状态。

0：选择双线单向通信数据模式

1：选择单线双向数据模式

位 14 **BIDIOE**：双向模式下的输出使能 (Output enable in bidirectional mode)

该位结合 BIDIMODE 位，用于选择双向模式下的传输方向

0：禁止输出（只接收模式）

1：使能输出（只发送模式）

注： 在主模式下，使用 MOSI 引脚；在从模式下，使用 MISO 引脚。

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： 为确保正确操作，只应在禁止 SPI（SPE = “0”）时对此位执行写操作。

位 12 **CRCNEXT**：发送下一个 CRC (Transmit CRC next)

0：下一个发送值来自发送缓冲区

1：下一个发送值来自发送 CRC 寄存器

注： 向 SPIx_DR 寄存器写入最后一个数据后，必须立即对该位执行写操作。

位 11 **CRCL**：CRC 长度 (CRC length)

此位由软件置 1 和清零，用以选择 CRC 长度。

0：8 位 CRC 长度

1：16 位 CRC 长度

注： 为确保正确操作，只应在禁止 SPI（SPE = “0”）时对此位执行写操作。

位 10 **RXONLY**：只接收模式使能 (Receive only mode enabled)。

该位用于使能通过一条双向线专门接收数据的单工通信。当只接收模式激活时，将 BIDIMODE 位保持清零状态。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：禁止输出（只接收模式）

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：　该位不适用于 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 I/O 值。
注：　该位不适用于 SPI TI 模式。

位 7 **LSBFIRST**：帧格式 (Frame format)
0：发送/接收数据时 MSB 在前
1：发送/接收数据时 LSB 在前
注：　1. 正在通信时不应更改此位。
2. 该位不适用于 SPI TI 模式。

位 6 **SPE**：SPI 使能 (SPI enable)
0：禁止外设
1：使能外设
注：　禁止 SPI 时，请按照第 1585 页的禁止 SPI 的步骤中所述的步骤操作。

位 5:3 **BR[2:0]**：波特率控制 (Baud rate control)
000：$f_{PCLK}/2$
001：$f_{PCLK}/4$
010：$f_{PCLK}/8$
011：$f_{PCLK}/16$
100：$f_{PCLK}/32$
101：$f_{PCLK}/64$
110：$f_{PCLK}/128$
111：$f_{PCLK}/256$
注：　正在通信时不应更改这些位。

位 2 **MSTR**：主模式选择 (Master selection)
0：从配置
1：主配置
注：　正在通信时不应更改此位。

位 1 **CPOL**：时钟极性 (Clock Polarity)
0：空闲状态时，CK 为 0
1：空闲状态时，CK 为 1
注：　正在通信时不应更改此位。
除了在 TI 模式下应用 CRC 的情况外，该位不会用于 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：　正在通信时不应更改此位。
除了在 TI 模式下应用 CRC 的情况外，该位不会用于 SPI TI 模式。

46.6.2 SPI 控制寄存器 2 (SPIx_CR2)

SPI control register 2

偏移地址：0x04

复位值：0x0700

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	LDMA_TX	LDMA_RX	FRXTH	DS [3:0]				TXEIE	RXNEIE	ERRIE	FRF	NSSP	SSOE	TXDMAEN	RXDMAEN
	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 **LDMA_TX**：发送的最后一次 DMA 传输 (Last DMA transfer for transmission)

此位用于数据封装模式，用于定义通过 DMA 发送的数据总数为奇数还是偶数。只有 SPIx_CR2 寄存器中的 TXDMAEN 位置 1 且使用封装模式（数据长度 =< 8 位，对 SPIx_DR 的写访问为 16 位宽）时，该位才有意义。当 SPI 禁止时（SPIx_CR1 寄存器中的 SPE = 0），必须对其执行写操作。

0：待传输数据数量为偶数

1：待传输数据数量为奇数

注： 如果 CRCEN 位置 1，请参见第 1585 页的禁止 SPI 的步骤。

位 13 **LDMA_RX**：接收的最后一次 DMA 传输 (Last DMA transfer for reception)

此位用于数据封装模式下，用于定义通过 DMA 接收的数据总数为奇数还是偶数。只有 SPIx_CR2 寄存器中的 RXDMAEN 位置 1 且使用封装模式（数据长度 =< 8 位，对 SPIx_DR 的写访问为 16 位宽）时，该位才有意义。当 SPI 禁止时（SPIx_CR1 寄存器中的 SPE = 0），必须对其执行写操作。

0：待传输数据数量为偶数

1：待传输数据数量为奇数

注： 如果 CRCEN 位置 1，请参见第 1585 页的禁止 SPI 的步骤。

位 12 **FRXTH**：FIFO 接收阈值 (FIFO reception threshold)

该位用于设置触发 RXNE 事件的 RXFIFO 阈值

0：如果 FIFO 占用水平大于或等于 1/2（16 位），将生成 RXNE 事件。

1：如果 FIFO 占用水平大于或等于 1/4（8 位），将生成 RXNE 事件。

位 11:8 **DS [3:0]**：数据大小 (Data size)

这些位用于配置 SPI 传输的数据长度：

0000：未使用
0001：未使用
0010：未使用
0011：4 位
0100：5 位
0101：6 位
0110：7 位
0111：8 位
1000：9 位
1001：10 位
1010：11 位
1011：12 位
1100：13 位
1101：14 位
1110：15 位
1111：16 位

如果软件尝试写入其中一个“未使用”值，这些位将被强制设为“0111”（8 位）。

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 模式下的 CRCERR、OVR 和 MODF，TI 模式下的 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： 只有在禁止 SPI (SPE=0) 后才能对此位执行写操作。

位 3 **NSSP**：NSS 脉冲管理 (NSS pulse management)

此位仅用于主模式。连续传输时，该位允许 SPI 在两个连续数据间生成 NSS 脉冲。单次数据传输时，该位强制 NSS 引脚在传输后变为高电平。

如果 CPHA =“1”或 FRF =“1”，该位无意义。

0：未生成 NSS 脉冲
1：生成 NSS 脉冲

注： 1. 只有在禁止 SPI (SPE=0) 后才能对此位执行写操作。
2. 该位不适用于 SPI TI 模式。

位 2 **SSOE**：SS 输出使能 (SS output enable)

0：在主模式下禁止 SS 输出，SPI 接口可在多主模式配置下工作

1：在主模式下以及使能 SPI 接口后使能 SS 输出。SPI 接口不能在多主模式环境下工作。

注：　该位不适用于 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，即产生 DMA 请求。

0：禁止发送缓冲区 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，即产生 DMA 请求。

0：禁止接收缓冲区 DMA

1：使能接收缓冲区 DMA

46.6.3 SPI 状态寄存器 (SPIx_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
Res.	Res.	Res.	FTLVL[1:0]		FRLVL[2:0]		FRE	BSY	OVR	MODF	CRC ERR	Res.	Res.	TXE	RXNE
			r	r	r	r	r	r	r	r	rc_w0			r	r

位 15:13 保留，必须保持复位值。

位 12:11 **FTLVL[1:0]：FIFO** 发送级别 (FIFO Transmission Level)
这些位将由硬件置 1 和清零。
00：FIFO 为空
01：1/4 FIFO
10：1/2 FIFO
11：FIFO 为满（当 FIFO 阈值大于 1/2 时即视为满）

位 10:9 **FRLVL[1:0]**：FIFO 接收级别 (FIFO reception level)
这些位将由硬件置 1 和清零。
00：FIFO 为空
01：1/4 FIFO
10：1/2 FIFO
11：FIFO 已满
注： 使能 CRC 计算时，这些位不适用于 SPI 仅接收模式。

位 8 **FRE**：帧格式错误 (Frame format error)
该标志在 TI 从模式下用于 SPI。请参见*第 46.4.11 节：SPI 错误标志*。
此标志由硬件置 1，在读取 SPIx_SR 时由软件复位。
0：未发生帧格式错误
1：发生帧格式错误

位 7 **BSY**：忙标志 (Busy flag)
0：SPI 不忙
1：SPI 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。
注： BSY 标志必须谨慎使用：请参见第 46.4.10 节：SPI 状态标志和第 1585 页的禁止 SPI 的步骤。

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，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，由软件序列复位。有关软件序列，请参见*第 1594 页的模式故障 (MODF)*一节。

位 4 **CRCERR**：CRC 错误标志 (CRC error flag)
0：接收到的 CRC 值与 SPIx_RXCRCR 值匹配
1：接收到的 CRC 值与 SPIx_RXCRCR 值不匹配
此标志由硬件置 1，通过软件写入 0 来清零。

位 3:2 保留，必须保持复位值。

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

46.6.4 SPI 数据寄存器 (SPIx_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。
数据寄存器用于连接 Rx 和 Tx FIFO。读取数据寄存器时，将访问 RxFIFO；而写入数据寄存器时，将访问 TxFIFO（请参见*第 46.4.9 节：数据发送和接收过程*）。
注：　数据始终右对齐。写入寄存器时将忽略未使用位，读取寄存器时会将未使用位读为 0。Rx 阈值设置必须始终与当前使用的读访问相符。

46.6.5 SPI CRC 多项式寄存器 (SPIx_CRCPR)

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：　多项式值只应为奇数。不支持任何偶数值。

46.6.6 SPI 接收 CRC 寄存器 (SPIx_RXCRCR)

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 SPIx_CR1 寄存器中的 CRCEN 位写入 1 时，此寄存器复位。 CRC 通过 SPIx_CRCPR 寄存器中编程的多项式连续计算。

CRC 帧格式设置为 8 位长度（SPIx_CR1 的 CRCL 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 CRC 帧格式（SPIx_CR1 寄存器的 CRCL 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。

当 BSY 标志置 1 时，读取此寄存器可能返回一个不正确的值。

46.6.7 SPI 发送 CRC 寄存器 (SPIx_TXCRCR)

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 SPIx_CR1 寄存器中的 CRCEN 位写入 1 时，此寄存器复位。CRC 通过 SPIx_CRCPR 寄存器中编程的多项式连续计算。

CRC 帧格式设置为 8 位长度（SPIx_CR1 的 CRCL 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 CRC 帧格式（SPIx_CR1 寄存器的 CRCL 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。

当 BSY 标志置 1 时，读取此寄存器可能返回一个不正确的值。

46.6.8 SPI 寄存器映射

表 276 给出了 SPI 寄存器映射和复位值。

表 276. SPI 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	**SPIx_CR1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BIDIMODE	BIDIOE	CRCEN	CRCNEXT	CRCL	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	**SPIx_CR2**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LDMA_TX	LDMA_RX	FRXTH	DS[3:0]				TXEIE	RXNEIE	ERRIE	FRF	NSSP	*SSOE*	TXDMAEN	RXDMAEN
	Reset value																		0	0	0	0	1	1	1	0	0	0	0	0	0	0	0
0x08	**SPIx_SR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FTLVL[1:0]		FRLVL[1:0]		FRE	BSY	OVR	MODF	CRCERR	Res.	Res.	TXE	RXNE
	Reset value																				0	0	0	0	0	0	0	0	0			1	0
0x0C	**SPIx_DR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DR[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x10	**SPIx_CRCPR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CRCPOLY[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1
0x14	**SPIx_RXCRCR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RxCRC[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x18	**SPIx_TXCRCR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TxCRC[15:0]															
	Reset value																	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*第 78 页的第 2.2.2 节*。

47 串行音频接口 (SAI)

47.1 简介

SAI 接口（串行音频接口）灵活性高、配置多样，可支持多种音频协议。该接口适用于许多立体声或单声道应用。例如，它可配置为支持 I2S 标准、LSB 或 MSB 对齐、PCM/DSP、TDM 和 AC’97 等协议。将音频模块配置为发送器时，SAI 接口可提供 SPDIF 输出。

SAI 通过两个完全独立的音频子模块来实现这种灵活性和可配置性。每个模块都有自己的时钟发生器和 I/O 线控制器。

SAI 可以配置为主模式或配置为从模式。音频子模块既可作为接收器，又可作为发送器；既可与另一模块同步，又可以不同步。

SAI 可与其他 SAI 相连接来同步运行。

47.2 SAI 主要特性

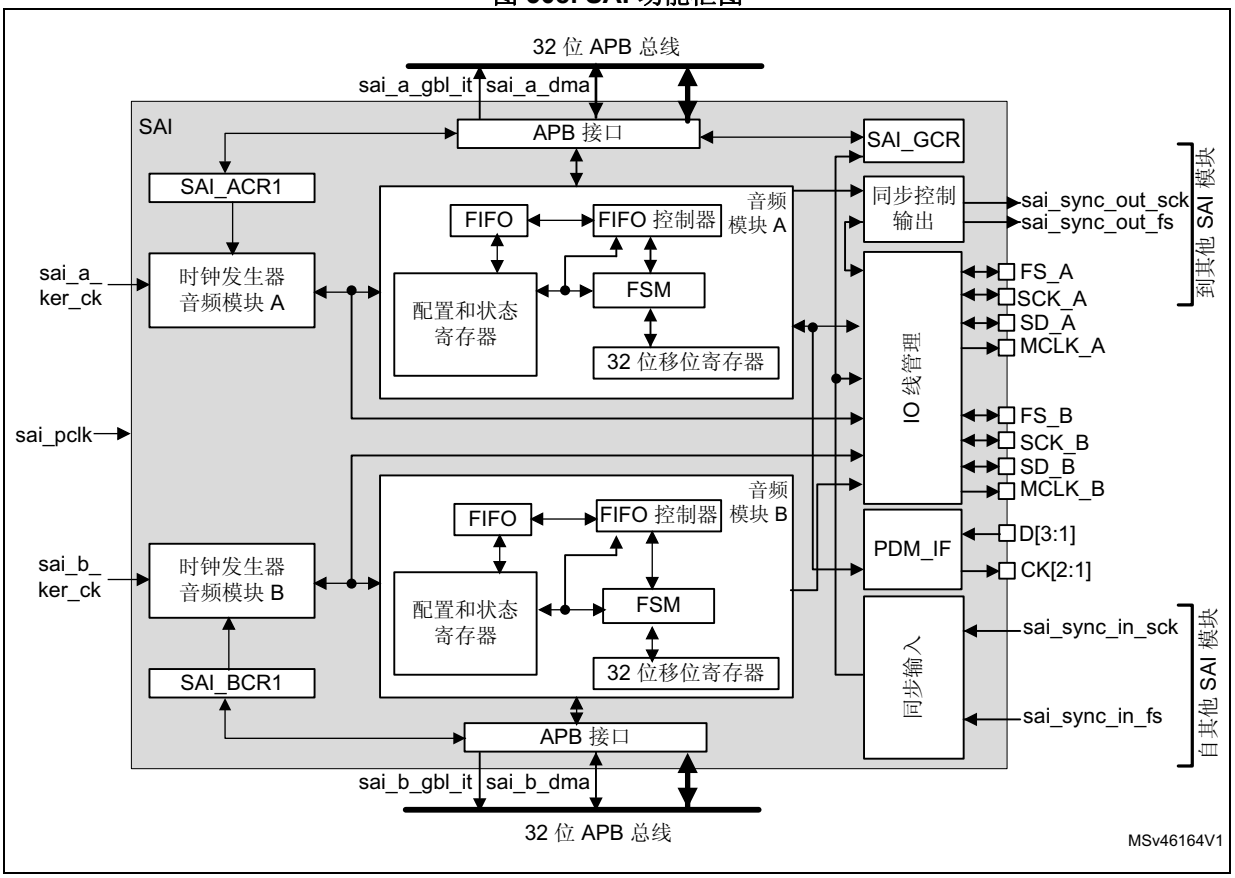
- 具有两个独立的音频子模块，子模块既可作为接收器，也可作为发送器，并带有自身的 FIFO。
- 每个音频子模块集成多达 8 个字的 FIFO。
- 两个音频子模块间可以是同步或异步模式。
- 多个 SAI 之间可实现同步。
- 两个音频子模块的主/从配置相互独立。
- 当两个音频子模块都配置为以主模式工作时，每个子模块的时钟发生器采用独立的音频采样频率。
- 数据大小可配置：8 位、10 位、16 位、20 位、24 位或 32 位。
- 音频协议：I2S、LSB 或 MSB 对齐、PCM/DSP、TDM、AC'97。
- PDM 接口，支持多达 4 对麦克风。
- 如果需要，可提供 SPDIF 输出。
- 高达 16 个大小可配置的 Slot。
- 每帧的位数可配置。
- 帧同步有效电平可配置（偏移、位长、电平）。
- 可配置 Slot 中第一个有效位的位置。
- 支持 LSB 或 MSB 数据传输。
- 支持静音模式。
- 具有立体声/单声道音频帧功能。
- 通信时钟选通边沿可配置 (SCK)。
- 错误标志对应相应中断（分别使能时）。
  - 上溢和下溢检测
  - 从模式下的帧同步信号提前检测
  - 从模式下的帧同步信号滞后检测
  - 接收时编码解码器未针对 AC'97 模式就绪
- 支持如下中断源（使能时）：
  - 错误
  - FIFO 请求
- DMA 接口有 2 个通道。

47.3 SAI 功能说明

47.3.1 SAI 框图

图 508 给出了 SAI 框图，*表 277* 和*表 278* 列出了 SAI 的内部信号和外部信号。

图 508. SAI 功能框图



SAI 主要由两个各自带有时钟发生器的音频子模块组成。每个音频模块集成一个 32 位移位寄存器，该寄存器由模块自身的功能状态机控制。数据存储和读取都是通过专用的 FIFO 来完成。FIFO 可通过 CPU 访问，也可通过 DMA 访问以减轻 CPU 的通信负担。每个音频模块是独立的。这两个音频子模块可彼此同步。

I/O 线控制器管理 SAI 中指定音频模块的 4 个专用引脚（SD、SCK、FS、MCLK）。如果将两个子模块声明为同步模块，则可以共用其中某些引脚，从而留出一些引脚用作通用 I/O。MCLK 引脚是否可用作输出引脚取决于实际应用和解码要求以及音频模块是否配置为主模式。

如果将一个 SAI 配置为与其他 SAI 同步运行，可以释放更多 I/O（引脚 SD_x 除外）。

可配置功能状态机来处理多种音频协议。一些寄存器用于设置所需协议（音频帧波形发生器）。

音频子模块在主模式或从模式下均可用作发送器或接收器。主模式意味着从 SAI 生成 SCK_x 位时钟和帧同步信号，而从模式则意味着 SCK_x 位时钟和帧同步信号来自另一外部或内部主器件。在特殊情况下，FS 信号方向与主模式或从模式定义不直接相关。在 AC'97 协议中，即使 SAI（链接控制器）设置为消耗 SCK 时钟，FS 信号也会是 SAI 输出（从模式下也是如此）。

注： *为方便阅读此部分，符号 SAI_x 指 SAI_A 或 SAI_B，其中 "x" 代表 SAI A 子模块或 SAI B 子模块。*

47.3.2 SAI 引脚和内部信号

表 277. SAI 内部输入/输出信号

内部信号名称	信号类型	说明
sai_a_gbl_it/ sai_b_gbl_it	输出	音频模块 A 和 B 全局中断。
sai_a_dma、 sai_b_dma	输入/输出	音频模块 A 和 B DMA 确认与请求。
sai_sync_out_sck、 sai_sync_out_fs	输出	与其他 SAI 模块交换的内部时钟和帧同步输出信号。
sai_sync_in_sck、 sai_sync_in_fs	输入	与其他 SAI 模块交换的内部时钟和帧同步输入信号。
sai_a_ker_ck/ sai_b_ker_ck	输入	音频模块 A/B 内核时钟。
sai_pclk	输入	APB 时钟。

表 278. SAI 输入/输出引脚

名称	信号类型	注释
SAI_SCK_A/B	输入/输出	音频模块 A/B 位时钟。
SAI_MCLK_A/B	输出	音频模块 A/B 主时钟。
SAI_SD_A/B	输入/输出	用于模块 A/B 的数据线。
SAI_FS_A/B	输入/输出	用于音频模块 A/B 的帧同步线。
SAI_CK[2:1]	输出	PDM 比特流时钟。
SAI_D[3:1]	输入	PDM 比特流数据。

47.3.3 SAI 的主要模式

SAI 的每个音频子模块均可通过所选音频模块的 SAI_xCR1 寄存器中的 MODE 位配置为主模式或从模式。

主模式

在主模式下，SAI 会向连接的外部器件提供时钟信号：

- 位时钟和帧同步分别在引脚 SCK_x 和 FS_x 上输出。
- 如果需要，SAI 也可以在 MCLK_x 引脚上生成主时钟。

SCK_x、FS_x 和 MCLK_x 均配置为输出。

从模式

SAI 从外部器件接收时钟信号。

- 如果将 SAI 子模块配置为异步模式，则 SCK_x 引脚和 FS_x 引脚将被配置为输入。
- 如果将 SAI 子模块配置为与另一个 SAI 接口或第二个音频子模块同步运行，则会释放相应的 SCK_x 引脚和 FS_x 引脚以用作通用 I/O。

在从模式下，不使用 MCLK_x 引脚，可将其分配给其他功能。

建议在使能主器件前先使能从器件。

配置和使能 SAI 模式

可通过相应音频模块的 SAI_xCR1 寄存器中的 MODE 位将每个音频子模块独立定义为发送器或接收器。为此，SAI_SD_x 引脚将分别被配置为输出或输入。

可使用两个不同的 MCLK 和 SCK 时钟频率配置同一 SAI 中的两个主音频模块。在这种情况下，必须将这两个音频模块配置为异步模式。

SAI 中的每个音频模块均通过 SAI_xCR1 寄存器中的 SAIEN 位使能。在从模式下，此位一经激活，发送器或接收器便会对时钟线、数据线和同步线上的活动敏感。

在主 TX 模式下，即使 FIFO 中没有数据，使能音频模块也会立即为外部从模块产生位时钟，但 FS 信号的产生受 FIFO 中是否存在数据的控制。FIFO 接收到要发送的第一个数据后，此数据将输出到外部从模块。如果 FIFO 中没有要发送的数据，则随后将在音频帧中传送值 0，并会产生一个下溢标志。

在从模式下，使能音频模块时和检测到 SOF 位时开始音频帧。

在从 TX 模式下，使能音频模块后的第一个帧上不可能出现下溢事件，因为此时的强制操作顺序如下：

1. 通过软件或 DMA 写入 SAI_xDR。
2. 等待至 FIFO 阈值 (FLH) 标志与 000b（FIFO 为空）不同。
3. 使能音频模块为从发送模式。

47.3.4 SAI 同步模式

同步模式存在两个同步级别，即音频子模块级别和 SAI 级别。

内部同步

音频子模块可以和同一 SAI 中的另一个音频子模块同步运行。在这种情况下，二者将共用位时钟和帧同步信号，以减少通信时占用的外部引脚数。配置为同步模式的音频模块将释放其 SCK_x、FS_x 和 MCLK_x 引脚以用作 GPIO，而配置为异步模式的音频模块将使用其 FS_x、SCK_x 和 MCLK_x I/O 引脚（如果该音频模块被配置为主模块）。

通常，同步模式下的音频模块可用于在全双工模式下配置 SAI。两个音频模块中的一个可配置为主模块，另一个为从模块；也可将两个均配置为从模块；一个模块为异步模块（SAI_xCR1 中的相应位 SYNCEN[1:0] 设置为 00），另一个为同步模块（SAI_xCR1 中的相应位 SYNCEN[1:0] 设置为 01）。

注： *由于存在内部重新同步阶段，因此 PCLK APB 频率必须大于比特率时钟频率的二倍。*

外部同步

音频子模块也可以配置为与其他 SAI 同步运行，具体方法如下：

1. 配置为其他 SAI 的同步源的 SAI 必须定义其音频子模块中的哪个子模块应向其他 SAI 提供 FS 和 SCK 信号。此操作通过编程 SYNCOUT[1:0] 位完成。
2. 接收同步信号的 SAI 必须通过为 SYNCIN[1:0] 位设置合适的值来选择哪个 SAI 将提供同步。之后，对于这两个 SAI 音频子模块的每个子模块，用户都必须通过 SYNCEN 位指定其是否与其他 SAI 同步运行。

注： *SYNCIN[1:0] 位和 SYNCOUT[1:0] 位位于 SAI_GCR 寄存器中，SYNCEN 位位于 SAI_xCR1 寄存器中。*

如果指定 SAI 中的两个音频子模块都需要与另一个 SAI 同步，则可以选择以下配置之一：

- 通过 SYNCEN[1:0] 位将每个音频子模块均配置为与另一个 SAI 模块同步。
- 通过 SYNCEN[1:0] 位将一个音频子模块配置为与另一个 SAI 模块同步。随后，通过 SYNCEN[1:0] 位将其他音频子模块配置为与第二个 SAI 音频子模块同步。

下表显示了如何根据所使用的 SAI 模块选择合适的同步信号。例如，SAI2 可通过将 SAI2 SYNCIN 设置为 0 来选择通过 SAI1 进行同步。如果 SAI1 要选择通过 SAI2 进行同步，则必须将 SAI1 SYNCIN 置 1。标注为“保留”的位不可使用。

表 279. 外部同步选择

模块实例	SYNCIN= 3	SYNCIN= 2	SYNCIN= 1	SYNCIN= 0
SAI1	保留	保留	SAI2 同步	保留
SAI2	保留	保留	保留	SAI1 同步

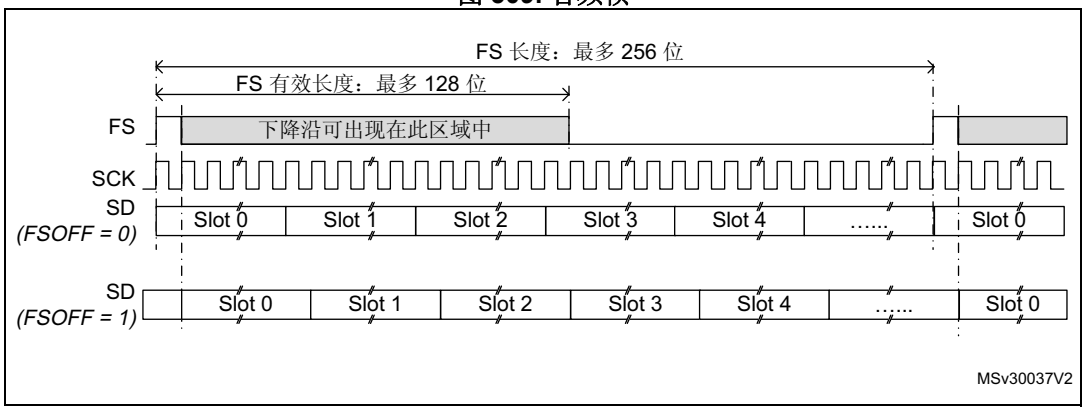
47.3.5 音频数据大小

通过配置 SAI_xCR1 寄存器中的 DS[2:0] 位，配置音频帧的数据大小。数据大小可以是 8 位、10 位、16 位、20 位、24 位或 32 位。在传输期间，将首先发送数据的 MSB 或 LSB，具体取决于 SAI_xCR1 寄存器中的 LSBFIRST 位的配置。

47.3.6 帧同步

FS 信号用作音频帧中的帧同步信号 (SOF)。此信号的波形完全可配置，以在帧同步时，支持各种具有特殊规格的音频协议。这一可重新配置性通过寄存器 SAI_xFRCR 来实现。*图 509* 说明了这种灵活性。

图 509. 音频帧



在 AC'97 模式或 SPDIF 模式下（SAI_xCR1 寄存器中的位 PRTCFG[1:0] = 10 或 PRTCFG[1:0] = 01），帧同步信号的波形被强制配置为支持 AC'97 协议。SAI_xFRCR 寄存器值被忽略。

每个音频模块相互独立，因此均需要特定的配置。

帧长度

- 主模式

  将 SAI_xFRCR 寄存器中的 FRL[7:0] 位域置 1，可将音频帧的长度配置为最长 256 个位时钟周期。

  如果帧长度大于为该帧声明的 Slot 数，则要发送的剩余位将用 0 填充，或者 SD 线将释放为高阻态，具体取决于 SAI_xCR2 寄存器中的位 TRIS 的状态（见 *FS 信号的作用*）。在接收模式下，剩余位被忽略。

  如果将位 NOMCK 清零，(FRL+1) 必须等于 2 的几次幂（从 8 到 256）以确保音频帧的每个位时钟周期都包含整数个 MCLK 脉冲。

  如果将位 NOMCK 置 1，则 (FRL+1) 字段可采用从 8 到 256 的任意值。请参见*第 47.3.8 节：SAI 时钟发生器*。

- 从模式

  音频帧的长度主要用于指定由外部主模块向从模块发送的每个音频帧的位时钟周期数。它主要用于从主模块中检测音频帧传输期间出现的提前或滞后帧同步信号。这种情况下会出现错误。有关详细信息，请参见*第 47.3.14 节：错误标志*。

  在从模式下，SAI_xFRCR 寄存器中的 FRL[7:0] 位的配置不受任何限制。

帧中的位数等于 FRL[7:0] + 1。

音频帧中要传输的最小位数为 8。

帧同步极性

SAI_xFRCR 寄存器中的 FSPOL 位用于设置 FS 引脚的有效极性，通过该极性来启动帧。SOF 信号对边沿敏感。

在从模式下，音频模块等待一个有效帧来启动发送或接收。SOF 信号与此信号同步。只有通信期间未检测到 SOF 信号并且 SOF 信号与预期 SOF 信号相同时，帧同步极性才有效（请参见*第 47.3.14 节：错误标志*）。

在主模式下，每次音频帧完成时均会发送帧同步信号，直至 SAI_xCR1 寄存器中的 SAIEN 位清零。如果前一个音频帧结束时 FIFO 中不存在数据，则将按照*第 47.3.14 节：错误标志*所述管理下溢条件，但音频通信流不会中断。

帧同步有效电平长度

SAI_xFRCR 寄存器的 FSALL[6:0] 位用于配置帧同步信号的有效电平的长度。该长度可设置为 1 到 128 个位时钟周期。

例如，有效长度在 I2S、LSB 或 MSB 对齐模式下为帧长的一半，在 PCM/DSP 或 TDM 模式下为 1 位。

帧同步偏移

基于应用中支持的音频协议（例如 I2S 标准协议和 MSB 对齐协议），可以在发送音频帧的最后一位或第一位时将帧同步信号置为有效。通过 SAI_xFRCR 寄存器中的 FSOFF 位选择两个配置之一。

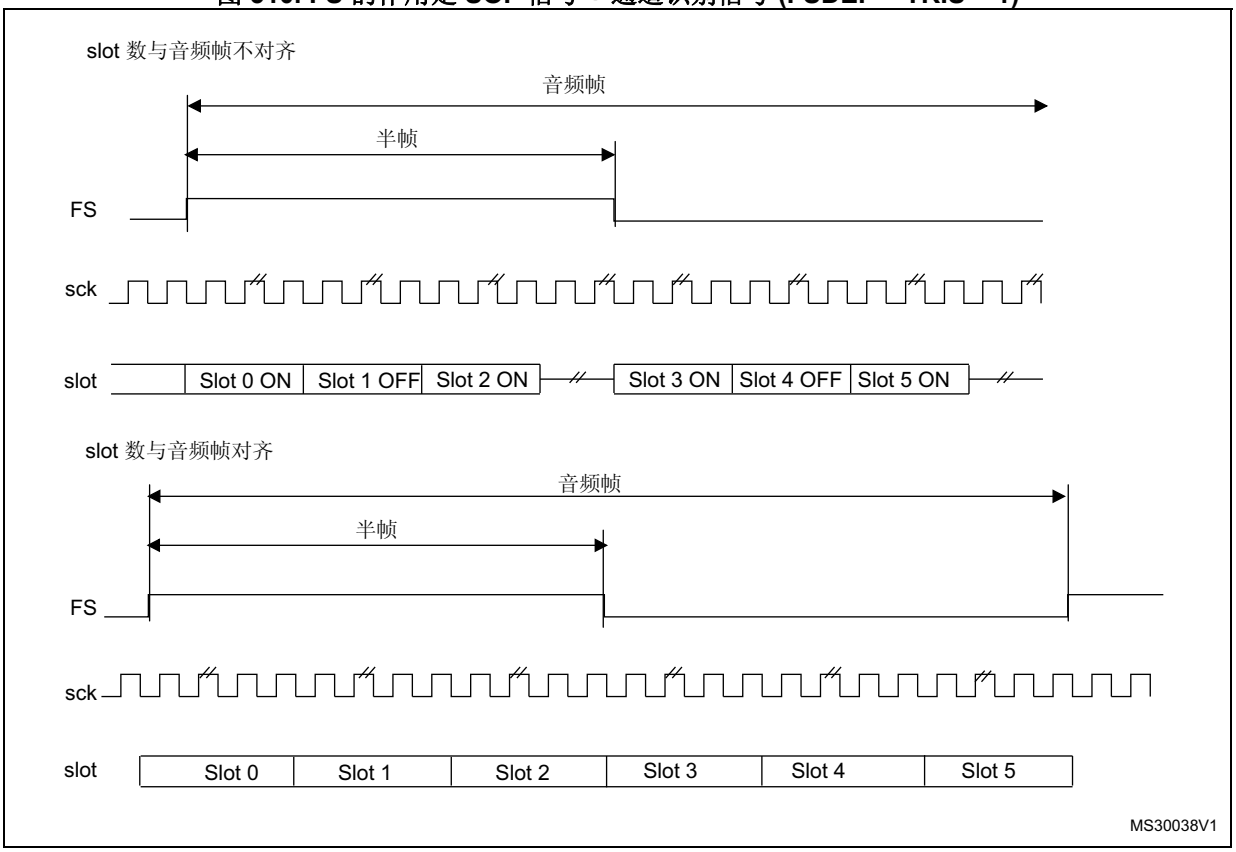
FS 信号的作用

FS 信号可以有不同的含义，具体取决于 FS 的功能。SAI_xFRCR 寄存器中的 FSDEF 位用于选择 FS 信号的含义。

- 0：SOF 信号，例如 PCM/DSP、TDM、AC’97 和音频协议。
- 1：音频帧内的 SOF 信号和通道识别信号，例如 I2S、MSB 或 LSB 对齐协议。

当 FS 信号被视为帧内的 SOF 信号和通道识别信号时，声明的 Slot 数必须是一半用于左通道，一半用于右通道。如果半个音频帧上的位时钟周期数大于某个通道的专用 Slot 数，若 TRIS = 0，则 SAI_xCR2 寄存器中的剩余位时钟周期将发送 0。否则若 TRIS = 1，SD 线将释放为高阻态。接收时，直到通道发生变化，才会考虑剩余位时钟。

图 510. FS 的作用是 SOF 信号 + 通道识别信号 (FSDEF = TRIS = 1)

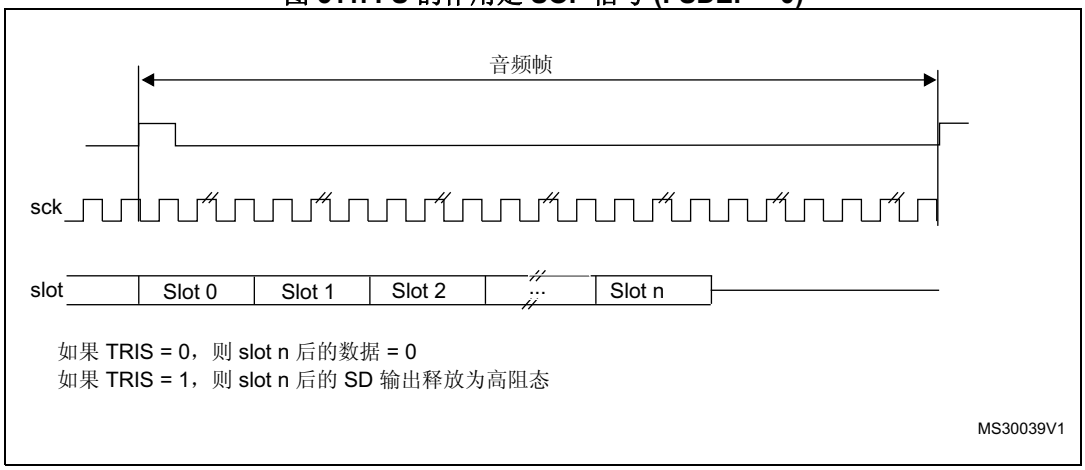


1. 帧长度应为偶数。

如果 SAI_xFRCR 中的 FSDEF 位保持清零状态，则 FS 信号等效于 SOF 信号，如果 SAI_xSLOTR 中的 NBSLOT[3:0] 位定义的 Slot 数乘以 SAI_xSLOTR 中的 SLOTSZ[1:0] 位配置的 Slot 位数所得的结果小于帧大小（SAI_xFRCR 寄存器中的 FRL[7:0] 位），则：

- 如果 SAI_xCR2 寄存器中的 TRIS = 0，则最后一个 Slot 后的剩余位将强制为 0，直至发送器的帧结束。
- 如果 TRIS = 1，则传输这些剩余位时，数据线将释放为高阻态。在接收模式下，这些位被丢弃。

图 511. FS 的作用是 SOF 信号 (FSDEF = 0)



在发送模式下将音频模块配置为获取 SD 线上的 SPDIF 输出时，将不使用 FS 信号。相应的 FS I/O 会被释放，留作他用。

47.3.7 Slot 配置

Slot 是音频帧中的基本元素。音频帧中的 Slot 数等于 NBSLOT[3:0] + 1。

每个音频帧的最大 Slot 数固定为 16。

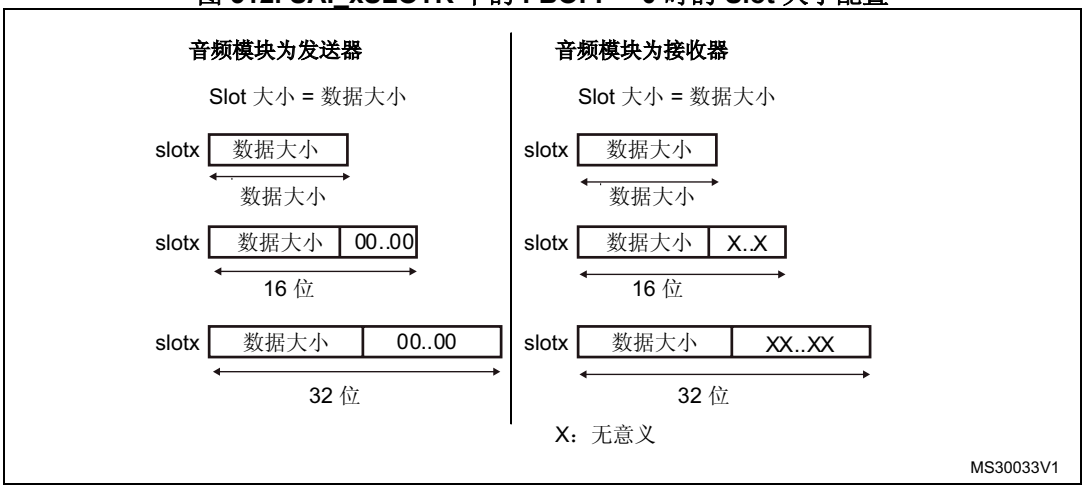
对于 AC'97 协议或 SPDIF 协议（位 PRTCFG[1:0] = 10 或 PRTCFG[1:0] = 01 时），Slot 数自动按照协议规范设置，NBSLOT[3:0] 的值被忽略。

通过设置 SAI_xSLOTR 寄存器的 SLOTEN[15:0] 位，可将各个 Slot 定义为有效 Slot 或无效 Slot。

传输一个无效 Slot 时，SD 数据线将强制为 0 或释放为高阻态，具体取决于 TRIS 位在发送模式下的配置（请参见 *无效 Slot 上的输出数据线管理*）。在接收模式下，从该 Slot 结束后接收的数据被忽略。结果，不会有 FIFO 访问，也不会有与此无效 Slot 状态相关的 FIFO 读/写请求。

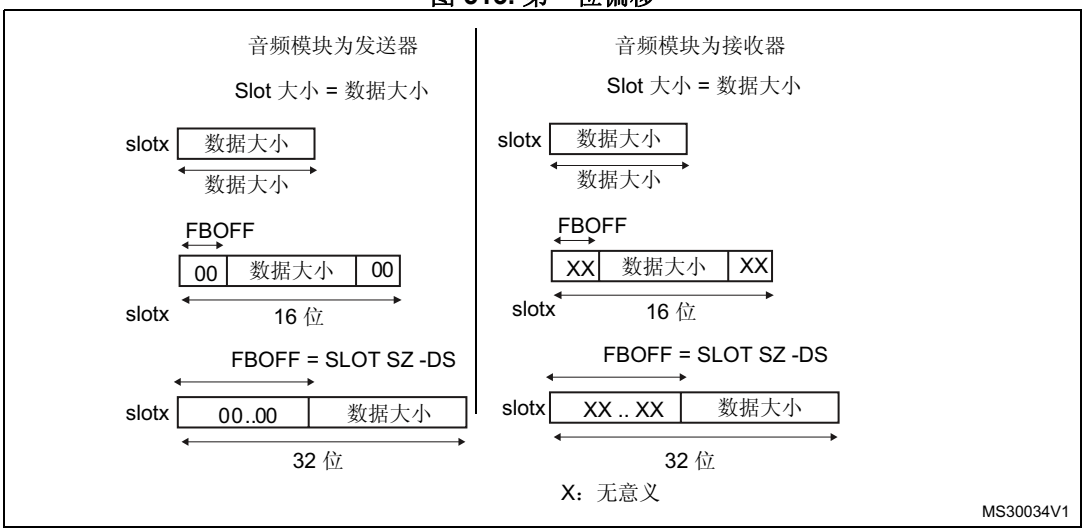
Slot 大小也可配置，如 *图 512* 所示。通过将 SAI_xSLOTR 寄存器中的 SLOTSZ[1:0] 位置 1 来选择 Slot 大小。该大小适用于音频帧中的每个 Slot。

图 512. SAI_xSLOTR 中的 FBOFF = 0 时的 Slot 大小配置



可以选择 Slot 内要传输的第一个数据位的位置，此偏移通过 SAI_xSLOTR 寄存器中的 FBOFF[4:0] 位配置。在发送模式下，将从 Slot 开始时注入 0 值，直至到达此偏移位置。接收时，偏移阶段中的位被忽略。此特性适用于 LSB 对齐协议（如果偏移等于 Slot 大小减去数据大小）。

图 513. 第一位偏移



要避免出现故障 SAI 行为，必须遵循以下条件：

FBOFF ≤ (SLOTSZ - DS)

DS ≤ SLOTSZ

NBSLOT x SLOTSZ ≤ FRL（帧长度）

SAI_xFRCR 寄存器中的 FSDEF 位置 1 时，Slot 数必须为偶数。

在 AC’97 和 SPDIF 协议（位 PRTCFG[1:0] = 10 或 PRTCFG[1:0] = 01）中，Slot 大小按照 *第 47.3.11 节：AC’97 链路控制器*中的定义自动设置。

47.3.8 SAI 时钟发生器

每个音频子模块都有自己的时钟发生器，旨在使这两个模块完全独立。这两个时钟发生器的功能没有任何区别。

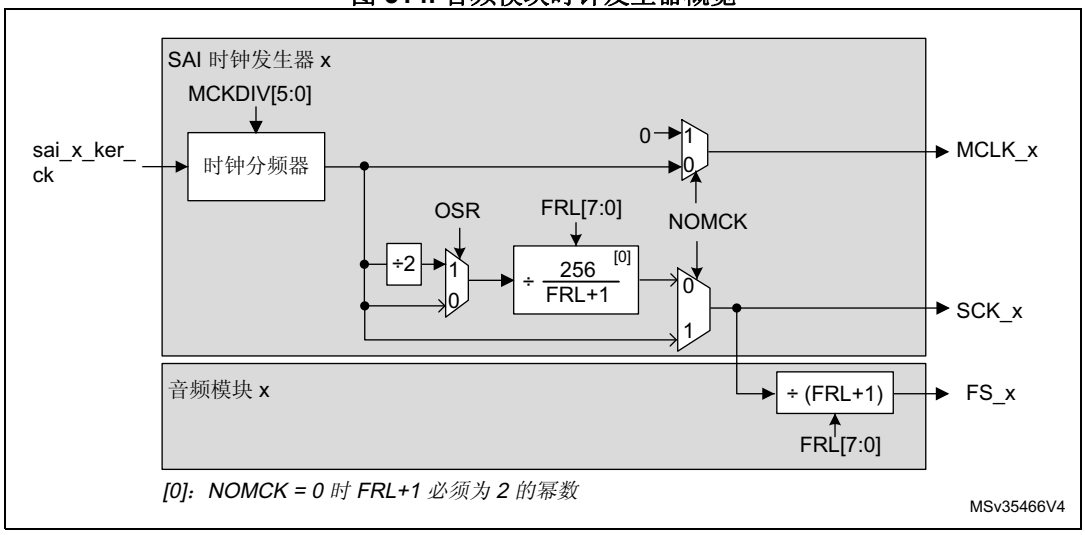
将音频模块配置为主模式时，时钟发生器将提供位时钟 (SCK_x) 以及用于外部解码器的主时钟 (MCLK_x)。帧同步 (FS_x) 还可源自时钟发生器提供的信号。SAI 时钟发生器的时钟源 (sai_x_ker_ck) 由芯片时钟控制器 (RCC) 提供。

将音频模块定义为从模式时，时钟发生器将关闭。NOMCK、MCKDIV 和 OSR 位的值将被忽略。此外，MCLK_x I/O 引脚也会被释放，可用作通用 I/O。

(SCK_x) 的位时钟选通边沿可通过 SAI_xCR1 寄存器中的 CKSTR 位配置。此位对主模式和从模式均有效。

图 514 给出了音频模块时钟发生器的架构。

图 514. 音频模块时钟发生器概览



SAI_xCR1 寄存器的 NOMCK 位用于定义是否生成主时钟。

将 SAI 用于主模式下时，时钟发生器配置有所不同，具体取决于是否需要提供主时钟 (MCLK_x)。

如果 NOMCK 置 1，则不生成主时钟，用户可以更灵活地选择帧长度和帧同步频率。此外，如果该引脚被配置为 GPIO 外设中的 SAI 引脚，则 MCLK_x 信号将驱动为低电平。MCKDIV 仍可用于将 SCK_x 时钟调整为所需频率。

如果 NOMCK 置 0，则会生成主时钟，并且可将主时钟用作外部解码器的参考时钟。在这种情况下，帧同步与主时钟之间的频率比为固定值 512 或 256，帧长度必须为 2 的幂数。下面进行了详细说明。

存在 MCLK (NOMCK = 0) 时的时钟发生器编程

此时，MCLK_x 频率将为：

$$F_{MCLK_x} = 256 \times F_{FS_x} \text{（OSR=0 时）}$$

$$F_{MCLK_x} = 512 \times F_{FS_x} \text{（OSR=1 时）}$$

MCKDIV 不为 0 时，MCLK_x 频率由如下公式计算得出：

$$F_{MCLK_x} = \frac{F_{sai_x_ker_ck}}{MCKDIV}$$

帧同步频率由如下公式计算得出：

$$F_{FS_x} = \frac{F_{sia_x_ker_ck}}{MCKDIV \times (OSR + 1) \times 256}$$

位时钟 (SCK_x) 的频率由以下表达式计算得出：

$$F_{SCK_x} = \frac{F_{MCLK_x} \times (FRL + 1)}{(OSR + 1) \times 256}$$

注： *如果 NOMCK = 0，则 (FRL+1) 必须为 2 的幂数。此外，(FRL+1) 必须介于 8 到 256 之间（请参见 FS 信号的作用一节）。*

MCKDIV 分频比为奇数时，MCLK 的占空比不会为 50%。如果 MCKDIV 为奇数、OSR 等于 0 或 (FRL+1) = 2^8，位时钟信号 (SCK_x) 的占空比也可以不是 50%。

建议将 MCKDIV 配置为偶数或较大的值（大于 10）。

请注意，MCKDIV = 0 时得到的结果与 MCKDIV = 1 时得到的结果相同。

无 MCLK (NOMCK = 1) 时的时钟发生器编程

MCKDIV 不为 0 时，SCK_x 频率如下公式计算得出：

$$F_{SCK_x} = \frac{F_{sai_x_ker_ck}}{MCKDIV}$$

帧同步 (FS_x) 的频率由如下公式计算得出：

$$F_{FS_x} = \frac{F_{sai_x_ker_ck}}{(FRL + 1) \times MCKDIV}$$

注： *NOMCK = 0 时，(FRL+1) 可采用 8 到 256 之间的任意值。*

请注意，MCKDIV = 0 时得到的结果与 MCKDIV = 1 时得到的结果相同。

时钟发生器编程示例

表 280 列出了一些针对 48 kHz、96 kHz 和 192 kHz 频率的编程示例。

表 280. 时钟发生器编程示例

输入 sai_x_ker_ck 时钟频率	MCLK	F_{MCLK}/F_{FS}	FRL (1)	OSR	NOMCK	MCKDIV[5:0]	音频采样频率 (F_{FS})
98.304 MHz	是	512	2^N-1	1	0	0 或 1	192 kHz
		512	2^N-1	1	0	2	96 kHz
		512	2^N-1	1	0	4	48 kHz
		256	2^N-1	0	0	2	192 kHz
		256	2^N-1	0	0	4	96 kHz
		256	2^N-1	0	0	8	48 kHz
	否	-	63	-	1	8	192 kHz
		-	63	-	1	16	96 kHz
		-	63	-	1	32	48 kHz

1. N 表示介于 3 到 8 之间的整数值。

47.3.9 内部 FIFO

SAI 中的每个音频模块都有自己的 FIFO。根据模块是定义为发送器还是接收器，可相应的读取或写入其 FIFO。因此只存在一个 FIFO 请求与 SAI_xSR 寄存器中的 FREQ 位相关。

如果 SAI_xIM 寄存器中的 FREQIE 位使能，将产生中断。这取决于：

- FIFO 阈值设置（SAI_xCR2 中的 FLVL 位）
- 通信方向 （发送器或接收器）。请参见*在发送模式下产生中断*和*在接收模式下产生中断*。

在发送模式下产生中断

根据发送模式下的 FIFO 配置产生中断：

- 当 SAI_xCR2 寄存器中的 FIFO 阈值位配置为 FIFO 为空时（FTH[2:0] 置为 000b），如果寄存器中没有数据（SAI_xSR 中的 FLVL[2:0] 位小于 001b），将产生中断（SAI_xSR 寄存器中的 FREQ 位由硬件置 1）。当 FIFO 不再为空时（SAI_xSR 中的 FLVL[2:0] 位不是 000b），即 FIFO 中存储一个或多个数据时，此中断（SAI_xSR 寄存器中的 FREQ 位）由硬件清零。
- 当 SAI_xCR2 寄存器中的 FIFO 阈值位配置为 FIFO 四分之一满时（FTH[2:0] 置为 001b），如果不到四分之一的 FIFO 包含数据（SAI_xSR 中的 FLVL[2:0] 位小于 010b），将产生中断（SAI_xSR 寄存器中的 FREQ 位由硬件置 1）。当至少四分之一的 FIFO 包含数据时（SAI_xSR 中的 FLVL[2:0] 位大于等于 010b），此中断（SAI_xSR 寄存器中的 FREQ 位）由硬件清零。
- 当 SAI_xCR2 寄存器中的 FIFO 阈值位配置为 FIFO 半满时（FTH[2:0] 置为 010b），如果不到一半的 FIFO 包含数据（SAI_xSR 中的 FLVL[2:0] 位小于 011b），将产生中断（SAI_xSR 寄存器中的 FREQ 位由硬件置 1）。当至少一半的 FIFO 包含数据时（SAI_xSR 中的 FLVL[2:0] 位大于或等于 011b），此中断（SAI_xSR 寄存器中的 FREQ 位）由硬件清零。

- 当 SAI_xCR2 寄存器中的 FIFO 阈值位配置为 FIFO 四分之三满时（FTH[2:0] 置为 011b），如果不到四分之三的 FIFO 包含数据（SAI_xSR 中的 FLVL[2:0] 位小于 100b），将产生中断（SAI_xSR 寄存器中的 FREQ 位由硬件置 1）。当至少四分之三的 FIFO 包含数据时（SAI_xSR 中的 FLVL[2:0] 位大于或等于 100b），此中断（SAI_xSR 寄存器中的 FREQ 位）由硬件清零。
- 当 SAI_xCR2 寄存器中的 FIFO 阈值位配置为 FIFO 为满时（FTH[2:0] 置为 100b），如果 FIFO 不满（SAI_xSR 中的 FLVL[2:0] 位小于 101b），将产生中断（SAI_xSR 寄存器中的 FREQ 位由硬件置 1）。当 FIFO 已满时（SAI_xSR 中的 FLVL[2:0] 位等于值 101b），此中断（SAI_xSR 寄存器中的 FREQ 位）由硬件清零。

在接收模式下产生中断

根据接收模式下的 FIFO 配置产生中断：

- 当 SAI_xCR2 寄存器中的 FIFO 阈值位配置为 FIFO 为空时（FTH[2:0] 置为 000b），如果 SAI_xDR 寄存器中至少有一个数据（SAI_xSR 中的 FLVL[2:0] 位大于或等于 001b），将产生中断（SAI_xSR 寄存器中的 FREQ 位由硬件置 1）。当 FIFO 变为空时（SAI_xSR 中的 FLVL[2:0] 位等于 000b），即 FIFO 中未存储数据时，此中断（SAI_xSR 寄存器中的 FREQ 位）由硬件清零。
- 当 SAI_xCR2 寄存器中的 FIFO 阈值位配置为 FIFO 四分之一满时（FTH[2:0] 置为 001b），如果至少有四分之一的 FIFO 数据单元可用（SAI_xSR 中的 FLVL[2:0] 位大于或等于 010b），将产生中断（SAI_xSR 寄存器中的 FREQ 位由硬件置 1）。当不到四分之一的 FIFO 数据单元可用时（SAI_xSR 中的 FLVL[2:0] 位小于 010b），此中断（SAI_xSR 寄存器中的 FREQ 位）由硬件清零。
- 当 SAI_xCR2 寄存器中的 FIFO 阈值位配置为 FIFO 半满时（FTH[2:0] 置为 010b），如果至少有一半的 FIFO 数据单元可用（SAI_xSR 中的 FLVL[2:0] 位大于或等于 011b），将产生中断（SAI_xSR 寄存器中的 FREQ 位由硬件置 1）。当不到一半的 FIFO 数据单元可用时（SAI_xSR 中的 FLVL[2:0] 位小于 011b），此中断（SAI_xSR 寄存器中的 FREQ 位）由硬件清零。
- 当 SAI_xCR2 寄存器中的 FIFO 阈值位配置为 FIFO 四分之三满时（FTH[2:0] 置为 011b），如果至少有四分之三的 FIFO 数据单元可用（SAI_xSR 中的 FLVL[2:0] 位大于或等于 100b），将产生中断（SAI_xSR 寄存器中的 FREQ 位由硬件置 1）。当不到四分之三的 FIFO 数据单元可用时（SAI_xSR 中的 FLVL[2:0] 位小于 100b），此中断（SAI_xSR 寄存器中的 FREQ 位）由硬件清零。
- 当 SAI_xCR2 寄存器中的 FIFO 阈值位配置为 FIFO 满时（FTH[2:0] 置为 100b），如果 FIFO 已满（SAI_xSR 中的 FLVL[2:0] 位等于 101b），将产生中断（SAI_xSR 寄存器中的 FREQ 位由硬件置 1）。当 FIFO 不满时（SAI_xSR 中的 FLVL[2:0] 位小于 101b），此中断（SAI_xSR 寄存器中的 FREQ 位）由硬件清零。

与中断的产生类似，如果 SAI_xCR1 寄存器中的 DMAEN 位置 1，则 SAI 可使用 DMA。FREQ 位的有效机制与上述 FREQIE 的中断产生机制相同。

每个 FIFO 均是一个 8 字 FIFO。无论访问大小为何，每次对 FIFO 进行读写时，均针对 1 个字的 FIFO 单元进行操作。每个 FIFO 字包含一个音频 Slot。每次访问 SAI_xDR 寄存器后，FIFO 指针递增一个字。

数据应以右对齐方式写入 SAI_xDR。

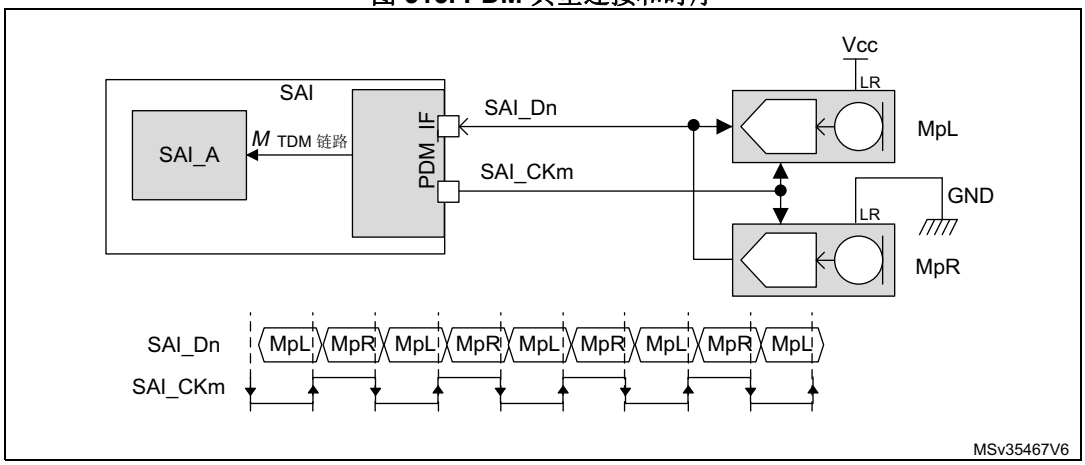
接收到的数据将以右对齐方式存储到 SAI_xDR。

将 SAI_xCR2 寄存器中的 FFLUSH 位置 1 禁止 SAI 后，可重新初始化 FIFO 指针。如果 FFLUSH 在 SAI 使能情况下置 1，则 FIFO 中的数据将自动清除。

47.3.10 PDM 接口

PDM（脉冲密度调制）接口旨在支持数字麦克风，最多可并联 4 对数字麦克风。*图 515* 所示为一对数字麦克风通过 PDM 接口实现的典型连接。两个麦克风共用同一比特流时钟和数据线。借助配置引脚 (LR)，其中一个麦克风可在 SAI_CK[m] 上升沿提供有效数据，而另一个麦克风则在 SAI_CK[m] 下降沿提供有效数据（m 表示时钟线的数量）。

图 515. PDM 典型连接和时序



1. **n** 表示数据线的数量，**p** 表示麦克风对的数量。

PDM 功能旨在与配置为 TDM 主模式的 SAI_A 子模块配合使用。它不能与 SAI_B 子模块搭配使用。PDM 接口使用由 SAI_A 的 TDM 接口提供的时序信号，并对其进行调整来生成时钟 (SAI_CK[m])。

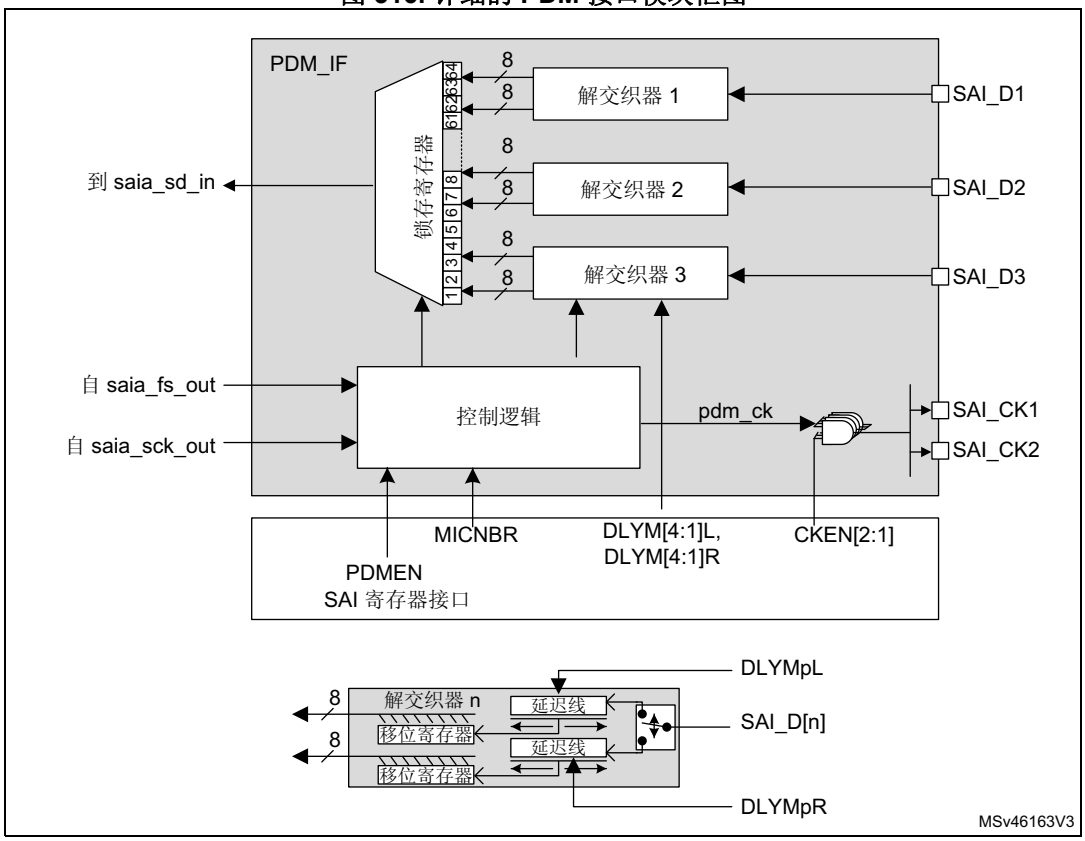
PDM 中的数据处理序列如下：

1. PDM 接口通过从 SAI_A 的 TDM 接口接收的位时钟构建比特流时钟。
2. 接收自麦克风的比特流数据 (SAI_D[n]) 进行解交织，然后经过 7 位延迟线，以根据比特流时钟的精度对每个麦克风的延迟进行微调。
3. 移位寄存器将每个串行比特流转换为字节。
4. 最后通过 TDM 接口的串行数据线将得到的字节移出到 SAI_A。

下面的*图 516* 给出了 PDM 接口框图，其中包括去交错的详细视图。

注： *PDM 接口未嵌入抽取滤波器，因此无法根据比特流构建 PCM 音频采样，需要借助应用软件来执行这一操作。*

图 516. 详细的 PDM 接口模块框图



1. **n** 表示数据线的数量，**p** 表示麦克风对的数量。

可通过 SAI_PDMCR 寄存器中的 PDMEN 位使能 PDM 接口。不过，必须在使能 SAI_A 模块之前使能 PDM 接口。

为减少存储器占用量，用户可根据应用需求选择麦克风的数量。这可以通过 MICNBR[1:0] 位来实现。用户可以选择 2、4、6 或 8 个麦克风。例如，如果应用需要使用 3 个麦克风，则用户必须选择 4 个麦克风。

使能 PDM 接口

要使能 PDM 接口，需遵循以下序列：

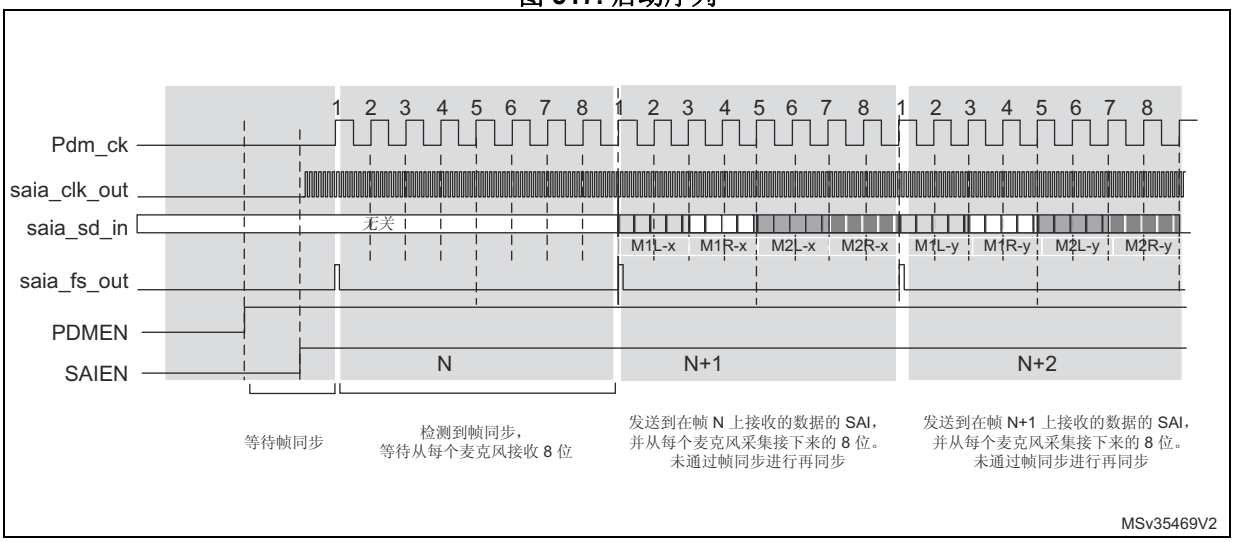
1. 将 SAI_A 配置为 TDM 主模式（请参见*表 281*）。
2. 按以下步骤配置 PDM 接口：
   a) 通过 MICNBR 定义数字麦克风的数量。
   b) 通过将 CKEN 的相应位置 1 来使能应用所需的比特流时钟。
3. 通过 PDMEN 位使能 PDM 接口。
4. 使能 SAI_A。

注： *使能 PDM 接口和 SAI_A 后，在 SAI_ADR 上接收到的前 2 个 TDMA 帧将是无效帧，应将其丢弃。*

启动序列

图 517 显示了启动序列：PDM 接口使能后，会等待帧同步事件，然后再开始采集麦克风采样。经过 8 个 SAI_CK 时钟周期之后，来自每个麦克风的数据字节将变为可用，并会通过 TDM 接口传输到 SAI。

图 517. 启动序列



SAI_ADR 数据格式

在 SAI_ADR 寄存器中，将根据以下参数对来自麦克风的数据进行排列：

- 麦克风的数量
- 所选 Slot 宽度
- 左对齐第一位

Slot 宽度定义了 SAI_ADR 寄存器中每个字的有效位数。

如果选择 32 位的 Slot 宽度，则 SAI_ADR 中的每个数据将包含 32 个有用位。这样可减少存储到存储器中的字数。不过，软件必须相应地执行某些操作来对每个麦克风的数据进行去交错。

另一方面，如果将 Slot 宽度设为 8 位，则 SAI_ADR 中的每个数据将包含 8 个有用位。这样会增加存储到存储器中的字数。不过，这样有助于避免额外的处理操作，因为每个字包含一个麦克风的信息。

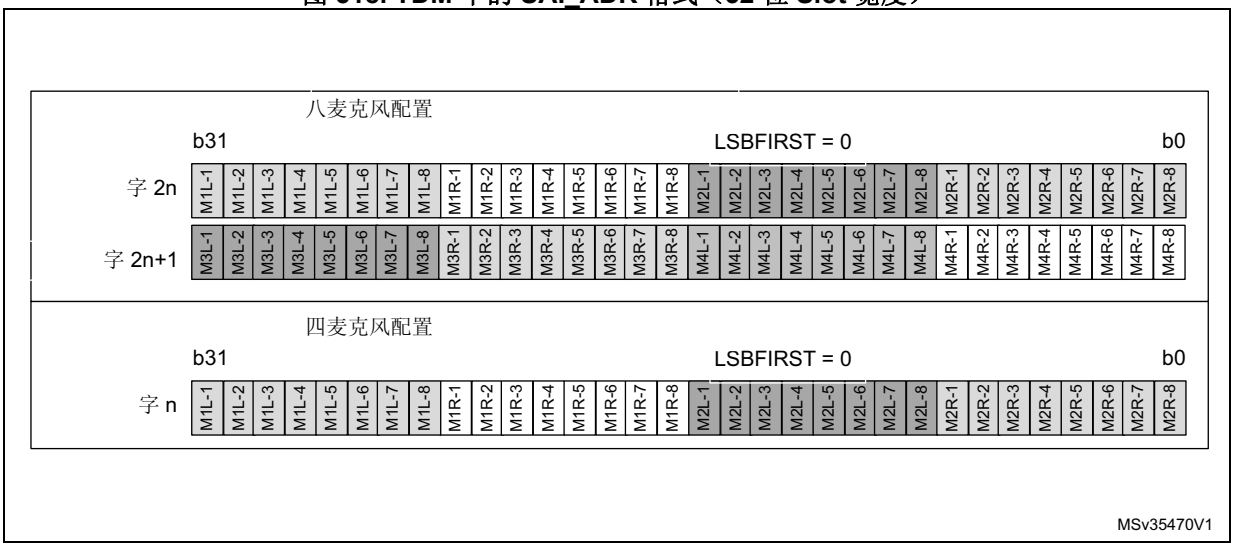
SAI_ADR 数据格式示例

- **32 位 Slot 宽度**（DS = 0b111 且 SLOTSZ = 0）。请参见*图 518*。

  对于 8 麦克风配置，连续对 SAI_ADR 寄存器进行两次字读操作可获取所有麦克风的数据字节。

  对于 4 麦克风配置，每次对 SAI_ADR 寄存器进行字读操作均可获取所有麦克风的数据字节。

图 518. TDM 中的 SAI_ADR 格式（32 位 Slot 宽度）

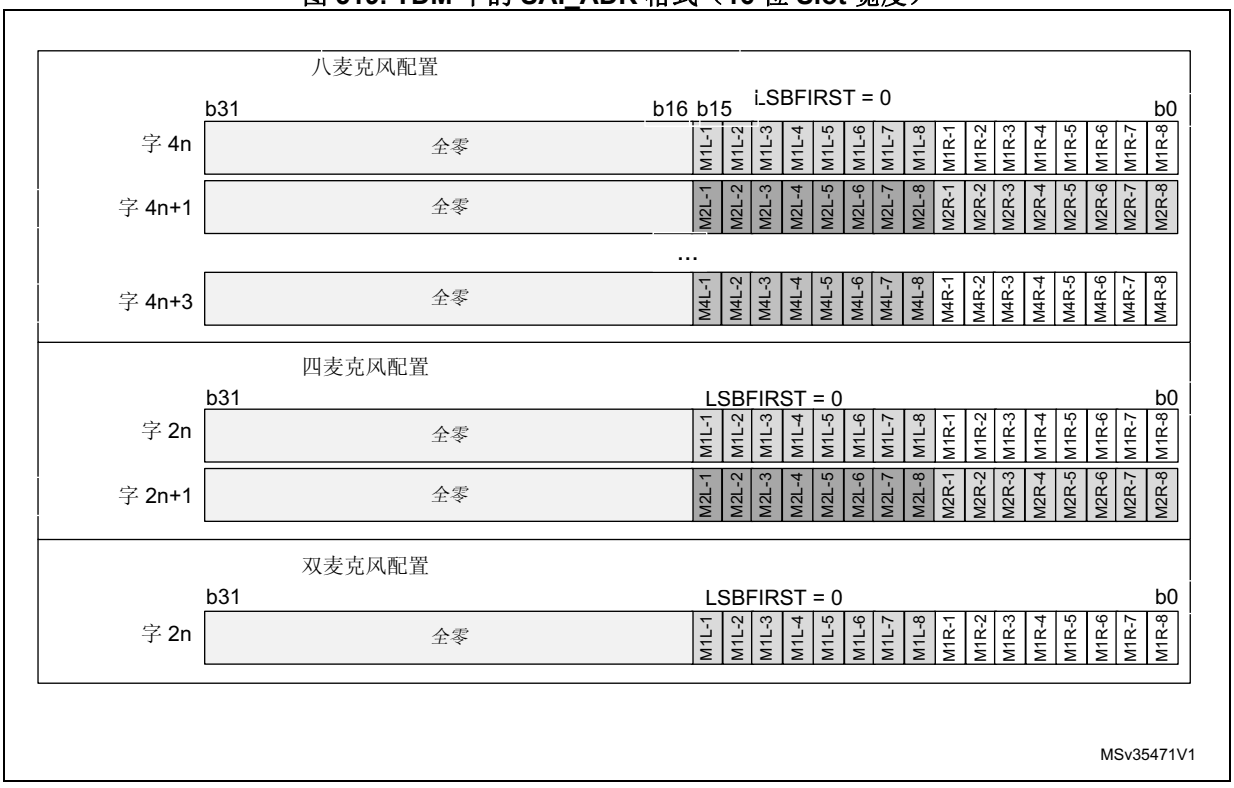


- **16 位 Slot 宽度**（DS = 0b100 且 SLOTSZ = 0）。请参见*图 519*。

  对于 8 麦克风配置，连续对 SAI_ADR 寄存器进行四次字读操作可获取所有麦克风的数据字节。请注意，SAI_ADR 的 16 位数据为右对齐。

  对于 4 麦克风配置或 2 麦克风配置，SAI 行为类似于 8 麦克风配置。获取 4 个麦克风的数据字节需要多达 2 个 16 位字；而获取 2 个麦克风的数据字节需要 1 个 16 位字。

图 519. TDM 中的 SAI_ADR 格式（16 位 Slot 宽度）

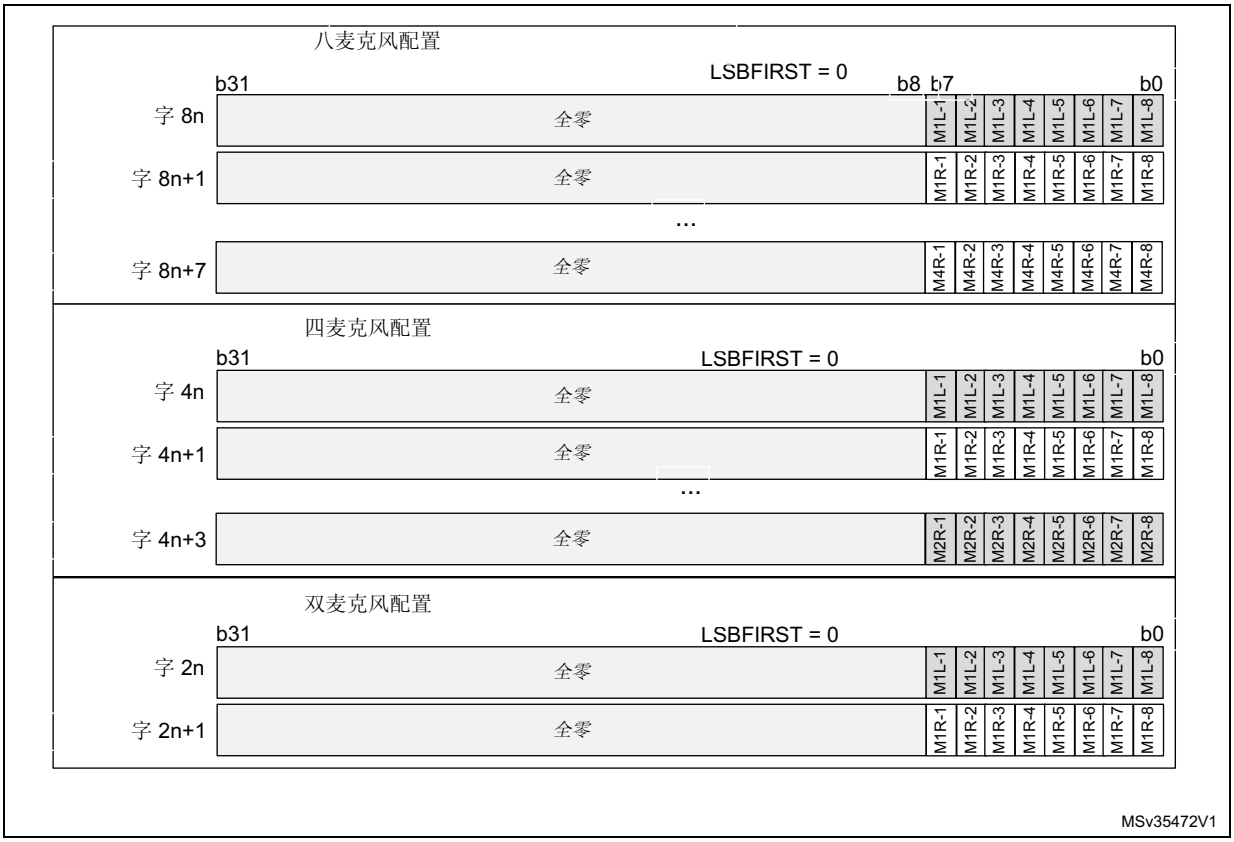


- **使用 8 位 Slot 宽度**（DS = 0b010 且 SLOTSZ = 0）。请参见*图 520*。

  对于 8 麦克风配置，连续对 SAI_ADR 寄存器进行 8 次字读操作可获取所有麦克风的数据字节。请注意，SAI_ADR 的 8 位数据为右对齐。

  对于 4 麦克风配置或 2 麦克风配置，SAI 行为类似于 8 麦克风配置。获取 4 个麦克风的数据字节需要多达 4 个 8 位字；而获取 2 个麦克风的数据字节需要 2 个 8 位字。

图 520. TDM 中的 SAI_ADR 格式（8 位 Slot 宽度）



PDM 接口的 TDM 配置

SAI_A TDM 接口从内部连接到 PDM 接口，以获取麦克风采样。用户应用程序必须按照*表 281* 所示配置 PDM 接口，以确保与 PDM 接口的连接良好。

表 281. TDM 设置

位域	值	注释
MODE	0b01	必须为主接收模式
PRTCFG	0b00	TDM 的自由协议
DS	X	根据所需数据格式，通过帧长度和 Slot 数（FRL 和 NBSLOT）进行调整。请参见*表 282*。
LSBFIRST	X	可根据所需数据格式使用该参数
CKSTR	0	信号在 SCK_A 位时钟的上升沿发生跳变，在位时钟的下降沿达到稳定状态。
MONO	0	立体声模式

表 281. TDM 设置（续）

位域	值	注释
FRL	X	根据麦克风数量 (MICNBR) 进行调整。请参见*表 282*。
FSALL	0	脉宽为一个位时钟周期
FSDEF	0	FS 信号为帧起始信号
FSPOL	1	FS 为高电平有效
FSOFF	0	在 Slot 0 的第一位上使能 FS
FBOFF	0	Slot 上无偏移
SLOTSZ	0	Slot 大小 = 数据大小
NBSLOT	X	根据所需数据格式，通过 Slot 大小和帧长度（FRL 和 DS）进行调整。 请参见*表 282*。
SLOTEN	X	根据 NBSLOT 进行调整
NOMCK	1	无需生成主时钟 MCLK
MCKDIV	X	取决于提供给 sai_a_ker_ck 输入的频率。 应对该参数进行调整，以生成合适的比特流时钟频率。请参见*表 282*。

调整比特流时钟频率

为正确编程 SAI TDM 接口，用户应用程序必须考虑*表 281* 中给出的设置，并遵循以下规则：

1. 使用以下公式，根据 PDM 比特流时钟所需的频率调整位时钟频率 (F_{SCK_A})：

$$F_{SCK_A} = F_{PDM_CK} \times (MICNBR + 1) \times 2$$

MICNBR 可以是 0、1、2 或 3（0 = 2 个麦克风，请参见*第 47.5.10 节*）

2. 使用以下公式设置帧长度 (FRL)。

$$FRL = (16 \times (MICNBR + 1)) - 1$$

3. 将 Slot 大小 (DS) 配置为 (FRL+1) 的倍数。

表 282. 允许的 TDM 帧配置(1)

麦克风采样率	麦克风数量	所需 SAI_CK[m] 频率 (2)	位时钟 (SCK_A) 频率	帧同步 (FS_A) 频率	FRL	DS	NBSLOT	注释
48 kHz	最多 8 个	3.072 MHz	24.576 MHz	384 kHz	63	0b100	3	每个帧 4 个 16 位 Slot
		3.072 MHz	24.576 MHz	384 kHz	63	0b010	7	每个帧 8 个 8 位 Slot
	最多 6 个	3.072 MHz	18.432 MHz	384 kHz	47	0b110	1	每个帧 2 个 24 位 Slot
		3.072 MHz	18.432 MHz	384 kHz	47	0b100	2	每个帧 3 个 16 位 Slot
		3.072 MHz	18.432 MHz	384 kHz	47	0b010	5	每个帧 6 个 8 位 Slot
	最多 4 个	3.072 MHz	12.288 MHz	384 kHz	31	0b111	0	每个帧 1 个 32 位 Slot
		3.072 MHz	12.288 MHz	384 kHz	31	0b100	1	每个帧 2 个 16 位 Slot
		3.072 MHz	12.288 MHz	384 kHz	31	0b010	3	每个帧 4 个 8 位 Slot
	最多 2 个	3.072 MHz	6.144 MHz	384 kHz	15	0b100	0	每个帧 1 个 16 位 Slot
		3.072 MHz	6.144 MHz	384 kHz	15	0b010	1	每个帧 2 个 8 位 Slot
16 kHz	最多 8 个	1.024 MHz	8.192 MHz	128 kHz	63	0b100	3	每个帧 4 个 16 位 Slot
		1.024 MHz	8.192 MHz	128 kHz	63	0b010	7	每个帧 8 个 8 位 Slot
	最多 6 个	1.024 MHz	6.144 MHz	128 kHz	47	0b110	1	每个帧 2 个 24 位 Slot
		1.024 MHz	6.144 MHz	128 kHz	47	0b010	5	每个帧 6 个 8 位 Slot
	最多 4 个	1.024 MHz	4.096 MHz	128 kHz	31	0b111	0	每个帧 1 个 32 位 Slot
		1.024 MHz	4.096 MHz	128 kHz	31	0b100	1	每个帧 2 个 16 位 Slot
		1.024 MHz	4.096 MHz	128 kHz	31	0b010	3	每个帧 4 个 8 位 Slot
	最多 2 个	1.024 MHz	2.048 MHz	128 kHz	15	0b100	0	每个帧 1 个 16 位 Slot
		1.024 MHz	2.048 MHz	128 kHz	15	0b010	1	每个帧 2 个 8 位 Slot

1. 有关 TDM 配置的更多信息，请参见 *表 281：TDM 设置*。提供给 SAI 的 sai_a_ker_ck 时钟频率应是 SCK_A 频率的倍数，并且应相应地编程 MCKDIV。
2. 上表给出了抽取率为 64 时所允许的设置。

调整延迟线

使能 PDM 接口时，应用程序可通过 SAI_PDMDLY 寄存器实时调整每个麦克风输入的延迟单元数。

新的延迟值将在两个 TDM 帧后生效。

47.3.11 AC’97 链路控制器

SAI 可用作 AC’97 链路控制器。在此协议中：

- Slot 数和 Slot 大小固定。
- 帧同步信号定义完善并且波形固定。

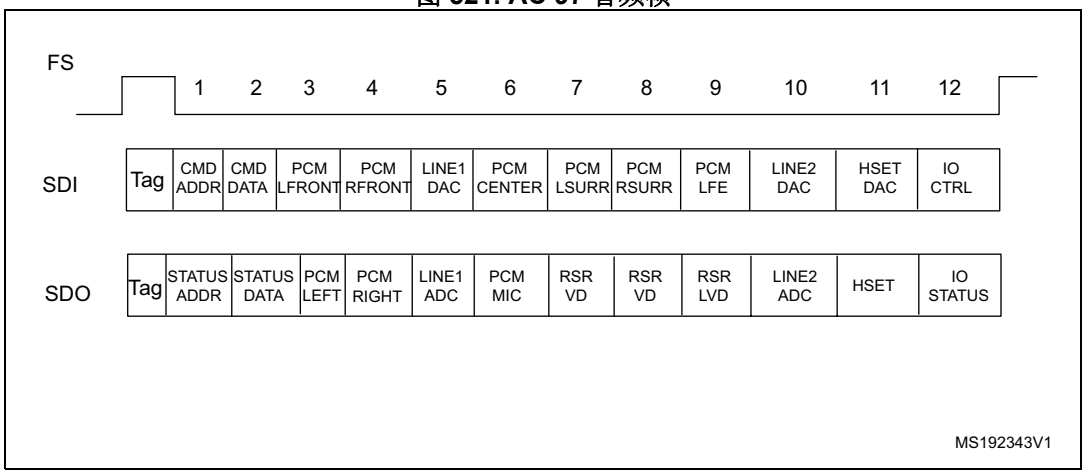
要选择此协议，可将 SAI_xCR1 寄存器中的 PRTCFG[1:0] 位置为 10。选择 AC’97 模式时，只能使用 16 位或 20 位的数据大小，否则将无法保证 SAI 运行正常。

- 因此，NBSLOT[3:0] 和 SLOTSZ[1:0] 位将被忽略。
- Slot 数固定为 13。第一个 Slot 为 16 位宽，所有其他 Slot 均为 20 位宽（数据 Slot）。
- SAI_xSLOTR 寄存器中的 FBOFF[4:0] 位被忽略。
- SAI_xFRCR 寄存器被忽略。
- 未使用 MCLK。

无论采用主配置还是从配置，由于 AC’97 链路控制器驱动 FS 信号，异步模块发出的 FS 信号将自动配置为输出。

图 521 给出了 AC’97 音频帧的结构。

图 521. AC’97 音频帧



注： *在 AC’97 协议中，TAG 的位 2 将保留（始终为 0），因此无论 SAI FIFO 中写入何值，TAG 的位 2 均强制为 0。*

有关 TAG 表示的详细信息，请参见 AC’97 协议标准。

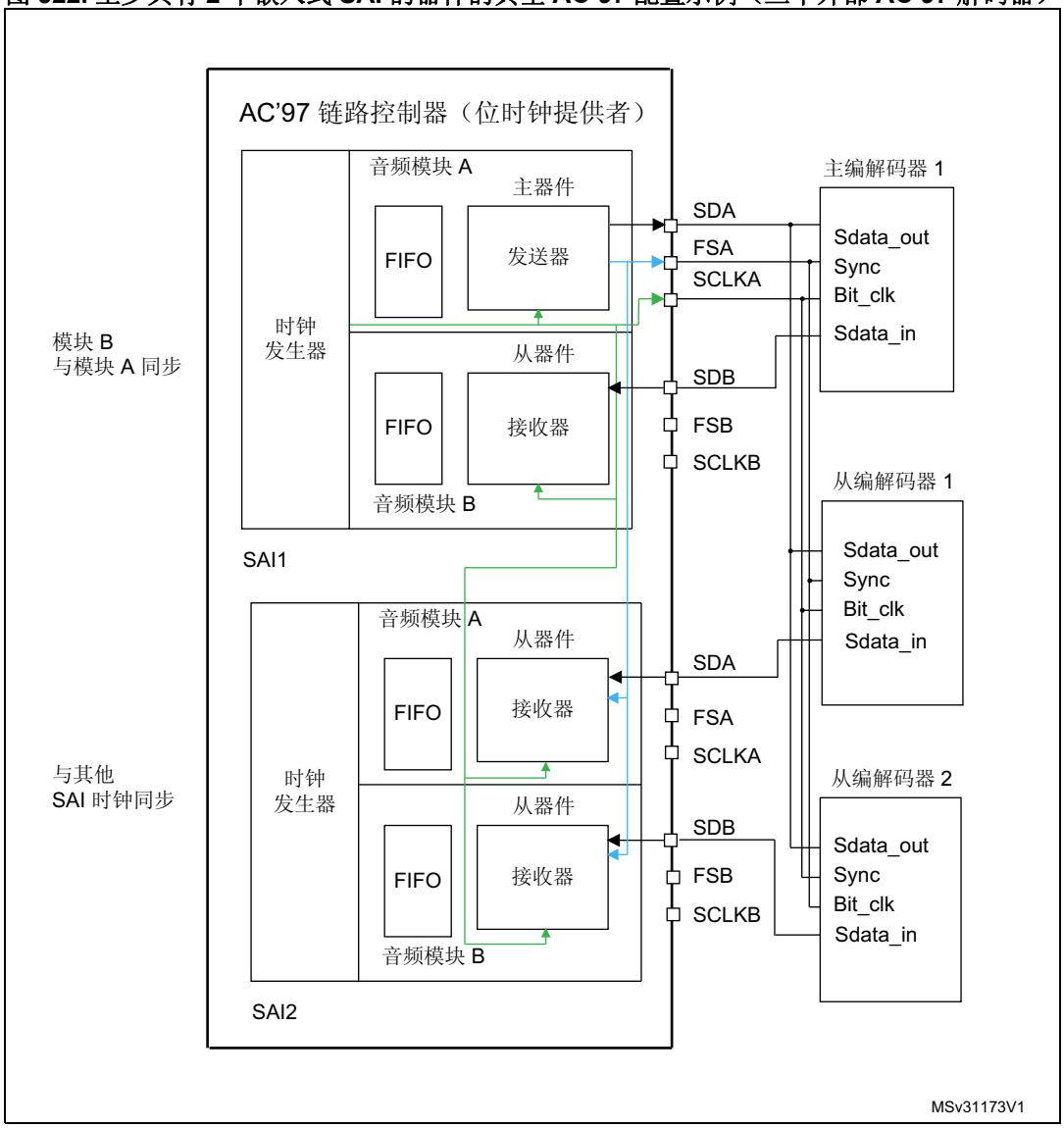
可将一个 SAI 用于 AC’97 点对点通信。

使用两个 SAI（针对具有两个嵌入式 SAI 的器件）可以控制三个外部 AC’97 解码器，如*图 522* 所示。

在 SAI1 中，必须将音频模块 A 声明为异步主发送器，而将音频模块 B 定义为从接收器，并在内部与音频模块 A 同步。

在从接收器模式下，同时针对音频模块 A 和音频模块 B 将 SAI2 配置为与外部 SAI1 同步。

图 522. 至少具有 2 个嵌入式 SAI 的器件的典型 AC’97 配置示例（三个外部 AC’97 解码器）



在接收器模式下，用作 AC’97 链路控制器的 SAI 无需 FIFO 请求，因此当 Slot 0 中的编码解码器就绪位解码为低电平时，FIFO 中不存储任何数据。如果 SAI_xIM 寄存器中的 CNRDYIE 位使能，则 SAI_xSR 寄存器中的标志 CNRDY 将置 1 并会产生一个中断。此标志专用于 AC’97 协议。

AC’97 模式下的时钟发生器编程

在 AC’97 模式下，帧长度固定为 256 位，其频率应设置为 48 kHz。*第 47.3.8 节：SAI 时钟发生器*中给出的规则应在 FRL = 255 时使用，以生成适当的帧速率 (F_{FS_x})。

47.3.12 SPDIF 输出

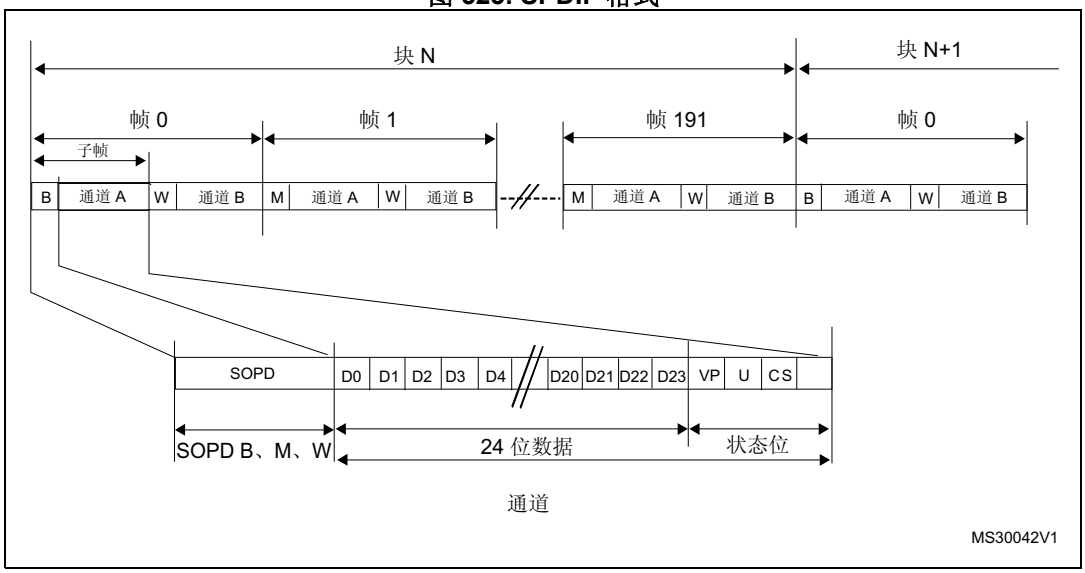
SPDIF 接口仅在发送模式下可用。它支持 IEC60958 音频标准。

要选择 SPDIF 模式，需将 SAI_xCR1 寄存器中的 PRTCFG[1:0] 位置为 01。

对于 SPDIF 协议：

- 仅使能 SD 数据线。
- 释放 FS、SCK、MCLK I/O 引脚。
- 为使能 SAI 的时钟发生器并管理 SD 线上的数据速率，将 MODE[1] 位强制设置为 0 以选择主模式。
- 数据大小强制设置为 24 位。忽略在 SAI_xCR1 寄存器的 DS[2:0] 位中设置的值。
- 必须配置时钟发生器以定义符号率，已知位时钟应为符号率的两倍。数据采用曼彻斯特协议进行编码。
- 忽略 SAI_xFRCR 和 SAI_xSLOTR 寄存器。在内部配置 SAI 使其符合 SPDIF 协议要求，如*图 523* 所示。

图 523. SPDIF 格式



SPDIF 块包含 192 个帧。每个帧由两个 32 位的子帧构成，通常一个子帧用于左通道，一个用于右通道。每个子帧由一个 SOPD 模式（4 位）构成，用于指定该子帧是否为块的开始（并用于识别通道 A），或者是否标识块中某处的通道 A，或者是否指代通道 B（请参见*表 283*）。接下来的 28 位是通道信息，由 24 个数据位和 4 个状态位组成。

表 283. SOPD 模式

SOPD	报头编码		说明
	最后一位是 0	最后一位是 1	
B	11101000	00010111	块开始处的通道 A 数据
W	11100100	00011011	块中某处的通道 B 数据
M	11100010	00011101	通道 A 数据

必须按如下方式填充 SAI_xDR 中存储的数据：

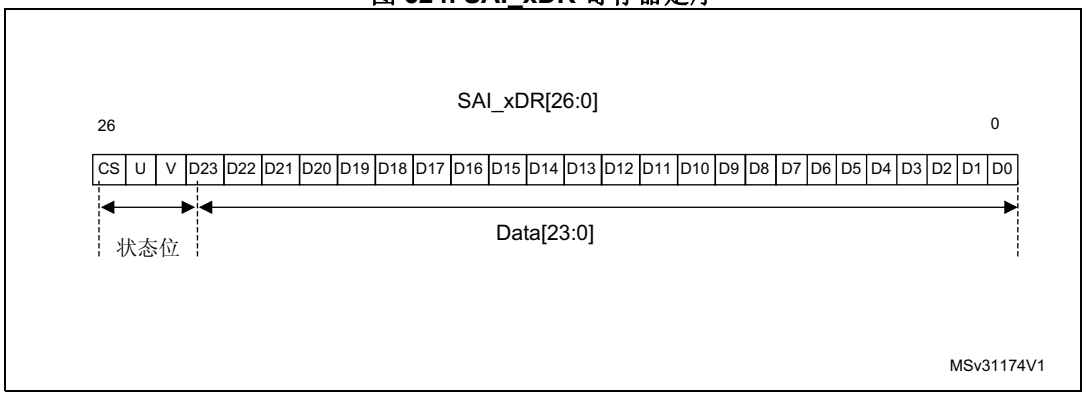
- SAI_xDR[26:24] 包含通道状态位、用户位和有效性位。
- SAI_xDR[23:0] 包含所考虑通道的 24 位数据。

如果数据大小为 20 位，应将数据映射到 SAI_xDR[23:4] 上。

如果数据大小为 16 位，应将数据映射到 SAI_xDR[23:8] 上。

SAI_xDR[23] 始终代表 MSB。

图 524. SAI_xDR 寄存器定序



注： *执行传输时，LSB 始终优先。*

SAI 首先在块中发送每个子帧的适当报头。随后在 SD 线上发送 SAI_xDR（以曼彻斯特协议进行编码）。SAI 通过传输按*表 284* 所述计算的奇偶校验位来结束子帧。

表 284. 奇偶校验位计算

SAI_xDR[26:0]	传输奇偶校验位 P 值
奇数个 0	0
奇数个 1	1

在 SPDIF 模式下，下溢是在 SAI_xSR 寄存器唯一个用到的错误标志。因为 SAI 只能在发送模式下工作。因此，要从通过下溢中断或下溢状态位检测到的下溢错误中恢复，应按顺序执行以下步骤：

1. 如果已使用 DMA，则禁止 DMA 流（通过 DMA 外设）。
2. 禁止 SAI 并通过轮询 SAI_xCR1 寄存器中的 SAIEN 位确认已从物理上禁止外设。
3. 将 SAI_xCLRFR 寄存器中的 COVRUNDR 标志清零。
4. 通过将 SAI_xCR2 寄存器中的 FFLUSH 位置 1 来刷新 FIFO。
   软件需要指向与新块开始（报头 B 的数据）相对应的后续数据的地址。如果已使用 DMA，应相应地更新 DMA 源起始地址指针。
5. 如果用于管理数据的 DMA 根据新的源起始地址来传输数据，则再次使能 DMA 流（DMA 外设）。
6. 通过将 SAI_xCR1 寄存器中的 SAIEN 位置 1 再次使能 SAI。

SPDIF 发生器模式下的时钟发生器编程

对于 SPDIF 发生器，SAI 应提供等于符号率两倍的位时钟。下表给出了符号率相对于音频采样率的通常示例。

表 285. 音频采样频率与符号率

音频采样频率 (F_S)	符号率
44.1 kHz	2.8224 MHz
48 kHz	3.072 MHz
96 kHz	6.144 MHz
192 kHz	12.288 MHz

通常，音频采样频率 (F_S) 与位时钟速率 (F_{SCK_X}) 之间的关系由以下公式给出：

$$F_S = \frac{F_{SCK_x}}{128}$$

按如下公式计算位时钟速率：

$$F_{SCK_x} = \frac{F_{SAI_CK_x}}{MCKDIV}$$

47.3.13 特性

根据所选的音频协议，SAI 接口内嵌了一些特定的实用功能。这些功能可通过 SAI_xCR2 寄存器的特定位来访问。

静音模式

当音频子模块用作发送器或接收器时，可使用静音模式。

发送模式下的音频子模块

在发送模式下，可随时选择静音模式。静音模式对于全部音频帧均有效。在帧传输过程中将 SAI_xCR2 寄存器的 MUTE 置 1，将使能静音模式。

该静音模式位仅在帧结束时选通。如果帧结束时置 1，静音模式将在新的音频帧开始时激活，并持续整个帧长度，直至下次帧结束；然后将选通该位，以确定下一帧是否仍为静音帧。

如果通过 SAI_xSLOTR 寄存器的 NBSLOT[3:0] 位设置的 Slot 数小于或等于 2，可指定静音模式下发送的值是否为 0 或该值是否为每个 Slot 的最后一个值。通过 SAI_xCR2 寄存器中的 MUTEVAL 位进行选择。

如果在 SAI_xSLOTR 寄存器的 NBSLOT[3:0] 位中设置的 Slot 数大于 2，由于在各 Slot 的每位上都发送值 0，因此 SAI_xCR2 中的 MUTEVAL 位没有意义。

在静音模式下，FIFO 指针仍递增，这意味着将丢弃 FIFO 中请求在静音模式下传输的数据。

接收模式下的音频子模块

在接收模式下，对于给定数量的连续音频帧（SAI_xCR2 寄存器中的 MUTECNT[5:0] 位），如果在音频帧所有声明的有效 Slot 接收到 0，则可检测到从外部发送器发来的静音模式。

检测到相应数量的静音帧时，SAI_xSR 寄存器中的 MUTEDET 标志置 1 并会在 SAI_xCR2 中的 MUTEDETIE 位置 1 情况下产生中断。

在音频子模块禁止时或在有效 Slot 内至少接收到音频帧中的一个数据时，静音帧计数器清零。计数器达到 MUTECNT[5:0] 位中指定的值时，仅产生一次中断。随后中断事件在计数器清零时重新初始化。

注： *静音模式不可用于 SPDIF 音频模块。*

单声道/立体声模式

在发送器模式下，如果 Slot 数等于 2（SAI_xSLOTR 中的 NBSLOT[3:0] = 0001），则可寻址单声道模式，而无需在存储器中进行任何数据预处理。在这种情况下，由于发送时 Slot 0 的数据被复制到数据 Slot 1 中，FIFO 的访问时间将减少一半。

要使能单声道模式，

1. 将 SAI_xCR1 寄存器中的 MONO 位置 1。
2. 将 SAI_xSLOTR 中的 NBSLOT 置 1，并将 SLOTEN 设置为 3。

在接收模式下，MONO 位可置 1 并且仅在 Slot 数等于 2（与发送模式下相同）时才有意义。当该位置 1 时，只有 Slot 0 的数据将存储到 FIFO 中。Slot 1 的数据由于被认为与前一个 Slot 的数据相同而被丢弃。如果接收模式下的数据流量是左声道数据和右声道数据明显不同的真立体声音频流，则 MONO 位没有意义。由软件完成从输出立体声文件到等效单声道文件的转换。

压扩模式

移动通信应用可能需要通过数据压扩算法处理待发送或待接收的数据。

应用可根据 SAI_xCR2 寄存器中的 COMP[1:0] 位（仅当选择 TDM 模式时使用），选择在 SD 串行输出线（压缩）发送数据前是否处理数据，以及是否在 SD 串行输入线（扩展）接收数据后扩展数据，如*图 525* 所示。所支持的两个压扩模式是 μ-Law 和 A-Law，二者是 CCITT G.711 推荐标准的一部分。

美国和日本采用的压扩标准是 μ-Law，该标准允许 14 位动态范围（SAI_xCR2 寄存器中的 COMP[1:0] = 10）。

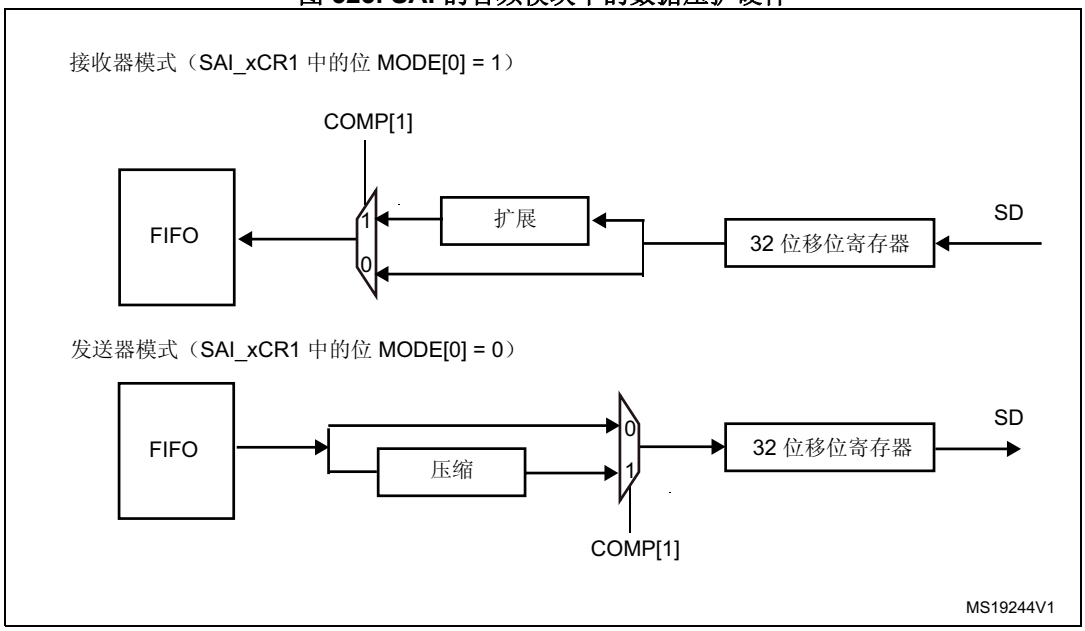
欧洲压扩标准是 A-Law，该标准支持 13 位动态范围（SAI_xCR2 寄存器中的 COMP[1:0] = 11）。

可根据 1 的补码或 2 的补码表示来计算 μ-Law 或 A-Law 压扩标准，具体取决于 SAI_xCR2 寄存器中的 CPL 位设置。

在 μ-Law 和 A-Law 标准中，数据将编码为采用 MSB 对齐的 8 位。压扩数据始终为 8 位宽。因此，当 SAI 音频模块使能（SAI_xCR1 寄存器中的位 SAIEN = 1）并且通过 COMP[1:0] 位选择这两个压扩模式之一时，SAI_xCR1 寄存器中的 DS[2:0] 位将强制为 010。

如果无需压扩处理，则 COMP[1:0] 位应保持清零。

图 525. SAI 的音频模块中的数据压扩硬件



1. 选择 AC'97 或 SPDIF 时不适用。

通过 SAI_xCR2 自动选择扩展模式和压缩模式：

- 如果 SAI 音频模块配置为发送器，且 SAI_xCR2 寄存器中的 COMP[1] 位置 1，将采用压缩模式。
- 如果 SAI 音频模块声明为接收器，将采用扩展算法。

无效 Slot 上的输出数据线管理

在发送模式下，当在数据线上发送无效 Slot 时，可选择 SD 线输出的行为（通过 TRIS 位实现）。

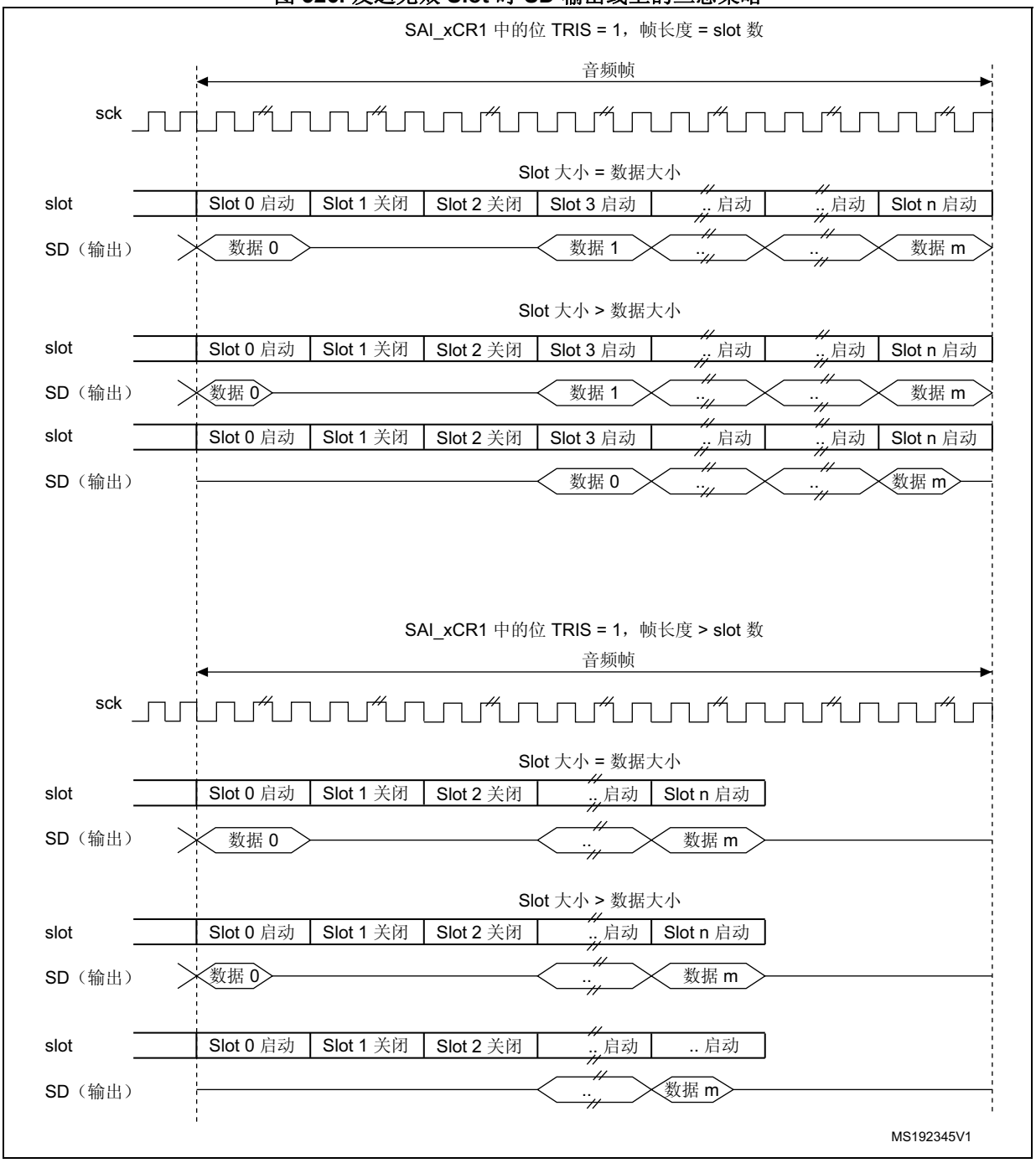
- 发送无效 Slot 时，SAI 将 SD 输出线强制为 0。
- 该输出线在最后一个数据位传输结束时释放为高阻态，为另一个连接此节点的发送器释放该数据线。

切记不要让两个发送器同时驱动同一个 SD 输出引脚，否则会导致短路。为确保存在发送间隙，如果数据低于 32 位，可通过在 SAI_xSLOTR 寄存器中设置 SLOTSZ[1:0] = 10 将数据扩展到 32 位。随后，如果下一 Slot 声明为无效，则 SD 输出引脚将在有效 Slot 的 LSD 结束时（将数据扩展到 32 位的填 0 阶段）置为三态。

此外，如果 Slot 数乘以 Slot 大小所得结果小于帧长度，则在通过填 0 来补充音频帧结束时，SD 输出线置为三态。

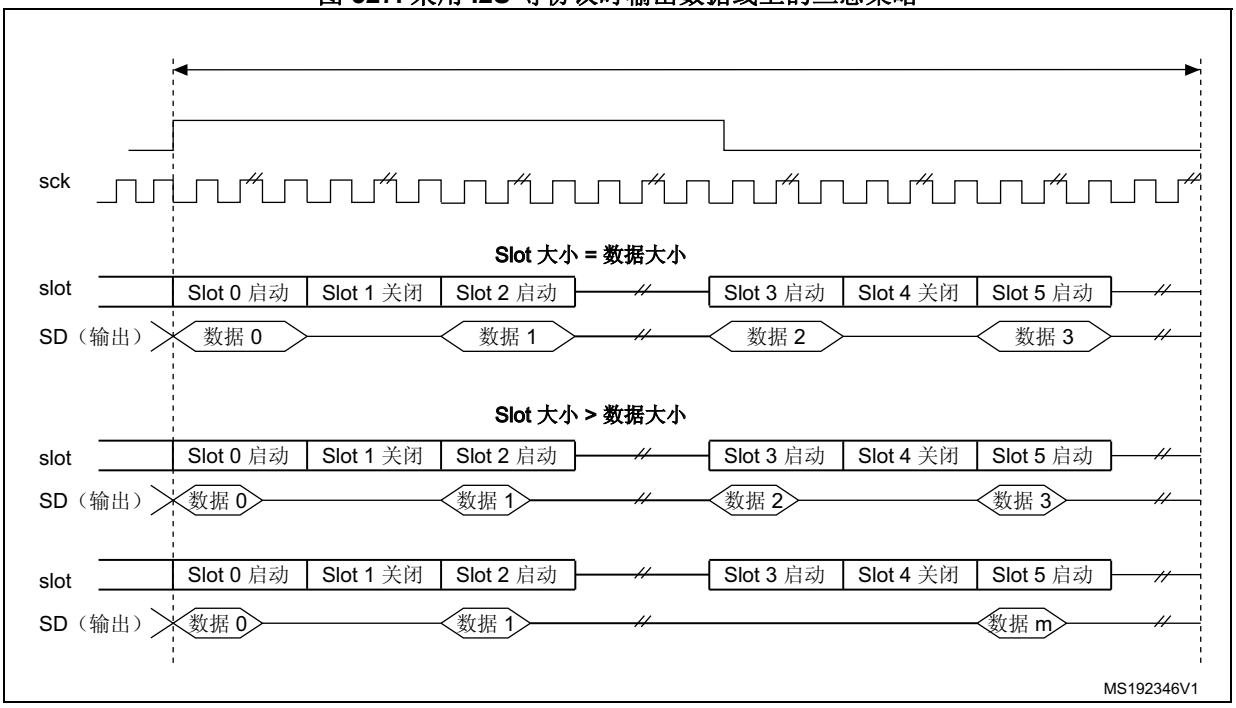
图 526 说明了这些行为。

图 526. 发送无效 Slot 时 SD 输出线上的三态策略



当所选音频协议使用 FS 信号作为 SOF 信号或通道识别信号（SAI_xFRCR 寄存器中的位 FSDEF = 1）时，将按照*图 527* 管理三态模式（其中，SAI_xCR1 寄存器中的位 TRIS = 1，FSDEF = 1，半帧长大于 Slot 数/2 且 NBSLOT = 6）。

图 527. 采用 I2S 等协议时输出数据线上的三态策略



如果 SAI_xCR2 寄存器中的 TRIS 位清零，*图 526* 和*图 527* 上的 SD 输出线上的所有高阻态将替换为使用值 0 驱动。

47.3.14 错误标志

SAI 使用以下错误标志：

- FIFO 上溢/下溢
- 帧同步提前检测
- 帧同步滞后检测
- 编解码器未就绪（仅限 AC'97）
- 主模式时钟配置错误

FIFO 上溢/下溢 (OVRUDR)

FIFO 上溢/下溢位是 SAI_xSR 寄存器中的 OVRUDR 位。

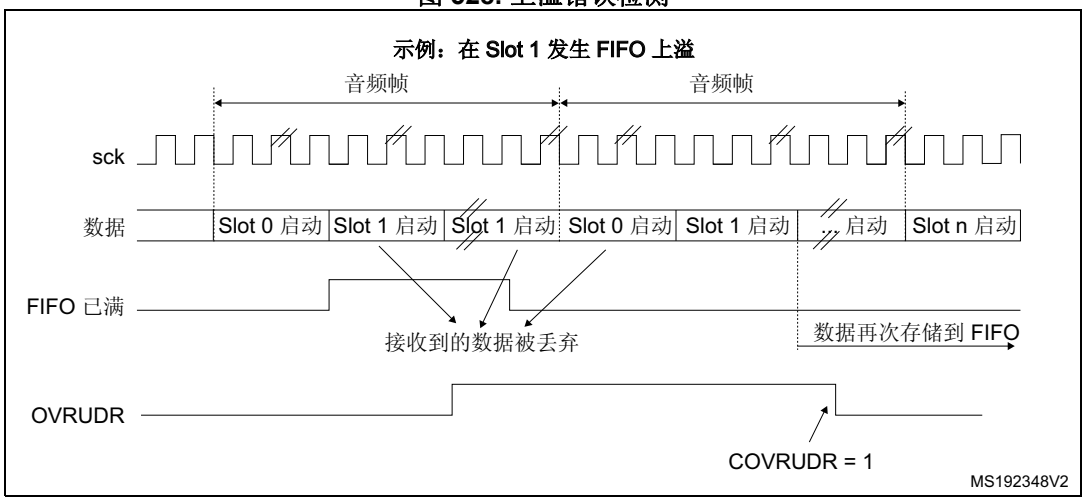
由于音频模块既可作为接收器，又可作为发送器，并且指定 SAI 中的每个音频模块都具有自己的 SAI_xSR 寄存器，因此上溢或下溢错误共用同一位。

上溢

若音频模块配置为接收器，则在 FIFO 已满且无法再存储接收数据的情况下又收到音频帧数据时，将出现上溢情况。这种情况下，接收数据将丢失，SAI_xSR 寄存器中的 OVRUDR 标志置 1；如果 SAI_xIM 寄存器中的 OVRUDRIE 位置 1，还将生成中断。内部将存储发生上溢时的 Slot 编号。FIFO 无法再存储更多数据，直至释放出空间存储新数据为止。在 FIFO 释放了至少一个数据的空间时，SAI 音频模块接收器将从检测到上溢后内部记录的 Slot 编号开始接收来自新音频帧的新数据，这样可避免目标存储器中出现数据 Slot 不对齐的情况（请参见*图 528*）。

SAI_xCLRFR 寄存器中的 COVRUDR 位置 1 时将清除 OVRUDR 标志。

图 528. 上溢错误检测



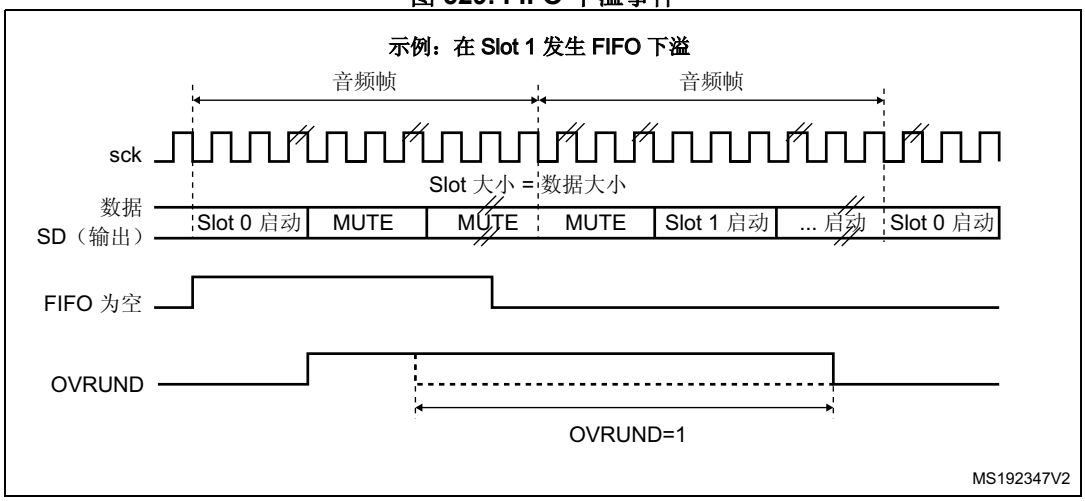
下溢

当 SAI 中的音频模块用作发送器时，如果需要发送数据时 FIFO 为空，则可能出现下溢。如果检测到下溢，则将存储发生事件的 Slot 编号并发送 MUTE 值 (00)，直到 FIFO 准备好发送与检测到下溢的 Slot 对应的数据（请参见*图 529*）。这样可避免存储器指针与音频帧中的 Slot 之间发生同步失效。

下溢事件会使 SAI_xSR 寄存器中的 OVRUDR 标志置 1，如果 SAI_xIM 寄存器中的 OVRUDRIE 位置 1，还将生成中断。要清除该标志，可将 SAI_xCLRFR 寄存器中的 COVRUDR 位置 1。

将音频子模块配置为主模式或从模式时，会发生下溢事件。

图 529. FIFO 下溢事件



帧同步提前检测 (AFSDET)

AFSDET 标志仅在从模式下使用。主模式下不会使能该标志。由于帧长度、帧极性和帧偏移已定义且已知，该标志用于指示是否比预期更早检测到帧同步 (FS) 信号。

出现帧检测提前时，SAI_xSR 寄存器中的 AFSDET 标志将置 1。

对 FS 提前不敏感的当前音频帧不受该检测影响。也就是说 FS 信号的“寄生”事件将被标记但不干扰当前音频帧。

如果 SAI_xIM 寄存器中的 AFSDETIE 位置 1，将生成中断。要清除 AFSDET 标志，必须将 SAI_xCLRFR 寄存器中的 CAFSDET 位置 1。

为了在出现帧检测提前错误后重新与主模块同步，需要执行以下四个步骤：

1. 通过复位 SAI_xCR1 寄存器中的 SAIEN 位禁止 SAI 模块。为确保禁止 SAI，读回 SAIEN 位并检查其是否已设置为 0。
2. 通过 SAI_xCR2 寄存器的 FFLUS 位刷新 FIFO。
3. 重新使能 SAI 外设（SAIEN 位置 1）。
4. SAI 模块将等待 FS 使能以重新开始与主模块同步。

注： *AC’97 中不使能 SAIEN 标志，原因是 SAI 音频模块作为链路控制器，即使在声明为从模块时也会生成 FS 信号。由于未使用 FS 信号，因此在 SPDIF 模式下无意义。*

帧同步滞后检测

只有当 SAI 音频模块用作从模块时，SAI_xSR 寄存器中的 LFSDET 标志才可置 1。SAI_xFRCR 寄存器中，帧长度、帧极性和帧偏移配置均已知。

如果外部主模块未在预定时间发送 FS 信号生成过晚，LFSDET 标志将置 1，如果 SAI_xIM 寄存器中的 LFSDETIE 位置 1，还将生成中断。

SAI_xCLRFR 寄存器中的 CLFSDET 位置 1 时将清除 LFSDET 标志。

在检测到相应错误时帧同步滞后检测标志置 1，SAI 需要重新与主模块同步（请参见*帧同步提前检测 (AFSDET)* 中所述的顺序）。

在噪声环境中，音频模块的状态机可能会错误检测到对 SCK 时钟的干扰，并将 SAI 数据移位到错误的帧位置。SAI 会检测到此事件，并将其报告为帧同步滞后检测错误。

如果外部主模块不是在连续模式下管理音频数据帧发送，则不会对帧造成破坏，大多数应用都不会出现这种情况。这种情况下 LFSDET 标志将置 1。

注： *AC’97 中不使能 LFSDET 标志，原因是 SAI 音频模块作为链路控制器，即使在声明为从模块时也会生成 FS 信号。由于协议未使用 FS 信号，因此在 SPDIF 模式下无意义。*

编解码器未就绪 (CNRDY AC’97)

仅当 SAI 音频模块配置为在 AC’97 模式下工作时（SAI_xCR1 寄存器中的 PRTCFG[1:0] = 10），SAI_xSR 寄存器中的 CNRDY 标志才有意义。如果 SAI_xIM 寄存器中的 CNRDYIE 位置 1，则在 CNRDY 标志置 1 时将生成中断。

在接收 AC’97 音频帧的 TAG 0 (slot0) 期间，当编解码器未准备好进行通信时，将使能 CNRDY。这种情况下，在 TAG 0 指示编解码器就绪之前，数据都不会自动存储到 FIFO，原因是编解码器未就绪。编解码器就绪后将捕获 SAI_xSLOTR 寄存器中定义的所有有效 Slot 。

要清除 CNRDY 标志，必须将 SAI_xCLRFR 寄存器中的 CCNRDY 位置 1。

主模式时钟配置错误 (NOMCK = 0)

当音频模块在主模式下工作（MODE[1] = 0）且 NOMCK 位等于 0 时，如果满足以下条件，则只要使能 SAI，WCKCFG 标志便会置 1：

- (FRL+1) 不是 2 的几次幂，并且
- (FRL+1) 不在 8 和 256 之间

MODE 位、NOMCK 位和 SAIEN 位属于 SAI_xCR1 寄存器，FRL 位属于 SAI_xFRCR 寄存器。

如果 WCKCFGIE 位置 1，则当 SAI_xSR 寄存器中的 WCKCFG 标志置 1 时将生成中断。要清除该标志，可将 SAI_xCLRFR 寄存器中的 CWCKCFG 位置 1。

当 WCKCFG 位置 1 时，音频模块会自动禁止，因此将对 SAIEN 位执行硬件清零。

47.3.15 禁止 SAI

可随时通过清零 SAI_xCR1 寄存器中的 SAIEN 位禁止 SAI 音频模块。所有已开始的帧将在 SAI 停止工作前自动完成。SAIEN 位将保持高电平，直到当前音频帧传输结束时 SAI 完全关闭。

如果 SAI 中有与另一个音频模块同步工作的音频模块，则必须先禁止以主模式工作的音频模块。

47.3.16 SAI DMA 接口

为减轻 CPU 负担和优化总线带宽，每个 SAI 音频模块都具有独立的 DMA 接口以便对 SAI_xDR 寄存器进行读/写操作（访问内部 FIFO）。每个音频子模块都有一个支持基本 DMA 请求/应答协议的 DMA 通道。

要为 DMA 传输配置音频子模块，可将 SAI_xCR1 寄存器中的 DMAEN 位置 1。DMA 请求直接由 FIFO 控制器管理，具体取决于 FIFO 阈值（更多详细信息，请参见*第 47.3.9 节：内部 FIFO*）。DMA 传输方向与 SAI 音频子模块配置相关：

- 如果音频模块用作发送器，则音频模块的 FIFO 控制器将输出 DMA 请求以向 FIFO 加载 SAI_xDR 寄存器中写入的数据。
- 如果音频模块用作接收器，则 DMA 请求与来自 SAI_xDR 寄存器的读取操作相关。

按照下面的顺序将 SAI 接口配置为 DMA 模式：

1. 配置 SAI 和 FIFO 阈值以指定何时启动 DMA 请求。
2. 配置 SAI DMA 通道。
3. 使能 DMA。
4. 使能 SAI 接口。

注： *配置 SAI 模块前，必须禁止 SAI DMA 通道。*

47.4 SAI 中断

SAI 支持 7 个中断源，如*表 286* 所示。

表 286. SAI 中断源

中断源	中断组	音频模块模式	中断使能	中断清零
FREQ	FREQ	主或从 接收器或发送器	SAI_xIM 寄存器中的 FREQIE	取决于： – FIFO 阈值设置（SAI_xCR2 中的 FLVL 位） – 通信方向（发送器或接收器） 更多详细信息，请参见*第 47.3.9 节：内部 FIFO*。
OVRUDR	ERROR	主或从 接收器或发送器	SAI_xIM 寄存器中的 OVRUDRIE	SAI_xCLRFR 寄存器中的 COVRUDR = 1
AFSDET	ERROR	从 （不适用于 AC’97 模式和 SPDIF 模式）	SAI_xIM 寄存器中的 AFSDETIE	SAI_xCLRFR 寄存器中的 CAFSDET = 1
LFSDET	ERROR	从 （不适用于 AC’97 模式和 SPDIF 模式）	SAI_xIM 寄存器中的 LFSDETIE	SAI_xCLRFR 寄存器中的 CLFSDET = 1
CNRDY	ERROR	从 （仅限 AC’97 模式）	SAI_xIM 寄存器中的 CNRDYIE	SAI_xCLRFR 寄存器中的 CCNRDY = 1
MUTEDET	MUTE	主或从 仅限接收模式	SAI_xIM 寄存器中的 MUTEDETIE	SAI_xCLRFR 寄存器中的 CMUTEDET = 1
WCKCFG	ERROR	主模式且 SAI_xCR1 寄存器中的 NOMCK = 0	SAI_xIM 寄存器中的 WCKCFGIE	SAI_xCLRFR 寄存器中的 CWCKCFG = 1

按照以下顺序使能中断：

1. 禁止 SAI 中断。
2. 配置 SAI。
3. 配置 SAI 中断源。
4. 使能 SAI。

47.5 SAI 寄存器

47.5.1 全局配置寄存器 (SAI_GCR)

Global configuration 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
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.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SYNCOUT[1:0]		Res.	Res.	SYNCIN[1:0]	
										rw	rw			rw	rw

位 31:6 保留，必须保持复位值。

位 5:4 **SYNCOUT[1:0]**：同步输出 (Synchronization outputs)

这些位将由软件置 1 和清零。

00：无同步输出信号。将音频模块配置为 SPDIF 模式时，应将 SYNCOUT[1:0] 配置为“无同步输出信号”

01：模块 A 用于其他 SAI 的进一步同步

10：模块 B 用于其他 SAI 的进一步同步

11：保留。必须在音频模块（A 和 B）禁止的情况下设置这些位。

位 3:2 保留，必须保持复位值。

位 1:0 **SYNCIN[1:0]**：同步输入 (Synchronization inputs)

这些位将由软件置 1 和清零。

有关如何编程此位域的信息，请参见*表 279：外部同步选择*。

必须在音频模块（A 和 B）禁止的情况下配置这些位。

如果将两个音频模块之一定义为与外部 SAI 在同步模式下工作（SAI_ACR1 或 SAI_BCR1 寄存器中的 SYNCEN[1:0] = 10），这些位便有意义。

47.5.2 配置寄存器 1 (SAI_ACR1/SAI_BCR1)

Configuration register 1

偏移地址：模块 A 为 0x004

偏移地址：模块 B 为 0x024

复位值：0x0000 0040

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	OSR	MCKDIV[5:0]						NOMCK	Res.	DMAEN	SAIEN
					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.	OUTDRIV	MONO	SYNCEN[1:0]		CKSTR	LSBFIRST	DS[2:0]			Res.	PRTCFG[1:0]		MODE[1:0]	
		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 **OSR**：主时钟的过采样率 (Oversampling ratio for master clock)
0：主时钟频率 = F_{FS} x 256
1：主时钟频率 = F_{FS} x 512

位 25:20 **MCKDIV[5:0]**：主时钟分频器 (Master clock divider)
这些位将由软件置 1 和清零。
000000：内核时钟输入 (sai_x_ker_ck) 1 分频。
否则，主时钟频率将根据*第 47.3.8 节：SAI 时钟发生器*中的公式计算。
音频模块为从模块时，这些位没有意义。
必须在音频模块禁止的情况下配置这些位。

位 19 **NOMCK**：无分频器 (No divider)
此位由软件置 1 和清零。
0：使能主时钟发生器。
1：禁止主时钟发生器。由 MCKDIV 控制的时钟分频器仍可用于生成位时钟。

位 18 保留，必须保持复位值。

位 17 **DMAEN**：DMA 使能 (DMA enable)
此位由软件置 1 和清零。
0：禁止 DMA
1：使能 DMA
注： 在接收模式下，必须在 DMAEN 位置 1 前配置 MODE[1:0] 位，以避免 DMA 请求，原因是复位后音频模块将默认以发送模式工作。

位 16 **SAIEN**：音频模块使能 (Audio block enable)
该位由软件置 1。
要关闭音频模块，必须由应用软件将该位编程为 0 并轮询该位，直到该位读回 0，这表示该模块已完全禁止。将该位置 1 前，先检查该位是否已置 0，否则不会执行此使能命令。
该位可用于控制 SAI 音频模块的状态。如果在音频帧传输期间禁止该位，则正在进行的传输将完成并且单元在该音频帧传输结束时完全禁止。
0：禁止 SAI 音频模块。
1：使能 SAI 音频模块。
注： 当 SAI 模块（A 或 B）配置为主模式时，SAI 模块输入中必须有时钟，然后才能将 SAIEN 位置 1。

位 15:14 保留，必须保持复位值。

位 13 **OUTDRIV**：输出驱动 (Output drive)
此位由软件置 1 和清零。
0：当 SAIEN 置 1 时驱动音频模块输出。
1：在该位置 1 后立即驱动音频模块输出。
注： 该位必须在音频模块配置后的使能前置 1。

位 12 **MONO**：单声道模式 (Mono mode)
此位由软件置 1 和清零。仅当 Slot 数为 2 时该位才有意义。如果选择了单声道模式，则当音频模块用作发送器时，Slot 0 的数据将复制到 Slot 1 上。在接收模式下，将丢弃 Slot 1 并仅存储从 Slot 0 接收的数据。更多详细信息，请参见*单声道/立体声模式*一节。
0：立体声模式。
1：单声道模式。

位 11:10 **SYNCEN[1:0]**：同步使能 (Synchronization enable)

这些位将由软件置 1 和清零。必须在音频子模块禁止的情况下配置这些位。

00：音频子模块处于异步模式。

01：音频子模块与另一个内部音频子模块同步。这种情况下，必须将该音频子模块配置为从模式。

10：音频子模块与外部 SAI 的嵌入式外设同步。这种情况下，应将音频子模块配置为从模式。

11：保留。

注： 使能 SPDIF 模式后，应将音频子模块配置为异步。

位 9 **CKSTR**：时钟选通边沿 (Clock strobing edge)

此位由软件置 1 和清零。必须在音频模块禁止的情况下配置该位。SPDIF 音频协议下该位没有意义。

0：在 SCK 上升沿更改 SAI 生成的信号，而在 SCK 下降沿对 SAI 接收的信号进行采样。

1：在 SCK 下降沿更改 SAI 生成的信号，而在 SCK 上升沿对 SAI 接收的信号进行采样。

位 8 **LSBFIRST**：最低有效位优先 (Least significant bit first)

此位由软件置 1 和清零。必须在音频模块禁止的情况下配置该位。AC'97 音频协议下该位没有意义，原因是传输 AC'97 数据时，数据的 MSB 位优先。SPDIF 音频协议下该位没有意义，原因是传输 SPDIF 数据时，数据的 LSB 位优先。

0：优先传输数据的 MSB

1：优先传输数据的 LSB

位 7:5 **DS[2:0]**：数据大小 (Data size)

这些位将由软件置 1 和清零。选择 SPDIF 协议时（位 PRTCFG[1:0]），将忽略这些位，原因是在这种情况下帧和数据大小是固定的。通过 COMP[1:0] 位选择压扩模式时，将忽略 DS[1:0]，原因是算法已将数据大小固定为 8 位。

必须在音频模块禁止的情况下配置这些位。

000：保留

001：保留

010：8 位

011：10 位

100：16 位

101：20 位

110：24 位

111：32 位

位 4 保留，必须保持复位值。

位 3:2 **PRTCFG[1:0]**：协议配置 (Protocol configuration)

这些位将由软件置 1 和清零。必须在音频模块禁止的情况下配置这些位。

00：自由协议。通过设置大部分配置寄存器位以及帧配置寄存器，自由协议允许用户使用音频模块这一强大的配置功能来处理特定的音频协议（如 I2S、LSB/MSB 对齐、TDM、PCM/DSP...）。

01：SPDIF 协议

10：AC'97 协议

11：保留

位 1:0 **MODE[1:0]**：SAIx 音频模块模式 (SAIx audio block mode)

这些位将由软件置 1 和清零。必须在 SAIx 音频模块禁止的情况下配置这些位。

00：主发送器

01：主接收器

10：从发送器

11：从接收器

注： 如果将音频模块配置为 SPDIF 模式，则将强制设置主发送模式 (MODE[1:0] = 00)。在主发送模式下，音频模块将立即开始生成 FS 和时钟。

47.5.3 配置寄存器 2 (SAI_ACR2/SAI_BCR2)

Configuration register 2

偏移地址：模块 A 为 0x008

偏移地址：模块 B 为 0x028

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
COMP[1:0]		CPL	MUTECNT[5:0]						MUTE VAL	MUTE	TRIS	F FLUSH	FTH[2:0]		
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	w	rw	rw	rw

位 31:16 保留，必须保持复位值。

位 15:14 **COMP[1:0]**：压扩模式 (Companding mode)。

这些位将由软件置 1 和清零。μ-Law 和 A-Law 算法是 CCITT G.711 建议的一部分，要使用何种补码类型取决于 *CPL 位*。

数据扩展还是数据压缩由 MODE[0] 位的状态确定。

如果将音频模块配置为发送器，则应用数据压缩。

如果将音频模块配置为接收器，则自动应用数据扩展。

更多详细信息，请参见*压扩模式*一节。

00：不支持压扩算法

01：保留。

10：μ-Law 算法

11：A-Law 算法

注： 仅当选择了 TDM 协议时才能使用压扩模式。

位 13 **CPL**：补码位 (Complement bit)。

此位由软件置 1 和清零。

该位定义用于压扩模式的补码类型。

0：1 的补码表示。

1：2 的补码表示。

注： 仅当压扩模式为 μ-Law 算法或 A-Law 算法时该位才有效。

位 12:7 **MUTECNT[5:0]**：静音计数器 (Mute counter)。

这些位将由软件置 1 和清零。这些位仅用于接收模式。

这些位中所设置的值将与接收模式下检测到的连续静音帧数量进行比较。当静音帧数量与该值相等时，MUTEDET 标志置 1，并且在 MUTEDETIE 位置 1 的情况下，还将生成中断。

更多详细信息，请参见*静音模式*一节。

位 6 **MUTEVAL**：静音值 (Mute value)。

该位由软件置 1 和清零。必须在使能音频模块 (SAIEN) 前写入。仅当音频模块用作发送器、Slot 数小于或等于 2 并且 MUTE 位置 1 时，该位才有意义。

如果声明了 2 个以上的 Slot，则无论 MUTEVAL 位的值为何，静音模式下发送的位值都将等于 0。

如果 Slot 数小于或等于 2 且 MUTEVAL = 1，则为每个 Slot 发送的 MUTE 值将是上一帧期间发送的值。

更多详细信息，请参见*静音模式*一节。

0：静音模式期间发送位值 0。

1：静音模式期间发送上一个值。

注：　该位对 SPDIF 音频模块无意义，从而也不使用。

位 5 **MUTE**：静音 (Mute)。

此位由软件置 1 和清零。仅当音频模块用作发送器时该位才有意义。Slot 数小于或等于 2 时，MUTE 值与 MUTEVAL 值相关；Slot 数大于 2 时，MUTE 值等于 0。

更多详细信息，请参见*静音模式*一节。

0：禁止静音模式。

1：使能静音模式。

注：　该位对 SPDIF 音频模块无意义，从而也不使用。

位 4 **TRIS**：数据线的三态管理 (Tristate management on data line)。

此位由软件置 1 和清零。仅当将音频模块配置为发送器时该位才有意义。当音频模块配置为 SPDIF 模式时不使用该位。应在 SAI 禁止时配置此位。

更多详细信息，请参见*无效 Slot 上的输出数据线管理*一节。

0：Slot 无效时，SD 输出线仍由 SAI 驱动。

1：SD 输出线将在上一个有效 Slot（下一个 Slot 无效）的最后一个数据位传输结束时释放（高阻态）。

位 3 **FFLUSH**：FIFO 刷新 (FIFO flush)。

该位由软件置 1，始终读为 0。应在 SAI 禁止时配置此位。

0：禁止 FIFO 刷新。

1：FIFO 刷新。将此位编程为 1 可触发 FIFO 刷新。所有的内部 FIFO 指针（读和写）将清零。这种情况下，仍存留在 FIFO 中的数据丢失（发送或接收数据不会继续丢失）。刷新 SAI 前，必须禁止 DMA 数据流 / 中断。

位 2:0 **FTH[2:0]**：FIFO 阈值 (FIFO threshold)。

此位由软件置 1 和清零。

000：FIFO 为空

001：¼ FIFO

010：½ FIFO

011：¾ FIFO

100：FIFO 已满

101：保留

110：保留

111：保留

47.5.4 帧配置寄存器 (SAI_AFRCR/SAI_BFRCR)

Frame configuration register

偏移地址：模块 A 为 0x00C

偏移地址：模块 B 为 0x02C

复位值：0x0000 0007

注： *该寄存器对于 AC'97 和 SPDIF 音频协议无意义*

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.	FSOFF	FSPOL	FSDEF
													rw	rw	r

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

位 31:19 保留，必须保持复位值。

位 18 **FSOFF**：帧同步偏移 (Frame synchronization offset)。

此位由软件置 1 和清零。该位对 AC'97 或 SPDIF 音频模块配置无意义，从而也不使用。必须在音频模块禁止的情况下配置该位。

0：在 Slot 0 的第一位上使能 FS。

1：在 Slot 0 第一位的前一位上使能 FS。

位 17 **FSPOL**：帧同步极性 (Frame synchronization polarity)。

此位由软件置 1 和清零。该位用于配置 FS 信号上的 SOF 电平。该位对 AC'97 或 SPDIF 音频模块配置无意义，从而也不使用。

必须在音频模块禁止的情况下配置该位。

0：FS 为低电平有效（下降沿）

1：FS 为高电平有效（上升沿）

位 16 **FSDEF**：帧同步定义 (Frame synchronization definition)。

此位由软件置 1 和清零。

0：FS 信号为起始帧信号

1：FS 信号为 SOF 信号 + 通道识别信号

此位置 1 时，SAI_xSLOTR 寄存器中定义的 Slot 数必须为偶数。这意味着有半数 Slot 将用于左通道，其他 Slot 用于右通道（例如，对于 I2S 或 MSB/LSB 对齐等协议，该位必须置 1）。

此位对 AC'97 或 SPDIF 音频模块配置无意义，从而也不使用。必须在音频模块禁止的情况下配置该位。

位 15 保留，必须保持复位值。

位 14:8 **FSALL[6:0]**：帧同步有效电平长度 (Frame synchronization active level length)。

这些位将由软件置 1 和清零。这些位用于指定音频帧中 FS 信号的有效电平长度，以位时钟数 (SCK) + 1 (FSALL[6:0] + 1) 表示。

这些位对 AC'97 或 SPDIF 音频模块配置无意义，从而也不使用。

必须在音频模块禁止的情况下配置这些位。

位 7:0 **FRL[7:0]**：帧长度 (Frame length)。

这些位将由软件置 1 和清零。这些位用于定义以 SCK 时钟周期数表示的音频帧长度：帧中的位数等于 FRL[7:0] + 1。

音频帧中发送的位数必须大于或等于 8，否则音频模块将出现操作异常。数据大小为 8 位且在 SAI_xSLOTR 寄存器的 NBSLOT[4:0] 中只定义了一个 Slot (NBSLOT[3:0] = 0000) 时便属于这种情况。

在主模式下，如果使用主时钟（MCLK_x 引脚上提供），则帧长度应为 8 到 256 之间的一个等于 2 的几次幂的数。不使用主时钟 (NOMCK = 1) 时，建议将帧长度编程为 8 到 256 之间的值。

这些位对 AC'97 或 SPDIF 音频模块配置无意义，从而也不使用。

47.5.5 Slot 寄存器 (SAI_ASLOTR/SAI_BSLOTR)

Slot register

偏移地址：模块 A 为 0x010

偏移地址：模块 B为 0x030

复位值：0x0000 0000

注： *该寄存器对于 AC'97 和 SPDIF 音频协议无意义*

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SLOTEN[15:0]															
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
Res.	Res.	Res.	Res.	NBSLOT[3:0]				SLOTSZ[1:0]		Res.	FBOFF[4:0]				
				rw	rw	rw	rw	rw	rw		rw	rw	rw	rw	rw

位 31:16 **SLOTEN[15:0]**：Slot 使能 (Slot enable)。

这些位将由软件置 1 和清零。

每个 SLOTEN 位都对应于从 0 到 15 的一个 Slot 位置（最多 16 个 Slot）。

0：无效 Slot。

1：有效 Slot。

必须在音频模块禁止的情况下使能 Slot。

在 AC'97 或 SPDIF 模式下会忽略这些位。

位 15:12 保留，必须保持复位值。

位 11:8 **NBSLOT[3:0]**：音频帧中的 Slot 数 (Number of slots in an audio frame)。

这些位将由软件置 1 和清零。

此位域中设置的值表示音频帧中的 Slot 数 + 1（包括无效 Slot 数）。Slot 数最大值为 16。

SAI_xFRCR 寄存器中的 FSDEF 位置 1 时，Slot 数应为偶数。

必须在音频模块禁止的情况下配置 Slot 数。

在 AC'97 或 SPDIF 模式下会忽略这些位。

位 7:6 **SLOTSZ[1:0]**：Slot 大小 (Slot size)
此位由软件置 1 和清零。
Slot 大小必须大于或等于数据大小。如果不满足该条件，SAI 的行为将不确定。
有关如何驱动 SD 线的信息，请参见*无效 Slot 上的输出数据线管理*一节。
必须在音频模块禁止的情况下配置这些位。
在 AC'97 或 SPDIF 模式下会忽略这些位。
00：Slot 大小与数据大小（在 SAI_xCR1 寄存器的 DS[3:0] 位中指定）相当。
01：16 位
10：32 位
11：保留

位 5 保留，必须保持复位值。

位 4:0 **FBOFF[4:0]**：第一个位偏移 (First bit offset)
这些位将由软件置 1 和清零。
此位域中设置的值定义 Slot 中第一个数据传输位的位置。它表示一个偏移值。在发送模式下，此数据位域以外的位将强制清零。在接收模式下，将丢弃额外接收的位。
必须在音频模块禁止的情况下配置这些位。
在 AC'97 或 SPDIF 模式下会忽略这些位。

47.5.6 中断屏蔽寄存器 2 (SAI_AIM/SAI_BIM)

Interrupt mask register 2

偏移地址：模块 A 为 0x014

偏移地址：模块 B 为 0x034

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.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LFSDETIE	AFSDETIE	CNRDYIE	FREQIE	WCKCFGIE	MUTEDETIE	OVRUDRIE
									rw	rw	rw	rw	rw	rw	rw

位 31:7 保留，必须保持复位值。

位 6 **LFSDETIE**：帧同步滞后检测中断使能 (Late frame synchronization detection interrupt enable)。
此位由软件置 1 和清零。
0：禁止中断
1：使能中断
该位置 1 时，若 SAI_xSR 寄存器中的 LFSDET 位置 1，则生成中断。
该位对于 AC'97、SPDIF 模式无意义；若音频模块为主模块，该位也无意义。

位 5 **AFSDETIE**：帧同步提前检测中断使能 (Anticipated frame synchronization detection interrupt enable)。
此位由软件置 1 和清零。
0：禁止中断
1：使能中断
该位置 1 时，若 SAI_xSR 寄存器中的 AFSDET 位置 1，则生成中断。
该位对于 AC'97、SPDIF 模式无意义；若音频模块为主模块，该位也无意义。

位 4 **CNRDYIE**：编解码器未就绪中断使能 (Codec not ready interrupt enable) (AC'97)。
此位由软件置 1 和清零。
0：禁止中断
1：使能中断
若使能该中断，音频模块将在 AC'97 帧的 Slot 0 (tag0) 中检测连接到该线路的编解码器是否就绪。
如果未就绪，SAI_xSR 寄存器中的 CNRDY 标志置 1，并生成中断。
仅当通过 PRTCFG[1:0] 位选择了 AC'97 模式且音频模块用作接收器时，该位才有意义。

位 3 **FREQIE**：FIFO 请求中断使能 (FIFO request interrupt enable)。
此位由软件置 1 和清零。
0：禁止中断
1：使能中断
该位置 1 时，若 SAI_xSR 寄存器中的 FREQ 位置 1，则生成中断。
在接收模式下，必须在 FREQIE 位置 1 前配置 MODE 位，以避免寄生中断，原因是复位后音频模块将默认用作发送器。

位 2 **WCKCFGIE**：时钟配置错误中断使能 (Wrong clock configuration interrupt enable)。
此位由软件置 1 和清零。
0：禁止中断
1：使能中断
仅在将音频模块配置为主模式 (MODE[1] = 0) 且 NOMCK = 0 时才执行此位。
该位在 SAI_xSR 寄存器中的 WCKCFG 标志置 1 时生成中断。
注：　该位仅用于 TDM 模式，其他模式下没有意义。

位 1 **MUTEDETIE**：静音检测中断使能 (Mute detection interrupt enable)。
此位由软件置 1 和清零。
0：禁止中断
1：使能中断
该位置 1 时，若 SAI_ASR 寄存器中的 MUTEDET 位置 1，则生成中断。
仅当音频模块配置为以发送模式工作时该位才有意义。

位 0 **OVRUDRIE**：上溢/下溢中断使能 (Overrun/underrun interrupt enable)。
此位由软件置 1 和清零。
0：禁止中断
1：使能中断
该位置 1 时，若 SAI_ASR 寄存器中的 OVRUDR 位置 1，则生成中断。

47.5.7 状态寄存器 (SAI_ASR/SAI_BSR)

Status register

偏移地址：模块 A 为 0x018

偏移地址：模块 B 为 0x038

复位值：0x0000 0008

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.	FLVL[2:0]		
													r	r	r

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LFSDET	AFSDET	CNRDY	FREQ	WCKCFG	MUTEDET	OVRUDR
									r	r	r	r	r	r	r

位 31:19 保留，必须保持复位值。

位 18:16 **FLVL[2:0]**：FIFO 阈值电压 (FIFO level threshold)。
该位为只读。 FIFO 阈值标志只通过硬件管理，其设置取决于 SAI 模块的配置 （发送器或接收器模式）。
如果将 SAI 模块配置为发送器：
000：FIFO 为空
001：FIFO <= ¼，但非空
010：¼ < FIFO <= ½
011：½ < FIFO <= ¾
100：¾ < FIFO，但未满
101：FIFO 已满
如果将 SAI 模块配置为接收器：
000：FIFO 为空
001：FIFO < ¼，但非空
010：¼ <= FIFO < ½
011：½ =< FIFO < ¾
100：¾ =< FIFO，但未满
101：FIFO 已满

位 15:7 保留，必须保持复位值。

位 6 **LFSDET**：帧同步滞后检测 (Late frame synchronization detection)。
该位为只读。
0：无错误。
1：帧同步信号未在正确的时刻出现。
仅当音频模块配置为以从模式工作时，此标志才能置 1。
不适用于 AC’97 或 SPDIF 模式。
它可在 SAI_xIM 寄存器中的 LFSDETIE 位置 1 时生成中断。
在软件将 SAI_xCLRFR 寄存器中的 CLFSDET 位置 1 时清除该标志。

位 5 **AFSDET**：帧同步提前检测 (Anticipated frame synchronization detection)。
该位为只读。
0：无错误。
1：提前检测到帧同步信号。
仅当音频模块配置为以从模式工作时，此标志才能置 1。
不适用于 AC’97 或 SPDIF 模式。
它可在 SAI_xIM 寄存器中的 AFSDETIE 位置 1 时生成中断。
在软件将 SAI_xCLRFR 寄存器中的 CAFSDET 位置 1 时清除该标志。

位 4 **CNRDY**：编解码器未就绪 (Codec not ready)。
该位为只读。
0：外部 AC’97 编解码器已就绪
1：外部 AC’97 编解码器未就绪
仅当在 SAI_xCR1 寄存器中选择了 AC’97 音频模块并且音频模块配置为接收器模式时，才使用该位。
它可在 SAI_xIM 寄存器中的 CNRDYIE 位置 1 时生成中断。
在软件将 SAI_xCLRFR 寄存器中的 CCNRDY 位置 1 时清除该标志。

位 3 **FREQ**：FIFO 请求 (FIFO request)。
该位为只读。
0：无 FIFO 请求。
1：FIFO 请求读取或写入 SAI_xDR。
请求内容取决于音频模块的配置：
– 如果模块配置为发送模式，则 FIFO 请求与向 SAI_xDR 中写入相关。
– 如果音频模块配置为接收模式，则 FIFO 请求与从 SAI_xDR 中读取相关。
此标志可在 SAI_xIM 寄存器中的 FREQIE 位置 1 时生成中断。

位 2 **WCKCFG**：时钟配置错误标志 (Wrong clock configuration flag)。
该位为只读。
0：时钟配置正确
1：时钟配置不符合*第 47.3.6 节：帧同步*中定义的帧长度规范（SAI_xFRCR 寄存器中 FRL[7:0] 位的配置）
该位仅在音频模块工作在主模式 (MODE[1] = 0) 下且 NOMCK = 0 时使用。
它可在 SAI_xIM 寄存器中的 WCKCFGIE 位置 1 时生成中断。
在软件将 SAI_xCLRFR 寄存器中的 CWCKCFG 位置 1 时清除该标志。

位 1 **MUTEDET**：静音检测 (Mute detection)。
该位为只读。
0：SD 输入线上未检测到 MUTE 值
1：在SD 输入线上检测到指定数量的连续音频帧中的 MUTE 值（0 值）
如果在指定音频帧的每个 Slot 或在一定数量（在 SAI_xCR2 寄存器中的 MUTECNT 位中设置）的连续音频帧中接收到连续的 0 值，则该标志置 1。
它可在 SAI_xIM 寄存器中的 MUTEDETIE 位置 1 时生成中断。
在软件将 SAI_xCLRFR 寄存器中的 CMUTEDET 位置 1 时清除该标志。

位 0 **OVRUDR**：上溢/下溢 (Overrun/underrun)。
该位为只读。
0：无上溢/下溢错误。
1：检测到上溢/下溢错误。
仅当音频模块分别被配置为接收器和发送器时才会发生上溢和下溢情况。
它可在 SAI_xIM 寄存器中的 OVRUDRIE 位置 1 时生成中断。
在软件将 SAI_xCLRFR 寄存器中的 COVRUDR 位置 1 时清除该标志。

47.5.8 清除标志寄存器 (SAI_ACLRFR/SAI_BCLRFR)

Clear flag register

偏移地址：模块 A 为 0x01C

偏移地址：模块 B 为 0x03C

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.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CLFSDET	CAFSDET	CCNRDY	Res.	CWCKCFG	CMUTEDET	COVRUDR
									w	w	w		w	w	w

位 31:7 保留，必须保持复位值。

位 6 **CLFSDET**：清除帧同步滞后检测标志 (Clear late frame synchronization detection flag)。
该位为只写位。
将此位编程为 1 可清除 SAI_xSR 寄存器中的 LFSDET 标志。
该位不适用于 AC'97 或 SPDIF 模式
读取该位将始终返回值 0。

位 5 **CAFSDET**：清除帧同步提前检测标志 (Clear anticipated frame synchronization detection flag)。
该位为只写位。
将此位编程为 1 可清除 SAI_xSR 寄存器中的 AFSDET 标志。
不适用于 AC'97 或 SPDIF 模式。
读取该位将始终返回值 0。

位 4 **CCNRDY**：清除编解码器未就绪标志 (Clear codec not ready flag)。
该位为只写位。
将此位编程为 1 可清除 SAI_xSR 寄存器中的 CNRDY 标志。
仅当在 SAI_xCR1 寄存器中选择了 AC'97 音频协议时，才使用该位。
读取该位将始终返回值 0。

位 3 保留，必须保持复位值。

位 2 **CWCKCFG**：清除时钟配置错误标志 (Clear wrong clock configuration flag)。
该位为只写位。
将此位编程为 1 可清除 SAI_xSR 寄存器中的 WCKCFG 标志。
仅当音频模块设置为主模块时 (MODE[1] = 0) 且 SAI_xCR1 寄存器中的 NOMCK = 0 时，才使用此位。
读取该位将始终返回值 0。

位 1 **CMUTEDET**：清除静音检测标志 (Clear Mute detection flag)。
该位为只写位。
将此位编程为 1 可清除 SAI_xSR 寄存器中的 MUTEDET 标志。
读取该位将始终返回值 0。

位 0 **COVRUDR**：清除上溢/下溢标志 (Clear overrun/underrun)。
该位为只写位。
将此位编程为 1 可清除 SAI_xSR 寄存器中的 OVRUDR 标志。
读取该位将始终返回值 0。

47.5.9 数据寄存器 (SAI_ADR/SAI_BDR)

Data register

偏移地址：模块 A 为 0x020

偏移地址：模块 B 为 0x040

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	rw	rw	rw	rw

位 31:0 **DATA[31:0]**：数据 (Data)
若 FIFO 未满，写入该寄存器可向 FIFO 加载数据。
若 FIFO 非空，读取该寄存器可清空 FIFO 。

47.5.10 PDM 控制寄存器 (SAI_PDMCR)

PDM control register

偏移地址：0x0044

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.	Res.	Res.	Res.	Res.	CKEN2	CKEN1	Res.	Res.	MICNBR[1:0]		Res.	Res.	Res.	PDMEN
						rw	rw			rw	rw				rw

位 31:16 保留，必须保持复位值。

位 15:10 保留，必须保持复位值。

位 9 **CKEN2**：2 号比特流时钟使能 (Clock enable of bitstream clock number 2)
此位由软件置 1 和清零。
0：禁止 SAI_CK2 时钟
1：使能 SAI_CK2 时钟
注： 建议不要在 PDMEN = 1 时配置该位。

位 8 **CKEN1**：1 号比特流时钟使能 (Clock enable of bitstream clock number 1)
此位由软件置 1 和清零。
0：禁止 SAI_CK1 时钟
1：使能 SAI_CK1 时钟
注： 建议不要在 PDMEN = 1 时配置该位。

位 7:6 保留，必须保持复位值。

位 5:4 **MICNBR[1:0]**：麦克风数量 (Number of microphones)
此位由软件置 1 和清零。
0b00：配置有 2 个麦克风
0b01：配置有 4 个麦克风
0b10：配置有 6 个麦克风
0b11：配置有 8 个麦克风
注： 建议不要在 PDMEN = 1 时配置该字段。

位 3:1 保留，必须保持复位值。

位 0 **PDMEN**：PDM 使能 (PDM enable)
此位由软件置 1 和清零。该位可用于控制 PDM 接口模块的状态。
在使能 PDM 接口之前，确保 SAI 已工作在 TDM 主模式下。
0：禁止 PDM 接口
1：使能 PDM 接口

47.5.11 PDM 延迟寄存器 (SAI_PDMDLY)

PDM delay register

偏移地址：0x0048

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	DLYM4R[2:0]			Res.	DLYM4L[2:0]			Res.	DLYM3R[2:0]			Res.	DLYM3L[2:0]		
	rw	rw	rw		rw	rw	rw		rw	rw	rw		rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	DLYM2R[2:0]			Res.	DLYM2L[2:0]			Res.	DLYM1R[2:0]			Res.	DLYM1L[2:0]		
	rw	rw	rw		rw	rw	rw		rw	rw	rw		rw	rw	rw

位 31 保留，必须保持复位值。

位 30:28 **DLYM4R[2:0]**：**第 4 对**的第二个麦克风的延迟线 (Delay line for second microphone of **pair 4**)
此位由软件置 1 和清零。
0b000：无延迟
0b001：延迟 1 个 T_{SAI_CK} 周期
0b010：延迟 2 个 T_{SAI_CK} 周期
...
0b111：延迟 7 个 T_{SAI_CK} 周期

该位域可实时更改。

位 27 保留，必须保持复位值。

位 26:24 **DLYM4L[2:0]**：第 4 对的第一个麦克风的延迟线 (Delay line for first microphone of pair 4)
此位由软件置 1 和清零。
0b000：无延迟
0b001：延迟 1 个 T_{SAI_CK} 周期
0b010：延迟 2 个 T_{SAI_CK} 周期
...
0b111：延迟 7 个 T_{SAI_CK} 周期

该位域可实时更改。

位 23 保留，必须保持复位值。

位 22:20 **DLYM3R[2:0]**：第 3 对的第二个麦克风的延迟线 (Delay line for second microphone of pair 3)
此位由软件置 1 和清零。
0b000：无延迟
0b001：延迟 1 个 T_{SAI_CK} 周期
0b010：延迟 2 个 T_{SAI_CK} 周期
...
0b111：延迟 7 个 T_{SAI_CK} 周期

该位域可实时更改。

位 19 保留，必须保持复位值。

位 18:16 **DLYM3L[2:0]**：第 3 对的第一个麦克风的延迟线 (Delay line for first microphone of pair 3)
此位由软件置 1 和清零。
0b000：无延迟
0b001：延迟 1 个 T_{SAI_CK} 周期
0b010：延迟 2 个 T_{SAI_CK} 周期
...
0b111：延迟 7 个 T_{SAI_CK} 周期

该位域可实时更改。

位 15 保留，必须保持复位值。

位 14:12 **DLYM2R[2:0]**：第 2 对的第二个麦克风的延迟线 (Delay line for second microphone of pair 2)
此位由软件置 1 和清零。
0b000：无延迟
0b001：延迟 1 个 T_{SAI_CK} 周期
0b010：延迟 2 个 T_{SAI_CK} 周期
...
0b111：延迟 7 个 T_{SAI_CK} 周期

该位域可实时更改。

位 11 保留，必须保持复位值。

位 10:8 **DLYM2L[2:0]**：第 2 对的第一个麦克风的延迟线 (Delay line for first microphone of pair 2)
此位由软件置 1 和清零。
0b000：无延迟
0b001：延迟 1 个 T_{SAI_CK} 周期
0b010：延迟 2 个 T_{SAI_CK} 周期
...
0b111：延迟 7 个 T_{SAI_CK} 周期

该位域可实时更改。

位 7 保留，必须保持复位值。

位 6:4 **DLYM1R[2:0]**：第 1 对的第二个麦克风的延迟线调整 (Delay line adjust for second microphone of pair 1)
此位由软件置 1 和清零。
0b000：无延迟
0b001：延迟 1 个 T_{SAI_CK} 周期
0b010：延迟 2 个 T_{SAI_CK} 周期
...
0b111：延迟 7 个 T_{SAI_CK} 周期

该位域可实时更改。

位 3 保留，必须保持复位值。

位 2:0 **DLYM1L[2:0]**：第 1 对的第一个麦克风的延迟线调整 (Delay line adjust for first microphone of pair 1)
此位由软件置 1 和清零。
0b000：无延迟
0b001：延迟 1 个 T_{SAI_CK} 周期
0b010：延迟 2 个 T_{SAI_CK} 周期
...
0b111：延迟 7 个 T_{SAI_CK} 周期

该位域可实时更改。

47.5.12 SAI 寄存器映射

下表对 SAI 寄存器进行了汇总。

表 287. SAI 寄存器映射和复位值

偏移	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
0x0000	SAI_GCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SYNCOUT[1:0]		Res.	Res.	SYNCIN[1:0]	
	Reset value																											0	0			0	0
0x0004 or 0x0024	SAI_xCR1	Res.	Res.	Res.	Res.	Res.	OSR	MCKDIV[3:0]						NOMCK	Res.	DMAEN	SAIEN	Res.	Res.	OUTDRIV	MONO	SYNCEN[1:0]		CKSTR	LSBFIRST	DS[2:0]		Res.	PRTCFG[1:0]		MODE[1:0]		
	Reset value						0	0	0	0	0	0	0	0	-	0	0			0	0	0	0	0	0	0	1	0		0	0	0	0
0x0008 or 0x0028	SAI_xCR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	COMP[1:0]		CPL	MUTECN[5:0]						MUTE VAL	MUTE	TRIS	FFLUS	FTH		
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x000C or 0x002C	SAI_xFRCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FSOFF	FSPOL	FSDEF	Res.	FSALL[6:0]							FRL[7:0]							
	Reset value														0	0	0		0	0	0	0	0	0	0	0	0	0	0	0	1	1	1
0x0010 or 0x0030	SAI_xSLOTR	SLOTEN[15:0]																Res.	Res.	Res.	Res.	NBSLOT[3:0]				SLOTSZ[1:0]		Res.	FBOFF[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	0		0	0	0	0	0
0x0014 or 0x0034	SAI_xIM	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LFSDET	AFSDETIE	CNRDYIE	FREQIE	WCKCFG	MUTEDET	OVRUDRIE
	Reset value																										0	0	0	0	0	0	0
0x0018 or 0x0038	SAI_xSR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FLVL[2:0]			Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LFSDET	AFSDET	CNRDY	FREQ	WCKCFG	MUTEDET	OVRUDR
	Reset value														0	0	0										0	0	0	0	1	0	0
0x001C or 0x003C	SAI_xCLRFR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LFSDET	CAFSDET	CNRDY	Res.	WCKCFG	MUTEDET	OVRUDR
	Reset value																										0	0	0		0	0	0
0x0020 or 0x0040	SAI_xDR	DATA[31:0]																															
	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

表 287. SAI 寄存器映射和复位值（续）

偏移	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
0x0044	SAI_PDMCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CKEN2	CKEN1	Res.	Res.	MICNBR[1:0]		Res.	Res.	Res.	PDMEN
	Reset value																							0	0			0	0				0
0x0048	SAI_PDMDLY	Res.	DLYM4R[2:0]			Res.	DLYM4L[2:0]			Res.	DLYM3R[2:0]			Res.	DLYM3L[2:0]			Res.	DLYM2R[2:0]			Res.	DLYM2L[2:0]			Res.	DLYM1R[2:0]			Res.	DLYM1L[2:0]		
	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

有关寄存器边界地址的信息，请参见*第 77 页的第 2.2 节*。

48 安全数字输入/输出多媒体卡接口 (SDMMC)

48.1 SDMMC 主要特性

SD/SDIO 多媒体卡 (MMC) 主机接口 (SDMMC) 提供 AHB 总线与 SD 存储卡、SDIO 卡以及 MMC 设备之间的接口。

多媒体卡协会网站 www.mmca.org 中提供了由 MMCA 技术委员会发布的多媒体卡系统规范。

SD 卡协会网站 www.sdcard.org 中提供了 SD 存储卡和 SD I/O 卡系统规范。

SDMMC 具有以下特性：

- 完全兼容*多媒体卡系统规范版本 4.51*。
  卡支持三种不同数据总线模式：1 位（默认）、4 位和8 位。
- 完全兼容先前版本的多媒体卡（向后兼容性）。
- 完全兼容 *SD 存储卡规范版本 4.1*。
  （不支持 SPI 模式和 UHS-II 模式）。
- 完全兼容 *SDIO 卡规范版本 4.0*。
  卡支持两种不同数据总线模式：1 位（默认）和 4 位。
  （不支持 SPI 模式和 UHS-II 模式）。
- 对于 8 位模式，数据传输高达 104 MB/s。
  （取决于允许的最大 I/O 速度）。
- 数据和命令输出使能信号，控制外部双向驱动程序。

多媒体卡/SD 总线将卡连接到主机。

当前版本的 SDMMC 在同一时间只支持一个协议栈版本为 MMC V4.51 或先前版本的 SD/SDIO/MMC 卡。

48.2 SDMMC 总线拓扑

总线的通信基于命令/响应和数据传输。

SD/SDIO/MMC 总线上的基本事务是命令/响应事务。这些类型的总线事务直接在命令或响应结构中传输其信息。此外，某些操作具有数据令牌。

数据传输通过以下方式完成：

- 块模式：块大小为 2^N 字节的数据块，N 为 0-14 范围内的值
- SDIO 多字节模式：块大小为 1-512 字节的单一数据块
- MMC 流模式：连续数据流

与 MMC 卡的数据相互传输在数据块或流中执行。

图 530. SDMMC“无响应”和“无数据”操作

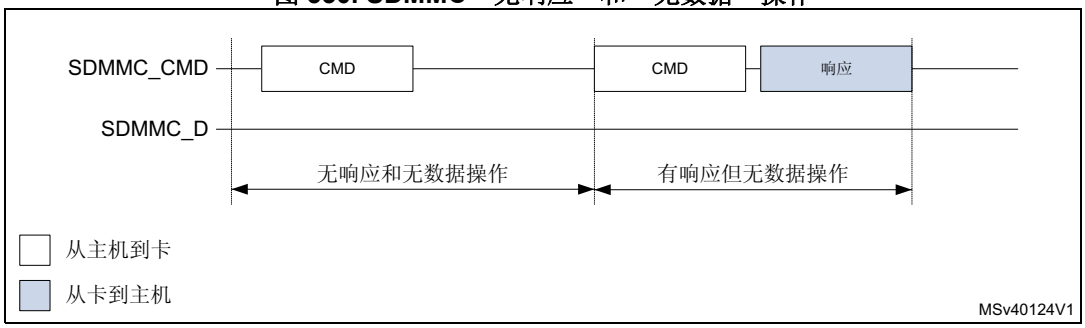
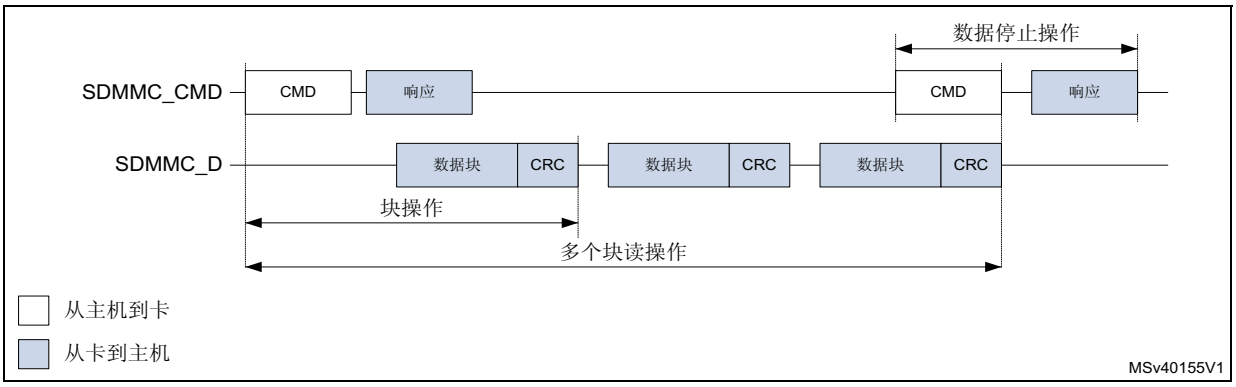
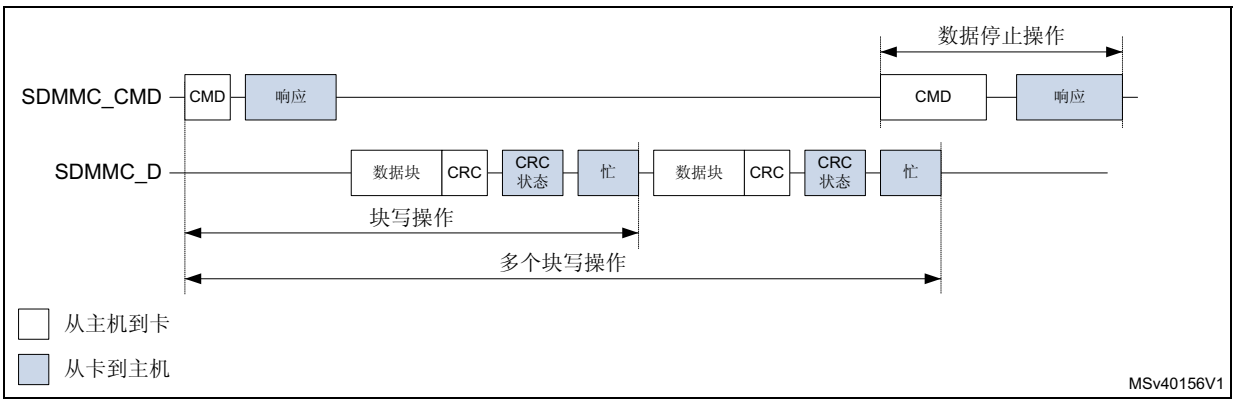


图 531. SDMMC（多个）块读取操作



注：　在具有预定义块数的 MMC 多块读取操作结束时，不需要停止传输命令。

图 532. SDMMC（多个）块写入操作



注：　在具有预定义块数的 MMC 多块写入操作结束时，不需要停止传输命令。

注：　只要发出“繁忙”信号（SDMMC_D0 被拉到低电平），SDMMC 便不会发送任何数据。

图 533. SDMMC（连续）流读取操作

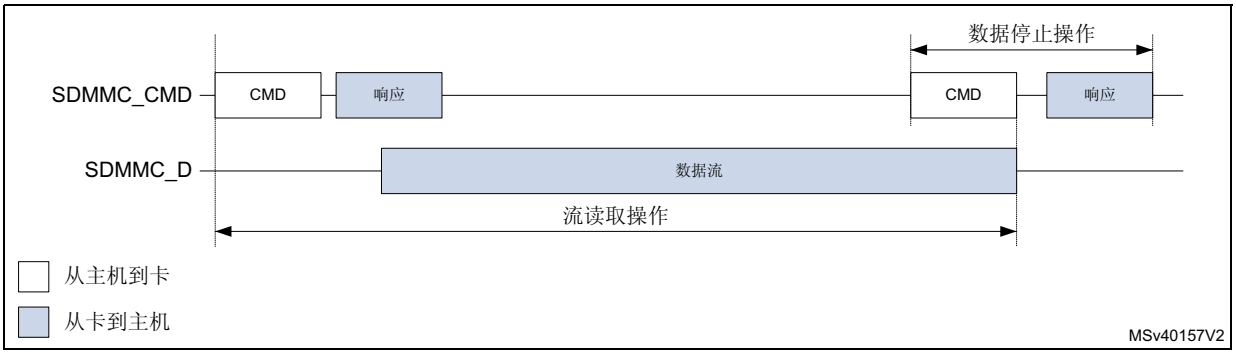


图 534. SDMMC（连续）流写入操作



数据流传输操作只适用于总线配置位宽为 1，且工作模式为单倍速率模式（DS、HS 和 SDR）下的 SDMMC_D0 上。

48.3 SDMMC 工作模式

表 288. SDMMC 工作模式 SD 和 SDIO

SDIO 总线速度模式(1)(2)	最大总线速度 (3) [MB/s]	最大时钟频率 [MHz](4)	信号电压 [V]
DS（默认速度）	12.5	25	3.3
HS（高速）	25	50	3.3
SDR12	12.5	25	1.8
SDR25	25	50	1.8
DDR50	50	50	1.8
SDR50	50	100	1.8

1. SDR 单倍数据速率信号传输。
2. DDR 双倍数据速率信号传输。（在 SDMMC_CK 的两种时钟边沿上均采样数据）。
3. SDIO 总线速度，4 位总线宽度。
4. 最大频率取决于允许的最大 IO 速度。

表 289. SDMMC 工作模式 eMMC

eMMC 总线速度模式 (1)(2)	最大总线速度 (3) [MB/s]	最大时钟频率 [MHz](4)	信号电压 [V]
传统兼容型	26	26	3/1.8/1.2V
高速 SDR	52	52	3/1.8/1.2V
高速 DDR	104	52	3/1.8/1.2V

1. SDR 单倍数据速率信号传输。
2. DDR 双倍数据速率信号传输。（在 SDMMC_CK 的两种时钟边沿上均采样数据）。
3. eMMC 总线速度，8 位总线宽度。
4. 最大频率取决于允许的最大 IO 速度。

48.4 SDMMC 功能说明

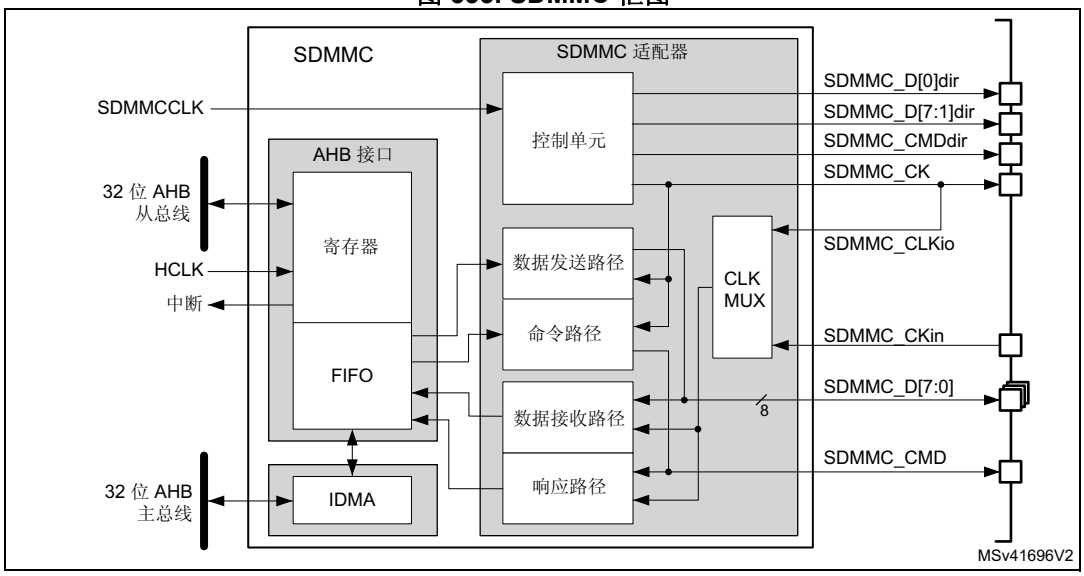
SDMMC 由三部分组成：

- AHB 从接口访问 SDMMC 适配器寄存器，并且生成中断信号和 IDMA 控制信号。
- SDMMC 适配器块提供特定于 MMC/SD/SD I/O 卡的所有功能，如时钟生成单元、命令和数据传输。
- 内部 DMA（IDMA）模块及其 AHB 主接口。

48.4.1 SDMMC 图

图 535 显示了 SDMMC 框图。

图 535. SDMMC 框图



48.4.2 SDMMC 引脚和内部信号

表 290 列出了 SDMMC 内部输入/输出信号，*表 291* 列出了 SDMMC 引脚（复用功能）。

表 290. SDMMC 内部输入/输出信号

信号名称	信号类型	说明
SDMMCCLK	数字输入	SDMMC 内核时钟
HCLK	数字输入	AHB2 时钟
中断	数字输出	SDMMC 全局中断
SDMMC_CKio	数字输入	SD/SDIO/MMC 卡反馈时钟。此信号在内部连接到 SDMMC_CK 引脚（针对 DS 和 HS 模式）。

表 291. SDMMC 引脚

信号名称	信号类型	说明
SDMMC_CK	数字输出	SD/SDIO/MMC 卡的时钟。
SDMMC_CKin	数字输入	来自 SD/SDIO/MMC 卡的外部驱动器的时钟反馈。（用于 SDR12、SDR25、SDR50 和 DDR50）
SDMMC_CMD	数字输入/输出	SD/SDIO/MMC 卡双向命令/响应信号。
SDMMC_CMDdir	数字输出	SDMMC_CMD 信号的 SD/SDIO/MMC 卡 I/O 方向指示。
SDMMC_D[7:0]	数字输入/输出	SD/SDIO/MMC 卡双向数据线。
SDMMC_D[0]dir	数字输出	SDMMC_D0 数据线的 SD/SDIO/MMC 卡 I/O 方向指示。
SDMMC_D[7:1]dir	数字输出	SDMMC_D[3:1] 数据线的 SD/SDIO/MMC 卡 I/O 方向指示。

48.4.3 概述

SDMMC_D[7:0] 线具有不同的工作模式：

- 默认情况下，SDMMC_D0 线用于数据传输。初始化后，主机可以更改数据总线宽度。
- 对于 MMC，可以使用 1 位 (SDMMC_D0)、4 位 (SDMMC_D[3:0]) 或 8 位 (SDMMC_D[7:0]) 的数据总线宽度。
- 对于 SD 或 SDIO 卡，可以使用 1 位 (SDMMC_D0) 或 4 位 (SDMMC_D[3:0]) 的数据总线宽度。所有数据线均以推挽模式运行。

为了连接外部驱动器（电压切换收发器），使用 I/O 方向信号指示数据线上的数据流方向。**SDMMC_D[0]dir** 信号指示 SDMMC_D0 数据线的 I/O 方向，**SDMMC_D[7:1]dir** 则指示 SDMMC_D[3:1] 数据线的方向。

SDMMC_CMD 仅在推挽模式下运行：

为了连接外部驱动器（电压切换收发器），使用 I/O 方向信号 **SDMMC_CMDdir** 指示 SDMMC_CMD 线上的数据流方向。

卡的 **SDMMC_CK** 时钟源自 **SDMMCCLK**：

- 当 SDMMCCLK 时钟的占空比为 50% 时，即使在旁路模式 (CLKDIV = 0) 下也可以使用。
- 当 SDMMCCLK 占空比不是 50% 时，必须使用 CLKDIV 以 2 或更大值 (CLKDIV > 0) 对其进行分频。
- SDMMC_CMD/SDMMC_D[7:0] 输出和 SDMMC_CK 之间的相位关系可通过 NEGEDGE 位进行选择。相位关系取决于 CLKDIV、NEGEDGE 和 DDR 设置。请参见*图 536*。

图 536. SDMMC 命令和数据相位关系

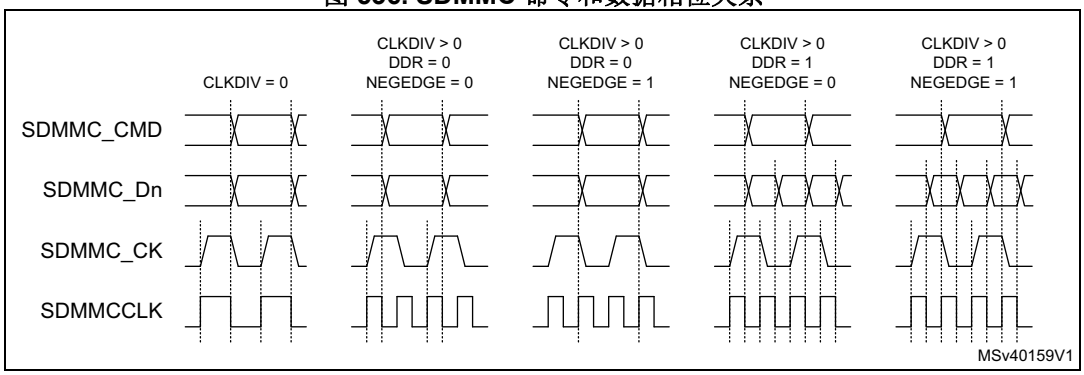


表 292. SDMMC 命令和数据相位选择

CLKDIV	DDR	NEGEDGE	SDMMC_CK	命令输出	数据输出
0	x	x	= SDMMCCLK	在 SDMMCCLK 下降沿生成。	
>0	0	0	在 SDMMCCLK 上升沿生成	在 SDMMC_CK 上升沿后的 SDMMCCLK 下降沿生成。	
		1		在生成 SDMMC_CK 下降沿的同一 SDMMCCLK 上升沿生成。	
	1	0		在 SDMMC_CK 上升沿后的 SDMMCCLK 下降沿生成。	在 SDMMC_CK 边沿后的 SDMMCCLK 下降沿生成。
		1		在生成 SDMMC_CK 下降沿的同一 SDMMCCLK 上升沿生成。	

默认情况下，选择 **sdmmc_io_in_ck** 反馈时钟输入来采样 SDMMC 接收路径中的传入数据。它源自 SDMMC_CK 引脚。

当使用外部驱动器（电压切换收发器）时，可以选择 SDMMC_CKin 反馈时钟输入来对接收数据进行采样。

对于 SD/SDIO/MMC 卡，时钟频率可在 0 到 208 MHz 之间变化（受限于最大 I/O 速度）。

根据选择的总线模式（SDR 或 DDR），每个时钟周期在 SDMMC_D[7:0] 线上传输一位或两位。SDMMC_CMD 线每个时钟周期只传输一位。

48.4.4 SDMMC 适配器

SDMMC 适配器（参见*图 535：SDMMC 框图*）是一个多媒体卡/安全数字存储卡总线主设备，提供与多媒体卡堆栈或安全数字存储卡的接口。它由以下子单元组成：

- 控制单元
- 数据发送路径
- 命令路径
- 数据接收路径
- 响应路径
- 接收数据路径时钟复用器
- 适配器寄存器模块
- 数据 FIFO
- 内部 DMA (IDMA)

注： *适配器寄存器和 FIFO 使用 AHB 时钟域 (HCLK)。控制单元、命令路径和数据发送路径使用 SDMMC 适配器时钟域 (SDMMCCLK)。响应路径和数据接收路径使用来自 SDMMC_CLKio 或 SDMMC_CKin 的 SDMMC 适配器反馈时钟域。*

适配器寄存器模块

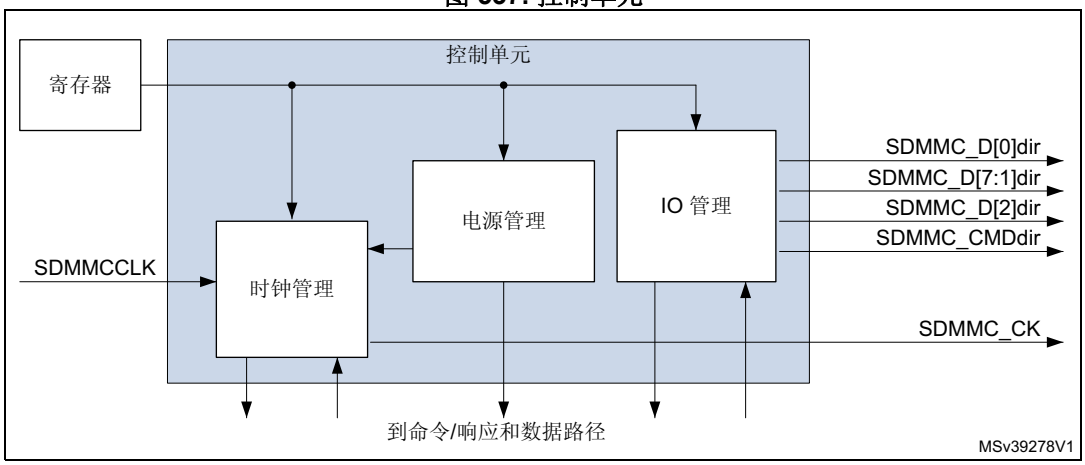
适配器寄存器模块包含所有系统控制寄存器、SDMMC 命令和响应寄存器以及数据 FIFO。

该模块还从 SDMMC 清零寄存器中的相应位单元生成信号，该寄存器用于清零 SDMMC 适配器中的静态标志。

控制单元

控制单元如*图 537* 所示，其包含电源管理功能、带有分频器的 SDMMC_CK 时钟管理功能和 I/O 方向管理功能。

图 537. 控制单元



电源管理子单元会在断电阶段和上电阶段中禁止卡总线输出信号。

有三个电源阶段：

- 掉电
- 上电
- 通电

时钟管理子单元使用 SDMMCCLK 来生成 SDMMC_CK，并提供分频控制。它还负责停止 SDMMC_CK 以进行流控制等。

时钟输出在以下情况下无效：

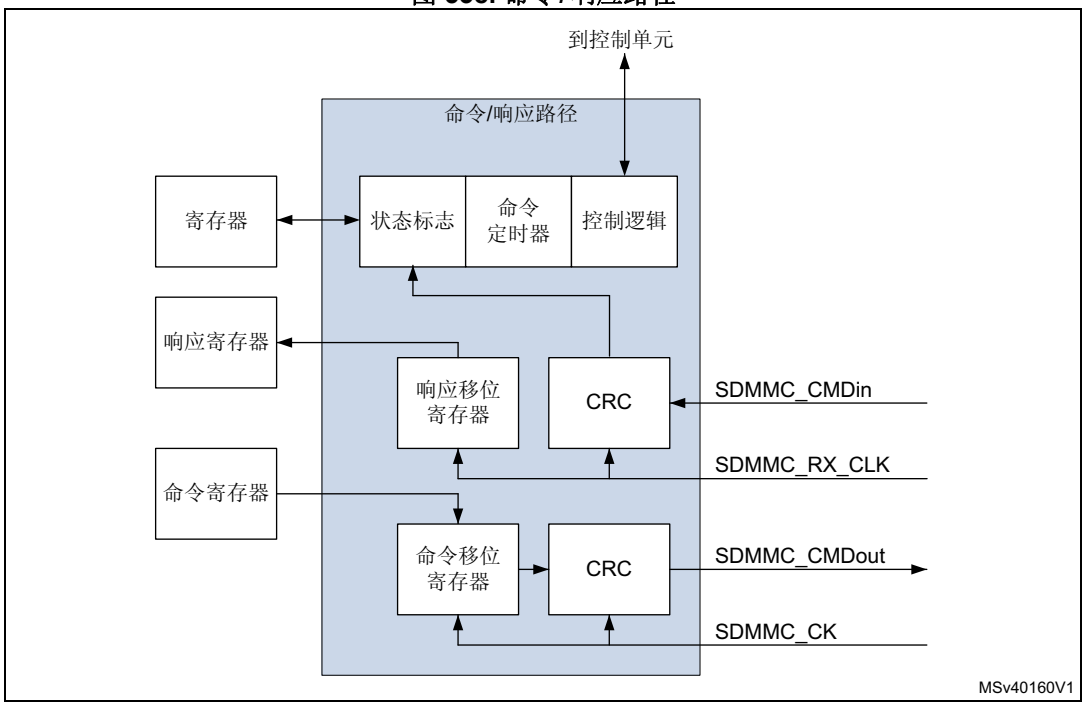
- 复位后。
- 在断电或上电阶段中。
- 节能模式（寄存器位 PWRSAV）已使能并且卡总线处于空闲状态的时间持续 8 个时钟周期。在命令/响应 CPSM 和数据路径 DPSM 子单元进入空闲阶段后的八个周期后，时钟停止。当命令/响应 CPSM 或数据路径 DPSM 激活（使能）时，时钟将重新启动。

I/O 管理子单元负责处理用于控制外部电压收发器的 SDMMC_Dn 和 SDMMC_CMD I/O 方向信号。

命令/响应路径

命令/响应路径子单元在 SDMMC_CMD 线上传送命令和响应。命令路径由 SDMMC_CK 提供时钟，并向卡发送命令。响应路径由 SDMMC_RX_CLK 提供时钟，并从卡接收响应。

图 538. 命令/响应路径

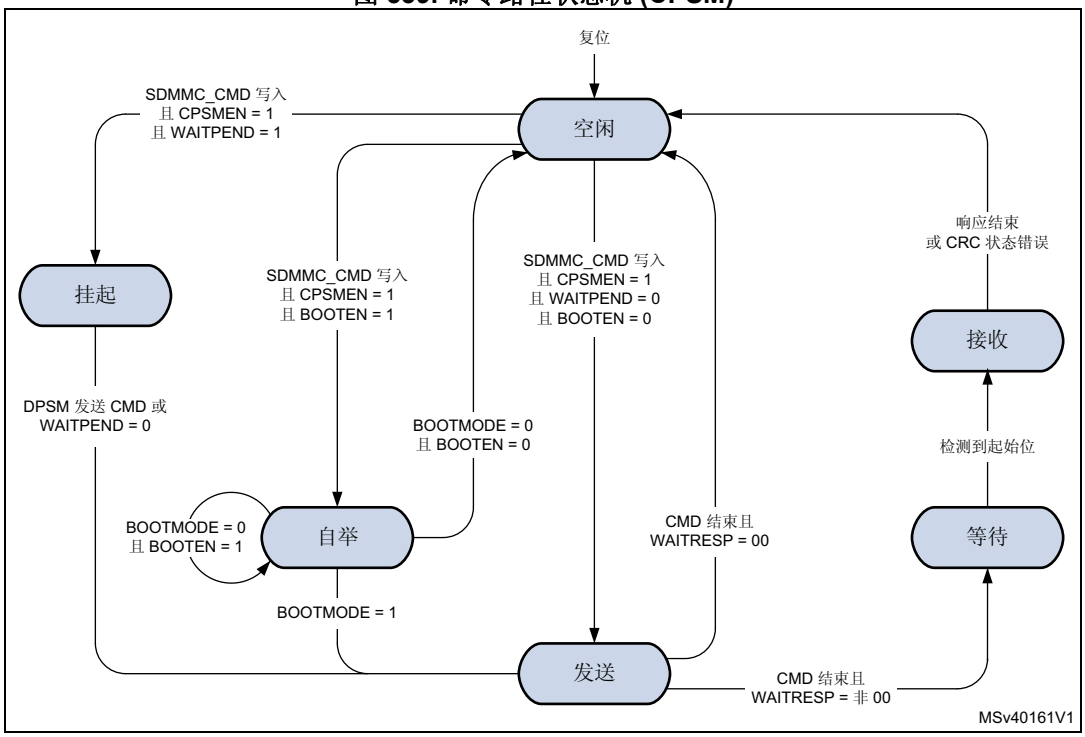


命令/响应路径状态机 (CPSM)

- 写入命令寄存器并且将使能位置 1 后，命令传输将开始。命令发送完成后，将附加 CRC，命令路径状态机 (CPSM) 将设置状态标志，并且：
  - 如果不需要响应，则进入空闲状态。
  - 如果需要响应，则等待响应。
- 接收到响应时，
  - 对于带有 CRC 的响应，将比较接收到的 CRC 代码和内部生成代码，并且会根据结果将相应的状态标志置 1。
  - 对于没有 CRC 的响应，不会进行 CRC 校验，并且不会将相应的状态标志置 1。

当 CPSM 有效（即不处于空闲状态）时，CPSMACT 位置 1。

图 539. 命令路径状态机 (CPSM)



- **空闲**：命令路径无效。当写入命令控制寄存器并且使能位 (CPSMEN) 置 1 后，CPSM 将激活 SDMMC_CK 时钟（因节能 PWRSAV 位而停止）并且
  - 在 WAITPEND = 0 且 BOOTEN = 0 时变为发送状态。
  - 在 WAITPEND = 1 时变为挂起状态。
  - 在 BOOTEN = 1 时变为启动状态。

- **发送**：发送命令并附加 CRC。
  - 当 CMDTRANS 位置 1 或 BOOTEN 位置 1、BOOTMODE 为备用启动状态且 DTDIR = 接收时，CPSM 数据使能信号将在命令结束时发送给 DPSM。
  - 当 CMDTRANS 位置 1 且 CMDSUSPEND 位为 0 时，中断周期将在命令结束时终止。
  - 当 CMDSTOP 位置 1 时，CPSM 中止信号将在命令结束时发送给 DPSM。
  - 如果不需要响应 (WAITRESP = 00)，CPSM 将变为空闲状态并生成 CMDSENT 标志。当 BOOTMODE = 1 且 BOOTEN = 0 时，CMDSENT 标志将在命令结束位后延迟 56 个周期生成，否则 CMDSENT 标志将在命令结束位后立即生成。
    不会修改 RESPCMDR 和 RESPxR 寄存器。
  - 如果需要命令响应（WAITRESP 不为 00），CPSM 将变为等待状态并启动响应超时。
- **等待**：命令路径等待响应。
  - 当 WAITINT 位为 0 时，命令定时器开始运行，CPSM 等待起始位。

  a) 如果在超时之前检测到起始位，CPSM 将变为接收状态。

  b) 如果在 CPSM 检测到响应起始位之前便已超时，则超时标志 (CTIMEOUT) 将置 1，CPSM 将变为空闲状态。
  不会修改 RESPCMDR 和 RESPxR 寄存器。
  - 当 WAITINT 位为 1 时，定时器将禁止并且 CPSM 将等待来自其中一个卡的中断请求（响应起始位）。

  a) 如果检测到起始位，CPSM 将变为接收状态。

  b) 向 WAITINT 写入 0（中断模式中止）时，主机将自行发送响应，并且检测到起始位时，CPSM 将变为接收状态。
- **接收**：将收到命令响应。响应可短可长，可带 CRC 可不带 CRC，具体取决于命令控制寄存器中的响应模式位 WAITRESP。将根据内部生成的 CRC 代码对接收到的 CRC 代码（存在时）进行校验。
  - 当 CMDSUSPEND 位置 1 且 SDIO 响应位 BS = 0（响应位 [39]）时，中断周期将在响应后启动。
    当 CMDSUSPEND 位清零或 CMDSUSPEND 位为 1 且 SDIO 响应位 BS = 1（响应位 [39]）时，不会有中断周期启动。
  - 当 CMDTRANS 位置 1、CMDSUSPEND 位置 1 且 SDIO 响应位 DF = 1（响应位 [32]）时，中断周期将在响应后终止。
  - 当 CRC 状态通过校验或不存在 CRC 时，CMDREND 标志将置 1，CPSM 变为空闲状态。
    RESPCMDR 和 RESPxR 寄存器通过接收到的响应进行更新。
    - 当 BOOTMODE = 1 且 BOOTEN = 0 时，CMDREND 标志将在响应结束位后延迟 56 个周期生成，否则 CMDREND 标志将在响应结束位后立即生成。
    - 当 CMDTRANS 位置 1 且 DTDIR = 发送时，CPSM 数据使能信号将在命令响应结束时发送给 DPSM。
  - 当 CRC 状态未通过校验时，CCRCFAIL 标志将置 1，CPSM 变为空闲状态。
    RESPCMDR 和 RESPxR 寄存器通过接收到的响应进行更新。

- **挂起**：CPSM 根据命令寄存器中的挂起 WAITPEND 位进入挂起状态。
  - 当 DATALENGTH =< 5 个字节时，CPSM 变为发送状态，并生成数据使能信号以启动与 CMD12 停止传输命令对齐的数据传输。
  - 当 DATALENGTH > 5 个字节时，将向 DPSM 发出 CPSM 数据使能信号以启动数据传输。CPSM 会等待来自 DPSM 的 sendCMD 信号，然后再变为发送状态。这样一来，便能够发送与数据对齐的 CMD12 停止传输命令。
  - 向 WAITPEND 写入 0 时，CPSM 将变为发送状态。
- **启动**：如果命令寄存器中的 BOOTEN 位置 1，则 CPSM 将进入启动状态，并且：
  - 当 BOOTMODE = 0 时，SDMMC_CMD 线驱动为低电平；当 CMDTRANS 位置 1 且 DTDIR = 接收时，CPSM 数据使能信号将发送到 DPSM，从而进入正常启动操作。启动过程结束后，将寄存器位 BOOTEN 清零以退出启动状态，此时会将 SDMMC_CMD 线驱动为高电平并将 CPSM 中止信号发送到 DPSM，然后再进入空闲状态。CMDSENT 标志在 SDMMC_CMD 线变为高电平后的 56 个周期后生成。
  - 当 BOOTMODE = 1 时，变为发送状态。此时可以发送 CMD0（启动）。将 BOOTEN 清零不起作用。

注： *CPSM 保持空闲状态至少八个 SDMMC_CK 周期以满足 N_{CC} 和 N_{RC} 时序限制。N_{CC} 是两个主机命令之间的最小延迟，N_{RC} 是主机命令和卡响应之间的最小延迟。*

注： *响应超时为固定值：64 个 SDMMC_CK 时钟周期。*

命令是用于启动操作的令牌。命令从主机发送到单个卡（编址命令），或发送到所有已连接的卡（MMC V3.31 或更低版本可以使用广播命令）。命令在 SDMMC_CMD 线上以串行方式传输。所有命令都为固定长度 48 位。*表 293* 中显示了 SD 存储卡、SDIO 卡和 MMC 卡的命令令牌的常规格式。

命令令牌数据取自 2 个寄存器，一个包含 32 位参数，另一个包含 6 位命令索引（发送到卡的 6 位）。

表 293. 命令令牌格式

位的位置	宽度	值	说明
47	1	0	起始位
46	1	1	传输位
[45:40]	6	x	命令索引
[39:8]	32	x	参数
[7:1]	7	x	CRC7
0	1	1	结束位

命令数据旁为命令类型 (WAITRESP) 位，用于控制命令路径状态机 (CPSM)。这些位还用于确定命令是否需要响应，响应是短（48 位）还是长（136 位）以及是否存在 CRC。

响应是一个令牌，它作为对先前接收命令的应答，从编址卡或从所有已连接的卡同步发送到主机。所有响应都是通过命令线 SDMMC_CMD 来发送。响应传输始终从对应于响应代码字的位字符串的左侧位开始。代码长度取决于响应类型。响应令牌 R1、R2、R3、R4、R5 和 R6 具有多种编码方案，具体取决于其内容。*表 294*、*表 295* 和 *表 296* 分别给出了 SD 存储卡、SDIO 卡和 MMC 卡的响应令牌的常规格式。

响应始终从起始位（始终为 0）开始，后跟用于指示传输方向的位 (card = 0)。下表中标有 x 的值表示变量项。大多数响应均受 CRC 保护。每个命令代码字由结束位（始终为 1）来终止。

响应令牌数据存储在 5 个寄存器中，其中 4 个寄存器包含 32 位卡状态、OCR 寄存器、参数或包括内部 CRC 的 127 位 CID 或 CSD 寄存器，另一个寄存器包含 6 位命令索引。

表 294. 带 CRC 的短响应令牌格式

位的位置	宽度	值	说明
47	1	0	起始位
46	1	0	传输位
[45:40]	6	x	命令索引（或保留为 111111）
[39:8]	32	x	参数
[7:1]	7	x	CRC7
0	1	1	结束位

表 295. 不带 CRC 的短响应令牌格式

位的位置	宽度	值	说明
47	1	0	起始位
46	1	0	传输位
[45:40]	6	x	命令索引（或保留为 111111）
[39:8]	32	x	参数
[7:1]	7	1111111	（保留为 1111111）
0	1	1	结束位

表 296. 带 CRC 的长响应令牌格式

位的位置	宽度	值	说明
135	1	0	起始位
134	1	0	传输位
[133:128]	6	111111	保留
[127:1]	127:8	x	CID 或 CSD 片段
	7:1	x	CRC7（包含在 CID 或 CSD 中）
0	1	1	结束位

命令/响应路径以半双工模式运行，因此既可发送命令，也可接收响应。如果 CPSM 未处于发送状态，则 SDMMC_CMD 输出处于高阻抗状态。在 SDMMC_CMD 上发送的数据将根据 NEGEDGE 寄存器位与 SDMMC_CK 进行同步，请参见*图 536*。

对于命令和带 CRC 的短响应，CRC 生成器计算 CRC 代码前面全部 40 个位的 CRC 校验和。这包括起始位、传输位、命令索引和命令参数（或卡状态）。

对于长响应，仅计算 R2 CID 或 CSD 的 120 个位的 CRC 校验和。请注意，CRC 计算中不使用起始位、发送位和 6 个保留位。

CRC 校验和是一个 7 位值：

CRC[6:0] = 余数 [(M(x) * x^7) / G(x)]

G(x) = $x^7 + x^3 + 1$

M(x) = (第 1 个位) * x^n + (第 2 个位) * x^{n-1}+... + (CRC 前的最后一位) * x^0

其中，n = 39 或 119。

CPSM 可在 CPSMEN 置 1 时发送一些特定命令来处理各种工作模式，请参见*表 297*。

表 297. 特定命令概述

VSWITCH	BOOTEN	BOOTMODE	CMDTRANS	WAITPEND	CMDSTOP	WAITINT	说明
1	x	x	x	x	x	x	启动电压切换序列。
0	1	x	x	x	x	x	开始正常启动。
0	1	1	x	x	x	x	开始备用启动。
0	0	1	x	x	x	x	停止备用启动。
0	0	0	1	x	x	x	发送包含相关数据传输的命令。
0	0	0	0	1	1	x	MMC 流数据传输，命令 (STOP_TRANSMISSION) 挂起至数据传输结束。
0	0	0	0	1	0	x	MMC 流数据传输，(STOP_TRANSMISSION) 以外的命令挂起至数据传输结束。
0	0	0	0	0	1	x	发送命令 (STOP_TRANSMISSION)，停止任何正在进行的数据传输。
0	0	0	0	0	0	1	进入 MMC 等待中断 (Wait-IRQ) 模式。
0	0	0	0	0	0	0	任何其他非特定命令。

命令/响应路径实现了*表 298* 所示的状态标志和相关清零位：

表 298. 命令路径状态标志

标志	说明
CMDSENT	在命令（无响应）结束时置 1。（CPSM 从发送状态变为空闲状态）
CMDREND	当 CRC 正常时，在命令响应结束时置 1。（CPSM 从接收状态变为空闲状态）
CCRCFAIL	当 CRC 失败时，在命令响应结束时置 1。（CPSM 从接收状态变为空闲状态）
CTIMEOUT	如果未在超时前接收到响应起始位，则在命令后置 1。（CPSM 从等待状态变为空闲状态）
CKSTOP	当 CRC 正常并且 SDMMC_CK 停止时，在电压切换 (VSWITCHEN = 1) 命令响应后置 1。（对 CPSM 无影响）
VSWEND	在 5 ms + 1 ms 的电压切换 (VSWITCH = 1) 超时后置 1。（对 CPSM 无影响）
CPSMACT	正在进行命令传输。（CPSM 不处于空闲状态）

命令路径错误处理如*表 302* 所示：

表 299. 命令路径错误处理

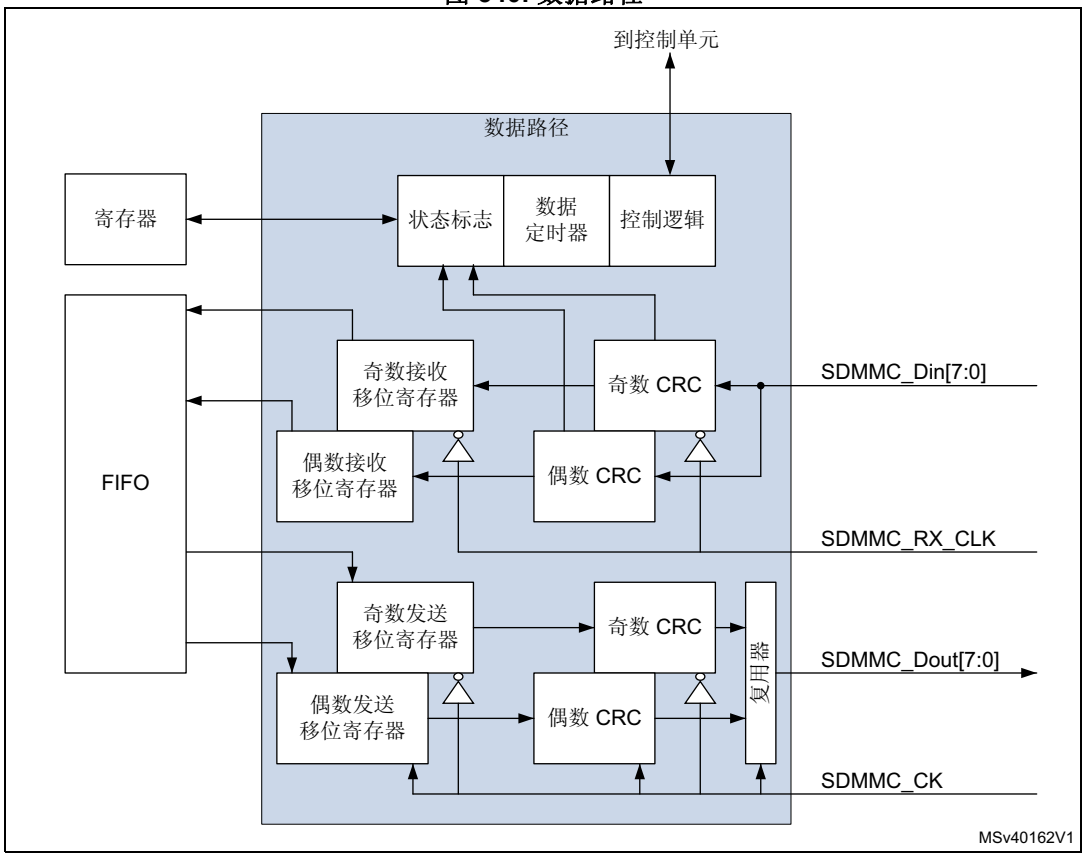
错误	CPSM 状态	原因	卡操作	主机操作	CPSM 操作
超时	等待	未及时出现起始位	未知	对卡进行复位或掉电再上电(1)	变为空闲状态
CRC 状态	接收	否定状态	忽略命令	重新发送命令(1)	变为空闲状态
		传输错误	接受命令	重新发送命令(1)	

1. 当 CMDTRANS 置 1 时，还需要发送 stop_transmission 命令才能使 DPSM 变为空闲状态。

数据路径

数据路径子单元负责通过 SDMMC_D[7:0] 线与卡相互传输数据。数据传输路径由 SDMMC_CK 提供时钟，并向卡发送数据。数据接收路径由 SDMMC_RX_CLK 提供时钟，并从卡接收数据。*图 540* 显示了数据路径框图。

图 540. 数据路径



卡数据总线宽度可以在时钟控制寄存器位 WIDBUS 中进行编程。支持的数据总线宽度模式如下：

- 如果未使能宽总线模式，则仅通过 SDMMC_D0 传输一位。
- 如果使能了 4 位宽度的总线模式，则通过 SDMMC_D[3:0] 传输 4 个数据位。
- 如果使能了 8 位宽度的总线模式，则通过 SDMMC_D[7:0] 传输 8 个数据位。

与数据总线宽度相似，数据采样模式可以在时钟控制寄存器位 DDR 中进行编程。支持的数据采样模式如下：

- 单倍数据速率信号传输 (SDR)，在时钟的上升沿驱动数据。
- 双倍数据速率信号传输 (DDR)，在时钟的两个边沿驱动数据。只有宽总线模式（4 位宽和 8 位宽）才支持 DDR 模式。

注： *数据采样模式仅适用于 SDMMC_D[7:0] 线。（不适用于 SDMMC_CMD 线。）*

在 DDR 模式下，根据以下规则在 SDMMC_CK 的两个边沿对数据进行采样，另请参见*图 541*和*图 542*：

- 在时钟的上升沿，对奇数字节进行采样。
- 在时钟的下降沿，对偶数字节进行采样。
- 数据有效负载大小始终为 2 字节的倍数。
- 每个数据线计算两个 CRC16。
  - 奇数位 CRC16，在时钟的下降沿驱动。
  - 偶数位 CRC16，在时钟的上升沿驱动。
- 起始位、结束位和空闲条件均为完整周期。
- CRC 状态/启动确认和繁忙信号均为完整周期，仅在时钟的上升沿进行采样。

在 DDR 模式下，SDMMC_CK 时钟分频比应 >= 2。

图 541. DDR 模式数据包时钟驱动

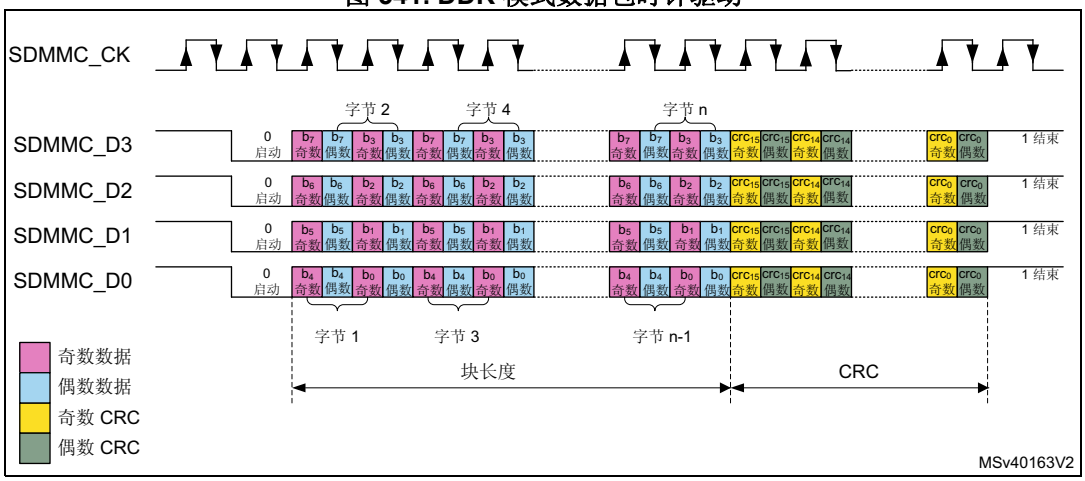
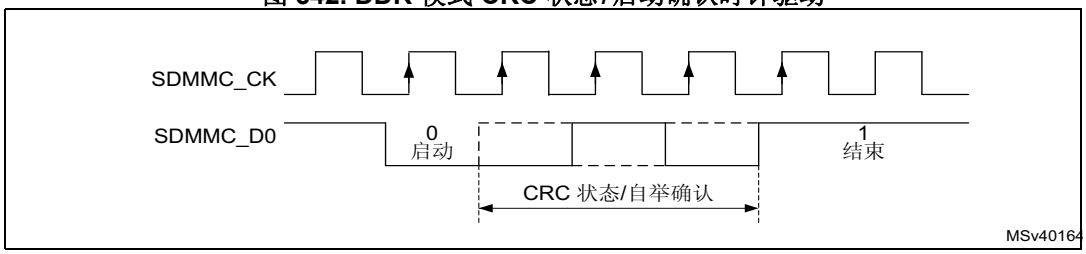


图 542. DDR 模式 CRC 状态/启动确认时钟驱动



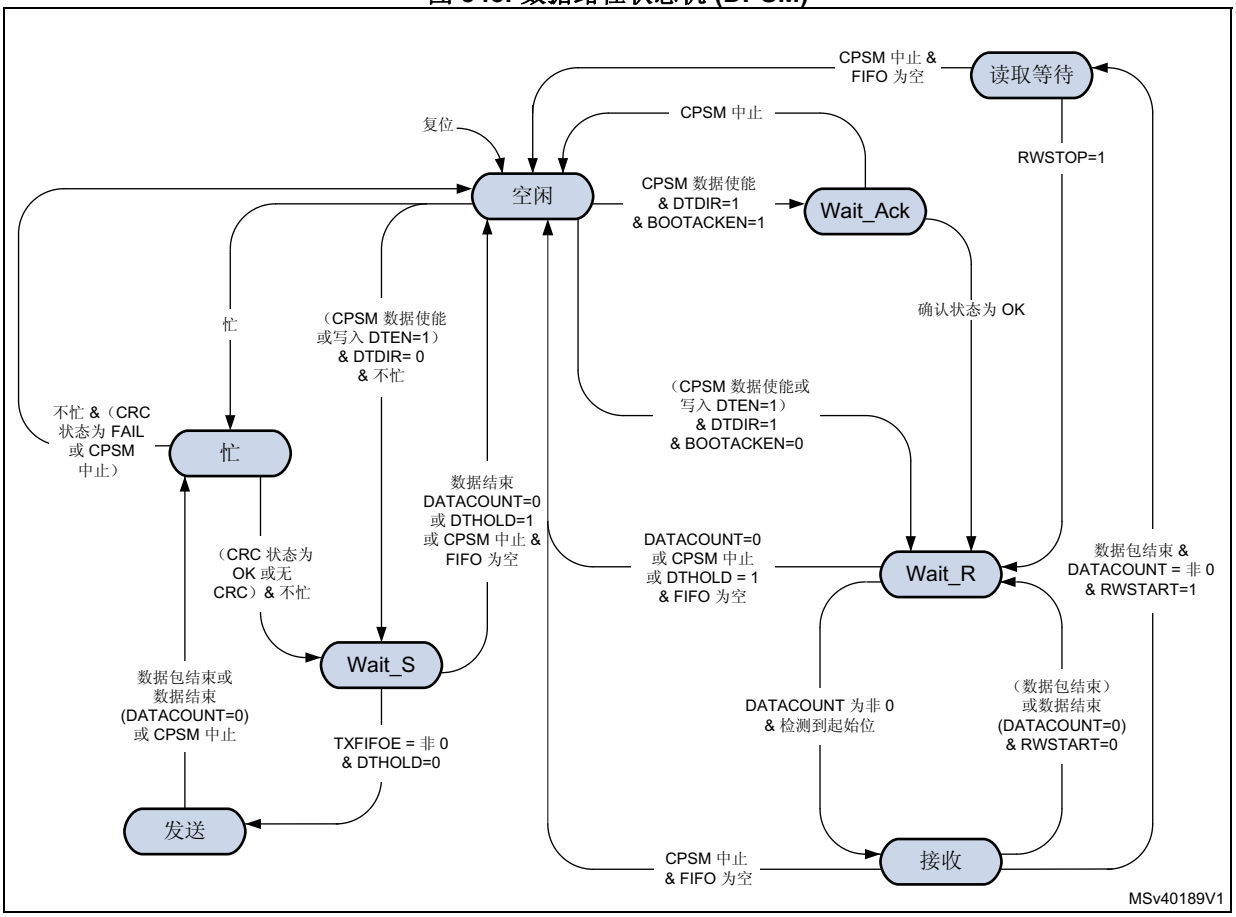
数据路径状态机 (DPSM)

根据传输方向（发送或接收），数据路径状态机 (DPSM) 将变为 Wait_S 或 Wait_R 状态（如果已使能）：

- 发送：DPSM 变为 Wait_S 状态。如果传输 FIFO 中有数据，则 DPSM 变为发送状态并且数据路径子单元开始向卡发送数据。
- 接收：DPSM 变为 Wait_R 状态并等待起始位。在 DPSM 收到起始位时，将变为接收状态，并且数据路径子单元开始从卡中接收数据。
  对于包含确认的启动操作，DPSM 变为 Wait_Ack 状态并等待启动确认，然后再变为 Wait_R 状态。

DPSM 以 SDMMC_CK 工作。DPSM 具有以下状态，如*图 543* 所示。当 DPSM 有效（即不处于空闲状态）时，DPSMACT 位置 1。

图 543. 数据路径状态机 (DPSM)



- **空闲**状态：数据路径无效，SDMMC_D[7:0] 输出取决于 PWRCTRL 设置。DPSM 将通过以下方式激活：发送将 CMDTRANS 位置 1 的命令或将 DTEN 位置 1，或者检测 SDMMC_D0 上的繁忙状态（即，在包含 R1b 响应的命令之后）。

  当不忙时，DPSM 将激活 SDMMC_CK 时钟（因节能 PWRSAV 位而停止时），并使用新的 (DATALENGTH) 值加载数据计数器，并且：

  – 当数据方向位 (DTDIR) 指示发送时，变为 Wait_S。

  – 当数据方向位 (DTDIR) 指示接收时，
  - 如果 BOOTACKEN 寄存器位清零，则变为 Wait_R。
  - 如果 BOOTACKEN 寄存器位置 1 且启动确认超时，则变为 Wait_Ack。

  当繁忙时，DPSM 将使 SDMMC_CK 时钟保持有效并变为繁忙状态。

注： *DTEN 不能用于启动与 SD、SDIO 和 eMMC 卡之间的数据传输。*

- **Wait_Ack** 状态：数据路径等待启动确认令牌。
  - 如果 DPSM 在超时之前收到无错确认，则它将变为 Wait_R 状态。
  - 当接收到确认以外的模式时，将生成确认状态错误，并且确认失败状态标志 (ACKFAIL) 将置 1。DPSM 保持 Wait_Ack 状态。
  - 如果在检测到起始位前便已超时 (ACKTIME)，则它会将超时状态标志 (ACKTIMEOUT) 置 1。DPSM 保持 Wait_Ack 状态。
  - 当 CPSM 中止信号置 1 时，它将变为空闲状态并将 DABORT 标志置 1。
- **Wait_R** 状态：如果数据计数器不为零并且数据未保存，则数据路径将等待 SDMMC_D[n:0] 上的起始位。如果数据计数器为零或数据已保存，则数据路径将等待 FIFO 变为空。
  - 在块模式下，如果在超时之前接收到起始位，DPSM 将变为接收状态，并使用 DBLOCKSIZE 加载数据块计数器。
  - 在 SDIO 多字节模式下，如果在超时之前接收到起始位，DPSM 将变为接收状态，并使用 DATALENGTH 加载数据块计数器。
  - 在流模式下，如果在超时之前接收到起始位，DPSM 将变为接收状态，并使用 DATALENGTH 加载数据计数器。
  - 如果数据计数器 (DATACOUNT) 等于零（数据结束），当接收 FIFO 为空并且 DATAEND 标志置 1 时，DPSM 将变为空闲状态。
  - 如果在检测到起始位前便已超时 (DATATIME)，则它会将超时状态标志 (DTIMEOUT) 置 1 且 DPSM 会保持 WAIT_R 状态。
  - 如果 CPSM 中止信号置 1：
    如果 DATACOUNT > 0，则当 FIFO 为空时，DPSM 变为空闲状态，当 IDMAEN = 0 时，使用 FIFORST 复位，并将 DABORT 标志置 1。
    如果 DATACOUNT 为零，则继续正常操作，由于传输已经正常完成，将不会有 DABORT 标志。
  - 如果 DTHOLD 位置 1：
    - 如果 DATACOUNT > 0，则当接收 FIFO 为空时，DPSM 变为空闲状态，当 IDMAEN = 0 时，使用 FIFORST 复位，并发出 DHOLD 标志。保持期间禁止超时。如果在保持期间接收到 CPSM 中止信号，则传输中止。
    - 当 DATACOUNT = 0 时，传输正常完成，不会有 DHOLD 标志。
  - 如果 DPSM 通过 DTEN 启动，在发生错误 (DTIMEOUT) 之后，当 FIFO 为空时，DPSM 将变为空闲状态，当 IDMAEN = 0 时，将使用 FIFORST 复位。
- **读取等待**状态：数据路径对总线进行读取等待。
  - 当读取等待停止位 (RWSTOP) 置 1 时，DPSM 将变为 Wait_R 状态，并且启动接收超时。
  - 如果 CPSM 中止信号置 1，则等待 FIFO 变为空，并且当 IDMAEN = 0 时，使用 FIFORST 复位，然后变为空闲状态并将 DABORT 标志置 1。

- **接收**状态：数据路径从卡接收串行数据。将数据以字节为单位打包，并写入到数据 FIFO。根据数据控制寄存器中选择的传输模式 (DTMODE)，数据传输模式可以是块或流：
  - 在块模式中，当接收到的数据字节数达到数据块大小 (DBLOCKSIZE) 时，DPSM 将等待直至其收到 CRC 代码。
  - 在 SDIO 多字节模式中，当接收到的数据字节数达到数据块大小 (DATALENGTH) 时，DPSM 将等待直至其收到 CRC 代码。

  a) 如果收到的 CRC 代码与内部生成的 CRC 代码相匹配，则
  - 当 RWSTART = 0 且启动接收超时时，DPSM 将变为 Wait_R 状态。
  - 当 RWSTART = 1 且 DATACOUNT > 0 时，DPSM 将变为读取等待状态，并会生成 DBCKEND 标志。

  b) 如果收到的 CRC 代码与内部生成的 CRC 代码不匹配，则将阻止任何后续的数据接收操作。
  - 当并未接收到所有数据 (DATACOUNT > 0) 时，CRC 失败状态标志 (DCRCFAIL) 将置 1，DPSM 将保持接收状态。
  - 当接收到所有数据 (DATACOUNT = 0) 时，将等待 FIFO 变为空，之后 CRC 失败状态标志 (DCRCFAIL) 将置 1，DPSM 将变为空闲状态。

  - 在流模式中，DPSM 在数据计数器 DATACOUNT > 0 时接收数据。当计数器为零时，会将移位寄存器中其余数据写入到数据 FIFO，并且 DPSM 将变为 Wait_R 状态。
  - 如果发生 FIFO 溢出错误，则 DPSM 将 FIFO 溢出错误标志 (RXOVERR) 置 1 并且会阻止任何后续的数据接收操作。DPSM 保持接收状态。
  - 收到 CPSM_Abort 信号时：
    - 如果 DATACOUNT = 0 时在数据的最后 2 位之前接收到 CPSM_Abort 信号，则传输中止。移位寄存器中的剩余数据将写入数据 FIFO，等待 FIFO 变为空，当 IDMAEN = 0 时，使用 FIFORST 复位，DPSM 随后变为空闲状态且 DABORT 标志置 1。
    - 如果 DATACOUNT = 0 时在传输的最后 2 位期间或之后接收到 CPSM_Abort 信号，则传输正常完成。DPSM 保持接收状态，不生成 DABORT 标志。
  - 如果 DPSM 通过 DTEN 启动，在发生错误（DATACOUNT > 0 时为 DCRCFAIL，否则为 RXOVERR）之后，当 FIFO 为空时，DPSM 将变为空闲状态，当 IDMAEN = 0 时，将使用 FIFORST 复位。

- **Wait_S** 状态：数据路径等待 FIFO 中有数据可用。
  - 如果数据计数器 DATACOUNT > 0，则等待直到数据 FIFO 为空标志 (TXFIFOE) 置为无效并且 DTHOLD 未置 1，然后变为发送状态。
  - 如果数据计数器 (DATACOUNT) = 0，DPSM 将变为空闲状态并生成 DATAEND 标志。
  - 当 DTHOLD 置 1 并且 DATACOUNT > 0 时
    - 如果使能 IDMA，FIFO 将被清空，之后 DPSM 将变为空闲状态并发出 DHOLD 标志。（DBCKEND 标志也将置 1，并且可以与 DHOLD 标志同时清零。）
    - 如果禁止 IDMA，则会生成 DBCKEND 标志。等待 FIFO 使用 FIFORST 复位，之后 DPSM 将变为空闲状态并发出 DHOLD 标志。
  - 当 DTHOLD 置 1 且 DATACOUNT = 0 时，传输正常完成。
  - 当收到 CPSM 中止信号时
    - 如果 DATACOUNT = 0 时在数据的最后 2 位之前接收到 CPSM_Abort 信号，则传输将中止，等待 FIFO 变为空并且当 IDMAEN = 0 时，使用 FIFORST 复位，之后 DPSM 变为空闲状态并将 DABORT 标志置 1。
    - 如果 DATACOUNT = 0 时在传输的最后 2 位期间或之后接收到 CPSM_Abort 信号，则继续正常操作，并且由于传输正常完成，将没有 DABORT 标志。

注： *DPSM 在 Wait_S 状态下需要至少保持两个时钟周期以满足 N_{WR} 计时要求，N_{WR} 是从收到卡响应到开始从主机传输数据这一期间的时钟周期数。*

- **发送**状态：DPSM 开始向卡发送数据。根据数据控制寄存器中的传输模式，数据传输模式可以是块、SDIO 多字节或流：
  - 在块模式中，当发送的数据字节数达到数据块大小 (DBLOCKSIZE) 时，DPSM 将发送一个内部生成的 CRC 代码和结束位，然后变为繁忙状态并启动发送超时。
  - 在 SDIO 多字节模式下，当发送的数据字节数达到数据块大小 (DATALENGTH) 时，DPSM 将发送一个内部生成的 CRC 代码和结束位，然后变为繁忙状态并启动发送超时。
  - 在流模式中，DPSM 在数据计数器 DATACOUNT > 0 时向卡发送数据。当数据计数器达到 0 时，DPSM 变为繁忙状态并启动发送超时。
    在根据 DATACOUNT 发送最后一个流字节之前，DPSM 会触发 sendCMD 信号。CPSM 使用该信号来发送任何挂起的命令。（即 CMD12 停止传输命令）
  - 如果发生 FIFO 下溢错误，DPSM 会将 FIFO 下溢错误标志 (TXUNDERR) 置 1。DPSM 保持发送状态。
  - 当收到 CPSM 中止信号时
    - 如果 DATACOUNT = 0 时在传输的最后 2 位之前接收到 CPSM_Abort 信号，则传输中止。DPSM 将发送最后一个数据位，后跟一个结束位。FIFO 将被禁止/清空，并且 DPSM 将变为繁忙状态，等待至不忙状态后将 DABORT 标志置 1。
    - 如果 DATACOUNT = 0 时在传输的最后 2 位期间或之后接收到 CPSM_Abort 信号，则传输正常完成，将不会有 DABORT 标志。
- **繁忙**状态：DPSM 等待 CRC 状态令牌（需要时），并等待不忙信号：
  - 如果需要 CRC 状态令牌并且该令牌指示“无错误传输”或者不需要 CRC 时：
    - 如果 SDMMC_D0 未处于低电平（卡未处于繁忙状态），则它变为 Wait_S 状态。
    - 当卡处于繁忙状态时，SDMMC_D0 处于低电平，它将保持繁忙状态。
  - 如果需要 CRC 状态令牌并且该令牌指示“错误传输”。
    - 当并未发送所有数据 (DATACOUNT > 0) 时。DPSM 等待至不忙状态后将 CRC 失败状态标志 (DCRCFAIL) 置 1。FIFO 将被禁止/清空并且 DPSM 保持繁忙状态。
    - 当已发送所有数据 (DATACOUNT = 0) 时。DPSM 等待至不忙状态后将 CRC 失败状态标志 (DCRCFAIL) 置 1，并且 DPSM 将变为空闲状态。
  - 如果 DPSM 处于繁忙状态时发生 CRC 状态 (Ncrc) 超时，则它会将数据超时标志 (DTIMEOUT) 置 1 并保持繁忙状态。
  - 如果 DPSM 处于繁忙状态时发生繁忙超时，则它会将数据超时标志 (DTIMEOUT) 置 1 并保持繁忙状态。
  - 当在繁忙状态下接收到 CPSM 中止信号时：
    - 如果 DATACOUNT > 0 时在 CRC 响应的最后 2 位之前接收到 CPSM_Abort 信号，则数据传输中止。DPSM 等待至不忙状态且 FIFO 被禁止/清空后变为空闲状态，并将 DABORT 标志置 1。
    - 如果 DATACOUNT = 0 时在 CRC 响应的最后 2 位期间或之后接收到 CPSM_Abort 信号，或者当不需要 CRC、DATACOUNT = 0 且没有 DTIMEOUT 错误时，DPSM 将保持繁忙状态，并且由于传输可正常完成，将不会生成 DABORT 标志。
    - 如果在已出现 DTIMEOUT 错误时接收到 CPSM_Abort 信号，则 DPSM 将等待至不忙状态且 FIFO 禁止/清空后变为空闲状态，并将 DABORT 标志置 1。

- 当由于在发送状态下中止而进入繁忙状态时，DPSM 将等待至不忙状态后变为空闲状态，并将 DABORT 标志置 1。
- 如果 DPSM 通过 DTEN 启动，在发生错误 (DATACOUNT > 0 时为 DCRCFAIL，否则为 DTIMEOUT) 之后，当 FIFO 复位时，DPSM 将变为空闲状态。
- 当 DPSM 因 SDMMC_D0 上繁忙而启动时，它将等待至不忙状态后将繁忙结束状态标志 (BUSYD0END) 置 1，并且 DPSM 将变为空闲状态。

当 DPSM 在数据块结束位、数据读取命令结束位或 R1b 响应后 2 个周期后进入 Wait_R 或繁忙状态时，将使能数据定时器 (DATATIME)，并会生成数据超时错误 (DTIMEOUT)：

- 发送数据时，发生超时
  - 当需要 CRC 状态并且在 8 个 SDMMC_CK 周期内未接收到起始位时，DTIMEOUT 标志将置 1。
  - 当繁忙状态的时长超过编程的超时周期时，DTIMEOUT 标志将置 1。
- 接收数据时，发生超时
  - 当仍然有数据要接收、DATACOUNT > 0 并且未在编程的超时周期前接收到起始位时，DTIMEOUT 标志将置 1。
- 在 R1b 响应后，发生超时
  - 当繁忙状态的时长超过编程的超时周期时，DTIMEOUT 标志将置 1。

DATATIME = 0 时，

- 接收期间，起始位应在数据块结束位或数据读取命令结束位后的 2 个周期后出现。
- 发送期间，繁忙状态在 CRC 令牌结束位或流数据结束位后超时 2 个周期。
- 在 R1b 响应之后，繁忙状态在响应结束位后超时 2 个周期。

可以将数据从卡传输到主机（发送），或从主机传输到卡（接收）。数据通过 SDMMC_Dn 数据线传输，并且存储在 FIFO 中。

表 300. 数据令牌格式

说明	起始位	数据(1)	CRC16	结束位	DTMODE
块数据	0	（DBLOCKSIZE 和 DATALENGTH）	是	1	00
SDIO 多字节	0	(DATALENGTH)	是	1	01
MMC 数据流	0	(DATALENGTH)	否	1	10

1. 要传输的数据总量由 DATALENGTH 给出。对于块数据而言，每个块中的数据量由 DBLOCKSIZE 给出。

数据令牌格式通过相应的寄存器位 DTMODE 选择。

数据路径实现了*表 301* 所示的状态标志和相关清零位：

表 301. 数据路径状态标志和清零位

标志		说明
DATAEND	TX	如果 CRC 正常且繁忙状态已结束，则在完整数据传输结束时置 1。(DATACOUNT = 0)。（DPSM 从 WAIT_S 状态变为空闲状态）
	RX	如果 CRC 正常且所有数据读取完毕，则在完整数据传输结束时置 1（DATACOUNT = 0 且 FIFO 为空）。（DPSM 从 WAIT_R 状态变为空闲状态）
	BOOT	
DCRCFAIL	TX	如果 CRC 失败且繁忙状态已结束，则在 CRC 结束时置 1。（当仍然有数据要发送时，DPSM 保持繁忙状态并等待中止）（当所有数据发送完毕或 DPSM 已通过 DTEN 启动时，DPSM 将从繁忙状态变为空闲状态）
	RX	如果 CRC 失败且 FIFO 为空，则在 CRC 结束时置 1。（当仍然有数据要接收时，DPSM 保持接收状态并等待中止）（当所有数据接收完毕或 DPSM 已通过 DTEN 启动时，DPSM 将从接收状态变为空闲状态）
	BOOT	
ACKFAIL	BOOT	如果失败，则在 BOOT ACK 结束时置 1。（DPSM 保持 Wait_Ack 状态并等待中止）
DTIMEOUT	CMD R1b	如果未在超时前接收到繁忙结束信号，则在命令响应后置 1。（DPSM 保持繁忙状态并等待中止）
	TX	当未在 Ncrc 内接收到 CRC 令牌起始位或未在超时前接收到繁忙结束信号时置 1。（DPSM 保持繁忙状态并等待中止）（当 DPSM 已通过 DTEN 启动时，将变为空闲状态）注意：在繁忙超时之前，如果 CRC 失败，则还会将 DCRCFAIL 标志置 1。
	RX	如果未在超时前接收到起始位，则置 1。（DPSM 保持 WAIT_R 状态并等待中止）（当 DPSM 已通过 DTEN 启动时，将变为空闲状态）
	BOOT	
ACKTIMEOUT	BOOT	如果未在超时前接收到起始位，则置 1。（DPSM 保持 Wait_Ack 状态并等待中止）
DBCKEND	TX	当 DTHOLD = 1 时：在数据传输未完成的情况下 (DATACOUNT > 0)，如果 CRC 正常且繁忙状态已结束，则在数据块传输结束时置 1。（DPSM 从 WAIT_S 状态变为空闲状态）
	RX	当 RWSTART = 1 时：在数据传输未完成的情况下 (DATACOUNT > 0)，如果 CRC 正常，则在数据块传输结束时置 1。（DPSM 从接收状态变为读取等待状态）
	BOOT	
DHOLD	TX	当 DTHOLD = 1 时：在数据传输未完成的情况下 (DATACOUNT > 0)，如果 CRC 正常且繁忙状态已结束，则在数据块传输结束时置 1。（DPSM 从 WAIT_S 状态变为空闲状态）
	RX	当 DTHOLD = 1 时：在数据传输未完成的情况下 (DATACOUNT > 0)，如果 CRC 正常且所有数据读取完毕（FIFO 为空），则在数据块传输结束时置 1。（DPSM 从 WAIT_R 状态变为空闲状态）
DABORT	CMD R1b	当 CPSM 已发送中止事件且繁忙状态结束时。（DPSM 从繁忙状态变为空闲状态）
	TX	当 CPSM 在传输的最后 2 位之前已发送中止事件时。（DPSM 从任何状态变为空闲状态）
	RX	
	BOOT	
BUSYD0END	CMD R1b	如果繁忙状态在超时前结束，则在命令响应后置 1。（DPSM 从繁忙状态变为空闲状态）
DPSMACT		正在进行数据传输。（DPSM 不处于空闲状态）

数据路径错误处理如*表 302* 所示：

表 302. 数据路径错误处理

误差	DPSM 状态	原因	卡操作	主机操作	DPSM 操作
超时	Wait_Ack	未及时确认	未知	对卡进行掉电再上电	保持 Wait_Ack 状态（通过 RCC.SDMMCxRST 寄存器位复位 SDMMC）
	Wait_R	未及时出现起始位	未知	停止数据接收 发送停止传输命令	出现 CPSM_Abort 时，变为空闲状态
			未知	停止启动过程	
	繁忙	繁忙状态太久（由于数据传输）	未知	停止数据接收 发送停止传输命令	
		繁忙状态太久（由于 R1b）	未知	发送复位命令	
CRC	接收	传输错误	发送后续数据	停止数据接收 发送停止传输命令	出现 CPSM_Abort 时，变为空闲状态
CRC 状态	繁忙	否定状态	忽略后续数据	停止数据发送 发送停止传输命令	出现 CPSM_Abort 时，变为空闲状态
		传输错误	等待后续数据		
确认状态	Wait_Ack	传输错误	发送启动数据	停止启动过程	出现 CPSM_Abort 时，变为空闲状态
上溢	接收	FIFO 已满	发送后续数据	停止数据接收 发送停止传输命令	出现 CPSM_Abort 时，变为空闲状态
下溢	发送	FIFO 为空	接收后续数据	停止数据发送 发送停止传输命令	出现 CPSM_Abort 时，变为空闲状态

数据 FIFO

数据 FIFO（先进先出）子单元包含发送和接收数据缓冲器。单个 FIFO 用于由 DTDIR 位选择的发送或接收操作。FIFO 包含一个宽度为 32 位且深度为 16 字的数据缓冲器和控制逻辑。由于数据 FIFO 在 AHB 时钟域 (HCLK) 中运行，因此来自 SDMMC 时钟域 (SDMMC_CK/SDMMC_RX_CLK) 的子单元中的所有信号都将重新同步。

FIFO 可进入以下状态之一：

- 向卡发送数据时，发送 FIFO 引用发送逻辑和数据缓冲器。(DTDIR = 0)
- 从卡接收数据时，接收 FIFO 引用接收逻辑和数据缓冲器。(DTDIR = 1)

当来自 FIFO 的 SDMMC 数据传输正确完成时，将由数据路径子单元驱动 DATAEND 标志来指示。当来自 FIFO 的 SDMMC 数据传输出错（中止）时，将由数据路径子单元驱动其中一个错误标志（DCRCFAIL、DTIMEOUT 和 DABORT）或由 FIFO 控制驱动其中一个 FIFO 错误标志（TXUNDERR 和 RXOVERR）来指示。

可以通过以下方式访问数据 FIFO，请参见*表 303*。

表 303. 数据 FIFO 访问

数据 FIFO 访问	IDMAEN
在固件的控制下通过 AHB 从接口访问	0
在 IDMA 的控制下通过 AHB 主接口访问	1

发送 FIFO：

当 DPSM 已激活 (DPSMACT = 1) 时，数据可以写入发送 FIFO。

当 IDMAEN = 1 时，FIFO 由 IDMA 完全处理。

当 IDMAEN = 0 时，FIFO 由固件通过 AHB 从接口控制。可以通过连续地址来访问发送 FIFO。传输 FIFO 包含一个数据输出寄存器，其中存储了读取指针所指向的数据字。在数据路径子单元已加载了其移位寄存器之后，该子单元会将读取指针递增并将新数据推出。发送 FIFO 的处理方式如下：

1. 将数据长度写入 DATALENGTH，将块长度写入 DBLOCKSIZE。
   – 对于块数据传输 (DTMODE = 0)，DATALENGTH 应为 DBLOCKSIZE 的倍数。
2. 将 SDMMC 设置为发送模式 (DTDIR = 0)。
   – 将 FIFO 配置为发送模式。
3. 使能数据传输。
   – 通过将 CMDTRANS 位置 1 从 CPSM 发送一条命令。
   – 通过将 DTEN 位置 1。
4. 当 (DPSMACT = 1) 将数据写入 FIFO 时。
   – DPSM 将保持 Wait_S 状态，直到 FIFO 已满 (TXFIFOF = 1) 或者达到 DATALENGTH 指示的数字。
   – 只要 FIFO 不为空，SDMMC 就会开始发送数据。
5. 当 FIFO 为半空（TXFIFOHE 标志）时，将数据写入 FIFO，直到 FIFO 已满 (TXFIFOF = 1)，或者写入最后一个数据。
6. 当写入最后一个数据后，等待数据结束（DATAEND 标志）。
   – SDMMC 已发送完所有数据且 DPSM 处于禁止状态 (DPSMACT = 0)。

如果在 IDMAEN = 0 时发生数据传输错误或保持传输，固件将停止写入 FIFO，并使用 FIFORST 寄存器位清空和复位 FIFO。

表 304 中列出了发送 FIFO 状态标志。

表 304. 传输 FIFO 状态标志

标志	说明
TXFIFOF	当所有发送 FIFO 字都包含有效数据时，设置为高电平。
TXFIFOE	当传输 FIFO 不包含有效数据时，设置为高电平。
TXFIFOHE	当一半或更多个发送 FIFO 字为空时，设置为高电平。
TXUNDERR	发生下溢错误时，设置为高电平。通过写入到 SDMMC 清零寄存器可将该位清零。

接收 FIFO：

当 DPSM 已激活 (DPSMACT = 1) 时，可以从接收 FIFO 读取数据。

当 IDMAEN = 1 时，FIFO 由 IDMA 完全处理。

当 IDMAEN = 0 时，FIFO 由固件通过 AHB 从接口控制。当数据路径子单元接收到数据字时，它会驱动写入数据总线上的数据。在写入操作完成后，写入指针递增。在读取端，读取指针的当前值指向的 FIFO 字内容被推入读取数据总线中。可以通过连续地址来访问接收 FIFO。接收 FIFO 的处理方式如下：

1. 将数据长度写入 DATALENGTH，将块长度写入 DBLOCKSIZE。
   – 对于块数据传输 (DTMODE = 0)，DATALENGTH 应为 DBLOCKSIZE 的倍数。
2. 将 SDMMC 设置为接收模式（DTDIR = 1）。
   – 将 FIFO 配置为接收模式。
3. 使能 DPSM 传输。
   – 通过将 CMDTRANS 位置 1 从 CPSM 发送一条命令。
   – 或者通过将 DTEN 位置 1。
4. 当 (DPSMACT = 1) FIFO 准备好接收数据时。
   – DPSM 会将接收到的数据写入 FIFO。
5. 当 FIFO 为半满（RXFIFOHF 标志）时，从 FIFO 读取数据，直到 FIFO 为空 (RXFIFOE = 1)。
6. 如果在数据结束（DATAEND 标志）时接收到最后一个数据，则从 FIFO 读取数据，直到 FIFO 为空 (RXFIFOE = 1)。
   – SDMMC 已接收完所有数据且 DPSM 处于禁止状态 (DPSMACT = 0)。

如果在 IDMAEN = 0 时保持数据传输，固件应读取剩余数据，直到 FIFO 为空，并使用 FIFORST 寄存器位复位 FIFO。这将导致 DPSM 进入空闲状态 (DPSMACT = 0)。

如果在 IDMAEN = 0 时出现数据传输错误，固件应停止读取 FIFO，并使用 FIFORST 寄存器位清空和复位 FIFO。这将导致 DPSM 进入空闲状态 (DPSMACT = 0)。

表 305 中列出了接收 FIFO 状态标志。

表 305. 接收 FIFO 状态标志

标志	说明
RXFIFOF	当所有接收 FIFO 字都包含有效数据时，设置为高电平。
RXFIFOE	当接收 FIFO 不包含有效数据时，设置为高电平。
RXFIFOHF	当一半或更多个接收 FIFO 字包含有效数据时，设置为高电平。
RXOVERR	发生溢出错误时，设置为高电平。通过写入到 SDMMC 清零寄存器可将该位清零。

CLKMUX 单元

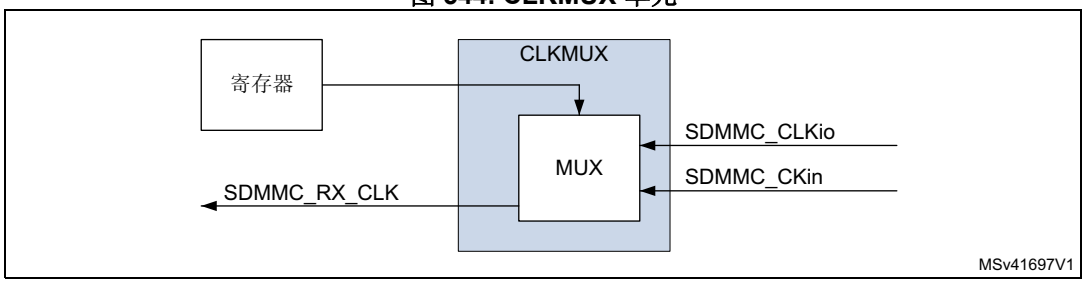
CLKMUX 选择用于与接收到的数据和命令响应搭配使用的 SDMMC_RX_CLK 时钟源。可以通过时钟控制寄存器位 SELCLKRX 从以下几项中选择接收数据时钟源：

- SDMMC_CLKio 总线主设备主反馈时钟。
- SDMMC_CKin 外部总线反馈时钟。

当没有外部驱动器且使用 DS 和 HS 时，选择 SDMMC_CLKio。

当有外部驱动器且使用 SDR12、SDR25、SDR50 和 DDR50 时，选择 SDMMC_CKin。

图 544. CLKMUX 单元



当 CPSM 和 DPSM 处于空闲状态时，应更改 SDMMC_RX_CLK 源。

48.4.5 SDMMC AHB 从接口

AHB 从接口生成中断请求，并且访问 SDMMC 适配器寄存器和数据 FIFO。该接口由数据路径、寄存器解码器和中断逻辑构成。

SDMMC FIFO

FIFO 访问仅限于字访问：

- 在发送 FIFO 模式下
  - 数据按字（32 位）写入 FIFO，直到 DATALENGTH 所指示的所有数据均传输完毕。当 DATALENGTH 不是 4 的倍数时，最后剩余的数据（1、2 或 3 字节）通过字传输的方式写入。
- 在接收 FIFO 模式下
  - 数据按字（32 位）从 FIFO 读取，直到 DATALENGTH 所指示的所有数据均传输完毕。当 DATALENGTH 不是 4 的倍数时，最后剩余的数据（1、2 或 3 字节）通过以 0 值字节填充的字传输读取。

当通过半字或字节访问的方式访问 FIFO 时，会生成 AHB 总线故障。

SDMMC 中断

中断逻辑生成一个中断请求信号，当至少其中一个非屏蔽状态标志有效时，便发出此信号。提供了一个屏蔽寄存器，用于选择将生成中断的条件。如果对应的屏蔽标志置 1，则状态标志将生成中断请求。一些状态标志需要在清零寄存器中隐式清零。

48.4.6 SDMMC AHB 主接口

AHB 主接口用于在存储器和 FIFO 之间使用 SDMMC IDMA 传输数据。

SDMMC IDMA

直接存储器访问 (DMA) 用于在 SDMMC FIFO 与存储器之间提供高速传输。AHB 主设备可优化系统总线的带宽。SDMMC 内部 DMA (IDMA) 提供一个用于发送或接收的通道。

IDMA 通过 IDMAEN 位使能，并支持 8 节拍的突发传输。

- 在发送突发传输模式下：
  - 只要 FIFO 中的突发传输次数为空，就会以突发形式从存储器中获取数据，直到 DATALENGTH 所指示的所有数据均已传输完毕。如果 DATALENGTH 不是突发大小的倍数，则不足突发大小的剩余数据将使用单次传输模式进行传输。
    当 DATALENGTH 不是 4 的倍数时，最后剩余的数据（1、2 或 3 字节）通过字传输的方式获取。
- 在接收突发传输模式下：
  - 只要 FIFO 中包含突发传输次数，就会以突发形式将数据存储到存储器中，直到 DATALENGTH 所指示的所有数据均已传输完毕。
    如果 DATALENGTH 不是突发传输的倍数，则不足突发大小的剩余数据将使用单次传输模式进行传输。
    当 DATALENGTH 不是 4 的倍数时，最后剩余的数据（1、2 或 3 字节）通过半字或字节传输的方式存储。

此外，IDMA 还提供有两种通道配置（通过位 IDMABMODE 选择）：

- 单缓冲通道
- 双缓冲通道

在单缓冲区配置中，存储器端的数据从基址 IDMABASE0 开始以线性方式进行访问。当 IDMA 完成所有数据的传输并且 DPSM 也完成传输后，DATAEND 标志将置 1。

在双缓冲区配置中，存储器端的数据从 2 个缓冲区进行访问，一个从基址 IDMABASE0 开始，另一个从基址 IDMABASE1 开始。这样一来，当 IDMA 访问其中一个存储器缓冲区时，固件可以处理另一个存储器缓冲区。存储器缓冲区的大小由 IDMABSIZE 定义。缓冲区大小应为突发大小的倍数。使能通道时，可以即时更新缓冲区的基址，适用的规则如下：

- 当 IDMABACT 位为“0”时，IDMA 硬件使用 IDMABASE0 来访问存储器。当尝试通过固件写入该寄存器时，写操作将被丢弃，IDMABASE0 数据将不会发生变化。允许固件写入 IDMABASE1。
- 当 IDMABACT 位为“1”时，IDMA 硬件使用 IDMABASE1 来访问存储器。当尝试通过固件写入该寄存器时，写操作将被丢弃，IDMABASE1 数据将不会发生变化。允许固件写入 IDMABASE0。

当 IDMA 完成其中一个缓冲区的数据传输时，缓冲区传输完成标志 (IDMABTC) 将置 1 且 IDMABACT 位将发生翻转，之后 IDMA 继续从另一个缓冲区传输数据。当 IDMA 完成所有数据的传输并且 DPSM 也完成传输后，DATAEND 标志将置 1。

IDMABASEn 地址应按字对齐。

错误管理

当对保留的地址空间执行读写操作时，将出现 IDMA 传输错误。在出现 IDMA 传输错误时，后续的 IDMA 传输将禁止，IDMATE 标志将置 1。IDMATE 标志的行为取决于 SDMMC 传输期间出现 IDMA 传输错误的时间：

- 在出现任何 SDMMC 传输错误（TXUNDERR、RXOVERR、DCRCFAIL 或 DTIMEOUT）之前检测到 IDMA 传输错误：
  - IDMATE 标志与 SDMMC 传输错误标志同时置 1。
  - 生成 TXUNDERR、RXOVERR、DCRCFAIL 或 DTIMEOUT 中断。
- 在 STOP_TRANSNMISSION 命令期间检测到 IDMA 传输错误：
  - IDMATE 标志与 DABORT 标志同时置 1。
  - 生成 DABORT 中断。
- 在 SDMMC 传输（HOLD 或 DATAEND）结束时检测到 IDMA 传输错误。
  - IDMATE 标志在 SDMMC 传输结束时置 1。
  - 生成 SDMMC 传输结束中断且 HOLD 或 DATAEND 标志置 1。

IDMATE 将在出现其他 SDMMC 传输中断（TXUNDERR、RXOVERR、DCRCFAIL、DTIMEOUT、DABORT、HOLD 或 DATAEND）时生成。

48.4.7 AHB 和 SDMMC_CK 时钟的关系

AHB 带宽应至少为 SDMMC 总线带宽的 3 倍，即对于 SDR50 4 位模式 (50 MB/s)，最小 HCLK 频率为 37.5 MHz (150 MB/s)。

表 306. AHB 和 SDMMC_CK 时钟频率的关系

SDMMC 总线模式	SDMMC 总线带宽	最大 SDMMC_CK [MHz]	最小 AHB 时钟 [MHz]
MMC DS	8	26	19.5
MMC HS	8	52	39
MMC DDR52	8	52	78
SD DS / SDR12	4	25	9.4
SD HS / SDR25	4	50	18.8
SD DDR50	4	50	37.5
SD SDR50	4	100	37.5

48.5 卡功能说明

48.5.1 SD I/O 模式

以下功能是专用于 SDMMC 的操作：

- SDIO 中断
- SDIO 挂起/恢复操作（写入和读取挂起）
- 通过停止时钟执行的 SDIO 读取等待操作
- 通过触发 SDMMC_D2 执行的 SDIO 读取等待操作

表 307. SDIO 特殊操作控制

操作模式	SDIOEN	RWMOD	RWSTOP	RWSTART	DTDIR
中断检测	1	X	X	X	X
挂起/恢复操作	X	X	X	X	X
读取等待 SDMMC_CK 时钟停止 (START)	X	1	0	1	1
读取等待 SDMMC_CK 时钟停止 (STOP)	X	1	1	1	1
读取等待 SDMMC_D2 信号 (START)	X	0	0	1	1
读取等待 SDMMC_D2 信号 (STOP)	X	0	1	1	1

SD I/O 中断

为使 SD I/O 卡能够中断主机，在 SD 接口的引脚 8（4 位模式下与 SDMMC_D1 共用）上提供了一个中断功能。每个卡或者卡中的功能都可以使用中断。SD I/O 中断为电平敏感型，这意味着，在主机识别出中断线并对其采取操作或者由于中断周期结束而使其失效之前，中断线必须保持有效状态（低电平）。在主机处理完中断后，将通过在 SD I/O 卡内部寄存器中的相应位中进行 I/O 写入来清除中断状态位。所有 SD I/O 卡的中断输出均为低电平有效，并且应用程序必须在所有数据线 (SDMMC_D[3:0]) 上提供外部上拉电阻。

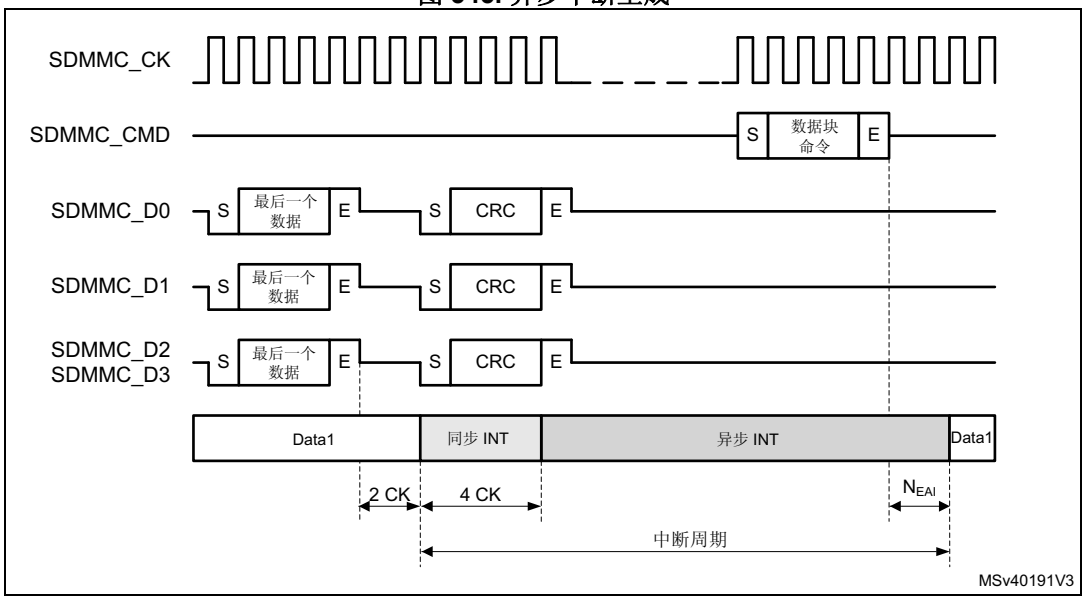
在 SD 1 位模式下，引脚 8 专用于中断功能 (IRQ)，并且中断没有时序限制。

在 SD 4 位模式下，只有在中断周期内，主机才会将引脚 8 (SDMMC_D1/IRQ) 的电平采样到中断检测器中。而在所有其他时候，主机中断将忽略此值。当卡上使能中断且 SDIOEN 位置 1 时，中断周期开始，请参见*表 307* 中的寄存器设置。

在 4 位模式下，卡可以生成同步或异步中断，分别通过卡 CCCR 寄存器中的 SAI 和 EAI 位指示。

- 同步中断，要求 SDMMC_CK 有效。
- 异步中断，可以在 SDMMC_CK 停止时生成，即在最后一个数据块后的卡中断周期开始后的 4 个周期后生成。

图 545. 异步中断生成



中断周期的时序取决于总线速度模式：

在 DS、HS、SDR12 和 SDR25 模式（通过寄存器位 BUSSPEED 选择）下，中断周期与 SD 时钟同步。

- 中断周期在数据块传输命令（在 CMDTRANS 位置 1 时发送命令）的结束位的下一个时钟结束或在 DTEN 位置 1 时结束。
- 中断周期在数据块完成传输后的 2 个 SDMMC_CK 后恢复。
- 在数据块间隙，中断周期被限制为 2 个 SDMMC_CK 周期。

注： *DTEN 不能用于启动与 SD 和 eMMC 卡之间的数据传输。*

图 546. 同步中断周期数据读取

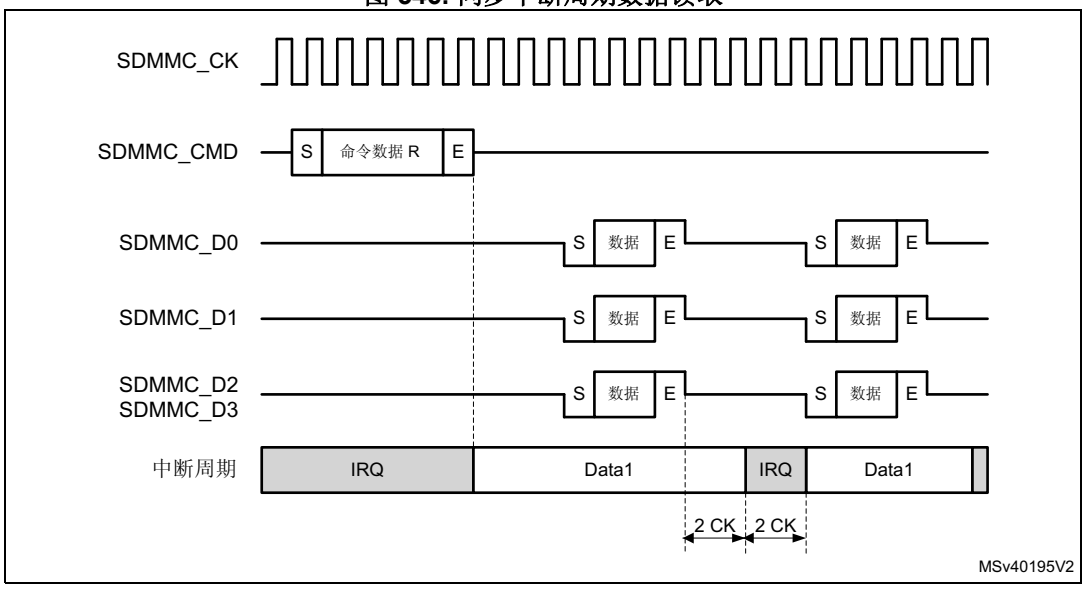
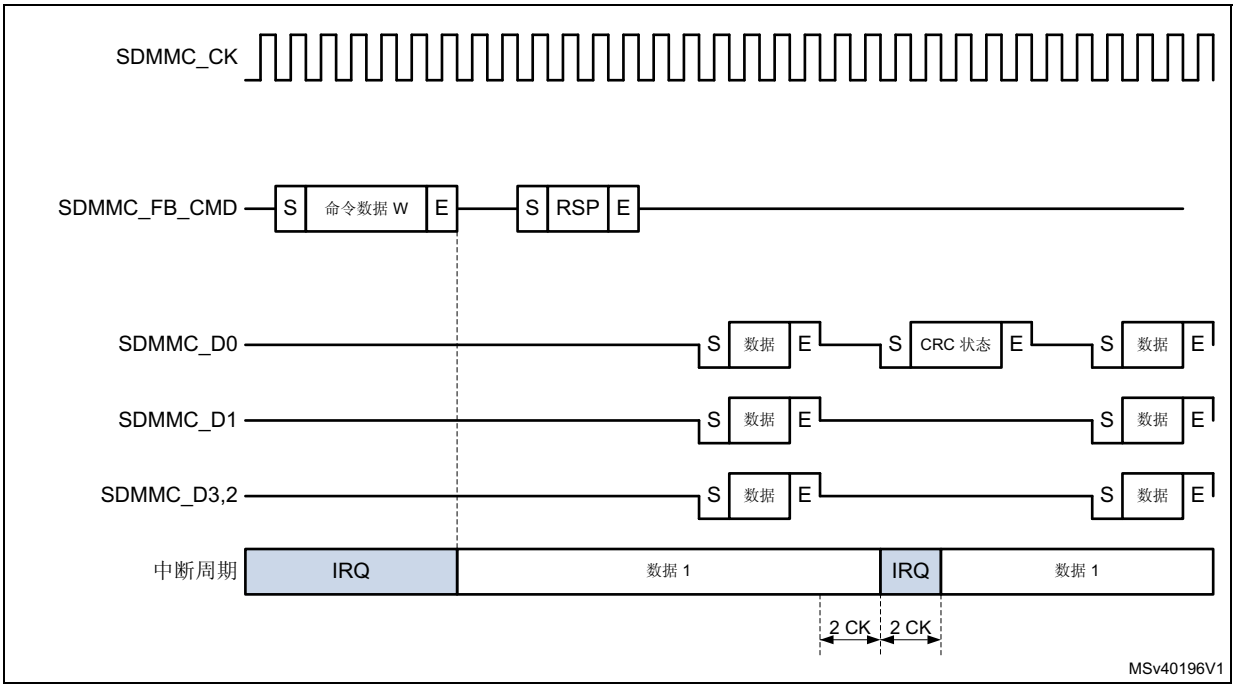


图 **547.** 同步中断周期数据写入



在 SDR50 和 DDR50（通过寄存器位 BUSSPEED 选择）下，由于卡与主机之间存在传播延迟，因此中断周期是异步的。

- 卡中断周期在数据块传输命令（在 CMDTRANS 位置 1 时发送命令）的结束位后的 0 到 2 个 SDMMC_CK 周期后结束或在 DTEN 位置 1 时结束。在主机中，中断周期在数据块传输命令的结束位之后结束。如果在此中断周期内主机未检测到命令结束位，则将在 1 到 2 个周期后发出卡中断。
- 卡中断周期在最后一个数据块完成传输后的 2 到 4 个 SDMMC_CK 后恢复。主机将始终在最后一个数据块之后的 2 个周期后恢复中断周期。
- 数据块间隙内没有中断周期。

注： *DTEN 不能用于启动与 SD 和 eMMC 卡之间的数据传输。*

图 548. 异步中断周期数据读取

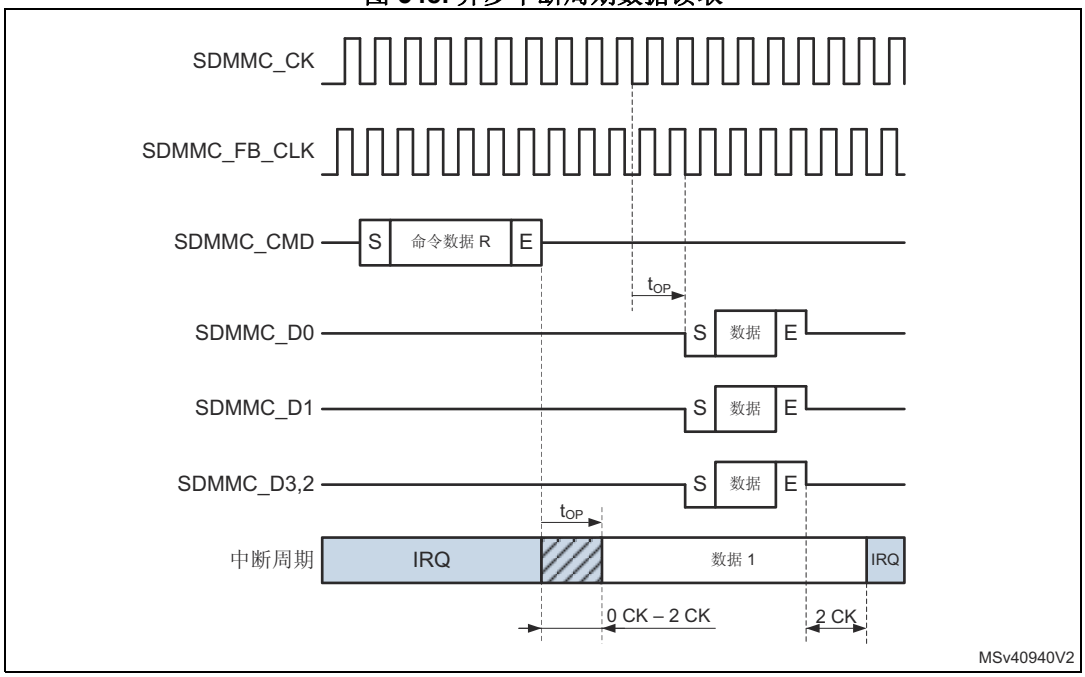
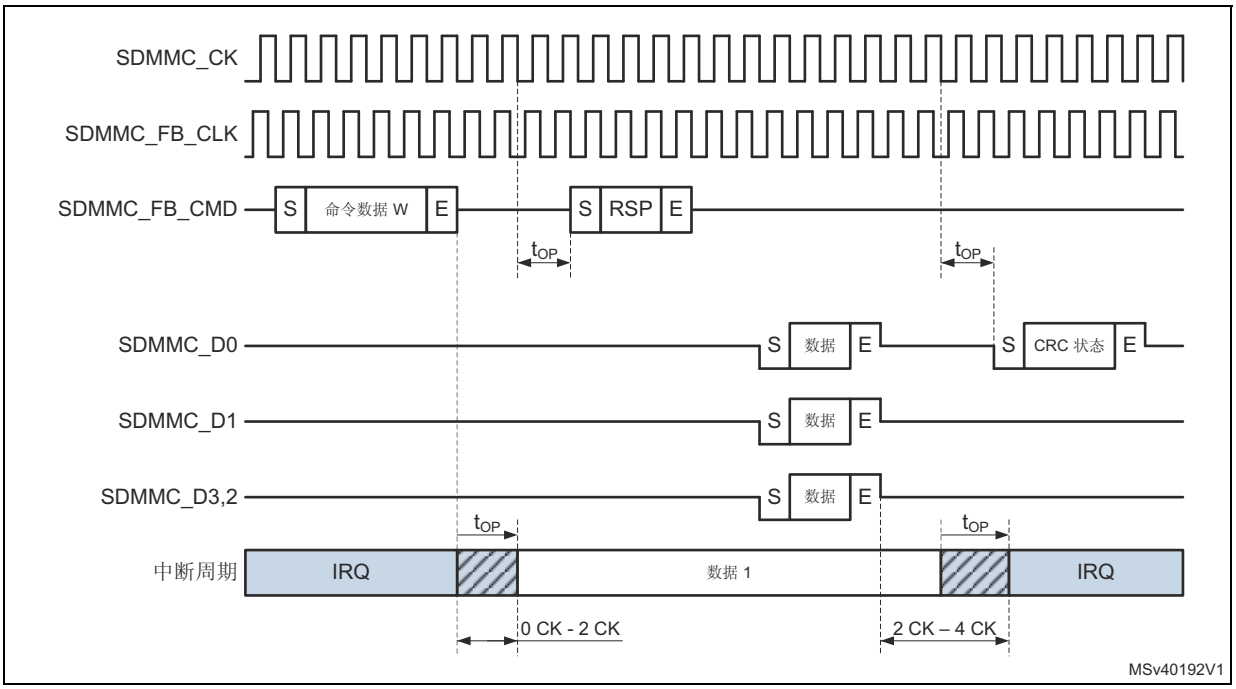


图 549. 异步中断周期数据写入



当传输开放式多块数据并使用 DTMODE “以 STOP_TRANSMISSION 命令结束的块数据传输”时，SDMMC 将在最后一个数据块之后屏蔽中断周期，直到 CMD12 STOP_TRANSMISSION 命令结束。

中断周期适用于存储器和 I/O 操作。

在 4 位模式下，可以根据*表 308* 来区分中断与其他信号。

表 308. 4 位模式启动、中断和 CRC 状态信号检测

SDMMC 数据线	启动	中断	CRC 状态
SDMMC_D0	0	1 或 CRC 状态	0
SDMMC_D1	0	0	X
SDMMC_D2	0	1 或读取等待	X
SDMMC_D3	0	1	X

SD I/O 挂起和恢复

SDIO V4.00 或更高版本不支持此功能。

在多功能 SD I/O 中或者在同时具备 I/O 和存储器功能的卡中，有多个设备（I/O 和存储器）共享对 MMC/SD 总线的访问权限。要在多个设备之间共享对主机的访问权限，SD I/O 和组合卡可以有选择性地实施“挂起/恢复”这一概念。如果卡支持挂起/恢复，则主机可以临时停止（挂起）对某一功能或存储器的数据传输操作以释放总线，从而将总线用于其他功能或存储器的更高优先级传输。此更高优先级传输完成后，原始传输将从中断位置重新开始（恢复）。

要在总线上执行挂起/恢复操作，主机将执行以下步骤：

1. 确定当前使用 SDMMC_D [3:0] 线的功能
2. 请求将优先级较低或速度较慢的事务挂起
3. 等待事务挂起完成
4. 开始更高优先级的事务
5. 等待更高优先级事务完成
6. 恢复挂起的事务

接收到挂起命令的卡将使用其当前总线状态进行响应。只有当总线通过卡挂起后，总线状态才会指示挂起完成。

有多种挂起情况：

- 在数据传输开始之前接受挂起请求。
- 未接受挂起请求（由于同时在传输数据），主机持续检查请求，直到接受请求。（数据传输已停止）。
- 写繁忙时的挂起请求。
- 多次写入时的挂起请求。
- 读取等待期间的挂起请求。

为了让主机获知总线是否已释放，应检查挂起请求的状态（挂起完成）。

当挂起请求响应的总线状态指示挂起完成时，表示卡已释放总线。此时，应保存挂起操作的状态，之后可启动其他操作。

挂起命令应在 CMDSUSPEND 位置 1 时发送。这样，当总线挂起时（响应位 BS = 0），可在挂起命令响应之后启动中断周期。

硬件不会保存挂起的操作恢复后还需传输的剩余数据量。固件负责确定已传输的数据，从而保证恢复后的剩余字节数是正确的。

从卡中接收数据时，SDMMC 可以在读取数据块结束（DPSM 处于 Wait_R 状态）之后挂起读取操作。在收到卡的挂起确认响应后，固件应执行以下步骤：

1. 应通过将 DTHOLD 位置 1 来停止正常接收过程。
   a) 应读取 FIFO 中的剩余数据字节数，直到接收 FIFO 为空（RXFIFOE 标志置 1），当 IDMAEN = 0 时，FIFO 应使用 FIFORST 复位。
2. 通过 DHOLD 标志指示对已从 FIFO 中读取所有数据以及挂起完成的确认。
   a) 恢复操作后仍需读取的剩余数据字节数（数据块的倍数）应通过 DATACOUNT 指示的剩余字节数确定。

注： *挂起过程中出现的 DTIMEOUT 标志将被忽略。*

要继续从卡中接收数据，固件应执行以下步骤：

1. 应在 DATALENGTH 中编程剩余的数据字节数（数据块的倍数）。
2. 应在 DTDIR 位中将 DPSM 配置为接收数据。
3. 应从 CPSM 发送恢复命令，此时 CMDTRANS 位置 1 且 CMDSUSPEND 位置 1，该命令将在数据传输恢复（响应位 DF = 1）且 DPSM 使能时结束中断周期，之后卡将继续发送数据。

在向卡发送数据时，SDMMC 可以在写入数据块 CRC 状态结束（DPSM 处于繁忙状态）之后挂起写入操作。在向卡发送挂起命令前，固件应执行以下步骤：

1. 使能 DHOLD 标志（当 IDMAEN = 0 时使能 DBCKEND 标志）。
2. 应防止 DPSM 通过将 DTHOLD 置 1 开始发送新的数据块。
3. 当 IDMAEN = 0 时：如果接收到 DBCKEND 标志，将停止数据传输。固件可以停止填充 FIFO，之后应使用 FIFORST 复位 FIFO。恢复操作后，仍保留在 FIFO 中的任何字节都需要重写。
4. 当接收到 DHOLD 标志时，将停止数据传输。恢复操作后仍需写入的剩余数据字节数应通过 DATACOUNT 指示的剩余字节数确定。
5. 要挂起卡，应在 CMDSUSPEND 位置 1 时通过 CPSM 发送挂起命令。这样，当总线挂起时（响应位 BS = 0），可在挂起命令响应之后启动中断周期。

要继续向卡发送数据，固件应执行以下步骤：

1. 应在 DATALENGTH 中编程剩余的数据字节数。
2. 应通过设置 DTDIR 来配置 DPSM 的传输方向，通过 CPSM 发送恢复命令（CMDTRANS 位置 1 且 CMDSUSPEND 位置 1）的方式来使能 DPSM。这将结束中断周期并开始数据传输。DPSM 将在 SDMMC_D0 未发出繁忙信号时进入 Wait_S 状态，而在 SDMMC_D0 发出繁忙信号时进入繁忙状态。
3. 当 IDMAEN = 1 时：需要针对要传输的剩余字节重新编程 DMA。
4. 当 IDMAEN = 0 时：固件应开始使用剩余的数据填充 FIFO。

SD I/O 读取等待

有两种方法可以在块间隙内暂停数据传输：

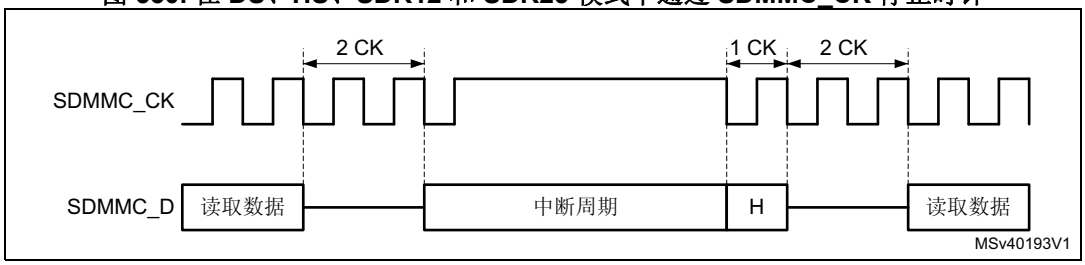
1. 停止 SDMMC_CK。
2. 在 SDMMC_D2 上使用读取等待信号。

SDMMC 可以根据*表 307* 通过寄存器设置执行读取等待。

根据 SDMMC 工作模式（DS、HS、SDR12 和 SDR25）或（SDR50 和 DDR），每种方法都有不同的特性。

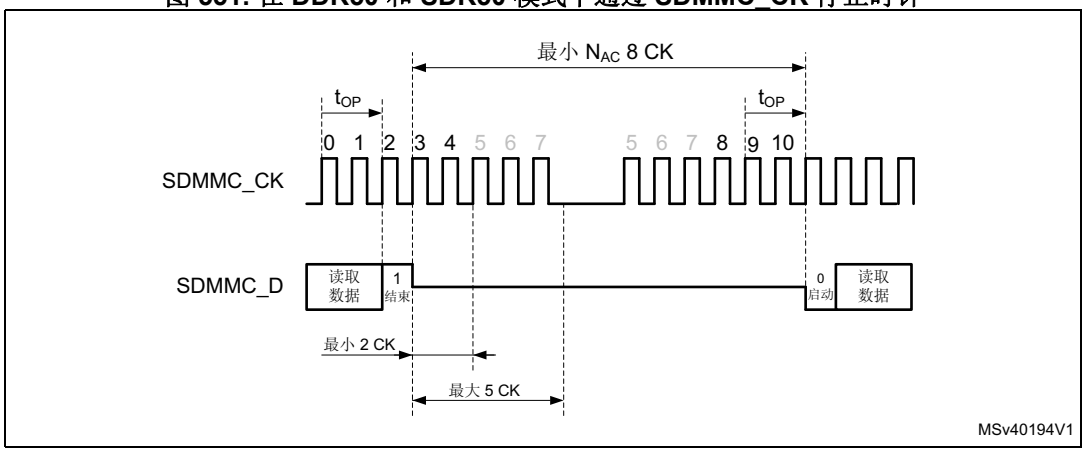
在 DS、HS、SDR12 和 SDR25 模式下通过停止 SDMMC_CK 实现暂停读取操作的时序中，SDMMC_CK 可在结束位之后的 2 个 SDMMC_CK 周期后停止。主机就绪后将通过重启时钟进行恢复，请参见*图 550*。

图 550. 在 DS、HS、SDR12 和 SDR25 模式下通过 SDMMC_CK 停止时钟



在 SDR50 和 DDR50 模式下通过停止 SDMMC_CK 实现暂停读取操作的时序中， SDMMC_CK 可在结束位之后的 2-5 个 SDMMC_CK 周期后停止。主机就绪后将通过重启时钟进行恢复，请参见*图 551*。（在 DDR50 模式下，如果时钟线为低电平，则只应在下降沿后停止 SDMMC_CK）。

图 551. 在 DDR50 和 SDR50 模式下通过 SDMMC_CK 停止时钟



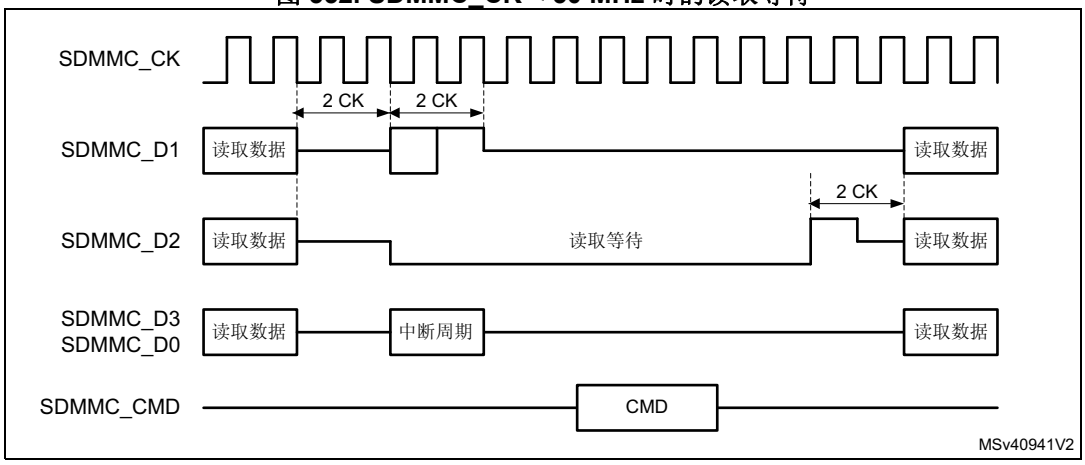
在读取等待 SDMMC_CK 时钟停止期间，如果 RWSTART 置 1，DSPM 会在当前接收到的数据块 CRC 的结束位之后停止时钟。时钟将在向 RWSTOP 位写入 1 后再次启动，之后 DPSM 将等待来自卡的起始位。

SDMMC_CK 停止时，无法向卡发出任何命令。在读取等待间隔内，SDMMC 仍可在 SDMMC_D1 上检测 SDIO 中断。

SDMMC_D2 上可选的读取等待信号 (RW) 操作仅针对 SD 的 1 位和 4 位模式进行了定义。“读取等待”操作允许主机发出如下信号：卡正在读取多个寄存器 (IO_RW_EXTENDED, CMD53) 以临时停止数据传输，同时允许主机向 SD I/O 设备中的任何功能发送命令。要确定卡是否支持“读取等待”协议，主机必须测试内部卡寄存器中的功能位。

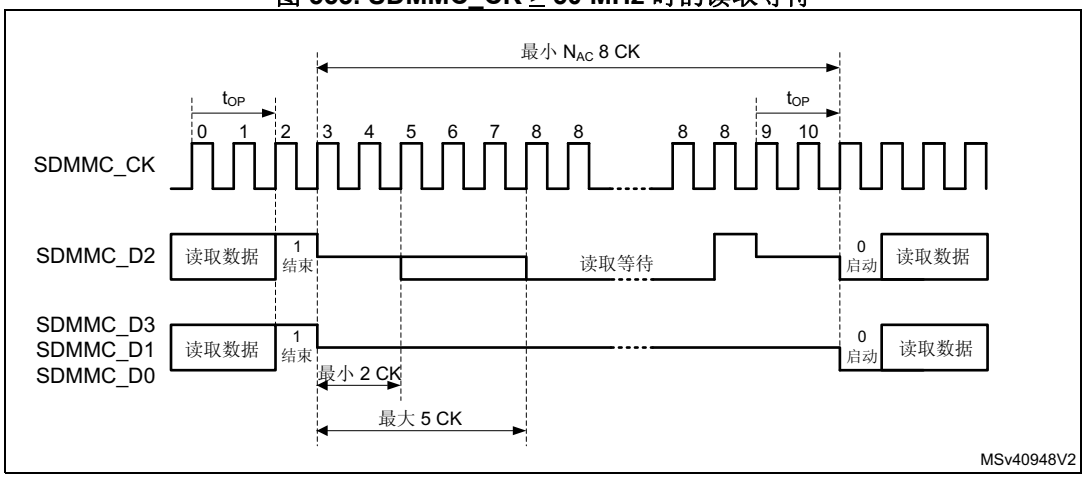
SDMMC_CK 小于 50 MHz（DS、HS、SDR12 和 SDR25）的读取等待的时序基于卡在 SDMMC_D1 上生成的中断周期。主机在中断周期内将 SDMMC_D2 置为低电平，请求卡进入读取等待状态。要退出读取等待状态，主机应在将 SDMMC_D2 设置为高阻态之前的一个 SDMMC_CK 周期内将 SDMMC_D2 升为高电平，请参见*图 552*。

图 552. SDMMC_CK < 50 MHz 时的读取等待



对于 SDMMC_CK 大于 50 MHz 的 SDR50 以及 DDR50，卡会将 SDMMC_D2 上的读取等待请求视为异步事件。主机在 2-5 个 SDMMC_CK 周期后将 SDMMC_D2 置为低电平，请求卡进入读取等待状态。要退出读取等待状态，主机应在将 SDMMC_D2 设置为高阻态之前的一个 SDMMC_CK 周期内将 SDMMC_D2 升为高电平。主机应基于 SDMMC_CK 时钟将 SDMMC_D2 升为高电平（参见*图 553*）。

图 553. SDMMC_CK ≥ 50 MHz 时的读取等待



在读取等待 SDMMC_D2 信号期间，如果 RWSTART 置 1，DPSM 将在当前接收的数据块 CRC 的结束位后驱动 SDMMC_D2。当将 1 写入 RWSTOP 位时，SDMMC_D2 上的读取等待信号将被删除。DPSM 保持读取等待状态的时间将延长两个 SDMMC_CK 时钟周期，以将 SDMMC_D2 驱动为 1 并保持一个时钟周期（根据 SDIO 规范），之后 DPSM 将等待来自卡的起始位。

在 SDMMC_D2 上的读取等待信号传输期间，可以向卡发出命令。在读取等待间隔内，SDMMC 可以在 SDMMC_D1 上检测 SDIO 中断。

48.5.2 CMD12 发送时序

CMD12 用于停止/中止数据传输，卡数据传输在停止传输命令的结束位后的两个时钟周期后终止。

表 309. CMD12 用例

数据操作	停止传输命令 CMD12 的说明
MMC 流写入	通过发送停止传输命令来停止/中止数据传输。
MMC 开放式多块写入	通过发送停止传输命令来停止/中止数据传输。 如果卡检测到错误，主机必须通过发送停止传输命令来中止操作。
预定义块数的 MMC 块写入	在该类型的多块写入操作结束时，不需要停止传输命令。（在卡接收到最后一个块后发送停止传输命令被视为非法命令。） 如果卡检测到错误，主机必须通过发送停止传输命令来中止操作。
MMC 流读取	通过发送停止传输命令来停止/中止数据传输。
MMC 开放式多块读取	通过发送停止传输命令来停止/中止数据传输。 如果卡检测到错误，主机必须通过发送停止传输命令来中止操作。
预定义块数的 MMC 块读取	在该类型的多块读取操作结束时，不需要停止传输命令。（在卡发送最后一个块后发送停止传输命令被视为非法命令。） 通过发送停止传输命令可中止事务。 如果卡检测到错误，主机必须通过发送停止传输命令来中止操作。

所有数据写入和读取命令都可以随时通过停止传输命令 CMD12 中止。如果正在进行数据传输，请按照以下数据中止步骤操作：

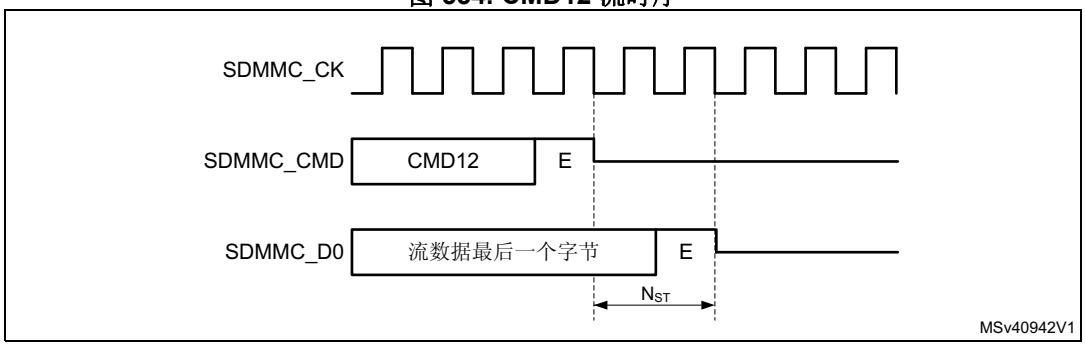
1. 在寄存器中加载 CMD12 停止传输命令并将 CMDSTOP 位置 1。
   a) 当命令发送到 DPSM 时，这将导致 CPSM 生成中止信号。
2. 将 CPSM 配置为立即发送命令（将 WAITPEND 位清零）。
   a) 发送数据时，卡将在停止传输命令结束位后的 2 个周期后停止数据传输。
   未发送数据时，卡将不会开始发送任何新数据。
   b) 发送数据时，主机将在停止传输命令结束位之后发送最后一个数据位，后跟一个结束位。
   未发送数据时，主机将不会开始发送任何新数据。
3. 当 IDMAEN = 0 时，需要使用 FIFORST 复位 FIFO。
   a) 向卡写入数据时。出现 CMDREND 标志时，固件将停止向 FIFO 写入数据。随后，将使用 FIFORST 复位 FIFO，这将清空 FIFO。
   b) 从卡读取数据时。出现 CMDREND 标志时，固件将从 FIFO 读取剩余数据。随后，将使用 FIFORST 复位 FIFO。
4. 当 IDMAEN = 1 时，硬件将处理 FIFO。
   a) 向卡写入数据时。出现中止信号时，硬件将停止 IDMA，随后 FIFO 将被清空。
   b) 从卡读取数据时。出现中止信号时，硬件将指示 IDMA 将剩余数据从 FIFO 传输到 RAM。
5. 当 FIFO 为空/复位时，将生成 DABORT 标志。

流操作和 CMD12

要在待传输的最后一个字节之后停止流传输，应在数据流的最后一个字节结束时发送 CMD12 结束位时序。请按照以下流数据写入步骤操作：

1. 在 DPSM 中初始化流数据，DTMODE = MCC 流数据传输。
2. 从 CPSM 发送 WRITE_DATA_STREAM 命令 (CMDTRANS = 1)。
3. 在命令寄存器中预装载 CMD12，CMDSTOP 位应置 1。
4. 将 CPSM 配置为仅在最后一个数据（根据 DATALENGTH）的等待挂起 (WAITPEND = 1) 结束后发送命令。
5. 使能 CPSM 发送 STOP_TRANSMISSION 命令，流数据结束位和命令结束位将对齐。
   a) 当 DATALENGTH > 5 字节时，CPSM 中将等待命令 CMD12 与数据传输结束位对齐。
   b) 当 DATALENGHT < 5 字节时，命令 CMD12 将提前启动，DPSM 将保持 Wait_S 状态，以将数据传输结束位与 CMD12 结束位对齐。
6. 可通过将 WAITPEND 位清零随时中止流数据写入过程。这将导致预装载的 CMD12 立即发送，并停止数据流写入过程。

图 554. CMD12 流时序



要在最后一个字节之后停止流传输读取过程，应在数据流最后一个字节后发送 CMD12 结束位时序。请按照以下流数据读取步骤操作：

1. 等待 DPSM 接收到所有数据（DATAEND 标志）。
   a) 即使卡发送更多数据，DPSM 接收的数据也不会超过 DATALENGTH 所指示的数据量。
2. 通过 CPSM 发送 CMD12。
   a) CMD12 将使卡停止发送数据。

注： *即使卡继续发送数据，SDMMC 也不会在 DATACOUNT = 0 时从卡中继续接收任何数据。*

块操作和 CMD12

要在数据结束时停止块传输，应在最后一个块结束位之后发送 CMD12 结束位。

向卡写入数据时，将在写入数据块 CRC 令牌结束位之后发送 CMD12 结束位。这要求 CMD12 发送过程遵循数据块传输时序。要停止开放式多块写入操作，请按照以下步骤操作：

1. 在开始数据传输之前，将 DTMODE 设置为“以 STOP_TRANSMISSION 命令结束块数据传输”。
2. 等待 DPSM 发送完所有数据并等待接收到 CRC 令牌（DATAEND 标志）。
   a) DPSM 发送的数据不会超过 DATALENGTH 所指示的数据量。
3. 通过 CPSM 发送 CMD12。
   a) CMD12 会将卡设置为空闲模式。

当从卡读取数据时，CMD12 结束位应最早发送，即在卡读取数据块最后一个数据位时发送。这要求 CMD12 发送过程遵循数据块接收时序。请按照以下停止开放式多块读取数据块步骤操作：

1. 在开始数据传输之前，将 DTMODE 设置为“以 STOP_TRANSMISSION 命令结束块数据传输”。
2. 等待 DPSM 接收到所有数据（DATAEND 标志）。
   a) 即使卡发送更多数据，DPSM 接收的数据也不会超过 DATALENGTH 所指示的数据量。
3. 通过 CPSM 发送 CMD12 并将 CMDSTOP 位置 1。
   a) CMD12 将使卡停止发送更多数据，并将卡设置为空闲模式。卡将中止正在进行的任何块传输操作。

注： *即使卡继续发送数据，SDMMC 也不会在 DATACOUNT = 0 时从卡中继续接收任何数据。*

48.5.3 睡眠 (CMD5)

MMC 卡可以通过 CMD5 在睡眠状态和待机状态之间转换。在睡眠状态下，卡的功耗降至最低，此时可关闭 Vcc 电源。

CMD5 (SLEEP) 用于启动从待机状态到睡眠状态的状态转换。在转换阶段，卡指示繁忙，同时会将 SDMMC_D0 拉为低电平。当卡停止将 SDMMC_DO 线拉为低电平时，表示已进入睡眠状态。

要将卡设置为睡眠状态，请按照以下步骤操作：

1. 使能 BUSYD0END 中断。
2. 发送 CMD5 (SLEEP)。
3. 出现 BUSYD0END 中断时，卡处于睡眠状态。
4. 允许关闭 Vcc 电源。

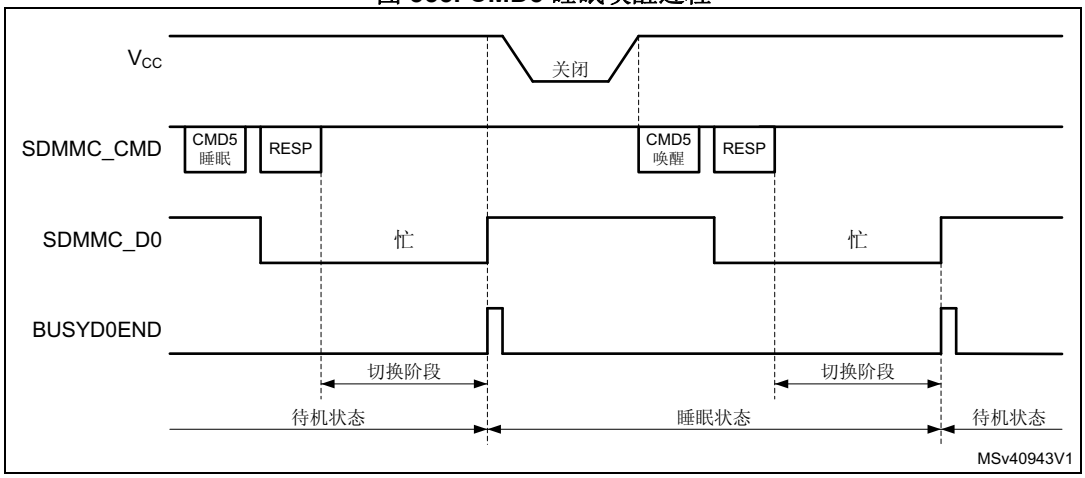
CMD5 (AWAKE) 用于启动从睡眠状态到待机状态的状态转换。在转换阶段，卡指示繁忙，同时会将 SDMMC_D0 拉为低电平。当卡停止将 SDMMC_DO 线拉为低电平时，表示已进入待机状态。

要将卡设置为睡眠状态，请按照以下步骤操作：

1. 接通 Vcc 电源，等待其达到最低工作电压。
2. 使能 BUSYD0END 中断。
3. 发送 CMD5 (AWAKE)。
4. 出现 BUSYD0END 中断时，卡处于待机状态。

只有在进入睡眠状态后，才允许关闭 Vcc 电源。在发送 CMD5 (AWAKE) 之前，应重新安装 Vcc 电源。

图 555. CMD5 睡眠唤醒过程



48.5.4 中断模式 (Wait-IRQ)

主机和卡同时进入和退出中断模式 (Wait-IRQ)。在中断模式下，不会进行任何数据传输。唯一允许的消息是来自卡或主机的中断服务请求响应。为了使中断模式正常工作，应根据开漏模式下可实现的 SDMMC_CMD 数据速率设置 SDMMC_CK 频率，具体取决于电容负载和上拉电阻。CLKDIV 应设置为 > 1，SETCLKRX 应选择 SDMMC_CLKio 或 SDMMC_CLKin 源。

主机必须确保发出 CMD40 (GO_IRQ_STATE) 之前卡处于待机状态。在等待中断响应时，SDMMC_CK 时钟信号必须保持有效。

中断模式下的卡（IRQ 状态）：

- 等待内部卡中断事件。出现中断事件后，卡开始发送中断服务请求响应。该响应在开漏模式下发送。
- 等待内部卡中断事件时，卡还会监视 SDMMC_CMD 线上的起始位。检测到启动位后，卡将中止中断模式并转换到待机状态。

中断模式下的主机（CPSM 等待状态，等待中断）：

- 等待卡中断服务请求响应（起始位）。
- 等待卡中断服务请求响应时，主机可中止中断模式（通过将 WAITINT 寄存器位清零），这会导致主机在开漏模式下发送 RCA = 0x0000 的中断服务请求响应 R5。

发送中断服务请求响应时，发送方逐位监视 SDMMC_CMD 位流。如果发送方的中断服务请求响应位与 SDMMC_CMD 线上的位不对应，则将停止发送。当存在多个发送方时，只有一个发送方将成功发送其完整的中断服务请求响应。即，如果主机与其中一个发送方同时发送，则主机将在传输位之后停止发送。

要处理中断模式，请按照以下步骤操作：

1. 根据开漏模式下可实现的 SDMMC_CMD 数据速率设置 SDMMC_CK 频率，CLKDIV 应设置为 > 1，SETCLKRX 应选择 SDMMC_CLKio。
2. 在命令寄存器中加载 CMD40 (GO_IRQ_STATE)。
3. 通过将 WAITINT 寄存器位置 1 使能等待中断。
4. 将 CPSM 配置为立即发送命令。
   a) 这将会发送 CMD40，并且使 CPSM 在等待状态下停止，等待中断服务请求响应。
5. 要退出等待中断状态（CPSM 等待状态）：
   a) 检测到中断服务请求响应起始位后，CPSM 将变为接收状态以接收响应。是否成功接收到响应通过 CMDREND 或命令 CRC 错误标志来指示。
   b) 为中止中断模式，主机会将 WAITINT 寄存器位清零。这将导致主机自行发送中断服务请求响应，从而使 CPSM 进入接收状态。是否成功接收到响应通过 CMDREND 或命令 CRC 错误标志来指示。

注： *当同步发送中断服务请求响应起始位发生冲突时，主机将在传输位之后丢失总线访问。*

48.5.5 启动操作

在启动操作模式下，主机可以通过 2 种启动操作功能从卡中读取启动数据：

1. 正常启动。（使 CMD 线保持低电平）
2. 备用启动。（发送具有参数 0xFFFFFFFA 的 CMD0）

可以根据以下配置选项读取启动数据，具体取决于卡寄存器设置：

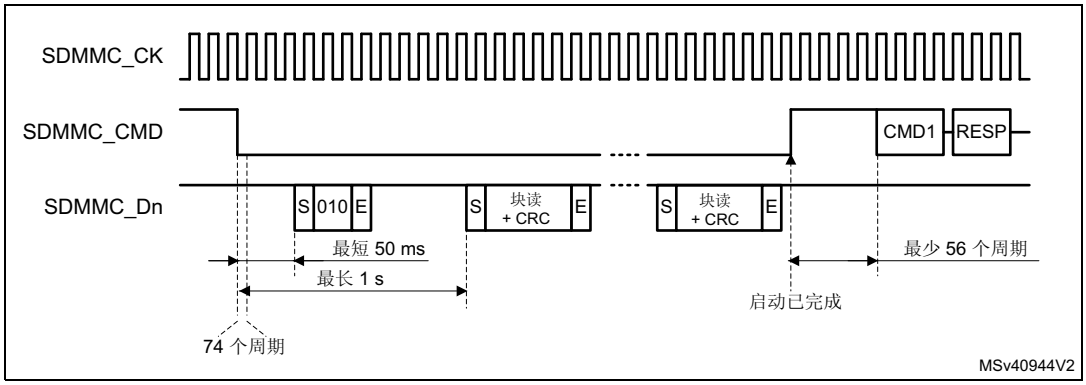
- 从中读取启动数据的分区（EXT_CSD 字节 [179]）
- 启动数据大小（EXT_CSD 字节 [226]）
- 启动期间的总线配置（EXT_CSD 字节 [177]）
- 从卡接收启动确认。（EXT_CSD 字节 [179]）

如果使能启动确认，则卡将在请求启动模式（通过将 CMD 线置为低电平）或发送具有参数 0xFFFFFFFA 的 CMD0 后的 50 ms 内在 SDMMC_D0 上发送模式 010，并将提供启动确认超时 (ACKTIMEOUT) 和确认状态 (ACKFAIL)。

正常启动操作

如果在卡上电或复位后 SDMMC_CMD 线的低电平状态持续至少 74 个时钟周期，则卡将在发出第一个命令之前识别出正在发起启动模式。在 CMD 线变为低电平后的 1 秒内，卡开始在 SDMMC_Dn 线上发送第一个启动代码数据。主机必须使 SDMMC_CMD 线保持低电平，直到读取完所有启动数据为止。主机可通过将 SDMMC_CMD 线拉高来终止启动模式。

图 556. 正常启动模式操作



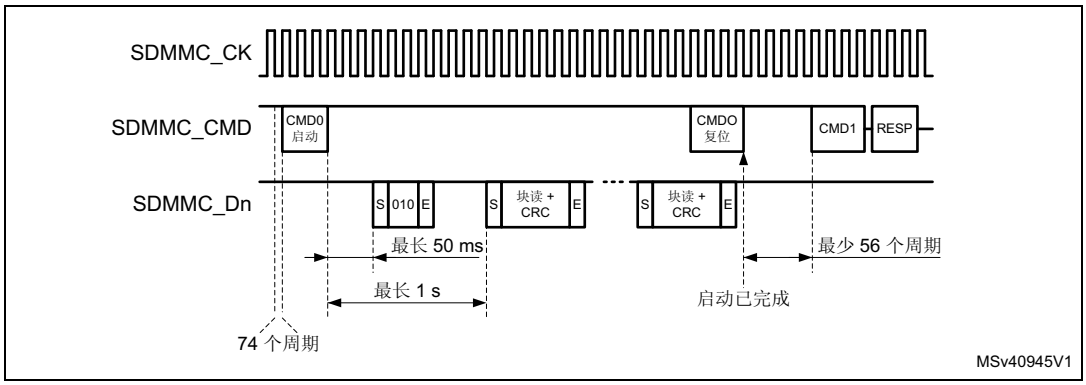
要执行正常的启动过程，需按照以下步骤操作：

1. 将卡复位。
2. 如果请求启动确认，需使能 BOOTACKEN、设置 ACKTIME 并使能 ACKFAIL 和 ACKTIMEOUT 中断。
3. 将 DPSM 设置为接收模式 (DTDIR) 并在 DATALENGTH 中设置要接收的数据字节数，以使能数据接收。
4. 使能 DTIMEOUT、DATAEND 和 CMDSENT 中断，用于完成启动命令确认。
5. 通过 BOOTMODE 选择正常启动操作模式，并通过 BOOTEN 使能启动。启动过程的启动方法是通过 CPSMEN 使能 CPSM。这将导致：
   - SDMMC_CMD 被驱动为低电平。（BOOTMODE = 正常启动）。
   - ACK 超时被启动。
   - DPSM 被使能。
6. 可通过 ACKFAIL 标志或 ACKTIMEOUT 标志（使能时）检测是否接收到不正确的启动确认。
   - 当接收到不正确的启动确认时，将出现 ACKFAIL 标志。
   - 当未及时接收到启动确认时，将出现 ACKTIMEOUT 标志。
7. 当接收到所有启动数据时，将出现 DATAEND 标志。
   - 当数据 CRC 失败时，还会生成 DCRCFAIL 标志。
   - 当数据发生超时时，还会生成 DTIMEOUT 标志。
8. 当接收到最后一个数据时，从 FIFO 读取数据，直到 FIFO 为空 (RXFIFOE = 1)，之后将生成数据结束 DATAEND 标志。
   - SDMMC 已接收完所有数据且 DPSM 处于禁止状态。
9. 通过固件清零 BOOTEN 将终止启动过程，这会导致 SDMMC_CMD 线变为高电平。56 个周期后将生成 CMDSENT 标志，以指示可以发送新命令。
   a) 如果在接收到所有数据之前通过固件中止启动过程，CPSM 中止信号将停止数据接收并禁止 DPSM，此时将生成 DABORT 标志（使能时）。
10. CMDSENT 标志用于指示启动过程结束，卡已准备好接收新命令。

备用启动操作

卡上电或复位后，如果主机在 74 个时钟周期后发送具有参数 0xFFFFFFFA 的 CMD0，则卡将在发出 CMD0 之前识别出正在发起启动模式。在发送具有参数 0xFFFFFFFA 的 CMD0 后的 1 秒内，卡开始在 SDMMC_Dn 线上发送第一个启动代码数据。主机将通过发送 CMD0（复位）来终止启动操作。

图 557. 备用启动模式操作



要执行备用启动过程，需按照以下步骤操作：

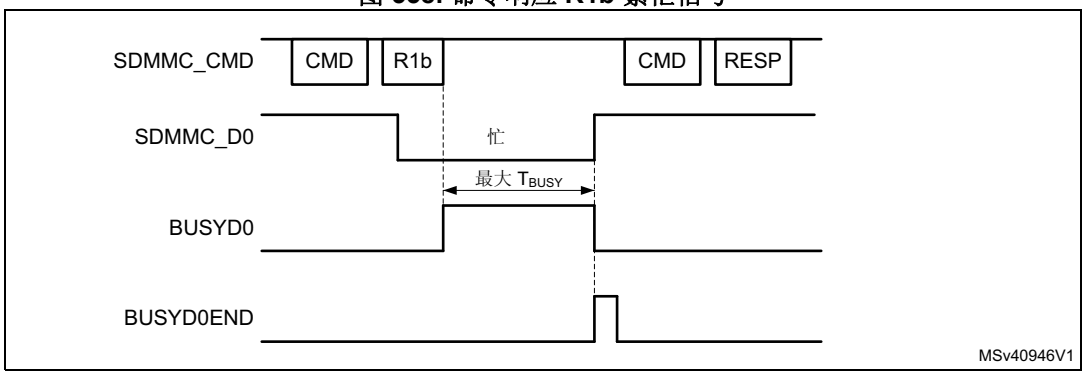
1. 使 SDMMC 变为掉电状态，并将卡复位。
2. 使 SDMMC 变为上电状态。这将保证在任何命令之前提供 74 个 SCDMMC_CK 周期的时钟驱动。
3. 如果请求启动确认，需使能 BOOTACKEN、设置 ACKTIME 并使能 ACKTIMEOUT 标志。
4. 将 DPSM 设置为接收模式 (DTDIR) 并在 DATALENGTH 中设置要接收的数据字节数，以使能数据接收。使能 DTIMEOUT 和 DATAEND 标志。
5. 通过 BOOTMODE 选择备用启动操作模式，在命令寄存器中加载具有参数 0xFFFFFFFA 的 CMD0。使能 CMDSENT 标志以指示完成启动命令确认，并通过 BOOTEN 使能启动。启动过程的启动方法是通过 CPSMEN 使能 CPSM。这将导致：
   - 加载的命令和参数被发送出去。（BOOTMODE = 备用启动）。
   - ACK 超时被启动。
   - DPSM 被使能。
6. 发送命令后，将生成 CMDSENT 标志，此时 BOOTEN 位应清零。
7. 可通过 ACKFAIL 标志（使能时）检测是否接收到启动确认。
   - 当未及时接收到启动确认时，将出现 ACKTIMEOUT 标志。
8. 当接收到所有启动数据时，将出现 DATAEND 标志。
   - 当数据 CRC 失败时，还会生成 DCRCFAIL 标志。
   - 当数据发生超时时，还会生成 DTIMEOUT 标志。
9. 当接收到最后一个数据时，从 FIFO 读取数据，直到 FIFO 为空 (RXFIFOE = 1)，之后将生成数据结束 DATAEND 标志。
   - SDMMC 已接收完所有数据且 DPSM 处于禁止状态。

10. 在通过发送 BOOTMODE = 备用启动的 CMD0（复位）以中止启动过程之前，应将 BOOTEN 位清零。这将导致在命令后的 56 个周期后出现 CMDSENT 标志。
    – 如果在接收到所有数据之前通过固件中止启动过程，CPSM 中止信号将停止数据传输并禁止 DPSM，此时将生成 DABORT 标志（使能时）。
11. CMDSENT 标志用于指示启动过程结束，卡已准备好接收新命令。当 RESET 命令成功发送后，必须将 BOOTMODE 控制位清零才能终止启动操作。

48.5.6 响应 R1b 的处理

当发送具有 R1b 响应的命令时，繁忙信号将反映在 BUSYD0 寄存器位上，而繁忙释放信号将通过 BUSYD0END 标志反映。SDMMC_D0 线在 R1b 响应结束时进行采样，在 BUSYD0 寄存器位中进行指示。当 SDMMC_D0 线释放繁忙状态时，BUSYD0 寄存器位将复位为不忙，同时将生成 BUSYD0END 标志。

图 558. 命令响应 R1b 繁忙信号



发送命令之前，应在 DATATIME 寄存器中设置预期的最长繁忙时间。当 R1b 响应繁忙状态保持有效的时间超过编程的时间时，DTIMEOUT 标志（使能时）将置 1。

发送具有 R1b 响应的命令时，如果要检测 SDMMC_D0 繁忙信号，请按照以下步骤操作：

- 使能 CMDREND 标志。
- 通过 CPSM 发送命令。
- 出现 CMDREND 标志时，检查 BUSYD0 寄存器位。
  – 如果 BUSYD0 指示不忙，则向固件发出繁忙释放信号
  – 如果 BUSYD0 指示繁忙，则等待 BUSYD0END 标志
- 出现 BUSYD0END 标志时，向固件发出繁忙已释放信号。
- 出现 DTIMEOUT 标志时，表示繁忙状态保持有效的时间超过编程的时间。

48.5.7 复位和卡掉电再上电

复位

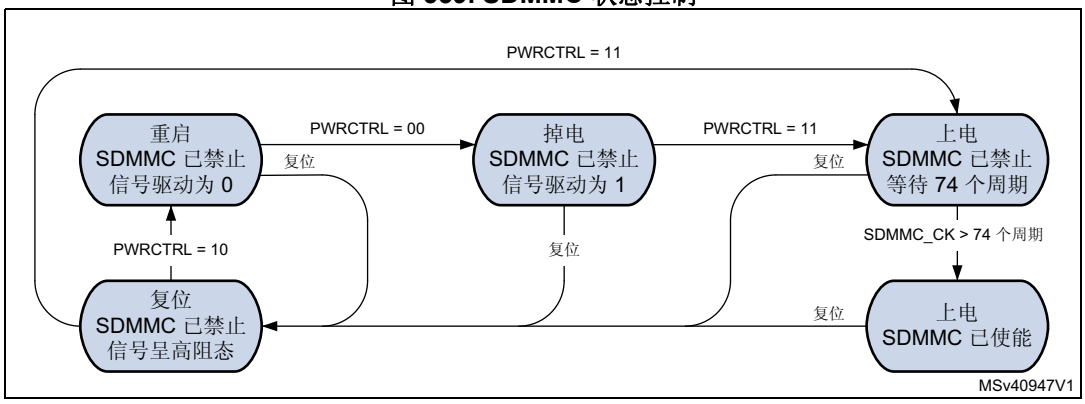
复位后，SDMMC 将处于复位状态。在此状态下，SDMMC 将被禁止，不能传输任何命令和数据。SDMMC_D[7:0] 和 SDMMC_CMD 处于高阻态，SDMMC_CK 驱动为低电平。

SDMMC 应在变为上电状态之前进行配置。

在上电状态下，SDMMC_CK 时钟处于运行状态。首先将提供 74 个 SDMMC_CK 周期的时钟驱动，之后将使能 SDMMC，此时即可传输命令和数据。

SDMMC 状态由固件通过 PWRCTL 寄存器位进行控制，如*图 559* 所示。

图 559. SDMMC 状态控制

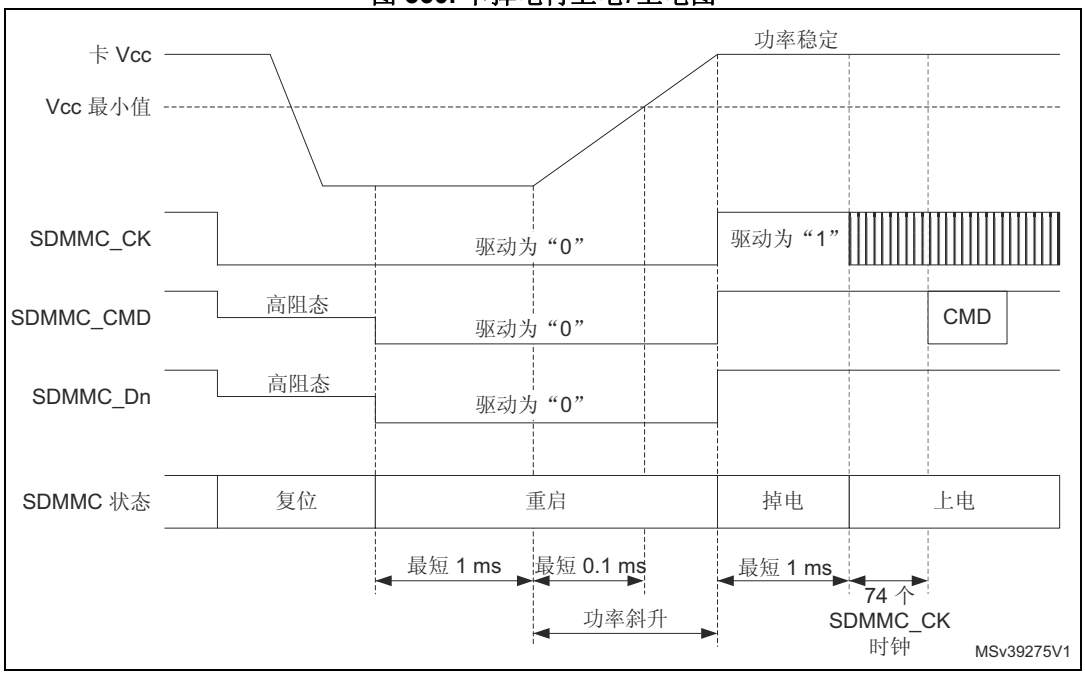


对卡进行掉电再上电

要对卡进行掉电再上电，请按照以下步骤进行操作：

1. 通过 RCC.SDMMCxRST 寄存器位复位 SDMMC。这将使 SDMMC 复位为复位状态，CPSM 和 DPSM 复位为空闲状态。
2. 禁止卡的 Vcc 电源。
3. 将 SDMMC 设置为掉电再上电状态。这将使 SDMMC_D[7:0]、SDMMC_CMD 和 SDMMC_CK 被驱动为低电平，以防止卡通过信号线供电。
4. 至少等待 1 ms 后，才能使能卡的 Vcc 电源。
5. 在电源斜升周期后，将 SDMMC 设置为掉电状态并至少持续 1 ms。SDMMC_D[7:0]、SDMMC_CMD 和 SDMMC_CK 将设置为“1”。
6. 经过 1 ms 的延迟后，将 SDMMC 设置为上电状态，在此期间将使能 SDMMC_CK 时钟。
7. 经过 74 个 SDMMC_CK 周期后，可以向卡发送第一个命令。

图 560. 卡掉电再上电/上电图



48.6 硬件流控制

硬件流控制功能用于避免 FIFO 下溢（发送模式）和上溢（接收模式）错误。

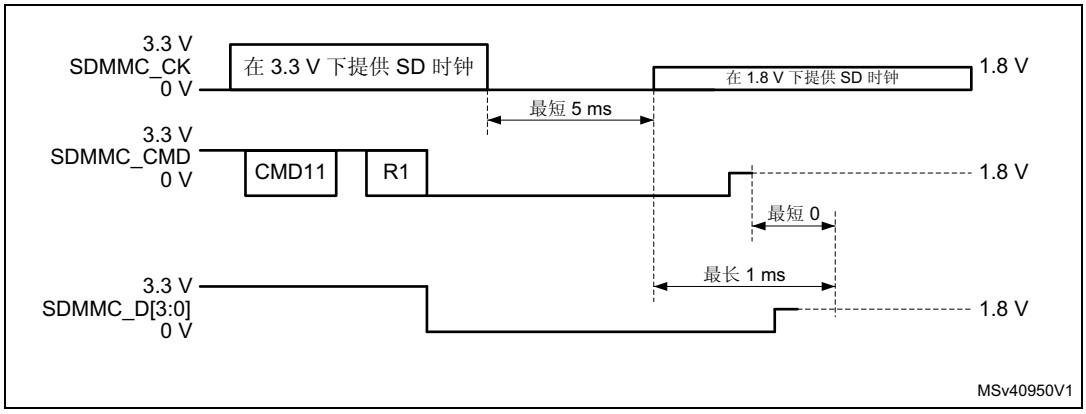
具体行为是在数据传输过程中停止 SDMMC_CK 并冻结 SDMMC 状态机。当 FIFO 无法发送或接收数据时，将停止数据传输。数据传输一直保持停止状态，直到发送 FIFO 半满或已存储 DATALENGHT 所指示的所有数据，或者直到接收 FIFO 半空。只有由 SDMMC_CK 提供时钟的状态机才会冻结，AHB 接口仍保持活动状态。因此，即使激活流控制，仍可填充或清空 FIFO。

要使能硬件流控制，HWFC_EN 寄存器位必须置 1。复位后，硬件流控制将禁止。

48.7 超高速 I 相 (UHS-I) 电压切换

UHS-I 模式（SDR12、SDR25、SDR50 和 DDR50）需要 1.8V 信号支持。上电后，卡将以 3.3V 模式启动。CMD11 将调用电压切换序列以切换到 1.8V 模式。当电压序列成功完成时，卡将以默认 SDR12 进入 UHS-I 模式，卡输入和输出时序将发生变化。

图 561. CMD11 信号电压切换序列



要执行信号电压切换序列，需按以下步骤进行操作：

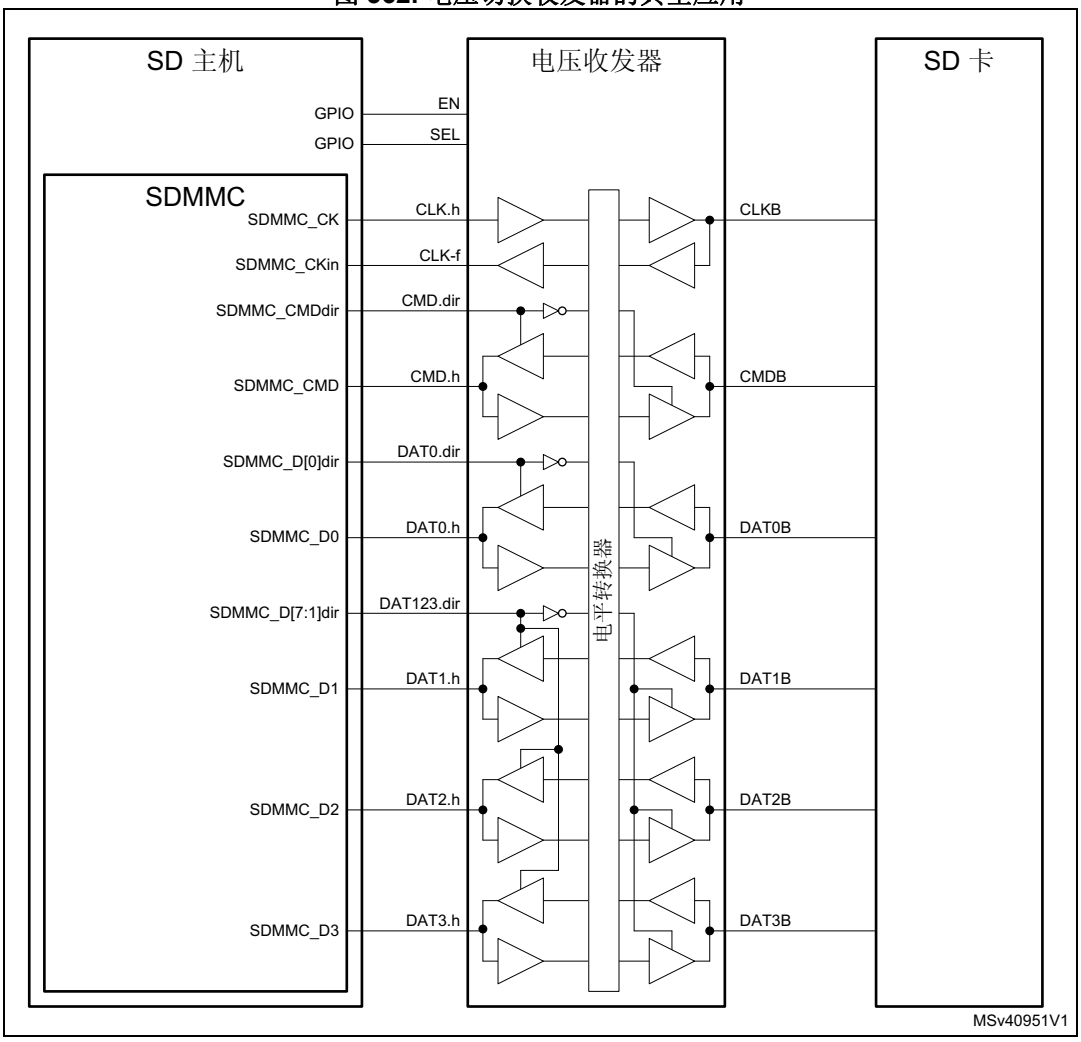
1. 在启动电压切换过程之前，应将 SDMMC_CK 频率设置在 100 kHz - 400 kHz 范围内。
2. 主机通过在发送 CMD11 之前将 VSWITCHEN 位置 1 来启动电压切换过程。
3. 卡返回 R1 响应。
   - 如果响应 CRC 通过，电压切换过程将继续执行，主机将不再驱动 CMD 和 SDMMC_D[3:0] 信号，直到电压切换序列完成。在响应后再经过若干周期，SDMMC_CK 将停止，CKSTOP 标志将置 1。
   - 如果响应 CRC 失败（CCRCFAIL 标志）或未在超时（CTIMEOUT 标志）之前收到响应，则电压切换过程将停止。
4. 在 R1 响应之后的下一个时钟，卡将 CMD 和 SDMMC_D[3:0] 驱动为低电平。
5. 收到 R1 响应后，主机可以使用 BUSYD0 寄存器位监视 SDMMC_D0 线。SDMMC_D0 线在响应后的两个 SDMMC_CK 时钟周期后进行采样。固件可以读取 CKSTOP 标志后的 BUSYD0 寄存器位。
   - 当检测到 BUSYD0 为低电平时，主机固件会将稳压器切换到 1.8V，然后通过将寄存器位 VSWITCH 置 1 来指示 SDMMC 启动电压切换序列的时序关键部分。硬件将继续停止 SDMMC_CK，方法是使 SDMMC_CK 保持低电平并持续至少 5 ms。
   - 当检测到 BUSYD0 为高电平时，主机将中止电压切换序列并对卡进行掉电再上电。
6. 检测到 SDMMC_CK 为低电平后，卡会开始将信号电压切换到 1.8V。
7. 主机 SDMMC 硬件将在至少 5 ms 后重新启动 SDMMC_CK。
8. 在检测到 SDMMC_CK 切换后的 1 ms 内，卡会将 CMD 和 DAT[3:0] 驱动为高电平并至少持续 1 个 SDMMC_CK 周期，然后停止驱动 CMD 和 DAT[3:0]。
9. 在 SDMMC_CK 重启后的 1 ms 后，主机 SDMMC 硬件会将 SDMMC_D0 采样到 BUSYD0 中并生成 VSWEND 标志。
10. 出现 VSWEND 标志时，主机将使用 BUSYD0 寄存器位检查 SDMMC_D0 线，以确认电压切换序列是否完成：
    - 当检测到 BUSYD0 为高电平时，表示电压切换成功完成。
    - 当检测到 BUSYD0 为低电平时，表示电压切换失败，主机将对卡执行掉电再上电。

停止 SDMMC_CK 所需的 5 ms 最短时间来自内部非门控 SDMMC_CK 时钟，其最大频率为 25 MHz（SD 模式），该频率由时钟分频比 CLKDIV 设置。> 5 ms 的时间将以 2^12 个周期（400 kHz 时为 10.24 ms）计数。如果通过时钟分频比 CLKDIV 选择较低的 SDMMC_CK 频率，则停止 SDMMC_CK 时钟所需的时间将会更长。

卡将 SDMMC_Dn 和 SDMMC_CMD 线驱动为高电平所需的 1 ms 最长时间来自内部非门控 SDMMC_CK 时钟，其最大频率为 25 MHz（SD 模式），该频率由时钟分频比 CLKDIV 设置。SDMMC 将在 > 1 ms 的时间后检查线路，该时间将以 2^9 个周期（25 MHz 时为 1.28 ms）计数。如果通过时钟分频比 CLKDIV 选择较低的 SDMMC_CK 频率，则检查线路所需的时间将会更长。

信号电压通过外部电压切换收发器（即 ST6G3244ME）来提供支持。

图 562. 电压切换收发器的典型应用



为了连接标准信号旁的外部驱动器（电压切换收发器），SDMMC 使用以下信号：

SDMMC_CKin 反馈输入时钟。

SDMMC_CMDdir CMD 的 I/O 方向控制信号。

SDMMC_D[0]dir SDMMC_D0 的 I/O 方向控制信号。

SDMMC_D[7:1]dir SDMMC_D1、SDMMC_D2 和 SDMMC_D3 的 I/O 方向控制信号。

电压收发器信号 **EN** 和 **SEL** 将通过通用 I/O 进行处理。

SDMMC_CMDdir、SDMMC_D[0]dir 和 SDMMC_D[7:1]dir 信号的极性可通过 SDMMC_POWER.DIRPOL 控制位进行选择。

48.8 SDMMC 寄存器

器件通过 32 位控制寄存器（可通过 AHB 从接口访问）与系统进行通信。

外设寄存器必须按字（32 位）进行访问。按字节（8 位）和半字（16 位）访问会生成 AHB 总线错误。

48.8.1 SDMMC 电源控制寄存器 (SDMMC_POWER)

SDMMC power control register

偏移地址：0x000

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.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DIRPOL	VSWITCHEN	VSWITCH	PWRCTRL[1:0]	
											rw	rw	rw	rw	rw

位 31:5 保留，必须保持复位值。

位 4 **DIRPOL**：数据和命令方向信号极性选择 (Data and command direction signals polarity selection)。
只有在 SDMMC 处于掉电状态时 (PWRCTRL = 00) 才能写入该位。
0：当方向信号为低电平时，电压收发器 IO 驱动为输出。
1：当方向信号为高电平时，电压收发器 IO 驱动为输出。

位 3 **VSWITCHEN**：电压切换过程使能 (Voltage switch procedure enable)。
只有在禁止 CPSM (CPSMEN=0) 时才能通过固件写入此位。
该位用于在电压切换命令响应后停止 SDMMC_CK：
0：SDMMC_CK 时钟在成功接收到命令响应后保持不变。
1：SDMMC_CK 时钟在成功接收到命令响应后停止。

位 2 **VSWITCH**：电压切换序列启动 (Voltage switch sequence start)。
该位用于启动电压切换序列的时序关键部分：
0：电压切换序列未启动且未激活。
1：电压切换序列已启动或已激活。

位 1:0 **PWRCTRL[1:0]**：SDMMC 状态控制位 (SDMMC state control bits)。

只有在 SDMMC 未处于上电状态时 (PWRCTRL ≠ 11) 才能写入这些位。

这些位用于定义 SDMMC 信号的功能状态：

00：复位后，复位：SDMMC 禁止，卡的时钟停止，SDMMC_D[7:0] 和 SDMMC_CMD 处于高阻态，SDMMC_CK 驱动为低电平。
写入 00 时，掉电：SDMMC 禁止，卡的时钟停止，SDMMC_D[7:0]、SDMMC_CMD 和 SDMMC_CK 驱动为高电平。

01：保留。（写入 01 时，PWRCTRL 值不会改变）

10：掉电再上电，SDMMC 禁止，卡的时钟停止，SDMMC_D[7:0]、SDMMC_CMD 和 SDMMC_CK 驱动为低电平。

11：上电：卡由时钟驱动，前 74 个 SDMMC_CK 周期内，SDMMC 仍禁止。74 个周期后，使能 SDMMC，并根据 SDMMC 操作控制 SDMMC_D[7:0]、SDMMC_CMD 和 SDMMC_CK。

任何后续写操作都将被忽略，PWRCTRL 值将保持为 11。

48.8.2 SDMMC 时钟控制寄存器 (SDMMC_CLKCR)

SDMMC clock control register

偏移地址：0x004

复位值：0x0000 0000

SDMMC_CLKCR 寄存器控制 SDMMC_CK 输出时钟、SDMMC_RX_CLK 接收时钟和总线宽度。

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.	SELCLKRX[1:0]		BUS SPEED	DDR	HWFC_EN	NEG EDGE
										rw	rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WID BUS[1:0]		Res.	PWR SAV	Res.	Res.	CLKDIV[9:0]									
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 **SELCLKRX[1:0]**：接收时钟选择 (Receive clock selection)。

只有在 CPSM 和 DPSM 未激活时（CPSMACT = 0 且 DPSMACT = 0），才能写入这些位

00：选择 SDMMC_CLKio 作为接收时钟

01：选择 SDMMC_CKin 反馈时钟作为接收时钟

10：保留

11：保留（选择 SDMMC_CLKio）

位 19 **BUSSPEED**：在 DS、HS、SDR12、SDR25 和 SDR50、DDR50 总线速度模式间进行选择 (Bus speed mode selection between DS, HS, SDR12, SDR25 and SDR50, DDR50)。

只有在 CPSM 和 DPSM 未激活时（CPSMACT = 0 且 DPSMACT = 0），才能写入该位

0：选择 DS、HS、SDR12、SDR25 总线速度模式

1：选择 SDR50、DDR50 总线速度模式

位 18 **DDR**：数据速率信号传输选择 (Data rate signaling selection)

只有在 CPSM 和 DPSM 未激活时（CPSMACT = 0 且 DPSMACT = 0），才能写入该位。
只能在 4 位或 8 位宽的总线模式下选择 DDR 速率。(WIDBUS > 00)。当 WIDBUS = 00（1 位宽总线）时，DDR = 1 不起作用。
只能在时钟分频比 > 1 的情况下选择 DDR 速率。(CLKDIV > 0)
0：SDR 单倍数据速率信号传输。
1：DDR 双倍数据速率信号传输。

位 17 **HWFC_EN**：硬件流控制使能 (Hardware flow control enable)

只有在 CPSM 和 DPSM 未激活时（CPSMACT = 0 且 DPSMACT = 0），才能写入该位。
0：禁止硬件流控制
1：使能硬件流控制
如果使能硬件流控制，TXFIFOE 和 RXFIFOF 标志的含义将发生变化，请参见*第 48.8.11 节*中的 SDMMC 状态寄存器定义。

位 16 **NEGEDGE**：数据和命令的 SDMMC_CK 移相选择位 (SDMMC_CK dephasing selection bit for data and Command)。

只有在 CPSM 和 DPSM 未激活时（CPSMACT = 0 且 DPSMACT = 0），才能写入该位。
当时钟分频比 = 1 (CLKDIV = 0) 时，该位不起作用。数据和命令在 SDMMC_CK 下降沿发生变化。
当时钟分频比 > 1 (CLKDIV > 0) 且 DDR = 0 时：
0： - 命令和数据在 SDMMC_CK 上升沿后的 SDMMCCLK 下降沿发生变化。
- SDMMC_CK 边沿出现在 SDMMCCLK 上升沿。
1： - 命令和数据在生成 SDMMC_CK 下降沿的同一 SDMMCCLK 上升沿发生变化。
当时钟分频比 > 1 (CLKDIV > 0) 且 DDR = 1 时：
0： - 命令在 SDMMC_CK 上升沿后的 SDMMCCLK 下降沿发生变化。
- 数据在 SDMMC_CK 边沿后的 SDMMCCLK 下降沿发生变化。
- SDMMC_CK 边沿出现在 SDMMCCLK 上升沿。
1： - 命令在生成 SDMMC_CK 下降沿的同一 SDMMCCLK 上升沿发生变化。
- 数据在 SDMMC_CK 边沿后的 SDMMC_CK 下降沿发生变化。
- SDMMC_CK 边沿出现在 SDMMCCLK 上升沿。

位 15:14 **WIDBUS[1:0]**：宽总线模式使能位 (Wide bus mode enable bit)

只有在 CPSM 和 DPSM 未激活时（CPSMACT = 0 且 DPSMACT = 0），才能写入该位。
00：默认 1 位宽总线模式：使用 SDMMC_D0（不支持 DDR）
01：4 位宽总线模式：使用 SDMMC_D[3:0]
10：8 位宽总线模式：使用 SDMMC_D[7:0]

位 13 保留，必须保持复位值。

位 12 **PWRSAV**：节能模式配置位 (Power saving configuration bit)

只有在 CPSM 和 DPSM 未激活时（CPSMACT = 0 且 DPSMACT = 0），才能写入该位。
要实现节能模式，可在总线空闲时通过将 PWRSAV 置 1 来禁止 SDMMC_CK 时钟输出：
0：始终使能 SDMMC_CK 时钟
1：仅在总线激活时使能 SDMMC_CK

位 11:10 保留，必须保持复位值。

位 9:0 **CLKDIV[9:0]**：时钟分频系数 (Clock divide factor)

只有在 CPSM 和 DPSM 未激活时（CPSMACT = 0 且 DPSMACT = 0），才能写入该位。
该位域定义输入时钟 (SDMMCCLK) 与输出时钟 (SDMMC_CK) 之间的分频系数：SDMMC_CK 频率 = SDMMCCLK / [2 * CLKDIV]。
0x000：SDMMC_CK 频率 = SDMMCCLK / 1（不支持 DDR）
0x001：SDMMC_CK 频率 = SDMMCCLK / 2
0x002：SDMMC_CK 频率 = SDMMCCLK / 4
0x0XX：...
0x080：SDMMC_CK 频率 = SDMMCCLK / 256
0xXXX：...
0x3FF：SDMMC_CK 频率 = SDMMCCLK / 2046

注： *1 当 SD/SDIO 卡或 MMC 处于识别模式时，SDMMC_CK 频率必须小于 400 kHz。*

2 如果为所有卡分配相对卡地址，则可以将时钟频率更改为卡总线最大频率。

3 此寄存器的两次写访问至少需要相隔七个 HCLK 时钟周期。在 SD I/O 卡的读取等待间隔期间也可以停止 SDMMC_CK；在此情况下，SDMMC_CLKCR 寄存器不对 SDMMC_CK 进行控制。

48.8.3 SDMMC 参数寄存器 (SDMMC_ARGR)

SDMMC argument register

偏移地址：0x008

复位值：0x0000 0000

SDMMC_ARGR 寄存器包含一个 32 位命令参数，该参数作为命令消息的一部分发送到卡。

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

位 31:0 **CMDARG[31:0]**：命令参数 (Command argument)。

只有在禁止 CPSM (CPSMEN=0) 时才能通过固件写入这些位。
作为命令消息的一部分发送给卡的命令参数。如果命令包含参数，则在将命令写入到命令寄存器之前，必须将参数加载到此寄存器中。

48.8.4 SDMMC 命令寄存器 (SDMMC_CMDR)

SDMMC command register

偏移地址：0x00C

复位值：0x0000 0000

SDMMC_CMDR 寄存器包含命令索引和命令类型位。命令索引作为命令消息的一部分而发送给卡。命令类型位控制命令路径状态机 (CPSM)。

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.	CMD SUS PEND
															rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BOOT EN	BOOT MODE	DT HOLD	CPSM EN	WAITP END	WAIT INT	WAITRESP[1:0]		CMD STOP	CMD TRANS	CMDINDEX[5:0]					
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 **CMDSUSPEND**：CPSM 将命令视为挂起或恢复命令并发出中断周期开始/结束信号 (The CPSM treats the command as a Suspend or Resume command and signals interrupt period start/end)。
只有在禁止 CPSM (CPSMEN=0) 时才能通过固件写入此位。
CMDSUSPEND = 1 且 CMDTRANS = 0 时为挂起命令，响应位 BS = 0 时开始中断周期。
CMDSUSPEND = 1 且 CMDTRANS = 1 时为含数据的恢复命令，响应位 DF = 1 时结束中断周期。

位 15 **BOOTEN**：使能启动模式过程 (Enable boot mode procedure)。
0：禁止启动模式过程
1：使能启动模式过程

位 14 **BOOTMODE**：选择要使用的启动模式过程 (Select the boot mode procedure to be used)。
只有在禁止 CPSM (CPSMEN = 0) 时才能写入该位
0：选择正常启动模式过程
1：选择备用启动模式过程

位 13 **DTHOLD**：在 DPSM 中保持新的数据块发送和接收 (Hold new data block transmission and reception in the DPSM)。
如果该位置 1，DPSM 将不会从 Wait_S 状态变为发送状态或从 Wait_R 状态变为接收状态。

位 12 **CPSMEN**：命令路径状态机 (CPSM) 使能位 (Command path state machine (CPSM) Enable bit)
该位由固件写 1，并在 CPSM 进入空闲状态时由硬件清零。
如果此位置 1，则使能 CPSM。
当 DTEN = 1 时，不会传输命令，也不会开始启动过程。CPSMEN 将清零。
在读取等待期间，当 SDMMC_CK 停止时，不会发送任何命令，CPSMEN 保持为 0。

位 11 **WAITPEND**：CPSM 等待来自 DPSM 的数据传输结束（CmdPend 内部信号）(CPSM Waits for end of data transfer (CmdPend internal signal) from DPSM)。
如果此位置 1，则 CPSM 将等到数据传输结束触发信号后才开始发送命令。
仅当 DTMODE = MMC 流数据传输、WIDBUS = 1 位宽总线模式、DPSMACT = 1 且 DTDIR = 从主机到卡时，才考虑 WAITPEND。

位 10 **WAITINT**：CPSM 等待中断请求 (CPSM waits for interrupt request)。
如果此位置 1，则 CPSM 禁止命令超时并等待卡中断请求（响应）。
如果该位在 CPSM 等待状态下清零，将导致中断模式中止。

位 9:8 **WAITRESP[1:0]**：等待响应位 (Wait for response bits)。
只有在禁止 CPSM (CPSMEN=0) 时才能通过固件写入此位。
这些位用于配置 CPSM 是否等待响应，如果等待，将等待哪种类型的响应。
00：无响应，将出现 CMDSENT 标志
01：短响应，将出现 CMDREND 或 CCRCFAIL 标志
10：短响应，将出现 CMDREND 标志（无 CRC）
11：长响应，将出现 CMDREND 或 CCRCFAIL 标志

位 7 **CMDSTOP**：CPSM 将命令视为停止传输命令并向 DPSM 发出中止信号 (The CPSM treats the command as a Stop Transmission command and signals Abort to the DPSM)。
只有在禁止 CPSM (CPSMEN=0) 时才能通过固件写入此位。
如果该位置 1，当发送命令时，CPSM 将向 DPSM 发出中止信号。

位 6 **CMDTRANS**：CPSM 将命令视为数据传输命令、停止中断周期并向 DPSM 发出数据使能信号 (The CPSM treats the command as a data transfer command, stops the interrupt period, and signals DataEnable to the DPSM)
只有在禁止 CPSM (CPSMEN=0) 时才能通过固件写入此位。
如果该位置 1，当发送命令时，CPSM 将发出中断周期结束信号并向 DPSM 发出数据使能信号。

位 5:0 **CMDINDEX[5:0]**：命令索引 (Command index)。
只有在禁止 CPSM (CPSMEN=0) 时才能通过固件写入此位。
命令索引作为命令消息的一部分发送给卡。

注： 1 此寄存器的两次写访问至少需要相隔七个 HCLK 时钟周期。

2 多媒体卡可以发送两种类型的响应：短响应（48 位）或长响应（136 位）。SD 卡和 SD I/O 卡则只能发送短响应，参数因响应类型而异：软件将根据发送的命令区分响应类型。

48.8.5 SDMMC 命令响应寄存器 (SDMMC_RESPCMDR)

SDMMC command response register

偏移地址：0x010

复位值：0x0000 0000

SDMMC_RESPCMDR 寄存器包含接收到的最后一个命令响应的命令索引位域。如果命令响应传输不包含命令索引位域（长响应或 OCR 响应），则 RESPCMD 位域为未知，但该位域必须包含 111111b（响应中保留位域的值）。

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.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RESPCMD[5:0]					
										r	r	r	r	r	r

位 31:6 保留，必须保持复位值。

位 5:0 **RESPCMD[5:0]**：响应命令索引 (Response command index)
只读位域。包含接收到的最后一个命令响应的命令索引。

48.8.6 SDMMC 响应 x 寄存器 (SDMMC_RESPxR)

SDMMC response x register

偏移地址：0x010 + 0x004 * x，（x = 1 到 4）

复位值：0x0000 0000

SDMMC_RESP1/2/3/4R 寄存器包含卡的状态，该状态来自接收的响应。

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

位 31:0 **CARDSTATUSx[31:0]**：请参见*表 310*。

卡状态为 32 位或 128 位，具体取决于响应类型。

表 310. 响应类型和 SDMMC_RESPxR 寄存器

寄存器(1)	短响应	长响应
SDMMC_RESP1R	卡状态 [31:0]	卡状态 [127:96]
SDMMC_RESP2R	全 0	卡状态 [95:64]
SDMMC_RESP3R	全 0	卡状态 [63:32]
SDMMC_RESP4R	全 0	卡状态 [31:0](2)

1. 首先接收卡状态的最高有效位。
2. SDMMC_RESP4R 寄存器 LSB 始终为 0。

48.8.7 SDMMC 数据定时器寄存器 (SDMMC_DTIMER)

SDMMC data timer register

偏移地址：0x024

复位值：0x0000 0000

SDMMC_DTIMER 寄存器包含以卡总线时钟周期表示的数据超时周期。

计数器加载 SDMMC_DTIMER 寄存器的值，并在数据路径状态机进入 Wait_R 或繁忙状态后开始递减。如果定时器达到 0 而 DPSM 处于上述一种状态，则超时状态标志置 1。

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

位 31:0 **DATATIME[31:0]**：数据和 R1b 繁忙超时周期 (Data and R1b busy timeout period)

只有在 CPSM 和 DPSM 未激活时（CPSMACT = 0 且 DPSMACT = 0），才能写入该位。

以卡总线时钟周期表示的数据和 R1b 繁忙超时周期。

注： *数据传输必须首先写入到数据计时器寄存器和数据长度寄存器，然后才写入到数据控制寄存器。*

48.8.8 SDMMC 数据长度寄存器 (SDMMC_DLENR)

SDMMC data length register

偏移地址：0x028

复位值：0x0000 0000

SDMMC_DLENR 寄存器包含要传输的数据字节数。当数据传输开始时，值将加载到数据计数器中。

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

位 31:25 保留，必须保持复位值。

位 24:0 **DATALENGTH[24:0]**：数据长度值 (Data length value)

只有在未激活 DPSM (DPSMACT = 0) 时才能通过固件写入此寄存器。

要传输的数据字节数量。

当 DDR = 1 时，DATALENGTH 被截断为 2 的倍数。（不传输最后一个奇数字节）

当 DATALENGTH = 0 时，不会传输任何数据，当通过 CPSMEN 和 CMDTRANS = 1 请求时，也不会传输任何命令。DTEN 和 CPSMEN 均清零。

注： *对于块数据传输，数据长度寄存器中的值必须是块大小的倍数（请参见 SDMMC_DCTRL）。数据传输必须首先写入到数据计时器寄存器和数据长度寄存器，然后才写入到数据控制寄存器。*

对于 SDMMC 多字节传输，数据长度寄存器中的值必须在 1 到 512 之间。

48.8.9 SDMMC 数据控制寄存器 (SDMMC_DCTRL)

SDMMC data control register

偏移地址：0x02C

复位值：0x0000 0000

SDMMC_DCTRL 寄存器控制数据路径状态机 (DPSM)。

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.	FIFO RST	BOOT ACKE N	SDIO EN	RW MOD	RW STOP	RW START	DBLOCKSIZE[3:0]				DTMODE[1:0]		DTDIR	DTEN
		rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:14 保留，必须保持复位值。

位 13 **FIFORST**：FIFO 复位，将清空任何剩余的数据 (FIFO reset, will flush any remaining data)。
只有在 IDMAEN = 0 且未激活 DPSM (DPSMACT = 1) 时才能通过固件写入该位。只有在发生传输错误或传输保持时，该位才会生效。
0：FIFO 不受影响。
1：清空任何剩余的数据并复位 FIFO 指针。当 DPSM 变为非激活状态 (DPSMACT = 0) 时，该位由硬件自动清零。

位 12 **BOOTACKEN**：使能启动确认接收 (Enable the reception of the boot acknowledgment)。
只有在未激活 DPSM (DPSMACT = 0) 时才能通过固件写入此位。
0：禁止启动确认，将不会接收
1：使能启动确认，将会接收

位 11 **SDIOEN**：SD I/O 中断使能功能 (SD I/O interrupt enable functions)
只有在未激活 DPSM (DPSMACT = 0) 时才能通过固件写入此位。
如果将该位置 1，则 DPSM 使能特定于 SD I/O 卡的中断操作。

位 10 **RWMOD**：读取等待模式 (Read wait mode)。
只有在未激活 DPSM (DPSMACT = 0) 时才能通过固件写入此位。
0：使用 SDMMC_D2 进行读取等待控制
1：通过停止 SDMMC_CK 进行读取等待控制

位 9 **RWSTOP**：读取等待停止 (Read wait stop)
该位由固件写入，并在 DPSM 从 READ_WAIT 状态变为 WAIT_R 或空闲状态时由硬件自动清零。
0：无读取等待停止。
1：当 DPSM 处于 READ_WAIT 状态时，使能读取等待停止。

位 8 **RWSTART**：读取等待开始 (Read wait start)。
如果将该位置 1，则读取等待操作开始。

位 7:4 **DBLOCKSIZE[3:0]**：数据块大小 (Data block size)

只有在未激活 DPSM (DPSMACT = 0) 时才能通过固件写入此位。

定义在选择了块数据传输模式时数据块的长度：

0000：（十进制数 0）块长度 = 2^0 = 1 字节

0001：（十进制数 1）块长度 = 2^1 = 2 字节

0010：（十进制数 2）块长度 = 2^2 = 4 字节

0011：（十进制数 3）块长度 = 2^3 = 8 字节

0100：（十进制数 4）块长度 = 2^4 = 16 字节

0101：（十进制数 5）块长度 = 2^5 = 32 字节

0110：（十进制数 6）块长度 = 2^6 = 64 字节

0111：（十进制数 7）块长度 = 2^7 = 128 字节

1000：（十进制数 8）块长度 = 2^8 = 256 字节

1001：（十进制数 9）块长度 = 2^9 = 512 字节

1010：（十进制数 10）块长度 = 2^{10} = 1024 字节

1011：（十进制数 11）块长度 = 2^{11} = 2048 字节

1100：（十进制数 12）块长度 = 2^{12} = 4096 字节

1101：（十进制数 13）块长度 = 2^{13} = 8192 字节

1110：（十进制数 14）块长度 = 2^{14} = 16384 字节

1111：（十进制数 15）保留

当 DATALENGTH 不是 DBLOCKSIZE 的倍数时，传输的数据将截断为 DBLOCKSIZE 的倍数。（将不会传输任何剩余数据。）

当 DDR = 1 时，不应使用 DBLOCKSIZE = 0000。（将不会传输任何数据）

位 3:2 **DTMODE[1:0]**：数据传输模式选择 (Data transfer mode selection)。

只有在未激活 DPSM (DPSMACT = 0) 时才能通过固件写入此位。

00：按块数结束的块数据传输。

01：SDIO 多字节数据传输。

10：MMC 流数据传输。（WIDBUS 应选择 1 位宽总线模式）

11：以 STOP_TRANSMISSION 命令结束的块数据传输（不与 DTEN 启动的数据传输一起使用）。

位 1 **DTDIR**：数据传输方向选择 (Data transfer direction selection)

只有在未激活 DPSM (DPSMACT = 0) 时才能通过固件写入此位。

0：从主机到卡。

1：从卡到主机。

位 0 **DTEN**：数据传输使能位 (Data transfer enable bit)

只有在未激活 DPSM (DPSMACT = 0) 时才能通过固件写入此位。当数据传输完成时，该位由硬件清零。

该位仅在不使用相关数据传输命令时才用于传输数据，即不能与 SD 或 eMMC 卡一起使用。

0：不使用 CPSM 数据传输命令时不启动数据传输。

1：不使用 CPSM 数据传输命令时启动数据传输。

48.8.10 SDMMC 数据计数器寄存器 (SDMMC_DCNTR)

SDMMC data counter register

偏移地址：0x030

复位值：0x0000 0000

当 DPSM 从空闲状态变为 Wait_R 或 Wait_S 状态时，SDMMC_DCNTR 寄存器将从数据长度寄存器中加载值（请参见 SDMMC_DLENR）。在传输数据时，计数器将值递减直至计数器达到 0。然后 DPSM 将变为空闲状态，并且在无错误时将数据状态结束标志 (DATAEND) 置 1。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	DATACOUNT[24:16]								
							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
DATACOUNT[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:25 保留，必须保持复位值。

位 24:0 **DATACOUNT[24:0]**：数据计数值 (Data count value)

读取时，将返回要传输的剩余数据字节数。写入没有任何效果。

注： *仅当数据传输完成或保持后才应读取该寄存器。在错误事件后读取时，读取的数据计数值可能与传输的实际数据字节数不同。*

48.8.11 SDMMC 状态寄存器 (SDMMC_STAR)

SDMMC status register

偏移地址：0x034

复位值：0x0000 0000

SDMMC_STAR 寄存器是一个只读寄存器。它包含两种类型的标志：

- 静态标志（位 [29,21,11:0]）：在通过写入到 SDMMC 中断清零寄存器来清零这些位之前，会一直保持这些位（请参见 SDMMC_ICR）。
- 动态标志（位 [20:12]）：这些位根据底层逻辑的状态来更改状态（例如，随着数据写入到 FIFO，发出和停止发出 FIFO 满和空标志）。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	IDMABTC	IDMATE	CKSTOP	VSWEND	ACKTIMEOUT	ACKFAIL	SDIOIT	BUSYD0END	BUSYD0	RXFIFOE	TXFIFOE	RXFIFOF	TXFIFOF
			r	r	r	r	r	r	r	r	r	r	r	r	r

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RXFIFOHF	TXFIFOHE	CPSMACT	DPSMACT	DABORT	DBCKEND	DHOLD	DATAEND	CMDSENT	CMDREND	RXOVERR	TXUNDERR	DTIMEOUT	CTIMEOUT	DCRCFAIL	CCRCFAIL
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:29 保留，必须保持复位值。

位 28 **IDMABTC**：IDMA 缓冲区传输完成 (IDMA buffer transfer complete)。通过写入 SDMMC_ICR 中的相应中断清除位来清除中断标志。

位 27 **IDMATE**：IDMA 传输错误 (IDMA transfer error)。通过写入 SDMMC_ICR 中的相应中断清除位来清除中断标志。

位 26 **CKSTOP**：SDMMC_CK 在电压切换过程中停止 (SDMMC_CK stopped in Voltage switch procedure)。通过写入 SDMMC_ICR 中的相应中断清除位来清除中断标志。

位 25 **VSWEND**：电压切换时序关键部分完成 (Voltage switch critical timing section completion)。通过写入 SDMMC_ICR 中的相应中断清除位来清除中断标志。

位 24 **ACKTIMEOUT**：启动确认超时 (Boot acknowledgment timeout)。通过写入 SDMMC_ICR 中的相应中断清除位来清除中断标志。

位 23 **ACKFAIL**：接收到启动确认（启动确认校验失败）(Boot acknowledgment received (boot acknowledgment check fail))。通过写入 SDMMC_ICR 中的相应中断清除位来清除中断标志。

位 22 **SDIOIT**：收到了 SDIO 中断 (SDIO interrupt received)。通过写入 SDMMC_ICR 中的相应中断清除位来清除中断标志。

位 21 **BUSYD0END**：检测到 CMD 响应后 SDMMC_D0 繁忙状态结束 (end of SDMMC_D0 Busy following a CMD response detected)。

该位仅指示 CMD 响应后繁忙状态结束，而不会指示因数据传输引起的繁忙状态。通过写入 SDMMC_ICR 中的相应中断清除位来清除中断标志。

0：卡 SDMMC_D0 信号未从繁忙状态变为不忙状态。

1：卡 SDMMC_D0 信号从繁忙状态变为不忙状态。

位 20 **BUSYD0**：SDMMC_D0 线的反转值（繁忙）(Inverted value of SDMMC_D0 line (Busy))，在 CMD 响应结束时进行采样，并在 CMD 响应后的 2 个 SDMMC_CK 周期后进行二次采样。

当 SDMMCD0 线从繁忙状态变为不忙状态时，该位复位为不忙状态。该位不会指示因数据传输引起的繁忙状态。它只是一个硬件状态标志，不会生成中断。

0：卡在 SDMMC_D0 上发出不忙信号。

1：卡在 SDMMC_D0 上发出繁忙信号。

位 19 **RXFIFOE**：接收 FIFO 为空 (Receive FIFO empty)

它只是一个硬件状态标志，不会生成中断。该位在一个 FIFO 单元已满时清零。

位 18 **TXFIFOE**：发送 FIFO 为空 (Transmit FIFO empty)

该位在一个 FIFO 单元已满时清零。

位 17 **RXFIFOF**：接收 FIFO 已满 (Receive FIFO full)

该位在一个 FIFO 单元为空时清零。

位 16 **TXFIFOF**：发送 FIFO 已满 (Transmit FIFO full)

它只是一个硬件状态标志，不会生成中断。该位在一个 FIFO 单元为空时清零。

位 15 **RXFIFOHF**：接收 FIFO 半满 (Receive FIFO half full)

FIFO 中至少有一半字数。该位在 FIFO 为“半 + 1”空时清零。

位 14 **TXFIFOHE**：发送 FIFO 半空 (Transmit FIFO half empty)

FIFO 中至少可写入一半字数。该位在 FIFO 为“半 + 1”满时清零。

位 13 **CPSMACT**：命令路径状态机激活，即不处于空闲状态 (Command path state machine active, i.e. not in Idle state)。

它只是一个硬件状态标志，不会生成中断。

位 12 **DPSMACT**：数据路径状态机激活，即不处于空闲状态 (Data path state machine active, i.e. not in Idle state)。

它只是一个硬件状态标志，不会生成中断。

位 11 **DABORT**：数据传输由 CMD12 中止 (Data transfer aborted by CMD12)。

通过写入 SDMMC_ICR 中的相应中断清除位来清除中断标志。

位 10 **DBCKEND**：已发送/接收数据块 (Data block sent/received)。

（CRC 校验已通过）且 DPSM 变为读取等待状态。通过写入 SDMMC_ICR 中的相应中断清除位来清除中断标志。

位 9 **DHOLD**：数据传输保持 (Data transfer Hold)。

通过写入 SDMMC_ICR 中的相应中断清除位来清除中断标志。

位 8 **DATAEND**：数据传输正确结束 (Data transfer ended correctly)。

（数据计数器，DATACOUNT 为零，未出现错误）。通过写入 SDMMC_ICR 中的相应中断清除位来清除中断标志。

位 7 **CMDSENT**：命令已发送（不需要响应）(Command sent (no response required))

通过写入 SDMMC_ICR 中的相应中断清除位来清除中断标志。

位 6 **CMDREND**：已接收命令响应（CRC 校验通过或无 CRC）(Command response received (CRC check passed, or no CRC))。

通过写入 SDMMC_ICR 中的相应中断清除位来清除中断标志。

位 5 **RXOVERR**：已接收 FIFO 上溢错误或 IDMA 写入传输错误 (Received FIFO overrun error or IDMA write transfer error)。

通过写入 SDMMC_ICR 中的相应中断清除位来清除中断标志。

位 4 **TXUNDERR**：发送 FIFO 下溢错误或 IDMA 读取传输错误 (Transmit FIFO underrun error or IDMA read transfer error)。
通过写入 SDMMC_ICR 中的相应中断清除位来清除中断标志。

位 3 **DTIMEOUT**：数据超时 (Data timeout)。
通过写入 SDMMC_ICR 中的相应中断清除位来清除中断标志。

位 2 **CTIMEOUT**：命令响应超时 (Command response timeout)。
通过写入 SDMMC_ICR 中的相应中断清除位来清除中断标志。
命令超时周期为固定值 64 个 SDMMC_CK 时钟周期。

位 1 **DCRCFAIL**：已发送/接收数据块（CRC 校验失败）(Data block sent/received (CRC check failed))。
通过写入 SDMMC_ICR 中的相应中断清除位来清除中断标志。

位 0 **CCRCFAIL**：已接收命令响应（CRC 校验失败）(Command response received (CRC check failed))。
通过写入 SDMMC_ICR 中的相应中断清除位来清除中断标志。

注： *使用 IDMA 模式时，SDMMC_MASKR 中应屏蔽 FIFO 中断标志。*

48.8.12 SDMMC 中断清零寄存器 (SDMMC_ICR)

SDMMC interrupt clear register

偏移地址：0x038

复位值：0x0000 0000

SDMMC_ICR 寄存器是一个只写寄存器。向某个位写入 1 会将 SDMMC_STAR 状态寄存器中的对应位清零。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	IDMABTCC	IDMATEC	CKSTOPC	VSWENDC	ACKTIMEOUTC	ACKFAILC	SDIOITC	BUSYD0ENDC	Res.	Res.	Res.	Res.	Res.
			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.	Res.	DABORTC	DBCKENDC	DHOLDC	DATAENDC	CMDSENTC	CMDRENDC	RXOVERRC	TXUNDERRC	DTIMEOUTC	CTIMEOUTC	DCRCFAILC	CCRCFAILC
				rw	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 **IDMABTCC**：IDMA 缓冲区传输完成清零位 (IDMA buffer transfer complete clear bit)
由软件置 1，用于将 IDMABTC 标志清零。
0：未将 IDMABTC 清零
1：已将 IDMABTC 清零

位 27 **IDMATEC**：IDMA 缓冲区错误清零位 (IDMA transfer error clear bit)
由软件置 1，用于将 IDMATE 标志清零。
0：未将 IDMATE 清零
1：已将 IDMATE 清零

位 26 **CKSTOPC**：CKSTOP 标志清零位 (CKSTOP flag clear bit)
由软件置 1，用于将 CKSTOP 标志清零。
0：未将 CKSTOP 清零
1：已将 CKSTOP 清零

位 25 **VSWENDC**：VSWEND 标志清零位 (VSWEND flag clear bit)
由软件置 1，用于将 VSWEND 标志清零。
0：未将 VSWEND 清零
1：已将 VSWEND 清零

位 24 **ACKTIMEOUTC**：ACKTIMEOUT 标志清零位 (ACKTIMEOUT flag clear bit)
由软件置 1，用于将 ACKTIMEOUT 标志清零。
0：未将 ACKTIMEOUT 清零
1：已将 ACKTIMEOUT 清零

位 23 **ACKFAILC**：ACKFAIL 标志清零位 (ACKFAIL flag clear bit)
由软件置 1，用于将 ACKFAIL 标志清零。
0：未将 ACKFAIL 清零
1：已将 ACKFAIL 清零

位 22 **SDIOITC**：SDIOIT 标志清零位 (SDIOIT flag clear bit)
由软件置 1，用于将 SDIOIT 标志清零。
0：未将 SDIOIT 清零
1：已将 SDIOIT 清零

位 21 **BUSYD0ENDC**：BUSYD0END 标志清零位 (BUSYD0END flag clear bit)
由软件置 1，用于将 BUSYD0END 标志清零。
0：未将 BUSYD0END 清零
1：已将 BUSYD0END 清零

位 20:12 保留，必须保持复位值。

位 11 **DABORTC**：DABORT 标志清零位 (DABORT flag clear bit)
由软件置 1，用于将 DABORT 标志清零。
0：未将 DABORT 清零
1：已将 DABORT 清零

位 10 **DBCKENDC**：DBCKEND 标志清零位 (DBCKEND flag clear bit)
由软件置 1，用于将 DBCKEND 标志清零。
0：未将 DBCKEND 清零
1：已将 DBCKEND 清零

位 9 **DHOLDC**：DHOLD 标志清零位 (DHOLD flag clear bit)
由软件置 1，用于将 DHOLD 标志清零。
0：未将 DHOLD 清零
1：已将 DHOLD 清零

位 8 **DATAENDC**：DATAEND 标志清零位 (DATAEND flag clear bit)
由软件置 1，用于将 DATAEND 标志清零。
0：未将 DATAEND 清零
1：已将 DATAEND 清零

位 7 **CMDSENTC**：CMDSENT 标志清零位 (CMDSENT flag clear bit)
由软件置 1，用于将 CMDSENT 标志清零。
0：未将 CMDSENT 清零
1：已将 CMDSENT 清零

位 6 **CMDRENDC**：CMDREND 标志清零位 (CMDREND flag clear bit)
由软件置 1，用于将 CMDREND 标志清零。
0：未将 CMDREND 清零
1：已将 CMDREND 清零

位 5 **RXOVERRC**：RXOVERR 标志清零位 (RXOVERR flag clear bit)
由软件置 1，用于将 RXOVERR 标志清零。
0：未将 RXOVERR 清零
1：已将 RXOVERR 清零

位 4 **TXUNDERRC**：TXUNDERR 标志清零位 (TXUNDERR flag clear bit)
由软件置 1，用于将 TXUNDERR 标志清零。
0：未将 TXUNDERR 清零
1：已将 TXUNDERR 清零

位 3 **DTIMEOUTC**：DTIMEOUT 标志清零位 (DTIMEOUT flag clear bit)
由软件置 1，用于将 DTIMEOUT 标志清零。
0：未将 DTIMEOUT 清零
1：已将 DTIMEOUT 清零

位 2 **CTIMEOUTC**：CTIMEOUT 标志清零位 (CTIMEOUT flag clear bit)
由软件置 1，用于将 CTIMEOUT 标志清零。
0：未将 CTIMEOUT 清零
1：已将 CTIMEOUT 清零

位 1 **DCRCFAILC**：DCRCFAIL 标志清零位 (DCRCFAIL flag clear bit)
由软件置 1，用于将 DCRCFAIL 标志清零。
0：未将 DCRCFAIL 清零
1：已将 DCRCFAIL 清零

位 0 **CCRCFAILC**：CCRCFAIL 标志清零位 (CCRCFAIL flag clear bit)
由软件置 1，用于将 CCRCFAIL 标志清零。
0：未将 CCRCFAIL 清零
1：已将 CCRCFAIL 清零

48.8.13 SDMMC 屏蔽寄存器 (SDMMC_MASKR)

SDMMC mask register

偏移地址：0x03C

复位值：0x0000 0000

中断屏蔽寄存器通过将对应的位置 1 来确定哪一个状态标志位可以产生中断请求。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	IDMABTCIE	Res.	CKSTOPIE	VSWENDIE	ACKTIMEOUTIE	ACKFAILIE	SDIOITIE	BUSYD0ENDIE	Res.	Res.	TXFIFOEIE	RXFIFOFIE	Res.
			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
RXFIFOHFIE	TXFIFOHEIE	Res.	Res.	DABORTIE	DBCKENDIE	DHOLDIE	DATAENDIE	CMDSENTIE	CMDRENDIE	RXOVERRIE	TXUNDERRIE	DTIMEOUTIE	CTIMEOUTIE	DCRCFAILIE	CCRCFAILIE
rw	rw			rw	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 **IDMABTCIE**：IDMA 缓冲区传输完成中断使能 (IDMA buffer transfer complete interrupt enable)
由软件置 1 和清零，用于使能/禁止 IDMA 传输完存储器缓冲区内的所有数据时生成的中断。
0：禁止 IDMA 缓冲区传输完成中断
1：使能 IDMA 缓冲区传输完成中断

位 27 保留，必须保持复位值。

位 26 **CKSTOPIE**：电压切换时钟停止中断使能 (Voltage Switch clock stopped interrupt enable)
由软件置 1 和清零，用于使能/禁止由电压切换时钟停止引起的中断。
0：禁止电压切换时钟停止中断
1：使能电压切换时钟停止中断

位 25 **VSWENDIE**：电压切换时序关键部分完成中断使能 (Voltage switch critical timing section completion interrupt enable)
由软件置 1 和清零，用于使能/禁止电压切换时序关键部分完成时生成的中断。
0：禁止电压切换时序关键部分完成中断
1：使能电压切换时序关键部分完成中断

位 24 **ACKTIMEOUTIE**：确认超时中断使能 (Acknowledgment timeout interrupt enable)
由软件置 1 和清零，用于使能/禁止由确认超时引起的中断。
0：禁止确认超时中断
1：使能确认超时中断

位 23 **ACKFAILIE**：确认失败中断使能 (Acknowledgment Fail interrupt enable)
由软件置 1 和清零，用于使能/禁止由确认失败引起的中断。
0：禁止确认失败中断
1：使能确认失败中断

位 22 **SDIOITIE**：接收到 SDIO 模式中断时中断使能 (SDIO mode interrupt received interrupt enable)
由软件置 1 和清零，用于使能/禁止收到 SDIO 模式中断时生成中断。
0：禁止接收到 SDIO 模式中断时中断
1：使能接收到 SDIO 模式中断时中断

位 21 **BUSYD0ENDIE**：BUSYD0END 中断使能 (BUSYD0END interrupt enable)
由软件置 1 和清零，用于使能/禁止 CMD 响应后 SDMMC_D0 信号从繁忙状态变为不忙状态时生成的中断。
0：禁止 BUSYD0END 中断
1：使能 BUSYD0END 中断

位 20:19 保留，必须保持复位值。

位 18 **TXFIFOEIE**：Tx FIFO 为空时中断使能 (Tx FIFO empty interrupt enable)
由软件置 1 和清零，用于使能/禁止由 Tx FIFO 为空引起的中断。
0：禁止 Tx FIFO 为空时中断
1：使能 Tx FIFO 为空时中断

位 17 **RXFIFOFIE**：Rx FIFO 变满时中断使能 (Rx FIFO full interrupt enable)
由软件置 1 和清零，用于使能/禁止由 Rx FIFO 变满引起的中断。
0：禁止 Rx FIFO 变满时中断
1：使能 Rx FIFO 变满时中断

位 16 保留，必须保持复位值。

位 15 **RXFIFOHFIE**：Rx FIFO 半满时中断使能 (Rx FIFO half full interrupt enable)
由软件置 1 和清零，用于使能/禁止由 Rx FIFO 半满引起的中断。
0：禁止 Rx FIFO 半满时中断
1：使能 Rx FIFO 半满时中断

位 14 **TXFIFOHEIE**：Tx FIFO 半空时中断使能 (Tx FIFO half empty interrupt enable)
由软件置 1 和清零，用于使能/禁止由 Tx FIFO 半空引起的中断。
0：禁止 Tx FIFO 半空时中断
1：使能 Tx FIFO 半空时中断

位 13:12 保留，必须保持复位值。

位 11 **DABORTIE**：数据传输中止中断使能 (Data transfer aborted interrupt enable)
由软件置 1 和清零，用于使能/禁止由数据传输中止引起的中断。
0：禁止数据传输中止中断
1：使能数据传输中止中断

位 10 **DBCKENDIE**：数据块结束中断使能 (Data block end interrupt enable)
由软件置 1 和清零，用于使能/禁止由数据块结束引起的中断。
0：禁止数据块结束中断
1：使能数据块结束中断

位 9 **DHOLDIE**：数据保持中断使能 (Data hold interrupt enable)
由软件置 1 和清零，用于使能/禁止发送过程中新数据保持 DPSM Wait_S 状态时生成的中断。
0：禁止数据保持中断
1：使能数据保持中断

位 8 **DATAENDIE**：数据结束中断使能 (Data end interrupt enable)
由软件置 1 和清零，用于使能/禁止由数据结束引起的中断。
0：禁止数据结束中断
1：使能数据结束中断

位 7 **CMDSENTIE**：命令发送中断使能 (Command sent interrupt enable)
由软件置 1 和清零，用于使能/禁止由发送命令引起的中断。
0：禁止命令发送中断
1：使能命令发送中断

位 6 **CMDRENDIE**：命令响应接收中断使能 (Command response received interrupt enable)
由软件置 1 和清零，用于使能/禁止由接收命令响应引起的中断。
0：禁止命令响应接收中断
1：使能命令响应接收中断

位 5 **RXOVERRIE**：Rx FIFO 上溢错误中断使能 (Rx FIFO overrun error interrupt enable)
由软件置 1 和清零，用于使能/禁止由 Rx FIFO 上溢错误引起的中断。
0：禁止 Rx FIFO 上溢错误中断
1：使能 Rx FIFO 上溢错误中断

位 4 **TXUNDERRIE**：Tx FIFO 下溢错误中断使能 (Tx FIFO underrun error interrupt enable)
由软件置 1 和清零，用于使能/禁止由 Tx FIFO 下溢错误引起的中断。
0：禁止 Tx FIFO 下溢错误中断
1：使能 Tx FIFO 下溢错误中断

位 3 **DTIMEOUTIE**：数据超时中断使能 (Data timeout interrupt enable)
由软件置 1 和清零，用于使能/禁止由数据超时引起的中断。
0：禁止数据超时中断
1：使能数据超时中断

位 2 **CTIMEOUTIE**：命令超时中断使能 (Command timeout interrupt enable)
由软件置 1 和清零，用于使能/禁止由命令超时引起的中断。
0：禁止命令超时中断
1：使能命令超时中断

位 1 **DCRCFAILIE**：数据 CRC 失败中断使能 (Data CRC fail interrupt enable)
由软件置 1 和清零，用于使能/禁止由数据 CRC 失败引起的中断。
0：禁止数据 CRC 失败中断
1：使能数据 CRC 失败中断

位 0 **CCRCFAILIE**：命令 CRC 失败中断使能 (Command CRC fail interrupt enable)
由软件置 1 和清零，用于使能/禁止由命令 CRC 失败引起的中断。
0：禁止命令 CRC 失败中断
1：使能命令 CRC 失败中断

48.8.14 SDMMC 确认定时器寄存器 (SDMMC_ACKTIMER)

SDMMC acknowledgment timer register

偏移地址：0x040

复位值：0x0000 0000

SDMMC_ACKTIMER 寄存器包含以 SDMMC_CK 总线时钟周期表示的确认超时周期。

计数器加载 SDMMC_ACKTIMER 寄存器的值，并在数据路径状态机 (DPSM) 进入 Wait_Ack 状态后开始递减。如果定时器达到 0 而 DPSM 处于上述状态，则确认超时状态标志置 1。

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

位 31:25 保留，必须保持复位值。

位 24:0 **ACKTIME[24:0]**：启动确认超时时间 (Boot acknowledgment timeout period)
只有在禁止 CPSM (CPSMEN=0) 时才能通过固件写入此位。
以卡总线时钟周期表示的启动确认超时周期。

注： *数据传输必须首先写入到确认定时器寄存器，然后才写入到数据控制寄存器。*

48.8.15 SDMMC 数据 FIFO 寄存器 (SDMMC_FIFOR)

SDMMC data FIFO register

偏移地址：0x080 到 0x0BC

复位值：0x0000 0000

接收和发送 FIFO 只能被当作 1 字（32 位）宽的寄存器来读出或写入。FIFO 在连续地址上包含 16 个入口。这使 CPU 可以使用其加载 (load) 和存储 (store) 多操作符命令来读写 FIFO。

当通过半字或字节访问的方式访问 SDMMC_FIFOR 时，会生成 AHB 总线故障。

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

位 31:0 **FIFODATA[31:0]**：接收和发送 FIFO 数据 (Receive and transmit FIFO data)
只有在激活 DPSM (DPSMACT = 1) 时才能通过固件读取或写入此寄存器。
FIFO 数据占用 32 位字的 16 个入口。

48.8.16 SDMMC DMA 控制寄存器 (SDMMC_IDMACTRLR)

SDMMC DMA control register

偏移地址：0x050

复位值：0x0000 0000

接收和发送 FIFO 可以被当做 32 位宽的寄存器来读出或写入。FIFO 在 32 个连续地址上包含 32 个入口。这使 CPU 可以使用其加载 (load) 和存储 (store) 多操作符命令来读写 FIFO。

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.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	IDMABACT	IDMABMODE	IDMAEN
													rw	rw	rw

位 31:3 保留，必须保持复位值。

位 2 **IDMABACT**：双缓冲区模式激活缓冲区指示 (Double buffer mode active buffer indication)

只有在未激活 DPSM (DPSMACT = 0) 时才能通过固件写入此位。使能 IDMA 时，该位由硬件切换。

0：使能 IDMA 时，使用缓冲区 0，禁止固件对 IDMABASE0 进行写访问。

1：使能 IDMA 时，使用缓冲区 1，禁止固件对 IDMABASE1 进行写访问。

位 1 **IDMABMODE**：缓冲区模式选择 (Buffer mode selection)。

只有在未激活 DPSM (DPSMACT = 0) 时才能通过固件写入此位。

0：单缓冲区模式。

1：双缓冲区模式。

位 0 **IDMAEN**：IDMA 使能 (IDMA enable)

只有在未激活 DPSM (DPSMACT = 0) 时才能通过固件写入此位。

0：禁止 IDMA

1：使能 IDMA

48.8.17 SDMMC IDMA 缓冲区大小寄存器 (SDMMC_IDMABSIZER)

SDMMC IDMA buffer size register

偏移地址：0x054

复位值：0x0000 0000

SDMMC_IDMABSIZER 寄存器包含双缓冲区配置下的缓冲区大小。

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.	Res.	IDMABNDT[7:0]								Res.	Res.	Res.	Res.	Res.
			rw	rw	rw	rw	rw	rw	rw	rw					

位 31:13 保留，必须保持复位值。

位 12:5 **IDMABNDT[7:0]**：每个缓冲区的传输次数 (Number of transfers per buffer)。
该 8 位值应乘以 8（可得到用 32 位字表示的缓冲区大小）和 32（可得到用字节表示的缓冲区大小）。
示例：IDMABNDT = 0x01：缓冲区大小 = 8 字 = 32 字节。
只有在未激活 DPSM (DPSMACT = 0) 时才能通过固件写入这些位。

位 4:0 保留，必须保持复位值。

48.8.18 SDMMC IDMA 缓冲区 0 基址寄存器 (SDMMC_IDMABASE0R)

SDMMC IDMA buffer 0 base address register

偏移地址：0x058

复位值：0x0000 0000

SDMMC_IDMABASE0R 寄存器包含单缓冲区配置下的存储器缓冲区基址和双缓冲区配置下的缓冲区 0 基址。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IDMABASE0[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
IDMABASE0[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	r	r

位 31:0 **IDMABASE0[31:0]**：缓冲区 0 存储器基址位 [31:2]，应按字对齐（位 [1:0] 始终为 0 且为只读位）。
当 DPSM 未激活 (DPSMACT = 0) 时，可通过固件写入该寄存器，当 DPSM 激活 (DPSMACT = 1) 且存储器缓冲区 0 未激活（IDMABACT = “1”）时，可通过固件动态写入该寄存器。

48.8.19 SDMMC IDMA 缓冲区 1 基址寄存器 (SDMMC_IDMABASE1R)

SDMMC IDMA buffer 1 base address register

偏移地址：0x05C

复位值：0x0000 0000

SDMMC_IDMABASE1R 寄存器包含双缓冲区配置下的第二个缓冲区存储器基址。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IDMABASE1[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
IDMABASE1[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	r	r

位 31:0 **IDMABASE1[31:0]**：缓冲区 1 存储器基址，应按字对齐（位 [1:0] 始终为 0 且为只读位）。当 DPSM 未激活 (DPSMACT = 0) 时，可通过固件写入该寄存器，当 DPSM 激活 (DPSMACT = 1) 且存储器缓冲区 1 未激活（IDMABACT = “0”）时，可通过固件动态写入该寄存器。

48.8.20 SDMMC 寄存器映射

下表对 SDMMC 寄存器进行了汇总。

表 311. SDMMC 寄存器映射

偏移	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	SDMMC_ POWER	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DIRPOL	VSWITCHEN	VSWITCH	PWRCTRL[1:0]	
	Reset value																												0	0	0	0	0
0x04	SDMMC_ CLKCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SELCLKRX[1:0]		BUSSPEED	DDR	HWFC_EN	NEGEDGE	WIDBUS[1:0]		Res.	PWRSAV	Res.	Res.	CLKDIV[9:0]									
	Reset value											0	0	0	0	0	0	0	0		0			0	0	0	0	0	0	0	0	0	0
0x08	SDMMC_ ARGR	CMDARG[31:0]																															
	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	SDMMC_ CMDR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CMDSUSPEND	BOOTEN	BOOTMODE	DTHOLD	CPSMEN	WAITPEND	WAITINT	WAITRESP[1:0]		CMDSTOP	CMDTRANS	CMDINDEX[5:0]					
	Reset value																																
0x10	SDMMC_ RESPCMDR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RESPCMD[5:0]					
	Reset value																											0	0	0	0	0	0
0x14	SDMMC_ RESP1R	CARDSTATUS1[31:0]																															
	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
0x18	SDMMC_ RESP2R	CARDSTATUS2[31:0]																															
	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
0x1C	SDMMC_ RESP3R	CARDSTATUS3[31:0]																															
	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
0x20	SDMMC_ RESP4R	CARDSTATUS4[31:0]																															
	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	SDMMC_ DTIMER	DATATIME[31:0]																															
	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
0x28	SDMMC_ DLENR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DATALENGTH[24:0]																								
	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

表 311. SDMMC 寄存器映射（续）

偏移	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
0x2C	**SDMMC_ DCTRLR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FIFORST	BOOTACKEN	SDIOEN	RWMOD	RWSTOP	RWSTART	DBLOCK SIZE[3:0]				DTMODE		DTDIR	DTEN
	Reset value																			0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x30	**SDMMC_ DCNTR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DATACOUNT[24:0]																								
	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
0x34	**SDMMC_ STAR**	Res.	Res.	Res.	IDMABTC	IDMATE	CKSTOP	VSWEND	ACKTIMEOUT	ACKFAIL	SDIOIT	BUSYD0END	BUSYD0	RXFIFOE	TXFIFOE	RXFIFOF	TXFIFOF	RXFIFOHF	TXFIFOHE	CPSMACT	DPSMACT	DABORT	DBCKEND	DHOLD	DATAEND	CMDSENT	CMDREND	RXOVERR	TXUNDERR	DTIMEOUT	CTIMEOUT	DCRCFAIL	CCRCFAIL
	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
0x38	**SDMMC_ ICR**	Res.	Res.	Res.	IDMABTCC	IDMATEC	CKSTOPC	VSWEDNDC	ACKTIMEOUTC	ACKFAILC	SDIOITC	BUSYD0ENDC	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DABORTC	DBCKENDC	DHOLDC	DATAENDC	CMDSENTC	CMDRENDC	RXOVERRC	TXUNDERRC	DTIMEOUTC	CTIMEOUTC	DCRCFAILC	CCRCFAILC
	Reset value				0	0	0	0	0	0	0	0										0	0	0	0	0	0	0	0	0	0	0	0
0x3C	**SDMMC_ MASKR**	Res.	Res.	Res.	IDMABTCIE	Res.	CKSTOPIE	VSWENDIE	ACKTIMEOUTIE	ACKFAILIE	SDIOITIE	BUSYD0ENDIE	Res.	Res.	TXFIFOEIE	RXFIFOFIE	Res.	RXFIFOHFIE	TXFIFOHEIE	Res.	Res.	DABORTIE	DBCKENDIE	DHOLDIE	DATAENDIE	CMDSENTIE	CMDRENDIE	RXOVERRIE	TXUNDERRIE	DTIMEOUTIE	CTIMEOUTIE	DCRCFAILIE	CCRCFAILIE
	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
0x40	**SDMMC_ ACKTIMER**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ACKTIME[24:0]																								
	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
0x44 - 0x4C	**Reserved**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
0x50	**SDMMC_ IDMACTRLR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	IDMABACT	IDMABMODE	IDMAEN
	Reset value																														0	0	0
0x54	**SDMMC_ IDMABSIZER**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	IDMABNDT[7:0]								Res.	Res.	Res.	Res.	Res.
	Reset value																				0	0	0	0	0	0	0	0					
0x58	**SDMMC_ IDMABASE0R**	IDMABASE0[31:0]																															
	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
0x5C	**SDMMC_ IDMABASE1R**	IDMABASE1[31:0]																															
	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
0x60 - 0x7C	**Reserved**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

表 311. SDMMC 寄存器映射（续）

偏移	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
0x80	**SDMMC_ FIFOR**	FIF0Data[31:0]																															
	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
0x84 - x3FC	**Reserved**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

有关寄存器边界地址的信息，请参见*第 78 页的第 2.2.2 节*。

49 控制器局域网络 (bxCAN)

49.1 简介

基本扩展 CAN 外设又称 **bxCAN**，可与 CAN 网络进行交互。该外设支持 2.0A 和 B 版本的 CAN 协议，旨在以最少的 CPU 负载高效管理大量的传入消息，并可按需要的优先级实现消息发送。

在有关安全性的应用中，CAN 控制器提供所有必要的硬件功能来支持 CAN 时间触发通信方案。

49.2 bxCAN 主要特性

- 支持 2.0 A 及 2.0 B Active 版本 CAN 协议
- 比特率高达 1 Mb/s
- 支持时间触发通信方案

发送

- 三个发送邮箱
- 可配置的发送优先级
- SOF 发送时间戳

接收

- 两个具有三级深度的接收 FIFO
- 可扩展的过滤器组：
  - 对于双 CAN，CAN1 和 CAN2 之间共享 28 个过滤器组
- 标识符列表功能
- 可配置的 FIFO 上溢
- SOF 接收时间戳

时间触发通信方案

- 禁止自动重发送模式
- 16 位自由运行定时器
- 在最后两个数据字节发送时间戳

管理

- 可屏蔽中断
- 在唯一地址空间通过软件实现高效的邮箱映射

双 CAN 外设配置

- CAN1：主 bxCAN，用于管理 bxCAN 与 512 字节 SRAM 存储器之间的通信
- CAN2：从 bxCAN，无法直接访问 SRAM 存储器
- 两个 bxCAN 单元共享 512 字节 SRAM 存储器（请参见*图 564：Dual-CAN 框图*）

49.3 bxCAN 一般说明

在如今的 CAN 应用中，网络节点数量日益增多，经常需要通过网关将数个网络连接在一起。通常，系统中的消息（各个节点需要处理的消息）数量也有了显著增加。除应用程序消息外，还引入了网络管理和诊断消息。

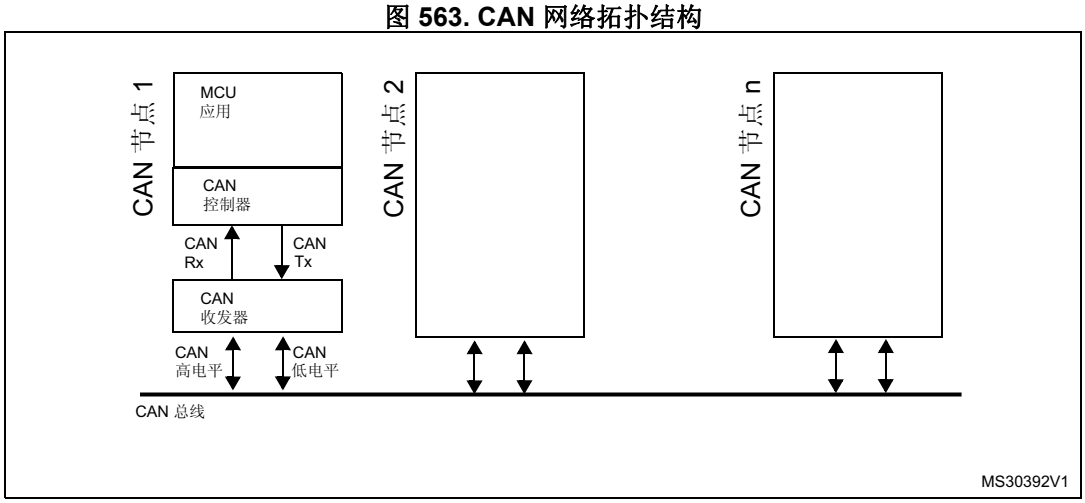
- 各种类型的消息需要一个增强的筛选机制进行处理。

此外，应用程序任务需要更多的 CPU 时间，因此必须减少因消息接收而对实时处理造成的限制。

- 接收 FIFO 方案使 CPU 能够长时间专门处理应用程序任务，而又不致丢失消息。

基于标准 CAN 驱动程序的标准 HLP（更高层协议）需要一个高效接口来与 CAN 控制器连接。

图 563. CAN 网络拓扑结构



49.3.1 CAN 2.0B 主动内核

bxCAN 模块可完全自主地处理 CAN 消息的发送和接收。标准标识符（11 位）和扩展标识符（29 位）完全由硬件支持。

49.3.2 控制、状态和配置寄存器

应用程序使用这些寄存器进行以下操作：

- 配置 CAN 参数，例如波特率
- 请求发送
- 处理接收
- 管理中断
- 获取诊断信息

49.3.3 发送邮箱

软件可通过三个发送邮箱设置消息。发送调度程序负责决定首先发送哪个邮箱的内容。

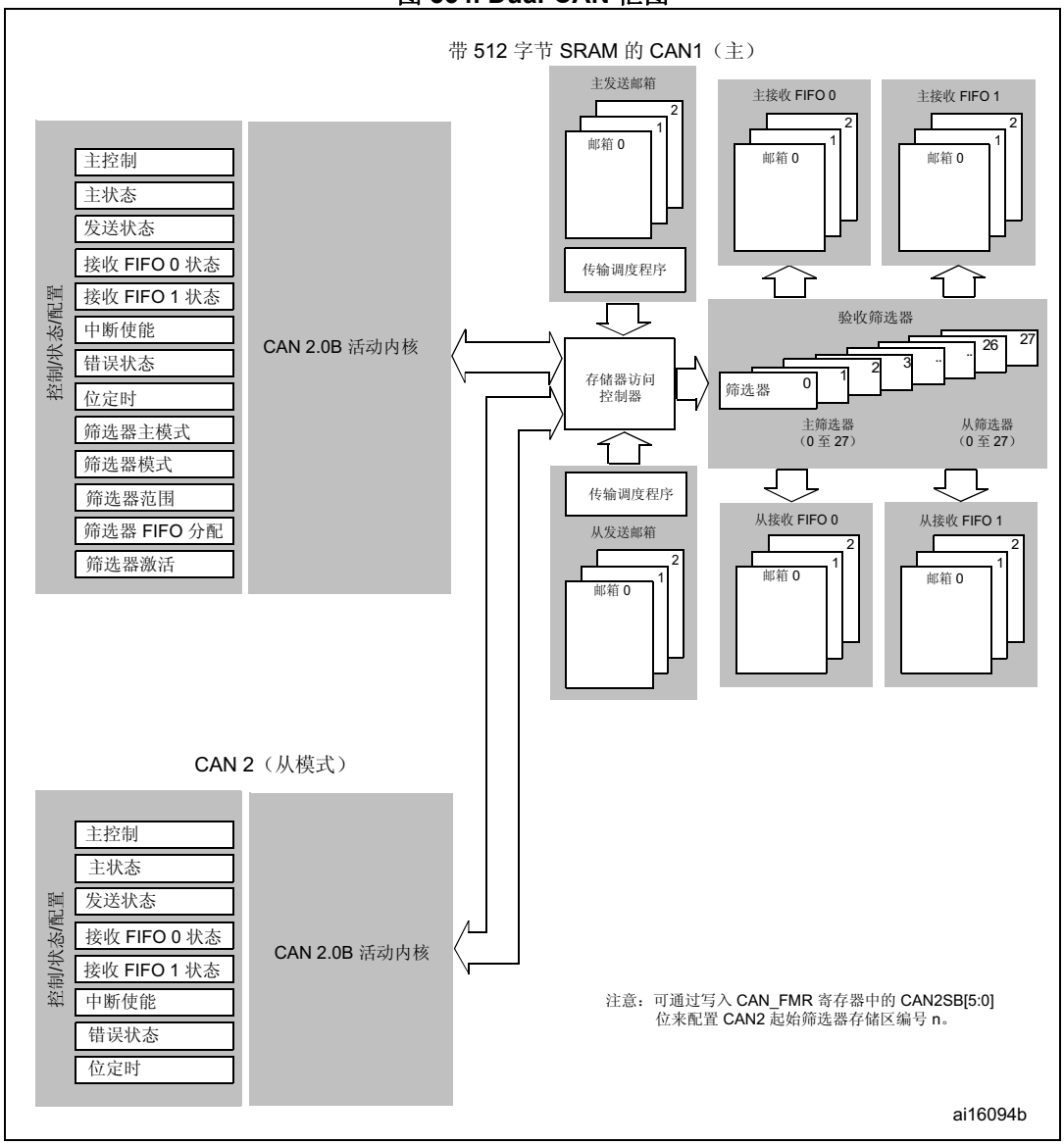
49.3.4 验收过滤器

双 CAN 配置的 bxCAN 提供了多达 28 个可调整/可配置的标识符过滤器组，用于选择软件所需的传入消息并丢弃其余消息。

接收 FIFO

硬件使用两个接收 FIFO 来存储传入消息。每个 FIFO 中可以存储三条完整消息。FIFO 完全由硬件管理。

图 564. Dual-CAN 框图



49.4 bxCAN 工作模式

bxCAN 有三种主要的工作模式：**初始化模式**、**正常模式**和**睡眠模式**。硬件复位后，bxCAN 进入睡眠模式以降低功耗，同时 CANTX 上的内部上拉电阻激活。软件将 CAN_MCR 寄存器的 INRQ 或 SLEEP 位置 1，以请求 bxCAN 进入**初始化**或**睡眠**模式。一旦进入该模式，bxCAN 即将 CAN_MSR 寄存器的 INAK 或 SLAK 位置 1，以确认该模式，同时禁止内部上拉电阻。如果 INAK 和 SLAK 均未置 1，则 bxCAN 将处于**正常**模式。进入**正常**模式之前，bxCAN 必须始终在 CAN 总线上实现**同步**。为了进行同步，bxCAN 将等待 CAN 总线空闲（即，已监测到 CANRX 上的 11 个连续隐性位）。

49.4.1 初始化模式

当硬件处于初始化模式时，可以进行软件初始化。为进入该模式，软件将 CAN_MCR 寄存器的 INRQ 位置 1，并等待硬件通过将 CAN_MCR 寄存器的 INAK 位置 1 来确认请求。

为退出初始化模式，软件将 INQR 位清零。一旦硬件将 INAK 位清零，bxCAN 即退出初始化模式。

在初始化模式下，所有从 CAN 总线传入和传出的消息都将停止，并且 CAN 总线输出 CANTX 的状态为隐性（高）。

进入初始化模式不会更改任何配置寄存器。

为初始化 CAN 控制器，软件必须设置位定时 (CAN_BTR) 和 CAN 选项 (CAN_MCR) 寄存器。

为初始化与 CAN 过滤器组相关的寄存器（模式、尺度、FIFO 分配、激活和过滤器值），软件必须将 FINIT 位 (CAN_FMR) 置 1。过滤器的初始化也可以在初始化模式之外进行。

注： *FINIT=1 时，CAN 接收停用。*

过滤器值也可通过停用（CAN_FA1R 寄存器的）相关过滤器激活位来修改。

如果某个过滤器组未使用，建议将其保持未激活状态（将相应 FACT 位保持清零）。

49.4.2 正常模式

一旦初始化完成，软件必须向硬件请求进入正常模式，这样才能在 CAN 总线上进行同步，并开始接收和发送。

进入正常模式的请求可通过将 CAN_MCR 寄存器的 INRQ 位清零来发出。bxCAN 进入正常模式，并与 CAN 总线上的数据传输实现同步后，即可参与总线活动。执行这一步时，需要等待出现一个由 11 个连续隐性位（总线空闲状态）组成的序列。硬件通过将 CAN_MSR 寄存器的 INAK 位清零，来确认切换到正常模式。

过滤器值的初始化与初始化模式无关，但必须要在过滤器处于未激活状态（相应 FACTx 位清零）时进行。过滤器尺度和模式配置必须在进入正常模式之前完成。

49.4.3 睡眠模式（低功耗）

为降低能耗功耗，bxCAN 具有低功耗模式，称为睡眠模式。软件通过将 CAN_MCR 寄存器的 SLEEP 位置 1 而发出请求后，即可进入该模式。该模式下，bxCAN 时钟停止，但软件仍可访问 bxCAN 邮箱。

在 bxCAN 处于**睡眠**模式时，如果软件通过将 INRQ 位置 1 来请求进入**初始化**模式，则必须同时将 SLEEP 位清零。

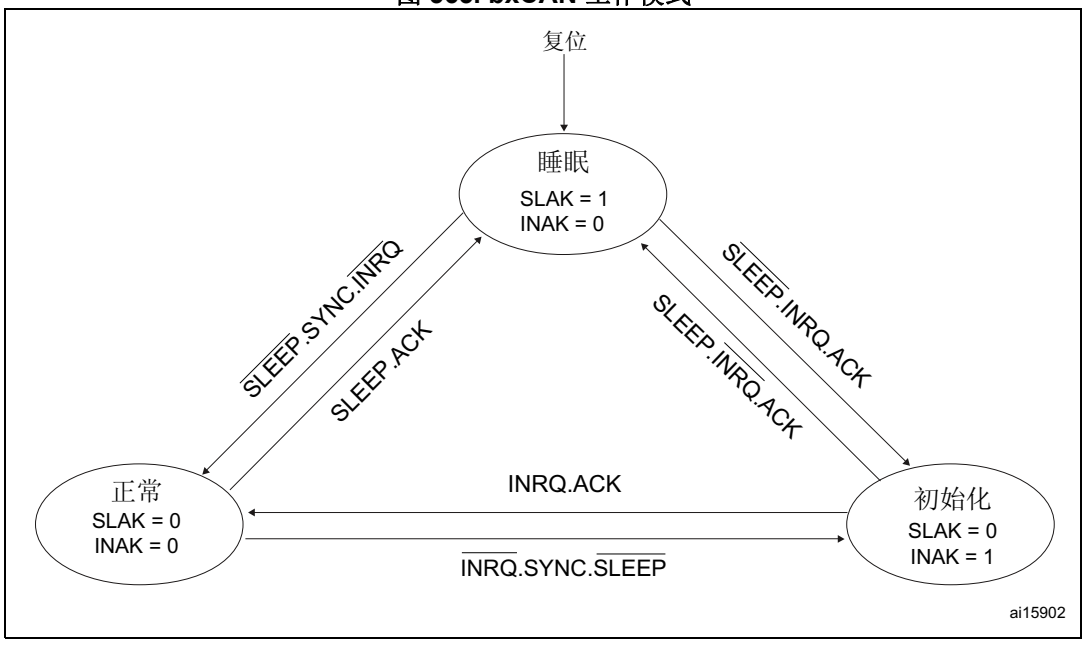
软件将 SLEEP 位清零或是检测到 CAN 总线活动时，bxCAN 即被唤醒（退出睡眠模式）。

检测到 CAN 总线活动后，如果 CAN_MCR 寄存器的 AWUM 位置 1，硬件将通过清零 SLEEP 位来自动执行唤醒序列。如果 AWUM 位清零，在发生唤醒中断时，软件必须将 SLEEP 位清零才能退出睡眠模式。

注： *如果使能唤醒中断（CAN_IER 寄存器的 WKUIE 位置 1），即使 bxCAN 自动执行唤醒序列，一旦检测到 CAN 总线活动，也会发生唤醒中断。*

SLEEP 位清零后，一旦 bxCAN 与 CAN 总线同步，即会退出睡眠模式，请参见 *图 565：bxCAN 工作模式*。一旦硬件将 SLAK 位清零，即会退出睡眠模式。

图 565. bxCAN 工作模式



1. ACK = 硬件通过将 CAN_MSR 寄存器的 INAK 或 SLAK 位置 1 来确认请求的等待状态
2. SYNC = bxCAN 等待 CAN 总线变为空闲（即在 CANRX 上监测到连续 11 个隐性位）的状态

49.5 测试模式

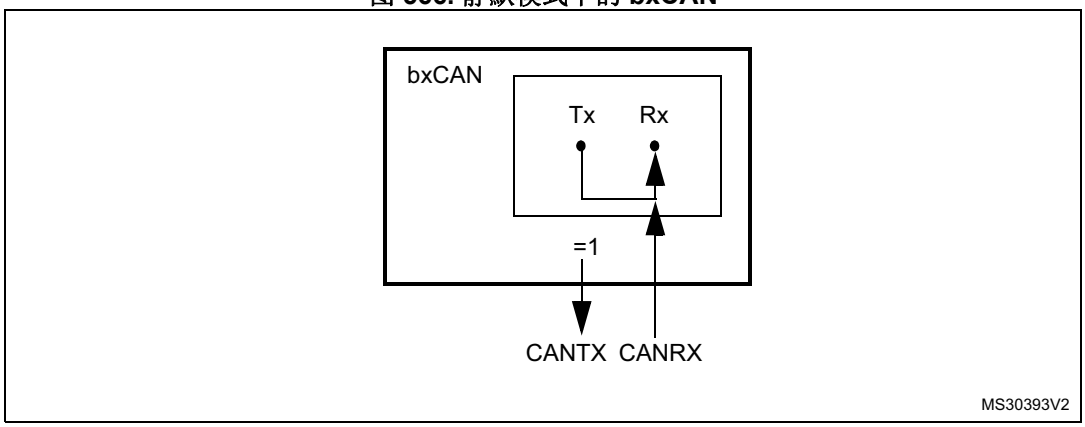
可以通过 CAN_BTR 寄存器中的 SILM 和 LBKM 位来选择测试模式。这些位必须在 bxCAN 处于初始化模式时进行配置。选择测试模式后，必须复位 CAN_MCR 寄存器中的 INRQ 位才能进入正常模式。

49.5.1 静默模式

可以通过将 CAN_BTR 寄存器的 SILM 位置 1，将 bxCAN 置于静默模式。

在静默模式下，bxCAN 可以接收有效数据帧和有效遥控帧，但仅在 CAN 总线上发送隐性位，并且无法启动发送。如果 bxCAN 必须发送一个显性位（ACK 位、溢出标志、活动错误标志），该位将在内部被改道发送，以便 CAN 内核可以监视该显性位，但 CAN 总线可以保持隐性状态。静默模式可用于分析 CAN 总线上的流量，同时又不会因发送显性位（确认位、错误帧）对其造成影响。

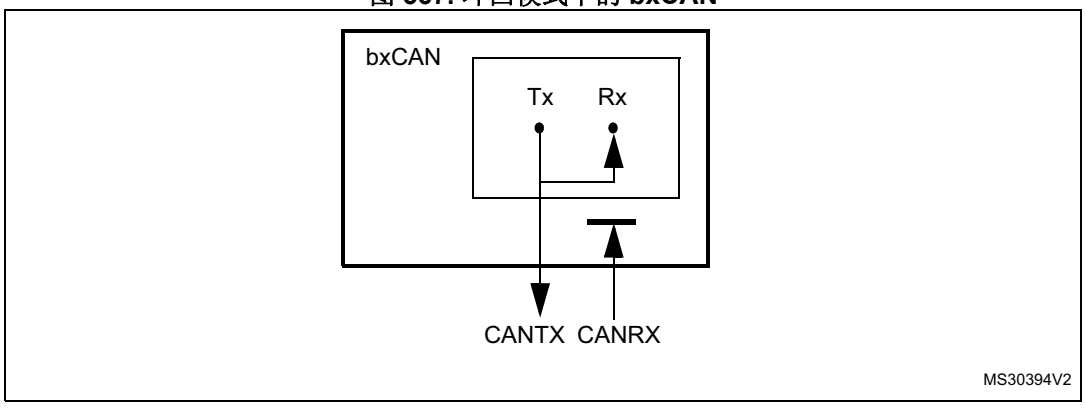
图 566. 静默模式下的 bxCAN



49.5.2 环回模式

可以通过将 CAN_BTR 寄存器的 LBKM 位置 1，将 bxCAN 置于环回模式。在环回模式下，bxCAN 将其自身发送的消息作为接收的消息来处理并存储（如果这些消息通过了验收筛选）在接收邮箱中。

图 567. 环回模式下的 bxCAN

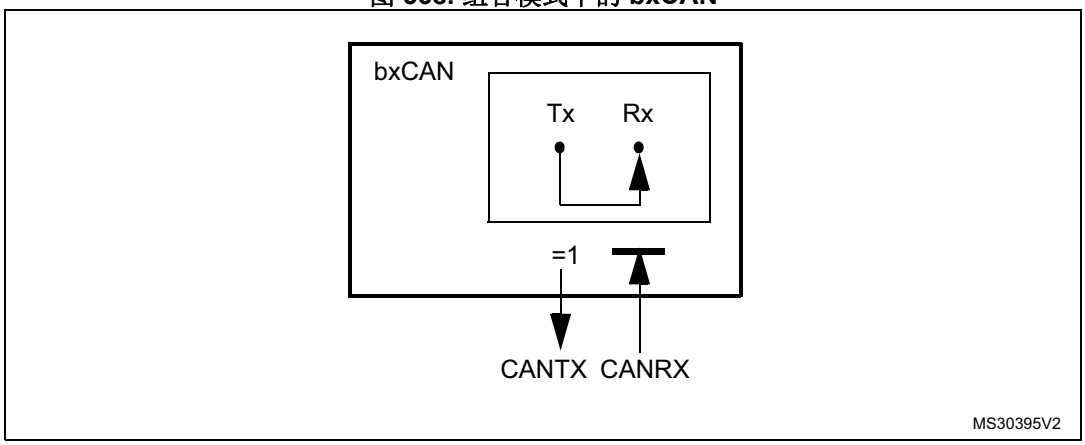


该模式为自检功能提供。为了不受外部事件的影响，CAN 内核在环回模式下将忽略确认错误（在数据/远程帧的确认时隙不对显性位采样）。在此模式下，bxCAN 将执行从发送输出到接收输入的内部反馈。bxCAN 将忽略 CANRX 输入引脚的实际值。从 CANTX 引脚可以监视发送的消息。

49.5.3 环回与静默组合模式

可以通过将 CAN_BTR 寄存器的 LBKM 和 SILM 位置 1，将环回模式和静默模式组合起来。该模式可用于“热自检”，也就是说，bxCAN 可以像在环回模式下一样进行检测，同时又不会影响与 CANTX 和 CANRX 引脚相连接的运行中的 CAN 系统。在此模式下，CANRX 引脚与 bxCAN 断开连接，CANTX 引脚则保持隐性。

图 568. 组合模式下的 bxCAN



49.6 调试模式下的行为

当微控制器进入调试模式（Cortex®-M4 内核停止）时，bxCAN 可以继续正常工作，也可以停止工作，具体取决于如下条件：

- 双重模式下 DBG 模块中用于 CAN1 的 DBG_CAN1_STOP 位或者用于 CAN2 的 DBG_CAN2_STOP 位。
- CAN_MCR 中的 DBF 位，有关更多详细信息，请参见*第 49.9.2 节：CAN 控制和状态寄存器*。

49.7 bxCAN 功能说明

49.7.1 发送处理

为了发送消息，应用程序必须在请求发送前，通过将 CAN_TIxR 寄存器的相应 TXRQ 位置 1，选择一个**空**发送邮箱，并设置标识符、数据长度代码 (DLC) 和数据。一旦邮箱退出**空**状态，软件即不再具有对邮箱寄存器的写访问权限。TXRQ 位置 1 后，邮箱立即进入**挂起**状态，等待成为优先级最高的邮箱，请参见*发送优先级*。一旦邮箱拥有最高优先级，即被**安排**发送。CAN 总线变为空闲后，被安排好的邮箱中的消息即开始发送（进入**发送**状态）。邮箱一旦发送成功，即恢复**空**状态。硬件通过将 CAN_TSR 寄存器的 RQCP 和 TXOK 位置 1，来表示发送成功。

如果发送失败，失败原因将由 CAN_TSR 寄存器的 ALST 位（仲裁丢失）和/或 TERR 位（检测到发送错误）指示。

发送优先级

按标识符

当多个发送邮箱挂起时，发送顺序由邮箱中所存储消息的标识符来确定。根据 CAN 协议的仲裁，标识符值最低的消息具有最高的优先级。如果标识符值相等，则首先安排发送编号较小的邮箱。

按发送请求顺序

可以通过设置 CAN_MCR 寄存器中的 TXFP 位，将发送邮箱配置为发送 FIFO。在此模式下，优先级顺序按照发送请求顺序来确定。

该模式对分段发送非常有用。

中止

可以通过将 CAN_TSR 寄存器的 ABRQ 位置 1，来中止发送请求。在**挂起**或**已安排**状态下，邮箱立即中止。如果在邮箱处于**发送**状态时请求中止，则会出现两种结果。如果邮箱发送成功，将变为**空**状态，同时 CAN_TSR 寄存器的 TXOK 位置 1。如果发送失败，邮箱变为**已安排**状态，发送中止并变为**空**状态，同时 TXOK 位清零。在所有情况下，邮箱至少在当前发送结束时都会恢复**空**状态。

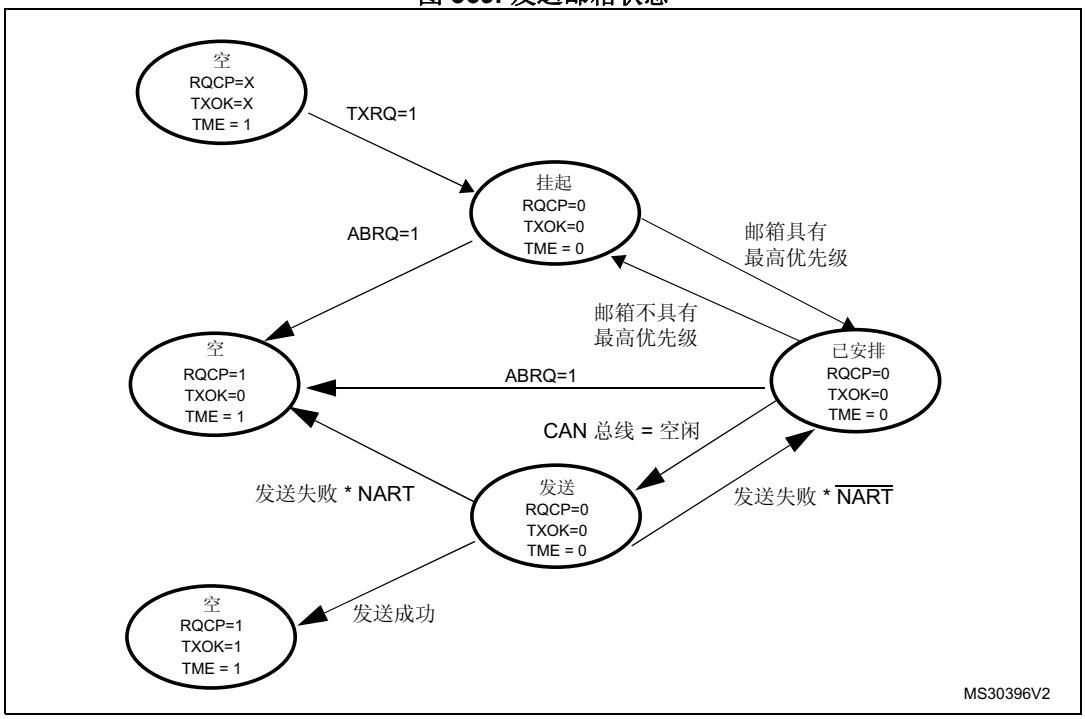
禁止自动重发送模式

该模式旨在满足 CAN 标准的时间触发通信方案的要求。要将硬件配置为此模式，必须将 CAN_MCR 寄存器的 NART 位置 1。

在此模式下，每个发送仅启动一次。如果第一次尝试失败，由于仲裁丢失或错误，硬件将不会自动重新启动消息发送。

第一次发送尝试结束时，硬件将认为请求已完成，并将 CAN_TSR 寄存器的 RQCP 位置 1。发送结果由 CAN_TSR 寄存器的 TXOK、ALST 和 TERR 位来指示。

图 569. 发送邮箱状态



49.7.2 时间触发通信模式

在此模式下，CAN 硬件的内部计数器激活，用于为接收和发送邮箱生成时间戳值，这些值分别存储在 CAN_RDTxR/CAN_TDTxR 寄存器中。内部计数器在每个 CAN 位时间递增（请参见*第 49.7.7 节：位时序*）。在接收和发送时，都会在帧起始位的采样点捕获内部计数器。

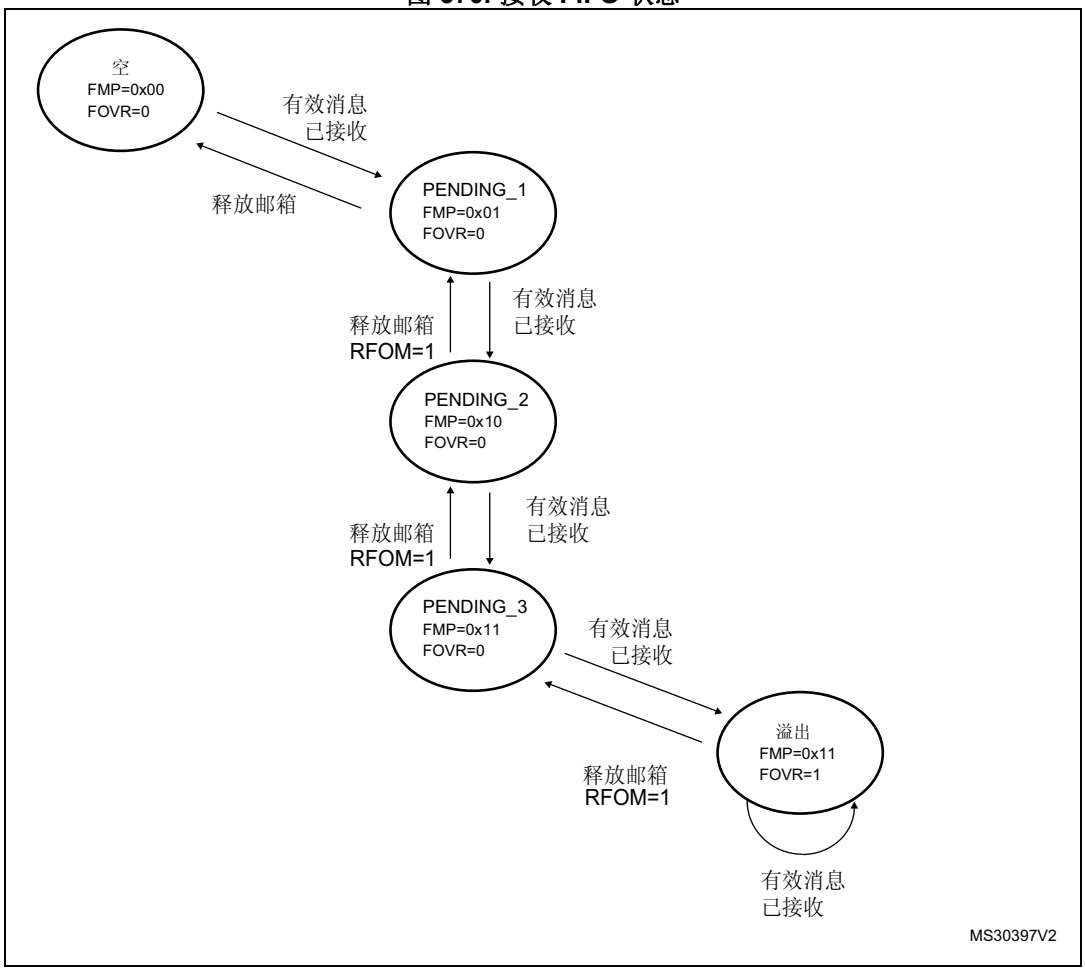
49.7.3 接收处理

为了接收 CAN 消息，提供了构成 FIFO 的三个邮箱。为了节约 CPU 负载，简化软件并保证数据一致性，FIFO 完全由硬件进行管理。应用程序通过 FIFO 输出邮箱访问 FIFO 中所存储的消息。

有效消息

当消息依据 CAN 协议正确接收（直到 EOF 字段的倒数第二位都没有发送错误）并且成功通过标识符筛选后，该消息将视为有效，请参见*第 49.7.4 节：标识符筛选*。

图 570. 接收 FIFO 状态



FIFO 管理

FIFO 开始时处于**空**状态，在接收的第一条有效消息存储在其中后，变为 **pending_1** 状态。硬件通过将 CAN_RFR 寄存器的 FMP[1:0] 位置为 01b 来指示该事件。消息将在 FIFO 输出邮箱中供读取。软件将读取邮箱内容，并通过将 CAN_RFR 寄存器的 RFOM 位置 1，来将邮箱释放。FIFO 随即恢复**空**状态。如果同时接收到新的有效消息，FIFO 将保持 **pending_1** 状态，新消息将在输出邮箱中供读取。

如果应用程序未释放邮箱，下一条有效消息将存储在 FIFO 中，使其进入 **pending_2** 状态 (FMP[1:0] = 10b)。下一条有效消息会重复该存储过程，同时将 FIFO 变为 **pending_3** 状态 (FMP[1:0] = 11b)。此时，软件必须通过将 RFOM 位置 1 来释放输出邮箱，从而留出一个空邮箱来存储下一条有效消息。否则，下一次接收到有效消息时，将导致消息丢失。

另请参见*第 49.7.5 节：消息存储*。

上溢

一旦 FIFO 处于 **pending_3** 状态（即三个邮箱均已满），则下一次接收到有效消息时，将导致**上溢**并丢失一条消息。硬件通过将 CAN_RFR 寄存器的 FOVR 位置 1 来指示上溢状况。丢失的消息取决于 FIFO 的配置：

- 如果禁止 FIFO 锁定功能（CAN_MCR 寄存器的 RFLM 位清零），则新传入的消息将覆盖 FIFO 中存储的最后一条消息。在这种情况下，应用程序将始终能访问到最新的消息。
- 如果使能 FIFO 锁定功能（CAN_MCR 寄存器的 RFLM 位置 1），则将丢弃最新的消息，软件将提供 FIFO 中最早的三条消息。

与接收相关的中断

消息存储到 FIFO 中后，FMP[1:0] 位即会更新，如果 CAN_IER 寄存器的 FMPIE 位置 1，将产生中断请求。

FIFO 存满消息（即存储了第三条消息）后，CAN_RFR 寄存器的 FULL 位置 1，如果 CAN_IER 寄存器的 FFIE 位置 1，将产生中断。

出现上溢时，FOVR 位将置 1，如果 CAN_IER 寄存器的 FOVIE 位置 1，将产生中断。

49.7.4 标识符筛选

在 CAN 协议中，消息的标识符与节点地址无关，但与消息内容有关。因此，发送器将消息广播给所有接收器。在接收到消息时，接收器节点会根据标识符的值来确定软件是否需要该消息。如果需要，该消息将复制到 SRAM 中。如果不需要，则必须在无软件干预的情况下丢弃该消息。

为满足该要求，bxCAN 控制器为应用程序提供了 28 个可配置且可调整的过滤器组 (27-0)，以便仅接收软件需要的消息。

此硬件筛选功能可以节省软件筛选所需的 CPU 资源。每个过滤器组 x 均包含两个 32 位寄存器，分别是 CAN_FxR0 和 CAN_FxR1。

可调整的宽度

为了根据应用程序的需求来优化和调整过滤器，每个过滤器组可分别进行伸缩调整。根据过滤器尺度不同，一个过滤器组可以：

- 为 STDID[10:0]、EXTID[17:0]、IDE 和 RTR 位提供一个 32 位过滤器。
- 为 STDID[10:0]、RTR、IDE 和 EXTID[17:15] 位提供两个 16 位过滤器。

请参见*图 571*。

此外，过滤器还可配置为掩码模式或标识符列表模式。

掩码模式

在**掩码**模式下，标识符寄存器与掩码寄存器关联，用以指示标识符的哪些位“必须匹配”，哪些位“无关”。

标识符列表模式

在**标识符列表**模式下，掩码寄存器用作标识符寄存器。这时，不会定义一个标识符和一个掩码，而是指定两个标识符，从而使单个标识符的数量加倍。传入标识符的所有位都必须与过滤器寄存器中指定的位匹配。

过滤器组尺度和模式配置

过滤器组通过相应的 CAN_FMR 寄存器进行配置。为了配置过滤器组，必须通过将 CAN_FAR 寄存器的 FACT 位清零而将其停用。过滤器尺度通过 CAN_FS1R 寄存器的相应 FSCx 位进行配置，请参见*图 571*。相应掩码/标识符寄存器的**标识符列表**或**标识符掩码**模式通过 CAN_FMR 寄存器的 FBMx 位进行配置。

要筛选一组标识符，应将掩码/标识符寄存器配置为掩码模式。

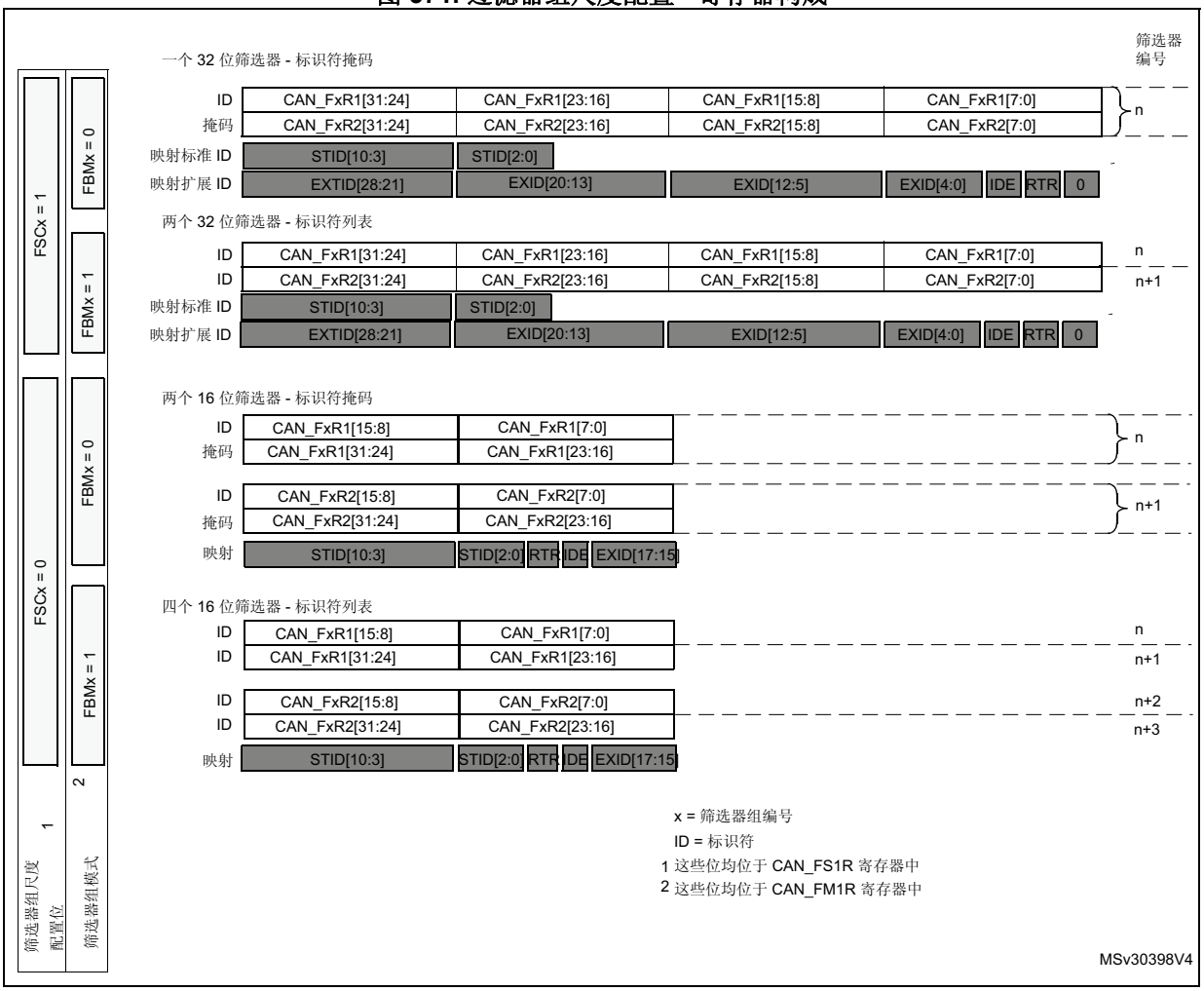
要选择单个标识符，应将掩码/标识符寄存器配置为标识符列表模式。

未由应用程序使用的过滤器应保持停用。

过滤器组中的每个过滤器将按从 0 到最大值的顺序进行编号（称为过滤器编号），具体取决于每个过滤器组的模式和尺度。

有关过滤器配置，请参见*图 571*。

图 571. 过滤器组尺度配置 - 寄存器构成



过滤器匹配索引

消息接收到 FIFO 中后，即可供应用程序使用。应用程序数据通常会复制到 SRAM 中的位置。为了将数据复制到正确的位置，应用程序必须通过标识符来识别数据。为了避免这种情况，方便访问 SRAM 位置，CAN 控制器提供了一个过滤器匹配索引。

该索引根据过滤器优先级规则与消息一同存储在邮箱中。因此，每条收到的消息都有相关联的过滤器匹配索引。

过滤器匹配索引的使用方法有两种：

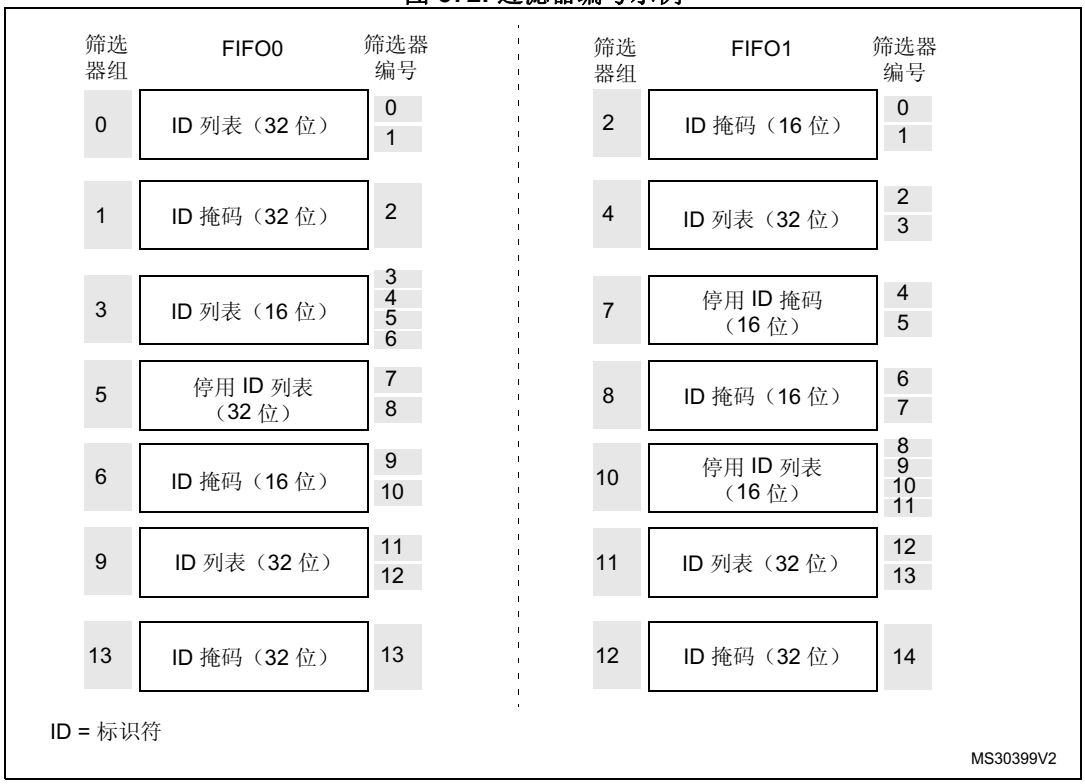
- 将过滤器匹配索引与预期值列表进行比较。
- 将过滤器匹配索引用作阵列索引，以访问数据目标位置。

对于非屏蔽过滤器，软件不再需要比较标识符。

如果过滤器有屏蔽，软件则只需比较屏蔽位。

过滤器编号的索引值与过滤器组的激活状态无关。此外，还将使用两个独立的编号方案，每个 FIFO 各一个。相关示例，请参见*图 572*。

图 572. 过滤器编号示例

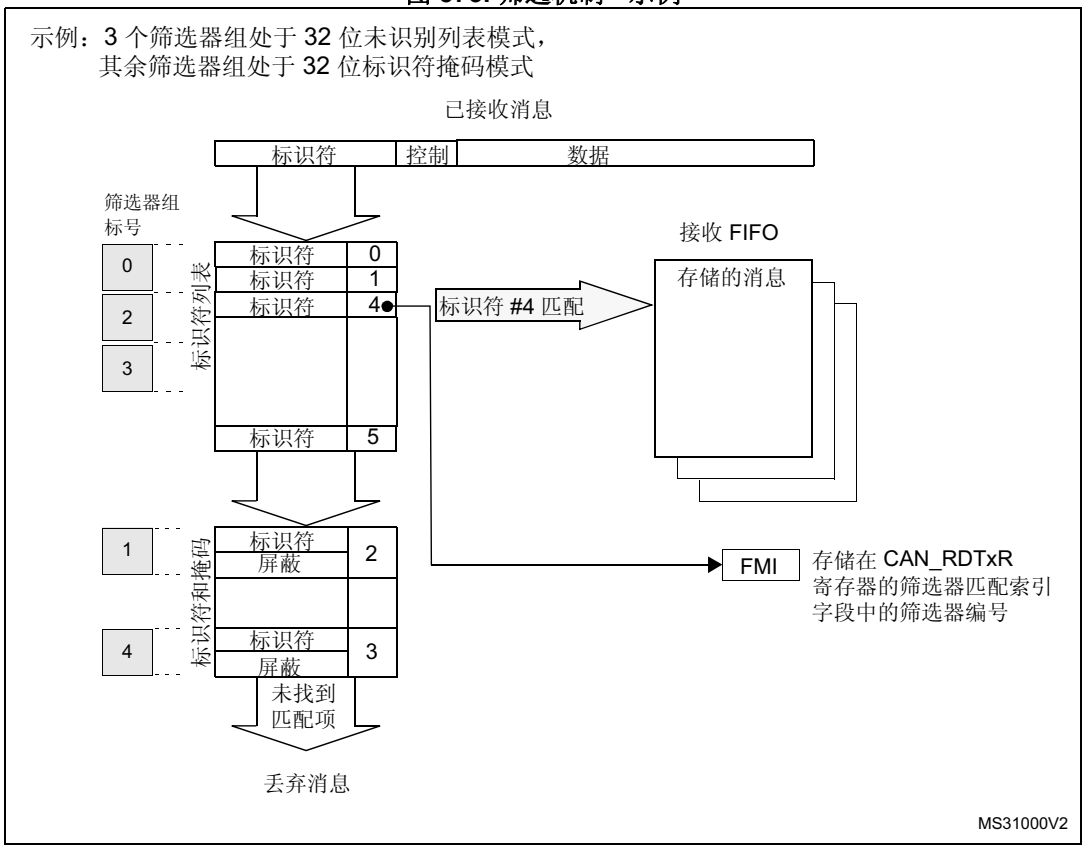


过滤器优先级规则

根据过滤器组合，可能会出现一个标识符成功通过数个过滤器的情况。这种情况下，将根据以下优先级规则选择接收邮箱中存储的过滤器匹配值：

- 32 位过滤器优先于 16 位过滤器。
- 对于尺度相等的过滤器，标识符列表模式优先于标识符掩码模式。
- 对于尺度和模式均相等的过滤器，则按过滤器编号确定优先级（编号越低，优先级越高）。

图 573. 筛选机制 - 示例



以上示例说明了 bxCAN 的筛选原则。接收到消息后，首先将标识符与标识符列表模式中配置的过滤器进行比较。如果匹配，消息将存储在相应 FIFO 中，匹配过滤器的索引则存储在过滤器匹配索引中。如本例所示，标识符与标识符 #2 匹配，因此消息内容和 FMI 2 存储到该 FIFO 中。

如果不匹配，则将传入标识符与掩码模式中配置的过滤器进行比较。

如果标识符与过滤器中配置的任何标识符均不匹配，硬件会在不干扰软件的情况下丢弃该消息。

49.7.5 消息存储

CAN 消息软件与硬件之间的接口通过邮箱实现。邮箱中包含所有与消息相关的信息：标识符、数据、控制、状态和时间戳信息。

发送邮箱

软件在空发送邮箱中设置将要发送的消息。发送状态由硬件在 CAN_TSR 寄存器中进行指示。

表 312. 发送邮箱映射

与发送邮箱基址之间的偏移	寄存器名称
0	CAN_TIxR
4	CAN_TDTxR
8	CAN_TDLxR
12	CAN_TDHxR

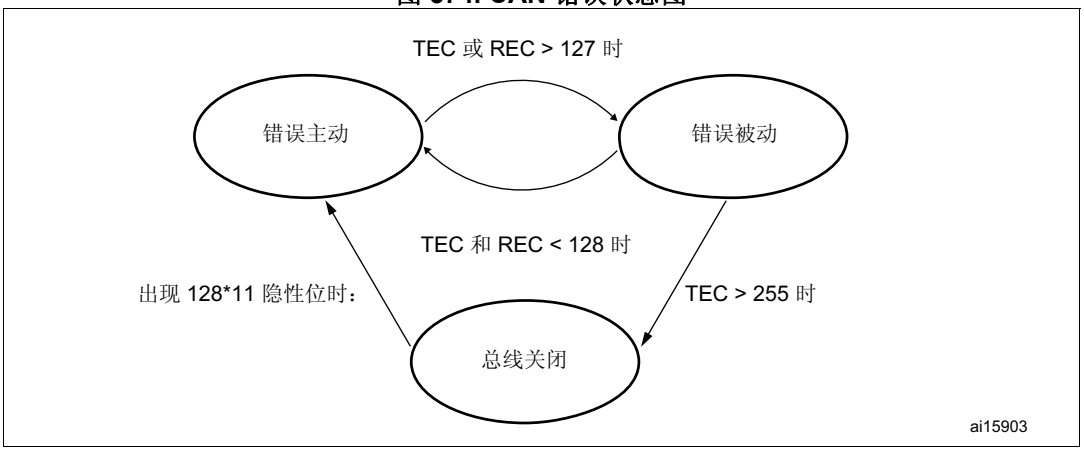
接收邮箱

消息在接收到后，将在 FIFO 输出邮箱中供软件使用。一旦软件对消息进行了处理（例如读取），则必须通过 CAN_RFR 寄存器的 RFOM 位释放 FIFO 输出邮箱，以接收下一条传入消息。过滤器匹配索引存储在 CAN_RDTxR 寄存器的 MFMI 字段中。16 位时间戳值则存储在 CAN_RDTxR 的 TIME[15:0] 字段中。

表 313. 接收邮箱映射

与接收邮箱基址之间的偏移（字节）	寄存器名称
0	CAN_RIxR
4	CAN_RDTxR
8	CAN_RDLxR
12	CAN_RDHxR

图 574. CAN 错误状态图



49.7.6 错误管理

如 CAN 协议所述，错误管理完全由硬件通过发送错误计数器（CAN_ESR 寄存器中的 TEC 值）和接收错误计数器（CAN_ESR 寄存器中的 REC 值）来处理，这两个计数器根据错误状况进行递增或递减。有关 TEC 和 REC 管理的详细信息，请参见 CAN 标准。

两者均可由软件读取，用以确定网络的稳定性。此外，CAN 硬件还将在 CAN_ESR 寄存器中提供当前错误状态的详细信息。通过 CAN_IER 寄存器（ERRIE 位等），软件可以非常灵活地配置在检测到错误时生成的中断。

总线关闭恢复

当 TEC 大于 255 时，达到总线关闭状态，该状态由 CAN_ESR 寄存器的 BOFF 位指示。在总线关闭状态下，bxCAN 不能再发送和接收消息。

bxCAN 可以自动或者应软件请求而从总线关闭状态中恢复（恢复错误主动状态），具体取决于 CAN_MCR 寄存器的 ABOM 位。但在两种情况下，bxCAN 都必须至少等待 CAN 标准中指定的恢复序列完成（在 CANRX 上监测到 128 次 11 个连续隐性位）。

如果 ABOM 位置 1，bxCAN 将在进入总线关闭状态后自动启动恢复序列。

如果 ABOM 位清零，则软件必须请求 bxCAN 先进入再退出初始化模式，从而启动恢复序列。

注： *在初始化模式下，bxCAN 不会监视 CANRX 信号，因此无法完成恢复序列。**要进行恢复，bxCAN 必须处于正常模式**。*

49.7.7 位时序

位时序逻辑将监视串行总线，执行采样并调整采样点，在调整采样点时，需要在起始位边沿进行同步并后续的边沿进行再同步。

通过将标称位时间划分为以下三段，即可解释其工作过程：

- **同步段 (SYNC_SEG)**：位变化应该在此时间段内发生。它只有一个时间片的固定长度 (1 x t_q)。
- **位段 1 (BS1)**：定义采样点的位置。它包括 CAN 标准的 PROP_SEG 和 PHASE_SEG1。其持续长度可以在 1 到 16 个时间片之间调整，但也可以自动加长，以补偿不同网络节点的频率差异所导致的正相位漂移。
- **位段 2 (BS2)**：定义发送点的位置。它代表 CAN 标准的 PHASE_SEG2。其持续长度可以在 1 到 8 个时间片之间调整，但也可以自动缩短，以补偿负相位漂移。

再同步跳转宽度 (SJW) 定义位段加长或缩短的上限。它可以在 1 到 4 个时间片之间调整。

有效边沿是指一个位时间内总线电平从显性到隐性的第一次转换（前提是控制器本身不发送隐性位）。

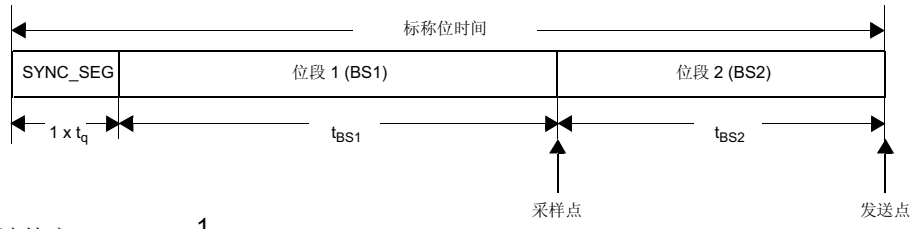
如果在 BS1 而不是 SYNC_SEG 中检测到有效边沿，则 BS1 会延长最多 SJW，以便延迟采样点。

相反地，如果在 BS2 而不是 SYNC_SEG 中检测到有效边沿，则 BS2 会缩短最多 SJW，以便提前发送点。

为了避免编程错误，位时序寄存器 (CAN_BTR) 只能在器件处于待机模式时进行配置。

注： *有关 CAN 位时序和再同步机制的详细说明，请参见 ISO 11898 标准。*

图 575. 位时序



$$波特率 = \frac{1}{NominalBitTime}$$

$$NominalBitTime = 1 \times t_q + t_{BS1} + t_{BS2}$$

公式中：

$t_{BS1} = t_q \times (TS1[3:0] + 1)$

$t_{BS2} = t_q \times (TS2[2:0] + 1)$

$t_q = (BRP[9:0] + 1) \times t_{PCLK}$

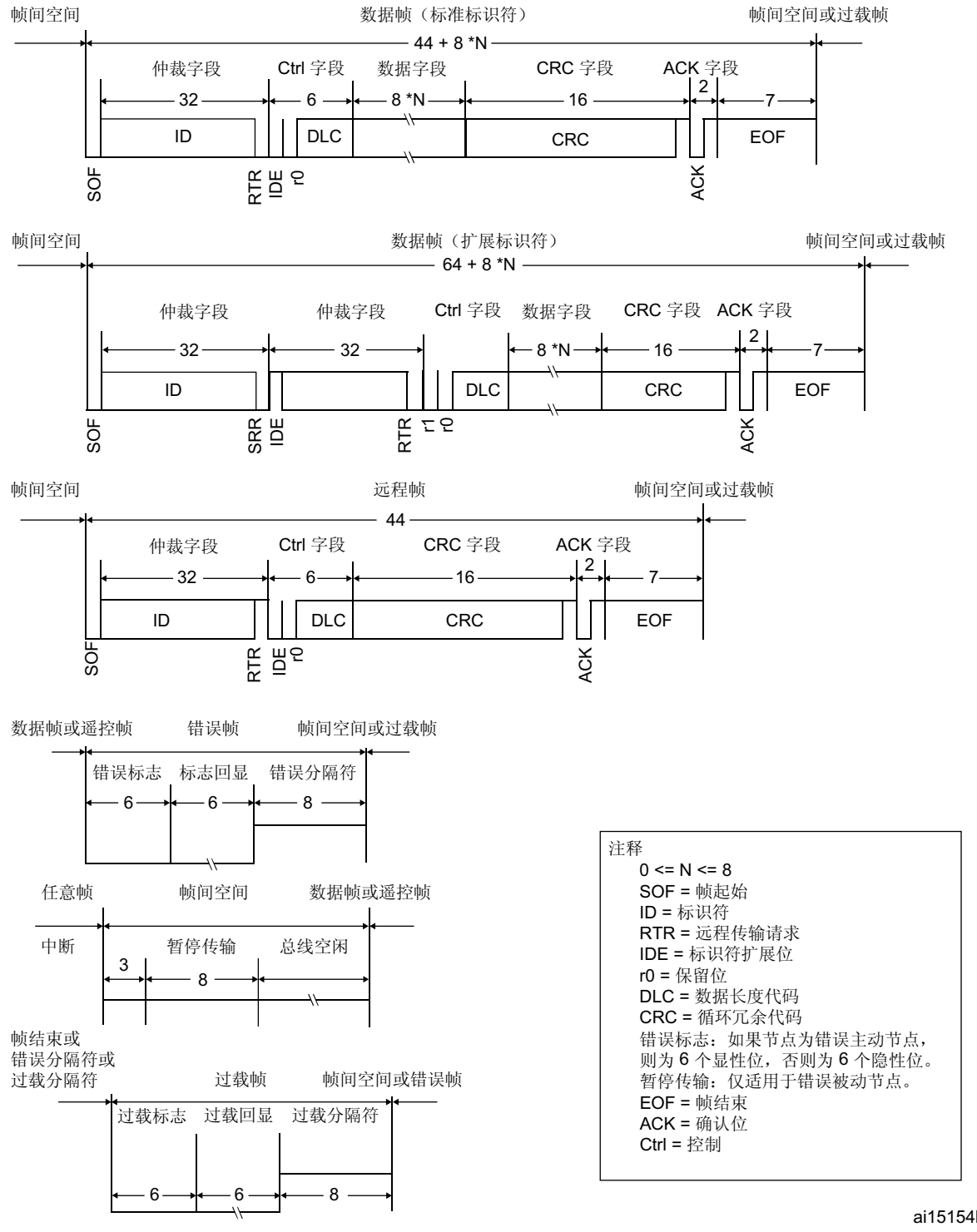
其中 t_q 为时间片

t_{PCLK} = APB 时钟的时间周期

BRP[9:0]、TS1[3:0] 和 TS2[2:0] 在 CAN_BTR 寄存器中定义。

MS31001V2

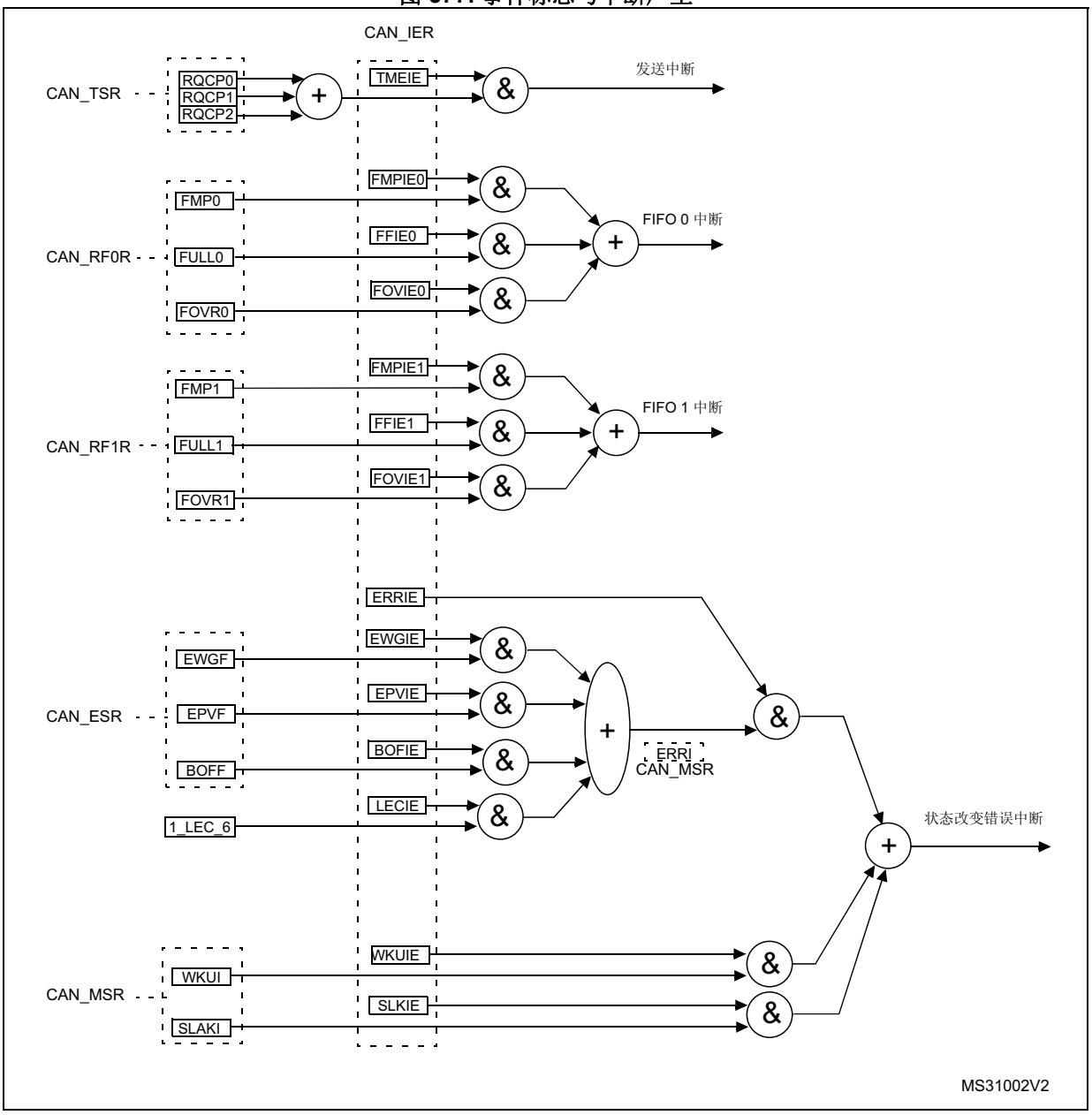
图 576. CAN 帧


帧间空间
数据帧（标准标识符）
帧间空间或过载帧
44 + 8 *N
仲裁字段
Ctrl 字段
数据字段
CRC 字段
ACK 字段
32
6
8 *N
16
2
7
ID
DLC
CRC
EOF
SOF
RTR
IDE
r0
ACK
帧间空间
数据帧（扩展标识符）
帧间空间或过载帧
64 + 8 *N
仲裁字段
仲裁字段
Ctrl 字段
数据字段
CRC 字段
ACK 字段
32
32
8 *N
16
2
7
ID
DLC
CRC
EOF
SOF
SRR
IDE
RTR
r1
r0
ACK
帧间空间
远程帧
帧间空间或过载帧
44
仲裁字段
Ctrl 字段
CRC 字段
ACK 字段
32
6
16
2
7
ID
DLC
CRC
EOF
SOF
RTR
IDE
r0
ACK
数据帧或遥控帧
错误帧
帧间空间或过载帧
错误标志
标志回显
错误分隔符
6
6
8
任意帧
帧间空间
数据帧或遥控帧
中断
暂停传输
总线空闲
3
8
帧结束或
错误分隔符或
过载分隔符
过载帧
帧间空间或错误帧
过载标志
过载回显
过载分隔符
6
6
8
注释
0 <= N <= 8
SOF = 帧起始
ID = 标识符
RTR = 远程传输请求
IDE = 标识符扩展位
r0 = 保留位
DLC = 数据长度代码
CRC = 循环冗余代码
错误标志：如果节点为错误主动节点，则为 6 个显性位，否则为 6 个隐性位。
暂停传输：仅适用于错误被动节点。
EOF = 帧结束
ACK = 确认位
Ctrl = 控制
ai15154b

49.8 bxCAN 中断

bxCAN 共有四个专用的中断向量。每个中断源均可通过 CAN 中断使能寄存器 (CAN_IER) 来单独地使能或禁止。

图 577. 事件标志与中断产生



- **发送中断**可由以下事件产生：
  - 发送邮箱 0 变为空，CAN_TSR 寄存器的 RQCP0 位置 1。
  - 发送邮箱 1 变为空，CAN_TSR 寄存器的 RQCP1 位置 1。
  - 发送邮箱 2 变为空，CAN_TSR 寄存器的 RQCP2 位置 1。
- **FIFO 0 中断**可由以下事件产生：
  - 接收到新消息，CAN_RF0R 寄存器的 FMP0 位不是“00”。
  - FIFO0 满，CAN_RF0R 寄存器的 FULL0 位置 1。
  - FIFO0 上溢，CAN_RF0R 寄存器的 FOVR0 位置 1。
- **FIFO 1 中断**可由以下事件产生：
  - 接收到新消息，CAN_RF1R 寄存器的 FMP1 位不是“00”。
  - FIFO1 满，CAN_RF1R 寄存器的 FULL1 位置 1。
  - FIFO1 上溢，CAN_RF1R 寄存器的 FOVR1 位置 1。
- **错误和状态改变中断**可由以下事件产生：
  - 错误状况，有关错误状况的更多详细信息，请参见 CAN 错误状态寄存器 (CAN_ESR)。
  - 唤醒状况，CAN Rx 信号上监测到 SOF。
  - 进入睡眠模式。

49.9 CAN 寄存器

外设寄存器必须按字（32 位）进行访问。

49.9.1 寄存器访问保护

错误访问某些配置寄存器可能会导致硬件暂时性地干扰整个 CAN 网络。因此，只有在 CAN 硬件处于初始化模式时，才可以通过软件修改 CAN_BTR 寄存器。

尽管传输错误的数据不会引发 CAN 网络级别的故障，但可能会对应用程序造成严重干扰。发送邮箱只能在处于空状态时通过软件进行修改，请参见*图 569：发送邮箱状态*。

可以通过停用相应过滤器组或将 FINIT 位置 1 来修改过滤器值。此外，只有在 CAN_FMR 寄存器的过滤器初始化模式位置 1 (FINIT=1) 时，才能对 CAN_FMxR、CAN_FSxR 和 CAN_FFAR 寄存器中的过滤器配置（大小、模式和 FIFO 分配）进行修改。

49.9.2 CAN 控制和状态寄存器

有关寄存器说明中使用的缩写，请参见*第 1.1 节*。

CAN 主控制寄存器 (CAN_MCR)

CAN master control register

偏移地址：0x00
复位值：0x0001 0002

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.	DBF
															rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESET	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TTCM	ABOM	AWUM	NART	RFLM	TXFP	SLEEP	INRQ
rs								rw	rw	rw	rw	rw	rw	rw	rw

位 31:17 保留，必须保持复位值。

位 16 **DBF**：调试冻结 (Debug freeze)
0：调试期间 CAN 处于工作状态。
1：调试期间 CAN 处于接收/发送冻结状态。接收 FIFO 仍可正常访问/控制。

位 15 **RESET**：bxCAN 软件主复位 (bxCAN software master reset)
0：正常工作。
1：强制 bxCAN 进行主复位 -> 复位后激活睡眠模式（FMP 位和 CAN_MCR 寄存器初始化为复位值）。此位自动复位为 0。

位 14:8 保留，必须保持复位值。

位 7 **TTCM**：时间触发通信模式 (Time triggered communication mode)
0：禁止时间触发通信模式。
1：使能时间触发通信模式
注：　有关时间触发通信模式的更多信息，请参见第 49.7.2 节：时间触发通信模式。

位 6 **ABOM**：自动总线关闭管理 (Automatic bus-off management)
此位控制 CAN 硬件在退出总线关闭状态时的行为。
0：在软件发出请求后，一旦监测到 128 次连续 11 个隐性位，并且软件将 CAN_MCR 寄存器的 INRQ 位先置 1 再清零，即退出总线关闭状态。
1：一旦监测到 128 次连续 11 个隐性位，即通过硬件自动退出总线关闭状态。
有关总线关闭状态的详细信息，请参见*第 49.7.6 节：错误管理*。

位 5 **AWUM**：自动唤醒模式 (Automatic wakeup mode)
此位控制 CAN 硬件在睡眠模式下接收到消息时的行为。
0：在软件通过将 CAN_MCR 寄存器的 SLEEP 位清零发出请求后，退出睡眠模式。
1：一旦监测到 CAN 消息，即通过硬件自动退出睡眠模式。
CAN_MCR 寄存器的 SLEEP 位和 CAN_MCR 寄存器的 SLAK 位由硬件清零。

位 4 **NART**：禁止自动重发送 (No automatic retransmission)
0：CAN 硬件将自动重发送消息，直到根据 CAN 标准成功发送消息。
1：无论发送结果如何（成功、错误或仲裁丢失），消息均只发送一次。

位 3 **RFLM**：接收 FIFO 锁定模式 (Receive FIFO locked mode)

0：接收 FIFO 上溢后不锁定。接收 FIFO 装满后，下一条传入消息将覆盖前一条消息。

1：接收 FIFO 上溢后锁定。接收 FIFO 装满后，下一条传入消息将被丢弃。

位 2 **TXFP**：发送 FIFO 优先级 (Transmit FIFO priority)

此位用于控制在几个邮箱同时挂起时的发送顺序。

0：优先级由消息标识符确定

1：优先级由请求顺序（时间顺序）确定

位 1 **SLEEP**：睡眠模式请求 (Sleep mode request)

此位由软件置 1，用于请求 CAN 硬件进入睡眠模式。一旦当前 CAN 活动（发送或接收 CAN 帧）结束，即进入睡眠模式。

此位由软件清零时，将退出睡眠模式。

当 AWUM 位置 1 以及在 CAN RX 信号上检测到 SOF 位时，硬件即将此位清零。

复位后，此位将置 1 - CAN 启动睡眠模式。

位 0 **INRQ**：初始化请求 (Initialization request)

软件通过将此位清零，来将硬件切换到正常模式。一旦在 Rx 信号上监测到连续 11 个隐性位，CAN 硬件即完成同步并准备进行发送和接收。硬件通过将 CAN_MSR 寄存器的 INAK 位清零来指示此事件。

软件通过将此位置 1 来请求 CAN 硬件进入初始化模式。一旦软件将 INRQ 位置 1，CAN 硬件将等待当前 CAN 活动（发送或接收）结束，然后进入初始化模式。硬件通过将 CAN_MSR 寄存器的 INAK 位置 1 来指示此事件。

CAN 主状态寄存器 (CAN_MSR)

CAN master status register

偏移地址：0x04
复位值：0x0000 0C02

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.	Res.	Res.	RX	SAMP	RXM	TXM	Res.	Res.	Res.	SLAKI	WKUI	ERRI	SLAK	INAK
				r	r	r	r				rc_w1	rc_w1	rc_w1	r	r

位 31:12 保留，必须保持复位值。

位 11 **RX**：CAN Rx 信号 (CAN Rx signal)

监视 **CAN_RX** 引脚的实际值。

位 10 **SAMP**：上一个采样点 (Last sample point)

上一个采样点的 RX 值（当前接收的位值）。

位 9 **RXM**：接收模式 (Receive mode)

CAN 硬件当前为接收器。

位 8 **TXM**：发送模式 (Transmit mode)

CAN 硬件当前为发送器。

位 7:5 保留，必须保持复位值。

位 4 **SLAKI**：睡眠确认中断 (Sleep acknowledge interrupt)

当 SLKIE=1 时，硬件将此位置 1，以指示 bxCAN 已经进入睡眠模式。如果 CAN_IER 寄存器的 SLKIE 位置 1，则当此位置 1 后将产生状态改变中断。
SLAK 清零后，此位由软件或硬件清零。

注： SLKIE=0 时，无法对 SLAKI 进行轮询。在这种情况下，可以轮询 SLAK 位。

位 3 **WKUI**：唤醒中断 (Wakeup interrupt)

此位由硬件置 1，用于指示在 CAN 硬件处于睡眠模式期间检测到一个 SOF 位。如果 CAN_IER 寄存器的 WKUIE 位置 1，则当此位置 1 后将产生状态改变中断。
此位由软件清零。

位 2 **ERRI**：错误中断 (Error interrupt)

如果在检测到错误时 CAN_ESR 的一个位置 1，并且使能 CAN_IER 中的相应中断，则硬件将此位置 1。如果 CAN_IER 寄存器的 ERRIE 位置 1，则当此位置 1 后将产生状态改变中断。
此位由软件清零。

位 1 **SLAK**：睡眠确认 (Sleep acknowledge)

此位由硬件置 1，用于向软件指示 CAN 硬件此时处于睡眠模式。此位可确认软件的睡眠模式请求（CAN_MCR 寄存器的 SLEEP 位置 1）。
CAN 硬件退出睡眠模式（以在 CAN 总线上进行同步）时，此位由硬件清零。要进行同步，硬件必须在 CAN RX 信号上监测到由 11 个连续隐性位组成的一个序列。

注： CAN_MCR 寄存器的 SLEEP 位清零时，将触发退出睡眠模式程序。有关 SLEEP 位清零的详细信息，请参阅 CAN_MCR 寄存器 AWUM 位的说明。

位 0 **INAK**：初始化确认 (Initialization acknowledge)

此位由硬件置 1，用于向软件指示 CAN 硬件此时处于初始化模式。此位可确认软件的初始化请求（CAN_MCR 寄存器的 INRQ 位置 1）。
CAN 硬件退出初始化模式（以在 CAN 总线上进行同步）时，此位由硬件清零。要进行同步，硬件必须在 CAN RX 信号上监测到由 11 个连续隐性位组成的一个序列。

CAN 发送状态寄存器 (CAN_TSR)

CAN transmit status register

偏移地址：0x08
复位值：0x1C00 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LOW2	LOW1	LOW0	TME2	TME1	TME0	CODE[1:0]		ABRQ2	Res.	Res.	Res.	TERR2	ALST2	TXOK2	RQCP2
r	r	r	r	r	r	r	r	rs				rc_w1	rc_w1	rc_w1	rc_w1
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ABRQ1	Res.	Res.	Res.	TERR1	ALST1	TXOK1	RQCP1	ABRQ0	Res.	Res.	Res.	TERR0	ALST0	TXOK0	RQCP0
rs				rc_w1	rc_w1	rc_w1	rc_w1	rs				rc_w1	rc_w1	rc_w1	rc_w1

位 31 **LOW2**：邮箱 2 最低优先级标志 (Lowest priority flag for mailbox 2)

当多个邮箱挂起等待发送且邮箱 2 优先级最低时，此位由硬件置 1。

位 30 **LOW1**：邮箱 1 最低优先级标志 (Lowest priority flag for mailbox 1)

当多个邮箱挂起等待发送且邮箱 1 优先级最低时，此位由硬件置 1。

位 29 **LOW0**：邮箱 0 最低优先级标志 (Lowest priority flag for mailbox 0)

当多个邮箱挂起等待发送且邮箱 0 优先级最低时，此位由硬件置 1。

注： 当只有一个邮箱挂起时，LOW[2:0] 位置为零。

位 28 **TME2**：发送邮箱 2 空 (Transmit mailbox 2 empty)
当邮箱 2 没有挂起的发送请求时，此位由硬件置 1。

位 27 **TME1**：发送邮箱 1 空 (Transmit mailbox 1 empty)
当邮箱 1 没有挂起的发送请求时，此位由硬件置 1。

位 26 **TME0**：发送邮箱 0 空 (Transmit mailbox 0 empty)
当邮箱 0 没有挂起的发送请求时，此位由硬件置 1。

位 25:24 **CODE[1:0]**：邮箱代码 (Mailbox code)
如果至少一个发送邮箱空闲，代码值等于下一个空闲发送邮箱的编号。
如果所有发送邮箱均挂起，则代码值等于优先级最低的发送邮箱的编号。

位 23 **ABRQ2**：邮箱 2 中止请求 (Abort request for mailbox 2)
由软件置 1，用于中止相应邮箱的发送请求。
邮箱变为空后，此位由硬件清零。
邮箱未挂起等待发送时，将此位置 1 没有任何作用。

位 22:20 保留，必须保持复位值。

位 19 **TERR2**：邮箱 2 发送错误 (Transmission error of mailbox 2)
如果上一次发送因错误而失败，此位将置 1。

位 18 **ALST2**：邮箱 2 仲裁丢失 (Arbitration lost for mailbox 2)
如果上一次发送因仲裁丢失而失败，此位将置 1。

位 17 **TXOK2**：邮箱 2 发送成功 (Transmission OK of mailbox 2)
每次发送尝试后，硬件都将更新此位。
0：上一次发送失败
1：上一次发送成功
当邮箱 2 的发送请求成功完成时，此位由硬件置 1。请参见*图 569*。

位 16 **RQCP2**：邮箱 2 请求完成 (Request completed mailbox 2)
最后一个请求（发送或中止）执行完毕时，由硬件置 1。
由软件通过写入“1”清零，或是在发生发送请求（CAN_TMID2R 寄存器中的 TXRQ2 位置 1）时由硬件清零。
如果将此位清零，邮箱 2 的所有状态位（TXOK2、ALST2 和 TERR2）都将清零。

位 15 **ABRQ1**：邮箱 1 中止请求 (Abort request for mailbox 1)
由软件置 1，用于中止相应邮箱的发送请求。
邮箱变为空后，此位由硬件清零。
邮箱未挂起等待发送时，将此位置 1 没有任何作用。

位 14:12 保留，必须保持复位值。

位 11 **TERR1**：邮箱 1 发送错误 (Transmission error of mailbox 1)
如果上一次发送因错误而失败，此位将置 1。

位 10 **ALST1**：邮箱 1 仲裁丢失 (Arbitration lost for mailbox 1)
如果上一次发送因仲裁丢失而失败，此位将置 1。

位 9 **TXOK1**：邮箱 1 发送成功 (Transmission OK of mailbox 1)
每次发送尝试后，硬件都将更新此位。
0：上一次发送失败
1：上一次发送成功
当邮箱 1 的发送请求成功完成时，此位由硬件置 1。请参见*图 569*。

位 8 **RQCP1**：邮箱 1 请求完成 (Request completed mailbox 1)
最后一个请求（发送或中止）执行完毕时，由硬件置 1。
由软件通过写入“1”清零，或是在发生发送请求（CAN_TI1R 寄存器中的 TXRQ1 位）时由硬件清零。
如果将此位清零，邮箱 1 的所有状态位（TXOK1、ALST1 和 TERR1）都将清零。

位 7 **ABRQ0**：邮箱 0 中止请求 (Abort request for mailbox 0)
由软件置 1，用于中止相应邮箱的发送请求。
邮箱变为空后，此位由硬件清零。
邮箱未挂起等待发送时，将此位置 1 没有任何作用。

位 6:4 保留，必须保持复位值。

位 3 **TERR0**：邮箱 0 发送错误 (Transmission error of mailbox 0)
如果上一次发送因错误而失败，此位将置 1。

位 2 **ALST0**：邮箱 0 仲裁丢失 (Arbitration lost for mailbox 0)
如果上一次发送因仲裁丢失而失败，此位将置 1。

位 1 **TXOK0**：邮箱 0 发送成功 (Transmission OK of mailbox 0)
每次发送尝试后，硬件都将更新此位。
0：上一次发送失败
1：上一次发送成功
当邮箱 1 的发送请求成功完成时，此位由硬件置 1。请参见*图 569*。

位 0 **RQCP0**：邮箱 0 请求完成 (Request completed mailbox 0)
最后一个请求（发送或中止）执行完毕时，由硬件置 1。
由软件通过写入“1”清零，或是在发生发送请求（CAN_TI0R 寄存器的 TXRQ0 位置 1）时由硬件清零。
如果将此位清零，邮箱 0 的所有状态位（TXOK0、ALST0 和 TERR0）都将清零。

CAN 接收 FIFO 0 寄存器 (CAN_RF0R)

CAN receive FIFO 0 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
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.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RFOM0	FOVR0	FULL0	Res.	FMP0[1:0]	
										rs	rc_w1	rc_w1		r	r

位 31:6 保留，必须保持复位值。

位 5 **RFOM0**：释放 FIFO 0 输出邮箱 (Release FIFO 0 output mailbox)
由软件置 1，用于释放 FIFO 的输出邮箱。FIFO 中至少有一条消息挂起时，才能释放输出邮箱。FIFO 为空时，将此位置 1 没有任何作用。如果 FIFO 中至少有两条消息挂起，软件必须释放输出邮箱，才能访问下一条消息。
输出邮箱释放后，此位由硬件清零。

位 4 **FOVR0**：FIFO 0 上溢 (FIFO 0 overrun)

FIFO 填满时，如果接收到新消息并且通过过滤器，此位将由硬件置 1。
此位由软件清零。

位 3 **FULL0**：FIFO 0 满 (FIFO 0 full)

FIFO 中存储三条消息后，由硬件置 1。
此位由软件清零。

位 2 保留，必须保持复位值。

位 1:0 **FMP0[1:0]**：FIFO 0 消息挂起 (FIFO 0 message pending)

这些位用于指示接收 FIFO 中挂起的消息数。
硬件每向 FIFO 存储一条新消息，FMP 即会增加。软件每次通过将 RFOM0 位置 1 来释放输出邮箱，FMP 即会减小。

CAN 接收 FIFO 1 寄存器 (CAN_RF1R)

CAN receive FIFO 1 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
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.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RFOM1	FOVR1	FULL1	Res.	FMP1[1:0]	
										rs	rc_w1	rc_w1		r	r

位 31:6 保留，必须保持复位值。

位 5 **RFOM1**：释放 FIFO 1 输出邮箱 (Release FIFO 1 output mailbox)

由软件置 1，用于释放 FIFO 的输出邮箱。FIFO 中至少有一条消息挂起时，才能释放输出邮箱。FIFO 为空时，将此位置 1 没有任何作用。如果 FIFO 中至少有两条消息挂起，软件必须释放输出邮箱，才能访问下一条消息。
输出邮箱释放后，此位由硬件清零。

位 4 **FOVR1**：FIFO 1 上溢 (FIFO 1 overrun)

FIFO 填满时，如果接收到新消息并且通过过滤器，此位将由硬件置 1。
此位由软件清零。

位 3 **FULL1**：FIFO 1 满 (FIFO 1 full)

FIFO 中存储三条消息后，由硬件置 1。
此位由软件清零。

位 2 保留，必须保持复位值。

位 1:0 **FMP1[1:0]**：FIFO 1 消息挂起 (FIFO 1 message pending)

这些位用于指示接收 FIFO1 中挂起的消息数。
硬件每向 FIFO1 存储一条新消息，FMP1 即会增加。软件每次通过将 RFOM1 位置 1 来释放输出邮箱，FMP 即会减小。

CAN 中断使能寄存器 (CAN_IER)

CAN interrupt enable 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
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SLKIE	WKUIE
														rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ERRIE	Res.	Res.	Res.	LEC IE	BOF IE	EPV IE	EWG IE	Res.	FOV IE1	FF IE1	FMP IE1	FOV IE0	FF IE0	FMP IE0	TME IE
rw				rw	rw	rw	rw		rw	rw	rw	rw	rw	rw	rw

位 31:18 保留，必须保持复位值。

位 17 **SLKIE**：睡眠中断使能 (Sleep interrupt enable)
0：SLAKI 位置 1 时不产生中断。
1：SLAKI 位置 1 时产生中断。

位 16 **WKUIE**：唤醒中断使能 (Wakeup interrupt enable)
0：WKUI 置 1 时不产生中断。
1：WKUI 位置 1 时产生中断。

位 15 **ERRIE**：错误中断使能 (Error interrupt enable)
0：CAN_ESR 中有挂起的错误状况时，不会产生中断。
1：CAN_ESR 中有挂起的错误状况时，会产生中断。

位 14:12 保留，必须保持复位值。

位 11 **LECIE**：上一个错误代码中断使能 (Last error code interrupt enable)
0：如果在检测到错误后硬件将 LEC[2:0] 中的错误代码置 1，则不会将 ERRI 位置 1。
1：如果在检测到错误后硬件将 LEC[2:0] 中的错误代码置 1，会将 ERRI 位置 1。

位 10 **BOFIE**：总线关闭中断使能 (Bus-off interrupt enable)
0：BOFF 置 1 时，不会将 ERRI 位置 1。
1：BOFF 置 1 时，将 ERRI 位置 1。

位 9 **EPVIE**：错误被动中断使能 (Error passive interrupt enable)
0：EPVF 置 1 时，不会将 ERRI 位置 1。
1：EPVF 置 1 时，将 ERRI 位置 1。

位 8 **EWGIE**：错误警告中断使能 (Error warning interrupt enable)
0：EWGF 置 1 时，不会将 ERRI 位置 1。
1：EWGF 置 1 时，将 ERRI 位置 1。

位 7 保留，必须保持复位值。

位 6 **FOVIE1**：FIFO 上溢中断使能 (FIFO overrun interrupt enable)
0：FOVR 置 1 时不产生中断。
1：FOVR 置 1 时产生中断。

位 5 **FFIE1**：FIFO 变满时中断使能 (FIFO full interrupt enable)
0：FULL 位置 1 时不产生中断。
1：FULL 位置 1 时产生中断。

位 4 **FMPIE1**：FIFO 消息挂起中断使能 (FIFO message pending interrupt enable)
0：FMP[1:0] 位的状态不是 00b 时，不产生中断。
1：FMP[1:0] 位的状态不是 00b 时，产生中断。

位 3 **FOVIE0**：FIFO 上溢中断使能 (FIFO overrun interrupt enable)
0：FOVR 位置 1 时不产生中断。
1：FOVR 位置 1 时产生中断。

位 2 **FFIE0**：FIFO 变满时中断使能 (FIFO full interrupt enable)
0：FULL 位置 1 时不产生中断。
1：FULL 位置 1 时产生中断。

位 1 **FMPIE0**：FIFO 消息挂起中断使能 (FIFO message pending interrupt enable)
0：FMP[1:0] 位的状态不是 00b 时，不产生中断。
1：FMP[1:0] 位的状态不是 00b 时，产生中断。

位 0 **TMEIE**：发送邮箱空中断使能 (Transmit mailbox empty interrupt enable)
0：RQCPx 位置 1 时不产生中断。
1：RQCPx 位置 1 时产生中断。

注： 请参见第 49.8 节：bxCAN 中断。

CAN 错误状态寄存器 (CAN_ESR)

CAN error status 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
REC[7:0]								TEC[7:0]							
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LEC[2:0]			Res.	BOFF	EPVF	EWGF
									rw	rw	rw		r	r	r

位 31:24 **REC[7:0]**：接收错误计数器 (Receive error counter)
CAN 协议故障隔离机制的实施部分。如果接收期间发生错误，该计数器按 1 或 8 递增，具体取决于 CAN 标准所定义的错误状况。每次成功接收后，该计数器按 1 递减，如果其数值大于 128，则复位为 120。计数器值超过 127 时，CAN 控制器进入错误被动状态。

位 23:16 **TEC[7:0]**：9 位发送错误计数器最低有效字节 (Least significant byte of the 9-bit transmit error counter)
CAN 协议故障隔离机制的实施部分。

位 15:7 保留，必须保持复位值。

位 6:4 **LEC[2:0]：** 上一个错误代码 (Last error code)

该字段由硬件置 1，其中的代码指示 CAN 总线上检测到的上一个错误的错误状况。如果消息成功传送（接收或发送）且未发生错误，该字段将清为“0”。

LEC[2:0] 位可由软件置为 0b111 值。这些位由硬件更新，以指示当前通信状态。

000：无错误
001：填充错误
010：格式错误
011：确认错误
100：位隐性错误
101：位显性错误
110：CRC 错误
111：由软件置 1

位 3 保留，必须保持复位值。

位 2 **BOFF：** 总线关闭标志 (Bus-off flag)

此位由硬件在进入睡眠状态时置 1。TEC 上溢（超过 255）时，进入总线关闭状态，请参见 *第 1750 页的第 49.7.6 节*。

位 1 **EPVF：** 错误被动标志 (Error passive flag)

达到错误被动极限（接收错误计数器或发送错误计数器 > 127）时，此位由硬件置 1。

位 0 **EWGF：** 错误警告标志 (Error warning flag)

达到警告极限时，此位由硬件置 1
（接收错误计数器或发送错误计数器≥96）。

CAN 位时序寄存器 (CAN_BTR)

CAN bit timing register

偏移地址：0x1C
复位值：0x0123 0000

只有 CAN 硬件处于初始化模式时，才能由软件访问此寄存器。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SILM	LBKM	Res.	Res.	Res.	Res.	SJW[1:0]		Res.	TS2[2:0]			TS1[3:0]			
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
Res.	Res.	Res.	Res.	Res.	Res.	BRP[9:0]									
						rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31 **SILM：** 静默模式（调试）(Silent mode (debug))

0：正常工作
1：静默模式

位 30 **LBKM：** 环回模式（调试）(Loop back mode (debug))

0：禁止环回模式
1：使能环回模式

位 29:26 保留，必须保持复位值。

位 25:24 **SJW[1:0]：** 再同步跳转宽度 (Resynchronization jump width)

这些位定义 CAN 硬件在执行再同步时最多可以将位加长或缩短的时间片数目。

$t_{RJW} = t_q \times (SJW[1:0] + 1)$

位 23 保留，必须保持复位值。

位 22:20 **TS2[2:0]**：时间段 2 (Time segment 2)
这些位定义时间段 2 中的时间片数目。
$t_{BS2} = t_q \times (TS2[2:0] + 1)$

位 19:16 **TS1[3:0]**：时间段 1 (Time segment 1)
这些位定义时间段 1 中的时间片数目。
$t_{BS1} = t_q \times (TS1[3:0] + 1)$
有关位时序的详细信息，请参见*第 1750 页的第 49.7.7 节：位时序*。

位 15:10 保留，必须保持复位值。

位 9:0 **BRP[9:0]**：波特率预分频器 (Baud rate prescaler)
这些位定义一个时间片的长度。
$t_q = (BRP[9:0]+1) \times t_{PCLK}$

49.9.3 CAN 邮箱寄存器

CAN mailbox registers

本章介绍发送和接收邮箱的寄存器。有关详细的寄存器映射，请参见*第 1749 页的第 49.7.5 节：消息存储*。

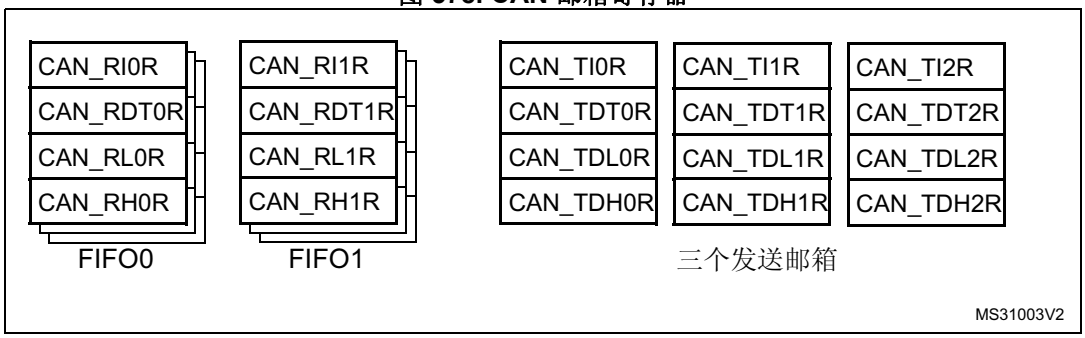
除了以下情况外，发送邮箱和接收邮箱使用相同的寄存器：

- CAN_RDTxR 寄存器中的 FMI 字段。
- 接收邮箱始终处于写保护状态。
- 发送邮箱仅在为空时才处于可写状态（CAN_TSR 寄存器中的相应 TME 位置 1）。

发送邮箱共有 3 个，接收邮箱共有 2 个。每个接收邮箱允许访问一个深度为 3 级的 FIFO，访问仅限于 FIFO 中最早接收到的消息。

每个邮箱由 4 个寄存器组成。

图 578. CAN 邮箱寄存器



CAN 发送邮箱标识符寄存器 (CAN_TIxR) (x=0..2)

CAN TX mailbox identifier register

偏移地址：0x180、0x190、0x1A0
复位值：0xXXXX XXXX（位 0 除外，TXRQ = 0）

当邮箱处于发送挂起状态（TMEx 复位）时，所有发送寄存器均为写保护状态。

该寄存器还会进行发送请求控制（位 0）- 复位值 0。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
STID[10:0]/EXID[28:18]											EXID[17:13]				
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
EXID[12:0]													IDE	RTR	TXRQ
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:21 **STID[10:0]/EXID[28:18]**：标准标识符或扩展标识符 (Standard identifier or extended identifier)
标准标识符或扩展标识符的 MSB（取决于 IDE 位的值）。

位 20:3 **EXID[17:0]**：扩展标识符 (Extended identifier)
扩展标识符的 LSB。

位 2 **IDE**：标识符扩展 (Identifier extension)
此位用于定义邮箱中消息的标识符类型。
0：标准标识符。
1：扩展标识符。

位 1 **RTR**：远程发送请求 (Remote transmission request)
0：数据帧
1：远程帧

位 0 **TXRQ**：发送邮箱请求 (Transmit mailbox request)
由软件置 1，用于请求发送相应邮箱的内容。
邮箱变为空后，此位由硬件清零。

CAN 邮箱数据长度控制和时间戳寄存器 (CAN_TDTxR) (x = 0..2)

CAN mailbox data length control and time stamp register

当邮箱未处于空状态时，该寄存器的所有位均为写保护状态。

偏移地址：0x184、0x194、0x1A4
复位值：0xXXXX XXXX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TIME[15:0]															
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
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DLC[3:0]			
												rw	rw	rw	rw

位 31:16 **TIME[15:0]**：消息时间戳 (Message time stamp)
此字段包含在进行 SOF 发送时所捕获的 16 位定时器值。

位 15:9 保留，必须保持复位值。

位 8 **TGT**：发送全局时间 (Transmit global time)
只有硬件处于时间触发通信模式（CAN_MCR 寄存器的 TTCM 位置 1）时，此位才会激活。
0：不发送时间戳 TIME[15:0]。
1：在 8 字节消息的最后两个数据字节中发送时间戳 TIME[15:0] 的值。数据字节 7 对应 TIME[7:0]，数据字节 6 对应 TIME[15:8]，该值将替换 CAN_TDHxR[31:16] 寄存器（DATA6[7:0] 和 DATA7[7:0]）中写入的数据。DLC 必须编程为 8，才能通过 CAN 总线发送这两个字节。

位 7:4 保留，必须保持复位值。

位 3:0 **DLC[3:0]**：数据长度代码 (Data length code)
该字段定义数据帧或遥控帧请求中的数据字节数。
一条消息可以包含 0 到 8 个数据字节，具体取决于 DLC 字段的值。

CAN 邮箱数据低位寄存器 (CAN_TDLxR) (x=0..2)

CAN mailbox data low register

当邮箱未处于空状态时，该寄存器的所有位均为写保护状态。

偏移地址：0x188、0x198、0x1A8
复位值：0xXXXX XXXX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DATA3[7:0]								DATA2[7:0]							
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
DATA1[7:0]								DATA0[7:0]							
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:24 **DATA3[7:0]**：数据字节 3 (Data byte 3)
消息的数据字节 3。

位 23:16 **DATA2[7:0]**：数据字节 2 (Data byte 2)
消息的数据字节 2。

位 15:8 **DATA1[7:0]**：数据字节 1 (Data byte 1)
消息的数据字节 1。

位 7:0 **DATA0[7:0]**：数据字节 0 (Data byte 0)
消息的数据字节 0。
一条消息可以包含 0 到 8 个数据字节，从字节 0 开始。

CAN 邮箱数据高位寄存器 (CAN_TDHxR) (x=0..2)

CAN mailbox data high register

当邮箱未处于空状态时，该寄存器的所有位均为写保护状态。

偏移地址：0x18C、0x19C、0x1AC
复位值：0xXXXX XXXX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DATA7[7:0]								DATA6[7:0]							
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
DATA5[7:0]								DATA4[7:0]							
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:24 **DATA7[7:0]**：数据字节 7 (Data byte 7)
消息的数据字节 7。
注： 如果 TTCM 以及此消息的 TGT 有效，则 DATA7 和 DATA6 将以时间戳值替换。

位 23:16 **DATA6[7:0]**：数据字节 6 (Data byte 6)
消息的数据字节 6。

位 15:8 **DATA5[7:0]**：数据字节 5 (Data byte 5)
消息的数据字节 5。

位 7:0 **DATA4[7:0]**：数据字节 4 (Data byte 4)
消息的数据字节 4。

CAN 接收FIFO 邮箱标识符寄存器 (CAN_RIxR) (x = 0..1)

CAN receive FIFO mailbox identifier register

偏移地址：0x1B0、0x1C0
复位值：0xXXXX XXXX

所有接收寄存器均受到写保护。

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

位 31:21 **STID[10:0]/EXID[28:18]**：标准标识符或扩展标识符 (Standard identifier or extended identifier)
标准标识符或扩展标识符的 MSB（取决于 IDE 位的值）。

位 20:3 **EXID[17:0]**：扩展标识符 (Extended identifier)
扩展标识符的 LSB。

位 2 **IDE**：标识符扩展 (Identifier extension)
此位用于定义邮箱中消息的标识符类型。
0：标准标识符。
1：扩展标识符。

位 1 **RTR**：远程发送请求 (Remote transmission request)
0：数据帧
1：远程帧

位 0 保留，必须保持复位值。

CAN 接收 FIFO 邮箱数据长度控制和时间戳寄存器 (CAN_RDTxR) (x = 0..1)

CAN receive FIFO mailbox data length control and time stamp register

偏移地址：0x1B4、0x1C4
复位值：0xXXXX XXXX

所有接收寄存器均受到写保护。

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

位 31:16 **TIME[15:0]**：消息时间戳 (Message time stamp)
此字段包含在进行 SOF 检测时所捕获的 16 位定时器值。

位 15:8 **FMI[7:0]**：过滤器匹配索引 (Filter match index)
该寄存器包含过滤器索引，邮箱中存储的消息需要经过该过滤器。有关标识符筛选的更多详细信息，请参见*第 1745 页的第 49.7.4 节：标识符筛选* - **过滤器匹配索引**一段。

位 7:4 保留，必须保持复位值。

位 3:0 **DLC[3:0]**：数据长度代码 (Data length code)
该字段定义一个数据帧所包含的数据字节数（0 到 8）。如果是远程帧请求，则为 0。

CAN 接收 FIFO 邮箱数据低位寄存器 (CAN_RDLxR) (x = 0..1)

CAN receive FIFO mailbox data low register

当邮箱未处于空状态时，该寄存器的所有位均为写保护状态。

偏移地址：0x1B8、0x1C8
复位值：0xXXXX XXXX

所有接收寄存器均受到写保护。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DATA3[7:0]								DATA2[7:0]							
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA1[7:0]								DATA0[7:0]							
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:24 **DATA3[7:0]**：数据字节 3 (Data byte 3)
消息的数据字节 3。

位 23:16 **DATA2[7:0]**：数据字节 2 (Data byte 2)
消息的数据字节 2。

位 15:8 **DATA1[7:0]**：数据字节 1 (Data byte 1)
消息的数据字节 1。

位 7:0 **DATA0[7:0]**：数据字节 0 (Data byte 0)
消息的数据字节 0。
一条消息可以包含 0 到 8 个数据字节，从字节 0 开始。

CAN 接收 FIFO 邮箱数据高位寄存器 (CAN_RDHxR) (x = 0..1)

CAN receive FIFO mailbox data high register

偏移地址：0x1BC、0x1CC
复位值：0xXXXX XXXX

所有接收寄存器均受到写保护。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DATA7[7:0]								DATA6[7:0]							
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA5[7:0]								DATA4[7:0]							
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:24 **DATA7[7:0]**：数据字节 7 (Data Byte 7)
消息的数据字节 3。

位 23:16 **DATA6[7:0]**：数据字节 6 (Data Byte 6)
消息的数据字节 2。

位 15:8 **DATA5[7:0]**：数据字节 5 (Data Byte 5)
消息的数据字节 1。

位 7:0 **DATA4[7:0]**：数据字节 4 (Data Byte 4)
消息的数据字节 0。

49.9.4 CAN 过滤器寄存器

CAN 过滤器主寄存器 (CAN_FMR)

CAN filter master register

偏移地址：0x200
复位值：0x2A1C 0E01

此寄存器的所有位均由软件置 1 和清零。

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.	CANSB[5:0]						Res.	Res.	Res.	Res.	Res.	Res.	Res.	FINIT
		rw	rw	rw	rw	rw	rw								rw

位 31:14 保留，必须保持复位值。

位 13:8 **CANSB[5:0]**：CAN 起始存储区 (CAN start bank)
这些位将由软件置 1 和清零。使用两个 CAN 时，这些位定义每个 CAN 接口的起始存储区：
000001 = 1 个过滤器分配给 CAN1，27 个过滤器分配给 CAN2
011011 = 27 个过滤器分配给 CAN1，1 个过滤器分配给 CAN2
- 要将所有过滤器分配给一个 CAN：将 CANSB 值设为零，并停用未使用的 CAN
- 要仅使用 CAN1：停止 CAN2 的时钟和/或将 CAN2 的 CAN_MCR.INRQ 置 1
- 要仅使用 CAN2：将 CAN1 的 CAN_MCR.INRQ 置 1，或停用 CAN1 的中断寄存器 CAN_IER

位 7:1 保留，必须保持复位值。

位 0 **FINIT**：过滤器初始化模式 (Filter initialization mode)
过滤器组的初始化模式
0：过滤器工作模式。
1：过滤器初始化模式。

CAN 过滤器模式寄存器 (CAN_FM1R)

CAN filter mode register

偏移地址：0x204
复位值：0x0000 0000

仅当 CAN_FMR 寄存器中设置了过滤器初始化模式 (FINIT=1) 时，才能对此寄存器执行写操作。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	FBM27	FBM26	FBM25	FBM24	FBM23	FBM22	FBM21	FBM20	FBM19	FBM18	FBM17	FBM16
				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
FBM15	FBM14	FBM13	FBM12	FBM11	FBM10	FBM9	FBM8	FBM7	FBM6	FBM5	FBM4	FBM3	FBM2	FBM1	FBM0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

注： *请参见第 1746 页上的图 571：过滤器组尺度配置 - 寄存器构成。*

位 31:28 保留，必须保持复位值。

位 27:0 **FBM*x***：过滤器模式 (Filter mode)
过滤器 x 的寄存器的模式。
0：过滤器组 x 的两个 32 位寄存器处于标识符屏蔽模式。
1：过滤器组 x 的两个 32 位寄存器处于标识符列表模式。

CAN 过滤器尺度寄存器 (CAN_FS1R)

CAN filter scale register

偏移地址：0x20C
复位值：0x0000 0000

仅当 CAN_FMR 寄存器中设置了过滤器初始化模式 (FINIT=1) 时，才能对此寄存器执行写操作。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	FSC27	FSC26	FSC25	FSC24	FSC23	FSC22	FSC21	FSC20	FSC19	FSC18	FSC17	FSC16
				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
FSC15	FSC14	FSC13	FSC12	FSC11	FSC10	FSC9	FSC8	FSC7	FSC6	FSC5	FSC4	FSC3	FSC2	FSC1	FSC0
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 27:0 **FSCx**：过滤器尺度配置 (Filter scale configuration)
这些位用于定义过滤器 27-0 尺度配置。
0：双 16 位尺度配置
1：单 32 位尺度配置

注： *请参见第 1746 页上的图 571：过滤器组尺度配置 - 寄存器构成。*

CAN 过滤器 FIFO 分配寄存器 (CAN_FFA1R)

CAN filter FIFO assignment register

偏移地址：0x214
复位值：0x0000 0000

仅当 CAN_FMR 寄存器中设置了过滤器初始化模式 (FINIT=1) 时，才能对此寄存器执行写操作。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	FFA27	FFA26	FFA25	FFA24	FFA23	FFA22	FFA21	FFA20	FFA19	FFA18	FFA17	FFA16
				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
FFA15	FFA14	FFA13	FFA12	FFA11	FFA10	FFA9	FFA8	FFA7	FFA6	FFA5	FFA4	FFA3	FFA2	FFA1	FFA0
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 27:0 **FFA*x***：过滤器 x 的过滤器 FIFO 分配 (Filter FIFO assignment for filter x)
通过此过滤器的消息将存储在指定的 FIFO 中。
0：过滤器分配给 FIFO 0
1：过滤器分配给 FIFO 1

CAN 过滤器激活寄存器 (CAN_FA1R)

CAN filter activation register

偏移地址：0x21C
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.	FACT27	FACT26	FACT25	FACT24	FACT23	FACT22	FACT21	FACT20	FACT19	FACT18	FACT17	FACT16
				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
FACT15	FACT14	FACT13	FACT12	FACT11	FACT10	FACT9	FACT8	FACT7	FACT6	FACT5	FACT4	FACT3	FACT2	FACT1	FACT0
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 27:0 **FACT*x***：过滤器激活 (Filter active)
软件将此位置 1 可激活过滤器 x。要修改过滤器 x 寄存器 (CAN_FxR[0:7])，必须将 FACTx 位清零，或将 CAN_FMR 寄存器中的 FINIT 位置 1。
0：过滤器 x 未激活
1：过滤器 x 激活

过滤器组 i 寄存器 x (CAN_FiRx)（i = 0..27，x = 1，2）

Filter bank i register x

偏移地址：0x240 到 0x31C
复位值：0xXXXX XXXX

共有 28 个过滤器组，i = 0 到 27。每个筛选器组 i 由两个 32 位寄存器组成，即 CAN_FiR[2:1]。

仅当 CAN_FAxR 寄存器的 FACTx 位清零，或者 CAN_FMR 寄存器的 FINIT 位置 1 时，才能修改此寄存器。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FB31	FB30	FB29	FB28	FB27	FB26	FB25	FB24	FB23	FB22	FB21	FB20	FB19	FB18	FB17	FB16
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
FB15	FB14	FB13	FB12	FB11	FB10	FB9	FB8	FB7	FB6	FB5	FB4	FB3	FB2	FB1	FB0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

在所有配置中：

位 31:0 **FB[31:0]**：过滤器位 (Filter bits)

标识符
寄存器的每一位用于指定预期标识符相应位的级别。
0：需要显性位
1：需要隐性位

掩码
寄存器的每一位用于指定相关标识符寄存器的位是否必须与预期标识符的相应位匹配。
0：无关，不使用此位进行比较。
1：必须匹配，传入标识符的此位必须与过滤器相应标识符寄存器中指定的级别相同。

注： *每个寄存器的功能可能因过滤器的尺度和模式配置而异。有关过滤器映射、功能说明和掩码寄存器关联的信息，请参见第 1745 页的第 49.7.4 节：标识符筛选。*

掩码模式中，掩码/标识符寄存器的位映射与**标识符列表**模式中的相同。

有关过滤器组的寄存器映射/地址信息，请参见第 1775 页上的表 314。

49.9.5 bxCAN 寄存器映射

有关寄存器边界地址的信息，请参见*第 78 页的第 2.2.2 节*。寄存器仅在 CAN1 中位于偏移量为 0x200 到 0x31C 的地址处。

表 314. bxCAN 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	**CAN_MCR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DBF	RESET	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TTCM	ABOM	AWUM	NART	RFLM	TXFP	SLEEP	INRQ
	Reset value																1	0								0	0	0	0	0	0	1	0
0x004	**CAN_MSR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RX	SAMP	RXM	TXM	Res.	Res.	Res.	SLAKI	WKUI	ERRI	SLAK	INAK
	Reset value																					1	1	0	0				0	0	0	1	0
0x008	**CAN_TSR**	LOW[2:0]			TME[2:0]			CODE[1:0]		ABRQ2	Res.	Res.	Res.	TERR2	ALST2	TXOK2	RQCP2	ABRQ1	Res.	Res.	Res.	TERR1	ALST1	TXOK1	RQCP1	ABRQ0	Res.	Res.	Res.	TERR0	ALST0	TXOK0	RQCP0
	Reset value	0	0	0	1	1	1	0	0	0				0	0	0	0	0				0	0	0	0	0				0	0	0	0
0x00C	**CAN_RF0R**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RFOM0	FOVR0	FULL0	Res.	FMP0[1:0]	
	Reset value																											0	0	0		0	0
0x010	**CAN_RF1R**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RFOM1	FOVR1	FULL1	Res.	FMP1[1:0]	
	Reset value																											0	0	0		0	0
0x014	**CAN_IER**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SLKIE	WKUIE	ERRIE	Res.	Res.	Res.	LECIE	BOFIE	EPVIE	EWGIE	Res.	FOVIE1	FFIE1	FMPIE1	FOVIE0	FFIE0	FMPIE0	TMEIE
	Reset value															0	0	0				0	0	0	0		0	0	0	0	0	0	0
0x018	**CAN_ESR**	REC[7:0]								TEC[7:0]									Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LEC[2:0]			Res.	BOFF	EPVF	EWGF
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0										0	0	0		0	0	0
0x01C	**CAN_BTR**	SILM	LBKM	Res.	Res.	Res.	Res.	SJW[1:0]		Res.	TS2[2:0]			TS1[3:0]					Res.	Res.	Res.	Res.	Res.	Res.	BRP[9:0]									
	Reset value	0	0					0	0		0	1	0	0	0	1	1							0	0	0	0	0	0	0	0	0	0
0x020-0x17F	-	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
0x180	**CAN_TI0R**	STID[10:0]/EXID[28:18]											EXID[17:0]																		IDE	RTR	TXRQ
	Reset value	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0

表 314. bxCAN 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
0x184	**CAN_TDT0R**	TIME[15:0]																Res.	Res.	Res.	Res.	Res.	Res.	Res.	TGT	Res.	Res.	Res.	Res.	DLC[3:0]			
	Reset value	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	-	-	-	-	-	-	-	x	-	-	-	-	x	x	x	x
0x188	**CAN_TDL0R**	DATA3[7:0]								DATA2[7:0]								DATA1[7:0]								DATA0[7:0]							
	Reset value	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
0x18C	**CAN_TDH0R**	DATA7[7:0]								DATA6[7:0]								DATA5[7:0]								DATA4[7:0]							
	Reset value	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
0x190	**CAN_TI1R**	STID[10:0]/EXID[28:18]								EXID[17:0]																					IDE	RTR	TXRQ
	Reset value	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0
0x194	**CAN_TDT1R**	TIME[15:0]																Res.	Res.	Res.	Res.	Res.	Res.	Res.	TGT	Res.	Res.	Res.	Res.	DLC[3:0]			
	Reset value	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	-	-	-	-	-	-	-	x	-	-	-	-	x	x	x	x
0x198	**CAN_TDL1R**	DATA3[7:0]								DATA2[7:0]								DATA1[7:0]								DATA0[7:0]							
	Reset value	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
0x19C	**CAN_TDH1R**	DATA7[7:0]								DATA6[7:0]								DATA5[7:0]								DATA4[7:0]							
	Reset value	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
0x1A0	**CAN_TI2R**	STID[10:0]/EXID[28:18]								EXID[17:0]																					IDE	RTR	TXRQ
	Reset value	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0
0x1A4	**CAN_TDT2R**	TIME[15:0]																Res.	Res.	Res.	Res.	Res.	Res.	Res.	TGT	Res.	Res.	Res.	Res.	DLC[3:0]			
	Reset value	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	-	-	-	-	-	-	-	x	-	-	-	-	x	x	x	x
0x1A8	**CAN_TDL2R**	DATA3[7:0]								DATA2[7:0]								DATA1[7:0]								DATA0[7:0]							
	Reset value	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
0x1AC	**CAN_TDH2R**	DATA7[7:0]								DATA6[7:0]								DATA5[7:0]								DATA4[7:0]							
	Reset value	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
0x1B0	**CAN_RI0R**	STID[10:0]/EXID[28:18]								EXID[17:0]																					IDE	RTR	Res.
	Reset value	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	-

表 314. bxCAN 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
0x1B4	**CAN_RDT0R**	TIME[15:0]																FMI[7:0]								Res.	Res.	Res.	Res.	DLC[3:0]			
	Reset value	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	-	-	-	-	x	x	x	x
0x1B8	**CAN_RDL0R**	DATA3[7:0]								DATA2[7:0]								DATA1[7:0]								DATA0[7:0]							
	Reset value	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
0x1BC	**CAN_RDH0R**	DATA7[7:0]								DATA6[7:0]								DATA5[7:0]								DATA4[7:0]							
	Reset value	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
0x1C0	**CAN_RI1R**	STID[10:0]/EXID[28:18]											EXID[17:0]																		IDE	RTR	Res.
	Reset value	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	-
0x1C4	**CAN_RDT1R**	TIME[15:0]																FMI[7:0]								Res.	Res.	Res.	Res.	DLC[3:0]			
	Reset value	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	-	-	-	-	x	x	x	x
0x1C8	**CAN_RDL1R**	DATA3[7:0]								DATA2[7:0]								DATA1[7:0]								DATA0[7:0]							
	Reset value	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
0x1CC	**CAN_RDH1R**	DATA7[7:0]								DATA6[7:0]								DATA5[7:0]								DATA4[7:0]							
	Reset value	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
0x1D0-0x1FF	-	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
0x208	-	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	-																																
0x20C	**CAN_FS1R**	Res.	Res.	Res.	Res.	FSC[27:0]																											
	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
0x210	-	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
0x214	**CAN_FFA1R**	Res.	Res.	Res.	Res.	FFA[27:0]																											
	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
0x218	-	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

表 314. bxCAN 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
0x21C	**CAN_FA1R**	Res.	Res.	Res.	Res.	FACT[27:0]																											
	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
0x220	-	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
0x224-0x23F	-	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
0x240	**CAN_F0R1**	FB[31:0]																															
	Reset value	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
0x244	**CAN_F0R2**	FB[31:0]																															
	Reset value	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
0x248	**CAN_F1R1**	FB[31:0]																															
	Reset value	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
0x24C	**CAN_F1R2**	FB[31:0]																															
	Reset value	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
. . . .	. . . .	. . . .																															
0x318	**CAN_F27R1**	FB[31:0]																															
	Reset value	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
0x31C	**CAN_F27R2**	FB[31:0]																															
	Reset value	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

50 USB on-the-go 全速 (OTG_FS)

50.1 简介

Portions Copyright (c) Synopsys, Inc. 保留所有权利。使用须经许可。

本节介绍了 OTG_FS 控制器的架构和编程模型。

本节中使用以下缩写：

FS	全速
LS	低速
MAC	介质访问控制器
OTG	On-the-go
PFC	数据包 FIFO 控制器
PHY	物理层
USB	通用串行总线
UTMI	USB 2.0 收发器宏单元接口 (UTMI)
ADP	连接检测协议
LPM	链路层电源管理
BCD	电池充电检测
HNP	主机协商协议
SRP	会话请求协议

参考文档如下：

- USB On-The-Go 补充标准，第 2.0 版
- 通用串行总线规范第 2.0 版
- 基于 USB 2.0 规范的 USB 2.0 链路层电源管理补充工程变更通知，2007 年 7 月 16 日
- USB 2.0 ECN 的勘误表：链路层电源管理 (LPM) - 2007 年 7 月
- 电池充电规范第 1.2 版

USB OTG 是一款双角色设备 (DRD) 控制器，同时支持从机功能和主机功能，完全符合 *USB 2.0 规范的 On-The-Go 补充标准*。此外，该控制器也可配置为“仅主机”模式或“仅从机”模式，完全符合 *USB 2.0 规范*。OTG_FS 支持下面*表 315：OTG_FS 支持的速度*中定义的速度。USB OTG 既支持 HNP 也支持 SRP。OTG 模式下需要的唯一外部器件是提供 V_{BUS} 电压的电荷泵。

表 315. OTG_FS 支持的速度

-	HS (480 Mb/s)	FS (12 Mb/s)	LS (1.5 Mb/s)
主机模式	-	X	X
设备模式	-	X	-

50.2 OTG 主要特性

主要特性可分为三类：通用特性、主机模式特性和从机模式特性。

50.2.1 通用特性

OTG_FS 接口的通用特性如下：

- 经 USB-IF 认证，符合通用串行总线规范第 2.0 版
- 模块内嵌的 PHY 还完全支持定义在标准规范 OTG 补充第 2.0 版中的 OTG 协议
  - 支持 A-B 设备识别（ID 线）
  - 支持主机协商协议 (HNP) 和会话请求协议 (SRP)
  - 允许主机关闭 V_{BUS} 以在 OTG 应用中节省电池电量
  - 支持通过内部比较器对 V_{BUS} 电平采取 OTG 监控
  - 支持主机到从机的角色动态切换
- 可通过软件配置为以下角色：
  - 具有 SRP 功能的 USB FS 从机（B 设备）
  - 具有 SRP 功能的 USB FS/LS 主机（A 设备）
  - USB On-The-Go 全速双角色设备
- 支持 FS SOF 和 LS Keep-alive 令牌
  - SOF 脉冲可通过 PAD 输出
  - SOF 脉冲通过内部连接到定时器 (TIMx)
  - 可配置的帧周期
  - 可配置的帧结束中断
- 具有省电功能，例如在 USB 挂起期间停止系统、关闭数字模块内部时钟域、对 PHY 和 DFIFO 电源加以管理。
- 具有采用高级 FIFO 控制的 1.25 KB 专用 RAM：
  - 可将 RAM 空间划分为不同 FIFO，以便灵活有效地使用 RAM
  - 每个 FIFO 可存储多个数据包
  - 动态分配存储区
  - FIFO 大小可配置为非 2 的幂次方值，以便连续使用存储单元
- 一帧之内可以无需要应用程序干预，以达到最大 USB 带宽
- 它支持电池充电规范第 1.2 版中介绍的充电端口检测（仅限 FS PHY 收发器）
- 它支持连接检测协议 (ADP)

50.2.2 主机模式特性

OTG_FS 接口在主机模式下具有以下主要特性和要求：

- 通过外部电荷泵生成 V_{BUS} 电压。
- 多达 12 个主机通道（又称之为管道）：每个通道都可以动态实现重新配置，可支持任何类型的 USB 传输。
- 内置硬件调度器可：
  - 在周期性硬件队列中存储多达 12 个中断加同步传输请求
  - 在非周期性硬件队列中存储多达 12 个控制加批量传输请求
- 管理一个共享 Rx FIFO、一个周期性传输 Tx FIFO 和一个非周期性传输 Tx FIFO，以有效使用 USB 数据 RAM。

50.2.3 从机模式特性

OTG_FS 接口在从机模式下具有以下特性：

- 1 个双向控制端点 0
- 5 个 IN 端点 (EP)，可配置为支持批量传输、中断传输或同步传输
- 5 个 OUT 端点，可配置为支持批量传输、中断传输或同步传输
- 管理一个共享 Rx FIFO 和一个 Tx-OUT FIFO，以高效使用 USB 数据 RAM
- 管理多达 6 个专用 Tx-IN FIFO（分别用于每个使能的 IN EP），以降低应用程序负荷
- 支持软断开功能

50.3 OTG 实现

表 316. OTG 实现(1)

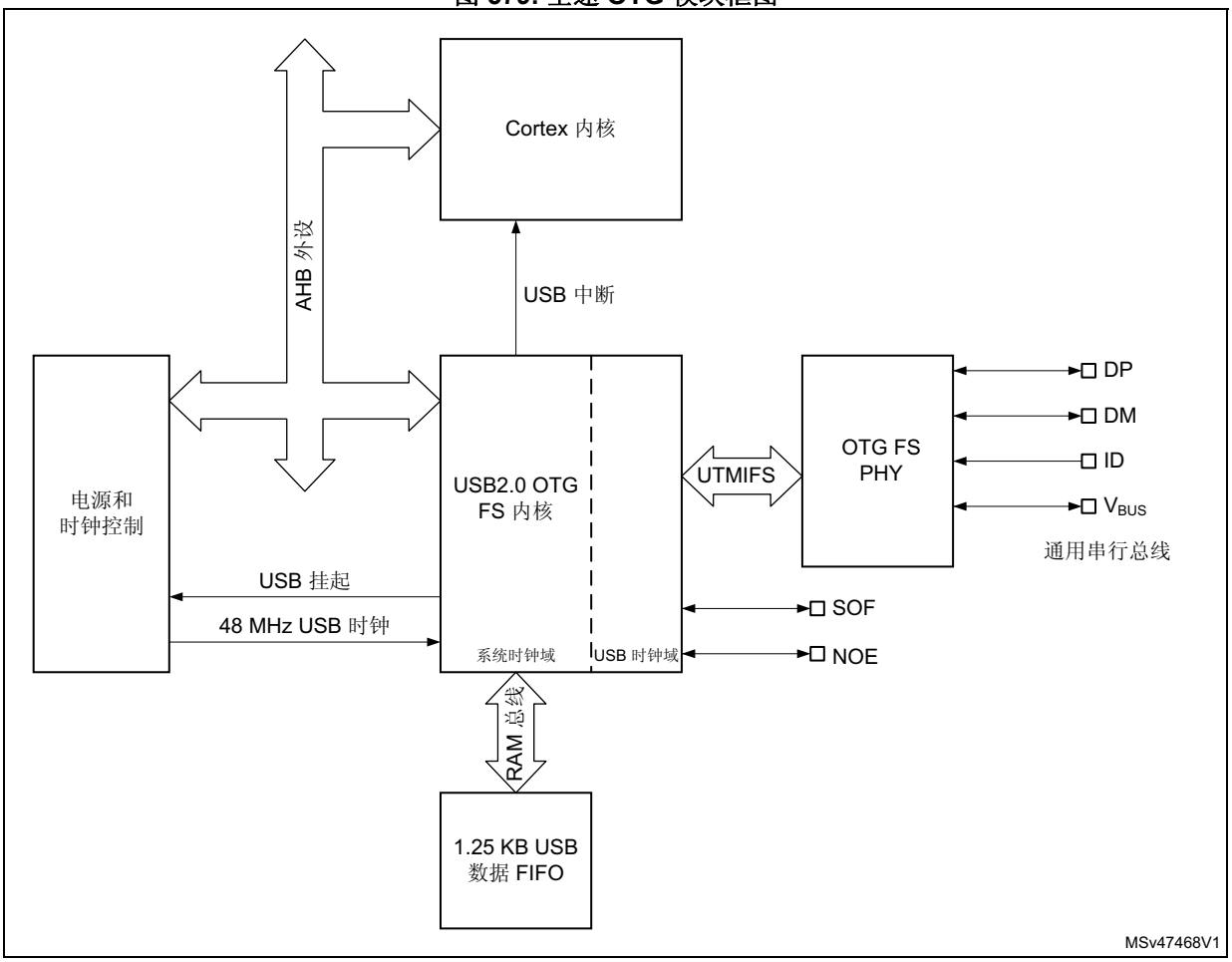
USB 功能	OTG_FS
设备双向端点（包括 EP0）	6
主机模式通道	12
专用 SRAM 的大小	1.2 KB
USB 2.0 链路层电源管理 (LPM)	X
支持 OTG 版本	2.0
连接检测协议 (ADP) 支持	X
电池充电检测 (BCD) 支持	X
PSRST/ERRATIM/STSPHST/CURMOD 位	X

1. “X” = 支持，“-” = 不支持

50.4 OTG 功能说明

50.4.1 OTG 框图

图 579. 全速 OTG 模块框图



50.4.2 USB OTG 引脚和内部信号

表 317. OTG_FS 输入/输出引脚

信号名称	信号类型	说明
OTG_FS_DP	数字输入/输出	USB OTG D+ 线
OTG_FS_DM	数字输入/输出	USB OTG D- 线
OTG_FS_ID	数字输入	USB OTG ID
OTG_FS_VBUS	模拟输入	USB OTG VBUS
OTG_FS_SOF	数字输出	USB OTG 帧起始位（可见）
OTG_FS_NOE	数字输出	D+/D- 的 USB OTG 输出使能（可见）

表 318. OTG_FS 输入/输出信号

信号名称	信号类型	说明
usb_sof	数字输出	USB OTG 帧起始事件（用于片上外设）
usb_wkup	数字输出	USB OTG 唤醒事件输出
usb_gbl_it	数字输出	USB OTG 全局中断

50.4.3 OTG 模块

USB OTG 通过复位和时钟控制模块 (RCC) 接收来自外部晶振的 48 MHz 的时钟。USB 时钟用于在全速通信时驱动 48 MHz 时钟域，必须在配置 OTG 模块前使能。

CPU 通过 AHB 外设总线对 OTG 模块寄存器进行读写操作，通过*第 50.13 节：OTG_FS 中断*中所述的 USB OTG 中断线接收 USB 事件通知。

CPU 通过向特定的 OTG 单元（压栈寄存器）写入 32 位字来向 USB 提交数据。数据随即自动存储到 USB 数据 RAM 中配置的数据发送 FIFO 中。每个 IN 端点（从机模式）或 OUT 通道（主机模式）都有一个 Tx FIFO 压栈寄存器。

CPU 从特定的 OTG 地址（出栈寄存器）读取 32 位字，以读取来自 USB 的数据。数据随即从在 1.25 KB USB 数据 RAM 内配置的共享 Rx FIFO 中弹出。每个 OUT 端点或 IN 通道都有一个 Rx FIFO 出栈寄存器。

USB 协议层通过串行接口引擎 (SIE) 驱动，并通过片上物理层 (PHY) 中的收发器模块经由 USB 进行数据的串行通信。

50.4.4 全速 OTG PHY

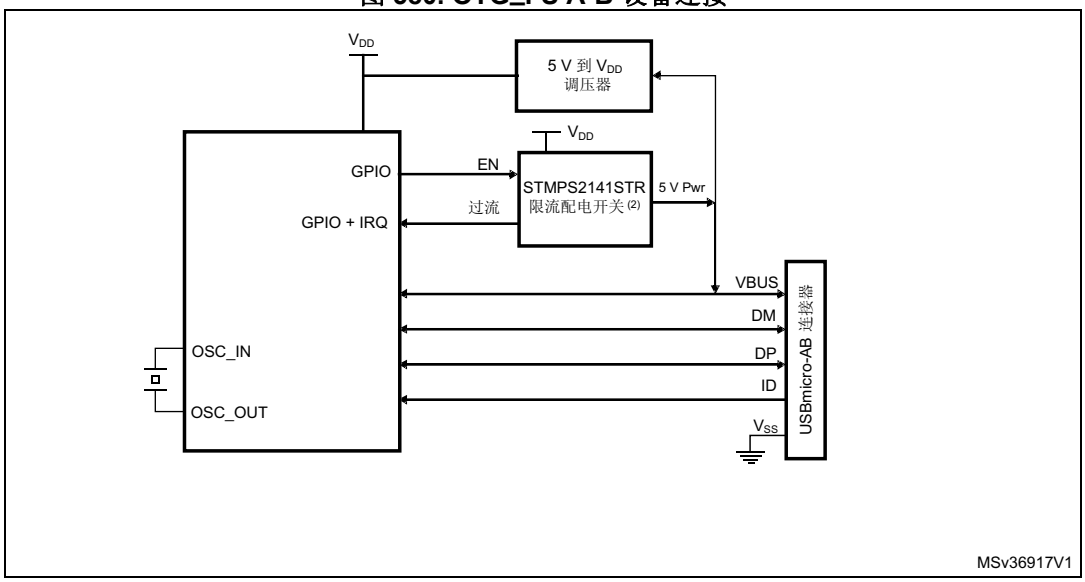
嵌入式全速 OTG PHY 由 OTG FS 模块控制，通过 UTMI+ 总线 (UTMIFS) 的全速子集传送 USB 控制和数据信号。为 USB 连接提供物理支持。
全速 OTG PHY 包括以下组成部分：

- 供主机和设备使用的 FS/LS 收发器模块。直接在单端 USB 线上驱动发送和接收操作。
- 集成 ID 上拉电阻，用于对 ID 线进行采样，以便识别 A/B 设备。
- 由 OTG_FS 模块控制的 DP/DM 集成上拉电阻和下拉电阻，具体使能哪种电阻取决于设备的当前角色。作为外设使用时，只要检测到 V_{BUS} 为有效电平（B 会话有效），立即使能 DP 上拉电阻，以告知全速设备的连接。主机模式下则使能 DP/DM 上的下拉电阻。通过主机协商协议 (HNP) 更改设备角色时，将在上拉电阻和下拉电阻之间动态切换。
- 上拉/下拉电阻 ECN 电路。根据适用于 USB 2.0 版本的电阻 ECN 规定，DP 上拉电路包括两个由 OTG_FS 单独进行控制的电阻。对 DP 上拉阻值的动态调整可以提高噪声抑制能力和 Tx/Rx 信号质量。
- 带滞回功能的 V_{BUS} 感应比较器，用于检测 V_{BUS} 有效、A-B 会话有效和会话端电压阈值。执行 USB 操作期间，这些比较器用于驱动会话请求协议 (SRP)、检测会话的有效启动和结束条件，以及持续监视 V_{BUS} 供电情况。
- V_{BUS} 脉冲电路，用于在 SRP 期间通过电阻对 V_{BUS} 充电/放电（驱动力较弱）。

注意： 为确保 USB OTG FS 模块正常工作，AHB 频率应大于 14.2 MHz。

50.5 OTG 双角色设备 (DRD)

图 580. OTG_FS A-B 设备连接



1. 只有在构建由 VBUS 供电的设备时才需要外部调压器。
2. 只有在应用必须支持由 VBUS 供电的设备时才需要 STMPS2141STR。如果应用电路板提供 5 V 电源，则可以使用基本电源开关。

50.5.1 ID 线检测

采取主机还是从机（默认设置）角色取决于 ID 输入引脚的电平。插入 USB 电缆时可根据是 MicroA 还是 MicroB 插头连接到 micro-AB 插座来确定 ID 线状态。

- 如果 USB 电缆的 B 端连入，其 ID 线浮空，则由于设备在 ID 线上的集成上拉电阻设备将检测到 ID 高电平并确认采取默认的从机角色。在此配置中，OTG_FS 符合“USB2.0 On-The-Go 规范第 2.0 版补充标准中第 4.2.4 节 ID 引脚”中所述的 FSM 标准。
- 如果 USB 电缆的 A 端连入，其 ID 线接地，则 OTG_FS 将发出 ID 线状态更改中断（OTG_GINTSTS 中的 CIDSCHG 位）以初始化主机软件，并自动切换为主机角色。在此配置中，OTG_FS 符合“USB2.0 On-The-Go 规范第 2.0 版补充标准中第 4.2.4 节 ID 引脚”中所述的 FSM 标准。

50.5.2 HNP 双角色设备

全局 USB 配置寄存器中的 HNP 使能位（OTG_ GUSBCFG 中的 HNPCAP 位）可使 OTG_FS 模块根据主机协商协议 (HNP) 动态切换角色，例如从 A 主机切换为 A 从机（反之亦然），或者从 B 从机切换为 B 主机（反之亦然）。通过全局 OTG 控制和状态寄存器中的连接器 ID 状态位（OTG_GOTGCTL 中的 CIDSTS 位）及全局中断和状态寄存器中的当前工作模式位（OTG_GINTSTS 中的 CMOD 位）二者的组合值可读取设备当前状态。

*第 50.16 节：OTG_FS 编程模块*详细介绍了 HNP 编程模型。

50.5.3 SRP 双角色设备

全局 USB 配置寄存器中的 SRP 使能位（OTG_GUSBCFG 中的 SRPCAP 位）可使 OTG_FS 模块关闭 V_{BUS} 供电，为 A 设备节省电能。注意，无论 OTG_FS 采取主机角色还是从机角色，A 设备将始终负责 V_{BUS} 的提供。

*第 50.16 节：OTG_FS 编程模块*详细介绍了 SRP A/B 设备编程模型。

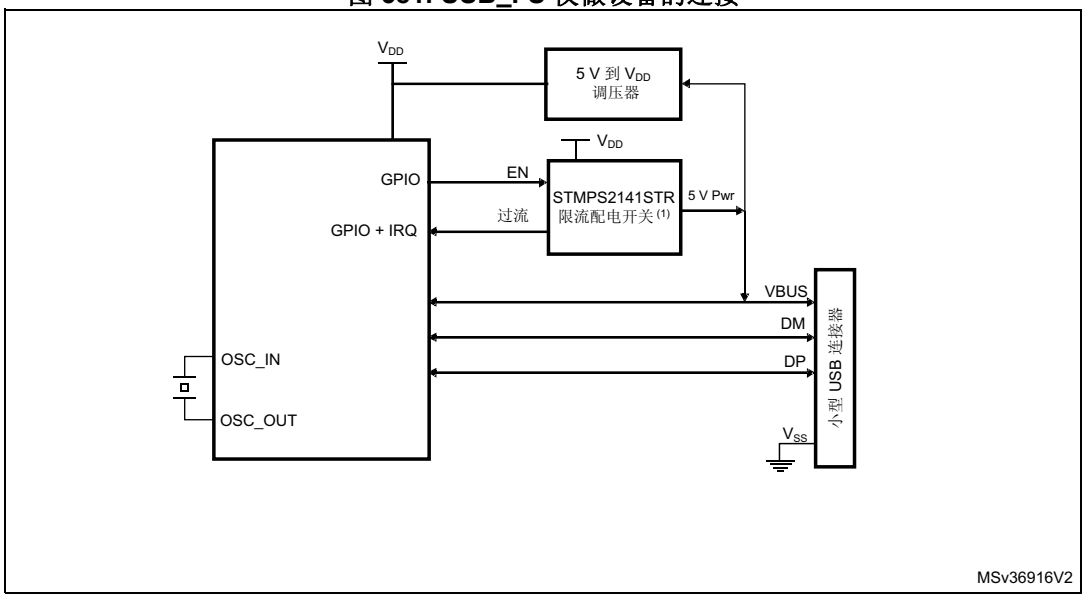
50.6 USB 设备

本节介绍了 OTG_FS 在 USB 设备模式下所具有的功能。在以下情形下，OTG_FS 用作 USB 设备：

- OTG B 设备
  - OTG B 设备插入 USB 电缆 B 端时的默认状态
- OTG A 设备
  - HNP 将 OTG_FS 切换到其设备角色后的 OTG A 设备状态
- B 设备
  - 如果 ID 线有连接，设备与 USB 电缆的 B 端相连，并且全局 USB 配置寄存器中的 HNP 功能位（OTG_GUSBCFG 中的 HNPCAP 位）清零。
- 仅作设备（请参见*图 581：USB_FS 仅做设备的连接*）
  - 将*第 50.15.4 节：OTG USB 配置寄存器 (OTG_GUSBCFG)* 中的强制设备模式位 (FDMOD) 置 1，从而将 OTG_FS 模块强制用作纯 USB 设备。这种情况下，即使 USB 连接器上存在 ID 线，也会将该 ID 线忽略。

注： *要在 B 设备或仅作设备配置情形下构建总线供电的设备方案，需要添加一个外部调压器，用于从 V_{BUS} 生成所需电源。*

图 581. USB_FS 仅做设备的连接



1. 使用调压器构建总线供电设备。

50.6.1 支持 SRP 功能的 USB 设备

全局 USB 配置寄存器中的 SRP 功能位（OTG_GUSBCFG 中的 SRPCAP 位）可使 OTG_FS 支持会话请求协议 (SRP)。这样一来，远程 A 设备便可以在 USB 会话挂起时，通过关闭 V_{BUS} 来节省电能。

*B 设备会话请求协议*一节详细介绍了 SRP 设备的编程模型。

50.6.2 USB 设备状态

供电状态

V_{BUS} 输入检测到 B 会话有效电压，就会使 USB 设备进入供电状态（请参见 USB2.0 第 9.1 部分）。然后，OTG_FS 自动连接 DP 上拉电阻，发出全速设备与主机相连的信号并生成会话请求中断（OTG_GINTSTS 中的 SRQINT 位），指示进入供电状态。

此外，V_{BUS} 输入还可确保主机在 USB 操作期间提供有效的 V_{BUS} 电平。如果检测到 V_{BUS} 降至 B 会话有效电压以下（例如，因电源干扰或主机端口关闭引发），OTG_FS 将自动断开连接并生成检测到会话结束中断（OTG_GOTGINT 中的 SEDET 位），指示 OTG_FS 已退出供电状态。

供电状态下，OTG_FS 期望收到来自主机的复位信号。其他 USB 操作则无法执行。收到复位信号后，立即生成检测到复位中断（OTG_GINTSTS 中的 USBRST）。复位信号结束后，将生成枚举完成中断（OTG_GINTSTS 中的 ENUMDNE 位），OTG_FS 随即进入默认状态。

软断开

供电状态可借助软断开功能通过软件退出。将设备控制寄存器中的软断开位（OTG_DCTL 中的 SDIS 位）置 1 即可移除 DP 上拉电阻，此时尽管没有从主机端口实际拔出 USB 电缆，但主机端仍会发生设备断开检测中断。

默认状态

默认状态下，OTG_FS 期望从主机收到 SET_ADDRESS 命令。其他 USB 操作则无法执行。当 USB 上解码出有效 SET_ADDRESS 命令时，应用程序会将相应的数值写入设备配置寄存器中的设备地址字段（OTG_DCFG 中的 DAD 位）。OTG_FS 随即进入地址状态，并准备好以所配置的 USB 地址对主机事务进行应答。

挂起状态

OTG_FS 设备持续监视 USB 活动。在 USB 空闲时间达到 3 ms 后，将发出早期挂起中断（OTG_GINTSTS 中的 ESUSP 位），并在 3 ms 后由挂起中断（OTG_GINTSTS 中的 USBSUSP 位）确认设备进入挂起状态。然后，设备状态寄存器中的设备挂起位（OTG_DSTS 中的 SUSPSTS 位）自动置 1，OTG_FS 随即进入挂起状态。

可通过设备本身退出挂起状态。这种情况下，应用程序会将设备控制寄存器中的远程唤醒信号位（OTG_DCTL 中的 RWUSIG 位）置 1，并在 1 ms 到 15 ms 后将其清零。

但若设备检测到主机发出的恢复信号时，将生成恢复中断（OTG_GINTSTS 中的 WKUPINT 位），设备挂起位自动清零。

50.6.3 USB 设备端点

OTG_FS 模块实现了以下 USB 端点：

- 控制端点 0：
  - 双向且仅处理控制消息
  - 使用一组单独的寄存器来处理 IN 和 OUT 事务
  - 专用控制 (OTG_DIEPCTL0/OTG_DOEPCTL0) 寄存器、传输配置 (OTG_DIEPTSIZ0/OTG_DOEPTSIZ0) 寄存器和状态中断 (OTG_DIEPINT0/OTG_DOEPINT0) 寄存器。控制和传输大小寄存器中可用的位组与其他端点中稍有不同
- 5 个 IN 端点
  - 每个端点都可配置为支持同步传输、批量传输或中断传输类型
  - 每个端点都有专用控制 (OTG_DIEPCTL*x*) 寄存器、传输配置 (OTG_DIEPTSIZ*x*) 寄存器和状态中断 (OTG_DIEPINT*x*) 寄存器
  - 设备 IN 端点通用中断屏蔽寄存器 (OTG_DIEPMSK) 可用于使能/禁止所有 IN 端点（包括 EP0）上的同一类端点中断源
  - 支持未完成的同步 IN 传输中断（OTG_GINTSTS 中的 IISOIXFR 位），该中断将在当前帧中至少有一个同步 IN 端点上的传输未完成时触发。该中断和周期帧结束中断 (OTG_GINTSTS/EOPF) 一起触发。
- 5 个 OUT 端点
  - 每个端点都可配置为支持同步传输、批量传输或中断传输类型
  - 每个端点都有专用控制 (OTG_DOEPCTL*x*) 寄存器、传输配置 (OTG_DOEPTSIZ*x*) 寄存器和状态中断 (OTG_DOEPINT*x*) 寄存器
  - 设备 OUT 端点通用中断屏蔽寄存器 (OTG_DOEPMSK) 可用于使能/禁止所有 OUT 端点（包括 EP0）上的同一类端点中断源
  - 支持未完成的同步 OUT 传输中断（OTG_GINTSTS 中的 INCOMPISOOUT 位），该中断将在当前帧中至少有一个同步 OUT 端点上的传输未完成时触发。该中断和周期帧结束中断 (OTG_GINTSTS/EOPF) 一起触发

端点控制

- 应用程序可通过设备端点 *x* IN/OUT 控制寄存器 (OTG_DIEPCTL*x*/OTG_DOEPCTL*x*) 对端点采取以下控制：
  - 端点使能/禁止
  - 在当前配置下激活端点
  - 设置 USB 传输类型（同步、批量和中断）
  - 设置支持的数据包大小
  - 设置与 IN 端点相关的 Tx FIFO 编号
  - 设置希望收到的或发送时要使用到的 data0/data1 PID（仅限批量/中断传输）
  - 设置接收或发送事务时所对应的奇/偶帧（仅限同步传输）
  - 可以设置 NAK 位，从而不论此时 FIFO 的状态如何，都对主机的请求回复 NAK
  - 可以设置 STALL 位，使得主机对该端点的令牌都被硬件回复 STALL
  - 可以将 OUT 端点设置为侦听模式，即对接收到的数据不进行 CRC 检查

端点传输

设备端点 *x* 传输大小寄存器 (OTG_DIEPTSIZ*x*/OTG_DOEPTSIZ*x*) 允许应用程序对传输大小参数进行编程并读取传输状态。必须在端点控制寄存器中的端点使能位置 1 之前完成对此寄存器的设置。使能端点后，这些字段立即变为只读状态，同时 OTG_FS 模块根据当前传输状态对这些字段进行更新。

可对以下传输参数进行编程：

- 以字节为单位的传输大小
- 构成整个传输的数据包个数

端点状态/中断

设备端点 *x* 中断寄存器 (OTG_DIEPINT*x*/OTG_DOPEPINT*x*) 指示端点在出现 USB 和 AHB 相关事件时的状态。当模块中断寄存器中的 OUT 端点中断位或 IN 端点中断位（分别为 OTG_GINTSTS 中的 OEPINT 位或 OTG_GINTSTS 中的 IEPINT 位）置 1 时，应用程序必须读取这些寄存器以获得详细信息。在应用程序读取这些寄存器之前，必须先读取设备全体端点中断 (OTG_DAINT) 寄存器，以获取设备端点 *x* 中断寄存器的端点编号。应用程序必须将此寄存器中的相应位清零，才能将 OTG_DAINT 和 OTG_GINTSTS 寄存器中的相应位清零。

模块提供以下状态检查和中断产生功能：

- 传输完成中断，指示应用程序 (AHB) 和 USB 端均已完成数据传输
- Setup 阶段已完成（仅针对控制传输类型的 OUT 端点）
- 相关的发送 FIFO 为半空或全空状态（IN 端点）
- NAK 应答已发送到主机（仅针对同步传输的 IN 端点）
- Tx FIFO 为空时接收到 IN 令牌（仅针对批量和中断传输类型的 IN 端点）
- 尚未使能端点时接收到 OUT 令牌
- 检测到串扰错误
- 应用程序关闭端点生效
- 应用程序对端点设置 NAK 生效（仅针对同步传输类型的 IN 端点）
- 接收到 3 个以上连续 setup 数据包（仅针对控制类型的 OUT 端点）
- 检测到超时状况（仅针对控制传输类型的 IN 端点）
- 同步传输类型的数据包未产生中断而丢失

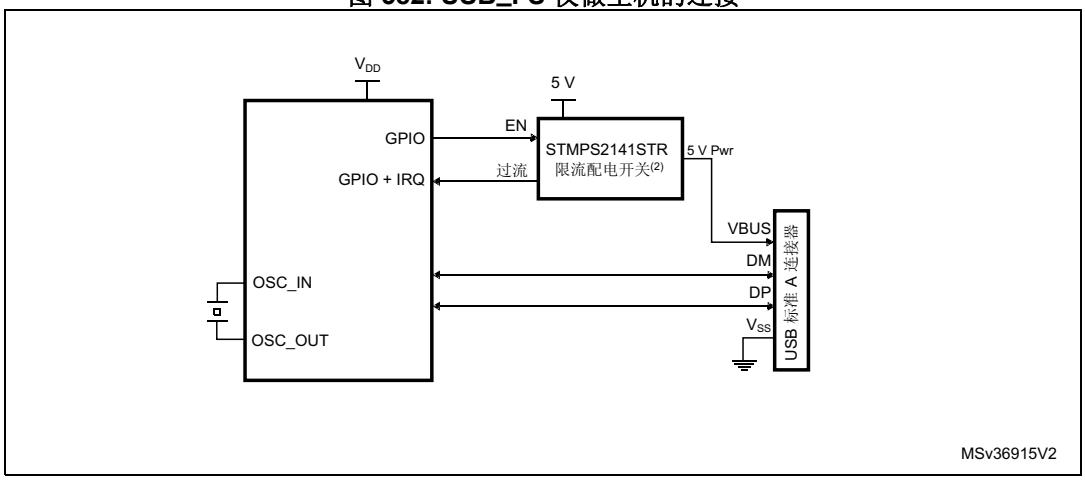
50.7 USB 主机

本节介绍了 OTG_FS 在 USB 主机模式下所具有的功能。在以下情形下，OTG_FS 用作 USB 主机：

- OTG A 主机
  - OTG A 设备在插入 USB 电缆 A 端时的默认状态
- OTG B 主机
  - OTG B 设备被 HNP 切换为主机角色后的状态
- A 主机
  - 如果 ID 线有连接，设备与 USB 电缆的 A 端相连，全局 USB 配置寄存器中的 HNP 功能位（OTG_GUSBCFG 中的 HNPCAP 位）清零。DP/DM 线上的集成下拉电阻自动使能。
- 仅作主机
  - *OTG USB 配置寄存器 (OTG_GUSBCFG)* 中的强制主机模式位 (FHMOD) 将 OTG_FS 模块强制用作纯 USB 设备。这种情况下，即使 USB 连接器上存在 ID 线，模块也会忽略 ID 线上的电平。DP/DM 线上的集成下拉电阻自动使能。

注： *微控制器不能输出 5 V 以提供 V_{BUS}。为此，必须在微控制器以外添加电荷泵或电源开关（如果应用电路板提供 5 V 电源）来驱动 5 V V_{BUS} 线。外部电荷泵可通过任何 GPIO 输出驱动。OTG A 主机、A 设备和仅作主机配置都需要使用电荷泵。*

图 582. USB_FS 仅做主机的连接



1. V_{DD} 范围介于 2 V 到 3.6 V 之间。

50.7.1 支持 SRP 功能的 USB 主机

全局 USB 配置寄存器中的 SRP 使能位（OTG_GUSBCFG 中的 SRPCAP 位）可提供 SRP 支持。使能 SRP 功能后，主机可在 USB 会话挂起时通过关闭 V_{BUS} 电源来节省电能。

*A 设备会话请求协议*一节详细介绍了 SRP 主机的编程模型。

50.7.2 USB 主机状态

给主机端口供电

微控制器不能输出 5 V 以提供 V_{BUS}。为此，必须在微控制器以外添加电荷泵或电源开关（如果应用电路板提供 5 V 电源）来驱动 5 V V_{BUS} 线。外部电荷泵可通过任何 GPIO 输出驱动，或者通过连接至外部 PMIC（电源管理 IC）的 I^2C 接口驱动。当应用程序确定控制 V_{BUS}，还必须将主机端口控制和状态寄存器中的端口电源位（OTG_HPRT 中的 PPWR 位）置 1。

V_{BUS} 有效

使能 HNP 或 SRP 后，应将 V_{BUS} 感应引脚连接到 V_{BUS}。V_{BUS} 输入可确保电荷泵在 USB 操作期间提供有效的 V_{BUS} 电平。如果 V_{BUS} 电压意外降至 V_{BUS} 有效阈值 (4.4 V) 以下，将通过会话结束检测位（OTG_GOTGINT 中的 SEDET 位）触发 OTG 中断。之后应用必须断开 V_{BUS} 电源并使端口电源位清零。

在 HNP 和 SRP 同时关闭的情况下，无需将 VBUS 感应引脚连接到 V_{BUS}。

电荷泵过流标志也可用来防止电气损坏。将电荷泵的过流标志输出连接到任意 GPIO 输入，然后将其配置为出现有效电平时生成端口中断。过流 ISR 必须立即关闭 V_{BUS} 并清零端口电源位。

主机检测设备连接

如果使能 SRP 或 HNP，即使可以随时连接 USB 设备或 B 设备，但是 OTG_FS 也只有在 V_{BUS} 有效后 (5 V) 才能检测到设备的连接。当 V_{BUS} 处于有效电平且已连接远程 B 设备时，OTG_FS 模块将发出主机端口中断信号，该中断由主机端口控制和状态寄存器中的设备连接位（OTG_HPRT 中的 PCDET 位）触发。

在 HNP 和 SRP 同时关闭的情况下，USB 设备或 B 设备将在连接后立即被检测到。OTG_FS 模块将发出主机端口中断信号，该中断由主机端口控制和状态寄存器中的设备连接位（OTG_HPRT 中的 PCDET 位）触发。

主机检测设备断开

设备断开事件将触发断开连接检测中断（OTG_GINTSTS 中的 DISCINT 位）。

主机枚举

检测到设备连接后，若又有新的设备连接进来，主机必须通过向新的设备发送 USB 复位和配置命令来启动枚举过程。

开始驱动 USB 复位前，应用程序必须等待去抖动完成位（OTG_GOTGINT 中的 DBCDNE 位）触发 OTG 中断，这表示由于在 DP (FS) 或 DM (LS) 上连接上拉电阻而发生电气抖动之后，总线恢复稳定状态。

应用程序通过将主机端口控制和状态寄存器中的端口复位位（OTG_HPRT 中的 PRST 位）置 1，并保持最少 10 ms，最多 20 ms，来在 USB 总线上发出 USB 复位信号（单端零）。应用程序计算这个过程的持续时间，然后将端口复位位清零。

USB 复位序列完成后，端口使能/禁止更改位（OTG_HPRT 中的 PENCHNG 位）立即触发主机端口中断，进而向应用程序发出通知，指示可从主机端口控制和状态寄存器中的端口速度字段（OTG_HPRT 中的 PSPD）读取枚举的设备速度，以及主机已经开始驱动 SOF (FS) 或 Keep-alive 令牌 (LS)。此时主机已就绪，可通过对设备发送命令来完成对设备的枚举。

主机挂起

应用程序通过将主机端口控制和状态寄存器中的端口挂起位（OTG_HPRT 中的 PSUSP）置 1 来挂起 USB 活动。OTG_FS 模块停止发送 SOF 并进入挂起状态。

可由远程设备的自主活动（远程唤醒）使总线退出挂起状态。这种情况下，远程唤醒信号将触发远程唤醒中断（OTG_GINTSTS 中的 WKUPINT 位），硬件把主机端口控制和状态寄存器中的端口恢复位（OTG_HPRT 中的 PRES 位）自动置位，并通过 USB 自动驱动恢复信号。应用程序必须为恢复窗口定时，然后将端口恢复位清零以退出挂起状态并重新发送 SOF。

如果由主机发起退出挂起状态，则应用程序必须将端口恢复位置 1 以启动主机端口上的恢复信号，为恢复窗口定时并最终将端口恢复位清零。

50.7.3 主机通道

OTG_FS 模块实现了 12 个主机通道。每个主机通道均可用于 USB 主机传输（USB 管道）。主机最多能同时处理 12 个传输请求。如果应用程序有 12 个以上的传输请求挂起，则在通道从之前任务释放后（即，接收到传输完成和通道停止中断后），主机控制器驱动器 (HCD) 必须为未处理的传输请求重新对通道进行分配。

每个主机通道都可配置为支持输入/输出以及周期性/非周期性事务。每个主机通道都使用专用控制 (OTG_HCCHAR*x*) 寄存器、传输配置 (OTG_HCTSIZ*x*) 寄存器状态/中断 (OTG_HCINT*x*) 寄存器以及和其相关的中断屏蔽寄存器 (OTG_HCINTMSK*x*)。

主机通道控制

- 应用程序可通过主机通道 *x* 特性寄存器 (OTG_HCCHAR*x*) 对主机通道作以下控制：
  - 通道使能/禁止
  - 设置目标 USB 设备的 FS/LS 速度
  - 设置目标 USB 设备的地址
  - 设置与该通道通信的目标 USB 设备上的端点的编号
  - 设置该通道上的传输方向：IN/OUT
  - 设置该通道上的 USB 传输的类型：控制/批量/中断/同步
  - 设置与该通道通信的设备端点的最大包长
  - 设置要进行周期传输的帧：奇帧/偶帧

主机通道传输

主机通道传输大小寄存器 (OTG_HCTSIZ*x*) 允许应用程序对传输大小参数进行编程并读取传输状态。必须在主机通道特性寄存器中的通道使能位置 1 之前完成对此寄存器的设置。使能端点后，数据包计数字段立即变为只读状态，同时 OTG_FS 模块根据当前传输状态对该字段进行更新。

- 可对以下传输参数进行编程：
  - 以字节为单位的传输大小
  - 构成整个传输大小的数据包个数
  - 初始数据 PID

主机通道状态/中断

主机通道 *x* 中断寄存器 (OTG_HCINT*x*) 指示通道在出现 USB 和 AHB 相关事件时的状态。当模块中断寄存器中的主机通道中断位（OTG_GINTSTS 中的 HCINT 位）置 1 时，应用程序必须读取这些寄存器以获得详细信息。在读取这些寄存器之前，应用程序必须先读取主机全体通道中断 (OTG_HAINT) 寄存器，以获取主机通道 x 中断寄存器的通道编号。应用程序必须将该寄存器中的相应位清零，才能将 OTG_HAINT 和 OTG_GINTSTS 寄存器中的对应位清零。OTG_HCINTMSKx 寄存器还提供每个通道各中断源的屏蔽位。

- 主机模块提供以下状态检查和中断产生功能：
  - 传输完成中断，指示应用程序 (AHB) 和 USB 端均已完成数据传输
  - 通道因传输完成、USB 事务错误或应用程序发出禁止命令而停止
  - 相关的发送 FIFO 为半空或全空状态（IN 端点）
  - 接收到 ACK 响应
  - 接收到 NAK 响应
  - 接收到 STALL 响应
  - 由于 CRC 校验失败、超时、位填充错误和错误的 EOP 导致 USB 事务错误
  - 串扰错误
  - 帧上溢
  - 数据同步错误

50.7.4 主机调度器

主机模块内置硬件调度器，可自主对应用程序发出的 USB 事务请求重新排序和管理。每一帧开始时，主机都先执行周期性（同步和中断）事务，然后执行非周期性（控制和批量）事务，以符合 USB 规范对同步和中断传输高优先级的保证。

主机通过请求队列（一个周期性请求队列和一个非周期请求队列）处理 USB 事务。每个请求队列最多可存储 8 个条目。每个条目代表一个应用程序发起但还未得到响应的 USB 事务请求，并存储了执行该 USB 事务所用到的 IN 或 OUT 通道的编号，以及其他相关信息。USB 事务请求在队列中的写入顺序决定了事务在 USB 接口上的执行顺序。

每一帧开始时，主机都先处理周期性请求队列，然后处理非周期性请求队列。如果当前帧结束时，计划在当前帧执行的同步或中断类型的 USB 传输事务请求仍处于挂起状态，则主机将发出未完成周期性传输中断（OTG_GINTSTS 中的 IPXFR 位）。OTG_FS 模块全面负责对周期性和非周期性请求队列的管理。周期性发送 FIFO 和队列状态寄存器 (OTG_HPTXSTS) 与非周期性发送 FIFO 和队列状态寄存器 (OTG_HNPTXSTS) 都为只读寄存器，应用程序可使用它们来读取各请求队列的状态。其中包括：

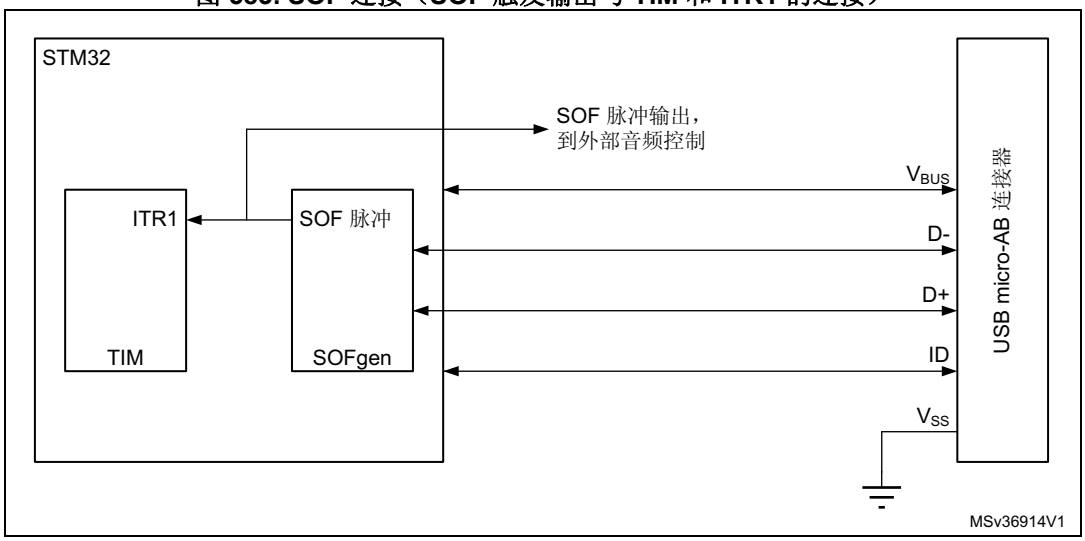
- 周期性（非周期性）请求队列中当前可用的空闲条目数（最多 8 个）
- 周期性（非周期性）Tx FIFO（OUT 事务）中当前可用的空闲空间
- IN/OUT 令牌、主机通道编号和其他状态信息

由于每个请求队列最多可存储 8 个 USB 事务请求，因此应用程序可以把主机 USB 事务请求提前发送给调度器；实际的通信最晚会在调度器处理完已挂起的 8 个周期事务和 8 个非周期事务完成之后出现在 USB 总线上。

要向主机调度器（队列）发出事务请求，应用程序必须读取 OTG_HNPTXSTS 寄存器中的 PTXQSAV 位或 OTG_HNPTXSTS 寄存器中的 NPTQXSAV 位，确保周期性（非周期性）请求队列中至少有一个可用空间来存储当前请求。

50.8 SOF 触发

图 583. SOF 连接（SOF 触发输出与 TIM 和 ITR1 的连接）



OTG_FS 模块在主机和设备模式下都可以监视、跟踪和配置 SOF 帧并且还具备 SOF 脉冲输出连接功能。

此功能尤其适用于自适应音频时钟生成，其中音频设备需要与 PC 提供的同步音频数据流实现同步，或者主机需要根据音频设备的要求调整数据帧率。

50.8.1 主机 SOF

主机模式下，可以在主机帧间隔寄存器 (HFIR) 中对所产生的两个连续 SOF (FS) 或 Keep-alive (LS) 令牌期间所出现的 PHY 时钟数进行编程，进而应用程序可对 SOF 帧周期进行控制。帧开始（OTH_GINTSTS 中的 SOF 位）时都将生成中断。当前帧编号和出现下一个 SOF 前剩余的时间应用程序在主机帧编号寄存器 (HFNUM) 中能够进行跟踪。

SOF 令牌发出的同时会产生 SOF 脉冲信号，并且宽度为 20 个 HCLK 周期。此外，SOF 脉冲还与定时器的输入触发相连，因此可通过 SOF 脉冲触发输入捕获功能、输出比较功能和定时器。

50.8.2 设备 SOF

在设备模式下，USB 每次接收到 SOF 令牌时，都将触发帧开始中断（OTH_GINTSTS 中的 SOF 位）。相应的帧编号可从设备状态寄存器（OTG_DSTS 中的 FNSOF 位）读取。还可以生成宽度为 20 个 HCLK 周期的 SOF 脉冲信号。此外，SOF 脉冲信号还在内部与 TIM 的输入触发相连，因此可通过 SOF 脉冲触发输入捕获功能、输出比较功能和定时器。

周期性帧结束中断 (OTG_GINTSTS/EOPF) 用于在经过了 80%、85%、90% 或 95% 的帧间隔时间时通知应用程序，具体取决于设备配置寄存器中的周期性帧间隔字段（OTG_DCFG 中的 PFIVL 位）。此功能可用于确定该帧的所有同步通信是否完成。

50.9 OTG 低功耗模式

下文的*表 319* 定义了 STM32 低功耗模式及其与 OTG 的兼容性。

表 319. STM32 低功耗模式与 OTG 的兼容性

模式	说明	USB 兼容性
运行	MCU 完全激活。	USB 不处于挂起状态时需要。
睡眠	USB 挂起退出可使器件退出睡眠模式。外设寄存器内容保持不变。	USB 处于挂起状态时可用。
停止	USB 挂起退出可使器件退出停止模式。外设寄存器内容保持不变(1)。	USB 处于挂起状态时可用。
待机	下电。在退出待机模式后，必须重新初始化外设。	与 USB 应用程序不兼容。

1. 在停止模式下，有不同的可能设置。也可能存在一些限制，请参见*第 5 节：电源控制 (PWR)* 以了解使用 OTG 的适用限制（如果存在）。

以下位和步骤降低了功耗。

OTG PHY 的功耗由通用模块配置寄存器中的两个或三个位控制，具体取决于 OTG 支持的版本。

- PHY 掉电 (OTG_GCCFG/PWRDWN)
  用于开启/关闭 PHY 的全速收发器模块。先置位后才允许后续的 USB 操作。
- V_{BUS} 检测使能 (OTG_GCCFG/VBDEN)
  用于开启/关闭与 OTG 操作关联的 V_{BUS} 感应比较器。

USB 会话没有开始或设备未连接时，可以在 USB 挂起状态下使用功率降低技术。

- 停止 PHY 时钟（OTG_PCGCCTL 中的 STPPCLK 位）
  将时钟门控控制寄存器中的停止 PHY 时钟位置 1 时，OTG 全速模块的大多数 48 MHz 内部时钟域均由时钟门控关闭。即使应用程序仍提供时钟输入，也会节省掉模块由于时钟信号翻转带来的动态功耗。
  还会关掉收发器的大部分单元，只有负责检测异步恢复事件或远程唤醒事件的部分还保持工作状态。
- HCLK 门控（OTG_PCGCCTL 中的 GATEHCLK 位）
  将时钟门控控制寄存器中的 Gate HCLK 位置 1 时，OTG_FS 模块内部的大多数系统时钟域均由时钟门控关闭。只有寄存器读取和写入接口保持活动状态。即使应用程序出于其他用途仍提供系统时钟，也会节省掉由于 USB 时钟信号翻转带来的动态功耗。
- USB 系统停止
  当 OTG_FS 处于 USB 挂起状态时，应用程序可通过将 USB 系统中的所有时钟源全部关闭来显著降低总功耗。USB 系统停止可通过以下方式激活：首先将停止 PHY 时钟位置 1，然后在电源控制系统模块 (PWR) 中将系统配置为深度睡眠模式。
  OTG_FS 模块通过对 USB 上的远程唤醒（作为主机）或恢复（作为设备）信号进行异步检测，自动重新激活系统时钟和 USB 时钟。

为了节省动态功耗，只在 USB 数据 FIFO 被 OTG_FS 模块访问时为其提供时钟。

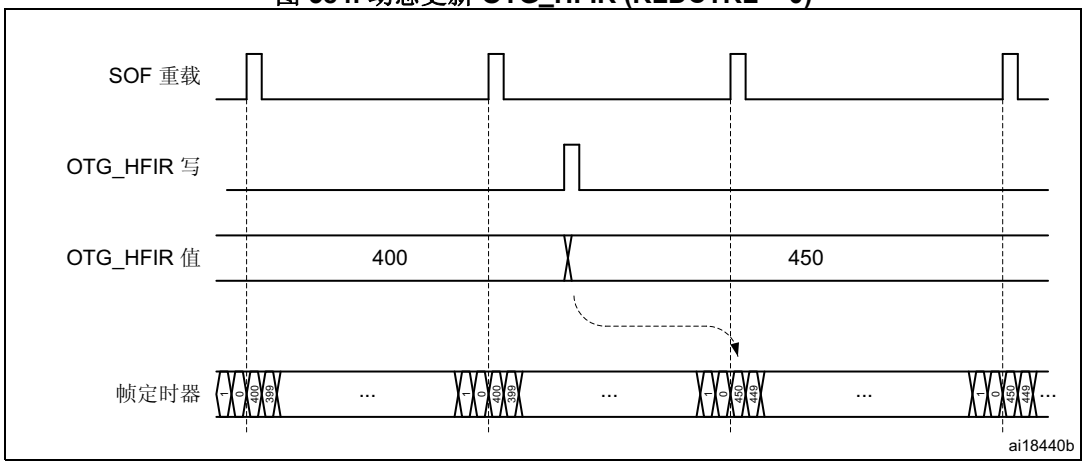
50.10 动态更新 OTG_HFIR 寄存器

主机模式下，USB 模块具有对 SOF 帧周期进行动态微调的功能，能够将外部设备与 SOF 帧进行同步。

如果 OTG_HFIR 寄存器在当前 SOF 帧内发生更改，则将在下一个帧中对 SOF 周期进行相应修正，具体说明请参见*图 584*。

对于动态更新，需要设置 RLDCTRL=0。

图 584. 动态更新 OTG_HFIR (RLDCTRL = 0)

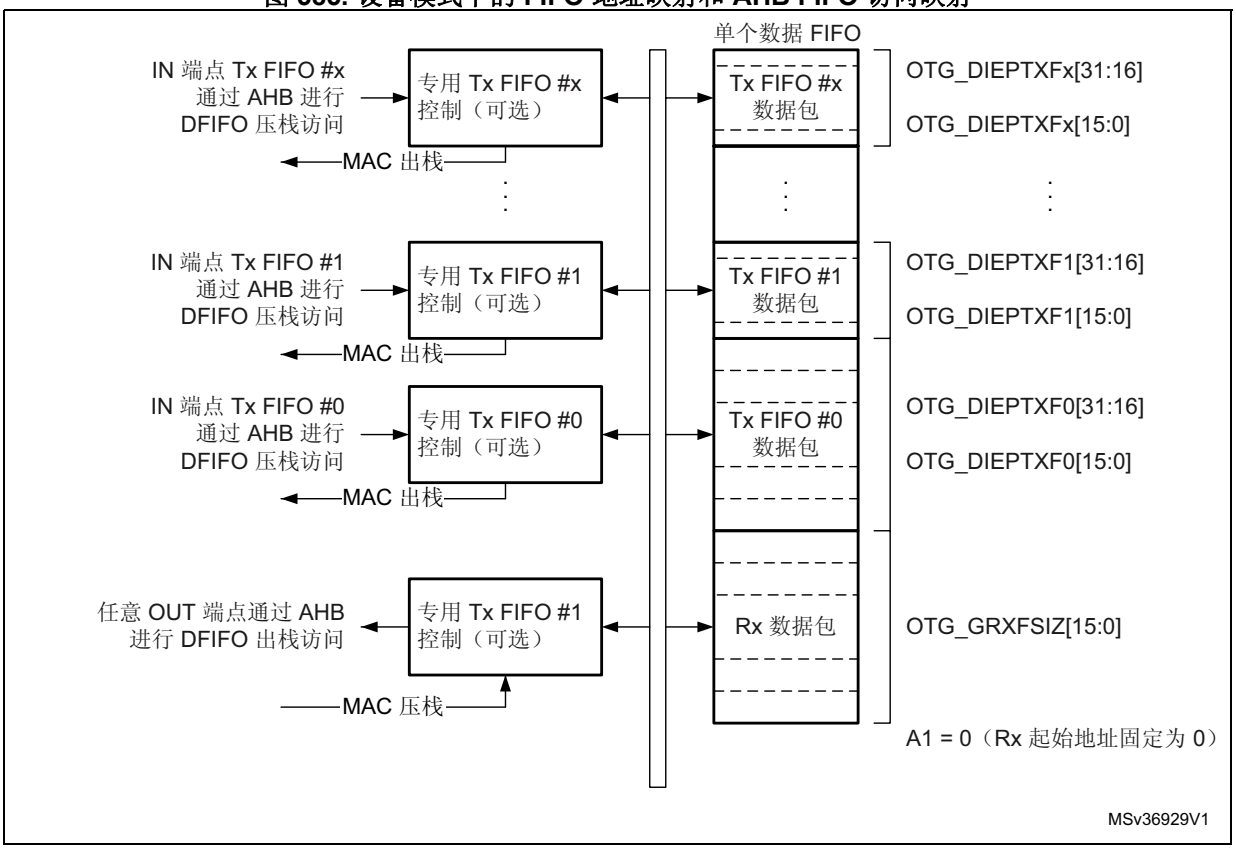


50.11 USB 数据 FIFO

USB 系统具有 1.25 KB 专用 RAM，采用复杂的 FIFO 控制机制。OTG_FS 模块中的数据包 FIFO 控制器模块将 RAM 空间划分为多个 Tx-FIFO（USB 传输前，应用程序将数据压入其中进行短暂存储）和单个 Rx FIFO（从 USB 接收到的数据被应用程序读取之前，在其中进行短暂存储）。RAM 中所构建的 FIFO 的数量与组织方式取决于设备的角色。设备模式下，为每个激活的 IN 端点配置一个 Tx FIFO。FIFO 的大小均由软件配置，以更好地满足应用要求。

50.11.1 设备 FIFO 架构

图 585. 设备模式下的 FIFO 地址映射和 AHB FIFO 访问映射



设备 Rx FIFO

OTG 设备使用单个接收 FIFO 接收发送到所有 OUT 端点的数据。只要 Rx FIFO 中有空余空间，收到的数据包就挨个填入 Rx FIFO。除了有效数据外，接收到的数据包状态（包含 OUT 端点目标编号、字节数、数据 PID 和对所接收数据的验证）也由模块进行存储。没有可用空间时，设备会回复主机事务 NACK 应答并在被寻址的端点上触发中断。接收 FIFO 的大小在接收 FIFO 大小寄存器 (OTG_GRXFSIZ) 中配置。

单个接收 FIFO 架构使得 USB 设备更高效地填充接收 RAM 缓冲区：

- 所有 OUT 端点共享同一个 RAM 缓冲区（共享 FIFO）
- OTG_FS 模块可将主机发出的任何 OUT 令牌序列填充到接收 FIFO，直到没有多余空闲空间

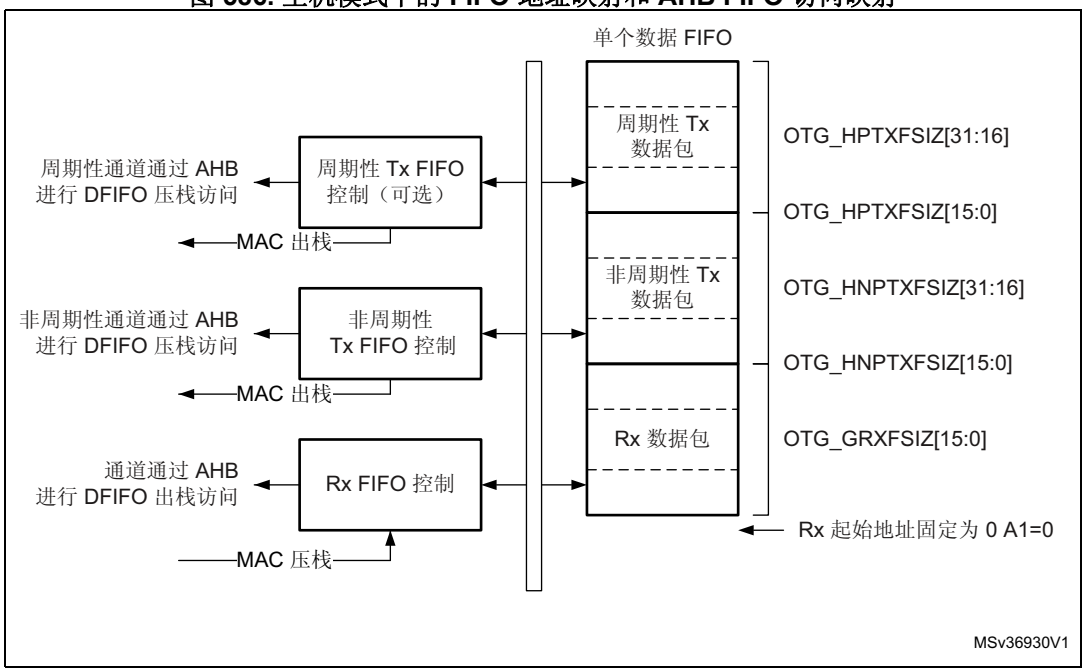
只要至少有一个数据包在 Rx FIFO 中可供读取，应用程序就会一直接收 Rx FIFO 非空中断（OTG_GINTSTS 中的 RXFLVL 位）。应用程序从接收状态读取和出栈寄存器 (OTG_GRXSTSP) 中读取数据包信息，最后通过读取与端点相关的出栈地址从接收 FIFO 读出相应数据。

设备 Tx FIFO

模块为各个 IN 端点提供了专用的 FIFO。应用程序通过端点 0 发送 FIFO 大小寄存器 (OTG_DIEPTXF0) 为 IN 端点 0 配置 FIFO 大小；通过设备 IN 端点发送 FIFOx 寄存器 (OTG_DIEPTXFx) 为 IN 端点 x 配置 FIFO 大小。

50.11.2 主机 FIFO 架构

图 586. 主机模式下的 FIFO 地址映射和 AHB FIFO 访问映射



主机 Rx FIFO

主机使用一个接收 FIFO 处理所有周期和非周期事务。FIFO 用作接收缓冲区以保存从 USB 接收到的数据（接收到的数据包的数据部分），直至这些数据传输到系统存储器。只要 FIFO 中有空间，来自设备 IN 端点的数据包就接收进来并挨个存储。接收到的每个数据包的状态（包含主机目标通道、字节数、数据 PID 和对所接收数据的校验）也存储在 FIFO 中。接收 FIFO 的大小在接收 FIFO 大小寄存器 (OTG_GRXFSIZ) 中配置。

单个接收 FIFO 架构使得 USB 主机高效地填充接收数据缓冲区：

- 所有 IN 配置主机通道共享同一个 RAM 缓冲区（共享 FIFO）
- OTG_FS 模块可将主机软件发出的任何 IN 令牌序列填充到接收 FIFO，直到没有多余空闲空间

只要至少有一个数据包在 Rx FIFO 中可供读取，应用程序就会接收 Rx FIFO 非空中断。应用程序从接收状态读取和出栈寄存器中读取数据包信息，最后从接收 FIFO 中读出数据。

主机 Tx FIFO

主机使用一个发送 FIFO 处理所有非周期（控制和批量）OUT 事务，使用另一个发送 FIFO 处理所有周期（同步和中断）OUT 事务。FIFO 用作发送缓冲区以保存要通过 USB 发送的数据（发送数据包）。周期（非周期）Tx FIFO 的大小在主机周期（非周期）发送 FIFO 大小 (OTG_HPTXFSIZ/OTG_HNPTXFSIZ) 寄存器中配置。

两个 Tx FIFO 按优先级实施操作，周期性通信的优先级较高，因此在 USB 一帧的时间内首先进行周期性通信。帧起始时，内置的主机调度器先处理周期请求队列，再处理非周期请求队列。

两个发送 FIFO 的架构使得 USB 主机能够对周期和非周期发送数据缓冲区分别进行优化管理：

- 配置为支持周期（非周期）OUT 事务的所有主机通道共享同一个 RAM 缓冲区（共享 FIFO）
- OTG_FS 模块可将主机软件发出的任何 OUT 令牌序列填充到周期性（非周期性）发送 FIFO，直到没有多余空闲空间

只要周期性 Tx FIFO 为半空或全空，OTG_FS 模块就会发出周期性 Tx FIFO 空中断（OTG_GINTSTS 中的 PTXFE 位），具体取决于 AHB 配置寄存器中的周期性 Tx FIFO 空门限位（OTG_GAHBCFG 中的 PTXFELVL 位）的值。只要周期性 Tx FIFO 和周期性请求队列中均存在空闲空间，应用程序便可提前写入发送数据。可通过读取主机周期性发送 FIFO 和队列状态寄存器 (OTG_HPTXSTS) 来了解二者的可用空间。

只要非周期性 Tx FIFO 为半空或全空，OTG_FS 模块就会发出非周期性 Tx FIFO 空中断（OTG_GINTSTS 中的 NPTXFE 位），具体取决于 AHB 配置寄存器中的非周期性 Tx FIFO 空门限位（OTG_GAHBCFG 中的 TXFELVL 位）。只要非周期性 Tx FIFO 和非周期性请求队列中均存在空闲空间，应用程序便可写入发送数据。可通过读取主机非周期性发送 FIFO 和队列状态寄存器 (OTG_HNPTXSTS) 来了解二者的可用空间。

50.11.3 FIFO RAM 分配

设备模式

接收 FIFO RAM 分配：应用程序应为 SETUP 数据包分配 RAM：

- 接收 FIFO 中必须保留 10 个位置以在控制端点上接收 SETUP 数据包。OTG 模块不会向这些为 SETUP 数据包保留的位置写入任何其他数据。
- 将会为全局 OUT NAK 分配一个位置。
- 状态信息随各个接收数据包写入 FIFO。因此，必须至少为接收数据包分配（最大数据包大小/4）+ 1 的空间。如果使能了多个同步端点，则为接收连续数据包分配的空间必须至少为（最大数据包大小/4）的两倍 + 1。通常，推荐的空间为（最大数据包/4 + 1）的两倍，这样当上一个数据包向 CPU 传送时，USB 可同时接收后续的数据包。
- 传输完成状态信息和该端点收到的最后一个数据包会一起被推入 FIFO。推荐为每个 OUT 端点分配一个位置存储该端点上的传输状态信息。

器件 RxFIFO =

(5 * 控制端点数量 + 8) + ((所使用的最大 USB 数据包/4) + 用于状态信息的 1) + (2 * OUT 端点数量) + 用于全局 NAK 的 1

例如：周期性 USB 数据包的 MPS 是 1024 个字节，非周期性 USB 数据包的 MPS 是 512 个字节。有三个 OUT 端点、三个 IN 端点、一个控制端点和三个主机通道。

则器件 RxFIFO = (5 * 1 + 8) + ((1,024/4) +1) + (2 * 4) + 1 = 279

发送 FIFO RAM 分配：各个 IN 端点发送 FIFO 所需的最小 RAM 空间为该特定 IN 端点的最大数据包大小。

注： *为发送 IN 端点 FIFO 分配的空间越多，USB 的性能就越高。*

主机模式

接收 FIFO RAM 分配：

状态信息随各个接收数据包写入 FIFO。因此，必须至少为接收数据包分配（最大数据包大小/4）+ 1 的空间。如果使能了多个同步通道，则为接收连续数据包分配的空间必须至少为（最大数据包大小/4）的两倍 + 1。通常，推荐的空间为（最大数据包/4 + 1）的两倍，这样当上一个数据包向 CPU 传送时，USB 可同时接收后续的数据包。

传输完成状态信息和主机通道中的最后一个数据包会一起被推入 FIFO。因此，必须为此分配一个位置。

主机 RxFIFO = ((所用的最大 USB 数据包/4) + 用于状态信息的 1) + 用于传输完成的 1

例如：主机 RxFIFO = ((1024/4）+ 1) + 1 = 258

发送 FIFO RAM 分配：

主机非周期性发送 FIFO 所需的最小 RAM 为所支持的所有非周期性 OUT 通道上传输的最大数据包的大小。

通常，推荐的空间为最大数据包大小的两倍，这样当 USB 正在发送当前数据包的同时，CPU 可获得下一个数据包。

非周期性 TxFIFO = 所用的最大非周期性 USB 数据包/4

例如：非周期性 TxFIFO = (512/4) = 128

主机周期性发送 FIFO 所需的最小 RAM 为所支持的所有周期性 OUT 通道上传输的最大数据包的大小。如果至少有一个同步 OUT 端点，则空间必须至少为该通道中最大数据包大小的两倍。

主机周期性 TxFIFO = 所用的最大周期性 USB 数据包/4

例如：主机周期性 TxFIFO = (1024/4) = 256

注： *为非周期性发送 FIFO 分配的空间越多，USB 的性能就越高。*

50.12 OTG_FS 系统性能

凭借大容量 RAM 缓冲区、高度可配置的 FIFO 大小、通过 AHB 压栈/出栈寄存器进行 32 位 FIFO 快速访问，尤其是高级 FIFO 控制机制可获得最佳 USB 和系统性能。实际上，无论当前 USB 序列如何，OTG_FS 均可通过该机制高效填充可用的 RAM 空间。借助这些特性：

- 应用程序有足够的裕量来计算并校正CPU的负载，从而优化 CPU 带宽利用率：
  - 应用程序可先积累大量发送数据，再通过 USB 发送出去
  - 可带来足够的时间裕量，以从接收 FIFO 读取数据
- USB 模块能够保持全速工作状态，也就是提供最大的全速带宽（尽量多的硬件自动运行，尽量少的软件参与）：
  - USB 模块可提前积累大量发送数据供其支配，从而可对 USB 数据发送进行自主管理
  - 接收缓冲区中有大量空白空间，可通过 USB 中的数据自动填满

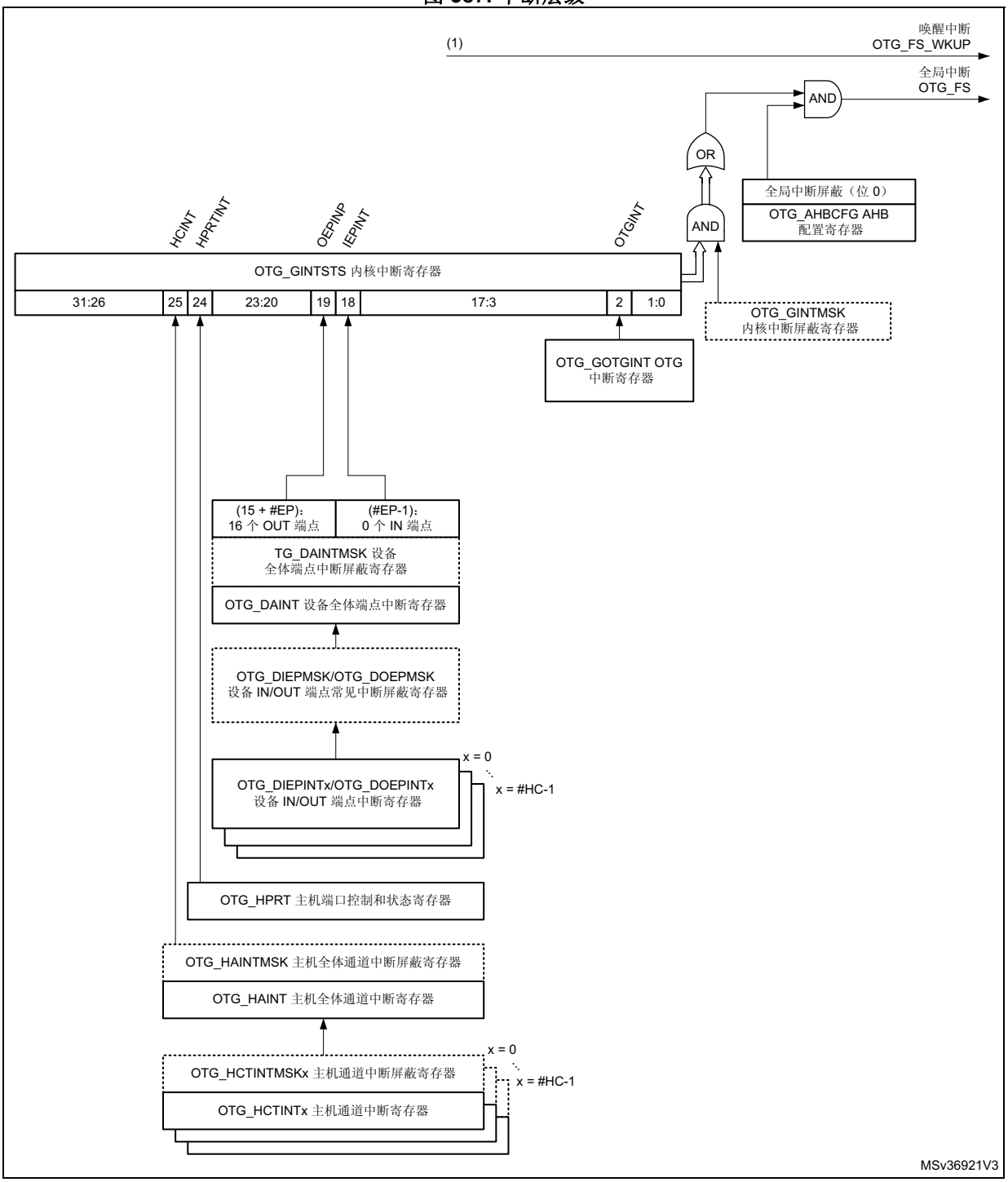
由于 OTG_FS 模块能够高效填充 1.25 KB RAM 缓冲区且 1.25 KB 发送/接收数据足以满足一个全速帧所能容纳的数据量，因此 USB 系统在一帧 (1 ms) 之内可以无需 CPU 干预达到最大 USB 带宽。

50.13 OTG_FS 中断

当 OTG_FS 控制器在一种模式下（设备模式或主机模式）工作时，应用程序不得以另一种角色模式访问寄存器。如果发生了非法访问，将会产生模式不匹配中断并在模块中断寄存器（OTG_GINTSTS 寄存器中的 MMIS 位）中反映。当模块从一种角色模式切换到另一种角色模式时，新工作模式下的寄存器必须重新编程为上电复位后的状态。

图 587 显示了中断层级。

图 587. 中断层级



1. 处于 L1 睡眠或 L2 挂起状态期间发生恢复条件时，OTG_FS_WKUP 变为有效状态（高电平状态）。

50.14 OTG_FS 控制和状态寄存器

应用程序通过 AHB 从接口对控制和状态寄存器 (CSR) 进行读写操作，以此来控制 OTG_FS 控制器。这些都是 32 位寄存器，其地址按 32 位对齐，因此只能以 32 位的方式访问 OTG_FS 寄存器。

CSR 分为以下几类：

- 模块全局寄存器
- 主机模式寄存器
- 主机全局寄存器
- 主机端口 CSR
- 主机通道相关寄存器
- 设备模式寄存器
- 设备全局寄存器
- 设备端点相关寄存器
- 电源和时钟门控寄存器
- 数据 FIFO (DFIFO) 访问寄存器

只有模块全局寄存器、电源和时钟门控寄存器、数据 FIFO 访问寄存器以及主机端口控制和状态寄存器，可在主机模式和设备模式下进行访问。当 OTG_FS 控制器在一种模式下（设备模式或主机模式）工作时，应用程序不得以另一种角色模式访问寄存器。如果发生了非法访问，将会产生模式不匹配中断并在模块中断寄存器（OTG_GINTSTS 寄存器中的 MMIS 位）中反映。当模块从一种角色模式切换到另一种角色模式时，新工作模式下的寄存器必须重新编程为上电复位后的状态。

50.14.1 CSR 存储器映射

主机模式寄存器和设备模式寄存器占据不同的地址。所有寄存器均在 AHB 时钟域内实现。

全局 CSR 地址映射

这些寄存器在主机模式和设备模式下均可用。

表 320. 模块全局控制和状态寄存器 (CSR)

缩略语	偏移地址	寄存器名称
OTG_GOTGCTL	0x000	*第 50.15.1 节：OTG 控制和状态寄存器 (OTG_GOTGCTL)*
OTG_GOTGINT	0x004	*第 50.15.2 节：OTG 中断寄存器 (OTG_GOTGINT)*
OTG_GAHBCFG	0x008	*第 50.15.3 节：OTG AHB 配置寄存器 (OTG_GAHBCFG)*
OTG_GUSBCFG	0x00C	*第 50.15.4 节：OTG USB 配置寄存器 (OTG_GUSBCFG)*
OTG_GRSTCTL	0x010	*第 50.15.5 节：OTG 复位寄存器 (OTG_GRSTCTL)*
OTG_GINTSTS	0x014	*第 50.15.6 节：OTG 模块中断寄存器 (OTG_GINTSTS)*
OTG_GINTMSK	0x018	*第 50.15.7 节：OTG 中断屏蔽寄存器 (OTG_GINTMSK)*

表 320. 模块全局控制和状态寄存器 (CSR)（续）

缩略语	偏移地址	寄存器名称
OTG_GRXSTSR	0x01C	*第 50.15.8 节：OTG 接收状态调试读取/OTG 状态读取和出栈寄存器 (OTG_GRXSTSR/OTG_GRXSTSP)*
OTG_GRXSTSP	0x020	
OTG_GRXFSIZ	0x024	*第 50.15.9 节：OTG 接收 FIFO 大小寄存器 (OTG_GRXFSIZ)*
OTG_HNPTXFSIZ/ OTG_DIEPTXF0(1)	0x028	*第 50.15.10 节：OTG 主机非周期性发送 FIFO 大小寄存器 (OTG_HNPTXFSIZ)/端点 0 发送 FIFO 大小 (OTG_DIEPTXF0)*
OTG_HNPTXSTS	0x02C	*第 50.15.11 节：OTG 非周期性发送 FIFO/队列状态寄存器 (OTG_HNPTXSTS)*
OTG_GCCFG	0x038	*第 50.15.12 节：OTG 通用模块配置寄存器 (OTG_GCCFG)*
OTG_CID	0x03C	*第 50.15.13 节：OTG 模块 ID 寄存器 (OTG_CID)*
OTG_GLPMCFG	0x54	*第 50.15.14 节：OTG 模块 LPM 配置寄存器 (OTG_GLPMCFG)*
OTG_GPWRDN	0x058	*第 50.15.15 节：OTG 掉电寄存器 (OTG_GPWRDN)*
OTG_GADPCTL	0x060	*第 50.15.16 节：OTG ADP 定时器、控制和状态寄存器 (OTG_GADPCTL)*
OTG_HPTXFSIZ	0x100	*第 50.15.17 节：OTG 主机周期性发送 FIFO 大小寄存器 (OTG_HPTXFSIZ)*
OTG_DIEPTXFx	0x104 0x108 ... 0x114	*第 50.15.18 节：OTG 设备 IN 端点发送 FIFO 大小寄存器 (OTG_DIEPTXFx)（x = 1..5，其中 x 为 FIFO 号）*

1. 通用规则是：主机模式下使用 OTG_HNPTXFSIZ；设备模式下使用 OTG_DIEPTXF0。

主机模式 CSR 地址映射

模块每次切换到主机模式时都必须对这些寄存器进行设置。

表 321. 主机模式控制和状态寄存器 (CSR)

缩略语	偏移地址	寄存器名称
OTG_HCFG	0x400	*第 50.15.20 节：OTG 主机配置寄存器 (OTG_HCFG)*
OTG_HFIR	0x404	*第 50.15.21 节：OTG 主机帧间隔寄存器 (OTG_HFIR)*
OTG_HFNUM	0x408	*第 50.15.22 节：OTG 主机帧编号/帧剩余时间寄存器 (OTG_HFNUM)*
OTG_HPTXSTS	0x410	*第 50.15.23 节：OTG_Host 周期性发送 FIFO/队列状态寄存器 (OTG_HPTXSTS)*
OTG_HAINT	0x414	*第 50.15.24 节：OTG 主机全体通道中断寄存器 (OTG_HAINT)*
OTG_HAINTMSK	0x418	*第 50.15.25 节：OTG 主机全体通道中断屏蔽寄存器 (OTG_HAINTMSK)*

表 321. 主机模式控制和状态寄存器 (CSR)（续）

缩略语	偏移地址	寄存器名称
OTG_HPRT	0x440	*第 50.15.26 节：OTG 主机端口控制和状态寄存器 (OTG_HPRT)*
OTG_HCCHARx	0x500 0x520 ... 0x660	*第 50.15.27 节：OTG 主机通道 x 特性寄存器 (OTG_HCCHARx)（x = 0..11，其中 x 表示通道编号）*
OTG_HCINTx	0x508 0x528 .... 0x668	*第 50.15.28 节：OTG 主机通道 x 中断寄存器 (OTG_HCINTx)（x = 0..11，其中 x 表示通道编号）*
OTG_HCINTMSKx	0x50C 0x52C 0x66C	*第 50.15.29 节：OTG 主机通道 x 中断屏蔽寄存器 (OTG_HCINTMSKx)（x = 0..11，其中 x 表示通道编号）*
OTG_HCTSIZx	0x510 0x530 .... 0x670	*第 50.15.30 节：OTG 主机通道 x 传输大小寄存器 (OTG_HCTSIZx)（x = 0..11，其中 x 表示通道编号）*

设备模式 CSR 地址映射

模块每次切换到设备模式时都必须对这些寄存器进行编程。

表 322. 设备模式控制和状态寄存器

缩略语	偏移地址	寄存器名称
OTG_DCFG	0x800	*第 50.15.32 节：OTG 设备配置寄存器 (OTG_DCFG)*
OTG_DCTL	0x804	*第 50.15.33 节：OTG 设备控制寄存器 (OTG_DCTL)*
OTG_DSTS	0x808	*第 50.15.34 节：OTG 设备状态寄存器 (OTG_DSTS)*
OTG_DIEPMSK	0x810	*第 50.15.35 节：OTG 设备 IN 端点通用中断屏蔽寄存器 (OTG_DIEPMSK)*
OTG_DOEPMSK	0x814	*第 50.15.36 节：OTG 设备 OUT 端点通用中断屏蔽寄存器 (OTG_DOEPMSK)*
OTG_DAINT	0x818	*第 50.15.37 节：OTG 设备全体端点中断寄存器 (OTG_DAINT)*
OTG_DAINTMSK	0x81C	*第 50.15.38 节：OTG 全体端点中断屏蔽寄存器 (OTG_DAINTMSK)*
OTG_DVBUSDIS	0x828	*第 50.15.39 节：OTG 设备 VBUS 放电时间寄存器 (OTG_DVBUSDIS)*
OTG_DVBUSPULSE	0x82C	*第 50.15.40 节：OTG 设备 VBUS 脉冲时间寄存器 (OTG_DVBUSPULSE)*

表 322. 设备模式控制和状态寄存器（续）

缩略语	偏移地址	寄存器名称
OTG_DIEPEMPMSK	0x834	*第 50.15.41 节：OTG 设备 IN 端点 FIFO 空中断屏蔽寄存器 (OTG_DIEPEMPMSK)*
OTG_DIEPCTL0	0x900	*第 50.15.42 节：OTG 设备控制 IN 端点 0 控制寄存器 (OTG_DIEPCTL0)*
OTG_DIEPCTLx	0x920 0x940 ... 0x9A0	*第 50.15.43 节：OTG 设备 IN 端点 x 控制寄存器 (OTG_DIEPCTLx)（x = 1..5，其中 x 表示端点编号）*
OTG_DIEPINTx	0x908 0x928 .... 0x988	*第 50.15.44 节：OTG 设备 IN 端点 x 中断寄存器 (OTG_DIEPINTx)（x = 0..5，其中 x 表示端点编号）*
OTG_DIEPTSIZ0	0x910	*第 50.15.45 节：OTG 设备 IN 端点 0 传输大小寄存器 (OTG_DIEPTSIZ0)*
OTG_DTXFSTSx	0x918 0x938 .... 0x998	*第 50.15.46 节：OTG 设备 IN 端点发送 FIFO 状态寄存器 (OTG_DTXFSTSx)（x = 0..5，其中 x = 端点编号）*
OTG_DIEPTSIZx	0x930 0x950 ... 0x9B0	*第 50.15.47 节：OTG 设备 IN 端点 x 传输大小寄存器 (OTG_DIEPTSIZx)（x = 1..5，其中 x 表示端点编号）*
OTG_DOEPCTL0	0xB00	*第 50.15.48 节：OTG 设备控制 OUT 端点 0 控制寄存器 (OTG_DOEPCTL0)*
OTG_DOEPINTx	0xB08 0xB28 ... 0xBA8	*第 50.15.49 节：OTG 设备 OUT 端点 x 中断寄存器 (OTG_DOEPINTx)（x = 0..5，其中 x 表示端点编号）*
OTG_DOEPTSIZ0	0xB10	*第 50.15.50 节：OTG 设备 OUT 端点 0 传输大小寄存器 (OTG_DOEPTSIZ0)*
OTG_DOEPCTLx	0xB20 0xB40 ... 0xBA0	*第 50.15.51 节：OTG 设备 OUT 端点 x 控制寄存器 (OTG_DOEPCTLx)（x = 1..5，其中 x 表示端点编号）*
OTG_DOEPTSIZx	0xB30 0xB50 ... 0xBB0	*第 50.15.52 节：OTG 设备 OUT 端点 x 传输大小寄存器 (OTG_DOEPTSIZx)（x = 1..5，其中 x 表示端点编号）*

数据 FIFO (DFIFO) 访问寄存器地址映射

这些寄存器在主机模式和设备模式下均可用，用于对给定方向的特定端点或通道的 FIFO 空间进行读写操作。如果主机通道类型为 IN，则只能对该通道上的 FIFO 进行读操作。同样地，如果主机通道类型为 OUT，则只能对该通道上的 FIFO 进行写操作。

表 323. 数据 FIFO (DFIFO) 访问寄存器地址映射

FIFO 访问寄存器段	偏移地址	访问
设备 IN 端点 0/主机 OUT 通道 0：DFIFO 写访问 设备 OUT 端点 0/主机 IN 通道 0：DFIFO 读访问	0x1000–0x1FFC	w r
设备 IN 端点 1/主机 OUT 通道 1：DFIFO 写访问 设备 OUT 端点 1/主机 IN 通道 1：DFIFO 读访问	0x2000–0x2FFC	w r
...	...	...
设备 IN 端点 x(1)/主机 OUT 通道 x(1)：DFIFO 写访问 设备 OUT 端点 x(1)/主机 IN 通道 x(1)：DFIFO 读访问	0xX000–0xXFFC	w r

1. 其中，x 为 5（设备模式）或 11（主机模式）。

电源和时钟门控 CSR 地址映射

由一个单独寄存器对电源和时钟门控进行管理。在主机模式和设备模式下均可使用。

表 324. 电源和时钟门控控制和状态寄存器

缩略语	偏移地址	寄存器名称
OTG_PCGCCTL	0xE00–0xE04	*第 50.15.53 节：OTG 电源和时钟门控控制寄存器 (OTG_PCGCCTL)*

50.15 OTG_FS 寄存器

这些寄存器在主机模式和设备模式下都可用，且在这两个模式间切换时无需对其进行重新编程。

除非特别说明，否则寄存器描述中的位值以二进制表示。

50.15.1 OTG 控制和状态寄存器 (OTG_GOTGCTL)

OTG control and status register

偏移地址：0x000

复位值：0x0001 0000

OTG_GOTGCTL 寄存器控制模块的 OTG 功能并反映其状态。

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.	CUR MOD	OTG VER	BSVLD	ASVLD	DBCT	CID STS
										r	rw	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	EHEN	DHNP EN	HSHNP EN	HNP RQ	HNG SCS	BVALO VAL	BVALO EN	AVALO VAL	AVALO EN	VBVAL OVAL	VBVAL OEN	SRQ	SRQ SCS
			rw	rw	rw	rw	r	rw	rw	rw	rw	rw	rw	rw	r

位 31:22 保留，必须保持复位值。

位 21 **CURMOD**：当前工作模式 (Current mode of operation)

该位指示当前模式（主机或设备）。

0：设备模式

1：主机模式

位 20 **OTGVER**：OTG 版本 (OTG version)

选择 OTG 版本。

0：OTG 版本 1.3。OTG1.3 在新产品开发中遭到淘汰。

1：OTG 版本 2.0。在此版本中，模块仅支持通过数据线脉冲实现 SRP。

位 19 **BSVLD**：B 会话有效 (B-session valid)

指示设备模式下收发器的状态。

0：B 会话无效。

1：B 会话有效。

在 OTG 模式下，该位可用于确定设备处于连接状态还是断开状态。

注： 仅可在设备模式下访问。

位 18 **ASVLD**：A 会话有效 (A-session valid)

指示主机模式下收发器的状态。

0：A 会话无效

1：A 会话有效

注： 仅可在主机模式下访问。

位 17 **DBCT**：长/短去抖动时间 (Long/short debounce time)

指示已检测到连接的去抖动时间。

0：长去抖动时间，用于物理连接 (100 ms + 2.5 µs)

1：短去抖动时间，用于软连接 (2.5 µs)

注： 仅可在主机模式下访问。

位 16 **CIDSTS**：连接器 ID 状态 (Connector ID status)

指示发生连接事件时的连接器 ID 状态。

0：OTG_FS 控制器处于 A 设备模式

1：OTG_FS 控制器处于 B 设备模式

注： 在设备模式和主机模式均可访问。

位 15:13 保留，必须保持复位值。

位 12 **EHEN**：嵌入式主机使能 (Embedded host enable)

用于在 OTG A 设备状态机和嵌入式主机状态机之间选择。

0：选择 OTG A 设备状态机

1：选择嵌入式主机状态机

位 11 **DHNPEN**：使能设备 HNP 特性 (Device HNP enabled)

从所连 USB 主机处成功接收到 SetFeature.SetHNPEnable 命令时，应用程序将该位置 1。

0：应用不使能 HNP

1：应用使能 HNP

注： 仅可在设备模式下访问。

位 10 **HSHNPEN**：主机设置 HNP 使能 (Host set HNP enable)

（使用 SetFeature.SetHNPEnable 命令）在所连接设备上成功使能 HNP 后，应用程序将该位置 1。

0：主机未对设备设置了 HNP 使能

1：主机已对设备设置了 HNP 使能

注： 仅可在主机模式下访问。

位 9 **HNPRQ**：HNP 请求 (HNP request)

应用程序将该位置 1 时，将对所连接 USB 主机发起 HNP 请求。当 OTG_GOTGINT 寄存器中的主机协商成功状态更改位（OTG_GOTGINT 中的 HNSSCHG 位）置 1 时，应用程序可通过写 0 将该位清零。HNSSCHG 位清零时，模块会将该位清零。

0：无 HNP 请求

1：HNP 请求

注： 仅可在设备模式下访问。

位 8 **HNGSCS**：主机协商成功 (Host negotiation success)

当主机协商成功时，模块会将该位置 1。当该寄存器中的 HNP 请求位 (HNPRQ) 置 1 时，模块会将该位清零。

0：主机协商失败

1：主机协商成功

注： 仅可在设备模式下访问。

位 7 **BVALOVAL**：B 设备会话有效覆盖值 (B-peripheral session valid override value)

此位用于在 BVALOEN 位置 1 时设置 Bvalid 信号的覆盖值。

0：BVALOEN = 1 时，Bvalid 值为“0”

1：BVALOEN = 1 时，Bvalid 值为“1”

注： 仅可在设备模式下访问。

位 6 **BVALOEN**：B 设备会话有效覆盖使能 (B-peripheral session valid override enable)

该位用于使能/禁止软件通过 BVALOVAL 位来覆盖 Bvalid 信号。

0：禁止覆盖，模块使用来自所选 PHY 的 Bvalid 信号

1：从 PHY 内部接收的 Bvalid 由 BVALOVAL 位的值覆盖

注： 仅可在设备模式下访问。

位 5 **AVALOVAL**：A 设备会话有效覆盖值 (A-peripheral session valid override value)

此位用于在 AVALOEN 位置 1 时设置 Avalid 信号的覆盖值。

0：AVALOEN = 1 时，Avalid 值为“0”

1：AVALOEN = 1 时，Avalid 值为“1”

注： 仅可在主机模式下访问。

位 4 **AVALOEN**：A 设备会话有效覆盖使能 (A-peripheral session valid override enable)

该位用于使能/禁止软件通过 AVALOVAL 位来覆盖 Avalid 信号。

0：禁止覆盖，模块使用来自所选 PHY 的 Avalid 信号

1：从 PHY 内部接收的 Avalid 由 AVALOVAL 位的值覆盖

注： 仅可在主机模式下访问。

位 3 **VBVALOVAL**：V_{BUS} 有效覆盖值 (V_{BUS} valid override value)

此位用于在 VBVALOEN 位置 1 时设置 vbusvalid 信号的覆盖值。

0：VBVALOEN = 1 时，vbusvalid 值为“0”

1：VBVALOEN = 1 时，vbusvalid 值为“1”

注： 仅可在主机模式下访问。

位 2 **VBVALOEN**：V_{BUS} 有效覆盖使能 (V_{BUS} valid override enable)

该位用于使能/禁止软件通过 VBVALOVAL 位来覆盖 vbusvalid 信号。

0：禁止覆盖，模块使用来自所选 PHY 的 vbusvalid 信号

1：从 PHY 内部接收的 vbusvalid 由 VBVALOVAL 位的值覆盖

注： 仅可在主机模式下访问。

位 1 **SRQ**：会话请求 (Session request)

应用程序将该位置 1 时，将在 USB 上发起会话请求。当 OTG_GOTGINT 寄存器中的主机协商成功状态更改位（OTG_GOTGINT 中的 HNSSCHG 位）置 1 时，应用程序可通过写 0 将该位清零。HNSSCHG 位清零时，模块会将该位清零。

如果使用 USB 1.1 全速串行收发器接口来启动会话请求，则应用程序必须在该寄存器中的 B 会话有效位（OTG_GOTGCTL 中的 BSVLD 位）清零后，等待 V_{BUS} 放电至 0.2 V。

0：无会话请求

1：会话请求

注： 仅可在设备模式下访问。

位 0 **SRQSCS**：会话请求成功 (Session request success)

当会话请求成功时，模块会将该位置 1。

0：会话请求失败

1：会话请求成功

注： 仅可在设备模式下访问。

50.15.2 OTG 中断寄存器 (OTG_GOTGINT)

OTG interrupt register

偏移地址：0x04

复位值：0x0000 0000

应用程序会在出现 OTG 中断时读取该寄存器，并通过将该寄存器中的位清零来清除 OTG 中断。

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.	IDCHNG	DBCDNE	ADTOCHG	HNGDET	Res.
											rc_w1	rc_w1	rc_w1	rc_w1	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	HNSSCHG	SRSSCHG	Res.	Res.	Res.	Res.	Res.	SEDET	Res.	Res.
						rc_w1	rc_w1						rc_w1		

位 31:21 保留，必须保持复位值。

位 20 **IDCHNG**：
该位置 1 时，表示 ID 输入引脚的值发生了变化。

位 19 **DBCDNE**：去抖动完成 (Debounce done)
模块会在设备连接后且去抖动结束时将该位置 1。应用程序会在看见该中断后，开始驱动 USB 复位。该位仅在 OTG_GUSBCFG 寄存器中的 HNP 功能位或 SRP 功能位（分别为 OTG_GUSBCFG 中的 HNPCAP 位或 SRPCAP 位）置 1 时有效。
注： 仅可在主机模式下访问。

位 18 **ADTOCHG**：A 设备超时更改 (A-device timeout change)
模块将该位置 1 时，指示 A 设备在等待 B 设备连接时超时。
注： 在设备模式和主机模式均可访问。

位 17 **HNGDET**：检测到主机协商 (Host negotiation detected)
当检测到 USB 上的主机协商请求时，模块会将该位置 1。
注： 在设备模式和主机模式均可访问。

位 16:10 保留，必须保持复位值。

位 9 **HNSSCHG**：主机协商成功状态更改 (Host negotiation success status change)
模块将在 USB 主机协商请求成功或失败时将此位置 1。应用程序必须读取 OTG_GOTGCTL 寄存器中的主机协商成功位（OTG_GOTGCTL 中的 HNGSCS 位）来检查请求成功还是失败。
注： 在设备模式和主机模式均可访问。

位 8 **SRSSCHG**：会话请求成功状态更改 (Session request success status change)
模块将在会话请求成功或失败时将此位置 1。应用程序必须读取 OTG_GOTGCTL 寄存器中的会话请求成功位（OTG_GOTGCTL 中的 SRQSCS 位）来检查请求成功还是失败。
注： 在设备模式和主机模式均可访问。

位 7:3 保留，必须保持复位值。

位 2 **SEDET**：检测到会话结束 (Session end detected)
模块将该位置 1 时，指示 V_{BUS} < 0.8 V 时，V_{BUS} 上的电压不再适用于 B 设备会话。
注： 在设备模式和主机模式均可访问。

位 1:0 保留，必须保持复位值。

50.15.3 OTG AHB 配置寄存器 (OTG_GAHBCFG)

OTG AHB configuration register

偏移地址：0x008

复位值：0x0000 0000

该寄存器可用于在上电后或更改角色模式时对模块进行配置。该寄存器主要包含 AHB 系统相关的配置参数。请勿在初始编程后更改该寄存器。应用程序必须在开始任何 AHB 或 USB 事务前对该寄存器进行编程。

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.	Res.	Res.	Res.	Res.	Res.	PTXFELVL	TXFELVL	Res.	Res.	Res.	Res.	Res.	Res.	GINTMSK
							rw	rw							rw

位 31:9 保留，必须保持复位值。

位 8 **PTXFELVL**：周期性 Tx FIFO 空门限 (Periodic Tx FIFO empty level)

指示 OTG_GINTSTS 寄存器中的周期性 Tx FIFO 空中断位（OTG_GINTSTS 中的 PTXFE 位）何时触发。

0：PTXFE（位于 OTG_GINTSTS）中断表示周期性 Tx FIFO 为半空状态

1：PTXFE（位于 OTG_GINTSTS）中断表示周期性 Tx FIFO 为全空状态

注： 仅可在主机模式下访问。

位 7 **TXFELVL**：Tx FIFO 空门限 (Tx FIFO empty level)

在设备模式下，该位指示 IN 端点发送 FIFO 空中断（OTG_DIEPINTx 中的 TXFE）何时触发：

0：TXFE（位于 OTG_DIEPINTx）中断表示 IN 端点 Tx FIFO 为半空状态

1：TXFE（位于 OTG_DIEPINTx）中断表示 IN 端点 Tx FIFO 为全空状态

在主机模式下，该位指示非周期性 Tx FIFO 空中断（OTG_GINTSTS 中的 NPTXFE 位）何时触发。

0：NPTXFE（位于 OTG_GINTSTS）中断表示非周期性 Tx FIFO 为半空状态

1：NPTXFE（位于 OTG_GINTSTS）中断表示非周期性 Tx FIFO 为全空状态

位 6:1 保留，必须保持复位值。

位 0 **GINTMSK**：全局中断屏蔽 (Global interrupt mask)

该位用于屏蔽全局中断或使能全局中断。中断状态寄存器由模块进行更新，与此位的设置无关。

0：对应用程序屏蔽中断。

1：不对应用程序屏蔽中断。

注： 在设备模式和主机模式均可访问。

50.15.4 OTG USB 配置寄存器 (OTG_GUSBCFG)

OTG USB configuration register

偏移地址：0x00C

复位值：0x0000 1440

该寄存器可用于在上电或更改角色模式后对模块进行配置。其中包含与 USB 和 USB-PHY 相关的配置参数。应用程序必须在开始任何 AHB 或 USB 事务前对该寄存器进行编程。请勿在初始编程后更改该寄存器。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	FD MOD	FH MOD	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	rw	rw													
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	TRDT				HNP CAP	SRP CAP	Res.	PHY SEL	Res.	Res.	Res.	TOCAL		
		rw				rw	rw		r				rw		

位 31 保留，必须保持复位值。

位 30 **FDMOD**：强制设备模式 (Force device mode)

向该位写入 1 时，可将模块强制为设备模式，而无需考虑 OTG_ID 输入引脚。

0：正常模式

1：强制设备模式

将强制位置 1 后，应用程序必须等待至少 25 ms 后更改方可生效。

注： 在设备模式和主机模式均可访问。

位 29 **FHMOD**：强制主机模式 (Force host mode)

向该位写入 1 时，可将模块强制为主机模式，而无需考虑 OTG_ID 输入引脚。

0：正常模式

1：强制主机模式

将强制位置 1 后，应用程序必须等待至少 25 ms 后更改方可生效。

注： 在设备模式和主机模式均可访问。

位 28:26 保留，必须保持复位值。

位 25:23 保留，必须保持复位值。

位 22 保留，必须保持复位值。

位 21:16 保留，必须保持复位值。

位 15 保留，必须保持复位值。

位 14 保留，必须保持复位值。

位 13:10 **TRDT[3:0]**：USB 周转时间 (USB turnaround time)

这些位允许以 PHY 时钟数为单位设置周转时间。必须根据 *表 325：TRDT 值 (FS)* 来配置这些位，具体取决于应用程序 AHB 频率。TRDT 值越高，USB 对 IN 令牌的响应时间就越长，从而可以弥补 AHB 对数据 FIFO 的较长读访问延迟。

注： 仅可在设备模式下访问。

位 9 **HNPCAP**：HNP 功能 (HNP-capable)

应用程序使用该位控制 OTG_FS 控制器的 HNP 功能。

0：不使能 HNP 功能。

1：使能 HNP 功能。

注： 在设备模式和主机模式均可访问。

位 8 **SRPCAP**：SRP 功能 (SRP-capable)

应用程序使用该位控制 OTG_FS 控制器的 SRP 功能。如果模块作为无 SRP 功能的 B 设备，则无法请求连接的 A 设备（主机）激活 V_{BUS} 并开始会话。

0：不使能 SRP 功能。

1：使能 SRP 功能。

注： 在设备模式和主机模式均可访问。

位 7 保留，必须保持复位值。

位 6 **PHYSEL**：全速串行收发器选择 (Full Speed serial transceiver select)

此位为只读位且始终为 1。

位 5 保留，必须保持复位值。

位 4 保留，必须保持复位值。

位 3 保留，必须保持复位值。

位 2:0 **TOCAL[2:0]**：FS 超时校准 (FS timeout calibration)

PHY 引入的额外延迟包括应用程序在该字段中设置的 PHY 时钟数，以及模块的全速数据包间超时间隔。不同 PHY 引入的延迟对数据线状态的影响是不同的。

全速操作的 USB 标准超时值为 16 到 18（含）个位时间。应用程序必须根据枚举速度编程该字段。每个 PHY 时钟增加的位时间数为 0.25 个位时间。

表 325. TRDT 值 (FS)

AHB 频率范围 (MHz)		TRDT 最小值
最小值	最大值	
14.2	15	0xF
15	16	0xE
16	17.2	0xD
17.2	18.5	0xC
18.5	20	0xB
20	21.8	0xA
21.8	24	0x9
24	27.5	0x8
27.5	32	0x7
32	-	0x6

50.15.5 OTG 复位寄存器 (OTG_GRSTCTL)

OTG reset register

偏移地址：0x10

复位值为：0x8000 0000

应用程序通过此寄存器复位模块中的各项硬件特性。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
AHB IDL	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
r															r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	TXFNUM					TXF FLSH	RXF FLSH	Res.	FCRST	PSRST	CSRST
					rw					rs	rs		rs	rs	r

位 31 **AHBIDL**：AHB 主设备空闲 (AHB master idle)

指示 AHB 主设备状态机处于空闲情况。

注：　在设备模式和主机模式均可访问。

位 30:11 保留，必须保持复位值。

位 10:6 **TXFNUM[4:0]**：Tx FIFO 编号 (Tx FIFO number)

使用 Tx FIFO 刷新位进行刷新的 FIFO 编号。只有在模块将 Tx FIFO 刷新位清零后，方可更改此字段。

00000：

– 主机模式下刷新非周期性 Tx FIFO

– 设备模式下刷新 Tx FIFO 0

00001：

– 主机模式下刷新周期性 Tx FIFO

– 设备模式下刷新 Tx FIFO 1

00010：设备模式下刷新 Tx FIFO 2

...

01111：设备模式下刷新 Tx FIFO 15

10000：在设备模式或主机模式下刷新所有的发送 FIFO

注：　在设备模式和主机模式均可访问。

位 5 **TXFFLSH**：Tx FIFO 刷新 (Tx FIFO flush)

此位选择性地刷新一个或所有的发送 FIFO，但当模块处理通信事务时无法执行该操作。

只有在确认模块当前未对 Tx FIFO 执行读写操作后，应用程序方可对此位执行写操作。使用以下寄存器进行确认：

读——NAK 有效中断可确保模块当前未对 FIFO 执行读操作。

写——OTG_GRSTCTL 中的 AHBIDL 位可确保模块当前未对 FIFO 执行任何写操作。

通常建议在重新配置 FIFO 时进行刷新。还建议在设备端点禁止期间进行 FIFO 刷新。应用程序必须等待模块将此位清零，才能执行任意操作。该位需要八个时钟来清零（使用较慢的 phy_clk 或 hclk 时钟）。

注：　在设备模式和主机模式均可访问。

位 4 **RXFFLSH**：Rx FIFO 刷新 (Rx FIFO flush)

应用程序可使用此位刷新整个 Rx FIFO，但必须首先确保模块当前未在处理事务。
只有在确认模块当前未对 Rx FIFO 执行读写操作后，应用程序方可对此位执行写操作。
应用程序必须等到此位清零后，方可执行其他操作。通常需要等待 8 个时钟周期（以 PHY 或 AHB 时钟中最慢的为准）。

注： 在设备模式和主机模式均可访问。

位 3 保留，必须保持复位值。

位 2 **FCRST**：主机帧计数器复位 (Host frame counter reset)

应用程序对该位执行写操作时，模块中的帧数计数器复位。帧计数器复位后，由模块发送的下一个 SOF 的帧号为 0。
当应用程序向此位写入“1”后，可能无法回读该值，原因是模块将在几个时钟周期内将此位清零。

注： 仅可在主机模式下访问。

位 1 **PSRST**：部分软复位 (Partial soft reset)

复位内部状态机，但保留枚举信息。可用于恢复一些特定的 PHY 错误。

注： 在设备模式和主机模式均可访问。

位 0 **CSRST**：模块软复位 (Core soft reset)

按如下所述将 HCLK 和 PHY 时钟域复位：
除以下各位外，将各个中断和所有 CSR 寄存器位清零：
– OTG_PCGCCTL 中的 GAYEHCLK 位
– OTG_PCGCCTL 中的 STPPCLK 位
– OTG_HCFG 中的 FSLSPCS 位
– OTG_DCFG 中的 DSPD 位
– OTG_DCTL 中的 SDIS 位
– OTG_GCCFG 寄存器
– OTG_GPWRDN 寄存器
– OTG_GADPCTL 寄存器
将所有模块状态机（AHB 从设备除外）复位至空闲状态，并清空所有发送 FIFO 和接收 FIFO。
在 AHB 传输的最后数据阶段结束后，尽快终止 AHB 主设备上的所有事务。立即终止 USB 上的所有事务。
当使能 ADP 功能时，电源管理模块不会被模块软复位复位。
应用程序可在需要复位模块时随时对该位执行写操作。该位为自清零位，模块将在其中所有必要逻辑复位后将该位清零，该过程需要若干个时钟的时间，具体取决于模块的当前状态。该位一旦清零，软件必须等待至少 3 个 PHY 时钟后才可以访问 PHY 域（同步延迟）。此外，软件还必须在确定该寄存器中的位 31 置 1（AHB 主设备空闲）后方可开始运行。
软件复位通常在两种情况下使用，一是软件开发期间，二是用户动态更改以上所列 USB 配置寄存器中的 PHY 选择位后。用户更改 PHY 时，将为 PHY 选择相应的时钟并用于 PHY 域中。一旦选择了新的时钟，则必须复位 PHY 域，才能保证正常运行。

注： 在设备模式和主机模式均可访问。

50.15.6 OTG 模块中断寄存器 (OTG_GINTSTS)

OTG core interrupt register

偏移地址：0x014

复位值：0x1400 0020

该寄存器用于在当前模式（设备模式或主机模式）下借助系统级别的事件来中断应用程序。

该寄存器中的某些位仅在主机模式下有效，而其他位则仅在设备模式下有效。此外，该寄存器还可指示当前模式。要将 rc_w1 类型中断状态位清零，应用程序必须向该位写入 1。

FIFO 状态中断为只读；如果软件在处理这些中断期间对 FIFO 执行读写操作，则 FIFO 中断标志将自动清零。

使能中断位前，应用程序必须在初始化时将 OTG_GINTSTS 寄存器清零，才可以避免在初始化前产生任何中断。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
WKUP INT	SRQ INT	DISC INT	CIDS CHG	LPM INT	PTXFE	HCINT	HPRT INT	RST DET	Res.	IPXFR/ IN COMP ISO OUT	IISOI XFR	OEP INT	IEPINT	Res.	Res.
rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	r	r	r	rc_w1		rc_w1	rc_w1	r	r		
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EOPF	ISOO DRP	ENUM DNE	USB RST	USB SUSP	ESUSP	Res.	Res.	GO NAK EFF	GI NAK EFF	NPTXF E	RXF LVL	SOF	OTG INT	MMIS	CMOD
rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1			r	r	r	r	rc_w1	r	rc_w1	r

位 31 **WKUPINT**：检测到恢复/远程唤醒中断 (Resume/remote wakeup detected interrupt)

挂起 (L2) 或 LPM(L1) 状态期间的唤醒中断。

– 挂起 (L2) 期间：

在设备模式下，当 USB 总线上检测到恢复信号时，将触发该中断。在主机模式下，当 USB 上检测到远程唤醒时，将触发该中断。

– LPM(L1) 期间：

USB 上检测到主机发起的恢复信号或设备发起的远程唤醒时，将触发该中断。

注： 在设备模式和主机模式均可访问。

位 30 **SRQINT**：检测到会话请求/新会话中断 (Session request/new session detected interrupt)

在主机模式下，当检测到来自设备的会话请求时，将触发该中断。在设备模式下，当 V_{BUS} 在 B 外设设备的有效范围内时，将触发该中断。在设备模式和主机模式均可访问。

位 29 **DISCINT**：检测到断开连接中断 (Disconnect detected interrupt)

当检测到设备断开连接时触发该中断。

注： 仅可在主机模式下访问。

位 28 **CIDSCHG**：连接器 ID 线状态更改 (Connector ID status change)

当连接器 ID 线状态发生更改时，模块将该位置 1。

注： 在设备模式和主机模式均可访问。

位 27 **LPMINT**：LPM 中断 (LPM interrupt)

在设备模式下，当设备接收到 LPM 事务并以非 ERROR 应答时，将触发该中断。
在主机模式下，当设备以非 ERROR 应答 LPM 事务或主机模块完成所编程次数（OTG_GLPMCFG 中的 RETRYCNT 位）的 LPM 事务时，将触发该中断。
只有 OTG_GLPMCFG 中的 LPMEN 位置 1 时，该字段才有效。

位 26 **PTXFE**：周期性 Tx FIFO 为空 (Periodic Tx FIFO empty)

当周期性发送 FIFO 为半空或全空状态，且周期性请求队列中存在可写入至少一个条目的空间时，将触发该中断。该 FIFO 为半空状态还是全空状态由 OTG_GAHBCFG 寄存器中的周期性 Tx FIFO 空门限位（OTG_GAHBCFG 中的 PTXFELVL 位）决定。

注： 仅可在主机模式下访问。

位 25 **HCINT**：主机通道中断 (Host channels interrupt)

模块将该位置 1 时，指示模块中一个通道上存在挂起的中断（在主机模式下）。应用程序必须读取 OTG_HAINT 寄存器，以确定发生中断的通道的准确编号，然后读取相应的 OTG_HCINTx 寄存器，以确定引发中断的确切原因。应用程序必须先将 OTG_HCINTx 寄存器的相应状态位清零，之后才能将该位清零。

注： 仅可在主机模式下访问。

位 24 **HPRTINT**：主机端口中断 (Host port interrupt)

模块将该位置 1 时，指示主机模式下 OTG_FS 控制器端口的状态发生变化。应用程序必须读取 OTG_HPRT 寄存器，以确定引发此中断的确切事件。应用程序必须先将 OTG_HPRT 寄存器的相应状态位清零，之后才能将该位清零。

注： 仅可在主机模式下访问。

位 23 **RSTDET**：检测到复位中断 (Reset detected interrupt)

在设备模式下，若设备处于挂起状态，则在部分断电模式下的 USB 上检测到复位时，将触发该中断。

注： 仅可在设备模式下访问。

位 22 保留，必须保持复位值

位 21 **IPXFR**：未完成周期性传输 (Incomplete periodic transfer)

在主机模式下，如果存在仍处于挂起状态的未完成周期性事务，而这些事务计划在当前帧期间完成，则模块会将该中断位置 1。

INCOMPISOOUT：未完成 OUT 同步传输 (Incomplete isochronous OUT transfer)

在设备模式下，模块将该中断置 1 时，指示当前帧中至少有一个同步 OUT 端点上的传输未完成。该中断随该寄存器中的周期性帧结束中断 (EOPF) 位一同触发。

位 20 **IISOIXFR**：未完成 IN 同步传输 (Incomplete isochronous IN transfer)

模块将该中断置 1 时，指示当前帧中至少有一个同步 IN 端点上的传输未完成。该中断随该寄存器中的周期性帧结束中断 (EOPF) 位一同触发。

注： 仅可在设备模式下访问。

位 19 **OEPINT**：OUT 端点中断 (OUT endpoint interrupt)

模块将该位置 1 时，指示模块中一个 OUT 端点上存在挂起的中断（在设备模式下）。应用程序必须读取 OTG_DAINT 寄存器，以确定发生中断的 OUT 端点的准确编号，然后读取相应的 OTG_DOEPINTx 寄存器，以确定引发中断的确切原因。应用程序必须先将相应 OTG_DOEPINTx 寄存器的相应状态位清零，之后才能将该位清零。

注： 仅可在设备模式下访问。

位 18 **IEPINT**：IN 端点中断 (IN endpoint interrupt)

模块将该位置 1 时，指示模块中一个 IN 端点上存在挂起的中断（在设备模式下）。应用程序必须读取 OTG_DAINT 寄存器，以确定发生中断的 IN 端点的准确编号，然后读取相应的 OTG_DIEPINTx 寄存器，以确定引发中断的确切原因。应用程序必须先将相应 OTG_DIEPINTx 寄存器的相应状态位清零，之后才能将该位清零。

注： 仅可在设备模式下访问。

位 17:16 保留，必须保持复位值。

位 15 **EOPF**：周期性帧结束中断 (End of periodic frame interrupt)

指示当前帧已达到 OTG_DCFG 寄存器中周期性帧间隔字段（OTG_DCFG 中的 PFIVL 位）所指定的周期。

注： 仅可在设备模式下访问。

位 14 **ISOODRP**：丢弃同步 OUT 数据包中断 (Isochronous OUT packet dropped interrupt)

如果由于 Rx FIFO 空间不足，无法容纳同步 OUT 端点的最大数据包，从而导致模块无法向 Rx FIFO 写入同步 OUT 数据包，模块会将该位置 1。

注： 仅可在设备模式下访问。

位 13 **ENUMDNE**：枚举完成 (Enumeration done)

模块将该位置 1 时，指示速率枚举已完成。应用程序必须读取 OTG_DSTS 寄存器来获取枚举速率。

注： 仅可在设备模式下访问。

位 12 **USBRST**：USB 复位 (USB reset)

模块将该位置 1 时，指示在 USB 上检测到复位信号。

注： 仅可在设备模式下访问。

位 11 **USBSUSP**：USB 挂起 (USB suspend)

模块将该位置 1 时，指示在 USB 上检测到挂起状态。当数据线上的空闲状态保持一段额外的时间后，模块进入挂起状态。

注： 仅可在设备模式下访问。

位 10 **ESUSP**：早期挂起 (Early suspend)

模块将该位置 1 时，指示已检测到 USB 处于空闲状态的时间达到 3 ms。

注： 仅可在设备模式下访问。

位 9:8 保留，必须保持复位值。

位 7 **GONAKEFF**：全局 OUT NAK 有效 (Global OUT NAK effective)

指示 OTG_DCTL 寄存器中由应用程序设置的“将全局 OUT NAK 置 1”位（OTG_DCTL 中的 SGONAK 位）已在模块中生效。通过写入 OTG_DCTL 寄存器中的“将全局 OUT NAK 清零”位（OTG_DCTL 中的 CGONAK 位），可将该位清零。

注： 仅可在设备模式下访问。

位 6 **GINAKEFF**：全局非周期性 IN NAK 有效 (Global IN nonperiodic NAK effective)

指示 OTG_DCTL 寄存器中由应用程序设置的“将全局非周期性 IN NAK 置 1”位（OTG_DCTL 中的 SGINAK 位）已在模块中生效。也就是说，模块已对应用程序设置的全局 IN NAK 位进行采样，结果已生效。通过清零 OTG_DCTL 寄存器中的“将全局非周期性 IN NAK 清零”位（OTG_DCTL 中的 CGINAK 位），可将该位清零。

此中断不一定表示 USB 上已发送了一个 NAK 握手信号。STALL 位优先级高于 NAK 位。

注： 仅可在设备模式下访问。

位 5 **NPTXFE**：非周期性 Tx FIFO 为空 (Non-periodic Tx FIFO empty)

当非周期性 Tx FIFO 为半空或全空状态，且非周期性发送请求队列中至少存在可写入一个条目的空间时，将触发该中断。该 FIFO 为半空状态还是全空状态由 OTG_GAHBCFG 寄存器中的非周期性 Tx FIFO 空门限位（OTG_GAHBCFG 中的 TXFELVL 位）决定。

注： 仅可在主机模式下访问。

位 4 **RXFLVL**：Rx FIFO 非空 (Rx FIFO non-empty)

指示 Rx FIFO 中至少有一个数据包等待读取。

注： 在主机模式和设备模式均可访问。

位 3 **SOF**：帧起始 (Start of frame)

在主机模式下，模块将该位置 1 时，指示 USB 上已发送一个 SOF (FS) 或 Keep-Alive (LS) 信号。应用程序必须将此位置 1 才可清除该中断。

在设备模式下，模块将该位置 1 时，指示 USB 上已接收到一个 SOF 令牌。应用程序可通过读取 OTG_DSTS 寄存器来获得当前的帧编号。只有在模块以 FS 模式运行时，才会出现此中断。

注： 如果上电复位后立即读取，该寄存器可能返回“1”。如果寄存器位在上电复位后立即读取“1”，则不表示已发送 SOF（主机模式下）或已接收 SOF（设备模式下）。只有在主机和设备之间建立有效连接后，此中断的读取值才有效。如果此位在上电复位后置 1，应用程序可把该位清零。

注： 在主机模式和设备模式均可访问。

位 2 **OTGINT**：OTG 中断 (OTG interrupt)

模块将该位置 1 时，指示出现 OTG 协议事件。应用程序必须读取 OTG 中断状态 (OTG_GOTGINT) 寄存器，以确定引发此中断的确切事件。应用程序必须先将 OTG_GOTGINT 寄存器的相应状态位清零，之后才能将该位清零。

注： 在主机模式和设备模式均可访问。

位 1 **MMIS**：模式不匹配中断 (Mode mismatch interrupt)

当应用程序尝试访问以下寄存器时，模块将该位置 1：

– 模块运行在设备模式下访问主机模式寄存器

– 模块运行在主机模式下访问设备模式寄存器

寄存器访问在 AHB 上以 OKAY 响应结束，但该访问在内部被模块忽略并且不会影响模块运行。

注： 在主机模式和设备模式均可访问。

位 0 **CMOD**：当前工作模式 (Current mode of operation)

指示当前模式。

0：设备模式

1：主机模式

注： 在主机模式和设备模式均可访问。

50.15.7 OTG 中断屏蔽寄存器 (OTG_GINTMSK)

OTG interrupt mask register

偏移地址：0x018

复位值：0x0000 0000

该寄存器与模块中断寄存器结合使用，以中断应用程序。如果将某个中断位屏蔽，则不会产生与该位相关的中断。但是，与该中断相对应的模块中断 (OTG_GINTSTS) 寄存器位仍会置 1。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
WUIM	SRQIM	DISCINT	CIDSCHGM	LPMINTM	PTXFEM	HCIM	PRTIM	RSTDETM	Res.	IPXFRM/IISOOXFRM	IISOIXFRM	OEPINT	IEPINT	Res.	Res.
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
EOPFM	ISOODRPM	ENUMDNEM	USBRST	USBSUSPM	ESUSPM	Res.	Res.	GONAKEFFM	GINAKEFFM	NPTXFEM	RXFLVLM	SOFM	OTGINT	MMISM	Res.
rw	rw	rw	rw	rw	rw			rw	rw	rw	rw	rw	rw	rw	

位 31 **WUIM**：检测到恢复/远程唤醒中断屏蔽 (Resume/remote wakeup detected interrupt mask)
0：屏蔽中断
1：使能中断
注： 在主机模式和设备模式均可访问。

位 30 **SRQIM**：检测到会话请求/新会话中断屏蔽 (Session request/new session detected interrupt mask)
0：屏蔽中断
1：使能中断
注： 在主机模式和设备模式均可访问。

位 29 **DISCINT**：检测到断开连接中断屏蔽 (Disconnect detected interrupt mask)
0：屏蔽中断
1：使能中断
注： 仅可在主机模式下访问。

位 28 **CIDSCHGM**：连接器 ID 状态更改屏蔽 (Connector ID status change mask)
0：屏蔽中断
1：使能中断
注： 在主机模式和设备模式均可访问。

位 27 **LPMINTM**：LPM 中断屏蔽 (LPM interrupt mask)
0：屏蔽中断
1：使能中断
注： 在主机模式和设备模式均可访问。

位 26 **PTXFEM**：周期性 Tx FIFO 空屏蔽 (Periodic Tx FIFO empty mask)
0：屏蔽中断
1：使能中断
注： 仅可在主机模式下访问。

位 25 **HCIM**：主机通道中断屏蔽 (Host channels interrupt mask)
0：屏蔽中断
1：使能中断
注： 仅可在主机模式下访问。

位 24 **PRTIM**：主机端口中断屏蔽 (Host port interrupt mask)
0：屏蔽中断
1：使能中断
注： 仅可在主机模式下访问。

位 23 **RSTDETM**：检测到复位中断屏蔽 (Reset detected interrupt mask)
0：屏蔽中断
1：使能中断
注： 仅可在设备模式下访问。

位 22 保留，必须保持复位值

位 21 **IPXFRM**：未完成周期性传输中断屏蔽 (Incomplete periodic transfer mask)
0：屏蔽中断
1：使能中断
注： 仅可在主机模式下访问。
IISOOXFRM：未完成 OUT 同步传输中断屏蔽 (Incomplete isochronous OUT transfer mask)
0：屏蔽中断
1：使能中断
注： 仅可在设备模式下访问。

位 20 **IISOIXFRM**：未完成 IN 同步传输中断屏蔽 (Incomplete isochronous IN transfer mask)
0：屏蔽中断
1：使能中断
注： 仅可在设备模式下访问。

位 19 **OEPINT**：OUT 端点中断屏蔽 (OUT endpoints interrupt mask)
0：屏蔽中断
1：使能中断
注： 仅可在设备模式下访问。

位 18 **IEPINT**：IN 端点中断屏蔽 (IN endpoints interrupt mask)
0：屏蔽中断
1：使能中断
注： 仅可在设备模式下访问。

位 17:16 保留，必须保持复位值。

位 15 **EOPFM**：周期性帧结束中断屏蔽 (End of periodic frame interrupt mask)
0：屏蔽中断
1：使能中断
注： 仅可在设备模式下访问。

位 14 **ISOODRPM**：丢弃同步 OUT 数据包中断屏蔽 (Isochronous OUT packet dropped interrupt mask)
0：屏蔽中断
1：使能中断
注： 仅可在设备模式下访问。

位 13 **ENUMDNEM**：枚举完成中断屏蔽 (Enumeration done mask)
0：屏蔽中断
1：使能中断
注： 仅可在设备模式下访问。

位 12 **USBRST**：USB 复位中断屏蔽 (USB reset mask)
0：屏蔽中断
1：使能中断
注： 仅可在设备模式下访问。

位 11 **USBSUSPM**：USB 挂起中断屏蔽 (USB suspend mask)
0：屏蔽中断
1：使能中断
注： 仅可在设备模式下访问。

位 10 **ESUSPM**：早期挂起中断屏蔽 (Early suspend mask)
0：屏蔽中断
1：使能中断
注： 仅可在设备模式下访问。

位 9:8 保留，必须保持复位值。

位 7 **GONAKEFFM**：全局 OUT NAK 生效中断屏蔽 (Global OUT NAK effective mask)
0：屏蔽中断
1：使能中断
注： 仅可在设备模式下访问。

位 6 **GINAKEFFM**：全局非周期性 IN NAK 生效中断屏蔽 (Global non-periodic IN NAK effective mask)
0：屏蔽中断
1：使能中断
注： 仅可在设备模式下访问。

位 5 **NPTXFEM**：非周期性 Tx FIFO 空中断屏蔽 (Non-periodic Tx FIFO empty mask)
0：屏蔽中断
1：使能中断
注： 仅可在主机模式下访问。

位 4 **RXFLVLM**：接收 FIFO 非空中断屏蔽 (Receive FIFO non-empty mask)
0：屏蔽中断
1：使能中断
注： 在设备模式和主机模式均可访问。

位 3 **SOFM**：帧起始中断屏蔽 (Start of frame mask)
0：屏蔽中断
1：使能中断
注： 在设备模式和主机模式均可访问。

位 2 **OTGINT**：OTG 中断屏蔽 (OTG interrupt mask)
0：屏蔽中断
1：使能中断
注： 在设备模式和主机模式均可访问。

位 1 **MMISM**：模式不匹配中断屏蔽 (Mode mismatch interrupt mask)
0：屏蔽中断
1：使能中断
注： 在设备模式和主机模式均可访问。

位 0 保留，必须保持复位值。

50.15.8 OTG 接收状态调试读取/OTG 状态读取和出栈寄存器 (OTG_GRXSTSR/OTG_GRXSTSP)

OTG receive status debug read/OTG status read and pop registers

读取的偏移地址：0x01C

出栈的偏移地址：0x020

复位值：0x0000 0000

读取接收状态调试读取寄存器将返回接收 FIFO 顶部的内容。读取接收状态读取和出栈寄存器将额外弹出 Rx FIFO 顶部的数据条目。

接收状态内容在主机模式和设备模式下的解释不同。当接收 FIFO 为空时，模块会忽略对该寄存器的读取或出栈操作，并返回值 0x0000 0000。当模块中断寄存器的接收 FIFO 非空位（OTG_GINTSTS 中的 RXFLVL 位）置位时，应用程序必须仅弹出接收状态 FIFO。

主机模式：

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.	PKTSTS[3:0]				DPID
											r	r	r	r	r

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

位 31:21 保留，必须保持复位值。

位 20:17 **PKTSTS[3:0]**：数据包状态 (Packet status)
指示接收的数据包的状态
0010：接收到 IN 数据包
0011：IN 传输完成（触发中断）
0101：数据翻转错误（触发中断）
0111：通道停止（触发中断）
其他值：保留

位 16:15 **DPID**：数据 PID (Data PID)
指示接收的数据包的数据 PID
00：DATA0
10：DATA1

位 14:4 **BCNT[10:0]**：字节数 (Byte count)
指示接收的 IN 数据包的字节数。

位 3:0 **CHNUM[3:0]**：通道编号 (Channel number)
指示当前接收的数据包所属的通道编号。

设备模式：

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	STSPHST	Res.	Res.	FRMNUM[3:0]				PKTSTS[3:0]				DPID[1]
				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
DPID[0]	BCNT[10:0]											EPNUM[3:0]			
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:28 保留，必须保持复位值。

位 27 **STSPHST**：起始状态阶段 (Status phase start)
指示控制写传输的起始状态阶段。该位与 OUT 传输完成的 PKTSTS 模式均置 1。

位 26:25 保留，必须保持复位值。

位 24:21 **FRMNUM[3:0]**：帧数 (Frame number)
这是在 USB 上接收的数据包帧编号的 4 个最低有效位。只有当支持同步 OUT 端点时才支持此字段。

位 20:17 **PKTSTS[3:0]**：数据包状态 (Packet status)
指示接收的数据包的状态
0001：全局 OUT NAK（触发中断）
0010：接收到 OUT 数据包
0011：OUT 传输完成（触发中断）
0100：SETUP 事务完成（触发中断）
0110：接收到 SETUP 数据包
其他值：保留

位 16:15 **DPID[1:0]**：数据 PID (Data PID)
指示接收的 OUT 数据包的数据 PID
00：DATA0
10：DATA1

位 14:4 **BCNT[10:0]**：字节数 (Byte count)
指示接收的数据包的字节数。

位 3:0 **EPNUM[3:0]**：端点编号 (Endpoint number)
指示当前接收的数据包所属的端点编号。

50.15.9 OTG 接收 FIFO 大小寄存器 (OTG_GRXFSIZ)

OTG receive FIFO size register

偏移地址：0x024

复位值：0x0000 0200

此应用程序可以对分配给 Rx FIFO 的 RAM 大小进行编程。

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
RXFD[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 **RXFD[15:0]**：Rx FIFO 深度 (Rx FIFO depth)
以 32 位字为单位。
最小值为 16。
编程值必须遵循可分配的 FIFO 存储区，不得超过上电值。

50.15.10 OTG 主机非周期性发送 FIFO 大小寄存器 (OTG_HNPTXFSIZ)/ 端点 0 发送 FIFO 大小 (OTG_DIEPTXF0)

OTG host non-periodic transmit FIFO size register/Endpoint 0 Transmit FIFO size

偏移地址：0x028

复位值：0x0200 0200

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NPTXFD/TX0FD[15:0]															
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
NPTXFSA/TX0FSA[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

主机模式

位 31:16 **NPTXFD[15:0]**：非周期性 Tx FIFO 深度 (Non-periodic Tx FIFO depth)
以 32 位字为单位。
最小值为 16。
编程值必须遵循可分配的 FIFO 存储区，不得超过上电值。

位 15:0 **NPTXFSA[15:0]**：非周期性发送 RAM 起始地址 (Non-periodic transmit RAM start address)
此字段配置非周期性发送 FIFO RAM 的存储器起始地址。

设备模式

位 31:16 **TX0FD**：端点 0 Tx FIFO 深度 (Endpoint 0 Tx FIFO depth)
以 32 位字为单位。
最小值为 16。
编程值必须遵循可分配的 FIFO 存储区，不得超过上电值。

位 15:0 **TX0FSA**：端点 0 发送 RAM 起始地址 (Endpoint 0 transmit RAM start address)
此字段配置端点 0 发送 FIFO RAM 的存储器起始地址。

50.15.11 OTG 非周期性发送 FIFO/队列状态寄存器 (OTG_HNPTXSTS)

OTG non-periodic transmit FIFO/queue status register

偏移地址：0x02C

复位值：0x0008 0200

注： *在设备模式下，此寄存器无效。*

此只读寄存器包含非周期性 Tx FIFO 和非周期性发送请求队列的空闲空间信息。

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

位 31 保留，必须保持复位值。

位 30:24 **NPTXQTOP[6:0]**：非周期性发送请求队列顶部 (Top of the non-periodic transmit request queue)
非周期性发送请求队列中 MAC 目前正在处理的条目。
位 30:27：通道/端点编号 (Channel/endpoint number)
位 26:25：
00：IN/OUT 令牌
01：长度为零的发送数据包（设备 IN/主机 OUT）
11：通道停止命令
位 24：结束（所选通道/端点的最后一个条目）(Terminate (last entry for selected channel/endpoint))

位 23:16 **NPTQXSAV[7:0]**：非周期性发送请求队列可用空间 (Non-periodic transmit request queue space available)

指示非周期性发送请求队列中的可用空闲空间大小。该队列既包含 IN 请求，又包含 OUT 请求。

0：非周期性发送请求队列已满

1：1 个位置可用

2：2 个位置可用

n：n 个位置可用 (0 ≤ n ≤ 8)

其他值：保留

位 15:0 **NPTXFSAV[15:0]**：非周期性 Tx FIFO 可用空间 (Non-periodic Tx FIFO space available)

指示非周期性 Tx FIFO 中的可用空闲空间大小。

以 32 位字为单位。

0：非周期性 TxFIFO 已满

1：1 个字可用

2：2 个字可用

n：n 个位置可用（其中，0 ≤ n ≤ 512）

其他值：保留

50.15.12 OTG 通用模块配置寄存器 (OTG_GCCFG)

OTG general core configuration register

偏移地址：0x038

复位值：0x0000 XXXX

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.	VBDEN	SDEN	PDEN	DCD EN	BCDEN	PWR DWN
										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.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PS2 DET	SDET	PDET	DCDET
												r	r	r	r

位 31:22 保留，必须保持复位值。

位 21 **VBDEN**：USB V_{BUS} 检测使能 (USB V_{BUS} detection enable)

使能 V_{BUS} 感应比较器检测 USB 主机和设备工作模式下的 V_{BUS} 引脚上的 V_{BUS} 有效电平。如果使能 HNP 和/或 SRP 支持，将自动使能 V_{BUS} 比较器，而与 VBDEN 值无关。

0 = 禁止 V_{BUS} 检测

1 = 使能 V_{BUS} 检测

位 20 **SDEN**：二次检测 (SD) 模式使能 (Secondary detection (SD) mode enable)

该位由软件置 1，用于将 BCD 置于 SD 模式。要正确工作，应只选择一个检测模式（DCD、PD、SD 或 OFF）

位 19 **PDEN**：初次检测 (PD) 模式使能 (Primary detection (PD) mode enable)

该位由软件置 1，用于将 BCD 置于 PD 模式。要正确工作，应只选择一个检测模式（DCD、PD、SD 或 OFF）。

位 18 **DCDEN**：数据接触点检测 (DCD) 模式使能 (Data contact detection (DCD) mode enable)
该位由软件置 1，用于将 BCD 置于 DCD 模式。要正确工作，应只选择一个检测模式（DCD、PD、SD 或 OFF）。

位 17 **BCDEN**：电池充电器检测 (BCD) 使能 (Battery charging detector (BCD) enable)
该位由软件置 1，用于在 USB 设备内使能 BCD 支持。使能后，USB PHY 完全由 BCD 控制，无法用于正常通信。BCD 检测完成后，应通过将该位清零使 BCD 置于 OFF 模式，以实现正常的 USB 操作。

位 16 **PWRDWN**：掉电控制 (Power down control)
用于在发送/接收时激活收发器。复位时，收发器保持掉电状态。置 1 时，必须关闭 BCD 功能 (BCDEN=0)。
0 = 禁止 USB FS 收发器
1 = 使能 USB FS 收发器

位 15:4 保留，必须保持复位值。

位 3 **PS2DET**：DM 上拉检测状态 (DM pull-up detection status)
该位只在 PD 期间有效，用于指示 DM 电压电平与 VLGC 阈值之间的比较结果。正常情况下，DM 电平应低于此阈值。如果 DM 电平高于此阈值，则意味着 DM 已外部拉为高电平。原因可能是连接到 PS2 端口（同时上拉 DP 和 DM 线）或某些不遵循 BCD 规范的专有充电器。
0：检测到正常端口（连接到 SDP、CDP 或 DCP）
1：检测到 PS2 端口或专有充电器

位 2 **SDEN**：二次检测 (SD) 状态 (Secondary detection (SD) status)
此位指示 SD 的结果。
0：检测到 CDP
1：检测到 DCP

位 1 **PDEN**：初次检测 (PD) 状态 (Primary detection (PD) status)
此位指示 PD 的结果。
0：未检测到 BCD 支持（连接到 SDP 或专有设备）。
1：检测到 BCD 支持（连接到 CDP 或 DCP）。

位 0 **DCDEN**：数据接触点检测 (DCD) 状态 (Data contact detection (DCD) status)
此位指示 DCD 的结果。
0：未检测到数据点接触
1：检测到数据点接触

50.15.13 OTG 模块 ID 寄存器 (OTG_CID)

OTG core ID register

偏移地址：0x03C

复位值：0x0000 3000

这是一个包含产品 ID 作为复位值的寄存器。

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

位 31:0 **PRODUCT_ID[31:0]**：产品 ID 字段 (Product ID field)
可通过应用程序编程的 ID 字段。

50.15.14 OTG 模块 LPM 配置寄存器 (OTG_GLPMCFG)

OTG core LPM configuration register

偏移地址：0x54

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	EN BESL	LPMRCNTSTS[2:0]			SND LPM	LPMRCNT[2:0]			LPMCHIDX[3:0]				L1RSM OK
			rw	r	r	r	rs	rw	rw	rw	rw	rw	rw	rw	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SLP STS	LPMRSP[1:0]		L1DS EN	BESLTHRS[3:0]				L1SS EN	REM WAKE	BESL[3:0]				LPM ACK	LPM EN
r	r	r	rw	rw	rw	rw	rw	rw	rw/r	rw/r	rw/r	rw/r	rw/r	rw	rw

位 31:29 保留，必须保持复位值。

位 28 **ENBESL**：使能尽力服务延迟 (Enable best effort service latency)
该位用于使能 LPM 勘误表中定义的 BESL 功能：
0：模块按照以下文档中的说明工作：
基于 USB 2.0 规范的 USB 2.0 链路层电源管理补充工程变更通知，2007 年 7 月 16 日
1：模块按照 LPM 勘误表中的说明工作：
USB 2.0 ECN 的勘误表：链路层电源管理 (LPM) - 2007 年 7 月
注： 本文档只考虑（LPM 勘误表中说明的）更新行为，因此 ENBESL 位应由应用程序软件置 1。

位 27:25 **LPMRCNTSTS[2:0]**：LPM 重试计数状态 (LPM retry count status)
当前 LPM 序列仍要发送的 LPM 主机重试次数
注： 仅可在主机模式下访问。

位 24 **SNDLPM**：发送 LPM 事务 (Send LPM transaction)

当应用程序软件将该位置 1 时，将发送包含两个令牌 EXT 和 LPM 的 LPM 事务。从设备接收到有效响应（STALL、NYET 或 ACK）或模块发送完编程的 LPM 重试次数后，硬件即将该位清零。

注： 只有当主机连接到本地端口时，该位才可置 1。

注： 仅可在主机模式下访问。

位 23:21 **LPMRCNT[2:0]**：LPM 重试计数 (LPM retry count)

当设备发出 ERROR 响应后，该位表示接收到有效设备响应（STALL、NYET 或 ACK）之前，主机需额外执行的 LPM 重试次数。

注： 仅可在主机模式下访问。

位 20:17 **LPMCHIDX[3:0]**：LPM 通道索引 (LPM Channel Index)

向本地设备发送 LPM 事务时，必须应用到 LPM 事务的通道编号。基于 LPM 通道编号，模块可自动向 LPM 事务插入相应通道中编程的设备地址和端点号。

注： 仅可在主机模式下访问。

位 16 **L1RSMOK**：睡眠状态恢复正常 (Sleep state resume OK)

表示设备或主机可以从睡眠状态启动恢复。该位在 LPM 睡眠 (L1) 状态下有效。它在睡眠模式下经 50 µs ($T_{L1Residency}$) 延时后置 1。

该位在 SLPSTS 为 0 时复位。

1：应用程序或主机可以从睡眠状态启动恢复

0：应用程序或主机无法从睡眠状态启动恢复

位 15 **SLPSTS**：端口睡眠状态 (Port sleep status)

设备模式：

只要 USB 总线上存在睡眠条件，该位就会置 1。当 ACK 响应发送给 LPM 事务且 $T_{L1TokenRetry}$ 定时器过期时，模块将进入睡眠模式。要停止 PHY 时钟，应用程序必须将 OTG_PCGCCTL 中的 STPPCLK 位置 1，这将触发 PHY 挂起输入信号。

应用程序必须依靠 LPMRSP 中的 SLPSTS 而不是 ACK 来确认切换到睡眠状态。

出现以下情况时，模块会退出睡眠状态：

– USB 线上有活动时。

– 当应用程序对 OTG_DCTL 中的 RWUSIG 位执行写操作或者复位或软断开设备时。

主机模式：

模块通过来自设备的 ACK 响应成功将 LPM 事务发送给本地端口后，主机将切换到睡眠 (L1) 状态。此位的读取值反映该端口的当前睡眠状态。

在以下事件后，模块会将该位清零：

– 模块检测到远程 L1 唤醒信号，

– 应用程序将 OTG_HPRT 寄存器中的 PRST 位或 PRES 位置 1，或者

– 应用程序将模块中断寄存器中的“检测到 L1 恢复/远程唤醒中断”位或“检测到断开连接中断”位（分别为 OTG_GINTSTS 中的 WKUPINT 或 DISCINT 位）置 1。

0：模块未处于 L1

1：模块处于 L1

位 14:13 **LPMRST[1:0]**：LPM 响应 (LPM response)

设备模式：

这两位反映模块对所接收的 LPM 事务的响应。

主机模式：

从本地设备接收、用于 LPM 事务的握手响应。

11：ACK

10：NYET

01：STALL

00：ERROR（无握手响应）

位 12 **L1DSEN**：L1 深度睡眠使能 (L1 deep sleep enable)

在 L1 睡眠模式下使能挂起 PHY。为了在 L1 睡眠模式下最大程度节能，所有情况下该位均应由应用程序软件置 1。

位 11:8 **BESLTHRS[3:0]**：BESL 阈值 (BESL threshold)

设备模式：

当 BESL 值大于或等于 BESL_Thres[3:0] 字段中定义的值时，模块将 PHY 置于 L1 下的深度低功耗模式。

主机模式：

模块将 PHY 置于 L1 下的深度低功耗模式。BESLTHRS[3:0] 用于指定要由主机在 USB 总线上反映的恢复信号的持续时间 ($T_{L1HubDrvResume2}$)。

在主机模式下，BESLTHRS 不得编程为大于 1100b 的值，原因是这将超出最大 $T_{L1HubDrvResume2}$。

Thres[3:0] 主机模式恢复信号的持续时间 (µs)：

0000: 75

0001: 100

0010: 150

0011: 250

0100: 350

0101: 450

0110: 950

所有其他值：保留

位 7 **L1SSEN**：L1 浅度睡眠使能 (L1 Shallow Sleep enable)

在 L1 睡眠模式下使能挂起 PHY。为了在 L1 睡眠模式下最大程度节能，所有情况下该位均应由应用程序软件置 1。

位 6 **REMWAKE**：bRemoteWake 值 (bRemoteWake value)

主机模式：

要在 LPM 事务的 wIndex 字段中发送的远程唤醒值。

设备模式（只读）：

当 ACK、NYET 或 STALL 响应发送给 LPM 事务时，该字段用接收的 LPM 令牌 bRemoteWake bmAttribute 进行更新。

位 5:2 **BESL[3:0]**：尽力服务延迟 (Best effort service latency)

主机模式：

要在 LPM 事务中发送的 BESL 值。该值还用于发起持续时间为 $T_{L1HubDrvResume1}$ 的恢复信号，以实现主机发起的恢复操作。

设备模式（只读）：

当 ACK、NYET 或 STALL 响应发送给 LPM 事务时，该字段用接收的 LPM 令牌 BESL bmAttribute 进行更新。

BESL[3:0]T_{BESL} (µs)

0000: 125
0001: 150
0010: 200
0011: 300
0100: 400
0101: 500
0110: 1000
0111: 2000
1000: 3000
1001: 4000
1010: 5000
1011: 6000
1100: 7000
1101: 8000
1110: 9000
1111: 10000

位 1 **LPMACK**：LPM 令牌应答使能 (LPM token acknowledge enable)

设备应用程序软件预编程的 LPM 令牌握手响应。

1：ACK

即使已预编程 ACK，也只有在成功完成 LPM 事务后，模块才能以 ACK 进行响应。LPM 事务成功的前提是：

– EXT 令牌或 LPM 令牌中没有 PID/CRC5 错误（否则会出现 ERROR）
– LPM 事务中接收到 Valid bLinkState = 0001B (L1)（否则会出现 STALL）
– 发送队列中没有待处理的数据（否则会出现 NYET）

0：NYET

在以下情况下，为响应 LPM 令牌，会重写预编程的软件位：

– 接收到的 bLinkState 不是 L1（STALL 响应），或者
– 由于任一 LPM 令牌数据包损坏而在其中检测到错误（ERROR 响应）

注：　仅可在设备模式下访问。

位 0 **LPMEN**：LPM 支持使能 (LPM support enable)

应用程序使用该位控制 OTG_FS 模块的 LPM 功能。

如果模块以不支持 LPM 功能的主机工作，则无法请求所连接的设备或集线器激活 LPM 模式。

如果模块以不支持 LPM 功能的设备工作，则无法响应任何 LPM 事务。

0：不使能 LPM 功能

1：使能 LPM 功能

50.15.15 OTG 掉电寄存器 (OTG_GPWRDN)

OTG power down register

偏移地址：0x058

复位值：0x0000 0010

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ADPIF	Res.	Res.	Res.	Res.	Res.	Res.	Res.
								rc_w1							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ADPMEN
															rw

位 31:24 保留，必须保持复位值。

位 23 **ADPIF**：ADP 中断标志 (ADP interrupt flag)
只要发生 ADP 事件，该位便置 1。

位 22:1 保留，必须保持复位值。

位 0 **ADPMEN**：ADP 模块使能 (ADP module enable)
该位用于使能或禁止 ADP 逻辑。
0：禁止 ADP 模块
1：使能 ADP 模块

50.15.16 OTG ADP 定时器、控制和状态寄存器 (OTG_GADPCTL)

OTG ADP timer, control and status register

偏移地址：0x060

复位值：0x0000 0000

OTG_GADPCTL 寄存器必须按如下方式访问：

- 为了读取 OTG_GADPCTL 寄存器，向 AR 域写入 0b01，然后轮询回读，直到读回的值是 0b00。模块更新该寄存器的其他字段，并使 AR=0b00。仅当 AR=0b00 时，该寄存器的读出值才有效。
- 要对 OTG_GADPCTL 寄存器执行写操作，写 AR 域写入 0b10，并同时设置寄存器内的其他字段，然后轮询回读 AR 域，直到读回的值是 0b00。当 AR 变为 0b00 时，意味着编程的值已在模块内部生效。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	AR		ADP TOIM	ADP SNSIM	ADP PRBIM	ADP TOIF	ADP SNSIF	ADP PRBIF	ADPEN	ADP RST	ENA SNS	ENA PRB	RTIM
			rw	rw	rw	rw	rw	rc_w1	rc_w1	rc_w1	rw	rs	rw	rw	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RTIM										PRBPER		PRBDELTA		PRBDSCHG	
r	r	r	r	r	r	r	r	r	r	rw	rw	rw	rw	rw	rw

位 31:29 保留，必须保持复位值。

位 28:27 **AR**：访问请求 (Access request)
该位域用于定义 OTG_GADPCTL 寄存器的访问模式。
00 读/写有效（由模块更新）
01 读请求
10 写请求
11 保留

位 26 **ADPTOIM**：ADP 超时中断屏蔽 (ADP timeout interrupt mask)
该位置 1 时，会使能来自 ADPTOIF 的中断。

位 25 **ADPSNSIM**：ADP 感应中断屏蔽 (ADP sense interrupt mask)
该位置 1 时，会使能来自 ADPSNSIF 的中断。

位 24 **ADPPRBIM**：ADP 探测中断屏蔽 (ADP probe interrupt mask)
该位置 1 时，会使能来自 ADPPRBIF 的中断。

位 23 **ADPTOIF**：ADP 超时中断标志 (ADP timeout interrupt flag)
此位仅用于 ADP 探测。该位置 1 时，意味着已达到斜坡时间（RTIM 已达到其终值 0x7FF）。这是一种调试功能，允许应用程序在每个周期后读取斜坡时间。

位 22 **ADPSNSIF**：ADP 感应中断标志 (ADP sense interrupt flag)
该位置 1 时，意味着 V_{BUS} 电压大于 V_{ADPSNS} 值或已达到 V_{ADPSNS}。

位 21 **ADPPRBIF**：ADP 探测中断标志 (ADP probe interrupt flag)
该位置 1 时，意味着 V_{BUS} 电压大于 V_{ADPPRB} 值或已达到 V_{ADPPRB}。

位 20 **ADPEN**：ADP 使能 (ADP enable)
置 1 时，模块基于 ENAPRB 或 ENASNS 执行 ADP 探测或感应。

位 19 **ADPRST**：ADP 复位 (ADP reset)
置 1 时，ADP 控制器复位。ADP 控制器中完成复位步骤后，该位自动清零。

位 18 **ENASNS**：使能感应 (Enable sense)
编程为 1 时，模块执行感应操作。

位 17 **ENAPRB**：使能探测 (Enable probe)
编程为 1 时，模块执行探测操作。

位 16:6 **RTIM**：斜坡时间 (Ramp time)
这些位用于捕获 V_{BUS} 从 $V_{ADPSINK}$ 斜升到 V_{ADPPRB} 的最新时间。这些位以 32 kHz 时钟周期为单位定义：
000：1 个周期
001：2 个周期
002：3 个周期
...
7FF：2048 个周期
1024 个 32 kHz 周期的时间为 32 ms。

位 5:4 **PRBPER**：探测周期 (Probe period)

这些位按如下方式设置 T_{ADPPRD}：

00：0.625 到 0.925 s（典型值为 0.775 s）

01：1.25 到 1.85 s（典型值为 1.55 s）

10：1.9 到 2.6 s（典型值为 2.275 s）

11：保留

位 3:2 **PRBDELTA**：探测增量 (Probe delta)

这些位用于设置 RTIM 值的分辨率。这些位以 32 kHz 时钟周期为单位定义，具体如下：

00：1 个周期

01：2 个周期

10：3 个周期

11：4 个周期

例如，如果将该值选为 01，则意味着每两个 32 kHz 时钟周期后 RTIM 递增。

位 1:0 **PRBDSCHG**：探测放电 (Probe discharge)

这些位用于设置 T_{ADP_DSCHG} 的时间。这些位按如下定义：

00：4 ms

01：8 ms

10：16 ms

11：32 ms

50.15.17 OTG 主机周期性发送 FIFO 大小寄存器 (OTG_HPTXFSIZ)

OTG host periodic transmit FIFO size register

偏移地址：0x100

复位值：0x0200 0400

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

位 31:16 **PTXFSIZ[15:0]**：主机周期性传输 Tx FIFO 深度 (Host periodic Tx FIFO depth)

以 32 位字为单位。

最小值为 16。

位 15:0 **PTXSA[15:0]**：主机周期性传输 Tx FIFO 起始地址 (Host periodic Tx FIFO start address)

此字段用于配置周期性发送 FIFO RAM 的存储器起始地址。

50.15.18 OTG 设备 IN 端点发送 FIFO 大小寄存器 (OTG_DIEPTXFx)（x = 1..5，其中 x 为 FIFO 号）

OTG device IN endpoint transmit FIFO size register

偏移地址：0x104 + (x – 1) * 0x04

复位值：0x0200 0200 + (x * 0x200)

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

位 31:16 **INEPTXFD[15:0]**：IN 端点 Tx FIFO 深度 (IN endpoint Tx FIFO depth)
以 32 位字为单位。
最小值为 16

位 15:0 **INEPTXSA[15:0]**：IN 端点发送 FIFOx RAM 起始地址 (IN endpoint FIFOx transmit RAM start address)
此字段包含 IN 端点发送 FIFOx 的存储器起始地址。该地址必须与 32 位存储器位置对齐。

50.15.19 主机模式寄存器

除非特别说明，否则寄存器描述中的位值以二进制表示。

主机模式寄存器会影响主机模式下的模块操作。在设备模式下不得访问主机模式寄存器，因为产生的结果不明确。主机模式寄存器可进行如下分类：

50.15.20 OTG 主机配置寄存器 (OTG_HCFG)

OTG host configuration register

偏移地址：0x400

复位值：0x0000 0000

此寄存器将在上电后对模块进行配置。请勿在初始化主机后更改此寄存器。

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

位 31:3 保留，必须保持复位值。

位 2 **FSLSS**：仅支持 FS 和 LS (FS- and LS-only support)

应用程序使用此位控制模块的枚举速度。使用此位，应用程序可使模块工作为 FS 主机，即使所连接的设备支持 HS 通信也是如此。请勿在初始编程后更改此字段。

1：仅限 FS/LS，即使所连接设备可支持 HS（只读）。

位 1:0 **FSLSPCS[1:0]**：FS/LS PHY 时钟选择 (FS/LS PHY clock select)

当模块处于 FS 主机模式时

01：PHY 时钟以 48 MHz 运行

其他值：保留

当模块处于 LS 主机模式时

00：保留

01：选择 48 MHz PHY 时钟频率

10：选择 6 MHz PHY 时钟频率

11：保留

注： 当设备连上主机时，必须依照所连接设备的速度设置 FSLSPCS（更改此位后，必须进行软件复位）。

50.15.21 OTG 主机帧间隔寄存器 (OTG_HFIR)

OTG host frame interval register

偏移地址：0x404

复位值：0x0000 EA60

此寄存器用于存储 OTG_FS 控制器对已连接设备当前速度所设定的帧间隔信息。

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.	RLD CTRL
															rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FRIVL[15:0]															
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 **RLDCTRL**：重载控制 (Reload control)

该位允许在运行期间动态重载 HFIR 寄存器。

0：运行期间可动态重载 HFIR。

1：无法动态重载 HFIR。

该位需要在初始配置期间编程，并且不得在运行期间更改其值。

注意： 不建议 RLDCTRL = 1。

位 15:0 **FRIVL[15:0]**：帧间隔 (Frame interval)

应用程序在此字段编程的值用于指定两个连续 SOF (FS) 或 Keep-Alive 令牌 (LS) 之间的时间间隔。此字段包含构成所需帧间隔的 PHY 时钟数。只有将主机端口控制和状态寄存器的端口使能位（OTG_HPRT 的 PENA 位）置 1 后，应用程序才能向此寄存器中写入值。如果未对值进行编程，模块将根据在主机配置寄存器的 FS/LS PHY 时钟选择字段（OTG_HCFG 中的 FSLSPCS）中指定的 PHY 时钟来计算。除非 RLDCTRL 位置 1，否则请勿在初始配置后更改此字段的值。在这种情况下，FRIVL 将在发生每个 SOF 事件时进行重载。

– 帧间隔 = 1 ms × (FRIVL - 1)

50.15.22 OTG 主机帧编号/帧剩余时间寄存器 (OTG_HFNUM)

OTG host frame number/frame time remaining register

偏移地址：0x408

复位值：0x0000 3FFF

此寄存器用于指示当前帧编号。它还指示当前帧的剩余时间（以 PHY 时钟数为单位）。

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

位 31:16 **FTREM[15:0]**：帧剩余时间 (Frame time remaining)

指示当前帧的剩余时间（以 PHY 时钟数为单位）。每过去 1 个 PHY 时钟，此字段递减 1。当值达到零时，此字段将重新装载帧间隔寄存器中的值，并由模块在 USB 上发送一个新 SOF。

位 15:0 **FRNUM[15:0]**：帧数 (Frame number)

当在 USB 上发送 1 个新 SOF 时此字段的值将递增 1，当达到 0x3FFF 时会清零。

50.15.23 OTG_Host 周期性发送 FIFO/队列状态寄存器 (OTG_HPTXSTS)

OTG_Host periodic transmit FIFO/queue status register

偏移地址：0x410

复位值：0x0008 0100

此只读寄存器包含周期性 Tx FIFO 和周期性发送请求队列的空闲空间信息。

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

位 31:24 **PTXQTOP[7:0]**：周期性传输发送请求队列顶部 (Top of the periodic transmit request queue)

指示周期性传输 Tx 请求队列中 MAC 当前正在处理的项。

该寄存器用于调试。

位 31：奇数/偶数帧 (Odd/Even frame)

0：以偶数帧发送

1：以奇数帧发送

位 30:27：通道/端点编号 (Channel/endpoint number)

位 26:25：类型 (Type)

00：IN/OUT

01：零长度数据包

11：禁止通道命令

位 24：结束（所选通道/端点的最后一个条目）(Terminate (last entry for the selected channel/endpoint))

位 23:16 **PTXQSAV[7:0]**：周期性传输发送请求队列可用空间 (Periodic transmit request queue space available)

指示可供写入的周期性传输发送请求队列的空闲位置的数量。该队列既包含 IN 请求，又包含 OUT 请求。

00：周期性发送请求队列已满

01：1 个位置可用

10：2 个位置可用

bxn：n 个位置可用 (0 ≤ n ≤ 8)

其他值：保留

位 15:0 **PTXFSAVL[15:0]**：周期性传输发送数据 FIFO 可用空间 (Periodic transmit data FIFO space available)

指示可供写入的周期性传输 Tx FIFO 的空闲位置的数量。

以 32 位字为单位

0000：周期性 Tx FIFO 已满

0001：1 个字可用

0010：2 个字可用

bxn：n 个字可用（其中 0 ≤ n ≤ PTXFD）

其他值：保留

50.15.24 OTG 主机全体通道中断寄存器 (OTG_HAINT)

OTG host all channels interrupt register

偏移地址：0x414

复位值：0x0000 0000

当通道上有事件发生时，主机全体通道中断寄存器会使用模块中断寄存器中的主机通道中断位（OTG_GINTSTS 中的 HCINT 位）中断应用程序。相关内容如*图 587* 所示。每个通道对应 1 个中断位，最多有 16 个位。当应用程序通过相应主机通道 x 中断寄存器清零中断时，该寄存器中的位也会清零。

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
HAINT[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 **HAINT[15:0]**：通道中断 (Channel interrupts)

每个通道一位：通道 0 对应位 0，通道 15 对应位 15

50.15.25 OTG 主机全体通道中断屏蔽寄存器 (OTG_HAINTMSK)

OTG host all channels interrupt mask register

偏移地址：0x418

复位值：0x0000 0000

主机全体通道中断屏蔽寄存器与主机全体通道中断寄存器结合使用，进而在通道上发生事件时中断应用程序。每个通道对应 1 个中断屏蔽位，最多有 16 个位。

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
HAINTM[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 **HAINTM[15:0]**：通道中断屏蔽 (Channel interrupt mask)

0：屏蔽中断

1：使能中断

每个通道一位：通道 0 对应位 0，通道 15 对应位 15

50.15.26 OTG 主机端口控制和状态寄存器 (OTG_HPRT)

OTG host port control and status register

偏移地址：0x440

复位值：0x0000 0000

该寄存器仅在主机模式下可用。当前，OTG 主机仅支持一个端口。

该寄存器包含与 USB 端口相关的信息，例如 USB 复位、使能、挂起、恢复、连接状态以及测试模式。相关内容在*图 587* 中进行了说明。该寄存器中的 rc_w1 位可通过模块中断寄存器中的主机端口中断位（OTG_GINTSTS 中的 HPRTINT 位）触发应用程序中断。发生端口中断时，应用程序必须读取该寄存器，并将引起中断的位清零。对于 rc_w1 位，应用程序必须向该位写入 1 以清除该中断。

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.	PSPD[1:0]		PTCTL[3]
													r	r	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PTCTL[2:0]			PPWR	PLSTS[1:0]		Res.	PRST	PSUSP	PRES	POC CHNG	POCA	PEN CHNG	PENA	PCDET	PCSTS
rw	rw	rw	rw	r	r		rw	rs	rw	rc_w1	r	rc_w1	rc_w1	rc_w1	r

位 31:19 保留，必须保持复位值。

位 18:17 **PSPD[1:0]**：端口速度 (Port speed)
指示连接到该端口的设备的速度。
01：全速
10：低速
11：保留

位 16:13 **PTCTL[3:0]**：端口测试控制 (Port test control)
应用程序向该字段写入一个非零值，以将端口置于测试模式，同时端口上会产生对应模式的信号。
0000：禁止测试模式
0001：Test_J 模式
0010：Test_K 模式
0011：Test_SE0_NAK 模式
0100：Test_Packet 模式
0101：Test_Force_Enable
其他值：保留

位 12 **PPWR**：端口电源 (Port power)
应用程序使用该字段控制该端口的电源，且发生过流情况时，模块会将该位清零。
0：掉电
1：上电

位 11:10 **PLSTS[1:0]**：端口线状态 (Port line status)
指示 USB 数据线的当前逻辑电平
位 10：OTG_DP 的逻辑电平
位 11：OTG_DM 的逻辑电平

位 9 保留，必须保持复位值。

位 8 **PRST**：端口复位 (Port reset)

应用程序将该位置 1 时，会在该端口上启动复位序列。应用程序必须为复位周期定时，并在复位序列完成后将该位清零。

0：端口未处于复位状态

1：端口处于复位状态

应用程序必须将该位置 1 并最少保持 10 ms，以在端口上启动复位。除所需的最少持续时间之外，在将该位清零前，应用程序可将该位置 1 的状态再保持 10 ms，即使 USB 标准并没有设置最大限制。

高速：50 ms

全速/低速：10 ms

位 7 **PSUSP**：端口挂起 (Port suspend)

应用程序将此位置 1 以将此端口置于挂起模式。只有此位置 1 时，模块才会停止发送 SOF。要停止 PHY 时钟，应用程序必须将端口时钟停止位置 1，这会使能 PHY 的挂起输入引脚。

此位的读取值反映该端口的当前挂起状态。检测到远程唤醒信号，或者应用程序将此寄存器中的端口复位位或端口恢复位置 1 后，模块可将此位清零；或应用程序将模块中断寄存器中的检测到恢复/远程唤醒中断位或检测到断开连接中断位（分别为 OTG_GINTSTS 中的 WKUPINT 或 DISCINT）置 1，模块也可将此位清零。

0：端口未处于挂起模式

1：端口处于挂起模式

位 6 **PRES**：端口恢复 (Port resume)

应用程序将此位置 1 以在该端口上驱动恢复信号。模块会持续驱动恢复信号直到应用程序将此位清零。

如模块中断寄存器中的检测到端口恢复/远程唤醒中断位（OTG_GINTSTS 中的 WKUPINT 位）指示，如果模块检测到 USB 远程唤醒序列，则开始驱动恢复信号，而无需应用程序进行干预；如果模块检测到断开连接的情况，则将此位清零。此位的读取值指示当前模块是否正在驱动恢复信号。

0：不驱动恢复信号

1：驱动恢复信号

当 LPM 已使能且模块处于 L1 状态时，此位的影响如下：

1. 应用程序将此位置 1 以在该端口上驱动恢复信号。

2. 模块继续驱动恢复信号，直至达到 OTG_GLPMCFG 寄存器的 BESLTHRS[3:0] 字段预定的时间。

3. 如模块中断寄存器中的“检测到端口 L1 恢复/远程 L1 唤醒中断”位（OTG_GINTSTS 中的 WKUPINT 位）指示，如果模块检测到 USB 远程唤醒序列，则开始驱动恢复信号，而无需应用程序进行干预，并在恢复结束时，将此位清零。此位可以由模块和应用程序置 1 和清零。即使主机未连接任何设备，此位也可由模块清零。

位 5 **POCCHNG**：端口过流变化 (Port overcurrent change)

该寄存器中端口过流激活位（位 4）状态发生变化时，模块将此位置 1。

位 4 **POCA**：端口过流激活 (Port overcurrent active)

此位指示端口的过流状况。

0：无过流状况

1：有过流状况

位 3 **PENCHNG**：端口使能/禁止变化 (Port enable/disable change)

该寄存器中的端口使能位 2 的状态发生变化时，模块将此位置 1。

位 2 **PENA**：端口使能 (Port enable)

端口执行复位序列后，只能由模块使能，并且可以由过流状况、断开连接状况或应用程序将此位清零来禁止。应用程序无法通过对寄存器执行写操作将此位置 1。只能将此位清零来禁止端口。对此位的操作不会触发应用程序的任何中断。

0：禁止端口

1：使能端口

位 1 **PCDET**：检测到端口连接 (Port connect detected)

当检测到设备连接时，模块将此位置 1，以使用模块中断寄存器中的主机端口中断位（OTG_GINTSTS 中的 HPRTINT 位）触发应用程序的中断。应用程序必须将此位置 1 才可清除该中断。

位 0 **PCSTS**：端口连接状态 (Port connect status)

0：端口未连接设备

1：端口已连接设备

50.15.27 OTG 主机通道 x 特性寄存器 (OTG_HCCHARx)（x = 0..11，其中 x 表示通道编号）

OTG host channel x characteristics register

偏移地址：0x500 + (x * 0x20)

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CHENA	CHDIS	ODD FRM	DAD[6:0]							MCNT[1:0]		EPTYP[1:0]		LSDEV	Res.
rs	rs	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
EPDIR	EPNUM[3:0]				MPSIZ[10:0]										
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31 **CHENA**：通道使能 (Channel enable)

此字段由应用程序软件置 1，并由 OTG 主机硬件清零。

0：禁止通道

1：使能通道

位 30 **CHDIS**：通道禁止 (Channel disable)

应用程序将此位置 1 以停止通过通道发送/接收数据，即使通过该通道的传输还未完成，停止操作仍然生效。应用程序必须等待禁止通道的中断以确认通道已经被禁止。

位 29 **ODDFRM**：奇数帧 (Odd frame)

此字段由应用程序置位或复位，以分别指示 OTG 主机必须传输奇数帧或偶数帧。此字段只适用于周期性（同步和中断）事务。

0：偶数帧

1：奇数帧

位 28:22 **DAD[6:0]**：设备地址 (Device Address)

此字段用于指定要与该主机通信的特定设备。

位 21:20 **MCNT[1:0]**：多重计数 (Multicount)

此字段向主机指示该周期性端点每帧必须执行的事务数。该字段不用于非周期性传输。

00：保留。对该字段的操作会产生不明确的结果

01：1 个事务

10：该端点每帧需要发出 2 个事务

11：该端点每帧需要发出 3 个事务

注： 此字段至少须置为 01。

位 19:18 **EPTYP[1:0]**：端点类型 (Endpoint type)

指示选择的传输类型。

00：控制

01：同步

10：批量

11：中断

位 17 **LSDEV**：低速设备 (Low-speed device)

此字段由应用程序置 1，表示此通道正在与一个低速设备进行通信。

位 16 保留，必须保持复位值。

位 15 **EPDIR**：端点方向 (Endpoint direction)

指示通信事务的方向是输入还是输出。

0：OUT

1：IN

位 14:11 **EPNUM[3:0]**：端点编号 (Endpoint number)

指示要与该主机通道通信的 USB 设备的端点号。

位 10:0 **MPSIZ[10:0]**：最大数据包大小 (Maximum packet size)

指示与该主机通道通信的设备端点的最大数据包大小。

50.15.28 OTG 主机通道 x 中断寄存器 (OTG_HCINTx)（x = 0..11，其中 x 表示通道编号）

OTG host channel x interrupt register

偏移地址：0x508 + (x * 0x20)

复位值：0x0000 0000

该寄存器指示在出现 USB 和 AHB 相关事件时通道的状态。相关内容在*图 587* 中进行了说明。当模块中断寄存器中的主机通道中断位（OTG_GINTSTS 中的 HCINT 位）置 1 时，应用程序必须读取该寄存器。在对寄存器执行读操作之前，应用程序必须先读取主机全体通道中断 (OTG_HAINT) 寄存器，以获取主机通道 x 中断寄存器的准确通道编号。应用程序必须将该寄存器中的相应位清零，才能将 OTG_HAINT 和 OTG_GINTSTS 寄存器中的对应位清零。

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.	Res.	Res.	Res.	DTERR	FRMOR	BBERR	TXERR	Res.	ACK	NAK	STALL	Res.	CHH	XFRC
					rc_w1	rc_w1	rc_w1	rc_w1		rc_w1	rc_w1	rc_w1		rc_w1	rc_w1

位 31:11 保留，必须保持复位值。

位 10 **DTERR**：数据翻转错误 (Data toggle error)。

位 9 **FRMOR**：帧溢出 (Frame overrun)。

位 8 **BBERR**：串扰错误 (Babble error)。

位 7 **TXERR**：通信事务错误 (Transaction error)。
指示 USB 上发生下列错误之一：
CRC 校验失败
超时
位填充错误
错误的 EOP

位 6 保留，必须保持复位值。

位 5 **ACK**：收到/发出 ACK 响应中断 (ACK response received/transmitted interrupt)。

位 4 **NAK**：收到 NAK 响应中断 (NAK response received interrupt)。

位 3 **STALL**：收到 STALL 响应中断 (STALL response received interrupt)。

位 2 保留，必须保持复位值。

位 1 **CHH**：通道停止 (Channel halted)。
指示因任意 USB 事务错误或为响应应用程序的禁止请求而导致传输非正常结束。

位 0 **XFRC**：传输完成 (Transfer completed)。
未出现任何错误，正常完成传输。

50.15.29 OTG 主机通道 x 中断屏蔽寄存器 (OTG_HCINTMSKx)（x = 0..11，其中 x 表示通道编号）

OTG host channel x interrupt mask register

偏移地址：0x50C + (x * 0x20)

复位值：0x0000 0000

此寄存器反映了先前部分中介绍的各通道状态的屏蔽情况。

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.	Res.	Res.	Res.	DTERRM	FRMORM	BBERRM	TXERRM	Res.	ACKM	NAKM	STALLM	Res.	CHHM	XFRCM
					rw	rw	rw	rw		rw	rw	rw		rw	rw

位 31:11 保留，必须保持复位值。

位 10 **DTERRM**：数据翻转错误屏蔽 (Data toggle error mask)。
0：屏蔽中断
1：使能中断

位 9 **FRMORM**：帧溢出屏蔽 (Frame overrun mask)。
0：屏蔽中断
1：使能中断

位 8 **BBERRM**：串扰错误屏蔽 (Babble error mask)。
0：屏蔽中断
1：使能中断

位 7 **TXERRM**：通信事务错误屏蔽 (Transaction error mask)。
0：屏蔽中断
1：使能中断

位 6 保留，必须保持复位值。

位 5 **ACKM**：接收/发送 ACK 响应中断屏蔽 (ACK response received/transmitted interrupt mask)。
0：屏蔽中断
1：使能中断

位 4 **NAKM**：NAK 响应接收中断屏蔽 (NAK response received interrupt mask)。
0：屏蔽中断
1：使能中断

位 3 **STALLM**：STALL 响应接收中断屏蔽 (STALL response received interrupt mask)。
0：屏蔽中断
1：使能中断

位 2 保留，必须保持复位值。

位 1 **CHHM**：通道停止中断屏蔽 (Channel halted mask)。
0：屏蔽中断
1：使能中断

位 0 **XFRCM**：传输完成中断屏蔽 (Transfer completed mask)。
0：屏蔽中断
1：使能中断

50.15.30 OTG 主机通道 x 传输大小寄存器 (OTG_HCTSIZx)（x = 0..11，其中 x 表示通道编号）

OTG host channel x transfer size register

偏移地址：0x510 + (x * 0x20)

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	DPID[1:0]		PKTCNT[9:0]										XFRSIZ[18:16]		
	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
XFRSIZ[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31 保留，必须保持复位值。

位 30:29 **DPID[1:0]**：数据 PID (Data PID)

应用程序在此字段设置数据通信的初始同步 PID。后续传输的数据 PID 由主机硬件维护。

00：DATA0

10：DATA1

11：SETUP（控制传输）/保留（非控制传输）

位 28:19 **PKTCNT[9:0]**：数据包计数 (Packet count)

应用程序在此字段中设置将要发送 (OUT) 或接收 (IN) 的数据包数。

主机每成功发送或接收一个 OUT/IN 数据包便递减一次计数值。此值达到 0 后，将中断应用程序来指示操作正常完成。

位 18:0 **XFRSIZ[18:0]**：传输大小 (Transfer size)

对于 OUT 操作，此字段为传输期间主机发送的数据字节数。

对于 IN 操作，此字段为应用程序保留给传输的缓冲区大小。对于 IN 事务（周期性和非周期性），应用程序会将此字段编程为最大数据包大小的整数倍。

50.15.31 设备模式寄存器

模块每次切换到设备模式时都必须对这些寄存器进行编程。

50.15.32 OTG 设备配置寄存器 (OTG_DCFG)

OTG device configuration register

偏移地址：0x800

复位值：0x0220 0000

此寄存器在上电、执行某些控制命令或枚举后，会将模块配置为设备模式。请勿在初始编程后更改该寄存器。

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
ERRATIM	Res.	Res.	PFIVL[1:0]		DAD[6:0]							Res.	NZLSOHSK	DSPD[1:0]	
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 **ERRATIM**：不定错误中断屏蔽 (Erratic error interrupt mask)

1：发送不定错误时不触发早期挂起中断

0：发生不定错误时生成早期挂起中断

位 14 保留，必须保持复位值。

位 13 保留，必须保持复位值。

位 12:11 **PFIVL[1:0]**：周期性帧间隔 (Periodic frame interval)

指示一帧内必须使用周期性帧中断通知应用程序的时间点。此功能可用于确定该帧的所有同步通信是否完成。

00：80% 帧间隔

01：85% 帧间隔

10：90% 帧间隔

11：95% 帧间隔

位 10:4 **DAD[6:0]**：设备地址 (Device Address)

应用程序必须在执行每个 SetAddress 控制命令后根据命令参数对该字段进行设置。

位 3 保留，必须保持复位值。

位 2 **NZLSOHSK**：非零长度状态 OUT 握手信号 (Non-zero-length status OUT handshake)

在控制传输状态阶段的 OUT 事务期间，当模块收到非零长度数据包后，应用程序可以使用此字段选择要发送的握手信号。

1：收到非零长度状态 OUT 事务时，回复 STALL 握手信号，收到的 OUT 数据包不发送给应用程序。

0：将收到的 OUT 数据包（零长度或非零长度）发送给应用程序，并基于设备端点控制寄存器中端点的 NAK 和 STALL 位发送握手信号。

位 1:0 **DSPD[1:0]**：设备速度 (Device speed)

指示应用程序要求模块进行枚举所采用的速度，或应用程序支持的最大速度。但是，实际总线速度只有在完成 chirp 序列后才能确定，同时此速度基于与模块连接的 USB 主机的速度。

00：保留

01：保留

10：保留

11：全速（USB 1.1 收发器时钟为 48 MHz）

50.15.33 OTG 设备控制寄存器 (OTG_DCTL)

OTG device control register

偏移地址：0x804

复位值：0x0000 0002

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.	DS BESL RJCT	Res.	Res.
													rw		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	PO PRG DNE	CGO NAK	SGO NAK	CGI NAK	SGI NAK	TCTL[2:0]			GON STS	GIN STS	SDIS	RWU SIG
				rw	w	w	w	w	rw	rw	rw	r	r	rw	rw

位 31:19 保留，必须保持复位值。

位 18 **DSBESLRJCT**：深度睡眠 BESL 拒绝 (Deep sleep BESL reject)

模块拒绝 BESL 值大于编程的 BESL 阈值的 LPM 请求。将为 BESL 值大于 BESL 阈值的 LPM 令牌发送 NYET 响应。默认情况下，禁用深度睡眠 BESL 拒绝功能。

位 17:12 保留，必须保持复位值。

位 11 **POPRGDNE**：上电编程完成 (Power-on programming done)
应用程序使用此位指示寄存器从掉电模式唤醒后已完成编程。

位 10 **CGONAK**：将全局 OUT NAK 清零 (Clear global OUT NAK)
对此字段写入 1 会将全局 OUT NAK 清零。

位 9 **SGONAK**：将全局 OUT NAK 置 1 (Set global OUT NAK)
对此字段写入 1 会将全局 OUT NAK 置 1。
应用程序使用此位在所有 OUT 端点发送 NAK 握手信号。
应用程序只有确定模块中断寄存器中的全局 OUT NAK 有效位（OTG_GINTSTS 中 GONAKEFF 位）已清零时，才可以将此位置 1。

位 8 **CGINAK**：将全局 IN NAK 清零 (Clear global IN NAK)
对此字段写入 1 会将全局 IN NAK 清零。

位 7 **SGINAK**：将全局 IN NAK 置 1 (Set global IN NAK)
对此字段写入 1 会将全局非周期性 IN NAK 置 1。应用程序使用此位使所有非周期性 IN 端点发送 NAK 握手信号。
应用程序只有确定模块中断寄存器中的全局 IN NAK 有效位（OTG_GINTSTS 中的 GINAKEFF 位）已清零时，才可以将此位置 1。

位 6:4 **TCTL[2:0]**：测试控制 (Test control)
000：禁止测试模式
001：Test_J 模式
010：Test_K 模式
011：Test_SE0_NAK 模式
100：Test_Packet 模式
101：Test_Force_Enable
其他值：保留

位 3 **GONSTS**：全局 OUT NAK 状态 (Global OUT NAK status)
0：根据 FIFO 状态和 NAK 和 STALL 位设置发送握手信号。
1：无论 Rx FIFO 中是否还有空闲空间都不接收数据。除 SETUP 事务之外，对所有收到的数据包回复 NAK 握手信号。所有同步类型的 OUT 数据包都将被丢弃。

位 2 **GINSTS**：全局 IN NAK 状态 (Global IN NAK status)
0：根据发送 FIFO 中是否有待发送的数据回复握手信号。
1：使所有非周期性 IN 端点回复 NAK 握手信号，无需考虑发送 FIFO 中是否有待发送的数据。

位 1 **SDIS**：软断开 (Soft disconnect)
应用程序使用该位向 USB OTG 模块发出执行软断开的信号。该位置 1 时，主机不会看到设备已连接，且该设备也不会接收 USB 上的信号。在应用程序将此位清零之前，模块会保持断开状态。
0：正常工作。此位在软断开之后清零，会使主机收到设备已连接的事件。重新连接设备之后，USB 主机会重新启动设备枚举。
1：模块使 USB 主机收到设备断开连接的事件。

位 0 **RWUSIG**：发送远程唤醒信号 (Remote wakeup signaling)
应用程序将此位置 1 时，模块会启动远程发送信号，以唤醒 USB 主机。应用程序必须将此位置 1 以使模块退出挂起状态。根据 USB 2.0 规范，应用程序必须在将此位置 1 之后的 1 ms 到 15 ms 内将其清零。
如果 LPM 已使能并且模块处于 L1（睡眠）状态，则当应用程序将此位置 1 时，模块会启动 L1 远程信号来唤醒 USB 主机。应用程序必须将此位置 1 以使模块退出睡眠状态。按照 LPM 规范中的规定，在应用程序将此位置 1 后经过 50 µs ($T_{L1DevDrvResume}$)，硬件会自动将此位清零。当前一个 LPM 事务中的 bRemoteWake 为零时（请参见 GLPMCFG 寄存器中的 REMWAKE 位），应用程序不得将此位置 1。

表 326 显示了为使 USB 主机检测到设备断开连接所需的软断开 (SDIS) 位置 1 的最短时间（取决于设备状态）。为了协调时钟抖动，建议应用程序在指定的最小时间基础上再加入一段延迟。

表 326. 软断开的最小时间

运行速度	设备状态	最小时间
全速	挂起	1 ms + 2.5 µs
全速	空闲	2.5 µs
全速	非空闲或挂起（正在通信时）	2.5 µs

50.15.34 OTG 设备状态寄存器 (OTG_DSTS)

OTG device status register

偏移地址：0x808

复位值：0x0000 0010

此寄存器指示模块在出现 USB 相关事件时的状态。必须在发生设备全体中断寄存器 (OTG_DAINT) 中断时读取它。

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

位 31:24 保留，必须保持复位值。

位 23:22 **DEVLNSTS[1:0]**：设备线状态 (Device line status)
指示 USB 数据线的当前逻辑电平。
位 [23]：D+ 的逻辑电平
位 [22]：D- 的逻辑电平

位 21:8 **FNSOF[13:0]**：接收 SOF 的帧编号 (Frame number of the received SOF)

位 7:4 保留，必须保持复位值。

位 3 **EERR**：不定错误 (Erratic error)

模块将该位置 1 以报告任何不定错误。

由于不定错误，OTG_FS 控制器会进入挂起状态，并且会以 OTG_GINTSTS 寄存器的早期挂起位（OTG_GINTSTS 中的 ESUSP 位）为应用程序生成一个中断。如果早期挂起中断是由不定错误触发，则应用程序只能执行软断开以恢复通信。

位 2:1 **ENUMSPD[1:0]**：枚举速度 (Enumerated speed)

指示 OTG_FS 控制器通过 chirp 序列检测速度后被枚举成的速度。

01：保留

10：保留

11：全速（PHY 时钟运行频率为 48 MHz）

其他值：保留

位 0 **SUSPSTS**：挂起状态 (Suspend status)

在设备模式下，只要在 USB 上检测到挂起状态，该位就会置 1。当 USB 数据线上的空闲状态保持 3 ms，模块便会进入挂起状态。出现以下情况时，模块会退出挂起状态：

– USB 数据线上有活动

– 应用程序对 OTG_DCTL 寄存器的远程唤醒信号位（OTG_DCTL 中的 RWUSIG 位）执行写操作。

50.15.35 OTG 设备 IN 端点通用中断屏蔽寄存器 (OTG_DIEPMSK)

OTG device IN endpoint common interrupt mask register

偏移地址：0x810

复位值：0x0000 0000

此寄存器与全体端点的各个 OTG_DIEPINTx 寄存器配合使用，以便在每个 IN 端点上生成中断。通过对此寄存器的相应位执行写操作，可屏蔽 OTG_DIEPINTx 寄存器中特定状态的 IN 端点中断。默认情况下，状态中断都被屏蔽。

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.	NAKM	Res.	Res.	Res.	Res.	TXFURM	Res.	INEPNEM	INEPNMM	ITTXFEMSK	TOM	Res.	EPDM	XFRCM
		rw					rw		rw	rw	rw	rw		rw	rw

位 31:14 保留，必须保持复位值。

位 13 **NAKM**：NAK 中断屏蔽 (NAK interrupt mask)

0：屏蔽中断

1：使能中断

位 12:10 保留，必须保持复位值。

位 9 保留，必须保持复位值。

位 8 **TXFURM**：FIFO 下溢中断屏蔽 (FIFO underrun mask)

0：屏蔽中断

1：使能中断

位 7 保留，必须保持复位值。

位 6 **INEPNEM**：IN 端点 NAK 有效中断屏蔽 (IN endpoint NAK effective mask)
0：屏蔽中断
1：使能中断

位 5 **INEPNMM**：EP 不匹配时接收到 IN 令牌中断屏蔽 (IN token received with EP mismatch mask)
0：屏蔽中断
1：使能中断

位 4 **ITTXFEMSK**：Tx FIFO 为空时接收到 IN 令牌中断屏蔽 (IN token received when Tx FIFO empty mask)
0：屏蔽中断
1：使能中断

位 3 **TOM**：超时中断屏蔽（非同步端点）(Timeout condition mask (Non-isochronous endpoints))
0：屏蔽中断
1：使能中断

位 2 保留，必须保持复位值。

位 1 **EPDM**：端点禁止中断屏蔽 (Endpoint disabled interrupt mask)
0：屏蔽中断
1：使能中断

位 0 **XFRCM**：传输完成中断屏蔽 (Transfer completed interrupt mask)
0：屏蔽中断
1：使能中断

50.15.36 OTG 设备 OUT 端点通用中断屏蔽寄存器 (OTG_DOEPMSK)

OTG device OUT endpoint common interrupt mask register

偏移地址：0x814

复位值：0x0000 0000

此寄存器与所有端点的各个 OTG_DOEPINTx 寄存器配合使用，以便可以在每个 OUT 端点上生成中断。通过对此寄存器的相应位执行写操作，可屏蔽 OTG_DOEPINTx 寄存器中特定状态的 OUT 端点中断。默认情况下，状态中断都被屏蔽。

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.	NYET MSK	NAK MSK	BERR M	Res.	Res.	Res.	OUT PKT ERRM	Res.	Res.	STS PHSR XM	OTEPD M	STUPM	Res.	EPDM	XFRC M
	rw	rw	rw				rw			rw	rw	rw		rw	rw

位 31:15 保留，必须保持复位值。

位 14 **NYETMSK**：NYET 中断屏蔽 (NYET interrupt mask)
0：屏蔽中断
1：使能中断

位 13 **NAKMSK**：NAK 中断屏蔽 (NAK interrupt mask)
0：屏蔽中断
1：使能中断

位 12 **BERRM**：Babble 错误中断屏蔽 (Babble error interrupt mask)
0：屏蔽中断
1：使能中断

位 11:10 保留，必须保持复位值。

位 9 保留，必须保持复位值。

位 8 **OUTPKTERRM**：输出数据包错误屏蔽 (Out packet error mask)
0：屏蔽中断
1：使能中断

位 7 保留，必须保持复位值。

位 6 保留，必须保持复位值。

位 5 **STSPHSRXM**：接收到用于控制写的状态阶段中断屏蔽 (Status phase received for control write mask)
0：屏蔽中断
1：使能中断

位 4 **OTEPDM**：端点禁止时接收到 OUT 令牌中断屏蔽 (OUT token received when endpoint disabled mask)。仅适用于控制 OUT 端点。
0：屏蔽中断
1：使能中断

位 3 **STUPM**：SETUP 阶段完成中断屏蔽 (SETUP phase done mask)。仅适用于控制端点。
0：屏蔽中断
1：使能中断

位 2 保留，必须保持复位值。

位 1 **EPDM**：端点禁止中断屏蔽 (Endpoint disabled interrupt mask)
0：屏蔽中断
1：使能中断

位 0 **XFRCM**：传输完成中断屏蔽 (Transfer completed interrupt mask)
0：屏蔽中断
1：使能中断

50.15.37 OTG 设备全体端点中断寄存器 (OTG_DAINT)

OTG device all endpoints interrupt register

偏移地址：0x818

复位值：0x0000 0000

当端点上发生有效事件时，OTG_DAINT 寄存器将通过 OTG_GINTSTS 寄存器中的设备 OUT 端点中断位或设备 IN 端点中断位（分别为 OTG_GINTSTS 中的 OEPINT 或 IEPINT 位）来中断应用程序。每个端点对应一个中断位，OUT 端点和 IN 端点均最多有 16 个中断位。双向端点将使用相应的 IN 和 OUT 中断位。当应用程序将相应设备端点 x 中断寄存器 (OTG_DIEPINTx/OTG_DOEPINTx) 中的位置 1 和清零时，此寄存器中的相应位也将置 1 和清零。

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

位 31:16 **OEPINT[15:0]**：OUT 端点中断位 (OUT endpoint interrupt bits)
每个 OUT 端点对应一位：
OUT 端点 0 对应位 16，而 OUT 端点 3 对应位 19。

位 15:0 **IEPINT[15:0]**：IN 端点中断位 (IN endpoint interrupt bits)
每个 IN 端点对应一位：
IN 端点 0 对应位 0，而 IN 端点 3 对应位 3。

50.15.38 OTG 全体端点中断屏蔽寄存器 (OTG_DAINTMSK)

OTG all endpoints interrupt mask register

偏移地址：0x81C

复位值：0x0000 0000

OTG_DAINTMSK 寄存器与设备端点中断寄存器结合使用，在设备端点上发生事件时中断应用程序。但是，与该中断相对应的 OTG_DAINT 寄存器位仍会置 1。

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

位 31:16 **OEPM[15:0]**：OUT EP 中断屏蔽位 (OUT EP interrupt mask bits)
每个 OUT 端点对应一位：
OUT EP 0 对应位 16，而 OUT EP 3 对应位 19
0：屏蔽中断
1：使能中断

位 15:0 **IEPM[15:0]**：IN EP 中断屏蔽位 (IN EP interrupt mask bits)
每个 IN 端点对应一位：
IN EP 0 对应位 0，而 IN EP 3 对应位 3
0：屏蔽中断
1：使能中断

50.15.39 OTG 设备 V_{BUS} 放电时间寄存器 (OTG_DVBUSDIS)

OTG device V_{BUS} discharge time register

偏移地址：0x0828

复位值：0x0000 17D7

该寄存器指定 SRP 期间 V_{BUS} 发出脉冲之后的 V_{BUS} 放电时间。

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
VBUSDT[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 **VBUSDT[15:0]**：设备 V_{BUS} 放电时间 (Device V_{BUS} discharge time)
指定 SRP 期间 V_{BUS} 发出脉冲之后的 V_{BUS} 放电时间。该时间值等于：
V_{BUS} 放电时间（PHY 时钟数）/1024
该值可基于不同的 V_{BUS} 负载根据需要进行调整。

50.15.40 OTG 设备 V_{BUS} 脉冲时间寄存器 (OTG_DVBUSPULSE)

OTG device V_{BUS} pulsing time register

偏移地址：0x082C

复位值：0x0000 05B8

该寄存器指定 SRP 期间的 V_{BUS} 脉冲时间。

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
DVBUSP[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 **DVBUSP[15:0]**：设备 V_{BUS} 脉冲时间 (Device V_{BUS} pulsing time)。该功能仅与 OTG1.3 相关。
指定 SRP 期间的 V_{BUS} 脉冲时间。
V_{BUS} 脉冲时间（PHY 时钟数）/1024

50.15.41 OTG 设备 IN 端点 FIFO 空中断屏蔽寄存器 (OTG_DIEPEMPMSK)

OTG device IN endpoint FIFO empty interrupt mask register

偏移地址：0x834

复位值：0x0000 0000

此寄存器用于控制 IN 端点 FIFO 空中断的生成 (TXFE_OTG_DIEPINTx)。

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
INEPTXFEM[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 **INEPTXFEM[15:0]**：IN EP Tx FIFO 空中断屏蔽位 (IN EP Tx FIFO empty interrupt mask bits)
这些位用作 OTG_DIEPINTx 的屏蔽位。
每个位对应一个 IN 端点的 TXFE 中断：
IN 端点 0 对应位 0，而 IN 端点 3 对应位 3
0：屏蔽中断
1：使能中断

50.15.42 OTG 设备控制 IN 端点 0 控制寄存器 (OTG_DIEPCTL0)

OTG device control IN endpoint 0 control register

偏移地址：0x900

复位值：0x0000 0000

本节介绍 USB_OTG FS 的 OTG_DIEPCTL0 寄存器。非零控制端点使用端点 1-3 的寄存器。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
EPENA	EPDIS	Res.	Res.	SNAK	CNAK	TXFNUM[3:0]				STALL	Res.	EPTYP		NAK STS	Res.
rs	rs			w	w	rw	rw	rw	rw	rs		r	r	r	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
USBA EP	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MPSIZ[1:0]	
r														rw	rw

位 31 **EPENA**：端点使能 (Endpoint enable)

应用程序将此位置 1 以在端点 0 上启动数据发送。

在此端点上触发以下任一中断之前，模块会将此位清零：

– 端点禁止

– 传输完成

位 30 **EPDIS**：端点禁止 (Endpoint disable)

即使在该端点上的传输完成之前，应用程序也可将此位置 1，以停止端点上的数据发送。应用程序必须等到发生端点禁止中断后，才能将端点视为禁止端点。在端点禁止中断位置 1 前，模块会将此位清零。只有在该端点的端点使能位置 1 后，应用程序才可将该位置 1。

位 29:28 保留，必须保持复位值。

位 27 **SNAK**：将 NAK 置 1 (Set NAK)

对此位进行写操作会将端点的 NAK 位置 1。

通过此位，应用程序可以控制端点上 NAK 握手信号的发送。模块也可在端点接收到 SETUP 数据包后将该端点的此位置 1。

位 26 **CNAK**：将 NAK 清零 (Clear NAK)

对此位进行写操作会将端点的 NAK 位清零。

位 25:22 **TXFNUM[3:0]**：Tx FIFO 编号 (Tx FIFO number)

该值设置为分配给 IN 端点 0 的 FIFO 编号。

位 21 **STALL**：STALL 握手 (STALL handshake)

应用程序只能将此位置 1，端点接收到 SETUP 令牌时，模块会将此位清零。如果 NAK 位、全局 IN NAK 或全局 OUT NAK 与此位均置 1，则 STALL 位优先。

位 20 保留，必须保持复位值。

位 19:18 **EPTYP**：端点类型 (Endpoint type)

硬件设置为‘00’，表示控制类型的端点。

位 17 **NAKSTS**：NAK 状态 (NAK status)

指示以下结果：

0：模块根据 FIFO 状态回复非 NAK 握手。

1：模块在此端点上回复 NAK 握手。

当应用程序或模块将此位置 1 时，即使 Tx FIFO 中仍有数据可用，模块也会停止发送数据。无论此位如何设置，模块总是通过 ACK 握手响应 SETUP 数据包。

位 16 保留，必须保持复位值。

位 15 **USBAEP**：USB 活动端点 (USB active endpoint)
此位总是置 1，指示在所有配置和接口中控制端点 0 始终处于激活状态。

位 14:2 保留，必须保持复位值。

位 1:0 **MPSIZ[1:0]**：最大数据包大小 (Maximum packet size)
应用程序必须将此字段编程为当前逻辑端点的最大数据包大小。
00：64 字节
01：32 字节
10：16 字节
11：8 字节

50.15.43 OTG 设备 IN 端点 x 控制寄存器 (OTG_DIEPCTLx)（x = 1..5，其中 x 表示端点编号）

OTG device IN endpoint x control register

偏移地址：0x900 + (x * 0x20)

复位值：0x0000 0000

应用程序使用此寄存器控制各个逻辑端点（端点 0 除外）的行为。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
EPENA	EPDIS	SODD FRM	SD0 PID/ SEVN FRM	SNAK	CNAK	TXFNUM[3:0]				STALL	Res.	EPTYP[1:0]		NAK STS	EO NUM/ DPID
rs	rs	w	w	w	w	rw	rw	rw	rw	rw/rs		rw	rw	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
USBA EP	Res.	Res.	Res.	Res.	MPSIZ[10:0]										
rw					rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31 **EPENA**：端点使能 (Endpoint enable)
应用程序将此位置 1 以在端点上启动数据发送。
在此端点上触发以下任一中断之前，模块会将此位清零：
– SETUP 阶段完成 (SETUP phase done)
– 端点禁止
– 传输完成

位 30 **EPDIS**：端点禁止 (Endpoint disable)
即使在该端点上的传送完成之前，应用程序也可将此位置 1，以停止端点上的数据发送/接收。应用程序必须等到发生端点禁止中断后，才能将端点视为禁止端点。在端点禁止中断位置 1 前，模块会将此位清零。只有在该端点的端点使能位置 1 后，应用程序才可将该位置 1。

位 29 **SODDFRM**：设置奇数帧 (Set odd frame)
仅适用于同步 IN 和 OUT 端点。
对此字段进行写操作会将偶数/奇数帧 (EONUM) 字段设置为奇数帧。

位 28 **SD0PID**：设置 DATA0 PID (Set DATA0 PID)

仅适用于中断/批量 IN 端点。

对此字段进行写操作会将此寄存器中的端点数据 PID (DPID) 字段设置为 DATA0。

SEVNFRM：设置偶数帧 (Set even frame)

仅适用于同步 IN 端点。

对此字段进行写操作会将偶数/奇数帧 (EONUM) 字段设置为偶数帧。

位 27 **SNAK**：将 NAK 置 1 (Set NAK)

对此位进行写操作会将端点的 NAK 位置 1。

通过此位，应用程序可以控制端点上 NAK 握手信号的发送。发生传输完成中断时或端点上接收到 SETUP 后，模块也可以将 OUT 端点的该位置 1。

位 26 **CNAK**：将 NAK 清零 (Clear NAK)

对此位进行写操作会将端点的 NAK 位清零。

位 25:22 **TXFNUM**：Tx FIFO 编号 (Tx FIFO number)

这些位用于指定与此端点相关联的 FIFO 编号。必须为每个有效的 IN 端点设置单独的 FIFO 编号。

此字段仅针对 IN 端点有效。

位 21 **STALL**：STALL 握手 (STALL handshake)

仅适用于非控制、非同步 IN 端点（访问类型为 rw）。

应用程序将此位置 1 使得设备对来自 USB 主机的所有令牌都回复 STALL。如果 NAK 位、全局 IN NAK 或全局 OUT NAK 与此位同时置 1，则 STALL 位优先。只有应用程序能够将此位清零，而模块则不能。

仅适用于控制端点（访问类型为 rs）

此端点接收到 SETUP 令牌时，应用程序只能将此位置 1，而模块会将其清零。如果 NAK 位、全局 IN NAK 或全局 OUT NAK 与此位同时置 1，则 STALL 位优先。无论此位如何设置，模块总是通过 ACK 握手响应 SETUP 数据包。

位 20 保留，必须保持复位值。

位 19:18 **EPTYP[1:0]**：端点类型 (Endpoint type)

以下是这个逻辑端点支持的传输类型。

00：控制

01：同步

10：批量

11：中断

位 17 **NAKSTS**：NAK 状态 (NAK status)

它指示以下结果：

0：模块根据 FIFO 状态回复非 NAK 握手。

1：模块在此端点上回复 NAK 握手。

当应用程序或模块将此位置 1 时：

对于非同步 IN 端点：即使 Tx FIFO 中存在可用数据，模块也会停止通过 IN 端点发送任何数据。

对于同步 IN 端点：即使 Tx FIFO 中存在可用数据，模块也会发送长度为零的数据包。

无论此位如何设置，模块总是通过 ACK 握手响应 SETUP 数据包。

位 16 **EONUM**：偶数/奇数帧 (Even/odd frame)

仅适用于同步 IN 端点。

指示模块为此端点发送/接收同步的数据所在的帧的编号。应用程序必须通过此寄存器中的 SEVNFRM 和 SODDFRM 字段对偶数/奇数帧编号进行编程，以便此端点发送/接收同步数据。

0：偶数帧

1：奇数帧

DPID：端点数据 PID (Endpoint data PID)

仅适用于中断/批量 IN 端点。

包含此端点上将要接收或发送的数据包的 PID。端点激活后，应用程序必须对要在此端点上接收或发送的首个数据包的 PID 进行编程。应用程序使用 SD0PID 寄存器字段对 DATA0 或 DATA1 PID 进行编程。

0：DATA0

1：DATA1

位 15 **USBAEP**：USB 活动端点 (USB active endpoint)

指示此端点在当前配置和接口中是否激活。检测到 USB 复位后，模块会为所有端点（端点 0 除外）将此位清零。接收到 SetConfiguration 和 SetInterface 命令后，应用程序必须相应地对端点寄存器进行编程并将此位置 1。

位 14:11 保留，必须保持复位值。

位 10:0 **MPSIZ[10:0]**：最大数据包大小 (Maximum packet size)

应用程序必须将此字段编程为当前逻辑端点的最大数据包大小。此值以字节为单位。

50.15.44 OTG 设备 IN 端点 x 中断寄存器 (OTG_DIEPINTx)（x = 0..5，其中 x 表示端点编号）

OTG device IN endpoint x interrupt register

偏移地址：0x908 + (x * 0x20)

复位值：0x0000 0080

此寄存器指示端点在出现 USB 和 AHB 相关事件时的状态。相关内容在*图 587* 中进行了说明。当模块中断寄存器中的 IN 端点中断位（OTG_GINTSTS 中的 IEPINT 位）置 1 时，应用程序必须读取此寄存器。在应用程序能够读取此寄存器之前，必须先读取设备全体端点中断 (OTG_DAINT) 寄存器，以获取设备端点 x 中断寄存器的准确端点编号。应用程序必须将该寄存器中的相应位清零，才能将 OTG_DAINT 和 OTG_GINTSTS 寄存器中的对应位清零。

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.	NAK	Res.	PKTD RPSTS	Res.	Res.	TXFIF OUD RN	TXFE	IN EPNE	IN EPNM	ITTXFE	TOC	Res.	EP DISD	XFRC
		rc_w1		rc_w1			rc_w1	r	r	rc_w1	rc_w1	rc_w1		rc_w1	rc_w1

位 31:14 保留，必须保持复位值。

位 13 **NAK**：NAK 输入 (NAK input)

当设备发出或收到 NAK 时，模块将生成该中断。如果是同步 IN 端点，由于 Tx FIFO 中无数据可发而发送长度为零的数据包时也会生成该中断。

位 12 保留，必须保持复位值。

位 11 **PKTDRPSTS**：数据包丢弃状态 (Packet dropped status)

该位用于向应用程序指示有 ISOC OUT 数据包被丢弃。该位没有相应的中断屏蔽位，也不会生成中断。

位 10 保留，必须保持复位值。

位 9 保留，必须保持复位值。

位 8 **TXFIFOUDRN**：发送FIFO 下溢 (TxfifoUndrn) (Transmit Fifo Underrun (TxfifoUndrn))

当模块检测到该端点的发送 FIFO 下溢时，将生成该中断。相关性：该中断仅在使能了阈值时有效

位 7 **TXFE**：发送 FIFO 为空 (Transmit FIFO empty)

当此端点的 Tx FIFO 为半空或全空时，此中断被置位。Tx FIFO 为半空还是全空状态由 OTG_GAHBCFG 寄存器中的 Tx FIFO 空门限位（OTG_GAHBCFG 中的 TXFELVL 位）决定。

位 6 **INEPNE**：IN 端点 NAK 有效 (IN endpoint NAK effective)

当应用程序通过向 OTG_DIEPCTLx 中的 CNAK 位写入数据来将 IN 端点 NAK 清零时，此位可被清零。

该中断指示模块已对（由应用程序或模块）置 1 的 NAK 采样，结果已生效。该中断指示由应用程序置 1 的 IN 端点 NAK 位已在模块中起作用。

此中断不一定表示 USB 上已发送了一个 NAK 握手信号。STALL 位优先级高于 NAK 位。

位 5 **INEPNM**：EP 不匹配时收到 IN 令牌 (IN token received with EP mismatch)

指示非周期性 TxFIFO 顶部数据所属的端点不是收到 IN 令牌的端点。此中断会在接收到 IN 令牌的端点上产生。

位 4 **ITTXFE**：Tx FIFO 为空时接收到 IN 令牌 (IN token received when Tx FIFO is empty)

指示当和该端点相对应的 Tx FIFO（周期性/非周期性）为空时，接收到 IN 令牌。此中断会在接收到 IN 令牌的端点上产生。

位 3 **TOC**：超时条件 (Timeout condition)

指示该端点对最近收到的 IN 令牌响应超时。

位 2 保留，必须保持复位值

位 1 **EPDISD**：端点禁止中断 (Endpoint disabled interrupt)

此位指示该端点已经由应用程序禁止掉。

位 0 **XFRC**：传输完成中断 (Transfer completed interrupt)

此字段指示在此端点上设置的传输已经在 USB 和 AHB 上传输完成。

50.15.45 OTG 设备 IN 端点 0 传输大小寄存器 (OTG_DIEPTSIZ0)

OTG device IN endpoint 0 transfer size register

偏移地址：0x910

复位值：0x0000 0000

在使能端点 0 之前，应用程序必须修改此寄存器。通过设备控制端点 0 控制寄存器中的端点使能位（OTG_DIEPCTL0 中的 EPENA）使能端点 0 后，模块对此寄存器进行修改。当模块将端点使能位清零后，应用程序才能读取此寄存器。

非零端点使用端点 1-3 的寄存器。

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.	PKTCNT[1:0]		Res.	Res.	Res.
											rw	rw			

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

位 31:21 保留，必须保持复位值。

位 20:19 **PKTCNT[1:0]**：数据包计数 (Packet count)

指示端点 0 的一次数据传输包含的 USB 数据包个数。

每次从 Tx FIFO 读取数据包（最大大小数据包或短数据包）时，此字段将递减。

位 18:7 保留，必须保持复位值。

位 6:0 **XFRSIZ[6:0]**：传输大小 (Transfer size)

指示端点 0 的一次数据传输包含的数据量，以字节为单位。仅当应用程序传输完这些数据后，模块才会中断该应用程序。传输大小可以设置为端点的最大数据包大小，以在每个数据包结束时中断。

每次向 Tx FIFO 写入来自外部存储器的数据包时，模块会使此字段递减。

50.15.46 OTG 设备 IN 端点发送 FIFO 状态寄存器 (OTG_DTXFSTSx)（x = 0..5，其中 x = 端点编号）

OTG device IN endpoint transmit FIFO status register

IN 端点的偏移地址：0x918 + (x * 0x20)。此只读寄存器包含设备 IN 端点 Tx FIFO 的空闲空间信息。

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
INEPTFSAV[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 **INEPTFSAV[15:0]**：IN 端点 Tx FIFO 可用空间 (IN endpoint Tx FIFO space available)
指示端点 Tx FIFO 中的可用空闲空间大小。
以 32 位字为单位：
0x0：端点 Tx FIFO 已满
0x1：1 个字可用
0x2：2 个字可用
0xn：n 个字可用
其他值：保留

50.15.47 OTG 设备 IN 端点 x 传输大小寄存器 (OTG_DIEPTSIZx)（x = 1..5，其中 x 表示端点编号）

OTG device IN endpoint x transfer size register

偏移地址：0x910 + (x * 0x20)

复位值：0x0000 0000

在使能该端点之前，应用程序必须修改此寄存器。通过 OTG_DIEPCTLx 寄存器中的端点使能位（OTG_DIEPCTLx 中的 EPENA 位）使能该端点后，模块对此寄存器进行修改。当模块将端点使能位清零后，应用程序才能读取此寄存器。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	MCNT[1:0]		PKTCNT[9:0]										XFRSIZ[18:16]		
	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
XFRSIZ[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31 保留，必须保持复位值。

位 30:29 **MCNT[1:0]**：多重计数 (Multi count)

对于周期性 IN 端点，此字段指示在 USB 上每帧必须发送的数据包数。模块使用此字段计算同步 IN 端点的数据 PID。

01：1 个数据包

10：2 个数据包

11：3 个数据包

位 28:19 **PKTCNT[9:0]**：数据包计数 (Packet count)

指示该端点上的一次数据传输包含的 USB 数据包个数。

每次从 Tx FIFO 读取数据包（最大大小数据包或短数据包）时，此字段将递减。

位 18:0 **XFRSIZ[18:0]**：传输大小 (Transfer size)

此字段包含当前端点的一次数据传输包含的数据量，以字节为单位。仅当应用程序传输完这些数据后，模块才会中断该应用程序。传输大小可以设置为端点的最大数据包大小，以在每个数据包结束时中断。

每次向 Tx FIFO 写入来自外部存储器的数据包时，模块会使此字段递减。

50.15.48 OTG 设备控制 OUT 端点 0 控制寄存器 (OTG_DOEPCTL0)

OTG device control OUT endpoint 0 control register

偏移地址：0xB00

复位值：0x0000 8000

本节介绍 OTG_DOEPCTL0 寄存器。非零控制端点使用编号 1-3 的寄存器。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
EPENA	EPDIS	Res.	Res.	SNAK	CNAK	Res.	Res.	Res.	Res.	STALL	SNPM	EPTYP[1:0]		NAK STS	Res.
w	r			w	w					rs	rw	r	r	r	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
USBA EP	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MPSIZ[1:0]	
r														r	r

位 31 **EPENA**：端点使能 (Endpoint enable)

应用程序将此位置 1 以在端点 0 上启动数据发送。

在此端点上触发以下任一中断之前，模块会将此位清零：

– SETUP 阶段完成

– 端点禁止

– 传输完成

位 30 **EPDIS**：端点禁止 (Endpoint disable)

应用程序无法禁止控制 OUT 端点 0。

位 29:28 保留，必须保持复位值。

位 27 **SNAK**：将 NAK 置 1 (Set NAK)

对此位进行写操作会将端点的 NAK 位置 1。

通过此位，应用程序可以控制端点上 NAK 握手信号的发送。发生传输完成中断时或端点上接收到 SETUP 后，模块也可以将此位置 1。

位 26 **CNAK**：将 NAK 清零 (Clear NAK)

对此位进行写操作会将端点的 NAK 位清零。

位 25:22 保留，必须保持复位值。

位 21 **STALL**：STALL 握手 (STALL handshake)

此端点接收到 SETUP 令牌时，应用程序只能将此位置 1，而模块会将其清零。如果 NAK 位、全局 OUT NAK 与此位同时置 1，则 STALL 位优先。无论此位如何设置，模块总是通过 ACK 握手响应 SETUP 数据包。

位 20 **SNPM**：监听模式 (Snoop mode)

此位用于将端点配置为监听模式。在监听模式下，模块不会在将 OUT 数据包传输到应用存储区前检查其是否正确。

位 19:18 **EPTYP[1:0]**：端点类型 (Endpoint type)

硬件固定为二进制 00，表示端点为控制传输类型。

位 17 **NAKSTS**：NAK 状态 (NAK status)

指示以下结果：

0：模块根据 FIFO 状态回复非 NAK 握手。

1：模块在此端点上回复 NAK 握手。

当应用程序或模块将此位置 1 时，即使 Rx FIFO 中存在空间可继续容纳收到的数据包，模块也会停止接收数据。无论此位如何设置，模块总是通过 ACK 握手响应 SETUP 数据包。

位 16 保留，必须保持复位值。

位 15 **USBAEP**：USB 活动端点 (USB active endpoint)

此位总是置 1，指示在所有配置和接口中控制端点 0 始终处于激活状态。

位 14:2 保留，必须保持复位值。

位 1:0 **MPSIZ[1:0]**：最大数据包大小 (Maximum packet size)

控制 OUT 端点 0 的最大数据包大小与在控制 IN 端点 0 中进行编程的值相同。

00：64 字节

01：32 字节

10：16 字节

11：8 字节

50.15.49 OTG 设备 OUT 端点 x 中断寄存器 (OTG_DOEPINTx)（x = 0..5，其中 x 表示端点编号）

OTG device OUT endpoint x interrupt register

偏移地址：0xB08 + (x * 0x20)

复位值：0x0000 0080

此寄存器指示端点在出现 USB 和 AHB 相关事件时的状态。相关内容在*图 587* 中进行了说明。当 OTG_GINTSTS 寄存器中的 OUT 端点中断位（OTG_GINTSTS 中的 OEPINT 位）置 1 时，应用程序必须读取此寄存器。在应用程序能够读取此寄存器之前，必须先读取 OTG_DAINT 寄存器，以获取 OTG_DOEPINTx 寄存器的准确端点编号。应用程序必须将该寄存器中的相应位清零，才能将 OTG_DAINT 和 OTG_GINTSTS 寄存器中的对应位清零。

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.	NYET	NAK	BERR	Res.	Res.	Res.	OUT PKT ERR	Res.	Res.	STSPH SRX	OTEP DIS	STUP	Res.	EP DISD	XFRC
	rc_w1	rc_w1	rc_w1				rc_w1			rc_w1	rc_w1	rc_w1		rc_w1	rc_w1

位 31:16 保留，必须保持复位值。

位 15 保留，必须保持复位值。

位 14 **NYET**：NYET 中断 (NYET interrupt)
当非同步 OUT 端点回复 NYET 握手信号时将产生此中断。

位 13 **NAK**：NAK 输入 (NAK input)
当设备发出或收到 NAK 时，模块将生成该中断。如果是同步 IN 端点，由于 Tx FIFO 中无数据可发而发送长度为零的数据包时也会生成该中断。

位 12 **BERR**：Babble 错误中断 (Babble error interrupt)
当端点收到 babble 时，模块会生成此中断。

位 11:10 保留，必须保持复位值。

位 9 保留，必须保持复位值。

位 8 **OUTPKTERR**：OUT 数据包错误 (OUT packet error)
当模块检测到 OUT 数据包发生上溢或 CRC 错误时会产生该中断。该中断仅在阈值已使能时有效。

位 7 保留，必须保持复位值。

位 6 保留，必须保持复位值。

位 5 **STSPHSRX**：接收到控制写的状态阶段 (Status phase received for control write)
该中断仅对控制 OUT 端点有效。只有当 OTG_FS 将主机在控制写传输的数据阶段发送的所有数据全部传输到系统存储器缓冲区后，才会生成该中断。该中断向应用程序指示主机已从控制写传输的数据阶段切换到状态阶段。完成数据阶段的解码后，应用程序可使用该中断向状态阶段作出 ACK 或 STALL 响应。

位 4 **OTEPDIS**：端点禁止时接收到 OUT 令牌 (OUT token received when endpoint disabled)
仅适用于控制 OUT 端点。
指示在尚未使能端点时接收到 OUT 令牌，从而产生中断。

位 3 **STUP**：SETUP 阶段完成 (SETUP phase done)
仅适用于控制 OUT 端点。
指示控制端点的 SETUP 阶段已完成，当前控制传输中不再接收到连续的 SETUP 数据包。在此中断上，应用程序可以对接收到的 SETUP 数据包进行解码。

位 2 保留，必须保持复位值。

位 1 **EPDISD**：端点禁止中断 (Endpoint disabled interrupt)
此位指示该端点已经由应用程序禁止掉。

位 0 **XFRC**：传输完成中断 (Transfer completed interrupt)
此字段指示在此端点上设置的传输已经在 USB 和 AHB 上传输完成。

50.15.50 OTG 设备 OUT 端点 0 传输大小寄存器 (OTG_DOEPTSIZ0)

OTG device OUT endpoint 0 transfer size register

偏移地址：0xB10

复位值：0x0000 0000

在使能端点 0 之前，应用程序必须修改此寄存器。通过 OTG_DOEPCTL0 寄存器中的端点使能位（OTG_DOEPCTL0 中的 EPENA 位）使能端点 0 后，模块对此寄存器进行修改。当模块将端点使能位清零后，应用程序才能读取此寄存器。

非零端点使用端点 1–5 的寄存器。

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

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

位 31 保留，必须保持复位值。

位 30:29 **STUPCNT[1:0]**：SETUP 数据包计数 (SETUP packet count)

此字段指定端点能连续接收的 SETUP 数据包数量。

01：1 个数据包

10：2 个数据包

11：3 个数据包

位 28:20 保留，必须保持复位值。

位 19 **PKTCNT**：数据包计数 (Packet count)

每向 Rx FIFO 写入一个数据包，此字段递减。

位 18:7 保留，必须保持复位值。

位 6:0 **XFRSIZ[6:0]**：传输大小 (Transfer size)

指示端点 0 的一次数据传输包含的数据量，以字节为单位。仅当应用程序传输完这些数据后，模块才会中断该应用程序。传输大小可以设置为端点的最大数据包大小，以在每个数据包结束时中断。

每次从 Rx FIFO 读取数据包并将其写入外部存储器时，模块会使此字段递减。

50.15.51 OTG 设备 OUT 端点 x 控制寄存器 (OTG_DOEPCTLx)（x = 1..5，其中 x 表示端点编号）

OTG device OUT endpoint x control register

OUT 端点的偏移地址：0xB00 + (x * 0x20)

复位值：0x0000 0000

应用程序使用此寄存器控制各个逻辑端点（端点 0 除外）的行为。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
EPENA	EPDIS	SD1 PID/ SODD FRM	SD0 PID/ SEVN FRM	SNAK	CNAK	Res.	Res.	Res.	Res.	STALL	SNPM	EPTYP[1:0]		NAK STS	EO NUM/ DPID
rs	rs	w	w	w	w					rw/rs	rw	rw	rw	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
USBA EP	Res.	Res.	Res.	Res.	MPSIZ[10:0]										
rw					rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31 **EPENA**：端点使能 (Endpoint enable)

适用于 IN 和 OUT 端点。

应用程序将此位置 1 以在端点上启动数据发送。

在此端点上触发以下任一中断之前，模块会将此位清零：

– SETUP 阶段完成

– 端点禁止

– 传输完成

位 30 **EPDIS**：端点禁止 (Endpoint disable)

即使在该端点上的传送完成之前，应用程序也可将此位置 1，以停止端点上的数据发送/接收。应用程序必须等到发生端点禁止中断后，才能将端点视为禁止端点。在端点禁止中断位置 1 前，模块会将此位清零。只有在该端点的端点使能位置 1 后，应用程序才可将该位置 1。

位 29 **SD1PID**：设置 DATA1 PID (Set DATA1 PID)

仅适用于中断/批量 IN 和 OUT 端点。对此字段进行写操作会将此寄存器中的端点数据 PID (DPID) 字段设置为 DATA1。

SODDFRM：设置奇数帧 (Set odd frame)

仅适用于同步 IN 和 OUT 端点。对此字段进行写操作会将偶数/奇数帧 (EONUM) 字段设置为奇数帧。

位 28 **SD0PID**：设置 DATA0 PID (Set DATA0 PID)

仅适用于中断/批量 OUT 端点。

对此字段进行写操作会将此寄存器中的端点数据 PID (DPID) 字段设置为 DATA0。

SEVNFRM：设置偶数帧 (Set even frame)

仅适用于同步 OUT 端点。

对此字段进行写操作会将偶数/奇数帧 (EONUM) 字段设置为偶数帧。

位 27 **SNAK**：将 NAK 置 1 (Set NAK)

对此位进行写操作会将端点的 NAK 位置 1。

通过此位，应用程序可以控制端点上 NAK 握手信号的发送。发生传输完成中断时或端点上接收到 SETUP 后，模块也可以将 OUT 端点的该位置 1。

位 26 **CNAK**：将 NAK 清零 (Clear NAK)

对此位进行写操作会将端点的 NAK 位清零。

位 25:22 保留，必须保持复位值。

位 21 **STALL**：STALL 握手 (STALL handshake)

仅适用于非控制、非同步 OUT 端点（访问类型为 rw）。

应用程序将此位置 1 使得设备对来自 USB 主机的所有令牌都回复 STALL。如果 NAK 位、全局 IN NAK 或全局 OUT NAK 与此位同时置 1，则 STALL 位优先。只有应用程序能够将此位清零，而模块则不能。

仅适用于控制端点（访问类型为 rs）

此端点接收到 SETUP 令牌时，应用程序只能将此位置 1，而模块会将其清零。如果 NAK 位、全局 IN NAK 或全局 OUT NAK 与此位同时置 1，则 STALL 位优先。无论此位如何设置，模块总是通过 ACK 握手响应 SETUP 数据包。

位 20 **SNPM**：监听模式 (Snoop mode)

此位用于将端点配置为监听模式。在监听模式下，模块不会在将 OUT 数据包传输到应用存储区前检查其是否正确。

位 19:18 **EPTYP[1:0]**：端点类型 (Endpoint type)

以下是这个逻辑端点支持的传输类型。

00：控制

01：同步

10：批量

11：中断

位 17 **NAKSTS**：NAK 状态 (NAK status)

指示以下结果：

0：模块根据 FIFO 状态回复非 NAK 握手。

1：模块在此端点上回复 NAK 握手。

当应用程序或模块将此位置 1 时：

即使 Rx FIFO 存在空间可容纳传入数据包，模块也会停止在 OUT 端点上接收任何数据。

无论此位如何设置，模块总是通过 ACK 握手响应 SETUP 数据包。

位 16 **EONUM**：偶数/奇数帧 (Even/odd frame)

仅适用于同步 IN 和 OUT 端点。

指示模块为此端点发送/接收同步的数据所在的帧的编号。应用程序必须通过此寄存器中的 SEVNFRM 和 SODDFRM 字段对偶数/奇数帧编号进行编程，以便此端点发送/接收同步数据。

0：偶数帧

1：奇数帧

DPID：端点数据 PID (Endpoint data PID)

仅适用于中断/批量 OUT 端点。

包含此端点上将要接收或发送的数据包的 PID。端点激活后，应用程序必须对要在此端点上接收或发送的首个数据包的 PID 进行编程。应用程序使用 SD0PID 寄存器字段对 DATA0 或 DATA1 PID 进行编程。

0：DATA0

1：DATA1

位 15 **USBAEP**：USB 活动端点 (USB active endpoint)
指示此端点在当前配置和接口中是否激活。检测到 USB 复位后，模块会为所有端点（端点 0 除外）将此位清零。接收到 SetConfiguration 和 SetInterface 命令后，应用程序必须相应地对端点寄存器进行编程并将此位置 1。

位 14:11 保留，必须保持复位值。

位 10:0 **MPSIZ[10:0]**：最大数据包大小 (Maximum packet size)
应用程序必须将此字段编程为当前逻辑端点的最大数据包大小。此值以字节为单位。

50.15.52 OTG 设备 OUT 端点 x 传输大小寄存器 (OTG_DOEPTSIZx)（x = 1..5，其中 x 表示端点编号）

OTG device OUT endpoint x transfer size register

偏移地址：0xB10 + (x * 0x20)

复位值：0x0000 0000

在使能该端点之前，应用程序必须修改此寄存器。通过 OTG_DOEPCTLx 寄存器中的端点使能位（OTG_DOEPCTLx 中的 EPENA 位）使能该端点后，模块对此寄存器进行修改。当模块将端点使能位清零后，应用程序才能读取此寄存器。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	RXDPID/ STUPCNT[1:0]		PKTCNT[9:0]										XFRSIZ		
	r/rw	r/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
XFRSIZ															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31 保留，必须保持复位值。

位 30:29 **RXDPID[1:0]**：接收到的数据 PID (Received data PID)
仅适用于同步 OUT 端点。
这是此端点收到的上一个数据包的 PID。
00：DATA0
10：DATA1

STUPCNT[1:0]：SETUP 数据包计数 (SETUP packet count)
仅适用于控制 OUT 端点。
此字段指定端点能连续接收的 SETUP 数据包数量。
01：1 个数据包
10：2 个数据包
11：3 个数据包

位 28:19 **PKTCNT[9:0]**：数据包计数 (Packet count)

指示该端点上的一次数据传输包含的 USB 数据包个数。

每次向 Rx FIFO 写入数据包（最大大小数据包或短数据包）后，此字段将递减。

位 18:0 **XFRSIZ**：传输大小 (Transfer size)

此字段包含当前端点的一次数据传输包含的数据量，以字节为单位。仅当应用程序传输完这些数据后，模块才会中断该应用程序。传输大小可以设置为端点的最大数据包大小，以在每个数据包结束时中断。

每次从 Rx FIFO 读取数据包并将其写入外部存储器时，模块会使此字段递减。

50.15.53 OTG 电源和时钟门控控制寄存器 (OTG_PCGCCTL)

OTG power and clock gating control register

偏移地址：0xE00

复位值：0x200B 8000

此寄存器在主机模式和设备模式下均可用。

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.	Res.	Res.	Res.	Res.	Res.	Res.	SUSP	PHY SLEEP	ENL1 GTG	PHY SUSP	Res.	Res.	GATE HCLK	STPP CLK
								r	r	rw	r			rw	rw

位 31:8 保留，必须保持复位值。

位 7 **SUSP**：深度睡眠 (Deep Sleep)

在 L1 状态下时，此位表示 PHY 处于深度睡眠状态。

位 6 **PHYSLEEP**：PHY 处于睡眠状态 (PHY in Sleep)

此位指示 PHY 处于睡眠状态。

位 5 **ENL1GTG**：使能睡眠时钟门控 (Enable sleep clock gating)

此位置 1 时，如果模块无法触发 utmi_l1_suspend_n，则会在睡眠状态下使能模块内部时钟门控。当此位不置 1 时，不会在睡眠状态下使能 PHY 时钟门控。

位 4 **PHYSUSP**：PHY 挂起 (PHY suspended)

指示 PHY 已挂起。应用程序将 STPPCLK 位置 1 后，一旦 PHY 挂起，此位就会更新。

位 3:2 保留，必须保持复位值。

位 1 **GATEHCLK**：门控 HCLK (Gate HCLK)

当 USB 通信挂起或会话无效时，应用程序会将此位置 1，以停止对除 AHB 总线从接口、主接口和唤醒逻辑之外的模块提供时钟。当 USB 恢复通信或新会话启动时，应用程序将此位清零。

位 0 **STPPCLK**：停止 PHY 时钟 (Stop PHY clock)

当 USB 通信挂起、会话无效或设备断开连接时，应用程序将此位置 1 以停止 PHY 时钟。当 USB 恢复通信或新会话启动时，应用程序将此位清零。

50.15.54 OTG_FS 寄存器映射

下表提供了 USB OTG 寄存器映射和复位值。

表 327. OTG_FS 寄存器映射和复位值

偏移	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
0x000	OTG_GOTGCTL	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CURMOD	OTGVER	BSVLD	ASVLD	DBCT	CIDSTS	Res.	Res.	Res.	EHEN	DHNPEN	HSHNPEN	HNPRQ	HNGSCS	BVALOVAL	BVALOEN	AVALOVAL	AVALOEN	VBVALOVAL	VBVALOEN	SRQ	SRQSCS
	Reset value											0	0	0	0	0	1				0	0	0	0	0	0	0	0	0	0	0	0	0
0x004	OTG_GOTGINT	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	IDCHNG	DBCDNE	ADTOCHG	HNGDET	Res.	Res.	Res.	Res.	Res.	Res.	Res.	HNSSCHG	SRSSCHG	Res.	Res.	Res.	Res.	Res.	SEDET	Res.	Res.
	Reset value												0	0	0	0							-	0	0						0		
0x008	OTG_GAHBCFG	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PTXFELVL	TXFELVL	Res.	Res.	Res.	Res.	Res.	Res.	GINTMSK
	Reset value																								0	0							0
0x00C	OTG_GUSBCFG	Res.	FDMOD	FHMOD	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TRDT				HNPCAP	SRPCAP	Res.	PHYSEL	Res.	Res.	Res.	TOCAL		
	Reset value		0	0																0	1	0	1	0	0		1				0	0	0
0x010	OTG_GRSTCTL	AHBIDL	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TXFNUM					TXFFLSH	RXFFLSH	Res.	FCRST	PSRST	CSRST
	Reset value	1																					0	0	0	0	0	0	0		0	0	0
0x014	OTG_GINTSTS	WKUPINT	SRQINT	DISCINT	CIDSCHG	LPMINT	PTXFE	HCINT	HPRTINT	RSTDET	Res.	IPXFR/INCOMPISOOUT	IISOIXFR	OEPINT	IEPINT	Res.	Res.	EOPF	ISOODRP	ENUMDNE	USBRST	USBSUSP	ESUSP	Res.	Res.	GONAKEFF	GINAKEFF	NPTXFE	RXFLVL	SOF	OTGINT	MMIS	CMOD
	Reset value	0	0	0	1	0	1	0	0	0		0	0	0	0			0	0	0	0	0	0			0	0	1	0	0	0	0	0
0x018	OTG_GINTMSK	WUIM	SRQIM	DISCINT	CIDSCHGM	LPMINTM	PTXFEM	HCIM	PRTIM	RSTDETM	Res.	IPXFRM/IISOOXFRM	IISOIXFRM	OEPINT	IEPINT	Res.	Res.	EOPFM	ISOODRPM	ENUMDNEM	USBRST	USBSUSPM	ESUSPM	Res.	Res.	GONAKEFFM	GINAKEFFM	NPTXFEM	RXFLVLM	SOFM	OTGINT	MMISM	Res.
	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	

表 327. OTG_FS 寄存器映射和复位值（续）

偏移	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
0x01C	**OTG_ GRXSTSR (host mode)**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PKTSTS				DPID		BCNT											CHNUM			
	Reset value												0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	OTG_ GRXSTSR (Device mode)	Res.	Res.	Res.	Res.	STSPHST	Res.	Res.	FRMNUM				PKTSTS				DPID		BCNT											EPNUM			
	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
0x020	**OTG_ GRXSTSP (host mode)**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PKTSTS				DPID		BCNT											CHNUM			
	Reset value												0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	OTG_ GRXSTSP (Device mode)	Res.	Res.	Res.	Res.	STSPHST	Res.	Res.	FRMNUM				PKTSTS				DPID		BCNT											EPNUM			
	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
0x024	**OTG_ GRXFSIZ**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RXFD															
	Reset value																	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
0x028	**OTG_ HNPTXFSIZ/ OTG_ DIEPTXF0**	NPTXFD/TX0FD																NPTXFSA/TX0FSA															
	Reset value	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
0x02C	**OTG_ HNPTXSTS**	Res.	NPTXQTOP							NPTQXSAV								NPTXFSAV															
	Reset value		0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
0x038	**OTG_ GCCFG**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	VBDEN	SDEN	PDEN	DCDEN	BCDEN	PWRDWN	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PS2DET	SDET	PDET	DCDET
	Reset value											0	0	0	0	0	0													X	X	X	X
0x03C	**OTG_CID**	PRODUCT_ID																															
	Reset value	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	0	0	0	0	0	0
0x054	**OTG_ GLPMCFG**	Res.	Res.	Res.	ENBESL	LPMR CNTSTS			SNDLPM	LPM RCNT			LPMCHIDX				L1RSMOK	SLPSTS	LPM RSP		L1DSEN	BESLTHRS				L1SSEN	REMWAKE	BESL			LPMACK	LPMEN	
	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
0x058	**OTG_ GPWRDN**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ADPIF	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ADPMEN
	Reset value									0																							0

表 327. OTG_FS 寄存器映射和复位值（续）

偏移	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
0x060	**OTG_ GADPCTL**	Res.	Res.	Res.	AR		ADPTOIM	ADPSNSIM	ADPPRBIM	ADPTOIF	ADPSNSIF	ADPPRBIF	ADPEN	ADPRST	ENASNS	ENAPRB	RTIM											PRB PER		PRB DEL TA		PRB DS CHG	
	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
0x100	**OTG_ HPTXFSIZ**	PTXFSIZ																PTXSA															
	Reset value	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
0x104	**OTG_ DIEPTXF1**	INEPTXFD																INEPTXSA															
	Reset value	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
0x108	**OTG_ DIEPTXF2**	INEPTXFD																INEPTXSA															
	Reset value	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0
. . . .	. . . .	. . . .																															
0x114	**OTG_ DIEPTXF5**	INEPTXFD																INEPTXSA															
	Reset value	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0
0x400	**OTG_ HCFG**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FSLSS	FSLS PCS	
	Reset value																														0	0	0
0x404	**OTG_ HFIR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RLDCTRL	FRIVL															
	Reset value																0	1	1	1	0	1	0	1	0	0	1	1	0	0	0	0	0
0x408	**OTG_ HFNUM**	FTREM																FRNUM															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0x410	**OTG_ HPTXSTS**	PTXQTOP								PTXQSAV								PTXFSAVL															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
0x414	**OTG_ HAINT**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	HAINT															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x418	**OTG_ HAINTMSK**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	HAINTM															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x440	**OTG_ HPRT**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PSPD		PTCTL				PPWR	PLSTS		Res.	PRST	PSUSP	PRES	POCCHNG	POCA	PENCHNG	PENA	PCDET	PCSTS
	Reset value														0	0	0	0	0	0	0	0	0		0	0	0	0	0	0	0	0	0

表 327. OTG_FS 寄存器映射和复位值（续）

偏移	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
0x500	**OTG_ HCCHAR0**	CHENA	CHDIS	ODDFRM	DAD							MCNT		EPTYP		LSDEV	Res.	EPDIR	EPNUM				MPSIZ										
	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
0x508	**OTG_ HCINT0**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DTERR	FRMOR	BBERR	TXERR	Res.	ACK	NAK	STALL	Res.	CHH	XFRC
	Reset value																						0	0	0	0		0	0	0		0	0
0x50C	**OTG_ HCINTMSK0**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DTERRM	FRMORM	BBERRM	TXERRM	Res.	ACKM	NAKM	STALLM	Res.	CHHM	XFRCM
	Reset value																						0	0	0	0		0	0	0		0	0
0x510	**OTG_ HCTSIZ0**	Res.	DPID		PKTCNT											XFRSIZ																	
	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
0x520	**OTG_ HCCHAR1**	CHENA	CHDIS	ODDFRM	DAD							MCNT		EPTYP		LSDEV	Res.	EPDIR	EPNUM				MPSIZ										
	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
0x528	**OTG_ HCINT1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DTERR	FRMOR	BBERR	TXERR	Res.	ACK	NAK	STALL	Res.	CHH	XFRC
	Reset value																						0	0	0	0		0	0	0		0	0
0x52C	**OTG_ HCINTMSK1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DTERRM	FRMORM	BBERRM	TXERRM	Res.	ACKM	NAKM	STALLM	Res.	CHHM	XFRCM
	Reset value																						0	0	0	0		0	0	0		0	0
0x530	**OTG_ HCTSIZ1**	Res.	DPID		PKTCNT											XFRSIZ																	
	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
. . . .	. . . .	.																															
0x660	**OTG_ HCCHAR11**	CHENA	CHDIS	ODDFRM	DAD							MCNT		EPTYP		LSDEV	Res	EPDIR	EPNUM				MPSIZ										
	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
0x668	**OTG_ HCINT11**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res	Res	Res	Res.	DTERR	FRMOR	BBERR	TXERR	Res.	ACK	NAK	STALL	Res.	CHH	XFRC
	Reset value																						0	0	0	0		0	0	0		0	0

表 327. OTG_FS 寄存器映射和复位值（续）

偏移	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
0x66C	**OTG_ HCINTMSK11**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DTERRM	FRMORM	BBERRM	TXERRM	Res.	ACKM	NAKM	STALLM	Res.	CHHM	XFRCM
	Reset value																						0	0	0	0		0	0	0		0	0
0x670	**OTG_ HCTSIZ11**	Res	DPID		PKTCNT										XFRSIZ																		
	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
0x800	**OTG_ DCFG**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ERRATIM	XCVRDLY	Res.	PFIVL		DAD							Res.	NZLSOHSK	DSPD	
	Reset value																	0	0		0	0	0	0	0	0	0	0	0		0	0	0
0x804	**OTG_ DCTL**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DSBESLRJCT	Res.	Res.	Res.	Res.	Res.	Res.	POPRGDNE	CGONAK	SGONAK	CGINAK	SGINAK	TCTL			GONSTS	GINSTS	SDIS	RWUSIG
	Reset value														0							0	0	0	0	0	0	0	0	0	0	1	0
0x808	**OTG_ DSTS**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DEV LN STS		FNSOF											Res.	Res.	Res.	Res.	EERR	ENUMSPD		SUSPSTS			
	Reset value									0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0					0	0	0	0
0x810	**OTG_ DIEPMSK**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	NAKM	Res.	Res.	Res.	Res.	TXFURM	Res.	INEPNEM	INEPNMM	ITTXFEMSK	TOM	Res.	EPDM	XFRCM
	Reset value																			0					0		0	0	0	0		0	0
0x814	**OTG_ DOEPMSK**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res	Res.	Res.	Res	Res	Res.	Res.	Res.	Res	NYETMSK	NAKMSK	BERRM	Res.	Res.	Res.	OUTPKTERRM	Res.	Res.	Res	OTEPDM	STUPM	Res.	EPDM	XFRCM
	Reset value																		0	0	0				0				0	0		0	0
0x818	**OTG_ DAINT**	OEPINT													IEPINT																		
	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
0x81C	**OTG_ DAINTMSK**	OEPM													IEPM																		
	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
0x828	**OTG_ DVBUSDIS**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	VBUSDT															
	Reset value																	0	0	0	1	0	1	1	1	1	1	0	1	0	1	1	1
0x82C	**OTG_DVB USPULSE**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DVBUSP															
	Reset value																	0	0	0	0	0	1	0	1	1	0	1	1	1	0	0	0

表 327. OTG_FS 寄存器映射和复位值（续）

偏移	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
0x834	**OTG_DIEPEMPMSK**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	INEPTXFEM															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x900	**OTG_DIEPCTL0**	EPENA	EPDIS	Res.	Res.	SNAK	CNAK	TXFNUM				STALL	Res.	EPTYP		NAKSTS	Res.	USBAEP	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MPSIZ	
	Reset value	0	0			0	0	0	0	0	0	0		0	0	0		1														0	0
0x908	**OTG_DIEPINT0**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	NAK	Res.	PKTDRPSTS	Res.	Res.	TXFIFOUDRN	TXFE	INEPNE	INEPNM	ITTXFE	TOC	Res.	EPDISD	XFRC
	Reset value																			0		0			0	1	0	0	0	0		0	0
0x910	**OTG_DIEPTSIZ0**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PKT CNT		Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	XFRSIZ						
	Reset value												0	0													0	0	0	0	0	0	0
0x918	**OTG_DTXFSTS0**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	INEPTFSAV															
	Reset value																	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
0x920	**OTG_DIEPCTL1**	EPENA	EPDIS	SODDFRM/SD1PID	SD0PID/SEVNFRM	SNAK	CNAK	TXFNUM				STALL	Res.	EPTYP		NAKSTS	EONUM/DPID	USBAEP	Res.	Res.	Res.	Res.	MPSIZ										
	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
0x928	**OTG_DIEPINT1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	NAK	Res.	PKTDRPSTS	Res.	Res.	TXFIFOUDRN	TXFE	INEPNE	INEPNM	ITTXFE	TOC	Res.	EPDISD	XFRC
	Reset value																			0		0			0	1	0	0	0	0		0	0
0x930	**OTG_DIEPTSIZ1**	Res.	MCNT		PKTCNT										XFRSIZ																		
	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
0x938	**OTG_DTXFSTS1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	INEPTFSAV															
	Reset value																	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
0x940	**OTG_DIEPCTL2**	EPENA	EPDIS	SODDFRM	SD0PID/SEVNFRM	SNAK	CNAK	TXFNUM				STALL	Res.	EPTYP		NAKSTS	EONUM/DPID	USBAEP	Res.	Res.	Res.	Res.	MPSIZ										
	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

表 327. OTG_FS 寄存器映射和复位值（续）

偏移	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
. . . .	. . . .	. . . .																															
0x9A0	**OTG_ DIEPCTL5**	EPENA	EPDIS	SODDFRM	SD0PID/SEVNFRM	SNAK	CNAK	TXFNUM				STALL	Res.	EPTYP		NAKSTS	EONUM/DPID	USBAEP	Res.	Res.	Res.	Res.	MPSIZ										
	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
. . . .	. . . .	. . . .																															
0x9A8	**OTG_ DIEPINT5**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	NAK	Res.	PKTDRPSTS	Res.	Res.	TXFIFOUDRN	TXFE	INEPNE	INEPNM	ITTXFE	TOC	Res.	EPDISD	XFRC
	Reset value																			0		0			0	1	0	0	0	0		0	0
. . . .	. . . .	. . . .																															
0x9B8	**OTG_ DTXFSTS5**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	INEPTFSAV															
	Reset value																	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
. . . .	. . . .	. . . .																															
0x9B0	**OTG_ DIEPTSIZ5**	Res.	MCNT		PKTCNT										XFRSIZ																		
	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
0xB00	**OTG_ DOEPCTL0**	EPENA	EPDIS	Res.	Res.	SNAK	CNAK	Res.	Res.	Res.	Res.	STALL	SNPM	EPTYP		NAKSTS	Res.	USBAEP	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MPSIZ	
	Reset value	0	0			0	0					0	0	0	0	0		1													0	0
0xB08	**OTG_ DOEPINT0**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	NYET	NAK	BERR	Res.	Res.	Res.	OUTPKTERR	Res.	Res.	STSPHSRX	OTEPDIS	STUP	Res.	EPDISD	XFRC
	Reset value																		0	0	0				0			0	0	0		0	0
0xB10	**OTG_ DOEPTSIZ0**	Res.	STUP CNT		Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PKTCNT	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	XFRSIZ						
	Reset value		0	0										0													0	0	0	0	0	0	0

表 327. OTG_FS 寄存器映射和复位值（续）

偏移	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
0xB20	OTG_ DOEPCTL1	EPENA	EPDIS	SODDFRM	SD0PID/SEVNFRM	SNAK	CNAK	Res.	Res.	Res.	Res.	STALL	SNPM	EP TYP		NAKSTS	EONUM/DPID	USBAEP	Res.	Res.	Res.	Res.	MPSIZ										
	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
0xB28	OTG_ DOEPINT1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	NYET	NAK	BERR	Res.	Res.	Res.	OUTPKTERR	Res.	Res.	STSPHSRX	OTEPDIS	STUP	Res.	EPDISD	XFRC
	Reset value																		0	0	0				0			0	0	0		0	0
0xB30	OTG_ DOEPTSIZ1	Res.	RXDPID/ STUPCNT		PKTCNT											XFRSIZ																	
	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
. . . .	. . . .	. . .																															
0xBA0	OTG_ DOEPCTL5	EPENA	EPDIS	SODDFRM	SD0PID/SEVNFRM	SNAK	CNAK	Res.	Res.	Res.	Res.	STALL	SNPM	EPTYP		NAKSTS	EONUM/DPID	USBAEP	Res.	Res.	Res.	Res.	MPSIZ										
	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
0xBA8	OTG_ DOEPINT5	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	NYET	NAK	BERR	Res.	Res.	Res.	OUTPKTERR	Res.	Res.	STSPHSRX	OTEPDIS	STUP	Res.	EPDISD	XFRC
	Reset value																		0	0	0				0	-		0	0	0		0	0
0xBB0	OTG_ DOEPTSIZ5	Res.	RXDPID/ STUPCNT		PKTCNT											XFRSIZ																	
	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
0xE00	OTG_ PCGCCTL	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SUSP	PHYSLEEP	ENL1GTG	PHYSUSP	Res.	Res.	GATEHCLK	STPPCLK
	Reset value																									0	0	0	0			0	0

有关寄存器边界地址的信息，请参见*第 78 页的第 2.2.2 节*。

50.16 OTG_FS 编程模块

50.16.1 模块初始化

应用程序必须执行模块初始化序列。如果上电期间连接电缆，则 OTG_GINTSTS 中的当前工作模式位（OTG_GINTSTS 中的 CMOD 位）将指示模式。连接“A 型”插头后，OTG_FS 控制器进入主机模式；连接“B 型”插头后，OTG_FS 控制器进入设备模式。

本节介绍了 OTG_FS 控制器在上电后的初始化过程。无论是以主机模式还是设备模式工作，应用程序都必须遵循初始化序列。根据模块配置对所有模块全局寄存器进行初始化：

1. 在 OTG_GAHBCFG 寄存器中编程以下字段：
   - 全局中断屏蔽位 GINTMSK = 1
   - Rx FIFO 非空（OTG_GINTSTS 中的 RXFLVL 位）
   - 周期性 Tx FIFO 空门限
2. 在 OTG_GUSBCFG 寄存器中编程以下字段：
   - HNP 功能位
   - SRP 功能位
   - OTG_FS 超时校准字段
   - USB 周转时间字段
3. 软件必须使能 OTG_GINTMSK 寄存器中的以下位：

   OTG 中断屏蔽

   模式不匹配中断屏蔽
4. 通过读取 OTG_GINTSTS 中的 CMOD 位，软件可确定 OTG_FS 控制器是在主机模式还是设备模式下工作。

50.16.2 主机初始化

要将模块作为主机进行初始化，应用程序必须执行以下步骤：

1. 编程 OTG_GINTMSK 寄存器中的 HPRTINT 以打开中断。
2. 编程 OTG_HCFG 寄存器以选择全速主机。
3. 将 OTG_HPRT 中的 PPWR 位编程为 1，给 USB 总线提供 V_{BUS}。
4. 等待 OTG_HPRT0 中的 PCDET 中断。这表示某设备已连接到主机端口。
5. 将 OTG_HPRT 中的 PRST 位编程为 1，在 USB 总线上发出复位信号。
6. 至少等待 10 ms，以便完成复位过程。
7. 将 OTG_HPRT 中的 PRST 位编程为 0。
8. 等待 OTG_HPRT 中的 PENCHNG 中断。
9. 读取 OTG_HPRT 中的 PSPD 位以获取枚举速度。
10. 使用所选 PHY 时钟 ，相应地设置 HFIR 寄存器。
11. 根据步骤 9 中检测到的设备速度编程 OTG_HCFG 寄存器中的 FSLSPCS 字段。如果 FSLSPCS 发生更改，则必须执行端口复位。
12. 编程 OTG_GRXFSIZ 寄存器以选择接收 FIFO 的大小。
13. 编程 OTG_HNPTXFSIZ 寄存器，以选择用于非周期性通信事务的非周期性发送 FIFO 的大小和起始地址。
14. 编程 OTG_HPTXFSIZ 寄存器，以选择用于周期性通信事务的周期性发送 FIFO 的大小和起始地址。

要与设备通信，系统软件必须初始化并使能至少一个通道。

50.16.3 设备初始化

上电期间或者从主机模式切换为设备模式后，应用程序必须执行下列步骤来将模块作为设备进行初始化。

1. 在 OTG_DCFG 寄存器中编程以下字段：
   - 设备速度
   - 非零长度状态 OUT 握手信号
2. 编程 OTG_GINTMSK 寄存器以使能以下中断：
   - USB 复位
   - 枚举完成
   - 早期挂起
   - USB 挂起
   - SOF
3. 等待 OTG_GINTSTS 中的 USBRST 中断。这表示已在 USB 上检测到复位信号，复位过程自接收到此中断后约持续 10 ms。

等待 OTG_GINTSTS 中的 ENUMDNE 中断。此中断指示 USB 上复位过程结束。接收到此中断时，应用程序必须读取 OTG_DSTS 寄存器以确定枚举速度并执行*第 1902 页的枚举完成时的端点初始化*中所列的步骤。

此时，设备已准备好接受 SOF 数据包并在控制端点 0 上执行控制传输。

50.16.4 主机编程模型

通道初始化

应用程序必须初始化一个或多个通道，之后才能与所连接的设备通信。要初始化和使能通道，应用程序必须执行以下步骤：

1. 编程 OTG_GINTMSK 寄存器以取消对以下位的中断屏蔽：
2. 通道中断
   - 用于 OUT 事务的非周期性发送 FIFO 为空（在流水线事务级别工作且数据包计数字段编程值大于 1 时适用）。
   - 用于 OUT 事务的非周期性发送 FIFO 为半空（在流水线事务级别工作且数据包计数字段编程值大于 1 时适用）。
3. 编程 OTG_HAINTMSK 寄存器以使能所选通道中断。
4. 编程 OTG_HCINTMSK 寄存器，以使能主机通道中断寄存器中反映的和通信事务有关的中断。
5. 编程所选通道的 OTG_HCTSIZx 寄存器，指定以字节为单位的总传输大小和包括短数据包在内的预期数据包个数。应用程序必须使用初始数据 PID（用于第一个 OUT 事务或预期从第一个 IN 事务获取）编程 PID 字段。
6. 编程所选通道的 OTG_HCCHARx 寄存器，指定设备的端点特性，例如类型、速度、方向等。（仅当应用程序准备好发送或接收数据包时，才能通过将通道使能位置 1 来使能通道）。

通道的停止

应用程序可以通过编程 OTG_HCCHARx 寄存器将 CHDIS 和 CHENA 位置 1 来禁止任何通道。这会使 OTG_FS 主机清空之前在该通道上发出的请求（如果有）并生成通道停止中断。应用程序在将通道重新分配给其他通信事务之前，必须等待 OTG_HCINTx 中的 CHH 中断。OTG_FS 主机不会中断已在 USB 上启动的通信事务。

禁止通道前，应用程序必须确保非周期性请求队列（禁止非周期性通道时）或周期性请求队列（禁止周期性通道时）中至少有一个空闲空间。应用程序可以在请求队列已满时（禁止通道之前），通过编程 OTG_HCCHARx 寄存器将 CHDIS 位置 1 和将 CHENA 位清零，清空已发出的请求。

出现以下任一情况时，应用程序将禁止通道：

1. IN 或 OUT 通道的 OTG_HCINTx 中接收到 STALL、TXERR、BBERR 或 DTERR 中断。应用程序在接收到通道停止信号之前，必须能够接收相同通道的其他中断（DTERR、Nak、data、TXERR）。
2. 接收到 OTG_GINTSTS 中的 DISCINT（断开设备连接）中断。（应用程序将禁止所有已使能的通道）。
3. 应用程序在传输正常完成之前将其中止。

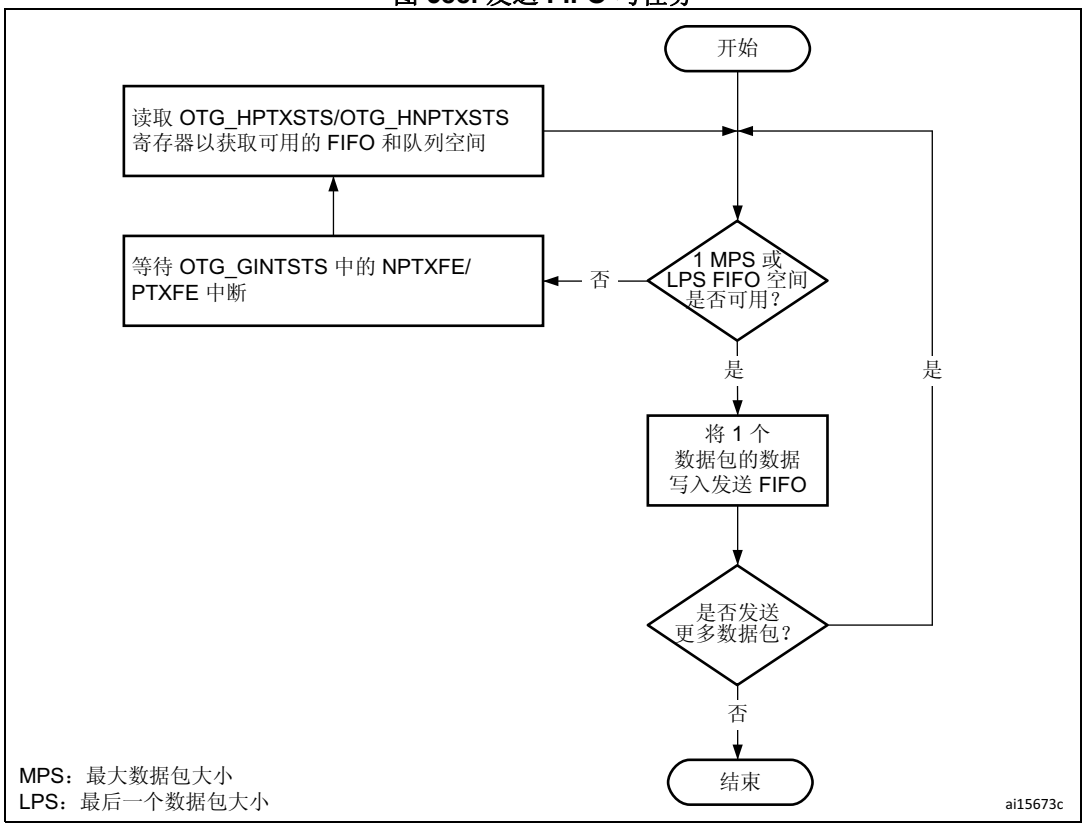
操作模型

应用程序必须初始化一个通道，之后才能与所连接的设备通信。本节介绍了针对不同 USB 事务类型要执行的操作序列。

- **写入发送 FIFO**

  应用程序对数据包执行最后一个 32 位字写操作的同时，OTG_FS 主机自动向周期性/非周期性请求队列写入一个条目（OUT 请求）。因此开始向发送 FIFO 写入数据之前，应用程序必须确保周期性/非周期性请求队列中至少有一个空闲空间。应用程序必须始终以 32 位字形式向发送 FIFO 写入数据。如果数据包大小不是 32 位字的整数倍，则应用程序必须将数据包填充到 32 位字的整数倍。OTG_FS 主机根据设定的最大数据包大小和传输大小确定实际数据包大小。

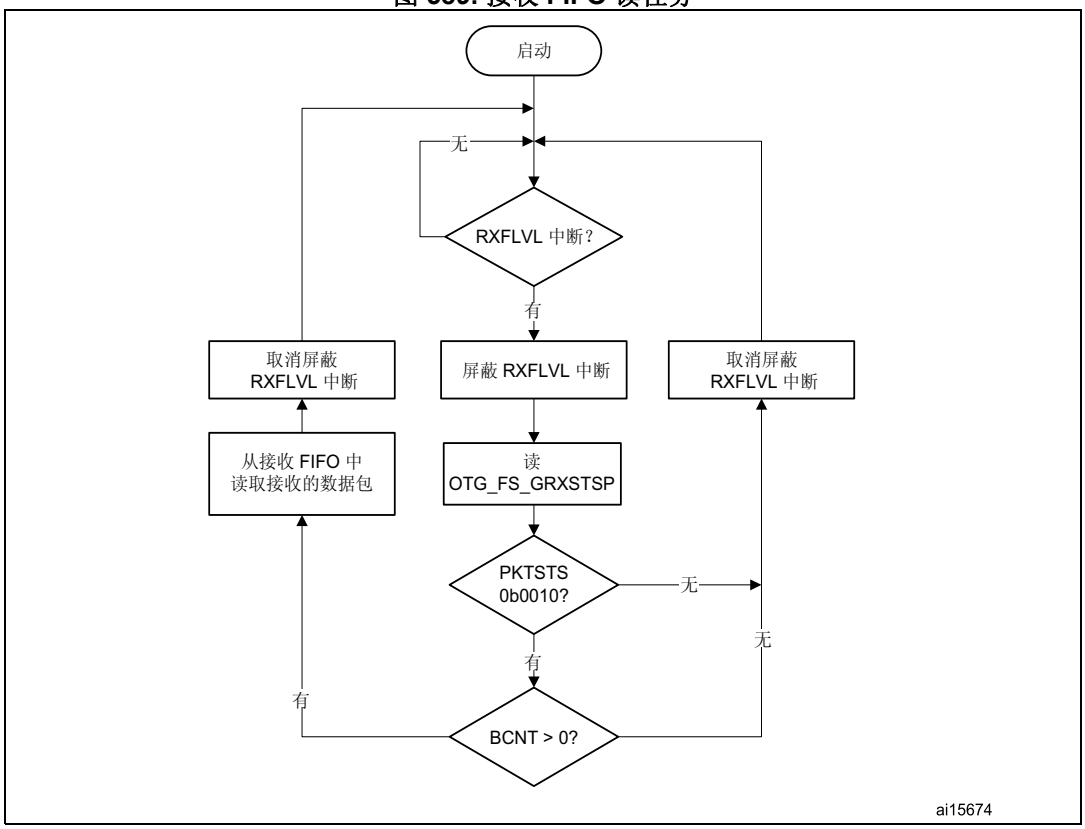
图 588. 发送 FIFO 写任务



- **读取接收 FIFO**

  应用程序必须忽略除 IN 数据包 (bx0010) 以外的所有数据包状态。

图 589. 接收 FIFO 读任务



- **批量和控制传输类型的 OUT/SETUP 通信事务**

  图 590 显示了典型的批量或控制 OUT/SETUP 流水线事务级操作。请参见通道 1 (ch_1)。该通道发送了两个批量 OUT 数据包。控制 SETUP 事务的工作方式相同，只不过只包含一个数据包。假设：

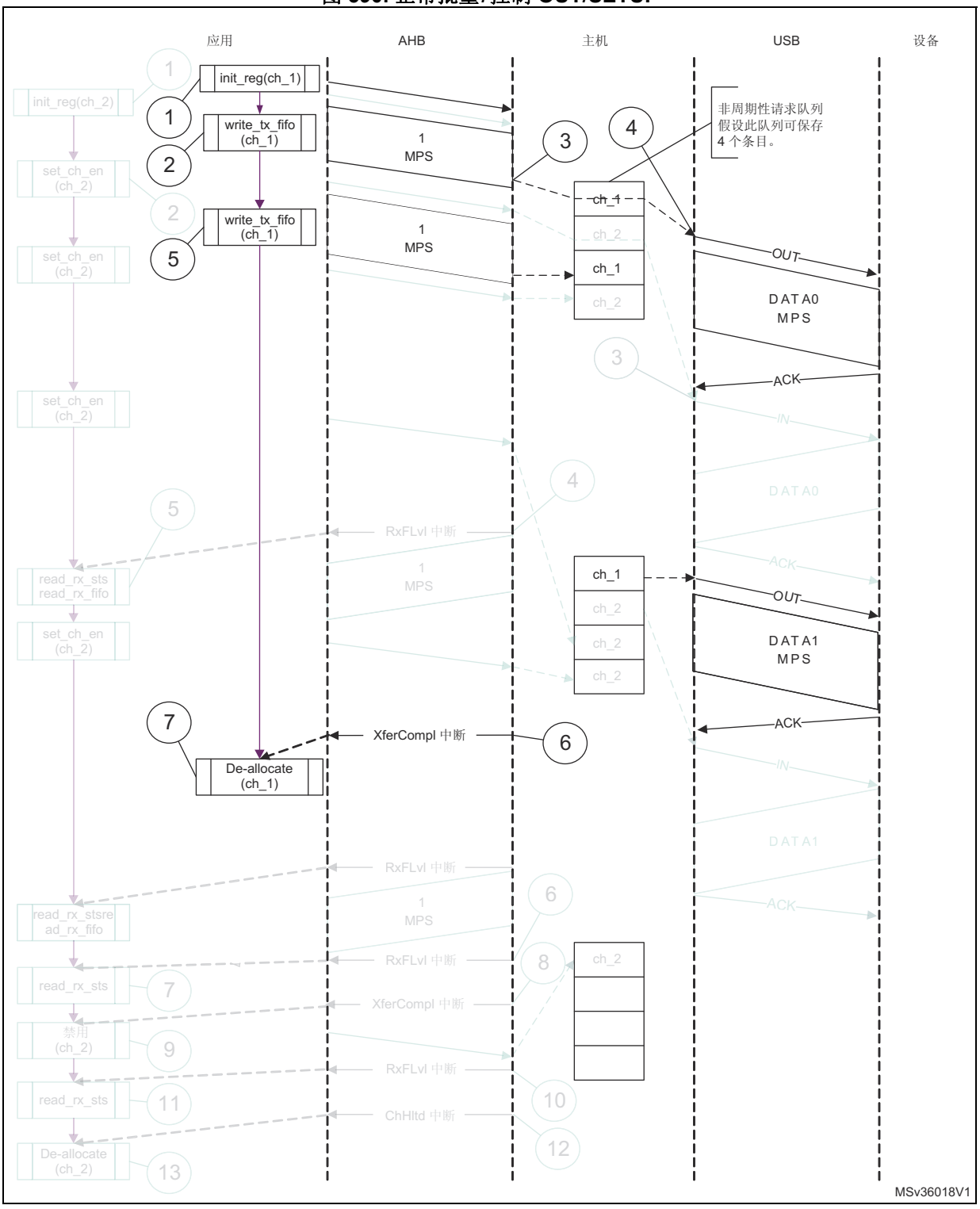
  – 应用程序尝试发送两个最大数据包大小的数据包（传输大小 = 1024 字节）。
  – 非周期性发送 FIFO 可存储两个数据包（FS 对应 128 字节）。
  – 非周期性请求队列深度 = 4。

- **正常批量和控制传输类型的 OUT/SETUP 操作**

  （通道 1）中的操作顺序如下：

1. 初始化通道 1
2. 写入通道 1 的第一个数据包
3. 在应用执行最后一次字写操作时，模块将向非周期性请求队列写入一个请求条目
4. 只要非周期性队列非空，模块即会尝试在当前帧内发送一个 OUT 令牌
5. 写入通道 1 的第二个（最后一个）数据包
6. 成功完成最后一个事务后，模块立即生成 XFRC 中断
7. 为了响应 XFRC 中断，将释放通道以供其他传输操作使用
8. 处理非 ACK 响应

图 590. 正常批量/控制 OUT/SETUP



1. 灰显元素与此图的上下文不相关。

以下代码示例说明了批量和控制 OUT/SETUP 传输类型的通信事务的通道相关中断服务程序。

- **批量/控制 OUT/SETUP 和批量/控制 IN 事务的中断服务程序**

  a) 批量/控制 OUT/SETUP

```
Unmask (NAK/TXERR/STALL/XFRC)
if (XFRC)
   {
   Reset Error Count
   Mask ACK
   De-allocate Channel
   }
else if (STALL)
   {
   Transfer Done = 1
   Unmask CHH
   Disable Channel
   }
else if (NAK or TXERR )
   {
   Rewind Buffer Pointers
   Unmask CHH
   Disable Channel
   if (TXERR)
      {
      Increment Error Count
      Unmask ACK
      }
   else
      {
      Reset Error Count
      }
      }
else if (CHH)
   {
   Mask CHH
   if (Transfer Done or (Error_count == 3))
      {
      De-allocate Channel
      }
   else
      {
      Re-initialize Channel
      }
   }
```

```
   else if (ACK)
      {
      Reset Error Count
      Mask ACK
      }
```

当发送 FIFO 和请求队列中有可用空间时，应用程序会将数据包写入发送 FIFO。应用程序可利用 OTG_GINTSTS 中的 NPTXFE 中断确定发送 FIFO 空间。

b) 批量/控制 IN

```
Unmask (TXERR/XFRC/BBERR/STALL/DTERR)
if (XFRC)
   {
   Reset Error Count
   Unmask CHH
   Disable Channel
   Reset Error Count
   Mask ACK
   }
else if (TXERR or BBERR or STALL)
   {
   Unmask CHH
   Disable Channel
   if (TXERR)
      {
      Increment Error Count
      Unmask ACK
      }
   }
else if (CHH)
   {
   Mask CHH
   if (Transfer Done or (Error_count == 3))
      {
      De-allocate Channel
      }
   else
      {
      Re-initialize Channel
      }
   }
else if (ACK)
   {
   Reset Error Count
   Mask ACK
   }
```

```
else if (DTERR)
   {
   Reset Error Count
   }
```

当请求队列空间可用时，应用程序会写入请求，直到接收到 XFRC 中断。

- **批量和控制 IN 事务**

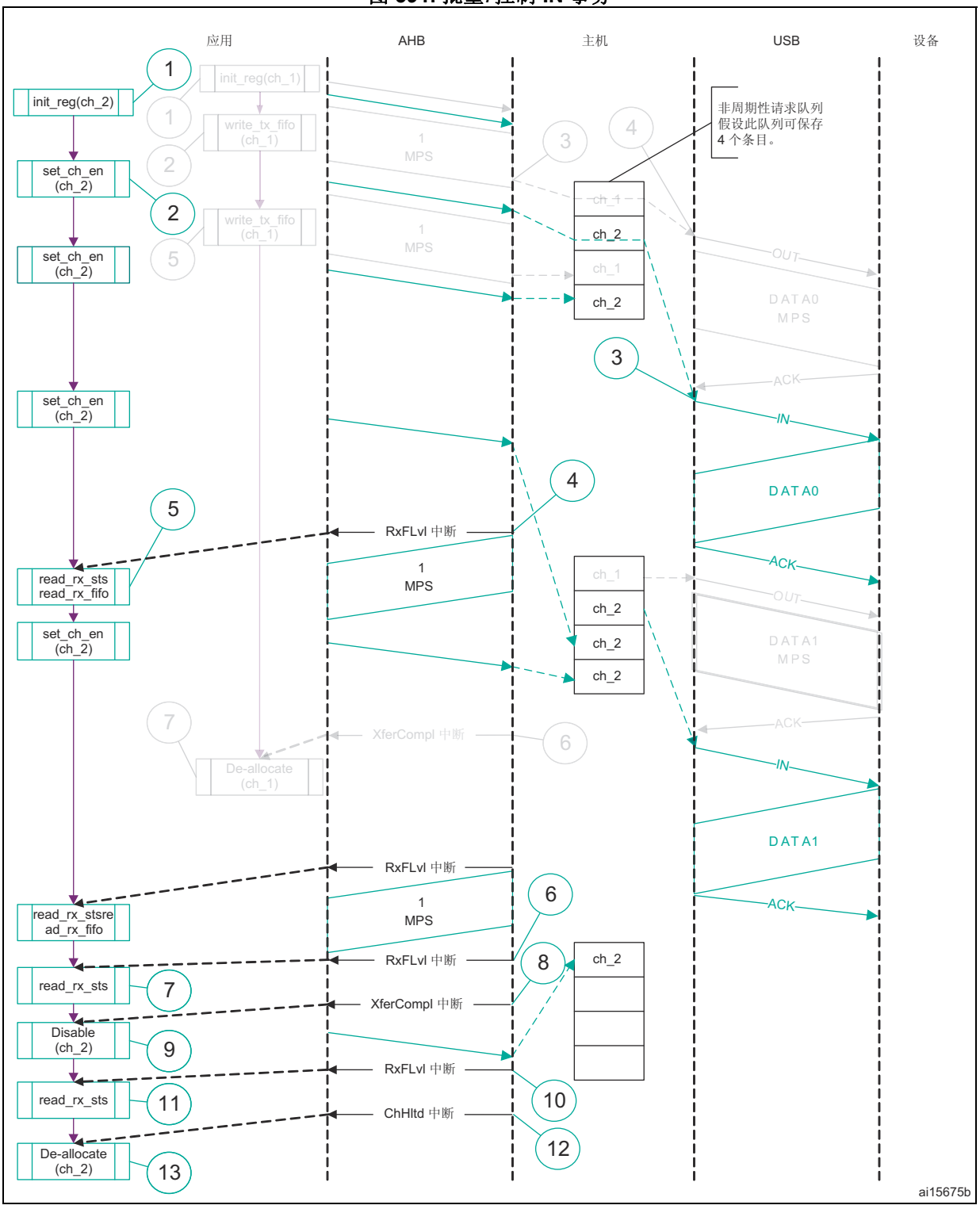
  图 591 显示了典型的批量或控制传输类型的 IN 流水线事务级操作。请参见通道 2 (ch_2)。假设：

  – 应用程序要接收两个最大数据包大小的数据包（传输大小 = 1024 字节）。

  – 接收 FIFO 可以包含至少一个最大数据包大小的数据包和每个数据包的两个状态字（FS 对应 72 字节）。

  – 非周期性请求队列深度 = 4。

图 591. 批量/控制 IN 事务



1. 灰显元素与此图的上下文不相关。

操作顺序如下：

1. 初始化通道 2。
2. 将 OTG_HCCHAR2 中的 CHENA 位置 1，以向非周期性请求队列写入 IN 请求。
3. 模块尝试在完成当前 OUT 事务后发送 IN 令牌。
4. 接收到的数据包写入接收 FIFO 后，模块立即生成 RXFLVL 中断。
5. 为了响应 RXFLVL 中断，将屏蔽 RXFLVL 中断并读取接收到的数据包状态，以此确定接收的字节数，然后相应地读取接收 FIFO。之后使能 RXFLVL 中断。
6. 模块针对接收 FIFO 中的传输完成状态条目生成 RXFLVL 中断。
7. 应用程序必须读取接收数据包状态，并在不是 IN 数据包（OTG_GRXSTSR 中的 PKTSTS ≠ 0b0010）时忽略它。
8. 读取接收数据包状态后，模块立即生成 XFRC 中断。
9. 为了响应 XFRC 中断，将禁止通道并停止针对其他请求向 OTG_HCCHAR2 写入数据。向 OTG_HCCHAR2 寄存器写入数据时，模块立即向非周期性请求队列写入通道禁止请求。
10. 停止状态写入接收 FIFO 后，模块立即生成 RXFLVL 中断。
11. 读取并忽略接收数据包状态。
12. 只要停止状态从接收 FIFO 弹出，模块立即生成 CHH 中断。
13. 为了响应 CHH 中断，将释放通道以供其他传输操作使用。
14. 处理非 ACK 响应。

- **控制事务**

  控制传输的建立、数据和状态阶段必须作为三个独立的传输过程来执行。建立、数据和状态阶段的 OUT 事务与上文所述的批量 OUT 事务的执行方式类似。数据或状态阶段的 IN 事务与上文所述的批量 IN 事务的执行方式类似。在所有这三个阶段，应用程序都会将 OTG_HCCHAR1 中的 EPTYP 字段设置为控制传输类型。在建立阶段期间，应用程序会将 OTG_HCTSIZ1 中的 PID 字段设置为 SETUP。

- **中断 OUT 事务**

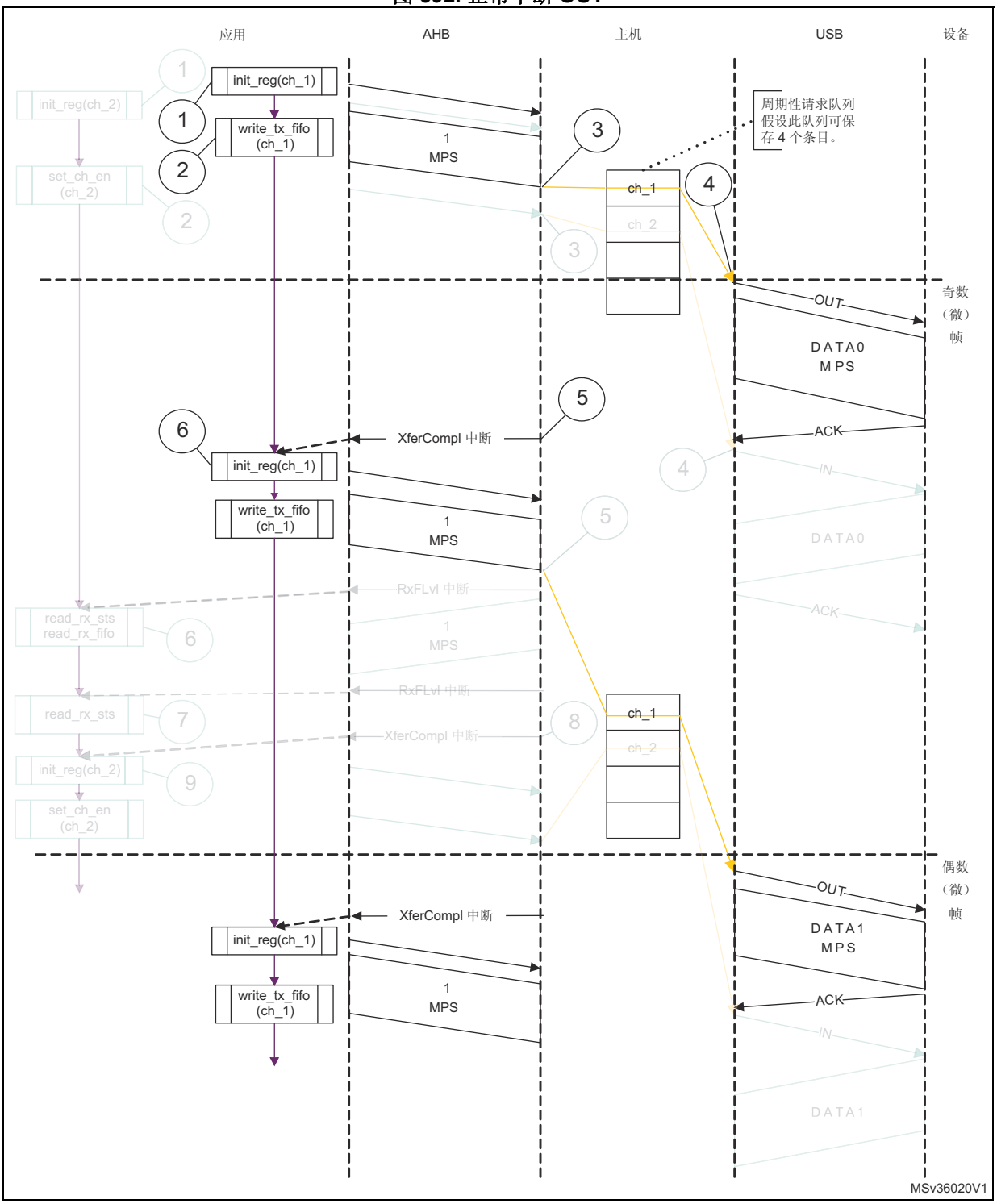
  图 592 显示了典型的中断 OUT 操作。假设：

  – 应用程序尝试从奇数帧开始，每帧发送一个数据包（高达 1 个最大数据包大小）（传输大小 = 1024 字节）
  – 周期性发送 FIFO 可存储一个数据包 (1 KB)
  – 周期性请求队列深度 = 4

  操作顺序如下：

1. 初始化和使能通道 1。应用程序必须将 OTG_HCCHAR1 中的 ODDFRM 位置 1。
2. 写入通道 1 的第一个数据包。
3. 应用程序在执行每个数据包的最后一次字写操作时，OTG_FS 主机向周期性请求队列写入一个请求条目。
4. OTG_FS 主机尝试在下一奇数帧发送 OUT 令牌。
5. 成功发送最后一个数据包后，OTG_FS 主机立即生成 XFRC 中断。
6. 为了响应 XFRC 中断，将重新初始化通道以供下一传输操作使用。

图 592. 正常中断 OUT



1. 灰显元素与此图的上下文不相关。

- **中断 OUT/IN 事务的中断服务程序**

  a) 中断 OUT

```
Unmask (NAK/TXERR/STALL/XFRC/FRMOR)
```

```
if (XFRC)
   {
   Reset Error Count
   Mask ACK
   De-allocate Channel
   }
else
   if (STALL or FRMOR)
      {
      Mask ACK
      Unmask CHH
      Disable Channel
      if (STALL)
         {
         Transfer Done = 1
         }
      }
   else
      if (NAK or TXERR)
         {
         Rewind Buffer Pointers
         Reset Error Count
         Mask ACK
         Unmask CHH
         Disable Channel
         }
      else
         if (CHH)
         {
         Mask CHH
         if (Transfer Done or (Error_count == 3))
            {
            De-allocate Channel
            }
         else
            {
            Re-initialize Channel (in next b_interval - 1 Frame)
            }
         }
      else
         if (ACK)
         {
         Reset Error Count
         Mask ACK
         }
```

应用程序利用 OTG_GINTSTS 中的 NPTXFE 中断确定发送 FIFO 空间。

```
中断 IN
Unmask (NAK/TXERR/XFRC/BBERR/STALL/FRMOR/DTERR)
if (XFRC)
   {
   Reset Error Count
   Mask ACK
   if (OTG_HCTSIZx.PKTCNT == 0)
      {
      De-allocate Channel
      }
   else
      {
      Transfer Done = 1
      Unmask CHH
      Disable Channel
      }
   }
else
   if (STALL or FRMOR or NAK or DTERR or BBERR)
      {
      Mask ACK
      Unmask CHH
      Disable Channel
      if (STALL or BBERR)
         {
         Reset Error Count
         Transfer Done = 1
         }
      else
         if (!FRMOR)
         {
         Reset Error Count
         }
      }
else
   if (TXERR)
      {
      Increment Error Count
      Unmask ACK
      Unmask CHH
      Disable Channel
      }
else
```

```
   if (CHH)
      {
      Mask CHH
      if (Transfer Done or (Error_count == 3))
         {
         De-allocate Channel
         }
      else
         Re-initialize Channel (in next b_interval - 1 /Frame)
      }
   }
else
   if (ACK)
      {
      Reset Error Count
      Mask ACK
      }
```

- **中断 IN 事务**

  假设：

  – 应用程序要从奇数帧开始，每帧接收一个数据包（最大 1 个最大数据包大小）（传输大小 = 1024 字节）。

  – 接收 FIFO 可以保存至少一个最大数据包大小的数据包和每个数据包的两个状态字（1031 字节）。

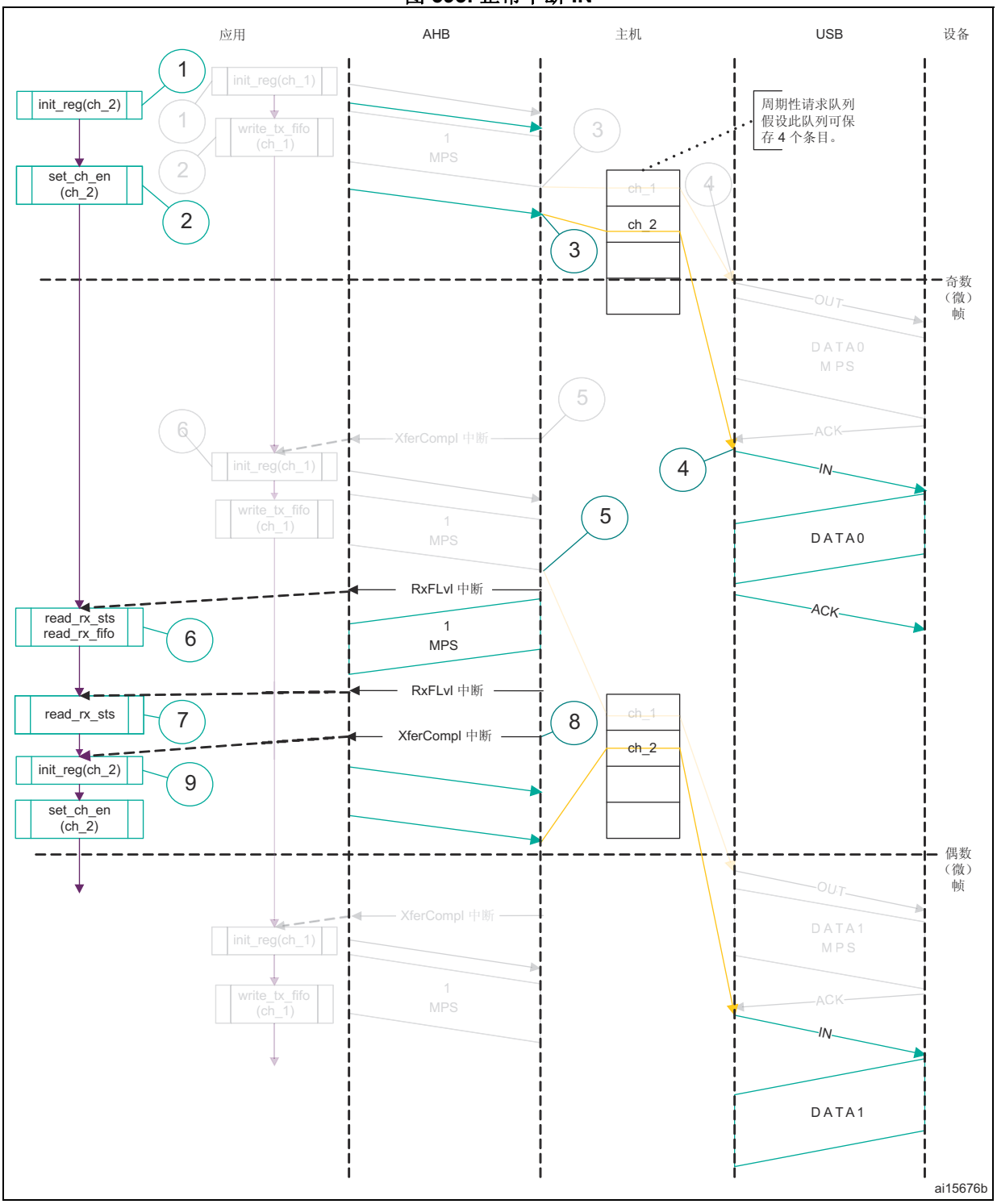
  – 周期性请求队列深度 = 4。

- **正常中断 IN 操作**

  操作顺序如下：

1. 初始化通道 2。应用程序必须将 OTG_HCCHAR2 中的 ODDFRM 位置 1。
2. 将 OTG_HCCHAR2 中的 CHENA 位置 1，以将 IN 请求写入周期性请求队列。
3. 只要 CHENA 置位，对于每次 OTG_HCCHAR2 寄存器写操作，OTG_FS 主机都会将一个 IN 请求写入周期性请求队列。
4. OTG_FS 主机尝试在下一奇数帧发送 IN 令牌。
5. 接收到 IN 数据包并写入接收 FIFO 后，OTG_FS 主机便会立即生成 RXFLVL 中断。
6. 为响应 RXFLVL 中断，读取接收到的数据包状态以确定接收的字节数，然后相应地读取接收 FIFO。应用程序必须在读取接收 FIFO 前屏蔽 RXFLVL 中断，并且在读取整个数据包后使能。
7. 模块针对接收 FIFO 中的传输完成状态条目生成 RXFLVL 中断。应用程序必须读取接收数据包状态，并在不是 IN 数据包（GRXSTSR 中的 PKTSTS ≠ 0b0010）时忽略它。
8. 读取接收数据包状态后，模块立即生成 XFRC 中断。
9. 为响应 XFRC 中断，将读取 OTG_HCTSIZ2 中的 PKTCNT 字段。如果 OTG_HCTSIZ2 中的 PKTCNT 位不等 0，则在重新初始化通道以进行下次传输前（如果存在），禁止该通道。如果 OTG_HCTSIZ2 中的 PKTCNT 位等于 0，则重新初始化通道以进行下次传输。此时，应用程序必须复位 OTG_HCCHAR2 中的 ODDFRM 位。

图 593. 正常中断 IN



1. 灰显元素与此图的上下文不相关。

- **同步 OUT 事务**

  图 593 中显示了典型的同步 OUT 操作。假设：

  – 应用程序尝试从奇数帧开始，每帧发送一个数据包（最大 1 个最大数据包大小）。（传输大小 = 1024 字节）。

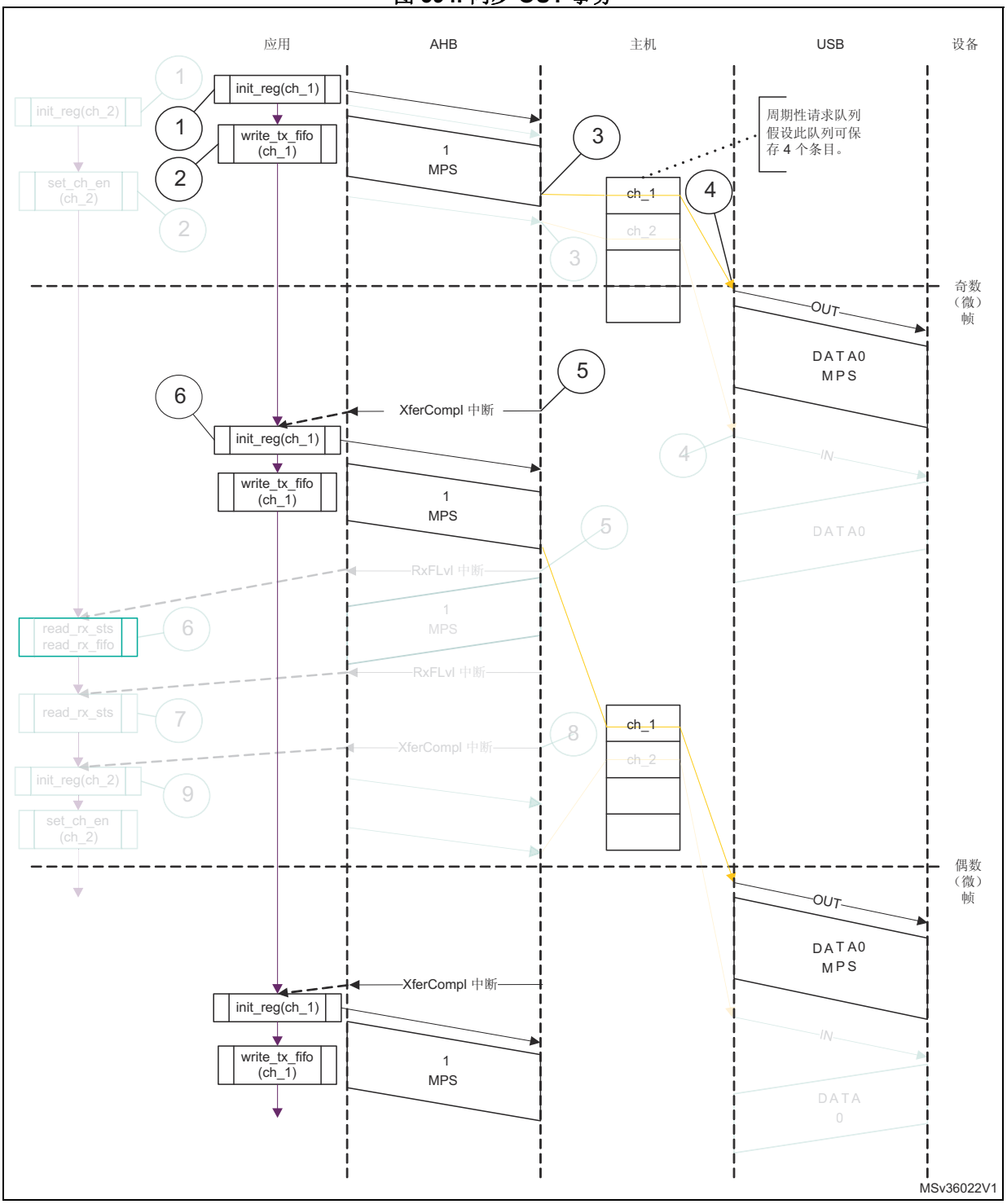
  – 周期性发送 FIFO 可存储一个数据包 (1 KB)。

  – 周期性请求队列深度 = 4。

  操作顺序如下：

1. 初始化和使能通道 1。应用程序必须将 OTG_HCCHAR1 中的 ODDFRM 位置 1。
2. 写入通道 1 的第一个数据包。
3. 应用程序在执行每个数据包的最后一次字写操作时，OTG_FS 主机向周期性请求队列写入一个请求条目。
4. OTG_FS 主机尝试在下一（奇数）帧发送 OUT 令牌。
5. 成功发送最后一个数据包后，OTG_FS 主机立即生成 XFRC 中断。
6. 为了响应 XFRC 中断，将重新初始化通道以供下一传输操作使用。
7. 处理非 ACK 响应。

图 594. 同步 OUT 事务



1. 灰显元素与此图的上下文不相关。

- **同步 OUT/IN 事务的中断服务程序**

  代码示例：同步 OUT

```
Unmask (FRMOR/XFRC)
if (XFRC)
   {
   De-allocate Channel
   }
else
   if (FRMOR)
      {
      Unmask CHH
      Disable Channel
      }
   else
   if (CHH)
      {
      Mask CHH
      De-allocate Channel
      }
```

代码示例：同步 IN

```
Unmask (TXERR/XFRC/FRMOR/BBERR)
if (XFRC or FRMOR)
   {
   if (XFRC and (OTG_HCTSIZx.PKTCNT == 0))
      {
      Reset Error Count
      De-allocate Channel
      }
   else
      {
      Unmask CHH
      Disable Channel
      }
   }
   else
      if (TXERR or BBERR)
         {
         Increment Error Count
         Unmask CHH
         Disable Channel
         }
      else
         if (CHH)
            {
            Mask CHH
```

```
if (Transfer Done or (Error_count == 3))
   {
   De-allocate Channel
   }
else
   {
   Re-initialize Channel
   }
}
```

- **同步 IN 事务**

  假设：

  – 应用程序尝试从下一个奇数帧开始，每帧接收一个数据包（最大 1 个最大数据包大小）（传输大小 = 1024 字节）。

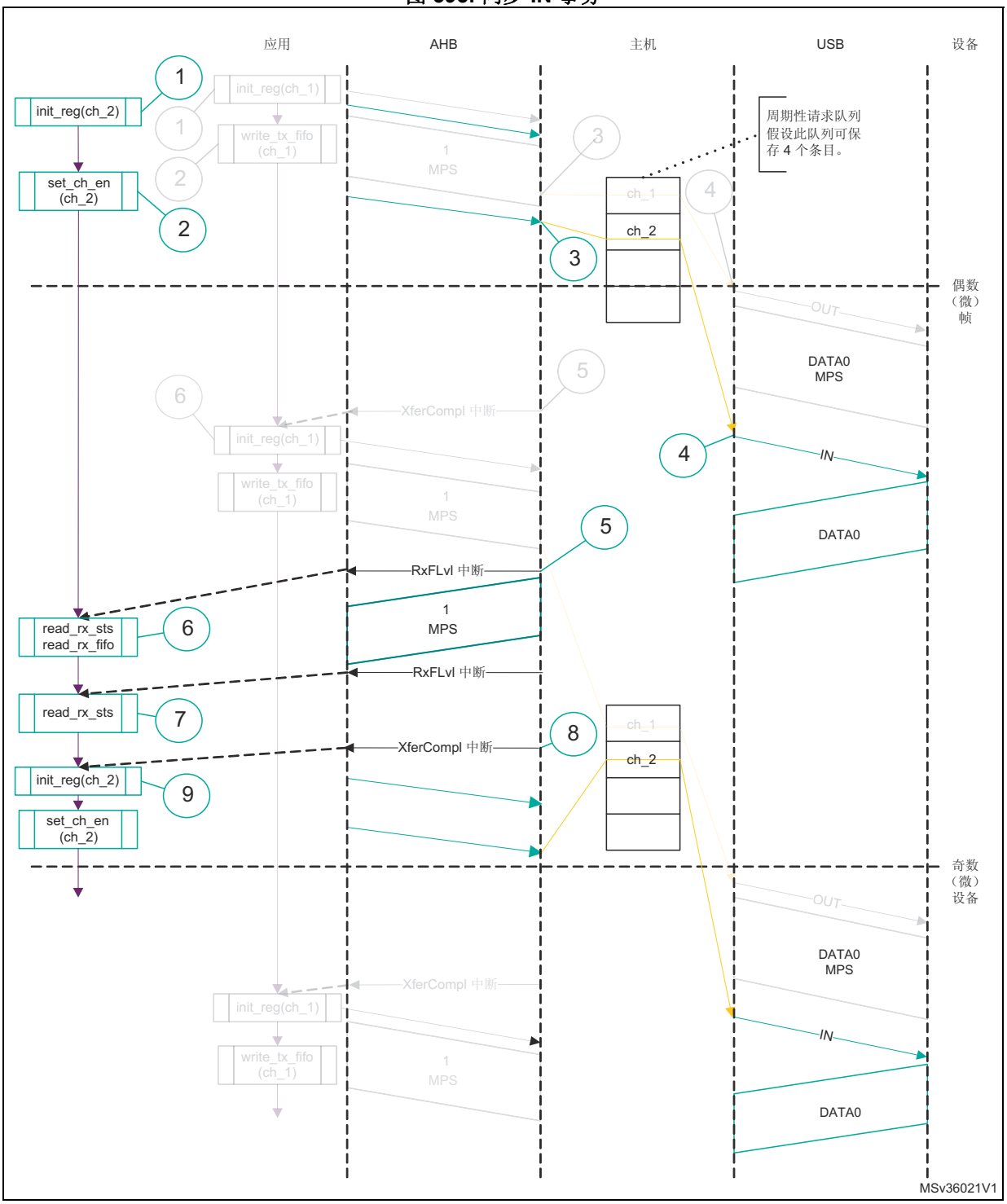
  – 接收 FIFO 可以保存至少一个最大数据包大小的数据包和每个数据包的两个状态字（1031 字节）。

  – 周期性请求队列深度 = 4。

  操作顺序如下：

1. 初始化通道 2。应用程序必须将 OTG_HCCHAR2 中的 ODDFRM 位置 1。
2. 将 OTG_HCCHAR2 中的 CHENA 位置 1，以将 IN 请求写入周期性请求队列。
3. 只要 CHENA 置位，对于每次 OTG_HCCHAR2 寄存器写操作，OTG_FS 主机都会将一个 IN 请求写入周期性请求队列。
4. OTG_FS 主机尝试在下一奇数帧发送 IN 令牌。
5. 接收到 IN 数据包并写入接收 FIFO 后，OTG_FS 主机便会立即生成 RXFLVL 中断。
6. 为响应 RXFLVL 中断，读取接收到的数据包状态以确定接收的字节数，然后相应地读取接收 FIFO。应用程序必须在读取接收 FIFO 前屏蔽 RXFLVL 中断，并且在读取整个数据包后使能。
7. 模块针对接收 FIFO 中的传输完成状态条目生成 RXFLVL 中断。应用程序必须读取接收数据包状态，并在不是 IN 数据包（OTG_GRXSTSR 中的 PKTSTS 位 ≠ 0b0010）时忽略它。
8. 读取接收数据包状态后，模块立即生成 XFRC 中断。
9. 为响应 XFRC 中断，将读取 OTG_HCTSIZ2 中的 PKTCNT 字段。如果在 OTG_HCTSIZ2 中的 PKTCNT ≠ 0，则在重新初始化通道以进行下次传输前（如果存在），禁止该通道。如果 OTG_HCTSIZ2 中的 PKTCNT = 0，则重新初始化通道以进行下次传输。此时，应用程序必须复位 OTG_HCCHAR2 中的 ODDFRM 位。

图 595. 同步 IN 事务



1. 灰显元素与此图的上下文不相关。

- **选择队列深度**

  请谨慎选择周期性和非周期性请求队列深度，以与要访问的周期性/非周期性端点数量相匹配。

  非周期性请求队列深度会影响非周期性传输的性能。队列越深（加上足够的 FIFO 大小），模块就更能对非周期性传输进行流水线处理。如果队列大小太小，则仅在队列空间释放时模块才能放入新请求。

  要按调度计划执行周期性传输，模块的周期性请求队列深度至关重要。根据一个微帧内要安排的周期性传输次数，选择周期性队列深度。如果周期性请求队列深度小于一个微帧内要安排的周期性传输数，则会发生帧溢出情况。

- **处理串扰情况**

  OTG_FS 控制器处理两种情况的 babble：数据包 babble 和端口 babble。如果设备发送的数据量超过通道的最大数据包大小，则会发生数据包串扰。如果模块从 EOF2（非常接近下一帧的 SOF）时刻还在从设备接收数据，则发生端口串扰。

  当 OTG_FS 控制器检测到数据包 babble 时，它停止向 Rx 缓冲区中写入数据并等待数据包结束信号 (EOP)。当该控制器检测到 EOP 时，它会清空 Rx 缓冲区中的已写入数据并对应用程序生成串扰中断。

  当 OTG_FS 控制器检测到端口 babble 时，它会清空 Rx FIFO 并禁止端口。模块随后将生成“端口已禁止”中断（OTG_GINTSTS 中的 HPRTINT 位和 OTG_HPRT 中的 PENCHNG 位）。在收到该中断时，应用程序必须通过检查 OTG_HPRT 中的 POCA 位，来确定该中断并非由过流 （“端口已禁止”中断的另一个原因）所致，然后执行软复位。检测到端口串扰情况后，模块不再发送任何其他令牌。

50.16.5 设备编程模型

USB 复位时的端点初始化

1. 为所有 OUT 端点将 NAK 位置 1
   - OTG_DOEPCTLx 中，SNAK = 1（对于所有 OUT 端点）
2. 使能以下中断位
   - OTG_DAINTMSK 中，INEP0 = 1（控制 0 IN 端点）
   - OTG_DAINTMSK 中，OUTEP0 = 1（控制 0 OUT 端点）
   - OTG_DOEPMSK 中的 STUPM = 1
   - OTG_DOEPMSK 中的 XFRCM = 1
   - OTG_DIEPMSK 中的 XFRCM = 1
   - OTG_DIEPMSK 中的 TOM = 1
3. 为每个 FIFO 设置数据 FIFO RAM
   - 对 OTG_GRXFSIZ 寄存器进行编程，以能够接收控制传输的 OUT 数据和设置数据。如果未使能阈值，则该寄存器必须至少等于控制端点 0 的 1 个最大数据包大小 + 2 个字（用于控制 OUT 数据包的状态）+ 10 个字（用于 SETUP 数据包）。
   - 对 OTG_DIEPTXF0 寄存器进行编程（取决于所选的 FIFO 编号），以能够发送控制 IN 数据。该寄存器至少必须等于控制端点 0 的 1 个最大数据包大小。
4. 对端点相关寄存器中的以下字段进行编程，以使控制 OUT 端点 0 接收 SETUP 数据包
   - OTG_DOEPTSIZ0 中的 STUPCNT = 3（接收最多 3 个连续的 SETUP 数据包）

此时，接收 SETUP 数据包所需的所有初始化工作便已完成。

枚举完成时的端点初始化

1. 在枚举完成中断（OTG_GINTSTS 中的 ENUMDNE）中，读取 OTG_DSTS 寄存器以确定设备的枚举速度。
2. 对 OTG_DIEPCTL0 中的 MPSIZ 字段进行编程以设置最大数据包大小。该步骤配置控制端点 0。控制端点的最大数据包大小取决于枚举速度。

此时，设备已准备好接收 SOF 数据包并配置为在控制端点 0 上执行控制传输。

收到 SetAddress 命令时的端点初始化

本节介绍了应用程序在 SETUP 数据包中接收到 SetAddress 命令时必须执行的操作。

1. 使用在 SetAddress 命令中接收到的设备地址来对 OTG_DCFG 寄存器进行编程
2. 对模块进行编程以发出状态阶段的 IN 数据包

收到 SetConfiguration/SetInterface 命令时的端点初始化

本节介绍了应用程序在 SETUP 包中接收 SetConfiguration 或 SetInterface 命令时必须执行的操作。

1. 接收到 SetConfiguration 命令时，应用程序必须对端点寄存器进行编程，以使用新配置中有效端点的特性来配置这些端点寄存器。
2. 接收到 SetInterface 命令时，应用程序必须对命令指定的端点的端点寄存器进行编程。
3. 在先前配置或其他设置中有效的端点在新的配置或其他设置中无效。必须停用这些无效端点。
4. 使用 OTG_DAINTMSK 寄存器使能有效端点的中断，屏蔽无效端点的中断。
5. 为每个 FIFO 设置数据 FIFO RAM。
6. 配置完所有必需的端点后，应用程序必须对模块进行编程以发送状态阶段的 IN 数据包。

此时，设备模块已可以接收和发送任何类型的数据包。

端点激活

本节介绍激活设备端点或者将现有设备端点配置为新类型所需的步骤。

1. 在 OTG_DIEPCTLx 寄存器（对于 IN 或双向端点）或 OTG_DOEPCTLx 寄存器（对于 OUT 或双向端点）的以下字段中，对所需端点的特性进行编程。
   - 最大数据包大小
   - USB 活动端点位置 1
   - 端点初始数据同步位（对于中断和批量端点）
   - 端点类型
   - Tx FIFO 编号
2. 激活端点后，模块便开始解码发送到该端点的令牌，并在收到的令牌有效的情况下回复有效握手信号。

端点停用

本节介绍停用现有端点所需的步骤。

1. 在要停用的端点中，将 OTG_DIEPCTLx 寄存器 （对于 IN 或双向端点）或 OTG_DOEPCTLx 寄存器 （对于 OUT 或双向端点）中的 USB 活动端点位清零。
2. 停用端点后，模块便会忽略发送到该端点的令牌，从而导致 USB 超时。

注： *应用程序必须满足以下条件才能设置设备模块以处理通信：*
必须将 OTG_GINTMSK 寄存器中的 NPTXFEM 和 RXFLVLM 清零。

操作模型

SETUP 和 OUT 数据传输：

本节介绍了数据 OUT 传输和 SETUP 事务期间的内部数据流和应用程序操作步骤。

- **数据包读取**

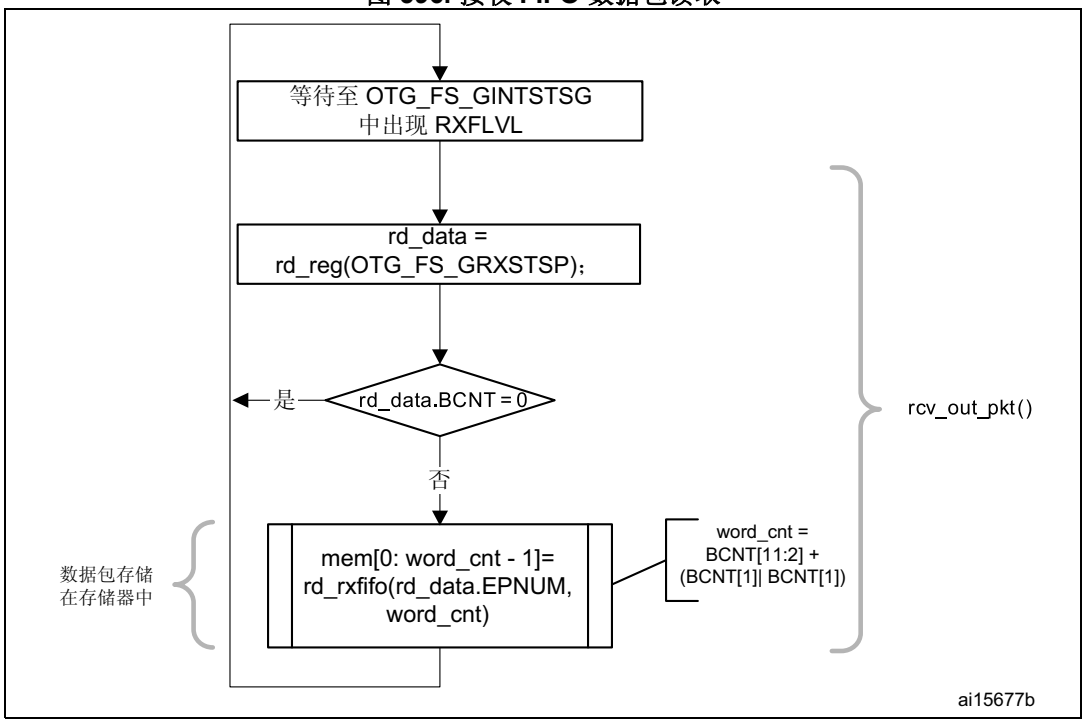
本节介绍如何从接收 FIFO 读取数据包（OUT 数据和 SETUP 数据包）。

1. 捕获到 RXFLVL 中断（OTG_GINTSTS 寄存器）时，应用程序必须读取接收状态弹出寄存器 (OTG_GRXSTSP)。
2. 应用程序可以通过写入 RXFLVLM = 0（在 OTG_GINTMSK 中）来屏蔽 RXFLVL 中断（在 OTG_GINTSTS 中），直到它把数据包从接收 FIFO 中读取出来。
3. 如果已接收数据包的字节计数不是 0，则从接收数据 FIFO 中弹出数据的字节计数并存储在存储器中。如果接收到的数据包字节计数为 0，则不会从接收数据 FIFO 中弹出任何数据。
4. 从接收 FIFO 读出的数据包状态有以下几种状态：
   a) 全局 OUT NAK：
   PKTSTS = 全局 OUT NAK，BCNT = 0x000，EPNUM = (0x0)，DPID = (0b00)。
   这些数据表示全局 OUT NAK 位已生效。
   b) SETUP 数据包：
   PKTSTS = SETUP，BCNT = 0x008，EPNUM = 控制端点编号，DPID = DATA0。
   这些数据表示指定端点上收到的 SETUP 数据包现在可从接收 FIFO 中读取。
   c) 建立阶段完成：
   PKTSTS = 建立阶段完成，BCNT = 0x0，EPNUM = 控制端点编号，DPID = (0b00)。
   这些数据表示指定端点的建立阶段完成并且数据阶段已启动。在此状态条目从接收 FIFO 中弹出后，模块将在指定控制 OUT 端点上产生建立中断。
   d) OUT 数据包：
   PKTSTS = DataOUT，BCNT = 接收的 OUT 数据包的大小 (0 ≤ BCNT ≤ 1 024)，EPNUM = 收到数据包的端点编号，DPID = 实际数据 PID。
   e) 数据传输完成：
   PKTSTS = OUT 数据传输完成，BCNT = 0x0，EPNUM = 完成数据传输的 OUT 端点编号，DPID = (0b00)。
   这些数据表示指定 OUT 端点的 OUT 数据传输完成。在此状态条目从接收 FIFO 中弹出后，模块将在指定的 OUT 端点上引发“传输完成”中断。

5. 从接收 FIFO 中弹出数据后，必须使能 RXFLVL 中断 (OTG_GINTSTS)。
6. 每次应用程序检测到 OTG_GINTSTS 中的 RXFLVL 中断时，都将重复步骤 1 到 5。读取空的接收 FIFO 可能导致未定义的模块行为。

图 596 提供了上述过程的流程图。

图 596. 接收 FIFO 数据包读取



SETUP 事务

本节介绍了模块处理 SETUP 数据包的方式以及应用程序处理 SETUP 事务的顺序。

- **应用程序要求**

1. 要接收 SETUP 数据包，必须将控制 OUT 端点中的 STUPCNT 字段 (OTG_DOEPTSIZx) 编程为非零值。如果应用程序将 STUPCNT 字段编程为非零值，模块会接收 SETUP 数据包并将其写入接收 FIFO，而不考虑 NAK 状态和 OTG_DOEPCTLx 中的 EPENA 位设置。控制端点每收到一个 SETUP 数据包后，STUPCNT 字段都会递减。如果在接收 SETUP 数据包之前，未将 STUPCNT 字段编程为适当值，模块仍能接收 SETUP 数据包并使 STUPCNT 字段递减，但应用程序可能无法确定在控制传输的建立阶段中接收的 SETUP 数据包的正确数量。
   - 在 OTG_DOEPTSIZx 中，STUPCNT = 3
2. 应用程序必须始终在接收数据 FIFO 中分配一些额外空间，以便能够在控制端点上接收最多三个 SETUP 数据包。
   - 预留空间 10 个字。第一个 SETUP 数据包需要 3 个字，“建立阶段完成”状态双字需要 1 个字，还需要 6 个字以在所有控制端点间存储两个额外的 SETUP 数据包。
   - 每个 SETUP 数据包需要 3 个字以存储 8 个字节的 SETUP 数据和 4 个字节的 SETUP 状态（SETUP 数据包模式）。模块会在接收数据 FIFO 中保留这些空间，仅写入 SETUP 数据，而不会将此空间用于其他数据包。
3. 应用程序必须从接收 FIFO 中读取 SETUP 数据包的 2 个字。
4. 应用程序必须从接收 FIFO 中读取并丢弃“建立阶段完成”状态字。

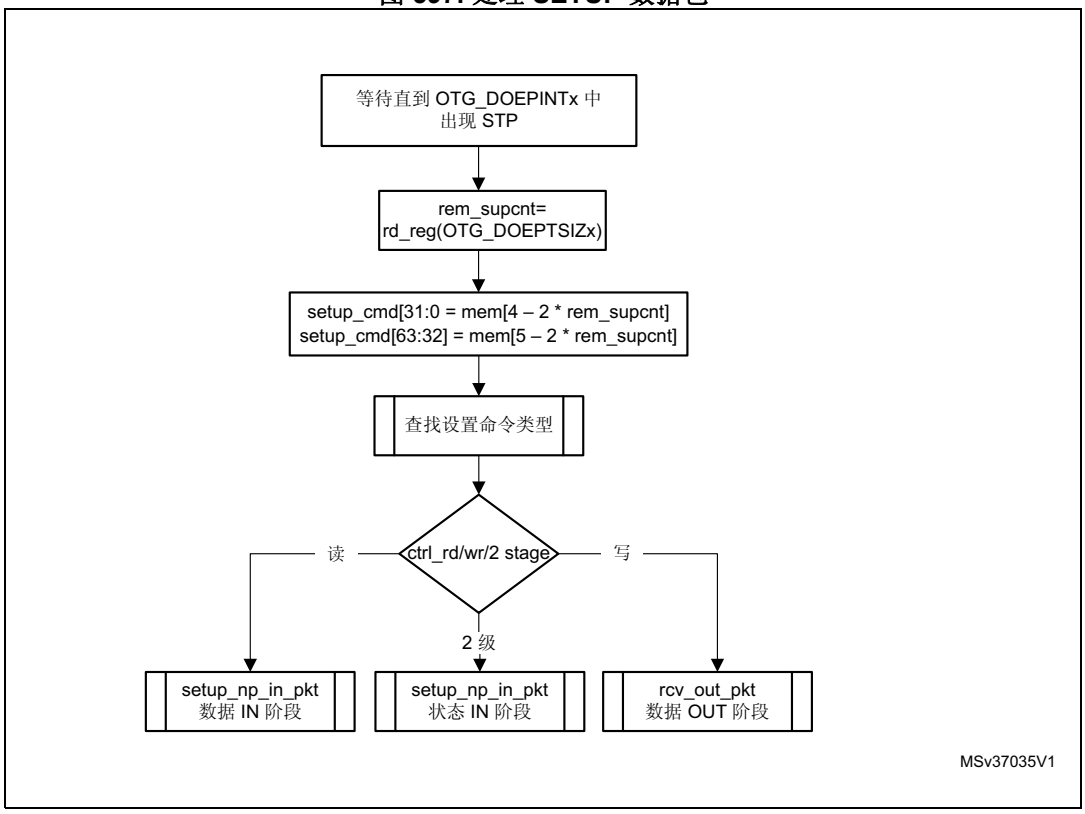
- **内部数据流**

1. 接收到 SETUP 数据包时，模块会将接收到的数据写入接收 FIFO，而不会检查接收 FIFO 中的可用空间，且不考虑端点的 NAK 和 STALL 位设置。
   - 模块会在内部将接收到 SETUP 数据包的控制 IN/OUT 端点的 IN NAK 和 OUT NAK 位置 1。
2. USB 上接收到的每个 SETUP 数据包，模块会将 3 个字的数据写入接收 FIFO，并且将 STUPCNT 字段递减 1。
   - 第一个字包含由模块内部使用的控制信息
   - 第二个字包含 SETUP 命令的前 4 个字节
   - 第三个字包含 SETUP 命令的最后 4 个字节
3. 当建立阶段结束，数据 IN/OUT 阶段开始时，模块会将一个状态条目（“建立阶段完成”字）写入接收 FIFO，指示建立阶段完成。
4. 在 AHB 端，SETUP 数据包被应用程序读取。
5. 当应用程序从接收 FIFO 中弹出“建立阶段完成”字时，模块将使用 STUP 中断 (OTG_DOEPINTx) 来中断应用程序，表示其可以处理接收到的 SETUP 数据包。
6. 模块会将控制 OUT 端点的端点使能位清零。

- **应用程序编程顺序**

1. 对 OTG_DOEPTSIZx 寄存器进行编程。
   – STUPCNT = 3
2. 等待 RXFLVL 中断 (OTG_GINTSTS) 并且从接收 FIFO 中读取数据包。
3. STUP 中断的触发 (OTG_DOEPINTx) 表示 SETUP 数据传输成功完成。
   – 发生该中断时，应用程序必须读取 OTG_DOEPTSIZx 寄存器以确定接收的 SETUP 数据包数量并处理最后接收的 SETUP 数据包。

图 597. 处理 SETUP 数据包



- **处理三个以上连续的 SETUP 数据包**

根据 USB 2.0 规范，在 SETUP 数据包错误中，主机通常不会向同一个端点发送 3 个以上连续的 SETUP 数据包。但是，USB 2.0 规范并未限制主机可以向同一个端点发送的连续 SETUP 数据包数量。出现这种情况时，OTG_FS 控制器将生成中断（OTG_DOEPINTx 中的 B2BSTUP）。

- **将全局 OUT NAK 置 1**

内部数据流：

1. 如果应用程序将全局 OUT NAK（OTG_DCTL 中的 SGONAK 位）置 1，模块将停止向接收 FIFO 中写入 SETUP 数据包以外的数据。无论接收 FIFO 中可用空间大小如何，设备都会对主机发送的非同步 OUT 令牌回复 NAK 握手，而对同步 OUT 数据包直接予以忽略。
2. 模块将全局 OUT NAK 写入接收 FIFO。应用程序必须为此留出足够空间。
3. 当应用程序从接收 FIFO 中弹出全局 OUT NAK 字时，模块会将 GONAKEFF 中断 (OTG_GINTSTS) 置 1。
4. 应用程序检测到该中断后，会认为模块处于全局 OUT NAK 模式。应用程序可以通过将 OTG_DCTL 中的 SGONAK 位清零来清除该中断。

应用程序编程顺序：

1. 要停止接收任何类型的数据到接收 FIFO 中，应用程序必须通过编程以下字段以将全局 OUT NAK 位置 1。
   – 在 OTG_DCTL 中，SGONAK = 1
2. 等待 OTG_GINTSTS 中的 GONAKEFF 中断。一旦被触发，该中断表示模块已停止接收 SETUP 数据包以外的任何类型数据。
3. 如果应用程序已将 OTG_DCTL 中的 SGONAK 位置 1，则在模块引发 GONAKEFF 中断 (OTG_GINTSTS) 之前，应用程序可以接收有效 OUT 数据包。
4. 应用程序可通过对 OTG_GINTMSK 寄存器中的 GONAKEFFM 位执行写操作来暂时屏蔽此中断。
   – 在 OTG_GINTMSK 寄存器中，GONAKEFFM = 0
5. 当应用程序准备退出全局 OUT NAK 模式时，必须将 OTG_DCTL 中的 SGONAK 位清零。此操作还会清除 GONAKEFF 中断 (OTG_GINTSTS)。
   – 在 OTG_DCTL 中，CGONAK = 1
6. 如果应用程序在之前已屏蔽此中断，则必须按以下方式使能该中断：
   – 在 OTG_GINTMSK 中，GONAKEFFM = 1

- **禁止 OUT 端点**

应用程序必须使用以下顺序禁止已使能的 OUT 端点。

应用程序编程顺序：

1. 禁止任何 OUT 端点前，应用程序必须在模块中使能全局 OUT NAK 模式。
   – 在 OTG_DCTL 中，SGONAK = 1
2. 等待 GONAKEFF 中断 (OTG_GINTSTS)
3. 通过编程以下字段来禁止 OUT 端点：
   – 在 OTG_DOEPCTLx 中，EPDIS = 1
   – 在 OTG_DOEPCTLx 中，SNAK = 1
4. 等待 EPDISD 中断 (OTG_DOEPINTx)，该中断表示已完全禁止 OUT 端点。引发 EPDISD 中断时，模块还会将以下位清零：
   – 在 OTG_DOEPCTLx 中，EPDIS = 0
   – 在 OTG_DOEPCTLx 中，EPENA = 0
5. 应用程序必须将全局 OUT NAK 位清零，以开始从其他未禁止的 OUT 端点接收数据。
   – 在 OTG_DCTL 中，SGONAK = 0

- **通用非同步 OUT 数据传输**

本节介绍一种常规非同步 OUT 数据传输（控制、批量或中断）。

应用程序要求：

1. 建立 OUT 传输前，应用程序必须在存储器中分配一个缓冲区，以容纳要作为 OUT 传输的一部分而接收的所有数据。
2. 对于 OUT 传输，端点的传输大小寄存器中的传输大小字段必须是端点的最大数据包大小的倍数（且以字对齐）。
   – 传输大小[EPNUM] = *n* × (MPSIZ[EPNUM] + 4 – (MPSIZ[EPNUM] mod 4))
   – 数据包计数[EPNUM] = *n*
   – *n* > 0
3. 发生 OUT 端点中断时，应用程序必须读取端点的传输大小寄存器以计算存储器中有效数据量。接收的有效数据量可能小于编程的传输大小。
   – 存储器中的有效数据量 = 应用程序设置的初始传输量 – 模块更新后的剩余传输量
   – 接收到 USB 数据包数 = 应用程序设置的初始数据包数 – 模块更新后的剩余数据包数

内部数据流：

1. 应用程序必须在端点相关寄存器中设置传输大小和数据包计数字段，将 NAK 位清零，并使能端点来接收数据。
2. NAK 位清零后，模块便开始接收数据并将数据写入接收 FIFO（只要接收 FIFO 中有空间）。对于 USB 上接收的每个数据包，数据包及其状态都会写入接收 FIFO。写入接收 FIFO 的每个数据包 （数据量达到最大数据包大小的数据包或短数据包）都会使该端点的数据包计数字段递减 1。
   – 收到的数据包若 CRC 无效，则自动被从接收 FIFO 中清除。
   – 在 USB 上为数据包回复 ACK 后，模块将丢弃主机因无法检测到 ACK 而重新发送的非同步 OUT 数据包。应用程序不会在具有相同数据 PID 的相同端点上检测到多个连续的 OUT 数据包。在这种情况下，数据包计数不会递减。
   – 如果接收 FIFO 中没有空间，则会忽略同步或非同步数据包并且不会将它们写入接收 FIFO。此外，非同步 OUT 令牌将会收到 NAK 握手应答。
   – 在上述所有三种情况中，数据包计数都不会递减，因为没有任何数据写入接收 FIFO。
3. 当数据包计数变为 0 或者在端点上接收到短数据包时，该端点的 NAK 位将置 1。NAK 位置 1 后，将忽略同步或非同步数据包并且不会将它们写入接收 FIFO，同时非同步 OUT 令牌会收到 NAK 握手应答。
4. 在数据写入接收 FIFO 后，应用程序将从接收 FIFO 中读取数据并将数据写入外部存储器，一次一个数据包，逐个端点过来。
5. 在 AHB 上向外部存储器写入完每个数据包后，端点的传输大小都会自动减去该数据包的大小。
6. 在以下情况时，OUT 端点的 OUT 数据传输完成状态将写入接收 FIFO：
   – 传输大小为 0 并且数据包计数为 0
   – 写入接收 FIFO 的最后一个 OUT 数据包是短数据包
   （0 ≤ 数据包大小 < 最大数据包大小）
7. 当应用程序弹出此状态条目（OUT 数据传输完成），并生成该端点的传输完成中断，同时清零端点使能位。

应用程序编程顺序：

1. 使用传输大小和相应数据包个数对 OTG_DOEPTSIZx 寄存器进行编程。
2. 使用端点特性对 OTG_DOEPCTLx 寄存器进行编程，并将 EPENA 和 CNAK 位置 1。
   – 在 OTG_DOEPCTLx 中，EPENA = 1
   – 在 OTG_DOEPCTLx 中，CNAK = 1
3. 等待 RXFLVL 中断（在 OTG_GINTSTS 中）并且从接收 FIFO 中读走数据包。
   – 此步骤可重复多次，具体取决于传输大小。
4. 触发 XFRC 中断 (OTG_DOEPINTx)，以表示非同步 OUT 数据传输成功完成。
5. 读取 OTG_DOEPTSIZx 寄存器，以确定有效数据量。

- **通用同步 OUT 数据传输**

本节介绍常规的同步 OUT 数据传输。

应用程序要求：

1. 非同步 OUT 数据传输的所有应用程序要求均适用于同步 OUT 数据传输。
2. 对于同步 OUT 数据传输中的传输大小和数据包计数字段，必须始终将其设置为单个帧中可接收的最大数据包大小的数据包数目。同步类型的 OUT 数据传输事务必须在一个帧内完成。
3. 在周期性帧结束（OTG_GINTSTS 中的 EOPF 中断）之前，应用程序必须从接收 FIFO 中读取所有同步 OUT 数据包（数据条目和状态条目）。
4. 要接收下一帧中的数据，必须在 EOPF (OTG_GINTSTS) 之后 SOF (OTG_GINTSTS) 之前使能一个同步 OUT 端点。

内部数据流：

1. 同步 OUT 端点的内部数据流与非同步 OTU 端点的基本相同，但稍有差异。
2. 同步 OUT 端点通过将端点使能位置 1 并将 NAK 位清零来使能时，必须相应地将偶数/奇数帧位置 1。仅当符合以下条件时，模块才会在同步 OUT 端点上接收特定帧中的数据：
   – EONUM （在 OTG_DOEPCTLx 中）= FNSOF[0]（在 OTG_DSTS 中）
3. 当应用程序从接收 FIFO 中完整地读取一个同步 OUT 数据包（数据和状态）后，模块会根据从接收 FIFO 中读取的最后一个同步 OUT 数据包的数据 PID 更新 OTG_DOEPTSIZx 中的 RXDPID 字段。

应用程序编程顺序：

1. 使用传输大小和相应数据包个数对 OTG_DOEPTSIZx 寄存器进行编程
2. 使用端点特性对 OTG_DOEPCTLx 寄存器进行编程，并将端点使能位、清除 NAK 位和奇数/偶数帧位置 1。
   – EPENA = 1
   – CNAK = 1
   – EONUM =（0：偶数/1：奇数）
3. 等待 RXFLVL 中断（在 OTG_GINTSTS 中）并且从接收 FIFO 中读走数据包
   – 此步骤可重复多次，具体取决于传输大小。
4. XFRC 中断（在 OTG_DOEPINTx 中）表示同步 OUT 数据传输完成。该中断不一定意味着存储器中的数据是有效的。
5. 对于同步 OUT 传输，应用程序可能并不总会检测到该中断。而是可以检测到 OTG_GINTSTS 中的 INCOMPISOOUT 中断。

6. 读取 OTG_DOEPTSIZx 寄存器以确定接收的传输大小以及帧中所接收数据的有效性。仅当符合以下条件之一时，应用程序才必须将存储器中接收的数据视为有效数据：
   – RXDPID = DATA0（在 OTG_DOEPTSIZx 中）并且接收该有效数据的 USB 数据包数量 = 1
   – RXDPID = DATA1（在 OTG_DOEPTSIZx 中）并且接收该有效数据的 USB 数据包数量 = 2

   接收该有效数据的 USB 数据包数量 =
   应用程序编程的初始数据包个数 – 模块更新后的剩余数据包个数

   应用程序可将无效数据包丢弃。

- **不完整的同步 OUT 数据传输**

本节介绍了同步 OUT 数据包出现丢包时应用程序编程顺序。

内部数据流：

1. 对于同步 OUT 端点，可能并不总是触发 XFRC 中断（在 OTG_DOEPINTx 中）。如果模块丢弃同步 OUT 数据包，则在以下情况下，应用程序可能无法检测到 XFRC 中断 (OTG_DOEPINTx)：
   – 在接收 FIFO 无法容纳完整的 ISO OUT 数据包时，模块将丢弃接收到的 ISO OUT 数据
   – 接收到的同步 OUT 数据包存在 CRC 错误
   – 模块接收到的同步 OUT 令牌损坏
   – 应用程序从接收 FIFO 中读取数据的速度非常缓慢
2. 如果模块在所有同步 OUT 端点的传输完成前检测到周期性帧结束，将触发未完成同步 OUT 数据中断（OTG_GINTSTS 中的 INCOMPISOOUT），指示至少有一个同步 OUT 端点上未触发 XFRC 中断（在 OTG_DOEPINTx 中）。此时，未完成传输的端点仍保持使能，但在 USB 的该端点上，没有进行中的有效传输。

应用程序编程顺序：

1. 硬件触发 INCOMPISOOUT 中断 (OTG_GINTSTS) 表示当前帧中至少有一个同步 OUT 端点具有未完成的传输。
2. 如果因未从端点完全读取同步 OUT 数据而发生这种情况，应用程序必须确保首先从接收 FIFO 读取走所有同步 OUT 数据（包括数据条目和状态条目），然后再继续处理。
   – 从接收 FIFO 读取所有数据后，应用程序即可检测到 XFRC 中断 (OTG_DOEPINTx)。在此情况下，应用程序必须重新使能端点以接收下一个帧中的同步 OUT 数据。
3. 当应用程序接收到 INCOMPISOOUT 中断（在 OTG_GINTSTS 中）时，应用程序必须读取所有同步 OUT 端点的控制寄存器 (OTG_DOEPCTLx)，以确定哪些端点在当前微帧中具有不完整的传输。同时满足以下两个条件时，表示端点传输未完成：
   – EONUM 位（在 OTG_DOEPCTLx 中）= FNSOF[0]（在 OTG_DSTS 中）
   – EPENA = 1（在 OTG_DOEPCTLx 中）
4. 在检测到 SOF 中断（在 OTG_GINTSTS 中）前，必须执行完成上一步操作，以确保当前帧编号未发生更改。
5. 对于具有未完成传输的同步 OUT 端点，应用程序必须丢弃存储器中的数据，并通过将 OTG_DOEPCTLx 中的 EPDIS 位置 1 来禁止该端点。
6. 等待 EPDISD 中断（在 OTG_DOEPINTx 中），并使能该端点，以便接收下一个帧中的新数据。
   – 由于模块可能需要一些时间才能禁止端点，因此应用程序在接收到无效同步数据后，可能无法接收下一个帧中的数据。

- **停止非同步 OUT 端点**

本节介绍应用程序如何才能停止非同步端点。

1. 将模块置于全局 OUT NAK 模式。
2. 禁止所需的端点
   – 禁止端点时，请设置 STALL = 1（在 OTG_DOEPCTL 中），而不是将 OTG_DOEPCTL 中的 SNAK 位置 1。

     STALL 位的优先级始终高于 NAK 位。
3. 当应用程序不再需要端点回复 STALL 握手信号时，必须将 STALL 位（在 OTG_DOEPCTLx 中）清零。
4. 如果应用程序由于收到主机的 SetFeature.Endpoint Halt 或 ClearFeature.Endpoint Halt 命令来设置或清除端点的 STALL 状态，则必须在应用程序建立该控制端点上的状态阶段传输前，将 STALL 位置 1 或清零。

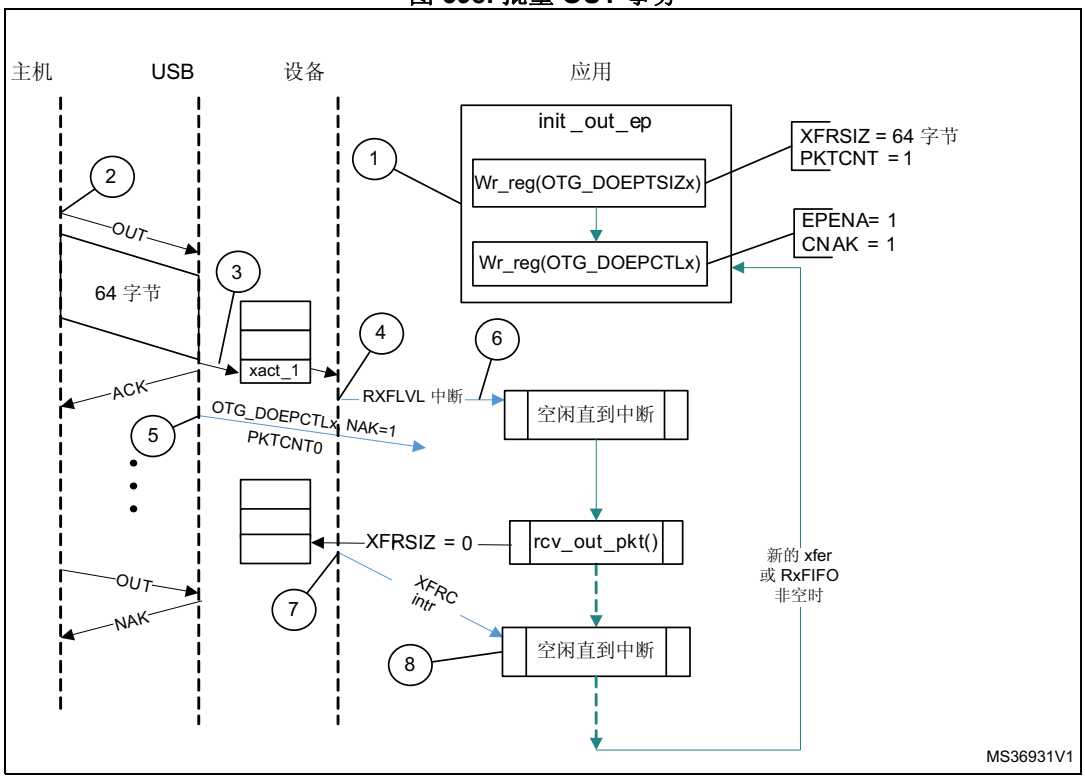
示例

本节介绍并描述了一些基本的传输类型和情景。

- 批量 OUT 事务

图 598 描述了将单个批量 OUT 数据包从 USB 接收到 AHB 中的过程，并介绍了这一过程中涉及到的事件。

图 598. 批量 OUT 事务



接收到 SetConfiguration/SetInterface 命令后，应用程序将初始化所有 OUT 端点：将 CNAK 和 EPENA 置 1（在 OTG_DOEPCTLx 中），并将 OTG_DOEPTSIZx 寄存器中的 XFRSIZ 和 PKTCNT 设置成合适的值。

1. 主机尝试将数据（OUT 令牌）发送到端点。
2. 当模块在 USB 上接收到 OUT 令牌时，模块会将数据包存储在 Rx FIFO 中，因为其中有可用空间。
3. 在 Rx FIFO 中写入完整数据包后，模块随后会触发 RXFLVL 中断（在 OTG_GINTSTS 中）。
4. 接收到 PKTCNT 所指定数量的 USB 数据包后，模块在内部将该端点的 NAK 位置 1，以避免其再接收任何数据包。
5. 应用程序处理中断并从 Rx FIFO 中读取数据。
6. 应用程序读取完所有数据后（相当于 XFRSIZ），模块将生成 XFRC 中断（在 OTG_DOEPINTx 中）。
7. 应用程序处理中断并通过 XFRC 中断位（在 OTG_DOEPINTx 中）的触发得知本次传输完成。

IN 数据传输

- **数据包写入**

本节介绍在已使能专用发送 FIFO 的情况下应用程序如何将数据包写入端点 FIFO。

1. 应用程序可以选择轮询模式或中断模式。
   - 在轮询模式下，应用程序通过读取 OTG_DTXFSTSx 寄存器来监视端点发送数据 FIFO 的状态，从而确定该数据 FIFO 中是否有足够空间。
   - 在中断模式中，应用程序等待 TXFE 中断（在 OTG_DIEPINTx 中），然后读取 OTG_DTXFSTSx 寄存器，以确定数据 FIFO 中是否有足够空间。
   - 要写入单个非零长度的数据包，数据 FIFO 中必须有足够的空间来容纳整个数据包。
   - 要写入零长度的数据包，应用程序不能查看 FIFO 空间。
2. 如果使用上述方法之一，当应用程序确定有足够空间来写入发送数据包时，应用程序必须首先对端点控制寄存器进行相应写操作，然后再将数据写入数据 FIFO。通常，应用程序必须对 OTG_DIEPCTLx 寄存器执行读-修改-写操作，以避免在将端点使能位置 1 的同时，修改寄存器中的其他内容。

如果有足够空间，应用程序可将同一端点的多个数据包写入发送 FIFO。对于周期性 IN 端点，应用程序只能一次写入一个微帧内的多个数据包。只有先前一个微帧的通信事务传输完成之后，应用程序才会写入下一个微帧内要发送的所有数据包。

- **将 IN 端点 NAK 置 1**

内部数据流：

1. 当应用程序将特定端点的 IN NAK 置 1 时，模块将停止端点上的数据发送，而不考虑端点发送 FIFO 中的数据是否可用。
2. 非同步端点收到 IN 令牌，回复 NAK 握手应答
   - 同步端点收到 IN 令牌，返回零长度数据包
3. 模块在 OTG_DIEPINTx 中触发 INEPNE（IN 端点 NAK 有效）中断以响应 OTG_DIEPCTLx 中的 SNAK 位。
4. 应用程序检测到该中断后，便会认为端点处于 IN NAK 模式。应用程序可以通过将 OTG_DIEPCTLx 中的 CNAK 位置 1 来清除该中断。

应用程序编程顺序：

1. 要在特定 IN 端点上停止发送任何数据，应用程序必须将 IN NAK 位置 1。要将该位置 1，必须编程以下字段。
   – 在 OTG_DIEPCTLx 中，SNAK = 1
2. 等待 OTG_DIEPINTx 中的 INEPNE 中断触发。该中断表示模块已在端点上停止发送数据。
3. 在应用程序将 NAK 位置 1 但“NAK 有效”中断尚未触发时，模块可以在端点上发送有效 IN 数据。
4. 应用程序可通过写入 OTG_DIEPMSK 中的 INEPNEM 位来临时屏蔽该中断。
   – 在 OTG_DIEPMSK 中，INEPNEM = 0
5. 要退出端点 NAK 模式，应用程序必须将 OTG_DIEPCTLx 中的 NAK 状态位 (NAKSTS) 清零。此操作还会将 INEPNE 中断位（在 OTG_DIEPINTx 中）清零。
   – 在 OTG_DIEPCTLx 中，CNAK = 1
6. 如果应用程序已将该中断屏蔽，则必须按以下方式使能：
   – 在 OTG_DIEPMSK 中，INEPNEM = 1

- **禁止 IN 端点**

使用以下顺序来禁止先前已使能的特定 IN 端点。

应用程序编程顺序：

1. 应用程序必须先停止在 AHB 上写入数据，之后才能禁止 IN 端点。
2. 应用程序必须将端点设置为 NAK 模式。
   – 在 OTG_DIEPCTLx 中，SNAK = 1
3. 等待 OTG_DIEPINTx 中的 INEPNE 中断。
4. 将必须禁止的端点的 OTG_DIEPCTLx 寄存器中的以下位置 1。
   – 在 OTG_DIEPCTLx 中，EPDIS = 1
   – 在 OTG_DIEPCTLx 中，SNAK = 1
5. OTG_DIEPINTx 中的 EPDISD 中断的触发表示模块已完全禁止指定的端点。在触发中断的同时，模块还会将以下位清零：
   – 在 OTG_DIEPCTLx 中，EPENA = 0
   – 在 OTG_DIEPCTLx 中，EPDIS = 0
6. 应用程序必须读取周期性 IN EP 的 OTG_DIEPTSIZx 寄存器，以计算该端点在 USB 上发送的数据量。
7. 应用程序必须通过将 OTG_GRSTCTL 寄存器中的以下字段置 1，来清空端点发送 FIFO 中的数据：
   – TXFNUM（在 OTG_GRSTCTL 中）= 端点发送 FIFO 编号
   – TXFFLSH（在 OTG_GRSTCTL 中）= 1

应用程序必须轮询 OTG_GRSTCTL 寄存器，直至模块将 TXFFLSH 位清零，这表示 FIFO 清空操作结束。要在该端点上发送新数据，应用程序可以稍后重新使能该端点。

- **通用非周期性 IN 数据传输**

应用程序要求：

1. 建立 IN 传输前，应用程序必须确保组成一次 IN 传输的每个数据包都可以容纳在单个缓冲区中。
2. 对于 IN 传输，端点传输大小寄存器中的传输大小字段表示本次传输的有效数据量，它由多个最大数据包大小的数据包和单个短数据包组成。该短数据包在传输结束时发送。
   – 要发送多个最大数据包大小的数据包并在传输结束时外加一个短数据包：
     传输大小[EPNUM] = *x* × MPSIZ[EPNUM] + sp
     如果 (sp > 0)，数据包计数[EPNUM] = x + 1
     否则，数据包计数[EPNUM] = *x*
   – 要发送单个零长度数据包：
     传输大小[EPNUM] = 0
     数据包计数[EPNUM] = 1
   – 要发送多个最大数据包大小的数据包并在传输结束时外加一个零长度数据包，应用程序必须将传输拆分为两个部分。第一部分发送最大数据包大小的数据包，第二部分仅发送零长度数据包。
     第一次传输：传输大小[EPNUM] = *x* × MPSIZ[epnum]；数据包计数 = *n*；
     第二次传输：传输大小[EPNUM] = 0；数据包计数 = 1；
3. 使能某个端点进行数据传输后，模块会更新传输大小寄存器。在 IN 传输结束时，应用程序必须读取传输大小寄存器，以确定送入发送 FIFO 中的数据已有多少通过 USB 发送出去。
4. 送入发送 FIFO 中的数据量 = 应用程序编程的初始传输大小 - 模块更新后的最终传输大小
   – 已经通过 USB 发送的数据量 = (应用程序编程的初始数据包个数 – 模块更新后的剩余数据包个数) × MPSIZ[EPNUM]
   – 要通过 USB 发送的剩余数据量 = (应用程序编程的初始传输大小 – 已通过 USB 发送的数据量)

内部数据流：

1. 应用程序必须在特定端点的寄存器中设置传输大小和数据包计数字段，并使能该端点来发送数据。
2. 应用程序还必须向该端点的发送 FIFO 写入必需的数据。
3. 应用程序每向发送 FIFO 写入一个数据包，该端点的传输大小便会自动减去该数据包大小。应用程序持续从存储器获取数据来写入发送 FIFO，直到该端点的传输大小变为 0。向 FIFO 写入数据后，“FIFO 中的数据包数”计数会递增 （这是一个 3 位计数，由模块在内部进行维护，每个 IN 端点发送 FIFO 对应一个。在 IN 端点 FIFO 中，模块所维护的最大数据包数始终为八个） 。对于零长度数据包，每个 FIFO 均另有一个单独的标志，FIFO 中没有任何数据。
4. 当数据写入发送 FIFO 后，模块会在接收到 IN 令牌时将这些数据送出。每个非同步 IN 数据包发送出去并收到回复的 ACK 握手信号后，该端点的数据包计数都会递减 1，直到数据包计数变 0 为止。发生超时时，数据包计数不会递减。
5. 对于零长度数据包（由内部零长度标志指示），模块会针对 IN 令牌发出一个零长度数据包，并递减数据包计数字段的值。
6. 如果接收到 IN 令牌的端点对应的 FIFO 中无数据，且该端点的数据包计数字段为零，则模块会针对该端点生成一个“Tx FIFO 为空时接收到 IN 令牌”(ITTXFE) 中断（前提是该端点的 NAK 位未置 1）。模块在该非同步端点上回复 NAK 握手信号。

7. 模块会在内部使 FIFO 指针重新返回到开头，并且不会生成超时中断。
8. 当传输大小为 0 且数据包计数为 0 时，将生成该端点的传输完成 (XFRC) 中断，同时将端点使能清零。

应用程序编程顺序：

1. 使用传输大小和相应数据包计数对 OTG_DIEPTSIZx 寄存器进行编程。
2. 使用端点特性对 OTG_DIEPCTLx 寄存器进行编程，并将 CNAK 和 EPENA（端点使能）位置 1。
3. 发送非零长度数据包时，应用程序必须轮询 OTG_DTXFSTSx 寄存器（其中 x 为与该端点相关联的 FIFO 编号），以确定数据 FIFO 中是否有足够的空间。写入数据前，应用程序可以选择使用 TXFE（在 OTG_DIEPINTx 中）。

- **通用周期性 IN 数据传输**

本节介绍典型的周期性 IN 数据传输。

应用程序要求：

1. *第 1914 页的通用非周期性 IN 数据传输*的应用程序要求 1、2、3 和 4 对周期性 IN 数据传输同样适用（只是对要求 2 稍加修改）。
   - 应用程序只能发送若干个最大数据包大小的数据包或若干个最大数据包大小的包，外加传输结束时的一个短数据包。要发送多个最大数据包大小的数据包并在传输结束时外加一个短数据包，必须满足以下条件：

     传输大小[EPNUM] = x × MPSIZ[EPNUM] + sp
     （其中 *x* 是 ≥ 0 的整数，且 0 ≤ sp < MPSIZ[EPNUM]）

     如果 (sp > 0)，数据包计数[EPNUM] = *x* + 1
     否则，数据包计数[EPNUM] = *x*；

     MCNT[EPNUM] = 数据包计数[EPNUM]
   - 应用程序无法在传输结束时发送零长度数据包。应用程序可以单独发送一个零长度数据包。要发送单个零长度数据包：
   - 传输大小[EPNUM] = 0

     数据包计数[EPNUM] = 1

     MCNT[EPNUM] = 数据包计数[EPNUM]
2. 应用程序一次只能安排一帧的数据传输。
   - (MCNT – 1) × MPSIZ ≤ XFERSIZ ≤ MCNT × MPSIZ
   - PKTCNT = MCNT（在 OTG_DIEPTSIZx 中）
   - 如果 XFERSIZ < MCNT × MPSIZ，则传输的最后一个数据包为短数据包。
   - 注意：MCNT 在 OTG_DIEPTSIZx 中，MPSIZ 在 OTG_DIEPCTLx 中，PKTCNT 在 OTG_DIEPTSIZx 中，而 XFERSIZ 在 OTG_DIEPTSIZx 中
3. 接收到 IN 令牌前，应用程序必须将要在帧中发送的完整数据写入到发送 FIFO 中。在接收到 IN 令牌时，即使发送 FIFO 中该帧要发送的数据只差 1 个双字未写进来，模块也会执行 FIFO 为空时的操作。当发送 FIFO 为空时：
   - 同步端点上将回复零长度数据包
   - 中断端点上将回复 NAK 握手信号

内部数据流：

1. 应用程序必须在特定端点的寄存器中设置传输大小和数据包计数字段，并使能该端点来发送数据。
2. 应用程序还必须向与该端点相关联的发送 FIFO 写入必需的数据。
3. 应用程序每向发送 FIFO 写入一个数据包，该端点的传输大小便会自动减去该数据包大小。应用程序持续从存储器获取数据来写入发送 FIFO，直到该端点的传输大小变为 0。
4. 当周期性端点接收到 IN 令牌时，模块将开始发送 FIFO 中的数据（如果 FIFO 中有数据）。如果 FIFO 中没有该帧要发送的数据的完整数据包，则模块将为该端点生成一个 “TxFIFO 为空时接收到 IN 令牌”中断。
   – 同步端点上将回复零长度数据包
   – 中断端点上将回复 NAK 握手信号
5. 端点的数据包计数会在下列情况下递减 1：
   – 对于同步端点，发送一个零长度或非零长度的数据包时
   – 对于中断端点，在发送 ACK 握手信号时递减
   – 当传输大小和数据包计数均为 0 时，将生成该端点的传输完成中断，同时将端点使能位清零
6. 在“周期性帧间隔”（由 OTG_DCFG 中的 PFIVL 位控制）内，当模块发现任何在当前帧内应为空的同步 IN 端点 FIFO 中的数据还未发送完成时，都会在 OTG_GINTSTS 中生成一个 IISOIXFR 中断。

应用程序编程顺序：

1. 使用端点特性对 OTG_DIEPCTLx 寄存器进行编程，并将 CNAK 和 EPENA 位置 1。
2. 将需要在下一帧中发送的数据写入发送 FIFO。
3. 硬件触发 ITTXFE 中断（在 OTG_DIEPINTx 中）表示应用程序尚未将需要发送的全部数据写入发送 FIFO。
4. 如果在检测到中断前已使能中断端点，则将忽略该中断。如果中断端点未使能，则使能此端点，以便数据能够在收到下一次 IN 令牌时发送出去。
5. 硬件触发 XFRC 中断（在 OTG_DIEPINTx 中）时如果 OTG_DIEPINTx 中未产生 ITTXFE 中断，则表示成功完成同步 IN 传输。读取 OTG_DIEPTSIZx 寄存器时始终得到传输大小 = 0 且数据包计数 = 0，表示所有数据都已通过 USB 发送完毕。
6. 无论是否产生 ITTXFE 中断（在 OTG_DIEPINTx 中），只要触发 XFRC 中断（在 OTG_DIEPINTx 中），即表示中断 IN 传输成功完成。读取 OTG_DIEPTSIZx 寄存器时始终得到传输大小 = 0 且数据包计数 = 0，表示所有数据都已通过 USB 发送完毕。
7. 如果在未产生任何前述中断的情况下在 OTG_GINTSTS 中触发未完成同步 IN 传输 (IISOIXFR) 中断，则表示模块在当前帧中至少未收到 1 个周期性 IN 令牌。

- **未完成同步 IN 数据传输**

本节介绍应用程序针对未完成同步 IN 数据传输必须执行的操作。

内部数据流：

1. 符合下列条件之一时，即认为同步 IN 传输未完成：
   a) 模块在至少一个同步 IN 端点上接收到损坏的同步 IN 令牌。此时，应用程序检测到未完成同步 IN 传输中断（OTG_GINTSTS 中的 IISOIXFR 位）。
   b) 应用程序向发送 FIFO 写入数据的速度过慢，在将完整数据写入 FIFO 之前便接收到 IN 令牌。此时，应用程序在 OTG_DIEPINTx 中检测到“Tx FIFO 为空时接收到 IN 令牌”中断。应用程序可忽略此中断，因为最终将在周期性帧结束时产生一个未完成同步 IN 传输中断（OTG_GINTSTS 中的 IISOIXFR 位）。
   模块会通过 USB 发送一个零长度数据包来响应接收到的 IN 令牌。
2. 应用程序必须尽快停止向发送 FIFO 写入数据。
3. 应用程序必须将端点的 NAK 位和禁止位置 1。
4. 模块会禁止该端点，将禁止位清零并触发端点的“端点禁止”中断。

应用程序编程顺序：

1. 应用程序可以在任何同步 IN 端点上忽略 OTG_DIEPINTx 中的“Tx FIFO 为空时接收到 IN 令牌”中断，因为最终这将产生一个未完成同步 IN 传输中断（在 OTG_GINTSTS 中）。
2. 硬件触发未完成同步 IN 传输中断（在 OTG_GINTSTS 中）表示在至少一个同步 IN 端点上存在未完成的同步 IN 传输。
3. 应用程序必须读取所有同步 IN 端点的“端点控制”寄存器来检测存在未完成 IN 数据传输的端点。
4. 应用程序必须停止向与这些端点相关联的“周期性发送 FIFO”写入数据。
5. 对 OTG_DIEPCTLx 寄存器中的下列字段进行编程以禁止端点：
   – 在 OTG_DIEPCTLx 中，SNAK = 1
   – 在 OTG_DIEPCTLx 中，EPDIS = 1
6. 硬件触发 OTG_DIEPINTx 中的“端点禁止”中断表示模块已禁止该端点。
   – 此时，应用程序必须清空相关联的发送 FIFO 中的数据，或者通过在下一微帧中使能新传输的端点来覆盖 FIFO 中的现有数据。要刷新数据，应用程序必须使用 OTG_GRSTCTL 寄存器。

- **停止非同步 IN 端点**

本节介绍应用程序如何才能停止非同步端点。

应用程序编程顺序：

1. 禁止要停止的 IN 端点。同时将 STALL 位置 1。
2. OTG_DIEPCTLx 中的 EPDIS = 1（当端点已使能时）
   - OTG_DIEPCTLx 中的 STALL = 1
   - STALL 位的优先级始终高于 NAK 位
3. 硬件触发“端点禁止”中断（在 OTG_DIEPINTx 中），应用程序获知模块已禁止指定的端点。
4. 应用程序必须根据端点类型清空非周期性或周期性发送 FIFO。对于非周期性端点，应用程序必须重新使能另一个无需停止的非周期性端点来发送数据。
5. 当应用程序准备好结束该端点的 STALL 握手信号时，必须将 OTG_DIEPCTLx 的 STALL 位清零。
6. 如果应用程序因收到来自主机的 SetFeature.Endpoint Halt 命令或 ClearFeature.Endpoint Halt 命令来设置或清除端点的 STALL 状态，则必须在该控制端点的状态阶段传输之前将 STALL 位置 1 或清零。

特例：停止控制 OUT 端点

如果在控制传输的数据阶段，主机发送的 IN/OUT 令牌数超过 SETUP 数据包指定的值，则模块必须对这些多余的 IN/OUT 令牌回复 STALL。在这种情况下，在控制传输的数据阶段，当模块传输完 SETUP 数据包指定的数据量后，应用程序必须使能 OTG_DIEPINTx 中的 ITTXFE 中断和 OTG_DOEPINTx 中的 OTEPDIS 中断。随后，当应用程序收到此中断时，必须将相应端点控制寄存器中的 STALL 位置 1 并清除此中断。

50.16.6 最坏情况下的响应时间

当 OTG_FS 控制器作为设备使用时，对于任何跟随在同步 OUT 之后的令牌，都存在一个最坏响应时间。这个最坏情况响应时间随 AHB 时钟频率的不同而异。

模块寄存器位于 AHB 域内，在更新这些寄存器值之前，模块不会接受新令牌。对于任何跟随在同步 OUT 之后的令牌，最坏的情况是：由于同步事务不需要握手信号，所以下一个令牌可能很快就到达。当 AHB 与 PHY 时钟频率相同时，这个最坏情况值为 7 个 PHY 时钟。AHB 时钟越快，此值越小。

如果发生这种最坏情况，模块将以 NAK 响应批量/中断令牌，并丢弃同步和 SETUP 令牌。对于 SETUP，主机会将这种情况视为超时，并尝试重新发送 SETUP 数据包。对于同步传输，会产生未完成同步 IN 传输中断 (IISOIXFR) 和未完成同步 OUT 传输中断 (IISOOXFR)，来通知应用程序同步 IN/OUT 数据包被丢弃。

选择 OTG_GUSBCFG 中的 TRDT 的值

TRDT 的值 (OTG_GUSBCFG) 是指在接收到 IN 令牌后以 PHY 时钟计算的时间长度，MAC 将在这段时间内获取 FIFO 状态并从 PFC 模块获取第一个数据。这段时间包含 PHY 和 AHB 时钟之间的同步延迟。最坏情况延迟是当 AHB 时钟与 PHY 时钟相等时的延迟。这种情况下的延迟为 5 个时钟周期。

MAC 接收到 IN 令牌后，此信息（接收到的令牌）将由 PFC（PFC 以 AHB 时钟运行）同步到 AHB 时钟。随后，PFC 从 SPRAM 中读取数据并将它们写入双时钟源缓冲区。MAC 再从源缓冲区读出数据（4 个深度）。

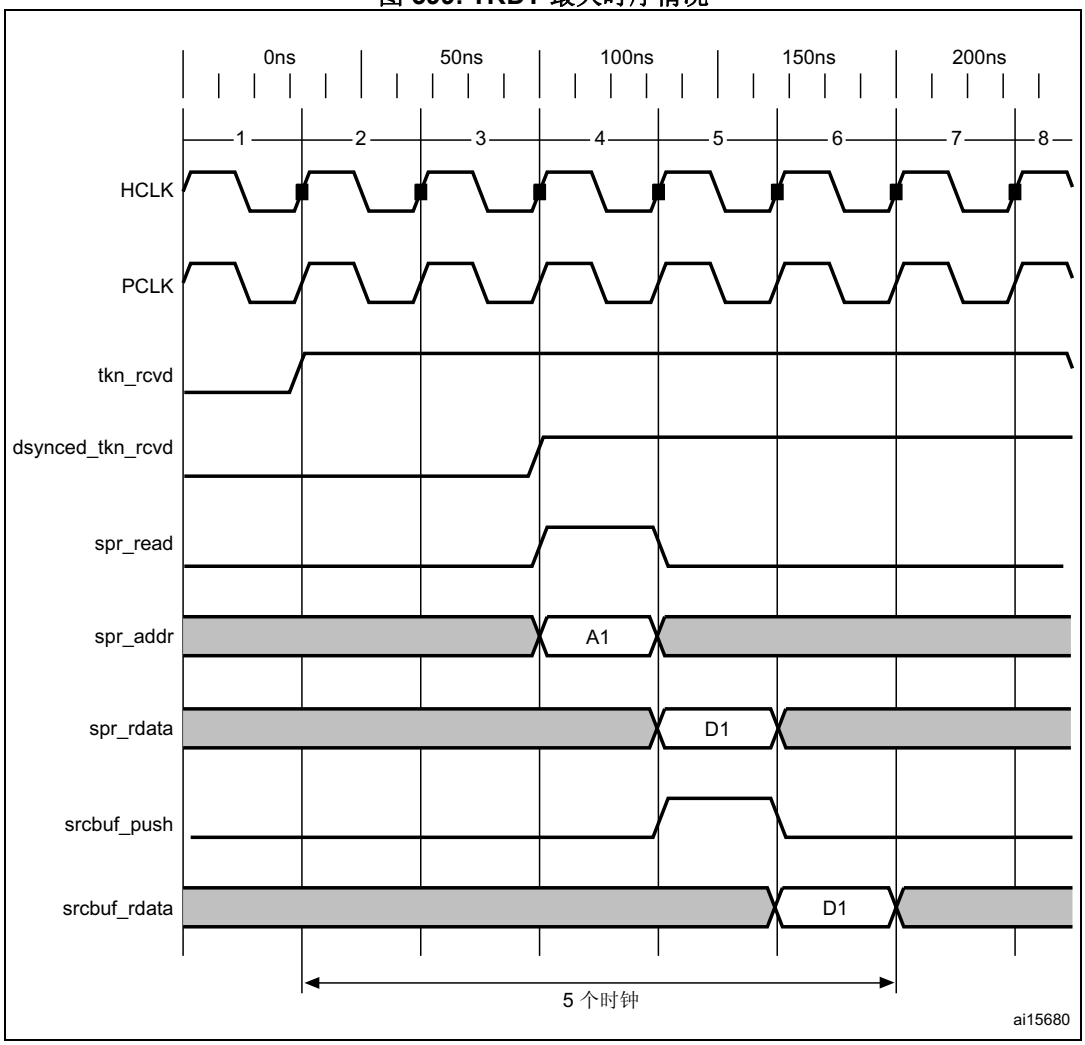
如果 AHB 的运行频率高于 PHY，应用程序可以使用较小的 TRDT 值（在 OTG_GUSBCFG 中）。

图 599 具有以下信号：

- tkn_rcvd：接收到令牌信息（从 MAC 到 PFC）
- dynced_tkn_rcvd：双倍同步 tkn_rcvd（从 PCLK 到 HCLK 域）
- spr_read：读取到 SPRAM
- spr_addr：寻址到 SPRAM
- spr_rdata：从 SPRAM 中读取数据
- srcbuf_push：压入源缓冲区
- srcbuf_rdata：从源缓冲区读取数据。MAC 看到的数据

要计算 TRDT 的值，请参见*表 325：TRDT 值 (FS)*。

图 599. TRDT 最大时序情况



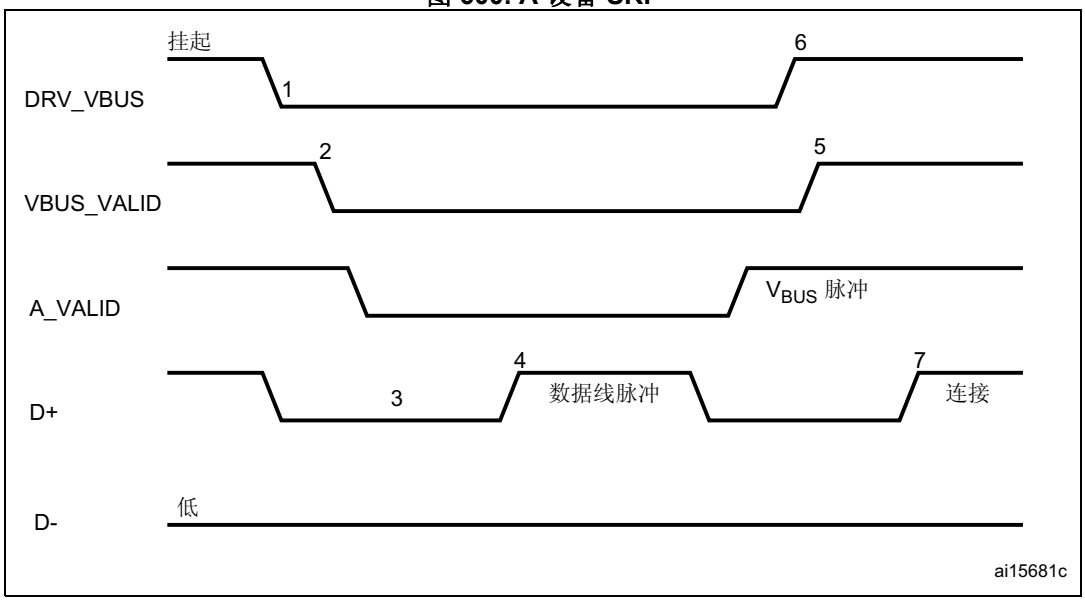
50.16.7 OTG 编程模型

OTG_FS 控制器是一种支持 HNP 和 SRP 的 OTG 设备。该模块接口与“A 型”插头连接时，该模块称为 A 设备。该模块接口与“B 型”插头连接时，该模块称为 B 设备。在主机模式下，OTG_FS 控制器将关闭 V_{BUS} 以节省电能。B 设备可以借助 SRP 请求 A 设备打开 V_{BUS} 电源。设备必须同时执行数据线脉冲和 V_{BUS} 脉冲，但主机可以检测数据线脉冲或者 V_{BUS} 脉冲中的一个用于 SRP。B 设备可以借助 HNP 协商并切换到主机角色。在协商模式下执行 HNP 后，B 设备会挂起总线并恢复到设备角色。

A 设备会话请求协议

应用程序必须将模块 USB 配置寄存器中的 SRP 功能位置 1。这将使能 OTG_FS 控制器在 A 设备模式下检测到 SRP。

图 600. A 设备 SRP



1. DRV_VBUS = 发送到 PHY 的 V_{BUS} 驱动信号
   VBUS_VALID = 来自 PHY 的 V_{BUS} 有效信号
   A_VALID = 发送到 PHY 的 A 设备 V_{BUS} 电平信号
   D+ = 正向数据线
   D- = 负向数据线

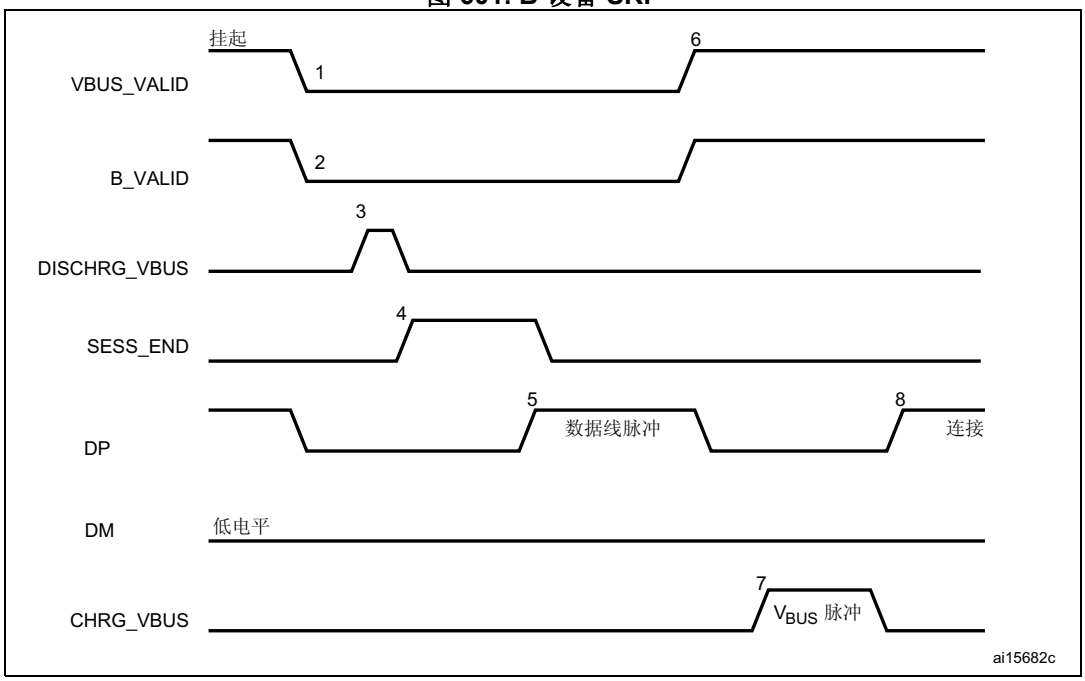
以下几点引用并描述了*图 600* 所示的信号过程：

1. 为节省电能，在总线空闲时，应用程序会通过在主机端口控制和状态寄存器中写入端口挂起位和端口电源位来挂起设备并关闭端口电源。
2. PHY 通过禁止 VBUS_VALID 信号来指示端口电源已关闭。
3. 当 V_{BUS} 电源关闭时，设备必须检测到至少 2 ms 的 SE0 信号才可以启动 SRP。
4. 要启动 SRP，设备需要打开其数据线的上拉电阻并维持 5 到 10 ms。OTG_FS 控制器会检测到数据线脉冲。
5. 设备会驱动 V_{BUS} 到 A 设备会话有效电平（最小 2.0 V）以上，以产生 V_{BUS} 脉冲。OTG_FS 控制器检测到 SRP 时将中断应用程序。在全局中断状态寄存器中，检测到会话请求位（OTG_GINTSTS 中的 SRQINT 位）将置 1。
6. 应用程序必须响应“检测到会话请求”中断，并通过在主机端口控制和状态寄存器中写入端口电源位来打开端口电源位。PHY 通过输出 VBUS_VALID 信号来指示端口已通电。
7. 当 USB 通电后，设备将连接，从而完成 SRP 过程。

B 设备会话请求协议

应用程序必须将模块 USB 配置寄存器中的 SRP 功能位置 1。这将使能 OTG_FS 控制器在 B 设备模式下启动 SRP。OTG_FS 控制器可以借助 SRP 向主机请求新会话。

图 601. B 设备 SRP



1. VBUS_VALID = 来自 PHY 的 V_{BUS} 有效信号
   B_VALID = 发送到 PHY 的 B 设备有效会话
   DISCHRG_VBUS = 发送到 PHY 的放电信号
   SESS_END = 发送到 PHY 的会话结束信号
   CHRG_VBUS = 发送到 PHY 的 V_{BUS} 充电信号
   DP = 正向数据线
   DM = 负向数据线

以下几点引用并描述了*图 601* 所示的信号过程：

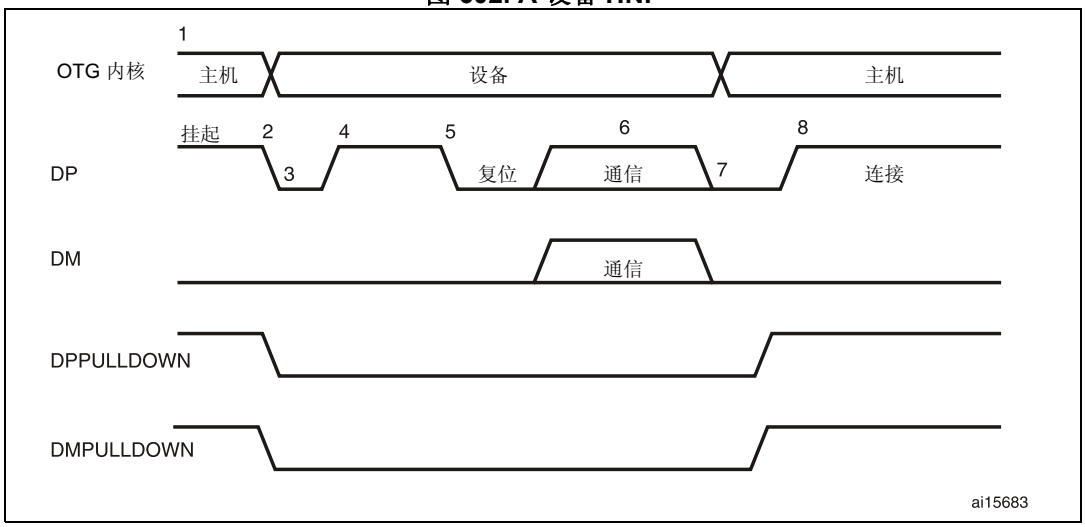
1. 为节省电能，在总线空闲时主机会挂起并关闭端口电源。
   OTG_FS 控制器会在总线空闲 3 ms 后，将模块中断寄存器中的早期挂起位置 1。随后，OTG_FS 控制器会将模块中断寄存器中的 USB 挂起位置 1。
   OTG_FS 控制器会通知 PHY 对 V_{BUS} 进行放电。
2. PHY 指示会话结束。这是 SRP 的初始条件。启动 SRP 之前，OTG_FS 控制器需要 2 ms 的 SE0 信号。
   对于 USB 1.1 全速串行收发器，应用程序必须在 BSVLD 位（位于 OTG_GOTGCTL）被禁止后，等待 V_{BUS} 放电至 0.2 V。此放电时间可从收发器供应商处获取，该值可能因收发器的不同而异。
3. OTG_FS 模块通知 PHY 对 V_{BUS} 加速放电。
4. 应用程序通过将 OTG 控制和状态寄存器中的会话请求位置 1 来启动 SRP。OTG_FS 控制器先执行数据线脉冲，随后执行 V_{BUS} 脉冲。
5. 主机会从数据线脉冲或 V_{BUS} 脉冲检测到 SRP，然后打开 V_{BUS}。PHY 指示 V_{BUS} 对设备上电。

6. OTG_FS 控制器执行 V_{BUS} 脉冲。
   主机通过打开 V_{BUS} 启动一个新会话，指示 SRP 已成功。OTG_FS 控制器通过将 OTG 中断状态寄存器中的会话请求成功状态更改位置 1 来中断应用程序。应用程序读取 OTG 控制和状态寄存器中的会话请求成功位。
7. 当 USB 通电时，OTG_FS 控制器将进行连接，并完成 SRP 过程。

A 设备主机协商协议

通过 HNP 可以将 USB 主机角色从 A 设备切换到 B 设备。应用程序必须将模块 USB 配置寄存器中的 HNP 功能位置 1，以使能 OTG_FS 控制器在 A 设备模式下的 HNP 功能。

图 602. A 设备 HNP



1. DPPULLDOWN = 从模块发送到 PHY 的信号，用于使能/禁止 PHY 内部 DP 线的下拉电阻。
   DMPULLDOWN = 从模块发送到 PHY 的信号，用于使能/禁止 PHY 内部 DM 线的下拉电阻。

以下几点引用并描述了*图 602* 所示的信号过程：

1. OTG_FS 控制器向 B 设备发送 SetFeature b_hnp_enable 描述符以使能 HNP 支持。B 设备的 ACK 响应表示 B 设备支持 HNP。应用程序必须将 OTG 控制和状态寄存器中的主机设置 HNP 使能位置 1，以向 OTG_FS 控制器指示 B 设备支持 HNP。
2. 应用程序不再使用总线时，会通过在主机端口控制和状态寄存器中写入端口挂起位来挂起总线。
3. B 设备检测到 USB 挂起时，将断开连接，指示 HNP 的初始条件。B 设备只有在切换到主机角色时才会启动 HNP，否则总线会保持挂起状态。

   OTG_FS 控制器将 OTG 中断状态寄存器中检测到的主机协商中断位置 1，指示 HNP 的启动。

   OTG_FS 控制器将禁止 PHY 中 DM 线和 DM 线的下拉，以指示设备角色。PHY 使能 OTG_DP 上拉电阻，以告知 B 设备 A 设备的连接。

   应用程序必须读取 OTG 控制和状态寄存器中的当前模式位，以确定设备工作模式。
4. B 设备检测到连接，发出 USB 复位信号，并枚举 OTG_FS 控制器进行数据通信。
5. B 设备继续担当主机角色，发起数据通信，并在结束时挂起总线。

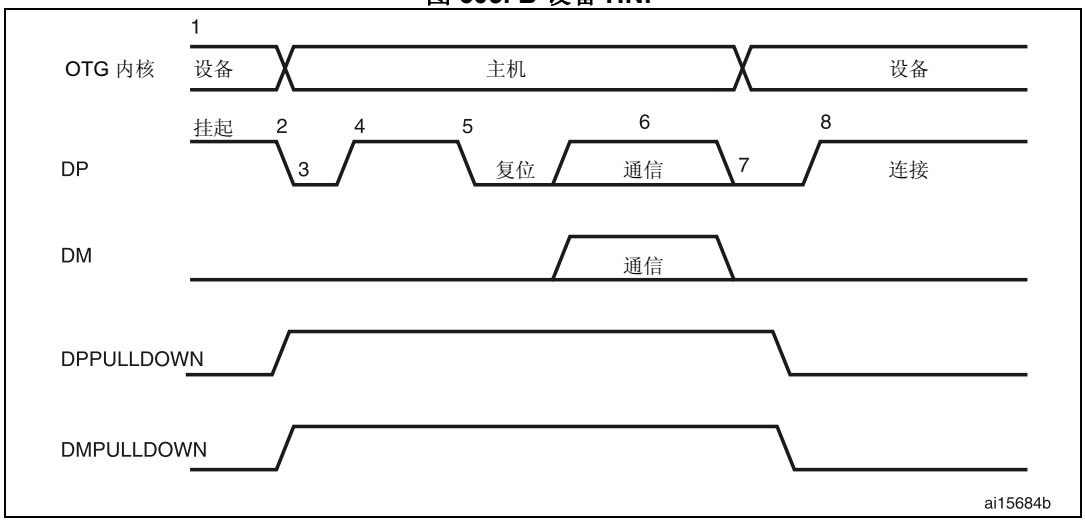
   OTG_FS 控制器会在总线空闲 3 ms 后，将模块中断寄存器中的早期挂起位置 1。随后，OTG_FS 控制器会将模块中断寄存器中的 USB 挂起位置 1。
6. 在协商模式下，OTG_FS 控制器会检测到总线挂起，连接断开，然后切换回主机角色。OTG_FS 控制器将使能 PHY 中 DM 和 DM 下拉电阻，以指示其承担主机角色。

7. OTG_FS 控制器将 OTG 中断状态寄存器中的连接器 ID 状态变化中断位置 1。应用程序必须读取 OTG 控制和状态寄存器中的连接器 ID 状态位，以确定 OTG_FS 控制器在 A 设备模式下工作。这向应用程序表示 HNP 已经完成。应用程序必须读取 OTG 控制和状态寄存器中的当前模式位，以确定主机工作模式。
8. 连接 B 设备，完成 HNP 过程。

B 设备主机协商协议

通过 HNP 可以将 USB 主机角色从 B 设备切换到 A 设备。应用程序必须将模块 USB 配置寄存器中的 HNP 功能位置 1，以使能 OTG_FS 控制器在 B 设备模式下的 HNP 功能。

图 603. B 设备 HNP



1. DPPULLDOWN = 从模块发送到 PHY 的信号，用于使能/禁止 PHY 内部 DP 线的下拉电阻。
DMPULLDOWN = 从模块发送到 PHY 的信号，用于使能/禁止 PHY 内部 DM 线的下拉电阻。

以下几点引用并描述了*图 603* 所示的信号过程：

1. A 设备发出一个 SetFeature b_hnp_enable 描述符以使能 HNP 支持。OTG_FS 控制器的 ACK 响应表示它支持 HNP。应用程序必须将 OTG 控制和状态寄存器中的设备 HNP 使能位置 1，以指示支持 HNP。

   应用程序将 OTG 控制和状态寄存器中的 HNP 请求位置 1，以向 OTG_FS 控制器指示启动 HNP。
2. A 设备不再使用总线时，会通过在主机端口控制和状态寄存器中写入端口挂起位来挂起总线。

   OTG_FS 控制器会在总线空闲 3 ms 后，将模块中断寄存器中的早期挂起位置 1。随后，OTG_FS 控制器会将模块中断寄存器中的 USB 挂起位置 1。

   OTG_FS 控制器会断开连接，A 设备在总线上检测到 SE0，指示 HNP 即将开始。OTG_FS 控制器将使能 PHY 中的 DP 和 DM 下拉电阻，以指示其承担主机角色。

   在检测到 SE0 的 3 ms 内，A 设备会通过激活 OTG_DP 上拉电阻进行响应。OTG_FS 控制器检测到此信号时认为有设备接入。

   OTG_FS 控制器将 OTG 中断状态寄存器中的主机协商成功状态变化中断位置 1，指示 HNP 的状态。应用程序必须读取 OTG 控制和状态寄存器中的主机协商成功位，以确定主机协商成功。应用程序必须读取模块中断寄存器 (OTG_GINTSTS) 中的当前模式位，以确定主机工作模式。

3. 应用程序将复位位（OTG_FS_HPRT 中的 PRST 位）置 1，同时 OTG_FS 控制器发出 USB 复位信号，并枚举 A 设备进行数据通信。
4. OTG_FS 控制器继续保持发起通信的主机角色，在通信结束后，会通过将主机端口控制和状态寄存器中的端口挂起位置 1 来挂起总线。
5. 在协商模式下，当 A 设备检测到挂起时，会断开连接，然后切换回主机角色。OTG_FS 控制器将禁止 PHY 中 DP 和 DM 下拉电阻，以指示其承担设备角色。
6. 应用程序必须读取模块中断寄存器 (OTG_GINTSTS) 中的当前模式位，以确定主机工作模式。
7. OTG_FS 控制器进行连接，完成 HNP 过程。

51 调试支持 (DBG)

51.1 概述

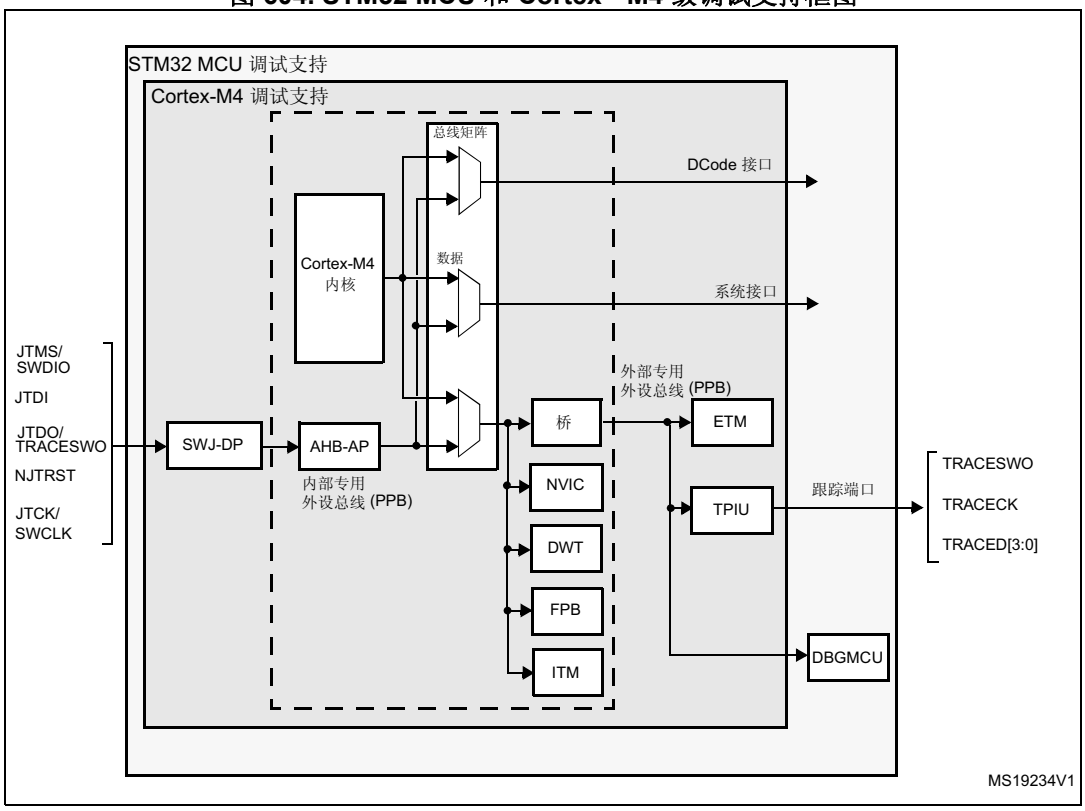
STM32L4Rxxx 和 STM32L4Sxxx 器件的内核是 Cortex®-M4，该内核包含用于高级调试功能的硬件扩展。调试扩展允许内核可以在取指（指令断点）或取访问数据（数据断点）时停止内核。内核停止时，可以查询内核的内部状态和系统的外部状态。查询完成后，将恢复内核和系统并恢复程序执行。

当调试主机与 STM32L4Rxxx 和 STM32L4Sxxx MCU 相连并进行调试时，将使用调试功能。

提供两个调试接口：

- 串行接口
- JTAG 调试端口

图 604. STM32 MCU 和 Cortex®-M4 级调试支持框图



注： *Cortex®-M4 内核中内置的调试功能是 Arm® CoreSight 设计套件的一部分。*

Arm® Cortex®-M4 内核提供集成的片上调试支持。它包括：

- SWJ-DP：串行线/JTAG 调试端口
- AHP-AP：AHB 访问端口
- ITM：指令跟踪宏单元
- FPB：Flash 指令断点
- DWT：数据观察点触发
- TPUI：跟踪端口单元接口（大封装上提供，其中会映射相应引脚）
- ETM：嵌入式跟踪宏单元（STM32L4Rxxx 和 STM32L4Sxxx 器件大封装上提供，其中会映射相应引脚）

它还包括专用于 STM32L4Rxxx 和 STM32L4Sxxx 的调试功能：

- 灵活调试引脚分配
- MCU 调试盒（支持低功耗模式和对外设时钟的控制等）

注： *有关 Arm® Cortex®-M4 内核支持的调试功能的更多信息，请参见《Cortex®-M4-r0p1 技术参考手册》和《CoreSight 设计套件 r0p1 技术参考手册》（请参见第 51.2 节：Arm® 参考文档）。*

51.2 Arm® 参考文档

- Cortex®-M4 r0p1 技术参考手册 (TRM)，在 http://infocenter.arm.com 网站中搜索“Cortex®-M4 技术参考手册”
- Arm® 调试接口 V5
- Arm® CoreSight 设计套件版本 r0p1 技术参考手册

51.3 SWJ 调试端口（串行接口和 JTAG）

STM32L4Rxxx 和 STM32L4Sxxx 内核集成了串行/JTAG调试端口 (SWJ-DP)。该端口是 Arm® 标准 CoreSight 调试端口，具有 JTAG-DP（5 引脚）接口和 SW-DP（2 引脚）接口。

- JTAG 调试端口 (JTAG-DP) 提供用于连接到 AHP-AP 端口的 5 引脚标准 JTAG 接口。
- 串行线调试端口 (SW-DP) 提供用于连接到 AHP-AP 端口的 2 引脚（时钟 + 数据）接口。

在 SWJ-DP 中，SW-DP 的 2 个 JTAG 引脚与 JTAG-DP 的 5 个 JTAG 引脚中的部分引脚复用。

图 605. SWJ 调试端口

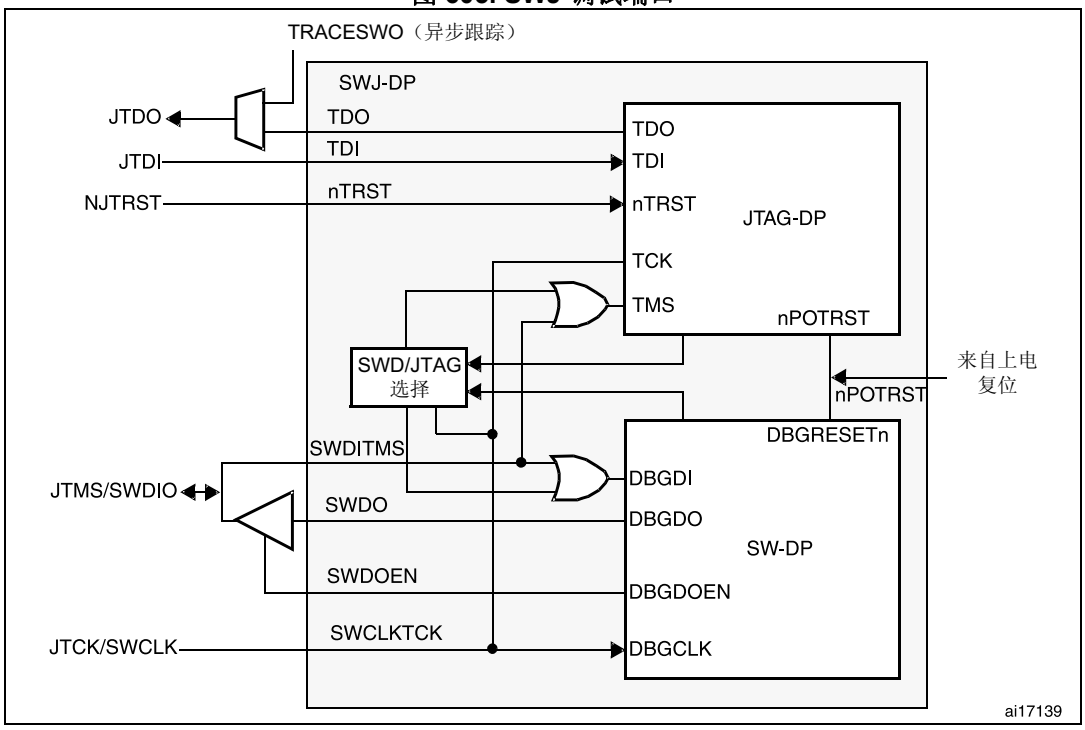


图 605 显示了异步 TRACE 输出 (TRACESWO) 与 TDO 复用。这意味着异步跟踪只能在 SW-DP 上实现，不能在 JTAG-DP 上实现。

51.3.1 JTAG-DP 或 SW-DP 的切换机制

默认调试接口是 JTAG 接口。

如果调试主机想要切换到 SW-DP，它必须在 TMS/TCK（分别映射到 SWDIO 和 SWCLK）上提供专用的 JTAG 序列，用于禁止 JTAG-DP 并使能 SW-DP。这样便可仅使用 SWCLK 和 SWDIO 引脚来激活 SWDP。

该序列为：

1. 输出超过 50 个 TCK 周期的 TMS (SWDIO) = 1 信号
2. 输出 16 个 TMS (SWDIO) 信号 0111100111100111 (MSB)
3. 输出超过 50 个 TCK 周期的 TMS (SWDIO) = 1 信号

51.4 引脚排列和调试端口引脚

STM32L4Rxxx 和 STM32L4Sxxx MCU 的不同封装有不同的有效引脚数。因此，某些与引脚相关的功能 (ETM) 可能随封装而有不同。

51.4.1 SWJ 调试端口引脚

STM32L4Rxxx 和 STM32L4Sxxx 的五个通用 I/O 可用作 SWJ-DP 接口引脚。所有封装都提供这些引脚。

表 328. SWJ 调试端口引脚

SWJ-DP 引脚名称	JTAG 调试端口		SW 调试端口		引脚分配
	类型	说明	类型	调试分配	
JTMS/SWDIO	I	JTAG 测试模式选择	IO	串行线数据输入/输出	PA13
JTCK/SWCLK	I	JTAG 测试时钟	I	串行线时钟	PA14
JTDI	I	JTAG 测试数据输入	-	-	PA15
JTDO/TRACESWO	O	JTAG 测试数据输出	-	TRACESWO（如果使能异步跟踪）	PB3
NJTRST	I	JTAG 测试 nReset	-	-	PB4

51.4.2 灵活的 SWJ-DP 引脚分配

复位（SYSRESETn 或 PORESETn）后，会将用于 SWJ-DP 的全部 5 个引脚指定为专用引脚，可供调试工具立即使用（请注意，除非由调试工具明确编程，否则不分配跟踪输出）。

然而 STM32L4Rxxx 和 STM32L4Sxxx MCU 可以禁止一部分或所有 SWJ-DP 端口，因此可以释放相关引脚（除了保持断开的 NJTRST）以供通用 I/O (GPIO) 使用，但是调试器连接断开时不能将这些引脚用作通用 GPIO。有关如何禁止 SWJ-DP 端口引脚的更多详细信息，请参见*第 8.3.2 节：I/O 引脚复用功能复用器和映射*。

表 329. 灵活的 SWJ-DP 引脚分配

可用的调试端口	分配的 SWJ IO 引脚				
	PA13/ JTMS/ SWDIO	PA14/ JTCK/ SWCLK	PA15/ JTDI	PB3/ JTDO	PB4/ 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				

注： *当 APB 桥的写缓冲区已满后，还需要一个额外的 APB 周期来写入 AFIO_MAPR 寄存器。这是因为释放 JTAGSW 引脚需要两个周期，以保证输入内核的 nTRST 和 TCK 信号的平稳。*

- 周期 1：输入 1/0 的 JTAGSW 信号到内核（nTRST、TDI 和 TMS 为 1，TCK 为 0）。
- 周期 2：GPI/O 控制器获得 SWJTAG I/O 引脚的控制信号（如对方向，上拉/下拉，施密特触发等的控制）。

51.4.3 JTAG引脚上的内部上拉和下拉

必须确保 JTAG 输入引脚不浮空，因为这些引脚直接连接到用于控制调试模式功能的触发器。还必须特别注意 SWCLK/TCK 引脚，该引脚直接连接到一些触发器的时钟。

为避免 IO 电平浮空，器件在 JTAG 引脚上内置有内部上拉和下拉：

- NJTRST：内部上拉
- JTDI：内部上拉
- JTMS/SWDIO：内部上拉
- TCK/SWCLK：内部下拉

用户软件释放 JTAG IO 后，GPIO 控制器便会再次取得控制权。GPIO 控制寄存器的复位状态会将 I/O 置于：

- NJTRST：输入上拉
- JTDI：输入上拉
- JTMS/SWDIO：输入上拉
- JTCK/SWCLK：输入下拉
- JTDO：输入浮空

软件可以把这些 I/O 口作为普通的 I/O 口使用。

注： *JTAG IEEE 标准建议在 TDI、TMS 和 nTRST 上添加上拉，但对 TCK 没有特殊建议。但是，对于 JTCK，器件需要集成下拉。*

由于带有上拉和下拉电阻，因此无需添加外部电阻。

51.4.4 使用串行接口以及释放未使用的调试引脚以用作 GPIO

为了利用串行调试接口以便释放一些 GPIO，用户软件必须在 GPIO_MODER 寄存器中更改 GPIO（PA15、PB3 和 PB4）配置模式。这样便可释放 PA15、PB3 和 PB4，以将这些引脚用作 GPIO。

调试时，主机执行以下操作：

- 在系统复位状态下，分配所有 SWJ 引脚 (JTAG-DP + SW-DP)。
- 在系统复位状态下，调试主机发送 JTAG 序列，以从 JTAG-DP 切换到 SW-DP。
- 仍然在系统复位状态下，调试主机在复位地址处设置断点。
- 释放复位信号，内核停止在复位地址处。
- 从此所有调试通信均使用 SW-DP 完成，然后通过用户软件将其他 JTAG 引脚重新分配为 GPIO。

注： *对于用户软件设计，请注意：*

请记住，要释放调试引脚，在复位后一直到用户软件释放引脚这段期间，这些引脚仍然处于输入上拉（nTRST、TMS 和 TDI）、下拉 (TCK) 或输出三态 (TDO)。

当这些引脚被配置为专用引脚时（JTAG 或者 SW 或者 TRACE），修改相应的普通 I/O 口配置寄存器是无效的。

51.5 STM32L4Rxxx 和 STM32L4Sxxx JTAG TAP 连接

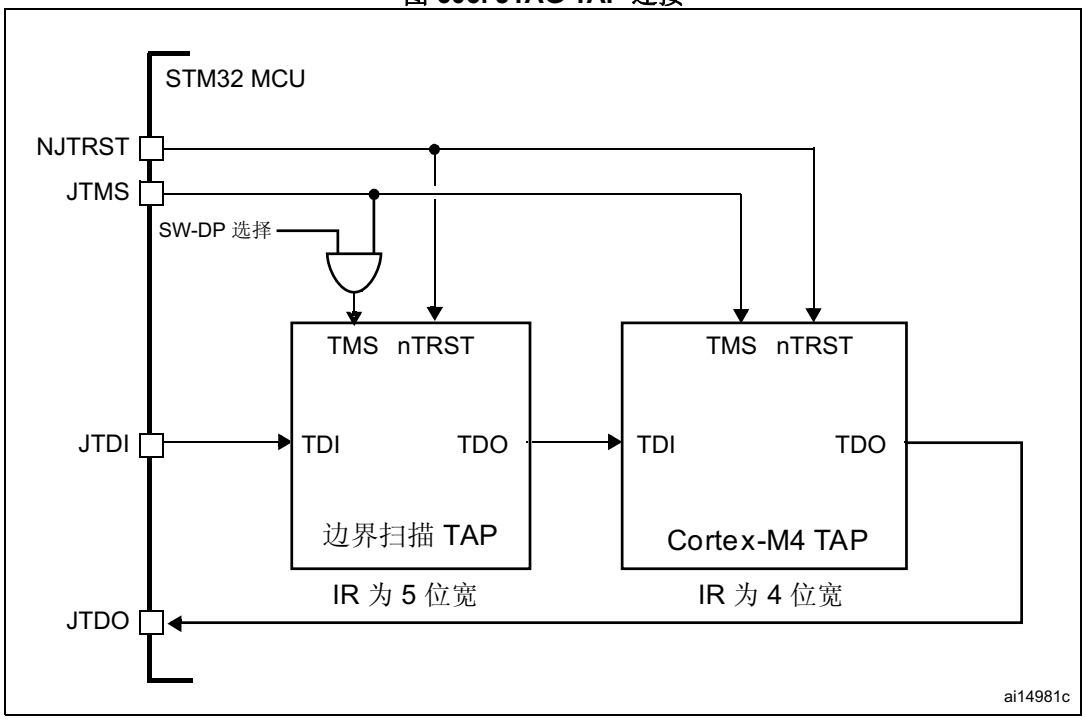
STM32L4Rxxx 和 STM32L4Sxxx MCU 集成了两个串连的 JTAG TAP、边界扫描 TAP（IR 宽度为 5 位）和 Cortex®-M4 TAP（IR 宽度为 4 位）。

要访问 Cortex®-M4 的 TAP 以进行调试：

1. 首先，必须将 BYPASS 指令移位输入边界扫描 TAP。
2. 其次，在移位输入 IR 时，每个扫描链包含 9 个比特位 (=5+4)，对于不用的 TAP，必须输入 BYPASS 指令。
3. 移位输入数据时，不用的 TAP 处于 BYPASS 模式下，因此数据扫描链需要额外添加一位比特位。

注： ***重要提示：****使用专用 Arm® JTAG 序列选择串行线后，将自动禁止边界扫描 TAP（JTMS 强制为高电平）。*

图 606. JTAG TAP 连接



51.6 ID 代码和锁定机制

STM32L4Rxxx 和 STM32L4Sxxx MCU 中提供了一些 ID 代码。ST 强烈建议工具设计人员使用位于外部 PPB 存储器映射的地址 0xE0042000 中的 MCU DEVICE ID 代码锁定调试工具。

51.6.1 MCU 器件 ID 代码

STM32L4Rxxx 和 STM32L4Sxxx MCU 集成了 MCU ID 代码。此 ID 标识 ST MCU 的料号和晶片版本。它是 DBG_MCU 组件的一部分并映射到外部 PPB 总线上（请参见*第 1943 页的第 51.16 节*）。可通过 JTAG 调试端口（4 到 5 个引脚）或 SW 调试端口（2 个引脚）或者用户软件访问此代码。即使在 MCU 处于系统复位状态下也可以访问。

调试软件/编程工具只能将 DEV_ID(11:0) 用于识别芯片。

DBGMCU_IDCODE

地址：0xE004 2000

仅支持 32 位访问。只读

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

位 31:16 **REV_ID[15:0]**：版本标识符 (Revision identifier)
此字段指示器件的版本。
0x1000：版本 A
0x1001：版本 Z
0x1003：版本 Y

位 15:12 保留，必须保持复位值。

位 11:0 **DEV_ID[11:0]**：器件标识符 (Device identifier)
器件 ID 为：0x470

51.6.2 边界扫描 TAP

JTAG ID 代码

STM32L4Rxxx 和 STM32L4Sxxx BSC（边界扫描）的 TAP 集成了 JTAG ID 代码（等于 0x06470041）。

51.6.3 Cortex®-M4 TAP

Arm® Cortex®-M4 的 TAP 集成了 JTAG ID 代码。此 ID 代码是 Arm® 的默认代码，未经修改。此代码只能通过 JTAG 调试端口进行访问。
此代码为 **0x4BA00477**（对应于 Cortex®-M4 r0p1，请参见*第 51.2 节：Arm® 参考文档*）。

51.6.4 Cortex®-M4 JEDEC-106 ID 代码

Arm® Cortex®-M4 集成了 JEDEC-106 ID 代码。它位于映射到内部 PPB 总线地址为 0xE00FF000_0xE00FFFFF 的 4KB ROM 表中。

可通过 JTAG 调试端口（4 到 5 个引脚）或 SW 调试端口（2 个引脚）或者用户软件访问此代码。

51.7 JTAG 调试端口

标准 JTAG 状态机使用一个 4 位指令寄存器 (IR) 和五个数据寄存器实现（有关完整详细信息，请参见 Cortex®-M4 带 FPU 的 r0p1 技术参考手册 (TRM)；有关参考信息，请参见第 51.2 节：Arm® 参考文档）。

表 330. JTAG 调试端口数据寄存器

IR(3:0)	数据寄存器	详细信息
1111	BYPASS [1 位]	-
1110	IDCODE [32 位]	ID CODE 0x3BA00477（Arm® Cortex®-M4 r0p1-01rel0 ID 代码）
1010	DPACC [35 位]	调试接口寄存器 初始化调试接口，并允许访问调试接口寄存器。 – 传输 IN 数据时： 位 34:3 = DATA[31:0] = 为写请求传输的 32 位数据 位 2:1 = A[3:2] = 调试端口寄存器的 2 位地址 位 0 = RnW = 读请求 (1) 或写请求 (0) – 传输 OUT 数据时： 位 34:3 = DATA[31:0] = 读请求后读取的 32 位数据 位 2:0 = ACK[2:0] = 3 位确认： 010 = OK/FAULT 001 = WAIT 其他 = 保留 有关 A(3:2) 位的说明，请参见表 331
1011	APACC [35 位]	存取接口寄存器 初始化存取接口并允许访问存取接口寄存器。 – 传输 IN 数据时： 位 34:3 = DATA[31:0] = 为写请求移入的 32 位数据 位 2:1 = A[3:2] = 2 位地址（子地址 AP 寄存器） 位 0 = RnW = 读请求 (1) 或写请求 (0) – 传输 OUT 数据时： 位 34:3 = DATA[31:0] = 读请求后读取的 32 位数据 位 2:0 = ACK[2:0] = 3 位确认： 010 = OK/FAULT 001 = WAIT 其他 = 保留 有多个 AP 寄存器（请参见 AHB-AP），这些寄存器按以下组合进行寻址： – 移位值 A[3:2] – DP SELECT 寄存器的当前值
1000	ABORT [35 位]	中止寄存器 – 位 31:1 = 保留 – 位 0 = DAPABORT：写入 1 以生成 DAP 中止

表 331. 32 位调试端口寄存器，通过移位值 A[3:2] 进行寻址

地址	A(3:2) 值	说明
0x0	00	保留，必须保持复位值。
0x4	01	DP CTRL/STAT 寄存器。用于： – 请求系统或调试上电 – 配置 AP 访问的传输操作 – 控制比较和验证操作 – 读取一些状态标志（上溢和上电确认）
0x8	10	DP SELECT 寄存器。用于选择当前访问端口和活动的 4 字寄存器窗口。 – 位 31:24：APSEL：选择当前 AP – 位 23:8：保留 – 位 7:4：APBANKSEL：在当前 AP 上选择活动的 4 字寄存器窗口 – 位 3:0：保留
0xC	11	DP RDBUFF 寄存器：用于通过调试器在执行一系列操作后获取最后结果（无需请求新的 JTAG-DP 操作）

51.8 SW 调试端口

51.8.1 SW 协议简介

此同步串行协议使用两个引脚：

- SWCLK：从主机到目标的时钟
- SWDIO：双向

利用该协议，可以同时读取和写入两组寄存器组（DPACC 寄存器组和 APACC 寄存器组）。

传输数据时，LSB 在前。

对于 SWDIO 双向管理，必须在电路板上对线路进行上拉（Arm® 建议采用 100 kΩ）。

每次在协议中更改 SWDIO 的方向时，都会插入转换时间，此时线路即不受主机驱动也不受目标驱动。默认情况下，此转换时间为一位时间，但可以通过配置 SWCLK 频率来调整。

51.8.2 SW 协议序列

每个序列包括三个阶段：

1. 主机发送的数据包请求（8 位）
2. 目标发送的确认响应（3 位）
3. 主机或目标发送的数据传输阶段（33 位）

表 332. 数据包请求（8 位）

位	名称	说明
0	启动	必须为 1
1	APnDP	0：DP 访问 1：AP 访问
2	RnW	0：写请求 1：读请求
4:3	A(3:2)	DP 或 AP 寄存器的地址字段（请参见*表 331*）
5	奇偶校验	前面几位的单位奇偶校验
6	停止	0
7	驻留	不受主机驱动。由于存在上拉，因此必须由目标读为 1

有关 DPACC 和 APACC 寄存器的详细说明，请参见 Cortex®-M4 r0p1 *TRM*。

数据包请求后面始终为转换时间（默认 1 位），此时主机和目标都不会驱动线路。

表 333. ACK 响应（3 位）

位	名称	说明
0..2	ACK	001：FAULT 010：WAIT 100：OK

仅当发生 READ 事务或者接收到 WAIT 或 FAULT 确认时，ACK 响应后才必须是转换时间。

表 334. DATA 传输（33 位）

位	名称	说明
0..31	WDATA 或 RDATA	写入或读取数据
32	奇偶校验	32 个数据位的单奇偶校验

仅当发生 READ 事务时，DATA 传输后才必须是转换时间。

51.8.3 SW-DP 状态机（复位、空闲状态、ID 代码）

SW-DP 的状态机有一个用于标识 SW-DP 的内部 ID 代码。该代码符合 JEP-106 标准。此 ID 代码是默认的 Arm® 代码，设置为 **0x1BA01477**（相当于 Cortex®-M4 r0p1）。

注： *请注意，在目标读取此 ID 代码前，SW-DP 状态机是不工作的。*

- 在上电复位后或是 DP 从 JTAG 切换到 SWD 后或者线路处于高电平超过 50 个周期后，SW-DP 状态机处于复位状态。
- 如果在复位状态后线路处于低电平至少两个周期，SW-DP 状态机处于空闲状态。
- 复位状态后，该状态机**必须**首先进入空闲状态，然后对 DP-SW ID CODE 寄存器执行读访问。否则，目标将在另一个事务上发出 FAULT 确认响应。

有关 SW-DP 状态机的更多详细信息，请参见 *《Cortex®-M4 r0p1 TRM》*和 *《CoreSight 设计套件 r0p1 TRM》*。

51.8.4 DP 和 AP 读/写访问

- 对 DP 的读访问还没有发出：可以立即发送目标响应（如果 ACK=OK），也可以延迟发送目标响应（如果 ACK=WAIT）。
- 对 AP 的读访问已经发出。这意味着会在下次传输时返回访问结果。如果要执行的下次访问不是 AP 访问，则必须读取 DP-RDBUFF 寄存器来获取结果。
  每次进行 AP 读访问或 RDBUFF 读请求时都会更新 DP-CTRL/STAT 寄存器的 READOK 标志，以便了解 AP 读访问是否成功。
- SW-DP 有写缓冲区（用于 DP 或 AP 写入），这样即使在其他操作仍未完成时，也可以接受写入操作。如果写缓冲区已满，则目标确认响应为 WAIT。但 IDCODE 读取、CTRL/STAT 读取或 ABORT 写入除外，这几项操作在写缓冲区已满时也会被接受。
- 由于存在异步时钟域 SWCLK 和 HCLK，因此写操作后（奇偶校验位后）还需要两个额外的 SWCLK 周期，以使写入操作在内部生效。应在将线路驱动为低电平时（空闲状态）应用这些周期。
  在写 CTRL/STAT 寄存器以提出一个上电请求时，这一点特别重要。如果下一个操作（在内核上电后才有效的操作）立即执行，这将导致失败。

51.8.5 SW-DP 寄存器

当 APnDP=0 时能够访问这些寄存器

表 335. SW-DP 寄存器

A(3:2)	R/W	SELECT 寄存器的 CTRLSEL 位	寄存器	注释
00	读取	-	IDCODE	制造商代码未设置为 ST 代码 **0x2BA01477**（标识 SW-DP）
00	写	-	ABORT	-
01	读/写	0	DP-CTRL/STAT	目的： – 请求系统或调试上电 – 配置 AP 访问的传输操作 – 控制比较和验证操作 – 读取一些状态标志（上溢和上电确认）
01	读/写	1	WIRE CONTROL	用于配置物理串行端口协议（如转换时间的持续时间）
10	读取	-	READ RESEND	允许从已损坏的调试软件传输中恢复读取数据，无需重复执行原始 AP 传输。
10	写	-	SELECT	用于选择当前访问端口和活动的 4 字寄存器窗口
11	读/写	-	READ BUFFER	由于已发出 AP 访问，因此该读缓冲区非常有用（在执行下个 AP 事务时提供读取 AP 请求的结果） 此读取缓冲区捕获 AP 中的数据，显示为前一次读取的结果，无需启动新操作

51.8.6 SW-AP 寄存器

当 APnDP=1 时能够访问这些寄存器

有多个 AP 寄存器（请参见 AHB-AP），这些寄存器按以下组合进行寻址：

- 移位值 A[3:2]
- DP SELECT 寄存器的当前值

51.9 AHB-AP（AHB 访问端口）——对 JTAG-DP 和 SW-DP 同时有效

特性：

- 系统访问与处理器状态无关。
- SW-DP 或 JTAG-DP 访问 AHB-AP。
- AHB-AP 是总线矩阵内的 AHB 主设备。因此，它可以访问所有数据总线（Dcode 总线、系统总线、内部和外部 PPB 总线），但不能访问 ICode 总线。
- 支持位寻址的传输。
- 旁路 FPB 的 AHB-AP 传输。

32 位 AHP-AP 寄存器的地址宽度为 6 位（最多达 64 字或 256 字节），其中包括：

c) 位 [7:4] = DP SELECT 寄存器的位 [7:4] APBANKSEL

d) 位 [3:2] = SW-DP 的 35 位数据包请求的 2 个地址位 A(3:2)

Cortex®-M4 的 AHB-AP 包括 9 个 32 位寄存器：

表 336. Cortex®-M4 AHB-AP 寄存器

偏移地址	寄存器名称	注释
0x00	AHB-AP 控制和状态字	配置和控制通过 AHB 接口的传输（大小、hprot、当前传输的状态、地址递增类型）
0x04	AHB-AP 传输地址	-
0x0C	AHB-AP 数据读/写	-
0x10	AHB-AP 分组数据 0	直接访问 4 个相连的字而不用重写访问地址
0x14	AHB-AP 分组数据 1	
0x18	AHB-AP 分组数据 2	
0x1C	AHB-AP 分组数据 3	
0xF8	AHB-AP 调试 ROM 地址	调试接口的基址
0xFC	AHB-AP ID 寄存器	-

有关更多详细信息，请参见《*Cortex®-M4 r0p1 TRM*》。

51.10 内核调试

通过内核调试寄存器调试内核。对这些寄存器的调试访问通过*先进高性能总线* (AHB-AP) 端口进行。处理器可通过内部*专用外设总线* (PPB) 直接访问这些寄存器。

它由 4 个寄存器组成：

表 337. 内核调试寄存器

寄存器	说明
DHCSR	32 位调试停止控制和状态寄存器： 此寄存器提供有关处理器状态的信息，能够使内核进入调试停止状态并提供处理器步进功能。
DCRSR	17 位调试内核寄存器选择器寄存器： 此寄存器选择需要进行读写操作的处理器寄存器。
DCRDR	32 位调试内核寄存器数据寄存器： 此寄存器保存在寄存器与 DCRSR（选择器）寄存器选择的处理器之间读取和写入的数据。
DEMCR	32 位调试异常和监视控制寄存器： 此寄存器提供向量捕获和调试监视控制。此寄存器包含一个名为 ***TRCENA*** 的位，该位用于使能 TRACE 功能。

注： ***重要提示：****这些寄存器在系统复位时不复位。它们只能通过上电复位来复位。*

有关更多详细信息，请参见《Cortex®-M4 r0p1 TRM》。

为了在复位后立即使内核进入调试停止状态，必须：

- 使能调试和异常监视控制寄存器的位 0 (VC_CORRESET)
- 使能调试停止控制和状态寄存器的位 0 (C_DEBUGEN)

51.11 调试主机在系统复位状态下建立连接的功能

STM32L4Rxxx 和 STM32L4Sxxx MCU 的复位系统由以下复位源组成：

- POR（上电复位），在每次上电时将执行复位
- 内部看门狗复位
- 软件复位
- 外部复位

Cortex®-M4 将调试部分的复位（通常为 PORRESETn）和其他部分 (SYSRESETn) 的复位分开。

这样，调试主机便能够在系统复位期间建立连接，对内核调试寄存器进行编程，以在获取复位向量时停止内核。随后，调试主机释放系统复位，内核将立即停止并且不执行任何指令。此外，还可以在内核处于复位状态下时配置调试特性。

注： *强烈建议调试主机在系统复位状态下建立连接（在复位向量处设置断点）。*

51.12 FPB（Flash 补丁断点）

FPB 单元：

- 实现硬件断点。
- 用系统区域的代码和数据取代代码区域的代码和数据。利用此功能可以修正位于代码存储空间中的软件错误。

软件补丁和硬件断点都必须单独使用。

FPB 包括：

- 2 个内容比较器，用来比较代码区域取得的内容并重映射到系统区域的相关地址。
- 6 个指令比较器，用来比较代码区域的指令。这些比较器可用来实现软件补丁或者硬件断点功能。

51.13 DWT（数据观察点触发）

DWT 单元由四个比较器组成。这些比较器可配置为：

- 硬件观察点或
- ETM 的触发或
- PC 采样器或
- 数据地址采样器

DWT 还能以某种方式提供一些概要信息。为此，可访问某些计数器以获得以下数据：

- 时钟周期
- 分支指令
- 存取单元操作
- 睡眠周期
- CPI（每条指令的执行时间）
- 中断开销

51.14 ITM（指令跟踪宏单元）

51.14.1 概述

ITM 是应用驱动的跟踪源，支持 *printf* 类的调试以跟踪*操作系统* (OS) 和应用程序事件，并发布诊断系统信息。ITM 以数据包形式发送跟踪信息，这些信息包括：

- **软件跟踪。**软件可以直接写入 ITM 激励寄存器以发布包信息。
- **硬件跟踪。**DWT 生成数据包，然后 ITM 发送这些数据包。
- **时间戳。**发送与数据包相关的时间戳。ITM 包含 21 位计数器，用于生成时间戳。计数器由 Cortex®-M4 时钟或*串行线查看器 (SWV) 输出的位时钟驱动。*

ITM 发送的数据包输出到 TPIU（跟踪端口接口单元）。TPIU 的格式化器添加一些额外的数据包（请参见 TPIU），然后将整个数据包序列输出到调试主机。

在设置或使用 ITM 前，必须使能调试异常和监视控制寄存器的位 TRCEN。

51.14.2 时间戳数据包、同步和溢出数据包

时间戳数据包会对时间戳信息、通用控制和同步进行编码。它使用 21 位时间戳计数器（带可能的预分频器），该计数器在每次发送时间戳数据包时复位。此计数器可由 CPU 时钟或 SWV 时钟驱动。

同步数据包由 6 个字节组成，等于 0x80_00_00_00_00_00，以 00 00 00 00 00 80 形式（首先发送 LSB）发送到 TPIU。

同步数据包是时间戳数据包的控制信号。它在每次触发 DWT 时发送。

为此，DWT 必须配置为触发 ITM：必须将 DWT 控制寄存器的位 CYCCNTENA（位 0）置 1。此外，还必须将 ITM 跟踪控制寄存器的位 2 (SYNCENA) 置 1。

注： *如果未将 SYNENA 位置 1，DWT 产生给 TPIU 的同步触发，将仅发送 TPIU 同步数据包，不发送 ITM 同步数据包。*

溢出数据包是一种特殊的时间戳数据包，用于指示已写入数据但 FIFO 已满。

表 338. 主要的 ITM 寄存器

地址	寄存器	详细信息
@E0000FB0	ITM 锁定访问	写入 0xC5ACCE55 以解锁对其他 ITM 寄存器的写访问
@E0000E80	ITM 跟踪控制	位 31-24 = 始终为 0
		位 23 = 忙碌
		位 22-16 = 7 位 ATB ID，用于标识跟踪数据源
		位 15-10 = 始终为 0
		位 9:8 = TSPrescale = 时间戳预分频器
		位 7-5 = 保留
		位 4 = SWOENA = 使能 SWV 行为（由 SWV 时钟驱动时间戳计数器）
		位 3 = DWTENA：使能 DWT 激励
		位 2 = SYNCENA：此位必须置为 1 以使能 DWT，进而生成同步触发，这样 TPIU 便可发送同步数据包
		位 1 = TSENA（时间戳使能）
		位 0 = ITMENA：ITM 的全局使能位
@E0000E40	ITM 跟踪特权	位 3：置 1 以使能跟踪端口 31:24
		位 2：置 1 以使能跟踪端口 23:16
		位 1：置 1 以使能跟踪端口 15:8
		位 0：置 1 以使能跟踪端口 7:0
@E0000E00	ITM 跟踪使能	每个位使能相应的激励端口以产生跟踪
@E0000000-E000007C	激励端口寄存器 0-31	在要跟踪的选定激励端口上写入 32 位数据（32 个可用位）

配置示例

从 TPIU 输出一个简单的数值：

- 通过配置 DBGMCU_CR 来配置 TPIU 并分配 TRACE I/O（请参见*第 51.17.2 节：TRACE 引脚分配*和*第 51.16.3 节：调试 MCU 配置寄存器 (DBGMCU_CR)*）
- 将 0xC5ACCE55 写入 ITM 锁定访问寄存器，以解锁对 ITM 寄存器的写保护
- 将 0x00010005 写入 ITM 跟踪控制寄存器，以使能 ITM（使能同步，ATB ID 不为 0x00）
- 将 0x1 写入 ITM 跟踪使能寄存器，以使能激励端口 0
- 将 0x1 写入 ITM 跟踪特权寄存器，以取消屏蔽激励端口 7:0
- 将要输出的值写入激励端口寄存器 0：此操作可用软件完成（使用 printf 函数）

51.15 ETM（嵌入式跟踪宏单元）

51.15.1 概述

ETM 能够重建程序执行。使用“数据观察点和跟踪”(DWT) 组件或“指令跟踪宏单元”(ITM) 跟踪数据，而使用“嵌入式跟踪宏单元”(ETM) 跟踪指令。

ETM 以数据包形式发送信息，由内置资源触发。这些资源必须单独编程，并且使用“触发事件寄存器”(0xE0041008) 选择触发源。事件可以是简单事件（地址比较器中的地址匹配），也可以是 2 个事件之间的逻辑运算结果。触发源是 DWT 模块的四个比较器之一，可以监视以下事件：

- 时钟周期匹配
- 数据地址匹配

有关触发资源的更多信息，请参见*第 51.13 节：DWT（数据观察点触发）*。

ETM 发送的数据包输出到 TPIU（跟踪端口接口单元）。TPIU 会添加一些额外的数据包（请参见*第 51.17 节：TPIU（跟踪端口接口单元）*），然后将整个数据包序列输出到调试主机。

51.15.2 信号协议和数据包类型

Arm® IHI 0014N 文档的第 7 章“ETMv3 信号协议”中介绍了这部分内容。

51.15.3 主要的 ETM 寄存器

有关寄存器的更多信息，请参见 Arm® IHI 0014N 规范的第 3 章。

表 339. 主要的 ETM 寄存器

地址	寄存器	详细信息
0xE0041FB0	ETM 锁定访问	写入 0xC5ACCE55 以解锁对其他 ETM 寄存器的写访问。
0xE0041000	ETM 控制	此寄存器控制 ETM 的一般操作，例如如何使能跟踪。
0xE0041010	ETM 状态	此寄存器提供有关跟踪和触发逻辑的当前状态的信息。
0xE0041008	ETM 触发事件	此寄存器定义控制触发的事件。
0xE004101C	ETM 跟踪使能控制	此寄存器定义选择哪个比较器。
0xE0041020	ETM 跟踪使能事件	此寄存器定义跟踪使能事件。
0xE0041024	ETM 跟踪启动/停止	此寄存器定义触发源启动和停止跟踪时分别使用的跟踪。

51.15.4 配置示例

从 TPIU 输出一个简单的数值：

- 配置 TPIU 并使能 I/IO_TRACEN，以分配 STM32L4Rxxx 和 STM32L4Sxxx 调试配置寄存器中的 TRACE I/O
- 将 0xC5ACCE55 写入 ETM 锁定访问寄存器，以解锁对 ITM 寄存器的写保护
- 将 0x00001D1E 写入控制寄存器（配置跟踪）
- 将 0000406F 写入触发事件寄存器（定义触发事件）
- 将 0000006F 写入跟踪使能事件寄存器（定义要启动/停止的事件）
- 将 00000001 写入跟踪启动/停止寄存器（使能跟踪）
- 将 0000191E 写入 ETM 控制寄存器（配置结束）

51.16 MCU 调试组件 (DBGMCU)

MCU 调试组件帮助调试器为以下各项提供支持：

- 低功耗模式
- 断点期间的定时器、看门狗、I^2C 和 bxCAN 的时钟控制
- 跟踪引脚分配的控制

51.16.1 对低功耗模式的调试支持

要进入低功耗模式，必须执行指令 WFI 或 WFE。

MCU 支持多个低功耗模式，这些模式可以禁止 CPU 时钟或降低 CPU 功耗。

内核不允许在调试会话期间关闭 FCLK 或 HCLK。由于调试期间需要使用它们进行调试连接，因此其必须保持激活状态。MCU 集成了特殊方法，允许用户在低功耗模式下调试软件。

为此，调试主机首先必须设置一些调试配置寄存器，以更改低功耗模式行为：

- 在睡眠模式下，调试主机必须事先将 DBGMCU_CR 寄存器的 DBG_SLEEP 位置 1。这样便可为 HCLK 和 FCLK 提供相同的时钟（之前配置的系统时钟）。
- 在停止模式下，调试主机必须事先将 DBG_STOP 位置 1。这样便可使能内部 RC 振荡器时钟，以在停止模式下为 FCLK 和 HCLK 提供时钟。
- 在待机模式下，调试主机必须事先将 DBG_STANDBY 位置 1。这样便可使稳压器保持开启并使能内部 RC 振荡器时钟，以在待机模式下为 FCLK 和 HCLK 提供时钟。会在内部产生系统复位，这样退出待机模式相当于取复位向量。

可在系统复位状态下用调试器写入 DBGMCU_CR 寄存器。如果调试主机不支持这些功能，仍然可以通过软件写入该寄存器。

51.16.2 对定时器、RTC、看门狗、bxCAN 和 I²C 的调试支持

断点期间，必须选择定时器、RTC 和看门狗的计数器的行为方式：

- 在产生断点时，计数器继续计数。例如，当 PWM 控制电机时，通常需要这种方式。
- 在产生断点时，计数器停止计数。用于看门狗时需要这种方式。

对于 bxCAN，用户可以选择在断点期间阻止更新接收寄存器。

对于 I²C，用户可以选择在断点期间阻止 SMBUS 超时。

可在系统复位状态下用调试器写入 DBGMCU 冻结寄存器。如果调试主机不支持这些功能，仍然可以通过软件写入这些寄存器。

51.16.3 调试 MCU 配置寄存器 (DBGMCU_CR)

Debug MCU configuration register

地址：0xE004 2004

上电复位：0x0000 0000

系统复位：不受影响

访问：仅支持 32 位访问

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.	Res.	Res.	Res.	Res.	Res.	Res.	TRACE_MODE [1:0]		TRACE_IOEN	Res.	Res.	DBG_STANDBY	DBG_STOP	DBG_SLEEP
								rw	rw	rw			rw	rw	rw

位 31:8 保留，必须保持复位值。

位 7:5 **TRACE_MODE[1:0] 和 TRACE_IOEN**：跟踪引脚分配控制 (Trace pin assignment control)

– TRACE_IOEN=0 时：
TRACE_MODE=xx：未分配 TRACE 引脚（默认状态）

– TRACE_IOEN=1 时：

– TRACE_MODE=00：异步模式下的 TRACE 引脚分配

– TRACE_MODE=01：同步模式下的 TRACE 引脚分配，TRACEDATA 大小为 1

– TRACE_MODE=10：同步模式下的 TRACE 引脚分配，TRACEDATA 大小为 2

– TRACE_MODE=11：同步模式下的 TRACE 引脚分配，TRACEDATA 大小为 4

位 4:3 保留，必须保持复位值。

位 2 **DBG_STANDBY**：调试待机模式 (Debug Standby mode)

0：（FCLK=关闭，HCLK=关闭）将整个数字部分断电。
从软件角度来看，退出待机模式相当于取复位向量（指示 MCU 从待机状态恢复的几个状态位除外）
1：（FCLK=开启，HCLK=开启）在这种情况下，数字部分不断电，FCLK 和 HCLK 由仍保持激活状态的内部 RC 振荡器提供。此外，MCU 会在待机模式期间产生系统复位，这样退出待机模式相当于取复位向量。

位 1 **DBG_STOP**：调试停止模式 (Debug Stop mode)

0：（FCLK=关闭，HCLK=关闭）在停机模式下，时钟控制器禁止所有时钟（包括 HCLK 和 FCLK）。退出停机模式时，时钟配置与复位后的时钟配置相同（CPU 时钟由 8 MHz 内部 RC 振荡器 (HSI16) 提供）。因此，软件必须重新编程时钟控制器以使能 PLL 和晶振等。
1：（FCLK=开启，HCLK=开启）在这种情况下，进入停机模式时，FCLK 和 HCLK 由在停机模式下仍保持激活状态的内部 RC 振荡器提供。退出停机模式时，软件必须重新编程时钟控制器以使能 PLL 和晶振等（操作步骤与在 DBG_STOP=0 时相同）

位 0 **DBG_SLEEP**：调试睡眠模式 (Debug Sleep mode)

0：（FCLK=开启，HCLK=关闭）在睡眠模式下，FCLK 由原先在 HCLK 禁止时通过软件配置好的系统时钟驱动。
在睡眠模式下，时钟控制器配置不复位，保持先前设置的状态。因此，退出睡眠模式时，软件不需要重新配置时钟控制器。
1：（FCLK=开启，HCLK=开启）在这种情况下，进入睡眠模式时，将为 HCLK 和 FCLK 馈送相同的时钟（软件先前配置的系统时钟）。

51.16.4 MCU APB1 调试冻结寄存器 1 (DBGMCU_APB1FZR1)

Debug MCU APB1 freeze register1

地址：0xE004 2008

上电复位 (POR)：0x0000 0000

系统复位：不受影响

访问：仅支持 32 位访问。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LPTIM1_STOP	Res.	Res.	Res.	Res.	Res.	DBG_CAN1_STOP	Res.	DBG_I2C3_STOP	DBG_I2C2_STOP	DBG_I2C1_STOP	Res.	Res.	Res.	Res.	Res.
rw						rw		rw	rw	rw					
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	DBG_IWDG_STOP	DBG_WWDG_STOP	DBG_RTC_STOP	Res.	Res.	Res.	Res.	DBG_TIM7_STOP	DBG_TIM6_STOP	DBG_TIM5_STOP	DBG_TIM4_STOP	DBG_TIM3_STOP	DBG_TIM2_STOP
			rw	rw	rw					rw	rw	rw	rw	rw	rw

位 31 **DBG_LPTIM1_STOP**：内核停止时 LPTIM1 计数器停止 (LPTIM1 counter stopped when core is halted)
0：即使内核停止，仍然馈送 LPTIM1 计数器时钟
1：内核停止时，LPTIM1 计数器时钟停止

位 30:26 保留，必须保持复位值。

位 25 **DBG_CAN1_STOP**：内核停止时 bxCAN1 停止 (bxCAN1 stopped when core is halted)
0：行为方式与正常模式下相同
1：冻结 bxCAN1 接收寄存器

位 24 保留，必须保持复位值。

位 23 **DBG_I2C3_STOP**：内核停止时停止 I2C3 SMBUS 超时计数器 (I2C3 SMBUS timeout counter stopped when core is halted)
0：行为方式与正常模式下相同
1：冻结 I2C3 SMBus 超时

位 22 **DBG_I2C2_STOP**：内核停止时停止 I2C2 SMBUS 超时计数器 (I2C2 SMBUS timeout counter stopped when core is halted)
0：行为方式与正常模式下相同
1：冻结 I2C2 SMBus 超时

位 21 **DBG_I2C1_STOP**：内核停止时停止 I2C1 SMBUS 超时计数器 (I2C1 SMBUS timeout counter stopped when core is halted)
0：行为方式与正常模式下相同
1：冻结 I2C1 SMBus 超时

位 20:13 保留，必须保持复位值。

位 12 **DBG_IWDG_STOP**：内核停止时停止独立看门狗计时器 (Independent watchdog counter stopped when core is halted)
0：即使内核停止，独立看门狗计数器时钟仍继续工作
1：内核停止时，独立看门狗计数器时钟停止

位 11 **DBG_WWDG_STOP**：内核停止时停止窗口看门狗计时器 (Window watchdog counter stopped when core is halted)
0：即使内核停止，窗口看门狗计数器时钟仍继续工作
1：内核停止时，窗口看门狗计数器时钟停止

位 10 **DBG_RTC_STOP**：内核停止时 RTC 计数器停止 (RTC counter stopped when core is halted)
0：即使内核停止，仍然馈送 RTC 计数器时钟
1：内核停止时，RTC 计数器时钟停止

位 9:6 保留，必须保持复位值。

位 5 **DBG_TIM7_STOP**：内核停止时 TIM7 计数器停止 (TIM7 counter stopped when core is halted)
0：即使内核停止，仍然馈送 TIM7 计数器时钟
1：内核停止时，TIM7 计数器时钟停止

位 4 **DBG_TIM6_STOP**：内核停止时 TIM6 计数器停止 (TIM6 counter stopped when core is halted)
0：即使内核停止，仍然馈送 TIM6 计数器时钟
1：内核停止时，TIM6 计数器时钟停止

位 3 **DBG_TIM5_STOP**：内核停止时 TIM5 计数器停止 (TIM5 counter stopped when core is halted)
0：即使内核停止，仍然馈送 TIM5 计数器时钟
1：内核停止时，TIM5 计数器时钟停止

位 2 **DBG_TIM4_STOP**：内核停止时 TIM4 计数器停止 (TIM4 counter stopped when core is halted)
0：即使内核停止，仍然馈送 TIM4 计数器时钟
1：内核停止时，TIM4 计数器时钟停止

位 1 **DBG_TIM3_STOP**：内核停止时 TIM3 计数器停止 (TIM3 counter stopped when core is halted)
0：即使内核停止，仍然馈送 TIM3 计数器时钟
1：内核停止时，TIM3 计数器时钟停止

位 0 **DBG_TIM2_STOP**：内核停止时 TIM2 计数器停止 (TIM2 counter stopped when core is halted)
0：即使内核停止，仍然馈送 TIM2 计数器时钟
1：内核停止时，TIM2 计数器时钟停止

51.16.5 调试 MCU APB1 冻结寄存器 2 (DBGMCU_APB1FZR2)

Debug MCU APB1 freeze register 2

地址：0xE004 200C

上电复位 (POR)：0x0000 0000

系统复位：不受影响

访问：仅支持 32 位访问。

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.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DBG_LPTIM2_STOP	Res.	Res.	Res.	DBG_I2C4_STOP	Res.
										rw				rw	

位 31:6 保留，必须保持复位值。

位 5 **DBG_LPTIM2_STOP**：内核停止时 LPTIM2 计数器停止 (LPTIM2 counter stopped when core is halted)
0：即使内核停止，仍然馈送 LPTIM2 计数器时钟
1：内核停止时，LPTIM2 计数器时钟停止

位 4:2 保留，必须保持复位值

位 1 **DBG_I2C4_STOP**：内核停止时停止 I2C4 SMBUS 超时计数器 (I2C4 SMBUS timeout counter stopped when core is halted)
0：行为方式与正常模式下相同
1：冻结 I2C4 SMBus 超时

位 0 保留，必须保持复位值

51.16.6 调试 MCU APB2 冻结寄存器 (DBGMCU_APB2FZR)

Debug MCU APB2 freeze register

地址：0xE004 2010

上电复位 (POR)：0x0000 0000

系统复位：不受影响

访问：仅支持 32 位访问。

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.	DBG_TIM17_STOP	DBG_TIM16_STOP	DBG_TIM15_STOP
													rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	DBG_TIM8_STOP	Res.	DBG_TIM1_STOP	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
		rw		rw											

位 31:19 保留，必须保持复位值。

位 18 **DBG_TIM17_STOP**：内核停止时 TIM17 计数器停止 (TIM17 counter stopped when core is halted)
0：即使内核停止，仍然馈送 TIM17 计数器时钟
1：内核停止时，TIM17 计数器时钟停止

位 17 **DBG_TIM16_STOP**：内核停止时 TIM16 计数器停止 (TIM16 counter stopped when core is halted)
0：即使内核停止，仍然馈送 TIM16 计数器时钟
1：内核停止时，TIM16 计数器时钟停止

位 16 **DBG_TIM15_STOP**：内核停止时 TIM15 计数器停止 (TIM15 counter stopped when core is halted)
0：即使内核停止，仍然馈送 TIM15 计数器时钟
1：内核停止时，TIM15 计数器时钟停止

位 15:14 保留，必须保持复位值。

位 13 **DBG_TIM8_STOP**：内核停止时 TIM8 计数器停止 (TIM8 counter stopped when core is halted)
0：即使内核停止，仍然馈送 TIM8 计数器时钟
1：内核停止时，TIM8 计数器时钟停止

位 12 保留，必须保持复位值。

位 11 **DBG_TIM1_STOP**：内核停止时 TIM1 计数器停止 (TIM1 counter stopped when core is halted)
0：即使内核停止，仍然馈送 TIM1 计数器时钟
1：内核停止时，TIM1 计数器时钟停止

位 10:0 保留，必须保持复位值。

51.17 TPIU（跟踪端口接口单元）

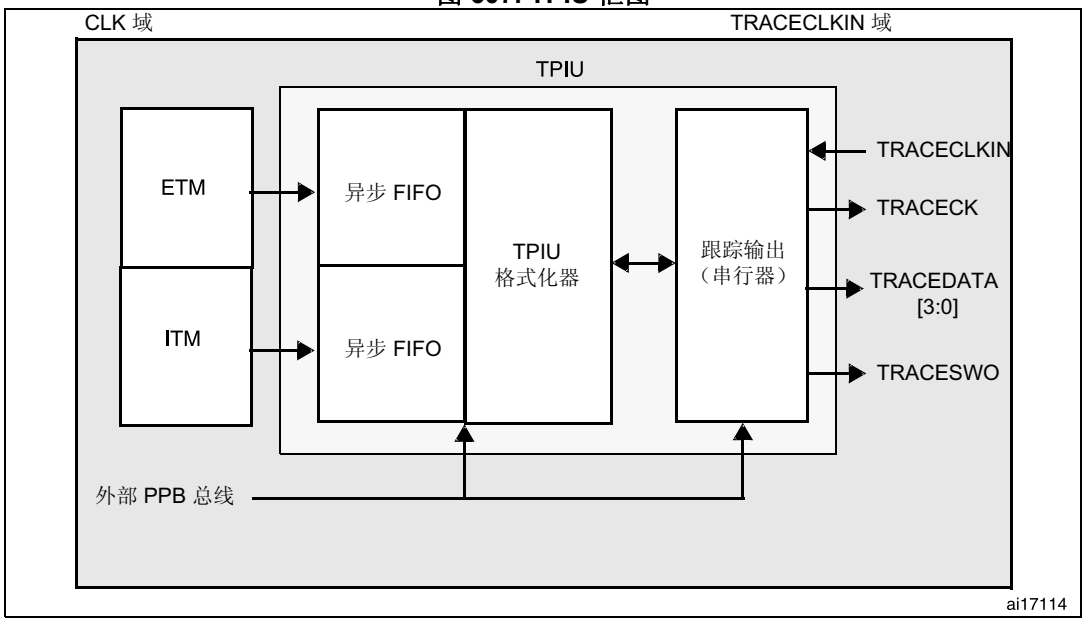
51.17.1 简介

TPIU 用作 ITM 和 ETM 中的片上跟踪数据之间的桥接器。

输出数据流封装成跟踪源 ID，然后被*跟踪端口分析器* (TPA) 捕获。

内核内置专为低成本调试设计的简单 TPIU（由特殊版本的 CoreSight TPIU 组成）。

图 607. TPIU 框图



51.17.2 TRACE 引脚分配

- 异步模式

  异步模式需要 1 个额外引脚，并且适用于所有封装。仅在使用串线行模式时异步模式才可用（在 JTAG 模式下不可用）。

表 340. 异步 TRACE 引脚分配

TPUI 引脚名称	跟踪同步模式		STM32L4Rxxx 和 STM32L4Sxxx 引脚分配
	类型	说明	
TRACESWO	O	TRACE 异步数据输出	PB3

- 同步模式

  同步模式需要 2 到 6 个额外引脚，具体取决于所跟踪数据的长度，并且仅适用于较大型的封装。此外，同步模式在 JTAG 模式和串行模式下均可用，并可提供比异步跟踪更高的带宽输出能力。

表 341. 同步 TRACE 引脚分配

TPUI 引脚名称	跟踪同步模式		STM32L4Rxxx 和 STM32L4Sxxx 引脚分配
	类型	说明	
TRACECK	O	TRACE 时钟	PE2
TRACED[3:0]	O	TRACE 同步数据输出 可以是 1、2 或 4。	PE[6:3] PC1、PC8、PD2、PC12

TPUI TRACE 引脚分配

默认情况下，不分配这些引脚。可以通过设置 *调试 MCU 配置寄存器 (DBGMCU_CR)* 中的 TRACE_IOEN 和 TRACE_MODE 位来分配这些引脚。必须由调试主机来完成此配置。

此外，要分配的引脚数目取决于跟踪配置（异步跟踪或同步跟踪）。

- **异步模式：**需要 1 个额外引脚
- **同步模式：**需要 2 到 5 个额外引脚，具体数目取决于数据跟踪端口寄存器的数据长度（1、2 或 4）：
  - TRACECK
  - TRACED(0)（如果端口数据长度配置为 1、2 或 4）
  - TRACED(1)（如果端口数据长度配置为 2 或 4）
  - TRACED(2)（如果端口数据长度配置为 4）
  - TRACED(3)（如果端口数据长度配置为 4）

要分配 TRACE 引脚，调试主机必须对 *调试 MCU 配置寄存器 (DBGMCU_CR)* 的 TRACE_IOEN 和 TRACE_MODE[1:0] 位进行编程。默认情况下不分配 TRACE 引脚。

此寄存器映射到外部 PPB 上，通过 PORESET（而非系统复位）复位。可在系统复位状态下用调试器写入该寄存器。

表 342. 灵活的 TRACE 引脚分配

DBGMCU_CR 寄存器		引脚用途：	分配的 TRACE IO 引脚					
TRACE_IOEN	TRACE_MODE [1:0]		PB3 / JTDO/ TRACESWO	PE2 / TRACECK	PE3 / TRACED[0]	PE4 / TRACED[1]	PE5 / TRACED[2]	PE6 / TRACED[3]
0	XX	无跟踪（默认状态）	已释放 (1)	-				
1	00	异步跟踪	TRACESWO	-	-	已释放（可用作 GPIO）		

表 342. 灵活的 TRACE 引脚分配（续）

DBGMCU_CR 寄存器		引脚用途：	分配的 TRACE IO 引脚					
TRACE_IOEN	TRACE_MODE [1:0]		PB3 / JTDO/ TRACESWO	PE2 / TRACECK	PE3 / TRACED[0]	PE4 / TRACED[1]	PE5 / TRACED[2]	PE6 / TRACED[3]
1	01	同步跟踪 1 位	已释放 (1)	TRACECK	TRACED[0]	-	-	-
1	10	同步跟踪 2 位		TRACECK	TRACED[0]	TRACED[1]	-	-
1	11	同步跟踪 4 位		TRACECK	TRACED[0]	TRACED[1]	TRACED[2]	TRACED[3]

1. 使用串行线模式时释放此引脚，但使用 JTAG 时，此引脚分配给 JTDO。

注： *默认情况下，TPIU 的 TRACECLKIN 输入时钟接地。因此，将在 TRACE_IOEN 位置 1 后的两个时钟周期将该时钟分配给 HCLK。*

调试器必须通过对 TPIU 的 SPP_R（选择引脚协议）寄存器的 PROTOCOL[1:0] 位执行写操作来配置跟踪模式。

- PROTOCOL=00：跟踪端口模式（同步）
- PROTOCOL=01 或 10：串行线（曼彻斯特或 NRZ）模式（异步模式）。
  默认状态为 01

随后，它还会通过对 TPIU 的 CPSPS_R（当前同步端口大小寄存器）中的位 [3:0] 执行写操作来配置跟踪端口大小：

- 0x1 表示 1 个引脚（默认状态）
- 0x2 表示 2 个引脚
- 0x8 表示 4 个引脚

51.17.3 TPUI 格式化器

格式化器协议以 16 字节帧形式输出数据：

- 七个字节的数据
- 八个字节的多用字节，包括：
  - 1 位 (LSB) 指示该字节是数据字节 (‘0) 还是 ID 字节 (‘1)。
  - 7 位 (MSB)，既可以是数据，也可用于更改源 ID 跟踪。
- 一个字节的辅助位，每个辅助位对应于八个多用字节中的一个：
  - 如果对应的字节是数据，则此位表示该数据的位 0。
  - 如果对应的字节是 ID 更改，则此位表示 ID 更改生效的时间。

注： *有关更多信息，请参见《Arm® CoreSight 架构规范 v1.0》(Arm IHI 0029B)*

51.17.4 TPUI 帧同步数据包

TPUI 可产生两种类型的同步数据包：

- 帧同步数据包（或全字同步数据包）

  它包含字：0x7F_FF_FF_FF（首先发送 LSB）。只要未使用 ID 源代码 0x7F，就不会在任何其他时间出现此序列。

  它会在各数据帧*之间*定期输出。

  在连续模式下，找到同步帧后，TPA 必须丢弃所有这些帧。

- 半字同步数据包

  它包含半字：0x7F_FF（首先发送 LSB）。

  它会在各数据帧*之间或内部*定期输出。

  这些数据包仅在连续模式下生成，能够使 TPA 检测到跟踪端口是否处于空闲模式（没有要捕获的 TRACE）。当 TPA 检测到这些数据包后，必须将其丢弃。

51.17.5 发送同步帧数据包

内核的 TPIU 中没有同步计数器寄存器。因此，只能通过 **DWT** 生成同步触发。请参见寄存器 DWT 控制寄存器（位 SYNCTAP[11:10]）和 DWT 当前 PC 采样器循环计数寄存器。

发送 TPUI 帧同步数据包 (0x7F_FF_FF_FF) 的条件：

- 每次 TPIU 复位释放后。此复位在 TRACECLKIN 时钟出现上升沿时同步释放。这意味着，当 DBGMCU_CFG 寄存器中的 TRACE_IOEN 位置 1 时传送此数据包。在这种情况下，字 0x7F_FF_FF_FF 后面不带任何格式化的数据包。
- 在每次触发 DWT 时（假设先前已配置 DWT）。存在两种情况：
  - 如果复位 ITM 的 SYNENA 位复位，则仅发送字 0x7F_FF_FF_FF，后面不带任何格式化的数据流。
  - 如果 ITM 的 SYNENA 位置 1，则 ITM 同步数据包将跟在 (0x80_00_00_00_00_00) 后面，并由 TPUI 格式化（添加跟踪源 ID）。

51.17.6 同步模式

跟踪数据输出大小可配置为 4 个、2 个或 1 个引脚：TRACED(3:0)

输出时钟输出到调试器 (TRACECK)

其中，TRACECLKIN 由内部驱动，并且仅在使用 TRACE 时连接到 HCLK。

注： *在此同步模式下，不需要提供稳定的时钟频率。*

TRACE I/O（包括 TRACECK）由 TRACLKIN 的上升沿驱动（等于 HCLK）。因此，TRACECK 的输出频率等于 HCLK/2。

51.17.7 异步模式

这是一种只使用 1 个引脚输出跟踪的低成本备用方案：该引脚是异步输出引脚 TRACESWO。显然，这种方案的带宽有限。

使用 SW-DP 引脚时，TRACESWO 与 JTDO 复用。这样，所有 STM32L4Rxxx 和 STM32L4Sxxx 封装都可提供此功能。

此异步模式要求 TRACECLKIN 具有恒定的频率。对于标准 UART (NRZ) 捕获机制，需要 5% 的精度。曼彻斯特编码的容限最高可达 10%。

51.17.8 STM32L4Rxxx 和 STM32L4Sxxx 内的 TRACECLKIN 连接 STM32L4Rxxx 和 STM32L4Sxxx

在 STM32L4Rxxx 和 STM32L4Sxxx 中，此 TRACECLKIN 输入内部连接到 HCLK。这意味着，在异步跟踪模式下，应用程序应限制使用时间帧保证 CPU 频率的稳定。

注： ***重要提示：****使用异步跟踪时，了解以下内容非常重要：*

STM32L4Rxxx 和 STM32L4Sxxx MCU 的默认时钟是内部 RC 振荡器。复位状态下的频率与复位释放后的频率不同。原因是，由于系统复位状态下默认采用 RC 校准值，而在每次系统复位释放时会更新该 RC 校准值。

因此，跟踪端口分析器 (TPA) 在系统复位状态下不应使能跟踪（使用 TRACE_IOEN 位），原因是，在复位状态下的同步帧包的比特宽度与复位后的包不同。

51.17.9 TPIU 寄存器

仅当调试异常和监视控制寄存器 (DEMCR) 的位 TRCENA 置 1 时才能对 TPIU APB 寄存器进行读写操作。否则，这些寄存器将读为零（此位的输出会使能 TPIU 的 PCLK）。

表 343. 重要的 TPIU 寄存器

地址	寄存器	说明
0xE0040004	当前端口大小	允许选择跟踪端口大小： 位 0：端口大小 = 1 位 1：端口大小 = 2 位 2：端口大小 = 3，不支持 位 3：端口大小 = 4 只有 1 位必须置 1。默认情况下，端口大小为一位。(0x00000001)
0xE00400F0	选择引脚协议	允许选择跟踪端口协议： 位 1:0= 00：同步跟踪端口模式 01：串行线输出 - 曼彻斯特（默认值） 10：串行线输出 - NRZ 11：保留
0xE0040304	格式化器和刷新控制	位 31-9 = 始终为“0” 位 8 = TrigIn = 始终为“1”，表明已指示这些触发 位 7-4 = 始终为 0 位 3-2 = 始终为 0 位 1 = EnFCont。在同步跟踪模式下（选择引脚协议寄存器位 1:0=00），此位会强制置为 1：在连续模式下自动使能格式化器。在异步模式下（选择引脚协议寄存器位 1:0 <> 00），可写入该位以激活或取消激活格式化器。 位 0 = 始终为“0” 产生的默认值为 0x102 **注意：**在同步模式下，由于不会在片外映射 TRACECTL 引脚，因此在连续模式下将始终使能格式化器。这样，格式化器便可插入一些控制数据包来标识跟踪数据包的源。
0xE0040300	格式化器和刷新状态	不适用于 Cortex®-M4，始终读为 0x00000008

51.17.10 配置示例

- 将调试异常和监视控制寄存器 (DEMCR) 中的位 TRCENA 置 1
- 将所需值写入 TPIU 当前端口大小寄存器（对于 1 位端口大小，默认值为 0x1）
- 将 0x102 写入 TPIU 格式化器和刷新控制寄存器（默认值）
- 写入 TPIU 选择引脚协议以选择同步模式或异步模式。示例：0x2 表示异步 NRZ 模式（类似于 UART）
- 将 0x20 写入 DBGMCU 控制寄存器（位 IO_TRACEN），为异步模式分配 TRACE I/O。此时发送 TPIU 同步数据包 (FF_FF_FF_7F)
- 配置 ITM 并对 ITM 激励寄存器进行写操作以输出值

51.18 DBG 寄存器映射

下表对调试寄存器进行了汇总

表 344. DBG 寄存器映射和复位值

地址	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
0xE0042000	DBGMCU_IDCODE	REV_ID																Res.	Res.	Res.	Res.	DEV_ID											
	Reset value(1)	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X					X	X	X	X	X	X	X	X	X	X	X	X
0xE0042004	DBGMCU_CR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TRACE_MODE[1:0]		TRACE_IOEN	Res.	Res.	DBG_STANDBY	DBG_STOP	DBG_SLEEP
	Reset value																									0	0	0			0	0	0
0xE004 2008	DBGMCU_APB1FZR1	DBG_LPTIM1_STOP	Res.	Res.	Res.	Res.	Res.	DBG_CAN1_STOP	Res.	DBG_I2C3_STOP	DBG_I2C2_STOP	DBG_I2C1_STOP	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DBG_IWDG_STOP	DBG_WWDG_STOP	DBG_RTC_STOP	Res.	Res.	Res.	Res.	DBG_TIM7_STOP	DBG_TIM6_STOP	DBG_TIM5_STOP	DBG_TIM4_STOP	DBG_TIM3_STOP	DBG_TIM2_STOP
	Reset value	0						0		0	0	0									0	0	0					0	0	0	0	0	0
0xE004 200C	DBGMCU_APB1FZR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DBG_LPTIM2_STOP	Res.	Res.	Res.	DBG_I2C4_STOP	Res.
	Reset value																											0				0	
0xE004 2010	DBGMCU_APB2FZR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DBG_TIM17_STOP	DBG_TIM16_STOP	DBG_TIM15_STOP	Res.	Res.	Res.	Res.	DBG_TIM1_STOP	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	Reset value														0	0	0					0											

1. 复位值与产品有关。有关详细信息，请参见*第 51.6.1 节：MCU 器件 ID 代码*。

52 器件电子签名

器件电子签名存储在 Flash 模块的系统存储区中，可以使用调试接口或 CPU 对其进行读取。它包含出厂前编程的标识和校准数据，这些数据允许用户固件或其他外部器件与 STM32L4Rxxx 和 STM32L4Sxxx 微控制器的特性自动匹配。

52.1 唯一器件 ID 寄存器（96 位）

Unique device ID register

唯一器件标识符最适合：

- 用作序列号（例如 USB 字符串序列号或其他终端应用程序）
- 在对内部 Flash 进行编程前将唯一 ID 与软件加密原语和协议结合使用时用作安全密钥的一部分以提高 Flash 中代码的安全性
- 激活安全自举过程等

96 位的唯一器件标识符提供了一个对于任何器件和任何上下文都唯一的参考号码。用户不能更改这些位。

基址：0x1FFF 7590

偏移地址：0x00

只读 = 0xXXXX XXXX，其中 X 是出厂前编程的

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

位 31:0 **UID[31:0]**：晶圆上的 X 和 Y 坐标 (X and Y coordinates on the wafer)

偏移地址：0x04

只读 = 0xXXXX XXXX，其中 X 是出厂前编程的

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
UID[63:48]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
UID[47:32]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:8 **UID[63:40]**：LOT_NUM[23:0]
批号（ASCII 编码）

位 7:0 **UID[39:32]**：WAF_NUM[7:0]
晶圆编号（8 位无符号数）

偏移地址：0x08

只读 = 0xXXXX XXXX，其中 X 是出厂前编程的

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
UID[95:80]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
UID[79:64]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:0 **UID[95:64]**：LOT_NUM[55:24]
批号（ASCII 编码）

52.2 Flash 大小数据寄存器

Flash size data register

基址：0x1FFF 75E0

偏移地址：0x00

只读 = 0xXXXX，其中 X 是出厂前编程的

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

位 15:0 **FLASH_SIZE[15:0]**：Flash 大小 (Flash memory size)
此处的位域指示以 KB 表示的器件 Flash 大小。
例如，0x040 对应于 64 KB。

52.3 封装数据寄存器

Package data register

基址：0x1FFF 7500

偏移地址：0x00

只读 = 0xXXXX，其中 X 是出厂前编程的

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

位 15:5 保留，必须保持复位值

位 4:0 **PKG[4:0]**：封装类型 (Package type)

00010：LQFP100，无 DSI
00011：UFBGA132，无 DSI
001XX：LQFP144，无 DSI
10000：UFBGA169，无 DSI
10010：LQFP100，带 DSI
10011：UFBGA144，带 DSI；WLCSP144，带 DSI
10100：UFBGA169，带 DSI
10101：LQFP144，带 DSI
其他值：保留

53 版本历史

表 345. 文档版本历史

日期	版本	变更
2017 年 10 月 13 日	1	初始版本。
2018 年 2 月 5 日	2	增加了： – *第 3.7.17 节：Flash 配置寄存器 (FLASH_CFGR)* – *第 12.6.7 节：DMAMUX 大小标识寄存器 (DMAMUX_SIDR)* – *第 12.6.8 节：DMAMUX IP 标识寄存器 (DMAMUX_IPIDR)* – *第 12.6.9 节：DMAMUX 版本寄存器 (DMAMUX_VERR)* – *第 12.6.10 节：DMAMUX 硬件配置 1 寄存器 (DMAMUX_HWCFGR1)* – *第 12.6.11 节：DMAMUX 硬件配置 2 寄存器 (DMAMUX_HWCFGR2)* – *第 19.2.4 节：常规命令模式与 HyperBus™ 的通用功能*中的 *CS 边界、刷新和通信调整功能* – *第 19.4.3 节：OCTOSPI 常规命令模式配置*中的 *状态标志轮询模式配置* – *第 19.4.7 节：OCTOSPI 重新配置或禁用* – *第 19.6.5 节：OCTOSPI 器件配置寄存器 4 (OCTOSPI_DCR4)* – *第 38.4.13 节：定时器计数器复位* – *第 50.4.2 节：USB OTG 引脚和内部信号* – *表 319：STM32 低功耗模式与 OTG 的兼容性* *图 584：动态更新 OTG_HFIR (RLDCTRL = 0)* 删除了： – CS 高电平时间和时钟控制 CS 高电平定时器图 – OCTOSPI 寄存器：OCTOSPI HW 配置寄存器、(OCTOSPI_HWCFGR)、OCTOSPI 版本寄存器 (OCTOSPI_VER)、OCTOSPI 标识 (OCTOSPI_ID)、OCTOSPI HW 魔术 ID (OCTOSPI_MID)

表 345. 文档版本历史（续）

日期	版本	变更
2018 年 2 月 5 日	2 续	更新（续）： – *第 1.1 节：寄存器相关缩写词列表* – *第 2.4 节：嵌入式 SRAM* – *第 2.6.1 节：自举配置* – *第 3.3.2 节：误码校正 (ECC)* – *表 15：Flash 接口—— 寄存器映射和复位值* – *第 5.1 节：电源* – *图 7：STM32L4S5xx/R5xx 和 STM32L4S7xx/L4R7xx 电源概述* – *图 8：STM32L4S9xx/L4R9xx 电源概述* – *第 5.3.9 节：待机模式* – *第 5.3.10 节：关断模式* – *第 5.3.9 节：待机模式* – *第 6.2 节：时钟* – *第 6.2.11 节：LSE 上的时钟安全系统* – *第 6.4.4 节：PLL 配置寄存器 (RCC_PLLCFGR)* – *第 8.3.8 节：外部中断线/唤醒线* – *第 8.4.5 节：GPIO 端口输入数据寄存器 (GPIOx_IDR) (x = A..I)* – *表 41：STM32L4Rxxx 和 STM32L4Sxxx 外设互连矩阵* – *第 10.3.3 节：从 ADC 到定时器 (TIM1/TIM8)* – *第 10.3.5 节：从定时器 (TIM1/TIM3/TIM4/TIM6/TIM7/TIM8/TIM16/LPTIM1/LPTIM2) 和 EXTI 到 DFSDM1* – *表 48：DMAMUX：复用器输入到资源的分配* – *第 12.4.4 节：DMAMUX 请求线复用器* – *第 12.4.5 节：DMAMUX 请求发生器* – *第 12.6.1 节：DMAMUX 请求线复用器通道 x 配置寄存器 (DMAMUX_CxCR)* – *第 19.6.7 节：DMAMUX1 请求发生器通道 x 配置寄存器 (DMAMUX1_RGxCR)* – *图 36：虚拟缓冲区和物理缓冲区存储器映射* – *第 19.2.1 节：OCTOSPI 常规命令模式*中的*空指令周期阶段* – *第 19.3.1 节：OCTOSPI 间接模式* – *第 19.3.2 节：OCTOSPI 状态标志轮询模式* – *第 19.3.3 节：OCTOSPI 内存映射模式* – *第 19.4.1 节：OCTOSPI 系统配置* – *第 19.4.2 节：OCTOSPI 器件配置* – *第 19.4.3 节：OCTOSPI 常规命令模式配置*中的*内存映射模式配置* – *第 19.4.6 节：OCTOSPI 的繁忙位和中止功能* – *第 19.6.1 节：OCTOSPI 控制寄存器 (OCTOSPI_CR)* – *第 19.6.2 节：OCTOSPI 器件配置寄存器 1 (OCTOSPI_DCR1)*

表 345. 文档版本历史（续）

日期	版本	变更
2018 年 2 月 5 日	2 续	更新（续）： – *第 19.6.3 节：OCTOSPI 器件配置寄存器 2 (OCTOSPI_DCR2)* – *第 19.6.4 节：OCTOSPI 器件配置寄存器 3 (OCTOSPI_DCR3)* – *第 19.6.14 节：OCTOSPI 通信配置寄存器 (OCTOSPI_CCR)* – *第 19.6.15 节：OCTOSPI 时序配置寄存器 (OCTOSPI_TCR)* – *第 19.6.19 节：OCTOSPI 回卷通信配置寄存器 (OCTOSPI_WPCCR)* – *第 19.6.20 节：OCTOSPI 回卷时序配置寄存器 (OCTOSPI_WPTCR)* – *第 19.6.21 节：OCTOSPI 回卷指令寄存器 (OCTOSPI_WPIR)* – *第 19.6.22 节：OCTOSPI 回卷交替字节寄存器 (OCTOSPI_WPABR)* – *第 19.6.24 节：OCTOSPI 写入时序配置寄存器 (OCTOSPI_WTCR)* – *第 19.6.25 节：OCTOSPI 写入指令寄存器 (OCTOSPI_WIR)* – *第 19.6.26 节：OCTOSPI 写入交替字节寄存器 (OCTOSPI_WABR)* – *第 19.6.27 节：OCTOSPI HyperBus™ 延迟配置寄存器 (OCTOSPI_HLCR)* – *第 19.6.28 节：OCTOSPI 寄存器映射* – *表 112：ADC 内部输入/输出信号* – *表 113：ADC 输入/输出引脚* – *图 85：ADC1 连接功能* – *第 21.4.7 节：单端通道和差分输入通道* – *表 119：偏移计算与数据分辨率* – *第 21.6 节：ADC 寄存器*（所有寄存器命名更新，除非另有说明，否则不对寄存器内容进行更新） – *第 21.4.29 节：模拟窗口看门狗（AWD1EN、JAWD1EN、AWD1SGL、AWD1CH、AWD2CH、AWD3CH、AWD_HTx、AWD_LTx、AWDx）* – *第 22.7.15 节：DAC 校准控制寄存器 (DAC_CCR)* – *第 23.7 节：DCMI 寄存器说明*（更新了本节所有寄存器的复位值格式） – *表 163：DFSDM 断路连接* – *第 27.4.4 节：串行通道收发器*中的*通道输入选择* – *第 27.7 节：DFSDM 通道 y 寄存器 (y=0..7)*（所有寄存器命名更新，除非另有说明，否则不对寄存器内容进行更新） – *第 27.8.13 节：DFSDM 滤波器 x 极值检测器最大值寄存器 (DFSDM_FLTxEXMAX)*

表 345. 文档版本历史（续）

日期	版本	变更
2018 年 2 月 5 日	2 续	更新（续）： – *第 28.6 节：LTDC 编程步骤* – *第 28.7.21 节：LTDC 第 x 层混合系数配置寄存器 (LTDC_LxBFCR)* – *第 30.6.10 节：TSC I/O 组 x 计数器寄存器 (TSC_IOGxCR)* – *第 34.4.23 节：TIM1/TIM8 捕获/比较模式寄存器 3 (TIMx_CCMR3)* – *图 341：捕获/比较通道 1 主电路* – *图 342：捕获/比较通道的输出阶段（通道 1）* – *图 408：低功耗定时器框图* – *第 38.7.5 节：LPTIM 控制寄存器 (LPTIM_CR)* – *第 40.4.4 节：状态寄存器 (IWDG_SR)* – *图 417：看门狗框图* – *图 418：窗口看门狗时序图* – *第 42.3.11 节：RTC 参考时钟检测* – *第 42.6.3 节：RTC 控制寄存器 (RTC_CR)* – *第 43 节：内部集成电路 (I2C) 接口*：所有章节更新了 SMBus 版本参考 – *第 43.2 节：I2C 主要特性* – *表 246：I2C 特性实现* – *图 420：I2C 框图* – *图 422：建立和保持时序* – *第 43.4.1 节：I2C 框图* – *第 43.6 节：I2C 中断* – *表 264：BRR [3:0] = 0000 时的 USART 接收器容差* – *表 265：BRR[3:0] 不等于 0000 时的 USART 接收器容差* – *第 44.7 节：USART 寄存器*在所有小节中，更新了与保留位/位域相关的注释 – *第 45.5 节：LPUART 寄存器*在所有小节中，更新了与保留位/位域相关的注释 – *第 47.3.12 节：SPDIF 输出*中的 *SPDIF 发生器模式下的时钟发生器编程* – *第 47.5.11 节：PDM 延迟寄存器 (SAI_PDMDLY)* – *图 563：CAN 网络拓扑结构* – *图 566：静默模式下的 bxCAN* – *图 569：发送邮箱状态* – *图 571：过滤器组尺度配置 - 寄存器构成* – *图 572：过滤器编号示例* – *图 575：位时序* – *图 573：筛选机制 - 示例* – *图 577：事件标志与中断产生* – *图 578：CAN 邮箱寄存器* – *第 50.1 节：简介* – *表 316：OTG 实现*

表 345. 文档版本历史（续）

日期	版本	变更
2018 年 2 月 5 日	2 续	更新（续）： – *图 579：全速 OTG 模块框图* – *第 50.7 节：USB 主机* – *第 50.9 节：OTG 低功耗模式* – *图 587：中断层级* – *第 50.15.1 节：OTG 控制和状态寄存器 (OTG_GOTGCTL)* – *第 50.15.4 节：OTG USB 配置寄存器 (OTG_GUSBCFG)* – *第 50.15.7 节：OTG 中断屏蔽寄存器 (OTG_GINTMSK)* – *第 50.15.8 节：OTG 接收状态调试读取/OTG 状态读取和出栈寄存器 (OTG_GRXSTSR/OTG_GRXSTSP)* – *第 50.15.18 节：OTG 设备 IN 端点发送 FIFO 大小寄存器 (OTG_DIEPTXFx)（x = 1..5，其中 x 为 FIFO 号）* – *第 50.15.21 节：OTG 主机帧间隔寄存器 (OTG_HFIR)* – *第 50.15.27 节：OTG 主机通道 x 特性寄存器 (OTG_HCCHARx)（x = 0..11，其中 x 表示通道编号）* – *第 50.15.28 节：OTG 主机通道 x 中断寄存器 (OTG_HCINTx)（x = 0..11，其中 x 表示通道编号）* – *第 50.15.29 节：OTG 主机通道 x 中断屏蔽寄存器 (OTG_HCINTMSKx)（x = 0..11，其中 x 表示通道编号）* – *第 50.15.32 节：OTG 设备配置寄存器 (OTG_DCFG)* – *第 50.15.35 节：OTG 设备 IN 端点通用中断屏蔽寄存器 (OTG_DIEPMSK)* – *第 50.15.35 节：OTG 设备 IN 端点通用中断屏蔽寄存器 (OTG_DIEPMSK)* – *第 50.15.44 节：OTG 设备 IN 端点 x 中断寄存器 (OTG_DIEPINTx)（x = 0..5，其中 x 表示端点编号）* – *第 50.15.49 节：OTG 设备 OUT 端点 x 中断寄存器 (OTG_DOEPINTx)（x = 0..5，其中 x 表示端点编号）* – *第 50.16.4 节：主机编程模型* – *图 600：A 设备 SRP* – *图 601：B 设备 SRP* – *表 342：灵活的 TRACE 引脚分配*

表 345. 文档版本历史（续）

日期	版本	变更
2018 年 3 月 5 日	3	增加了： – *第 22.3 节：DAC 实现* 更新了： – *第 1.1 节：寄存器相关缩写词列表* – *第 11 节：直接存储器访问控制器 (DMA)* 和所有小节内容 – *第 22.7 节：DAC 寄存器*：所有小节的命名规则 – *第 22.7.1 节：DAC 控制寄存器 (DAC_CR)* – *第 18.5.6 节：NOR/PSRAM 控制器寄存器* – *第 18.6.7 节：NAND Flash 控制器寄存器* – *表 107：FSMC 寄存器映射* – *第 28.7 节：LTDC 寄存器*：所有小节的命名规则和复位值格式 – *第 45.5.6 节：中断和状态寄存器 (LPUART_ISR)* – *第 45.5.7 节：中断标志清零寄存器 (LPUART_ICR)* – *表 274：LPUART 寄存器映射和复位值* – *第 50.8.1 节：主机 SOF* – *第 50.8.2 节：设备 SOF* – *第 50.11.3 节：FIFO RAM 分配* – *第 50.15 节：OTG_FS 寄存器*：所有小节的命名规则 删除了： – *第 12 节：DMA 请求复用器 (DMAMUX)* 中：DMAMUX 大小标识寄存器 (DMAMUX_SIDR)、DMAMUX IP 标识寄存器 (DMAMUX_IPIDR)、DMAMUX 版本寄存器 (DMAMUX_VERR)、DMAMUX 硬件配置 1 寄存器 (DMAMUX_HWCFGR1) 和 DMAMUX 硬件配置 2 寄存器 (DMAMUX_HWCFGR2)

索引

A

ADCx_AWD2CR 633
ADCx_AWD3CR 633
ADCx_CALFACT 634
ADCx_CCR 636
ADCx_CFGR 617
ADCx_CFGR2 621
ADCx_CR 614
ADCx_CSR 635
ADCx_DIFSEL 634
ADCx_DR 630
ADCx_IER 612
ADCx_ISR 610
ADCx_JDRy 632
ADCx_JSQR 630
ADCx_OFRy 631
ADCx_SMPR1 623
ADCx_SMPR2 623
ADCx_SQR1 626
ADCx_SQR2 627
ADCx_SQR3 628
ADCx_SQR4 629
ADCx_TR1 624
ADCx_TR2 625
ADCx_TR3 625
AES_CR 1004
AES_DINR 1008
AES_DOUTR 1008
AES_IVR0 1011
AES_IVR1 1012
AES_IVR2 1012
AES_IVR3 1013
AES_KEYR1 1010
AES_KEYR2 1010
AES_KEYR3 1011
AES_KEYR4 1013
AES_KEYR5 1014
AES_KEYR6 1014
AES_KEYR7 1015
AES_KEYRx 1009
AES_SR 1007
AES_SUSPxR 1015

C

CAN_BTR 1763
CAN_ESR 1762
CAN_FA1R 1773
CAN_FFA1R 1773
CAN_FiRx 1774
CAN_FM1R 1772
CAN_FMR 1771
CAN_FS1R 1772
CAN_IER 1761
CAN_MCR 1755
CAN_MSR 1756
CAN_RDHxR 1770
CAN_RDLxR 1770
CAN_RDTxR 1769
CAN_RF0R 1759
CAN_RF1R 1760
CAN_RIxR 1768
CAN_TDHxR 1767
CAN_TDLxR 1767
CAN_TDTxR 1766
CAN_TIxR 1765
CAN_TSR 1757
COMP1_CSR 709
COMP2_CSR 711
CRC_CR 435
CRC_DR 434
CRC_IDR 435
CRC_INIT 436
CRC_POL 436
CRS_CFGR 287
CRS_CR 286
CRS_ICR 290
CRS_ISR 288

D

DAC_CCR 669
DAC_CR 658
DAC_DHR12L1 663
DAC_DHR12L2 664
DAC_DHR12LD 666
DAC_DHR12R1 662
DAC_DHR12R2 664
DAC_DHR12RD 665
DAC_DHR8R1 663
DAC_DHR8R2 665
DAC_DHR8RD 666
DAC_DOR1 667
DAC_DOR2 667
DAC_MCR 669
DAC_SHHR 672
DAC_SHRR 673

DAC_SHSR1 671
DAC_SHSR2 671
DAC_SR 668
DAC_SWTRGR 662
DBGMCU_APB1FZR1 1945
DBGMCU_APB1FZR2 1947
DBGMCU_APB2FZR 1948
DBGMCU_CR 1944
DBGMCU_IDCODE 1931
DCMI_CR 687
DCMI_CWSIZE 697
DCMI_CWSTRT 697
DCMI_DR 698
DCMI_ESCR 695
DCMI_ESUR 696
DCMI_ICR 694
DCMI_IER 692
DCMI_MIS 693
DCMI_RIS 691
DCMI_SR 690
DFSDM_CHyAWSCDR 759
DFSDM_CHyCFGR1 755
DFSDM_CHyCFGR2 758
DFSDM_CHyDATINR 760
DFSDM_CHyDLYR 761
DFSDM_CHyWDATR 760
DFSDM_FLTxAWCFR 774
DFSDM_FLTxAWHTR 772
DFSDM_FLTxAWLTR 772
DFSDM_FLTxAWSR 773
DFSDM_FLTxCNVTIMR 775
DFSDM_FLTxCR1 762
DFSDM_FLTxCR2 765
DFSDM_FLTxEXMAX 774
DFSDM_FLTxEXMIN 775
DFSDM_FLTxFCR 769
DFSDM_FLTxICR 768
DFSDM_FLTxISR 766
DFSDM_FLTxJCHGR 769
DFSDM_FLTxJDATAR 770
DFSDM_FLTxRDATAR 771
DMA_CCRx 343
DMA_CMARx 347
DMA_CNDTRx 346
DMA_CPARx 347
DMA_IFCR 342
DMA_ISR 340
DMAMUX_CFR 361
DMAMUX_CSR 361
DMAMUX_CxCR 360
DMAMUX_RGCFR 363
DMAMUX_RGSR 363
DMAMUX_RGxCR 362
DSIHOST_CCR 874
DSIHOST_CLCR 894
DSIHOST_CLTCR 894
DSIHOST_CMCR 886
DSIHOST_CR 873
DSIHOST_DLTCR 895
DSIHOST_FIR0 905
DSIHOST_FIR1 907
DSIHOST_GHCR 888
DSIHOST_GPDR 888
DSIHOST_GPSR 889
DSIHOST_GVCIDR 878
DSIHOST_IER0 901
DSIHOST_IER1 904
DSIHOST_ISR0 899
DSIHOST_ISR1 900
DSIHOST_LCCR 885
DSIHOST_LCOLCR 875
DSIHOST_LCVCIDR 908
DSIHOST_LPCR 875
DSIHOST_LPMCCR 910
DSIHOST_LPMCR 876
DSIHOST_LVCIDR 874
DSIHOST_MCR 878
DSIHOST_PCONFR 896
DSIHOST_PCR 876
DSIHOST_PCTLR 895
DSIHOST_PSR 898
DSIHOST_PTTCR 897
DSIHOST_PUCR 896
DSIHOST_TCCR1 890
DSIHOST_TCCR2 891
DSIHOST_TCCR3 891
DSIHOST_TCCR4 892
DSIHOST_TCCR5 893
DSIHOST_TCCR6 893
DSIHOST_VCCCR 912
DSIHOST_VCCR 881
DSIHOST_VHBPCCR 914
DSIHOST_VHBPCR 882
DSIHOST_VHSACCR 913
DSIHOST_VHSACR 882
DSIHOST_VLCCR 914
DSIHOST_VLCR 883
DSIHOST_VMCCR 910
DSIHOST_VMCR 879
DSIHOST_VNPCCR 913
DSIHOST_VPCCR 912
DSIHOST_VPCR 880
DSIHOST_VR 873
DSIHOST_VSCR 908

DSIHOST_VVACCR 916
DSIHOST_VVACR 885
DSIHOST_VVBPCCR 915
DSIHOST_VVBPCR 884
DSIHOST_VVFPCCR 916
DSIHOST_VVFPCR 884
DSIHOST_VVSACCR 915
DSIHOST_VVSACR 883
DSIWRAP_CFGR 917
DSIWRAP_CR 918
DSIWRAP_IER 919
DSIWRAP_IFCR 921
DSIWRAP_ISR 920
DSIWRAP_PCR1 922
DSIWRAP_PCR2 924
DSIWRAP_PCR3 926
DSIWRAP_PCR4 927
DSIWRAP_PCR5 928
DSIWRAP_RPCR 928

E

EXTI_EMR1 424
EXTI_EMR2 428
EXTI_FTSR1 425
EXTI_FTSR2 429
EXTI_IMR1 424
EXTI_IMR2 427
EXTI_PR1 427
EXTI_PR2 430
EXTI_RTSR1 425
EXTI_RTSR2 428
EXTI_SWIER1 426
EXTI_SWIER2 429

F

FLASH_ACR 121
FLASH_CR 126
FLASH_ECCR 128
FLASH_KEYR 123
FLASH_OPTKEYR 123
FLASH_OPTR 130
FLASH_PCROP1ER 132
FLASH_PCROP1SR 132
FLASH_PCROP2ER 135
FLASH_PCROP2SR 134
FLASH_PDKEYR 122
FLASH_SR 124
FLASH_WRP1AR 133
FLASH_WRP1BR 134
FLASH_WRP2AR 135
FLASH_WRP2BR 136
FSMC_BCR1..4 473
FSMC_BTR1..4 475
FSMC_BWTR1..4 478
FSMC_ECCR 490
FSMC_PATT 489
FSMC_PCR 485
FSMC_PMEM 488
FSMC_SR 487
FW_CR 150
FW_CSL 147
FW_CSSA 146
FW_NVDSL 148
FW_NVDSSA 147
FW_VDSL 149
FW_VDSSA 148

G

GFXMMU_CR 407
GPIOx_AFRH 306
GPIOx_AFRL 305
GPIOx_BRR 306
GPIOx_BSRR 304
GPIOx_IDR 303
GPIOx_LCKR 304
GPIOx_MODER 301
GPIOx_ODR 303
GPIOx_OSPEEDR 302
GPIOx_OTYPER 301
GPIOx_PUPDR 302

H

HASH_CR 1030
HASH_CSR0 1038
HASH_CSRx 1038
HASH_DIN 1032
HASH_HR0 1034
HASH_HR1 1034
HASH_HR2 1035
HASH_HR3 1035
HASH_HR4 1035
HASH_HR5 1035
HASH_HR6 1036
HASH_HR7 1036
HASH_IMR 1036
HASH_SR 1037
HASH_STR 1033

I

I2C_CR1 1437
I2C_CR2 1440

I2C_ICR 1449
I2C_ISR 1447
I2C_OAR1 1443
I2C_OAR2 1444
I2C_PECR 1450
I2C_RXDR 1451
I2C_TIMEOUTR 1446
I2C_TIMINGR 1445
I2C_TXDR 1451
I2Cx_CR2 . . . . . 147-148, 150, 1440
IWDG_KR 1334
IWDG_PR 1335
IWDG_RLR 1336
IWDG_SR 1337
IWDG_WINR 1338

L

LPTIM_ARR 1327
LPTIM_CFGR 1323
LPTIM_CMP 1326
LPTIM_CNT 1327
LPTIM_CR 1325
LPTIM_ICR 1321
LPTIM_IER 1322
LPTIM_ISR 1320
LPTIM1_OR 1328
LPTIM2_OR 1328
LPUART_BRR 1563
LPUART_CR1 1556
LPUART_CR2 1559
LPUART_CR3 1560
LPUART_ICR 1569
LPUART_ISR 1564
LPUART_PRESC 1571
LPUART_RDR 1570
LPUART_RQR 1564
LPUART_TDR 1570
LTDC_AWCR 799
LTDC_BCCR 802
LTDC_BPCR 798
LTDC_CDSR 806
LTDC_CPSR 806
LTDC_GCR 800
LTDC_ICR 804
LTDC_IER 803
LTDC_ISR 804
LTDC_LIPCR 805
LTDC_LxBFCR 813
LTDC_LxCACR 811
LTDC_LxCFBAR 814
LTDC_LxCFBLNR 815
LTDC_LxCFBLR 814
LTDC_LxCKCR 810
LTDC_LxCLUTWR 816
LTDC_LxCR 807
LTDC_LxDCCR 812
LTDC_LxPFCR 810
LTDC_LxWHPCR 808
LTDC_LxWVPCR 809
LTDC_SRCR 802
LTDC_SSCR 798
LTDC_TWCR 800

O

OCTOSPI _PIR 529
OCTOSPI _PSMAR 528
OCTOSPI _PSMKR 528
OCTOSPI_ABR 533
OCTOSPI_AR 527
OCTOSPI_CCR 529
OCTOSPI_CR 518
OCTOSPI_DCR1 520
OCTOSPI_DCR2 522
OCTOSPI_DCR3 523
OCTOSPI_DLR 526
OCTOSPI_DR 527
OCTOSPI_FCR 525
OCTOSPI_IR 532
OCTOSPI_LPTR 533
OCTOSPI_TCR 532
OCTOSPI_WCCR 538
OCTOSPI_WPABR 537
OCTOSPI_WPCCR 534
OCTOSPI_WPIR 537
OCTOSPI_WPTCR 536
OCTOSPIIOM_PnCR) 549
OPAMP1_CSR 722
OPAMP1_LPOTR 724, 726
OPAMP1_OTR 723
OTG_CID 1829
OTG_DAINT 1854
OTG_DAINTMSK 1854
OTG_DCFG 1847
OTG_DCTL 1848
OTG_DIEPCTL0 1857
OTG_DIEPCTLx 1858
OTG_DIEPEMPMSK 1856
OTG_DIEPINTx 1860
OTG_DIEPMSK 1851
OTG_DIEPTSIZ0 1862
OTG_DIEPTSIZx 1863
OTG_DIEPTXF0 1825

OTG_DIEPTXFx 1836
OTG_DOEPCTL0 1864
OTG_DOEPCTLx 1868
OTG_DOEPINTx 1865
OTG_DOEPMSK 1852
OTG_DOEPTSIZ0 1867
OTG_DOEPTSIZx 1870
OTG_DSTS 1850
OTG_DTXFSTSx 1863
OTG_DVBUSDIS 1855
OTG_DVBUSPULSE 1856
OTG_GADPCTL 1833
OTG_GAHBCFG 1811
OTG_GCCFG 1827
OTG_GINTMSK 1820
OTG_GINTSTS 1816
OTG_GLPMCFG 1829
OTG_GOTGCTL 1807
OTG_GOTGINT 1810
OTG_GPWRDN 1833
OTG_GRSTCTL 1814
OTG_GRXFSIZ 1825
OTG_GRXSTSP 1823
OTG_GRXSTSR 1823
OTG_GUSBCFG 1812
OTG_HAINT 1840
OTG_HAINTMSK 1840
OTG_HCCHARx 1843
OTG_HCFG 1836
OTG_HCINTMSKx 1845
OTG_HCINTx 1844
OTG_HCTSIZx 1846
OTG_HFIR 1837
OTG_HFNUM 1838
OTG_HNPTXFSIZ 1825
OTG_HNPTXSTS 1826
OTG_HPRT 1841
OTG_HPTXFSIZ 1835
OTG_HPTXSTS 1839
OTG_PCGCCTL 1871

P

PWR_CR1 181
PWR_CR2 182
PWR_CR3 183
PWR_CR4 184
PWR_PDCRA 190
PWR_PDCRB 191
PWR_PDCRC 192
PWR_PDCRD 193
PWR_PDCRE 194
PWR_PDCRF 195, 198
PWR_PDCRG 196
PWR_PDCRH 197
PWR_PUCRA 189
PWR_PUCRB 190
PWR_PUCRC 191
PWR_PUCRD 192
PWR_PUCRE 193
PWR_PUCRF 194, 197
PWR_PUCRG 195
PWR_PUCRH 196
PWR_SCR 188
PWR_SR1 185
PWR_SR2 186

R

RCC_AHB1ENR 245
RCC_AHB1RSTR 237
RCC_AHB1SMENR 255
RCC_AHB2ENR 247
RCC_AHB2RSTR 238
RCC_AHB2SMENR 256
RCC_AHB3ENR 249
RCC_AHB3RSTR 240
RCC_AHB3SMENR 259
RCC_APB1ENR1 250
RCC_APB1ENR2 252
RCC_APB1RSTR1 241
RCC_APB1RSTR2 243
RCC_APB1SMENR1 260
RCC_APB1SMENR2 264
RCC_APB2ENR 253
RCC_APB2RSTR 244
RCC_APB2SMENR 265
RCC_BDCR 269
RCC_CCIPR 267
RCC_CFGR 221
RCC_CICR 235
RCC_CIER 232
RCC_CIFR 234
RCC_CR 218
RCC_CRRCR 273
RCC_CSR 271
RCC_ICSCR 221
RCC_PLLCFGR 224
RCC_PLLSAI1CFGR 227
RCC_PLLSAI2CFGR 230
RNG_CR 962
RNG_DR 964
RNG_SR 963
RTC_ALRMAR 1373

RTC_ALRMBR 1374
RTC_ALRMBSSR 1384
RTC_BKPxR 1385
RTC_CALR 1379
RTC_CR 1365
RTC_DR 1364
RTC_ISR 1368
RTC_OR 1385
RTC_PRER 1371
RTC_SHIFTR 1376
RTC_SSR 1375
RTC_TR 1363
RTC_TSDR 1378
RTC_TSSSR 1378
RTC_TSTR 1377
RTC_WPR 1375
RTC_WUTR 1372

S

SAI_ACLRFR 1653
SAI_ACR1 1643
SAI_ACR2 1646
SAI_ADR 1654
SAI_AFRCR 1648
SAI_AIM 1650
SAI_ASLOTR 1649
SAI_ASR 1651
SAI_BCLRFR 1653
SAI_BCR1 1643
SAI_BCR2 1646
SAI_BDR 1654
SAI_BFRCR 1648
SAI_BIM 1650
SAI_BSLOTR 1649
SAI_BSR 1651
SAI_GCR 1643
SAI_PDMCR 1655
SAI_PDMDLY 1656
SDMMC_ACKTIMER 1728
SDMMC_ARGR 1712
SDMMC_CLKCR 1710
SDMMC_CMDR 1713
SDMMC_DCNTR 1719
SDMMC_DCTRL 1717
SDMMC_DLENR 1716
SDMMC_DTIMER 1715
SDMMC_FIFOR 1728
SDMMC_ICR 1722
SDMMC_IDMABASE0R 1730
SDMMC_IDMABASE1R 1731
SDMMC_IDMABSIZER 1730
SDMMC_IDMACTRLR 1729
SDMMC_MASKR 1725
SDMMC_POWER 1709
SDMMC_RESPCMDR 1714
SDMMC_RESPxR 1715
SDMMC_STAR 1720
SPIx_CR1 1599
SPIx_CR2 1601
SPIx_CRCPR 1605
SPIx_DR 1605
SPIx_RXCRCR 1606
SPIx_SR 1604
SPIx_TXCRCR 1606
SYSCFG_CFGR1 310
SYSCFG_CFGR2 319
SYSCFG_EXTICR1 312
SYSCFG_EXTICR2 314
SYSCFG_EXTICR3 315
SYSCFG_EXTICR4 317
SYSCFG_MEMRMP 309
SYSCFG_SCSR 318
SYSCFG_SKR 320
SYSCFG_SWPR 320-321

T

TIM1_OR1 1126
TIM1_OR2 1129
TIM1_OR3 1131
TIM15_ARR 1263
TIM15_BDTR 1265
TIM15_CCER 1260
TIM15_CCMR1 1256
TIM15_CCR1 1264
TIM15_CCR2 1265
TIM15_CNT 1263
TIM15_CR1 1248
TIM15_CR2 1249
TIM15_DCR 1268
TIM15_DIER 1252
TIM15_DMAR 1268
TIM15_EGR 1255
TIM15_OR1 1269
TIM15_OR2 1269
TIM15_PSC 1263
TIM15_RCR 1264
TIM15_SMCR 1251
TIM15_SR 1254
TIM16_OR1 1289
TIM16_OR2 1290
TIM17_OR1 1291
TIM17_OR2 1291

TIM2_OR1 1206
TIM2_OR2 1207
TIM3_OR1 1206
TIM3_OR2 1207
TIM8_OR1 1126
TIM8_OR2 1132
TIM8_OR3 1133
TIMx_ARR 1118, 1202, 1284, 1306
TIMx_BDTR 1121, 1286
TIMx_CCER 1113, 1199, 1281
TIMx_CCMR1 . . . . . 1108, 1194, 1279
TIMx_CCMR2 . . . . . 1111, 1197
TIMx_CCMR3 1127
TIMx_CCR1 . . . . . 1119, 1202, 1285
TIMx_CCR2 . . . . . 1119, 1203
TIMx_CCR3 . . . . . 1120, 1203
TIMx_CCR4 . . . . . 1120, 1204
TIMx_CCR5 1128
TIMx_CCR6 1129
TIMx_CNT 1117, 1201, 1284, 1305
TIMx_CR1 . . . . . 1097, 1185, 1274, 1302
TIMx_CR2 . . . . . 1099, 1187, 1275, 1304
TIMx_DCR 1124, 1205, 1288
TIMx_DIER 1103, 1190, 1276, 1304
TIMx_DMAR 1125, 1205, 1289
TIMx_EGR 1107, 1193, 1278, 1305
TIMx_PSC 1117, 1201, 1284, 1306
TIMx_RCR 1118, 1285
TIMx_SMCR 1101, 1188
TIMx_SR 1105, 1191, 1277, 1305
TSC_CR 944
TSC_ICR 947
TSC_IER 946
TSC_IOASCR 949
TSC_IOCCR 950
TSC_IOGCSR 950
TSC_IOGxCR 951
TSC_IOHCR 948
TSC_IOSCR 949
TSC_ISR 948
通用定时器 1211

U

USART_BRR 1514
USART_CR1 1502
USART_CR2 1506
USART_CR3 1510
USART_GTPR 1514
USART_ICR 1523
USART_ISR 1517
USART_PRESC 1526
USART_RDR 1525
USART_RQR 1516
USART_RTOR 1515
USART_TDR 1525

V

VREFBUF_CCR 702
VREFBUF_CSR 701

W

WWDG_CFR 1344
WWDG_CR 1343
WWDG_SR 1344

重要通知 - 请仔细阅读

意法半导体公司及其子公司（“ST”）保留随时对ST 产品和/ 或本文档进行变更、更正、增强、修改和改进的权利，恕不另行通知。买方在订货之前应获取关于 ST 产品的最新信息。ST 产品的销售依照订单确认时的相关 ST 销售条款。

买方自行负责对 ST 产品的选择和使用， ST 概不承担与应用协助或买方产品设计相关的任何责任。

ST 不对任何知识产权进行任何明示或默示的授权或许可。

转售的ST 产品如有不同于此处提供的信息的规定，将导致ST 针对该产品授予的任何保证失效。

ST 和 ST 徽标是 ST 的商标。所有其他产品或服务名称均为其各自所有者的财产。

本文档中的信息取代本文档所有早期版本中提供的信息。本文档的中文版本为英文版本的翻译件，仅供参考之用；若中文版本与英文版本有任何冲突或不一致，则以英文版本为准。

© 2015 STMicroelectronics - 保留所有权利