RM0433
参考手册
STM32H7x3
基于 ARM 内核的 32 位高级 MCU

前言

本参考手册面向应用开发人员， 提供有关使用 STM32H7x3 微控制器存储器与外设的完整信息。

STM32H7x3 构成一个微控制器系列，各产品具有不同的存储器大小、封装和外设。

有关订购信息以及器件的机械与电气特性，请参见相应的数据手册。

有关带 FPU 的 ARM® Cortex®-M7 内核的信息，请参见相应的 ARM *技术参考手册*。

相关文档

- ARM® Cortex®-M7 技术参考手册，可从 www.arm.com 获取。
- Cortex®-M7 编程手册 (PM0253)。

目录

1 文档约定 94

1.1 寄存器相关缩写词列表 94

1.2 词汇表 94

1.3 外设可用性 94

2 存储器和总线架构 95

2.1 系统架构 95

2.1.1 总线矩阵 97

2.1.2 总线-总线桥 97

2.1.3 域间总线 97

2.1.4 CPU 总线 98

2.1.5 总线主设备外设 98

2.1.6 功能模块的时钟 99

2.2 存储器组织结构 100

2.2.1 前言 100

2.2.2 存储器映射和寄存器边界地址 100

2.3 内部 SRAM 104

2.4 Flash 概述 105

2.5 启动配置 105

3 嵌入式 Flash (FLASH) 107

3.1 前言 107

3.2 Flash 主要特性 107

3.3 Flash 功能说明 108

3.3.1 框图 108

3.3.2 引脚和内部信号 108

3.3.3 Flash 架构 109

3.3.4 Flash 读操作 110

3.3.5 误码校正 (ECC) 113

3.3.6 循环冗余校验模块 113

3.3.7 Flash 编程/擦除操作 114

3.3.8 更改用户选项字节 120

3.3.9 Flash 接口错误标志 121

3.3.10 同时在存储区 1 和存储区 2 上执行读取/编程/擦除操作 122
3.3.11 FLASH 选项字节 122
3.3.12 保护机制 125
3.3.13 Flash 存储区交换 129
3.4 Flash 中断 131
3.5 Flash 接口寄存器 131
3.5.1 Flash 访问控制寄存器 (FLASH_ACR) 131
3.5.2 存储区 1 的 FLASH 密钥寄存器 (FLASH_KEYR1) 132
3.5.3 Flash 选项密钥寄存器 (FLASH_OPTKEYR) 133
3.5.4 存储区 1 的 FLASH 控制寄存器 (FLASH_CR1) 133
3.5.5 存储区 1 的 FLASH 状态寄存器 (FLASH_SR1) 136
3.5.6 存储区 1 的 FLASH 清零控制寄存器 (FLASH_CCR1) 139
3.5.7 Flash 选项控制寄存器 (FLASH_OPTCR) 140
3.5.8 FLASH 选项状态寄存器（当前值）(FLASH_OPTSR_CUR) 141
3.5.9 FLASH 选项状态寄存器（要编程的值）(FLASH_OPTSR_PRG) 144
3.5.10 FLASH 选项清空控制寄存器 (FLASH_OPTCCR) 146
3.5.11 存储区 1 的 FLASH 保护地址（当前值）(FLASH_PRAR_CUR1) 147
3.5.12 存储区 1 的 FLASH 保护地址（要编程的值）(FLASH_PRAR_PRG1) 148
3.5.13 存储区 1 的 FLASH 安全地址（当前值）(FLASH_SCAR_CUR1) 149
3.5.14 存储区 1 的 FLASH 安全地址（要编程的值）(FLASH_SCAR_PRG1) 150
3.5.15 存储区 1 的 FLASH 扇区写保护（当前值）(FLASH_WPSN_CUR1R) 150
3.5.16 存储区 1 的 FLASH 扇区写保护（要编程的值）(FLASH_WPSN_PRG1R) 151
3.5.17 包含自举地址的 FLASH 寄存器（当前值）(FLASH_BOOT_CURR) 151
3.5.18 包含自举地址的 FLASH 寄存器（要编程的值）(FLASH_BOOT_PRGR) 152
3.5.19 存储区 1 的 FLASH CRC 控制寄存器 (FLASH_CRCCR1) 152
3.5.20 存储区 1 的 FLASH CRC 起始地址寄存器 (FLASH_CRCSADD1R) 154
3.5.21 存储区 1 的 FLASH CRC 结束地址寄存器 (FLASH_CRCEADD1R) 154
3.5.22 FLASH CRC 数据寄存器 (FLASH_CRCDATAR) 155
3.5.23 存储区 1 的 FLASH ECC 失效地址 FLASH_ECC_FA1R) 155
3.5.24 存储区 2 的 FLASH 密钥寄存器 (FLASH_KEYR2) 156
3.5.25 存储区 2 的 FLASH 控制寄存器 (FLASH_CR2) 156
3.5.26 存储区 2 的 FLASH 状态寄存器 (FLASH_SR2) 160
3.5.27 存储区 2 的 FLASH 清零控制寄存器 (FLASH_CCR2) 163
3.5.28 存储区 2 的 FLASH 保护地址（当前值）(FLASH_PRAR_CUR2) 164
3.5.29 存储区 2 的 FLASH 保护地址（要编程的值）(FLASH_PRAR_PRG2) 165

3.5.30 存储区 2 的 FLASH 安全地址（当前值）(FLASH_SCAR_CUR2) 166
3.5.31 存储区 2 的 FLASH 安全地址（要编程的值）(FLASH_SCAR_PRG2) 167
3.5.32 存储区 2 的 FLASH 扇区写保护（当前值）(FLASH_WPSN_CUR2R) 167
3.5.33 存储区 2 的 FLASH 扇区写保护（要编程的值）(FLASH_WPSN_PRG2R) 168
3.5.34 存储区 2 的 FLASH CRC 控制寄存器 (FLASH_CRCCR2) 168
3.5.35 存储区 2 的 FLASH CRC 起始地址寄存器 (FLASH_CRCSADD2R) 170
3.5.36 存储区 2 的 FLASH CRC 结束地址寄存器 (FLASH_CRCEADD2R) 170
3.5.37 存储区 2 的 FLASH ECC 失效地址 (FLASH_ECC_FA2R) 171
3.6 Flash 寄存器映射与复位值 172

4 安全存储管理 177
4.1 前言 177
4.2 词汇表 177
4.3 Flash 保护 178
4.4 安全访问模式 178
4.4.1 相关特性 179
4.4.2 自举状态机 179
4.4.3 安全访问模式配置 180
4.5 根安全服务 (RSS) 181
4.5.1 调用根安全服务 181
4.5.2 根安全服务说明 182
4.6 安全用户软件 182
4.6.1 访问规则 183
4.6.2 设置安全用户存储区 183
4.6.3 移除安全用户存储区 183
4.6.4 选择安全用户软件 184
4.7 Flash 保护机制汇总 185

5 AXI 互连 186
5.1 AXI简介 186
5.2 AXI 互连主要特性 186
5.3 AXI 互连功能说明 187
5.3.1 框图 187
5.3.2 ASIB配置 187
5.3.3 AMIB配置 188

5.3.4 服务质量 (QoS) 188
5.3.5 全局编程器视图 (GPV) 188
5.4 AXI 互连寄存器 189
5.4.1 AXI 互连 - 外设 ID4 寄存器 (AXI_PERIPH_ID_4) 189
5.4.2 AXI 互连 - 外设 ID0 寄存器 (AXI_PERIPH_ID_0) 189
5.4.3 AXI 互连 - 外设 ID1 寄存器 (AXI_PERIPH_ID_1) 190
5.4.4 AXI 互连 - 外设 ID2 寄存器 (AXI_PERIPH_ID_2) 190
5.4.5 AXI 互连 - 外设 ID3 寄存器 (AXI_PERIPH_ID_3) 191
5.4.6 AXI 互连 - 组件 ID0 寄存器 (AXI_PERIPH_ID_0) 191
5.4.7 AXI 互连 - 组件 ID1 寄存器 (AXI_PERIPH_ID_1) 192
5.4.8 AXI 互连 - 组件 ID2 寄存器 (AXI_PERIPH_ID_2) 192
5.4.9 AXI 互连 - 组件 ID3 寄存器 (AXI_PERIPH_ID_3) 193
5.4.10 AXI 互连 - TARG x 总线矩阵发布功能寄存器 (AXI_TARGx_FN_MOD_ISS_BM) 193
5.4.11 AXI 互连 - TARG x 总线矩阵功能 2 寄存器 (AXI_TARGx_FN_MOD2) 194
5.4.12 AXI 互连 - TARG x 长突发功能修改寄存器 (AXI_TARGx_FN_MOD_LB) . . . 194
5.4.13 AXI 互连 - TARG x 发布功能修改寄存器 (AXI_TARGx_FN_MOD) 195
5.4.14 AXI 互连 - INI x 功能修改 2 寄存器 (AXI_INIx_FN_MOD2) 195
5.4.15 AXI 互连 - INI x AHB 功能修改寄存器 (AXI_INIx_FN_MOD_AHB) 196
5.4.16 AXI 互连 - INI x 读取 QoS 寄存器 (AXI_INIx_READ_QOS) 196
5.4.17 AXI 互连 - INI x 写入 QoS 寄存器 (AXI_INIx_WRITE_QOS) 197
5.4.18 AXI 互连 - INI x 发布功能修改寄存器 (AXI_INIx_FN_MOD) 197
5.5 AXI 互连寄存器映射 198

6 电源控制 (PWR) 206
6.1 前言 206
6.2 PWR 主要特性 206
6.3 PWR 框图 207
6.3.1 PWR 引脚和内部信号 208
6.4 电源 209
6.4.1 系统电源启动 212
6.4.2 内核域 213
6.4.3 PWR 外部电源 214
6.4.4 备份域 215
6.4.5 VBAT 电池充电 216
6.4.6 模拟电源 217
6.4.7 USB 稳压器 217

6.5 电源监控 218
6.5.1 上电复位(POR)/掉电复位(PDR) 218
6.5.2 欠压复位 (BOR) 219
6.5.3 可编程电压检测器（PVD） 220
6.5.4 模拟电压检测器 (AVD) 221
6.5.5 电池电压阈值 222
6.5.6 温度阈值 223
6.6 电源管理 223
6.6.1 工作模式 224
6.6.2 电压调节 227
6.6.3 电源控制模式 228
6.6.4 电源管理示例 231
6.7 低功耗模式 237
6.7.1 降低系统时钟速度 237
6.7.2 控制外设时钟 237
6.7.3 进入低功耗模式 237
6.7.4 退出低功耗模式 238
6.7.5 CSleep 模式 238
6.7.6 CStop 模式 239
6.7.7 DStop 模式 240
6.7.8 停止模式 241
6.7.9 DStandby 模式 243
6.7.10 待机模式 245
6.8 PWR 寄存器说明 247
6.8.1 PWR 控制寄存器 1 (PWR_CR1) 247
6.8.2 PWR 控制状态寄存器 1 (PWR_CSR1) 248
6.8.3 PWR 控制寄存器 2 (PWR_CR2) 249
6.8.4 PWR 控制寄存器 3 (PWR_CR3) 250
6.8.5 PWR CPU 控制寄存器 (PWR_CPUCR) 252
6.8.6 PWR D3 域控制寄存器 (PWR_D3CR) 253
6.8.7 PWR 唤醒清除寄存器 (PWR_WKUPCR) 254
6.8.8 PWR 唤醒标志寄存器 (PWR_WKUPFR) 254
6.8.9 PWR 唤醒使能和极性寄存器 (PWR_WKUPEPR) 255
6.8.10 PWR 寄存器映射 256

7 低功耗 D3 域 257
7.1 前言 257
7.2 EXTI、RCC 和 PWR 互连 257
7.2.1 中断和唤醒 259
7.2.2 模块交互 259
7.2.3 D3 域 DMAMUX2 的角色 260
7.3 基于 LPUART1 发送的低功耗应用示例 260
7.3.1 存储器保留 260
7.3.2 使用 LPUART1 接口的存储器至外设传输 260
7.3.3 基于 LPUART1 发送的低功耗应用示例总体说明 265
7.3.4 备选实现方式 266
7.4 其它低功耗应用 266

8 复位和时钟控制 (RCC) 267
8.1 RCC 主要特性 267
8.2 RCC 框图 268
8.3 RCC 引脚和内部信号 268
8.4 RCC 复位模块功能说明 270
8.4.1 上电/掉电复位 270
8.4.2 系统复位 271
8.4.3 本地复位 272
8.4.4 复位源标识 273
8.4.5 低功耗模式安全复位 (lpwr_rst) 274
8.4.6 备份域复位 275
8.4.7 上电和唤醒序列 275
8.5 RCC 时钟模块功能说明 277
8.5.1 时钟命名约定 279
8.5.2 振荡器说明 279
8.5.3 时钟安全系统 (CSS) 283
8.5.4 时钟输出生成 (MCO1/MCO2) 284
8.5.5 PLL 描述 285
8.5.6 系统时钟 (sys_ck) 289
8.5.7 在停止和待机模式下处理时钟发生器 291
8.5.8 内核时钟选择 292
8.5.9 常规时钟概念概述 303
8.5.10 外设分配 306
8.5.11 外设时钟门控 308
8.5.12 CPU 和总线矩阵时钟门控 311

8.6 RCC 中断 313
8.7 RCC 寄存器说明 314
8.7.1 寄存器映射概述 314
8.7.2 RCC 源控制寄存器 (RCC_CR) 315
8.7.3 RCC 内部时钟源校准寄存器 (RCC_ICSCR) 318
8.7.4 RCC 时钟恢复 RC 寄存器 (RCC_CRRCR) 319
8.7.5 RCC 时钟配置寄存器 (RCC_CFGR) 320
8.7.6 RCC 域 1 时钟配置寄存器 (RCC_D1CFGR) 322
8.7.7 RCC 域 2 时钟配置寄存器 (RCC_D2CFGR) 324
8.7.8 RCC 域 3 时钟配置寄存器 (RCC_D3CFGR) 325
8.7.9 RCC PLL 时钟源选择寄存器 (RCC_PLLCKSELR) 326
8.7.10 RCC PLL 配置寄存器 (RCC_PLLCFGR) 328
8.7.11 RCC PLL1 分频器配置寄存器 (RCC_PLL1DIVR) 331
8.7.12 RCC PLL1 小数分频器寄存器 (RCC_PLL1FRACR) 333
8.7.13 RCC PLL2 分频器配置寄存器 (RCC_PLL2DIVR) 334
8.7.14 RCC PLL2 小数分频器寄存器 (RCC_PLL2FRACR) 336
8.7.15 RCC PLL3 分频器配置寄存器 (RCC_PLL3DIVR) 337
8.7.16 RCC PLL3 小数分频器寄存器 (RCC_PLL3FRACR) 339
8.7.17 RCC 域 1 内核时钟配置寄存器 (RCC_D1CCIPR) 340
8.7.18 RCC 域 2 内核时钟配置寄存器 (RCC_D2CCIP1R) 341
8.7.19 RCC 域 2 内核时钟配置寄存器 (RCC_D2CCIP2R) 343
8.7.20 RCC 域 3 内核时钟配置寄存器 (RCC_D3CCIPR) 345
8.7.21 RCC 时钟源中断使能寄存器 (RCC_CIER) 347
8.7.22 RCC 时钟源中断标志寄存器 (RCC_CIFR) 349
8.7.23 RCC 时钟源中断清零寄存器 (RCC_CICR) 351
8.7.24 RCC 备份域控制寄存器 (RCC_BDCR) 353
8.7.25 RCC 时钟控制和状态寄存器 (RCC_CSR) 355
8.7.26 RCC AHB3 复位寄存器 (RCC_AHB3RSTR) 356
8.7.27 RCC AHB1 外设复位寄存器 (RCC_AHB1RSTR) 357
8.7.28 RCC AHB2 外设复位寄存器 (RCC_AHB2RSTR) 358
8.7.29 RCC AHB4 外设复位寄存器 (RCC_AHB4RSTR) 359
8.7.30 RCC APB3 外设复位寄存器 (RCC_APB3RSTR) 361
8.7.31 RCC APB1 外设复位寄存器 (RCC_APB1LRSTR) 362
8.7.32 RCC APB1 外设复位寄存器 (RCC_APB1HRSTR) 365
8.7.33 RCC APB2 外设复位寄存器 (RCC_APB2RSTR) 366
8.7.34 RCC APB4 外设复位寄存器 (RCC_APB4RSTR) 368
8.7.35 RCC 全局控制寄存器 (RCC_GCR) 369

8.7.36 RCC D3 自主模式寄存器 (RCC_D3AMR) 370
8.7.37 RCC 复位状态寄存器 (RCC_RSR) 373
8.7.38 RCC AHB3 时钟寄存器 (RCC_AHB3ENR) 375
8.7.39 RCC AHB1 时钟寄存器 (RCC_AHB1ENR) 376
8.7.40 RCC AHB2 时钟寄存器 (RCC_AHB2ENR) 378
8.7.41 RCC AHB4 时钟寄存器 (RCC_AHB4ENR) 380
8.7.42 RCC APB3 时钟寄存器 (RCC_APB3ENR) 382
8.7.43 RCC APB1 时钟寄存器 (RCC_APB1LENR) 383
8.7.44 RCC APB1 时钟寄存器 (RCC_APB1HENR) 386
8.7.45 RCC APB2 时钟寄存器 (RCC_APB2ENR) 387
8.7.46 RCC APB4 时钟寄存器 (RCC_APB4ENR) 390
8.7.47 RCC AHB3 睡眠时钟寄存器 (RCC_AHB3LPENR) 392
8.7.48 RCC AHB1 睡眠时钟寄存器 (RCC_AHB1LPENR) 394
8.7.49 RCC AHB2 睡眠时钟寄存器 (RCC_AHB2LPENR) 396
8.7.50 RCC AHB4 睡眠时钟寄存器 (RCC_AHB4LPENR) 398
8.7.51 RCC APB3 睡眠时钟寄存器 (RCC_APB3LPENR) 400
8.7.52 RCC APB1 低位睡眠时钟寄存器 (RCC_APB1LLPENR) 401
8.7.53 RCC APB1 高位睡眠时钟寄存器 (RCC_APB1HLPENR) 405
8.7.54 RCC APB2 睡眠时钟寄存器 (RCC_APB2LPENR) 406
8.7.55 RCC APB4 睡眠时钟寄存器 (RCC_APB4LPENR) 409
8.8 RCC 寄存器映射 412

9 时钟恢复系统 (CRS) 421
9.1 前言 421
9.2 CRS主要特性 421
9.3 CRS 功能说明 422
9.3.1 CRS框图 422
9.4 CRS 内部信号 423
9.4.1 同步输入 423
9.4.2 频率误差测量 423
9.4.3 频率误差评估和自动微调 425
9.4.4 CRS 初始化和配置 425
9.5 CRS 低功耗模式 426
9.6 CRS 中断 426

9.7 CRS 寄存器 427
9.7.1 CRS 控制寄存器 (CRS_CR) 427
9.7.2 CRS 配置寄存器 (CRS_CFGR) 428
9.7.3 CRS 中断和状态寄存器 (CRS_ISR) 429
9.7.4 CRS 中断标志清零寄存器 (CRS_ICR) 431
9.7.5 CRS寄存器映射 432

10 硬件信号量 (HSEM) 433
10.1 硬件信号量简介 433
10.2 硬件信号量的主要特性 433
10.3 HSEM 功能说明 433
10.3.1 HSEM框图 433
10.3.2 HSEM 内部信号 434
10.3.3 HSEM 锁定步骤 434
10.3.4 HSEM 写/读/读锁定寄存器地址 436
10.3.5 HSEM 清零步骤 436
10.3.6 HSEM MasterID 信号量清零 436
10.3.7 HSEM 中断 437
10.3.8 AHB 总线主控 ID 验证 438
10.4 HSEM寄存器 439
10.4.1 HSEM 寄存器 (HSEM_R0 - HSEM_R31) 439
10.4.2 HSEM 读取锁定寄存器 (HSEM_RLR0 - HSEM_RLR31) 440
10.4.3 HSEM 中断使能寄存器 (HSEM_CnIER) 441
10.4.4 HSEM 中断清零寄存器 (HSEM_CnICR) 441
10.4.5 HSEM 中断状态寄存器 (HSEM_CnISR) 442
10.4.6 HSEM 屏蔽中断状态寄存器 (HSEM_CnMISR) 442
10.4.7 HSEM 清零寄存器 (HSEM_CR) 443
10.4.8 HSEM 中断清零寄存器 (HSEM_KEYR) 443
10.4.9 HSEM寄存器映射 444

11 通用 I/O (GPIO) 445
11.1 简介 445
11.2 GPIO 主要特性 445
11.3 GPIO 功能描述 445
11.3.1 通用 I/O (GPIO) 448
11.3.2 I/O 引脚复用功能复用器和映射 448

11.3.3 I/O 端口控制寄存器 449
11.3.4 I/O 端口数据寄存器 449
11.3.5 I/O 数据位操作 449
11.3.6 GPIO 锁定机制 450
11.3.7 I/O 复用功能输入/输出 450
11.3.8 外部中断线/唤醒线 450
11.3.9 输入配置 450
11.3.10 输出配置 451
11.3.11 I/O 补偿单元 452
11.3.12 复用功能配置 452
11.3.13 模拟配置 453
11.3.14 将 HSE 或 LSE 振荡器引脚用作 GPIO 454
11.3.15 在备份电源域中使用 GPIO 引脚 454
11.4 GPIO寄存器 454
11.4.1 GPIO 端口模式寄存器 (GPIOx_MODER) (x =A..K) 455
11.4.2 GPIO 端口输出类型寄存器 (GPIOx_OTYPER) (x = A..K) 455
11.4.3 GPIO 端口输出速度寄存器 (GPIOx_OSPEEDR) (x = A..K) 456
11.4.4 GPIO 端口上拉/下拉寄存器 (GPIOx_PUPDR) (x = A..K) 456
11.4.5 GPIO 端口输入数据寄存器 (GPIOx_IDR) (x = A..K) 457
11.4.6 GPIO 端口输出数据寄存器 (GPIOx_ODR) (x = A..K) 457
11.4.7 GPIO 端口置位/复位寄存器 (GPIOx_BSRR) (x = A..K) 458
11.4.8 GPIO 端口配置锁定寄存器 (GPIOx_LCKR) (x = A..K) 458
11.4.9 GPIO 复用功能低位寄存器 (GPIOx_AFRL) (x = A..K) 459
11.4.10 GPIO 复用功能高位寄存器 (GPIOx_AFRH) (x = A..J) 460
11.4.11 GPIO 寄存器映射 461

12 系统配置控制器 (SYSCFG) 463
12.1 前言 463
12.2 SYSCFG主要特性 463
12.3 SYSCFG 寄存器说明 463
12.3.1 SYSCFG 外设模式配置寄存器 (SYSCFG_PMCR) 463
12.3.2 SYSCFG 外部中断配置寄存器 1 (SYSCFG_EXTICR1) 465
12.3.3 SYSCFG 外部中断配置寄存器 2 (SYSCFG_EXTICR2) 466
12.3.4 SYSCFG 外部中断配置寄存器 3 (SYSCFG_EXTICR3) 467
12.3.5 SYSCFG 外部中断配置寄存器 4 (SYSCFG_EXTICR4) 468
12.3.6 SYSCFG 补偿单元控制/状态寄存器 (SYSCFG_CCCSR) 469
12.3.7 SYSCFG 补偿单元值寄存器 (SYSCFG_CCVR) 470

12.3.8 SYSCFG 补偿单元代码寄存器 (SYSCFG_CCCR) 470
12.3.9 SYSCFG 封装寄存器 (SYSCFG_PKGR) 471
12.3.10 SYSCFG 用户寄存器 0 (SYSCFG_UR0) 471
12.3.11 SYSCFG 用户寄存器 2 (SYSCFG_UR2) 472
12.3.12 SYSCFG 用户寄存器 3 (SYSCFG_UR3) 473
12.3.13 SYSCFG 用户寄存器 4 (SYSCFG_UR4) 473
12.3.14 SYSCFG 用户寄存器 5 (SYSCFG_UR5) 474
12.3.15 SYSCFG 用户寄存器 6 (SYSCFG_UR6) 474
12.3.16 SYSCFG 用户寄存器 7 (SYSCFG_UR7) 475
12.3.17 SYSCFG 用户寄存器 8 (SYSCFG_UR8) 475
12.3.18 SYSCFG 用户寄存器 9 (SYSCFG_UR9) 476
12.3.19 SYSCFG 用户寄存器 10 (SYSCFG_UR10) 477
12.3.20 SYSCFG 用户寄存器 11 (SYSCFG_UR11) 477
12.3.21 SYSCFG 用户寄存器 12 (SYSCFG_UR12) 478
12.3.22 SYSCFG 用户寄存器 13 (SYSCFG_UR13) 478
12.3.23 SYSCFG 用户寄存器 14 (SYSCFG_UR14) 479
12.3.24 SYSCFG 用户寄存器 15 (SYSCFG_UR15) 480
12.3.25 SYSCFG 用户寄存器 16 (SYSCFG_UR16) 480
12.3.26 SYSCFG 用户寄存器 17 (SYSCFG_UR17) 481
12.3.27 SYSCFG 寄存器映射 482

13 块互连 485
13.1 外设互连 485
13.1.1 前言 485
13.1.2 连接概述 485
13.2 从低功耗模式唤醒 504
13.3 DMA 508
13.3.1 MDMA（D1 域） 509
13.3.2 DMAMUX1、DMA1 和 DMA2（D2 域） 510
13.3.3 DMAMUX2 和 BDMA（D3 域） 515

14 MDMA 控制器 (MDMA) 518
14.1 MDMA 简介 518
14.2 MDMA 主要特性 518
14.3 MDMA 功能说明 519
14.3.1 MDMA 框图 519
14.3.2 MDMA 内部信号 519

14.3.3 MDMA 概述 520
14.3.4 MDMA 通道 521
14.3.5 源、目标和传输模式 521
14.3.6 指针更新 521
14.3.7 MDMA 缓冲区传输 522
14.3.8 请求仲裁 522
14.3.9 FIFO 523
14.3.10 块传输 523
14.3.11 块重复模式 523
14.3.12 链表模式 523
14.3.13 MDMA 传输完成 523
14.3.14 MDMA 传输暂停 524
14.3.15 错误管理 524
14.4 MDMA 中断 524
14.5 MDMA寄存器 525
14.5.1 MDMA 全局中断/状态寄存器 (MDMA_GISR0) 525
14.5.2 MDMA 通道 x 中断/状态寄存器 (MDMA_CxISR) (x = 0..15) 525
14.5.3 MDMA 通道 x 中断标志清零寄存器 (MDMA_CxIFCR) (x = 0..15) 527
14.5.4 MDMA 通道 x 错误状态寄存器 (MDMA_CxESR) (x = 0..15) 527
14.5.5 MDMA 通道 x 控制寄存器 (MDMA_CxCR) (x = 0..15) 529
14.5.6 MDMA 通道 x 传输配置寄存器 (MDMA_CxTCR) (x = 0..15) 531
14.5.7 MDMA 通道 x 块数据数寄存器 (MDMA_CxBNDTR) (x = 0..15) 534
14.5.8 MDMA 通道 x 源地址寄存器 (MDMA_CxSAR) (x = 0..15) 535
14.5.9 MDMA 通道 x 目标地址寄存器 (MDMA_CxDAR) (x = 0..15) 536
14.5.10 MDMA 通道 x 块重复地址更新寄存器 MDMA_CxBRUR (x = 0..15) 537
14.5.11 MDMA 通道 x 链路地址寄存器 (MDMA_CxLAR) (x = 0..15) 538
14.5.12 MDMA 通道 x 触发和总线选择寄存器 (MDMA_CxTBR) (x = 0..15) . . . 539
14.5.13 MDMA 通道 x 掩码地址寄存器 (MDMA_CxMAR) (x = 0..15) 540
14.5.14 MDMA 通道 x 掩码数据寄存器 (MDMA_CxMDR) (x = 0..15) 540
14.5.15 MDMA寄存器映射 541

15 直接存储器访问控制器（DMA1、DMA2） 543
15.1 DMA 简介 543
15.2 DMA 主要特性 543

15.3 DMA 功能说明 544
15.3.1 DMA 框图 544
15.3.2 DMA 内部信号 544
15.3.3 DMA概述 545
15.3.4 DMA 传输 545
15.3.5 DMA 请求映射 545
15.3.6 仲裁器 546
15.3.7 DMA 数据流 546
15.3.8 源、目标和传输模式 546
15.3.9 指针递增 549
15.3.10 循环模式 550
15.3.11 双缓冲区模式 550
15.3.12 可编程数据宽度、封装/解封、字节序 551
15.3.13 单次传输和突发传输 553
15.3.14 FIFO 554
15.3.15 DMA 传输完成 556
15.3.16 DMA 传输暂停 557
15.3.17 流控制器 558
15.3.18 可能的 DMA 配置汇总 559
15.3.19 流配置过程 559
15.3.20 错误管理 560
15.4 DMA 中断 561
15.5 DMA 寄存器 562
15.5.1 DMA 低中断状态寄存器 (DMA_LISR) 562
15.5.2 DMA 高中断状态寄存器 (DMA_HISR) 563
15.5.3 DMA 低中断标志清零寄存器 (DMA_LIFCR) 564
15.5.4 DMA 高中断标志清零寄存器 (DMA_HIFCR) 564
15.5.5 DMA 数据流 x 配置寄存器 (DMA_SxCR) (x = 0..7) 565
15.5.6 DMA 数据流 x 数据项数寄存器 (DMA_SxNDTR) (x = 0..7) 568
15.5.7 DMA 数据流 x 外设地址寄存器 (DMA_SxPAR) (x = 0..7) 569
15.5.8 DMA 数据流 x 存储器 0 地址寄存器 (DMA_SxM0AR) (x = 0..7) 569
15.5.9 DMA 数据流 x 存储器 1 地址寄存器 (DMA_SxM1AR) (x = 0..7) 570
15.5.10 DMA 数据流 x FIFO 控制寄存器 (DMA_SxFCR) (x = 0..7) 570
15.5.11 DMA 寄存器映射 572

16 基本直接存储器访问控制器 (BDMA) 575
16.1 前言 575
16.2 BDMA主要特性 575
16.3 BDMA 功能说明 576
16.3.1 BDMA 传输 576
16.3.2 仲裁器 577
16.3.3 BDMA通道 577
16.3.4 可编程数据宽度、数据对齐和字节存储次序 579
16.3.5 错误管理 580
16.3.6 BDMA 中断 580
16.4 BDMA 寄存器 581
16.4.1 DMA 中断状态寄存器 (BDMA_ISR) 581
16.4.2 DMA 中断标志清零寄存器 (BDMA_IFCR) 582
16.4.3 DMA 通道 x 配置寄存器 (BDMA_CCRx) （x = 1..8，其中 x 表示通道编号） 583
16.4.4 DMA 通道 x 数据数寄存器 (BDMA_CNDTRx) （x = 1..8，其中 x 表示通道编号） 585
16.4.5 DMA 通道 x 外设地址寄存器 (BDMA_CPARx) （x = 1..8，其中 x 表示通道编号） 585
16.4.6 DMA 通道 x 存储器地址寄存器 (BDMA_CMARx) （x = 1..8，其中 x 表示通道编号） 586
16.4.7 BDMA寄存器映射 587

17 DMA 请求复用器 (DMAMUX) 590
17.1 简介 590
17.2 DMAMUX 主要特性 590
17.3 DMAMUX 实现 591
17.3.1 DMAMUX1 和 DMAMUX2 实例化 591
17.3.2 DMAMUX1 映射 591
17.3.3 DMAMUX2 映射 593
17.4 DMAMUX 功能说明 595
17.4.1 DMAMUX 框图 595
17.4.2 DMAMUX 信号 596
17.4.3 DMAMUX 通道 596
17.4.4 DMAMUX 请求线复用器 596
17.4.5 DMAMUX 请求发生器 598
17.5 DMAMUX 中断 599

17.6 DMAMUX1 寄存器 600
17.6.1 DMAMUX1 请求线复用器通道 x 配置寄存器 (DMAMUX1_CxCR) 600
17.6.2 DMAMUX2 请求线复用器通道 x 配置寄存器 (DMAMUX2_CxCR) 601
17.6.3 DMAMUX1 请求线复用器中断通道状态寄存器 (DMAMUX1_CSR) 602
17.6.4 DMAMUX2 请求线复用器中断通道状态寄存器 (DMAMUX2_CSR) 602
17.6.5 DMAMUX1 请求线复用器中断清除标志寄存器 (DMAMUX1_CFR) 603
17.6.6 DMAMUX2 请求线复用器中断清除标志寄存器 (DMAMUX2_CFR) 603
17.6.7 DMAMUX1 请求发生器通道 x 配置寄存器 (DMAMUX1_RGxCR) 604
17.6.8 DMAMUX2 请求发生器通道 x 配置寄存器 (DMAMUX2_RGxCR) 604
17.6.9 DMAMUX1 请求发生器中断状态寄存器 (DMAMUX1_RGSR) 606
17.6.10 DMAMUX2 请求发生器中断状态寄存器 (DMAMUX2_RGSR) 606
17.6.11 DMAMUX1 请求发生器中断清除标志寄存器 (DMAMUX1_RGCFR) 607
17.6.12 DMAMUX2 请求发生器中断清除标志寄存器 (DMAMUX2_RGCFR) 607
17.6.13 DMAMUX 寄存器映射 608

18 Chrom-Art Accelerator™ 控制器 (DMA2D) 610

18.1 DMA2D 简介 610
18.2 DMA2D 的主要特性 610
18.3 DMA2D 功能说明 611
18.3.1 概述 611
18.4 DMA2D 引脚和内部信号 612
18.4.1 DMA2D 控制 612
18.4.2 DMA2D 前景层 FIFO 和背景层 FIFO 612
18.4.3 DMA2D 前景层和背景层像素格式转换器 (PFC) 613
18.4.4 DMA2D 前景层 FIFO 和背景层 CLUT 接口 615
18.4.5 DMA2D 混合器 616
18.4.6 DMA2D 输出 PFC 616
18.4.7 DMA2D 输出 FIFO 617
18.4.8 DMA2D AXI 主设备端口定时器 617
18.4.9 DMA2D 事务 617
18.4.10 DMA2D 配置 618
18.4.11 支持 YCbCr 620
18.4.12 DMA2D 传输控制（启动、挂起、中止和完成） 621
18.4.13 水印 621
18.4.14 错误管理 621
18.4.15 AXI 死区 621

18.5 DMA2D 中断 622
18.6 DMA2D 寄存器 623
18.6.1 DMA2D 控制寄存器 (DMA2D_CR) 623
18.6.2 DMA2D 中断状态寄存器 (DMA2D_ISR) 625
18.6.3 DMA2D 中断标志清零寄存器 (DMA2D_IFCR) 626
18.6.4 DMA2D 前景层存储器地址寄存器 (DMA2D_FGMAR) 627
18.6.5 DMA2D 前景层偏移寄存器 (DMA2D_FGOR) 627
18.6.6 DMA2D 背景层存储器地址寄存器 (DMA2D_BGMAR) 628
18.6.7 DMA2D 背景层偏移寄存器 (DMA2D_BGOR) 628
18.6.8 DMA2D 前景层 PFC 控制寄存器 (DMA2D_FGPFCCR) 629
18.6.9 DMA2D 前景层颜色寄存器 (DMA2D_FGCOLR) 631
18.6.10 DMA2D 背景层 PFC 控制寄存器 (DMA2D_BGPFCCR) 632
18.6.11 DMA2D 背景层颜色寄存器 (DMA2D_BGCOLR) 634
18.6.12 DMA2D 前景层 CLUT 存储器地址寄存器 (DMA2D_FGCMAR) 634
18.6.13 DMA2D 背景层 CLUT 存储器地址寄存器 (DMA2D_BGCMAR) 635
18.6.14 DMA2D 输出 PFC 控制寄存器 (DMA2D_OPFCCR) 635
18.6.15 DMA2D 输出颜色寄存器 (DMA2D_OCOLR) 636
18.6.16 DMA2D 输出存储器地址寄存器 (DMA2D_OMAR) 637
18.6.17 DMA2D 输出偏移寄存器 (DMA2D_OOR) 638
18.6.18 DMA2D 行数寄存器 (DMA2D_NLR) 638
18.6.19 DMA2D 行水印寄存器 (DMA2D_LWR) 639
18.6.20 DMA2D AXI 主设备定时器配置寄存器 (DMA2D_AMTCR) 639
18.6.21 DMA2D 寄存器映射 640

19 嵌套向量中断控制器 642
19.1 NVIC 特性 642
19.1.1 SysTick 校准值寄存器 642
19.1.2 中断和异常向量 642

20 扩展中断和事件控制器 (EXTI) 650
20.1 EXTI 主要特性 650
20.2 EXTI 框图 650
20.2.1 外设、CPU 和 D3 域之间的 EXTI 连接 651
20.3 EXTI 功能说明 652
20.3.1 EXTI 可配置事件输入 CPU 唤醒 652
20.3.2 EXTI 可配置事件输入任意唤醒 654
20.3.3 EXTI 直接事件输入 CPU 唤醒 655

20.3.4 EXTI 直接事件输入任意唤醒 655
20.3.5 EXTI D3 挂起请求清除选择 656
20.4 EXTI 事件输入映射 657
20.5 EXTI 功能行为 660
20.5.1 EXTI CPU 中断程序 661
20.5.2 EXTI CPU 事件程序 661
20.5.3 EXTI CPU 唤醒程序 661
20.5.4 自主运行模式下的 EXTI D3 域唤醒程序 661
20.5.5 EXTI 软件中断/事件触发程序 662
20.6 EXTI 寄存器说明 662
20.6.1 EXTI 上升沿触发选择寄存器 (EXTI_RTSR1) 662
20.6.2 EXTI 下降沿触发选择寄存器 (EXTI_FTSR1) 663
20.6.3 EXTI 软件中断事件寄存器 (EXTI_SWIER1) 663
20.6.4 EXTI D3 挂起屏蔽寄存器 (EXTI_D3PMR1) 664
20.6.5 EXTI D3 挂起清除选择寄存器低位字 (EXTI_D3PCR1L) 665
20.6.6 EXTI D3 挂起清除选择寄存器高位字 (EXTI_D3PCR1H) 665
20.6.7 EXTI 上升沿触发选择寄存器 (EXTI_RTSR2) 666
20.6.8 EXTI 下降沿触发选择寄存器 (EXTI_FTSR2) 667
20.6.9 EXTI 软件中断事件寄存器 (EXTI_SWIER2) 668
20.6.10 EXTI D3 挂起屏蔽寄存器 (EXTI_D3PMR2) 668
20.6.11 EXTI D3 挂起清除选择寄存器低位字 (EXTI_D3PCR2L) 669
20.6.12 EXTI D3 挂起清除选择寄存器高位字 (EXTI_D3PCR2H) 670
20.6.13 EXTI 上升沿触发选择寄存器 (EXTI_RTSR3) 670
20.6.14 EXTI 下降沿触发选择寄存器 (EXTI_FTSR3) 671
20.6.15 EXTI 软件中断事件寄存器 (EXTI_SWIER3) 672
20.6.16 EXTI D3 挂起屏蔽寄存器 (EXTI_D3PMR3) 672
20.6.17 EXTI D3 挂起清除选择寄存器低位字 (EXTI_D3PCR3L) 673
20.6.18 EXTI D3 挂起清除选择寄存器高位字 (EXTI_D3PCR3H) 673
20.6.19 EXTI 中断屏蔽寄存器 (EXTI_CPUIMR1) 674
20.6.20 EXTI 事件屏蔽寄存器 (EXTI_CPUEMR1) 674
20.6.21 EXTI 挂起寄存器 (EXTI_CPUPR1) 675
20.6.22 EXTI 中断屏蔽寄存器 (EXTI_CPUIMR2) 675
20.6.23 EXTI 事件屏蔽寄存器 (EXTI_CPUEMR2) 676
20.6.24 EXTI 挂起寄存器 (EXTI_CPUPR2) 677
20.6.25 EXTI 中断屏蔽寄存器 (EXTI_CPUIMR3) 677
20.6.26 EXTI 事件屏蔽寄存器 (EXTI_CPUEMR3) 678
20.6.27 EXTI 挂起寄存器 (EXTI_CPUPR3) 679
20.6.28 EXTI 寄存器映射 680

21 循环冗余校验计算单元 (CRC) 683
21.1 简介 683
21.2 CRC 主要特性 683
21.3 CRC 功能说明 684
21.3.1 CRC 框图 684
21.3.2 CRC 内部信号 684
21.3.3 CRC 操作 684
21.4 CRC 寄存器 686
21.4.1 数据寄存器 (CRC_DR) 686
21.4.2 独立数据寄存器 (CRC_IDR) 686
21.4.3 控制寄存器 (CRC_CR) 687
21.4.4 CRC 初始值 (CRC_INIT) 688
21.4.5 CRC 多项式 (CRC_POL) 688
21.4.6 CRC 寄存器映射 689

22 灵活存储控制器（FMC） 690
22.1 FMC主要特性 690
22.2 FMC 框图 691
22.3 FMC 内部信号 692
22.4 AHB 接口 693
22.5 AXI接口 693
22.5.1 支持的存储器和事务 693
22.6 外部器件地址映射 694
22.6.1 NOR/PSRAM 地址映射 696
22.6.2 NAND Flash 地址映射 696
22.6.3 SDRAM 地址映射 697
22.7 NOR Flash/PSRAM 控制器 700
22.7.1 外部存储器接口信号 702
22.7.2 支持的存储器和事务 704
22.7.3 通用时序规则 705
22.7.4 NOR Flash/PSRAM 控制器异步事务 705
22.7.5 同步事务 724
22.7.6 NOR/PSRAM 控制寄存器 730
22.8 NAND Flash 控制器 738
22.8.1 外部存储器接口信号 738
22.8.2 NAND Flash 支持的存储器和事务 739

22.8.3 NAND Flash 的时序图 740
22.8.4 NAND Flash 操作 741
22.8.5 NAND Flash 预等待功能 742
22.8.6 纠错码 (ECC) 计算 (NAND Flash) 743
22.8.7 NAND Flash 控制器寄存器 744
22.9 SDRAM 控制器 749
22.9.1 SDRAM 控制器主要特性 749
22.9.2 SDRAM 外部存储器接口信号 749
22.9.3 SDRAM 控制器功能说明 750
22.9.4 低功耗模式 756
22.9.5 SDRAM 控制寄存器 759
22.10 FMC 寄存器映射 766

23 QuadSPI接口（QUADSPI） 768
23.1 简介 768
23.2 QUADSPI 主要特性 768
23.3 QUADSPI 功能说明 768
23.3.1 QUADSPI 框图 768
23.3.2 QUADSPI 引脚和内部信号 769
23.3.3 QUADSPI 命令序列 770
23.3.4 QUADSPI 信号接口协议模式 772
23.3.5 QUADSPI 间接模式 775
23.3.6 QUADSPI 状态标志轮询模式 776
23.3.7 QUADSPI 内存映射模式 776
23.3.8 QUADSPI 自由运行时钟模式 777
23.3.9 QUADSPI FLASH 配置 777
23.3.10 QUADSPI 延迟数据采样 777
23.3.11 QUADSPI 配置 777
23.3.12 QUADSPI 的用法 778
23.3.13 指令仅发送一次 780
23.3.14 QUADSPI 差错管理 780
23.3.15 QUADSPI 的繁忙位和中止功能 780
23.3.16 nCS 行为 780
23.4 QUADSPI 中断 782

23.5 QUADSPI 寄存器 783
23.5.1 QUADSPI 控制寄存器 (QUADSPI_CR) 783
23.5.2 QUADSPI 器件配置寄存器 (QUADSPI_DCR) 786
23.5.3 QUADSPI 状态寄存器 (QUADSPI_SR) 787
23.5.4 QUADSPI 标志清零寄存器 (QUADSPI_FCR) 788
23.5.5 QUADSPI 数据长度寄存器 (QUADSPI_DLR) 788
23.5.6 QUADSPI 通信配置寄存器 (QUADSPI_CCR) 789
23.5.7 QUADSPI 地址寄存器 (QUADSPI_AR) 791
23.5.8 QUADSPI 交替字节寄存器 (QUADSPI_ABR) 792
23.5.9 QUADSPI 数据寄存器 (QUADSPI_DR) 792
23.5.10 QUADSPI 轮询状态屏蔽寄存器 (QUADSPI _PSMKR) 793
23.5.11 QUADSPI 轮询状态匹配寄存器 (QUADSPI _PSMAR) 793
23.5.12 QUADSPI 轮询间隔寄存器 (QUADSPI _PIR) 794
23.5.13 QUADSPI 低功耗超时寄存器 (QUADSPI_LPTR) 794
23.5.14 QUADSPI 寄存器映射 795

24 延迟模块 (DLYB) 796
24.1 简介 796
24.2 DLYB主要特性 796
24.3 DLYB 功能说明 796
24.3.1 DLYB 框图 796
24.3.2 DLYB 引脚和内部信号 797
24.3.3 概述 797
24.3.4 延迟线长度配置程序 797
24.3.5 输出时钟相位配置程序 798
24.4 DLYB寄存器 798
24.4.1 DLYB 控制寄存器 (DLYB_CR) 798
24.4.2 DLYB 配置寄存器 (DLYB_CFGR) 799
24.4.3 DLYB寄存器映射 800

25 模数转换器 (ADC) 801
25.1 简介 801
25.2 ADC 主要特性 801
25.3 ADC 功能说明 803
25.3.1 ADC框图 803
25.3.2 ADC 引脚和内部信号 804

25.3.3 时钟 805
25.3.4 ADC1/2/3 连接 806
25.3.5 从设备 AHB 接口 810
25.3.6 ADC 深度掉电模式 (DEEPPWD) 和 ADC 稳压器 (ADVREGEN) 810
25.3.7 单端通道和差分输入通道 810
25.3.8 校准（ADCAL、ADCALDIF、ADCALLIN、ADCx_CALFACT） 811
25.3.9 ADC 开关控制（ADEN、ADDIS、ADRDY） 816
25.3.10 写入 ADC 控制位时的限制 817
25.3.11 通道选择（SQRx、JSQRx） 817
25.3.12 通道预选寄存器 (ADCx_PCSEL) 818
25.3.13 可独立设置各通道采样时间（SMPR1、SMPR2） 818
25.3.14 单次转换模式 (CONT=0) 819
25.3.15 连续转换模式 (CONT=1) 820
25.3.16 开始转换（ADSTART、JADSTART） 820
25.3.17 时序 821
25.3.18 停止正在进行的转换（ADSTP、JADSTP） 822
25.3.19 外部触发转换和触发极性（EXTSEL、EXTEN、JEXTSEL、JEXTEN） 823
25.3.20 注入通道管理 827
25.3.21 不连续模式（DISCEN、DISCNUM、JDISCEN） 828
25.3.22 注入转换的上下文队列 829
25.3.23 可编程分辨率 (RES) - 快速转换模式 836
25.3.24 转换结束、采样阶段结束（EOC、JEOC、EOSMP） 837
25.3.25 转换序列结束（EOS、JEOS） 837
25.3.26 时序图示例（单次模式/连续模式，硬件/软件触发） 838
25.3.27 数据管理 839
25.3.28 使用 DFSDM 管理转换 846
25.3.29 动态低功耗特性 846
25.3.30 模拟窗口看门狗（AWD1EN、JAWD1EN、AWD1SGL、AWD1CH、AWD2CH、AWD3CH、AWD_HTRy、AWD_LTRy、AWDy） 851
25.3.31 过采样器 854
25.3.32 双重 ADC 模式 859
25.3.33 温度传感器 873
25.3.34 VBAT 电源监测 874
25.3.35 监测内部参考电压 875
25.4 ADC 中断 876

25.5 ADC 寄存器（每个 ADC） 877
25.5.1 ADC x 中断和状态寄存器 (ADCx_ISR)（x=1 到 3） 877
25.5.2 ADC x 中断使能寄存器 (ADCx_IER)（x=1 到 3） 879
25.5.3 ADC x 控制寄存器 (ADCx_CR)（x=1 到 3） 881
25.5.4 ADC x 配置寄存器 (ADCx_CFGR)（x=1 到 3） 885
25.5.5 ADC x 配置寄存器 2 (ADCx_CFGR2)（x=1 到 3） 889
25.5.6 ADC x 采样时间寄存器 1 (ADCx_SMPR1)（x=1 到 3） 891
25.5.7 ADC x 采样时间寄存器 2 (ADCx_SMPR2)（x=1 到 3） 892
25.5.8 ADC x 通道预选寄存器 (ADCx_PCSEL)（x=1 到 3） 893
25.5.9 ADC x 看门狗阈值寄存器 1 (ADCx_LTR1)（x=1 到 3） 893
25.5.10 ADC x 看门狗阈值寄存器 1 (ADCx_HTR1)（x=1 到 3） 894
25.5.11 ADC x 常规序列寄存器 1 (ADCx_SQR1)（x=1 到 3） 895
25.5.12 ADC x 常规序列寄存器 2 (ADCx_SQR2)（x=1 到 3） 896
25.5.13 ADC x 常规序列寄存器 3 (ADCx_SQR3)（x=1 到 3） 897
25.5.14 ADC x 常规序列寄存器 4 (ADCx_SQR4)（x=1 到 3） 898
25.5.15 ADC x 常规数据寄存器 (ADCx_DR)（x=1 到 3） 898
25.5.16 ADC x 注入序列寄存器 (ADCx_JSQR)（x=1 到 3） 899
25.5.17 ADC x 偏移寄存器 (ADCx_OFRy)（x=1 到 3） 901
25.5.18 ADC x 注入数据寄存器 (ADCx_JDRy)（x=1 到 3） 902
25.5.19 ADC x 模拟看门狗 2 配置寄存器 (ADCx_AWD2CR)（x=1 到 3） . . . 902
25.5.20 ADC x 模拟看门狗 3 配置寄存器 (ADCx_AWD3CR)（x=1 到 3） . . . 903
25.5.21 ADC x 看门狗阈值下限寄存器 2 (ADCx_LTR2)（x=1 到 3） 903
25.5.22 ADC x 看门狗阈值上限寄存器 2 (ADCx_HTR2)（x=1 到 3） 904
25.5.23 ADC x 看门狗阈值下限寄存器 3 (ADCx_LTR3)（x=1 到 3） 904
25.5.24 ADC x 看门狗阈值上限寄存器 3 (ADCx_HTR3)（x=1 到 3） 905
25.5.25 ADC x 差分模式选择寄存器 (ADCx_DIFSEL)（x=1 到 3） 905
25.5.26 ADC x 校准系数寄存器 (ADCx_CALFACT)（x=1 到 3） 906
25.5.27 ADC x 校准系数寄存器 2 (ADCx_CALFACT2)（x=1 到 3） 907
25.6 ADC 寄存器 908
25.6.1 ADC x 通用状态寄存器 (ADCx_CSR)（x=12 或 3） 908
25.6.2 ADC x 通用控制寄存器 (ADCx_CCR)（x=12 或 3） 910
25.6.3 适用于双重模式的 ADC x 通用常规数据寄存器 (ADCx_CDR)（x=12 或 3） 913
25.6.4 适用于 32 位双重模式的 ADC x 通用常规数据寄存器 (ADCx_CDR2)（x=12 或 3） 913
25.6.5 ADC 寄存器映射 914

26 数模转换器 (DAC) 918

26.1 简介 918

26.2 DAC 主要特性 918

26.3 DAC 功能说明 919

26.3.1 DAC框图 919

26.3.2 DAC 引脚和内部信号 920

26.3.3 DAC 通道使能 921

26.3.4 DAC 数据格式 921

26.3.5 DAC 转换 922

26.3.6 DAC 输出电压 922

26.3.7 DAC 触发选择 923

26.3.8 DMA 请求 924

26.3.9 生成噪声 924

26.3.10 生成三角波 925

26.3.11 DAC 通道模式 926

26.3.12 DAC 通道缓冲器校准 929

26.3.13 DAC 双通道转换（如果可用） 930

26.4 DAC 低功耗模式 933

26.5 DAC 中断 934

26.6 DAC 寄存器 934

26.6.1 DAC x 控制寄存器 (DACx_CR)（x=1 到 2） 934

26.6.2 DAC x 软件触发寄存器 (DACx_SWTRGR)（x=1 到 2） 937

26.6.3 DAC x 通道 1 12 位右对齐数据保持寄存器 (DACx_DHR12R1)（x = 1 到 2） 937

26.6.4 DAC x 通道 1 12 位左对齐数据保持寄存器 (DACx_DHR12L1)（x=1 到 2） 938

26.6.5 DAC x 通道 1 8 位右对齐数据保持寄存器 (DACx_DHR8R1)（x=1 到 2） 938

26.6.6 DAC x 通道 2 12 位右对齐数据保持寄存器 (DACx_DHR12R2)（x=1 到 2） 939

26.6.7 DAC x 通道 2 12 位左对齐数据保持寄存器 (DACx_DHR12L2)（x=1 到 2） 939

26.6.8 DAC x 通道 2 8 位右对齐数据保持寄存器 (DACx_DHR8R2)（x=1 到 2） 940

26.6.9 双 DAC x 12 位右对齐数据保持寄存器 (DACx_DHR12RD)（x=1 到 2） 940

26.6.10 双 DAC x 12 位左对齐数据保持寄存器 (DACx_DHR12LD)（x=1 到 2） 941

26.6.11 双 DAC x 8 位右对齐数据保持寄存器 (DACx_DHR8RD)（x=1 到 2） 941

26.6.12 DAC x 通道 1 数据输出寄存器 (DACx_DOR1)（x=1 到 2） 942
26.6.13 DAC x 通道 2 数据输出寄存器 (DACx_DOR2)（x=1 到 2） 942
26.6.14 DAC x 状态寄存器 (DACx_SR)（x=1 到 2） 943
26.6.15 DAC x 校准控制寄存器 (DACx_CCR)（x=1 到 2） 944
26.6.16 DAC x 模式控制寄存器 (DACx_MCR)（x=1 到 2） 944
26.6.17 DACx 采样和保持采样时间寄存器 1 (DACx_SHSR1)（x=1 到 2） 946
26.6.18 DACx 采样和保持采样时间寄存器 2 (DACx_SHSR2)（x=1 到 2） 946
26.6.19 DAC x 采样和保持保持时间寄存器 (DACx_SHHR)（x=1 到 2） 947
26.6.20 DAC x 采样和保持刷新时间寄存器 (DACx_SHRR)（x=1 到 2） 947
26.6.21 DAC 寄存器映射 949

27 电压参考缓冲器 (VREFBUF) 951

27.1 简介 951
27.2 VREFBUF 功能说明 951
27.3 VREFBUF 寄存器 952
27.3.1 VREFBUF 控制和状态寄存器 (VREFBUF_CSR) 952
27.3.2 VREFBUF 校准控制寄存器 (VREFBUF_CCR) 953
27.3.3 VREFBUF 寄存器映射 953

28 比较器 (COMP) 954

28.1 简介 954
28.2 COMP 主要特性 954
28.3 COMP 功能说明 955
28.3.1 COMP 框图 955
28.3.2 COMP 引脚和内部信号 955
28.3.3 COMP 复位和时钟 957
28.3.4 比较器锁定机制 957
28.3.5 窗口比较器 957
28.3.6 迟滞 957
28.3.7 比较器输出消隐功能 958
28.3.8 GPIO 上的比较器输出 959
28.3.9 比较器输出重定向 960
28.3.10 COMP 功耗和速度模式 960
28.4 COMP 低功耗模式 960
28.5 COMP 中断 961
28.5.1 通过 EXTI 模块实现的中断 961
28.5.2 通过 CPU 的 NVIC 实现中断 961

28.6 SCALER 功能 962
28.7 COMP 寄存器 963
28.7.1 比较器状态寄存器 (COMP_SR) 963
28.7.2 比较器中断清除标志寄存器 (COMP_ICFR) 964
28.7.3 比较器选项寄存器 (COMP_OR) 964
28.7.4 比较器配置寄存器 1 (COMP_CFGR1) 965
28.7.5 比较器配置寄存器 2 (COMP_CFGR2) 967
28.7.6 COMP寄存器映射 970

29 运算放大器（OPAMP） 971
29.1 简介 971
29.2 OPAMP 主要特性 971
29.3 OPAMP 功能说明 971
29.3.1 OPAMP 复位和时钟 971
29.3.2 初始配置 972
29.3.3 信号走线 972
29.3.4 OPAMP 模式 973
29.3.5 校准 979
29.4 OPAMP 低功耗模式 980
29.5 OPAMP PGA 增益 980
29.6 OPAMP 寄存器 981
29.6.1 OPAMP1 控制/状态寄存器 (OPAMP1_CSR) 981
29.6.2 正常模式下的 OPAMP1 微调寄存器 (OPAMP1_OTR) 983
29.6.3 高速模式下的 OPAMP1 微调寄存器 (OPAMP1_HSOTR) 984
29.6.4 OPAMP选项寄存器(OPAMP_OR) 984
29.6.5 OPAMP2 控制/状态寄存器 (OPAMP2_CSR) 985
29.6.6 正常模式下的 OPAMP2 微调寄存器 (OPAMP2_OTR) 987
29.6.7 高速模式下的 OPAMP2 微调寄存器 (OPAMP2_HSOTR) 987
29.6.8 OPAMP寄存器映射 988

30 数字滤波器，用于 ΣΔ 调制器（DFSDM） 989
30.1 简介 989
30.2 DFSDM 主要特性 990
30.3 DFSDM 实现 991
30.4 DFSDM 功能说明 992
30.4.1 DFSDM框图 992
30.4.2 DFSDM 引脚和内部信号 993

30.4.3 DFSDM 复位和时钟 994
30.4.4 串行通道收发器 995
30.4.5 配置输入串行接口 1004
30.4.6 并行数据输入 1004
30.4.7 通道选择 1006
30.4.8 数字滤波器配置 1007
30.4.9 积分器单元 1008
30.4.10 模拟看门狗 1008
30.4.11 短路检测器 1010
30.4.12 极值检测器 1011
30.4.13 数据单元模块 1011
30.4.14 有符号数据格式 1012
30.4.15 启动转换 1012
30.4.16 连续和快速连续模式 1013
30.4.17 请求优先级 1014
30.4.18 在运行模式下的功耗优化 1014
30.5 DFSDM 中断 1014
30.6 DFSDM DMA 传输 1016
30.7 DFSDM 通道 y 寄存器 (y=0..7) 1017
30.7.1 DFSDM 通道配置 y 寄存器 (DFSDM_CHyCFGR1) (y=0..7) 1017
30.7.2 DFSDM 通道配置 y 寄存器 (DFSDM_CHyCFGR2) (y=0..7) 1019
30.7.3 DFSDM 通道模拟看门狗和短路检测器寄存器 (DFSDM_CHyAWSCDR) (y = 0..7) 1020
30.7.4 DFSDM 通道看门狗滤波器数据寄存器 (DFSDM_CHyWDATR) (y=0..7) 1021
30.7.5 DFSDM 通道数据输入寄存器 (DFSDM_CHyDATINR) (y=0..7) 1022
30.8 DFSDM 滤波器 x 模块寄存器 (x=0..3) 1023
30.8.1 DFSDM 控制寄存器 1 (DFSDM_FLTxCR1) 1023
30.8.2 DFSDM 控制寄存器 2 (DFSDM_FLTxCR2) 1025
30.8.3 DFSDM 中断和状态寄存器 (DFSDM_FLTxISR) 1027
30.8.4 DFSDM 中断标志清零寄存器 (DFSDM_FLTxICR) 1029
30.8.5 DFSDM 注入通道组选择寄存器 (DFSDM_FLTxJCHGR) 1030
30.8.6 DFSDM 滤波器控制寄存器 (DFSDM_FLTxFCR) 1031
30.8.7 注入组的 DFSDM 数据寄存器 (DFSDM_FLTxJDATAR) 1032
30.8.8 常规通道的 DFSDM 数据寄存器 (DFSDM_FLTxRDATAR) 1033
30.8.9 DFSDM 模拟看门狗阈值上限寄存器 (DFSDM_FLTxAWHTR) 1034
30.8.10 DFSDM 模拟看门狗阈值下限寄存器 (DFSDM_FLTxAWLTR) 1035

30.8.11 DFSDM 模拟看门狗状态寄存器 (DFSDM_FLTxAWSR) 1036
30.8.12 DFSDM 模拟看门狗清零标志寄存器 (DFSDM_FLTxAWCFR) 1036
30.8.13 DFSDM 极值检测器最大值寄存器 (DFSDM_FLTxEXMAX) 1037
30.8.14 DFSDM 极值检测器最小值寄存器 (DFSDM_FLTxEXMIN) 1037
30.8.15 DFSDM 转换定时器寄存器 (DFSDM_FLTxCNVTIMR) 1038
30.8.16 DFSDM寄存器映射 1039

31 数字摄像头接口 (DCMI) 1048
31.1 DCMI 简介 1048
31.2 DCMI 主要特性 1048
31.3 DCMI 时钟 1048
31.4 DCMI 功能概述 1048
31.4.1 DCMI 框图 1049
31.4.2 DCMI 内部信号 1050
31.4.3 DMA 接口 1050
31.4.4 DCMI 物理接口 1050
31.4.5 同步 1052
31.4.6 捕获模式 1055
31.4.7 裁剪功能 1056
31.4.8 JPEG 格式 1057
31.4.9 FIFO 1057
31.5 数据格式说明 1057
31.5.1 数据格式 1057
31.5.2 单色格式 1058
31.5.3 RGB 格式 1058
31.5.4 YCbCr 格式 1059
31.5.5 YCbCr 格式——仅含 Y 分量 1059
31.5.6 半分辨率图像提取 1059
31.6 DCMI 中断 1060
31.7 DCMI 寄存器说明 1060
31.7.1 DCMI 控制寄存器 (DCMI_CR) 1060
31.7.2 DCMI 状态寄存器 (DCMI_SR) 1063
31.7.3 DCMI 原始中断状态寄存器 (DCMI_SR) 1064
31.7.4 DCMI 中断使能寄存器 (DCMI_IER) 1065
31.7.5 DCMI 屏蔽中断状态寄存器 (DCMI_MIS) 1066
31.7.6 DCMI 中断清零寄存器 (DCMI_ICR) 1067

31.7.7 DCMI 内嵌同步码寄存器 (DCMI_ESCR) 1068
31.7.8 DCMI 内嵌码同步取消屏蔽寄存器 (DCMI_ESUR) 1069
31.7.9 DCMI 裁剪窗口起点 (DCMI_CWSTRT) 1070
31.7.10 DCMI 裁剪窗口大小 (DCMI_CWSIZE) 1070
31.7.11 DCMI 数据寄存器 (DCMI_DR) 1071
31.7.12 DCMI 寄存器映射 1072

32 LCD-TFT 显示控制器 (LTDC) 1073

32.1 简介 1073
32.2 LTDC 主要特性 1073
32.3 LTDC 功能说明 1074
32.3.1 LTDC 框图 1074
32.3.2 LCD-TFT 内部信号 1074
32.3.3 LCD-TFT 引脚和外部信号接口 1075
32.3.4 LTDC 复位和时钟 1075
32.4 LTDC 可编程参数 1077
32.4.1 LTDC 全局配置参数 1077
32.4.2 层可编程参数 1079
32.5 LTDC 中断 1083
32.6 LTDC 编程步骤 1084
32.7 LTDC 寄存器 1085
32.7.1 LTDC 同步大小配置寄存器 (LTDC_SSCR) 1085
32.7.2 LTDC 后沿配置寄存器 (LTDC_BPCR) 1085
32.7.3 LTDC 有效宽度配置寄存器 (LTDC_AWCR) 1086
32.7.4 LTDC 总宽度配置寄存器 (LTDC_TWCR) 1087
32.7.5 LTDC 全局控制寄存器 (LTDC_GCR) 1087
32.7.6 LTDC 影子重载配置寄存器 (LTDC_SRCR) 1089
32.7.7 LTDC 背景色配置寄存器 (LTDC_BCCR) 1089
32.7.8 LTDC 中断使能寄存器 (LTDC_IER) 1090
32.7.9 LTDC 中断状态寄存器 (LTDC_ISR) 1091
32.7.10 LTDC 中断清零寄存器 (LTDC_ICR) 1091
32.7.11 LTDC 行中断位置配置寄存器 (LTDC_LIPCR) 1092
32.7.12 LTDC 当前位置状态寄存器 (LTDC_CPSR) 1093
32.7.13 LTDC 当前显示状态寄存器 (LTDC_CDSR) 1093
32.7.14 LTDC 第 x 层控制寄存器 (LTDC_LxCR)（其中，x = 1..2） 1094
32.7.15 LTDC 第 x 层窗口水平位置配置寄存器 (LTDC_LxWHPCR)（其中 x=1..2） 1095

32.7.16 LTDC 第 x 层窗口垂直位置配置寄存器 (LTDC_LxWVPCR)（其中 x=1..2） 1096
32.7.17 LTDC 第 x 层色键配置寄存器 (LTDC_LxCKCR)（其中 x=1..2） 1097
32.7.18 LTDC 第 x 层像素格式配置寄存器 (LTDC_LxPFCR)（其中 x=1..2） . . . 1097
32.7.19 LTDC 第 x 层常数 Alpha 配置寄存器 (LTDC_LxCACR)（其中 x=1..2） 1098
32.7.20 LTDC 第 x 层默认颜色配置寄存器 (LTDC_LxDCCR)（其中 x=1..2） . . . 1099
32.7.21 LTDC 第 x 层混合系数配置寄存器 (LTDC_LxBFCR)（其中 x=1..2） . . . 1099
32.7.22 LTDC 第 x 层颜色帧缓冲区地址寄存器 (LTDC_LxCFBAR)（其中 x=1..2） 1101
32.7.23 LTDC 第 x 层颜色帧缓冲区长度寄存器 (LTDC_LxCFBLR)（其中 x=1..2） 1101
32.7.24 LTDC 第 x 层颜色帧缓冲区行数寄存器 (LTDC_LxCFBLNR)（其中 x=1..2） 1102
32.7.25 LTDC 第 x 层 CLUT 写寄存器 (LTDC_LxCLUTWR)（其中 x=1..2） 1103
32.7.26 LTDC 寄存器映射 1104

33 JPEG 编解码器 (JPEG) 1107

33.1 简介 1107
33.2 JPEG 编解码器主要特性 1107
33.3 JPEG 编解码器功能说明 1108
33.3.1 概述 1108
33.3.2 JPEG 内部信号 1109
33.3.3 JPEG 解码程序 1109
33.3.4 JPEG 编码程序 1110
33.4 JPEG 编解码器中断 1112
33.5 JPEG 编解码器寄存器 1113
33.5.1 JPEG 编解码器控制寄存器 0 (JPEG_CONFR0) 1113
33.5.2 JPEG 编解码器配置寄存器 1 (JPEG_CONFR1) 1113
33.5.3 JPEG 编解码器配置寄存器 2 (JPEG_CONFR2) 1114
33.5.4 JPEG 编解码器配置寄存器 3 (JPEG_CONFR3) 1115
33.5.5 JPEG 编解码器配置寄存器 4-7 (JPEG_CONFR4-7) 1115
33.5.6 JPEG 控制寄存器 (JPEG_CR) 1116
33.5.7 JPEG 状态寄存器 (JPEG_SR) 1117
33.5.8 JPEG 清零标志寄存器 (JPEG_CFR) 1118
33.5.9 JPEG 数据输入寄存器 (JPEG_DIR) 1119
33.5.10 JPEG 数据输出寄存器 (JPEG_DOR) 1119
33.5.11 JPEG 编解码器寄存器映射 1120

34 真随机数发生器 (RNG) 1122
34.1 前言 1122
34.2 RNG主要特性 1122
34.3 RNG 功能说明 1122
34.3.1 RNG框图 1122
34.3.2 RNG 内部信号 1123
34.3.3 随机数生成 1123
34.3.4 RNG初始化 1126
34.3.5 RNG 操作 1127
34.3.6 RNG 时钟 1128
34.3.7 错误管理 1128
34.4 RNG 低功耗使用 1129
34.5 RNG 中断 1129
34.6 RNG 处理时间 1130
34.7 熵源验证 1130
34.7.1 前言 1130
34.7.2 验证条件 1130
34.7.3 数据采集 1130
34.8 RNG 寄存器 1131
34.8.1 RNG 控制寄存器 (RNG_CR) 1131
34.8.2 RNG 状态寄存器 (RNG_SR) 1132
34.8.3 RNG 数据寄存器 (RNG_DR) 1133
34.8.4 RNG 寄存器映射 1133

35 加密处理器 (CRYP) 1134
35.1 简介 1134
35.2 CRYP 主要特性 1134
35.3 CRYP 功能说明 1136
35.3.1 CRYP 框图 1136
35.3.2 CRYP 内部信号 1136
35.3.3 CRYP DES/TDES 加密内核 1137
35.3.4 CRYP AES 加密内核 1138
35.3.5 用于执行密码操作的 CRYP 过程 1143
35.3.6 CRYP 忙碌状态 1147
35.3.7 为解密准备 CRYP AES 密钥 1147
35.3.8 CRYP 窃取和数据填充 1148

35.3.9 CRYP 挂起/恢复操作 1150
35.3.10 CRYP DES/TDES 基本链接模式（ECB 和 CBC） 1151
35.3.11 CRYP AES 基本链接模式（ECB 和 CBC） 1155
35.3.12 CRYP AES 计数器模式 (AES-CTR) 1161
35.3.13 CRYP AES Galois/计数器模式 (GCM) 1164
35.3.14 CRYP AES Galois 消息认证码 (GMAC) 1168
35.3.15 CRYP AES CBC-MAC 计数器模式 (CCM) 1169
35.3.16 CRYP 数据寄存器和数据交换 1173
35.3.17 CRYP 密钥寄存器 1177
35.3.18 CRYP 初始化向量寄存器 1178
35.3.19 CRYP DMA 接口 1179
35.3.20 CRYP 差错管理 1181
35.4 CRYP 中断 1181
35.5 CRYP 处理时间 1182
35.6 CRYP 寄存器 1183
35.6.1 CRYP 控制寄存器 (CRYP_CR) 1183
35.6.2 CRYP 状态寄存器 (CRYP_SR) 1185
35.6.3 CRYP 数据输入寄存器 (CRYP_DIN) 1185
35.6.4 CRYP 数据输出寄存器 (CRYP_DOUT) 1186
35.6.5 CRYP DMA 控制寄存器 (CRYP_DMACR) 1187
35.6.6 CRYP 中断屏蔽置位/清零寄存器 (CRYP_IMSCR) 1187
35.6.7 CRYP 原始中断状态寄存器 (CRYP_RISR) 1188
35.6.8 CRYP 屏蔽中断状态寄存器 (CRYP_MISR) 1188
35.6.9 CRYP 密钥寄存器 0L (CRYP_K0LR) 1189
35.6.10 CRYP 密钥寄存器 0R (CRYP_K0RR) 1190
35.6.11 CRYP 密钥寄存器 1L (CRYP_K1LR) 1190
35.6.12 CRYP 密钥寄存器 1R (CRYP_K1RR) 1190
35.6.13 CRYP 密钥寄存器 2L (CRYP_K2LR) 1191
35.6.14 CRYP 密钥寄存器 2R (CRYP_K2RR) 1191
35.6.15 CRYP 密钥寄存器 3L (CRYP_K3LR) 1192
35.6.16 CRYP 密钥寄存器 3R (CRYP_K3RR) 1192
35.6.17 CRYP 初始化向量寄存器 0L (CRYP_IV0LR) 1192
35.6.18 CRYP 初始化向量寄存器 0R (CRYP_IV0RR) 1193
35.6.19 CRYP 初始化向量寄存器 1L (CRYP_IV1LR) 1193
35.6.20 CRYP 初始化向量寄存器 1R (CRYP_IV1RR) 1194
35.6.21 CRYP 上下文交换 GCM-CCM 寄存器 (CRYP_CSGCMCCMxR) . . . 1194
35.6.22 CRYP 上下文交换 GCM 寄存器 (CRYP_CSGCMxR) 1195
35.6.23 CRYP 寄存器映射 1196

36 散列处理器 (HASH) 1199
36.1 前言 1199
36.2 散列主要特性 1199
36.3 散列功能说明 1200
36.3.1 HASH 框图 1200
36.3.2 HASH 内部信号 1200
36.3.3 关于安全散列算法 1201
36.3.4 消息数据馈送 1201
36.3.5 消息摘要计算 1203
36.3.6 消息填充 1204
36.3.7 HMAC 运算 1205
36.3.8 上下文交换 1207
36.3.9 HASH DMA 接口 1209
36.3.10 HASH 错误管理 1209
36.4 HASH 中断 1209
36.5 HASH 处理时间 1210
36.6 散列寄存器 1211
36.6.1 HASH 控制寄存器 (HASH_CR) 1211
36.6.2 散列数据输入寄存器 (HASH_DIN) 1213
36.6.3 散列启动寄存器 (HASH_STR) 1214
36.6.4 HASH 摘要寄存器 (HASH_HR0..7) 1215
36.6.5 散列中断使能寄存器 (HASH_IMR) 1217
36.6.6 散列状态寄存器 (HASH_SR) 1218
36.6.7 散列上下文交换寄存器 (HASH_CSRx) 1219
36.6.8 散列寄存器映射 1220

37 高分辨率定时器 (HRTIM) 1221
37.1 简介 1221
37.2 主要特性 1221
37.3 功能描述 1222
37.3.1 概述 1222
37.3.2 HRTIM 引脚和内部信号 1223
37.3.3 时钟 1225
37.3.4 定时器 A..E 定时单元 1227
37.3.5 主定时器 1244
37.3.6 置位/复位事件优先级和窄脉冲管理 1245

37.3.7 外部事件全局调节 1246
37.3.8 定时单元中的外部事件过滤 1250
37.3.9 延迟保护 1255
37.3.10 寄存器预装载和更新管理 1261
37.3.11 事件在多个定时器之间的传播 1263
37.3.12 输出管理 1267
37.3.13 突发模式控制器 1269
37.3.14 斩波 1277
37.3.15 故障保护 1278
37.3.16 辅助输出 1281
37.3.17 将 HRTIM 与其他定时器或 HRTIM 实例同步 1284
37.3.18 ADC 触发 1287
37.3.19 DAC 触发 1288
37.3.20 HRTIM 中断 1289
37.3.21 DMA 1291
37.3.22 HRTIM 初始化 1294
37.3.23 调试 1295
37.4 应用用例 1296
37.4.1 降压转换器 1296
37.4.2 具有同步整流功能的降压转换器 1297
37.4.3 多相转换器 1298
37.4.4 过渡模式功率因数校正 1299
37.5 HRTIM 寄存器 1301
37.5.1 HRTIM 主定时器控制寄存器 (HRTIM_MCR) 1301
37.5.2 HRTIM 主定时器中断状态寄存器 (HRTIM_MISR) 1304
37.5.3 HRTIM 主定时器中断清零寄存器 (HRTIM_MICR) 1305
37.5.4 HRTIM 主定时器 DMA/中断使能寄存器 (HRTIM_MDIER) 1306
37.5.5 HRTIM 主定时器计数器寄存器 (HRTIM_MCNTR) 1308
37.5.6 HRTIM 主定时器周期寄存器 (HRTIM_MPER) 1308
37.5.7 HRTIM 主定时器重复寄存器 (HRTIM_MREP) 1309
37.5.8 HRTIM 主定时器比较 1 寄存器 (HRTIM_MCMP1R) 1309
37.5.9 HRTIM 主定时器比较 2 寄存器 (HRTIM_MCMP2R) 1310
37.5.10 HRTIM 主定时器比较 3 寄存器 (HRTIM_MCMP3R) 1310
37.5.11 HRTIM 主定时器比较 4 寄存器 (HRTIM_MCMP4R) 1311
37.5.12 HRTIM Timerx 控制寄存器 (HRTIM_TIMxCR) 1312
37.5.13 HRTIM Timerx 中断状态寄存器 (HRTIM_TIMxISR) 1316
37.5.14 HRTIM Timerx 中断清零寄存器 (HRTIM_TIMxICR) 1318

37.5.15 HRTIM Timerx DMA/中断使能寄存器 (HRTIM_TIMxDIER) 1319
37.5.16 HRTIM Timerx 计数器寄存器 (HRTIM_CNTxR) 1322
37.5.17 HRTIM Timerx 周期寄存器 (HRTIM_PERxR) 1322
37.5.18 HRTIM Timerx 重复寄存器 (HRTIM_REPxR) 1323
37.5.19 HRTIM Timerx 比较 1 寄存器 (HRTIM_CMP1xR) 1323
37.5.20 HRTIM Timerx 比较 1 复合寄存器 (HRTIM_CMP1CxR) 1324
37.5.21 HRTIM Timerx 比较 2 寄存器 (HRTIM_CMP2xR) 1324
37.5.22 HRTIM Timerx 比较 3 寄存器 (HRTIM_CMP3xR) 1325
37.5.23 HRTIM Timerx 比较 4 寄存器 (HRTIM_CMP4xR) 1325
37.5.24 HRTIM Timerx 捕获 1 寄存器 (HRTIM_CPT1xR) 1326
37.5.25 HRTIM Timerx 捕获 2 寄存器 (HRTIM_CPT2xR) 1326
37.5.26 HRTIM Timerx 死区寄存器 (HRTIM_DTxR) 1327
37.5.27 HRTIM Timerx 输出 1 置位寄存器 (HRTIM_SETx1R) 1329
37.5.28 HRTIM Timerx 输出 1 复位寄存器 (HRTIM_RSTx1R) 1331
37.5.29 HRTIM Timerx 输出 2 置位寄存器 (HRTIM_SETx2R) 1331
37.5.30 HRTIM Timerx 输出 2 复位寄存器 (HRTIM_RSTx2R) 1332
37.5.31 HRTIM Timerx 外部事件过滤寄存器 1 (HRTIM_EEFxR1) 1333
37.5.32 HRTIM Timerx 外部事件过滤寄存器 2 (HRTIM_EEFxR2) 1335
37.5.33 HRTIM Timerx 复位寄存器 (HRTIM_RSTxR) 1336
37.5.34 HRTIM Timerx 斩波寄存器 (HRTIM_CHPxR) 1340
37.5.35 HRTIM Timerx 捕获 1 控制寄存器 (HRTIM_CPT1xCR) 1341
37.5.36 HRTIM Timerx 捕获 2 控制寄存器 (HRTIM_CPT2xCR) 1342
37.5.37 HRTIM Timerx 输出寄存器 (HRTIM_OUTxR) 1345
37.5.38 HRTIM Timerx 故障寄存器 (HRTIM_FLTxR) 1348
37.5.39 HRTIM 控制寄存器 1 (HRTIM_CR1) 1349
37.5.40 HRTIM 控制寄存器 2 (HRTIM_CR2) 1351
37.5.41 HRTIM 中断状态寄存器 (HRTIM_ISR) 1352
37.5.42 HRTIM 中断清零寄存器 (HRTIM_ICR) 1353
37.5.43 HRTIM 中断使能寄存器 (HRTIM_IER) 1354
37.5.44 HRTIM 输出使能寄存器 (HRTIM_OENR) 1355
37.5.45 HRTIM 输出禁止寄存器 (HRTIM_ODISR) 1356
37.5.46 HRTIM 输出禁止状态寄存器 (HRTIM_ODSR) 1357
37.5.47 HRTIM 突发模式控制寄存器 (HRTIM_BMCR) 1358
37.5.48 HRTIM 突发模式触发寄存器 (HRTIM_BMTRGR) 1360
37.5.49 HRTIM 突发模式比较寄存器 (HRTIM_BMCMPR) 1362
37.5.50 HRTIM 突发模式周期寄存器 (HRTIM_BMPER) 1362
37.5.51 HRTIM 定时器外部事件控制寄存器 1 (HRTIM_EECR1) 1363

37.5.52 HRTIM 定时器外部事件控制寄存器 2 (HRTIM_EECR2) 1365
37.5.53 HRTIM 定时器外部事件控制寄存器 3 (HRTIM_EECR3) 1366
37.5.54 HRTIM ADC 触发 1 寄存器 (HRTIM_ADC1R) 1367
37.5.55 HRTIM ADC 触发 2 寄存器 (HRTIM_ADC2R) 1368
37.5.56 HRTIM ADC 触发 3 寄存器 (HRTIM_ADC3R) 1369
37.5.57 HRTIM ADC 触发 4 寄存器 (HRTIM_ADC4R) 1371
37.5.58 HRTIM 故障输入寄存器 1 (HRTIM_FLTINR1) 1373
37.5.59 HRTIM 故障输入寄存器 2 (HRTIM_FLTINR2) 1375
37.5.60 HRTIM 突发 DMA 主定时器更新寄存器 (HRTIM_BDMUPR) 1377
37.5.61 HRTIM 突发 DMA Timerx 更新寄存器 (HRTIM_BDTxUPR) 1378
37.5.62 HRTIM 突发 DMA 数据寄存器 (HRTIM_BDMADR) 1379
37.5.63 HRTIM 寄存器映射 1380

38 高级控制定时器 (TIM1/TIM8) 1389

38.1 TIM1/TIM8 简介 1389
38.2 TIM1/TIM8 主要特性 1389
38.3 TIM1/TIM8 功能描述 1391
38.3.1 时基单元 1391
38.3.2 计数器模式 1393
38.3.3 重复计数器 1404
38.3.4 外部触发输入 1406
38.3.5 时钟选择 1407
38.3.6 捕获/比较通道 1411
38.3.7 输入捕获模式 1414
38.3.8 PWM 输入模式 1415
38.3.9 强制输出模式 1415
38.3.10 输出比较模式 1416
38.3.11 PWM 模式 1417
38.3.12 不对称 PWM 模式 1420
38.3.13 组合 PWM 模式 1421
38.3.14 组合三相 PWM 模式 1422
38.3.15 互补输出和死区插入 1423
38.3.16 使用断路功能 1425
38.3.17 双向断路输入 1431
38.3.18 发生外部事件时清除 OCxREF 信号 1431
38.3.19 生成 6 步 PWM 1433
38.3.20 单脉冲模式 1434

38.3.21 可再触发单脉冲模式 (OPM) 1435
38.3.22 编码器接口模式 1436
38.3.23 UIF 位重映射 1438
38.3.24 定时器输入异或功能 1439
38.3.25 连接霍尔传感器 1439
38.3.26 定时器同步 1442
38.3.27 ADC 同步 1445
38.3.28 DMA 连续传送模式 1445
38.3.29 调试模式 1446
38.4 TIM1/TIM8 寄存器 1447
38.4.1 TIM1/TIM8控制寄存器 1 (TIMx_CR1) 1447
38.4.2 TIM1/TIM8控制寄存器 2 (TIMx_CR2) 1448
38.4.3 TIM1/TIM8 从模式控制寄存器 (TIMx_SMCR) 1451
38.4.4 TIM1/TIM8 DMA/中断使能寄存器 (TIMx_DIER) 1453
38.4.5 TIM1/TIM8 状态寄存器 (TIMx_SR) 1455
38.4.6 TIM1/TIM8 事件生成寄存器 (TIMx_EGR) 1456
38.4.7 TIM1/TIM8 捕获/比较模式寄存器 1 (TIMx_CCMR1) 1458
38.4.8 TIM1/TIM8 捕获/比较模式寄存器 2 (TIMx_CCMR2) 1461
38.4.9 TIM1/TIM8 捕获/比较使能寄存器 (TIMx_CCER) 1463
38.4.10 TIM1/TIM8计数器 (TIMx_CNT) 1466
38.4.11 TIM1/TIM8 预分频器 (TIMx_PSC) 1466
38.4.12 TIM1/TIM8自动重载寄存器 (TIMx_ARR) 1466
38.4.13 TIM1/TIM8重复计数器寄存器 (TIMx_RCR) 1467
38.4.14 TIM1/TIM8 捕获/比较寄存器 1 (TIMx_CCR1) 1467
38.4.15 TIM1/TIM8 捕获/比较寄存器 2 (TIMx_CCR2) 1468
38.4.16 TIM1/TIM8 捕获/比较寄存器 3 (TIMx_CCR3) 1468
38.4.17 TIM1/TIM8 捕获/比较寄存器 4 (TIMx_CCR4) 1469
38.4.18 TIM1/TIM8 断路和死区寄存器 (TIMx_BDTR) 1469
38.4.19 TIM1/TIM8 DMA 控制寄存器 (TIMx_DCR) 1472
38.4.20 TIM1/TIM8 全传输 DMA 地址 (TIMx_DMAR) 1473
38.4.21 TIM1/TIM8 捕获/比较模式寄存器 3 (TIMx_CCMR3) 1474
38.4.22 TIM1/TIM8 捕获/比较寄存器 5 (TIMx_CCR5) 1475
38.4.23 TIM1/TIM8 捕获/比较寄存器 6 (TIMx_CCR6) 1476
38.4.24 TIM1 复用功能选项寄存器 1 (TIM1_AF1) 1476
38.4.25 TIM1 复用功能寄存器 2 (TIM1_AF2) 1478
38.4.26 TIM8 复用功能选项寄存器 1 (TIM8_AF1) 1479
38.4.27 TIM8 复用功能选项寄存器 2 (TIM8_AF2) 1480

38.4.28 TIM1 定时器输入选择寄存器 (TIM1_TISEL) 1482
38.4.29 TIM8 定时器输入选择寄存器 (TIM8_TISEL) 1483
38.4.30 TIM1 寄存器映射 1484
38.4.31 TIM8 寄存器映射 1487

39 通用定时器 (TIM2/TIM3/TIM4/TIM5) 1490

39.1 TIM2/TIM3/TIM4/TIM5 简介 1490
39.2 TIM2/TIM3/TIM4/TIM5 主要特性 1490
39.3 TIM2/TIM3/TIM4/TIM5 功能描述 1492
39.3.1 时基单元 1492
39.3.2 计数器模式 1494
39.3.3 时钟选择 1504
39.3.4 捕获/比较通道 1508
39.3.5 输入捕获模式 1510
39.3.6 PWM 输入模式 1511
39.3.7 强制输出模式 1512
39.3.8 输出比较模式 1512
39.3.9 PWM 模式 1513
39.3.10 不对称 PWM 模式 1516
39.3.11 组合 PWM 模式 1517
39.3.12 发生外部事件时清除 OCxREF 信号 1518
39.3.13 单脉冲模式 1519
39.3.14 可再触发单脉冲模式 (OPM) 1520
39.3.15 编码器接口模式 1521
39.3.16 UIF 位重映射 1523
39.3.17 定时器输入异或功能 1523
39.3.18 定时器与外部触发同步 1523
39.3.19 定时器同步 1527
39.3.20 DMA 连续传送模式 1531
39.3.21 调试模式 1532
39.4 TIM2/TIM3/TIM4/TIM5 寄存器 1533
39.4.1 TIMx 控制寄存器 1 (TIMx_CR1) 1533
39.4.2 TIMx 控制寄存器 2 (TIMx_CR2) 1535
39.4.3 TIMx 从模式控制寄存器 (TIMx_SMCR) 1536
39.4.4 TIMx DMA/中断使能寄存器 (TIMx_DIER) 1539
39.4.5 TIMx 状态寄存器 (TIMx_SR) 1540
39.4.6 TIMx 事件生成寄存器 (TIMx_EGR) 1541

39.4.7 TIMx 捕获/比较模式寄存器 1 (TIMx_CCMR1) 1542
39.4.8 TIMx 捕获/比较模式寄存器 2 (TIMx_CCMR2) 1545
39.4.9 TIMx 捕获/比较使能寄存器 (TIMx_CCER) 1547
39.4.10 TIMx 计数器 (TIMx_CNT) 1549
39.4.11 TIMx 预分频器 (TIMx_PSC) 1549
39.4.12 TIMx 自动重载寄存器 (TIMx_ARR) 1550
39.4.13 TIMx 捕获/比较寄存器 1 (TIMx_CCR1) 1550
39.4.14 TIMx 捕获/比较寄存器 2 (TIMx_CCR2) 1551
39.4.15 TIMx 捕获/比较寄存器 3 (TIMx_CCR3) 1551
39.4.16 TIMx 捕获/比较寄存器 4 (TIMx_CCR4) 1552
39.4.17 TIMx DMA 控制寄存器 (TIMx_DCR) 1553
39.4.18 TIMx 全传输 DMA 地址 (TIMx_DMAR) 1553
39.4.19 TIM2 复用功能选项寄存器 1 (TIM2_AF1) 1554
39.4.20 TIM3 复用功能选项寄存器 1 (TIM3_AF1) 1554
39.4.21 TIM5 复用功能选项寄存器 1 (TIM5_AF1) 1555
39.4.22 TIM2 定时器输入选择寄存器 (TIM2_TISEL) 1555
39.4.23 TIM3 定时器输入选择寄存器 (TIM3_TISEL) 1556
39.4.24 TIM5 定时器输入选择寄存器 (TIM5_TISEL) 1557
39.4.25 TIMx 寄存器映射 1558

40 通用定时器 (TIM12/TIM13/TIM14) 1561

40.1 TIM12/TIM13/TIM14 简介 1561
40.2 TIM12/TIM13/TIM14 主要特性 1561
40.2.1 TIM12 主要特性 1561
40.2.2 TIM13/TIM14 主要特性 1562
40.3 TIM12/TIM13/TIM14 功能描述 1563
40.3.1 时基单元 1563
40.3.2 计数器模式 1565
40.3.3 时钟选择 1569
40.3.4 捕获/比较通道 1571
40.3.5 输入捕获模式 1573
40.3.6 PWM 输入模式（仅适用于 TIM12） 1573
40.3.7 强制输出模式 1574
40.3.8 输出比较模式 1575
40.3.9 PWM 模式 1576
40.3.10 组合 PWM 模式（仅适用于 TIM12） 1577
40.3.11 单脉冲模式 1578

40.3.12 可再触发单脉冲模式 (OPM)（仅限 TIM12） 1580
40.3.13 UIF 位重映射 1580
40.3.14 定时器输入异或功能 1581
40.3.15 TIM12 外部触发同步 1581
40.3.16 从模式——组合复位 + 触发模式 1583
40.3.17 定时器同步(TIM12) 1583
40.3.18 调试模式 1583
40.4 TIM12 寄存器 1584
40.4.1 TIM12控制寄存器 1 (TIMx_CR1) 1584
40.4.2 TIM12 从模式控制寄存器 (TIMx_SMCR) 1585
40.4.3 TIM12 中断使能寄存器 (TIMx_DIER) 1587
40.4.4 TIM12 状态寄存器 (TIMx_SR) 1588
40.4.5 TIM12 事件生成寄存器 (TIMx_EGR) 1589
40.4.6 TIM12 捕获/比较模式寄存器 1 (TIMx_CCMR1) 1590
40.4.7 TIM12 捕获/比较使能寄存器 (TIMx_CCER) 1593
40.4.8 TIM12计数器 (TIMx_CNT) 1594
40.4.9 TIM12 预分频器 (TIMx_PSC) 1594
40.4.10 TIM12自动重载寄存器 (TIMx_ARR) 1595
40.4.11 TIM12 捕获/比较寄存器 1 (TIMx_CCR1) 1595
40.4.12 TIM12 捕获/比较寄存器 2 (TIMx_CCR2) 1596
40.4.13 TIM12 定时器输入选择寄存器 (TIM12_TISEL) 1596
40.4.14 TIM12 寄存器映射 1597
40.5 TIM13/TIM14 寄存器 1599
40.5.1 TIM13/TIM14控制寄存器 1 (TIMx_CR1) 1599
40.5.2 TIM13/TIM14 中断使能寄存器 (TIMx_DIER) 1601
40.5.3 TIM13/TIM14 状态寄存器 (TIMx_SR) 1601
40.5.4 TIM13/TIM14 事件生成寄存器 (TIMx_EGR) 1602
40.5.5 TIM13/TIM14 捕获/比较模式寄存器 1 (TIMx_CCMR1) 1603
40.5.6 TIM13/TIM14 捕获/比较使能寄存器 (TIMx_CCER) 1605
40.5.7 TIM13/TIM14计数器 (TIMx_CNT) 1606
40.5.8 TIM13/TIM14 预分频器 (TIMx_PSC) 1606
40.5.9 TIM13/TIM14自动重载寄存器 (TIMx_ARR) 1606
40.5.10 TIM13/TIM14 捕获/比较寄存器 1 (TIMx_CCR1) 1607
40.5.11 TIM13 定时器输入选择寄存器 (TIM13_TISEL) 1607
40.5.12 TIM14 定时器输入选择寄存器 (TIM14_TISEL) 1607
40.5.13 TIM13/TIM14 寄存器映射 1608

41 通用定时器 (TIM15/TIM16/TIM17) 1610

41.1 TIM15/TIM16/TIM17前言 1610

41.2 TIM15主要特性 1610

41.3 TIM16/TIM17主要特性 1611

41.4 TIM15/TIM16/TIM17 功能描述 1614

41.4.1 时基单元 1614

41.4.2 计数器模式 1616

41.4.3 重复计数器 1619

41.4.4 时钟选择 1620

41.4.5 捕获/比较通道 1622

41.4.6 输入捕获模式 1625

41.4.7 PWM 输入模式（仅适用于 TIM15） 1626

41.4.8 强制输出模式 1627

41.4.9 输出比较模式 1627

41.4.10 PWM 模式 1628

41.4.11 组合 PWM 模式（仅适用于 TIM15） 1629

41.4.12 互补输出和死区插入 1630

41.4.13 使用断路功能 1632

41.4.14 单脉冲模式 1637

41.4.15 可再触发单脉冲模式 (OPM)（仅限 TIM15） 1638

41.4.16 UIF 位重映射 1639

41.4.17 定时器输入异或功能（仅适用于 TIM15） 1639

41.4.18 外部触发同步（仅适用于 TIM15） 1640

41.4.19 从模式——组合复位 + 触发模式 1642

41.4.20 DMA 连续传送模式 1642

41.4.21 定时器同步 (TIM15) 1643

41.4.22 调试模式 1643

41.5 TIM15寄存器 1644

41.5.1 TIM15 控制寄存器 1 (TIM15_CR1) 1644

41.5.2 TIM15 控制寄存器 2 (TIM15_CR2) 1645

41.5.3 TIM15 从模式控制寄存器 (TIM15_SMCR) 1647

41.5.4 TIM15 DMA/中断使能寄存器 (TIM15_DIER) 1648

41.5.5 TIM15 状态寄存器 (TIM15_SR) 1649

41.5.6 TIM15 事件产生寄存器 (TIM15_EGR) 1651

41.5.7 TIM15 捕获/比较模式寄存器 1 (TIM15_CCMR1) 1652

41.5.8 TIM15 捕获/比较使能寄存器 (TIM15_CCER) 1655

41.5.9 TIM15 计数器 (TIM15_CNT) 1658
41.5.10 TIM15 预分频器 (TIM15_PSC) 1658
41.5.11 TIM15 自动重载寄存器 (TIM15_ARR) 1658
41.5.12 TIM15 重复计数器寄存器 (TIM15_RCR) 1659
41.5.13 TIM15 捕获/比较寄存器 1 (TIM15_CCR1) 1659
41.5.14 TIM15 捕获/比较寄存器 2 (TIM15_CCR2) 1660
41.5.15 TIM15 断路和死区寄存器 (TIM15_BDTR) 1660
41.5.16 TIM15 DMA 控制寄存器 (TIM15_DCR) 1663
41.5.17 TIM15 全传输 DMA 地址 (TIM15_DMAR) 1663
41.5.18 TIM15 复用寄存器 1 (TIM15_AF1) 1664
41.5.19 TIM15 输入选择寄存器 (TIM15_TISEL) 1665
41.5.20 TIM15寄存器映射 1666
41.6 TIM16/TIM17 寄存器 1668
41.6.1 TIM16/TIM17控制寄存器 1 (TIMx_CR1) 1668
41.6.2 TIM16/TIM17控制寄存器 2 (TIMx_CR2) 1669
41.6.3 TIM16/TIM17 DMA/中断使能寄存器 (TIMx_DIER) 1670
41.6.4 TIM16/TIM17 状态寄存器 (TIMx_SR) 1671
41.6.5 TIM16/TIM17 事件生成寄存器 (TIMx_EGR) 1672
41.6.6 TIM16/TIM17 捕获/比较模式寄存器 1 (TIMx_CCMR1) 1673
41.6.7 TIM16/TIM17 捕获/比较使能寄存器 (TIMx_CCER) 1675
41.6.8 TIM16/TIM17计数器 (TIMx_CNT) 1678
41.6.9 TIM16/TIM17 预分频器 (TIMx_PSC) 1678
41.6.10 TIM16/TIM17自动重载寄存器 (TIMx_ARR) 1678
41.6.11 TIM16/TIM17重复计数器寄存器 (TIMx_RCR) 1679
41.6.12 TIM16/TIM17 捕获/比较寄存器 1 (TIMx_CCR1) 1679
41.6.13 TIM16/TIM17 断路和死区寄存器 (TIMx_BDTR) 1680
41.6.14 TIM16/TIM17 DMA 控制寄存器 (TIMx_DCR) 1682
41.6.15 TIM16/TIM17 全传输 DMA 地址 (TIMx_DMAR) 1683
41.6.16 TIM16 复用功能寄存器 1 (TIM16_AF1) 1683
41.6.17 TIM16 输入选择寄存器 (TIM16_TISEL) 1684
41.6.18 TIM17 复用功能寄存器 1 (TIM17_AF1) 1685
41.6.19 TIM17 输入选择寄存器 (TIM17_TISEL) 1686
41.6.20 TIM16/TIM17 寄存器映射 1687

42 基本定时器 (TIM6/TIM7) 1689
42.1 TIM6/TIM7 简介 1689
42.2 TIM6/TIM7 主要特性 1689

42.3 TIM6/TIM7 功能说明 1690
42.3.1 时基单元 1690
42.3.2 计数模式 1692
42.3.3 UIF 位重映射 1695
42.3.4 时钟源 1695
42.3.5 调试模式 1696
42.4 TIM6/TIM7 寄存器 1696
42.4.1 TIM6/TIM7 控制寄存器 1 (TIMx_CR1) 1696
42.4.2 TIM6/TIM7 控制寄存器 2 (TIMx_CR2) 1698
42.4.3 TIM6/TIM7 DMA/中断使能寄存器 (TIMx_DIER) 1698
42.4.4 TIM6/TIM7 状态寄存器 (TIMx_SR) 1699
42.4.5 TIM6/TIM7 事件产生寄存器 (TIMx_EGR) 1699
42.4.6 TIM6/TIM7 计数器 (TIMx_CNT) 1699
42.4.7 TIM6/TIM7 预分频器 (TIMx_PSC) 1700
42.4.8 TIM6/TIM7 自动重载寄存器 (TIMx_ARR) 1700
42.4.9 TIM6/TIM7 寄存器映射 1701

43 低功耗定时器 (LPTIM) 1702
43.1 前言 1702
43.2 LPTIM 主要特性 1702
43.3 LPTIM 实现 1702
43.4 LPTIM 功能说明 1703
43.4.1 LPTIM 框图 1703
43.4.2 LPTIM 引脚和内部信号 1705
43.4.3 LPTIM 输入和触发映射 1705
43.4.4 LPTIM 复位和时钟 1708
43.4.5 干扰滤波器 1708
43.4.6 预分频器 1709
43.4.7 触发多路复用器 1709
43.4.8 工作模式 1710
43.4.9 超时功能 1711
43.4.10 生成波形 1712
43.4.11 寄存器更新 1713
43.4.12 计数器模式 1713
43.4.13 定时器使能 1714
43.4.14 定时器计数器复位； 1714
43.4.15 编码器模式 1714

43.5 LPTIM 中断 1716
43.6 LPTIM 寄存器 1717
43.6.1 LPTIM 中断和状态寄存器 (LPTIM_ISR) 1717
43.6.2 LPTIM 中断清零寄存器 (LPTIM_ICR) 1718
43.6.3 LPTIM 中断使能寄存器 (LPTIM_IER) 1719
43.6.4 LPTIM 配置寄存器 (LPTIM_CFGR) 1720
43.6.5 LPTIM 控制寄存器 (LPTIM_CR) 1723
43.6.6 LPTIM 比较寄存器 (LPTIM_CMP) 1724
43.6.7 LPTIM 自动重载寄存器 (LPTIM_ARR) 1724
43.6.8 LPTIM 计数器寄存器 (LPTIM_CNT) 1725
43.6.9 LPTIM 配置寄存器 2 (LPTIM_CFGR2) 1725
43.6.10 LPTIM3 配置寄存器 2 (LPTIM3_CFGR2) 1726
43.6.11 LPTIM 寄存器映射 1727

44 系统窗口看门狗 (WWDG) 1729
44.1 前言 1729
44.2 WWDG 主要特性 1729
44.3 WWDG 功能说明 1729
44.3.1 WWDG框图 1730
44.3.2 WWDG 内部信号 1730
44.3.3 使能看门狗 1730
44.3.4 控制递减计数器 1731
44.3.5 看门狗中断高级特性 1731
44.3.6 如何设置看门狗超时 1731
44.3.7 调试模式 1732
44.4 WWDG 寄存器 1733
44.4.1 控制寄存器 (WWDG_CR) 1733
44.4.2 配置寄存器 (WWDG_CFR) 1734
44.4.3 状态寄存器 (WWDG_SR) 1735
44.4.4 WWDG 寄存器映射 1735

45 独立看门狗 (IWDG) 1736
45.1 前言 1736
45.2 IWDG 主要特性 1736
45.3 IWDG 功能说明 1736
45.3.1 IWDG 框图 1736
45.3.2 IWDG 内部信号 1737

45.3.3 窗口选项 1737
45.3.4 硬件看门狗 1738
45.3.5 低功耗冻结 1738
45.3.6 停止和待机模式下的行为 1738
45.3.7 寄存器访问保护 1738
45.3.8 调试模式 1738
45.4 IWDG 寄存器 1739
45.4.1 键寄存器 (IWDG_KR) 1739
45.4.2 预分频器寄存器 (IWDG_PR) 1740
45.4.3 重载寄存器 (IWDG_RLR) 1741
45.4.4 状态寄存器 (IWDG_SR) 1742
45.4.5 窗口寄存器 (IWDG_WINR) 1743
45.4.6 IWDG 寄存器映射 1743

46 实时时钟 (RTC) 1744
46.1 前言 1744
46.2 RTC 主要特性 1744
46.3 RTC 功能说明 1745
46.3.1 RTC 框图 1745
46.3.2 RTC 引脚和内部信号 1748
46.3.3 RTC 控制的 GPIO 1748
46.3.4 时钟和预分频器 1750
46.3.5 实时时钟和日历 1750
46.3.6 可编程闹钟 1751
46.3.7 周期性自动唤醒 1751
46.3.8 RTC 初始化和配置 1752
46.3.9 读取日历 1753
46.3.10 复位 RTC 1754
46.3.11 RTC 同步 1754
46.3.12 RTC 参考时钟检测 1755
46.3.13 RTC 精密数字校准 1755
46.3.14 时间戳功能 1757
46.3.15 入侵检测 1757
46.3.16 校准时钟输出 1759
46.3.17 闹钟输出 1759

46.4 RTC 低功耗模式 1760
46.5 RTC 中断 1760
46.6 RTC 寄存器 1761
46.6.1 RTC 时间寄存器 (RTC_TR) 1761
46.6.2 RTC 日期寄存器 (RTC_DR) 1762
46.6.3 RTC 控制寄存器 (RTC_CR) 1763
46.6.4 RTC 初始化和状态寄存器 (RTC_ISR) 1766
46.6.5 RTC 预分频器寄存器 (RTC_PRER) 1768
46.6.6 RTC 唤醒定时器寄存器 (RTC_WUTR) 1769
46.6.7 RTC 闹钟 A 寄存器 (RTC_ALRMAR) 1770
46.6.8 RTC 闹钟 B 寄存器 (RTC_ALRMBR) 1771
46.6.9 RTC 写保护寄存器 (RTC_WPR) 1772
46.6.10 RTC 亚秒寄存器 (RTC_SSR) 1772
46.6.11 RTC 平移控制寄存器 (RTC_SHIFTR) 1773
46.6.12 RTC 时间戳时间寄存器 (RTC_TSTR) 1774
46.6.13 RTC 时间戳日期寄存器 (RTC_TSDR) 1775
46.6.14 RTC 时间戳亚秒寄存器 (RTC_TSSSR) 1775
46.6.15 RTC 校准寄存器 (RTC_CALR) 1776
46.6.16 RTC 入侵配置寄存器 (RTC_TAMPCR) 1777
46.6.17 RTC 闹钟 A 亚秒寄存器 (RTC_ALRMASSR) 1780
46.6.18 RTC 闹钟 B 亚秒寄存器 (RTC_ALRMBSSR) 1781
46.6.19 RTC 选项寄存器 (RTC_OR) 1782
46.6.20 RTC 备份寄存器 (RTC_BKPxR) 1782
46.6.21 RTC 寄存器映射 1783

47 内部集成电路 (I2C) 接口 1785
47.1 前言 1785
47.2 I2C 主要特性 1785
47.3 I2C 特性实现 1786
47.4 I2C 功能说明 1786
47.4.1 I2C 框图 1787
47.4.2 I2C 时钟要求 1788
47.4.3 模式选择 1788
47.4.4 I2C 初始化 1789
47.4.5 软件复位 1793
47.4.6 数据传输 1794

47.4.7 I2C 从模式 1796
47.4.8 I2C 主模式 1804
47.4.9 I2C_TIMINGR 寄存器配置示例 1815
47.4.10 SMBus 特性 1817
47.4.11 SMBus 初始化 1820
47.4.12 SMBus：I2C_TIMEOUTR 寄存器配置示例 1822
47.4.13 SMBus 从模式 1822
47.4.14 地址匹配时从停止模式唤醒 1829
47.4.15 错误条件 1829
47.4.16 DMA 请求 1831
47.4.17 调试模式 1831
47.5 I2C 低功耗模式 1832
47.6 I2C 中断 1832
47.7 I2C 寄存器 1833
47.7.1 控制寄存器 1 (I2C_CR1) 1833
47.7.2 控制寄存器 2 (I2C_CR2) 1836
47.7.3 设备自身地址 1 寄存器 (I2C_OAR1) 1839
47.7.4 设备自身地址 2 寄存器 (I2C_OAR2) 1840
47.7.5 时序寄存器 (I2C_TIMINGR) 1841
47.7.6 超时寄存器 (I2C_TIMEOUTR) 1842
47.7.7 中断和状态寄存器 (I2C_ISR) 1843
47.7.8 中断清零寄存器 (I2C_ICR) 1845
47.7.9 PEC 寄存器 (I2C_PECR) 1846
47.7.10 接收数据寄存器 (I2C_RXDR) 1847
47.7.11 发送数据寄存器 (I2C_TXDR) 1847
47.7.12 I2C寄存器映射 1848

48 通用同步异步收发器 (USART) 1850
48.1 USART 简介 1850
48.2 USART 主要特性 1850
48.3 USART 扩展特性 1851
48.4 USART 实现 1851
48.5 USART 功能说明 1852
48.5.1 USART 框图 1852
48.5.2 USART 信号 1853
48.5.3 USART 字符说明 1854

48.5.4 USART FIFO 和阈值 1856
48.5.5 USART 发送器 1856
48.5.6 USART 接收器 1859
48.5.7 USART 波特率生成 1865
48.5.8 USART 接收器对时钟偏差的容差 1867
48.5.9 USART 自动波特率检测 1868
48.5.10 USART 多处理器通信 1869
48.5.11 USART Modbus 通信 1871
48.5.12 USART 极性控制 1871
48.5.13 USART LIN（局域互连网络）模式 1872
48.5.14 USART 同步模式 1874
48.5.15 USART 单线半双工通信 1878
48.5.16 USART 接收器超时 1878
48.5.17 USART 智能卡模式 1879
48.5.18 USART IrDA SIR ENDEC 模块 1882
48.5.19 使用 USART 和 DMA 进行连续通信 1884
48.5.20 RS232 硬件流控制和 RS485 驱动器使能 1886
48.5.21 USART 低功耗管理 1889
48.6 USART 中断 1891
48.7 USART 寄存器 1893
48.7.1 USART 控制寄存器 1 (USART_CR1) 1893
48.7.2 USART 控制寄存器 2 (USART_CR2) 1896
48.7.3 USART 控制寄存器 3 (USART_CR3) 1900
48.7.4 USART 波特率寄存器 (USART_BRR) 1905
48.7.5 USART 保护时间和预分频器寄存器 (USART_GTPR) 1905
48.7.6 USART 接收器超时寄存器 (USART_RTOR) 1906
48.7.7 USART 请求寄存器 (USART_RQR) 1907
48.7.8 USART 中断和状态寄存器 (USART_ISR) 1908
48.7.9 USART 中断标志清零寄存器 (USART_ICR) 1913
48.7.10 USART 接收数据寄存器 (USART_RDR) 1915
48.7.11 USART 发送数据寄存器 (USART_TDR) 1915
48.7.12 USART 预分频器寄存器 (USART_PRESC) 1916
48.7.13 USART 寄存器映射 1917

49 低功耗通用异步接收器 (LPUART) 1919
49.1 LPUART 简介 1919
49.2 LPUART 主要特性 1920

49.3 LPUART 功能说明 1921
49.3.1 LPUART框图 1921
49.3.2 LPUART 信号 1922
49.3.3 LPUART 字符说明 1922
49.3.4 LPUART FIFO 和阈值 1924
49.3.5 LPUART 发送器 1924
49.3.6 LPUART 接收器 1927
49.3.7 LPUART 波特率生成 1931
49.3.8 LPUART 接收器对时钟偏差的容差 1932
49.3.9 LPUART 多处理器通信 1933
49.3.10 LPUART 极性控制 1935
49.3.11 LPUART 单线半双工通信 1936
49.3.12 使用 DMA 和 LPUART 进行连续通信 1936
49.3.13 RS232 硬件流控制和 RS485 驱动器使能 1939
49.3.14 LPUART 低功耗管理 1941
49.4 LPUART 中断 1943
49.5 LPUART寄存器 1945
49.5.1 控制寄存器 1 (LPUART_CR1) 1945
49.5.2 控制寄存器 2 (LPUART_CR2) 1948
49.5.3 控制寄存器 3 (LPUART_CR3) 1949
49.5.4 波特率寄存器 (LPUART_BRR) 1952
49.5.5 请求寄存器 (LPUART_RQR) 1953
49.5.6 中断和状态寄存器 (LPUART_ISR) 1953
49.5.7 中断标志清零寄存器 (LPUART_ICR) 1957
49.5.8 接收数据寄存器 (LPUART_RDR) 1959
49.5.9 发送数据寄存器 (LPUART_TDR) 1959
49.5.10 预分频器寄存器 (LPUART_PRESC) 1960
49.5.11 LPUART 寄存器映射 1961

50 串行外设接口 (SPI) 1962
50.1 前言 1962
50.2 SPI 主要特性 1962
50.3 SPI 实现 1963
50.4 SPI 功能说明 1963
50.4.1 SPI 框图 1963
50.4.2 SPI 信号 1964

50.4.3 SPI 通信一般情况 1964
50.4.4 一个主器件和一个从器件之间的通信 1965
50.4.5 标准多从器件通信 1967
50.4.6 多主通信 1970
50.4.7 从器件选择 (SS) 引脚管理 1970
50.4.8 通信格式 1974
50.4.9 SPI 配置 1976
50.4.10 使能 SPI 的步骤 1977
50.4.11 SPI 数据发送和接收过程 1977
50.4.12 禁止 SPI 的步骤 1980
50.4.13 数据打包 1981
50.4.14 使用 DMA（直接存储器寻址）进行通信 1982
50.5 SPI 特定模式和控制 1984
50.5.1 TI 模式 1984
50.5.2 SPI 错误标志 1984
50.5.3 CRC 计算 1986
50.6 低功耗模式管理 1987
50.7 SPI 唤醒和中断 1989
50.8 I2S 主要特性 1990
50.9 I2S 功能说明 1990
50.9.1 I2S 一般说明 1990
50.9.2 与 SPI 功能共享的引脚 1991
50.9.3 I2S/PCM 模式下可用的位和位域 1991
50.9.4 从模式和主模式 1992
50.9.5 支持的音频协议 1992
50.9.6 更灵活的串行接口 1997
50.9.7 启动序列 1999
50.9.8 停止序列 2000
50.9.9 时钟发生器 2001
50.9.10 内部 FIFO 2004
50.9.11 FIFOs 状态标志 2005
50.9.12 下溢情况的处理 2006
50.9.13 上溢情况的处理 2007
50.9.14 帧错误检测 2007
50.9.15 DMA 接口 2009
50.9.16 编程示例 2010
50.9.17 从模式 I2S Philips 标准，接收 2012

50.10 I2S 唤醒和中断 2013
50.11 SPI/I2S 寄存器 2014
50.11.1 SPI2S 控制寄存器 1 (SPI/I2S_CR1) 2014
50.11.2 SPI 控制寄存器 2 (SPI_CR2) 2016
50.11.3 SPI 配置寄存器 1 (SPI_CFG1) 2016
50.11.4 SPI 配置寄存器 2 (SPI_CFG2) 2019
50.11.5 SPI/I2S 中断使能寄存器 (SPI2S_IER) 2021
50.11.6 SPI/I2S 状态寄存器 (SPI2S_SR) 2022
50.11.7 SPI/I2S 中断/状态标志清零寄存器 (SPI2S_IFCR) 2025
50.11.8 SPI/I2S 发送数据寄存器 (SPI2S_TXDR) 2026
50.11.9 SPI/I2S 接收数据寄存器 (SPI2S_RXDR) 2026
50.11.10 SPI 多项式寄存器 (SPI_CRCPOLY) 2027
50.11.11 SPI 发送器 CRC 寄存器 (SPI_TXCRC) 2027
50.11.12 SPI 接收器 CRC 寄存器 (SPI_RXCRC) 2028
50.11.13 SPI 下溢数据寄存器 (SPI_UDRDR) 2028
50.11.14 SPI/I2S 配置寄存器 (SPI_I2SCGFR) 2029
50.12 SPI 寄存器映射和复位值 2031

51 串行音频接口 (SAI) 2033
51.1 简介 2033
51.2 SAI 主要特性 2033
51.3 SAI 功能说明 2034
51.3.1 SAI 框图 2034
51.3.2 SAI 引脚和内部信号 2035
51.3.3 SAI 的主要模式 2036
51.3.4 SAI 同步模式 2037
51.3.5 音频数据大小 2038
51.3.6 帧同步 2038
51.3.7 Slot 配置 2041
51.3.8 SAI 时钟发生器 2043
51.3.9 内部 FIFO 2045
51.3.10 PDM 接口 2047
51.3.11 AC’97 链路控制器 2055
51.3.12 SPDIF 输出 2057
51.3.13 特性 2059
51.3.14 错误标志 2063

51.3.15 禁止 SAI 2066
51.3.16 SAI DMA 接口 2066
51.4 SAI 中断 2067
51.5 SAI 寄存器 2068
51.5.1 全局配置寄存器 (SAI_GCR) 2068
51.5.2 配置寄存器 1 (SAI_ACR1/SAI_BCR1) 2068
51.5.3 配置寄存器 2 (SAI_ACR2/SAI_BCR2) 2071
51.5.4 帧配置寄存器 (SAI_AFRCR/SAI_BFRCR) 2073
51.5.5 Slot 寄存器 (SAI_ASLOTR/SAI_BSLOTR) 2074
51.5.6 中断屏蔽寄存器 2 (SAI_AIM/SAI_BIM) 2075
51.5.7 状态寄存器 (SAI_ASR/SAI_BSR) 2076
51.5.8 清除标志寄存器 (SAI_ACLRFR/SAI_BCLRFR) 2078
51.5.9 数据寄存器 (SAI_ADR/SAI_BDR) 2080
51.5.10 PDM 控制寄存器 (SAI_PDMCR) 2080
51.5.11 PDM 延迟寄存器 (SAI_PDMDLY) 2081
51.5.12 SAI 寄存器映射 2083

52 SPDIF 接收器接口 (SPDIFRX) 2085
52.1 SPDIFRX 接口简介 2085
52.2 SPDIFRX 主要特性 2085
52.3 SPDIFRX 功能说明 2085
52.3.1 SPDIFRX 引脚和内部信号 2086
52.3.2 S/PDIF 协议 (IEC-60958) 2087
52.3.3 SPDIFRX 解码器 (SPDIFRX_DC) 2089
52.3.4 SPDIFRX 对时钟偏差的容差 2093
52.3.5 SPDIFRX 同步 2093
52.3.6 SPDIFRX 处理 2095
52.3.7 数据接收管理 2097
52.3.8 专用控制流 2099
52.3.9 接收错误 2100
52.3.10 时钟策略 2102
52.3.11 符号时钟生成 2103
52.3.12 DMA 接口 2104
52.3.13 中断生成 2105
52.3.14 寄存器保护 2106

52.4 编程步骤 2106
52.4.1 初始化阶段 2107
52.4.2 处理来自 SPDIFRX 的中断 2108
52.4.3 处理来自 DMA 的中断 2108
52.5 SPDIFRX 接口寄存器 2109
52.5.1 控制寄存器 (SPDIFRX_CR) 2109
52.5.2 中断屏蔽寄存器 (SPDIFRX_IMR) 2112
52.5.3 状态寄存器 (SPDIFRX_SR) 2113
52.5.4 中断标志清零寄存器 (SPDIFRX_IFCR) 2115
52.5.5 数据输入寄存器 (SPDIFRX_DR) 2116
52.5.6 数据输入寄存器 (SPDIFRX_DR) 2117
52.5.7 数据输入寄存器 (SPDIFRX_DR) 2118
52.5.8 通道状态寄存器 (SPDIFRX_CSR) 2119
52.5.9 调试信息寄存器 (SPDIFRX_DIR) 2120
52.5.10 SPDIFRX 版本寄存器 (SPDIFRX_VERR) 2121
52.5.11 SPDIFRX 标识寄存器 (SPDIFRX_IDR) 2121
52.5.12 SPDIFRX 大小标识寄存器 (SPDIFRX_SIDR) 2122
52.5.13 SPDIFRX 接口寄存器映射 2123

53 单线协议主接口 (SWPMI) 2124
53.1 简介 2124
53.2 SWPMI 主要特性 2125
53.3 SWPMI 功能说明 2126
53.3.1 SWPMI 框图 2126
53.3.2 SWPMI 引脚和内部信号 2126
53.3.3 SWP 初始化和激活 2127
53.3.4 SWP 总线状态 2127
53.3.5 SWPMI_IO（内部收发器）旁路 2129
53.3.6 SWPMI 比特率 2129
53.3.7 SWPMI 帧处理 2129
53.3.8 发送过程 2130
53.3.9 接收过程 2134
53.3.10 错误管理 2139
53.3.11 回送模式 2141
53.4 SWPMI 低功耗模式 2141
53.5 SWPMI 中断 2142

53.6 SWPMI 寄存器 2143
53.6.1 SWPMI 配置/控制寄存器 (SWPMI_CR) 2143
53.6.2 SWPMI 比特率寄存器 (SWPMI_BRR) 2144
53.6.3 SWPMI 中断和状态寄存器 (SWPMI_ISR) 2145
53.6.4 SWPMI 中断标志清零寄存器 (SWPMI_ICR) 2146
53.6.5 SWPMI 中断使能寄存器 (SMPMI_IER) 2147
53.6.6 SWPMI 接收帧长度寄存器 (SWPMI_RFL) 2149
53.6.7 SWPMI 发送数据寄存器 (SWPMI_TDR) 2149
53.6.8 SWPMI 接收数据寄存器 (SWPMI_RDR) 2150
53.6.9 SWPMI 选项寄存器 (SWPMI_OR) 2150
53.6.10 SWPMI 寄存器映射和复位值表 2151

54 管理数据输入/输出 (MDIOS) 2152
54.1 MDIOS 简介 2152
54.2 MDIOS 主要特性 2152
54.3 MDIOS 功能说明 2153
54.3.1 MDIOS 框图 2153
54.3.2 MDIOS 引脚和内部信号 2153
54.3.3 MDIOS 协议 2154
54.3.4 MDIOS 使能和禁止 2155
54.3.5 MDIOS 数据 2155
54.3.6 MDIOS APB 频率 2156
54.3.7 写入/读取标志和中断 2156
54.3.8 MDIOS 错误管理 2156
54.3.9 停止模式下的 MDIOS 2157
54.3.10 MDIOS 中断 2157
54.4 MDIOS 寄存器 2158
54.4.1 MDIOS 配置寄存器 (MDIOS_CR) 2158
54.4.2 MDIOS 写入标志寄存器 (MDIOS_WRFR) 2159
54.4.3 MDIOS 清零写入标志寄存器 (MDIOS_CWRFR) 2159
54.4.4 MDIOS 读取标志寄存器 (MDIOS_RDFR) 2160
54.4.5 MDIOS 清零读取标志寄存器 (MDIOS_CRDFR) 2160
54.4.6 MDIOS 状态寄存器 (MDIOS_SR) 2161
54.4.7 MDIOS 清零标志寄存器 (MDIOS_CLRFR) 2162
54.4.8 MDIOS 输入数据寄存器 (MDIOS_DINR0-MDIOS_DINR31) 2163
54.4.9 MDIOS 输出数据寄存器 (MDIOS_DOUTR0-MDIOS_DOUTR31) . . . 2163
54.4.10 MDIOS 寄存器映射 2164

55 安全数字输入/输出多媒体卡接口 (SDMMC) 2166

55.1 SDMMC 主要特性 2166

55.2 SDMMC 总线拓扑 2166

55.3 SDMMC 工作模式 2168

55.4 SDMMC 功能说明 2169

55.4.1 SDMMC 图 2169

55.4.2 SDMMC 引脚和内部信号 2170

55.4.3 概述 2170

55.4.4 SDMMC 适配器 2172

55.4.5 SDMMC AHB 从接口 2191

55.4.6 SDMMC AHB 主接口 2192

55.4.7 MDMA 请求生成 2193

55.4.8 AHB 和 SDMMC_CK 时钟的关系 2193

55.5 卡功能说明 2194

55.5.1 SD I/O 模式 2194

55.5.2 CMD12 发送时序 2202

55.5.3 睡眠 (CMD5) 2204

55.5.4 中断模式 (Wait-IRQ) 2205

55.5.5 启动操作 2206

55.5.6 响应 R1b 的处理 2209

55.5.7 复位和卡掉电再上电 2209

55.6 硬件流控制 2211

55.7 超高速 I 相 (UHS-I) 电压切换 2211

55.8 SDMMC 寄存器 2214

55.8.1 SDMMC 电源控制寄存器 (SDMMC_POWER) 2214

55.8.2 SDMMC 时钟控制寄存器 (SDMMC_CLKCR) 2215

55.8.3 SDMMC 参数寄存器 (SDMMC_ARGR) 2217

55.8.4 SDMMC 命令寄存器 (SDMMC_CMDR) 2217

55.8.5 SDMMC 命令响应寄存器 (SDMMC_RESPCMDR) 2219

55.8.6 SDMMC 响应 1..4 寄存器 (SDMMC_RESPxR) (x = 1..4) 2219

55.8.7 SDMMC 数据定时器寄存器 (SDMMC_DTIMER) 2220

55.8.8 SDMMC 数据长度寄存器 (SDMMC_DLENR) 2221

55.8.9 SDMMC 数据控制寄存器 (SDMMC_DCTRL) 2222

55.8.10 SDMMC 数据计数器寄存器 (SDMMC_DCNTR) 2224

55.8.11 SDMMC 状态寄存器 (SDMMC_STAR) 2225

55.8.12 SDMMC 中断清零寄存器 (SDMMC_ICR) 2227

55.8.13 SDMMC 屏蔽寄存器 (SDMMC_MASKR) 2230
55.8.14 SDMMC 确认定时器寄存器 (SDMMC_ACKTIMER) 2232
55.8.15 SDMMC 数据 FIFO 寄存器 (SDMMC_FIFOR) 2233
55.8.16 SDMMC DMA 控制寄存器 (SDMMC_IDMACTRLR) 2233
55.8.17 SDMMC IDMA 缓冲区大小寄存器 (SDMMC_IDMABSIZER) 2235
55.8.18 SDMMC IDMA 缓冲区 0 基址寄存器 (SDMMC_IDMABASE0R) 2235
55.8.19 SDMMC IDMA 缓冲区 1 基址寄存器 (SDMMC_IDMABASE1R) 2236
55.8.20 SDMMC 寄存器映射 2237

56 FD 控制器局域网络（FDCAN） 2239

56.1 简介 2239
56.2 FDCAN 主要特性 2241
56.3 FDCAN 功能说明 2242
56.3.1 工作模式 2243
56.3.2 消息 RAM 2251
56.3.3 FIFO 确认处理 2262
56.3.4 CAN 时钟校准 2262
56.3.5 TTCAN 操作（仅限 FDCAN1） 2266
56.3.6 TTCAN 配置 2268
56.3.7 消息调度 2270
56.3.8 TTCAN 间隙控制 2276
56.3.9 停止监视 2276
56.3.10 本地时间、周期时间、全局时间和外部时钟同步 2277
56.3.11 TTCAN 错误级别 2279
56.3.12 TTCAN 消息处理 2280
56.3.13 TTCAN 中断和错误处理 2283
56.3.14 0 级 2283
56.3.15 与外部时间调度同步 2286
56.3.16 FDCAN 接收缓冲区和 FIFO 元素 2286
56.3.17 FDCAN 发送缓冲区元素 2288
56.3.18 FDCAN 发送事件 FIFO 元素 2290
56.3.19 FDCAN 标准消息 ID 过滤器元素 2291
56.3.20 FDCAN 扩展消息 ID 过滤器元素 2293
56.3.21 FDCAN 触发存储器元素 2294
56.4 FDCAN 寄存器 2295
56.4.1 FDCAN 内核释放寄存器 (FDCAN_CREL) 2295
56.4.2 FDCAN 字节序寄存器 (FDCAN_ENDN) 2296

56.4.3 FDCAN 数据位定时和预分频器寄存器 (FDCAN_DBTP) 2296
56.4.4 FDCAN 测试寄存器 (FDCAN_TEST) 2297
56.4.5 FDCAN RAM 看门狗寄存器 (FDCAN_RWD) 2298
56.4.6 FDCAN CC 控制寄存器 (FDCAN_CCCR) 2299
56.4.7 FDCAN 标称位定时和预分频器寄存器 (FDCAN_NBTP) 2300
56.4.8 FDCAN 时间戳计数器配置寄存器 (FDCAN_TSCC) 2301
56.4.9 FDCAN 时间戳计数器值寄存器 (FDCAN_TSCV) 2303
56.4.10 FDCAN 超时计数器配置寄存器 (FDCAN_TOCC) 2303
56.4.11 FDCAN 超时计数器值寄存器 (FDCAN_TOCV) 2304
56.4.12 FDCAN 错误计数器寄存器 (FDCAN_ECR) 2305
56.4.13 FDCAN 协议状态寄存器 (FDCAN_PSR) 2306
56.4.14 FDCAN 发送器延迟补偿寄存器 (FDCAN_TDCR) 2308
56.4.15 FDCAN 中断寄存器 (FDCAN_IR) 2308
56.4.16 FDCAN 中断使能寄存器 (FDCAN_IE) 2311
56.4.17 FDCAN 中断线选择寄存器 (FDCAN_ILS) 2314
56.4.18 FDCAN 中断线使能寄存器 (FDCAN_ILE) 2315
56.4.19 FDCAN 全局过滤器配置寄存器 (FDCAN_GFC) 2316
56.4.20 FDCAN 标准 ID 过滤器配置寄存器 (FDCAN_SIDFC) 2317
56.4.21 FDCAN 扩展 ID 过滤器配置寄存器 (FDCAN_XIDFC) 2318
56.4.22 FDCAN 扩展 ID 和掩码寄存器 (FDCAN_XIDAM) 2318
56.4.23 FDCAN 高优先级消息状态寄存器 (FDCAN_HPMS) 2319
56.4.24 FDCAN 新数据 1 寄存器 (FDCAN_NDAT1) 2320
56.4.25 FDCAN 新数据 2 寄存器 (FDCAN_NDAT2) 2320
56.4.26 FDCAN 接收 FIFO 0 配置寄存器 (FDCAN_RXF0C) 2321
56.4.27 FDCAN 接收 FIFO 0 状态寄存器 (FDCAN_RXF0S) 2321
56.4.28 FDCAN 接收 FIFO 0 确认寄存器 (FDCAN_RXF0A) 2322
56.4.29 FDCAN 接收缓冲区配置寄存器 (FDCAN_RXBC) 2323
56.4.30 FDCAN 接收 FIFO 1 配置寄存器 (FDCAN_RXF1C) 2323
56.4.31 FDCAN 接收 FIFO 1 状态寄存器 (FDCAN_RXF1S) 2324
56.4.32 FDCAN 接收 FIFO 1 确认寄存器 (FDCAN_RXF1A) 2325
56.4.33 FDCAN 接收缓冲区元素大小配置寄存器 (FDCAN_RXESC) 2325
56.4.34 FDCAN 发送缓冲区配置寄存器 (FDCAN_TXBC) 2326
56.4.35 FDCAN 发送 FIFO/队列状态寄存器 (FDCAN_TXFQS) 2327
56.4.36 FDCAN 发送缓冲区元素大小配置寄存器 (FDCAN_TXESC) 2328
56.4.37 FDCAN 发送缓冲区请求挂起寄存器 (FDCAN_TXBRP) 2329
56.4.38 FDCAN 发送缓冲区添加请求寄存器 (FDCAN_TXBAR) 2330
56.4.39 FDCAN 发送缓冲区取消请求寄存器 (FDCAN_TXBCR) 2330

56.4.40 FDCAN 发送缓冲区发送已发生寄存器 (FDCAN_TXBTO) 2331
56.4.41 FDCAN 发送缓冲区取消完成寄存器 (FDCAN_TXBCF) 2331
56.4.42 FDCAN 发送缓冲区发送中断使能寄存器 (FDCAN_TXBTIE) 2332
56.4.43 FDCAN 发送缓冲区取消完成中断使能寄存器 (FDCAN_TXBCIE) . . . 2332
56.4.44 FDCAN 发送事件 FIFO 配置寄存器 (FDCAN_TXEFC) 2333
56.4.45 FDCAN 发送事件 FIFO 状态寄存器 (FDCAN_TXEFS) 2334
56.4.46 FDCAN 发送事件 FIFO 确认寄存器 (FDCAN_TXEFA) 2335
56.4.47 FDCAN TT 触发存储器配置寄存器 (FDCAN_TTTMC) 2335
56.4.48 FDCAN TT 参考消息配置寄存器 (FDCAN_TTRMC) 2336
56.4.49 FDCAN TT 操作配置寄存器 (FDCAN_TTOCF) 2337
56.4.50 FDCAN TT 矩阵限值寄存器 (FDCAN_TTMLM) 2338
56.4.51 FDCAN TUR 配置寄存器 (FDCAN_TURCF) 2339
56.4.52 FDCAN TT 操作控制寄存器 (FDCAN_TTOCN) 2340
56.4.53 FDCAN TT 全局时间预设寄存器 (CAN_TTGTP) 2342
56.4.54 FDCAN TT 时间标记寄存器 (FDCAN_TTTMK) 2343
56.4.55 FDCAN TT 中断寄存器 (FDCAN_TTIR) 2344
56.4.56 FDCAN TT 中断使能寄存器 (FDCAN_TTIE) 2346
56.4.57 FDCAN TT 中断线选择寄存器 (FDCAN_TTILS) 2348
56.4.58 FDCAN TT 工作状态寄存器 (FDCAN_TTOST) 2349
56.4.59 FDCAN TUR 分子实际寄存器 (FDCAN_TURNA) 2351
56.4.60 FDCAN TT 本地和全局时间寄存器 (FDCAN_TTLGT) 2352
56.4.61 FDCAN TT 周期时间和计数寄存器 (FDCAN_TTCTC) 2352
56.4.62 FDCAN TT 捕获时间寄存器 (FDCAN_TTCPT) 2353
56.4.63 FDCAN TT 周期同步标记寄存器 (FDCAN_TTCSM) 2353
56.4.64 FDCAN TT 触发选择寄存器 (FDCAN_TTTS) 2354
56.4.65 FDCAN 寄存器映射和复位值表 2355
56.5 CCU寄存器 2361
56.5.1 时钟校准单元内核释放寄存器 (CCU_CREL) 2361
56.5.2 校准配置寄存器 (CCU_CCFG) 2361
56.5.3 校准状态寄存器 (CCU_CSTAT) 2363
56.5.4 校准看门狗寄存器 (CCU_CWD) 2364
56.5.5 时钟校准单元中断寄存器 (CCU_IR) 2364
56.5.6 时钟校准单元中断使能寄存器 (CCU_IE) 2365
56.5.7 CCU 寄存器映射和复位值表 2366

57 USB on-the-go 高速(OTG_HS) 2367

57.1 前言 2367

57.2 OTG 主要特性 2368

57.2.1 通用特性 2368

57.2.2 主机模式特性 2369

57.2.3 从机模式特性 2369

57.3 OTG 实现 2370

57.4 OTG 功能说明 2370

57.4.1 OTG 框图 2370

57.4.2 USB OTG 引脚和内核信号 2372

57.4.3 OTG 模块 2373

57.4.4 嵌入式全速 OTG PHY 2373

57.4.5 高速 OTG PHY 2373

57.4.6 使用 I2C 接口的外部全速 OTG PHY 2373

57.5 OTG 双角色设备 (DRD) 2374

57.5.1 ID 线检测 2374

57.5.2 HNP 双角色设备 2374

57.5.3 SRP 双角色设备 2374

57.6 USB 设备 2374

57.6.1 支持 SRP 功能的 USB 设备 2375

57.6.2 USB 设备状态 2375

57.6.3 USB 设备端点 2376

57.7 USB 主机 2378

57.7.1 支持 SRP 功能的 USB 主机 2378

57.7.2 USB 主机状态 2378

57.7.3 主机通道 2379

57.7.4 主机调度器 2381

57.8 SOF 触发 2381

57.8.1 主机 SOF 2382

57.8.2 设备 SOF 2382

57.9 电源选项 2382

57.10 动态更新 OTG_HFIR 寄存器 2383

57.11 USB 数据 FIFO 2383

57.11.1 设备 FIFO 架构 2384

57.11.2 主机 FIFO 架构 2385

57.11.3 FIFO RAM 分配 2386

57.12 OTG_HS 中断 2388
57.13 OTG_HS 控制和状态寄存器 2389
57.13.1 CSR 存储器映射 2389
57.14 OTG_HS 寄存器 2394
57.14.1 OTG 控制和状态寄存器 (OTG_GOTGCTL) 2394
57.14.2 OTG 中断寄存器 (OTG_GOTGINT) 2397
57.14.3 OTG AHB 配置寄存器 (OTG_GAHBCFG) 2398
57.14.4 OTG USB 配置寄存器 (OTG_GUSBCFG) 2399
57.14.5 OTG 复位寄存器 (OTG_GRSTCTL) 2402
57.14.6 OTG 模块中断寄存器 (OTG_GINTSTS) 2404
57.14.7 OTG 中断屏蔽寄存器 (OTG_GINTMSK) 2408
57.14.8 OTG_FS 接收状态调试读取/OTG 状态读取和出栈寄存器 (OTG_GRXSTSR/OTG_GRXSTSP) 2411
57.14.9 OTG 接收 FIFO 大小寄存器 (OTG_GRXFSIZ) 2412
57.14.10 OTG 主机非周期性发送 FIFO 大小寄存器 (OTG_HNPTXFSIZ)/端点 0 发送 FIFO 大小 (OTG_DIEPTXF0) 2413
57.14.11 OTG 非周期性发送 FIFO/队列状态寄存器 (OTG_HNPTXSTS) 2414
57.14.12 OTG I^2C 访问寄存器 (OTG_GI2CCTL) 2415
57.14.13 OTG 通用模块配置寄存器 (OTG_GCCFG) 2416
57.14.14 OTG 模块 ID 寄存器 (OTG_CID) 2417
57.14.15 OTG 模块 LPM 配置寄存器 (OTG_GLPMCFG) 2418
57.14.16 OTG 主机周期性发送 FIFO 大小寄存器 (OTG_HPTXFSIZ) 2422
57.14.17 OTG 设备 IN 端点发送 FIFO 大小寄存器 (OTG_DIEPTXFx) （x = 1..8，其中 x 为 FIFO 编号） 2422
57.14.18 主机模式寄存器 2423
57.14.19 OTG 主机配置寄存器 (OTG_HCFG) 2423
57.14.20 OTG 主机帧间隔寄存器 (OTG_HFIR) 2424
57.14.21 OTG 主机帧编号/帧剩余时间寄存器 (OTG_HFNUM) 2424
57.14.22 OTG_Host 周期性发送 FIFO/队列状态寄存器 (OTG_HPTXSTS) . . . 2425
57.14.23 OTG 主机全体通道中断寄存器 (OTG_HAINT) 2426
57.14.24 OTG 主机全体通道中断屏蔽寄存器 (OTG_HAINTMSK) 2426
57.14.25 OTG 主机端口控制和状态寄存器 (OTG_HPRT) 2427
57.14.26 OTG 主机通道 x 特性寄存器 (OTG_HCCHARx) （x = 0..15，其中 x = 通道编号） 2429
57.14.27 OTG 主机通道 x 分离控制寄存器 (OTG_HCSPLTx) （x = 0..15，其中 x = 通道编号） 2430
57.14.28 OTG 主机通道 x 中断寄存器 (OTG_HCINTx) （x = 0..15，其中 x = 通道编号） 2431

57.14.29 OTG 主机通道 x 中断屏蔽寄存器 (OTG_HCINTMSKx)
（x = 0..15，其中 x = 通道编号） 2432
57.14.30 OTG 主机通道 x 传输大小寄存器 (OTG_HCTSIZx)
（x = 0..15，其中 x = 通道编号） 2433
57.14.31 OTG 主机通道 x DMA 地址寄存器 (OTG_HCDMAx)
（x = 0..15，其中 x = 通道编号） 2434
57.14.32 设备模式寄存器 2434
57.14.33 OTG 设备配置寄存器 (OTG_DCFG) 2435
57.14.34 OTG 设备控制寄存器 (OTG_DCTL) 2436
57.14.35 OTG 设备状态寄存器 (OTG_DSTS) 2438
57.14.36 OTG 设备 IN 端点通用中断屏蔽寄存器 (OTG_DIEPMSK) 2439
57.14.37 OTG 设备 OUT 端点通用中断屏蔽寄存器 (OTG_DOEPMSK) 2440
57.14.38 OTG 设备全体端点中断寄存器 (OTG_HAINT) 2441
57.14.39 OTG 全体端点中断屏蔽寄存器 (OTG_DAINTMSK) 2442
57.14.40 OTG 设备 V_{BUS} 放电时间寄存器 (OTG_DVBUSDIS) 2443
57.14.41 OTG 设备 V_{BUS} 脉冲时间寄存器 (OTG_DVBUSPULSE) 2443
57.14.42 OTG 设备阈值控制寄存器 (OTG_DTHRCTL) 2444
57.14.43 OTG 设备 IN 端点 FIFO 空中断屏蔽寄存器 (OTG_DIEPEMPMSK) . . . 2445
57.14.44 OTG 设备单个端点中断寄存器 (OTG_DEACHINT) 2445
57.14.45 OTG 设备单个端点中断屏蔽寄存器 (OTG_DEACHINTMSK) 2446
57.14.46 OTG 设备端点 x 控制寄存器 (OTG_DIEPCTLx)
（x = 0..8，其中 x = 端点编号） 2446
57.14.47 OTG 设备控制 OUT 端点 0 控制寄存器 (OTG_DOEPCTL0) 2448
57.14.48 OTG 设备端点 x 控制寄存器 (OTG_DOEPCTLx)
（x = 1..8，其中 x = 端点编号） 2450
57.14.49 OTG 设备端点 x 中断寄存器 (OTG_DIEPINTx)
（x = 0..8，其中 x = 端点编号） 2452
57.14.50 OTG 设备端点 x 中断寄存器 (OTG_DIEPINTx)
（x = 0..8，其中 x = 端点编号） 2454
57.14.51 OTG 设备 IN 端点 0 传输大小寄存器 (OTG_DIEPTSIZ0) 2455
57.14.52 OTG 设备通道 x DMA 地址寄存器 (OTG_DIEPDMAx)
（x = 0..15，其中 x= 通道编号） 2455
57.14.53 OTG 设备 OUT 端点 0 传输大小寄存器 (OTG_DOEPTSIZ0) 2456
57.14.54 OTG 设备通道 x DMA 地址寄存器 (OTG_DOEPDMAx)
（x = 0..15，其中 x= 通道编号） 2457
57.14.55 OTG 设备 IN 端点 x 传输大小寄存器 (OTG_DIEPTSIZx)
（x = 1..8，其中 x= 端点编号） 2457
57.14.56 OTG 设备 IN 端点发送 FIFO 状态寄存器 (OTG_DTXFSTSx)
（x = 0..8，其中 x = 端点编号） 2458
57.14.57 OTG 设备 OUT 端点 x 传输大小寄存器 (OTG_DOEPTSIZx)
（x = 1..8，其中 x = 端点编号） 2458

57.14.58 OTG 电源和时钟门控控制寄存器 (OTG_PCGCCTL) 2459
57.14.59 OTG_HS 寄存器映射 2460
57.15 OTG_HS 编程模型 2470
57.15.1 模块初始化 2470
57.15.2 主机初始化 2471
57.15.3 设备初始化 2471
57.15.4 DMA 模式 2472
57.15.5 主机编程模型 2472
57.15.6 设备编程模型 2502
57.15.7 最坏情况下的响应时间 2520
57.15.8 OTG 编程模型 2521

58 以太网 (ETH)：通过 DMA 控制器进行介质访问控制 (MAC) 2527
58.1 以太网简介 2527
58.2 以太网主要特性 2527
58.2.1 MAC 内核特性 2527
58.2.2 DMA 特性 2529
58.2.3 总线接口功能 2529
58.3 以太网引脚和内部信号 2530
58.4 以太网架构 2532
58.4.1 DMA 控制器 2533
58.4.2 MTL 2540
58.4.3 MAC 2540
58.5 以太网功能说明：MAC 2544
58.5.1 双 VLAN 处理 2544
58.5.2 源地址和 VLAN 插入、替换或删除 2545
58.5.3 数据包过滤 2546
58.5.4 IEEE 1588 时间戳 2553
58.5.5 IPv4 ARP 减荷 2564
58.5.6 TCP 分段减荷 2564
58.5.7 回送 2568
58.5.8 流控制 2568
58.5.9 校验和减荷引擎 2570
58.5.10 MAC 管理计数器 2573
58.5.11 MAC 生成的中断 2574
58.5.12 MAC 和 MMC 寄存器说明 2574

58.6 以太网功能说明：PHY 接口 2574
58.6.1 站管理代理 (SMA) 2575
58.6.2 介质独立接口 (MII) 2578
58.6.3 精简介质独立接口 (RMII) 2579
58.7 以太网低功耗模式 2582
58.7.1 节能型以太网 2582
58.7.2 电源管理 2583
58.7.3 掉电和唤醒序列 2585
58.8 以太网中断 2585
58.8.1 DMA 中断 2585
58.8.2 MTL 中断 2587
58.8.3 MAC 中断 2587
58.9 以太网编程模型 2588
58.9.1 DMA 初始化 2588
58.9.2 MTL 初始化 2589
58.9.3 MAC 初始化 2589
58.9.4 执行正常接收和发送操作 2590
58.9.5 停止和开始发送 2590
58.9.6 关于 MII 链路状态转换的编程指南 2591
58.9.7 关于 IEEE 1588 时间戳的编程指南 2592
58.9.8 关于节能型以太网 (EEE) 的编程指南 2593
58.9.9 关于每秒脉冲数 (PPS) 灵活的输出的编程指南 2594
58.9.10 关于 TSO 的编程指南 2595
58.9.11 关于在接收端执行 VLAN 过滤的编程指南 2596
58.10 描述符 2596
58.10.1 描述符概述 2596
58.10.2 描述符结构 2597
58.10.3 发送描述符 2599
58.10.4 接收描述符 2610
58.11 以太网 MAC 寄存器 2620
58.11.1 以太网寄存器映射 2620
58.11.2 以太网 DMA 寄存器 2620
58.11.3 以太网 MTL 寄存器 2646
58.11.4 以太网 MAC 和 MMC 寄存器 2658

59 HDMI-CEC 控制器 (HDMI-CEC) 2751

59.1 简介 2751

59.2 HDMI-CEC 控制器主要特性 2751

59.3 HDMI-CEC 功能说明 2752

59.3.1 HDMI-CEC 引脚和内核信号 2752

59.3.2 HDMI-CEC 框图 2752

59.3.3 消息说明 2753

59.3.4 位时序 2754

59.4 仲裁 2755

59.4.1 SFT 选项位 2756

59.5 错误处理 2757

59.5.1 位错误 2757

59.5.2 消息错误 2757

59.5.3 位上升错误 (BRE) 2757

59.5.4 短位周期错误 (SBPE) 2757

59.5.5 长位周期错误 (LBPE) 2758

59.5.6 发送错误检测 (TXERR) 2760

59.6 HDMI-CEC 中断 2761

59.7 HDMI-CEC 寄存器 2762

59.7.1 CEC 控制寄存器 (CEC_CR) 2762

59.7.2 CEC 配置寄存器 (CEC_CFGR) 2763

59.7.3 CEC 发送数据寄存器 (CEC_TXDR) 2765

59.7.4 CEC 接收数据寄存器 (CEC_RXDR) 2765

59.7.5 CEC 中断和状态寄存器 (CEC_ISR) 2765

59.7.6 CEC 中断使能寄存器 (CEC_IER) 2767

59.7.7 HDMI-CEC 寄存器映射 2769

60 调试基础结构 2770

60.1 前言 2770

60.2 调试基本接口特性 2771

60.3 调试基础结构功能描述 2771

60.3.1 调试基础结构框图 2771

60.3.2 调试基础结构引脚和内部信号 2772

60.3.3 调试基础结构电源、时钟和复位 2773

60.4 调试访问端口功能描述 2775

60.4.1 串行线和 JTAG 调试端口 (SWJ-DP) 2775

60.4.2 访问端口 2788

60.5 跟踪和调试子系统功能描述 2794
60.5.1 系统 ROM 表 2794
60.5.2 全局时间戳发生器 (TSG) 2802
60.5.3 交叉触发接口 (CTI) 和矩阵 (CTM) 2811
60.5.4 跟踪聚合器 (CSTF) 2831
60.5.5 嵌入式跟踪 FIFO (ETF) 2843
60.5.6 跟踪端口接口单元 (TPIU) 2865
60.5.7 串行线输出 (SWO) 和 SWO 跟踪聚合器 (SWTF) 2883
60.5.8 微控制器调试单元 (DBGMCU) 2906
60.6 Cortex-M7 调试功能描述 2914
60.6.1 Cortex-M7 ROM 表 2914
60.6.2 Cortex-M7 数据观察点和跟踪单元 (DWT) 2927
60.6.3 Cortex-M7 指令跟踪宏单元 (ITM) 2942
60.6.4 Cortex-M7 断点单元 (FPB) 2951
60.6.5 Cortex-M7 嵌入式跟踪宏单元 (ETM) 2958
60.6.6 Cortex-M7 交叉触发接口 (CTI) 2993
60.7 针对调试基础结构的参考文献 2993

61 设备电子签名 2994
61.1 唯一设备 ID 寄存器（96 位） 2994
61.2 Flash 大小 2995
61.3 封装数据寄存器 2995

版本历史 2996

表格索引

表 1. 总线主设备与总线从设备互连 . . . 95
表 2. 寄存器边界地址 . . . 100
表 3. 自举模式 . . . 105
表 4. 连接到封装引脚或焊球的 FLASH 输入 / 输出信号 . . . 108
表 5. FLASH 内部输入 / 输出信号 . . . 108
表 6. Flash 模块 - 1 MB 双存储区构成 . . . 109
表 7. 根据总线频率 (ACLK) 和 VCORE 范围确定的等待状态数 . . . 112
表 8. 并行位数参数 . . . 116
表 9. 编程速度 . . . 116
表 10. Flash 用户选项字节列表 . . . 123
表 11. RDP 值与读取保护级别 . . . 126
表 12. 允许的访问与读取保护级别 . . . 126
表 13. AXI 接口存储器映射 SWAP_BANK = “0” . . . 129
表 15. 存储区交换步骤 . . . 130
表 14. AXI 接口存储器映射 SWAP_BANK = “1” . . . 130
表 16. FLASH 寄存器映射和复位值 . . . 172
表 17. 首选术语列表 . . . 177
表 18. Flash 保护机制 . . . 178
表 19. 安全用户软件选择用例 . . . 184
表 20. Flash 保护区域访问权限汇总 . . . 185
表 21. ASIB 配置 . . . 187
表 22. AMIB 配置 . . . 188
表 23. AXI 寄存器映射和复位值 . . . 198
表 24. 连接到封装引脚或焊球的 PWR 输入 / 输出信号 . . . 208
表 25. PWR 内部输入 / 输出信号 . . . 208
表 26. 电源配置控制 . . . 211
表 27. 低功耗模式汇总 . . . 226
表 28. PDDS_Dn 低功耗模式控制 . . . 228
表 29. 低功耗退出模式标志 . . . 230
表 30. CSleep 模式 . . . 239
表 31. CStop 模式 . . . 240
表 32. DStop 模式概述 . . . 240
表 33. DStop 模式 . . . 241
表 34. 停止模式下的运行 . . . 242
表 35. 停止模式 . . . 243
表 36. DStandby 模式 . . . 244
表 37. 待机和停止标志 . . . 246
表 38. 待机模式 . . . 246
表 39. 电源控制寄存器映射和复位值 . . . 256
表 40. BDMA 和 DMAMUX2 初始化序列 (DMAMUX2_INIT) . . . 263
表 41. LPUART1 初始编程 (LPUART1_INIT) . . . 265
表 42. LPUART1 初始编程 (LPUART1_Start) . . . 265
表 43. 连接到封装引脚或焊球的 RCC 输入 / 输出信号 . . . 268
表 44. RCC 内部输入 / 输出信号 . . . 269
表 45. 复位分配汇总 . . . 272
表 46. 复位源标识 (RCC_RSR) . . . 274
表 47. 时钟定时器与 pclk 之比 . . . 290
表 48. STOPWUCK 和 STOPKERWUCK 说明 . . . 291

表 49. HSIKERON 和 CSIKERON 行为 . . . 292
表 50. 内核时钟分配概述 . . . 293
表 51. 系统状态概述 . . . 306
表 52. D1 和 D2 外设的外设时钟使能 . . . 309
表 53. D3 外设的外设时钟使能 . . . 310
表 54. 中断源和控制 . . . 313
表 55. RCC_RSR 偏移地址和复位值 . . . 373
表 56. RCC_AHB3ENR 偏移地址和复位值 . . . 375
表 57. RCC_AHB1ENR 偏移地址和复位值 . . . 376
表 58. RCC_AHB2ENR 偏移地址和复位值 . . . 378
表 59. RCC_AHB4ENR 偏移地址和复位值 . . . 380
表 60. RCC_APB3ENR 偏移地址和复位值 . . . 382
表 61. RCC_APB1ENR 偏移地址和复位值 . . . 383
表 62. RCC_APB1ENR 偏移地址和复位值 . . . 386
表 63. RCC_APB2ENR 偏移地址和复位值 . . . 387
表 64. RCC_APB4ENR 偏移地址和复位值 . . . 390
表 65. RCC_AHB3LPENR 偏移地址和复位值 . . . 392
表 66. RCC_AHB1LPENR 偏移地址和复位值 . . . 394
表 67. RCC_AHB2LPENR 偏移地址和复位值 . . . 396
表 68. RCC_AHB4LPENR 偏移地址和复位值 . . . 398
表 69. RCC_APB3LPENR 偏移地址和复位值 . . . 400
表 70. RCC_APB1LLPENR 偏移地址和复位值 . . . 401
表 71. RCC_APB1HLPENR 偏移地址和复位值 . . . 405
表 72. RCC_APB2LPENR 偏移地址和复位值 . . . 406
表 73. RCC_APB4LPENR 偏移地址和复位值 . . . 409
表 74. RCC 寄存器映射和复位值 . . . 412
表 75. CRS 内部输入 / 输出信号 . . . 423
表 76. 低功耗模式对 CRS 的作用 . . . 426
表 77. 中断控制位 . . . 426
表 78. CRS 寄存器映射和复位值 . . . 432
表 79. HSEM 内部输入 / 输出信号 . . . 434
表 80. 经授权的 AHB 总线主控 ID . . . 439
表 81. HSEM 寄存器映射和复位值 . . . 444
表 82. 端口位配置表 . . . 447
表 83. GPIO 寄存器映射和复位值 . . . 461
表 84. SYSCFG 寄存器映射和复位值 . . . 482
表 85. 外设互连矩阵 （D2 域） . . . 486
表 86. 外设互连矩阵 （D3 域） . . . 487
表 87. 外设互连矩阵详细信息 . . . 488
表 88. EXTI 唤醒输入 . . . 505
表 89. EXTI 待处理请求清零输入 . . . 508
表 90. MDMA . . . 509
表 91. DMAMUX1、 DMA1 和 DMA2 连接 . . . 510
表 92. DMAMUX2 和 BDMA 连接 . . . 515
表 93. MDMA 内部输入 / 输出信号 . . . 519
表 94. MDMA 中断请求 . . . 524
表 95. MDMA 寄存器映射和复位值 . . . 541
表 96. DMA 内部输入 / 输出信号 . . . 544
表 97. 源和目标地址 . . . 546
表 98. 双缓冲区模式下的源和目标地址寄存器 (DBM=1) . . . 551
表 99. 封装 / 解封和字节序行为 （位 PINC = MINC = 1） . . . 552
表 100. PSIZE 与 MSIZE 确定时对 NDT 的限制条件 . . . 552

表 101. FIFO 阈值配置 . . . 555
表 102. 可能的 DMA 配置 . . . 559
表 103. DMA 中断请求 . . . 561
表 104. DMA 寄存器映射和复位值 . . . 572
表 105. 可编程的数据宽度、字节存储次序 （位 PINC = MINC = 1 时） . . . 579
表 106. BDMA 中断请求 . . . 580
表 107. BDMA 寄存器映射和复位值 . . . 587
表 108. DMAMUX1 和 DMAMUX2 实例化 . . . 591
表 109. DMAMUX1：复用器输入到资源的分配 . . . 591
表 110. DMAMUX1：触发输入到资源的分配 . . . 592
表 111. DMAMUX1：同步输入到资源的分配 . . . 593
表 112. DMAMUX2：复用器输入到资源的分配 . . . 593
表 113. DMAMUX2：触发输入到资源的分配 . . . 593
表 114. DMAMUX2：同步输入到资源的分配 . . . 594
表 115. DMAMUX 信号 . . . 596
表 116. DMAMUX 中断 . . . 599
表 117. DMAMUX 寄存器映射和复位值 . . . 608
表 118. DMA2D 内部输入 / 输出信号 . . . 612
表 119. 输入时支持的颜色模式 . . . 613
表 120. 存储器中的数据顺序 . . . 614
表 121. Alpha 模式配置 . . . 614
表 122. 支持的 CLUT 颜色模式 . . . 615
表 123. 存储器中的 CLUT 数据顺序 . . . 615
表 124. 输出时支持的颜色模式 . . . 616
表 125. 存储器中的数据顺序 . . . 617
表 126. 存储器中的 MCU 顺序 . . . 620
表 127. DMA2D 中断请求 . . . 622
表 128. DMA2D 寄存器映射和复位值 . . . 640
表 129. NVIC . . . 642
表 130. EXTI 事件输入配置和寄存器控制 . . . 652
表 131. 可配置事件输入异步边沿检测器复位 . . . 654
表 132. EXTI 事件输入映射 . . . 657
表 133. 屏蔽功能 . . . 660
表 134. 异步中断 / 事件控制器寄存器映射和复位值 . . . 680
表 135. CRC 内部输入 / 输出信号 . . . 684
表 136. CRC 寄存器映射和复位值 . . . 689
表 137. FMC 引脚 . . . 692
表 138. FMC 存储区域映射选项 . . . 695
表 139. NOR/PSRAM 存储区域选择 . . . 696
表 140. NOR/PSRAM 外部存储器地址 . . . 696
表 141. NAND 存储映射和时序寄存器 . . . 696
表 142. NAND 存储区域选择 . . . 697
表 143. SDRAM 存储区域选择 . . . 697
表 144. SDRAM 地址映射 . . . 697
表 145. 数据总线宽度为 8 位时的 SDRAM 地址映射 . . . 698
表 146. 数据总线宽度为 16 位时的 SDRAM 地址映射 . . . 699
表 147. 32 位数据总线宽度时的 SDRAM 地址映射 . . . 700
表 148. NOR/PSRAM 的可编程访问参数 . . . 701
表 149. 非复用 I/O NOR Flash . . . 702
表 150. 16 位复用 I/O NOR Flash . . . 702
表 151. 非复用 I/O PSRAM/SRAM . . . 703
表 152. 16 位复用 I/O PSRAM . . . 703

表 153. NOR Flash/PSRAM：支持的存储器和事务示例 704
表 154. FMC_BCRx 位域 707
表 155. FMC_BTRx 位域 708
表 156. FMC_BCRx 位域 709
表 157. FMC_BTRx 位域 710
表 158. FMC_BWTRx 位域 710
表 159. FMC_BCRx 位域 712
表 160. FMC_BTRx 位域 713
表 161. FMC_BWTRx 位域 713
表 162. FMC_BCRx 位域 715
表 163. FMC_BTRx 位域 716
表 164. FMC_BWTRx 位域 716
表 165. FMC_BCRx 位域 718
表 166. FMC_BTRx 位域 719
表 167. FMC_BWTRx 位域 719
表 168. FMC_BCRx 位域 721
表 169. FMC_BTRx 位域 721
表 170. FMC_BCRx 位域 727
表 171. FMC_BTRx 位域 727
表 172. FMC_BCRx 位域 728
表 173. FMC_BTRx 位域 729
表 174. 可编程的 NAND Flash 访问参数 738
表 175. 8 位 NAND Flash 738
表 176. 16 位 NAND Flash 739
表 177. 支持的存储器和事务 739
表 178. ECC 结果相关位 748
表 179. SDRAM 信号 749
表 180. FMC 寄存器映射 766
表 181. QUADSPI 内部信号 769
表 182. QUADSPI 引脚 770
表 183. QUADSPI 中断请求 782
表 184. QUADSPI 寄存器映射和复位值 795
表 185. DLYB 内部输入 / 输出信号 797
表 186. 延迟模块控制 797
表 187. DLYB 寄存器映射和复位值 800
表 188. ADC 内部输入 / 输出信号 804
表 189. ADC 输入 / 输出引脚 804
表 190. 为常规外部触发配置触发极性 823
表 191. 为注入外部触发配置触发极性 824
表 192. ADC1、 ADC2 和 ADC3 - 常规通道的外部触发 825
表 193. ADC1、 ADC2 和 ADC3 - 注入通道的外部触发 826
表 194. TSAR 与分辨率的对应关系 837
表 195. 偏移计算与数据分辨率 840
表 196. 16 位数据格式 843
表 197. 16 位格式的数字示例 （粗体代表饱和） 843
表 198. 模拟看门狗通道选择 851
表 199. 模拟看门狗 1、 2、 3 比较 852
表 200. 过采样器工作模式汇总 859
表 201. 每个 ADC 的 ADC 中断 876
表 202. DELAY 位与 ADC 分辨率的关系 912
表 203. ADC 全局寄存器映射 914

表 204. 每个 ADC 的 ADC 寄存器映射和复位值 （主 ADC 偏移 =0x000，从 ADC 偏移 =0x100） 914
表 205. ADC 寄存器映射和复位值 （主 ADC 和从 ADC 通用寄存器，偏移 =0x300） 917
表 206. DAC 输入 / 输出引脚 920
表 207. DAC 内部输入 / 输出信号 920
表 208. DAC 触发选择 923
表 209. 采样时间和刷新时间 927
表 210. 通道输出模式汇总 928
表 211. DAC 低功耗模式的作用 933
表 212. DAC 中断 934
表 213. DAC 寄存器映射和复位值 949
表 214. VREF 缓冲器模式 951
表 215. VREFBUF 寄存器映射和复位值 953
表 216. COMP 输入 / 输出内部信号 956
表 217. COMP 输入 / 输出引脚 956
表 218. COMP1_OUT 的 GPIO 分配 959
表 219. COMP2_OUT 的 GPIO 分配 959
表 220. 低功耗模式下的比较器行为 960
表 221. 中断控制位 961
表 222. 中断控制位 961
表 223. COMP 寄存器映射和复位值 970
表 224. 运算放大器连接情形 972
表 225. 工作模式和校准 979
表 226. 低功耗模式对 OPAMP 的影响 980
表 227. OPAMP 寄存器映射和复位值 988
表 228. DFSDM1 实现 991
表 229. DFSDM 外部引脚 993
表 230. DFSDM 内部信号 993
表 231. DFSDM 触发器连接 993
表 232. DFSDM 断路连接 994
表 233. 滤波器最大输出分辨率 （来自滤波器输出的峰值数据值），基于某些 FOSR 值 1008
表 234. 积分器最大输出分辨率 （来自积分器输出的峰值数据值），基于某些 IOSR 值， FOSR = 256 以及 Sinc3 滤波器类型 （最大数据） 1008
表 235. DFSDM 中断请求 1016
表 236. DFSDM 寄存器映射和复位值 1039
表 237. DCMI 内部信号 1050
表 238. DCMI 外部信号 1050
表 239. 捕获的数据字节在 32 位字 （宽 8 位）中的位置排布 1051
表 240. 捕获的数据字节在 32 位字 （宽 10 位）中的位置排布 1051
表 241. 捕获的数据字节在 32 位字 （宽 12 位）中的位置排布 1052
表 242. 捕获的数据字节在 32 位字 （宽 14 位）中的位置排布 1052
表 243. 单色逐行视频格式的数据存储 1058
表 244. 以 RGB 逐行视频格式存储数据 1058
表 245. YCbCr 逐行视频格式下的数据存储 1059
表 246. YCbCr 逐行视频格式下的数据存储 ——Y 分量提取模式 1059
表 247. DCMI 中断 1060
表 248. DCMI 寄存器地址映射和复位值 1072
表 249. LCD-TFT 内部信号 1074
表 250. LCD-TFT 引脚和信号接口 1075
表 251. 各个寄存器的时钟域 1075
表 252. LTDC 寄存器访问和更新持续时间 1076
表 253. 像素数据映射与颜色格式的关系 1080

表 254. LTDC 中断请求 . . . 1083
表 255. LTDC 寄存器映射和复位值 . . . 1104
表 256. JPEG 内部信号 . . . 1109
表 257. JPEG 编解码器中断请求 . . . 1112
表 258. JPEG 编解码器寄存器映射和复位值 . . . 1120
表 259. RNG 内部输入 / 输出信号 . . . 1123
表 260. RNG 中断请求 . . . 1129
表 261. RNG 寄存器映射和复位映射 . . . 1133
表 262. CRYP 内部输入 / 输出信号 . . . 1136
表 263. 计数器模式初始化向量 . . . 1163
表 264. GCM 最后一个块定义 . . . 1165
表 265. GCM 模式 IV 寄存器初始化 . . . 1165
表 266. CCM 模式 IV 寄存器初始化 . . . 1170
表 267. DES/TDES 数据交换功能 . . . 1175
表 268. AES 数据交换功能 . . . 1177
表 269. 密钥寄存器 CRYP_KxR/LR 字节序 （TDES K1/2/3 和 AES 128/192/256 位密钥） . . . 1177
表 270. 初始化向量寄存器 CRYP_IVxR 字节序 . . . 1178
表 271. 针对存储器到外设的 DMA 传输的加密处理器配置 . . . 1179
表 272. 针对外设到存储器的 DMA 传输的加密处理器配置 . . . 1180
表 273. CRYP 中断请求 . . . 1181
表 274. ECB、 CBC 和 CTR 模式下每个 128 位块的处理时间 （以时钟周期计） . . . 1182
表 275. GCM 和 CCM 模式下每个 128 位块的处理时间 （以时钟周期计） . . . 1182
表 276. CRYP 寄存器映射和复位值 . . . 1196
表 277. HASH 内部输入 / 输出信号 . . . 1200
表 278. 散列处理器输出 . . . 1204
表 279. HASH 中断请求 . . . 1209
表 280. 处理时间 （时钟周期数） . . . 1210
表 281. HASH 寄存器映射和复位值 . . . 1220
表 282. HRTIM 输入 / 输出汇总 . . . 1223
表 283. 定时器分辨率和最小 PWM 频率 （f_{HRTIM} = 400 MHz 时） . . . 1225
表 284. 周期和比较寄存器最小值和最大值 . . . 1227
表 285. 定时器工作模式 . . . 1228
表 286. 定时器 A 到 E 之间的事件映射 . . . 1233
表 287. 死区分辨率和最大绝对值 . . . 1241
表 288. 外部事件映射和关联特性 . . . 1247
表 289. 输出置位 / 复位延迟和抖动与外部事件工作模式的关系 . . . 1248
表 290. 每个定时器的过滤信号映射 . . . 1251
表 291. 每个定时器的窗口信号映射 (EEFLTR[3:0] = 1111) . . . 1253
表 292. HRTIM 可预装载控制寄存器和关联的更新源 . . . 1261
表 293. 更新使能输入和源 . . . 1262
表 294. 主定时器更新事件传播 . . . 1264
表 295. TIMx 更新事件传播 . . . 1264
表 296. 能够生成更新事件的复位事件 . . . 1265
表 297. 用于定时器复位的更新事件传播 . . . 1266
表 298. 输出状态编程， x= A..E， y = 1 或 2 . . . 1267
表 299. 突发模式的定时器输出编程 . . . 1270
表 300. 来自通用定时器的突发模式时钟源 . . . 1271
表 301. 故障输入 . . . 1279
表 302. 采样速率和滤波器长度与 FLTFxF[3:0] 和时钟设置的关系 . . . 1280
表 303. 同步事件的作用与定时器工作模式之间的关系 . . . 1285
表 304. HRTIM 中断汇总 . . . 1290
表 305. HRTIM DMA 请求汇总 . . . 1291

表 306. RTIM 全局寄存器映射 1380
表 307. HRTIM 寄存器映射和复位值：主定时器 1380
表 308. HRTIM 寄存器映射和复位值：TIMx (x= A..E) 1382
表 309. HRTIM 寄存器映射和复位值：常用功能 1386
表 310. 定时器输出行为与 BRK/BRK2 输入 1430
表 311. 计数方向与编码器信号的关系 1437
表 312. TIMx 内部触发连接 1453
表 313. 具有断路功能的互补通道 OCx 和 OCxN 的输出控制位 1465
表 314. TIM1 寄存器映射和复位值 1484
表 315. TIM8 寄存器映射和复位值 1487
表 316. 计数方向与编码器信号的关系 1521
表 317. TIMx 内部触发连接 1538
表 318. 标准 OCx 通道的输出控制位 1548
表 319. TIM2/TIM3/TIM4/TIM5 寄存器映射和复位值 1558
表 320. TIMx 内部触发连接 1586
表 321. 标准 OCx 通道的输出控制位 1594
表 322. TIM12 寄存器映射和复位值 1597
表 323. 标准 OCx 通道的输出控制位 1605
表 324. TIM13/TIM14 寄存器映射和复位值 1608
表 325. TIMx 内部触发连接 1648
表 326. 具有断路功能的互补通道 OCx 和 OCxN 的输出控制位 TIM15 1657
表 327. TIM15 寄存器映射和复位值 1666
表 328. 具有断路功能的互补通道 OCx 和 OCxN 的输出控制位 (TIM16/17) 1677
表 329. TIM16/TIM17 寄存器映射和复位值 1687
表 330. TIM6/TIM7 寄存器映射和复位值 1701
表 331. STM32H7x3 LPTIM 特性 1702
表 332. LPTIM 输入 / 输出引脚 1705
表 333. LPTIM 内部信号 1705
表 334. LPTIM1 外部触发连接 1705
表 335. LPTIM2 外部触发连接 1706
表 336. LPTIM3 外部触发连接 1706
表 337. LPTIM4 外部触发连接 1706
表 338. LPTIM5 外部触发连接 1707
表 339. LPTIM1 输入 1 连接 1707
表 340. LPTIM1 输入 2 连接 1707
表 341. LPTIM2 输入 1 连接 1707
表 342. LPTIM2 输入 2 连接 1708
表 343. LPTIM3 输入 1 连接 1708
表 344. 预分频器的分频比 1709
表 345. 编码器计数方案 1715
表 346. 中断事件 1716
表 347. LPTIM 寄存器映射和复位值 1727
表 348. WWDG 内部输入 / 输出信号 1730
表 349. WWDG 寄存器映射和复位值 1735
表 350. IWDG 内部输入 / 输出信号 1737
表 351. IWDG 寄存器映射和复位值 1743
表 352. RTC 引脚和内部信号 1748
表 353. RTC 引脚 PC13 配置 1748
表 354. RTC_OUT 映射 1749
表 355. 各个模式下的 RTC 功能 1749
表 356. 低功耗模式对 RTC 的作用 1760
表 357. 中断控制位 1760

表 358. RTC 寄存器映射和复位值 . . . 1783
表 359. STM32H7x3 I2C 特性实现 . . . 1786
表 360. 模拟滤波器与数字滤波器对比 . . . 1789
表 361. I2C-SMBUS 规范数据建立和保持时间 . . . 1792
表 362. I2C 配置表 . . . 1796
表 363. I2C-SMBUS 规范时钟时序 . . . 1806
表 364. fI2CCLK = 8 MHz 时的时序设置示例 . . . 1816
表 365. fI2CCLK = 16 MHz 时的时序设置示例 . . . 1816
表 366. fI2CCLK = 48 MHz 时的时序设置示例 . . . 1817
表 367. SMBus 超时规范 . . . 1819
表 368. 带 PEC 的 SMBUS 配置 . . . 1820
表 369. 不同 i2c_ker_ck 频率下的 TIMEOUTA 设置示例 （最大 $t_{TIMEOUT}$ = 25 ms） . . . 1822
表 370. 不同 i2c_ker_ck 频率下的 TIMEOUTB 设置示例 . . . 1822
表 371. 不同 i2c_ker_ck 频率下的 TIMEOUTA 设置示例 （最大 t_{IDLE} = 50 µs） . . . 1822
表 372. 低功耗模式 . . . 1832
表 373. I2C 中断请求 . . . 1832
表 374. I2C 寄存器映射和复位值 . . . 1848
表 375. USART/LPUART 功能 . . . 1851
表 376. 通过采样数据进行噪声检测 . . . 1864
表 377. BRR [3:0] = 0000 时的 USART 接收器容差 . . . 1867
表 378. BRR[3:0] 不等于 0000 时的 USART 接收器容差 . . . 1867
表 379. USART 帧格式 . . . 1871
表 380. USART 中断请求 . . . 1891
表 381. USART 寄存器映射和复位值 . . . 1917
表 382. lpuart_ker_ck_pres = 32,768 KHz 时编程的波特率的误差计算 . . . 1931
表 383. 采用 16 倍过采样 (OVER8 = 0) 时，在 fCK = 100 MHz 下 . . . 1932
表 384. LPUART 接收器的容差 . . . 1933
表 386. LPUART 中断请求 . . . 1943
表 387. LPUART 寄存器映射和复位值 . . . 1961
表 388. STM32F7xx SPI 特性 . . . 1963
表 389. SPI 唤醒和中断请求 . . . 1989
表 390. PCM/I2S 模式下可用的位域 . . . 1991
表 391. AFCNTR = 1 时在使能 SPI/I2S 之前的 WS 和 CK 电平 . . . 1998
表 392. 串行数据线交换 . . . 1999
表 393. 适合常见 I2S 频率的 CLKGEN 编程示例 . . . 2003
表 394. I2S 中断请求 . . . 2013
表 395. SPI 寄存器映射和复位值 . . . 2031
表 396. SAI 内部输入 / 输出信号 . . . 2035
表 397. SAI 输入 / 输出引脚 . . . 2035
表 398. 外部同步选择 . . . 2037
表 399. 时钟发生器编程示例 . . . 2045
表 400. TDM 设置 . . . 2052
表 401. 允许的 TDM 帧配置 . . . 2054
表 402. SOPD 模式 . . . 2057
表 403. 奇偶校验位计算 . . . 2058
表 404. 音频采样频率与符号率 . . . 2059
表 405. SAI 中断源 . . . 2067
表 406. SAI 寄存器映射和复位值 . . . 2083
表 407. SPDIFRX 内部输入 / 输出信号 . . . 2086
表 408. SPDIFRX 引脚 . . . 2086
表 409. 报头的转换序列 . . . 2092
表 410. spdifrx_ker_ck 的最小频率与音频采样率 . . . 2102

表 411. spdifrx_symb_ck 的生成条件 2104
表 412. 位域属性与 SPDIFRX 状态 2106
表 413. SPDIFRX 接口寄存器映射和复位值 2123
表 414. 连接到封装引脚或焊球的 SWPMI 输入 / 输出信号 2126
表 415. SWPMI 内部输入 / 输出信号 2127
表 416. 低功耗模式对 SWPMI 的作用 2141
表 417. 中断控制位 2142
表 418. 针对发送 / 接收选择缓冲区模式 2144
表 419. SWPMI 寄存器映射和复位值 2151
表 420. 连接到封装引脚或焊球的 MDIOS 输入 / 输出信号 2153
表 421. MDIOS 内部输入 / 输出信号 2153
表 422. 中断控制位 2157
表 423. MDIOS 寄存器映射和复位值 2164
表 424. SDMMC 工作模式 SD 和 SDIO 2168
表 425. SDMMC 工作模式 eMMC 2169
表 426. SDMMC 内部输入 / 输出信号 2170
表 427. SDMMC 引脚 2170
表 428. SDMMC 命令和数据相位选择 2171
表 429. 命令令牌格式 2177
表 430. 带 CRC 的短响应令牌格式 2177
表 431. 不带 CRC 的短响应令牌格式 2178
表 432. 带 CRC 的长响应令牌格式 2178
表 433. 特定命令概述 2179
表 434. 命令路径状态标志 2179
表 435. 命令路径错误处理 2180
表 436. 数据令牌格式 2186
表 437. 数据路径状态标志和清零位 2187
表 438. 数据路径错误处理 2188
表 439. 数据 FIFO 访问 2189
表 440. 传输 FIFO 状态标志 2189
表 441. 接收 FIFO 状态标志 2190
表 442. SDMMC 与 MDMA 的连接 2193
表 443. AHB 和 SDMMC_CK 时钟频率的关系 2193
表 444. SDIO 特殊操作控制 2194
表 445. 4 位模式启动、中断和 CRC 状态信号检测 2198
表 446. 应用案例 2202
表 447. 响应类型和 SDMMC_RESPxR 寄存器 2220
表 448. SDMMC 寄存器映射 2237
表 449. CAN 子系统 I/O 信号 2239
表 450. CAN 子系统 I/O 引脚 2239
表 451. FDCAN 中的 DLC 编码 2245
表 452. 接收缓冲区过滤器配置示例 2257
表 453. 调试消息过滤器配置示例 2257
表 454. 帧发送的可能配置 2258
表 455. 发送缓冲区 /FIFO - 队列元素大小 2259
表 456. 1 级参考消息的第一个字节 2267
表 457. 2 级参考消息的前四个字节 2267
表 458. 0 级参考消息的前四个字节 2268
表 459. TUR 配置示例 2269
表 460. 系统矩阵，节点 A 2273
表 461. 触发列表，节点 A 2273
表 462. 随参考消息一起发送的数据字节数 2280

表 463. 接收缓冲区和 FIFO 元素 . . . 2286
表 464. 接收缓冲区和 FIFO 元素说明 . . . 2287
表 465. 发送缓冲区和 FIFO 元素 . . . 2288
表 466. 发送缓冲区元素说明 . . . 2289
表 467. 发送事件 FIFO 元素 . . . 2290
表 468. 发送事件 FIFO 元素说明 . . . 2290
表 469. 标准消息 ID 过滤器元素 . . . 2291
表 470. 标准消息 ID 过滤器元素位域说明 . . . 2292
表 471. 扩展消息 ID 过滤器元素 . . . 2293
表 472. 扩展消息 ID 过滤器元素位域说明 . . . 2293
表 473. 触发存储器元素 . . . 2294
表 474. 触发存储器元素说明 . . . 2294
表 475. FDCAN 寄存器映射和复位值 . . . 2355
表 476. CCU 寄存器映射和复位值 . . . 2366
表 477. OTG_HS 支持的速度 . . . 2367
表 478. STM32H7 的 OTG 实现 . . . 2370
表 479. OTG_FS/OTG_HS 输入 / 输出信号 . . . 2372
表 480. OTG_FS/OTG_HS 输入 / 输出信号 . . . 2372
表 481. 模块全局控制和状态寄存器 (CSR) . . . 2389
表 482. 主机模式控制和状态寄存器 (CSR) . . . 2390
表 483. 设备模式控制和状态寄存器 . . . 2391
表 484. 数据 FIFO (DFIFO) 访问寄存器地址映射 . . . 2393
表 485. 电源和时钟门控控制和状态寄存器 . . . 2393
表 486. TRDT 值 . . . 2401
表 487. TRDT 值 (HS) . . . 2402
表 488. 软断开的最小时间 . . . 2438
表 489. 寄存器映射和复位值 . . . 2460
表 490. 以太网外设引脚 . . . 2530
表 491. 以太网内部输入 / 输出信号 . . . 2531
表 492. Tx 路径中的双 VLAN 处理功能 . . . 2544
表 493. Rx 路径中的双 VLAN 处理功能 . . . 2545
表 494. 基于 VLTI 位进行 VLAN 插入或替换 . . . 2546
表 495. 目标地址过滤 . . . 2549
表 496. 源地址过滤 . . . 2550
表 497. VLAN 匹配状态 . . . 2551
表 498. 普通时钟：拍摄快照的 PTP 消息 . . . 2556
表 499. 端对端透明时钟：拍摄快照的 PTP 消息 . . . 2557
表 500. 点对点透明时钟：拍摄快照的 PTP 消息 . . . 2557
表 501. PTP 消息生成条件 . . . 2563
表 502. TSO：TCP 和 IP 报头字段 . . . 2566
表 503. 暂停数据包字段 . . . 2568
表 504. Tx MAC 流控制 . . . 2569
表 505. Rx MAC 流控制 . . . 2569
表 506. 针对不同数据包类型的发送校验和减荷引擎功能 . . . 2571
表 507. 针对不同数据包类型的接收校验和减荷引擎功能 . . . 2572
表 508. MCD 时钟选择 . . . 2575
表 509. MDIO 数据包字段说明 . . . 2576
表 510. RX 接口信号编码 . . . 2579
表 511. 传送完成中断行为 . . . 2586
表 512. TDES0 正常描述符 （读取格式） . . . 2599
表 513. TDES1 正常描述符 （读取格式） . . . 2600
表 514. TDES2 正常描述符 （读取格式） . . . 2600

表 515. TDES3 正常描述符 （读取格式）. 2601
表 516. DES0 正常描述符 （回写格式）. 2603
表 517. TDES1 正常描述符 （回写格式）. 2604
表 518. TDES2 正常描述符 （回写格式）. 2604
表 519. TDES3 正常描述符 （回写格式）. 2604
表 520. TDES0 上下文描述符 2607
表 521. TDES1 上下文描述符 2607
表 522. TDES2 上下文描述符 2608
表 523. TDES3 上下文描述符 2608
表 524. RDES0 正常描述符 （读取格式）. 2611
表 525. RDES1 正常描述符 （读取格式）. 2611
表 526. RDES2 正常描述符 （读取格式）. 2611
表 527. RDES3 正常描述符 （读取格式）. 2612
表 528. RDES0 正常描述符 （回写格式）. 2613
表 529. RDES1 正常描述符 （回写格式）. 2613
表 530. RDES2 正常描述符 （回写格式）. 2615
表 531. RDES3 正常描述符 （回写格式）. 2617
表 532. RDES0 上下文描述符 2619
表 533. RDES1 上下文描述符 2619
表 534. RDES2 上下文描述符 2620
表 535. RDES3 上下文描述符 2620
表 536. ETH_DMA 通用寄存器映射和复位值 2643
表 537. ETH_DMA_CH 寄存器映射和复位值 2643
表 538. ETH_MTL 寄存器映射和复位值 2656
表 539. 基于 S2KP 和 JE 位的大型数据包状态 2662
表 540. 基于 CST 和 ACS 位的数据包长度 2662
表 541. 远程唤醒数据包过滤寄存器. 2683
表 542. ETH_MACRWKPFR 2685
表 543. 远程唤醒数据包和 PMT 中断生成 2686
表 544. 时间戳快照对寄存器位的相依性 2723
表 545. 以太网 MAC 寄存器映射和复位值. 2742
表 546. HDMI 引脚 2752
表 547. HDMI-CEC 内部输入 / 输出信号 2752
表 548. 错误处理时序参数 2759
表 549. TXERR 时序参数 2760
表 550. HDMI-CEC 中断 2761
表 551. HDMI-CEC 寄存器映射和复位值 2769
表 552. JTAG/ 串行调试端口引脚 2772
表 553. 跟踪端口引脚 2772
表 554. 串行线跟踪端口引脚 2772
表 555. 触发引脚 2772
表 556. 数据包请求 2776
表 557. ACK 响应 2776
表 558. 数据传输 2776
表 559. JTAG-DP 数据寄存器 2779
表 560. 调试端口寄存器. 2780
表 561. MEM-AP 寄存器 2790
表 562. 系统 ROM 表 1 2794
表 563. 系统 ROM 表 2 2795
表 564. 系统 ROM 表寄存器映射和复位值 2801
表 565. TSG 寄存器映射和复位值 2810
表 566. 系统 CTI 输入 2812

表 567. 系统 CTI 输出 2812
表 568. Cortex-M7 CTI 输入 2812
表 569. Cortex-M7 CTI 输出 2813
表 570. CTI 寄存器映射和复位值 2829
表 571. CSTF 寄存器映射和复位值 2841
表 572. ETF 寄存器映射和复位值 2863
表 573. TPIU 寄存器映射和复位值 2881
表 574. SWO 寄存器映射和复位值 2893
表 575. SWTF 寄存器映射和复位值 2904
表 576. DBGMCU 寄存器映射和复位值 2913
表 577. Cortex-M7 CPU ROM 表 2914
表 578. Cortex-M7 PPB ROM 表 2915
表 579. Cortex-M7 CPU ROM 表寄存器映射和复位值 2920
表 580. Cortex-M7 PPB ROM 表寄存器映射和复位值 2926
表 581. Cortex-M7 DWT 寄存器映射和复位值 2940
表 582. Cortex-M7 ITM 寄存器映射和复位值 2950
表 583. Cortex-M7 FPB 寄存器映射和复位值 2957
表 584. Cortex-M7 ETM 寄存器映射和复位值 2989
表 585. 文档版本历史 2996

图片索引

图 1. STM32H7X3 系统架构 . . . 96
图 2. Flash 框图 . . . 108
图 3. Flash 概述 . . . 110
图 4. 突发读取时序图 - 4x64 位，延时 = 3 个等待状态 . . . 111
图 5. 突发读取时序图 - 8x64 位，延时 = 3 个等待状态 . . . 112
图 6. 保护转换方案 . . . 127
图 7. 标准与安全访问模式下的 Flash 区域和服务 . . . 179
图 8. 安全自举程序状态机 . . . 180
图 9. 根安全服务调用 . . . 181
图 10. 内核对 Flash 区域的访问 . . . 185
图 11. AXI 互连 . . . 187
图 12. 电源控制框图 . . . 207
图 13. 电源概述 . . . 210
图 14. 系统电源配置 . . . 211
图 15. 由稳压器提供 V_{CORE} 时的器件启动情形 . . . 213
图 16. 备份域 . . . 216
图 17. USB 电源配置 . . . 217
图 18. 上电复位/掉电复位波形 . . . 218
图 19. BOR 阈值 . . . 219
图 20. PVD 阈值 . . . 220
图 21. AVD 阈值 . . . 221
图 22. VBAT 阈值 . . . 222
图 23. 温度阈值 . . . 223
图 24. V_{CORE} 电压调节与系统电源模式 . . . 227
图 25. 电源控制模式详细状态图 . . . 229
图 26. 运行模式下的动态电压调节 . . . 232
图 27. D1、D2 和系统处于停止模式时的动态电压调节行为 . . . 233
图 28. D1、D2 和系统待机模式下的动态电压调节 . . . 234
图 29. D1 和 D2 处于 DStandby 模式且 D3 处于自动模式时的动态电压调节行为 . . . 236
图 30. EXTI、RCC 和 PWR 互连 . . . 258
图 31. SRAM4 到 LPUART1 传输时序图（BDMA 和 D3 域处于自动模式） . . . 261
图 32. BDMA 和 DMAMUX2 互连 . . . 263
图 33. LPUART1 发送时序图（D3 域处于自动模式） . . . 266
图 34. RCC 框图 . . . 268
图 35. 系统复位电路 . . . 271
图 36. 启动序列与系统状态 . . . 276
图 37. 顶级时钟树 . . . 278
图 38. HSE/LSE 时钟源 . . . 279
图 39. PLL 框图 . . . 285
图 40. PLL 初始化流程图 . . . 288
图 41. 内核与总线时钟生成 . . . 290
图 42. SAI 和 DFSDM 的内核时钟分配 . . . 295
图 43. SPI 和 SPI/I2S 的内核时钟分配 . . . 296
图 44. I2C 的内核时钟分配 . . . 297
图 45. UART、USART 和 LPUART1 的内核时钟分配 . . . 297
图 46. LTDC 的内核时钟分配 . . . 298
图 47. SDMMC、QUADSPI 和 FMC 的内核时钟分配 . . . 298
图 48. USB 的内核时钟分配 (2) . . . 299

图 49. 以太网的内核时钟分配 . . . 299
图 50. ADC、SWPMI、RNG 和 FDCAN 的内核时钟分配 (2) . . . 300
图 51. LPTIM 和 HDMI-CEC 的内核时钟分配 (2) . . . 301
图 52. 外设分配示例 . . . 303
图 53. 内核时钟切换 . . . 306
图 54. 外设内核时钟使能逻辑详细信息 . . . 308
图 55. 总线时钟使能逻辑 . . . 312
图 56. RCC 映射概览 . . . 314
图 57. CRS 框图 . . . 422
图 58. CRS 计数器行为 . . . 424
图 59. HSEM 框图 . . . 434
图 60. 步骤状态图 . . . 435
图 61. 中断状态图 . . . 437
图 62. I/O 端口位的基本结构 . . . 446
图 63. 5 V 容忍 I/O 端口位的基本结构 . . . 446
图 64. 输入浮空/上拉/下拉配置 . . . 451
图 65. 输出配置 . . . 451
图 66. 复用功能配置 . . . 452
图 67. 高阻态模拟配置 . . . 453
图 68. 连接到 ADC 输入的模拟输入 . . . 454
图 69. MDMA 框图 . . . 519
图 70. DMA 框图 . . . 544
图 71. 外设到存储器模式 . . . 547
图 72. 存储器到外设模式 . . . 548
图 73. 存储器到存储器模式 . . . 549
图 74. FIFO 结构 . . . 554
图 75. BDMA 框图 . . . 576
图 76. DMAMUX 框图 . . . 595
图 77. DMAMUX 请求线复用器通道的同步模式 . . . 597
图 78. DMA 请求线复用器通道的事件生成 . . . 598
图 79. DMA2D 框图 . . . 611
图 80. EXTI 框图 . . . 651
图 81. 触发逻辑 CPU 唤醒的可配置事件 . . . 653
图 82. 可配置事件触发逻辑任意唤醒 . . . 654
图 83. 直接事件触发逻辑 CPU 唤醒 . . . 655
图 84. 直接事件触发逻辑任意唤醒 . . . 656
图 85. D3 域挂起请求清除逻辑 . . . 657
图 86. CRC 计算单元框图 . . . 684
图 87. FMC 框图 . . . 692
图 88. FMC 存储区域 . . . 695
图 89. 模式 1 读取访问波形 . . . 706
图 90. 模式 1 写入访问波形 . . . 706
图 91. 模式 A 读取访问波形 . . . 708
图 92. 模式 A 写入访问波形 . . . 709
图 93. 模式 2 和模式 B 读取访问波形 . . . 711
图 94. 模式 2 写入访问波形 . . . 711
图 95. 模式 B 写入访问波形 . . . 712
图 96. 模式 C 读取访问波形 . . . 714
图 97. 模式 C 写入访问波形 . . . 714
图 98. 模式 D 读取访问波形 . . . 717
图 99. 模式 D 写入访问波形 . . . 717
图 100. 复用读取访问波形 . . . 720

图 101. 复用写入访问波形 . . . 720
图 102. 读取访问波形期间的异步等待 . . . 723
图 103. 写入访问波形期间的异步等待 . . . 723
图 104. 等待配置波形 . . . 726
图 105. 同步复用读取模式波形 - NOR、PSRAM (CRAM) . . . 726
图 106. 同步复用写入模式波形 - PSRAM (CRAM) . . . 728
图 107. NAND Flash 控制器的通用存储器访问波形 . . . 740
图 108. 访问非“CE 无关”NAND-Flash . . . 742
图 109. 突发写入 SDRAM 访问波形 . . . 751
图 110. 突发读 SDRAM 访问 . . . 752
图 111. RBURST 位置 1 时的读访问逻辑图（CAS=2，RPIPE=0） . . . 753
图 112. 跨行边界的读访问 . . . 755
图 113. 跨行边界的写访问 . . . 755
图 114. 自刷新模式 . . . 757
图 115. 掉电模式 . . . 758
图 116. QUADSPI 功能框图（双闪存模式禁止） . . . 768
图 117. QUADSPI 功能框图（双闪存模式使能） . . . 769
图 118. 四线模式下的读命令示例 . . . 770
图 119. 四线模式下 DDR 命令示例 . . . 773
图 120. CKMODE = 0 时的 nCS（T = CLK 周期） . . . 781
图 121. SDR 模式下 CKMODE = 1 时的 nCS（T = CLK 周期） . . . 781
图 122. DDR 模式下 CKMODE = 1 时的 nCS（T = CLK 周期） . . . 781
图 123. CKMODE = 1 且发生中止时的 nCS（T = CLK 周期） . . . 782
图 124. DLYB 框图 . . . 796
图 125. ADC 框图 . . . 803
图 126. ADC 时钟方案 . . . 805
图 127. ADC1 连接功能 . . . 807
图 128. ADC2 连接功能 . . . 808
图 129. ADC3 连接功能 . . . 809
图 130. ADC 校准 . . . 812
图 131. 更新 ADC 偏移校准系数 . . . 813
图 132. 混合单端通道和差分通道 . . . 813
图 133. 使能/禁止 ADC . . . 816
图 134. 模数转换时间 . . . 821
图 135. 停止正在进行的常规转换 . . . 822
图 136. 停止正在进行的常规转换和注入转换 . . . 823
图 137. 触发由主 ADC 和从 ADC 共享 . . . 824
图 138. 注入转换延迟 . . . 828
图 139. JSQR 上下文队列示例（队列更改） . . . 830
图 140. JSQR 上下文队列示例（触发更改） . . . 831
图 141. 转换前发生溢出的 JSQR 上下文队列示例 . . . 831
图 142. 转换期间发生溢出的 JSQR 上下文队列示例 . . . 832
图 143. 队列为空时的 JSQR 上下文队列示例（JQM=0 的情况） . . . 832
图 144. 队列为空时的 JSQR 上下文队列示例（JQM=1 的情况） . . . 833
图 145. 通过将 JADSTP 置 1 的方式清空 JSQR 上下文队列 (JQM=0)。
正在进行转换时发生 JADSTP 的情况。 . . . 833
图 146. 通过将 JADSTP 置 1 的方式清空 JSQR 上下文队列 (JQM=0)。
正在进行转换时发生 JADSTP 并出现新触发的情况。 . . . 834
图 147. 通过将 JADSTP 置 1 的方式清空 JSQR 上下文队列 (JQM=0)。
在正在进行转换的范围之外发生 JADSTP 的情况。 . . . 834
图 148. 通过将 JADSTP 置 1 的方式清空 JSQR 上下文队列 (JQM=1)。 . . . 835
图 149. 通过将 ADDIS 置 1 的方式清空 JSQR 上下文队列 (JQM=0)。 . . . 835

图 150. 通过将 ADDIS 置 1 的方式清空 JSQR 上下文队列 (JQM=1)。 836
图 151. 单次序列转换，软件触发 838
图 152. 连续序列转换，软件触发 838
图 153. 单次序列转换，硬件触发 839
图 154. 连续序列转换，硬件触发 839
图 155. 右对齐（偏移禁止，无符号值） 841
图 156. 右对齐（偏移使能，有符号值） 841
图 157. 左对齐（偏移禁止，无符号值） 842
图 158. 左对齐（偏移使能，有符号值） 842
图 159. 溢出示例 (OVR) 844
图 160. AUTODLY=1，连续模式下的常规转换，软件触发 847
图 161. AUTODLY=1，被注入转换中断的常规硬件转换（DISCEN=0；JDISCEN=0） 848
图 162. AUTODLY=1，被注入转换中断的常规硬件转换（DISCEN=1，JDISCEN=1） 849
图 163. AUTODLY=1，被注入转换中断的常规连续转换 850
图 164. AUTODLY=1，自动注入模式 (JAUTO=1) 850
图 165. 模拟看门狗的保护区域 851
图 166. ADCy_AWDx_OUT 信号生成（所有常规通道上） 853
图 167. ADCy_AWDx_OUT 信号生成（AWDx 标志未通过软件清零） 853
图 168. ADCy_AWDx_OUT 信号生成（单条常规通道上） 853
图 169. ADCy_AWDx_OUT 信号生成（所有注入通道上） 854
图 170. 采用 10 位右移和四舍五入进行过采样得到的 16 位结果 855
图 171. 已触发的常规过采样模式（TROVS 位 = 1） 856
图 172. 常规过采样模式（4x 过采样率） 857
图 173. 同时使用常规和注入过采样模式 857
图 174. 已触发常规过采样支持注入 858
图 175. 在自动模式下进行过采样 858
图 176. 双重 ADC 框图(1) 860
图 177. 4 通道的注入同步模式：双重 ADC 模式 861
图 178. 16 通道的常规同步模式：双重 ADC 模式 862
图 179. 连续转换模式下 1 通道的交替模式：双重 ADC 模式 864
图 180. 单次转换模式下 1 通道的交替模式：双重 ADC 模式 864
图 181. 支持注入的交替转换 865
图 182. 交替触发：各个 ADC 的注入组 866
图 183. 交替触发：不连续采样模式下的 4 个注入通道（各个 ADC） 867
图 184. 交替 + 常规同步 867
图 185. 在注入转换期间出现的触发事件 868
图 186. 单条交替通道 CH0，注入序列 CH11 和 CH12 868
图 187. 两条交替通道（CH1、CH2），注入序列 CH11 和 CH12 - 情况 1：先中断主 ADC 869
图 188. 两条交替通道（CH1、CH2），注入序列 CH11 和 CH12 - 情况 2：先中断从 ADC 869
图 189. 常规同步模式下的 DMA 请求（DAMDF=0b00 时） 870
图 190. 常规同步模式下的 DMA 请求（DAMDF=0b10 时） 871
图 191. 交替模式下的 DMA 请求（DAMDF=0b10 时） 871
图 192. 温度传感器通道框图 873
图 193. VBAT 通道框图 874
图 194. VREFINT 通道框图 875
图 195. DAC 通道框图 919
图 196. DAC 单通道模式下的数据寄存器 921
图 197. DAC 双通道模式下的数据寄存器 922
图 198. 关闭触发 (TEN = 0) 时的转换时序图 TEN = 0 922
图 199. DAC LFSR 寄存器计算算法 924
图 200. LFSR 产生波形的 DAC 转换（使能软件触发） 925
图 201. 生成 DAC 三角波 925

图 202. 生成三角波波形的 DAC 转换（使能软件触发）. 926
图 203. DAC 采样和保持模式阶段图. 928
图 204. 比较器功能框图. 955
图 205. 比较器迟滞 957
图 206. 比较器输出消隐. 958
图 207. 输出重定向 960
图 208. 框图. 962
图 209. 独立模式：外部增益设置模式. 973
图 210. 跟随器配置 974
图 211. PGA 模式，内部增益设置 (x2/x4/x8/x16)，未使用反相输入 975
图 212. PGA 模式，内部增益设置 (x2/x4/x8/x16)，用于滤波的反相输入. 976
图 213. PGA 模式，非反相增益设置 (x2/x4/x8/x16) 或反相增益设置 (x-1/x-3/x-7/x-15). 977
图 214. 配置示例 977
图 215. PGA 模式，非反相增益设置 (x2/x4/x8/x16) 或使用滤波的反相增益设置 (x-1/x-3/x-7/x-15) 978
图 216. 配置示例 978
图 217. 单个 DFSDM 框图. 992
图 218. 输入通道引脚重定向 996
图 219. 通道收发器时序图. 998
图 220. SPI 时钟缺失时序图 999
图 221. 曼彻斯特编码时钟缺失时序图. 1000
图 222. 曼彻斯特编码首次转换（曼彻斯特同步） 1002
图 223. DFSDM_CHyDATINR 寄存器操作模式和分配 1006
图 224. 示例：Sinc3 滤波器响应. 1007
图 225. DCMI 框图 1049
图 226. 顶级框图 1049
图 227. DCMI 信号波形 1051
图 228. 时序图. 1053
图 229. 快照模式下的帧捕获波形 1055
图 230. 连续采集模式下的帧捕获波形. 1055
图 231. 裁剪后窗口的坐标和大小 1056
图 232. 数据捕获波形 1056
图 233. 像素光栅扫描顺序. 1057
图 234. LTDC 框图 1074
图 235. LCD-TFT 同步时序 1077
图 236. 层窗口可编程参数： 1079
图 237. 两层与背景混合. 1082
图 238. 中断事件 1083
图 239. JPEG 编解码器框图 1108
图 240. RNG框图 1123
图 241. 熵源模型 1124
图 242. 随机采样调节过程. 1125
图 243. RNG 初始化概述. 1127
图 244. CRYP 框图 1136
图 245. AES-ECB 模式概览. 1139
图 246. AES-CBC 模式概览 1140
图 247. AES-CTR 模式概览. 1141
图 248. AES-GCM 模式概览 1142
图 249. AES-GMAC 模式概览 1142
图 250. AES-CCM 模式概览 1143
图 251. STM32 加密库 DES/TDES 流程图 1144
图 252. STM32 加密库 AES 流程图示例 1145

图 253. 挂起模式管理示例 . . . 1150
图 254. DES/TDES-ECB 模式加密 . . . 1151
图 255. DES/TDES-ECB 模式解密 . . . 1152
图 256. DES/TDES-CBC 模式加密 . . . 1153
图 257. DES/TDES-CBC 模式解密 . . . 1154
图 258. AES-ECB 模式加密 . . . 1156
图 259. AES-ECB 模式解密 . . . 1157
图 260. AES-CBC 模式加密 . . . 1158
图 261. AES-CBC 模式解密 . . . 1159
图 262. 计数器模式的消息结构 . . . 1161
图 263. AES-CTR 模式加密 . . . 1162
图 264. AES-CTR 模式解密 . . . 1163
图 265. Galois/计数器模式的消息结构 . . . 1164
图 266. Galois 消息认证码模式的消息结构 . . . 1168
图 267. CBC-MAC 计数器模式的消息结构 . . . 1169
图 268. 根据数据类型构建 64 位块 (IN FIFO) . . . 1174
图 269. 根据数据类型构建 128 位块 . . . 1176
图 270. CRYP 中断映射图表 . . . 1181
图 271. HASH框图 . . . 1200
图 272. 消息数据交换功能 . . . 1202
图 273. HASH 保存/恢复机制 . . . 1207
图 274. 散列中断映射图 . . . 1209
图 275. 高分辨率定时器框图 . . . 1223
图 276. 定时器 A..E 概览 . . . 1227
图 277. 连续定时器工作模式 . . . 1228
图 278. 单发定时器工作模式 . . . 1229
图 279. 定时器复位重新同步（预分频比大于 32） . . . 1230
图 280. 连续模式下的重复率与 HRTIM_REPxR 内容 . . . 1231
图 281. 单发模式下的重复计数器行为 . . . 1232
图 282. 比较事件对输出的操作：发生比较 1 时置位，发生比较 2 时复位 . . . 1233
图 283. 定时单元捕获电路 . . . 1235
图 284. 自动延迟概述（仅限比较 2 寄存器） . . . 1236
图 285. 自动延迟比较 . . . 1237
图 286. 推挽模式框图 . . . 1239
图 287. 推挽模式示例 . . . 1239
图 288. 已插入死区的互补输出 . . . 1240
图 289. 死区插入与死区符号（1 表示负死区） . . . 1241
图 290. 低脉宽的互补输出 (SDTRx = SDTFx = 0) . . . 1242
图 291. 低脉宽的互补输出 (SDTRx = SDTFx = 1) . . . 1242
图 292. 低脉宽的互补输出（SDTRx = 0，SDTFx = 1） . . . 1242
图 293. 低脉宽的互补输出（SDTRx = 1，SDTFx = 0） . . . 1243
图 294. 主定时器概览 . . . 1244
图 295. 外部事件条件概览（显示了 1 条通道） . . . 1246
图 296. 外部事件下降沿的延迟（计数器复位和输出置位） . . . 1249
图 297. 外部事件的延迟（发生外部事件时，输出复位） . . . 1249
图 298. 事件消隐模式 . . . 1250
图 299. 事件延迟模式 . . . 1250
图 300. 采用边沿有效触发的外部触发消隐 . . . 1252
图 301. 外部触发消隐，电平有效触发 . . . 1252
图 302. 事件窗口模式 . . . 1253
图 303. 采用边沿有效触发的外部触发窗口 . . . 1254
图 304. 外部触发窗口，电平有效触发 . . . 1254

图 305. 延迟空闲模式进入 . . . 1256
图 306. 突发模式和延迟保护优先级 (DIDL = 0) . . . 1257
图 307. 突发模式和延迟保护优先级 (DIDL = 1) . . . 1258
图 308. 均衡空闲保护示例 . . . 1259
图 309. 输出管理概览 . . . 1267
图 310. HRTIM 输出状态和转换 . . . 1268
图 311. 突发模式工作示例 . . . 1269
图 312. 发生外部事件时触发突发模式 . . . 1272
图 313. 使能死区且 IDLESx = 1 时的延迟突发模式进入 . . . 1273
图 314. 死区内的延迟突发模式进入 . . . 1274
图 315. 死区发生器使能时的突发模式退出 . . . 1275
图 316. 突发模式仿真示例 . . . 1277
图 317. 载波频率信号插入 . . . 1277
图 318. 已使能斩波模式的 HRTIM 输出 . . . 1278
图 319. 故障保护电路（完全显示了 FAULT1，部分显示了 FAULT2..5） . . . 1279
图 320. 故障信号过滤（FLTxF[3:0]= 0010：$f_{SAMPLING}$ = fHRTIM，N = 4） . . . 1280
图 321. 辅助输出 . . . 1282
图 322. 突发模式期间的辅助输出和主输出 (DIDLx = 0) . . . 1283
图 323. 退出突发模式时辅助输出上的死区失真 . . . 1283
图 324. 同步启动模式下计数器的行为 . . . 1286
图 325. ADC 触发选择概览 . . . 1287
图 326. 将多个更新合并到一路 hrtim_dac_trgx 输出 . . . 1288
图 327. DMA 突发概览 . . . 1292
图 328. 突发 DMA 工作流程图 . . . 1293
图 329. DMA 突发传输后进行寄存器更新 . . . 1294
图 330. 降压转换器拓扑 . . . 1296
图 331. 双降压转换器管理 . . . 1296
图 332. 同步整流取决于输出电流 . . . 1297
图 333. 采用同步整流功能实现降压 . . . 1297
图 334. 3 相交错降压转换器 . . . 1298
图 335. 3 相交错降压转换器控制 . . . 1299
图 336. 过渡模式 PFC . . . 1299
图 337. 过渡模式 PFC 波形 . . . 1300
图 338. 高级控制定时器框图 . . . 1390
图 339. 预分频器分频由 1 变为 2 时的计数器时序图 . . . 1392
图 340. 预分频器分频由 1 变为 4 时的计数器时序图 . . . 1392
图 341. 计数器时序图，1 分频内部时钟 . . . 1394
图 342. 计数器时序图，2 分频内部时钟 . . . 1394
图 343. 计数器时序图，4 分频内部时钟 . . . 1395
图 344. 计数器时序图，N 分频内部时钟 . . . 1395
图 345. 计数器时序图，ARPE=0 时更新事件（TIMx_ARR 未预装载） . . . 1396
图 346. 计数器时序图，ARPE=1 时更新事件（TIMx_ARR 已预装载） . . . 1396
图 347. 计数器时序图，1 分频内部时钟 . . . 1398
图 348. 计数器时序图，2 分频内部时钟 . . . 1398
图 349. 计数器时序图，4 分频内部时钟 . . . 1399
图 350. 计数器时序图，N 分频内部时钟 . . . 1399
图 351. 计数器时序图，未使用重复计数器时更新事件 . . . 1400
图 352. 计数器时序图，1 分频内部时钟，TIMx_ARR = 0x6 . . . 1401
图 353. 计数器时序图，2 分频内部时钟 . . . 1402
图 354. 计数器时序图，4 分频内部时钟，TIMx_ARR=0x36 . . . 1402
图 355. 计数器时序图，N 分频内部时钟 . . . 1403
图 356. 计数器时序图，ARPE=1 时的更新事件（计数器下溢） . . . 1403

图 357. 计数器时序图，ARPE=1 时的更新事件（计数器上溢） 1404
图 358. 不同模式和 TIMx_RCR 寄存器设置下的更新频率示例 1405
图 359. 外部触发输入模块 1406
图 360. TIM1/TIM8 ETR 输入电路 1406
图 361. 正常模式下的控制电路，1 分频内部时钟 1407
图 362. TI2 外部时钟连接示例 1408
图 363. 外部时钟模式 1 下的控制电路 1409
图 364. 外部触发输入模块 1409
图 365. 外部时钟模式 2 下的控制电路 1410
图 366. 捕获/比较通道（例如：通道 1 输入阶段） 1411
图 367. 捕获/比较通道 1 主电路 1412
图 368. 捕获/比较通道的输出阶段（通道 1、 通道 2 和通道 3） 1412
图 369. 捕获/比较通道的输出阶段（通道 4） 1413
图 370. 捕获/比较通道的输出阶段（通道 5 和通道 6） 1413
图 371. PWM 输入模式时序 1415
图 372. 输出比较模式，翻转 OC1 1417
图 373. 边沿对齐模式的 PWM 波形 (ARR=8) 1418
图 374. 中心对齐模式 PWM 波形 (ARR=8) 1419
图 375. 50% 占空比时产生的 2 个相移 PWM 信号 1421
图 376. 通道 1 和通道 3 上的组合 PWM 模式 1422
图 377. 三相组合 PWM 信号（每个周期多个触发脉冲） 1423
图 378. 带死区插入的互补输出 1424
图 379. 延迟时间大于负脉冲宽度的死区波形 1424
图 380. 延迟时间大于正脉冲宽度的死区波形 1425
图 381. 断路和断路 2 电路概述 1427
图 382. 响应 BRK 上的断路事件的不同输出行为 (OSSI = 1) 1429
图 383. BRK 和 BRK2 引脚使能后的 PWM 输出状态 (OSSI=1) 1430
图 384. BRK 使能后的 PWM 输出状态 (OSSI=0) 1431
图 385. 输出重定向 1431
图 386. 清除 TIMx 的 OCxREF 1432
图 387. COM 事件生成 6 步 PWM 的示例 (OSSR=1) 1433
图 388. 单脉冲模式示例： 1434
图 389. 可再触发单脉冲模式 1436
图 390. 编码器接口模式下的计数器工作示例 1437
图 391. TI1FP1 极性反相时的编码器接口模式示例 1438
图 392. 测量 3 个信号上边沿之间的时间间隔 1439
图 393. 霍尔传感器接口的示例 1441
图 394. 复位模式下的控制电路 1442
图 395. 门控模式下的控制电路 1443
图 396. 触发模式下的控制电路 1444
图 397. 外部时钟模式 2 + 触发模式下的控制电路 1445
图 398. 通用定时器框图 1491
图 399. 预分频器分频由 1 变为 2 时的计数器时序图 1493
图 400. 预分频器分频由 1 变为 4 时的计数器时序图 1493
图 401. 计数器时序图，1 分频内部时钟 1494
图 402. 计数器时序图，2 分频内部时钟 1495
图 403. 计数器时序图，4 分频内部时钟 1495
图 404. 计数器时序图，N 分频内部时钟 1496
图 405. 计数器时序图，ARPE=0 时更新事件（TIMx_ARR 未预装载） 1496
图 406. 计数器时序图，ARPE=1 时更新事件（TIMx_ARR 已预装载） 1497
图 407. 计数器时序图，1 分频内部时钟 1498
图 408. 计数器时序图，2 分频内部时钟 1498

图 409. 计数器时序图，4 分频内部时钟 1499
图 410. 计数器时序图，N 分频内部时钟 1499
图 411. 计数器时序图，不使用重复计数器时更新事件 1500
图 412. 计数器时序图，1 分频内部时钟，TIMx_ARR=0x6 1501
图 413. 计数器时序图，2 分频内部时钟 1502
图 414. 计数器时序图，4 分频内部时钟，TIMx_ARR=0x36 1502
图 415. 计数器时序图，N 分频内部时钟 1503
图 416. 计数器时序图，ARPE=1 时的更新事件（计数器下溢） 1503
图 417. 计数器时序图，ARPE=1 时的更新事件（计数器上溢） 1504
图 418. 正常模式下的控制电路，1 分频内部时钟 1505
图 419. TI2 外部时钟连接示例 1505
图 420. 外部时钟模式 1 下的控制电路 1506
图 421. 外部触发输入模块 1507
图 422. 外部时钟模式 2 下的控制电路 1507
图 423. 捕获/比较通道（例如：通道 1 输入阶段） 1508
图 424. 捕获/比较通道 1 主电路 1509
图 425. 捕获/比较通道的输出阶段（通道 1） 1509
图 426. PWM 输入模式时序 1511
图 427. 输出比较模式，翻转 OC1 1513
图 428. 边沿对齐模式的 PWM 波形 (ARR=8) 1514
图 429. 中心对齐模式 PWM 波形 (ARR=8) 1515
图 430. 50% 占空比时产生的 2 个相移 PWM 信号 1516
图 431. 通道 1 和通道 3 上的组合 PWM 模式 1517
图 432. 清除 TIMx 的 OCxREF 1518
图 433. 单脉冲模式示例 1519
图 434. 可再触发单脉冲模式 1520
图 435. 编码器接口模式下的计数器工作示例 1522
图 436. TI1FP1 极性反相时的编码器接口模式示例 1522
图 437. 复位模式下的控制电路 1524
图 438. 门控模式下的控制电路 1525
图 439. 触发模式下的控制电路 1525
图 440. 外部时钟模式 2 + 触发模式下的控制电路 1526
图 441. 主/从定时器示例 1527
图 442. 使用 TIM3 的 OC1REF 对 TIM2 实施门控控制 1528
图 443. 使用 TIM3 的使能信号对 TIM2 实施门控控制 1529
图 444. 使用 TIM3 的更新事件触发 TIM2 1529
图 445. 使用 TIM3 的使能信号触发 TIM2 1530
图 446. 使用 TIM3 的 TI1 输入触发 TIM3 和 TIM2 1531
图 447. 通用定时器框图(TIM12) 1562
图 448. 通用定时器框图(TIM13/TIM14) 1563
图 449. 预分频器分频由 1 变为 2 时的计数器时序图 1564
图 450. 预分频器分频由 1 变为 4 时的计数器时序图 1565
图 451. 计数器时序图，1 分频内部时钟 1566
图 452. 计数器时序图，2 分频内部时钟 1566
图 453. 计数器时序图，4 分频内部时钟 1567
图 454. 计数器时序图，N 分频内部时钟 1567
图 455. 计数器时序图，ARPE=0 时更新事件（TIMx_ARR 未预装载） 1568
图 456. 计数器时序图，ARPE=1 时更新事件（TIMx_ARR 已预装载） 1568
图 457. 正常模式下的控制电路，1 分频内部时钟 1569
图 458. TI2 外部时钟连接示例 1570
图 459. 外部时钟模式 1 下的控制电路 1571
图 460. 捕获/比较通道（例如：通道 1 输入阶段） 1571

图 461. 捕获/比较通道 1 主电路 1572
图 462. 捕获/比较通道的输出阶段（通道 1） 1572
图 463. PWM 输入模式时序 1574
图 464. 输出比较模式，翻转 OC1。 1576
图 465. 边沿对齐模式的 PWM 波形 (ARR=8) 1577
图 466. 通道 1 和通道 2 上的组合 PWM 模式 1578
图 467. 单脉冲模式示例： 1579
图 468. 可再触发单脉冲模式 1580
图 469. 测量 2 个信号边沿之间的时间间隔 1581
图 470. 复位模式下的控制电路 1582
图 471. 门控模式下的控制电路 1582
图 472. 触发模式下的控制电路 1583
图 473. TIM15框图 1612
图 474. TIM16/TIM17框图. 1613
图 475. 预分频器分频由 1 变为 2 时的计数器时序图. 1615
图 476. 预分频器分频由 1 变为 4 时的计数器时序图. 1615
图 477. 计数器时序图，1 分频内部时钟 1616
图 478. 计数器时序图，2 分频内部时钟 1617
图 479. 计数器时序图，4 分频内部时钟 1617
图 480. 计数器时序图，N 分频内部时钟 1618
图 481. 计数器时序图，ARPE=0 时更新事件（TIMx_ARR 未预装载） 1618
图 482. 计数器时序图，ARPE=1 时更新事件（TIMx_ARR 未预装载） 1619
图 483. 不同模式和 TIMx_RCR 寄存器设置下的更新频率示例 1620
图 484. 正常模式下的控制电路，1 分频内部时钟 1621
图 485. TI2 外部时钟连接示例. 1621
图 486. 外部时钟模式 1 下的控制电路. 1622
图 487. 捕获/比较通道（例如：通道 1 输入阶段） 1623
图 488. 捕获/比较通道 1 主电路 1623
图 489. 捕获/比较通道的输出阶段（通道 1） 1624
图 490. 捕获/比较通道的输出阶段（TIM15 的通道 2） 1624
图 491. PWM 输入模式时序 1626
图 492. 输出比较模式，翻转 OC1 1628
图 493. 边沿对齐模式的 PWM 波形 (ARR=8) 1629
图 494. 通道 1 和通道 2 上的组合 PWM 模式 1630
图 495. 带死区插入的互补输出。 1631
图 496. 延迟时间大于负脉冲宽度的死区波形。 1631
图 497. 延迟时间大于正脉冲宽度的死区波形。 1632
图 498. 断路电路概述 1634
图 499. 输出的断路响应行为。 1636
图 500. 单脉冲模式示例： 1637
图 501. 可再触发单脉冲模式 1638
图 502. 测量 2 个信号上边沿之间的时间间隔 1639
图 503. 复位模式下的控制电路 1640
图 504. 门控模式下的控制电路 1641
图 505. 触发模式下的控制电路 1642
图 506. 基本定时器框图. 1689
图 507. 预分频器分频由 1 变为 2 时的计数器时序图. 1691
图 508. 预分频器分频由 1 变为 4 时的计数器时序图. 1691
图 509. 计数器时序图，1 分频内部时钟 1692
图 510. 计数器时序图，2 分频内部时钟 1693
图 511. 计数器时序图，4 分频内部时钟 1693
图 512. 计数器时序图，N 分频内部时钟 1694

图 513. 计数器时序图，ARPE = 0 时更新事件（TIMx_ARR 未预装载）. 1694
图 514. 计数器时序图，ARPE=1 时更新事件（TIMx_ARR 预装载）. 1695
图 515. 正常模式下的控制电路，1 分频内部时钟 1696
图 516. 低功耗定时器框图（LPTIM1 和 LPTIM2）. 1703
图 517. 低功耗定时器框图 (LPTIM3). 1704
图 518. 低功耗定时器框图（LPTIM4 和 LPTIM5）. 1704
图 519. 干扰滤波器时序图. 1709
图 520. LPTIM 输出波形，单次计数模式配置 1710
图 521. LPTIM 输出波形，单次计数模式配置且激活置 1 一次模式（WAVE 位置 1） 1710
图 522. LPTIM 输出波形、连续计数模式配置 1711
图 523. 生成波形 1712
图 524. 编码器模式计数序列 1715
图 525. 看门狗框图 1730
图 526. 窗口看门狗时序图 1732
图 527. 独立看门狗框图. 1736
图 528. RTC 块概览. 1745
图 529. 详细的 RTC 框图 1746
图 530. 入侵检测 1747
图 531. I2C 框图 1787
图 532. I2C 总线协议. 1789
图 533. 建立和保持时序. 1790
图 534. I2C 初始化流程图 1793
图 535. 数据接收 1794
图 536. 数据发送 1795
图 537. 从器件初始化流程图 1798
图 538. I2C 从发送器的传输序列流程图 (NOSTRETCH=0) 1799
图 539. I2C 从发送器的传输序列流程图 (NOSTRETCH=1) 1800
图 540. I2C 从发送器的传输总线图 1801
图 541. 从接收器的传输序列流程图 (NOSTRETCH=0). 1802
图 542. 从接收器的传输序列流程图 (NOSTRETCH=1). 1803
图 543. I2C 从接收器的传输总线图 1803
图 544. 主时钟生成 1805
图 545. 主模式初始化流程图 1807
图 546. 10 位地址读访问 (HEAD10R=0). 1807
图 547. 10 位地址读访问 (HEAD10R=1). 1808
图 548. I2C 主发送器的传输序列流程图（N≤255 字节） 1809
图 549. I2C 主发送器的传输序列流程图（N>255 字节） 1810
图 550. I2C 主发送器的传输总线图 1811
图 551. I2C 主接收器的传输序列流程图（N≤255 字节） 1813
图 552. I2C 主接收器的传输序列流程图（N>255 字节） 1814
图 553. I2C 主接收器的传输总线图 1815
图 554. $t_{LOW:SEXT}$ 和 $t_{LOW:MEXT}$ 的超时间隔. 1819
图 555. SMBus 从发送器的传输序列流程图（N 字节 + PEC） 1823
图 556. SMBus 从发送器的传输总线图 (SBC=1) 1823
图 557. SMBus 从接收器的传输序列流程图（N 字节 + PEC） 1825
图 558. SMBus 从接收器的总线传输图 (SBC=1) 1826
图 559. SMBus 主发送器的总线传输图 1827
图 560. SMBus 主接收器的总线传输图 1828
图 561. I2C 中断映射图. 1833
图 562. USART 框图 1852
图 563. 字长编程 1855
图 564. 可配置的停止位. 1857

图 565. 发送时的 TC/TXE 行为 . . . 1858
图 566. 16 倍或 8 倍过采样时的起始位检测 . . . 1859
图 567. usart_ker_ck 时钟分频器框图 . . . 1862
图 568. 16 倍过采样时的数据采样 . . . 1863
图 569. 8 倍过采样时的数据采样 . . . 1864
图 570. 使用空闲线路检测时的静默模式 . . . 1869
图 571. 使用地址标记检测时的静默模式 . . . 1870
图 572. LIN 模式下的中断检测（11 位中断长度——LBDL 位置 1） . . . 1873
图 573. LIN 模式下的中断检测与帧错误检测 . . . 1874
图 574. USART 同步主发送示例 . . . 1875
图 575. USART 在同步主模式下的数据时钟时序图（M 位 = “00”） . . . 1875
图 576. USART 在同步主模式下的数据时钟时序图（M 位 = “01”） . . . 1876
图 577. USART 在同步从模式下的数据时钟时序图（M 位 = “00”） . . . 1877
图 578. ISO 7816-3 异步协议 . . . 1879
图 579. 使用 1.5 个停止位检测奇偶校验错误 . . . 1880
图 580. IrDA SIR ENDEC 框图 . . . 1883
图 581. IrDA 数据调制 (3/16)——正常模式 . . . 1884
图 582. 使用 DMA 进行发送 . . . 1885
图 583. 使用 DMA 进行接收 . . . 1886
图 584. 2 个 USART 间的硬件流控制 . . . 1886
图 585. RS232 RTS 流控制 . . . 1887
图 586. RS232 CTS 流控制 . . . 1888
图 587. 唤醒事件通过验证（唤醒事件 = 地址匹配，禁止 FIFO） . . . 1890
图 588. 唤醒事件未通过验证（唤醒事件 = 地址匹配，禁止 FIFO） . . . 1891
图 589. LPUART 框图 . . . 1921
图 590. LPUART 字长编程 . . . 1923
图 591. 可配置的停止位 . . . 1925
图 592. 发送时的 TC/TXE 行为 . . . 1926
图 593. lpuart_ker_ck 时钟分频器框图 . . . 1930
图 594. 使用空闲线路检测时的静默模式 . . . 1934
图 595. 使用地址标记检测时的静默模式 . . . 1935
图 596. 使用 DMA 进行发送 . . . 1937
图 597. 使用 DMA 进行接收 . . . 1938
图 598. 2 个 LPUART 间的硬件流控制 . . . 1939
图 599. RS232 RTS 流控制 . . . 1939
图 600. RS232 CTS 流控制 . . . 1940
图 601. 唤醒事件通过验证（唤醒事件 = 地址匹配，禁止 FIFO） . . . 1942
图 602. 唤醒事件未通过验证（唤醒事件为地址匹配，禁止 FIFO） . . . 1943
图 603. SPI2S 框图 . . . 1963
图 604. 全双工单个主器件/单个从器件应用 . . . 1965
图 605. 半双工单个主器件/单个从器件应用 . . . 1966
图 606. 单工单个主器件/单个从器件应用（主器件为只发送模式/从器件为只接收模式） . . . 1967
图 607. 使用星形拓扑的主器件和三个独立从器件 . . . 1968
图 608. 一主三从环形（菊花链）拓扑 . . . 1969
图 609. 多主应用 . . . 1970
图 610. SS 控制逻辑原理图 . . . 1972
图 611. 数据流时序控制（SSOE = 1，SSOM = 0，SSM = 0） . . . 1972
图 612. 数据之间的 SS 交错脉冲（SSOE = 1，SSOM = 1，SSM = 0） . . . 1973
图 613. 数据时钟时序图 . . . 1975
图 614. 数据大小不等于 8 位、16 位或 32 位时的数据对齐 . . . 1976
图 615. 将数据封装到 FIFO 中以进行发送和接收 . . . 1982
图 616. TI 模式传输 . . . 1984

图 617. 低功耗模式应用示例 1988
图 618. 波形示例 1993
图 619. 主模式 I2S Philips 协议波形（16/32 位全精度） 1993
图 620. I2S Philips 标准波形 1994
图 621. 主模式 MSB 对齐的 16 位或 32 位全精度长度 1994
图 622. 主模式 MSB 对齐的 16 位或 24 位数据长度 1995
图 623. 从模式 MSB 对齐 1995
图 624. LSB 对齐的 16 位或 24 位数据长度 1996
图 625. 帧长度等于数据长度时的主模式 PCM 1996
图 626. 主模式 PCM 标准波形（16 位或 24 位数据长度） 1997
图 627. 从模式 PCM 波形 1997
图 628. 启动序列，I2S Philips 标准，主模式 2000
图 629. 启动序列，I2S Philips 标准，从模式 2000
图 630. 停止序列，I2S Philips 标准，主模式 2001
图 631. I^2S 时钟发生器架构. 2001
图 632. 数据格式 2004
图 633. 下溢情况的处理. 2006
图 634. 上溢情况的处理. 2007
图 635. 帧错误检测，FIXCH = 0 2008
图 636. 帧错误检测，FIXCH = 1 2008
图 637. SAI 功能框图. 2034
图 638. 音频帧 2038
图 639. FS 的作用是 SOF 信号 + 通道识别信号 (FSDEF = TRIS = 1) 2040
图 640. FS 的作用是 SOF 信号 (FSDEF = 0) 2041
图 641. SAI_xSLOTR 中的 FBOFF = 0 时的 Slot 大小配置 2042
图 642. 第一位偏移 2042
图 643. 音频模块时钟发生器概览 2043
图 644. PDM 典型连接和时序 2047
图 645. 详细的 PDM 接口模块框图 2048
图 646. 启动序列 2049
图 647. TDM 中的 SAI_ADR 格式（32 位 Slot 宽度） 2050
图 648. TDM 中的 SAI_ADR 格式（16 位 Slot 宽度） 2051
图 649. TDM 中的 SAI_ADR 格式（8 位 Slot 宽度） 2052
图 650. AC’97 音频帧 2055
图 651. 至少具有 2 个嵌入式 SAI 的器件的典型 AC’97 配置示例（三个外部 AC’97 解码器） 2056
图 652. SPDIF 格式. 2057
图 653. SAI_xDR 寄存器定序 2058
图 654. SAI 的音频模块中的数据压扩硬件 2061
图 655. 发送无效 Slot 时 SD 输出线上的三态策略 2062
图 656. 采用 I2S 等协议时输出数据线上的三态策略 2063
图 657. 上溢错误检测 2064
图 658. FIFO 下溢事件 2064
图 659. SPDIFRX 框图 2086
图 660. S/PDIF 子帧格式. 2087
图 661. S/PDIF 块格式 2087
图 662. S/PDIF 报头 2088
图 663. 通道编码示例 2089
图 664. SPDIFRX 解码器 2090
图 665. 噪声滤波和边沿检测 2090
图 666. 阈值. 2092
图 667. 同步流程图 2094
图 668. 同步过程调度 2095

图 669. SPDIFRX 状态 . . . 2096
图 670. SPDIFRX_DR 寄存器格式 . . . 2098
图 671. 通道/用户数据格式 . . . 2099
图 672. RXSTEO = 0 时的 S/PDIF 上溢错误 . . . 2101
图 673. RXSTEO = 1 时的 S/PDIF 上溢错误 . . . 2102
图 674. SPDIFRX 接口中断映射图 . . . 2105
图 675. S1 信号编码 . . . 2124
图 676. S2 信号编码 . . . 2124
图 677. SWPMI 框图 . . . 2126
图 678. SWP 总线状态 . . . 2128
图 679. SWP 帧结构 . . . 2129
图 680. SWPMI 无软件缓冲区模式发送 . . . 2131
图 681. SWPMI 无软件缓冲区模式发送，连续帧 . . . 2131
图 682. SWPMI 多软件缓冲区模式发送 . . . 2134
图 683. SWPMI 无软件缓冲区模式接收 . . . 2135
图 684. SWPMI 单软件缓冲区模式接收 . . . 2137
图 685. SWPMI 多软件缓冲区模式接收 . . . 2139
图 686. SWPMI 单缓冲区模式接收（带 CRC 错误） . . . 2140
图 687. MDIOS 框图 . . . 2153
图 688. MDIO 协议写入帧波形 . . . 2154
图 689. MDIO 协议读取帧波形 . . . 2154
图 690. SDMMC“无响应”和“无数据”操作 . . . 2167
图 691. SDMMC（多个）块读取操作 . . . 2167
图 692. SDMMC（多个）块写入操作 . . . 2167
图 693. SDMMC（连续）流读取操作 . . . 2168
图 694. SDMMC（连续）流写入操作 . . . 2168
图 695. SDMMC 框图 . . . 2169
图 696. SDMMC 命令和数据相位关系 . . . 2171
图 697. 控制单元 . . . 2173
图 698. 命令/响应路径 . . . 2174
图 699. 命令路径状态机 (CPSM) . . . 2175
图 700. 数据路径 . . . 2180
图 701. DDR 模式数据包时钟驱动 . . . 2181
图 702. DDR 模式 CRC 状态/启动确认时钟驱动 . . . 2181
图 703. 数据路径状态机 (DPSM) . . . 2182
图 704. CLKMUX 单元 . . . 2191
图 705. 异步中断生成 . . . 2195
图 706. 同步中断周期数据读取 . . . 2196
图 707. 同步中断周期数据写入 . . . 2196
图 708. 异步中断周期数据读取 . . . 2197
图 709. 异步中断周期数据写入 . . . 2197
图 710. 在 DS、HS、SDR12 和 SDR25 模式下通过 SDMMC_CK 停止时钟 . . . 2200
图 711. 在 DDR50、SDR50 和 SDR104 模式下通过 SDMMC_CK 停止时钟 . . . 2200
图 712. SDMMC_CK < 50 MHz 时的读取等待 . . . 2201
图 713. SDMMC_CK > 50 MHz 时的读取等待 . . . 2201
图 714. CMD12 流时序 . . . 2203
图 715. CMD5 睡眠唤醒过程 . . . 2205
图 716. 正常启动模式操作 . . . 2207
图 717. 备用启动模式操作 . . . 2208
图 718. 命令响应 R1b 繁忙信号 . . . 2209
图 719. SDMMC 状态控制 . . . 2210
图 720. 卡掉电再上电/上电图 . . . 2210

图 721. 硬件流时序 . . . 2211
图 722. CMD11 信号电压切换序列 . . . 2211
图 723. 电压切换收发器的典型应用 . . . 2213
图 724. CAN 子系统 . . . 2240
图 725. FDCAN 框图 . . . 2242
图 726. 收发器延迟测量 . . . 2247
图 727. 总线监控模式下的引脚控制 . . . 2248
图 728. 环回模式下的引脚控制 . . . 2250
图 729. 消息 RAM 配置 . . . 2251
图 730. 标准消息 ID 过滤器路径 . . . 2254
图 731. 扩展消息 ID 过滤器路径 . . . 2255
图 732. 混合配置专用发送缓冲区/发送 FIFO 示例 . . . 2260
图 733. 混合配置专用发送缓冲区/发送队列示例 . . . 2261
图 734. 绕过操作 . . . 2262
图 735. FSM 校准 . . . 2264
图 736. 周期时间和全局时间同步 . . . 2277
图 737. TTCAN 0 级和 2 级偏移补偿 . . . 2278
图 738. 0 级调度同步状态机 . . . 2284
图 739. 0 级主控节点与被控节点的关系 . . . 2285
图 740. 高速 OTG 模块框图 (OTG_HS1) . . . 2371
图 741. 高速 OTG 模块框图 (OTG_HS2) . . . 2371
图 742. SOF 连接（SOF 触发输出与 TIM 和 ITR1 的连接） . . . 2381
图 743. 动态更新 OTG_HFIR . . . 2383
图 744. 设备模式下的 FIFO 地址映射和 AHB FIFO 访问映射 . . . 2384
图 745. 主机模式下的 FIFO 地址映射和 AHB FIFO 访问映射 . . . 2385
图 746. 中断层级 . . . 2388
图 747. 发送 FIFO 写任务 . . . 2474
图 748. 接收 FIFO 读任务 . . . 2475
图 749. 正常批量/控制 OUT/SETUP . . . 2476
图 750. 批量/控制 IN 事务 . . . 2480
图 751. 正常中断 OUT . . . 2482
图 752. 正常中断 IN . . . 2486
图 753. 同步 OUT 事务 . . . 2488
图 754. 同步 IN 事务 . . . 2491
图 755. 正常批量/控制 OUT/SETUP 事务 - DMA . . . 2493
图 756. 正常批量/控制 IN 事务 - DMA . . . 2495
图 757. 正常中断 OUT 事务 - DMA 模式 . . . 2496
图 758. 正常中断 IN 事务 - DMA 模式 . . . 2497
图 759. 正常同步 OUT 事务 - DMA 模式 . . . 2498
图 760. 正常同步 IN 事务 - DMA 模式 . . . 2499
图 761. 接收 FIFO 数据包读取 . . . 2505
图 762. 处理 SETUP 数据包 . . . 2507
图 763. 批量 OUT 事务 . . . 2513
图 764. TRDT 最大时序情况 . . . 2521
图 765. A 器件 SRP . . . 2522
图 766. B 器件 SRP . . . 2523
图 767. A 器件 HNP . . . 2524
图 768. B 器件 HNP . . . 2525
图 769. 以太网高级框图 . . . 2532
图 770. DMA 发送流程（标准模式） . . . 2535
图 771. DMA 发送流程（OSP 模式） . . . 2537
图 772. 接收 DMA 流程 . . . 2539

图 773. MAC 发送流程概览 . . . 2541
图 774. MAC 接收流程 . . . 2543
图 775. 数据包过滤序列 . . . 2547
图 776. 网络时间同步 . . . 2554
图 777. 支持点对点路径校准的时钟的传播延迟计算 . . . 2555
图 778. 使用精密方法更新系统时间 . . . 2559
图 779. TCP 分段减荷概览 . . . 2565
图 780. 分段数据包的报头和有效负载字段 . . . 2567
图 781. 支持的 PHY 接口 . . . 2574
图 782. SMA 接口模块 . . . 2575
图 783. MDIO 数据包结构 . . . 2576
图 784. 写数据包 . . . 2576
图 785. 读数据包 . . . 2577
图 786. 介质独立接口 (MII) 信号 . . . 2578
图 787. RMII 框图 . . . 2580
图 788. 发送位序 . . . 2580
图 789. 接收位序 . . . 2581
图 790. 描述符环结构 . . . 2597
图 791. DMA 描述符环 . . . 2598
图 792. 发送描述符（读取格式） . . . 2599
图 793. 发送描述符回写格式 . . . 2603
图 794. 发送上下文描述符格式 . . . 2607
图 795. 接收正常描述符（读取格式） . . . 2610
图 796. 接收正常描述符（回写格式） . . . 2612
图 797. 接收上下文描述符 . . . 2619
图 798. ETH_DMAISR 标志的生成 . . . 2636
图 799. HDMI-CEC 框图 . . . 2752
图 800. 消息结构 . . . 2753
图 801. 块 . . . 2753
图 802. 位时序 . . . 2754
图 803. 信号空闲时间 . . . 2755
图 804. 仲裁阶段 . . . 2755
图 805. 三个标称位周期的 SFT . . . 2755
图 806. 错误位时序 . . . 2757
图 807. 错误处理 . . . 2758
图 808. TXERR 检测 . . . 2760
图 809. 调试基础结构框图 . . . 2771
图 810. 调试基础结构的电源域 . . . 2773
图 811. 调试基础结构的时钟域 . . . 2774
图 812. SWD 成功的数据传输过程 . . . 2777
图 813. JTAG TAP 状态机 . . . 2778
图 814. 调试和访问端口连接 . . . 2788
图 815. APB-D CoreSight 组件拓扑 . . . 2796
图 816. 全局时间戳分布 . . . 2802
图 817. 嵌入式交叉触发 . . . 2811
图 818. 将触发输入映射到输出 . . . 2813
图 819. ETF 状态转换图 . . . 2844
图 820. Cortex-M7 CoreSight 拓扑 . . . 2915

1 文档约定

1.1 寄存器相关缩写词列表

寄存器说明中使用以下缩写词：

读/写 (rw)	软件可以读写该位。
只读 (r)	软件只能读取该位。
只写 (w)	软件只能写入该位。读取该位时将返回复位值。
读取/清零 (rc_w0)	软件可以读取该位，也可以通过写入 0 将该位清零。写入 1 对该位的值无影响。
读取/清零 (rc_w1)	软件可以读取该位，也可以通过写入 1 将该位清零。写入 0 对该位的值无影响。
读取/读取清零 (rc_r)	软件可以读取该位。读取该位时，将自动清零。写入该位对其值无影响。
读取/置位 (rs)	软件可以读取该位，也可将其置 1。写入 0 对该位的值无影响。
保留 (Res.)	保留位，必须保持复位值。

1.2 词汇表

本节简要介绍本文档中所用首字母缩略词和缩写词的定义：

- **字**：32 位数据。
- **半字**：16 位数据。
- **字节**：8 位数据。
- **双字**：64 位数据。
- Flash 字：256 位数据。
- **IAP（在应用中编程）**：IAP 是指可以在用户程序运行期间对微控制器的 Flash 进行重新编程。
- **ICP（在线编程）**：ICP 是指可以在器件安装于用户应用电路板上时使用 JTAG 协议、SWD 协议或自举程序对微控制器的 Flash 进行编程。
- **选项字节**：存储于 Flash 中的产品配置位。
- **AHB**：高级高性能总线。
- AXI：高级可扩展接口协议。
- **PCROP**：专有代码读保护。
- RDP：读保护。

1.3 外设可用性

有关各型号产品的外设可用性及数量信息，请参见特殊器件数据手册。

2 存储器和总线架构

2.1 系统架构

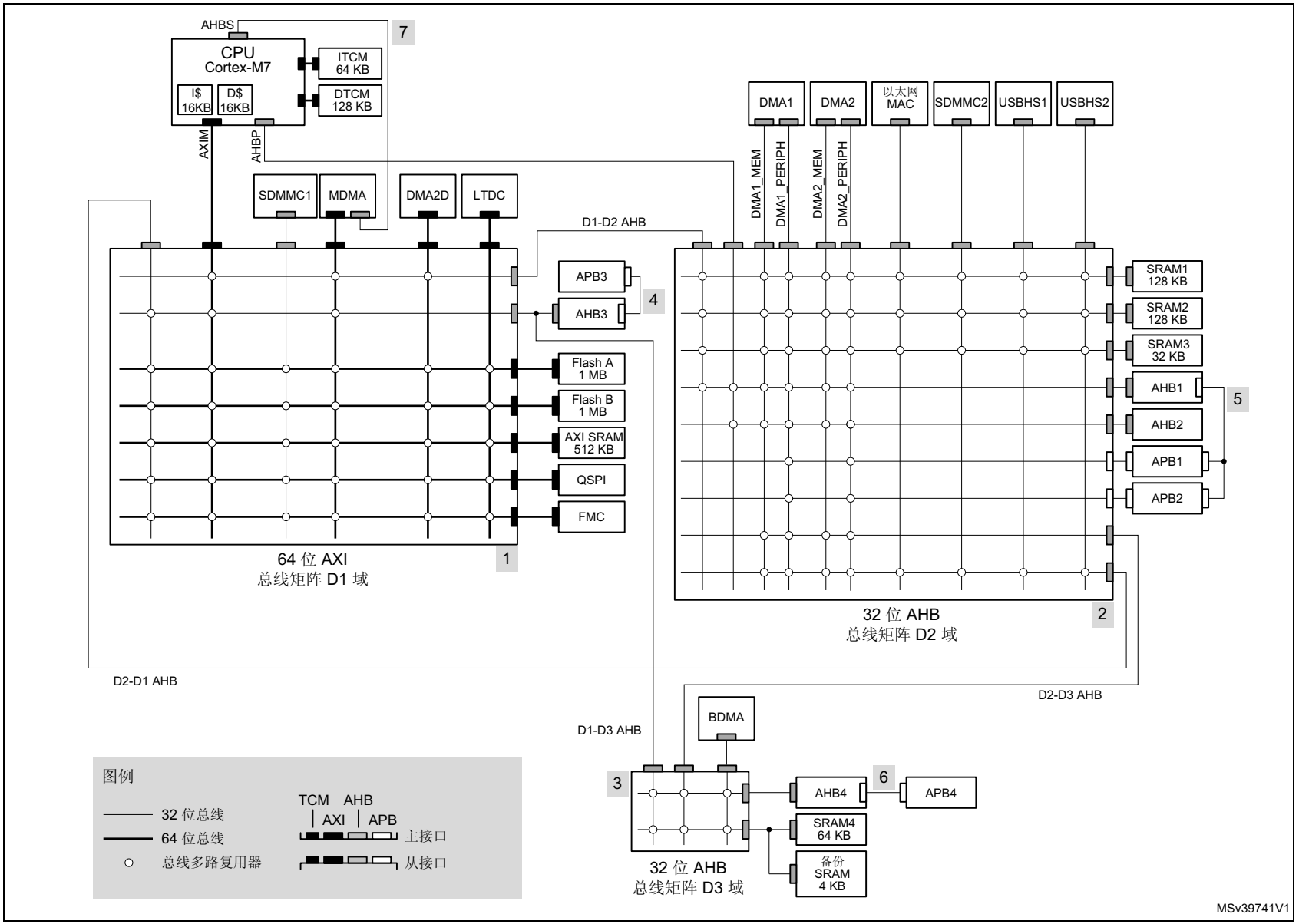
通过一个 AXI 总线矩阵、两个 AHB 总线矩阵和总线桥，可以将总线主设备与总线从设备实现互连，如*表 1* 和*图 1* 所示。

表 1. 总线主设备与总线从设备互连

	总线主设备/类型(1)																			
	Cortex-M7 - AXIM	Cortex-M7 - AHBP	**Cortex-M7 - ITCM**	**Cortex-M7 -DTCM**	SDMMC1	**MDMA - AXI**	MDMA - AHBS	**DMA2D**	**LTDC**	DMA1 - MEM	DMA1 - PERIPH	DMA2 - MEM	DMA2 - PERIPH	Eth MAC - AHB	SDMMC2 - AHB	USBHS1 - AHB	USBHS2 - AHB	BDMA - AHB
总线从设备/类型(1)	互连路径和类型(2)																			
ITCM	-	-	D	-	-	-	7	-	-	-	-	-	-	-	-	-	-	-	-	-
DTCM	-	-	-	D	-	-	7	-	-	-	-	-	-	-	-	-	-	-	-	-
AHB3 外设	1	-	-	-	-	1	-	-	-	21	21	-	21	21	-	21	21	21	21	-
APB3 外设	14	-	-	-	-	14	-	-	-	214	214	-	214	214	-	214	214	214	214	-
Flash A	**1**	-	-	-	*1*	**1**	-	**1**	**1**	*21*	*21*	-	*21*	*21*	-	*21*	*21*	*21*	*21*	-
Flash B	**1**	-	-	-	*1*	**1**	-	**1**	**1**	*21*	*21*	-	*21*	*21*	-	*21*	*21*	*21*	*21*	-
AXI SRAM	**1**	-	-	-	*1*	**1**	-	**1**	**1**	*21*	*21*	-	*21*	*21*	-	*21*	*21*	*21*	*21*	-
QUADSPI	**1**	-	-	-	*1*	**1**	-	**1**	**1**	*21*	*21*	-	*21*	*21*	-	*21*	*21*	*21*	*21*	-
FMC	**1**	-	-	-	*1*	**1**	-	**1**	**1**	*21*	*21*	-	*21*	*21*	-	*21*	*21*	*21*	*21*	-
SRAM 1	12	-	-	-	-	12	-	12	-	2	2	-	2	2	-	2	2	2	2	-
SRAM 2	12	-	-	-	-	12	-	12	-	2	2	-	2	2	-	2	2	2	2	-
SRAM 3	12	-	-	-	-	12	-	12	-	2	2	-	2	2	-	2	2	2	2	-
AHB1 外设	12	2	-	-	-	12	-	12	-	2	2	-	2	2	-	-	-	-	-	-
APB1 外设	125	25	-	-	-	125	-	125	-	25	25	2	25	25	2	-	-	-	-	-
AHB2 外设	-	2	-	-	-	-	-	-	-	2	2	-	2	2	-	-	-	-	-	-
APB2 外设	125	25	-	-	-	125	-	125	-	25	25	2	25	25	2	-	-	-	-	-
AHB4 外设	13	-	-	-	-	13	-	-	-	23	23	-	23	23	-	213	213	213	213	3
APB4 外设	136	-	-	-	-	136	-	-	-	236	236	-	236	236	-	213	213	213	213	36
SRAM4	13	-	-	-	-	13	-	-	-	23	23	-	23	23	-	213	213	213	213	3
备份 RAM	13	-	-	-	-	13	-	-	-	23	23	-	23	23	-	213	213	213	213	3

1. **粗体**字体类型表示 64 位总线，常规字体类型表示 32 位总线。
2. 表格主体中的单元格表示访问可行性、可用性、路径和类型：
   访问可行性和可用性：
   任意数字 = 可以访问，“-” = 不可访问，阴影 = 访问有用/可用
   访问路径：
   D = 直接访问，1 = 通过 AXI 总线矩阵访问，2 = 通过 D2 中的 AHB 总线矩阵访问，3 = 通过 D3 中的 AHB 总线矩阵访问，4 = 通过 D1 中的 AHB/APB 桥访问，5 = 通过 D2 中的 AHB/APB 桥访问，6 = 通过 D3 中的 AHB/APB 桥访问，7 = 通过 Cortex-M7 的 AHBS 总线访问，
   多位数字 = 互连路径按数字顺序经过多个矩阵或/和桥。
   访问类型：
   常规字体 = 32 位，*斜体* = 总线主设备端为 32 位/总线从设备端为 64 位，**粗体** = 64 位

图 1. STM32H7X3 系统架构



2.1.1 总线矩阵

D1 域中的 AXI 总线矩阵

D1 域中的多 AXI 总线矩阵为从多个主设备到多个从设备的并发访问提供保证和仲裁。这样可实现高速外设的高效同步运行。

仲裁采用带 QoS 功能的轮循调度算法。

DTCM 和 ITCM（数据和指令紧密耦合 RAM）通过专用 TCM 总线直接连接到 Cortex-M7 内核。MDMA 控制器可通过 AHBS（特定的 CPU 从设备 AHB）访问 DTCM 和 ITCM。ITCM 由 Cortex-M7 以 CPU 时钟速度（零等待周期）访问。

有关 AXI 互连的更多信息，请参见 *第 5 节：AXI 互连*。

D2 域和 D3 域中的 AHB 总线矩阵

D2 域和 D3 域中的 AHB 总线矩阵为从多个主设备到多个从设备的并发访问提供保证和仲裁。这样可实现高速外设的高效同步运行。

仲裁采用循环调度算法。

2.1.2 总线-总线桥

系统中具有大量的总线-总线桥，用于在不同总线类型的外设之间实现通信。

D1 和 D3 域中的 AHB/APB 总线桥可将 APB3 和 APB4 上的外设分别连接到 AHB3 和 AHB4。D2 域中的 AHB/APB 总线桥可将 APB1 和 APB2 上的外设连接到 AHB1。这些 AHB/APB 总线桥提供完全同步接口，允许 APB 外设依靠与其所连接的 AHB 无关的时钟来运行。

AHB/APB 总线桥还可将 APB1 和 APB2 外设分别连接到 DMA1 和 DMA2 外设总线，而无需通过 AHB1。

AHB/APB 总线桥可将 8 位/16 位 APB 数据转换为 32 位 AHB 数据，具体通过将 8 位/16 位 APB 数据复制到 32 位字的三个高位字节/高位半字来实现。

AXI 总线矩阵在其从总线接口上集成 AHB/AXI 总线桥功能。*图 1* 中标记为 32 位的主设备接口上的 AXI/AHB 总线桥在矩阵外部。

Cortex-M7 CPU 从其 AHBS 从设备 AHB 提供 AHB/TCM 总线（ITCM 和 DTCM 总线）转换，以允许 MDMA 控制器访问 ITCM 和 DTCM。

2.1.3 域间总线

D2-D1 AHB

该 32 位总线将 D2 域连接到 D1 域中的 AXI 总线矩阵。它使得 D2 域中的总线主设备能够访问 D1 域中的资源（总线从设备），以及通过 D1-D3 AHB 间接访问 D3 域中的资源（总线从设备）。

D1-D2 AHB

该 32 位总线将 D1 域连接到 D2 域 AHB 总线矩阵。它使得 D1 域中的总线主设备能够访问 D2 域中的资源（总线从设备）。

D1-D3 AHB

该 32 位总线将 D1 域连接到 D3 域 AHB 总线矩阵。它使得 D1 域中的总线主设备能够访问 D3 域中的资源（总线从设备）。

D2-D3 AHB

该 32 位总线将 D2 域连接到 D3 域 AHB 总线矩阵。它使得 D2 域中的总线主设备能够访问 D3 域中的资源（总线从设备）。

2.1.4 CPU 总线

Cortex®-M7 AXIM 总线

Cortex®-M7 CPU 使用 64 位 AXIM 总线访问所有存储器和外设（ITCM、DTCM、AHB2 外设除外），同时由于寻址不兼容性，AHB1、APB1 和 APB2 外设也排除在外。

该 AXIM 总线将 CPU 连接到 D1 域中的 AXI 总线矩阵。

Cortex®-M7 ITCM 总线

Cortex®-M7 CPU 使用 64 位 ITCM 总线从 ITCM 中获取指令和访问数据。

Cortex®-M7 DTCM 总线

Cortex®-M7 CPU 使用 64 位 DTCM 总线访问 DTCM 中的数据，而且也可以从其中获取指令。

Cortex®-M7 AHBS 总线

Cortex®-M7 CPU 使用 32 位 AHBS 从总线以允许 MDMA 控制器访问 ITCM 和 DTCM。

Cortex®-M7 AHBP 总线

Cortex®-M7 CPU 使用 32 位 AHBP 总线通过 D2 域中的 AHB 总线矩阵访问 AHB1、AHB2、APB1 和 APB2 外设。

2.1.5 总线主设备外设

SDMMC1

SDMMC1 可通过连接到 AXI 总线矩阵的 32 位总线来访问内部 AXI SRAM 和 Flash，以及通过 Quad-SPI 控制器和 FMC 来访问外部存储器。

SDMMC2

SDMMC2 使用 32 位总线，该总线连接到 D2 域中的 AHB 总线矩阵。SDMMC2 可通过系统总线矩阵访问内部 AXI SRAM、SRAM1、SRAM2、SRAM3 和 Flash，以及通过 Quad-SPI 控制器和 FMC 访问外部存储器。

MDMA 控制器

MDMA 控制器使用连接到 AXI 总线矩阵的 64 位总线，用于存储器之间及与外设之间的 DMA 数据传输。通过系统总线矩阵和 Cortex-M7 AHBS 从总线，该控制器可访问所有内部存储器和 AHB1 外设。它还可通过 Quad-SPI 控制器和 FMC 访问外部存储器。

DMA1 和 DMA2 控制器

DMA1 和 DMA2 控制器有两条 32 位总线（存储器总线和外设总线），这两条总线都连接到 D2 域中的 AHB 总线矩阵。

存储器总线可实现存储器之间的 DMA 数据传输。存储器总线可通过系统总线矩阵访问所有内部存储器（ITCM 和 DTCM 除外），以及通过 Quad-SPI 控制器和 FMC 访问外部存储器。

外设总线可实现两个外设之间、两个存储器之间或外设和存储器之间的 DMA 数据传输。外设总线可通过系统总线矩阵访问所有内部存储器（ITCM 和 DTCM 除外），以及通过 Quad-SPI 控制器和 FMC 访问外部存储器，另外还可访问所有 AHB 和 APB 外设。可以直接访问 APB1 和 APB2，而无需通过 AHB1。

BDMA 控制器

BDMA 控制器使用连接到 D3 域中 AHB 总线矩阵的 32 位总线，用于实现两个外设之间、两个存储器之间或外设与存储器之间的 DMA 数据传输。该控制器可通过 D3 域中的 AHB 总线矩阵访问内部 SRAM4、备份 RAM 以及 AHB4 和 APB4 外设。

Chrom-ART Accelerator™ (DMA2D)

DMA2D 图形加速器使用 64 位总线，该总线连接到 AXI 总线矩阵。该加速器可通过系统总线矩阵访问内部 AXI SRAM、SRAM1、SRAM2、SRAM3 和 Flash，以及通过 Quad-SPI 控制器和 FMC 访问外部存储器。

LCD-TFT 控制器 (LTDC)

LCD-TFT 显示控制器 (LTDC) 可通过连接到 AXI 总线矩阵的 64 位总线来访问内部 AXI SRAM 和 Flash，以及通过 Quad-SPI 控制器和 FMC 来访问外部存储器。

以太网 MAC

以太网 MAC 使用 32 位总线，该总线连接到 D2 域中的 AHB 总线矩阵。它可通过系统总线矩阵访问所有内部存储器（ITCM 和 DTCM 除外），以及通过 Quad-SPI 控制器和 FMC 访问外部存储器。

USBHS1 和 USBHS2 外设

USBHS1 和 USBHS2 外设使用 32 位总线，该总线连接到 D2 域中的 AHB 总线矩阵。这两个外设可通过系统总线矩阵访问所有内部存储器（ITCM 和 DTCM 除外），以及通过 Quad-SPI 控制器和 FMC 访问外部存储器。

2.1.6 功能模块的时钟

复位时，将禁用模块（例如，外设）和某些存储器（SRAM、DTCM、ITCM 和 Flash 除外）的时钟。如要操作复位后无时钟的模块，软件必须首先分别通过 RCC_AHBxENR 或 RCC_APBxENR 寄存器将其时钟使能。

2.2 存储器组织结构

2.2.1 前言

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

各字节按小端格式在存储器中编码。字中编号最低的字节被视为该字的最低有效字节，而编号最高的字节被视为最高有效字节。

可寻址的存储空间分为 8 个主要块，每个块为 512 MB。

未分配给片上存储器和外设的所有存储映射区域均视为“保留区”。有关可用存储器和寄存器区域的详细映射，请参见*存储器映射和寄存器边界地址*和外设章节。

2.2.2 存储器映射和寄存器边界地址

有关综合存储器映射图，请参见相应器件的数据手册。

下表给出了器件中可用外设的边界地址。

表 2. 寄存器边界地址

边界地址	外设	总线	寄存器映射
0x58026400 - 0x580267FF	HSEM	AHB4 (D3)	*第 10.4 节：HSEM 寄存器*
0x58026000 - 0x580263FF	ADC3		*第 25.6 节：ADC 寄存器*
0x58025800 - 0x58025BFF	DMAMUX2		*第 17.6 节：DMAMUX1 寄存器*
0x58025400 - 0x580257FF	BDMA		*第 16.4 节：BDMA 寄存器*
0x58024C00 - 0x58024FFF	CRC		*第 21.4 节：CRC 寄存器*
0x58024800 - 0x58024BFF	PWR		*第 6.8 节：PWR 寄存器说明*
0x58024400 - 0x580247FF	RCC		*第 8.7 节：RCC 寄存器说明*
0x58022800 - 0x58022BFF	GPIOK		*第 11.4 节：GPIO 寄存器*
0x58022400 - 0x580227FF	GPIOJ		*第 11.4 节：GPIO 寄存器*
0x58022000 - 0x580223FF	GPIOI		*第 11.4 节：GPIO 寄存器*
0x58021C00 - 0x58021FFF	GPIOH		*第 11.4 节：GPIO 寄存器*
0x58021800 - 0x58021BFF	GPIOG		*第 11.4 节：GPIO 寄存器*
0x58021400 - 0x580217FF	GPIOF		*第 11.4 节：GPIO 寄存器*
0x58021000 - 0x580213FF	GPIOE		*第 11.4 节：GPIO 寄存器*
0x58020C00 - 0x58020FFF	GPIOD		*第 11.4 节：GPIO 寄存器*
0x58020800 - 0x58020BFF	GPIOC		*第 11.4 节：GPIO 寄存器*
0x58020400 - 0x580207FF	GPIOB		*第 11.4 节：GPIO 寄存器*
0x58020000 - 0x580203FF	GPIOA		*第 11.4 节：GPIO 寄存器*

表 2. 寄存器边界地址（续）

边界地址	外设	总线	寄存器映射
0x58006400 - 0x58006BFF	保留	APB4 (D3)	保留
0x58005400 - 0x580057FF	SAI4		*第 51.5 节：SAI 寄存器*
0x58004C00 - 0x58004FFF	保留		保留
0x58004800 - 0x58004BFF	IWDG1		*第 45.4 节：IWDG 寄存器*
0x58004000 - 0x580043FF	RTC 和 BKP 寄存器		*第 46.6 节：RTC 寄存器*
0x58003C00 - 0x58003FFF	VREF		*第 27.3 节：VREFBUF 寄存器*
0x58003800 - 0x58003BFF	COMP1 - COMP2		*第 28.7 节：COMP 寄存器*
0x58003000 - 0x580033FF	LPTIM5		*第 43.6 节：LPTIM 寄存器*
0x58002C00 - 0x58002FFF	LPTIM4		*第 43.6 节：LPTIM 寄存器*
0x58002800 - 0x58002BFF	LPTIM3		*第 43.6 节：LPTIM 寄存器*
0x58002400 - 0x580027FF	LPTIM2		*第 43.6 节：LPTIM 寄存器*
0x58001C00 - 0x58001FFF	I2C4		*第 47.7 节：I2C 寄存器*
0x58001400 - 0x580017FF	SPI6		*第 50.11 节：SPI/I2S 寄存器*
0x58000C00 - 0x58000FFF	LPUART1		*第 49.5 节：LPUART 寄存器*
0x58000400 - 0x580007FF	SYSCFG		*第 12.3 节：SYSCFG 寄存器说明*
0x58000000 - 0x580003FF	EXTI		*第 20.6 节：EXTI 寄存器说明*
0x52008000 - 0x52008FFF	延迟模块 SDMMC1	AHB3 (D1)	*第 24.4 节：DLYB 寄存器*
0x52007000 - 0x52007FFF	SDMMC1		*第 55.8 节：SDMMC 寄存器*
0x52006000 - 0x52006FFF	延迟模块 QUADSPI		*第 24.4 节：DLYB 寄存器*
0x52005000 - 0x52005FFF	QUADSPI 控制寄存器		*第 23.5 节：QUADSPI 寄存器*
0x52004000 - 0x52004FFF	FMC 控制寄存器		*第 22.7.6 节：NOR/PSRAM 控制寄存器*, *第 22.8.7 节：NAND Flash 控制器寄存器*, *第 22.9.5 节：SDRAM 控制寄存器*
0x52003000 - 0x52003FFF	JPEG		*第 33.5 节：JPEG 编解码器寄存器*
0x52002000 - 0x52002FFF	Flash 接口寄存器		*第 3.5 节：Flash 接口寄存器*
0x52001000 - 0x52001FFF	Chrom-ART (DMA2D)		*第 18.6 节：DMA2D 寄存器*
0x52000000 - 0x52000FFF	MDMA		*第 14.5 节：MDMA 寄存器*
0x51000000 - 0x510FFFFF	GPV		*第 5.4 节：AXI 互连寄存器*
0x50003000 - 0x50003FFF	WWDG1	APB3 (D1)	*第 44.4 节：WWDG 寄存器*
0x50001000 - 0x50001FFF	LTDC		*第 32.7 节：LTDC 寄存器*
0x50000000 - 0x50000FFF	保留		
0x48022800 - 0x48022BFF	延迟模块 SDMMC2	AHB2 (D2)	*第 24.4 节：DLYB 寄存器*

表 2. 寄存器边界地址（续）

边界地址	外设	总线	寄存器映射
0x48022400 - 0x480227FF	SDMMC2	AHB2 (D2)	第 55.8 节：SDMMC 寄存器
0x48021800 - 0x48021BFF	RNG		第 34.8 节：RNG 寄存器
0x48021400 - 0x480217FF	HASH		第 36.6 节：散列寄存器
0x48021000 - 0x480213FF	CRYPTO		第 35.6 节：CRYP 寄存器
0x48020000 - 0x480203FF	DCMI		第 31.7 节：DCMI 寄存器说明
0x40080000 - 0x400BFFFF	USB2 OTG FS	AHB1 (D2)	第 57.14 节：OTG_HS 寄存器
0x40040000 - 0x4007FFFF	USB1 OTG HS/FS		第 57.14 节：OTG_HS 寄存器
0x40028000 - 0x400293FF	以太网 MAC		第 58.11 节：以太网 MAC 寄存器
0x40024400 - 0x400247FF	保留		保留
0x40022000 - 0x400223FF	ADC1 - ADC2		第 25.6 节：ADC 寄存器
0x40020800 - 0x40020BFF	DMAMUX1		第 17.6 节：DMAMUX1 寄存器
0x40020400 - 0x400207FF	DMA2		第 15.5 节：DMA 寄存器
0x40020000 - 0x400203FF	DMA1		第 15.5 节：DMA 寄存器
0x40017400 - 0x400177FF	HRTIM	APB2 (D2)	第 37.5 节：HRTIM 寄存器
0x40017000 - 0x400173FF	DFSDM1		第 30.7 节：DFSDM 通道 y 寄存器 (y=0..7)， 第 30.8 节：DFSDM 滤波器 x 模块寄存器 (x=0..3)
0x40016000 - 0x400163FF	SAI3		第 51.5 节：SAI 寄存器
0x40015C00 - 0x40015FFF	SAI2		第 51.5 节：SAI 寄存器
0x40015800 - 0x40015BFF	SAI1		第 51.5 节：SAI 寄存器
0x40015000 - 0x400153FF	SPI5		第 50.11 节：SPI/I2S 寄存器
0x40014800 - 0x40014BFF	TIM17		第 41.6 节：TIM16/TIM17 寄存器
0x40014400 - 0x400147FF	TIM16		第 41.6 节：TIM16/TIM17 寄存器
0x40014000 - 0x400143FF	TIM15		第 41.5 节：TIM15 寄存器
0x40013400 - 0x400137FF	SPI4		第 50.11 节：SPI/I2S 寄存器
0x40013000 - 0x400133FF	SPI1/I2S1		第 50.11 节：SPI/I2S 寄存器
0x40011400 - 0x400117FF	USART6		第 48.7 节：USART 寄存器
0x40011000 - 0x400113FF	USART1		第 48.7 节：USART 寄存器
0x40010400 - 0x400107FF	TIM8		第 38.4 节：TIM1/TIM8 寄存器
0x40010000 - 0x400103FF	TIM1		第 38.4 节：TIM1/TIM8 寄存器

表 2. 寄存器边界地址（续）

边界地址	外设	总线	寄存器映射
0x4000AC00 - 0x4000D3FF	CAN消息RAM	APB1 (D2)	*第 56.4 节：FDCAN 寄存器*
0x4000A800 - 0x4000ABFF	CAN, CCU		*第 56.4 节：FDCAN 寄存器*
0x4000A400 - 0x4000A7FF	FDCAN2		*第 56.4 节：FDCAN 寄存器*
0x4000A000 - 0x4000A3FF	FDCAN1		*第 56.4 节：FDCAN 寄存器*
0x40009400 - 0x400097FF	MDIOS		*第 54.4 节：MDIOS 寄存器*
0x40009000 - 0x400093FF	OPAMP		*第 29.6 节：OPAMP 寄存器*
0x40008800 - 0x40008BFF	SWPMI		*第 53.6 节：SWPMI 寄存器*
0x40008400 - 0x400087FF	CRS		*第 9.7 节：CRS 寄存器*
0x40007C00 - 0x40007FFF	UART8		*第 48.7 节：USART 寄存器*
0x40007800 - 0x40007BFF	UART7		*第 48.7 节：USART 寄存器*
0x40007400 - 0x400077FF	DAC1		*第 26.6 节：DAC 寄存器*
0x40006C00 - 0x40006FFF	HDMI-CEC		*第 59.7 节：HDMI-CEC 寄存器*
0x40005C00 - 0x40005FFF	I2C3		*第 47.7 节：I2C 寄存器*
0x40005800 - 0x40005BFF	I2C2		*第 47.7 节：I2C 寄存器*
0x40005400 - 0x400057FF	I2C1		*第 47.7 节：I2C 寄存器*
0x40005000 - 0x400053FF	UART5		*第 48.7 节：USART 寄存器*
0x40004C00 - 0x40004FFF	UART4		*第 48.7 节：USART 寄存器*
0x40004800 - 0x40004BFF	USART3		*第 48.7 节：USART 寄存器*
0x40004400 - 0x400047FF	USART2		*第 48.7 节：USART 寄存器*
0x40004000 - 0x400043FF	SPDIFRX		*第 52.5 节：SPDIFRX 接口寄存器*
0x40003C00 - 0x40003FFF	SPI3 / I2S3		*第 50.11 节：SPI/I2S 寄存器*
0x40003800 - 0x40003BFF	SPI2 / I2S2		*第 50.11 节：SPI/I2S 寄存器*
0x40002C00 - 0x40002FFF	保留		保留
0x40002400 - 0x400027FF	LPTIM1		*第 43.6 节：LPTIM 寄存器*
0x40002000 - 0x400023FF	TIM14		*第 39.4 节：TIM2/TIM3/TIM4/TIM5 寄存器*
0x40001C00 - 0x40001FFF	TIM13		*第 39.4 节：TIM2/TIM3/TIM4/TIM5 寄存器*
0x40001800 - 0x40001BFF	TIM12		*第 39.4 节：TIM2/TIM3/TIM4/TIM5 寄存器*
0x40001400 - 0x400017FF	TIM7		*第 42.4 节：TIM6/TIM7 寄存器*
0x40001000 - 0x400013FF	TIM6		*第 42.4 节：TIM6/TIM7 寄存器*
0x40000C00 - 0x40000FFF	TIM5		*第 39.4 节：TIM2/TIM3/TIM4/TIM5 寄存器*
0x40000800 - 0x40000BFF	TIM4		*第 39.4 节：TIM2/TIM3/TIM4/TIM5 寄存器*
0x40000400 - 0x400007FF	TIM3		*第 39.4 节：TIM2/TIM3/TIM4/TIM5 寄存器*
0x40000000 - 0x400003FF	TIM2		*第 39.4 节：TIM2/TIM3/TIM4/TIM5 寄存器*

2.3 内部 SRAM

STM32H7x3 器件具有：

- 多达 864 KB 的系统 SRAM
- 128 KB 的数据 TCM RAM
- 64 KB 的指令 TCM RAM
- 4 K 字节的备份 SRAM

嵌入式系统 SRAM 最多可分为五个块：

- AXI SRAM（D1 域）：
  - 映射到地址 0x2400 0000 的 AXI SRAM，可供除 BDMA 外的所有系统主设备通过 D1 域 AXI 总线矩阵访问
- AHB SRAM（D2 域）：
  - 映射到地址 0x3000 0000 的 AHB SRAM1，可供除 BDMA 外的所有系统主设备通过 D2 域 AHB 矩阵访问
  - 映射到地址 0x3002 0000 的 AHB SRAM2，可供除 BDMA 外的所有系统主设备通过 D2 域 AHB 矩阵访问
  - 映射到地址 0x3004 0000 的 AHB SRAM3，可供除 BDMA 外的所有系统主设备通过 D2 域 AHB 矩阵访问
- AHB SRAM（D3 域）：
  - 映射到地址 0x3800 0000 的 AHB SRAM4，可供大多数系统主设备通过 D3 域 AHB 矩阵访问

系统 AHB SRAM 可按字节、半字（16 位单元）或全字（32 位单元）访问，而系统 AXI SRAM 可按字节、半字、全字或双字（64 位单位）访问。这些存储器可在最高系统时钟频率下以 0 等待周期寻址。

当 AHB 主设备对 SRAM 中的一部分进行读/写访问的同时，以太网 MAC 或 USB OTG HS 外设可以访问 SRAM 的另一部分。例如，以太网 MAC 访问 SRAM2，同时 CPU 访问 SRAM1。

TCM SRAM 专用于 Cortex®-M7：

- TCM 接口上的 DTCM-RAM 映射到地址 0x2000 0000，可供 Cortex®-M7 访问，并且可供 MDMA 通过 Cortex®-M7 CPU 的 AHBS 从总线访问。
- TCM 接口上的 ITCM-RAM 映射到地址 0x0000 0000，可供 Cortex®-M7 访问，并且可供 MDMA 通过 Cortex®-M7 CPU 的 AHBS 从总线访问。

备份 RAM 映射到地址 0x3880 0000，可供大多数系统主设备通过 D3 域的 AHB 矩阵访问。

误码校正 (ECC)

SRAM 数据受 ECC 保护：

- 每个 32 位字增加 7 个 ECC 位。
- 对于 AXI-SRAM 和 ITCM-RAM，每个 64 位字增加 8 个 ECC 位。

ECC 机制基于 SECDED 算法。它支持单、双错误校正。

2.4 Flash 概述

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

Flash 结构如下：

- 两个主存储器分为多个扇区。
- 信息块：
  - 系统存储器，器件在系统存储器自举模式下从该存储器启动。
  - Option 字节，用于配置读写保护、BOR 级别、软件/硬件看门狗以及器件处于待机或停止模式下的复位。

更多详细信息，请参见*第 3 节：嵌入式 Flash (FLASH)*。

2.5 启动配置

在 STM32H7x3 中，通过 BOOT 引脚及 BOOT_ADD0/BOOT_ADD1 的组合配置，可以让系统从两个不同的区域启动基址如*表 3* 所示。

表 3. 自举模式

自举模式选择		自举区域
BOOT	**自举地址选项字节**	
0	BOOT_ADD0[15:0]	启动地址由用户选择字节 BOOT_ADD0[15:0] ST 编程值定义：Flash 位于 0x0800 0000
1	BOOT_ADD1[15:0]	启动地址由用户选择字节 BOOT_ADD1[15:0] ST 编程值定义：系统自举程序位于 0x1FF0 0000

复位后，在 SYSCLK 的第四个上升沿锁存 BOOT 引脚的值。复位后，用户可以使用复用于 BOOT 引脚的其他功能而不影响系统运行。。

器件退出待机模式时，还会对 BOOT 引脚重新采样。因此，当器件处于待机模式时，这些引脚必须保持所需的自举模式配置。

启动延迟后，系统会在释放处理器复位前选择自举区域。

BOOT_ADD0 和 BOOT_ADD1 地址选项字节允许将自举存储器地址配置为从 0x0000 0000 到 0x3FFF 0000 的任意存储器地址，包括：

- 所有 Flash 地址空间
- 所有 RAM 地址空间：ITCM、DTCM RAM 和 SRAM
- TCM-RAM

可在复位后修改 BOOT_ADD0/BOOT_ADD1 选项字节以在下次复位后从任何其它自举地址自举。

如果编程的自举存储器地址位于存储器映射区域或保留区域之外，则按如下方式编程默认自举获取地址：

- 自举地址 0：位于 0x0800 0000 的 FLASH
- 自举地址 1：位于 0x0000 0000 的 ITCM-RAM

当 Flash 的保护级别被配置为级别 2 之后，只能从 Flash 自举。如果 BOOT_ADD0/BOOT_ADD1 选项字节中自举地址被配置为位于存储器范围之外或属于 RAM 地址范围，则系统只能从地址 0x0800 0000 上的 Flash 开始执行。

内部 Bootloader

内部 Bootloader 代码位于系统存储器中， 在芯片生产期间由 ST 编程。它用于通过以下串行接口重新编程 Flash：

- 引脚 PA9/PA10 和 PB14/PB15 上的 USART1、引脚 PA3/PA2 上的 USART2 和引脚 PB10/PB11 上的 USART3。
- 引脚 PB6/PB9 上的 I2C1、引脚 PF0/PF1 上的 I2C2 和引脚 PA8/PC9 上的 I2C3。
- 引脚 PA11/PA12 上处于设备模式 (DFU) 下的 USB OTG FS。
- 引脚 PA7/PA6/PA5/PA4 上的 SPI1、引脚 PI3/PI2/PI1/PI0 上的 SPI2、引脚 PC12/PC11/PC10/PA15 上的 SPI3 和引脚 PE14/PE13/PE12/PE11 上的 SPI4。

有关详细信息，请参见应用笔记 AN2606。

3 嵌入式 Flash (FLASH)

3.1 前言

Flash 接口可管理任何主设备的 AXI 对 Flash 进行的访问。该接口可针对 Flash 执行读取、编程和擦除操作，并实施读取和编程/擦除保护机制。

Flash 接口还管理着复位时进行的选项字节加载以及动态选项字节更改。

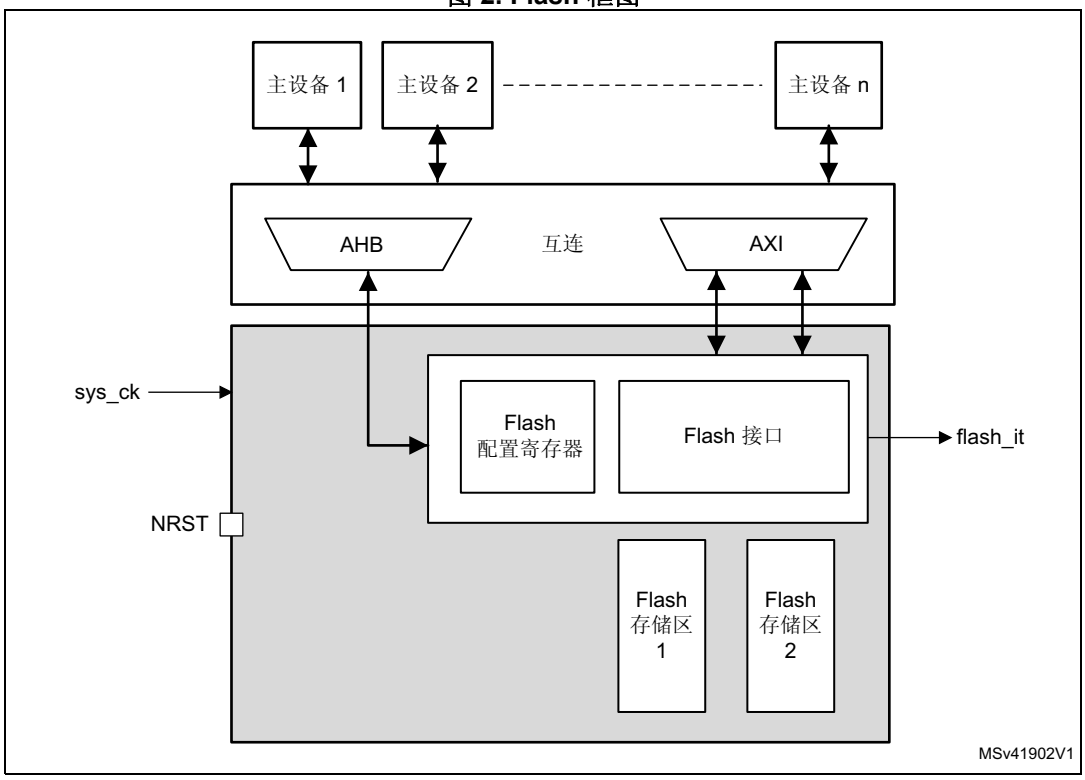
3.2 Flash 主要特性

- 存储器容量最高可达到 2 MB
- 误码校正 (Error Code Correction, ECC)：256 位 Flash 字 10 个 ECC 位
- 双字、字、半字和字节读操作
- 按 256 位进行 Flash 编程
- 双字、字、半字和字节写操作
- 扇区擦除、存储区擦除和批量擦除
- 双存储区构成支持同时操作：可在两个存储区中并行执行两个读取/编程/擦除操作
- 存储区交换：每个存储区的用户 Flash 的地址映射可进行交换。
- 增强型安全功能
  - 读保护 (RDP)
  - 扇区写保护
  - 2 个 PCROP 保护区（每个存储区 1 个）（仅执行存储器）
- 用户 Flash 中有 2 个安全区域（每个存储区 1 个）

3.3 Flash 功能说明

3.3.1 框图

图 2. Flash 框图



3.3.2 引脚和内部信号

表 4 列出了连接到封装引脚或焊球的 FLASH 输入与输出信号，*表 5* 则列出了内部 FLASH 信号。

表 4. 连接到封装引脚或焊球的 FLASH 输入/输出信号

信号名称	信号类型	说明
NRST	数字输入	外部复位信号。

表 5. FLASH 内部输入/输出信号

信号名称	信号类型	说明
sys_ck	数字输入	系统时钟
flash_it	数字输出	嵌入式 Flash 中断

3.3.3 Flash 架构

Flash 为 266 位的 Flash 字存储器，用于存储代码和数据常量。每个字包括：

- 一个 Flash 字（8 个字、32 个字节或 256 位）。
- 10 个 ECC 位。

Flash 分为两个独立的存储区。每个存储区的构成如下：

- 1 MB 的用户 Flash 块，包括八个 128 KB 的用户扇区（4 K Flash 字）。
- 128 KB 的系统 Flash，器件可从该 Flash 启动：

  此区域包含根安全服务 (root secure services, RSS) 和自举程序，两者分别用于通过 USART、USB (DFU)、I2C、SPI 或以太网接口进行安全或非安全 Flash 编程。系统 Flash 留给意法半导体使用。它由意法半导体在器件制造期间编程并加以保护以防止误编程/误擦除操作。更多详细信息，请参见 *http://www.st.com* 上提供的应用笔记 AN2606《STM32 微控制器系统 Flash 启动模式》。
- 2 KB（64 个 Flash 字）的用户选项字节，用于进行用户配置：

  此区域仅在存储区 1 中可用。与用户 Flash 和系统 Flash 不同，该区域并未映射到任何存储器地址，并且仅可通过 Flash 寄存器接口进行访问。

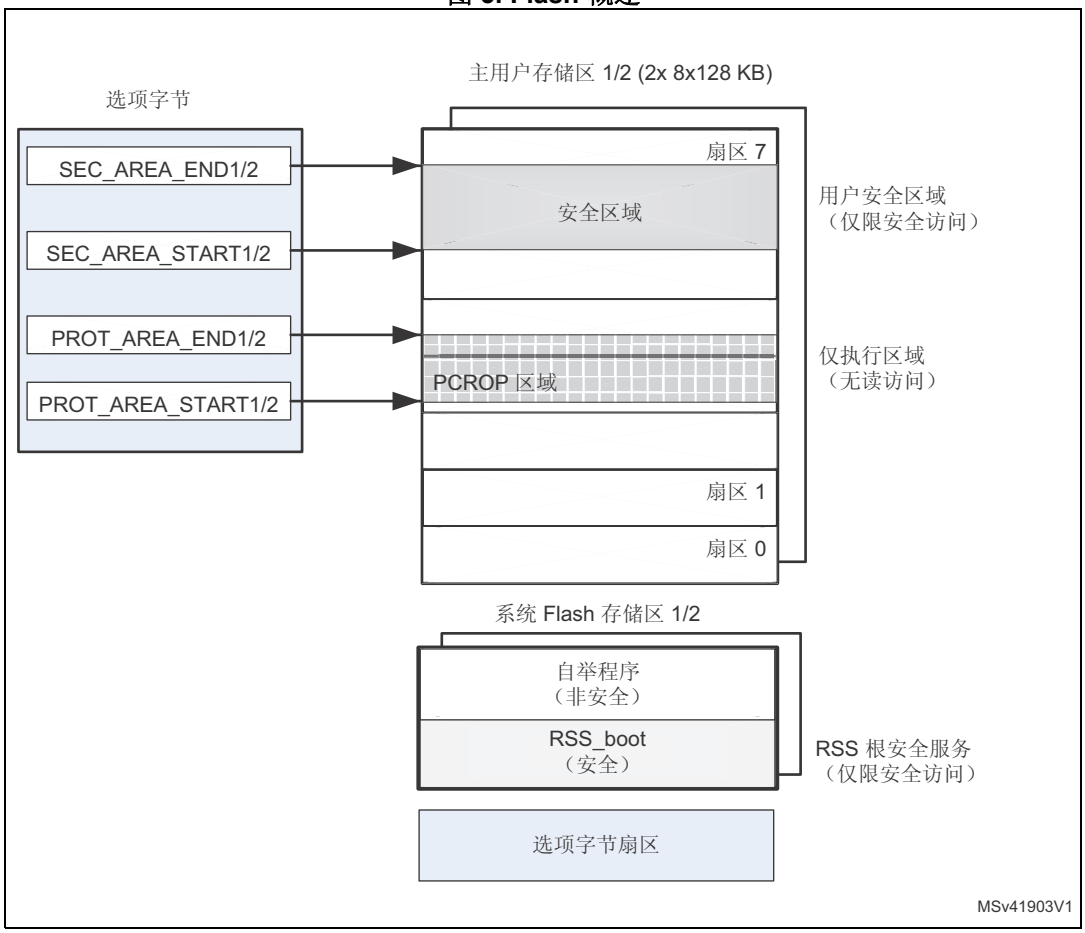
图 2 显示的是 Flash 高级框图，*表 6* 则介绍了 Flash 的构成。

图 3 概述了 Flash 映射以及保护机制。可为每个存储区定义 PCROP 和安全区域。这些受保护区域的属性在*第 3.3.12 节：保护机制*中进行了详细介绍，而系统 Flash 中存储的安全服务则在*第 4 节：安全存储管理*中加以介绍。

表 6. Flash 模块 - 1 MB 双存储区构成

Flash 区域		起始地址	结束地址	大小（字节）	区域名称
存储区域 1	用户 Flash	0x0800 0000	0x0801 FFFF	128 K	扇区 0
		0x0802 0000	0x0803 FFFF	128 K	扇区 1
		...	...	...	...
		0x080E 0000	0x080F FFFF	128 K	扇区 7
	系统 Flash	0x1FF0 0000	0x1FF1 FFFF	128 K	系统 Flash
存储区域 2	用户 Flash	0x0810 0000	0x0811 FFFF	128 K	扇区 0
		0x0812 0000	0x0813 FFFF	128 K	扇区 1
		...	...	...	...
		0x081E 0000	0x081F FFFF	128 K	扇区 7
	系统 Flash	0x1FF4 0000	0x1FF5 FFFF	128 K	系统 Flash

图 3. Flash 概述



3.3.4 Flash 读操作

读协议

Flash 接口支持以下访问类型：

- 双字（64 位）
- 单字（32 位）
- 半字（16 位）
- 字节（8 位）

从 Flash 读取数据时，Flash 接口时钟必须使能并处于运行状态。为了确保正确进行 Flash 接口读操作，必须根据 Flash 接口频率在 FLASH_ACR 寄存器中正确配置等待状态 (LATENCY) 的数量（请参见*读访问延迟*一节）。

如果等待状态的数量不正确，则可能导致读取的数值不正确（例如，未编程足够多的等待状态时）或者访问时间过长（例如，编程了过多的等待状态时）。

按照 Flash 接口接收操作的顺序执行操作。

Flash 接口设计为一次只能在给定的存储区上执行一个读取、编程或擦除操作。以交换用户 Flash 为例。第一个 AXI 接口请求对存储区 1 系统 Flash (ICP) 执行读操作，而第二个 AXI 接口请求对同一存储区的用户 Flash 执行读操作或取指操作。在这种情况下，会进行硬件仲裁，将接受两个读取请求，但会按顺序处理。

如果多个读取/编程/擦除操作针对的是不同存储区，则可同时执行。如果请求读操作的同时正在对同一存储区执行编程/擦除操作，则会缓冲读取命令，并在编程/擦除操作完成后执行读取命令。除非读取命令 FIFO 已满（已有 3 个读操作挂起），否则系统不会停止工作。

读取顺序

Flash 接口使用双 AXI 总线接口进行代码/数据访问，并使用 AHB 接口进行 Flash 接口配置。AXI 接口可同时请求读取/编程操作。

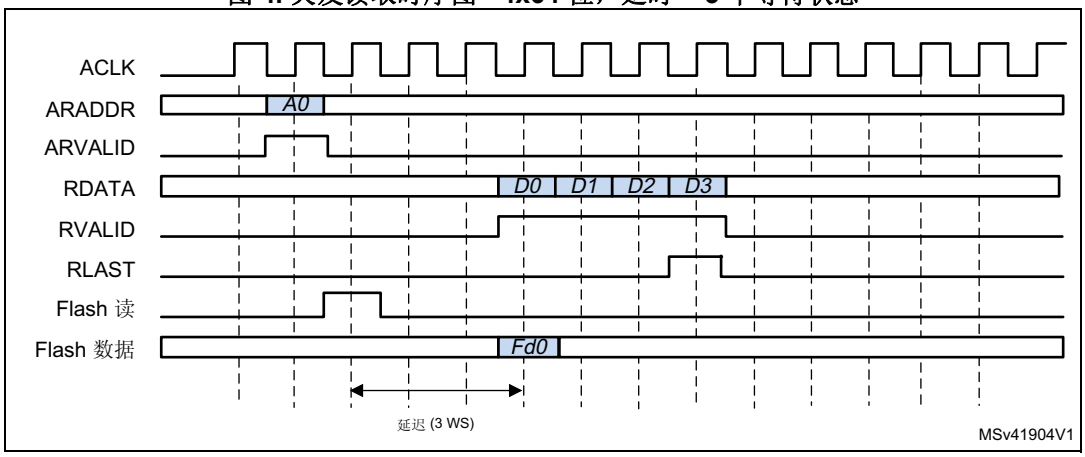
读取机制如下：

- 读取命令缓冲区深度固定设为 3 个请求。当缓冲区已满时（3 个读取请求在缓冲区中排队），如果有新的读取请求，总线接口便会停止工作，因此主机也会停止工作。
- 当前读取事务的最后一个数据从 Flash 传送到 Flash 接口内的读取数据缓冲区后，Flash 接口会立即释放读取命令队列缓冲区中的一个请求。
- 如果系统读取请求要读取的数据不在读取缓冲区中，将触发 Flash 读操作。读取到的数据随后将缓冲到读取数据缓冲区中。
- 如果多个连续读取访问请求的数据属于同一 Flash 数据字（256 位），则会直接从当前数据读取缓冲区中读取数据，不会触发额外的 Flash 读操作。

图 4 和*图 5* 分别显示了读取四个和八个双字数据的突发读取事务的时序图。延时配置为三个等待状态：

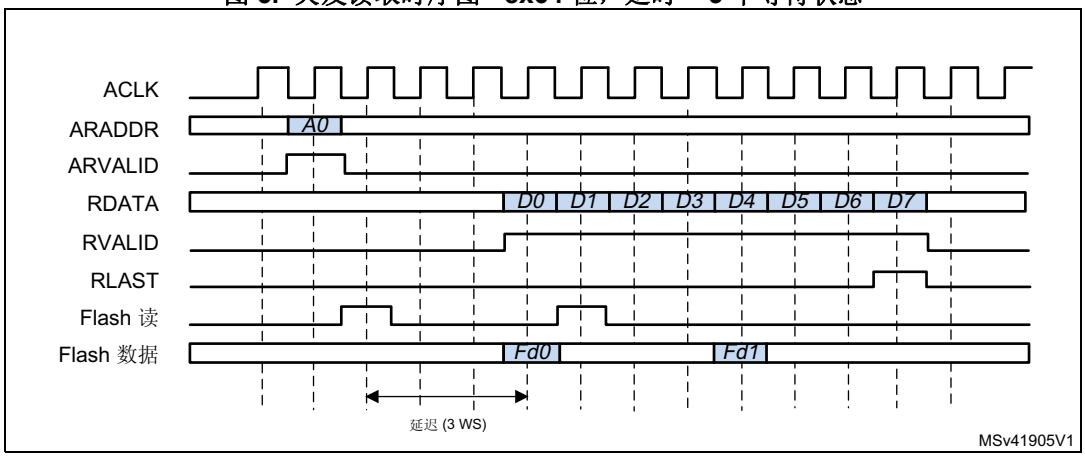
1. 第一个延时（三个等待状态）后，第一个数据可用。
2. 数据出现在当前读取缓冲区中后，Flash 接口便可执行读取命令队列中的下一条命令。
3. 第二个延时对应于下一次 Flash 字读取访问，它由当前缓冲区读操作屏蔽，该读操作可连续进行四次双字读取， 这相当于 0 等待状态执行顺序。

图 4. 突发读取时序图 - 4x64 位，延时 = 3 个等待状态



1. ACLK、ARADDR、ARVALID、RDATA、RVALID 和 RLAST 是 AXI 总线信号。
   Flash Read 和 Flash Data 是 Flash 接口信号。

图 5. 突发读取时序图 - 8x64 位，延时 = 3 个等待状态



1. ACLK、ARADDR、ARVALID、RDATA、RVALID 和 RLAST 是 AXI 总线信号。
   Flash Read 和 Flash Data 是 Flash 接口信号。

读访问延迟

为了正确地从 Flash 中读取数据，必须根据总线时钟频率 (aclk) 和器件 V_{CORE} 的内部电压范围在 Flash 访问控制寄存器 (FLASH_ACR) 中正确编程等待状态数 (LATENCY)。*表 7* 显示了等待状态数、总线时钟频率和 V_{CORE} 范围之间的对应关系。

表 7. 根据总线频率 (ACLK) 和 V_{CORE} 范围确定的等待状态数

V_{CORE} 范围		等待状态数	最大频率
VOS1 级别	1.15 V - 1.26 V	0	70 MHz
		1	140 MHz
		2	210 MHz
VOS2 级别	1.05 V - 1.15 V	0	55 MHz
		1	110 MHz
		2	16 MHz
		3	220 MHz
VOS3 级别	0.95 V - 1.05 V	0	45 MHz
		1	90 MHz
		2	135 MHz
		3	180 MHz
		4	225 MHz

上电后，使用的时钟是 HSI (64 MHz)，并且默认在 FLASH_ACR 寄存器中配置 0x7 个等待状态。要更改总线频率，必须应用*调整总线频率*一节中介绍的软件顺序来调节访问 Flash 所需的等待状态数。

调整总线频率

- 增大总线频率
  a) 如有必要，可将正确的等待状态数（请参见*表 7*）编程到 FLASH_ACR 寄存器中的 LATENCY 位。
  b) 通过读取 FLASH_ACR 寄存器，检查新的等待周期数是否设置成功
  c) 修改 RCC_CFGR 寄存器中的 Flash 接口时钟源和/或总线时钟预分频比。
  d) 通过读取 RCC_CFRG 寄存器中相应的时钟源状态和/或总线预分频值，检查新的 Flash 接口时钟源和/或新的 Flash 接口时钟预分频值是否设置成功。
- 减小总线频率
  a) 修改 RCC_CFGR 寄存器中的 Flash 接口时钟源和/或总线时钟预分频比。
  b) 通过读取 RCC_CFGR 寄存器中相应的 Flash 接口时钟源状态和/或总线预分频值，检查新的 Flash 接口时钟源和/或新的总线时钟预分频值是否设置成功。
  c) 如有必要，可将正确的等待状态数（请参见*表 7*）编程到 FLASH_ACR 寄存器中的 LATENCY 位。
  d) 通过读取 FLASH_ACR 寄存器检查新的等待状态数是否设置成功。

3.3.5 误码校正 (ECC)

Flash 中的数据为 266 位字：每个 256 位的 Flash 字增加了 10 个 ECC 位。ECC 机制基于 SECDED 算法。它支持：

- 单错误校正
- 双重错误检测

检测到错误并进行校正后，FLASH_SR1/2 寄存器中的 SNECCERR1/2 标志会置 1。如果 FLASH_CR1/2 寄存器中的 SNECCERRIE 位置 1，将生成中断。

如果检测到两个错误，FLASH_SR1/2 寄存器中的 DBECCERR1/2 标志会置 1，并会生成总线错误。在这种情况下，将不会对接收到的数据进行校正。如果 FLASH_CR1/2 寄存器中的 DBECCERRIE1/2 位置 1，将生成中断。

如果检测到 ECC 错误，出错 Flash 字的地址会保存到 FLASH_ECC_FA1/2R 寄存器中。如果连续检测到错误，将只会存储第一个错误对应的地址。当生成错误的相关标志复位后，该寄存器会自动清空。

3.3.6 循环冗余校验模块

Flash 接口嵌入了循环冗余校验硬件模块。此模块用于校验 Flash 区域内容的完整性。该区域可按扇区定义，也可按起始/结束地址定义。

每次只能在存储区 1 或存储区 2 上启动一个 CRC 校验操作。

CRC 操作与选项字节更改操作同时进行。这意味着如果在进行选项字节更改的同时请求执行 CRC 操作，则必须先完成选项字节更改操作，然后才能执行 CRC 操作，反之亦然。

CRC 硬件模块使能后，会按 4、16、64 或 256 Flash 字块的形式处理 Flash 数据。CRC 计算使用 CRC-32（以太网）多项式 0x4C11DB7。

$$X^{32} + X^{26} + X^{23} + X^{22} + X^{16} + X^{12} + X^{11} + X^{10} + X^{8} + X^{7} + X^{5} + X^{4} + X^{2} + X + 1$$

Flash 接口会在内部发出 4、16、64 或 256 连续 Flash 字读取访问请求。这些事务会与其他 AXI 读取请求一起在读取命令队列中排队，从而可避免拒绝 AXI 读取命令。队列命令缓冲区仅可包含一个 CRC 命令。

在存储区 1/2 中配置 CRC 操作的建议顺序如下：

1. 通过将 FLASH_CR1/2 中的 CRC_EN 位置 1 使能 CRC 功能。
2. 在 FLASH_CRCCR1/2 的 CRC_BURST 字段中编程所需数据大小。
3. 定义必须对其进行 CRC 计算的 Flash 区域。可采用两种解决方案：
   - 分别编程 FLASH_CRCSADD1/2R 和 FLASH_CRCEADD1/2R，定义区域起始地址和结束地址。
   - 或者将 FLASH_CRCCR1/2 中的 CRC_BY_SECT 位置 1，或连续在 FLASH_CRCCR1/2 寄存器的 CRC_SECT 字段中编程目标扇区数，以选择目标扇区。每次 CRC_SECT 编程完毕后都将 ADD_SECT 位置 1。
4. 将 START_CRC 位置 1，启动 CRC 操作。
5. 等待 CRC_BUSY 标志复位。
6. 获取 FLASH_CRCDATAR 寄存器中的 CRC 结果。

通过将 FLASH_CRCCR1/2 寄存器中的 ALL_BANK 位置 1，可为整个存储区计算 CRC。

注： *在 PCROP 或受到安全保护的用户 Flash 区域上运行 CRC 可能会改变预期的 CRC 值。*

3.3.7 Flash 编程/擦除操作

编程/擦除操作概述

Flash 接口可执行单错误校正和双重错误检测（请参见*第 3.3.5 节：误码校正 (ECC)*）。ECC 算法设计为使全“0”的数据生成全“0”的 ECC 码，并使全“1”的数据生成全“1”的 ECC 码。

由于 10 位 ECC 码关联到每个 256 位数据 Flash 字，因此只支持按 256 位进行的写操作。

要恢复初始状态，需要对整个扇区执行擦除操作。

Flash 接口支持多种编程操作：

- 写入用户扇区
- 擦除用户扇区
- 擦除存储区 1、存储区 2 或同时擦除两个存储区
- 更改用户选项字节

通过 AXI 接口发出的写访问可视为可缓冲、但不可缓存，不同之处在于不能回读 Flash 接口内的写缓冲区。

嵌入式 Flash 可采用在线编程或在应用中编程两种方式。

写缓冲区的结构和提示

Flash 接口写队列缓冲区可包含 2 个请求。因此，如果有多个写访问请求发送到 Flash 接口，写队列缓冲区被占满之前，会接受这些请求。缓冲区已满后，Flash 接口会使 AXI 总线停止工作。

按照 Flash 接口接收操作的顺序执行操作。除非写命令队列和写缓冲区已满（2 个写操作正在进行中），否则系统不会停止运行。

正常写操作期间，PG1/2 位必须置“1”（请参见*使能写操作*一节）。写访问随后必须填入 256 位写缓冲区中。缓冲区已满后，其内容会自动传送到写队列缓冲区中。Flash 就绪、并且之前请求的操作执行完毕后，会立即执行有效的写操作。

如果写缓冲区部分填满，那么在完全填满整个写缓冲区之前，可强制执行写操作， 具体可通过将 FLASH_CR1/2 中的 FW1/2 位置 1 来实现。在这种特殊情况下，未写入的位会自动设置为“高”值。如果写缓冲区中没有任何位设置为“低”值，则 FW1/2 位未起作用。

建议不要覆盖非原始 Flash 字， 否则可能造成 ECC 码不一致。Flash 接口可系统地报告 ECC 错误。可支持的有效方案是用全零数据进行覆盖。在这种情况下，相应的 ECC 码也会变为全零， 因此读取过程中不会检测到 ECC 错误。

写缓冲区还支持递增突发写访问。仅当突发写访问不超过 32 字节对齐地址（Flash 字 = 256 位）时，才允许进行回卷。

配置编程参数

用户应用程序必须先配置编程参数，然后才能执行编程/擦除操作：

1. 解锁 Flash 配置寄存器 (FLASH_CR1/2) 并进行编程：

   Flash 接口使用两个 Flash 控制寄存器：FLASH_CR1（用于存储区 1）和 FLASH_CR2（用于存储区 2）。复位后，会对这两个寄存器进行写保护，以免因电气干扰等原因对 Flash 执行意外操作。FLASH_CR1/2 寄存器的解锁顺序如下：

   a) 在 FLASH_KEYR1/2 寄存器中写入 KEY1 = 0x45670123。

   b) 在 FLASH_KEYR1/2 寄存器中写入 KEY2 = 0xCDEF89AB。

   FLASH_CR1/2 寄存器中的 LOCK1/2 位会自动清空。如果操作顺序不正确，会锁定 FLASH_CR1/2，下次系统复位才会解锁。这种情况下会生成总线错误。

   此外，如果执行以下操作，FLASH_CR1/2 会保持锁定状态，并会生成总线错误：

   – 编程第三个密钥值，
   – 在 FLASH_CR1/2 完全解锁（KEY1 已编程，但 KEY2 尚未编程）之前对与 FLASH_KEYR1/2 同属于一个存储区的另一寄存器执行写操作。
   – 对 KEY1 或 KEY2 写入不足一个 Flash 字的数据。

   要再次锁定 FLASH_CR1/2 寄存器，需通过软件将相应的 LOCK1/2 位设置为“1”。

注： 仅可在字级别访问 FLASH_KEYR1/2。

2. 解锁 Flash 选项配置寄存器 (FLASH_OPTCR)：

   要修改选项字节，需解锁 FLASH_OPTCR 寄存器。由于选项字节是两个存储区共用的，因此 Flash 接口中只存在一个 FLASH_OPTCR 寄存器，并且 FLASH_OPTCR 不受存储区交换的影响。寄存器有别名，因此能够在两个不同地址处访问：

   – 0x018
   – 0x118

   复位后，OPTLOCK 位会置“1”，FLASH_OPTCR 会锁定。FLASH_OPTCR 寄存器的解锁顺序如下：

   a) 将 0x08192A3B 写入到 FLASH_OPTKEYR 寄存器中的 OPTKEY1。

   b) 将 0x4C5D6E7F 写入到 FLASH_OPTKEYR 寄存器中的 OPTKEY2。

   执行该顺序后，OPTLOCK 位会清零，FLASH_OPTCR 会解锁。如果密钥操作顺序不正确，会在下一次系统复位前锁定 FLASH_OPTCR 并生成总线错误。

   此外，如果执行以下操作，FLASH_OPTCR 会保持锁定状态，并会生成总线错误：

   – 编程第三个密钥值，
   – 在 FLASH_OPTCR 完全解锁（OPTKEY1 已编程，但 OPTKEY2 尚未编程）之前对另一寄存器执行写操作。
   – 对 OPTKEY1 或 OPTKEY2 写入不足一个 Flash 字的数据。

   要再次锁定 FLASH_OPTCR 寄存器，需通过软件将 OPTLOCK 位设置为“1”。

注： *仅可在字级别访问 FLASH_OPTCR。*

3. 设置编程并行位数：

   并行位数是指写操作期间可在单触发中写为“0”的最大位数。扇区和存储区擦除期间也会使用编程并行位数。

   编程并行位数没有硬件限制。用户可根据应用程序需求选择不同的并行位数：并行位数越小，编程操作过程中的峰值功耗越低，但执行时间也越长。

   并行位数是通过 FLASH_CR1/2 中的 PSIZE1/2 位配置的。可为存储区 1 和存储区 2 定义两个不同值（请参见*表 8*）。

表 8. 并行位数参数

PSIZE1/2	并行位数
00	字节（8 位）
01	半字（16 位）
10	字（32 位）
11	双字（64 位）

4. 设置编程延时：

   编程操作时间限制取决于直接影响性能的 Flash 接口频率。如果时间限制过严，Flash 将无法正常工作，如果限制过松，编程速度将不会达到最佳水平。

   因此，用户必须通过 FLASH_ACR 寄存器中的 WRHIGHFREQ 参数调整最佳编程延时。该配置寄存器由两个存储区共用。有关 Flash 接口频率对应的建议编程延时，请参见*表 9*。

表 9. 编程速度

WRHIGHFREQ 建议值	Flash 接口频率 (MHz)
00	85
01	185

   修改 WRHIGHFREQ 后进行回读，以检查新值是否设置成功。

注： *如果修改编程速度的同时正在执行编程/擦除操作，则仅会在当前操作完成后设置新值。*

使能写操作

进行编程之前，必须将 FLASH_CR1/2 中的 PG1/2 位置 1。FLASH_CR1/2 必须已提前解锁。PG1/2 位置为“0”时提出的任何写访问请求都会被拒绝。这种情况下，不会在总线上生成错误，但 PGSERR1/2 标志会置 1。

检查写保护

在对 Flash 进行有效写操作之前，会在写队列缓冲区的输出处检查写事务目标的保护属性。如果 Flash 接口接受 AXI 写请求，则不会执行检查。如果检测到写保护违例，则会取消写操作，并且 FLASH_SR1/2 寄存器中与写保护违例相关的标志会置 1。如果写操作有效，10 位 ECC 码会连接到 256 位数据，对 Flash 的写操作会有效执行。

执行新的编程/擦除操作之前，必须将写保护标志清零。

写入状态忙标志

FLASH_SR1/2 中三个不同的状态标志可供每个存储区使用。这些标志指示正在进行的写操作的状态：

- **BSY1/2**：此位指示正在对 Flash 进行有效的写入、擦除或选项字节更改操作。
- **QW1/2**：此位指示编程、擦除或选项字节更改操作挂起。写操作完成之前，此位会保持高电平。此位取代了 BSY1/2 状态位。
- **WBNE1/2**：此位指示写缓冲区不为空。写入命令排队后，此位会立即复位。

注： *由于写缓冲区对应于与 32 字节地址对齐的 256 位 Flash 字，因此如果正在进行某一写操作，则不能开始写入其他新的 Flash 字。*

简单写入顺序（建议顺序）

1. 将目标存储区（存储区 1/2）的 FLASH_CR1/2 寄存器中的 PG1/2 位置 1（请参见*使能写操作*一节）。
2. 检查目标存储区是否受保护（请参见*检查写保护*一节）。
3. 从 32 字节对齐地址处开始写入一个对应于 32 字节数据的 Flash 字。
4. 检查 BSY1/2 是否置 1，并等待其复位（请参见*写入状态忙标志*一节）。

最佳块写入顺序

可使用此顺序将块编程到 Flash：

1. 将相应存储区（存储区 1/2）的 FLASH_CR1/2 寄存器中的 PG1/2 位置 1（请参见*使能写操作*一节）。
2. 检查目标存储区是否受保护（请参见*检查写保护*一节）。
3. 连续写入 32 个数据字节（Flash 字），直至整个块传送完成。每个 Flash 字必须从 32 字节对齐地址开始。

释放 Flash 接口，使系统切换为停止或待机模式

如果其中一个忙标志有效，微控制器不能将 D1 域切换为停止或待机模式（请参见*第 6 节：电源控制 (PWR)*）。可通过两种方法释放 Flash 接口：

- 通过以下任一操作复位 WBNE1/2 忙标志：
  a) 用缺失的数据填满写缓冲区。
  b) 通过激活 FLASH_CR1/2 中的 FW1/2 位强制执行写操作，而不填入缺失的数据。此操作会强制将所有缺失的数据设置为“高电平”。
  c) 复位 FLASH_CR1/2 中的 PG1/2 位。此操作将禁止写缓冲区，进而导致缓冲区内容丢失。
- 轮询 BSY1/2 和 QW1/2 忙位，直至其清零， 这种情况说明记录的所有写入、擦除和选项更改操作均已完成。

微控制器随后可切换为停止或待机模式。

Flash 扇区擦除

擦除 128 KB 用户扇区的具体步骤如下：

1. 检查并清零（可选）之前的编程 / 擦除操作所导致的全部错误标志（有关详细信息，请参见*第 3.3.9 节：Flash 接口错误标志*）。
2. 解锁 FLASH_CR1/2 寄存器，具体说明请参见*配置编程参数*一节。
3. 将相应 FLASH_CR1/2 寄存器中的 SER1/2 位和 SNB1/2 位域置 1。SER1/2 指示扇区擦除操作，而 SNB1/2 则包含目标扇区数。
4. 将 FLASH_CR1/2 寄存器中的 START1/2 位置 1。
5. 等待相应 FLASH_SR1/2 寄存器中的 BSY1/2 位清零。

注： *如果需要同时执行存储区擦除和扇区擦除（BER1/2 位置 1），则会执行存储区擦除操作而不是扇区擦除操作。*

标准 Flash 存储区擦除

要擦除包含安全和保护数据的存储区扇区以外的所有存储区扇区，请执行以下程序：

1. 检查并清零（可选）之前的编程 / 擦除操作所导致的全部错误标志（有关详细信息，请参见*第 3.3.9 节：Flash 接口错误标志*）。
2. 解锁 FLASH_CR1/2 寄存器，具体说明请参见*配置编程参数*一节。
3. 如果存在 PCROP 保护区域，FLASH_PRAR_CUR1/2 和 FLASH_PRAR_PRG1/2 寄存器中的 DMEP1/2 位应置 0。
4. 如果存在仅安全区域，FLASH_SCAR_CUR1/2 和 FLASH_SCAR_PRG1/2 寄存器中的 DMES1/2 位应置 0。
5. 将对应于目标存储区的 FLASH_CR1/2 寄存器中的 BER1/2 位置 1。
6. 将 FLASH_CR1/2 寄存器中的 START1/2 位置 1 以开始存储区擦除操作。然后等待相应 FLASH_SR1/2 寄存器中的 BSY1/2 位清零。

注： *BER1/2 和 START1/2 位可同时置 1，步骤 8 和步骤 9 可以合并。*

如果需要同时执行存储区擦除和扇区擦除（SER1/2 位置 1），则会执行存储区擦除操作而不是扇区擦除操作。

使用自动取消保护来执行 Flash 存储区擦除操作

要在不执行 RDP 降级的情况下擦除所有存储区扇区（包括包含安全数据和保护数据的扇区），请执行以下程序：

1. 监测 FLASH_SR1/2 寄存器中的 BSY1/2 位，以确认当前未执行任何 Flash 操作。
2. 检查并清零（可选）之前的编程 / 擦除操作所导致的全部错误标志（有关详细信息，请参见*第 3.3.9 节：Flash 接口错误标志*）。
3. 解锁 FLASH_OPTCR 寄存器，具体说明请参见*配置编程参数*一节。
4. 如果存在 PCROP 保护区域，则将 FLASH_PRAR_CUR1/2 或 FLASH_PRAR_PRG1/2 寄存器中的 DMEP1/2 位置 1。此外，还要设定 PCROP 区域的结束地址和起始地址，使两者的差值为负，即：

   PROT_AREA_END1/2 < PROT_AREA_START1/2
5. 如果存在仅安全区域，则将 FLASH_SCAR_CUR1/2 或 FLASH_SCAR_PRG1/2 寄存器中的 DMES1/2 位置 1。此外，还要设定安全区域的结束地址和起始地址，使两者的差值为负，即：

   SEC_AREA_END1/2 < SEC_AREA_START1/2

注： *步骤 5 只能通过 ST 安全库或在安全模式下运行的应用程序来执行。不过，只有当存在仅安全区域时才必须执行该步骤。*

6. 将 FLASH_WPSN_PRG1/2R 中的所有 WRPSn1/2 位置 1 以禁止所有扇区写保护。
7. 解锁 FLASH_CR1/2 寄存器。
8. 将对应于目标存储区的 FLASH_CR1/2 寄存器中的 BER1/2 位置 1。
9. 将 FLASH_CR1/2 寄存器中的 START1/2 位置 1 以开始存储区擦除操作和取消保护操作。然后等待相应 FLASH_SR1/2 寄存器中的 BSY1/2 位清零。此时，存储区擦除操作会擦除整个存储区（包括包含 PCROP 保护数据和/或安全数据的扇区），并会自动执行选项字节更改，以禁止所有保护。

注： *BER1/2 和 START1/2 位可同时置 1，步骤 8 和步骤 9 可以合并。*

警告： **对于上述顺序，请注意警告：**
- 无法仅在一个存储区中执行上述顺序，还会同时修改其他存储区的保护参数。
- 不得更改上述选项字节以外的其他选项字节，并且不得在存储区擦除请求和取消保护请求的目标存储区以外的其他存储区中执行保护更改操作。
- 当上述一个或两个事件发生时，将执行简单的存储区擦除操作，不会发生选项字节更改，也不会设置选项更改错误。

Flash 批量擦除

要同时擦除两个存储区中的全部扇区（包含安全数据和保护数据的扇区除外），用户应用程序可将 FLASH_OPTCR 寄存器中的 MER 位置 1，如下所述：

1. 监测 FLASH_SR1/2 中的 BSY1/2 位，以确认当前未执行任何 Flash 操作。
2. 检查并清零（可选）之前的编程 / 擦除操作所导致的全部错误标志（有关详细信息，请参见*第 3.3.9 节：Flash 接口错误标志*）。
3. 解锁两个 FLASH_CR1/2 寄存器和 FLASH_OPTCR 寄存器，具体说明请参见*配置编程参数*一节。
4. 如果存在 PCROP 保护区域，FLASH_PRAR_CUR1/2 和 FLASH_PRAR_PRG1/2 寄存器中的 DMEP1/2 位应置 0。
5. 如果存在仅安全区域，FLASH_SCAR_CUR1/2 和 FLASH_SCAR_PRG1/2 寄存器中的 DMES1/2 位应置 0。
6. 将 FLASH_OPTCR 寄存器中的 MER 位置 1， 这会自动将 BER1、BER2、START1 和 START2 置 1，从而会开始对两个存储区进行存储区擦除操作。然后等待相应 FLASH_SR1/2 寄存器中的 BSY1/2 位清零。

自动保护取消时的Flash批量擦除

要在不执行 RDP 降级的情况下同时擦除两个存储区的所有扇区（包括包含安全数据和保护数据的扇区），请执行以下程序：

1. 监测 FLASH_SR1/2 中的 BSY1/2 位，以确认当前未执行任何 Flash 操作。
2. 检查并清零（可选）之前的编程/擦除操作所导致的全部错误标志。
3. 解锁两个 FLASH_CR1/2 寄存器和 FLASH_OPTCR 寄存器。

4. 如果存在 PCROP 保护区域，则将 FLASH_PRAR_CUR1/2 或 FLASH_PRAR_PRG1/2 寄存器中的 DMEP1/2 位置 1。此外，还要设定 PCROP 区域的结束地址和起始地址，使两者的差值为负。
5. 如果存在仅安全区域，则将 FLASH_SCAR_CUR1/2 或 FLASH_SCAR_PRG1/2 寄存器中的 DMES1/2 位置 1。此外，还要设定安全区域的结束地址和起始地址，使两者的差值为负。

注： *步骤 5 只能通过 ST 安全库或在安全模式下运行的应用程序来执行。不过，只有当存在仅安全区域时才必须执行该步骤。*

6. 将 FLASH_WPSN_PRG1/2R 中的所有 WRPSn1/2 位置 1 以禁止所有扇区写保护。
7. 将 FLASH_OPTCR 寄存器中的 MER 位置 1，然后等待相应 FLASH_SR1/2 寄存器中的 BSY1/2 位清零。此时，将在两个存储区上执行 Flash 存储区擦除操作和自动取消保护操作。

3.3.8 更改用户选项字节

用户选项字节更改操作可用于修改保存为 Flash 选项字节区域形式的配置及保护设置（有关详细列表，请参见*表 10：Flash 用户选项字节列表*）。

Flash 接口具有两组选项字节寄存器：

- 第一组寄存器包含选项字节的当前值，其扩展名为 _CUR。所有“_CUR”寄存器均为只读寄存器。其值会在上电后或选项字节更改操作后自动加载。
- 第二组寄存器可修改选项字节， 其扩展名为 _PRG。所有“_PRG”寄存器均可在读取/写入模式下访问。

如果 FLASH_OPTCR 寄存器中的 OPTLOCK 位置 1，则禁止修改 FLASH_OPTXX_PRG 寄存器。

如果 OPTSTART 位置“1”，Flash 接口会将当前值 (_CUR) 与新值 (_PRG) 进行比较，以确认是否至少有一个选项字节需要编程。

仅当以下条件得到满足时，才允许执行选项字节更改操作：

- 当前 RDP 不是级别 2（SWAP 位除外，任何级别下都可以更改此位）。
- 如果存在 PCROP 保护区域，那么新设定的区域必须大于或等于当前区域（降级期间除外）。
- 如果存在安全保护区域，可在设定安全模式的情况下修改此区域（请参见*第 4 节：安全存储管理*）。
- 如果是降级的情况（从级别 1 变为级别 0），DMEP1/2 选项位可无限制地进行设置，否则只能设为“1”。
- 仅可在执行降级（从级别 1 变为级别 0）的同时修改 FLASH_SCAR_PRG1/2 中的 DMES1/2 选项位，否则该位只能设为“1”。
- 如果 SECURITY 选项位已置 1，并且存在 PCROP 或安全区域，可通过执行降级并将 PCROP 或安全区域结束地址与起始地址之差设为负值的方式复位保护（SECURITY 位复位）。

如果其中一个条件未满足，Flash 接口会将 FLASH_OPTSR_CUR 寄存器中的 OPTCHANGEERR 标志置“1”，并会中止选项字节更改操作。

选项字节修改顺序

要修改用户选项字节，请执行以下步骤：

1. 按照*配置编程参数*一节中介绍的特定顺序解锁 OPTLOCK 位。
2. 将所需选项字节值写入相应的选项寄存器：FLASH_XXX_PRG1/2。
3. 将 FLASH_OPTCR 寄存器中的选项启动 OPTSTART 位置 1。
4. 等待 OPT_BUSY 位清零。

注： *如果在进行选项字节修改时出现复位或掉电情况，会保留原有的选项字节值。需要新的选项字节修改顺序才能设定新值。*

3.3.9 Flash 接口错误标志

编程/擦除操作期间，如果发生错误，Flash 接口会报告。用户应用程序可分别针对每个错误使能中断。需要先清除一些错误，才能开始新的写入操作。

每个存储区都有一组专有的错误标志来标识哪一存储区生成了错误：这些标志位于 Flash 状态寄存器 1 或 2 (FLASH_SR1/2) 中。

每个错误标志都与 FLASH_CR1/2 中的中断使能位相关联。如果中断使能位置“1”，当相应的错误标志置“1”时，会产生中断。要将错误标志清零，应将 FLASH_CCR1/2 寄存器中的相应位置 1。

写保护错误 (WRPERR1/2)

WRPERR1/2 标志属于黏着位，当用户应用程序尝试对受保护区域进行编程/擦除操作时，会通过硬件将此位置 1。当此标志置 1 时，操作会中止，并且不会进行任何更改。将 FLASH_CCR1/2 寄存器中的 CLR_WRPERR1/2 位置 “1”可将此标志清零。如果 FLASH_CR1/2 寄存器中的 WRPERRIE1/2 位置 “1”，当 WRPERR1/2 标志置 1 时，会生成中断。

WRPERR1/2 必须先清零才能开始执行新的写操作，否则会生成顺序错误（PGSERR1/2 位），并且下一编程操作也会中止。

编程顺序错误 (PGSERR1/2)

PGSERR1/2 标志属于黏着位，当编程顺序不正确时，会通过硬件将此位置 1。满足下列条件之一时，此位会置“1”：

- AXI 总线发出了写操作请求，但 PG1/2 位尚未置 1。
- 写保护错误标志 (WRPERR1/2) 尚未复位便发出了新的写操作请求。
- 不一致错误 (INCERR1/2) 尚未复位便发出了新的写操作请求。
- Flash 操作错误 (OPERR1/2) 尚未复位便发出了新的写操作请求。

PGSERR1/2 标志置 1 时，编程操作中止。要将此标志清零，应将 FLASH_CCR1/2 寄存器中的 CLR_PGSERR1/2 位置“1”。如果 FLASH_CR1/2 中的 PGSERRIE1/2 位置 1，当 PGSERR1/2 标志置 1 时，会生成中断。

PGSERR1/2 必须先清零才能开始执行新的写操作。

选通错误 (STRBERR1/2)

STRBERR1/2 标志属于黏着位，当用户应用程序多次向写缓冲区中写入同一字节时，会通过硬件将此位置 1。写操作不会中止，应用程序可忽略此错误、继续执行当前写操作并请求执行新的写操作。将 FLASH_CCR1/2 中的 CLR_STRBERR1/2 位置 1 可将此标志清零。如果 FLASH_CR1/2 中的 STRBERRIE1/2 位置 1，当 STRBERR1/2 标志置 1 时，会生成中断。

不必先将 STRBERR1/2 标志清零便可以开始执行新的写操作。

不一致错误 (INCERR1/2)

INCERR1/2 标志属于黏着位，出现以下情况时，会通过硬件将此位置 1：

- 用户应用程序开始对 Flash 字执行写操作（使用第一次突发），在当前写操作完成之前向另一个 Flash 字地址发送新的突发写操作。在这种情况下，写缓冲区会被清空，新数据会被拒绝，INCERR1/2 标志会置 1。
- 主机开始对 Flash 字执行写操作（使用第一次突发），在第一台主机发起的操作完成之前，另一台主机向同一个或新的 Flash 字地址发送新的突发写操作。在这种情况下，写缓冲区会被清空，新数据会被拒绝，INCERR1/2 标志会置 1。
- 主机发出的回卷突发与两个或多个 Flash 字地址重叠（回卷突发的最大大小限制为 Flash 字大小，即 256 位）。

当 INCERR1/2 标志置 1 时，操作会中止，并且不会执行任何写操作。要将此标志清零，应将 FLASH_CCR1/2 寄存器中的 CLR_INCERR1/2 位置 1。如果 FLASH_CR 寄存器中的 INCERRIE1/2 位置 1，当 INCERR1/2 标志置 1 时，会生成中断。

INCERR1/2 标志必须先清零才能开始执行新的编程或擦除操作，否则会生成顺序错误（PGSERR1/2 位），并且下一编程操作也会中止。

操作错误 (OPERR1/2)

OPERR1/2 标志属于黏着位，当 Flash 在写操作或擦除操作期间检测到错误时，会通过硬件将此位置 1。出现此错误的原因可能是因循环问题造成 Flash 行为不正确。

要将此标志清零，应将 FLASH_CCR1/2 寄存器中的 CLR_OPERR1/2 位置“1”。如果 FLASH_CR1/2 寄存器中的 OPERRIE1/2 位为“1”，当 OPERR1/2 标志置 1 时，会生成中断。

OPERR1/2 必须先清零才能开始执行新的编程操作，否则会生成顺序错误（PGSERR1/2 位），并且新的编程操作也会中止。

3.3.10 同时在存储区 1 和存储区 2 上执行读取/编程/擦除操作

Flash 分为两个独立的存储区。Flash 接口可同时对每个存储区执行不同操作。对存储区 1 执行读取、编程或擦除操作的同时可对存储区 2 执行另一读取、编程或擦除操作。

例外情况是选项字节更改需要进行降级时：此时需要两个存储区均可用。

3.3.11 FLASH 选项字节

选项字节说明

选项字节是最终用户根据应用程序需求配置的。用户选项字节可通过 Flash 接口寄存器接口访问。选项字节的编程顺序在*选项字节修改顺序*一节中进行了介绍。

为了提高 Flash 接口中选项字节存储的稳健性，每个选项字节数据都关联到 Flash 中的误码校正 (ECC)。

表 10 介绍了 Flash 接口寄存器中所有可用的用户选项字节的列表。

表 10. Flash 用户选项字节列表

用户选项	寄存器(1)	大小（位）	默认出厂编程值(2)
SWAP_BANK	FLASH_OPTSR_CUR	1	0x0
RDP	FLASH_OPTSR_CUR	8	0xAA
BOR_LEV	FLASH_OPTSR_CUR	2	0x1
BOOT_ADD0	FLASH_BOOT_CURR	16	0x0800
BOOT_ADD1	FLASH_BOOT_CURR	16	0x1FF0
DMEP1	FLASH_PRAR_CUR1	1	0x0
DMES1	FLASH_SCAR_CUR1	1	0x0
WRPSn1	FLASH_WRP_CUR1R	8	0xFF
PROT_AREA_START1	FLASH_PRAR_CUR1	12	0xFF
PROT_AREA_END1	FLASH_PRAR_CUR1	12	0x00
SEC_AREA_START1	FLASH_SCAR_CUR1	12	0xFF
SEC_AREA_END1	FLASH_SCAR_CUR1	12	0x00
DMEP2	FLASH_PRAR_CUR2	1	0x0
DMES2	FLASH_SCAR_CUR2	1	0x0
WRPSn2	FLASH_WPSN_CUR1R	8	0xFF
PROT_AREA_START2	FLASH_PRAR_CUR2	12	0xFF
PROT_AREA_END2	FLASH_PRAR_CUR2	12	0x00
SEC_AREA_START2	FLASH_SCAR_CUR2	12	0xFF
SEC_AREA_END2	FLASH_SCAR_CUR2	12	0x00
IWDG1_HW	FLASH_OPTSR_CUR	1	0x1
SECURITY	FLASH_OPTSR_CUR	1	0x0
ST_RAM_SIZE	FLASH_OPTSR_CUR	2	0x10
nRST_STBY_D1	FLASH_OPTSR_CUR	1	0x1
nRST_STOP_D1	FLASH_OPTSR_CUR	1	0x1
FZ_IWDG_SDBY	FLASH_OPTSR_CUR	1	0x1
FZ_IWDG_STOP	FLASH_OPTSR_CUR	1	0x1
PERSO_OK	FLASH_OPTSR_CUR	1	0x1
IO_HSLV	FLASH_OPTSR_CUR	1	0x0

1. *第 3.5 节：Flash 接口寄存器*中介绍了位映射和详细的位说明。
2. 默认出厂选项字节值是首次选项字节更改之前的默认值。

选项字节加载

可通过三种方法加载 Flash 用户选项字节：

1. 上电时：

   Flash 接口自动从 Flash 加载用户选项字节。加载过程中，器件保持在复位状态下，Flash 接口不可访问。

2. 包含 Flash 接口的 D1 电源域从待机模式切换为运行模式时：

   这种情况下，Flash 接口会执行与上电顺序期间相同的操作，不同之处在于器件不会保持在复位状态下。

3. 如果用户应用程序通过 Flash 接口寄存器修改选项字节内容：

   这种情况下，在 Flash 接口对选项字节扇区进行重新编程后，会执行选项字节重载来更新 Flash 接口选项寄存器。

更改特定选项位的规则

一些选项字节字段必须遵循特定规则才能更新为新值。下文对这些字段以及相关限制条件进行了介绍：

- 安全区域大小

  安全区域大小是由位于 FLASH_SCAR_CUR1/2 和 FLASH_SCAR_PRG1/2 寄存器中的 SEC_AREA_START1/2、SEC_AREA_END1/2 和 DMES1/2 控制的。这些选项字节可在安全模式下进行修改。如果 SEC_AREA_END1/2 字段小于 SEC_AREA_START1/2，安全区域大小为零。如果 SEC_AREA_START1/2 = SEC_AREA_END1/2，所有存储区均受到安全保护。有关详细信息，请参见*第 4 节：安全存储管理*。

- DMES1/2

  如果此选项位置 1，降级（从级别 1 变为级别 0）或者取消所有保护的存储区擦除期间会擦除安全区域，否则会保留安全区域的内容。仅当同时请求降级或取消所有保护的存储区擦除时，才能完成此位的复位。有关详细信息，请参见*第 4 节：安全存储管理*。

- PCROP 区域大小（仅执行）

  PCROP 区域大小是由 FLASH_PRAR_CUR1/2 和 FLASH_PRAR_PRG1/2 寄存器中的 PROT_AREA_START1/2、PROT_AREA_END1/2 和 DMEP1/2 控制的。增大 PCROP 区域没有任何限制。要减小或移除 PCROP 区域，必须同时请求进行降级。可以执行取消所有保护的存储区擦除，以删除 PCROP 区域。如果 PROT_AREA_END1/2 字段小于 PROT_AREA_START1/2 字段，PCROP 区域大小为零。如果 PROT_AREA_START1/2 = PROT_AREA_END1/2，所有存储区均受到 PCROP 保护。有关详细信息，请参见*第 4 节：安全存储管理*。

- DMEP1/2

  如果此位已置 1，降级或取消所有保护的存储区擦除期间，会擦除 PCROP 区域。如果此位未置 1，降级（由级别 1 变为级别 0）期间，会保留 PCROP 区域的内容。将此位置 1 没有任何限制。但仅当同时请求降级或存储区擦除时，才能完成此位的复位。

- 读保护 (RDP)
  - 如果 RDP 设为级别 2，则无法返回到较低级别。如果用户应用程序尝试执行此类操作，会产生错误，并会忽略所需更改。
  - 如果 RDP 设为级别 1，在用户应用程序请求转换为 RDP 级别 2 的情况下，允许进行 RDP 选项字节修改，没有任何限制。
  - 但是，如果相应存储区的 FLASH_SCAR_CUR1/2 中的 DMES1/2 以及 FLASH_PRAR_CUR1/2 中的 DMEP1/2 位置“1”，则从级别 1 转换为级别 0 会对 Flash 进行批量擦除。

  - 如果用户应用程序希望在降级期间保留 PCROP 区域或安全区域，DMEP1/2 或 DMES1/2 位必须在执行降级之前置“0”。
  - 如果 RDP 设为级别 0，则可转换为级别 1 或级别 2，没有任何限制。
- 扇区写保护 (WRPSn1/2)

  无特定规则。
- 安全模式 (SECURITY)

  将此选项位置 1 可激活器件安全模式。如果不存在安全区域或 PCROP 区域，此位可清零，否则必须执行降级或安全自举服务才能将此位复位。有关详细信息，请参见*第 4 节：安全存储管理*。
- ST_RAM_SIZE

  此选项字节的值可增大，没有任何限制。仅允许通过根安全服务减小该值。仅当 SECURITY 选项字节置 1 时，此选项才有效。有关详细信息，请参见*第 4 节：安全存储管理*。

3.3.12 保护机制

Flash 接口使用不同的保护机制：

- 读保护 (RDP)
- PCROP（专有代码读保护）：仅执行区域
- 安全保护
- 扇区写保护。

RDP 读保护

可保护 Flash 用户区域，以防止不可信代码进行读操作。存在三种读保护级别（请参见*表 11：RDP 值与读取保护级别*、*表 12：允许的访问与读取保护级别*和*图 6：保护转换方案*）：

- 级别 0：无读保护

  将 0xAA 写入读保护选项字节 (RDP) 时，读保护级别即设为 0，此时，在所有自举配置（Flash 用户自举、调试或从 RAM 自举）中，均可执行对 Flash 或备份 SRAM 的读取/编程操作（如果未设置其他保护）。
- 级别 1

  将任意值（分别用于设置级别 0 和级别 2 的 0xAA 和 0xCC 除外）写入 RDP 选项字节时，即激活读保护级别 1。设置读保护级别 1 后：
  - 如果调试器已连接或自举配置不同于用户 Flash，则不允许访问（读取、擦除、编程）Flash 或备份 SRAM。如果请求对 Flash 执行读访问，则会生成总线错误。
  - 如果从 Flash 自举（未连接调试器），则允许通过用户代码对 Flash 和备份 SRAM 进行读取、擦除和编程访问。
  - 激活级别 1 后，如果将保护选项字节编程为级别 0（RDP 降级），则将对 Flash 和备份 SRAM 执行批量擦除。结果，用户 Flash 和备份 SRAM 会在 RDP 级别设为级别 0 之前清空。如果 DMEP1/2 和 DMES1/2 位为“0”，批量擦除操作会删除存储区和备份 SRAM 的内容，同时会保留 Flash PCROP 和安全区域的内容。

- 级别 2：禁止调试并使能 RDP

  将 0xCC 写入 RDP 选项字节时，可激活读保护级别 2。设置读保护级别 2 后：

  – 级别 1 提供的所有保护均有效。
  – 不再允许从 RAM 自举。
  – 如果使能了安全功能，会强制在安全系统 Flash 中进行自举（请参见*第 4 节：安全存储管理*）。
  – 除使能了安全功能的情况外，不再允许从系统 Flash 自举。强制在安全系统 Flash 中进行自举（请参见*第 4 节：安全存储管理*）。
  – 所有调试功能均禁止。
  – 除 SWAP 位之外，不再允许更改其他用户选项字节（请参见*第 3.3.13 节：Flash 存储区交换*）。
  – 从 Flash 自举时，允许通过用户代码对 Flash 和备份 SRAM 进行读取、擦除和编程访问。存储器读保护级别 2 是不可更改的。激活级别 2 后，保护级别不能再降回级别 0 或级别 1。

注： *激活级别 2 后，将永久性禁止 JTAG 端口（相当于 JTAG 熔断）。因此，意法半导体无法对设为保护级别 2 的器件做失效分析。*

表 11. RDP 值与读取保护级别

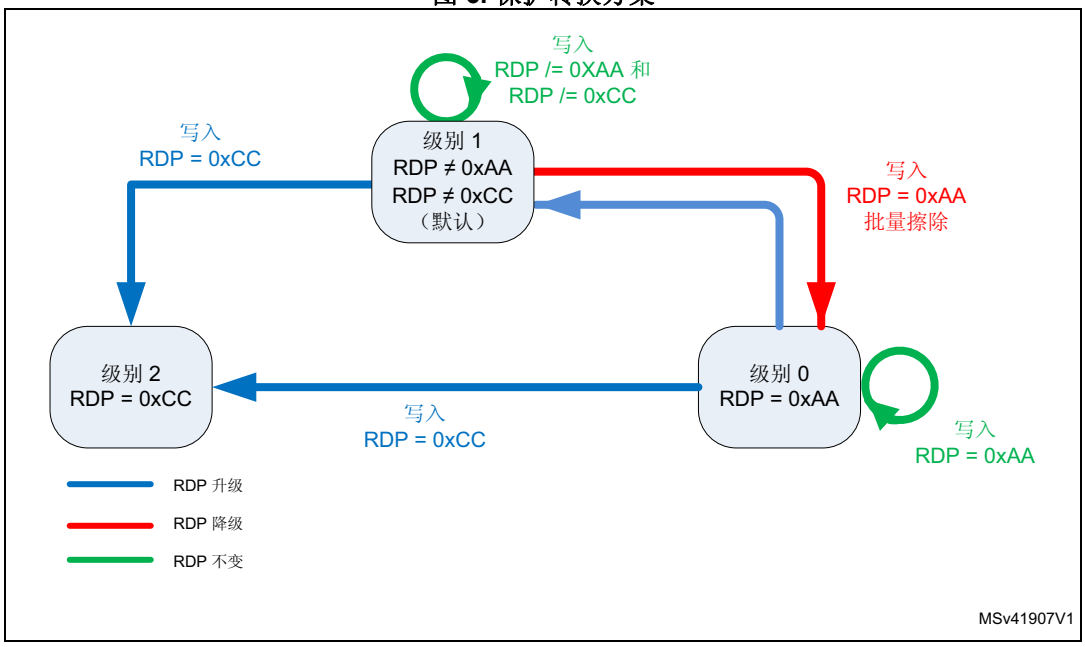
RDP 选项字节值	读取保护级别
0xAA	级别 0
0xCC	级别 2
其他任意值	级别 1

表 12. 允许的访问与读取保护级别

区域	保护级别 (RDP)	自举 = 用户 Flash (1)	自举 /= 用户 Flash 或连接了调试
用户 Flash	1	R/W/E	无访问
	2	R/W/E	-(2)
系统 Flash	1	R	R
	2	R	-(2)
选项字节	1	R/W/E	R/W/E
	2	R	-(2)
备份 SRAM	1	R/W	无访问
	2	R/W	-(2)

1. W：写入，R：读取，E：擦除。
2. RDP = 级别 2 时，不可使用调试功能，除用户 Flash 自举外不能执行其他自举操作，不允许修改 SWAP 位域之外的其他选项字节。

图 6. 保护转换方案



PCROP 区域（专有代码读保护，仅执行区域）

Flash 接口允许在每个 Flash 存储区中定义“仅执行”区域。此区域仅允许执行取指事务。不允许进行数据访问（数据或缓冲池）。必须使用“仅执行”选项相应地编译原生代码。仅执行区域可帮助保护软件知识产权。

可通过设置 PROT_AREA_END1/2 和 PROT_AREA_START1/2 选项字节来定义两个不同的 PCROP 区域（每个 Flash 存储区一个），使两者的差值为正，而不是零（END 地址大于 START 地址）。

可设置的最小 PCROP 区域为 16 个 Flash 字，也就是 512 字节。可设置的最大可执行区域为整个 Flash 存储区。具体方法是将 PROT_AREA_END1/2 和 PROT_AREA_START1/2 字段设为相同值。

PCROP 区域可使用 8 个 Flash 字的粒度来定义。也就是说，假定 PROT_AREA_END1/2 绝对大于 PROT_AREA_START1/2，那么 END 地址与 START 地址之差加 1 再乘以 8 表示进行 PCROP 配置的 Flash 字数。

要禁止两个 PCROP 区域中的任何一个，PROT_AREA_END1/2 地址字段必须设为小于 PROT_AREA_START1/2 的值，也就是得出的差值为负（END 地址小于 START 地址）。

举例来说，要设置从地址 0x08000000（包含）到地址 0x08000FFF（包含）的 PCROP 区域（对应于从第一个存储区的存储区 1 基址开始的 4 KB 空间），必须分别将 PROT_AREA_START1 和 PROT_AREA_END1 寄存器编程为：

- PROT_AREA_START1 = 0x000
- PROT_AREA_END1 = 0x00F

保护区域大小等于：

[(PROT_AREA_END - PROT_AREA_START) + 1] x 256 = 16 x 256 字节 = 4 KB。

PCROP 区域属性

- 对 PCROP 区域执行读取（非取指）事务会将 FLASH_SR1/2 寄存器中的 RDPERR 标志置 1。
- 对 PCROP 区域执行写入事务会将 FLASH_SR1/2 寄存器中的 WRPERR 标志置 1。
- 如果已定义 PCROP 区域，仍可通过减小 START 地址或增大 END 地址字段的方式增大区域大小。可通过以下两步禁止 PCROP 区域：
  - 执行 RDP 降级（从级别 1 变为级别 0），使能 DMEP1/2 位并在 FLASH_PRAR_PRG1/2 寄存器中将 PROT_AREA_START1/2 的值设为大于 PROT_AREA_END1/2。

    PCROP 保护与 RDP 保护无关。但这种情况下，在取消 PCROP 之前，应先将 RDP 设为级别 1，以使能 RDP 降级。
  - 请求执行存储区擦除操作，同时将 DMEP1/2、DMES1/2 和所有 WRPSn1/2 位置“1”，并在 FLASH_PRAR_PRG1/2 中将 PROT_AREA_START1/2 的值设为大于 PROT_AREA_END1/2，在 FLASH_SCAR_PRG1/2 中将 SEC_AREA_START1/2 的值设为大于 SEC_AREA_END1/2。

安全区域

Flash 接口允许在每个 Flash 存储区中设置“安全”区域。除非设置了安全模式，否则不能访问存储在此区域中的数据和程序。安全区域有助于将安全用户代码与应用程序非安全代码分离开。例如，安全区域可用于保护安全固件升级代码和安全自举代码。

可通过设置 SEC_AREA_END1/2 和 SEC_AREA_START1/2 选项字节来定义两个不同的安全区域（每个 Flash 存储区一个），使两者的差值为正，而不是零（END 地址大于 START 地址）。需要注意的是，只能在安全模式下修改这些选项字节（请参见*第 4 节：安全存储管理*）。

可设置的较小的安全区域大小为 512 字节。最大安全区域大小是整个 Flash 存储区。具体方法是将 SEC_AREA_END1/2 和 SEC_AREA_START1/2 字段设为相同值。

安全区域可使用 8 个 Flash 字的粒度来定义。也就是说，假定 SEC_AREA_END1/2 绝对大于 SEC_AREA_START1/2，那么 END 地址与 START 地址之差加 1 再乘以 8 表示受到安全保护的 Flash 字数。

要禁止两个安全区域中的任何一个，SEC_AREA_END1/2 地址字段必须设为小于 SEC_AREA_START1/2 的值，也就是得出的差值为负（END 地址小于 START 地址）。

举例来说，要设置从地址 0x08100000（包含）到地址 0x08101FFF（包含）的安全区域（对应于从第二个存储区基址开始的 8 KB 空间），必须分别将 SEC_AREA_START2 和 SEC_AREA_END2 寄存器编程为：

- SEC_AREA_START2 = 0x000
- SEC_AREA_END2 = 0x01F

安全区域大小等于：

[(SEC_AREA_END2 - SEC_AREA_START2) + 1] x 256 = 32 x 256 字节 = 8 KB。

安全区域属性

- 如果使能了安全功能，可在特定条件下修改安全区域大小（请参见*第 4 节：安全存储管理*）。可通过两种方法删除安全区域：
  - 执行 RDP 降级（从级别 1 变为级别 0），使能 DMES1/2 位并在 FLASH_SCAR_PRG1/2 寄存器中将 SEC_AREA_START1/2 地址的值设为大于 SEC_AREA_END1/2。

    RDP 保护与安全保护无关。但这种情况下，在取消 RDP 之前，应先将 RDP 设为级别 1，以使能 RDP 降级。
  - 请求执行存储区擦除操作，同时将 DMEP1/2、DMES1/2 和所有 WRPSn1/2 位置“1”，并在 FLASH_SCAR_PRG1/2 中将 SEC_AREA_START1/2 地址的值配置为大于 SEC_AREA_END1/2 的值，在 FLASH_PRAR_PRG1/2 中将 PROT_AREA_START1/2 地址的值配置为大于 PROT_AREA_END1/2 的值。

更多关于器件安全功能使用方法以及用户安全模式使能/禁止方法的详细信息，请参见*第 4 节：安全存储管理*。

扇区写保护

扇区写保护的作用是避免 Flash 的非易失性代码和/或数据被意外修改。将 FLASH_WPSN_PRG1/2R 寄存器中的相应 WRPSn1/2 位清零/置 1，可为任何 Flash 扇区单独执行写保护或取消保护。

受到写保护的扇区既不能擦除、也不能编程。因此，如果一个存储区扇区受到写保护，除非在降级范围内完成存储区擦除，否则不能执行存储区擦除。这种情况下，由于仅会擦除未受保护的扇区，因此存储区擦除操作会转变为一系列扇区擦除操作。

如果 RDP 级别设为级别 0 或级别 1，可对扇区写保护位（用户选项）进行修改，没有任何限制。如果 RDP 级别设为级别 2，则不能更改选项字节中的写保护位域。

注： *默认情况下，PCROP 区域也受到写保护。因此，部分或全部属于 PCROP 区域的扇区不能进行擦除和编程操作。*

3.3.13 Flash 存储区交换

Flash 接口允许交换存储区 1 和存储区 2 的存储器映射。固件升级后可使用此功能使器件在系统复位后以新固件重启。存储区交换是由 FLASH_OPTCR 寄存器中的 SWAP_BANK 位控制的。*表 13* 按 SWAP_BANK 选项位配置列出了两个 AXI 从 Flash 接口可使用的存储器映射。

表 13. AXI 接口存储器映射 SWAP_BANK = “0”

区域	AXI 1	AXI 2
用户扇区存储区 1	有(1)	无
用户扇区存储区 2	无	有(1)
系统 Flash 存储区 1	有(2)	无
系统 Flash 存储区 2	无	有(2)

1. 用户存储区 1 和用户存储区 2 分别从 0x0800 0000 映射到 0x080F FFFF，从 0x0810 0000 映射到 0x081F FFFF。
2. 系统 Flash 存储区 1 和系统 Flash 存储区 2 分别从 0x1FF0 0000 映射到 0x1FF1 FFFF，从 0x1FF4 0000 映射到 0x1FF5 FFFF。

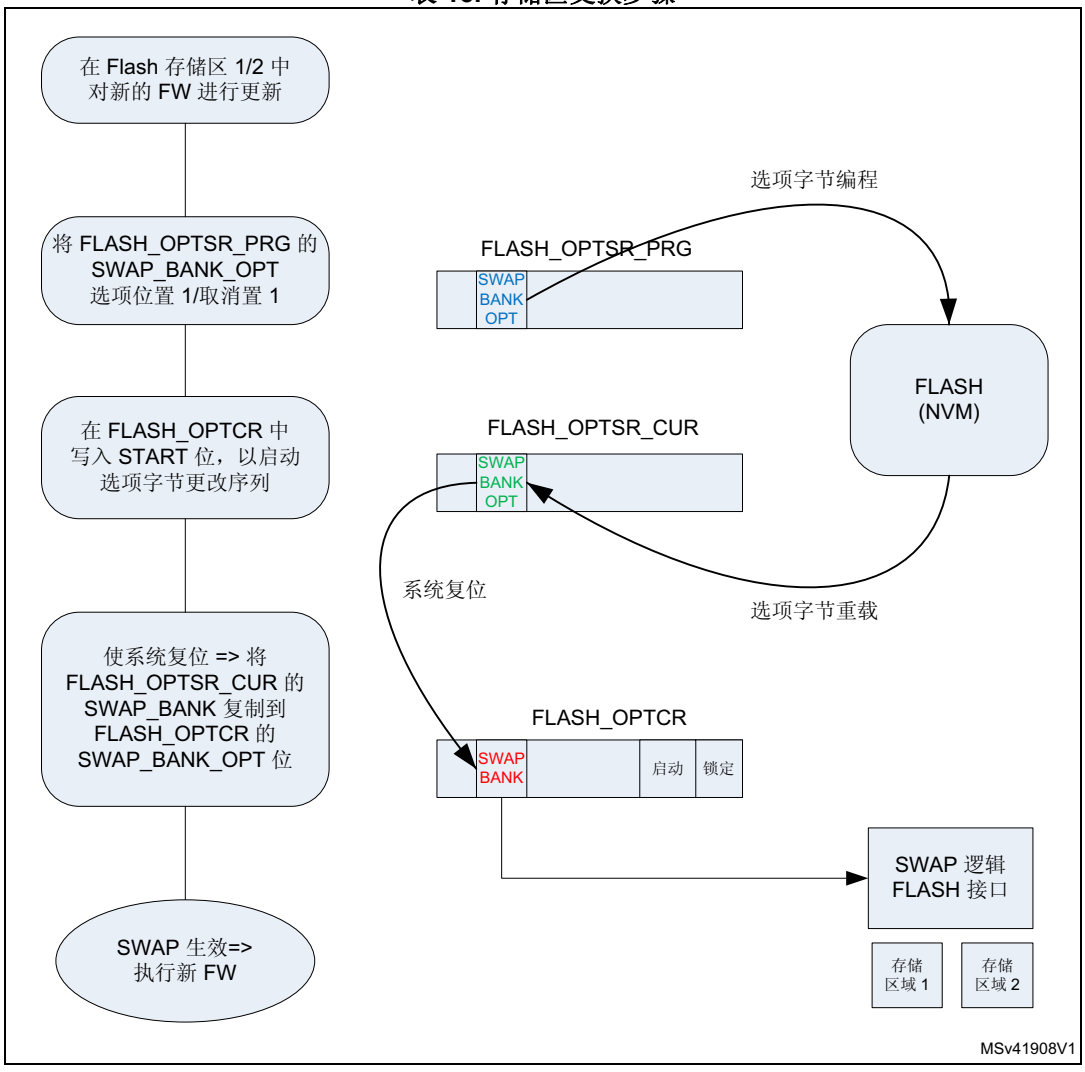
表 14. AXI 接口存储器映射 SWAP_BANK = “1”

区域	AXI 1	AXI 2
用户扇区存储区 1	无	有(1)
用户扇区存储区 2	有(1)	无
系统 Flash 存储区 1	有(2)	无
系统 Flash 存储区 2	无	有(2)

1. 用户存储区 1 和用户存储区 2 分别从 0x0810 0000 映射到 0x081F FFFF，从 0x0800 0000 映射到 0x080F FFFF。
2. 系统 Flash 存储区 1 和系统 Flash 存储区 2 分别从 0x1FF0 0000 映射到 0x1FF1 FFFF，从 0x1FF4 0000 映射到 0x1FF5 FFFF。

如果修改 FLASH_OPTSR_PRG 寄存器中的 SWAP_BANK_OPT 位，则仅当选项字节从选项字节扇区重新载入到 FLASH_OPTSR_CUR 寄存器中并且其值在系统复位后复制到 FLASH_OPTCR 寄存器的 SWAP_BANK 位时，新值才会设置成功。

表 15. 存储区交换步骤



下文介绍了在固件更新后交换存储区 1 和存储区 2 的存储器映射的建议步骤：

1. 使用新固件更新存储区 2 或存储区 1。
2. 对 FLASH_OPTSR_PRG 寄存器中的 SWAP_BANK_OPT 位进行相应修改。
3. 将 FLASH_OPTCR 寄存器中的 OPTSTART 位置 1，开始执行选项字节更改步骤。对寄存器进行编程之前，确保寄存器已解锁。
4. 执行系统复位。执行完选项字节加载步骤后，存储区交换生效，应执行新固件。

注： *无论 RDP 级别是多少（即使是级别 2），都可以修改 FLASH_OPTSR_PRG 中的 SWAP_BANK_OPT 选项位，从而可在任何读保护级别下进行高级固件升级。*

3.4 Flash 中断

如*第 3.3.9 节：Flash 接口错误标志*中所述，如果 FLASH_CR1/2 寄存器中的相应中断使能位置 1，每个标志错误均可生成中断。

擦除、编程或选项更改操作成功完成后，FLASH_SR1/2 寄存器中的操作结束 (EOP1/2) 位会置 1。如果将 FLASH_CR1/2 寄存器中的操作结束中断使能位 (EOPIE1/2) 置 1，则可在擦除、编程或选项更改操作完成时生成中断。

向 FLASH_FCCR1/2 寄存器中的相应位写入数据，可将 EOP1/2 标志清零。

3.5 Flash 接口寄存器

3.5.1 Flash 访问控制寄存器 (FLASH_ACR)

FLASH access control register

偏移地址：0x000 或 0x100

复位值：0x0000 0037

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

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

位 31:6 保留，必须保持复位值

位 5:4 **WRHIGHFREQ**：Flash 信号延时 (Flash signal delay)

这些位用于控制编程操作期间各 Flash 信号之间的延时。用户需要根据 Flash 接口频率写入正确的值（请参见*表 7*）。不会执行任何检查来验证配置是否正确。

00： ≤ 85 MHz
01： ≤ 185 MHz
10： ≤ 285 MHz
11： ≤ 385 MHz

位 3 保留，必须保持复位值

位 2:0 **LATENCY**：读延时 (Read latency)

这些位的值用于指定对两个 Flash 存储区执行读操作期间将会使用的等待状态数。用户需要根据*表 7* 所示的 Flash 接口频率和电源工作模式写入正确的值。不会执行任何检查来验证配置是否正确。

000：从 Flash 中读取字时需要使用 0 个等待状态
001：从 Flash 中读取字时需要使用 1 个等待状态
...
111：从 Flash 中读取字时需要使用 7 个等待状态

3.5.2 存储区 1 的 FLASH 密钥寄存器 (FLASH_KEYR1)

FLASH key register for bank 1

偏移地址：0x004

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
KEYR1															
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
KEYR1															
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

位 31:0 **KEYR1**：存储区 1 访问配置解锁密钥 (Bank 1 access configuration unlock key)

FLASH_KEYR1 为只写寄存器。要将 FLASH_CR1 寄存器解锁并允许对其执行编程/擦除操作，必须顺序编程以下值：

a) 第 1 个密钥 = 0x4567 0123
b) 第 2 个密钥 = 0xCDEF 89AB

有关详细信息，请参见*配置编程参数*一节。

3.5.3 Flash 选项密钥寄存器 (FLASH_OPTKEYR)

FLASH option key register

偏移地址：0x008 或 0x108

复位值：0x0000 0000

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

位 31:0 **OPTKEYR**：解锁密钥选项字节 (Unlock key option bytes)

FLASH_OPTKEYR 为只写寄存器。要将 FLASH_OPTCR 寄存器解锁并允许对该寄存器以及所有 _PRG 寄存器执行编程/擦除操作，必须顺序编程以下值：

a) 第 1 个密钥 = 0x0819 2A3B

b) 第 2 个密钥 = 0x4C5D 6E7F

有关更多详细信息，请参见*配置编程参数*一节。

3.5.4 存储区 1 的 FLASH 控制寄存器 (FLASH_CR1)

FLASH control register for bank 1

偏移地址：0x00C

复位值：0x0000 0031

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	CRCENDIE1	DBECCERRIE1	SNECCERRIE1	RDSERRIE1	RDPERRIE1	OPERRIE1	INCERRIE1	Res.	STRBERRIE1	PGSERRIE1	WRPERRIE1	EOPIE1
				rw	rw	rw	rw	rw	rw	rw		rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CRC_EN	Res.	Res.	Res.	Res.	SNB1			START1	FW1	PSIZE1		BER1	SER1	PG1	LOCK1
rw					rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rs

位 31:28 保留，必须保持复位值

位 27 **CRCENDIE1**：存储区 1 CRC 计算结束中断使能位 (Bank 1 end of CRC calculation interrupt enable bit)

如果 CRCENDIE1 位置“1”，对存储区 1 的 CRC 计算完成时，会产生中断。仅当 LOCK1 置“0”时，才能编程 CRCENDIE1。

0：对存储区 1 进行的 CRC 计算完成时不产生中断

1：对存储区 1 进行的 CRC 计算完成时产生中断

位 26 **DBECCERRIE1**：存储区 1 ECC 双重检测错误中断使能位 (Bank 1 ECC double detection error interrupt enable bit)

如果 DBECCERRIE1 位置“1”，对存储区 1 执行读操作过程中发生 ECC 双重检测错误时，会产生中断。仅当 LOCK1 置“0”时，才能编程 DBECCERRIE1。

0：存储区 1 发生 ECC 双重检测错误时不产生中断

1：存储区 1 发生 ECC 双重检测错误时产生中断

位 25 **SNECCERRIE1**：存储区 1 ECC 单校正错误中断使能位 (Bank 1 ECC single correction error interrupt enable bit)

如果 SNECCERRIE1 位置“1”，对存储区 1 执行读操作过程中发生 ECC 单校正错误时，会产生中断。仅当 LOCK1 置“0”时，才能编程 SNECCERRIE1。

0：存储区 1 发生 ECC 单校正错误时不产生中断

1：存储区 1 发生 ECC 单校正错误时产生中断

位 24 **RDSERRIE1**：存储区 1 安全错误中断使能位 (Bank 1 secure error interrupt enable bit)

如果 RDSERRIE1 位置“1”，在对存储区 1 执行读操作的过程中发生安全错误（未获得适当权限便访问安全保护地址）时，会产生中断。仅当 LOCK1 置“0”时，才能编程 RDSERRIE1。

0：存储区 1 发生安全错误时不产生中断

1：存储区 1 发生安全错误时产生中断

位 23 **RDPERRIE1**：存储区 1 读保护错误中断使能位 (Bank 1 read protection error interrupt enable bit)

如果 RDPERRIE1 位置“1”，在对存储区 1 执行读操作的过程中发生读保护错误（访问受 PCROP 保护的地址）时，会产生中断。仅当 LOCK1 置“0”时，才能编程 RDPERRIE1。

0：存储区 1 发生读保护错误时不产生中断

1：存储区 1 发生读保护错误时产生中断

位 22 **OPERRIE1**：存储区 1 写入/擦除错误中断使能位 (Bank 1 write/erase error interrupt enable bit)

如果 OPERRIE1 位置“1”，对存储区 1 执行写入/擦除操作过程中检测到错误时，会产生中断。仅当 LOCK1 置“0”时，才能编程 OPERRIE1。

0：存储区 1 发生写入/擦除错误时不产生中断

1：存储区 1 发生写入/擦除错误时产生中断

位 21 **INCERRIE1**：存储区 1 不一致错误中断使能位 (Bank 1 inconsistency error interrupt enable bit)

如果 INCERRIE1 位置“1”，对存储区 1 执行写入操作过程中发生不一致错误时，会产生中断。仅当 LOCK1 置“0”时，才能编程 INCERRIE1。

0：存储区 1 发生不一致错误时不产生中断

1：存储区 1 发生不一致错误时产生中断

位 20 保留，必须保持复位值

位 19 **STRBERRIE1**：存储区 1 选通错误中断使能位 (Bank 1 strobe error interrupt enable bit)

如果 STRBERRIE1 位置 “1”，对存储区 1 执行写入操作过程中发生选通错误时 （主机多次向写缓冲区中编程同一字节），会产生中断。仅当 LOCK1 置 “0”时，才能编程 STRBERRIE1。

0：存储区 1 发生选通错误时不产生中断

1：存储区 1 发生选通错误时产生中断

位 18 **PGSERRIE1**：存储区 1 编程顺序错误中断使能位 (Bank 1 programming sequence error interrupt enable bit)

如果 PGSERRIE1 位置“1”，对存储区 1 执行编程操作过程中发生顺序错误时，会产生中断。仅当 LOCK1 置“0”时，才能编程 PGSERRIE1。

0：存储区 1 发生顺序错误时不产生中断

1：存储区 1 发生顺序错误时产生中断

位 17 **WRPERRIE1**：存储区 1 写保护错误中断使能位 (Bank 1 write protection error interrupt enable bit)

如果 WRPERRIE1 位置“1”，对存储区 1 执行编程操作过程中发生保护错误时，会产生中断。仅当 LOCK1 置“0”时，才能编程 WRPERRIE1。

0：存储区 1 发生保护错误时不产生中断

1：存储区 1 发生保护错误时产生中断

位 16 **EOPIE1**：存储区 1 编程结束中断控制位 (Bank 1 end-of-program interrupt control bit)

将 EOPIE1 位置“1”可在对存储区 1 执行的编程操作结束时产生中断。仅当 LOCK1 置“0”时，才能编程 EOPIE1。

0：对存储区 1 执行的编程操作结束时不产生中断。

1：对存储区 1 执行的编程操作结束时使能中断。

位 15 **CRC_EN**：存储区 1 CRC 控制位 (Bank 1 CRC control bit)

将 CRC_EN 位置“1”可使能对存储区 1 进行 CRC 计算。CRC_EN 不会启动 CRC 计算，但会通过 FLASH_CRCCR1 寄存器使能 CRC 配置。

如果已在存储区 1 上执行 CRC 计算，则只能通过将 CRC_EN 位置“0”的方式禁止 CRC 计算。复位 CRC_EN 会复位 FLASH_CRCDATAR 寄存器的内容。

仅当 LOCK1 置“0”时，才能编程 CRC_EN。

位 14:11 保留，必须保持复位值

位 10:8 **SNB1**：存储区 1 扇区擦除选择编号 (Bank 1 sector erase selection number)

这些位用于选择扇区擦除操作的目标扇区。仅当 LOCK1 置“0”时，才能编程 SNB1。

000：存储区 1 的扇区 0

001：存储区 1 的扇区 1

...

111：存储区 1 的扇区 7

位 7 **START1**：存储区 1 的存储区或扇区擦除启动控制位 (Bank 1 bank or sector erase start control bit)

START1 位用于启动扇区擦除或存储区擦除操作。仅当 LOCK1 置“0”时，才能编程 START1。相应操作得到确认后，Flash 接口会复位 START1。操作得到确认之前，用户应用程序不能访问任何 Flash 寄存器。

位 6 **FW1**：存储区 1 写入强制控制位 (Bank 1 write forcing control bit)

即使写缓冲区未满，FW1 也会强制执行写操作。仅当 LOCK1 置“0”时，才能编程 FW1。相应操作得到确认后，Flash 接口会复位 FW1。操作得到确认之前，用户应用程序不能访问任何 Flash 寄存器。

仅当写缓冲区非空时，写入强制才有效（特别要说明的是，当连续执行写操作时，FW1 不会启动多次写操作）。

位 5:4 **PSIZE1**：存储区 1 程序大小 (Bank 1 program size)

PSIZE1 会选择对存储区 1 执行写操作和擦除操作过程中 Flash 使用的并行位数（更多详情，请参见*配置编程参数*一节）。仅当 LOCK1 置“0”时，才能编程 PSIZE1。

00：使用字节并行位数执行编程

01：使用半字并行位数执行编程

10：使用字并行位数执行编程

11：使用双字并行位数执行编程

位 3 **BER1**：存储区 1 擦除请求 (Bank 1 erase request)

将 BER1 位置“1”会请求对存储区 1 执行存储区擦除操作。仅当 LOCK1 置“0”时，才能编程 BER1。

BER1 的优先级要高于 SER1：如果两者均置 1，Flash 接口会执行存储区擦除操作（更多详细信息，请参见*标准 Flash 存储区擦除*一节）。

0：未请求对存储区 1 执行存储区擦除操作

1：请求对存储区 1 执行存储区擦除操作

位 2 **SER1**：存储区 1 扇区擦除请求 (Bank 1 sector erase request)

将 SER1 位置“1”会请求对存储区 1 执行扇区擦除操作。仅当 LOCK1 置“0”时，才能编程 SER1。

BER1 的优先级要高于 SER1：如果两者均置 1，Flash 接口会执行存储区擦除操作（更多详细信息，请参见*Flash 扇区擦除*一节）。

0：未请求对存储区 1 执行扇区擦除操作

1：请求对存储区 1 执行扇区擦除操作

位 1 **PG1**：存储区 1 编程使能位 (Bank 1 program enable bit)

将 PG1 位置“1”可使能对存储区 1 执行写操作， 因此，即使正在进行扇区或存储区擦除操作，也可准备编程操作。

仅当 LOCK1 置“0”时，才能编程 PG1。PG1 复位后，会禁止将内部缓冲区用于对存储区 1 执行的写操作，缓冲区中存储的所有尚未编程的数据都会丢失。

位 0 **LOCK1**：存储区 1 配置锁定位 (Bank 1 configuration lock bit)

此位用于锁定 FLASH_CR1 寄存器。

FLASH_CR1 寄存器解锁后，LOCK1 位会自动复位（请参见*配置编程参数*一节）。如果执行顺序不正确，此位会保持锁定状态，直至下一次系统复位。

将 LOCK1 编程为“1”可将其置 1。LOCK1 置“1”后，必须执行新的解锁顺序将其解锁。如果 LOCK1 从“0”变为“1”，FLASH_CR1 寄存器的其他位不会发生变化。

0：FLASH_CR1 寄存器已解锁

1：FLASH_CR1 寄存器已锁定

3.5.5 存储区 1 的 FLASH 状态寄存器 (FLASH_SR1)

FLASH status register for bank 1

偏移地址：0x010

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	CRCEND1	DBECCERR1	SNECCERR1	RDSERR1	RDPERR1	OPERR1	INCERR1	Res.	STRBERR1	PGSERR1	WRPERR1	EOP1
				r	r	r	r	r	r	r		r	r	r	r

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CRC_BUSY1	QW1	WBNE1	BSY1
												r	r	r	r

位 31:28 保留，必须保持复位值

位 27 **CRCEND1**：存储区 1 CRC 完成标志 (Bank 1 CRC-complete flag)

对存储区 1 完成 CRC 计算后，CRCEND1 位会置 1。如果 CRCENDIE1 置“1”，将产生中断。不需要复位 CRCEND1 便可重新开始进行 CRC 计算。向 FLASH_CCR1 寄存器中的 CLR_CRCEND1 位写入“1”可将 CRCEND1 清零。

0：对存储区 1 进行的 CRC 计算未完成

1：对存储区 1 进行的 CRC 计算已完成

位 26 **DBECCERR1**：存储区 1 ECC 双重检测错误标志 (Bank 1 ECC double detection error flag)

如果在对存储区 1 执行读操作的过程中发生 ECC 双重检测错误，DBECCERR1 标志会置 1。如果 DBECCERRIE1 置“1”，将产生中断。向 FLASH_CCR1 寄存器中的 CLR_DBECCERR1 位写入“1”可将 DBECCERR1 清零。

0：存储区 1 未发生 ECC 双重检测错误

1：存储区 1 发生 ECC 双重检测错误

位 25 **SNECCERR11**：存储区 1 单校正错误标志 (Bank 1 single correction error flag)

如果在对存储区 1 执行读操作的过程中发生 ECC 单校正错误，SNECCERR1 标志会置 1。如果 SNECCERRIE1 置“1”，将产生中断。向 FLASH_CCR1 寄存器中的 CLR_SNECCERR1 位写入“1”可将 SNECCERR1 清零。

0：存储区 1 未发生 ECC 单校正错误

1：存储区 1 发生 ECC 单校正错误

位 24 **RDSERR1**：存储区 1 安全错误标志 (Bank 1 secure error flag)

如果存储区 1 发生安全错误（未获得适当权限便访问安全保护字），RDSERR1 标志会置 1。如果 RDSERRIE1 置“1”，将产生中断。向 FLASH_CCR1 寄存器中的 CLR_RDSERR1 位写入“1”可将 RDSERR1 清零。

0：存储区 1 未发生安全错误

1：存储区 1 发生安全错误

位 23 **RDPERR1**：存储区 1 读保护错误标志 (Bank 1 read protection error flag)

如果存储区 1 发生读保护错误（访问受 PCROP 保护的字），RDPERR1 标志会置 1。如果 RDPERRIE1 置“1”，将产生中断。向 FLASH_CCR1 寄存器中的 CLR_RDPERR1 位写入“1”可将 RDPERR1 清零。

0：存储区 1 未发生读保护错误

1：存储区 1 发生读保护错误

位 22 **OPERR1**：存储区 1 写入/擦除错误标志 (Bank 1 write/erase error flag)

如果在对存储区 1 执行写入/擦除操作的过程中出错，OPERR1 标志会置 1。如果 OPERRIE1 置“1”，将产生中断。向 FLASH_CCR1 寄存器中的 CLR_OPERR1 位写入“1”可将 OPERR1 清零。

0：存储区 1 未发生写入/擦除错误

1：存储区 1 发生写入/擦除错误

位 21 **INCERR1**：存储区 1 不一致错误标志 (Bank 1 inconsistency error flag)

如果存储区 1 发生不一致错误，INCERR1 标志会置 1。如果 INCERRIE1 置“1”，将产生中断。向 FLASH_CCR1 寄存器中的 CLR_INCERR1 位写入“1”可将 INCERR1 清零。

请参见*第 3.3.9 节：Flash 接口错误标志*。

0：存储区 1 未发生不一致错误

1：存储区 1 发生不一致错误

位 20 保留，必须保持复位值

位 19 **STRBERR1**：存储区 1 选通错误标志 (Bank 1 strobe error flag)

如果存储区 1 发生选通错误（主机尝试多次将同一字节写入写缓冲区中），STRBERR1 标志会置 1。如果 STRBERRIE1 位置“1”，将产生中断。向 FLASH_CCR1 寄存器中的 CLR_STRBERR1 位写入 “1” 可将 STRBERR1 清零。

0：存储区 1 未发生选通错误

1：存储区 1 发生选通错误

位 18 **PGSERR1**：存储区 1 编程顺序错误标志 (Bank 1 programming sequence error flag)

如果存储区 1 发生顺序错误，PGSERR1 标志会置 1。如果 PGSERRIE1 位置“1”，将产生中断。向 FLASH_CCR1 寄存器中的 CLR_PGSERR1 位写入“1”可将 PGSERR1 清零。

0：存储区 1 未发生顺序错误

1：存储区 1 发生顺序错误

位 17 **WRPERR1**：存储区 1 写保护错误标志 (Bank 1 write protection error flag)

如果在对存储区 1 执行编程操作的过程中发生保护错误，WRPERR1 标志会置 1。如果 EOPIE1 置“1”，也会产生中断。向 FLASH_CCR1 寄存器中的 CLR_EOP1 位写入“1”可将 WRPERR1 清零。

0：存储区 1 未发生保护错误

1：存储区 1 发生保护错误

位 16 **EOP1**：存储区 1 编程结束标志 (Bank 1 end-of-program flag)

对存储区 1 进行的编程操作结束时，EOP1 标志会置 1。如果 EOPIE1 置“1”，将产生中断。不需要先复位 EOP1 便可开始执行新操作。向 FLASH_CCR1 寄存器中的 CLR_EOP1 位写入“1”可将 EOP1 位清零。

0：存储区 1 上的编程操作未结束

1：存储区 1 上的编程操作已结束

位 15:4 保留，必须保持复位值

位 3 **CRC_BUSY1**：存储区 1 CRC 忙标志 (Bank 1 CRC busy flag)

正在对存储区 1 进行 CRC 计算时，CRC_BUSY1 标志会置 1。此位不能强制设为“0”。用户必须等待存储区 1 的 CRC 计算完成或禁止对存储区 1 进行 CRC 计算。

0：当前未对存储区 1 执行 CRC 计算

1：正在对存储区 1 执行 CRC 计算

位 2 **QW1**：存储区 1 等待队列标志 (Bank 1 wait queue flag)

对存储区 1 进行的编程操作位于等待队里中时，QW1 标志会置 1。无法确定队列中是哪种类型的编程操作。所有编程操作均已执行完毕并从等待队列中删除时，此标志会由硬件复位。此位不能强制设为“0”。如果未请求执行其他操作，此位会在确定时间后复位。

0：没有编程操作在存储区 1 的操作队列中等待

1：至少有一个编程操作在存储区 1 的操作队列中等待

位 1 **WBNE1**：存储区 1 写缓冲区非空标志 (Bank 1 write buffer not empty flag)

如果存储区 1 写缓冲区不为空，并且 Flash 接口正在等待新数据填入，WBNE1 标志会置 1。每次写缓冲区清空时，WBNE1 会由硬件复位。只要发生以下任一事件，就会进行清空：

- 写缓冲区已满
- 用户强制执行写操作
- 发生了涉及到数据丢失的错误
- 写操作禁止。

此位不能强制设为“0”。要复位此位，应执行上述任一操作来清空写缓冲区。

0：存储区 1 的写缓冲区为空

1：存储区 1 的写缓冲区正在等待数据填入

位 0 **BSY1**：存储区 1 正在进行编程标志 (Bank 1 ongoing program flag)
正在对存储区 1 进行编程操作时，BSY1 标志会置 1。无法确定正在进行哪种类型的编程操作。
BSY1 不能强制设为“0”。操作完成后，如果未开始执行其他编程操作，则会由硬件复位此位。
0：未在对存储区 1 执行编程操作
1：正在对存储区 1 执行编程操作

3.5.6 存储区 1 的 FLASH 清零控制寄存器 (FLASH_CCR1)

FLASH clear control register for bank 1

偏移地址：0x014

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	CLR_CRCEND1	CLR_DBECCERR1	CLR_SNECCERR1	CLR_RDSERR1	CLR_RDPERR1	CLR_OPERR1	CLR_INCERR1	Res.	CLR_STRBERR1	CLR_PGSERR1	CLR_WRPERR1	CLR_EOP1
				w	w	w	w	w	w	w		w	w	w	w

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

位 31:28 保留，必须保持复位值

位 27 **CLR_CRCEND1**：存储区 1 CRCEND1 标志清零位 (Bank 1 CRCEND1 flag clear bit)
将此位置“1”可将 FLASH_SR1 寄存器的 CRCEND1 标志复位为“0”。

位 26 **CLR_DBECCERR1**：存储区 1 DBECCERR1 标志清零位 (Bank 1 DBECCERR1 flag clear bit)
将此位置“1”可将 FLASH_SR1 寄存器的 DBECCERR1 标志复位为“0”。如果 FLASH_SR1 寄存器的 SNECCERR1 标志置“0”，FLASH_ECC_FA1R 寄存器也会复位为“0”。

位 25 **CLR_SNECCERR1**：存储区 1 SNECCERR1 标志清零位 (Bank 1 SNECCERR1 flag clear bit)
将此位置“1”可将 FLASH_SR1 寄存器的 SNECCERR1 标志复位为“0”。如果 FLASH_SR1 寄存器的 DBECCERR1 标志置“0”，FLASH_ECC_FA1R 寄存器也会复位为“0”。

位 24 **CLR_RDSERR1**：存储区 1 RDSERR1 标志清零位 (Bank 1 RDSERR1 flag clear bit)
将此位置“1”可将 FLASH_SR1 寄存器的 RDSERR1 标志复位为“0”。

位 23 **CLR_RDPERR1**：存储区 1 RDPERR1 标志清零位 (Bank 1 RDPERR1 flag clear bit)
将此位置“1”可将 FLASH_SR1 寄存器的 RDPERR1 标志复位为“0”。

位 22 **CLR_OPERR1**：存储区 1 OPERR1 标志清零位 (Bank 1 OPERR1 flag clear bit)
将此位置“1”可将 FLASH_SR1 寄存器的 OPERR1 标志复位为“0”。

位 21 **CLR_INCERR1**：存储区 1 INCERR1 标志清零位 (Bank 1 INCERR1 flag clear bit)
将此位置“1”可将 FLASH_SR1 寄存器的 INCERR1 标志复位为“0”。

位 20 保留，必须保持复位值

位 19 **CLR_STRBERR1**：存储区 1 STRBERR1 标志清零位 (Bank 1 STRBERR1 flag clear bit)
将此位置“1”可将 FLASH_SR1 寄存器的 STRBERR1 标志复位为“0”。

位 18 **CLR_PGSERR1**：存储区 1 PGSERR1 标志清零位 (Bank 1 PGSERR1 flag clear bit)
将此位置“1”可将 FLASH_SR1 寄存器的 PGSERR1 标志复位为“0”。

位 17 **CLR_WRPERR1**：存储区 1 WRPERR1 标志清零位 (Bank 1 WRPERR1 flag clear bit)
将此位置“1”可将 FLASH_SR1 寄存器的 WRPERR1 标志复位为“0”。

位 16 **CLR_EOP1**：存储区 1 EOP1 标志清零位 (Bank 1 EOP1 flag clear bit)
将此位置“1”可将 FLASH_SR1 寄存器的 EOP1 标志复位为“0”。

位 15:0 保留，必须保持复位值

3.5.7 Flash 选项控制寄存器 (FLASH_OPTCR)

FLASH option control register

偏移地址：0x018 或 0x118

复位值：0x0000 0001

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MER	Res.	OPTSTART	OPTLOCK
												w		rw	rs

位 31 **SWAP_BANK**：存储区交换配置位 (Bank swapping configuration bit)
SWAP_BANK 用于控制是否交换存储区 1 和存储区 2。选项字节加载后，此位会载入 FLASH_OPTSR_CUR 寄存器的 SWAP_BANK_OPT 位。FLASH_OPTCR 寄存器解锁（OPTLOCK = “0”）后，主机可修改 SWAP_BANK 来交换/取消交换用户 Flash 存储区（请参见*第 3.3.13 节：Flash 存储区交换*）。
0：不交换存储区 1 和存储区 2
1：交换存储区 1 和存储区 2

位 30 **OPTCHANGEERRIE**：选项字节更改错误中断使能位 (Option byte change error interrupt enable bit)
OPTCHANGEERRIE 位用于控制选项字节更改期间出错时是否产生中断。
0：选项字节更改期间出错时不产生中断
1：选项字节更改期间出错时产生中断。

位 29:5 保留，必须保持复位值

位 4 **MER**：Flash 批量擦除使能位 (Flash mass erase enable bit)

要编程 MER 位，FLASH_CR1、FLASH_CR2 和 FLASH_OPTCR 寄存器必须提前解锁。

将 MER 位编程为“1”会自动将 BER1、BER2、START1 和 START2 置“1”， 从而允许同时对两个存储区进行批量擦除操作。

位 3:2 保留，必须保持复位值

位 1 **OPTSTART**：选项字节开始更改选项配置位 (Option byte start change option configuration bit)

OPTSTART 可触发选项字节更改操作。仅当 OPTLOCK 位置“0”时，用户才能将 OPTSTART 置 1。选项字节更改操作得到确认后，Flash 接口会复位 OPTSTART。

操作得到确认之前，用户应用程序不能访问任何 Flash 寄存器。

将此位置 1 之前，用户需要将所需值写入到 FLASH_XXX_PRG 寄存器中。FLASH_XXX_PRG 寄存器将保持锁定，直至选项字节更改操作已在 Flash 中执行。

如果正在对存储区 1 或存储区 2 执行 CRC 计算，则不能开始执行选项字节更改操作：如果在 FLASH_SR1/2 寄存器的 CRC_BUSY1/2 置 1 的情况下尝试将 OPTSTART 置 1，则不会有任何作用；选项字节更改操作不会开始，并且不会生成错误。

位 0 **OPTLOCK**：FLASH_OPTCR 锁定选项配置位 (FLASH_OPTCR lock option configuration bit)

OPTLOCK 位用于锁定 FLASH_OPTCR 寄存器。FLASH_OPTCR 解锁后，OPTLOCK 会自动复位（请参见*配置编程参数*一节）。如果执行顺序不正确，此位会保持锁定状态，直至下一次系统复位。

将 OPTLOCK 编程为“1”可将其置 1。LOCK1 置“1”后，必须执行新的解锁顺序将其解锁。如果 OPTLOCK 从“0”变为“1”，FLASH_OPTCR 寄存器的其他位不会发生变化。

0：FLASH_OPTCR 寄存器已解锁

1：FLASH_OPTCR 寄存器已锁定。

3.5.8 FLASH 选项状态寄存器（当前值）(FLASH_OPTSR_CUR)

FLASH option status register

偏移地址：0x01C 或 0x11C

复位值：0xXXXX XXXX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SWAP_BANK_OPT	OPTCHANGEERR	IO_HSLV	PERSO_OK	RSS2	RSS1	Res.	Res.	Res.	Res.	SECURITY	ST_RAM_SIZE		FZ_IWDG_SDBY	FZ_IWDG_STOP	Res.
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RDP								nRST_STBY_D1	nRST_STOP_D1	Res.	IWDG1_HW	BOR_LEV		Res.	OPT_BUSY
r	r	r	r	r	r	r	r	r	r	r	r	r	r		r

位 31 **SWAP_BANK_OPT**：存储区交换选项状态位 (Bank swapping option status bit)

SWAP_BANK_OPT 反映配置存储区 1/2 交换默认值的相应选项位的值。

0：自举加载后，不交换用户扇区

1：自举加载后，交换用户扇区

位 30 **OPTCHANGEERR**：选项字节更改错误标志 (Option byte change error flag)

OPTCHANGEERR 标志指示选项字节更改操作期间出错。如果 OPTCHANGEERR 置“1”，则表示选项字节更改操作未成功完成。如果 FLASH_OPTCR 寄存器的 OPTCHANGEERRIE 位置“1”，则会在此标志置 1 时产生中断。

向寄存器 FLASH_OPTCCR 的 CLR_OPTCHANGEERR 写入“1”会将 OPTCHANGEERR 清零。

0：未发生选项字节更改错误

1：选项字节更改操作期间发生一个或多个错误

位 29 **IO_HSLV**：低压条件下 I/O 高速状态位 (PRODUCT_BELOW_27V) (I/O high-speed at low-voltage status bit (PRODUCT_BELOW_27V))

此位可指示产品工作电压是否低于 2.7 V。

0：产品在全电压范围内工作，禁止在低压条件下进行 I/O 速度优化

1：产品工作电压低于 2.7 V，允许在低压条件下进行 I/O 速度优化

位 28 **PERSO_OK**：器件个性化状态位 (Device personalization status bit)

PERSO_OK 指示器件是否已进行个性化设置。

位 27 **RSS2**：用户选项状态位 2 (User option status bit 2)

RSS2 用于 ST 开发代码（RSS/ 自举程序）。

位 26 **RSS1**：用户选项状态位 1 (User option status bit 1)

RSS1 用于 ST 开发代码（RSS/ 自举程序）。

位 25:22 保留，必须保持复位值

位 21 **SECURITY**：安全使能选项状态位 (Security enable option status bit)

0：禁止安全功能

1：使能安全功能。

位 20:19 **ST_RAM_SIZE**：DTCM RAM 大小选项状态 (DTCM RAM size option status)

00：2 KB

01：4 KB

10：8 KB

11：16 KB

注： 仅当使能了安全功能（SECURITY = “1”）时，此位域才有效。

位 18 **FZ_IWDG_SDBY**：IWDG 待机模式冻结选项状态位 (IWDG Standby mode freeze option status bit)

此位反映了 IWDG_FZ_STOP 选项位的冻结状态。

0：在待机模式下冻结独立看门狗

1：在待机模式下运行独立看门狗

位 17 **FZ_IWDG_STOP**：IWDG 停止模式冻结选项状态位 (IWDG Stop mode freeze option status bit)

此位反映了 IWDG_FZ_STANDBY 选项位的冻结状态。

0：在停止模式下冻结独立看门狗

1：在停止模式下运行独立看门狗

位 16 保留，必须保持复位值

位 15:8 **RDP**：读保护级别选项状态字节 (Readout protection level option status byte)
关于读保护级别的详细信息，请参见*RDP 读保护*一节。提供三种不同的级别：
0xAA：保护级别 0
0xCC：保护级别 2
其他值：保护级别 1。

位 7 **nRST_STBY_D1**：D1 进入 DStandby 模式的复位选项状态位 (D1 DStandby entry reset option status bit)
0：D1 域进入 DStandby 模式时产生复位
1：不产生复位

位 6 **nRST_STOP_D1**：D1 进入 DStop 模式的复位选项状态位 (D1 DStop entry reset option status bit)
0：D1 域进入 DStop 模式时产生复位
1：不产生复位

位 5 保留，必须保持复位值

位 4 **IWDG1_HW**：IWDG1 控制选项状态位 (IWDG1 control option status bit)
0：IWDG1 通过软件控制
1：IWDG1 看门狗通过硬件控制。

位 3:2 **BOR_LEV**：欠压电压选项状态位 (Brownout level option status bit)
这些选项位用于定义生成系统复位的电源电压。
00：复位电压设为 2.1 V
01：复位电压设为 2.4 V
10：复位电压设为 2.7 V
11：复位电压设为 2.1 V（作为 00 配置）。

位 1 保留，必须保持复位值

位 0 **OPT_BUSY**：正在进行选项字节更改标志 (Option byte change ongoing flag)
OPT_BUSY 指示是否正在进行选项字节更改。如果此位置“1”，说明 Flash 接口正在执行选项更改，不能修改任何 Flash 寄存器。
0：未在进行选项字节更改
1：正在进行选项字节更改，选项字节更改完成之前，会阻止对 Flash 寄存器的所有写访问。

3.5.9 FLASH 选项状态寄存器（要编程的值）(FLASH_OPTSR_PRG)

FLASH option status register

偏移地址：0x020 或 0x120

复位值：0xXXXX XXXX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SWAP_BANK_OPT	Res.	IO_HSLV	Res.	RSS2	RSS1	Res.	Res.	Res.	Res.	SECURITY	ST_RAM_SIZE		FZ_IWDG_SDBY	FZ_IWDG_STOP	Res.
rw		rw		rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RDP								nRST_STBY_D1	nRST_STOP_D1	Res.	IWDG1_HW	BOR_LEV		Res.	Res.
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw		

位 31 **SWAP_BANK_OPT**：存储区交换选项配置位 (Bank swapping option configuration bit)

SWAP_BANK_OPT 选项位用于配置存储区 1/2 交换的默认值。

0：自举加载后，不交换用户扇区

1：自举加载后，交换用户扇区。

位 30 保留，必须保持复位值

位 29 **IO_HSLV**：低压条件下 I/O 高速 (PRODUCT_BELOW_27V) (I/O high-speed at low-voltage (PRODUCT_BELOW_27V))

此位可指示产品工作电压是否低于 2.7 V。仅当产品电源电压低于 2.7 V 时，才可将此位置 1。

0：产品在全电压范围内工作，禁止在低压条件下进行 I/O 速度优化

1：产品工作电压低于 2.7 V，允许在低压条件下进行 I/O 速度优化

位 28 保留，必须保持复位值

位 27 **RSS2**：用户选项配置位 2 (User option configuration bit 2)

RSS2 用于 ST 开发代码（RSS/ 自举程序）。修改此位不会产生影响。

位 26 **RSS1**：用户选项配置位 1 (User option configuration bit 1)

RSS1 用于 ST 开发代码（RSS/ 自举程序）。修改此位不会产生影响。

位 25:22 保留，必须保持复位值

位 21 **SECURITY**：安全选项配置位 (Security option configuration bit)

SECURITY 位可在选项字节更改期间在器件级使能安全功能。更改将在下次上电复位时应用。使能安全功能后，如果不存在受 PCROP 或安全模式保护的区域，可禁止安全功能。如果存在安全保护或 PCROP 保护区域，需执行降级（从级别 1 变为级别 0）并将所有位置 1 以取消对安全区域和 PCROP 区域的保护（请参见*第 3.3.12 节：保护机制*）。

0：禁止安全功能

1：使能安全功能。

位 20:19 **ST_RAM_SIZE**：DTCM 大小选择选项配置位 (DTCM size select option configuration bits)

ST_RAM_SIZE 位在选项字节更改期间使用，用于设置要保护的 DTCM RAM 的大小。

00：2 KB

01：4 KB

10：8 KB

11：16 KB

注： 仅当使能了安全功能（SECURITY = “1”）时，此位域才有效。

位 18 **FZ_IWDG_SDBY**：IWDG 待机模式冻结选项配置位 (IWDG Standby mode freeze option configuration bit)

FZ_IWDG_SDBY 在选项字节更改期间使用，用于选择是否在待机模式下冻结独立 看门狗。

0：在待机模式下冻结独立看门狗

1：在待机模式下运行独立看门狗

位 17 **FZ_IWDG_STOP**：IWDG 停止模式冻结选项配置位 (IWDG Stop mode freeze option configuration bit)

FZ_IWDG_STOP 在选项更改期间使用，用于选择是否在停止模式下冻结独立 看门狗。

0：在停止模式下冻结独立看门狗

1：在停止模式下运行独立看门狗

位 16 保留，必须保持复位值

位 15:8 **RDP**：读保护级别选项配置字节 (Readout protection level option configuration byte)

RDP 位用于更改读保护级别。仅当当前保护级别不是级别 2 时，才能进行此更改（请参见 *RDP 读保护* 一节）。可能的配置包括：

0xAA：设置保护级别 0

0xCC：设置保护级别 2

其他所有值：设置保护级别 1。

位 7 **nRST_STBY_D1**：D1 进入 DStandby 模式后选项字节擦除选项配置位 (Option byte erase after D1 DStandby option configuration bit)

nRST_STBY_D1 在选项字节更改期间使用。如果置“1”，则 D1 域进入 DStandby 模式时会擦除选项字节。

0：D1 域进入 DStandby 模式时擦除选项字节

1：D1 域进入 DStandby 模式时不擦除选项字节

位 6 **nRST_STOP_D1**：D1 进入 DStop 模式后选项字节擦除选项配置位 (Option byte erase after D1 DStop option configuration bit)

nRST_STOP_D1 在选项字节更改期间使用。如果置“1”，D1 域进入 DStop 模式时会擦除选项字节。

0：D1 域进入 DStop 模式时擦除选项字节

1：D1 域进入 DStop 模式时不擦除选项字节

位 5 保留，必须保持复位值

位 4 **IWDG1_HW**：IWDG1 选项配置位 (IWDG1 option configuration bit)
IWDG1_HW 选项位用于选择 IWDG1 独立看门狗需要由硬件控制还是软件控制。
0：IWDG1 通过软件控制
1：IWDG1 通过硬件控制。

位 3:2 **BOR_LEV**：BOR 复位电压选项配置位 (BOR reset level option configuration bits)
FLASH_OPTSR_PRG 用于更改 BOR_LEV 选项字节。要修改此选项，用户必须先将 BOR_LEV 编程为所需配置，然后才能开始进行选项字节更改。可能的配置包括：
00 和 11：复位电压设为 2.1 V
01：复位电压设为 2.4 V
10：复位电压设为 2.7 V

位 1:0 保留，必须保持复位值

3.5.10 FLASH 选项清空控制寄存器 (FLASH_OPTCCR)

FLASH option clear control register

偏移地址：0x024 或 0x124

复位值：0x0000 0000

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

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

位 31 保留，必须保持复位值

位 30 **CLR_OPTCHANGEERR**：OPTCHANGEERR 复位位 (OPTCHANGEERR reset bit)
FLASH_OPTCCR 用于复位 FLASH_OPTSR_CUR 寄存器的 OPTCHANGEERR 标志。
FLASH_OPTCCR 为只写。
将其编程为“1”可实现复位。

位 29:0 保留，必须保持复位值

3.5.11 存储区 1 的 FLASH 保护地址（当前值）(FLASH_PRAR_CUR1)

FLASH protection address for bank 1

偏移地址：0x028

复位值：0xXXXX XXXX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DMEP1	Res.	Res.	Res.	PROT_AREA_END1											
r				r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	PROT_AREA_START1											
				r	r	r	r	r	r	r	r	r	r	r	r

位 31 **DMEP1**：存储区 1 PCROP 保护擦除使能选项状态位 (Bank 1 PCROP protected erase enable option status bit)

如果 DMEP1 置“1”，发生保护降级（从级别 1 变为级别 0）时，会擦除 PCROP 保护区（请参见 *PCROP 区域（专有代码读保护，仅执行区域）*一节和 *标准 Flash 存储区擦除*一节）。

位 30:28 保留，必须保持复位值

位 27:16 **PROT_AREA_END1**：存储区 1 最高 PCROP 保护地址 (Bank 1 highest PCROP protected address)

这些位包含存储区 1 中的最后一个 PCROP 保护地址。
如果此地址等于 PROT_AREA_START1，整个存储区 1 会受到 PCROP 保护。
如果此地址小于 PROT_AREA_START1，则不会对存储区 1 设置任何保护功能。

位 15:12 保留，必须保持复位值

位 11:0 **PROT_AREA_START1**：存储区 1 最低 PCROP 保护地址 (Bank 1 lowest PCROP protected address)

这些位包含存储区 1 中的第一个 PCROP 保护地址。
如果此地址等于 PROT_AREA_END1，整个存储区 1 会受到 PCROP 保护。
如果此地址大于 PROT_AREA_END1，则不会对存储区 1 设置任何保护功能。

3.5.12 存储区 1 的 FLASH 保护地址（要编程的值）(FLASH_PRAR_PRG1)

FLASH protection address for bank 1

偏移地址：0x02C

复位值：0xXXXX XXXX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DMEP1	Res.	Res.	Res.	PROT_AREA_END1											
rw				rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	PROT_AREA_START1											
				rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31 **DMEP1**：存储区 1 PCROP 保护擦除使能选项配置位 (Bank 1 PCROP protected erase enable option configuration bit)

如果 DMEP1 置“1”，发生保护降级（从级别 1 变为级别 0）或存储区 1 擦除时，会擦除 PCROP 保护区（请参见*PCROP 区域（专有代码读保护，仅执行区域）*一节）。

位 30:28 保留，必须保持复位值

位 27:16 **PROT_AREA_END1**：存储区 1 最高 PCROP 保护地址配置 (Bank 1 highest PCROP protected address configuration)

这些位允许配置存储区 1 中的最后一个 PCROP 保护地址。

位 15:12 保留，必须保持复位值

位 11:0 **PROT_AREA_START1**：存储区 1 最低 PCROP 保护地址配置 (Bank 1 lowest PCROP protected address configuration)

这些位允许配置存储区 1 中的第一个 PCROP 保护地址。

3.5.13 存储区 1 的 FLASH 安全地址（当前值）(FLASH_SCAR_CUR1)

FLASH secure address for bank 1

偏移地址：0x030

复位值：0xXXXX XXXX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DMES1	Res.	Res.	Res.	SEC_AREA_END1											
r				r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	SEC_AREA_START1											
				r	r	r	r	r	r	r	r	r	r	r	r

位 31 **DMES1**：存储区 1 安全保护擦除使能选项状态位 (Bank 1 secure protected erase enable option status bit)

如果 DMES1 置“1”，发生保护降级（从级别 1 变为级别 0）时，会擦除安全保护区（请参见*安全区域*一节和*标准 Flash 存储区擦除*一节）。

位 30:28 保留，必须保持复位值

位 27:16 **SEC_AREA_END1**：存储区 1 最高安全保护地址 (Bank 1 highest secure protected address)

这些位包含存储区 1 中的最后一个安全保护地址。
如果此地址等于 SEC_AREA_START1，整个存储区 1 会受到安全保护。
如果此地址小于 SEC_AREA_START1，则不会对存储区 1 设置任何保护功能。

位 15:12 保留，必须保持复位值

位 11:0 **SEC_AREA_START1**：存储区 1 最低安全保护地址 (Bank 1 lowest secure protected address)

这些位包含存储区 1 中的第一个安全保护地址。
如果此地址等于 SEC_AREA_END1，整个存储区 1 会受到安全保护。
如果此地址大于 SEC_AREA_END1，则不会对存储区 1 设置任何保护功能。

3.5.14 存储区 1 的 FLASH 安全地址（要编程的值）(FLASH_SCAR_PRG1)

FLASH secure address for bank 1

偏移地址：0x034

复位值：0xXXXX XXXX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DMES1	Res.	Res.	Res.	SEC_AREA_END1											
rw				rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	SEC_AREA_START1											
				rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31 **DMES1**：存储区 1 安全保护擦除使能选项配置位 (Bank 1 secure protected erase enable option configuration bit)

如果 DMES1 置“1”，发生保护降级（从级别 1 变为级别 0）或存储区 1 擦除时，会擦除安全保护区（请参见*安全区域*一节）。

位 30:28 保留，必须保持复位值

位 27:16 **SEC_AREA_END1**：存储区 1 最高安全保护地址配置 (Bank 1 highest secure protected address configuration)

这些位允许配置存储区 1 中的最后一个安全保护地址。

位 15:12 保留，必须保持复位值

位 11:0 **SEC_AREA_START1**：存储区 1 最低安全保护地址配置 (Bank 1 lowest secure protected address configuration)

这些位允许配置存储区 1 中的第一个安全保护地址。

3.5.15 存储区 1 的 FLASH 扇区写保护（当前值）(FLASH_WPSN_CUR1R)

FLASH write sector protection for bank 1

偏移地址：0x038

复位值：0x0000 00XX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WRPSn1							
								r	r	r	r	r	r	r	r

位 31:8 保留，必须保持复位值

位 7:0 **WRPSn1**：存储区 1 扇区写保护选项状态字节 (Bank 1 sector write protection option status byte)
每个 FLASH_WPSN_CUR1R 位用于反映存储区 1 相应扇区的写保护状态（请参见*扇区写保护*一节）。

3.5.16 存储区 1 的 FLASH 扇区写保护（要编程的值）(FLASH_WPSN_PRG1R)

FLASH write sector protection for bank 1

偏移地址：0x03C

复位值：0x0000 00XX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WRPSn1							
								rw	rw	rw	rw	rw	rw	rw	rw

位 31:8 保留，必须保持复位值

位 7:0 **WRPSn1**：存储区 1 扇区写保护配置字节 (Bank 1 sector write protection configuration byte)
将 WRPSn1 位置“0”可对存储区 1 的相应扇区进行写保护（请参见*扇区写保护*一节）。

3.5.17 包含自举地址的 FLASH 寄存器（当前值）(FLASH_BOOT_CURR)

FLASH register with boot address

偏移地址：0x040

复位值：0xXXXX XXXX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
BOOT_ADD1															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BOOT_ADD0															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:16 BOOT_ADD1：自举地址 1 (Boot address 1)
这些位用于反映 BOOT 引脚为高电平时自举地址的 MSB。

位 15:0 BOOT_ADD0：自举地址 0 (Boot address 0)
这些位用于反映 BOOT 引脚为低电平时自举地址的 MSB。

3.5.18 包含自举地址的 FLASH 寄存器（要编程的值）(FLASH_BOOT_PRGR)

FLASH register with boot address

偏移地址：0x044

复位值：0xXXXX XXXX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
BOOT_ADD1															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BOOT_ADD0															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:16 BOOT_ADD1：自举地址 1 配置 (Boot address 1 configuration)
这些位可配置 BOOT 引脚为高电平时自举地址的 MSB。

位 15:0 BOOT_ADD1：自举地址 0 配置 (Boot address 0 configuration)
这些位可配置 BOOT 引脚为低电平时自举地址的 MSB。

3.5.19 存储区 1 的 FLASH CRC 控制寄存器 (FLASH_CRCCR1)

FLASH CRC control register for bank 1

偏移地址：0x050

复位值：0x001C 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CRC_BURST		Res.	Res.	CLEAN_CRC	START_CRC
										rw	rw			w	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	CLEAN_SECT	ADD_SECT	CRC_BY_SECT	ALL_BANK	Res.	Res.	Res.	Res.	CRC_SECT		
					w	w	rw	w					rw	rw	rw

位 31:22 保留，必须保持复位值

位 21:20 **CRC_BURST**：存储区 1 CRC 突发大小 (Bank 1 CRC burst size)

CRC_BURST 位用于设置 CRC 计算单元生成的突发大小。

00：每个突发的大小为 4 个 Flash 字

01：每个突发的大小为 16 个 Flash 字

10：每个突发的大小为 64 个 Flash 字

11：每个突发的大小为 256 个 Flash 字

位 19:18 保留，必须保持复位值

位 17 **CLEAN_CRC**：存储区 1 CRC 清空位 (Bank 1 CRC clear bit)

将 CLEAN_CRC 置“1”会将 FLASH_CRCDATAR 寄存器中存储的当前 CRC 结果清空。

位 16 **START_CRC**：存储区 1 CRC 起始位 (Bank 1 CRC start bit)

START_CRC 位会触发使用当前配置对存储区 1 进行 CRC 计算。如果正在进行选项字节更改操作，则不能开始 CRC 计算，这是因为选项字节更改操作完成之前，对 Flash 寄存器执行的所有写访问都会挂起。

位 15:13 保留，必须保持复位值

位 12:11 保留，必须保持复位值

位 10 **CLEAN_SECT**：存储区 1 CRC 扇区列表清空位 (Bank 1 CRC sector list clear bit)

将 CLEAN_SECT 置“1”会清空进行 CRC 计算的扇区列表。

位 9 **ADD_SECT**：存储区 1 CRC 扇区选择位 (Bank 1 CRC sector select bit)

将 ADD_SECT 置“1”会将编号为 CRC_SECT 的扇区添加到进行 CRC 计算的扇区列表中。

位 8 **CRC_BY_SECT**：存储区 1 CRC 扇区模式选择位 (Bank 1 CRC sector mode select bit)

如果 CRC_BY_SECT 置“1”，则会在扇区级对由 CRC_SECT 选择的扇区或全部存储区（如果 ALL_BANK 位置 1）执行 CRC 计算。

如果 CRC_BY_SECT 复位为“0”，则会对 CRC_START_ADDR 和 CRC_END_ADDR 之间的所有地址执行 CRC 计算。

位 7 **ALL_BANK**：存储区 1 CRC 选择位 (Bank 1 CRC select bit)

如果 ALL_BANK 置“1”，存储区 1 的所有用户扇区都会添加到执行 CRC 计算的扇区列表中。

位 6:3 保留，必须保持复位值

位 2:0 **CRC_SECT**：存储区 1 CRC 扇区编号 (Bank 1 CRC sector number)

CRC_SECT 用于选择一个或多个要添加到 CRC 计算的扇区。可在两个地址（起始地址和结束地址）之间执行 CRC 计算，也可以对扇区列表执行 CRC 计算。如果选择第二种方法，可在 CRC_SECT 中编程扇区编号，然后将 ADD_SECT 置“1”，以此将扇区添加到扇区列表中。

要擦除扇区列表，可将 CLEAN_SECT 位置 1，也可以禁止 CRC 计算。有关各扇区编号的信息，请参见对 FLASH_CR 寄存器中的 SNB1 位的说明（请参见 *第 3.5.4 节：存储区 1 的 FLASH 控制寄存器 (FLASH_CR1)*）。

仅当 FLASH_CR 寄存器的 CRC_EN 置“1”时，才能设置 CRC_SECT。

3.5.20 存储区 1 的 FLASH CRC 起始地址寄存器 (FLASH_CRCSADD1R)

FLASH CRC start address register for bank 1

偏移地址：0x054

复位值：0x0000 0000

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

位 31:0 **CRC_START_ADDR**：存储区 1 上的 CRC 起始地址 (CRC start address on bank 1)
当 CRC_BY_SECT 为“0”时，会使用 CRC_START_ADDR， 必须将它编程为执行 CRC 计算的存储区 1 的起始地址。

3.5.21 存储区 1 的 FLASH CRC 结束地址寄存器 (FLASH_CRCEADD1R)

FLASH CRC end address register for bank 1

偏移地址：0x058

复位值：0x0000 0000

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

位 31:0 **CRC_END_ADDR**：存储区 1 上的 CRC 结束地址 (CRC end address on bank 1)
当 CRC_BY_SECT 为“0”时，会使用 CRC_END_ADDR， 必须将它编程为执行 CRC 计算的存储区 1 的结束地址

3.5.22 FLASH CRC 数据寄存器 (FLASH_CRCDATAR)

FLASH CRC data register

偏移地址：0x05C 或 0x15C

复位值：0x0000 0000

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

位 31:0 **CRC_DATA**：CRC 结果 (CRC result)
CRC_DATA 包含 CRC 计算的结果。

3.5.23 存储区 1 的 FLASH ECC 失效地址 FLASH_ECC_FA1R)

FLASH ECC fail address for bank 1

偏移地址：0x060

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	FAIL_ECC_ADDR1														
	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:15 保留，必须保持复位值

位 14:0 **FAIL_ECC_ADDR1**：存储区 1 ECC 错误地址 (Bank 1 ECC error address)
如果对存储区 1 执行读操作时出现 ECC 错误（单校正错误或双重检测错误），FAIL_ECC_ADDR1 位域会包含产生此错误的地址。
如果 FLASH_SR1 寄存器（CLR_SNECCERR1 或 CLR_DBECCERR1）中的错误标志复位，FAIL_ECC_ADDR1 也会复位。
仅当没有 ECC 错误标志置 1 时，Flash 接口才会在该寄存器中编程地址。这意味着仅会保存第一个生成 ECC 错误的地址。

3.5.24 存储区 2 的 FLASH 密钥寄存器 (FLASH_KEYR2)

FLASH key register for bank 2

偏移地址：0x104

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
KEYR2															
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
KEYR2															
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

位 31:0 **KEYR2**：存储区 2 访问配置解锁密钥 (Bank 2 access configuration unlock key)

FLASH_KEYR2 为只写寄存器。要将 FLASH_CR2 寄存器解锁并允许对其执行编程/擦除操作，必须顺序编程以下值：

a) 第 1 个密钥 = 0x4567 0123

b) 第 2 个密钥 = 0xCDEF 89AB

有关更多详细信息，请参见*配置编程参数*一节。

3.5.25 存储区 2 的 FLASH 控制寄存器 (FLASH_CR2)

FLASH control register for bank 2

偏移地址：0x10C

复位值：0x0000 0031

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	CRCENDIE2	DBECCERRIE2	SNECCERRIE2	RDSERRIE2	RDPERRIE2	OPERRIE2	INCERRIE2	Res.	STRBERRIE2	PGSERRIE2	WRPERRIE2	EOPIE2
				rw	rw	rw	rw	rw	rw	rw		rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CRC_EN	Res.	Res.	Res.	Res.	SNB2			START2	FW2	PSIZE2		BER2	SER2	PG2	LOCK2
rw					rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rs

位 31:28 保留，必须保持复位值

位 27 **CRCENDIE2**：存储区 2 CRC 计算结束中断使能位 (Bank 2 end of CRC calculation interrupt enable bit)

如果 CRCENDIE2 位置“1”，对存储区 2 的 CRC 计算完成时，会产生中断。仅当 LOCK2 置“0”时，才能编程 CRCENDIE2。

0：对存储区 2 进行的 CRC 计算完成时不产生中断

1：对存储区 2 进行的 CRC 计算完成时产生中断

位 26 **DBECCERRIE2**：存储区 2 ECC 双重检测错误中断使能位 (Bank 2 ECC double detection error interrupt enable bit)

如果 DBECCERRIE2 位置“1”，对存储区 2 执行读操作过程中发生 ECC 双重检测错误时，会产生中断。仅当 LOCK2 置“0”时，才能编程 DBECCERRIE2。

0：存储区 2 发生 ECC 双重检测错误时不产生中断

1：存储区 2 发生 ECC 双重检测错误时产生中断

位 25 **SNECCERRIE2**：存储区 2 ECC 单校正错误中断使能位 (Bank 2 ECC single correction error interrupt enable bit)

如果 SNECCERRIE2 位置“1”，对存储区 2 执行读操作过程中发生 ECC 单校正错误时，会产生中断。仅当 LOCK2 置“0”时，才能编程 SNECCERRIE2。

0：存储区 2 发生 ECC 单校正错误时不产生中断

1：存储区 2 发生 ECC 单校正错误时产生中断

位 24 **RDSERRIE2**：存储区 2 安全错误中断使能位 (Bank 2 secure error interrupt enable bit)

如果 RDSERRIE2 位置“1”，在对存储区 2 执行读操作的过程中发生安全错误（未获得适当权限便访问安全保护地址）时，会产生中断。仅当 LOCK2 置“0”时，才能编程 RDSERRIE2。

0：存储区 2 发生安全错误时不产生中断

1：存储区 2 发生安全错误时产生中断

位 23 **RDPERRIE2**：存储区 2 读保护错误中断使能位 (Bank 2 read protection error interrupt enable bit)

如果 RDPERRIE2 位置“1”，在对存储区 2 执行读操作的过程中发生读保护错误（访问受 PCROP 保护的地址）时，会产生中断。仅当 LOCK2 置“0”时，才能编程 RDPERRIE2。

0：存储区 2 发生读保护错误时不产生中断

1：存储区 2 发生读保护错误时产生中断

位 22 **OPERRIE2**：存储区 2 写入/擦除错误中断使能位 (Bank 2 write/erase error interrupt enable bit)

如果 OPERRIE2 位置“1”，对存储区 2 执行写入/擦除操作过程中检测到错误时，会产生中断。仅当 LOCK2 置“0”时，才能编程 OPERRIE2。

0：存储区 2 发生写入/擦除错误时不产生中断

1：存储区 2 发生写入/擦除错误时产生中断

位 21 **INCERRIE2**：存储区 2 不一致错误中断使能位 (Bank 2 inconsistency error interrupt enable bit)

如果 INCERRIE2 位置“1”，对存储区 2 执行写入操作过程中发生不一致错误时，会产生中断。仅当 LOCK2 置“0”时，才能编程 INCERRIE2。

0：存储区 2 发生不一致错误时不产生中断

1：存储区 2 发生不一致错误时产生中断

位 20 保留，必须保持复位值

位 19 **STRBERRIE2**：存储区 2 选通错误中断使能位 (Bank 2 strobe error interrupt enable bit)

如果 STRBERRIE2 位置“1”，对存储区 2 执行写入操作过程中发生选通错误时（主机多次向写缓冲区中编程同一字节），会产生中断。仅当 LOCK2 置“0”时，才能编程 STRBERRIE2。

0：存储区 2 发生选通错误时不产生中断

1：存储区 2 发生选通错误时产生中断

位 18 **PGSERRIE2**：存储区 2 编程顺序错误中断使能位 (Bank 2 programming sequence error interrupt enable bit)

如果 PGSERRIE2 位置“1”，对存储区 2 执行编程操作过程中发生顺序错误时，会产生中断。仅当 LOCK2 置“0”时，才能编程 PGSERRIE2。

0：存储区 2 发生顺序错误时不产生中断

1：存储区 2 发生顺序错误时产生中断

位 17 **WRPERRIE2**：存储区 2 写保护错误中断使能位 (Bank 2 write protection error interrupt enable bit)

如果 WRPERRIE2 位置“1”，对存储区 2 执行编程操作过程中发生保护错误时，会产生中断。仅当 LOCK2 置“0”时，才能编程 WRPERRIE2。

0：存储区 2 发生保护错误时不产生中断

1：存储区 2 发生保护错误时产生中断

位 16 **EOPIE2**：存储区 2 编程结束中断控制位 (Bank 2 end-of-program interrupt control bit)

将 EOPIE2 位置“1”可在对存储区 2 的编程操作结束时产生中断。仅当 LOCK2 置“0”时，才能编程 EOPIE2。

0：对存储区 2 执行的编程操作结束时不产生中断。

1：对存储区 2 执行的编程操作结束时使能中断。

位 15 **CRC_EN**：存储区 2 CRC 控制位 (Bank 2 CRC control bit)

将 CRC_EN 位置“1”可使能对存储区 2 进行 CRC 计算。CRC_EN 不会启动 CRC 计算，但会通过 FLASH_CRCCR2 寄存器使能 CRC 配置。

如果已在存储区 2 上执行 CRC 计算，则只能通过将 CRC_EN 位置“0”的方式禁止 CRC 计算。复位 CRC_EN 会复位 FLASH_CRCDATAR 寄存器的内容。

仅当 LOCK2 置“0”时，才能编程 CRC_EN。

位 14:11 保留，必须保持复位值

位 10:8 **SNB2**：存储区 2 扇区擦除选择编号 (Bank 2 sector erase selection number)

这些位用于选择扇区擦除操作的目标扇区。仅当 LOCK2 置“0”时，才能编程 SNB2。如果最高有效位置“1”，会使用第三位在 ICP 扇区与选项扇区之间进行选择。000 0000：存储区 2 的扇区 0

000 0001：存储区 2 的扇区 1

...

000 0111：存储区 2 的扇区 7

000 1000 到 011 1111：保留（不能设置此配置）

...

011 1111：保留（不能设置此配置）

100 0000：存储区 2 的 ICP 扇区

其他配置：保留

位 7 **START2**：存储区 2 的存储区或扇区擦除启动控制位 (Bank 2 bank or sector erase start control bit)

START2 位用于启动扇区擦除或存储区擦除操作。仅当 LOCK2 置“0”时，才能编程 START2。相应操作得到确认后，Flash 接口会复位 START2。操作得到确认之前，用户应用程序不能访问任何 Flash 寄存器。

位 6 **FW2**：存储区 2 写入强制控制位 (Bank 2 write forcing control bit)

即使写缓冲区未满，FW2 也会强制执行写操作。仅当 LOCK2 置“0”时，才能编程 FW2。相应操作得到确认后，Flash 接口会复位 FW2。操作得到确认之前，用户应用程序不能访问任何 Flash 寄存器。

仅当写缓冲区非空时，写入强制才有效（特别要说明的是，当连续执行写操作时，FW2 不会启动多次写操作）。

位 5:4 **PSIZE2**：存储区 2 程序大小 (Bank 2 program size)

PSIZE2 会选择对存储区 2 执行写操作和擦除操作过程中 Flash 使用的并行位数（更多详情，请参见*配置编程参数*一节）。仅当 LOCK2 置“0”时，才能编程 PSIZE2。

00：使用字节并行位数执行编程

01：使用半字并行位数执行编程

10：使用字并行位数执行编程

11：使用双字并行位数执行编程

位 3 **BER2**：存储区 2 擦除请求 (Bank 2 erase request)

将 BER2 位置“1”会请求对存储区 2 执行存储区擦除操作。仅当 LOCK2 置“0”时，才能编程 BER2。

BER2 的优先级要高于 SER2：如果两者均置 1，Flash 接口会执行存储区擦除操作（更多详细信息，请参见*标准 Flash 存储区擦除*一节）。

0：未请求对存储区 2 执行存储区擦除操作

1：请求对存储区 2 执行存储区擦除操作

位 2 **SER2**：存储区 2 扇区擦除请求 (Bank 2 sector erase request)

将 SER2 位置“1” 会请求对存储区 2 执行扇区擦除操作。仅当 LOCK2 置“0”时，才能编程 SER2。

BER2 的优先级要高于 SER2：如果两者均置 1，Flash 接口会执行存储区擦除操作（更多详细信息，请参见*Flash 扇区擦除*一节）。

0：未请求对存储区 2 执行扇区擦除操作

1：请求对存储区 2 执行扇区擦除操作

位 1 **PG2**：存储区 2 编程使能位 (Bank 2 program enable bit)

将 PG2 位置“1”可使能对存储区 2 执行写操作， 因此，即使正在进行扇区或存储区擦除操作，也可准备编程操作。

仅当 LOCK2 置“0”时，才能编程 PG2。PG2 复位后，会禁止将内部缓冲区用于对存储区 2 执行的写操作，缓冲区中存储的所有尚未编程的数据都会丢失。

位 0 **LOCK2**：存储区 2 配置锁定位 (Bank 2 configuration lock bit)

此位用于锁定 FLASH_CR2 寄存器。

FLASH_CR2 寄存器解锁后，LOCK2 位会自动复位（请参见*配置编程参数*一节）。如果执行顺序不正确，此位会保持锁定状态，直至下一次系统复位。

将 LOCK2 编程为“1”可将其置 1。LOCK1 置“1”后，必须执行新的解锁顺序将其解锁。

如果 LOCK2 从“0”变为“1”，FLASH_CR2 寄存器的其他位不会发生变化。

3.5.26 存储区 2 的 FLASH 状态寄存器 (FLASH_SR2)

FLASH status register for bank 2

偏移地址：0x110

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	CRCEND2	DBECCERR2	SNECCERR2	RDSERR2	RDPERR2	OPERR2	INCERR2	Res.	STRBERR2	PGSERR2	WRPERR2	EOP2
				r	r	r	r	r	r	r		r	r	r	r

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CRC_BUSY2	QW2	WBNE2	BSY2
												r	r	r	r

位 31:28 保留，必须保持复位值

位 27 **CRCEND2**：存储区 2 CRC 完成标志 (Bank 2 CRC-complete flag)

对存储区 2 完成 CRC 计算后，CRCEND2 位会置 1。如果 CRCENDIE2 置“1”，会产生中断。不需要复位 CRCEND2 便可重新开始进行 CRC 计算。向 FLASH_CCR2 寄存器中的 CLR_CRCEND2 位写入“1”可将 CRCEND2 清零。

0：对存储区 2 进行的 CRC 计算未完成

1：对存储区 2 进行的 CRC 计算已完成

位 26 **DBECCERR2**：存储区 2 ECC 双重检测错误标志 (Bank 2 ECC double detection error flag)

如果在对存储区 2 执行读操作的过程中发生 ECC 双重检测错误，DBECCERR2 标志会置 1。如果 DBECCERRIE2 置“1”，将产生中断。向 FLASH_CCR2 寄存器中的 CLR_DBECCERR2 位写入“1”可将 DBECCERR2 清零。

0：存储区 2 未发生 ECC 双重检测错误

1：存储区 2 发生 ECC 双重检测错误

位 25 **SNECCERR2**：存储区 2 单校正错误标志 (Bank 2 single correction error flag)

如果在对存储区 2 执行读操作的过程中发生 ECC 单校正错误，SNECCERR2 标志会置 1。如果 SNECCERRIE2 置“1”，将产生中断。向 FLASH_CCR2 寄存器中的 CLR_SNECCERR2 位写入“1”可将 SNECCERR2 清零。

0：存储区 2 未发生 ECC 单校正错误

1：存储区 2 发生 ECC 单校正错误

位 24 **RDSERR2**：存储区 2 安全错误标志 (Bank 2 secure error flag)

如果存储区 2 发生安全错误（未获得适当权限便访问安全保护字），RDSERR2 标志会置 1。如果 RDSERRIE2 置“1”，将产生中断。向 FLASH_CCR2 寄存器中的 CLR_RDSERR2 位写入“1”可将 RDSERR2 清零。

0：存储区 2 未发生安全错误

1：存储区 2 发生安全错误

位 23 **RDPERR2**：存储区 2 读保护错误标志 (Bank 2 read protection error flag)

如果存储区 2 发生读保护错误（访问受 PCROP 保护的字），RDPERR2 标志会置 1。如果 RDPERRIE2 置“1”，将产生中断。向 FLASH_CCR2 寄存器中的 CLR_RDPERR2 位写入“1”可将 RDPERR2 清零。

0：存储区 2 未发生读保护错误

1：存储区 2 发生读保护错误

位 22 **OPERR2**：存储区 2 写入/擦除错误标志 (Bank 2 write/erase error flag)

如果在对存储区 2 执行写入/擦除操作的过程中出错，OPERR2 标志会置 1。如果 OPERRIE2 置“1”，将产生中断。向 FLASH_CCR2 寄存器中的 CLR_OPERR2 位写入“1”可将 OPERR2 清零。

0：存储区 2 未发生写入/擦除错误

1：存储区 2 发生写入/擦除错误

位 21 **INCERR2**：存储区 2 不一致错误标志 (Bank 2 inconsistency error flag)

如果存储区 2 发生不一致错误，INCERR2 标志会置 1。如果 INCERRIE2 置“1”，将产生中断。向 FLASH_CCR2 寄存器中的 CLR_INCERR2 位写入“1”可将 INCERR2 清零。

0：存储区 2 未发生不一致错误

1：存储区 2 发生不一致错误。

位 20 保留，必须保持复位值

位 19 **STRBERR2**：存储区 2 选通错误标志 (Bank 2 strobe error flag)

如果存储区 2 发生选通错误（主机尝试多次将同一字节写入写缓冲区中），STRBERR2 标志会置 1。如果 STRBERRIE2 位置“1”，将产生中断。向 FLASH_CCR2 寄存器中的 CLR_STRBERR2 位写入“1”可将 STRBERR2 清零。

0：存储区 2 未发生选通错误

1：存储区 2 发生选通错误。

位 18 **PGSERR2**：存储区 2 编程顺序错误标志 (Bank 2 programming sequence error flag)

如果存储区 2 发生顺序错误，PGSERR2 标志会置 1。如果 PGSERRIE2 位置“1”，将产生中断。向 FLASH_CCR2 寄存器中的 CLR_PGSERR2 位写入“1”可将 PGSERR2 清零。

0：存储区 2 未发生顺序错误

1：存储区 2 发生顺序错误。

位 17 **WRPERR2**：存储区 2 写保护错误标志 (Bank 2 write protection error flag)

如果在对存储区 2 执行编程操作的过程中发生保护错误，WRPERR2 标志会置 1。如果 EOPIE2 置“1”，也会产生中断。向 FLASH_CCR2 寄存器中的 CLR_EOP2 位写入“1”可将 WRPERR2 清零。

0：存储区 2 未发生保护错误

1：存储区 2 发生保护错误

位 16 **EOP2**：存储区 2 编程结束标志 (Bank 2 end-of-program flag)

对存储区 2 进行的编程操作结束时，EOP2 标志会置 1。如果 EOPIE2 置“1”，将产生中断。不需要先复位 EOP2 便可开始执行新操作。向 FLASH_CCR2 寄存器中的 CLR_EOP2 位写入“1”可将 EOP2 位清零。

0：存储区 2 上的编程操作未结束

1：存储区 2 上的编程操作已结束

位 15:4 保留，必须保持复位值

位 3 **CRC_BUSY2**：存储区 2 CRC 忙标志 (Bank 2 CRC busy flag)

正在对存储区 2 进行 CRC 计算时，CRC_BUSY2 标志会置 1。此位不能强制设为“0”。用户必须等待存储区 2 的 CRC 计算完成或禁止对存储区 2 进行 CRC 计算。

0：未在对存储区 2 执行 CRC 计算

1：正在对存储区 2 执行 CRC 计算

位 2 **QW2**：存储区 2 等待队列标志 (Bank 2 wait queue flag)

对存储区 2 进行的编程操作位于等待队列中时，QW2 标志会置 1。无法确定队列中是哪种类型的编程操作。所有编程操作均已执行完毕并从等待队列中删除时，此标志会由硬件复位。此位不能强制设为“0”。如果未请求执行其他操作，此位会在确定时间后复位。

0：没有编程操作在存储区 2 的操作队列中等待

1：至少有一个编程操作在存储区 2 的操作队列中等待

位 1 **WBNE2**：存储区 2 写缓冲区非空标志 (Bank 2 write buffer not empty flag)

如果存储区 2 写缓冲区不为空，并且 Flash 接口正在等待新数据填入，WBNE2 标志会置 1。每次写缓冲区清空时，WBNE2 会由硬件复位。只要发生以下任一事件，就会进行清空：

- 写缓冲区已满
- 用户强制执行写操作
- 发生了涉及到数据丢失的错误
- 写操作禁止

此位不能强制设为“0”。要复位此位，应执行上述任一操作来清空写缓冲区。

0：存储区 2 的写缓冲区为空

1：存储区 2 的写缓冲区正在等待数据填入

位 0 **BSY2**：存储区 2 正在进行编程标志 (Bank 2 ongoing program flag)

正在对存储区 2 进行编程操作时，BSY2 标志会置 1。无法确定正在进行哪种类型的编程操作。BSY2 不能强制设为“0”。操作完成后，如果未开始执行其他编程操作，则会由硬件复位此位。

0：未在对存储区 2 执行编程操作

1：正在对存储区 2 执行编程操作

3.5.27 存储区 2 的 FLASH 清零控制寄存器 (FLASH_CCR2)

FLASH clear control register for bank 2

偏移地址：0x114

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	CLR_CRCEND2	CLR_DBECCERR2	CLR_SNECCERR2	CLR_RDSERR2	CLR_RDPERR2	CLR_OPERR2	CLR_INCERR2	Res.	CLR_STRBERR2	CLR_PGSERR2	CLR_WRPERR2	CLR_EOP2
				w	w	w	w	w	w	w		w	w	w	w

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

位 31:28 保留，必须保持复位值

位 27 **CLR_CRCEND2**：存储区 2 CRCEND2 标志清零位 (Bank 2 CRCEND2 flag clear bit)
将此位置“1”可将 FLASH_SR2 寄存器的 CRCEND2 标志复位为“0”。

位 26 **CLR_DBECCERR2**：存储区 2 DBECCERR2 标志清零位 (Bank 2 DBECCERR2 flag clear bit)
将此位置“1”可将 FLASH_SR2 寄存器的 DBECCERR2 标志复位为“0”。如果 FLASH_SR2 寄存器的 SNECCERR2 标志置为“0”，FLASH_ECC_FA2R 寄存器也会复位为“0”。

位 25 **CLR_SNECCERR2**：存储区 2 SNECCERR2 标志清零位 (Bank 2 SNECCERR2 flag clear bit)
将此位置“1”可将 FLASH_SR2 寄存器的 SNECCERR2 标志复位为“0”。如果 FLASH_SR2 寄存器的 DBECCERR2 标志置为“0”，FLASH_ECC_FA2R 寄存器也会复位为“0”。

位 24 **CLR_RDSERR2**：存储区 2 RDSERR2 标志清零位 (Bank 2 RDSERR2 flag clear bit)
将此位置“1”可将 FLASH_SR2 寄存器的 RDSERR2 标志复位为“0”。

位 23 **CLR_RDPERR2**：存储区 2 RDPERR2 标志清零位 (Bank 2 RDPERR2 flag clear bit)
将此位置“1”可将 FLASH_SR2 寄存器的 RDPERR2 标志复位为“0”。

位 22 **CLR_OPERR2**：存储区 2 OPERR2 标志清零位 (Bank 2 OPERR2 flag clear bit)
将此位置“1”可将 FLASH_SR2 寄存器的 **OPERR**2 标志复位为“0”。

位 21 **CLR_INCERR2**：存储区 2 INCERR2 标志清零位 (Bank 2 INCERR2 flag clear bit)
将此位置“1”可将 FLASH_SR2 寄存器的 INCERR2 标志复位为“0”。

位 20 保留，必须保持复位值

位 19 **CLR_STRBERR2**：存储区 2 STRBERR2 标志清零位 (Bank 2 STRBERR2 flag clear bit)
将此位置“1”可将 FLASH_SR2 寄存器的 STRBERR2 标志复位为“0”。

位 18 **CLR_PGSERR2**：存储区 2 PGSERR2 标志清零位 (Bank 2 PGSERR2 flag clear bit)
将此位置“1”可将 FLASH_SR2 寄存器的 PGSERR2 标志复位为“0”。

位 17 **CLR_WRPERR2**：存储区 2 WRPERR2 标志清零位 (Bank 2 WRPERR2 flag clear bit)
将此位置“1”可将 FLASH_SR2 寄存器的 WRPERR2 标志复位为“0”。

位 16 **CLR_EOP2**：存储区 2 EOP2 标志清零位 (Bank 2 EOP2 flag clear bit)
将此位置“1”可将 FLASH_SR2 寄存器的 EOP2 标志复位为“0”。

位 15:0 保留，必须保持复位值

3.5.28 存储区 2 的 FLASH 保护地址（当前值）(FLASH_PRAR_CUR2)

FLASH protection address for bank 2

偏移地址：0x128

复位值：0xXXXX XXXX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DMEP2	Res.	Res.	Res.	PROT_AREA_END2											
r				r	r	r	r	r	r	r	r	r	r	r	r

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	PROT_AREA_START2											
				r	r	r	r	r	r	r	r	r	r	r	r

位 31 **DMEP2**：存储区 2 PCROP 保护擦除使能选项状态位 (Bank 2 PCROP protected erase enable option status bit)
如果 DMEP2 置“1”，发生保护降级（从级别 1 变为级别 0）时，会擦除 PCROP 保护区（请参见 *PCROP 区域（专有代码读保护，仅执行区域）*一节和 *标准 Flash 存储区擦除*一节）。

位 30:28 保留，必须保持复位值

位 27:16 **PROT_AREA_END2**：存储区 2 最高 PCROP 保护地址 (Bank 2 highest PCROP protected address)
这些位包含存储区 2 中的最后一个 PCROP 保护地址。
如果此地址等于 PROT_AREA_START2，整个存储区 2 会受到 PCROP 保护。
如果此地址小于 PROT_AREA_START2，则不会对存储区 2 设置任何保护功能。

位 15:12 保留，必须保持复位值

位 11:0 **PROT_AREA_START2**：存储区 2 最低 PCROP 保护地址 (Bank 2 lowest PCROP protected address)
这些位包含存储区 2 中的第一个 PCROP 保护地址。
如果此地址等于 PROT_AREA_END2，整个存储区 2 会受到 PCROP 保护。
如果此地址大于 PROT_AREA_END2，则不会对存储区 1 设置任何保护功能。

3.5.29 存储区 2 的 FLASH 保护地址（要编程的值）(FLASH_PRAR_PRG2)

FLASH protection address for bank 2

偏移地址：0x12C

复位值：0xXXXX XXXX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DMEP2	Res.	Res.	Res.	PROT_AREA_END2											
rw				rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	PROT_AREA_START2											
				rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31 **DMEP2**：存储区 2 PCROP 保护擦除使能选项配置位 (Bank 2 PCROP protected erase enable option configuration bit)

如果 DMEP2 置“1”，发生保护降级（从级别 1 变为级别 0）或存储区 2 擦除时，会擦除 PCROP 保护区（请参见 *PCROP 区域（专有代码读保护，仅执行区域）*一节）。

位 30:28 保留，必须保持复位值

位 27:16 **PROT_AREA_END2**：存储区 2 最高 PCROP 保护地址配置 (Bank 2 highest PCROP protected address configuration)

这些位允许配置存储区 2 中的最后一个 PCROP 保护地址。

位 15:12 保留，必须保持复位值

位 11:0 **PROT_AREA_START2**：存储区 2 最低 PCROP 保护地址配置 (Bank 2 lowest PCROP protected address configuration)

这些位允许配置存储区 2 中的第一个 PCROP 保护地址。

3.5.30 存储区 2 的 FLASH 安全地址（当前值）(FLASH_SCAR_CUR2)

FLASH secure address for bank 2

偏移地址：0x130

复位值：0xXXXX XXXX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DMES2	Res.	Res.	Res.	SEC_AREA_END2											
r				r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	SEC_AREA_START2											
				r	r	r	r	r	r	r	r	r	r	r	r

位 31 **DMES2**：存储区 2 安全保护擦除使能选项状态位 (Bank 2 secure protected erase enable option status bit)

如果 DMES2 置“1”，发生保护降级（从级别 1 变为级别 0）时，会擦除安全保护区（请参见*安全区域*一节和 *标准 Flash 存储区擦除*一节）。

位 30:28 保留，必须保持复位值

位 27:16 **SEC_AREA_END2**：存储区 2 最高安全保护地址 (Bank 2 highest secure protected address)

这些位包含存储区 2 中的最后一个安全保护地址。
如果此地址等于 SEC_AREA_START2，整个存储区 2 会受到安全保护。
如果此地址小于 SEC_AREA_START2，则不会对存储区 2 设置任何保护功能。

位 15:12 保留，必须保持复位值

位 11:0 **SEC_AREA_START2**：存储区 2 最低安全保护地址 (Bank 2 lowest secure protected address)

这些位包含存储区 2 中的第一个安全保护地址。
如果此地址等于 SEC_AREA_END2，整个存储区 2 会受到安全保护。
如果此地址大于 SEC_AREA_END2，则不会对存储区 2 设置任何保护功能。

3.5.31 存储区 2 的 FLASH 安全地址（要编程的值）(FLASH_SCAR_PRG2)

FLASH secure address for bank 2

偏移地址：0x134

复位值：0xXXXX XXXX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DMES2	Res.	Res.	Res.	SEC_AREA_END2											
rw				rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	SEC_AREA_START2											
				rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31 **DMES2**：存储区 2 安全保护擦除使能选项配置位 (Bank 2 secure protected erase enable option configuration bit)

如果 DMES2 置“1”，发生保护降级（从级别 1 变为级别 0）或存储区 2 擦除时，会擦除安全保护区（请参见*安全区域*一节），

位 30:28 保留，必须保持复位值

位 27:16 **SEC_AREA_END2**：存储区 2 最高安全保护地址配置 (Bank 2 highest secure protected address configuration)

这些位允许配置存储区 2 中的最后一个安全保护地址。仅可在安全模式下编程这些位。

位 15:12 保留，必须保持复位值

位 11:0 **SEC_AREA_START2**：存储区 2 最低安全保护地址配置 (Bank 2 lowest secure protected address configuration)

这些位允许配置存储区 2 中的第一个安全保护地址。仅可在安全模式下编程这些位。

3.5.32 存储区 2 的 FLASH 扇区写保护（当前值）(FLASH_WPSN_CUR2R)

FLASH write sector protection for bank 2

偏移地址：0x138

复位值：0x0000 00XX

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WRPSn2							
								r	r	r	r	r	r	r	r

位 31:8 保留，必须保持复位值

位 7:0 **WRPSn2**：存储区 2 扇区写保护状态字节 (Bank 2 sector write protection status byte)
每个 FLASH_WPSN_CUR2R 位用于反映存储区 2 相应扇区的写保护状态。

3.5.33 存储区 2 的 FLASH 扇区写保护（要编程的值）(FLASH_WPSN_PRG2R)

FLASH write sector protection for bank 2

偏移地址：0x13C

复位值：0x0000 00XX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WRPSn2							
								rw	rw	rw	rw	rw	rw	rw	rw

位 31:8 保留，必须保持复位值

位 7:0 **WRPSn2**：存储区 2 扇区写保护配置字节 (Bank 2 sector write protection configuration byte)
将 WRPSn2 位置“0”可对存储区 2 的相应扇区进行写保护。

3.5.34 存储区 2 的 FLASH CRC 控制寄存器 (FLASH_CRCCR2)

FLASH CRC control register for bank 2

地址偏移：0x150

复位值：0x001C 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CRC_BURST		Res.	Res.	CLEAN_CRC	START_CRC
										rw	rw			w	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	CLEAN_SECT	ADD_SECT	CRC_BY_SECT	ALL_BANK	Res.	Res.	Res.	Res.	CRC_SECT		
					w	w	rw	w					rw	rw	rw

位 31:22 保留，必须保持复位值

位 21:20 **CRC_BURST**：存储区 2 CRC 突发大小 (Bank 2 CRC burst size)

CRC_BURST 位用于设置 CRC 计算单元生成的突发大小。

00：每个突发的大小为 4 个字

01：每个突发的大小为 16 个字

10：每个突发的大小为 64 个字

11：每个突发的大小为 256 个字

位 19:18 保留，必须保持复位值

位 17 **CLEAN_CRC**：存储区 2 CRC 清空位 (Bank 2 CRC clear bit)

将 CLEAN_CRC 置“1”会将 FLASH_CRCDATAR 寄存器中存储的当前 CRC 结果清空。

位 16 **START_CRC**：存储区 2 CRC 起始位 (Bank 2 CRC start bit)

START_CRC 位会触发使用当前配置对存储区 2 进行 CRC 计算。如果正在进行选项字节更改操作，则不能开始 CRC 计算，这是因为选项字节更改操作完成之前，对 Flash 寄存器执行的所有写访问都会挂起。

位 15:13 保留，必须保持复位值

位 12:11 保留，必须保持复位值

位 10 **CLEAN_SECT**：存储区 2 CRC 扇区列表清空位 (Bank 2 CRC sector list clear bit)

将 CLEAN_SECT 置“1”会清空进行 CRC 计算的扇区列表。

位 9 **ADD_SECT**：存储区 2 CRC 扇区选择位 (Bank 2 CRC sector select bit)

将 ADD_SECT 置“1”会将编号为 CRC_SECT 的扇区添加到进行 CRC 计算的扇区列表中。

位 8 **CRC_BY_SECT**：存储区 2 CRC 扇区模式选择位 (Bank 2 CRC sector mode select bit)

如果 CRC_BY_SECT 置“1”，则会在扇区级对由 CRC_SECT 选择的扇区执行 CRC 计算。

如果 CRC_BY_SECT 复位为“0”，则会对 CRC_START_ADDR 和 CRC_END_ADDR 之间的所有地址执行 CRC 计算。

位 7 **ALL_BANK**：存储区 2 CRC 选择位 (Bank 2 CRC select bit)

如果 ALL_BANK 置“1”，存储区 2 的所有用户扇区都会添加到执行 CRC 计算的扇区列表中。

位 6:3 保留，必须保持复位值

位 2:0 **CRC_SECT**：存储区 2 CRC 扇区编号 (Bank 2 CRC sector number)

CRC_SECT 用于选择一个或多个要添加到 CRC 计算的扇区。可在两个地址（起始地址和结束地址）之间执行 CRC 计算，也可以对扇区列表执行 CRC 计算。如果选择第二种方法，可在 CRC_SECT 中编程扇区编号，然后将 ADD_SECT 置“1”，以此将扇区添加到扇区列表中。

要擦除扇区列表，可将 CLEAN_SECT 位置 1，也可以禁止 CRC 计算。有关各扇区编号的信息，请参见对 FLASH_CR 寄存器中的 SNB2 位的说明（请参见*第 3.5.25 节：存储区 2 的 FLASH 控制寄存器 (FLASH_CR2)*）。

仅当 FLASH_CR 寄存器的 CRC_EN 置“1”时，才能设置 CRC_SECT。

3.5.35 存储区 2 的 FLASH CRC 起始地址寄存器 (FLASH_CRCSADD2R)

FLASH CRC start address register for bank 2

偏移地址：0x154

复位值：0x0000 0000

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

位 31:0 **CRC_START_ADDR**：存储区 2 上的 CRC 起始地址 (CRC start address on bank 2)
当 CRC_BY_SECT 为“0”时，会使用 CRC_START_ADDR， 必须将它编程为执行 CRC 计算的存储区 2 的起始地址。

3.5.36 存储区 2 的 FLASH CRC 结束地址寄存器 (FLASH_CRCEADD2R)

FLASH CRC end address register for bank 2

偏移地址：0x158

复位值：0x0000 0000

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

位 31:0 **CRC_END_ADDR**：存储区 2 上的 CRC 结束地址 (CRC end address on bank 2)
当 CRC_BY_SECT 为“0”时，会使用 CRC_END_ADDR， 必须将它编程为执行 CRC 计算的存储区 2 的结束地址。

3.5.37 存储区 2 的 FLASH ECC 失效地址 (FLASH_ECC_FA2R)

FLASH ECC fail address for bank 2

偏移地址：0x160

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	FAIL_ECC_ADDR2														
	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:15 保留，必须保持复位值

位 14:0 **FAIL_ECC_ADDR2**：存储区 2 ECC 错误地址 (Bank 2 ECC error address)

如果在对存储区 2 执行读操作的过程中出现 ECC 错误（单错误校正错误或双重检测错误），FAIL_ECC_ADDR2 位域会包含生成此错误的地址。

如果 FLASH_SR2 寄存器（CLR_SNECCERR2 或 CLR_DBECCERR2）中的错误标志复位，FAIL_ECC_ADDR2 也会复位。

仅当没有 ECC 错误标志置 1 时，Flash 接口才会在该寄存器中编程地址。这意味着仅会保存第一个生成 ECC 错误的地址。

3.6 Flash 寄存器映射与复位值

表 16. FLASH 寄存器映射和复位值

偏移	寄存器名称和复位值	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x000	FLASH_ACR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WRHIGHFREQ [1:0]		Res.	LATENCY [3:0]		
	0x00000037																											1	1		1	1	1
0x004	FLASH_KEYR1	KEYR1																															
	0x00000000	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x008	FLASH_ OPTKEYR	OPTKEYR																															
	0x00000000	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x00C	FLASH_CR1	Res.	Res.	Res.	Res.	CRCENDIE1	DBECCERRIE1	SNECCERRIE1	RDSERRIE1	RDPERRIE1	OPERRIE1	INCERRIE1	Res.	STRBERRIE1	PGSERRIE1	WRPERRIE1	EOPIE1	CRC_EN	Res.	Res.	Res.	Res.	SNB1 [2:0]			START1	FW1	PSIZE1 [1:0]		BER1	SER1	PG1	LOCK1
	0x00000031					0	0	0	0	0	0	0		0	0	0	0	0					0	0	0	0	0	1	1	0	0	0	1
0x010	FLASH_SR1	Res.	Res.	Res.	Res.	CRCEND1	DBECCERR1	SNECCERR1	RDSERR1	RDPERR1	OPERR1	INCERR1	Res.	STRBERR1	PGSERR1	WRPERR1	EOP1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CRC_BUSY1	QW1	WBNE1	BSY1
	0x00000000					0	0	0	0	0	0	0		0	0	0	0													0	0	0	0
0x014	FLASH_CCR1	Res.	Res.	Res.	Res.	CLR_CRCEND1	CLR_DBECCERR1	CLR_SNECCERR1	CLR_RDSERR1	CLR_RDPERR1	CLR_OPERR1	CLR_INCERR1	Res.	CLR_STRBERR1	CLR_PGSERR1	CLR_WRPERR1	CLR_EOP1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	0x00000000					0	0	0	0	0	0	0		0	0	0	0																
0x018	FLASH_OPTCR	SWAP_BANK	OPTCHANGEERRIE	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MER	Res.	Res.	OPTSTART	OPTLOCK
	(0x00000001)	0	0																								1			0	1

表 16. FLASH 寄存器映射和复位值（续）

偏移	寄存器名称和复位值	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x01C	FLASH_OPTSR_CUR	SWAP_BANK_OPT	OPTCHANGEERR	IO_HSLV	PERSO_OK	RSS2.	RSS1.	Res.	Res.	Res.	Res.	保密性	ST_RAM_SIZE [1:0]		FZ_IWDG_SDBY	FZ_IWDG_STOP	Res.	RDP[7:0]								nRST_STBY_D1	nRST_STOP_D1	Res.	IWDG1_HW	BOR_LEV[1:0]		Res.	OPT_BUSY
	0xXXXX XXXX	X	X	X	X	X	X					X	X	X	X	X		X	X	X	X	X	X	X	X	X	X		X	X	X		X
0x020	FLASH_OPTSR_PRG	SWAP_BANK_OPT	Res.	IO_HSLV	Res.	RSS2.	RSS1.	Res.	Res.	Res.	Res.	保密性	ST_RAM_SIZE [1:0]		FZ_IWDG_SDBY	FZ_IWDG_STOP	Res.	RDP[7:0]								nRST_STBY_D1	nRST_STOP_D1	Res.	IWDG1_HW	BOR_LEV [1:0]		Res.	Res.
	0xXXXX XXXX	X		X		X	X					X	X	X	X	X		X	X	X	X	X	X	X	X	X	X			X	X		
0x024	FLASH_OPTCCR	Res.	CLR_OPTCHANGEERR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	0x00000000		0																														
0x028	FLASH_PRAR_CUR1	DMEP1	Res.	Res.	Res.	PROT_AREA_END1[11:0]												Res.	Res.	Res.	Res.	PROT_AREA_START1[11:0]												
	0xXXXX XXXX	X				X	X	X	X	X	X	X	X	X	X	X	X					X	X	X	X	X	X	X	X	X	X	X	X
0x02C	FLASH_PRAR_PRG1	DMEP1	Res.	Res.	Res.	PROT_AREA_END1[11:0]												Res.	Res.	Res.	Res.	PROT_AREA_START1[11:0]												
	0xXXXX XXXX	X				X	X	X	X	X	X	X	X	X	X	X	X					X	X	X	X	X	X	X	X	X	X	X	X
0x030	FLASH_SCAR_CUR1	DMES1	Res.	Res.	Res.	SEC_AREA_END1[11:0]												Res.	Res.	Res.	Res.	SEC_AREA_START1[11:0]												
	0xXXXX XXXX	X				X	X	X	X	X	X	X	X	X	X	X	X					X	X	X	X	X	X	X	X	X	X	X	X
0x034	FLASH_SCAR_PRG1	DMES1	Res.	Res.	Res.	SEC_AREA_END1[11:0]												Res.	Res.	Res.	Res.	SEC_AREA_START1[11:0]												
	0xXXXX XXXX	X				X	X	X	X	X	X	X	X	X	X	X	X					X	X	X	X	X	X	X	X	X	X	X	X
0x038	FLASH_WPSN_CUR1R	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WRPSn1[7:0]							
	0x000000FF																									X	X	X	X	X	X	X	X
0x03C	FLASH_WPSN_PRG1R	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WRPSn1[7:0]							
	0x000000FF																									X	X	X	X	X	X	X	X

表 16. FLASH 寄存器映射和复位值（续）

偏移	寄存器名称和复位值	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x040	FLASH_BOOT_CURR	BOOT_ADD1[15:0]																BOOT_ADD0[15:0]															
	0xXXXX XXXX	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
0x044	FLASH_BOOT_PRGR	BOOT_ADD1[15:0]																BOOT_ADD0[15:0]															
	0xXXXX XXXX	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
0x050	FLASH_CRCCR1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CRC_BURST[1:0]		Res.	Res.	CLEAN_CRC	START_CRC	Res.	Res.	Res.	Res.	Res.	CLEAN_SECT	ADD_SECT	CRC_BY_SECT	ALL_BANK	Res.	Res.	Res.	Res.	CRC_SECT[2:0]		
	0x001C0000											0	1			0	0						0	0	0	0					0	0	0
0x054	FLASH_CRCSADD1R	CRC_START_ADDR[31:0]																															
	0x00000000	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x058	FLASH_CRCEADD1R	CRC_END_ADDR[31:0]																															
	0x00000000	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x05C	FLASH_CRCDATAR	CRC_DATA[31:0]																															
	0x00000000	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x060	FLASH_ECC_FA1R	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FAIL_ECC_ADDR1[14:0]														
	0x00000000																		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x100	FLASH_ACR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WRHIGHFREQ[1:0]		LATENCY[3:0]			
	0x00000037																											1	1	0	1	1	1
0x104	FLASH_KEYR2	KEYR2																															
	0x00000000	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x108	FLASH_OPTKEYR	OPTKEYR																															
	0x00000000	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x10C	FLASH_CR2	Res.	Res.	Res.	Res.	CRCENDIE2	DBECCERRIE2	SNECCERRIE2	RDSERRIE2	RDPERRIE2	OPERRIE2	INCERRIE2	Res.	STRBERRIE2	PGSERRIE2	WRPERRIE2	EOPIE2	CRC_EN	Res.	Res.	Res.	Res.	SNB2[2:0]			START2	FW2	PSIZE2[1:0]		BER2	SER2	PG2	LOCK2
	0x00000031					0	0	0	0	0	0	0		0	0	0	0	0					0	0	0	0	0	1	1	0	0	0	1

表 16. FLASH 寄存器映射和复位值（续）

偏移	寄存器名称和复位值	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x110	FLASH_SR2	Res.	Res.	Res.	Res.	CRCEND2	DBECCERR2	SNECCERR2	RDSERR2	RDPERR2	OPERR2	INCERR2	Res.	STRBERR2	PGSERR2	WRPERR2	EOP2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CRC_BUSY2	QW2	WBNE2	BSY2
	0x00000000					0	0	0	0	0	0	0		0	0	0	0													0	0	0	0
0x114	FLASH_CCR2	Res.	Res.	Res.	Res.	CLR_CRCEND2	CLR_DBECCERR2	CLR_SNECCERR2	CLR_RDSERR2	CLR_RDPERR2	CLR_OPERR2	CLR_INCERR2	Res.	CLR_STRBERR2	CLR_PGSERR2	CLR_WRPERR2	CLR_EOP2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	0x00000000					0	0	0	0	0	0	0		0	0	0	0																
0x118	FLASH_OPTCR	SWAP_BANK	OPTCHANGEERRIE	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OBL_LAUNCH	OPTSTART	OPTLOCK	
	0x00000001	0	0																											0	0	1	
0x11C	FLASH_OPTSR_CUR	SWAP_BANK_OPT	OPTCHANGEERR	IO_HSLV	PERSO_OK	RSS2	RSS1	Res.	Res.	Res.	Res.	保密性	ST_RAM_SIZE [1:0]		FZ_IWDG_SDBY	FZ_IWDG_STOP	Res.	RDP[7:0]								nRST_STBY_D1	nRST_STOP_D1	Res.	IWDG1_HW	BOR_LEV [1:0]		Res.	OPT_BUSY	
	0x13C600F0	0	0		1	0	0					0	0	0	1	1		0	0	0	0	0	0	0	0	1	1		1	0	0		0
0x120	FLASH_OPTSR_PRG	SWAP_BANK_OPT	Res.	IO_HSLV	Res.	RSS2	RSS1	nRST_STBY_D2	nRST_STOP_D2	Res.	Res.	保密性	ST_RAM_SIZE [1:0]		FZ_IWDG_SDBY	FZ_IWDG_STOP	Res.	RDP[7:0]								nRST_STBY_D1	nRST_STOP_D1	Res.	IWDG1_HW	BOR_LEV [1:0]		Res.	Res.	
	0x13C600F0	0				0	0	1	1			0	0	0	1	1		0	0	0	0	0	0	0	0	1	1		1	0	0		
0x124	FLASH_OPTCCR	Res.	CLR_OPTCHANGEERR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	0x00000000		0																														
0x128	FLASH_PRAR_CUR2	DMEP2	Res.	Res.	Res.	PROT_AREA_END2[11:0]												Res.	Res.	Res.	Res.	PROT_AREA_START2[11:0]												
	0x80000000	X				X	X	X	X	X	X	X	X	X	X	X	X					X	X	X	X	X	X	X	X	X	X	X	X

表 16. FLASH 寄存器映射和复位值（续）

偏移	寄存器名称和复位值	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x12C	FLASH_PRAR_PRG2	DMEP2	Res.	Res.	Res.	PROT_AREA_END2[11:0]												Res.	Res.	Res.	Res.	PROT_AREA_START2[11:0]												
	0x80000000	X				X	X	X	X	X	X	X	X	X	X	X	X					X	X	X	X	X	X	X	X	X	X	X	X
0x130	FLASH_SCAR_CUR2	DMES2	Res.	Res.	Res.	SEC_AREA_END2[11:0]												Res.	Res.	Res.	Res.	SEC_AREA_START2[11:0]												
	0x80000000	X				X	X	X	X	X	X	X	X	X	X	X	X					X	X	X	X	X	X	X	X	X	X	X	X
0x134	FLASH_SCAR_PRG2	DMES2	Res.	Res.	Res.	SEC_AREA_END2[11:0]												Res.	Res.	Res.	Res.	SEC_AREA_START2[11:0]												
	0x80000000	X				X	X	X	X	X	X	X	X	X	X	X	X					X	X	X	X	X	X	X	X	X	X	X	X
0x138	FLASH_WPSN_CUR2R	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WRPSn2[7:0]							
	0x000000FF																									X	X	X	X	X	X	X	X
0x13C	FLASH_WPSN_PRG2R	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WRPSn2[7:0]							
	0x000000FF																									X	X	X	X	X	X	X	X
0x150	FLASH_CRCCR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CRC_BURST[1:0]		Res.	Res.	CLEAN_CRC	START_CRC	Res.	Res.	Res.	Res.	Res.	CLEAN_SECT	ADD_SECT	CRC_BY_SECT	ALL_BANK	Res.	Res.	Res.	Res.	CRC_SECT[2:0]		
	0x001C0000											0	x			0	0						0	0	0	0					0	0	0
0x154	FLASH_CRCSADD2R	CRC_START_ADDR[31:0]																															
	0x00000000	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x158	FLASH_CRCEADD2R	CRC_END_ADDR[31:0]																															
	0x00000000	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x15C	FLASH_CRCDATAR	CRC_DATA[31:0]																															
	0x00000000	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x160	FLASH_ECC_FA2R	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FAIL_ECC_ADDR2[14:0]														
	0x00000000																		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

4 安全存储管理

4.1 前言

STM32H7x3 微控制器提供了第一套保护机制，与最新的 STM32 系列（STM32F4/F7 和 STM32L0/L4）类似：

- 全局读取设备保护
- 写保护
- 专有代码读保护（PCROP）

有关这些保护机制的汇总，请参见*第 4.3 节：Flash 保护*。

STM32H7x3 还提供了全新的增强保护模式，即安全访问模式，可实现用户自定义安全服务（例如安全固件升级）开发、确保安全执行以及代码和数据保护。*第 4.4 节：安全访问模式*、*第 4.5 节：根安全服务 (RSS)* 和*第 4.6 节：安全用户软件*中详细介绍了这种新机制。

安全存储管理单元包含在 D1 域中。

4.2 词汇表

本文将使用以下术语：

表 17. 首选术语列表

术语	说明
器件安全等级	
标准模式	器件状态，可访问用户 Flash、选项字节和自举程序区
安全访问模式	器件状态，可访问该器件的所有存储区
存储区	
系统存储器	ST 保留存储区用于存储 ST ROM 代码
用户 Flash	Flash 区域用于存储用户代码和数据
安全用户存储区	可将此区域配置为在复位后访问一次，并在执行此区域中存储的代码后针对用户 Flash 中存储的固件隐藏
SW 服务	
自举程序	意法半导体软件在复位后执行，可从常规通信端口下载固件
根安全服务 (RSS)	意法半导体软件支持加密固件/模块解密，可安装到安全和非安全用户存储器中
安全自举程序	补充服务包括意法半导体自举程序和 ST 根安全服务
安全用户软件	用户软件在复位后运行一次，可用于存储 SFU 和固件初始化。安全用户 SW 位于安全用户存储区
安全固件更新 (SFU)	用户软件用于从用户 Flash 下载和更新已安装固件

4.3 Flash 保护

有三种保护机制可用于终端用户应用或在涉及专有代码（例如，第三方库）的开发期间使用：

- 读取器件保护 (RDP)

  这是一种全局保护机制，适用于终端产品。防止通过调试端口或者从 RAM 或自举程序自举来从外部访问器件。
- 专有代码读保护（PCROP）

  可在指定的 Flash 区域（必须为仅执行）中设置此机制。PCROP 用于避免敏感软件转储。
- 写保护

  可在 Flash 扇区级设置此保护。这可防止代码或数据被意外擦除。

表 18 汇总了 STM32H7x3 以及大多数 STM32 器件提供的保护机制。在 STM32H7x3 上，这些保护机制在标准和安全访问模式下均可用。

更多详细信息，请参见*第 3 节：嵌入式 Flash (FLASH)*。

表 18. Flash 保护机制

Flash 保护	范围	说明
RDP	全球 （全部存储区的所有扇区 + 备份 SRAM）	保护级别 1： – 当调试器已连接或自举配置与用户 Flash 不同时，不允许访问（读取、擦除和编程）Flash 或备份 SRAM – 执行 Flash 批量擦除可移除保护 保护级别 2： – 调试访问（SW 或 JTAG）禁用 – 无系统自举程序访问(1) – 不允许 RAM 自举 – 保护级别 2 是永久性的
写保护	扇区级 (128 KB)	防止对用户 Flash 扇区进行意外的写入/修改操作
PCROP/ execute-only 区域	区域级 （256 字节粒度）	只允许执行访问。 此保护在敏感代码上设置，可防止其它软件进行调试访问和代码复制。

1. 在安全访问模式下允许访问根安全服务。请参见*第 4.4 节：安全访问模式*。

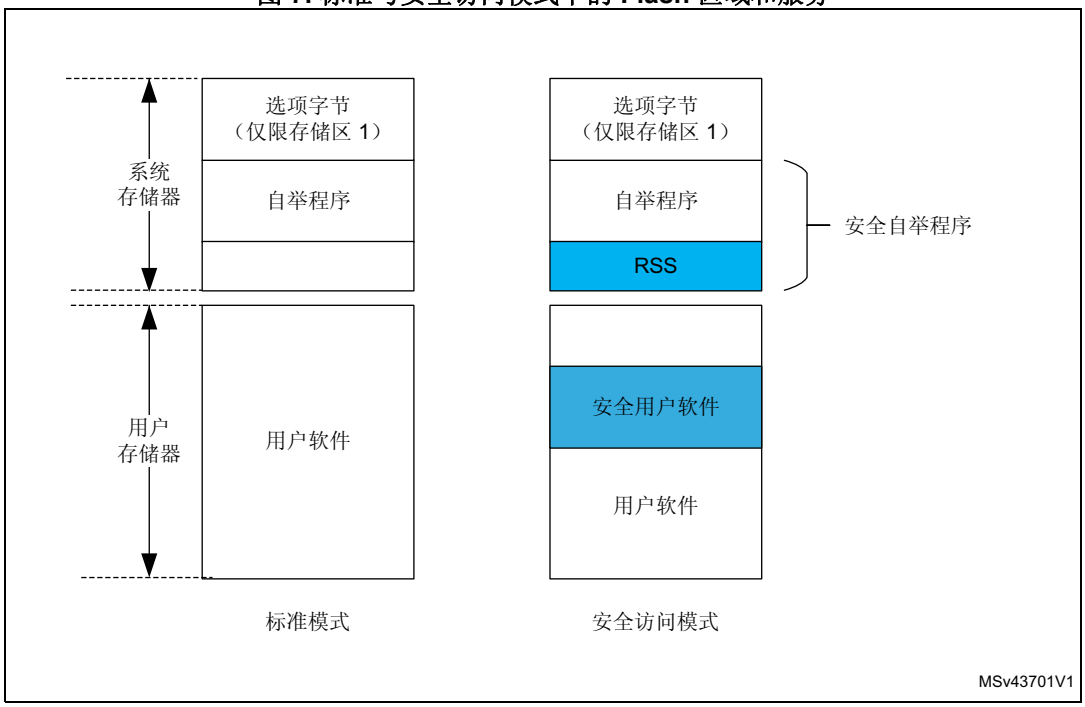
4.4 安全访问模式

某些敏感功能需要安全执行以免受潜在的恶意软件攻击。安全固件更新 (SFU) 软件是需要高保护级别的代码的良好示例，因为它控制的机密数据 （例如，加密固件或密钥）无法被其它进程恢复。

STM32H7x3 微控制器具有限制访问的安全存储区。这些存储区允许创建可在任何用户应用前执行的安全服务。仅当在安全访问模式下配置新器件时，这些安全区与其中包含的软件才能访问。

图 7 给出了标准与安全访问模式下的 Flash 区域和服务的概述。

图 7. 标准与安全访问模式下的 Flash 区域和服务



1. 只能在安全访问模式下访问的保护区域标记为蓝色。

4.4.1 相关特性

安全访问模式可通过选项字节配置。设置后，将使能访问：

- 安全自举程序，嵌入了自举程序加上意法半导体提供的一些安全服务（请参见*第 4.5 节：根安全服务 (RSS)*）。
- 安全用户存储器嵌入了安全用户代码和数据。

有关每个内核访问权限的汇总，请参见*第 4.7 节：Flash 保护机制汇总*。

4.4.2 自举状态机

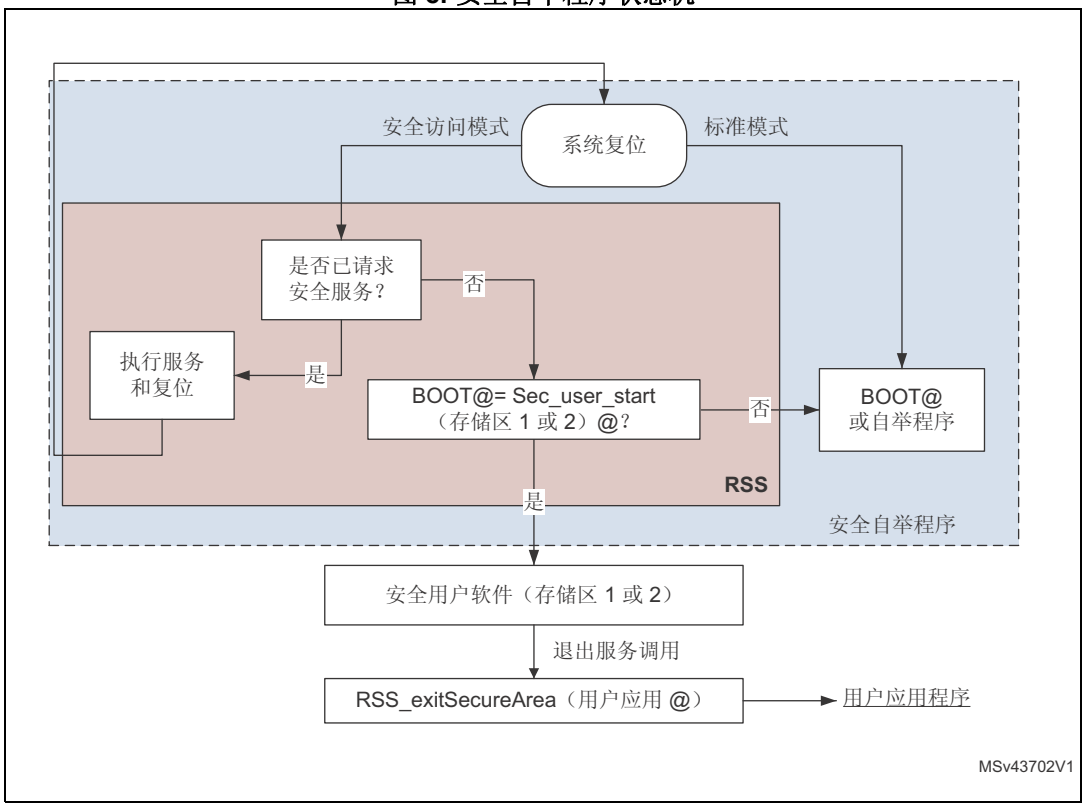
在安全访问模式下，安全自举程序将取代标准自举程序。RSS 中的自举强制执行，与自举配置（自举引脚和自举地址）无关。因此，自举安全服务和安全用户软件将优先于器件中的所有其他代码（包括传统自举程序）。

执行优先级

1. 在系统复位后，自举安全服务具有最高优先级。将执行调用的服务并在之后立即触发新的系统复位。关于可用安全服务的详细信息，请参见*第 4.5 节：根安全服务 (RSS)*。
2. 如果没有请求服务，则只要已定义安全用户软件并相应设置了自举地址，安全用户软件便会执行。关于安全用户软件的设置，请参见*第 4.6 节：安全用户软件*。
3. 否则，将遵循传统自举状态机：自举程序或从 Flash 自举。

图 8 显示 STM32H7x3 自举状态机。

图 8. 安全自举程序状态机



4.4.3 安全访问模式配置

使能安全访问模式

如何在器件上激活安全访问模式没有限制。它通过 FLASH_OPTSR_CUR 寄存器中的 SECURITY 选项位配置（请参见*第 3.5.8 节：FLASH 选项状态寄存器（当前值）(FLASH_OPTSR_CUR)*）。

安全访问模式在系统复位后激活。

禁止安全访问模式

禁用安全访问模式是一个更敏感的任务，因为这只能在器件上不再存在受保护代码时执行。因此，要返回标准模式，应当在清除 FLASH_OPTSR_CUR 寄存器中的 SECURITY 选项位之前移除安全用户存储器和 PCROP/仅执行区域。

可通过执行 Flash 批量擦除、存储区擦除或调用专用安全服务（仅适用于 PCROP 区域）来移除受保护区域：

- 关于移除 PCROP 保护，请参见 *PCROP 区域（专有代码读保护，仅执行区域）*一节和 *RSS_resetAndDestroyPCROPArea* 安全服务。
- 关于如何移除安全区域的说明，请参见*第 3.3.12 节：保护机制*。

4.5 根安全服务 (RSS)

意法半导体提供了用于配置安全区域的服务。在安全访问模式下，这些根安全服务在系统复位后执行，优先于器件中存储的任何其他软件。

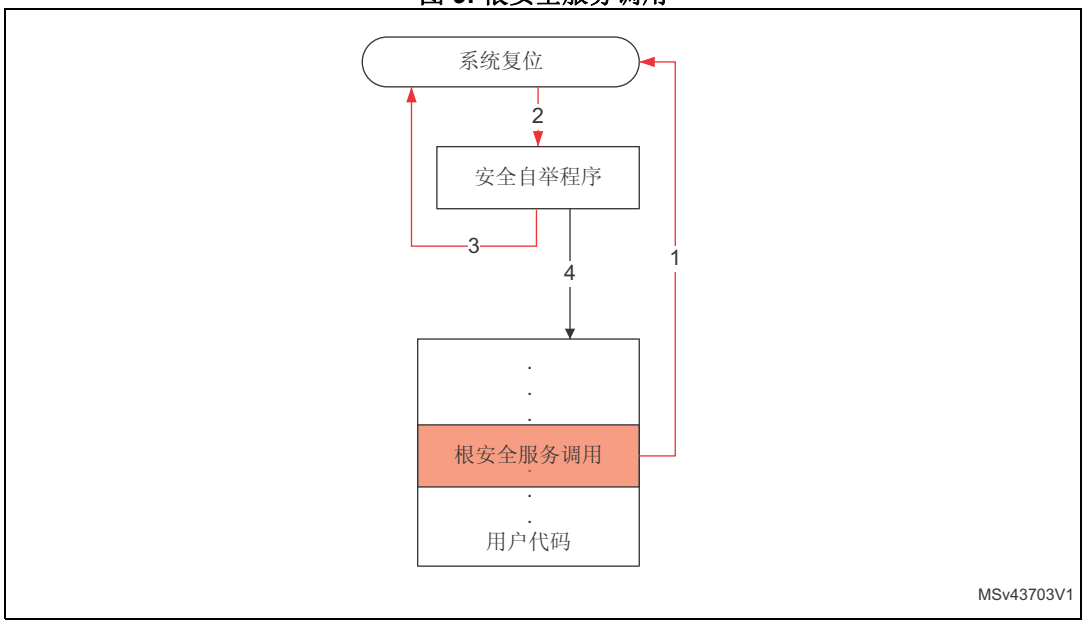
当 STM32H7x3 在标准模式下运行时，将无法访问（读取、写入、执行和调试）RSS 软件。

当在安全访问模式下配置 STM32H7x3 时，SS 软件在复位后仅运行一次。当 RSS 软件执行完成时，将无法再被访问（没有其他代码可以直接跳转到 RSS）。执行复位后可调用任何 RSS 例程。

4.5.1 调用根安全服务

通过可触发软件复位的公共 API 来调用安全服务（请参见*图 9*，步骤 1），这样便可通过可用的 RSS 代码执行安全自举程序（请参见*图 9*，步骤 2）。一旦执行，大多数服务会触发系统复位，（请参见*图 9*，步骤 3），由于不再需要服务，因此安全自举程序会跳转到用户软件（请参见*图 9*，步骤 4）。

图 9. 根安全服务调用



4.5.2 根安全服务说明

以下安全服务可用于管理 PCROP、安全用户存储器和安全模式设置。

RSS_resetAndInitializeSecureAreas

原型	void RSS_resetAndInitializeSecureAreas(RSS_SecureArea_t area)
参数	安全用户区域起始地址和结束地址。可设置一至两个安全用户区域。
说明	此服务按照选择字节寄存器中存储的值来设置安全用户区域边界： – 存储区 1 的 SEC_AREA_START1 和 SEC_AREA_END1 – 存储区 12 的 SEC_AREA_START1 和 SEC_AREA_END1 此服务仅在第一次设置安全区域时使用。 服务完成后将触发系统复位。

RSS_exitSecureArea

原型	void RSS_exitSecureArea(unsigned int vectors)
参数	退出后的应用向量跳转的地址
说明	此服务用于退出安全用户软件并跳转至用户主应用 此服务不会触发系统复位

RSS_resetAndDestroyPCROPArea

原型	RSS_resetAndDestroyPCROPArea(RSS_FlashBank_et bank)
参数	目标存储区编号
说明	此服务覆盖 PCROP 区域并移除 PCROP 保护 避免 Flash/存储器批量擦除 服务完成后将触发系统复位

4.6 安全用户软件

安全用户软件是一种可信代码，在器件上电或系统复位后执行。它能够创建安全应用，例如：

- 代码签名或完整性检查（用户安全自举）
- 软件许可证检查
- 安全固件更新

4.6.1 访问规则

只能在安全访问模式下访问，安全用户软件存储在安全用户存储器中，安全用户存储器是一个受保护的区域，属于用户主存储器的一部分。

每个存储区只能配置一个用户安全区域。

只有根安全服务具有较高的优先级，因此可抢占安全用户软件执行（请参见*第 4.4.2 节：自举状态机*）。

当安全用户软件执行后，代码会跳转到主用户应用，防止访问安全用户区域。这通过使用指定为参数的应用代码地址调用 *RSS_exitSecureArea* 安全服务来实现。

在应用代码中，对安全用户区域的任何访问都会触发 Flash 错误。

4.6.2 设置安全用户存储区

每个存储区可设置一个大小可配置的安全区域。每个区域的大小可在 512 字节到整个存储区的范围内设置，粒度为 256 字节：

- 存储区 1 中的安全区域

  通过 SEC_AREA_START1 和 FLASH_SCAR_CUR1 中的 SEC_AREA_END1 选项位配置边界（请参见*第 3.5.13 节：存储区 1 的 FLASH 安全地址（当前值）(FLASH_SCAR_CUR1)*）。
- 存储区 2 中的安全区域

  通过 SEC_AREA_START2 和 FLASH_SCAR_CUR2 中的 SEC_AREA_END2 选项位配置边界（请参见*第 3.5.30 节：存储区 2 的 FLASH 安全地址（当前值）(FLASH_SCAR_CUR2)*）。

注：　如果安全区域的起始地址等于结束地址，则认为整个存储区都处于安全保护下。

激活的设置根据是否在器件中定义了安全用户区域而不同：

- 不存在安全用户区域（首次设置）：

  应调用专用的根安全服务 *RSS_resetAndInitializeSecureAreas* 来初始化 ST 自举状态机中的空地址。两个安全区域（每个存储区一个）可设置为相同操作。
- 安全用户区域已存在：

  在这种情况下，安全用户区域代码可更新其自己的安全用户区域大小，或在其他存储区中创建一个新的安全用户区域。

  请注意，*RSS_resetAndInitializeSecureAreas* 功能无法用于创建第二个用户安全区域。

4.6.3 移除安全用户存储区

只能通过执行 Flash 批量擦除或存储区擦除操作来擦除安全用户区域。

- 当 RDP 级别 1 降级至 RDP 级别 0 时，将触发 Flash 批量擦除。
- 设置 FLASH_CR1/2 寄存器的 BER1/2 位来触发存储区擦除（请参见*第 3.5.4 节：存储区 1 的 FLASH 控制寄存器 (FLASH_CR1)* 和*第 3.5.25 节：存储区 2 的 FLASH 控制寄存器 (FLASH_CR2)*）。

在两种情况下，在 FLASH_PRAR_CUR1/2 寄存器中定义擦除策略的 DMES1/2 位均应置 1（请参见*第 3.5.11 节：存储区 1 的 FLASH 保护地址（当前值）(FLASH_PRAR_CUR1)* 和*第 3.5.28 节：存储区 2 的 FLASH 保护地址（当前值）(FLASH_PRAR_CUR2)*，区域的起始地址应高于其结束地址。

4.6.4 选择安全用户软件

当前自举地址会从两个安全用户软件（每个存储区一个）中选择一个。由于可以定义两个自举地址，因此必须考虑多个用例。

FLASH_BOOT_CURR 寄存器中 BOOT_ADD0 和 BOOT_ADD1 之间自举地址的选择由 BOOT 引脚（请参见*第 2.5 节：启动配置*）或交换存储区机制完成（请参见*第 3.3.13 节：Flash 存储区交换*）。

表 19 给出了不同的自举地址寄存器设置和潜在的相关场景。

表 19. 安全用户软件选择用例(1)

BOOT_ADD0（自举引脚 =0）	BOOT_ADD1（自举引脚 =1）	应用案例
SEC_AREA_START1	SEC_AREA_START2	通过 BOOT 引脚在两个安全用户软件之间切换（当前与恢复或当前与前一个）
SEC_AREA_START1	SEC_AREA_START1	从每个存储区中的相同偏移量开始，通过存储区交换功能在两个安全用户软件之间切换（前一个安全自举到当前）
SEC_AREA_START1	恢复固件	通过 BOOT 引脚在安全用户软件和非安全恢复自举之间切换
SEC_AREA_START1	自举程序	通过 BOOT 引脚进行基于自举程序的恢复（在非安全模式下）

1. 存储区编号和自举地址寄存器编号不相关。BOOT_ADDx 或者可设置为存储区 1 或存储区 2。

如果自举地址与两个安全用户区域不同，系统会直接跳转到所选地址，绕过安全用户软件。

4.7 Flash 保护机制汇总

图 10 和*表 20* 汇总了安全访问和标准模式下不同 Flash 区域的访问权限。

图 10. 内核对 Flash 区域的访问

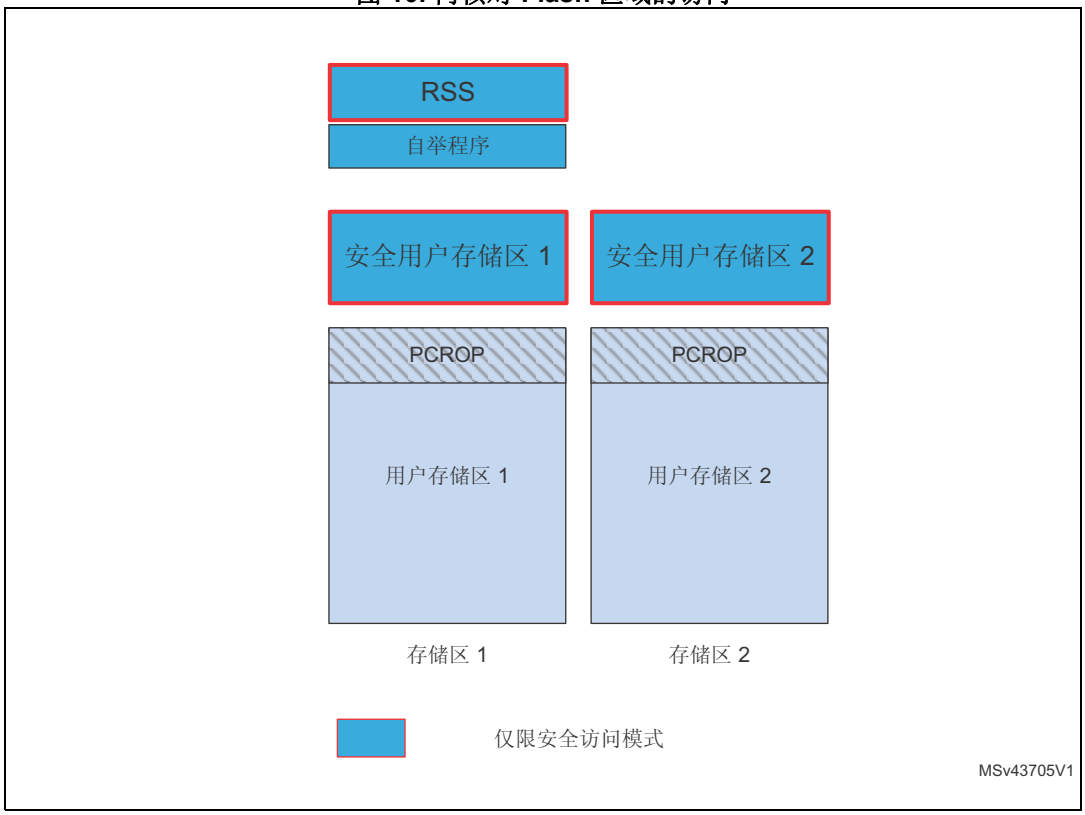


表 20. Flash 保护区域访问权限汇总

访问类型	SW 区域	安全模式	访问
执行	PCROP	任意	✓
	安全用户软件	安全访问	✓(1)
	根安全服务	安全访问	✓(1)
读访问	PCROP	任意	无
	安全用户软件	安全访问	✓(1)
	根安全服务	安全访问	✓
调试访问	PCROP	任意	无
	安全用户软件	安全访问	无
	根安全服务	安全访问	无

1. 仅在复位后、代码完成前才会授予访问权限。

5 AXI 互连

5.1 AXI简介

AXI（高级可扩展接口）互连基于 ARM® CoreLink™ NIC-400 网络互连。该互连具有六个发起者端口或 ASIB（AMBA 从接口模块）和七个目标端口或 AMIB（AMBA 主接口模块）。ASIB 通过一个 AXI 开关矩阵连接至 AMIB。

每个 ASIB 均可作为 AXI 总线 或 AHB（高级高性能总线）的从器件。类似地，每个 AMIB 均可作为 AXI 或 AHB 总线的主器件。当 ASIB 或 AMIB 连接至 AHB，其可在 AHB 和 AXI 协议之间转换。

AXI 互连包括 GPV（全局编程器视图）。该视图具有配置某些参数（例如每个 ASIB 的 QoS（服务质量）等级）的寄存器。

默认从器件处理全部未分配地址空间，进而产生返回信号。这确保了此类事务顺利完成并且不会阻断主器件和 ASIB 的发布。

5.2 AXI 互连主要特性

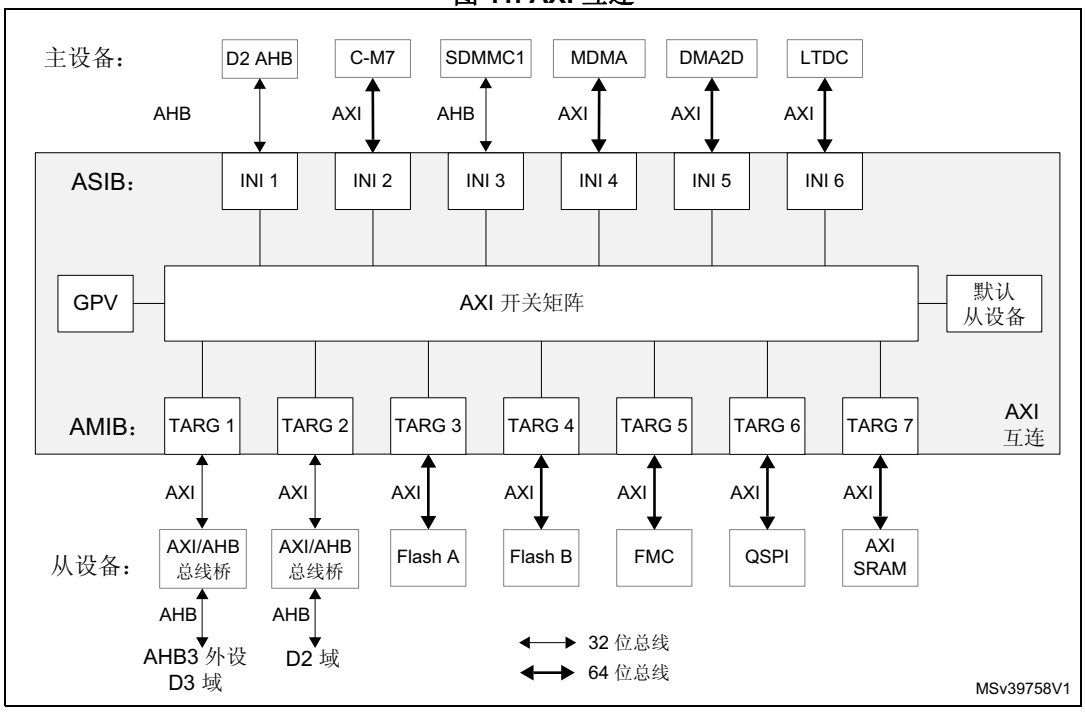
- 具有六个 ASIB 和七个 AMIB 的 64 位 AXI 总线开关矩阵，位于 D1 域中
- AHB/AXI 桥函数编入 ASIB
- 多个 ASIB 与多个 AMIB 的并发连接
- 可编程通信优先级管理（QoS - 服务质量）
- 通过 GPV 可由软件进行配置

5.3 AXI 互连功能说明

5.3.1 框图

AXI 互连如*图 11* 所示。

图 11. AXI 互连



5.3.2 ASIB配置

表 21 汇总了 ASIB 的特性。

表 21. ASIB配置

ASIB	连接的主器件	协议	总线宽度	R/W 发布
INI 1	D2 域的 AHB	AHB-Lite	32	1/4
INI 2	Cortex-M7	AXI4	64	7/32
INI 3	SDMMC1	AHB-Lite	32	1/4
INI 4	MDMA	AXI4	64	4/1
INI 5	DMA2D	AXI4	64	2/1
INI 6	LTDC	AXI4	64	1/1

5.3.3 AMIB配置

表 22 汇总了 AMIB 的特性。

表 22. AMIB配置

AMIB	连接的从器件	协议	总线宽度	R/W/ 总体验收
TARG 1	外设 3 和 D3 AHB	AXI4(1)	32	1/1/1
TARG 2	D2 AHB	AXI4(1)	32	1/1/1
TARG 3	Flash A	AXI4	64	3/2/5
TARG 4	Flash B	AXI4	64	3/2/5
TARG 5	FMC	AXI4	64	3/3/6
TARG 6	QUADSPI	AXI4	64	2/1/3
TARG 7	AXI SRAM	AXI3	64	2/2/2

1. 通过位于 AXI 互连和连接的从器件之间的 AXI/AHB 桥可转换为 AHB 协议。

5.3.4 服务质量 (QoS)

当两个 ASIB 同时尝试访问同一 AMIB 时，AXI 开关矩阵使用基于优先级的仲裁。每个 ASIB 均具有可编程读取通道和写入通道优先级，即 QoS，其范围为 0 到 15，数值越高表示优先级越高。读取通道 QoS 值在 *AXI 互连 - INI x 读取 QoS 寄存器 (AXI_INIx_READ_QOS)* 中编程，而写入通道 QoS 值在 *AXI 互连 - INI x 写入 QoS 寄存器 (AXI_INIx_WRITE_QOS)* 中编程。所有通道的默认 QoS 值是 0（最低优先级）。

如果两个重合事务到达同一 AMIB，则高优先级事务在低优先级事务前通过。如果两事务的 QoS 值相同，则采用最近最少使用 (LRU) 优先级方案。

QoS 值应根据应用的延时要求进行编程。为 ASIB 设置较高优先级，确保由相关总线主器件发起的事务延时较低。这对于受限的实时任务十分有用，例如图形处理（LTDC，DMA2D）。为多次频繁访问同一从器件（如 Cortex-M7 CPU）的主器件分配高优先级可阻止其它低优先级主器件访问该从器件。

5.3.5 全局编程器视图 (GPV)

GPV 包含适用于 AXI 互连的配置寄存器（请参见*第 5.4 节*）。这些寄存器仅可通过 Cortex-M7 CPU 进行访问。

5.4 AXI 互连寄存器

5.4.1 AXI 互连 - 外设 ID4 寄存器 (AXI_PERIPH_ID_4)

AXI interconnect - peripheral ID4 register

偏移地址：0x1FD0

复位值：0x0000 0004

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	4KCOUNT[3:0]				JEP106CON[3:0]			
								r				r			

位 31:8 保留，必须保持复位值。

位 7:4 **4KCOUNT[3:0]**：寄存器文件大小 (Register file size)
0x0：N/A

位 3:0 **JEP106CON[3:0]**：JEP106 连续代码 (JEP106 continuation code)
0x4：ARM®

5.4.2 AXI 互连 - 外设 ID0 寄存器 (AXI_PERIPH_ID_0)

AXI interconnect - peripheral ID0 register

偏移地址：0x1FE0

复位值：0x0000 0000

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

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

位 31:8 保留，必须保持复位值。

位 7:0 **PARTNUM[7:0]**：外设产品编号位 0 到 7 (Peripheral part number bits 0 to 7)
0x00：产品编号 = 0x400

5.4.3 AXI 互连 - 外设 ID1 寄存器 (AXI_PERIPH_ID_1)

AXI interconnect - peripheral ID1 register

偏移地址：0x1FE4

复位值：0x0000 00B4

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	JEP106ID[3:0]				PARTNUM[11:8]			
								r				r			

位 31:8 保留，必须保持复位值。

位 7:4 **JEP106ID[3:0]**：JEP106 识别位 0 到 3 (JEP106 identity bits 0 to 3)
0xB：ARM® JEDEC 代码

位 3:0 **PARTNUM[11:8]**：外设产品编号位 8 到 11 (Peripheral part number bits 8 to 11)
0x4：产品编号 = 0x400

5.4.4 AXI 互连 - 外设 ID2 寄存器 (AXI_PERIPH_ID_2)

AXI interconnect - peripheral ID2 register

偏移地址：0x1FE8

复位值：0x0000 002B

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REVISION[3:0]				JEDEC	JEP106ID[6:4]		
								r				r	r		

位 7:4 **REVISION[3:0]**：外设版本编号 (Peripheral revision number)
0x2：r0p2

位 3 **JEDEC**：JEP106 代码标志 (JEP106 code flag)
0x1：JEDEC 分配的代码

位 2:0 **JEP106ID[6:4]**：JEP106 识别位 4 到 6 (JEP106 Identity bits 4 to 6)
0x3：ARM® JEDEC 代码

5.4.5 AXI 互连 - 外设 ID3 寄存器 (AXI_PERIPH_ID_3)

AXI interconnect - peripheral ID3 register

偏移地址：0x1FEC

复位值：0x0000 0000

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

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

位 31:8 保留，必须保持复位值。

位 7:4 **REV_AND[3:0]**：客户版本 (Customer version)

0：无

位 3:0 **CUST_MOD_NUM[3:0]**：客户修改 (Customer modification)

0：无

5.4.6 AXI 互连 - 组件 ID0 寄存器 (AXI_PERIPH_ID_0)

AXI interconnect - component ID0 register

偏移地址：0x1FF0

复位值：0x0000 000D

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

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

位 31:8 保留，必须保持复位值。

位 7:0 **PREAMBLE7:0]**：报头位 0 到 7 (Preamble bits 0 to 7)

0xD：公共 ID 值

5.4.7 AXI 互连 - 组件 ID1 寄存器 (AXI_PERIPH_ID_1)

AXI interconnect - component ID1 register

偏移地址：0x1FF4

复位值：0x0000 00F0

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

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

位 31:8 保留，必须保持复位值。

位 7:4 **CLASS[3:0]**：组件类 (Component class)
0xF：通用 IP 组件类

位 3:0 **PREAMBLE[11:8]**：报头位 8 到 11 (Preamble bits 8 to 11)
0x0：公共 ID 值

5.4.8 AXI 互连 - 组件 ID2 寄存器 (AXI_PERIPH_ID_2)

AXI interconnect - component ID2 register

偏移地址：0x1FF8

复位值：0x0000 0005

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[19:12]							
								r							

位 31:8 保留，必须保持复位值。

位 7:0 **PREAMBLE[19:12]**：报头位 12 到 19 (Preamble bits 12 to 19)
0x05：公共 ID 值

5.4.9 AXI 互连 - 组件 ID3 寄存器 (AXI_PERIPH_ID_3)

AXI interconnect - component ID3 register

偏移地址：0x1FFC

复位值：0x0000 00B1

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[27:20]							
								r							

位 31:8 保留，必须保持复位值。

位 7:0 **PREAMBLE[27:20]**：报头位 20 到 27 (Preamble bits 20 to 27)
0xB1：公共 ID 值

5.4.10 AXI 互连 - TARG x 总线矩阵发布功能寄存器 (AXI_TARGx_FN_MOD_ISS_BM)

AXI interconnect - TARG x bus matrix issuing functionality register

偏移地址：0x1008 + 0x1000 * x，其中 x = 1 到 7

复位值：0x0000 0000

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

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

位 31:2 保留，必须保持复位值。

位 1 **WRITE_ISS_OVERRIDE**：针对目标器件的开关矩阵写入发布覆盖 (Switch matrix write issuing override for target)
0：正常发布功能
1：开关矩阵写入发布功能置 1

位 0 **READ_ISS_OVERRIDE**：针对目标器件的开关矩阵读取发布覆盖 (Switch matrix read issuing override for target)
0：正常发布功能
1：开关矩阵读取发布功能置 1

5.4.11 AXI 互连 - TARG x 总线矩阵功能 2 寄存器 (AXI_TARGx_FN_MOD2)

AXI interconnect - TARG x bus matrix functionality 2 register

偏移地址：0x1024 + 0x1000 * x，其中 x = 1、2 和 7

复位值：0x0000 0000

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

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

位 31:1 保留，必须保持复位值。

位 0 **BYPASS_MERGE**：禁止节拍封装，以匹配输出数据宽度。未对齐事务不会与输入数据字边界重新对齐。

0：正常工作

1：禁止封装

5.4.12 AXI 互连 - TARG x 长突发功能修改寄存器 (AXI_TARGx_FN_MOD_LB)

AXI interconnect - TARG x long burst functionality modification register

偏移地址：0x102C + 0x1000 * x，其中 x=1 和 2

复位值：0x0000 0000

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

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

位 31:1 保留，必须保持复位值。

位 0 **FN_MOD_LB**：控制长突发的突发中断 (Controls burst breaking of long bursts)

0：ASIB 输出不生成长突发

1：ASIB 输出可生成长突发

5.4.13 AXI 互连 - TARG x 发布功能修改寄存器 (AXI_TARGx_FN_MOD)

AXI interconnect - TARG x issuing functionality modification register

偏移地址：0x1108 + 0x1000 * x，其中 x = 1、2 和 7

复位值：0x0000 0000

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

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

位 31:2 保留，必须保持复位值。

位 1 **WRITE_ISS_OVERRIDE**：覆盖 AMIB 写入发布功能 (Override AMIB write issuing capability)
- 0：正常发布功能
- 1：将发布功能强制置 1

位 0 **READ_ISS_OVERRIDE**：覆盖 AMIB 读取发布功能 (Override AMIB read issuing capability)
- 0：正常发布功能
- 1：将发布功能强制置 1

5.4.14 AXI 互连 - INI x 功能修改 2 寄存器 (AXI_INIx_FN_MOD2)

AXI interconnect - INI x functionality modification 2 register

偏移地址：0x41024 + 0x1000 * x，其中 x = 1 和 3

复位值：0x0000 0000

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

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

位 31:1 保留，必须保持复位值。

位 0 **BYPASS_MERGE**：除非协议要求，否则禁止使用增大器进行事务变更。
- 0：正常工作
- 1：在允许的情况下，通过的事务未发生变化

5.4.15 AXI 互连 - INI x AHB 功能修改寄存器 (AXI_INIx_FN_MOD_AHB)

AXI interconnect - INI x AHB functionality modification register

偏移地址：0x41028 + 0x1000 * x，其中 x = 1 和 3

复位值：0x0000 0000

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

位 31:2 保留，必须保持复位值。

位 1 **WR_INC_OVERRIDE**：将所有 AHB-Lite 读取事务转换为一系列单节拍 AXI 事务。
0：禁止覆盖
1：使能覆盖

位 0 **RD_INC_OVERRIDE**：将所有 AHB-Lite 写入事务转换为一系列单节拍 AXI 事务，每个 AHB-Lite 写入节拍由 AXI 缓存的写入响应确认。
0：禁止覆盖
1：使能覆盖

5.4.16 AXI 互连 - INI x 读取 QoS 寄存器 (AXI_INIx_READ_QOS)

AXI interconnect - INI x read QoS register

偏移地址：0x41100 + 0x1000 * x，其中 x = 1 到 6

复位值：0x0000 0000

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

位 31:4 保留，必须保持复位值。

位 3:0 **AR_QOS[3:0]**：读取通道 QoS 设置 (Read channel QoS setting)
0x0：最低优先级
0xF：最高优先级

5.4.17 AXI 互连 - INI x 写入 QoS 寄存器 (AXI_INIx_WRITE_QOS)

AXI interconnect - INI x write QoS register

偏移地址：0x41104 + 0x1000 * x，其中 x = 1 到 6

复位值：0x0000 0000

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

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

位 31:4 保留，必须保持复位值。

位 3:0 **AW_QOS[3:0]**：写入通道 QoS 设置 (Write channel QoS setting)

0x0：最低优先级

0xF：最高优先级

5.4.18 AXI 互连 - INI x 发布功能修改寄存器 (AXI_INIx_FN_MOD)

AXI interconnect - INI x issuing functionality modification register

偏移地址：0x41108 + 0x1000 * x，其中 x = 1 到 6

复位值：0x0000 0000

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

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

位 31:2 保留，必须保持复位值。

位 1 **WRITE_ISS_OVERRIDE**：覆盖 ASIB 写入发布功能 (Override ASIB write issuing capability)

0：正常发布功能

1：将发布功能强制置 1

位 0 **READ_ISS_OVERRIDE**：覆盖 ASIB 读取发布功能 (Override ASIB read issuing capability)

0：正常发布功能

1：将发布功能强制置 1

5.5 AXI 互连寄存器映射

表 23. AXI 寄存器映射和复位值

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x1FD0	AXI_PERIPH_ID_4	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	4KCOUNT[3:0]				JEP106CON[3:0]			
	Reset value																									0	0	0	0	0	1	0	0
0x1FD4	AXI_PERIPH_ID_5	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	保留							
	Reset value																									0	0	0	0	0	0	0	0
0x1FD8	AXI_PERIPH_ID_6	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	保留							
	Reset value																									0	0	0	0	0	0	0	0
0x1FDC	AXI_PERIPH_ID_7	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	保留							
	Reset value																									0	0	0	0	0	0	0	0
0x1FE0	AXI_PERIPH_ID_0	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PARTNUM[7:0]							
	Reset value																									0	0	0	0	0	0	0	0
0x1FE4	AXI_PERIPH_ID_1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	JEP106ID[3:0]				PARTNUM[11:8]			
	Reset value																									1	0	1	1	0	1	0	0
0x1FE8	AXI_PERIPH_ID_2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	版本[3:0]				JEDEC	JEP106ID[6:4]		
	Reset value																									0	0	1	0	1	0	1	1
0x1FEC	AXI_PERIPH_ID_3	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REV_AND[3:0]				CUST_MOD_NUM[3:0]			
	Reset value																									0	0	0	0	0	0	0	0
0x1FF0	AXI_COMP_ID_0																									PREAMBLE[7:0]							
	Reset value																									0	0	0	0	1	1	0	1
0x1FF4	AXI_COMP_ID_1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CLASS[3:0]				报头[11:8]			
	Reset value																									1	1	1	1	0	0	0	0

表 23. AXI 寄存器映射和复位值（续）

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x1FF8	AXI_COMP_ID_2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[19:12]							
	Reset value																									0	0	0	0	0	1	0	1
0x1FFC	AXI_COMP_ID_3	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[27:20]							
	Reset value																									1	0	1	1	0	0	0	1
0x2000 - 0x2004	保留	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
0x2008	AXI_TARG1_FN_MOD_ISS_BM	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WRITE_ISS_OVERRIDE	READ_ISS_OVERRIDE
	Reset value																															0	0
0x200C-0x2020	保留	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
0x2024	AXI_TARG1_FN_MOD2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BYPASS_MERGE
	Reset value																																0
0x2028	保留	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
0x202C	AXI_TARG1_FN_MOD_LB	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FN_MOD_LB
	Reset value																																0
0x2030 - 0x2104	保留	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
0x2108	AXI_TARG1_FN_MOD	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WRITE_ISS_OVERRIDE	READ_ISS_OVERRIDE
	Reset value																															0	0
0x210C - 0x3004	保留	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

表 23. AXI 寄存器映射和复位值（续）

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x3008	AXI_TARG2_ FN_MOD_ ISS_BM	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WRITE_ISS_OVERRIDE	READ_ISS_OVERRIDE
	Reset value																															0	0
0x300C - 0x3020	保留	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
0x3024	AXI_TARG2_ FN_MOD2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BYPASS_MERGE
	Reset value																																0
0x3028	保留	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
0x302C	AXI_TARG2_ FN_MOD_LB	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FN_MOD_LB
	Reset value																																0
0x3030 - 0x3104	保留	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
0x3108	AXI_TARG2_ FN_MOD	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WRITE_ISS_OVERRIDE	READ_ISS_OVERRIDE
	Reset value																															0	0
0x310C - 0x4004	保留	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
0x4008	AXI_TARG3_ FN_MOD_ ISS_BM	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WRITE_ISS_OVERRIDE	READ_ISS_OVERRIDE
	Reset value																															0	0
0x400C - 0x5004	保留	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

表 23. AXI 寄存器映射和复位值（续）

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x5008	AXI_TARG4_FN_MOD_ISS_BM	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WRITE_ISS_OVERRIDE	READ_ISS_OVERRIDE
	Reset value																															0	0
0x500C - 0x6004	保留	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
0x6008	AXI_TARG5_FN_MOD_ISS_BM	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WRITE_ISS_OVERRIDE	READ_ISS_OVERRIDE
	Reset value																															0	0
0x600C - 0x7004	保留	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
0x7008	AXI_TARG6_FN_MOD_ISS_BM	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WRITE_ISS_OVERRIDE	READ_ISS_OVERRIDE
	Reset value																															0	0
0x700C - 0x8004	保留	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
0x8008	AXI_TARG7_FN_MOD_ISS_BM	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WRITE_ISS_OVERRIDE	READ_ISS_OVERRIDE
	Reset value																															0	0
0x800C - 0x8020	保留	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
0x8024	AXI_TARG7_FN_MOD2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BYPASS_MERGE
	Reset value																																0

表 23. AXI 寄存器映射和复位值（续）

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x8028 - 0x8104	保留	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
0x8108	AXI_TARG7_ FN_MOD	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WRITE_ISS_OVERRIDE	READ_ISS_OVERRIDE
	Reset value																															0	0
0x810C- 0x42020	保留	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
0x42024	AXI_INI1_ FN_MOD2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BYPASS_MERGE
	Reset value																																0
0x42028	AXI_INI1_FN_ MOD_AHB	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WR_INC_OVERRIDE	RD_INC_OVERRIDE
	Reset value																															0	0
0x4202C- 0x420FC	保留	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
0x42100	AXI_INI1_ READ_QOS	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	AR_QOS [3:0]			
	Reset value																													0	0	0	0
0x42104	AXI_INI1_ WRITE_QOS	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	AW_QOS [3:0]			
	Reset value																													0	0	0	0
0x42108	AXI_INI1_ FN_MOD	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WRITE_ISS_OVERRIDE	READ_ISS_OVERRIDE
	Reset value																															0	0
0x4210C- 0x430FC	保留	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

表 23. AXI 寄存器映射和复位值（续）

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x43100	AXI_INI2_ READ_QOS	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	AR_QOS [3:0]			
	Reset value																													0	0	0	0
0x43104	AXI_INI2_ WRITE_QOS	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	AW_QOS [3:0]			
	Reset value																													0	0	0	0
0x43108	AXI_INI2_ FN_MOD	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WRITE_ISS_OVERRIDE	READ_ISS_OVERRIDE
	Reset value																															0	0
0x4310C - 0x44020	保留	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
0x44024	AXI_INI3_ FN_MOD2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BYPASS_MERGE
	Reset value																																0
0x44028	AXI_INI3_ FN_MOD_AHB	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WR_INC_OVERRIDE	RD_INC_OVERRIDE
	Reset value																															0	0
0x4402C- 0x440FC	保留	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
0x44100	AXI_INI3_ READ_QOS	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	AR_QOS [3:0]			
	Reset value																													0	0	0	0
0x44104	AXI_INI3_ WRITE_QOS	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	AW_QOS [3:0]			
	Reset value																													0	0	0	0

表 23. AXI 寄存器映射和复位值（续）

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x44108	AXI_INI3_ FN_MOD	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WRITE_ISS_OVERRIDE	READ_ISS_OVERRIDE
	Reset value																															0	0
0x4410C- 0x450FC	保留	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
0x45100	AXI_INI4_ READ_QOS	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	AR_QOS [3:0]			
	Reset value																													0	0	0	0
0x45104	AXI_INI4_ WRITE_QOS	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	AW_QOS [3:0]			
	Reset value																													0	0	0	0
0x45108	AXI_INI4_ FN_MOD	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WRITE_ISS_OVERRIDE	READ_ISS_OVERRIDE
	Reset value																															0	0
0x4510C- 0x460FC	保留	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
0x46100	AXI_INI5_ READ_QOS	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	AR_QOS [3:0]			
	Reset value																													0	0	0	0
0x46104	AXI_INI5_ WRITE_QOS	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	AW_QOS [3:0]			
	Reset value																													0	0	0	0
0x46108	AXI_INI5_ FN_MOD	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WRITE_ISS_OVERRIDE	READ_ISS_OVERRIDE
	Reset value																															0	0
0x4610C- 0x470FC	保留	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
0x47100	AXI_INI6_ READ_QOS	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	AR_QOS [3:0]			
	Reset value																													0	0	0	0

表 23. AXI 寄存器映射和复位值（续）

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x47104	AXI_INI6_ WRITE_QOS	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	AW_QOS [3:0]			
	Reset value																													0	0	0	0
0x47108	AXI_INI6_ FN_MOD	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WRITE_ISS_OVERRIDE	READ_ISS_OVERRIDE
	Reset value																															0	0

有关寄存器边界地址的信息，请参见*第 100 页的第 2.2.2 节*。

6 电源控制 (PWR)

6.1 前言

电源控制部分 (PWR) 概述了不同电源域的电源架构以及电源配置控制器。

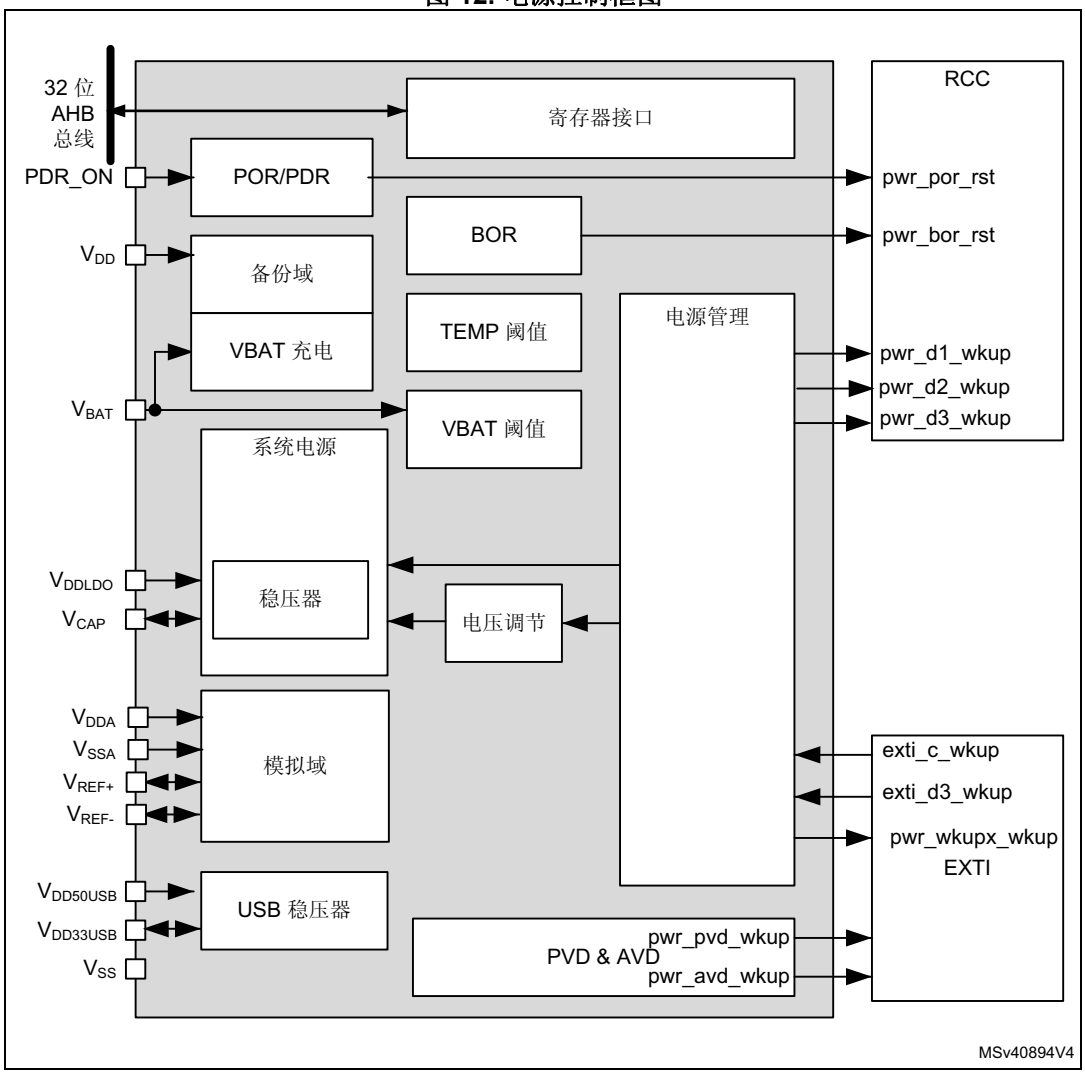
还介绍了电源监控器的特性并说明如何根据工作模式、所选性能（时钟频率）和电压调节来配置 V_{CORE} 电源域。

6.2 PWR主要特性

- 电源和电源域
  - 内核域 (V_{CORE})
  - VDD 域
  - 备份域（V_{SW} 和 V_{BKP}）
  - 模拟域 (V_{DDA})
- 系统电源稳压
  - 稳压器 (LDO)
- 外设电源稳压
  - USB 稳压器
- 电源监控
  - POR/PDR监控器
  - BOR监控器
  - PVD监控器
  - AVD监控器
  - VBAT 阈值
  - 温度阈值
- 电源管理
  - VBAT电池充电
  - 工作模式
  - 电压调节控制
  - 低功耗模式

6.3 PWR框图

图 12. 电源控制框图



6.3.1 PWR 引脚和内部信号

表 24 列出了连接到封装引脚或焊球的 PWR 输入与输出信号，*表 25* 则列出了内部 PWR 信号。

表 24. 连接到封装引脚或焊球的 PWR 输入/输出信号

引脚名称	信号类型	说明
V_{DD}	电源输入	主 I/O 和 V_{DD} 域电源输入
V_{DDA}	电源输入	模拟外设的外部模拟电源
V_{REF+} 和 V_{REF-}	电源输入 / 输出	ADC 和 DAC 的外部参考电压
V_{BAT}	电源输入	备份电池供电输入
V_{DDLDO}	电源输入	稳压器供电输入
V_{CAP}	电源输入 / 输出	数字内核域电源
$V_{DD50USB}$	电源输入	USB 稳压器供电输入
$V_{DD33USB}$	电源输入 / 输出	USB 稳压器供电输出
V_{SS}	电源输入	主接地
AHB	数字输入 / 输出	AHB 寄存器接口
PDR_ON	数字输入	掉电复位使能

表 25. PWR 内部输入/输出信号

信号名称	信号类型	说明
pwr_pvd_wkup	数字输出	可编程电压检测器输出
pwr_avd_wkup	数字输出	模拟电压检测器输出
pwr_wkupx_wkup	数字输出	CPU 唤醒信号（x=1 到 6）
pwr_por_rst	数字输出	上电复位
pwr_bor_rst	数字输出	欠压复位
exti_c_wkup	数字输入	CPU 唤醒请求
exti_d3_wkup	数字输入	D3 域唤醒请求
pwr_d1_wkup	数字输出	D1 域总线矩阵时钟唤醒请求
pwr_d2_wkup	数字输出	D2 域总线矩阵时钟唤醒请求
pwr_d3_wkup	数字输出	D3 域总线矩阵时钟唤醒请求

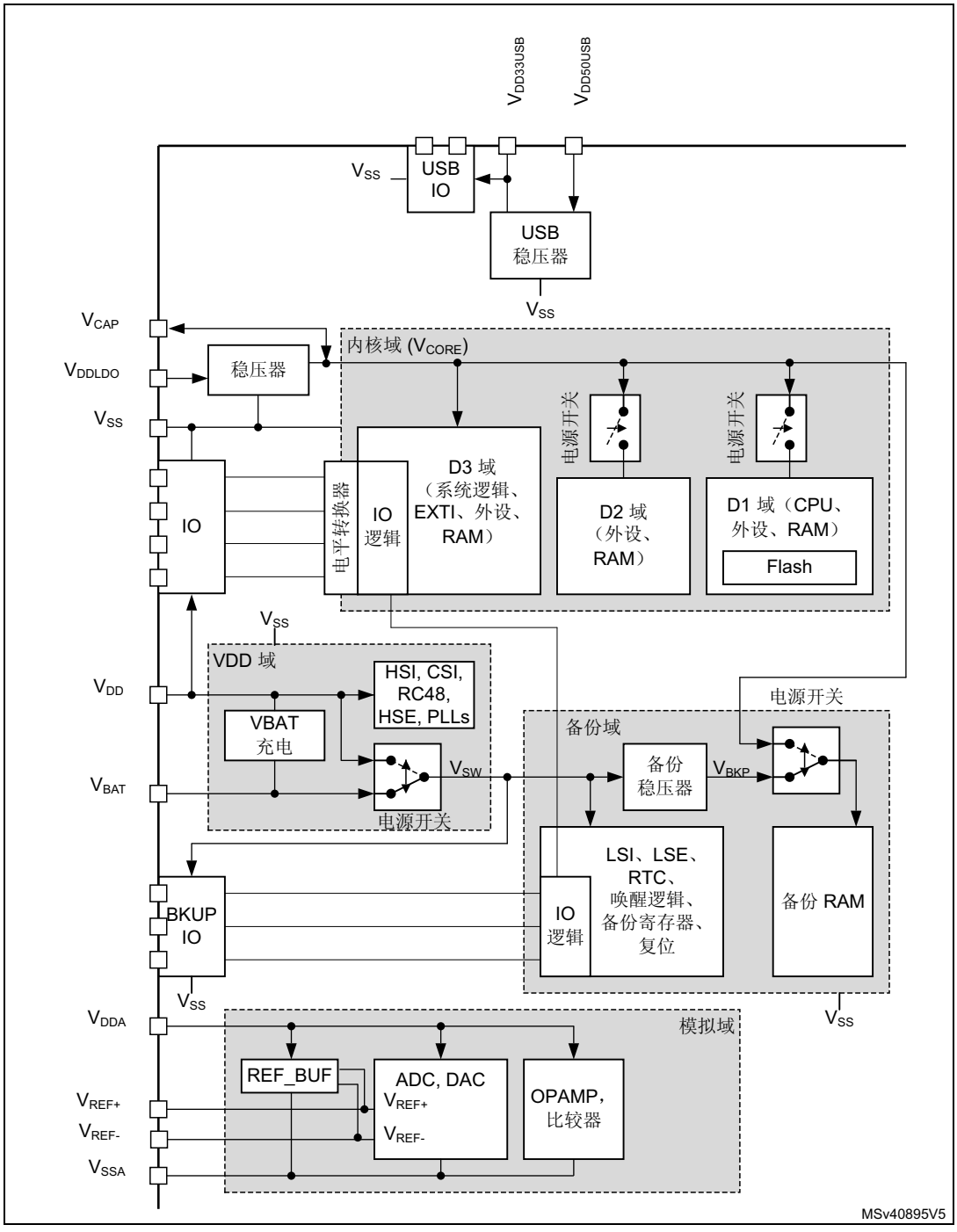
6.4 电源

器件需要 V_{DD} 电源以及用于 V_{DDLDO}、V_{DDA}、V_{DDUSB} 和 V_{CAP} 的独立电源。它还为特定功能提供稳压电源（稳压器和 USB 稳压器）。

- V_{DD}，为 I/O 和系统模拟模块（如复位、电源管理和时钟）供电的外部电源
- V_{BAT}，当 V_{DD} 不存在时（V_{BAT} 模式）为备份域供电的外部电源（可选）
  不使用电池时，该电源应连接到 V_{DD}
- V_{DDLDO}，为稳压器供电的外部电源
- V_{CAP} 数字内核域电源
  该电源独立于所有其它电源：
  - 当稳压器使能时，V_{CORE} 由内部稳压器提供。
  - 当稳压器禁止时，V_{CORE} 由外部电源通过 V_{CAP} 引脚提供。
- V_{DDA}，为 ADC、DAC、OPAMP、比较器和电压参考缓冲器供电的外部模拟电源。
  该电源独立于所有其它电源。
- V_{REF+}，ADC 和 DAC 的外部参考电压。
  - 当电压参考缓冲器使能时，V_{REF+} 和 V_{REF-} 由内部电压参考缓冲器提供。
  - 当电压参考缓冲器禁止时，V_{REF+} 由独立的外部参考电源提供。
- V_{SSA}，独立的模拟和参考电压地。
- $V_{DD50USB}$，为 USB 稳压器供电的外部电源。
- $V_{DD33USB}$，为 USB 接口供电的 USB 稳压器供电输出。
  - 当 USB 稳压器使能时，$V_{DD33USB}$ 由内部 USB 稳压器提供。
  - 当 USB 稳压器禁止时，$V_{DD33USB}$ 由独立的外部电源输入提供。
- V_{SS}，所有电源和模拟稳压器的公共地。

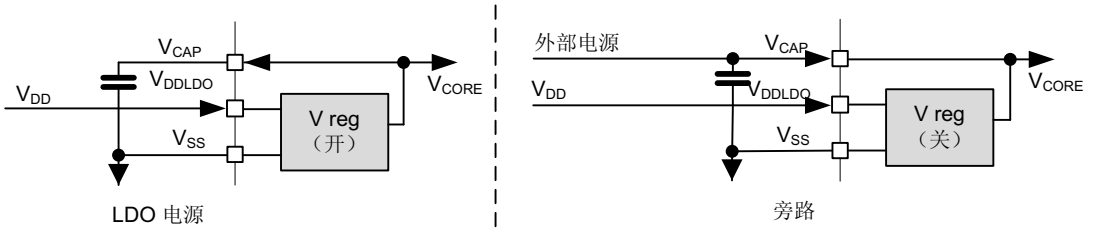
注：根据工作期间供电电压的不同，某些外设可能只提供有限的功能和性能。有关详细信息，请参见器件数据手册中“通用工作条件”一节。

图 13. 电源概述


VDD33USB
VDD50USB
VSS
USB IO
USB 稳压器
VSS
VCAP
VDDLDO
VSS
稳压器
内核域 (VCORE)
电源开关
电源开关
IO
电平转换器
IO 逻辑
D3 域（系统逻辑、EXTI、外设、RAM）
D2 域（外设、RAM）
D1 域（CPU、外设、RAM）
Flash
VSS
VDD 域
VDD
HSI, CSI, RC48, HSE, PLLs
VBAT 充电
VBAT
VSW
电源开关
电源开关
备份域
备份稳压器
VBKP
LSI、LSE、RTC、唤醒逻辑、备份寄存器、复位
IO 逻辑
备份 RAM
BKUP IO
VSS
VSS
VDDA
模拟域
REF_BUF
ADC, DAC
VREF+
VREF-
OPAMP，比较器
VREF+
VREF-
VSSA
MSv40895V5

通过配置稳压器，V_{CORE} 内核域和外部电源可支持*图 14* 所示的电源配置。

图 14. 系统电源配置



不同电源配置通过 *PWR 控制寄存器 3 (PWR_CR3)* 寄存器中的 LDOEN 和 BYPASS 位控制，如*表 26* 所示。

表 26. 电源配置控制

电源配置	LDOEN	BYPASS	说明
默认配置	1	0	– V_{CORE} 电源域由 LDO 根据 VOS 提供。
LDO 供电	1	0	– V_{CORE} 电源域由 LDO 根据 VOS 提供。 – LDO 电源模式（主模式、LP 模式和关闭模式）将遵循系统低功耗模式。
LDO 旁路	0	1	– V_{CORE} 通过外部电源提供。 – LDO 旁路，电压监控处于激活状态。
非法	0	0	– 非法组合，保持默认配置。（将忽略写入数据）。
	1	1	

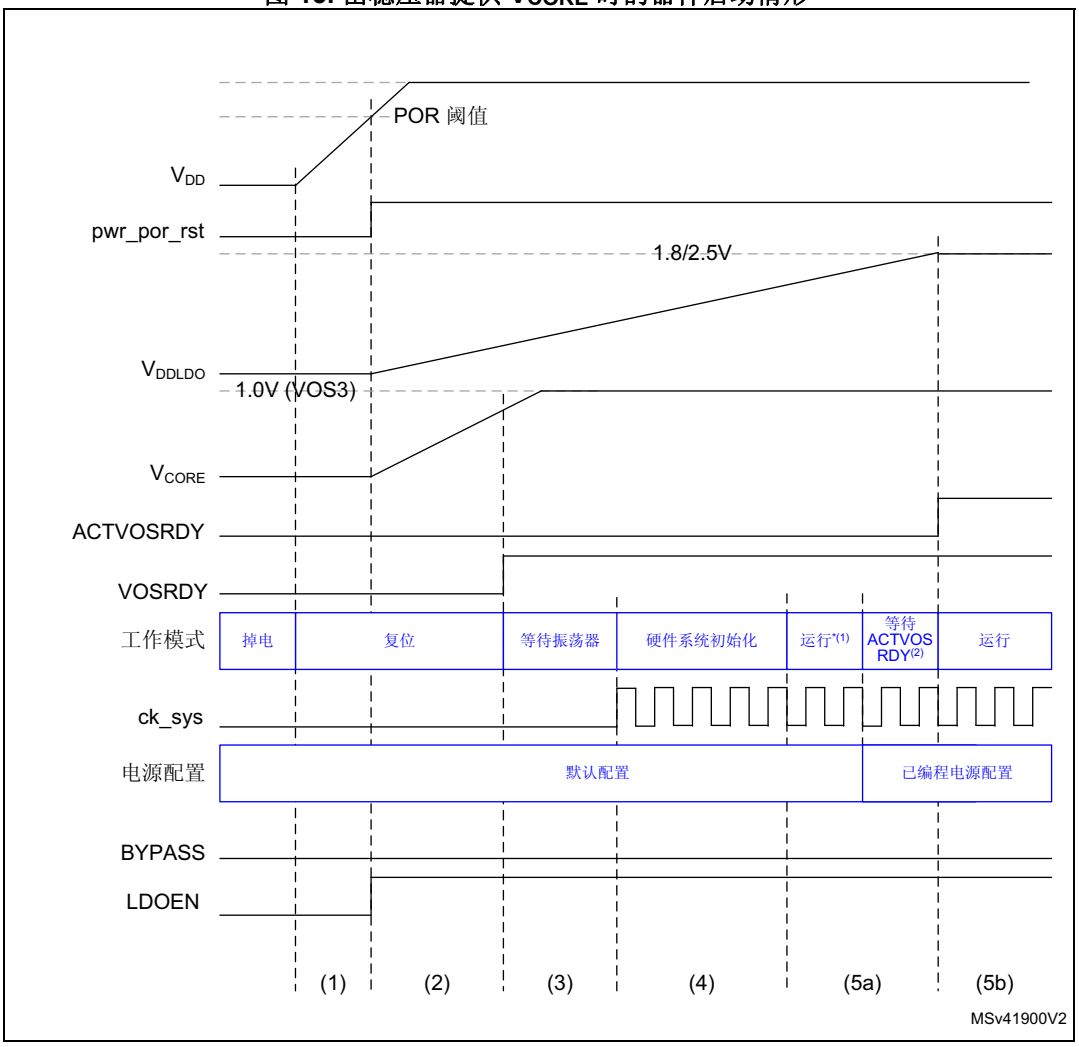
6.4.1 系统电源启动

不同电源配置下从上电状态开始的系统启动时序如下（对于 LDO 电源，请参见*图 15*）：

1. 系统上电时，POR 会监视 V_{DD} 电源。在默认电源配置下，一旦 V_{DD} 高于 POR 阈值，稳压器就使能：
   – 根据 *PWR D3 域控制寄存器 (PWR_D3CR)* 中配置的 VOS3 级别，电压转换器输出电平设置为 1.0 V。
2. 只要 V_{CORE} 不正确，系统就保持在复位模式。
3. 一旦 V_{CORE} 正确，系统便会退出复位模式，HSI 时钟将会使能。
4. 时钟稳定后，系统将进行初始化：Flash 和选项字节将进行装载，CPU 将在有限运行 (Run*) 模式下启动。
5. 之后，软件应初始化系统，包括 *PWR 控制寄存器 3 (PWR_CR3)* 中的电源配置编程。电源配置一旦配置完成，应检查 *PWR 控制状态寄存器 1 (PWR_CSR1)* 中的 ACTVOSRDY 位以确保电压级别有效：
   a) 只要 ACTVOSRDY 指示电压级别无效，系统便会处于 Run* 模式、不允许对 RAM 进行写访问并且不得更改 VOS。
   b) ACTVOSRDY 指示电压级别有效后，系统便会处于正常运行模式、允许对 RAM 进行写访问并且可更改 VOS。

软件必须在 *PWR 控制寄存器 3 (PWR_CR3)* 中对电源配置进行编程。

图 15. 由稳压器提供 V_{CORE} 时的器件启动情形



1. 在 Run* 模式下，不允许对 RAM 执行写操作。
2. 只有当 ACTVOSRDY 有效时，才允许对 RAM 执行写操作，才可更改 VOS。

从待机模式退出时，由于 *PWR 控制寄存器 3 (PWR_CR3)* 寄存器的内容保持不变，因此系统会识别电源配置。不过，软件仍应先等待 *PWR 控制状态寄存器 1 (PWR_CSR1)* 中的 ACTVOSRDY 位置 1（表示 V_{CORE} 电压级别有效），再写入 RAM 或更改 VOS。

6.4.2 内核域

V_{CORE} 内核域电源可通过稳压器或外部电源 (V_{CAP}) 提供。V_{CORE} 为除备份域和待机电路以外的所有数字电路供电。V_{CORE} 域分为 3 个部分：

- D1 域中的 CPU (Cortex-M7)、Flash 和外设。
- D2 域的外设。
- D3 域的系统控制、I/O 逻辑和低功耗外设。

当发生系统复位时，稳压器使能并为 V_{CORE} 供电。这允许系统以任何电源配置启动（请参见*图 14*）。

系统复位后，软件应先在 *PWR 控制寄存器 3 (PWR_CR3)* 寄存器中配置所使用的电源配置，再更改 VOS（在 *PWR D3 域控制寄存器 (PWR_D3CR)* 中）或 RCC ck_sys 频率。可控制不同的系统电源配置，如*表 26* 所示。

调压器

嵌入式稳压器 (LDO) 需要在 V_{CAP} 引脚上连接外部电容。

稳压器提供三种不同的工作模式：主 (MR) 模式、低功耗模式 (LP) 或关闭模式。这些模式将根据系统工作模式（运行、停止和待机）进行使用。

- 运行模式

  LDO 稳压器处于主模式并为 V_{CORE} 域（内核、存储器和数字外设）提供全功率。稳压器输出电源可通过软件调节为不同电压级别（VOS1、VOS2 和 VOS3），这些级别通过 *PWR D3 域控制寄存器 (PWR_D3CR)* 中的 VOS 位配置。系统运行在最高工作频率时，VOS 电压调节特性可使功耗得到优化。默认情况下，系统复位后选择 VOS3。可实时更改 VOS 来适应所需的系统性能。

- 停止模式

  稳压器为 V_{CORE} 供电以保存寄存器和内部存储器的内容。

  稳压器可保持主模式，此时可从停止模式快速退出；也可以设置为 LP 模式以获得较低的 V_{CORE} 电源电压级别，此时会增大从停止模式退出的延时。稳压器模式通过 *PWR 控制寄存器 1 (PWR_CR1)* 中的 SVOS 和 LPDS 位选择。如果选择了 SVOS3 电压调节，则可以选择主模式或 LP 模式；如果选择 SVOS4 和 SVOS5 调节，则只能选择 LP 模式。由于 SVOS4 和 SVOS5 调节的电压级别低，因此可进一步降低停止模式的功耗。

- 待机模式

  稳压器关闭且 V_{CORE} 域掉电。除待机电路和备份域外，寄存器和存储器的内容都将丢失。

注： *有关详细信息，请参见数据手册的调压器部分。*

6.4.3 PWR 外部电源

当 V_{CORE} 通过外部电源供电时，可根据系统工作模式（运行、停止或待机）使用不同的工作模式：

- 在运行模式下

  外部电源为 V_{CORE} 域（内核、存储器和数字外设）提供全功率。外部电源输出电压可调节为不同的电压级别（VOS1、VOS2 和 VOS3）。应当在 PWR_D3CR 寄存器的 VOS 位反映外部施加的电压级别。只有当外部施加的电压级别与 VOS 设置匹配时才应对 RAM 进行写访问。

- 在停止模式下

  外部电源为 V_{CORE} 域提供电源，以保存寄存器和内部存储器的内容。在停止模式下，稳压器可选择较低的 V_{CORE} 电源电压来降低功耗。

- 在待机模式下

  外部电源将关闭，V_{CORE} 域断电。除待机电路和备份域外，寄存器和存储器的内容都将丢失。退出待机模式时，外部电源将打开。

6.4.4 备份域

要在 V_{DD} 关断后保留备份域（RTC、备份寄存器和备份 RAM）的内容，需将 V_{BAT} 引脚连接到通过电池或其它电源供电的可选备用电压。

V_{BAT} 的开关由复位模块中内置的掉电复位电路进行控制，复位模块用于监控 V_{DD} 电源。

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

当 V_{DD} 电源存在时，备份域通过 V_{DD} 供电。这可以延长 V_{BAT} 电源电池的使用寿命。

如果没有使用任何外部电池，建议将该 V_{BAT} 通过 100 nF 外部陶瓷电容从外部连接到 V_{DD} 上。

当 V_{DD} 电源存在且其值高于 PDR 阈值时，备份域通过 V_{DD} 供电并提供以下功能：

- PC14 和 PC15 可用作 GPIO 或者 LSE 引脚。
- PC13 可用作 GPIO 或 RTC_AF1 或 RTC_TAMP1 引脚，前提是它们已通过 RTC 进行了相应配置。
- PI8/RTC_TAMP2 和 PC1/RTC_TAMP3 可通过 RTC 配置为入侵引脚。

注： *由于该开关的灌电流能力有限，因此使用 PC1、PC13 到 PC15 以及 PI8 GPIO 时存在以下限制：每次只能有一个 I/O 用作输出，最大负载为 30 pF 时速率不得超过 2 MHz 并且这些 I/O 不能用作电流源（例如，用于驱动 LED）。*

在 V_{BAT} 模式下，当 V_{DD} 电源不在且 V_{BAT} 上存在电源时，备份域通过 V_{BAT} 供电并提供以下功能：

- PC14 和 PC15 只可用作 LSE 引脚。
- PC13 可用作 RTC_AF1 或 RTC_TAMP1 引脚，前提是它们已通过 RTC 进行了相应配置。
- PI8/RTC_TAMP2 和 PC1/RTC_TAMP3 可通过 RTC 配置为入侵引脚。

访问备份域

复位后，备份域（RTC 寄存器和 RTC 备份寄存器）将受到保护，以防止意外的写访问。要访问备份域，需将 *PWR 控制寄存器 1 (PWR_CR1)* 中的 DBP 位置 1。

有关 RTC 和备份 RAM 访问的更多详细信息，请参见*第 8 节：复位和时钟控制 (RCC)*。

备份 RAM

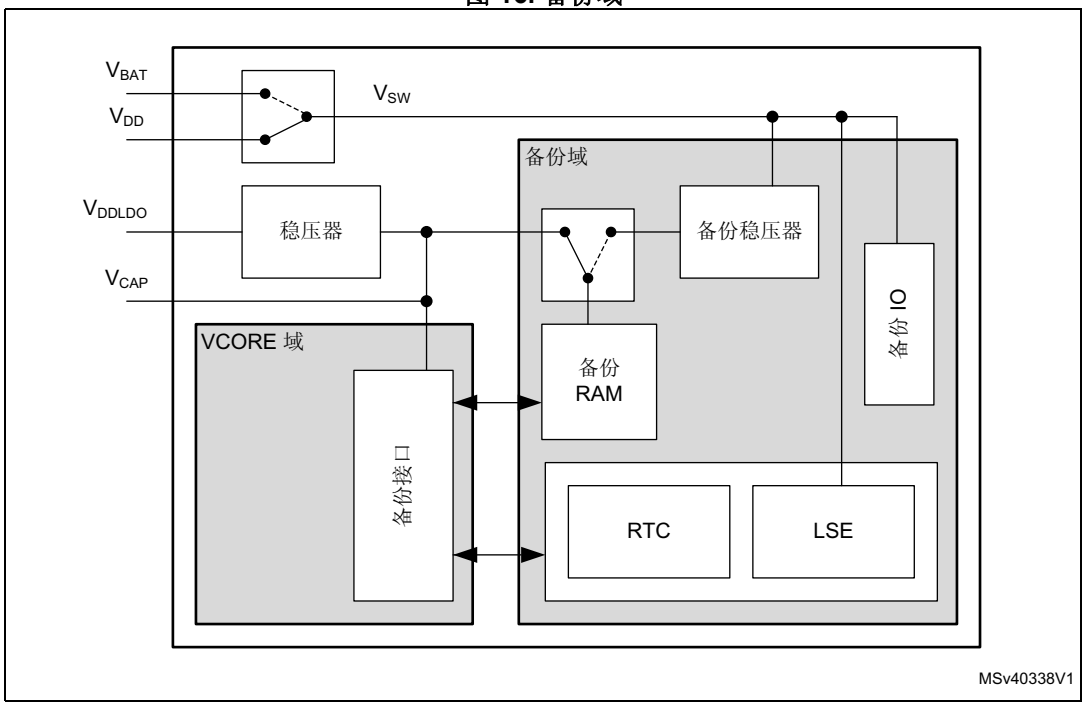
备份域包括可在 32 位、16 位 或 8 位模式下寻址的 4 KB 备份 RAM。备份 RAM 通过备份域中的备份稳压器供电。当备份稳压器通过 *PWR 控制寄存器 2 (PWR_CR2)* 中的 BREN 位使能时，即使处于待机和/或 V_{BAT} 模式，也会保存备份 RAM 的内容（如果 V_{BAT} 始终存在，则它可以视作内部 EEPROM。）

备份稳压器既可以处于打开状态，也可以处于关闭状态，具体取决于应用在待机模式或 V_{BAT} 模式下是否需要备份 RAM 功能。

备份 RAM 不会被入侵事件批量擦除，而是对其进行了读保护，以防止用户对加密私钥等机密数据进行访问。要在入侵事件后再次访问备份 RAM，首先需要擦除内存区域。备份 RAM 可通过以下方式擦除：

- 当要求保护级别从级别 1 更改为级别 0 时（请参见 Flash 编程手册中的读保护 (RDP) 说明），通过 Flash 接口进行擦除。
- 在入侵事件后，通过对备份 RAM 执行虚写操作向其写入数据 0，以此进行擦除。

图 16. 备份域



6.4.5 V_{BAT} 电池充电

当 V_{DD} 存在时，可通过内部电阻为 V_{BAT} 连接的外部电池充电。

可通过 5 kΩ 电阻或 1.5 kΩ 电阻为 V_{BAT} 充电，具体取决于 *PWR 控制寄存器 3 (PWR_CR3)* 中 VBRS 位的值。

通过将 *PWR 控制寄存器 3 (PWR_CR3)* 中的 VBE 位置 1 来使能电池充电。在VBAT模式下自动禁用。

6.4.6 模拟电源

独立的 V_{DDA} 模拟电源

模拟电源域通过专用的 V_{DDA} 和 V_{SSA} 引脚供电，这些引脚可为电源滤除和屏蔽 PCB 上的噪声，从而提高 ADC 和 DAC 的转换效率：

- 模拟电源电压从独立的 V_{DDA} 引脚输入。
- V_{SSA} 引脚提供了独立的电源接地连接。

模拟参考电压 V_{REF+}/V_{REF}

为获得精度更高的低电压信号，ADC 和 DAC 还通过 V_{REF+} 引脚提供独立的参考电压。用户可在 V_{REF+} 上连接独立的外部参考电压。

V_{REF+} 可控制最高电压（通过满量程值表示），而低参考电压 (V_{REF-}) 连接到 V_{SSA}。

当通过 VREFBUF 控制和状态寄存器中的 ENVR 位使能时（请参见*第 27 节：电压参考缓冲器 (VREFBUF)*），V_{REF+} 由内部电压参考缓冲器提供。内部电压参考缓冲器还可通过 V_{REF+}/V_{REF-} 引脚为外部元件提供参考电压。

当内部电压参考缓冲器通过 ENVR 禁止时，V_{REF+} 由独立的外部参考电源提供。

6.4.7 USB 稳压器

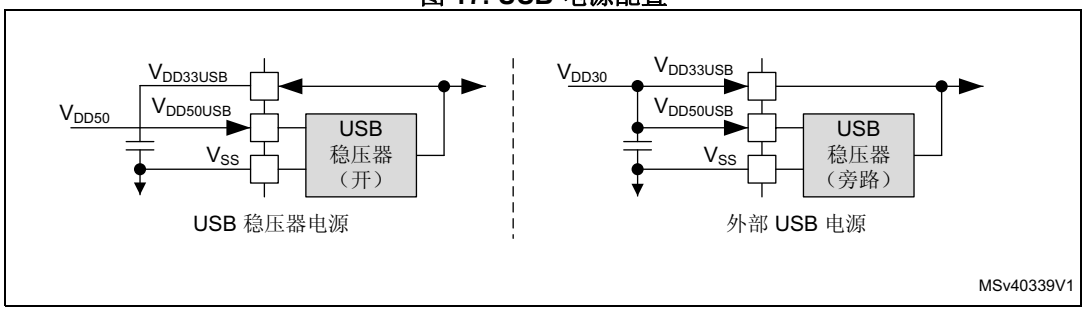
USB 收发器通过专用的 $V_{DD33USB}$ 电源供电，该电源可通过集成 USB 稳压器或外部 USB 电源提供。

当通过 *PWR 控制寄存器 3 (PWR_CR3)* 中的 USBREGEN 位使能时，$V_{DD33USB}$ 由 USB 稳压器提供。使用 $V_{DD33USB}$ 前，请通过监控 *PWR 控制寄存器 3 (PWR_CR3)* 中的 USB33RDY 位检查其是否可用。$V_{DD33USB}$ 电源电压检测器应通过 PWR_CR3 寄存器中的 USB33DEN 位使能。

当 USB 稳压器通过 USBREGEN 位禁止时，$V_{DD33USB}$ 可由外部电源提供。在这种情况下，$V_{DD33USB}$ 和 $V_{DD50USB}$ 应连接在一起。在使用 USB 收发器之前，$V_{DD33USB}$ 电源电压检测器必须通过 PWR_CR3 寄存器中的 USB33DEN 位使能。

有关 USB 稳压器的更多信息，请参见*第 57 节：USB on-the-go 高速(OTG_HS)*。

图 17. USB 电源配置



6.5 电源监控

可对以下电源执行电源电压监控：

- V_{DD}，通过 POR/PDR（请参见*第 6.5.1 节*）、BOR（请参见*第 6.5.2 节*）和 PVD 监控器（请参见*第 6.5.3 节*）监控
- V_{DDA}，通过 AVD 监控器监控（请参见*第 6.5.4 节*）
- V_{BAT}，通过 VBAT 阈值监控（请参见*第 6.5.5 节*）
- V_{SW}，通过 rst_vsw 监控，只要电压级别不正常，V_{SW} 域便一直保持复位模式
- V_{BKP}，通过 *PWR 控制寄存器 2 (PWR_CR2)* 中的 BRRDY 位监控
- $V_{DD33USB}$，通过 *PWR 控制寄存器 3 (PWR_CR3)* 中的 USBRDY 位监控

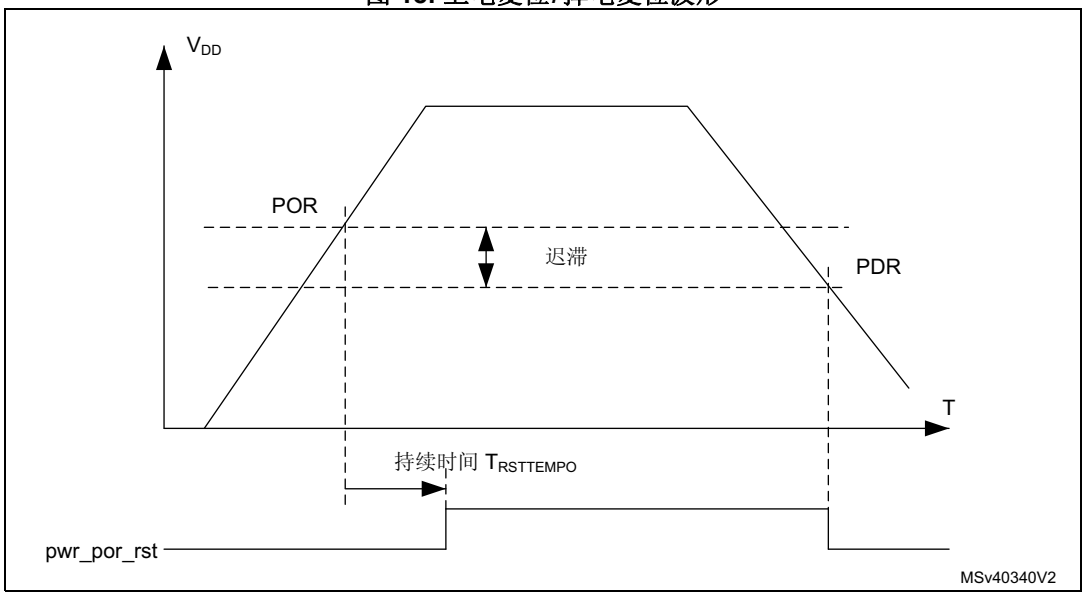
6.5.1 上电复位(POR)/掉电复位(PDR)

系统内部集成有 POR/PDR 电路，可确保正常的启动操作。

当 V_{DD} 低于指定 V_{POR} 阈值时，系统无需外部复位电路便会保持复位模式。一旦 V_{DD} 电源电压高于 V_{POR} 阈值，系统便会退出复位状态（请参见*图 18*）。有关上电/掉电复位阈值的相关详细信息，请参见数据手册的电气特性部分。

PDR 可通过器件 PDR_ON 输入引脚使能/禁止。

图 18. 上电复位/掉电复位波形



1. 有关阈值和迟滞值的信息，请参见数据手册。

6.5.2 欠压复位 (BOR)

上电期间，欠压复位 (BOR) 将使系统保持复位状态，直到 V_{DD} 电源电压达到指定的 V_{BOR} 阈值。

V_{BOR} 阈值通过系统选项字节进行配置。默认情况下，BOR 关闭。可选择以下可编程 V_{BOR} 阈值：

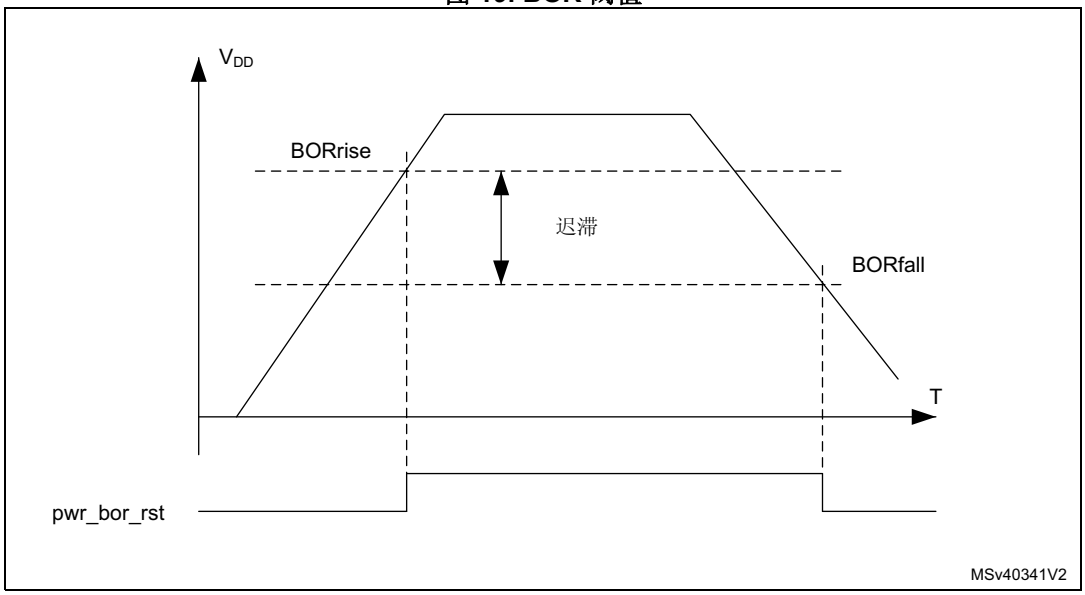
- BOR 关闭 (V_{BOR0})
- BOR 级别 1 (VBOR1)
- BOR 级别 2 (VBOR2)
- BOR 级别 3 (VBOR3)

有关欠压复位阈值的相关详细信息，请参见产品数据手册的“电气特性”部分。

BOR 使能且 V_{DD} 电源电压降至所选 V_{BOR} 阈值以下时，将产生系统复位。

通过对系统选项字节进行编程可以禁止 BOR。要禁止 BOR 功能，V_{DD} 必须高于 V_{BOR0}，以启动系统选项字节编程顺序。PDR 将在随后监视掉电过程（参见*第 6.5.1 节*）。

图 19. BOR 阈值



1. 有关阈值和迟滞值的信息，请参见数据手册。

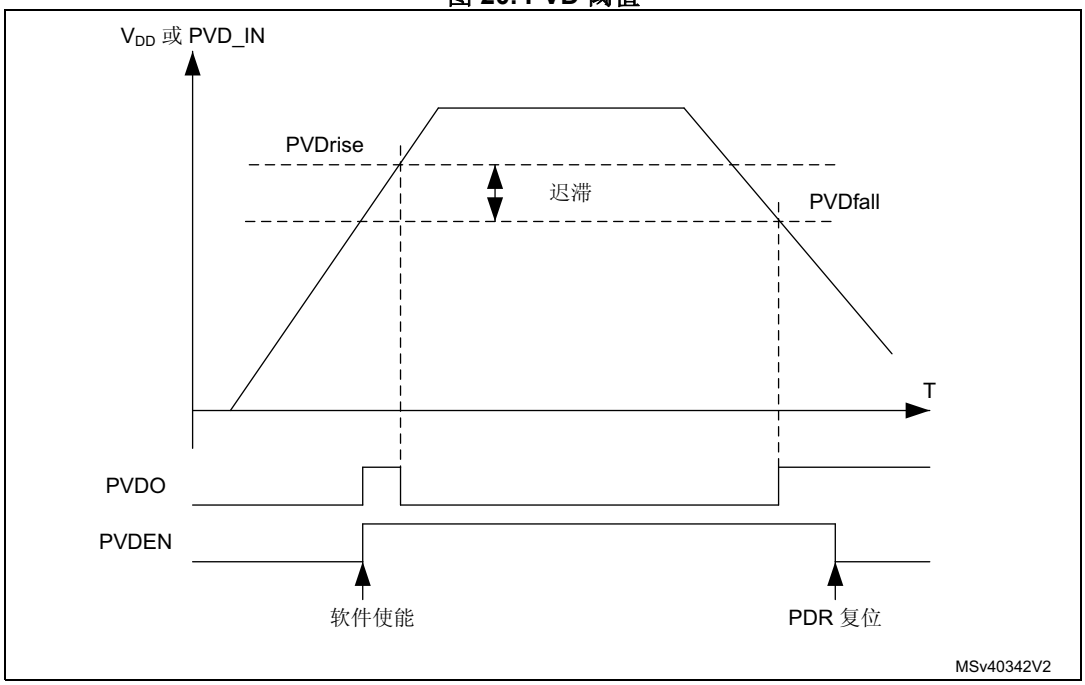
6.5.3 可编程电压检测器（PVD）

可以使用 PVD 监视 V_{DD} 电源，方法是将其与 *PWR 控制寄存器 1 (PWR_CR1)* 中的 PLS[2:0] 位所选的阈值进行比较。也可以使用 PVD 来监视 PVD_IN 引脚上的电压级别。在这种情况下，会将 PVD_IN 电压与内部 V_{REFINT} 级别进行比较。

通过将 *PWR 控制寄存器 1 (PWR_CR1)* 中的 PVDE 位置 1 来使能 PVD。

PWR 控制状态寄存器 1 (PWR_CSR1) 中提供了 PVDO 标志，用于指示 V_{DD} 或 PVD_IN 电压是大于还是小于 PVD 阈值。该事件内部连接到 EXTI，如果通过 EXTI 寄存器使能，则可以产生中断。当 V_{DD} 或 PVD_IN 电压降至 PVD 阈值以下以及/或者当 V_{DD} 或 PVD_IN 电压升至 PVD 阈值以上时，可以产生 pwr_pvd_wkup 输出中断，具体取决于 EXTI 上升沿/下降沿配置。该功能的用处之一就是可以在中断服务程序中执行紧急关闭系统的任务。

图 20. PVD 阈值



1. 有关阈值和迟滞值的信息，请参见数据手册。

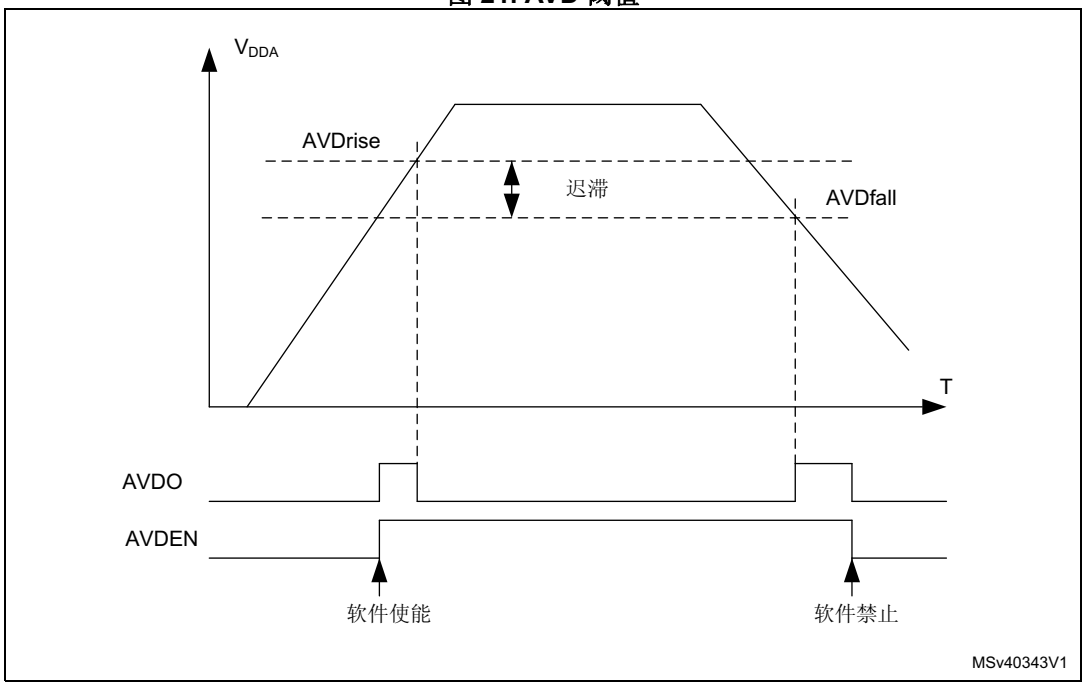
6.5.4 模拟电压检测器 (AVD)

可以使用 AVD 监视 V_{DDA} 电源，方法是将其与 *PWR 控制寄存器 1 (PWR_CR1)* 中的 ALS[1:0] 位所选的阈值进行比较。

通过将 *PWR 控制寄存器 1 (PWR_CR1)* 中的 AVDEN 位置 1 来使能 AVD。

PWR 控制状态寄存器 1 (PWR_CSR1) 中提供了 AVDO 标志，用于指示 V_{DDA} 是大于还是小于 AVD 阈值。该事件内部连接到 EXTI，如果通过 EXTI 寄存器使能，则可以产生中断。当 V_{DDA} 降至 AVD 阈值以下以及/或者当 V_{DDA} 升至 AVD 阈值以上时，可以产生 pwr_avd_wkup 中断，具体取决于 EXTI 上升沿/下降沿配置。例如，服务程序可以指示 V_{DDA} 电源何时降至最小值以下。

图 21. AVD 阈值



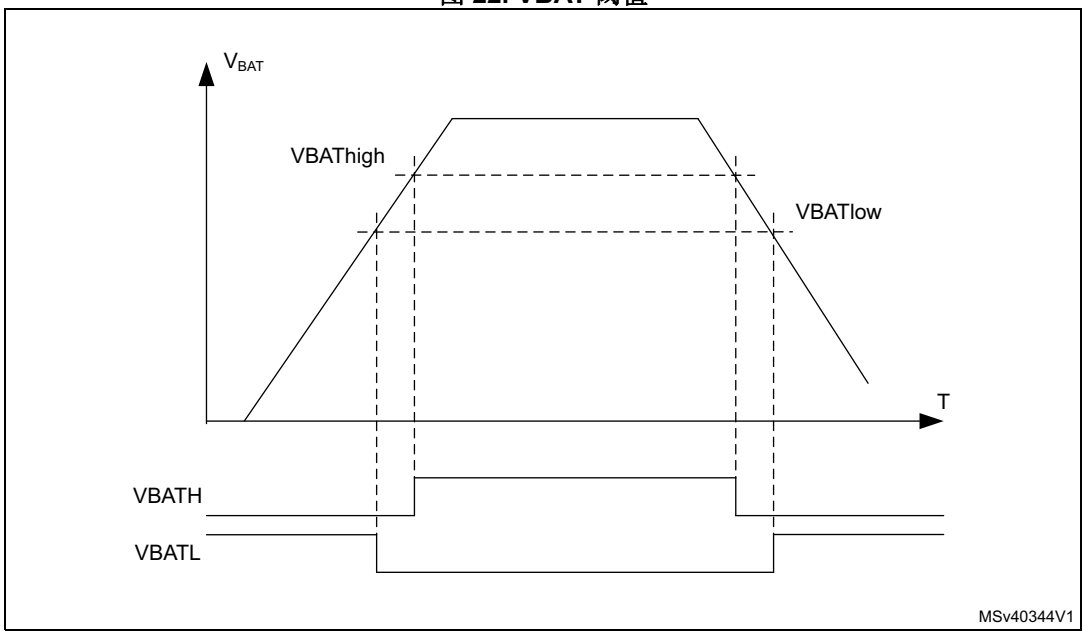
1. 有关阈值和迟滞值的信息，请参见数据手册。

6.5.5 电池电压阈值

可以监视 V_{BAT} 电池电压，方法是将其与以下两个阈值进行比较：$V_{BAThigh}$ 和 V_{BATlow}。*PWR 控制寄存器 2 (PWR_CR2)* 中的 VBATH 和 VBATL 标志用于指示 V_{BAT} 是大于还是小于阈值。可通过 *PWR 控制寄存器 2 (PWR_CR2)* 中的 MONEN 位使能/禁止 V_{BAT} 监视。当使能时，电池电压阈值会增加功耗。例如，电压级别可以用于触发程序以执行紧急保存任务。

RTC 入侵信号上提供了 VBATH 和 VBATL 唤醒中断（请参见*第 46 节：实时时钟 (RTC)*）。

图 22. VBAT 阈值



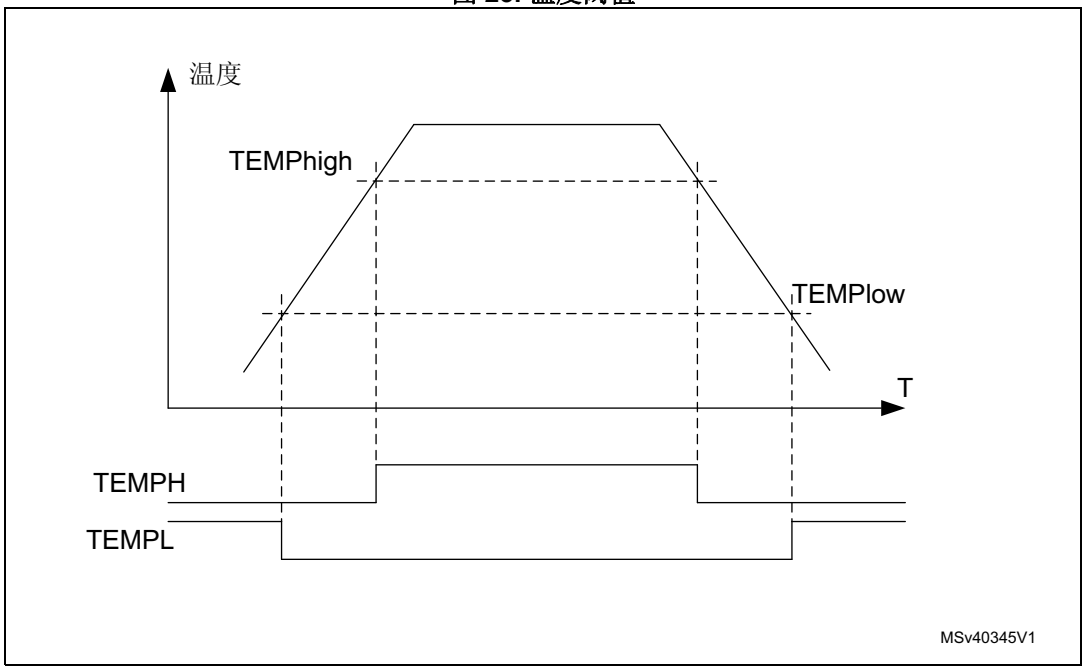
1. 有关阈值和迟滞值的信息，请参见数据手册。

6.5.6 温度阈值

可以监视结温，方法是将其与 $TEMP_{high}$ 和 $TEMP_{low}$ 这两个阈值进行比较。*PWR 控制寄存器 2 (PWR_CR2)* 中的 TEMPH 和 TEMPL 标志用于指示器件温度是大于还是小于阈值。可通过 *PWR 控制寄存器 2 (PWR_CR2)* 中的 MONEN 位使能/禁止温度监视。当使能时，温度阈值会增加功耗。例如，温度级别可以用于触发程序以执行温度控制任务。

RTC 入侵信号上提供了 TEMPH 和 TEMPL 唤醒中断（请参见*第 46 节：实时时钟 (RTC)*）。

图 23. 温度阈值



1. 有关阈值和迟滞值的信息，请参见数据手册。

6.6 电源管理

电源管理模块根据系统工作模式来控制 V_{CORE} 电源（请参见*第 6.6.1 节*）。

V_{CORE} 域分为以下电源域：

- D1 域中的一些外设和 Cortex® M7 CPU。
- D2 域中的大部分外设。
- D3 域中的一些外设和系统控制。

D1、D2 和系统 D3 电源域可在以下其中一种工作模式下工作：

- DRun/Run/Run*（电源打开，时钟打开）
- DStop/Stop（电源打开，时钟关闭）
- DStandby/Standby（电源关闭，时钟关闭）

D1 域和 D2 域的工作模式是相互独立的。而系统 D3 域的工作模式取决于 D1 和 D2 域的模式：

- 要使系统 D3 域在停止模式下工作，D1 和 D2 域必须处于 DStop 或 DStandby 模式。
- 要使系统 D3 域在待机模式下工作，D1 和 D2 域必须处于 DStandby 模式。

D1、D2 和系统 D3 域均通过一个稳压器以共同的 V_{CORE} 电压供电。V_{CORE} 电源电压取决于系统工作模式（运行、停止和待机）。当 D1 域和/或 D2 域电源处于 DStandby 模式时可单独掉电。

以下电压调节功能可根据所需的系统性能控制电源（请参见*第 6.6.2 节：电压调节*）：

- 要获得给定的系统性能，应根据系统时钟频率设置相应的电压调节。为此，需将 VOS 位配置为运行模式电压调节。
- 要在功耗和退出停止模式延时之间达到最佳平衡，需将 SVOS 位配置为停止模式电压调节。

6.6.1 工作模式

提供了多种系统工作模式，从而可以根据所需性能（比如当 CPU 不需要执行代码并且正在等待外部事件时）对系统进行调整。由用户选择具体的工作模式，以在低功耗、短启动时间和可用唤醒源之间寻求最佳平衡。

操作模式允许通过控制时钟去控制不同的系统模块并为其供电。系统工作模式通过 CPU 子系统、D2 域和系统 D3 自动唤醒来驱动。CPU 子系统可包含多个域，具体取决于其外设分配（请参见*第 8.5.11 节：外设时钟门控*）。

以下工作模式可用于不同的系统模块（请参见*表 27*）：

- CPU 子系统模式：
  - **CRun**

    CPU 和 CPU 子系统外设的时钟通过 RCC PERxEN 设定并运行。
  - **CSleep**：

    CPU 时钟停止，CPU 子系统分配的外设时钟视 RCC PERxLEN 而定。
  - **CStop**：

    CPU 和 CPU 子系统外设时钟停止。
- D1 域模式：
  - DRUN

    域总线矩阵时钟运行。CPU 子系统在 CRun 或 CSleep 模式下工作。
  - DStop

    域总线矩阵时钟停止。

    CPU 子系统在 CStop 模式下工作，PDDS_D1[1] 位选择 DStop 模式。
  - DStandby

    域掉电。

    CPU 子系统在 CStop 模式下工作，PDDS_D1 位选择 DStandby 模式。
- D2 域模式：
  - DRUN

    域总线矩阵时钟运行。

    CPU 子系统在 D2 域中分配了一个外设且 CPU 子系统在 CRun 或 CSleep 模式下工作。

1. PDDS_Dn 位属于 *PWR CPU 控制寄存器 (PWR_CPUCR)*。

– DStop

域总线矩阵时钟停止。

CPU 子系统未在 D2 域中分配外设且 PDDS_D2(1) 位选择 DStop 模式，

或

CPU 子系统在 D2 域中分配了一个外设，CPU 子系统在 CStop 模式下工作且 PDDS_D2 位选择 DStop 模式。

– DStandby

域掉电。

CPU 子系统未在 D2 域中分配外设且 PDDS_D2 位选择 DStandby 模式，

或

CPU 子系统在 D2 域中分配了一个外设，CPU 子系统在 CStop 模式下工作且 PDDS_D2 位选择 DStandby 模式。

- 系统/D3 域模式
  - **Run/Run***

    系统时钟和 D3 域总线矩阵时钟均在运行：
    - CPU 子系统处于 CRun 或 CSleep 模式
    或
    - 唤醒信号有效。（比如，系统 D3 自动模式）

    POR 复位和从待机模式唤醒后将进入 Run* 模式。在 Run* 模式下，性能受限且系统电源配置将在 *PWR 控制寄存器 3 (PWR_CR3)* 中编程。只有当 *PWR 控制状态寄存器 1 (PWR_CSR1)* 中的 ACTVOSRDY 位置 1 时，系统才会进入运行模式。
  - **Stop**

    系统时钟和 D3 域总线矩阵时钟停止：
    - CPU 子系统处于 CStop 模式。
    且
    - 所有唤醒信号均无效。
    且
    - 任何域中至少有一个 PDDS_Dn(1) 位选择停止模式。
  - **待机**

    系统掉电：
    - CPU 子系统处于 CStop 模式。
    且
    - 所有唤醒信号均无效。
    且
    - 所有域的所有 PDDS_Dn(1) 位均选择待机模式。

在运行模式下，可通过下列方法之一降低功耗：

- 通过减慢系统时钟速率降低系统性能，通过 VOS 电压调节位减小 V_{CORE} 电源电压。
- 不使用 APBx 和 AHBx 外设时，通过 PERxEN 位将对应的外设时钟关闭。

表 27. 低功耗模式汇总

系统	域	CPU	进入	唤醒	系统时钟	系统时钟	域总线矩阵时钟	外设时钟	CPU 时钟	调压器	域电源
运行	DRUN(1)	CRun	-	-	开启	开启	开启	开启	开启	开启	开启
		CSleep	WFI 或从 ISR 或 WFE 返回	任何中断或事件				开启/关闭(2)	关闭		
	DStop(3)	CStop	SLEEPDEEP 位 + WFI 或者从 ISR 或 WFE 返回	任何 EXTI 中断或事件			关闭	开启/关闭(4)			
	DStandby(3)										关闭
Stop(5)	DStop(3)		SLEEPDEEP 位 + WFI 或者从 ISR 或 WFE 返回或者清除唤醒源(6)		开启/关闭(7)	关闭		关闭			开启
	DStandby(3)										关闭
待机(8)	DStandby(3)		所有 PDDS_Dn 位 + SLEEPDEEP 位 + WFI 或者从 ISR 或 WFE 返回或者清除唤醒源(6)	WKUP 引脚上升沿或下降沿、RTC 闹钟（闹钟 A 或闹钟 B）、RTC 唤醒事件、RTC 入侵事件、RTC 时间戳事件、NRST 引脚外部复位、IWDG 复位	关闭					关闭	

1. CPU 子系统在 D2 域中分配了一个外设并且在 CRun 或 CSleep 模式下工作。
2. 具有 PERxLPEN 位的 CPU 子系统外设将相应地工作。
3. 如果 CPU 子系统在 D2 域中分配了一个外设，则它必须在 CStop 模式下工作。
4. 具有 PERxAMEN 位的 CPU 子系统外设将相应地工作。
5. 所有域都需要在 DStop 或 DStandby 模式下工作。
6. 当 CPU 处于 CStop 模式且 D3 域处于自动模式时，将清除最后一个 EXTI 唤醒源。
7. 当使用系统时钟 HSI 或 CSI 时，状态通过 HSIKERON 和 CSIKERON 控制，否则系统时钟关闭。
8. 所有域均处于 DStandby 模式。

6.6.2 电压调节

D1、D2 和 D3 域由单个稳压器供电，该稳压器通过以下功能支持电压调节：

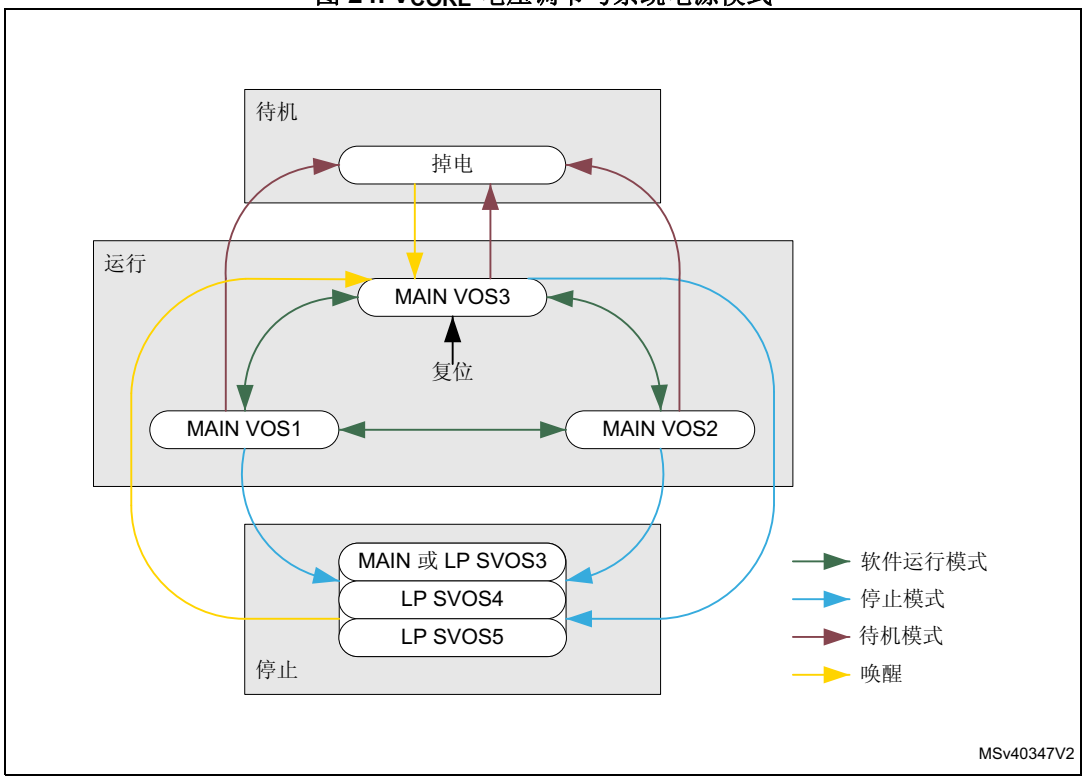
- 运行模式电压调节
  - VOS1：电压调节 1
  - VOS2：电压调节 2
  - VOS3：电压调节 3
- 停止模式电压调节
  - SVOS3：电压调节 3
  - LP-SVOS4：电压调节 4
  - LP-SVOS5：电压调节 5

有关电压调节值的详细信息，请参见产品数据手册。

复位后，系统以最低的运行模式电压调节 (VOS3) 启动。之后，可通过软件根据所需的系统性能编程 *PWR D3 域控制寄存器 (PWR_D3CR)* 中的 VOS 位，实时更改电压调节。当从停止模式或待机模式退出时，运行模式电压调节将复位为默认的 VOS3 值。

进入停止模式前，软件可在 *PWR 控制寄存器 1 (PWR_CR1)* 中预先选择 SVOS 级别。SVOS4 和 SVOS5 的停止模式电压调节还会将稳压器设置为低功耗 (LP) 模式，以进一步降低功耗。当预先选择 SVOS3 时，可通过 LPDS 寄存器位选择使用稳压器低功耗模式 (LP)。

图 24. V_{CORE} 电压调节与系统电源模式



6.6.3 电源控制模式

电源控制模块针对系统运行、停止和待机模式处理 V_{CORE} 电源。

系统工作模式取决于 CPU 子系统模式（CRun、CSleep 和 CStop）、域模式（DRun、DStop 和 DStandby）以及系统 D3 自动唤醒：

- 在运行模式下，V_{CORE} 通过 VOS 电压调节定义。
  CPU 子系统处于 CRun 或 CSleep 模式或者 EXTI 唤醒有效。
- 在停止模式下，V_{CORE} 通过 SVOS 电压调节定义。
  CPU 子系统处于 CStop 模式且所有 EXTI 唤醒均无效。D1 域和 D2 域处于 DStop 或 DStandby 模式。
- 在待机模式下，V_{CORE} 电源关闭。
  CPU 子系统处于 CStop 模式且所有 EXTI 唤醒均无效。D1 域和 D2 域均处于 DStandby 模式。

当在相应域中分配外设时，域工作模式取决于 CPU 子系统。在 DStop 和 DStandby 之间的域模式切换可以通过 *PWR CPU 控制寄存器 (PWR_CPUCR)* 中的域专用 PDDS_Dn 位配置。CPU 还可选择使一个域保持 DStop 模式，或使一个域进入 DStandby 模式。

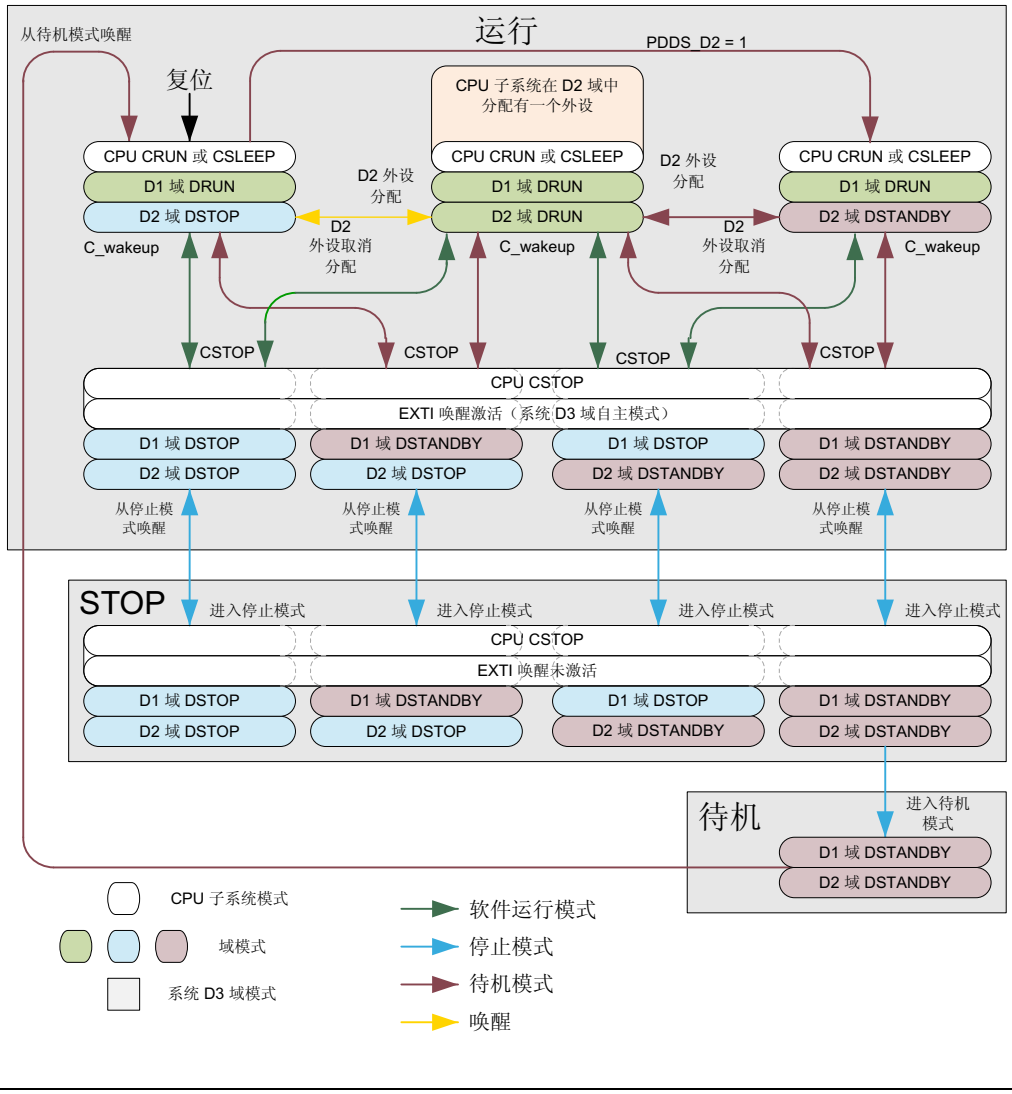
如果某个域处于 DStandby 模式，则相应电源将关闭。

所有域均可通过 *PWR CPU 控制寄存器 (PWR_CPUCR)* 中的 PDDS_Dn 位配置为相应的系统模式（停止或待机）。只有当所有域的所有 PDDS_Dn 位都允许时，系统才会进入待机状态。

表 28. PDDS_Dn 低功耗模式控制

PWR_CPUCR			D1 模式	D2 模式	D3 模式
PDDS_D1	PDDS_D2	PDDS_D3			
0	x	x	DStop	任意	Run 或 Stop
1			DStandby	任意	任意
x	0		任意	DStop	Run 或 Stop
	1		任意	DStandby	任意
至少一个 = 0			DStop 或 DStandby	DStop 或 DStandby	Stop
1	1	1	DStandby	DStandby	待机

图 25. 电源控制模式详细状态图



系统复位后，CPU 处于 CRun 模式。

电源控制状态转换由以下事件启动：

- CPU 进入 CStop 模式（运行模式下的状态转换标记为绿色和红色）
  - 绿色转换：CPU 从 CSleep 唤醒。
  - 红色转换：CPU 通过域复位唤醒。SBF_Dn 位置 1。
- 将一个外设分配到域中或从域中移除（运行模式下的状态转换标记为橙色和红色）
  - 橙色转换：域从 DStop 唤醒。
  - 红色转换：域从 DStandby 唤醒 SBF_Dn 位置 1。
- 系统进入或退出停止模式（状态转换标记为蓝色）
  - 蓝色转换：系统从停止模式唤醒。STOPF 置 1。
- 系统进入或退出待机模式（停止和待机模式下的状态转换标记为红色）。
  - 退出待机模式时，SBF 置 1。

当某个域退出 DStandby 时，该域的外设复位，同时该域的 SBF_Dn 位置 1（导致域复位的状态转换标记为红色）。

表 29 所示的标志用于指示域/系统从哪个模式退出。CPU 具有一组标志，这组标志可从 *PWR CPU 控制寄存器 (PWR_CPUCR)* 读取。

表 29. 低功耗退出模式标志

系统模式	D1 域模式	D2 域模式	SBF_D1	SBF_D2	SBF	STOPF	备注
运行	DRun 或 DStop	DRun 或 DStop	0	0	0	0	D1、D2 和系统的内容保持不变
运行	DStandby	DStop	1	0	0	0	D1 内容丢失，D2 和系统内容保留
运行	DRun 或 DStop	DStandby	0	1	0	0	D2 内容丢失，D1 和系统内容保留
运行	DStandby	DStandby	1	1	0	0	D1 和 D2 内容丢失，系统内容保留
Stop	DStop	DStop	0	0	0	1	D1、D2 和系统内容保留，时钟系统复位
Stop	DStandby	DStop	1	0	0	1	D1 内容丢失，D2 和系统内容保留，时钟系统复位
Stop	DStop	DStandby	0	1	0	1	D2 内容丢失，D1 和系统内容保留，时钟系统复位
Stop	DStandby	DStandby	1	1	0	1	D1 和 D2 内容丢失，系统内容保留，时钟系统复位
待机	DStandby	DStandby	0(1)	0(1)	1	0	D1、D2 和系统内容丢失

1. 从待机模式返回时，SBF_D1 和 SBF_D2 反映复位值。

6.6.4 电源管理示例

- *图 26* 显示了运行模式下的 V_{CORE} 电压调节行为。
- *图 27* 显示了停止模式下的 V_{CORE} 电压调节行为。
- *图 28* 显示了待机模式下的 V_{CORE} 稳压器和电压调节行为。
- *图 29* 显示了运行模式下且 D1 和 D2 处于 DStandby 模式时的 V_{CORE} 电压调节行为。

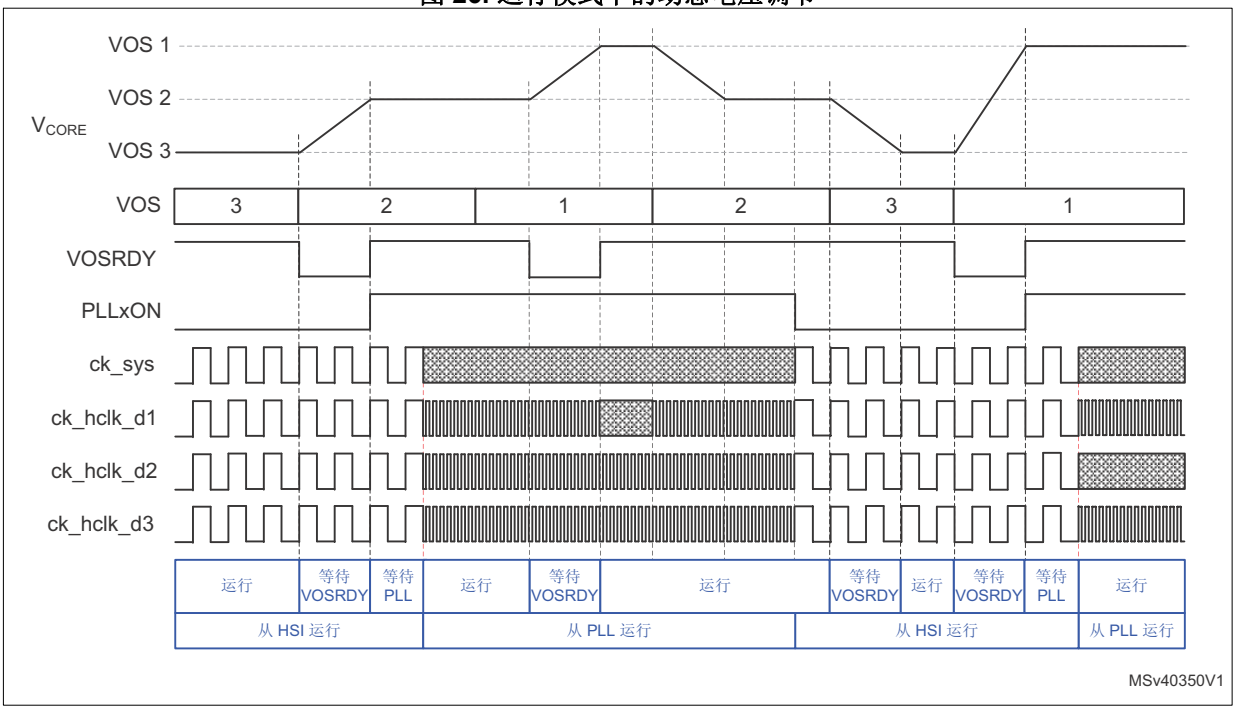
运行模式下的 V_{CORE} 电压调节行为的示例

图 26 说明了以下系统操作序列：

1. 复位后，系统以 VOS3 从 HSI 启动。
2. 首先，通过 PLL 和电压调节 VOS2 使系统性能升到中速时钟。为此：
   a) 将电压调节编程到 VOS2。
   b) V_{CORE} 电源达到 VOSRDY 指示的所需值后，通过使能 PLL 增大时钟频率。
   c) PLL 锁定后，切换系统时钟。
3. 之后，通过 PLL 和电压调节 VOS1 使系统性能升到高速时钟。为此：
   a) 将电压调节编程到 VOS1。
   b) V_{CORE} 电源达到 VOSRDY 指示的所需值后，增大时钟频率。
4. 然后，通过电压调节 VOS2 使系统性能降到中速时钟。为此：
   a) 首先，减小系统频率。
   b) 将电压调节降为 VOS2。
5. 下一步是通过电压调节 VOS3 使系统性能降到 HSI 时钟。为此：
   a) 将时钟切换到 HSI。
   b) 禁止 PLL。
   c) 将电压调节降为 VOS3。
6. 之后，可通过 PLL 使系统性能升到高速时钟。为此：
   a) 将电压调节编程到 VOS1。
   b) V_{CORE} 电源达到 VOSRDY 指示的所需值后，通过使能 PLL 增大时钟频率。
   c) PLL 锁定后，切换系统时钟。

当系统性能（时钟频率）发生变化时，VOS 应相应地设置。否则，系统可能会不可靠。

图 26. 运行模式下的动态电压调节



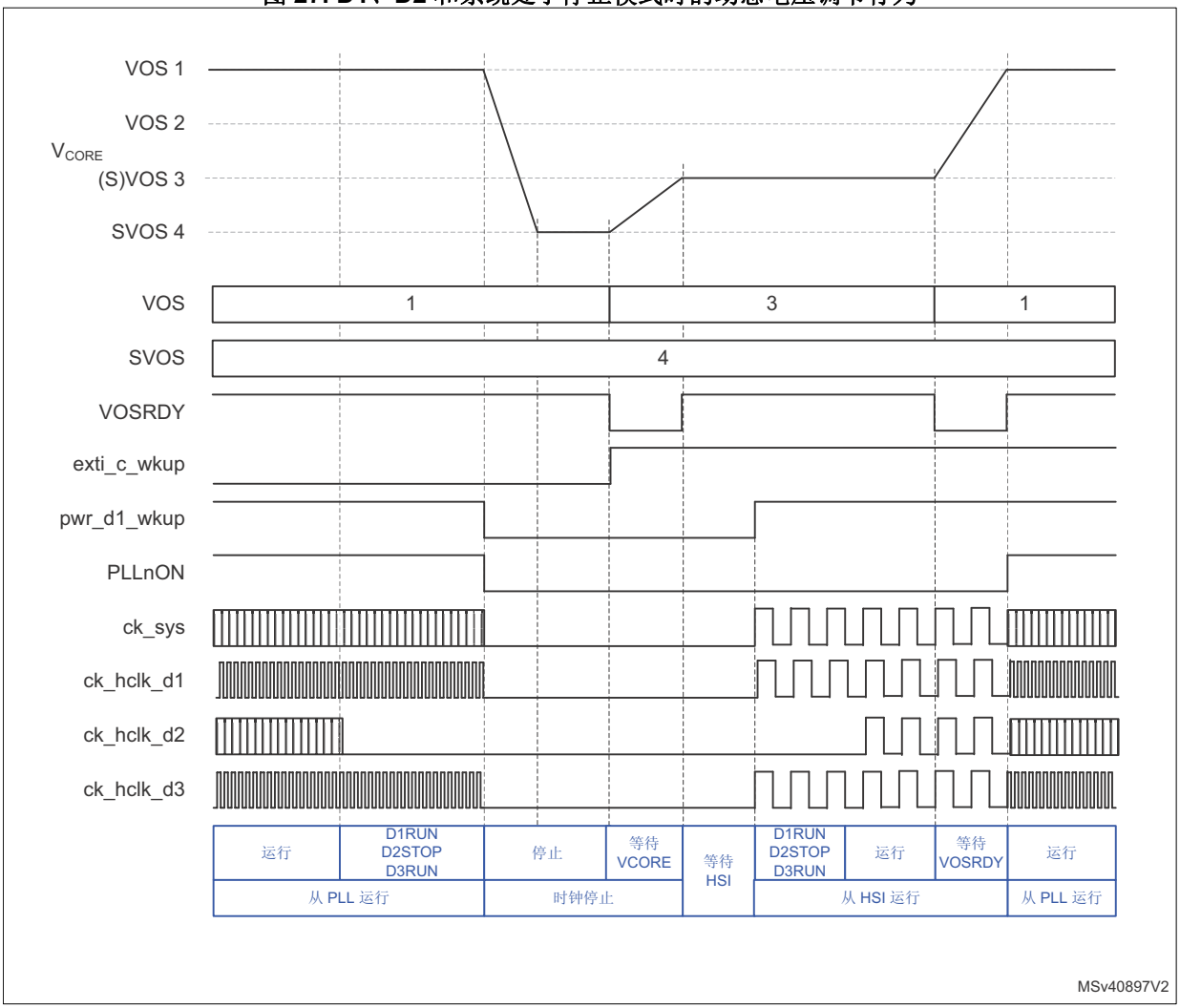
1. 每个步骤的寄存器位的状态以蓝色显示。

停止模式下的 V_{CORE} 电压调节行为的示例

图 27 说明了以下系统操作序列：

1. 系统在高性能模式（VOS1 电压调节）下通过 PLL 运行。
2. 如果 D2 域首先要进入 DStop 模式，则 CPU 子系统将释放其中的所有外设。D2 系统时钟停止。系统仍提供高性能系统时钟，因此电压调节应保持 VOS1 值。
3. 在第二步中，CPU 子系统进入 CStop 模式，D1 域进入 DStop 模式且系统进入停止模式。系统时钟停止且硬件将电压调节降到软件预先选择的 SVOS4 值。
4. CPU 子系统随后唤醒。系统退出停止模式，D1 域退出 DStop 模式且 CPU 子系统退出 CStop 模式。硬件随后将电压调节设置为 VOS3 值并等待达到所请求的电源级别，再使能 HSI 时钟。HSI 时钟稳定后，使能系统时钟和 D1 系统时钟。
5. CPU 子系统在 D2 域中分配一个外设。使能 D2 系统时钟。
6. 随后，系统性能提高。为此：
   a) 软件先将电压调节设置为 VOS1。
   b) V_{CORE} 电源达到 VOSRDY 指示的所需值后，通过使能 PLL 增大时钟频率。
   c) PLL 锁定后，可切换系统时钟。

图 27. D1、D2 和系统处于停止模式时的动态电压调节行为



1. 每个步骤的寄存器位的状态以蓝色显示。

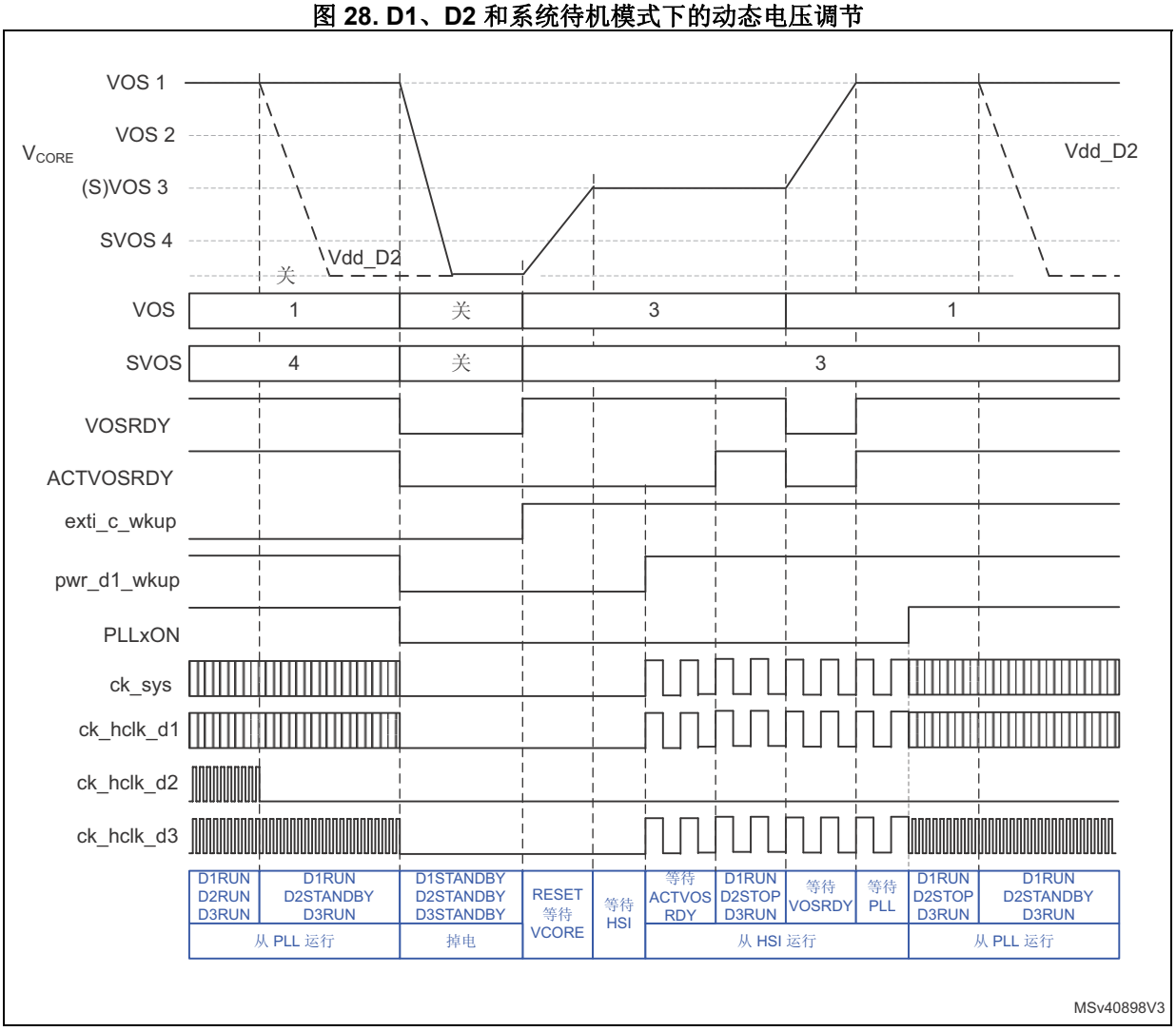
V_{CORE} 稳压器和电压调节行为的示例在待机模式下

图 28 说明了以下系统操作序列：

1. 系统在高性能模式（VOS1 电压调节）下通过 PLL 运行。
2. 如果 D2 域首先要进入 DStandby 模式，则 CPU 子系统将释放其中的所有外设。D2 域总线矩阵时钟停止且电源关闭。系统性能不变，因此电压调节也不变。
3. 随后，CPU 子系统进入 CStop 模式，D1 域进入 DStandby 模式且系统进入待机模式。系统时钟停止且稳压器关闭。
4. 随后，系统通过唤醒源唤醒。系统退出待机模式。硬件将电压调节设置为默认 VOS3 值并等待达到所请求的电源级别，再使能默认的 HSI 时钟。HSI 时钟稳定后，使能系统时钟和 D1 子系统时钟。由于 D2 域中未分配外设，因此该域保持 DStop 模式。之后，软件应先检查 ACTVOSRDY 是否有效，再更改系统性能。

5. 在下一步中，提高系统性能。为此：
   a) 软件先将电压调节提高到 VOS1 值。
   b) 使能 PLL 之前，它将等待达到所请求的电源级别（通过监视 VOSRDY 位确定）。
   c) PLL 锁定后，可切换系统时钟。
6. CPU 子系统将 D2 域置于 DStandby 模式。

图 28. D1、D2 和系统待机模式下的动态电压调节



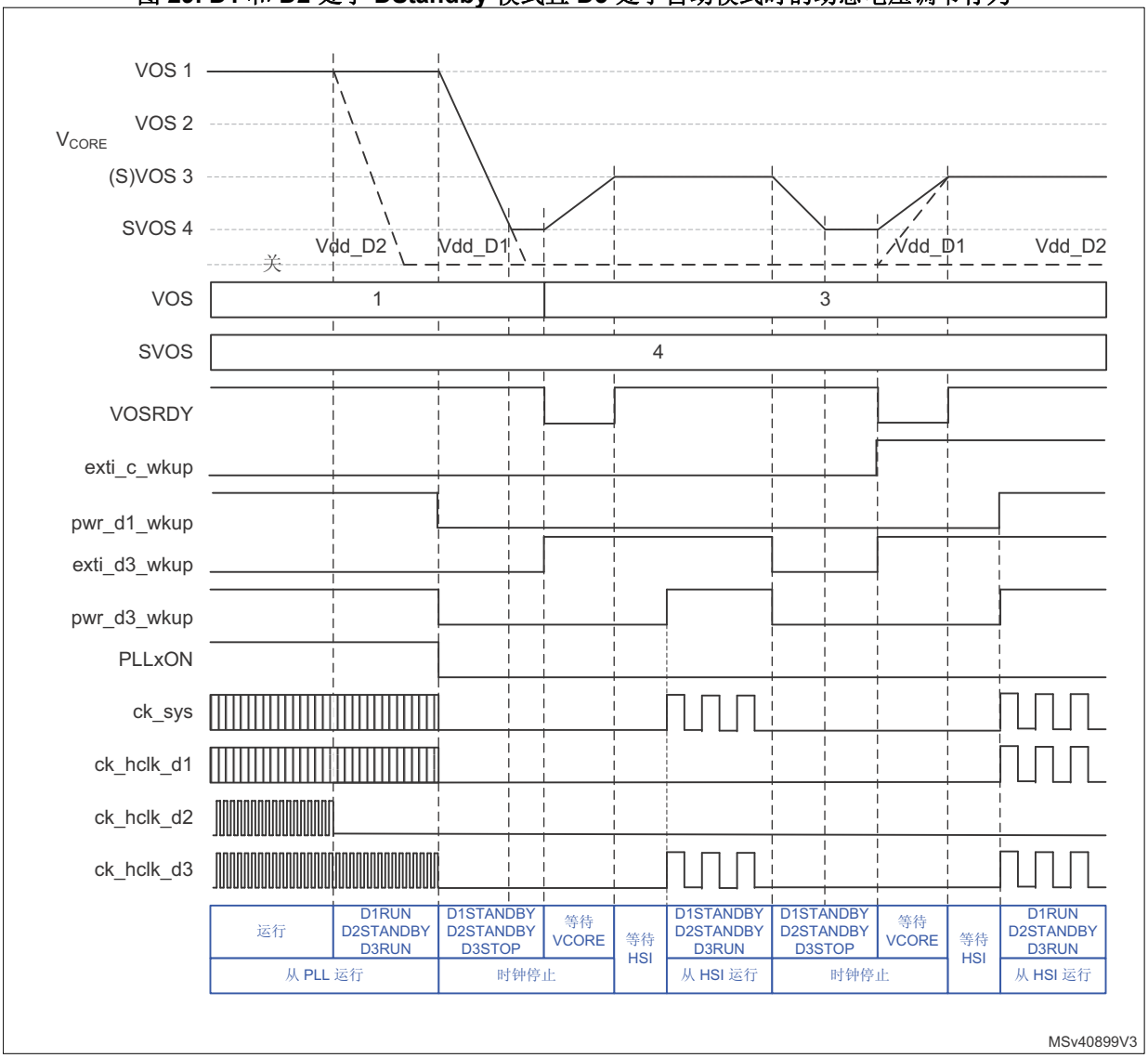
1. 每个步骤的寄存器位的状态以蓝色显示。

运行模式下且 D1 和 D2 域处于 DStandby 模式时的 V_{CORE} 电压调节行为的示例

图 29 说明了以下系统操作序列：

1. 系统在高性能模式（VOS1 电压调节）下通过 PLL 运行。
2. 如果 D2 域首先要进入 DStandby 模式，则 CPU 子系统将释放其中的所有外设。D2 域总线矩阵时钟停止且其电源关闭。系统性能不变，因此电压调节也不变。
3. CPU 子系统随后进入 CStop 模式且 D1 域进入 DStandby 模式。D1 域总线矩阵时钟停止且其电源关闭。同时系统进入停止模式。系统时钟停止且硬件将电压调节降到软件预先选择的 SVOS4 值。
4. 随后，系统通过 D3 自动模式唤醒事件唤醒。系统退出停止模式。硬件将电压调节设置为默认 VOS3 值并等待达到所请求的电源级别，再使能 HSI 时钟。HSI 时钟稳定后，使能系统时钟。系统在 D3 自动模式下运行。
5. D3 自动模式唤醒源随后清除，导致系统进入停止模式。系统时钟停止且电压调节下降到软件预先选择的 SVOS4 值。
6. CPU 子系统随后唤醒。系统退出停止模式，D1 域退出 DStandby 模式且 CPU 子系统退出 CStop 模式。硬件将电压调节设置为默认 VOS3 值并等待达到所请求的电源级别，再使能默认的 HSI 时钟。HSI 时钟稳定后，使能系统时钟和 D1 子系统时钟。D2 域保持 DStandby 模式。

图 29. D1 和 D2 处于 DStandby 模式且 D3 处于自动模式时的动态电压调节行为



1. 每个步骤的寄存器位的状态以蓝色显示。

6.7 低功耗模式

系统提供了多个低功耗模式，可在 CPU 不需要执行代码时（比如等待外部事件时）节省功耗。由用户根据应用选择具体的模式，以在低功耗、短启动时间和可用唤醒源之间寻求最佳平衡。

- 降低系统时钟速度（请参见*第 8.5.6 节：系统时钟 (sys_ck)*）
- 控制各个外设时钟（请参见*第 8.5.11 节：外设时钟门控*）
- 低功耗模式
  - CSleep（CPU 时钟停止）
  - CStop（CPU 子系统时钟停止）
  - DStop（域总线矩阵时钟停止）
  - 停止（系统时钟停止）
  - DStandby（域掉电）
  - 待机（系统掉电）

6.7.1 降低系统时钟速度

在运行模式下，可降低系统时钟 ck_sys 的速度。有关详细信息，请参见*第 8.5.6 节：系统时钟 (sys_ck)*。

6.7.2 控制外设时钟

在运行模式下，可随时停止各外设的 HCLKx 和 PCLKx 以降低功耗，方法是配置 RCC_C1_xxxxENR 或 RCC_DnxxxxENR 中的 PERxEN 位。

要降低 CSleep 模式下的功耗，可通过配置 RCC_C1_xxxxLPENR 或 RCC_DnxxxxLPENR 中的 PERxLPEN 位禁止各外设的时钟。对于在 CSleep 模式下仍接收时钟的外设，可在进入 CSleep 模式前降低其时钟速度。

6.7.3 进入低功耗模式

通过执行 WFI（等待中断）或 WFE（等待事件）指令，或者当从 ISR 返回后，Cortex®-M 系统控制寄存器中的 SLEEPONEXIT 位置 1 时，即可使得 MCU 进入 CPU 子系统 CSleep 和 CStop 低功耗模式。

当 CPU 子系统在某个域中分配了一个外设并进入 CStop 模式时，或者当 D2 域的外设都释放时，相应域可进入 DStop 或 DStandby 低功耗模式。

当所有 EXTI 唤醒源清除且其它域处于 DStop 或 DStandby 模式时，系统可进入停止或待机低功耗模式。

6.7.4 退出低功耗模式

CPU 子系统可通过任何中断或事件退出 CSleep 模式，具体取决于进入低功耗模式的方式：

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

    当 Cortex®-M7 系统控制寄存器中的 **SEVONPEND = 0** 时，通过使能外设控制寄存器和 NVIC 中的中断。

    当 MCU 从 WFE 恢复时，需要清除相应外设的中断挂起位和 NVIC 外设 IRQ 通道挂起位（在 NVIC 中断清除挂起寄存器中）。只有当 NVIC 中断的优先级足够高时，才能唤醒和中断 MCU。

    当 Cortex®-M7 系统控制寄存器中的 **SEVONPEND = 1** 时，通过使能外设控制寄存器和NVIC(可选)中的中断。当 MCU 从 WFE 恢复时，需要清零外设中断挂起位和使能的 NVIC 外设 IRQ 通道挂起位（在 NVIC 中断清除挂起寄存器中）。

    所有 NVIC 中断将唤醒 MCU，即使是禁止的亦是如此。

    只有当已使能 NVIC 中断的优先级足够高时，才能唤醒和中断 MCU。
  - 事件

    EXTI 线必须配置为事件模式。当 CPU 从 WFE 恢复时，因为对应事件线的挂起位没有被置 1，不必清零 EXTI 外设的中断挂起位或 NVIC IRQ 通道挂起位。可能需要清零相应外设中的中断标志。

CPU 子系统可通过使能 EXTI 中断或事件退出 CStop、DStop 和停止模式，具体取决于进入低功耗模式的方式（见上文）。

系统可通过使能 EXTI 唤醒从停止模式唤醒，而不会唤醒 CPU 子系统。在这种情况下，系统将在 D3 自动模式下工作。

CPU 子系统可通过使能 EXTI 中断或事件退出 DStandby 模式，而与进入 DStandby 模式的方式无关。程序将从 CPU 本地复位（例如，从系统配置块 (SYSCFG) 获取复位向量）重新执行。

当 CPU 在 D2 域中分配第一个外设时，该域可退出 DStop 或 DStandby 模式。

使能外部复位（NRST 引脚）、IWDG 复位、使能的 WKUPx 引脚上出现上升沿或者触发 RTC 事件时，CPU 子系统退出待机模式。程序将按照系统复位（例如，启动引脚采样、选项字节加载或复位向量读取）后的方式重新执行。

6.7.5 CSleep 模式

CSleep 模式仅适用于 CPU 子系统。在 CSleep 模式下，CPU 时钟停止。CPU 子系统外设时钟根据 RCC_C1_xxxxENR 或 RCC_DnxxxxENR 中 PERxLPEN 位的值工作。

进入 CSleep 模式

根据*第 6.7.3 节：进入低功耗模式*，当 Cortex®-M 系统控制寄存器中的 SLEEPDEEP 位清零时，将进入 CSleep 模式。

有关如何进入 CSleep 模式的详细信息，请参见*表 30*。

退出 CSleep 模式

根据*第 6.7.4 节：退出低功耗模式*退出 CSleep 模式。

有关如何退出 CSleep 模式的详细信息，请参见*表 30*。

表 30. CSleep 模式

CSleep 模式	说明
进入模式	WFI（等待中断）或 WFE（等待事件），且： – SLEEPDEEP = 0（请参见 Cortex®-M 系统控制寄存器。） – CPU NVIC 中断和事件清除。
	从 ISR 返回，且： – SLEEPDEEP = 0 及 – SLEEPONEXIT = 1（请参见 Cortex®-M 系统控制寄存器。） – CPU NVIC 中断和事件清除。
退出模式	如果使用 WFI 或从 ISR 返回进入： – 在 NVIC 中使能的任何中断：请参见*表 129：NVIC* 如果使用 WFE 进入且 SEVONPEND = 0： – 任何事件：请参见*第 20.5.3 节：EXTI CPU 唤醒程序* 如果使用 WFE 进入且 SEVONPEND = 1： – 任何中断（即使在 NVIC 中禁止时）：请参见*表 129：NVIC*，或者任何事件：请参见*第 20.5.3 节：EXTI CPU 唤醒程序*
唤醒延迟	无

6.7.6 CStop 模式

CStop 模式仅适用于 CPU 子系统。在 CStop 模式下，CPU 时钟停止。大多数 CPU 子系统外设时钟也停止，仅具有 PERxAMEN 位的 CPU 子系统外设相应地工作。

在 CStop 模式下，具有内核时钟请求的 CPU 子系统外设仍可请求其内核时钟。对于具有 PERxAMEN 位的外设，此位应置 1 以便请求内核时钟。

进入 CStop 模式

根据*第 6.7.3 节：进入低功耗模式*，当 Cortex®-M 系统控制寄存器中的 SLEEPDEEP 位置 1 时，将进入 CStop 模式。

有关如何进入 CStop 模式的详细信息，请参见*表 31*。

退出 CStop 模式

根据*第 6.7.4 节：退出低功耗模式*退出 CStop 模式。

有关如何退出 CStop 模式的详细信息，请参见*表 31*。

表 31. CStop 模式

CStop 模式	说明
进入模式	WFI（等待中断）或 WFE（等待事件），且： – SLEEPDEEP = 1（请参见 Cortex®-M 系统控制寄存器。） – CPU NVIC 中断和事件清除。 – 所有 CPU EXTI 唤醒源清除。
	从 ISR 返回，且： – SLEEPDEEP = 1 及 – SLEEPONEXIT = 1（请参见 Cortex®-M 系统控制寄存器。） – CPU NVIC 中断和事件清除。 – 所有 CPU EXTI 唤醒源清除。
退出模式	如果使用 WFI 或从 ISR 返回进入： – 在 NVIC 中使能的 EXTI 中断：请参见*表 129：NVIC*（对于未停止或掉电的外设）。 如果使用 WFE 进入且 SEVONPEND = 0： – EXTI 事件：请参见*第 20.5.3 节：EXTI CPU 唤醒程序*（对于未停止或掉电的外设）。 如果使用 WFE 进入且 SEVONPEND = 1： – EXTI 中断（即使在 NVIC 中禁止）：请参见*表 129：NVIC*，或者 EXTI 事件：请参见*第 20.5.3 节：EXTI CPU 唤醒程序*（对于未停止或掉电的外设）。
唤醒延迟	EXTI 和 RCC 唤醒同步（请参见*第 8.4.7 节：上电和唤醒序列*）

6.7.7 DStop 模式

只有当 CPU 子系统处于 CStop 模式且在 D1 域和/或 D2 域中分配了外设，相应域才进入 DStop 模式（请参见*表 32*）。在 DStop 模式下，域总线矩阵时钟停止。

当 Flash 通过 PWR_CR1 寄存器中的 FLPS 使能时，可进入低功耗停止模式。这可以在域 DStop 重启时间和低功耗之间达到最佳平衡。

表 32. DStop 模式概述

外设分配	CPU	D1 域	D2 域	备注
未在 D2 域中分配外设	CRun 或 CSleep	DRun	DStop	
	CStop	DStop	DStop	
在 D2 域中分配了外设	CRun 或 CSleep	DRun	DRun	CPU 子系统，使 D2 域保持有效。
	CStop	DStop	DStop	

在 DStop 模式下，使用 LSI 或 LSE 时钟的域外设和具有内核时钟请求的外设仍能够工作。

进入 DStop 模式

根据*第 6.7.3 节：进入低功耗模式*，当域的 *PWR CPU 控制寄存器 (PWR_CPUCR)* 中至少有一个 PDDS_Dn 位选择停止时，将进入 DStop 模式。

有关如何进入 DStop 模式的详细信息，请参见*表 33*。

如果正在执行 Flash 编程，DStop 模式的进入将延迟到存储器访问结束后执行。

如果正在访问域总线矩阵，DStop 模式的进入则延迟到域总线矩阵访问结束后执行。

退出 DStop 模式

根据*第 6.7.4 节：退出低功耗模式*退出 DStop 模式。

有关如何退出 DStop 模式的详细信息，请参见*表 33*。

当退出 DStop 模式时，CPU 子系统时钟、域总线矩阵时钟和电压调节取决于系统模式。

- 当系统未进入停止模式时，CPU 子系统时钟、域总线矩阵时钟和电压调节值与进入 DStop 模式时相同。
- 当系统进入停止模式后，CPU 子系统时钟、域总线矩阵时钟和电压调节将复位。

表 33. DStop 模式

DStop 模式	说明
进入模式	– 域 CPU 子系统进入 CStop。 – CPU 子系统在 D2 域中分配了一个外设并且进入 CStop。 – CPU 子系统释放其在 D2 域中的最后一个外设。 – 域的 PDDS_Dn 位选择停止模式。
退出模式	– 域 CPU 子系统退出 CStop 模式（请参见*表 31*）。 – CPU 子系统在 D2 域中分配了一个外设并且退出 CStop 模式（请参见*表 31*）。 – CPU 子系统在 D2 域中分配第一个外设。
唤醒延迟	EXTI 和 RCC 唤醒同步（请参见*第 8.5.7 节：在停止和待机模式下处理时钟发生器*）。

6.7.8 停止模式

只有当 CPU 子系统处于 CStop 模式、EXTI 唤醒源无效且任何域的 *PWR CPU 控制寄存器 (PWR_CPUCR)* 中至少有一个 PDDS_Dn 位请求停止，系统 D3 域才进入停止模式。在停止模式下，系统时钟（包括 PLL 和 D3 域总线矩阵时钟）停止。当选择 HSI 或 CSI 时，系统时钟根据 RCC_CR 寄存器中的 HSIKERON 和 CSIKERON 位工作。其它系统时钟源停止。

在系统 D3 域停止模式下，D1 域和 D2 域处于 DStop 和/或 DStandby 模式。

在停止模式下，使用 LSI 或 LSE 时钟的域外设和具有选择 HSI 或 CSI 作为源的内核时钟请求的外设仍能够工作。

在系统停止模式下，可以通过对各控制位进行编程来选择使以下功能保持有效：

- 独立看门狗 (IWDG)
  IWDG 通过写入其密钥寄存器或使用硬件选项来启动。而且一旦启动便无法停止，除非复位（请参见*第 45 节：独立看门狗 (IWDG)* 中的*第 45.3 节*）。
- 实时时钟 (RTC)
  通过 *RCC 备份域控制寄存器 (RCC_BDCR)* 中的 RTCEN 位进行配置。
- 内部 RC 时钟 (LSI RC)
  通过 *RCC 时钟控制和状态寄存器 (RCC_CSR)* 中的 LSION 位进行配置。
- 32.768 kHz 外部时钟 (LSE OSC)
  通过 *RCC 备份域控制寄存器 (RCC_BDCR)* 中的 LSEON 位进行配置。
- 能够通过 LSI 或 LSE 时钟运行的外设。
- 具有内核时钟请求的外设。
- 内部 RC 时钟（HSI 和 CSI）
- 通过 *RCC 时钟控制和状态寄存器 (RCC_CSR)* 中的 HSIKERON 和 CSIKERON 位进行配置。
- 在停止模式下，ADC 或 DAC 也会产生功耗，除非在进入该模式前将其禁止。要禁止这些转换器，必须将 ADC_CR2 寄存器中的 ADON 位和 DAC_CR 寄存器中的 ENx 位都清零。

退出系统停止模式时，所选的 SVOS4 和 SVOS5 值会额外增加一段启动延时（请参见*表 34*）。

表 34. 停止模式下的运行

SVOS	LPDS	停止模式 稳压器运行	唤醒延迟
SVOS3	0	**主路**	没有额外的唤醒时间。
	1	**LP**	稳压器从 LP 模式唤醒的时间。
SVOS4 或 SVOS5	x	**LP**	稳压器从 LP 模式唤醒的时间 + 电压从 SVOS4 或 SVOS5 值转换到 VOS3 值的唤醒时间。

进入停止模式

根据*第 6.7.3 节：进入低功耗模式*，当任何域的 *PWR CPU 控制寄存器 (PWR_CPUCR)* 中至少有一个 PDDS_Dn 位 n 请求停止时，将进入停止模式。

有关如何进入停止模式的详细信息，请参见*表 35*。

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

如果正在访问域总线矩阵（AXI、AHB 或 APB），停止模式的进入则延迟到域总线矩阵访问结束后执行。

注： *使用 DSB 指令确保未完成的存储器事务在进入停止模式之前完成。*

要使具有内核时钟请求的外设在停止模式下工作，系统必须使用 SVOS3 值。

退出停止模式

根据*第 6.7.4 节：退出低功耗模式*退出停止模式。

有关如何退出停止模式的详细信息，请参见*表 35*。

当退出停止模式时，系统时钟、D3 域总线矩阵时钟和电压调节将复位。

PWR CPU 控制寄存器 (PWR_CPUCR) 中的 STOPF 状态位表示系统已退出停止模式（请参见*表 29*）。

表 35. 停止模式

停止模式	说明
进入模式	– CPU 处于 CStop 模式、没有有效的 EXTI 唤醒源且 Run_D3 = 0 时。 – 任何域中至少有一个 PDDS_Dn 位选择停止模式。
退出模式	– EXTI 唤醒时。
唤醒延迟	系统时钟启动（禁止时）。 + EXTI 和 RCC 唤醒同步。 + 有关电压调节的信息，请参见*表 34*（请参见*第 6.6.2 节：电压调节*）。

停止模式下的 I/O 状态

在停止模式下，I/O 引脚配置保持不变。

6.7.9 DStandby 模式

与 DStop 模式相同，DStandby 模式基于 CPU 子系统 CStop 模式。但域 V_{CORE} 电源关闭。如果某个域中分配了外设，则只有当 CPU 子系统处于 CStop 模式，该域才会进入 DStandby 模式。

如果某个域中分配了外设，则只有当 CPU 子系统处于 CStop 模式，该域才会进入 DStandby 模式，*PWR CPU 控制寄存器 (PWR_CPUCR)* 中的 PDDS_Dn 位将相应地配置。在 DStandby 模式下，相应域将掉电且该域 RAM 和寄存器的内容将丢失。

进入 DStandby 模式

根据*第 6.7.3 节：进入低功耗模式*，当 Dn 域的 *PWR CPU 控制寄存器 (PWR_CPUCR)* 中的 PDDS_Dn 位选择待机模式时，将进入 DStandby 模式。

有关如何进入 DStandby 模式的详细信息，请参见*表 36*。

如果正在执行 Flash 编程，DStandby 模式的进入将延迟到存储器访问结束后执行。

如果正在访问域总线矩阵，DStandby 模式的进入则延迟到域总线矩阵访问结束后执行。

注： *当 CPU 设置 PDDS_D2 位选择待机模式时，D2 域将进入 DStandby 模式（CPU 未在 D2 域中分配外设）。*

退出 DStandby 模式

根据*第 6.7.4 节：退出低功耗模式*退出 DStandby 模式。

有关如何退出 DStandby 模式的详细信息，请参见*表 36*。

注： *当 D2 域处于 DStandby 模式且 CPU 设置域的 PDDS_D2 位选择停止模式时，D2 域将保持 DStandby 模式。只有当 CPU 在 D2 域中分配一个外设时，D2 域才会退出 DStandby 模式。*

退出 DStandby 模式时，域的 CPU 和外设将复位。不过，CPU 子系统时钟、域总线矩阵时钟和电压调节的状态取决于系统模式：

- 当系统未进入停止模式时，CPU 子系统时钟、域总线矩阵时钟和电压调节与进入 DStandby 模式时相同。
- 当系统进入停止或待机模式后，CPU 子系统时钟、域总线矩阵时钟和电压调节将复位。

当 D2 域因 CPU 子系统退出 DStandby 模式（比如，分配第一个外设时或者在 D2 域中分配外设且 CPU 子系统退出 CStop 模式时），CPU 应确保相应域已退出 DStandby 模式。为确保正确工作，建议遵守下列顺序：

1. 首先检查域总线矩阵时钟是否可用。可以在 RCC_CR 寄存器中检查域总线矩阵时钟的状态：
   – 当 RCC DnCKRDY = 0 时，域总线矩阵时钟停止。
   – 当 RCC DnCKRDY = 1 时，域总线矩阵时钟使能。
2. 随后，等待相应域退出 DStandby 模式。为此，需检查 *PWR CPU 控制寄存器 (PWR_CPUCR)* 中的 SBF_Dn 标志。相应域上电，并且只能在 SBF_Dn 清零时访问。下面是一个代码样例：

```
Loop
write PWR SBF_Dn = 0 ; try to clear bit.
read PWR SBF_Dn
While 1 ==> loop
```

表 36. DStandby 模式

DStandby 模式	说明
进入模式	– 域 CPU 子系统进入 CStop。 – CPU 子系统在 D2 域中分配了一个外设并且进入 CStop。 – CPU 子系统释放其在 D2 域中的最后一个外设。 – 域的 PDDS_Dn 位选择待机模式。 – 电源控制/状态寄存器 (PWR_WKUPFR) 中的所有 WKUPF 位均清零。
退出模式	– CPU 子系统退出 CStop 模式（请参见*表 31*）。 – CPU 子系统在 D2 域中分配了一个外设并且退出 CStop 模式（请参见*表 31*）。 – CPU 子系统在 D2 域中分配第一个外设。
唤醒延迟	EXTI 和 RCC 唤醒同步。 + 域上电和复位。 （参见*第 8.4.7 节：上电和唤醒序列*）

6.7.10 待机模式

待机模式下可达到最低功耗。与停止模式相同，它基于 CPU 子系统 CStop 模式。但 V_{CORE} 电源稳压器关闭。

只有当 D1 和 D2 域均处于 DStandby 模式，系统 D3 域才进入待机模式。当系统 D3 域进入待机模式时，稳压器将禁止。因此整个 V_{CORE} 域断电。PLL、HSI 时钟、CSI 时钟、HSI48 和 HSE 时钟也将关闭。除备份域（RTC 寄存器、RTC 备份寄存器和备份 RAM）和待机电路中的寄存器外，SRAM 和寄存器内容都将丢失（参见*第 6.4.4 节：备份域*）。

在系统待机模式下，可以通过对各控制位进行编程来选择以下功能：

- 独立看门狗 (IWDG)

  IWDG 通过编程其密钥寄存器或使用硬件选项来启动。而且一旦启动便无法停止，除非复位（请参见*第 45 节：独立看门狗 (IWDG)* 中的*第 45.3 节*）。
- 实时时钟 (RTC)

  通过备份域控制寄存器 (RCC_BDCR) 中的 RTCEN 位进行配置。
- 内部 RC 时钟 (LSI RC)

  通过控制/状态寄存器 (RCC_CSR) 中的 LSION 位进行配置。
- 32.768 kHz 外部时钟 (LSE OSC)

  通过备份域控制寄存器 (RCC_BDCR) 中的 LSEON 位进行配置。

进入待机模式

根据*第 6.7.3 节：进入低功耗模式*，当所有域的 *PWR CPU 控制寄存器 (PWR_CPUCR)* 中的所有 PDDS_Dn 位均请求待机时，将进入待机模式。

有关如何进入待机模式的详细信息，请参见*表 38*。

退出待机模式

根据*第 6.7.4 节：退出低功耗模式*退出待机模式。

有关如何退出待机模式的详细信息，请参见*表 38*。

检测到外部复位（NRST 引脚）、IWDG 复位、WKUP 引脚事件、RTC 闹钟、入侵事件或时间戳事件时，系统退出待机模式。除电源控制和状态寄存器（*PWR 控制寄存器 2 (PWR_CR2)* 和 *PWR 控制寄存器 3 (PWR_CR3)*）以及 *PWR CPU 控制寄存器 (PWR_CPUCR)*、*PWR 唤醒标志寄存器 (PWR_WKUPFR)* 和 *PWR 唤醒使能和极性寄存器 (PWR_WKUPEPR)* 的 SBF 位外的所有寄存器都将在从待机模式唤醒后复位。

从待机模式唤醒后，程序将按照系统复位（启动选项采样、启动向量复位获取等）后的方式重新执行。*PWR CPU 控制寄存器 (PWR_CPUCR)* 寄存器中的 SBF 状态标志用于指示系统从哪个模式退出（请参见*表 37*）。

表 37. 待机和停止标志

SBF_D2	SBF_D1	SBF	STOPF	说明
0	1	0	0	D1 域退出 DStandby 模式，而系统保持运行模式
0	1	0	1	D1 域退出 DStandby 模式，而系统已处于或退出停止模式
1	0	0	0	D2 域退出 DStandby 模式，而系统保持运行模式
1	0	0	1	D2 域退出 DStandby 模式，而系统已处于或退出停止模式
1	1	0	0	D1 和 D2 域退出 DStandby 模式，而系统保持运行模式
1	1	0	1	D1 和 D2 域退出 DStandby 模式，而系统处于停止模式或正在退出该模式
0	0	0	1	系统已处于或退出停止模式
0(1)	0(1)	1	0	系统退出待机模式

1. 退出待机模式时，SBF_D1 和 SBF_D2 反映复位值

表 38. 待机模式

待机模式	说明
进入模式	– CPU 子系统处于 CStop 模式、没有有效的 EXTI 唤醒源且 RUN_D3 = 0。 – 所有域的所有 PDDS_Dn 位均选择待机模式。 – 电源控制/状态寄存器 (PWR_WKUPFR) 中的所有 WKUPF 位均清零。
退出模式	– WKUP 引脚上升沿或下降沿、RTC 闹钟（闹钟 A 和闹钟 B）、RTC 唤醒事件、入侵事件、时间戳事件、NRST 引脚外部复位 和 IWDG 复位。
唤醒延迟	系统复位阶段（请参见*第 8.4.2 节：系统复位*）

待机模式下的 I/O 状态

在待机模式下，除以下各部分以外，所有 I/O 引脚都处于高阻态，无上下拉，除了：

- 复位引脚（仍可用）
- RTC_AF1 引脚（如果针对入侵、时间戳、RTC 闹钟输出或 RTC 时钟校准输出进行了配置）
- WKUP 引脚（如果使能）。可通过 *PWR 唤醒使能和极性寄存器 (PWR_WKUPEPR)* 中的 WKUPPUPD 寄存器位定义 WKUP 引脚的拉取配置

6.8 PWR 寄存器说明

除非另外说明，否则可按字、半字和字节格式访问 PWR 寄存器。

6.8.1 PWR 控制寄存器 1 (PWR_CR1)

PWR control register 1

偏移地址：0x000

复位值：0xF000 C000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SVOS		Res.	Res.	Res.	Res.	FLPS	DBP	PLS			PVDE	Res.	Res.	Res.	LPDS
rw	rw					rw	rw	rw	rw	rw	rw				rw

位 31:19 保留，必须保持复位值

位 18:17 **ALS**：模拟电压检测器电平选择 (Analog voltage detector level selection)
这些位用于选择通过 AVD 检测的电压阈值。
00：1.7 V
01：2.1 V
10：2.5 V
11：2.8 V

位 16 **AVDEN**：V_{DDA} 上的外设电压监测器 (Peripheral voltage monitor on V_{DDA} enable)
0：禁止 V_{DDA} 上的外设电压监测器
1：使能 V_{DDA} 上的外设电压监测器

位 15:14 **SVOS**：系统停止模式电压调节选择 (System Stop mode voltage scaling selection)
这些位用于控制系统停止模式下的 V_{CORE} 电压级别，以在功耗和性能之间实现最佳平衡。
00：保留
01：SVOS5 电压调节 5
10：SVOS4 电压调节 4
11：SVOS3 电压调节 3（默认）

位 13:10 保留，必须保持复位值

位 9 **FLPS**：DStop 模式下的 Flash 低功耗模式 (Flash low-power mode in DStop mode)
此位能够在退出 DStop 模式时实现低功耗和重启时间的最佳平衡。
将此位置 1 时，Flash 将在 D1 域处于 DStop 模式后进入低功耗模式。
0：当 D1 域进入 DStop 模式时，Flash 将保持正常模式（快速重启时间）。
1：当 D1 域进入 DStop 模式时，Flash 将进入低功耗模式（低功耗）。

位 8 **DBP**：禁止备份域写保护 (Disable backup domain write protection)
在复位状态下，RCC_BDCR 寄存器、RTC 寄存器（包括备份寄存器）以及 PWR_CR2 寄存器的 BREN 和 MOEN 位均受到写访问保护。必须将此位置 1 才能使能对这些寄存器的写访问。
0：禁止对 RTC、RTC 备份寄存器和备份 SRAM 的访问
1：使能对 RTC、RTC 备份寄存器和备份 SRAM 的访问

位 7:5 **PLS**：可编程电压检测器的电平选择 (Programmable voltage detector level selection)
这些位用于选择通过 PVD 检测的电压阈值。
000：1.95 V
001：2.1 V
010：2.25 V
011：2.4 V
100：2.55 V
101：2.7 V
110：2.85 V
111：PVD_IN 引脚上的外部电压级别（与内部 V_{REFINT} 值相比较）。
注：　有关详细信息，请参见产品数据手册的“电气特性”部分。

位 4 **PVDE**：可编程电压检测器使能 (Programmable voltage detector enable)
0：禁止可编程电压检测器。
1：使能可编程电压检测器。

位 3:1　保留，必须保持复位值

位 0 **LPDS**：电压调节为 SVOS3 的低功耗深度睡眠模式 (Low-power Deepsleep with SVOS3)（无论此位的设置为何，SVOS4 和 SVOS5 始终使用低功耗模式）
0：当选择电压调节为 SVOS3 的停止模式时，稳压器处于主模式 (MR)
1：当选择电压调节为 SVOS3 的停止模式时，稳压器处于低功耗模式 (LPR)

6.8.2 PWR 控制状态寄存器 1 (PWR_CSR1)

PWR control status register 1

偏移地址：0x004

复位值：0x0000 4000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	AVDO
															r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ACTVOS		ACTVOSRDY	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PVDO	Res.	Res.	Res.	Res.
r		r									r				

位 31:17　保留，必须保持复位值

位 16 **AVDO**：V_{DDA} 上的模拟电压检测器输出 (Analog voltage detector output on V_{DDA})
此位通过硬件置 1 和清零。仅当通过 AVDEN 位使能V_{DDA}上的 AVD 时此位才有效。
0：V_{DDA} 等于或高于 ALS[2:0] 位选择的 AVD 阈值。
1：VDDA 低于 ALS[2:0] 位选择的 AVD 阈值。
注：　由于在待机模式下禁止 AVD，因此，进入待机模式或执行复位后，此位等于 0，直到 AVDEN 位置 1。

位 15:14 **ACTVOS**：当前应用于 V_{CORE} 电压调节选择的 VOS (VOS currently applied for VCORE voltage scaling selection)。
这些位反映了应用于稳压器的最后一个 VOS 值。

位 13 **ACTVOSRDY**：当前使用的 VOS 电压级别就绪位 (Voltage levels ready bit for currently used VOS)

当禁止稳压器并在 PWR 控制寄存器 3 (PWR_CR3) 中选择旁路模式时，此位通过硬件置 1。

0：电压级别无效，高于或低于当前 VOS 选择的级别。

1：电压级别有效，等于当前 VOS 选择的级别。

位 12:5 保留，必须保持复位值

位 4 **PVDO**：可编程电压检测输出 (Programmable voltage detect output)

此位通过硬件置 1 和清零。仅当通过 PVDE 位使能 PVD 时此位才有效。

0：V_{DD} 或 PVD_IN 电压等于或高于通过 PLS[2:0] 位选择的 PVD 阈值。

1：V_{DD} 或 PVD_IN 电压低于通过 PLS[2:0] 位选择的 PVD 阈值。

注： *由于在待机模式下禁止 PVD，因此，进入待机模式或执行复位后，此位等于 0，直到 PVDE 位置 1。*

位 3:0 保留，必须保持复位值

6.8.3 PWR 控制寄存器 2 (PWR_CR2)

PWR control register 2

偏移地址：0x008

复位值：0x0000 0000

从待机模式唤醒、RESET 信号和 V_{DD} POR 不会复位此寄存器。只有 V_{SW} POR 和 VSWRST 复位才能将此寄存器复位。

当 RCC_BDCR 寄存器中的 VSWRST 位复位 V_{SW} 域时，不得访问此寄存器。

复位后，PWR_CR2 寄存器受写保护。修改其内容前，必须将 PWR_CR1 寄存器中的 DBP 位设置为禁止写保护。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TEMPH	TEMPL	VBATH	VBATL	Res.	Res.	Res.	BRRDY
								r	r	r	r				r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MONEN	Res.	Res.	Res.	BREN
											rw				rw

位 31:24 保留，必须保持复位值

位 23 **TEMPH**：温度级别监视与高阈值 (Temperature level monitoring versus high threshold)

0：温度低于高阈值级别。

1：温度等于或高于高阈值级别。

位 22 **TEMPL**：温度级别监视与低阈值 (Temperature level monitoring versus low threshold)

0：温度高于低阈值级别。

1：温度等于或低于低阈值级别。

位 21 **VBATH**：V_{BAT} 级别监视与高阈值 (V_{BAT} level monitoring versus high threshold)

0：V_{BAT} 级别低于高阈值级别。

1：V_{BAT} 级别等于或高于高阈值级别。

位 20 **VBATL**：V_{BAT} 级别监视与低阈值 (V_{BAT} level monitoring versus low threshold)

0：V_{BAT} 级别高于低阈值级别。

1：V_{BAT} 级别等于或低于低阈值级别。

位 19:17 保留，必须保持复位值

位 16 **BRRDY**：备份调压器就绪 (Backup regulator ready)
此位由硬件置 1，用以指示备份稳压器已就绪。
0：备份调压器未就绪
1：备份调压器已就绪

位 15:5 保留，必须保持复位值

位 4 **MONEN**：V_{BAT} 和温度监视使能 (V_{BAT} and temperature monitoring enable)
将此位置 1 时，使能 V_{BAT} 电源和温度监视。
0：禁止 V_{BAT} 和温度监视。
1：使能 V_{BAT} 和温度监视。

位 3:1 保留，必须保持复位值

位 0 **BREN**：使能备份调压器 (Backup regulator enable)
将此位置 1 时，使能备份稳压器（用于在待机模式和 V_{BAT} 模式下保持备份 RAM 内容）。
如果 BREN 复位，备份调压器关闭。备份 RAM 仍可在运行和停止模式下使用。但在待机模式和 V_{BAT} 模式下其内容将丢失。
如果 BREN 置 1，则在备份稳压器就绪标志 (BRRDY) 置 1 以指示在待机模式和 V_{BAT} 模式下会保持写入 SRAM 中的数据之前，应用程序必须等待。
0：禁止备份调压器
1：使能备份调压器

6.8.4 PWR 控制寄存器 3 (PWR_CR3)

PWR control register 3

偏移地址：0x00C

复位值：0x0000 0006（仅通过 POR 复位，而不会通过从待机模式唤醒和 RESET 引脚复位）。

此寄存器的低字节在 POR 后写入一次，应在更改 VOS 值或 ck_sys 时钟频率之前写入。高字节则没有限制。

将忽略无效的 LDOEN 和 BYPASS 位组合对应的编程数据（请参见 *表 26*）：将不会写入数据，写入一次的机制将锁定寄存器，因此会忽略其它任何写访问。将保持默认的电源配置，*PWR 控制状态寄存器 1 (PWR_CSR1)* 中的 ACTVOSRDY 位将继续指示无效电压级别。系统应在写入新值前断电再上电。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	USB33RDY	USBREGEN	USB33DEN	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
					r	rw	rw								

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	VBRS	VBE	Res.	Res.	Res.	Res.	Res.	SCUEN	LDOEN	BYPASS
						rw	rw						rw	rw	rw

位 31:27 保留，必须保持复位值

位 26 **USB33RDY**：USB 电源就绪 (USB supply ready)。
0：USB33 电源未就绪。
1：USB33 电源就绪。

位 25 **USBREGEN**：USB 稳压器使能 (USB regulator enable)。
0：禁止 USB 稳压器。
1：使能 USB 稳压器。

位 24 **USB33DEN**：$V_{DD33USB}$ 电压级别检测器使能 ($V_{DD33USB}$ voltage level detector enable)。
0：禁止 $V_{DD33USB}$ 电压级别检测器。
1：使能 $V_{DD33USB}$ 电压级别检测器。

位 23:10 保留，必须保持复位值

位 9 **VBRS**：V_{BAT} 充电电阻选择 (V_{BAT} charging resistor selection)
0：通过 5 kΩ 电阻为 V_{BAT} 充电。
1：通过 1.5 kΩ 电阻为 V_{BAT} 充电。

位 8 **VBE**：V_{BAT} 充电使能 (V_{BAT} charging enable)
0：禁止 V_{BAT} 电池充电。
1：使能 V_{BAT} 电池充电。

位 7:3 保留，必须保持复位值

位 2 **SCUEN**：电源配置更新使能 (Supply configuration update enable)
此位为只读：
0：电源配置更新已锁定。
1：已针对电源配置使能单次写操作（LDOEN 和 BYPASS）

位 1 **LDOEN**：低压降稳压器使能 (Low drop-out regulator enable)
0：禁止低压降稳压器。
1：使能低压降稳压器（默认）

位 0 **BYPASS**：电源管理单元旁路(Power management unit bypass)
0：电源管理单元正常工作。
1：电源管理单元旁路，电压监测仍处于激活状态。

6.8.5 PWR CPU 控制寄存器 (PWR_CPUCR)

PWR CPU control register

此寄存器可控制 CPU 电源。

偏移地址：0x010

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	RUN_D3	Res.	CSSF	SBF_D2	SBF_D1	SBF	SOPFF	Res.	Res.	PDDS_D3	PDDS_D2	PDDS_D1
				rw		rw	r	r	r	r			rw	rw	rw

位 31:12 保留，必须保持复位值

位 11： **RUN_D3**：将系统 D3 域保持在运行模式，无关乎 CPU 子系统模式 (Keep system D3 domain in Run mode regardless of the CPU subsystem modes)

0：D3 域跟随 CPU 子系统模式。

1：D3 域保持运行模式，无关乎 CPU 子系统模式。

位 10 保留，必须保持复位值

位 9 **CSSF**：将待机和停止标志清零 (Clear Standby and Stop flags)（始终读为 0）

此位通过硬件清零。

0：无影响。

1：STOPF、SBF、SBF_D1 和 SBF_D2 标志清零。

位 8 **SBF_D2**：D2 域 DStandby 标志 (D2 domain DStandby flag)

该位通过硬件置 1，通过系统复位或通过将 CSSF 位置 1 清零。置 1 后，只有 D2 域不再处于 DStandby 模式时，才能将该位清零。

0：D2 域未处于 DStandby 模式

1：D2 域已处于 DStandby 模式

位 7 **SBF_D1**：D1 域 DStandby 标志 (D1 domain DStandby flag)

该位通过硬件置 1，通过系统复位或通过将 CSSF 位置 1 清零。置 1 后，只有 D1 域不再处于 DStandby 模式时，才能将该位清零。

0：D1 域未处于 DStandby 模式

1：D1 域已处于 DStandby 模式

位 6 **SBF**：系统待机标志 (System Standby flag)

此位由硬件置 1，并且只能通过 POR（上电复位）或将 CSSF 位置 1 清零。

0：系统未处于待机模式

1：系统已处于待机模式

位 5 **STOPF**：STOP 标志 (STOP flag)

此位由硬件置 1，并且只能通过任何复位或将 CSSF 位置 1 清零。

0：系统未处于停止模式

1：系统已处于停止模式

位 4:3 保留，必须保持复位值

位 2 **PDDS_D3**：系统 D3 域掉电深度睡眠 (System D3 domain Power Down Deepsleep)
此位可为系统 D3 域定义深度睡眠模式。
0：在 D3 域进入深度睡眠时保持停止模式。
1：在 D3 域进入深度睡眠时允许待机模式。

位 1 **PDDS_D2**：D2 域掉电深度睡眠 (D2 domain Power Down Deepsleep)
此位可为 D2 域定义深度睡眠模式。
0：在 D2 域进入深度睡眠时保持 DStop 模式。
1：在 D2 域进入深度睡眠时允许 DStandby 模式。

位 0 **PDDS_D1**：D1 域掉电深度睡眠选择 (D1 domain Power Down Deepsleep selection)
此位可为 D1 域定义深度睡眠模式。
0：在 D1 域进入深度睡眠时保持 DStop 模式。
1：在 D1 域进入深度睡眠时允许 DStandby 模式。

6.8.6 PWR D3 域控制寄存器 (PWR_D3CR)

PWR D3 domain control register

此寄存器可控制 D3 域的电源。

偏移地址：0x018

复位值：0x0000 4000（复位后，VOSRDY 将通过软件读为 1）。

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

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

位 31:16 保留，必须保持复位值

位 15:14 **VOS**：根据性能选择电压调节 (Voltage scaling selection according to performance)
这些位用于控制 V_{CORE} 电压级别，以在功耗和性能之间实现最佳平衡：
– 提高性能时，应在增大系统频率前更改电压调节。
– 降低性能前，应先减小系统频率，再更改电压调节。
00：保留（选择电压调节 3）
01：电压调节 3（默认）
10：电压调节 2
11：电压调节 1

位 13 **VOSRDY**：V_{CORE} 电压调节输出选择的 VOS 就绪位 (VOS Ready bit for V_{CORE} voltage scaling output selection)。
当在 PWR 控制寄存器 3 (PWR_CR3) 中选择旁路模式时，此位通过硬件置 1。
0：未就绪，电压级别低于 VOS 选择值。
1：已就绪，电压级别等于或高于 VOS 选择值。

位 12:0 保留，必须保持复位值

6.8.7 PWR 唤醒清除寄存器 (PWR_WKUPCR)

PWR wakeup clear register

偏移地址：0x020

复位值：0x0000 0000（仅通过系统复位来复位，而不会通过从待机模式唤醒复位）。

写入此寄存器时需要 5 个等待状态（将 PWR_WKUPFR 中的 WKUPF 位清零时，AHB 写访问将在 WKUPF 清零后完成）。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WKUPC6	WKUPC5	WKUPC4	WKUPC3	WKUPC2	WKUPC1
										rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1

位 31:6 保留，始终读为 0。

位 5:0 **WKUPCn+1**：将 WKUPn+1 唤醒引脚标志清零 (Clear Wakeup pin flag for WKUPn+1)。
此位始终读为 0
0：无影响
1：写 1 会将 WKUPFn+1 唤醒引脚标志清零（此位通过硬件清零）

6.8.8 PWR 唤醒标志寄存器 (PWR_WKUPFR)

PWR wakeup flag register

偏移地址：0x024

复位值：0x0000 0000（仅通过系统复位来复位，而不会通过从待机模式唤醒复位）。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WKUPF6	WKUPF5	WKUPF4	WKUPF3	WKUPF2	WKUPF1
										r	r	r	r	r	r

位 31:6 保留，必须保持复位值

位 5:0 **WKUPn+1**：唤醒引脚 WKUPn+1 标志 (Wakeup pin WKUPn+1 flag)。
此位由硬件置 1，并且只能通过复位引脚或将 *PWR 唤醒清除寄存器 (PWR_WKUPCR)* 中的 WKUPCn+1 位置 1 清零。
0：未发生唤醒事件
1：从 WKUPn+1 引脚接收到唤醒事件

6.8.9 PWR 唤醒使能和极性寄存器 (PWR_WKUPEPR)

PWR wakeup enable and polarity register

偏移地址：0x028

复位值：0x0000 0000（仅通过系统复位来复位，而不会通过从待机模式唤醒复位）。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	WKUPPUPD6		WKUPPUPD5		WKUPPUPD4		WKUPPUPD3		WKUPPUPD2		WKUPPUPD1	
				rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	WKUPP6	WKUPP5	WKUPP4	WKUPP3	WKUPP2	WKUPP1	Res.	Res.	WKUPEN6	WKUPEN5	WKUPEN4	WKUPEN3	WKUPEN2	WKUPEN1
		rw	rw	rw	rw	rw	rw			rw	rw	rw	rw	rw	rw

位 31:28 保留，必须保持复位值

位 27:16 **WKUPPUPD[truncate(n/2)-7)**：WKUP(truncate(n/2)-7) 唤醒引脚拉取配置 (Wakeup pin pull configuration for WKUP(truncate(n/2)-7))

这些引脚用于定义 WKUPEN(truncate(n/2)-7) = 1 时所使用的 I/O 引脚上拉和下拉配置。相关的 GPIO 端口上拉和下拉配置应设置为相同值或“00”。

唤醒引脚的上拉和下拉配置保持待机模式下的配置。

00：无上拉

01：上拉

10：下拉

11：保留

位 15:14 保留，必须保持复位值

位 13:8 **WKUPPn-7**：WKUPn-7 唤醒引脚极性位 (Wakeup pin polarity bit for PA0)

这些位定义用于 WKUPn-7 外部唤醒引脚事件检测的极性。

0：在高电平（上升沿）检测

1：在低电平（下降沿）检测

位 7:6 保留，必须保持复位值

位 5:0 **WKUPENn+1**：使能唤醒引脚 WKUPn+1 (Enable Wakeup Pin WKUPn+1)

各个位由软件置 1 和清零。

0：WKUPn+1 引脚上的事件不会把系统从待机模式唤醒。

1：WKUPn+1 引脚上的上升沿或下降沿将系统从待机模式唤醒。

注： *如果 WKUPn+1 引脚电平在 WKUPPn+1 选择上升沿时已为高电平，或者在 WKUPPn+1 选择下降沿时已为低电平，则当 WKUPn+1 引脚使能时（将 WKUPENn+1 位置 1），会检测到额外的唤醒事件。*

6.8.10 PWR 寄存器映射

表 39. 电源控制寄存器映射和复位值

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x000	PWR_CR1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ALS		AVDEN	SVOS		Res.	Res.	Res.	Res.	FLPS	DBP	PLS			PVDE	Res.	Res.	Res.	LPDS
	Reset value														0	0	0	1	1					0	0	0	0	0	0				0
0x004	PWR_CSR1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	AVDO	ACTVOS		ACTVOSRDY	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PVDO	Res.	Res.	Res.	Res.
	Reset value																0	0	1	0									0				
0x008	PWR_CR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TEMPH	TEMPL	VBATH	VBATL	Res.	Res.	Res.	BRRDY	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MONEN	Res.	Res.	Res.	BREN
	Reset value									0	0	0	0				0												0				0
0x00C	PWR_CR3	Res.	Res.	Res.	Res.	Res.	USB33RDY	USBREGEN	USB33DEN	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	VBRS	VBE	Res.	Res.	Res.	Res.	Res.	SCUEN	LDOEN	BYPASS
	Reset value						0	0	0															0	0						1	1	0
0x010	PWR_CPUCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RUN_D3		CSSF	SBF_D2	SBF_D1	SBF	STOPF	Res.	Res.	PDDS_D3	PDDS_D2	PDDS_D1
	Reset value																					0		0	0	0	0	0			0	0	0
0x014	Reserved	Reserved																															
	Reset value																																
0x018	PWR_D3CR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	VOS		VOSRDY	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	Reset value																	0	1	0													
0x020	PWR_WKUPCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WKUPC6	WKUPC5	WKUPC4	WKUPC3	WKUPC2	WKUPC1
	Reset value																											0	0	0	0	0	0
0x024	PWR_WKUPFR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WKUPF6	WKUPF5	WKUPF4	WKUPF3	WKUPF2	WKUPF1
	Reset value																											0	0	0	0	0	0
0x028	PWR_WKUPEPR	Res.	Res.	Res.	Res.	WKUPPUPD6		WKUPPUPD5		WKUPPUPD4		WKUPPUPD3		WKUPPUPD2		WKUPPUPD1		Res.	Res.	WKUPP6	WKUPP5	WKUPP4	WKUPP3	WKUPP2	WKUPP1	Res.	Res.	WKUPEN6	WKUPEN5	WKUPEN4	WKUPEN3	WKUPEN2	WKUPEN1
	Reset value					0	0	0	0	0	0	0	0	0	0	0	0			0	0	0	0	0	0			0	0	0	0	0	0
0x030	Reserved																																

有关寄存器边界地址的信息，请参见*第 100 页的第 2.2.2 节*。

7 低功耗 D3 域

本部分通过一项示例说明如何使用 D3 域实现低功耗应用。

7.1 前言

说明的第一部分介绍了 EXTI、RCC 和 PWR 模块彼此之间或与其它系统模块如何交互。此外，详细介绍了如何使用 DMAMUX2 释放 CPU。

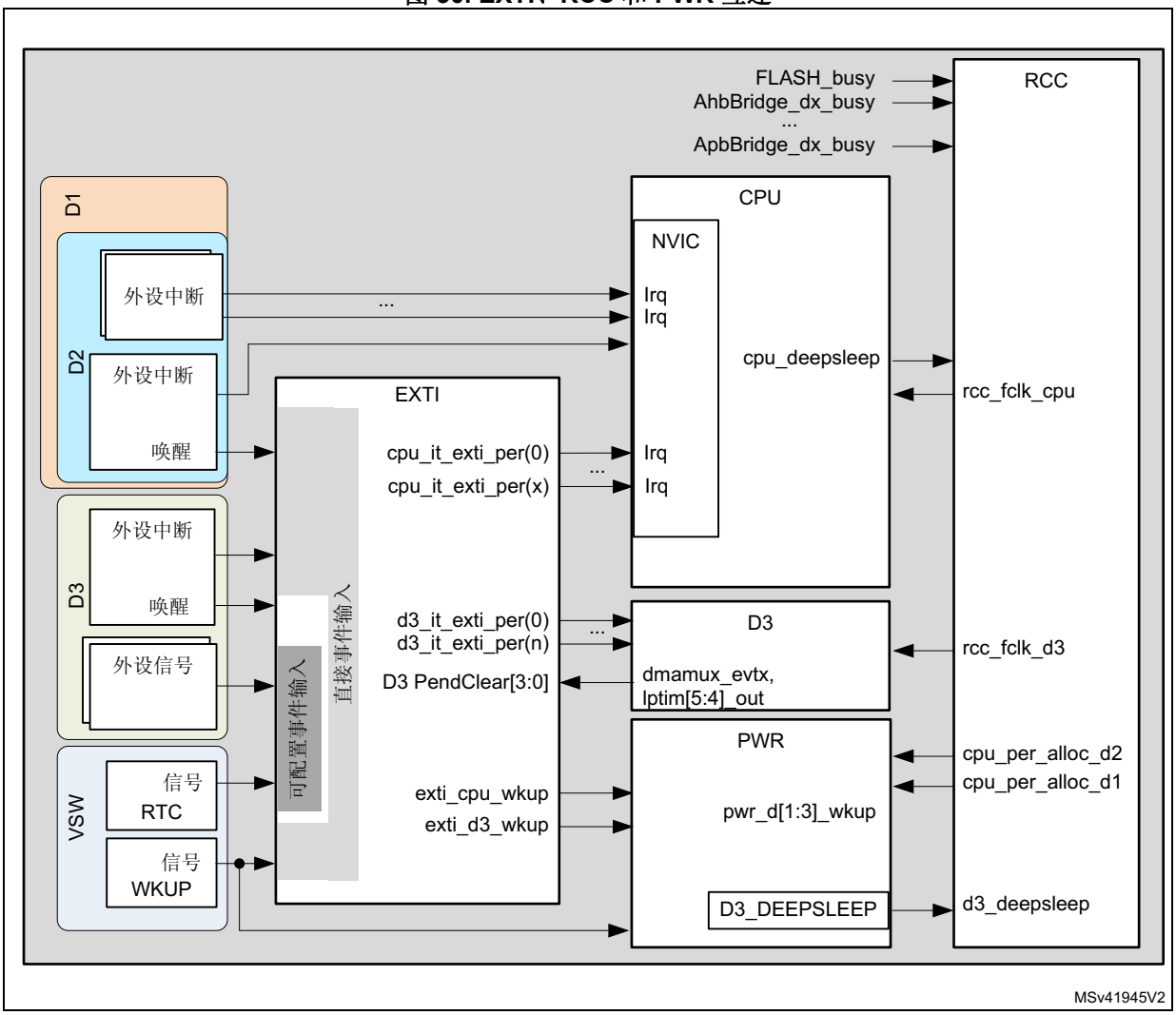
第二部分说明了如何通过 LINUART1 传输示例使用自动模式执行简单数据传输。

寄存器编程的详细介绍仅限与自动模式相关的模块。

7.2 EXTI、RCC 和 PWR 互连

图 30 显示了主要的 EXTI、RCC 和 PWR 互连。

图 30. EXTI、RCC 和 PWR 互连



7.2.1 中断和唤醒

三种信号在外设之间交换。它们用于将系统由停止模式唤醒：

- **唤醒事件**（或异步中断）

  部分外设可生成中断事件,即使其总线接口时钟并不存在。这些中断事件称为唤醒事件（或异步中断）。

  示例：i2c1_wkup、usart1_wkup 和 lptim1_wkup。

- 信号

  部分外设生成脉冲而非中断信号。这些脉冲称为信号。

  示例：lptim2_out 和 lptim3_out。

- 中断

  与信号相反，中断必须由 CPU 或其它总线主器件清除，具体方式是将外设寄存器中的相应事件位清零或更新 FIFO 中断等级。

  除了能够将系统由停止模式唤醒，或将 CPU 由 CStop 模式唤醒的外设外，所有与系统外设相关的中断均直接连接至 NVIC。在后一种情况下，中断、信号或唤醒事件均通过 EXTI 连接至 NVIC。

  示例：spi1_it、tim1_brk_it 和 tim1_upd_it。

需要在外设中自行清除的中断和唤醒源连接至 EXTI 直接事件输入。EXTI 不管理任何 CPU 状态挂起位。

外设信号连接至 EXTI 可配置事件输入。这些 EXTI 输入提供一个 CPU 状态挂起位，需要由应用程序清零。

7.2.2 模块交互

EXTI 和 PWR 模块的交互

EXTI 为 PWR 控制器提供唤醒请求信号（**exti_c_wkup**，**exti_d3_wkup**）。这些信号根据连接至 EXTI 的中断、信号或唤醒事件状态激活。这些唤醒请求通过 PWR 控制器提供给需要处理由外设产生的唤醒事件的电源域。

PWR 和 RCC 模块的交互

PWR 模块根据系统工作模式（CRun、CSleep 或 CStop）控制 V_{CORE} 电源。此外，PWR 模块控制为 D1 和 D2 域提供 V_{CORE} 电源的电源开关 (ePOD)。

RCC 模块根据系统工作模式控制时钟生成。其同样负责复位生成。

为了同步系统模式转换，RCC 模块与 PWR 控制器紧密耦合：

- 当 Dx 域中的外设由 CPU 进行分配后（**c_per_alloc_d2**，**c_per_alloc_d1**），RCC 向 PWR 控制器发出通知。
- 当域时钟处于激活/停用状态时，RCC 也会向 PWR 块发出警告。这些信号用于域由 DRun 转换到 DStop 或 DStandby 的情况。在这种情况下，PWR 控制器会等到域时钟被关断后再关闭该域。
- 相似地，PWR 控制器向 RCC 通知每个域的 V_{CORE} 电源状态 (**pwr_d[1:3]_wkup**)。RCC 在域由 DStop 或 DStandby 转换到 DRun 时使用该信息。

EXTI 与 D3 域的交互

在系统时钟再同步后，EXTI 从 D3 域中所有外设接收的所有唤醒事件输入均转发回 D3 域。D3 域使用这些事件执行自动操作，而无需激活 CPU。

接收自 D3 域的 EXTI **D3_PenClear[3:0]** 输入用于确认由 D3 域中外设生成的正在进行的唤醒命令。**D3_PenClear[3:0]** 输入允许系统 D3 域由运行模式切换到停止模式。

7.2.3 D3 域 DMAMUX2 的角色

DMAMUX2 在 D3 域中实现，允许连接 BDMA 传输。凭借预期数据量传输完成后可能生成的触发事件 (**dmamux2_evtx**)，BDMA 请求实现同步。

这些事件还可触发 DMAMUX2 请求发生器 (REQ_GEN[3:0])，从而连接多个 BDMA 传输。事实上，REQ_GEN[3:0] 可通过所有 D3 域外设生成的所有唤醒事件间接触发。

与 LPTIM5 和 LPTIM4 输出相似，dmamux2_evt7 和 dmamux2_evt6 事件连接至 EXTI。当唤醒事件请求的任务完成后，它们可将 D3 域由 DRun 切换到 DStop 模式。

7.3 基于 LPUART1 发送的低功耗应用示例

本部分通过一个示例展示了 D3 域在功耗方面的优势。为协助用户对器件进行编程，此处仅介绍关键寄存器设置。

其它详细信息，请参见*第 8 节：复位和时钟控制 (RCC)* 和*第 6 节：电源控制 (PWR)*。

7.3.1 存储器保留

D3 域具有 64 KB SRAM (SRAM4)，可用于在 D1 和 D2 域进入 DStandby 模式时保留数据。

该特性可用于多种用例：

- 保留应用程序代码，以从 DStandby 模式正常恢复。
- 保留来自/传输到传感器的数据，适用条件是 CPU 在两次连续操作之间进入 CStop 模式（D1 或 D2 域处于 DStandby 模式）。

注： *只要系统不处于待机模式，便可继续使用 SRAM4。*

如果系统处于待机模式，仍可使用 BKUP_SRAM。然而，其大小限制为 4 KB。

7.3.2 使用 LPUART1 接口的存储器至外设传输

示例说明

图 31 显示了建议的实现方式。在 LPTIM4 提供的常规时间间隔中，CPU 由 CStop 模式唤醒（其域处于 DStandby 模式）。CPU 在处于运行模式时准备通过 LPUART1 传输的数据，然后将其传输至 SRAM4 并返回 CStop 模式。配置 D3 域通过 LPUART1 执行传输数据并在传输完成后返回停止模式。

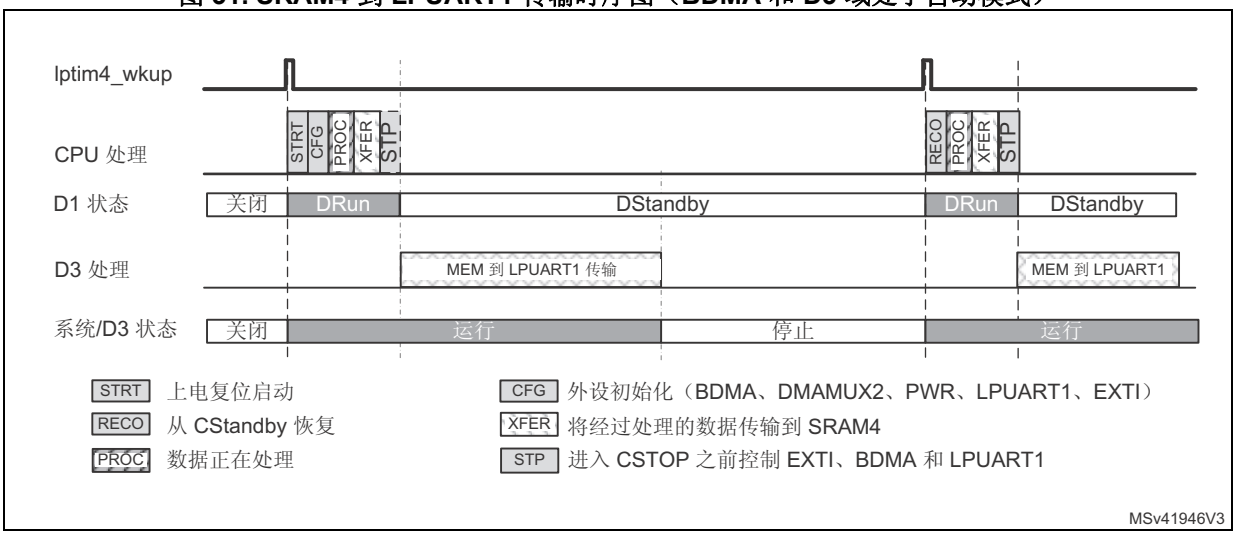
LPTIM4 接口用于在常规时间间隔内将系统由待机模式唤醒。
此后，CPU 必须执行以下操作：

1. 将应用由系统待机模式恢复 (RECO)。
2. 通过 LPUART1 处理待发送的新数据 (PROC)。
3. 将数据传输至 SRAM4 (XFER)。
4. 配置 DMAMUX2、BDMA、LPUART1 和 RCC (CFG)。
5. 配置 EXTI (CFG)。
6. 配置 PWR 模块，以允许 D1 域进入 DStandby (STP) 模式。
7. 将 CPU 设为停止模式。

D3 域在自动模式下执行以下任务：

1. 使用 BDMA 将 SRAM4 中的数据传输至 LPUART1。
2. 当 LPUART1 接口显示最后字节传输完毕，D3 域切换到停止模式。

图 31. SRAM4 到 LPUART1 传输时序图（BDMA 和 D3 域处于自动模式）



注： *在本部分介绍的示例中，当使用 PWR_CPUCR 寄存器的 RUN_D3 位使 D1 和 D2 域处于 DStop/DStandby 模式时，D3 域无法保持运行模式。RUN_D3 强制 D3 域处于运行模式，但其无法自行返回停止模式。*

如果应用需要 D3 域在停止模式和运行模式间切换，则运行模式必须由唤醒事件触发，以便 D3 域可根据需要清除该事件。

RCC 编程

在本示例中，CPU 子系统还包含可用于数据传输的 D3 域外设，即 BDMA、DMAMUX2、LPUART1 和 LPTIM4。这些外设必须在自动模式下编程，目的是即使 CPU 处于 CSTOP 模式依然保持运行。

LPUART1 可使用自身的 APB 时钟作为内核时钟。由于系统在 LPUART1 完成数据传输前将不会进入停止模式，PLLx 可用于为外设提供时钟。

PWR 编程

在本示例中，必须对 PWR 模块进行编程，从而：

- 在数据传输完成后，防止系统 D3 域进入待机模式。
- 允许 D1 域进入 DStandby 模式。
- 根据系统模式定义工作电压。

注： *D3 域也可进入待机模式，但在此情况下，LPTIM4 无法用于唤醒系统，同时应使用 AWU。此外，当系统唤醒时，应对所有事项进行编程。*

EXTI 编程

必须将 EXTI 模块配置为提供以下服务：

- 当 D1 域处于 DStandby 模式，使 D3 域保持运行。这将由软件事件完成。
- 当数据通过 LPUART1 传输完毕后，将器件设置为停止模式。
- 当 LPTIM4 时间间隔结束后，将产品由停止模式唤醒。

在首次执行数据传输前，将 EXTI 模块配置一次。对于传入数据传输，编程的配置保持不变，仅需触发或确认一些事件。

注： *CPU 使用事件输入编号 0 生成软件事件。LPTIM4 唤醒信号连接至事件输入编号 52（直接事件输入）。*

必须禁止其它所有事件输入：EXTI_RTSRx_TRy = “0” 和 EXTI_FTSRx_TRy = “0”。

为了生成 D3 域的唤醒事件，CPU 必须将 EXTI_SWIER1 的 SWIER0 位写为“1”。连接至本事件输入的 GPIO 禁止切换，避免意外唤醒。为防止 GPIO 产生干扰，可按以下顺序执行操作：

BDMA 和 DMAMUX2 编程

通过 LPUART1 执行数据传输需要使用两条 BDMA 通道。

- 第一条 BDMA 通道，例如通道 0，用于将数据由 SRAM4 传输到 LPUART1（使用 TXE 标志）。
- 第二条 BDMA 通道的作用是将 D3 域转换为停止模式。为此，DMAMUX2 请求发生器通道 0 (REQ_GEN0) 和 DMAMUX2 通道 7 同步块 (SYNC7) 与 BDMA 通道 7 结合使用。

BDMA 通道 0 不使用 DMAMUX2 触发功能。有关初始化的详细信息，请参见*表 40*。

BDMA 通道 7 使用 REQ_GEN0 生成 BDMA 请求。BDMA 请求的生成由 LPUART1 发送中断 (**lpuart1_tx_it**) 触发。检测到发送完成事件后，LPUART1 接口生成 lpuart1_tx_it 中断。此后，BDMA 通过对 LPUART1 执行写入操作清除挂起中断。

SYNC7 块在自由运行模式下进行编程。当 REQ_GEN0 生成的 BDMA 请求完成后，SYNC7 块在其 **dmamux2_evt7** 输出中生成一个脉冲。EXTI 使用 **dmamux2_evt7** 将 D3 域切换回停止模式。

图 32 显示经过 DMAMUX2 的有效信号路径。灰色块表示未使用的路径。

图 32. BDMA 和 DMAMUX2 互连

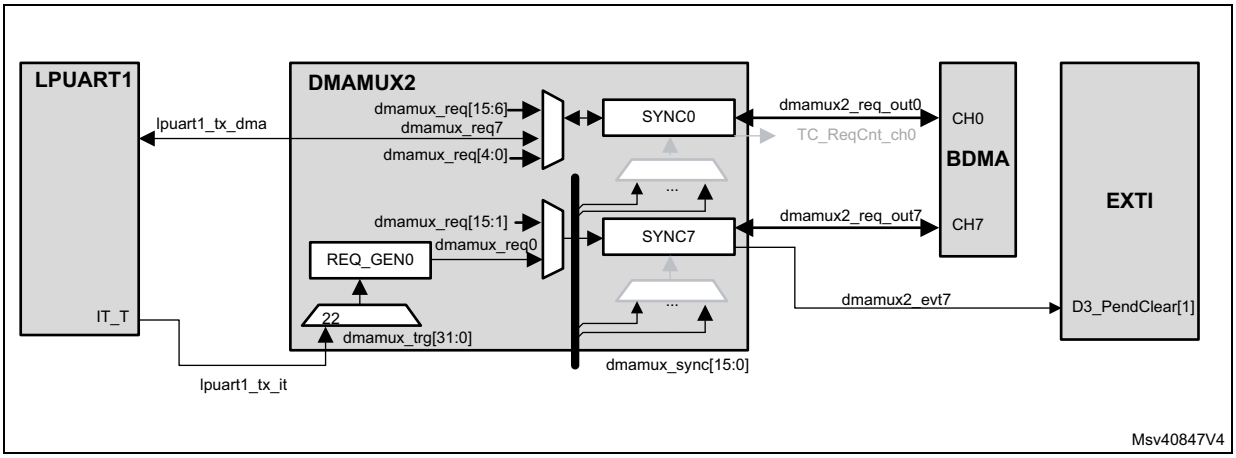


表 40 介绍了 BDMA 和 DMAMUX2 关键功能的编程方式。未介绍错误处理方式。

表 40. BDMA 和 DMAMUX2 初始化序列 (DMAMUX2_INIT)

外设	寄存器内容	相关操作
DMAMUX2 SYNC0	DMAMUX2_C0CR 的 DMAREQ_ID = “7” DMAMUX2_C0CR 的 SE = “0” DMAMUX2_C0CR 的 EGE = “0” DMAMUX2_C0CR 的 NBREQ = “0”	选择 LPUART_TX BDMA 请求。 禁止块同步。 无事件产生。 在每次 BDMA 传输过程中均生成一个事件（自由运行模式）。
DMAMUX2 SYNC7	DMAMUX2_C0CR 的 DMAREQ_ID = “0” DMAMUX2_C7CR 的 SE = “0” DMAMUX2_C7CR 的 EGE = “1” DMAMUX2_C7CR 的 NBREQ = “0”	选择 REQ_GEN0 作为 BDMA 请求。 禁止块同步。 使能事件生成。 在每次 BDMA 传输过程中均生成一个事件（自由运行模式）。
DMAMUX2 REQ_GEN0	DMAMUX2_RG0CR 的 SIG_ID = “0d24” DMAMUX2_RG0CR 的 GPOL = “0b01” DMAMUX2_C7CR 的 NBREQ = “0” DMAMUX2_C7CR 的 NBREQ = “1”	选择 LPUART TX 中断作为触发。 在事件的上升沿触发。 仅生成一个 BDMA 请求。 使能发生器。

表 40. BDMA 和 DMAMUX2 初始化序列 (DMAMUX2_INIT)（续）

外设	寄存器内容	相关操作
BDMA-CH0	BDMA_CNDTR0 的 NDT 位 = DatNber	待传输数据数量。
	BDMA_CPAR0 的 PA = &LPUART1_TDR	LPUART1_TDR 地址。
	BDMA_CMAR0 的 MA = &DatBuff	SRAM4 存储器缓冲区地址。
	BDMA_CCR0 的 DIR = “1”	从存储器读取。
	BDMA_CCR0 的 CIRC = “0”	禁止循环模式。
	BDMA_CCR0 的 PINC = “0”	禁止外设递增。
	BDMA_CCR0 的 MINC = “1”	使能存储器递增
	BDMA_CCR0 的 PSIZE = “0”	外设大小 = 8 位
	BDMA_CCR0 的 MSIZE = “1”	存储器大小 = 8 位
	BDMA_CCR0 的 MEM2MEM = “0”	禁止存储器到存储器模式
BDMA-CH7	BDMA_CNDTR7 的 NDT 位 = “1”	仅传输一个数据。
	BDMA_CPAR7 的 PA = &LPUART1_ICR	LPUART1_ICR 地址（中断标志清除寄存器）。
	BDMA_CMAR7 的 MA = &DatClrTC	SRAM4 中的变量地址。该变量必须包含 0x0040，以清零 TC 标志。
	BDMA_CCR7 的 DIR = “1”	从存储器读取。
	BDMA_CCR7 的 CIRC = “0”	禁止循环模式。
	BDMA_CCR7 的 PINC = “0”	禁止外设递增。
	BDMA_CCR7 的 MINC = “1”	禁用存储器递增。
	BDMA_CCR7 的 PSIZE = 2	外设大小 = 32 位。
	BDMA_CCR7 的 MSIZE = 2	存储器大小 = 32 位。
	BDMA_CCR7 的 MEM2MEM = “0”	禁止存储器到存储器模式。

LPTIM4 编程

发生 LPTIM4 唤醒事件时，CPU 重新启动，D3 域模式同时设置为运行模式。

由 LPTIM4 发出的中断在 CPU NVIC 中挂起。LPTIM4 中断处理程序必须通过将 LPTIM4_ICR 寄存器的 ARRMCF 位写入“1”确认该 LPTIM4 中断 (LPTIM4_Ack)。

LPUART 编程

在此处介绍的用例中，未使用 LPUART1 根据某些事件请求内核时钟的功能。

针对 LPUART1 进行编程，从而在其 TX-FIFO 未满时生成 BDMA 请求。

当 TX-FIFO 及其发送移位寄存器均为空时，LPUART1 同样生成中断。该中断用于将 D3 域切换到停止模式。

表 41 给出了有关 LPUART1 处理停止模式的关键设置。

表 41. LPUART1 初始编程 (LPUART1_INIT)

寄存器内容	相关操作
LPUART1_CR1 的 FIFOEN = “1”	使能 FIFO。此后，BDMA 将使用 TXFNF（TXFIFO 未满）标志生成 BDMA 请求。
LPUART1_CR1 的 TCIE = “0”	当发送缓冲区为空时，禁止中断。
LPUART1_CR1 的 UE = “1”	使能 BDMA。
LPUART1_CR1 的 TE = “1”	使能 LPUART1。
LPUART1_CR1 的 TXE = “1”	使能传输。
LPUART1_CR3 的 DMAT = “1”	使能 BDMA 模式进行传输。

遵循*表 42* 中介绍的顺序使能 LPUART1。

表 42. LPUART1 初始编程 (LPUART1_Start)

寄存器内容	相关操作
LPUART1_ICR 的 TCCF = “1”	清零 TC 标志，避免生成即时中断。该中断会将 EXTI 中的 D3_PendClear[1] 清零。
LPUART1_CR1 的 TCIE = “1”	当发送缓冲区为空时，使能中断。

7.3.3 基于 LPUART1 发送的低功耗应用示例总体说明

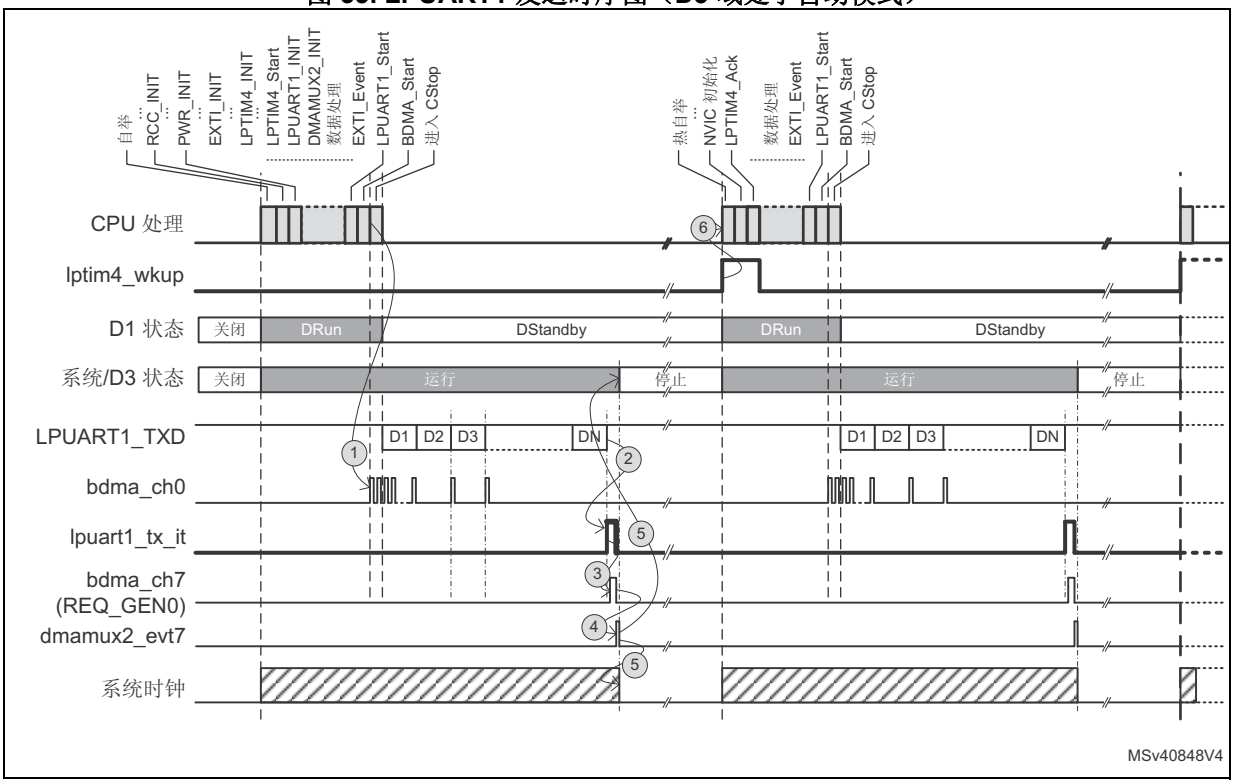
上电复位后，CPU 执行以下操作：

1. 启动序列（此处不进行介绍）。
2. RCC、PWR、EXTI、LPUART1、GPIO、LPTIM4、DMAMUX2、BDMA 和 NVIC 的完全初始化。
   此处仅介绍与自动模式相关的 RCC、EXTI、PWR、LPUART1、BDMA 和 DMAMUX2 初始化的关联步骤。其它详细信息，请参见先前的章节。
3. CPU 处理待传输的数据并将其复制到 SRAM4。
4. CPU 在 D1 进入 DStandby 模式时生成唤醒事件 (EXTI_Event)，维持 D3 处于运行模式。
5. CPU 使能 BDMA，以启动 LPUART 传输并进入停止模式。正如允许执行的操作，当 D3 保持运行模式时，D1 域进入 DStandby 模式。当 D1 域处于 DStandby 模式时，SRAM4 中储存的数据将会保留。
6. BDMA 使能后可处理 LPUART1 发出的请求，从而填充其 TX-FIFO。与此同时，可启动串行数据传输。
7. 预期数据量发送完毕后（BDMA_CNDTR0 的 NDT 位置 0），BDMA 将不再为 LPUART1 提供数据。当 TX-FIFO 和发送缓冲区均为空时，LPUART1 生成一个中断。
8. 该中断触发 DMAMUX2 REQ_GEN0，从而激活通过 BDMA 通道 7 (BDMA_Ch7) 进行的数据传输。该传输清零 LPUART1 TC 标志，**lpuart1_tx_it** 复位为“0”。
9. 该传输结束后将触发 dmamux2_evt7 信号。该信号用于清除 CPU 生成的唤醒请求。
10. 因此，D3 域（即该系统）进入停止模式且系统时钟进行门控。由于 LPTIM4 使用 **ck_lsi** 时钟，因此仍在运行。

11. LPTIM4 **lptim4_wkup** 中断唤醒系统。该器件通过 HSI 时钟退出停止模式。CPU 必须在热重启序列中恢复恰当的时钟配置并执行以下任务：
    d) 确认 LPTIM4 唤醒中断
    e) 处理下一数据块并将其传输到 SRAM4
    f) 再次为 D3 域生成唤醒事件
    g) 启动 BDMA
    h) 返回 CStop 模式

注： *CPU 无需再次初始化 BDMA、DMAMUX2 和 LPUART1。*

图 33. LPUART1 发送时序图（D3 域处于自动模式）



7.3.4 备选实现方式

电源效率更高的实现方式同样可行。例如，当数据传输到 LPUART1 TX-FIFO 后，系统时钟随即停止，而非按上一示例所述，系统在整个传输过程中始终处于激活状态。在这种情况下，当系统由运行模式切换到停止模式时，LPUART1 必须使用 ck_hsi or ck_csi 作为内核时钟。当 LPUART1 的 TX-FIFO 几乎为空时，必须将其编程为唤醒 D3 域。该异步中断可由 DMAMUX2 的 REQ_GENx 触发，可针对 LPUART1_TDR 执行给定次数的数据传输（例如 14），然后将 D3 域切换回停止模式。这种实现方式可行，原因是只要 TX-FIFO 与发送缓冲区不为空，LPUART1 即可请求内核时钟。

7.4 其它低功耗应用

D3 域中的其它外设，如 I2C4、SPI6、SAI4 或 ADC3，可用于实现低功耗应用。

8 复位和时钟控制 (RCC)

RCC 模块管理整个微控制器的时钟和复位信号的生成。

RCC 模块位于 D3 域中（有关详细说明，请参见*第 6 节：电源控制 (PWR)*）。

本部分涉及的工作模式在 PWR 模块的*第 6.6.1 节：工作模式*中定义。

8.1 RCC 主要特性

复位模块

- 生成本地和系统复位信号
- 双向引脚复位，可复位微处理器或外部器件
- 保持启动功能
- 支持 WWDG 复位

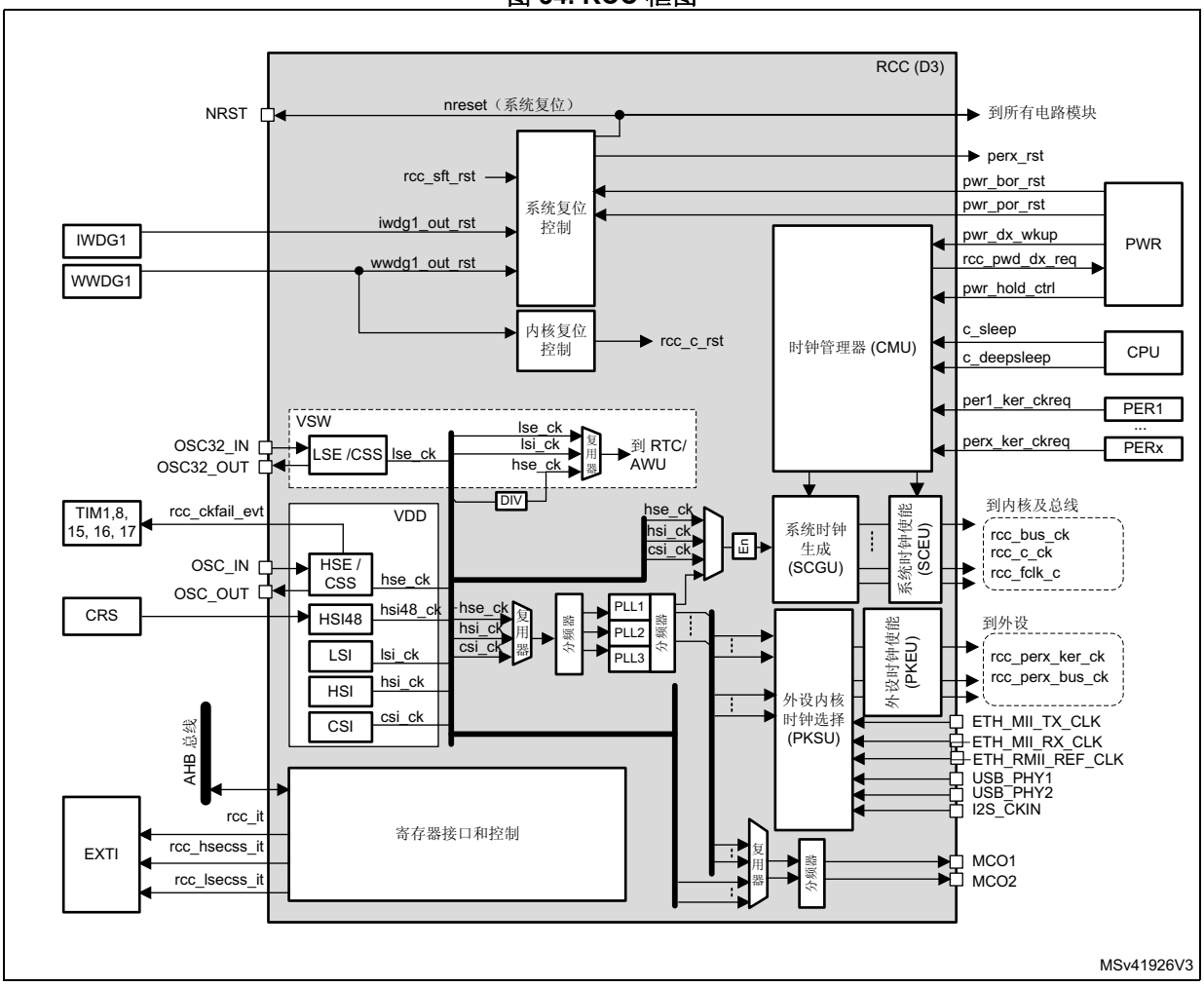
时钟生成模块

- 针对整个器件生成和调度时钟
- 3 个使用整数或小数分频比的独立 PLL
- 可实时更改 PLL 小数分频比
- 智能时钟门控，可降低功耗
- 2 个外部振荡器：
  - 高速外部振荡器 (HSE)，支持 4 MHz 到 48 MHz 频率范围内的晶振
  - 适用于 32 kHz 晶振的低速外部振荡器 (LSE)
- 4 个内部振荡器
  - 高速内部振荡器 (HSI)
  - 48 MHz RC 振荡器 (HSI48)
  - 低功耗内部振荡器 (CSI)
  - 低速内部振荡器 (LSI)
- 针对外部器件的缓冲时钟输出
- 生成两种类型的中断线：
  - 多条用于时钟安全管理的专用中断线
  - 一条用于其他事件的通用中断线
- 在停止和待机模式下的时钟生成处理
- D3 域自主模式

8.2 RCC 框图

图 34 显示了 RCC 框图。

图 34. RCC 框图



8.3 RCC 引脚和内部信号

表 43 列出了连接到封装引脚或焊球的 RCC 输入与输出信号。

表 43. 连接到封装引脚或焊球的 RCC 输入/输出信号

信号名称	信号类型	说明
NRST	I/O	系统复位，可用于为外部器件提供复位信号
OSC32_IN	I	32 KHz 振荡器输入
OSC32_OUT	O	32 KHz 振荡器输出
OSC_IN	I	系统振荡器输入

表 43. 连接到封装引脚或焊球的 RCC 输入/输出信号（续）

信号名称	信号类型	说明
OSC_OUT	O	系统振荡器输出
MCO1	O	外部器件的输出时钟 1
MCO2	O	外部器件的输出时钟 2
I2S_CKIN	I	数字音频接口的外部内核时钟输入：SPI/I2S、SAI 和 DFSDM
ETH_MII_TX_CLK	I	以太网 MII 接口提供的外部 TX 时钟
ETH_MII_RX_CLK	I	以太网 MII 接口提供的外部 RX 时钟
ETH_RMII_REF_CLK	I	以太网 RMII 接口提供的外部参考时钟
USB_PHY1	I	外部 USB PHY 提供的 USB 时钟输入
USB_PHY2	I	外部 USB PHY 提供的 USB 时钟输入

RCC 会与产品的所有组件交换大量内部信号，因此，*表 43* 仅给出了最重要的几个内部信号。

表 44. RCC 内部输入/输出信号

新信号名称	信号类型	说明
rcc_it	O	通用中断请求线。
rcc_hsecss_it	O	HSE 时钟安全故障中断。
rcc_lsecss_it	O	LSE 时钟安全故障中断。
rcc_ckfail_evt	O	指示检测到 HSE 时钟安全故障的事件。该信号连接到定时器。
nreset	I/O	系统复位。
iwdg1_out_rst	I	由 IWDG1 驱动的复位线，指示已发生超时。
wwdg1_out_rst	I	由 WWDG1 驱动的复位线，指示已发生超时。
pwr_bor_rst	I	由 PWR 模块生成的欠压复位信号。
pwr_por_rst	I	由 PWR 模块生成的上电复位信号。
pwr_vsw_rst	I	由 PWR 模块生成的 VSW 域上电复位信号。
rcc_perx_rst	O	RCC 针对外设生成的复位信号。
pwr_d[3:1]_wkup	I	由 PWR 生成的唤醒域请求。通常用于在域退出 DStop 模式时恢复该域的时钟。
rcc_pwd_d[3:1]_req	O	由 RCC 生成的低功耗请求。通常用于在域处于 DStop 模式时让 PWR 将域置于低功耗模式。
pwr_hold_ctrl	I	由 PWR 生成的信号，用于在退出系统停止模式时将处理器置于 CStop 模式。
c_sleep	I	由 CPU 生成的信号，用于指示 CPU 是处于 CRun、CSleep 还是 CStop 模式。
c_deepsleep	I	

表 44. RCC 内部输入/输出信号（续）

新信号名称	信号类型	说明
perx_ker_ckreq	I	由某些外设生成的信号，用于请求激活其内核时钟。
rcc_perx_ker_ck	O	由 RCC 针对某些外设生成的内核时钟信号。
rcc_perx_bus_ck	O	由 RCC 针对外设生成的总线接口时钟信号。
rcc_bus_ck	O	由 RCC 生成的 APB 桥时钟 (rcc_apb_ck)、AHB 桥时钟 (rcc_ahb_ck) 和 AXI 桥时钟 (rcc_axi_ck)。
rcc_c_ck	O	由 RCC 生成的 CPU 时钟。
rcc_fclk_c	O	

8.4 RCC 复位模块功能说明

复位信号源有以下几种：

- 外部器件（通过 NRST 引脚）
- V_{DD} 电源电压故障
- 看门狗超时
- 软件命令

复位范围取决于复位信号源。共有三种复位类型：

- 上电/掉电复位
- 系统复位
- 本地复位

8.4.1 上电/掉电复位

上电/掉电复位 (**pwr_por_rst**) 由电源控制器模块 (PWR) 生成。它在输入电压 (V_{DD}) 低于阈值电压时激活。此为最完整的复位，因为其可将除备份域外的整个电路复位。上电/掉电复位功能可通过 PDR_ON 引脚禁用（请参见*第 6.5 节：电源监控*）。

有关详细信息，请参见*表 45：复位分配汇总*。

8.4.2 系统复位

系统复位 (**nreset**) 可将所有寄存器均复位为其默认值，但 RCC_RSR（或 RCC_C1_RSR）寄存器中的复位状态标志、调试功能、Flash 和备份域寄存器除外。

系统复位信号源有以下几种：

- 通过 NRST 引脚复位（外部复位）
- 通过上电/掉电复位模块 (**pwr_por_rst**) 复位
- 通过欠压复位模块 (**pwr_bor_rst**) 复位

  有关 BOR 功能的详细说明，请参见*第 6.5.2 节：欠压复位 (BOR)*。
- 通过独立看门狗 (**iwdg1_out_rst**) 复位
- 通过 Cortex®-M7 内核实现软件复位

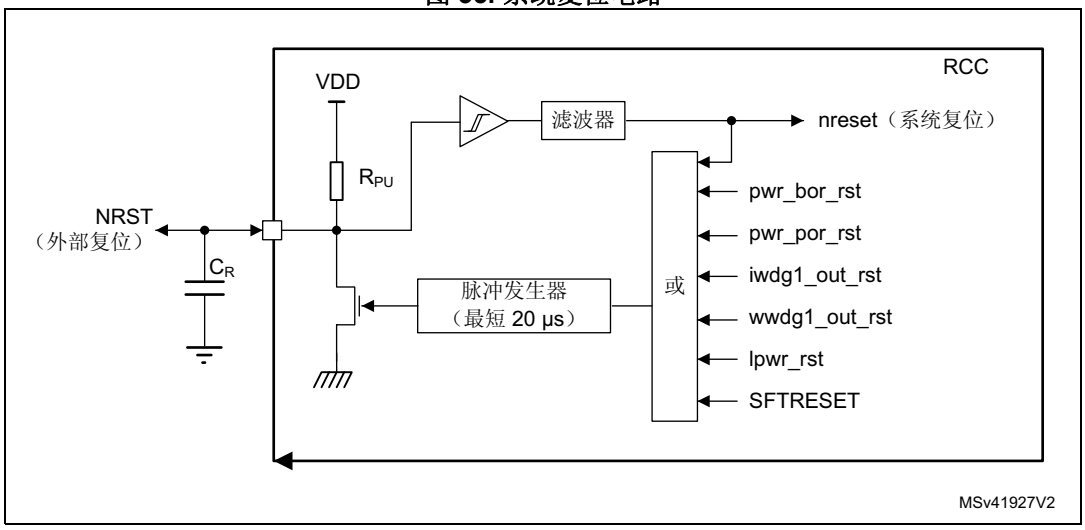
  此复位由 Cortex®-M7 内核发出的 SYSRESETREQ 信号生成。在本文档中，该信号还被称为 SFTRESET。
- 通过窗口看门狗复位，具体取决于 WWDG 配置 (**wwdg1_out_rst**)
- 通过低功耗模式安全复位实现复位，具体取决于选项字节配置 (**lpwr[2:1]_rst**)

注： *要对器件进行软件复位，必须将 Cortex®-M7 应用中断和复位控制寄存器中的 SYSRESETREQ 位置 1。更多详细信息，请参见带 FPU 的 Cortex®-M7 技术参考手册（请访问* http://infocenter.arm.com *）。*

如*图 35* 所示，某些内部复位源（例如 **pwr_por_rst**、**pwr_bor_rst**、**iwdg1_out_rst**）可执行电路的系统复位，这也会传播到 NRST 引脚以复位其连接的外部器件。脉冲发生器可确保每个内部复位源的复位脉冲都至少持续 20 µs。对于外部复位，在 NRST 引脚处于低电平时产生复位脉冲。

注： *不建议使 NRST 引脚处于未连接状态。如果未使用该引脚，则可通过一个容值介于* 10 nF 到 100 nF *之间的电容（图 35 中的 C_R）将其接地。*

图 35. 系统复位电路



8.4.3 本地复位

CPU 复位

CPU 可通过 *RCC AHB3 复位寄存器 (RCC_AHB3RSTR)* 中的 CPURST 位自行复位。

域复位

某些复位还取决于域状态。例如，如果 D1 域退出 DStandby 模式，则将发生复位 (**d1_rst**)。该机制对于 D2 同样适用。

系统退出待机模式时，将发生 **stby_rst** 复位。只要内部稳压器提供的 V_{CORE} 电压无效，**stby_rst** 信号就会生成针对整个 V_{CORE} 域的复位信号。

表 45 给出了复位源及其适用范围的详细说明。

表 45. 复位分配汇总

复位源	复位名称	D1 CPU	D1 互连	D1 外设	D1 调试	WWDG1	D2 互连	D2 外设	D3 外设	IWDG1	闪存	RTC 域	备份 RAM	系统电源	NRST 引脚	注释
引脚	NRST	x	x	x	-	x	x	x	x	x	-	-	-	-	x	– 复位 D1 和 D2 域及其所有外设 – 复位 D3 域外设 – 复位 V_{DD} 域：IWDG1、LDO... – 不复位调试功能、Flash、RTC 和备份 RAM
PWR	pwr_bor_rst	x	x	x	-	x	x	x	x	x	-	-	-	-	x	– 与引脚复位相同。也会使能相应引脚。
	pwr_por_rst	x	x	x	x	x	x	x	x	x	x	-	-	x	x	– 与 **pwr_bor_rst** 复位相同，另外： 还会复位 Flash 数字块（包括选项字节加载）。 复位调试块
	lpwr_rst	x	x	x	-	x	x	x	x	x	-	-	-	-	x	– 低功耗模式安全复位的适用范围与 **pwr_por_rst** 相同。 有关更多信息，请参见*第 8.4.5 节：低功耗模式安全复位 (lpwr_rst)*。

表 45. 复位分配汇总（续）

复位源	复位名称	D1 CPU	D1 互连	D1 外设	D1 调试	WWDG1	D2 互连	D2 外设	D3 外设	IWDG1	闪存	RTC 域	备份 RAM	系统电源	NRST 引脚	注释
RCC	BDRST	-	-	-	-	-	-	-	-	-	-	x	-	-	-	– 备份域复位可由软件触发。有关更多信息，请参见*第 8.4.6 节：备份域复位*。
	d1_rst	x	x	x	x	x	-	-	-	-	-	-	-	-	-	– 在 D1 域退出 DStandby 模式时复位该域及其所有外设。
	d2_rst	-	-	-	-	-	x	x	-	-	-	-	-	-	-	– 在 D2 域退出 DStandby 模式时复位该域及其所有外设。
	stby_rst	x	x	x	x	x	x	x	x	-	-	-	-	-	-	– 当器件退出待机模式时，只要 V_{CORE} 电压无效，就会对整个 V_{CORE} 域执行复位。V_{CORE} 由内部稳压器提供。 不触发 NRST 信号。
	CPURST	x	-	-	-	x	-	-	-	-	-	-	-	-	-	– 该复位由软件通过 *RCC AHB3 复位寄存器 (RCC_AHB3RSTR)* 中的位生成。 – 复位 CPU 和 WWDG1 模块。
CPU	SFTRESET	x	x	x	-	x	x	x	x	x	-	-	-	-	x	– 该复位在写入 Cortex®-M7 内核的 AIRCR 寄存器中的 SYSRESETREQ 位时由软件生成。 – 适用范围与 **pwr_bor_rst** 复位相同。
备份域	pwr_vsw_rst	-	-	-	-	-	-	-	-	-	-	x	-	-	-	– 该复位在 V_{SW} 电源电压超出工作范围时由备份域生成。
IWDG1	iwdg1_out_rst	x	x	x	-	x	x	x	x	x	-	-	-	-	x	– 与 **pwr_bor_rst** 复位相同。
WWDG1	wwdg1_out_rst	x	x	x	-	x	x	x	x	x	-	-	-	-	x	– 与 **pwr_bor_rst** 复位相同。

8.4.4 复位源标识

CPU 可通过检查 RCC_RSR（或 RCC_C1_RSR）寄存器中的复位标志来识别复位源。

CPU 可通过将 RMVF 位置 1 来复位标志。

表 46 给出了不同复位源条件下 RCC_RSR （或 RCC_C1_RSR）寄存器状态位的值。例如，发生 IWDG1 超时时（第 10 行），如果 CPU 正在启动阶段读取 RCC_RSR（或 RCC_C1_RSR）寄存器，则 PINRSTF 和 IWDG1RSTF 位都会置 1，以指示 IWDG1 也生成了一个引脚复位信号。

表 46. 复位源标识 (RCC_RSR)(1)

#	生成复位的条件	LPWRRSTF	WWDG1RSTF	IWDG1RSTF	SFTRSTF	PORRSTF	PINRSTF;	BORRSTF	D2RSTF	D1RSTF	CPURSTF
1	上电复位 (pwr_por_rst)	0	0	0	0	1	1	1	1	1	1
2	引脚复位 (NRST)	0	0	0	0	0	1	0	0	0	1
3	欠压复位 (pwr_bor_rst)	0	0	0	0	0	1	1	0	0	1
4	CPU 生成的系统复位 (SFTRESET)	0	0	0	1	0	1	0	0	0	1
5	CPU 复位 (CPURST)	0	0	0	0	0	0	0	0	0	1
6	WWDG1 复位 (wwdg1_out_rst)	0	1	0	0	0	**1**	0	0	0	**1**
8	IWDG1 复位 (iwdg1_out_rst)	0	0	1	0	0	1	0	0	0	1
10	D1 退出 DStandby 模式	0	0	0	0	0	0	0	0	1	0
11	D2 退出 DStandby 模式	0	0	0	0	0	0	0	1	0	0
12	D1 错误地进入 DStandby 模式或 CPU 错误地进入 CStop 模式	1	0	0	0	0	1	0	0	0	1

1. 灰色单元格突出显示了置 1 的寄存器位。

8.4.5 低功耗模式安全复位 (lpwr_rst)

为了防止关键应用错误地进入低功耗模式，提供了两种低功耗模式安全复位。通过 nRST_STOP_D1 选项字节使能时，如果满足以下条件则会生成系统复位：

- CPU 意外进入 CStop 模式

  可通过复位 nRST_STOP_D1 用户选项字节来使能此类复位。在这种情况下，只要成功执行 CPU 进入 CStop 模式序列，就会生成系统复位。

- D1 域意外进入 DStandby 模式

  可通过复位 nRST_STDBY_D1 用户选项字节来使能此类复位。在这种情况下，只要成功执行 D1 域进入 DStandby 模式序列，就会生成系统复位。

RCC 复位状态寄存器 (RCC_RSR) 中的 LPWRRSTF 位指示发生了低功耗模式安全复位（请参见*表 46* 的第 12 行）。

发生因 D1 或 CPU 导致的低功耗模式安全复位时，将激活 **lpwr_rst**。

有关更多信息，请参见*第 3.3.11 节：FLASH 选项字节*。

8.4.6 备份域复位

只要发生以下事件之一，就会产生备份域复位：

- 软件复位，通过将 *RCC 备份域控制寄存器 (RCC_BDCR)* 中的 BDRST 位置 1 触发。所有 RTC 寄存器和 RCC_BDCR 寄存器均复位为默认值。备份 RAM 不受影响。
- V_{SW} 电压超出工作范围。所有 RTC 寄存器和 RCC_BDCR 寄存器均复位为默认值。在这种情况下，备份 RAM 的内容不再有效。

备份 RAM 有两种复位方式：

- 通过 Flash 接口请求将保护级别从 1 更改为 0 的方式
- 发生入侵事件时。

有关详细信息，请参见 PWR 模块的*第 6.4.4 节：备份域*部分。

8.4.7 上电和唤醒序列

有关详细时序图，请参见 PWR 部分的*第 6.4.1 节：系统电源启动*。

从产品退出低功耗模式到 CPU 能执行代码的时间间隔取决于系统状态及其配置。*图 36* 所示为最常见的示例。

上电唤醒序列

图 36 所示的上电唤醒序列给出了上电序列的几个最重要阶段。由于电路未通电，因此这是最长的序列。请注意，无论是否存在 V_{BAT}，该序列都保持不变。

通过引脚复位启动

发生引脚复位时，仍存在 V_{DD}。因此：

- 参考电压已稳定，稳压器稳定时间变短。
- 如果在发生 NRST 时未使能 HSI，则可能需要 HSI 重启延时，否则会跳过此重启延时阶段。
- 如果在发生 NRST 时使能了 Flash，则还会跳过 Flash 供电恢复延时。

注： *pwr_bor_rst、lpwr_rst、STFxRESET、iwdg1_out_rst 和 wwdg1_out_rst（WW1RSC = '1' 时）的启动序列与此类似。*

从系统待机模式启动

从系统待机模式唤醒时，V_{DD} 并不移除，因此参考电压保持稳定。因此，稳压器稳定时间较短。由于不存在 V_{CORE}，因此不能跳过 HSI、Flash 供电恢复和选项字节重载的重启延时。

通过系统停止模式重启

从系统停止模式重启时，仍存在 V_{DD}。因此，该序列主要包括两步：

1. 为达到 VOS3（默认电压）而需要的稳压器稳定时间。
2. HSI/CSI 重启延时。如果 *RCC 源控制寄存器 (RCC_CR)* 中的 HSIKERON 或 CSIKERON 位置“1”，则可跳过该步。

从域 DStandby 模式启动

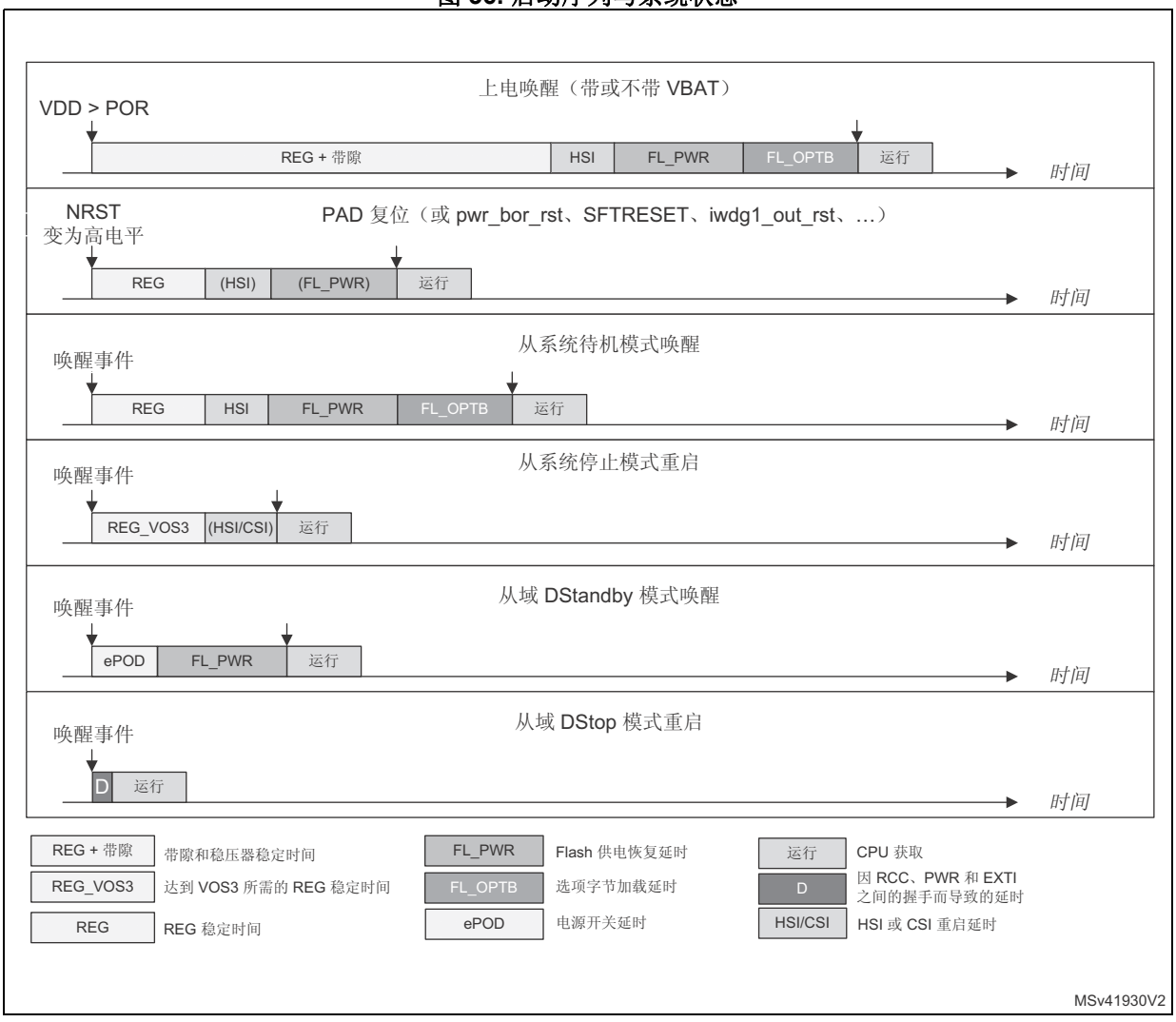
域从域 DStandby 模式启动的序列主要包括两步：

1. 电源开关稳定时间（稳压器已激活）。
2. Flash 恢复供电。

从域 DStop 模式重启

域从域 DStop 模式重启的序列主要包括 RCC、EXTI 和 PWR 模块之间的握手。

图 36. 启动序列与系统状态



8.5 RCC 时钟模块功能说明

RCC 有多种时钟发生器可供选择：

- HSI（高速内部振荡器）时钟：~ 8 MHz、16 MHz、32 MHz 或 64 MHz
- HSE（高速外部振荡器）时钟：4 MHz 到 48 MHz
- LSE（低速外部振荡器）时钟：32 kHz
- LSI（低速内部振荡器）时钟：~ 32 kHz
- CSI（低功耗内部振荡器）时钟：~ 4 MHz
- HSI48 （高速 48 MHz 内部振荡器）时钟：~48 MHz

该模块使得应用能够极其灵活地为 CPU 和外设（尤其是需要特定时钟的外设，例如，以太网、USB OTG-FS 和 HS、SPI/I2S、SAI 以及 SDMMC）选择合适的时钟。

为优化功耗，每个时钟源都可单独打开或关闭。

RCC 提供多达 3 个 PLL，每个 PLL 均可采用整数或小数分频比配置。

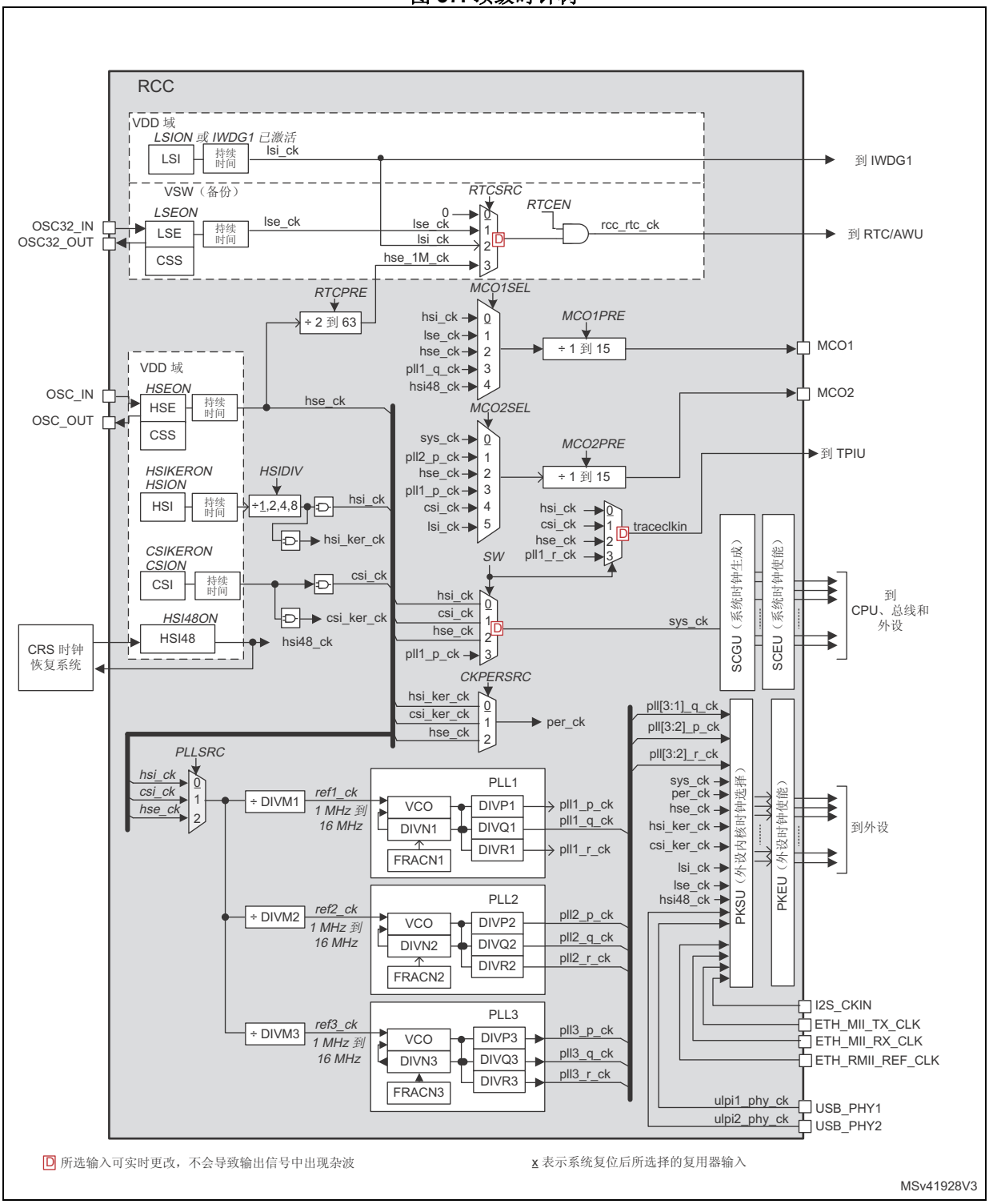
如*图 37* 所示，RCC 具有 2 个时钟输出（MCO1 和 MCO2），可非常灵活地进行时钟选择和频率调节。

SCGU 模块（系统时钟生成单元）包含多个预分频器，用于配置 CPU 和总线矩阵的时钟频率。

PKSU 模块（外设内核时钟选择单元）配有多个动态开关，支持丰富的外设内核时钟分配选项。

PKEU（外设内核时钟使能单元）和 SCEU（系统时钟使能单元）模块分别用于执行外设内核时钟门控和总线接口/内核/总线矩阵时钟门控。

图 37. 顶级时钟树


RCC
VDD 域
LSION 或 IWDG1 已激活
LSI
持续时间
lsi_ck
到 IWDG1
VSW（备份）
RTCSRC
RTCEN
LSEON
OSC32_IN
OSC32_OUT
LSE
CSS
lse_ck
rcc_rtc_ck
到 RTC/AWU
hse_1M_ck
RTCPRE
÷ 2 到 63
MCO1SEL
MCO1PRE
÷ 1 到 15
MCO1
hsi_ck
hse_ck
pll1_q_ck
hsi48_ck
OSC_IN
OSC_OUT
HSEON
MCO2
MCO2SEL
MCO2PRE
到 TPIU
sys_ck
pll2_p_ck
pll1_p_ck
csi_ck
lsi_ck
traceclkin
HSIKERON
HSION
HSI
HSIDIV
÷1,2,4,8
hsi_ker_ck
CSIKERON
CSION
CSI
csi_ker_ck
HSI48ON
HSI48
hsi48_ck
CRS 时钟恢复系统
SW
SCGU（系统时钟生成）
SCEU（系统时钟使能）
到 CPU、总线和外设
CKPERSRC
hsi_ker_ck
csi_ker_ck
per_ck
PLLSRC
÷ DIVM1
÷ DIVM2
÷ DIVM3
ref1_ck
ref2_ck
ref3_ck
1 MHz 到 16 MHz
PLL1
PLL2
PLL3
VCO
DIVN1
DIVN2
DIVN3
FRACN1
FRACN2
FRACN3
DIVP1
DIVQ1
DIVR1
DIVP2
DIVQ2
DIVR2
DIVP3
DIVQ3
DIVR3
pll1_p_ck
pll1_q_ck
pll1_r_ck
pll2_p_ck
pll2_q_ck
pll2_r_ck
pll3_p_ck
pll3_q_ck
pll3_r_ck
pll[3:1]_q_ck
pll[3:2]_p_ck
pll[3:2]_r_ck
per_ck
PKSU（外设内核时钟选择）
PKEU（外设时钟使能）
到外设
I2S_CKIN
ETH_MII_TX_CLK
ETH_MII_RX_CLK
ETH_RMII_REF_CLK
ulpi1_phy_ck
ulpi2_phy_ck
USB_PHY1
USB_PHY2
D 所选输入可实时更改，不会导致输出信号中出现杂波
x 表示系统复位后所选择的复用器输入
MSv41928V3

8.5.1 时钟命名约定

RCC 为整个电路提供时钟。为避免误解，本文档使用以下术语：

- 外设时钟

  外设时钟是指 RCC 为外设提供的时钟。具体包括两类时钟：

  – 总线接口时钟

  – 内核时钟

  外设从 RCC 接收总线接口时钟以访问其寄存器，从而控制外设操作。该时钟通常为 AHB、APB 或 AXI 时钟，具体取决于外设所连接的总线。某些外设仅需一个总线接口时钟（例如，RNG、TIMx）。

  而某些外设还需要一个专用时钟来处理接口功能。该时钟被称为“内核时钟”。例如，SAI 等外设必须精确生成特定的主时钟频率，这需要专用的内核时钟频率。将总线接口时钟与特定接口需求相分离的另一个优势是，可在不重新编程外设的情况下更改总线时钟。

- CPU 时钟

  CPU 时钟是指提供给 CPU 的时钟。其源自系统时钟 (**sys_ck**)。

- 总线矩阵时钟

  总线矩阵时钟是指提供给不同桥（APB、AHB 或 AXI）的时钟。这些时钟源自系统时钟 (**sys_ck**)。

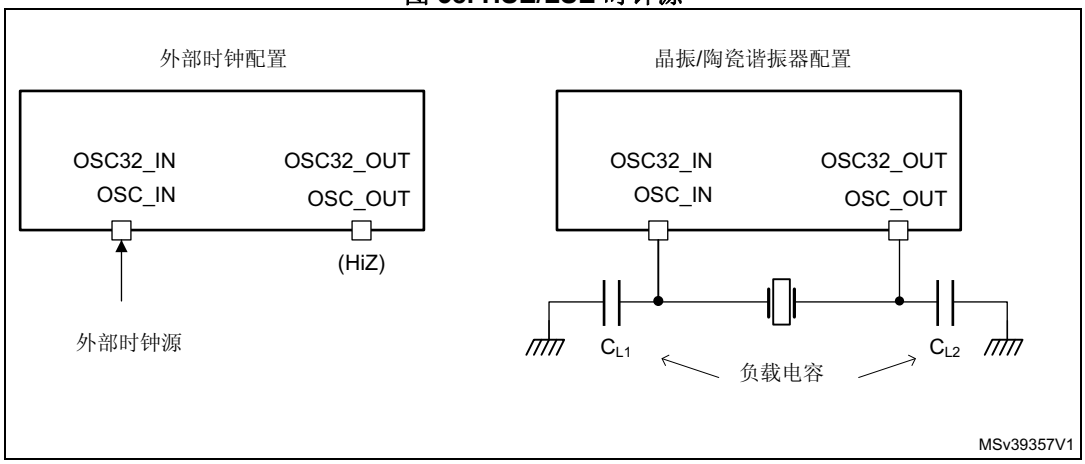
8.5.2 振荡器说明

HSE 振荡器

HSE 模块可通过两种可能的时钟源来生成时钟：

- 外部晶振/陶瓷谐振器
- 外部时钟源

图 38. HSE/LSE 时钟源



外部时钟源（HSE 旁路）

在此模式下，必须为 OSC_IN 引脚提供外部时钟源。通过将 *RCC 源控制寄存器 (RCC_CR)* 的 HSEBYP 和 HSEON 位置 1 来选择此模式。必须使用占空比约为 50% 的外部时钟源（方波、正弦波或三角波）来驱动 OSC_IN 引脚，同时 OSC_OUT 引脚应保持为 HI-Z（请参见 *图 38*）。

外部晶振/陶瓷谐振器

通过将 HSEBYP 位设为“0”并将 HSEON 位置“1”来使能该振荡器。

如果产品需要非常精确的高速时钟，可使用 HSE。

相关硬件配置如*图 38* 所示：谐振器和负载电容必须尽可能地靠近振荡器的引脚，以尽量减小输出失真和起振稳定时间。负载电容值必须根据所选晶振或陶瓷谐振器的不同做适当调整。有关详细信息，请参见数据手册的电气特性部分。

RCC 源控制寄存器 (RCC_CR) 的 HSERDY 标志指示 HSE 振荡器是否稳定。在启动时，硬件将此位置 1 后，**hse_ck** 时钟才可以使用。如在 *RCC 时钟源中断使能寄存器 (RCC_CIER)* 中使能中断，则可产生中断。

HSE 可通过 HSEON 位打开和关闭。请注意，如果满足以下两个条件之一，则不能关闭 HSE：

- HSE 直接（通过软件复用）用作系统时钟。
- 在使能并选择 PLL1 来提供系统时钟的情况下，将 HSE 选作 PLL1 的参考时钟（通过软件复用）。

此时，硬件不允许将 HSEON 位设为“0”。

系统进入停止或待机模式时，HSE 会自动由硬件禁止（更多详细信息，请参见*第 8.5.7 节：在停止和待机模式下处理时钟发生器*）。

此外，HSE 时钟可被驱动到 MCO1 和 MCO2 输出，以及用作其他应用组件的时钟源。

LSE 振荡器

LSE 模块可通过两种可能的时钟源来生成时钟：

- 外部晶振/陶瓷谐振器
- 外部用户时钟

外部时钟源（LSE 旁路）

在此模式下，必须为 OSC32_IN 引脚提供外部时钟源。输入时钟的最高频率不超过 1 MHz。通过将 *RCC 备份域控制寄存器 (RCC_BDCR)* 的 LSEBYP 和 LSEON 位置“1”来选择此模式。必须使用占空比约为 50% 的外部时钟信号（方波、正弦波或三角波）来驱动 OSC32_IN 引脚，同时 OSC32_OUT 引脚应保持为高阻态 (Hi-Z)。请参见*图 38*。

外部晶振/陶瓷谐振器（LSE 晶振）

LSE 时钟由 32.768 kHz 晶振或陶瓷谐振器生成。可作为实时时钟 (RTC) 的时钟源来提供时钟/日历或其他定时功能，具有功耗低且精度高的优点。

RCC 备份域控制寄存器 (RCC_BDCR) 的 LSERDY 标志指示 LSE 晶振是否稳定。在启动时，硬件将此位置 1 后，LSE 晶振输出时钟信号才可以使用。如在 *RCC 时钟源中断使能寄存器 (RCC_CIER)* 中使能中断，则可产生中断。

LSE 振荡器通过 LSEON 位打开和关闭。系统进入停止或待机模式时，LSE 保持使能状态。

此外，LSE 时钟可被驱动到 MCO1 输出，以及用作其他应用组件的时钟源。

LSE 还具有可编程驱动功能 (LSEDRV[1:0])，可用于调节放大器驱动能力。可动态将驱动能力从高驱动改为中高驱动，然后再改为中低驱动。

HSI 振荡器

HSI 模块为产品提供默认时钟。

HSI 为高速内部 RC 振荡器，其可直接用作系统时钟、外设时钟或 PLL 输入。应用可使用预分频器来选择 8 MHz、16 MHz、32 MHz 或 64 MHz 的 HSI 输出频率。此预分频器受 HSIDIV 控制。

HSI 的优势如下：

- 无需外部晶振，因此时钟源成本较低
- 与 HSE 相比，启动时间更短（几微秒）

但即使经过频率校准后，HSI 频率也不如外部晶振或陶瓷谐振器的频率精度高。

HSI 可通过 HSION 位打开和关闭。请注意，如果满足以下两个条件之一，则不能关闭 HSI：

- HSI 直接（通过软件复用）用作系统时钟。
- 在使能并选择 PLL1 来提供系统时钟的情况下，将 HSI 选作 PLL1 的参考时钟（通过软件复用）。

此时，硬件不允许将 HSION 位设为“0”。

请注意，如果将 HSI 选作至少一个已使能 PLL（PLLxON 位置“1”）的参考时钟，则不能更改 HSIDIV。此时，硬件不使用新值更新 HSIDIV。不过，如果将 HSI 直接用作系统时钟，则可更改 HSIDIV。

HSIRDY 标志指示 HSI 是否稳定。在启动时，硬件将此位置 1 后，HSI 输出时钟才可以使用。

HSI 时钟还可作为备份源（辅助时钟）使用，以防 HSE 发生故障（请参见*HSE 上的 CSS* 一节）。系统进入停止模式时，是否禁止 HSI 均可，更多详细信息请参见*第 8.5.7 节：在停止和待机模式下处理时钟发生器*。

此外，HSI 时钟可被驱动到 MCO1 输出，以及用作其他应用组件的时钟源。

将 HSI 用作通信外设的内核时钟时必须谨慎，应用必须考虑以下参数：

- 从外设生成内核时钟请求到时钟实际可用的时间间隔，
- 频率精度。

注： *系统处于停止模式时，HSI 可保持使能状态（更多相关信息，请参见第 8.5.7 节）。*

HSION、HSIRDY 和 HSIDIV 位在 RCC 源控制寄存器 (RCC_CR) 中。

HSI 校准

因为生产工艺不同，不同芯片的 RC 振荡器频率也不同。因此意法半导体会对每个器件进行出厂校准，以满足 ACC_{HSI} 精度要求（更多相关信息，请参见产品数据手册）。

上电复位后，工厂校准值将加载到 HSICAL[11:0] 位中。

如果应用受到电压或温度变化影响，则这可能也会影响到 RC 振荡器的频率。用户应用可使用 HSITRIM[5:0] 位对 HSI 频率进行微调。

注： *HSICAL[11:0] 和 HSITRIM[5:0] 位在 RCC 内部时钟源校准寄存器 (RCC_ICSCR) 中。*

CSI 振荡器

CSI 为低功耗 RC 振荡器，其可直接用作系统时钟、外设时钟或 PLL 输入。

CSI 的优势如下：

- 无需外部晶振，因此时钟源成本较低
- 与 HSE 相比，启动时间更短（几微秒）
- 极低功耗

CSI 提供的时钟频率为 4 MHz 左右，而 HSI 能提供高达 64 MHz 的时钟。

即使经过频率校准后，CSI 频率也不如外部晶振或陶瓷谐振器的频率精度高。

CSI 可通过 CSION 位打开和关闭。CSIRDY 标志指示 CSI 是否稳定。在启动时，硬件将此位置 1 后，CSI 输出时钟才可以使用。

如果满足以下两个条件之一，则不能关闭 CSI：

- CSI 直接（通过软件复用）用作系统时钟
- 在使能并选择 PLL1 来提供系统时钟的情况下，将 CSI 选作 PLL1 的参考时钟（通过软件复用）。

此时，硬件不允许将 CSION 位设为“0”。

系统进入停止模式时，是否禁止 CSI 均可（更多详细信息，请参见*第 8.5.7 节：在停止和待机模式下处理时钟发生器*）。

此外，CSI 时钟可被驱动到 MCO2 输出，以及用作其他应用组件的时钟源。

虽然 CSI 的稳定时间要比 HSI 的稳定时间短，但是将 CSI 用作通信外设的内核时钟时必须谨慎，应用必须考虑以下参数：

- 从外设生成内核时钟请求到时钟实际可用的时间间隔，
- 频率精度。

注： *CSION 和 CSIRDY 位在 RCC 源控制寄存器 (RCC_CR) 中。*

CSI校准

因为生产工艺不同，不同芯片的 RC 振荡器频率也不同，因此意法半导体会对每个器件进行出厂校准，以满足 ACC_{CSI} 精度要求（更多相关信息，请参见产品数据手册）。

复位后，工厂校准值将加载到 CSICAL[7:0] 位中。

如果应用受到电压或温度变化影响，则这可能也会影响到 RC 振荡器的频率。用户应用可使用 CSITRIM[4:0] 位对 CSI 频率进行微调。

注： *CSICAL[7:0] 和 CSITRIM[4:0] 位在 RCC 内部时钟源校准寄存器 (RCC_ICSCR) 中。*

HSI48 振荡器

HSI48 是一款 RC 振荡器，其提供的 48 MHz 时钟可直接用作某些外设的内核时钟。

HSI48 振荡器主要用于通过特殊时钟恢复系统 (CRS) 电路为 USB 外设提供高精度时钟，该电路可使用 USB SOF 信号、LSE 或外部信号以精细粒度实时自动调整振荡器频率。

系统进入停止或待机模式后，HSI48 振荡器会立即被禁止。如果未使用 CRS，则该振荡器将自由运行，并因此受到生产工艺差异的影响。因此意法半导体会对每个器件进行出厂校准，以满足 ACC_{HSI48} 精度要求（更多相关信息，请参见产品数据手册）。

有关如何配置和使用 CRS 的更多详细信息，请参见*第 9 节：时钟恢复系统 (CRS)*。

HSI48RDY 标志指示 HSI48 振荡器是否稳定。在启动时，硬件将此位置 1 后，HSI48 输出时钟才可以使用。

HSI48 可通过 HSI48ON 位打开和关闭。

HSI48 时钟还可被驱动到 MCO1 复用器，以及用作其他应用组件的时钟源。

注： *HSI48ON 和 HSI48RDY 位在 RCC 源控制寄存器 (RCC_CR) 中。*

LSI 振荡器

系统处于停止或待机模式时，LSI 可作为低功耗时钟源保持运行，供独立看门狗 (IWDG) 和自动唤醒单元 (AWU) 使用。时钟频率在 32 kHz 左右。有关详细信息，请参见数据手册的电气特性部分。

LSI 可通过 LSION 位打开和关闭。LSIRDY 标志指示 LSI 振荡器是否稳定。如果独立看门狗已通过硬件或软件启动，则 LSI 将强制打开且不可禁止。

系统进入停止或待机模式时，LSI 保持使能状态（更多详细信息，请参见*第 8.5.7 节：在停止和待机模式下处理时钟发生器*）。

LSI 启动时，硬件将 LSIRDY 位置 1 后，才会提供相应时钟。如在 *RCC 时钟源中断使能寄存器 (RCC_CIER)* 中使能中断，则可产生中断。

此外，LSI 时钟可被驱动到 MCO2 输出，以及用作其他应用组件的时钟源。

注： *LSION 和 LSIRDY 位在 RCC 时钟控制和状态寄存器 (RCC_CSR) 中。*

8.5.3 时钟安全系统 (CSS)

HSE 上的 CSS

时钟安全系统可由软件通过 HSECSSON 位使能。即使将 HSEON 设为“0”，也仍可使能 HSECSSON 位。

在 HSE 已使能并就绪且 HSECSSON 置“1”时，HSE 上的 CSS 将由硬件使能。

HSE 禁止时，将禁止 HSE 上的 CSS。因此当系统处于停止模式时，该功能不起作用。

不能由软件直接将 HSECSSON 位清零。

发生系统复位或系统进入待机模式时，由硬件将 HSECSSON 位清零（请参见*第 8.4.2 节：系统复位*）。

如果检测到 HSE 时钟故障，则系统会自动切换到 HSI 以提供安全时钟。HSE 随后将自动禁止，一个时钟故障事件将发送到高级控制定时器（TIM1、TIM8、TIM15、TIM16 和 TIM17）的断路输入，并且同时还将生成一个中断来向软件通知此故障（CSS 中断：***rcc_hsecss_it***），从而可使 MCU 能够执行救援操作。如果在发生故障时已将 HSE 输出用作 PLL 的时钟源，则还会禁止 PLL。

如果在使能 CSS 时发生 HSE 时钟故障，则 CSS 会生成一个中断，这会导致自动生成 NMI。*RCC 时钟源中断标志寄存器 (RCC_CIFR)* 中的 HSECSSF 标志置“1”，以允许应用识别故障源。NMI 程序将无限期执行，除非将 HSECSSF 位清零。因此，应用程序必须通过将 *RCC 时钟源中断清零寄存器 (RCC_CICR)* 中的 HSECSSC 位置 1 来清零 NMI ISR 中的 HSECSSF 标志。

LSE 上的 CSS

可由软件通过对 *RCC 备份域控制寄存器 (RCC_BDCR)* 中的 LSECSSON 位进行编程，来使能 LSE 振荡器上的时钟安全系统。

只有在满足以下条件时，该位才能由硬件禁止：

- 在 pwr_vsw_rst（V_{SW} 软件复位）后
- 或在检测到 LSE 故障后。

LSE 使能（LSEON 位由软件置 1）并就绪（LSERDY 位由硬件置 1）后，以及通过 RTCSEL 位选择 RTC 时钟后，必须写入 LSECSSON 位。

LSE 上的 CSS 适用于除 VBAT 外的所有模式（运行、停止和待机）。

如果检测到 LSE 故障，则不会再向 RTC 提供 LSE 时钟，但 RTCSEL、LSECSSON 和 LSEON 位的值不会由硬件更改。

在待机模式下，会产生唤醒。在其他模式下，可发送中断 (**rcc_lsecss_it**) 来唤醒软件。软件随后必须禁止 LSECSSON 位并停止故障 LSE（将 LSEON 位清零），且可通过 RTCSEL 位更改 RTC 时钟源（无时钟、LSI 或 HSE），或者采取任何必要措施来确保应用的安全。

8.5.4 时钟输出生成 (MCO1/MCO2)

此微控制器有两个时钟输出 (MCO) 引脚可供使用，分别为 MCO1 和 MCO2。可以为每个输出选择一个时钟源。所选时钟可通过可配置的预分频器进行分频（有关信号选择的更多信息，请参见*图 37*）。

MCO1 和 MCO2 输出通过 *RCC 时钟配置寄存器 (RCC_CFGR)* 中的 MCO1PRE[3:0]、MCO1[2:0]、MCO2PRE[3:0] 和 MCO2[2:0] 进行控制。

GPIO 端口对应于各个 MCO 引脚，必须在复用功能模式下进行编程。

为 MCO 输出提供的时钟不能超出最大引脚速度（有关支持的引脚速度的信息，请参见产品数据手册）。

8.5.5 PLL 描述

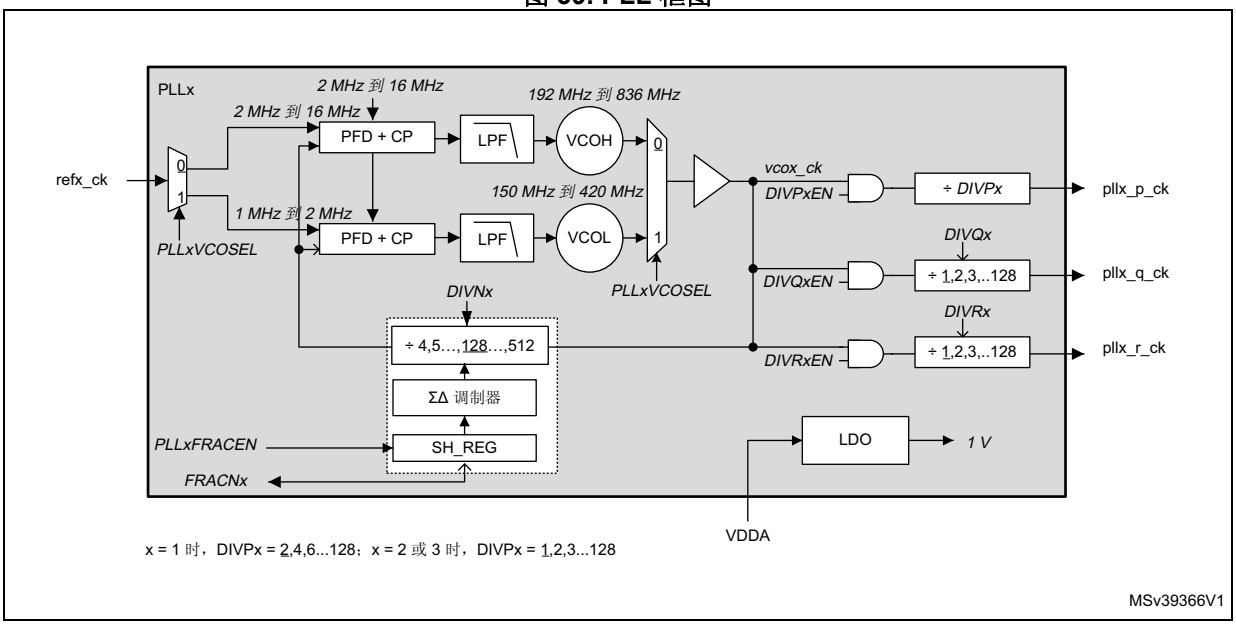
RCC 具有三个 PLL：

- 一个主 PLL (PLL1)，通常用于为 CPU 和某些外设提供时钟。
- 两个专用 PLL（PLL2 和 PLL3），用于为外设生成内核时钟。

集成在 RCC 中的 PLL 完全独立。它们具有以下特性：

- 两个嵌入式 VCO：
  - 一个宽范围 VCO (VCOH)
  - 一个低频 VCO (VCOL)
- 输入频率范围：
  - 1 MHz 到 2 MHz（使用 VCOL 时）
  - 2 MHz 到 16 MHz（使用 VCOH 时）
- 能够工作在整数或小数模式下
- 13 位 Sigma-Delta 调制器，能够对 VCO 频率进行 11 档微调（误差最高 0.3 ppm）。
- Sigma-Delta 调制器可实时更新，不会在 PLL 输出端产生频率过冲。
- 每个 PLL 均可通过后分频器提供 3 种输出

图 39. PLL 框图



PLL 通过 RCC_PLLxDIVR、RCC_PLLxFRACR、RCC_PLLCFGR 和 RCC_CR 寄存器进行控制。

为 PLL 提供的参考时钟的频率 (**refx_ck**) 必须介于 1 MHz 到 16 MHz 范围内。为满足此条件，用户应用必须正确编程 *RCC PLL 时钟源选择寄存器 (RCC_PLLCKSELR)* 的 DIVMx 分频器。此外，必须根据参考输入频率设置 *RCC PLL 配置寄存器 (RCC_PLLCFGR)* 字段的 PLLxRGE，以确保 PLL 达到最佳性能。

用户应用随后可配置合适的 VCO：如果参考时钟的频率低于或等于 2 MHz，则必须选择 VCOL，否则必须选择 VCOH。要降低功耗，建议将 VCO 输出配置为最低频率。

必须对 DIVNx 环分频器进行编程，以使 VCO 输出达到预期频率。此外，必须满足 VCO 输出范围。

将 SH_REG（FRACNx 影子寄存器）的值设为“0”时，PLL 工作在整数模式下。PLLxFRACEN 位从“0”变为“1”时，用 FRACNx 值更新 SH_REG。Sigma-Delta 调制器采用专门设计，可最大程度地降低抖动影响，同时实现非常小的频率阶跃。

可通过将 PLLxON 置“1”使能 PLL。位 PLLxRDY 指示 PLL 已就绪（即，已锁定）。

注： *在使能 PLL 之前，请确保为 PLL 提供的参考频率 **(refx_ck)** 稳定，以使硬件在 PLLx 开启时不允许更改 DIVMx，以及在其中的一个 PLL 开启时无法更改 PLLSRC。*

如果当前使用 PLL1 提供系统时钟，则硬件可防止向 PLL1ON 写入“0”。此外，时钟发生器还具备其他硬件保护（请参见 HSE 振荡器、HSI 振荡器和 CSI 振荡器部分）。
使能 PLL 后不能更改以下 PLL 参数：DIVNx、PLLxRGE、PLLxVCOSEL、DIVPx、DIVQx、DIVRx、DIVPxEN、DIVQxEN 和 DIVRxEN。

为确保 PLL 达到最佳性能，如果未使用其中的一个后分频器（DIVP、DIVQ 或 DIVR），则应用程序应将使能位 (DIVyEN) 置 1 以及将相应的后分频器位（DIVP、DIVQ 或 DIVR）设为“0”。

如果未遵循上述规则，则无法保证 PLL 输出频率。

输出频率计算

在整数模式 (SH_REG = “0”) 下配置 PLL 时，VCO 频率 (F_{VCO}) 通过以下表达式进行计算：

$$F_{VCO} = F_{REF_CK} \times DIVN$$

$$F_{PLL_y_CK} = (F_{VCO} / (DIVy + 1))，其中 y = P、Q 或 R$$

在小数模式（SH_REG 不为“0”）下配置 PLL 时，在使能 PLL 之前必须先初始化 DIVN 分频器。不过，可实时更改 FRACNx 值，而不会干扰 PLL 输出。

可使用该功能基于精度较高的晶振值生成特定频率，或实时微调频率。

对于每个 PLL，VCO 频率均通过以下公式进行计算：

$$F_{VCO} = F_{ref_ck} \times \left(DIVN + \frac{FRACN}{2^{(13)}} \right)$$

注： *对于 PLL1，DIVP 只能使用奇数值。*

在以下情况下，PLL 将由硬件禁止：

- 系统进入停止或待机模式。
- 将 HSE 或 PLL（由 HSE 提供时钟）用作系统时钟时发生 HSE 故障。

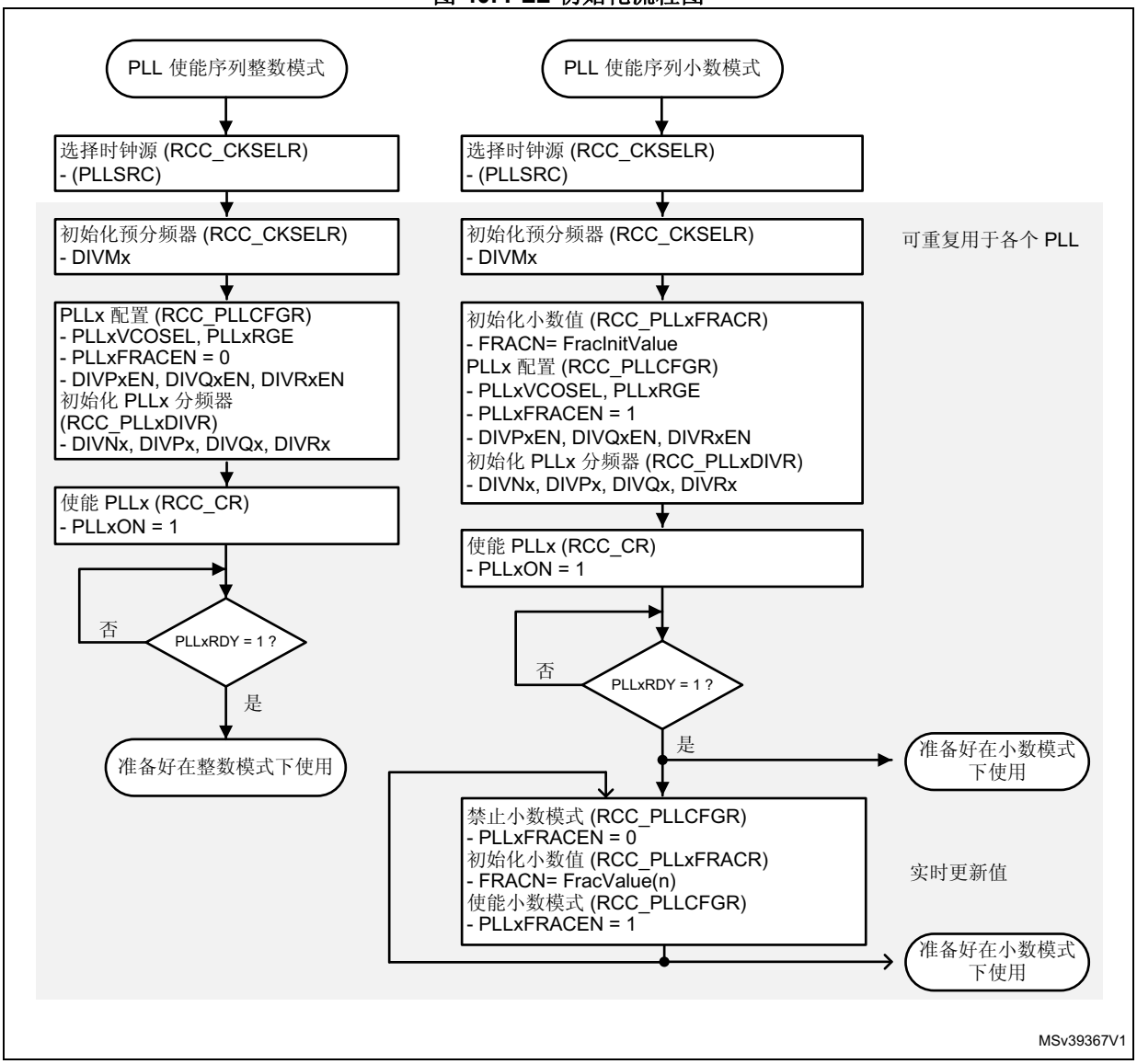
PLL 初始化阶段

图 40 给出了整数和小数模式下推荐的 PLL 初始化序列。在初始化序列开始时，PLLx 应处于禁止状态：

1. 根据所需频率初始化 PLL 寄存器。
   - 对于整数模式，将 *RCC PLL 配置寄存器 (RCC_PLLCFGR)* 的 PLLxFRACEN 设为“0”。
   - 对于小数模式，将 FRACN 设为所需初始值 (FracInitValue)，然后将 PLLxFRACEN 置“1”。
2. 将 PLLxON 位置“1”后，用户应用必须等待到 PLLxRDY 位置“1”。如果 PLLx 处于小数模式，则只要 PLLxRDY =“0”，PLLxFRACEN 位就不能设回“0”。
3. 将 PLLxRDY 位置“1”后，PLLx 即准备好进行使用。
4. 如果应用程序要实时调整 PLLx 频率（仅限小数模式），则：
   a) PLLxFRACEN 必须设为“0”，如果 PLLxFRACEN =“0”，则 Sigma-Delta 调制器仍使用锁存到 SH_REG 中的值。
   b) 新值必须上传到 PLLxFRACR (FracValue(n)) 中。
   c) PLLxFRACEN 必须置“1”，以将 PLLxFRACR 的内容锁存到其影子寄存器中。

注： *PLLxRDY 变为“1”时即表示，PLLx 输出频率和目标值之差小于 ±2%。*

图 40. PLL 初始化流程图



8.5.6 系统时钟 (sys_ck)

系统时钟选择

系统复位后，会将 HSI 选作系统时钟，并且所有 PLL 均将关闭。当时钟源用于系统时钟时，软件无法通过 xxxON 位禁止所选时钟源。

当然，系统时钟可在系统进入停止或待机模式时由硬件停止。

系统处于运行状态时，用户应用可在以下 4 种时钟源中选择系统时钟 (**sys_ck**)：

- HSE
- HSI
- CSI
- 或 pll1_p_ck

该功能通过编程 *RCC 时钟配置寄存器 (RCC_CFGR)* 进行控制。只有在目标时钟源已就绪时（时钟在启动延迟或 PLL 锁相后稳定时），才可从一个时钟源切换到另一个。如果选择尚未就绪的时钟源，则切换在该时钟源就绪时才会进行。
RCC 时钟配置寄存器 (RCC_CFGR) 的 SWS 状态位指示当前用作系统时钟的时钟。RCC_CR 寄存器中的其他状态位指示已就绪的时钟。

系统时钟生成

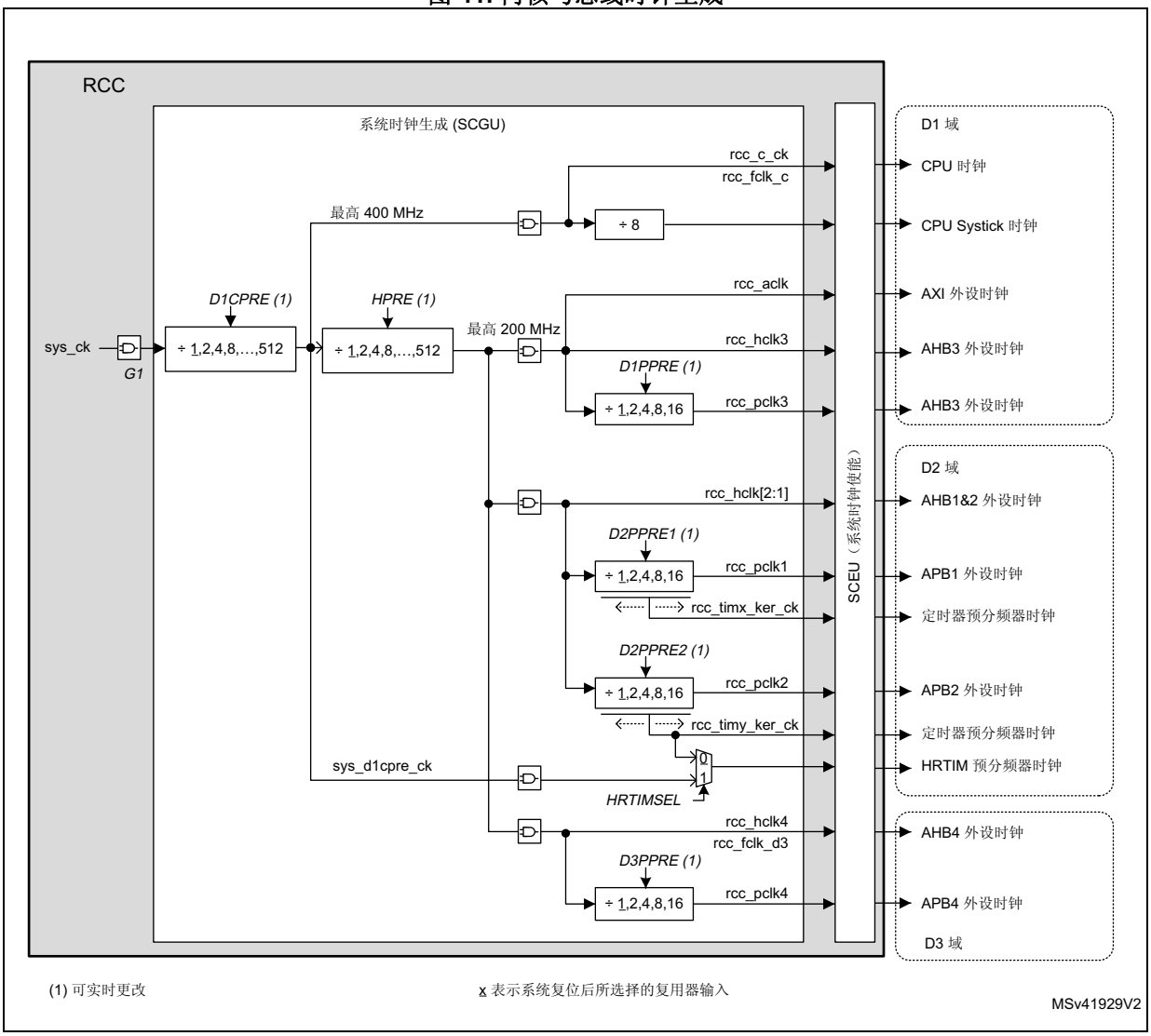
图 41 显示了 CPU 和总线的时钟分配的简化框图。框图中显示的所有分频器均可实时更改，且不会违反时序。如要根据应用需求调整总线频率并因此优化功耗，该功能是一种非常简单的解决方案。

可使用 D1CPRE 分频器调整 CPU 时钟。不过，这也会影响所有总线矩阵和 HRTIM 的时钟频率。

同样，可使用 HPRE 分频器调整 D1 域总线矩阵的时钟，但这也会影响 D2 和 D3 域总线矩阵的时钟频率。

大多数预分频器均通过 RCC_D1CFGR、RCC_D2CFGR 和 RCC_D3CFGR 寄存器进行控制。

图 41. 内核与总线时钟生成



该模块还为定时器（**rcc_timx_ker_ck 和 rcc_timy_ker_ck**）提供时钟。定时器时钟的频率取决于 APB 预分频器（对应于定时器所连的总线）和 TIMPRE 位。*表 47* 显示了如何选择定时器时钟频率。

表 47. 时钟定时器与 pclk 之比

D2PPRE1 (1) D2PPRE2	TIMPRE (2)		$F_{rcc_timx_ker_ck}$ $F_{rcc_timy_ker_ck}$	F_{rcc_pclk1} F_{rcc_pclk2}	注释
0xx	0	→	F_{rcc_hclk1}	F_{rcc_hclk1}	定时器时钟等效于总线时钟。
100	0	→	F_{rcc_hclk1}	F_{rcc_hclk1} / 2	定时器时钟频率是总线时钟频率的两倍。
101	0	→	F_{rcc_hclk1} / 2	F_{rcc_hclk1} / 4	
110	0	→	F_{rcc_hclk1} / 4	F_{rcc_hclk1} / 8	
111	0	→	F_{rcc_hclk1} / 8	F_{rcc_hclk1} / 16	

表 47. 时钟定时器与 pclk 之比（续）

D2PPRE1 (1) D2PPRE2	TIMPRE (2)		$F_{rcc_timx_ker_ck}$ $F_{rcc_timy_ker_ck}$	F_{rcc_pclk1} F_{rcc_pclk2}	注释
0xx	1	→	F_{rcc_hclk1}	F_{rcc_hclk1}	定时器时钟等效于总线时钟。
100	1	→	F_{rcc_hclk1}	F_{rcc_hclk1} / 2	定时器时钟频率是总线时钟频率的两倍。
101	1	→	F_{rcc_hclk1}	F_{rcc_hclk1} / 4	定时器时钟频率是总线时钟频率的 4 倍。
110	1	→	F_{rcc_hclk1} / 2	F_{rcc_hclk1} / 8	
111	1	→	F_{rcc_hclk1} / 4	F_{rcc_hclk1} / 16	

1. D2PPRE1 和 D2PPRE2 属于 *RCC 域 2 时钟配置寄存器 (RCC_D2CFGR)*。
2. TIMPRE 属于 *RCC 时钟配置寄存器 (RCC_CFGR)*。

8.5.7 在停止和待机模式下处理时钟发生器

当整个系统进入停止模式时，所有时钟（系统时钟和内核时钟）均停止，以下时钟源也会停止：

- CSI、HSI（取决于 HSIKERON 和 CSIKERON 位）
- HSE
- PLL1、PLL2 和 PLL3
- HSI48

除被设为“0”的 PLL1ON、PLL2ON、PLL3ON HSEON 和 HSI48ON 外，RCC 寄存器的内容未发生更改。

退出停止模式

微控制器通过唤醒事件使系统退出停止模式时，应用程序可选择将用于重启的振荡器（HSI 和/或 CSI）。STOPWUCK 位将振荡器选作系统时钟。STOPKERWUCK 位选择用作外设内核时钟的振荡器。如果在系统停止后，外设需要由非系统时钟振荡器来生成内核时钟，则 STOPKERWUCK 位十分有用。

所有这些位均属于 *RCC 时钟配置寄存器 (RCC_CFGR)*。*表 48* 给出了相应行为的详细说明。

表 48. STOPWUCK 和 STOPKERWUCK 说明

STOPWUCK	STOPKERWUCK		系统退出停止模式时激活的振荡器	系统退出停止模式时分配的时钟	
				系统时钟	内核时钟
0	0	→	HSI	HSI	HSI
	1	→	HSI 和 CSI		HSI 和/或 CSI
1	0	→		CSI	
	1	→	CSI		CSI

停止模式期间

系统处于停止模式时，在以下两种特定情况下可使能 HSI 或 CSI：

- 专用外设请求内核时钟时：

  在这种情况下，外设将接收到 HSI 或 CSI，具体取决于为该外设选择的内核时钟源（通过 PERxSRC）。
- 位 HSIKERON 或 CSIKERON 置 1 时：

  在这种情况下，HSI 和 CSI 在停止模式期间保持运行，但会对输出进行门控。这样，在系统退出停止模式时或在外设请求内核时钟时，时钟将立即可用（有关详细信息，请参见*表 49*）。

HSIKERON 和 CSIKERON 位属于 *RCC 源控制寄存器 (RCC_CR)*。*表 49* 给出了相应行为的详细说明。

表 49. HSIKERON 和 CSIKERON 行为

HSIKERON (CSIKERON)		停止模式期间的 HSI (CSI) 状态	HSI (CSI) 稳定时间
0	→	关闭	$t_{su(HSI)}$ ($t_{su(CSI)}$) (1)
1	→	正在运行且受门控	立即

1. $t_{su(HSI)}$ 和 $t_{su(CSI)}$ 为 HSI 和 CSI 振荡器的启动时间（有关这些参数值的信息，请参见产品数据手册）。

当微控制器使系统退出待机模式时，将 HSI 选作系统和内核时钟，除 RCC_RSR（或 RCC_C1_RSR）和 RCC_BDCR 寄存器外的 RCC 寄存器均复位为初始值。

另应注意，HSI 和 CSI 输出提供两条时钟路径（请参见*图 37*）：

- 一条路径用于系统时钟（**hsi_ck** 或 **csi_ck**）
- 一条路径用于外设内核时钟（**hsi_ker_ck** 或 **csi_ker_ck**）。

如果外设在系统处于停止模式时请求内核时钟，则只会激活提供 **hsi_ker_ck** 或 **csi_ker_ck** 的路径。

注意： 无法保证 CPU 会自动获得退出 CStop 模式时的时钟频率：这主要取决于系统状态。例如，如果 CPU 进入 CStop 模式，而 D3 域保持在 CRun 模式下，则在 CPU 退出 CStop 模式时，时钟设置保持不变。如果 D3 域进入 CStop 模式，且 CPU 也处于 CStop 模式下，则在 CPU 退出 CStop 模式时，CPU 将使用退出 CStop 模式时的 HSI 或 CSI。

8.5.8 内核时钟选择

某些外设经过专门设计，支持两种可异步运行的不同时钟域：

- 与寄存器和总线接口同步的时钟域（**ckg_bus_perx** 时钟）
- 通常与外设同步的时钟域（内核时钟）。

拥有支持这两种时钟域的外设具备以下优势：用户应用能更加自由地为 CPU、总线矩阵和外设的内核部分选择最佳时钟频率。

因此，用户应用无需对外设进行重新编程即可更改总线频率。例如，即使实时更改相应的 APB 时钟，也不会干扰正在通过 UART 进行的传输过程。

表 50 显示了 RCC 可为外设提供的内核时钟。*表 50* 的每一行都表示一个复用器以及与其输出相连的外设。从第 4 列开始的各个列表示时钟源。第 3 列给出了各个复用器输出所允许的最大频率。用户需遵循这些要求。

表 50. 内核时钟分配概述

外设	时钟复用器控制位	最大 允许频率 (MHz)	domain	时钟源																		
				pll1_q_ck	pll2_p_ck	pll2_q_ck	pll2_r_ck	pll3_p_ck	pll3_q_ck	pll3_r_ck	sys_ck	总线时钟 (1)	hse_ck	hsi_ker_ck	csi_ker_ck	hsi48_ck	lse_ck	lsi_ck	per_ck (2)	I2S_CKIN	USB_PHY1/2	输出关闭
LTDC	-	66	D1							x												
FMC	FMCSEL	200		1			2					0							3			
QUADSPI	QSPISEL	200		1			2					0							3			
SDMMC1	SDMMCSEL	200 (4)		0			1															
SDMMC2		200	D2																			
DFSDM1 Aclk	SAI1SEL	133		0	1			2											4	3		
DFSDM1 clk	DFSDM1SEL	133									1	0										
FDCAN	FDCANSEL	100		1		2							0									
HDMI-CEC	CECSEL	66													2 (3)		0	1				
I2C1/2/3	I2C123SEL	100								1		0		2	3							
LPTIM1	LPTIM1SEL	100			1					2		0					3	4	5			
TIM[8:1]、TIM[17:12]	-	200										x										
HRTIM	-	400										x										
RNG：	RNGSEL	66		1												0	2	3				
SAI1	SAI1SEL	133		0	1			2											4	3		
SAI2	SAI23SEL	133		0	1			2											4	3		
SAI3		133																				
SPDIFRX	SPDIFSEL	200		0			1			2				3								
SPI(I2S)1,2,3	SPI123SEL	200		0	1			2											4	3		
SPI4,5	SPI45SEL	100				1			2			0	5	3	4							
SWPMI	SWPSEL	100										0		1								
USART1,6	USART16SEL	100				1			2			0		3	4		5					
USART2,3 UART4,5,7,8	USART234578SEL	100				1			2			0		3	4		5					
USB1OTG USB2OTG	USBSEL	66		1					2							3						0
USB1ULPI	-	100																			x	
USB2ULPI	-	100																			x	
ADC1,2	ADCSEL	36 (4)			0					1									2			
ADC3			D3																			

表 50. 内核时钟分配概述（续）

外设	时钟复用器控制位	最大 允许频率 (MHz)	domain	时钟源																		
				pll1_q_ck	pll2_p_ck	pll2_q_ck	pll2_r_ck	pll3_p_ck	pll3_q_ck	pll3_r_ck	sys_ck	总线时钟 (1)	hse_ck	hsi_ker_ck	csi_ker_ck	hsi48_ck	lse_ck	lsi_ck	per_ck (2)	I2S_CKIN	USB_PHY1/2	输出关闭
I2C4	I2C4SEL	100	D3							1		0		2	3							
LPTIM2	LPTIM2SEL	100			1					2		0					3	4	5			
LPTIM3,4,5	LPTIM345SEL	100			1					2		0					3	4	5			
LPUART1	LPUART1SEL	100				1			2			0		3	4		5					
SAI4_A	SAI4ASEL	133		0	1			2											4	3		
SAI4_B	SAI4BSEL	133		0	1			2											4	3		
SPI6	SPI6SEL	100				1			2			0	5	3	4							
RTC/AWU	RTCSEL	1	VSW										3 (5)				1	2				0

1. 总线时钟为与外设相连的总线接口时钟，可为 APB、AHB 或 AXI 时钟。
2. per_ck 时钟可为 hse_ck、hsi_ker_ck 或 csi_ker_ck，具体取决于 CKPERSEL 选择。
3. 时钟 CSI 进行 122 分频。
4. 占空比接近 50%，这意味着 DIV[P/Q/R]x 值应为偶数。对于 SDMMCx，如果支持 DDR，则占空比应为 50%。
5. 时钟 HSE 由 RTCPRE 分频。

图 42 到*图 51* 给出了内核时钟分配的更详细说明。为简化框图，图中未表示总线接口时钟（pclk、hclk），即使其已通过使能信号进行门控。更多详细信息，请参见*第 8.5.11 节：外设时钟门控*。

为减少开关数，某些外设共用同一内核时钟源。不过，所有外设均具有各自的专用使能信号。

专用于音频应用的外设

除 SPDIFRX 外，音频外设通常需要特别精确的频率。如*图 42* 所示，SAI 或 SPI(I2S) 的内核时钟可由以下时钟源生成：

- PLL1（必须减小活动 PLL 数目时）
- PLL2 或 PLL3（用于在频率生成方面实现最佳灵活性）
- HSE、HSI 或 CSI（用于对电流消耗极为敏感的用例）
- I2S_CKIN（需要使用外部时钟参考时）。

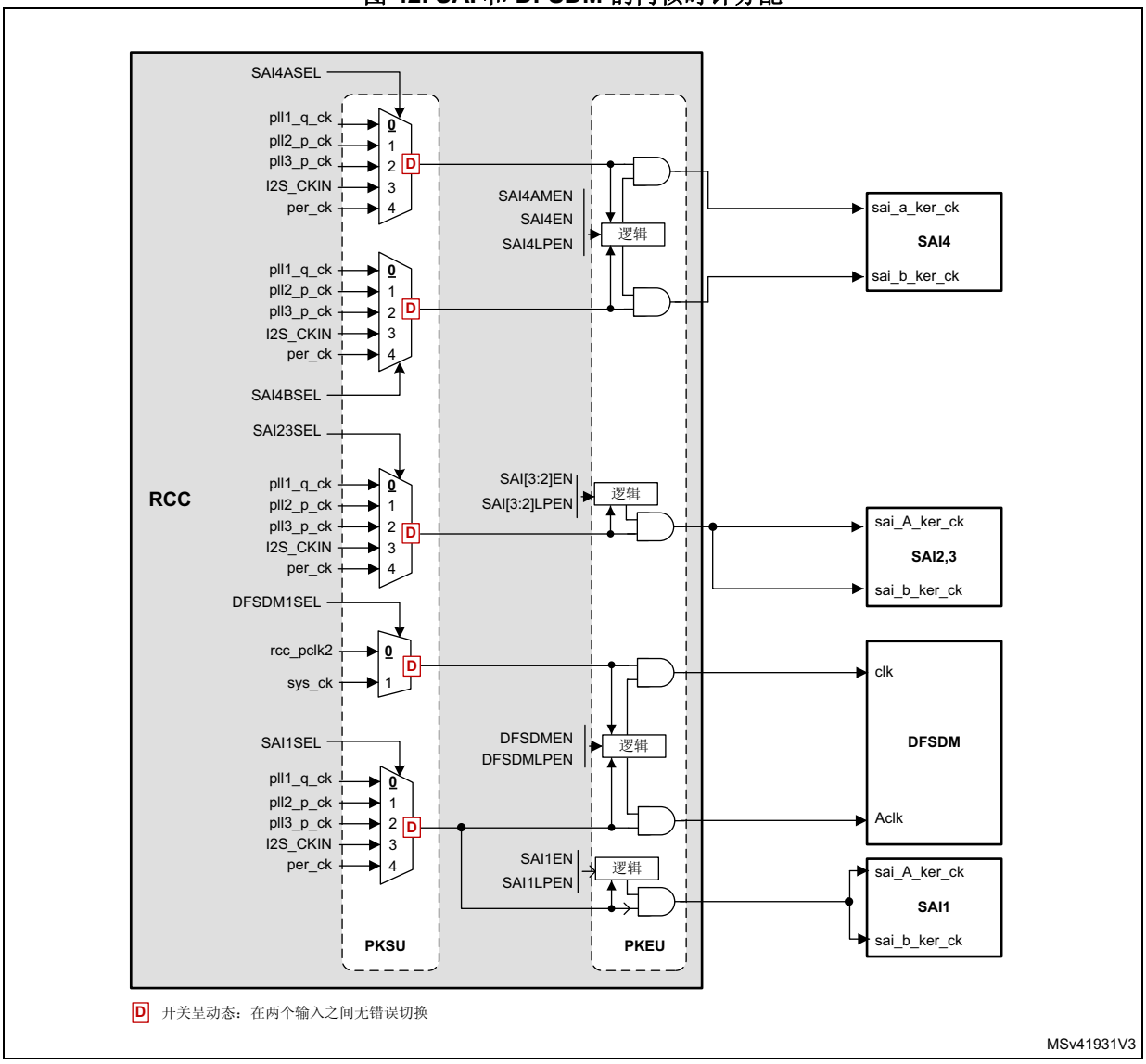
注： *SPDIFRX 无需特定频率，只需一个足够高的内核时钟频率即可使外设正常工作。更多详细信息，请参见 SPDIFRX 说明。*

DFSDM1 可与 SAI1A 使用同一时钟。这在将 DFSDM1 用于音频应用时十分有用。

为提高灵活性，SAI4 可对各个子模块使用不同的时钟。

SPI/I2S1、2 和 3 共用同一内核时钟源（请参见图 43）。

图 42. SAI 和 DFSDM 的内核时钟分配



1. **X** 表示系统复位后所选择的复用器输入。
2. 该图未显示总线接口时钟与外设的连接。有关各个使能单元的详细信息，请参见*第 8.5.11 节：外设时钟门控*。

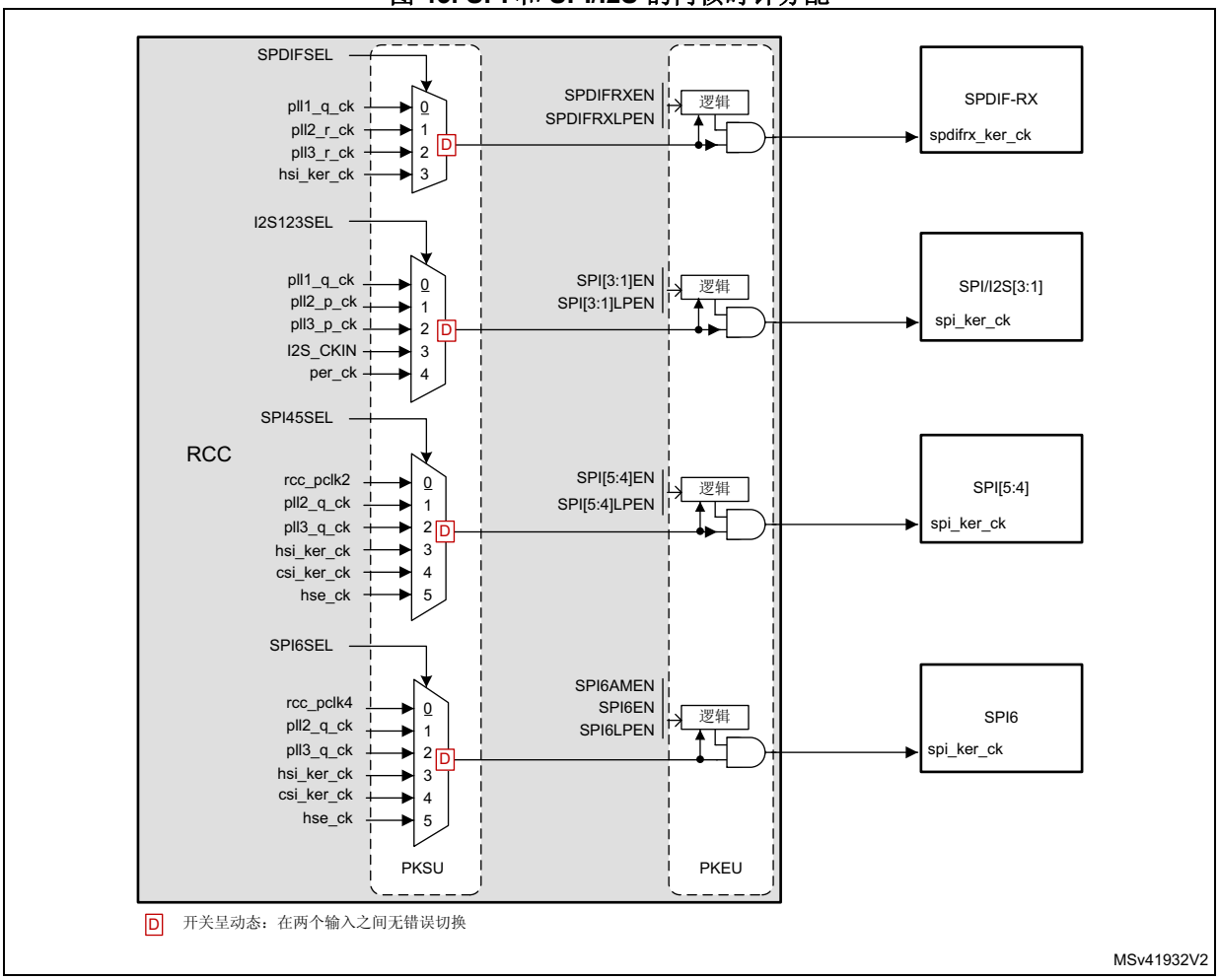
专用于控制和数据传输的外设

SPI、I2C、UART 等外设无需特定内核时钟频率，但时钟频率应足以生成合适的波特率，或串行接口上所需的位时钟。为此，可从以下时钟源中选择：

- PLL1（必须减小活动 PLL 数目时）
- PLL2 或 PLL3（需要更高的灵活性时）。例如，该解决方案可通过 PLL1 更改频率总线，且不会影响某些串行接口的速度。
- HSI 或 CSI（适用于低功耗条件，或在外设必须快速从停止模式唤醒的情况，即 UART、I2C...）。

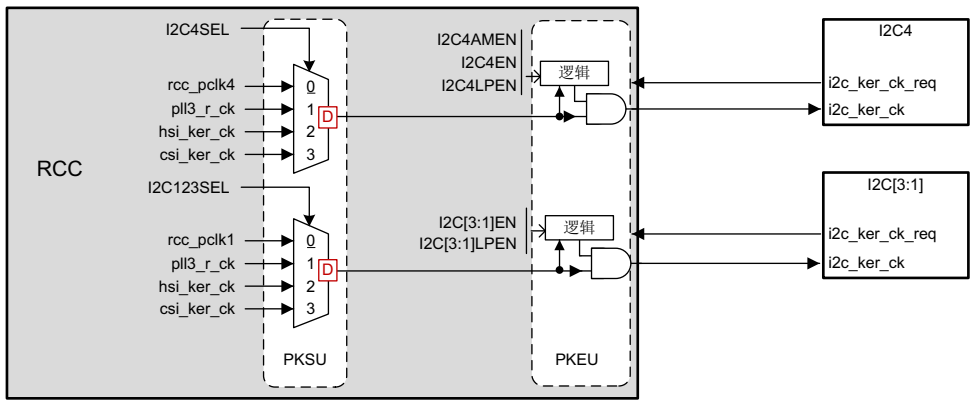
注：　如果无需高波特率，则 UART 还需要 LSE 时钟。

图 43. SPI 和 SPI/I2S 的内核时钟分配



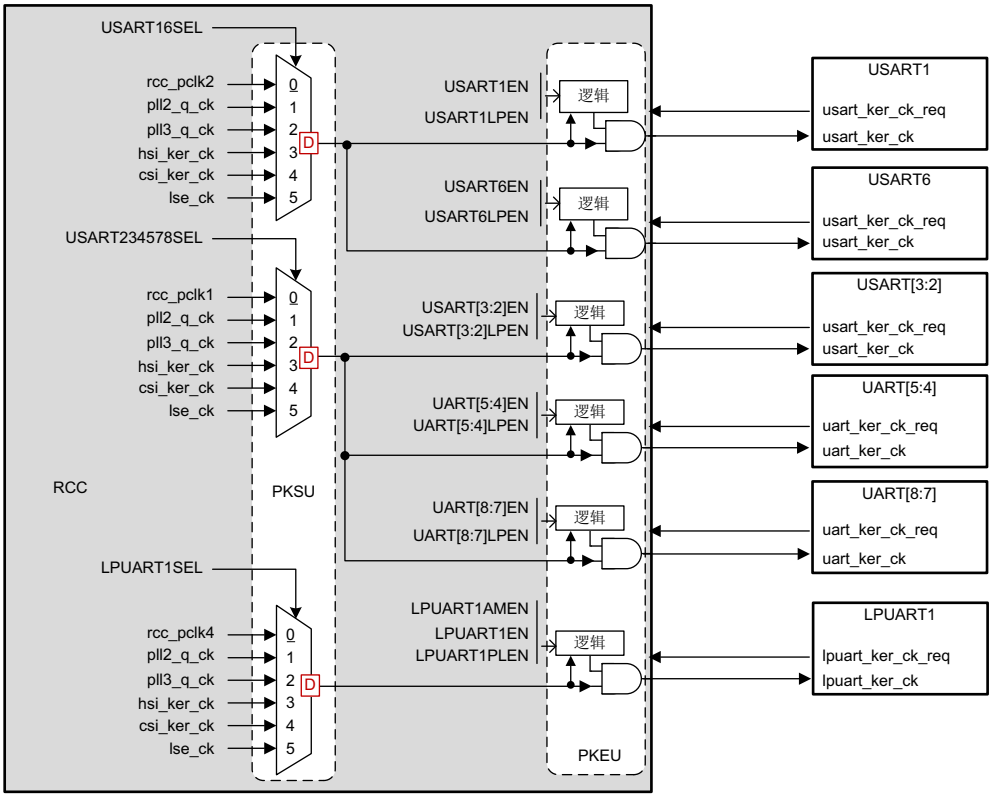
1. **X** 表示系统复位后所选择的复用器输入。
2. 该图未显示总线接口时钟与外设的连接。有关各个使能单元的详细信息，请参见*第 8.5.11 节：外设时钟门控*。

图 44. I2C 的内核时钟分配



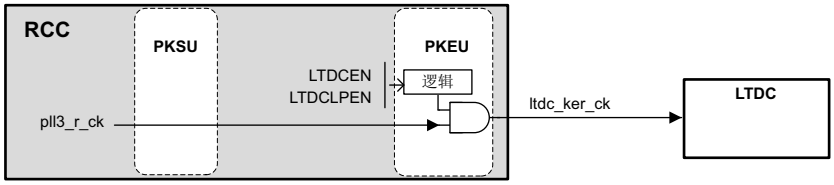
1. **X** 表示系统复位后所选择的复用器输入
2. 该图未显示总线接口时钟与外设的连接，有关各个使能单元的详细信息，请参见*第 8.5.11 节：外设时钟门控*。

图 45. UART、USART 和 LPUART1 的内核时钟分配



1. **X** 表示系统复位后所选择的复用器输入。
2. 该图未显示总线接口时钟与外设的连接，有关各个使能单元的详细信息，请参见*第 8.5.11 节：外设时钟门控*。

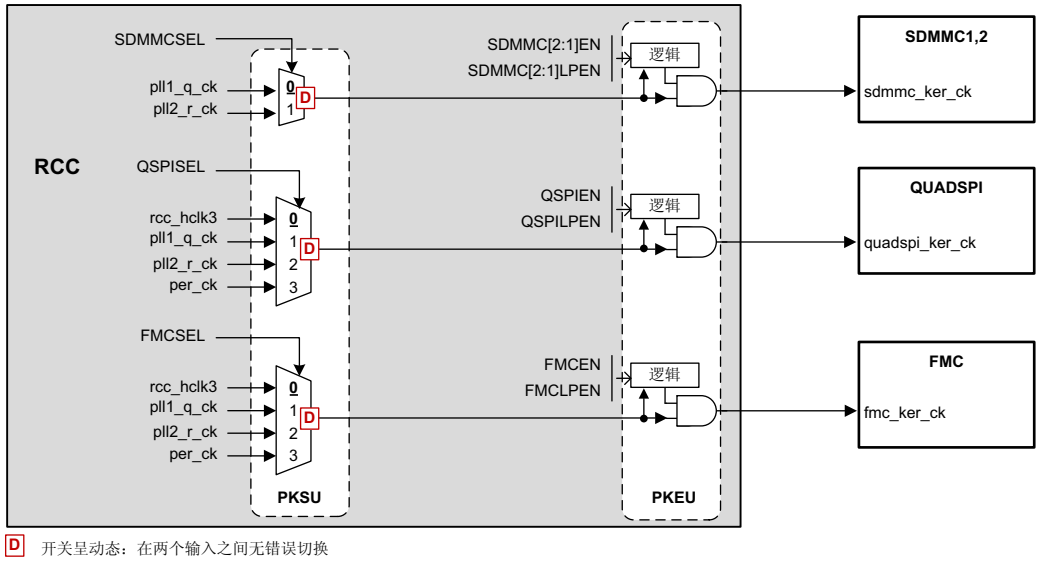
图 46. LTDC 的内核时钟分配



1. **X** 表示系统复位后所选择的复用器输入。
2. 该图未显示总线接口时钟与外设的连接。有关各个使能单元的详细信息，请参见*第 8.5.11 节：外设时钟门控*。

为提高灵活性，FMC、QUADSPI 和 SDMMC1/2 还可与总线接口使用不同的时钟。

图 47. SDMMC、QUADSPI 和 FMC 的内核时钟分配

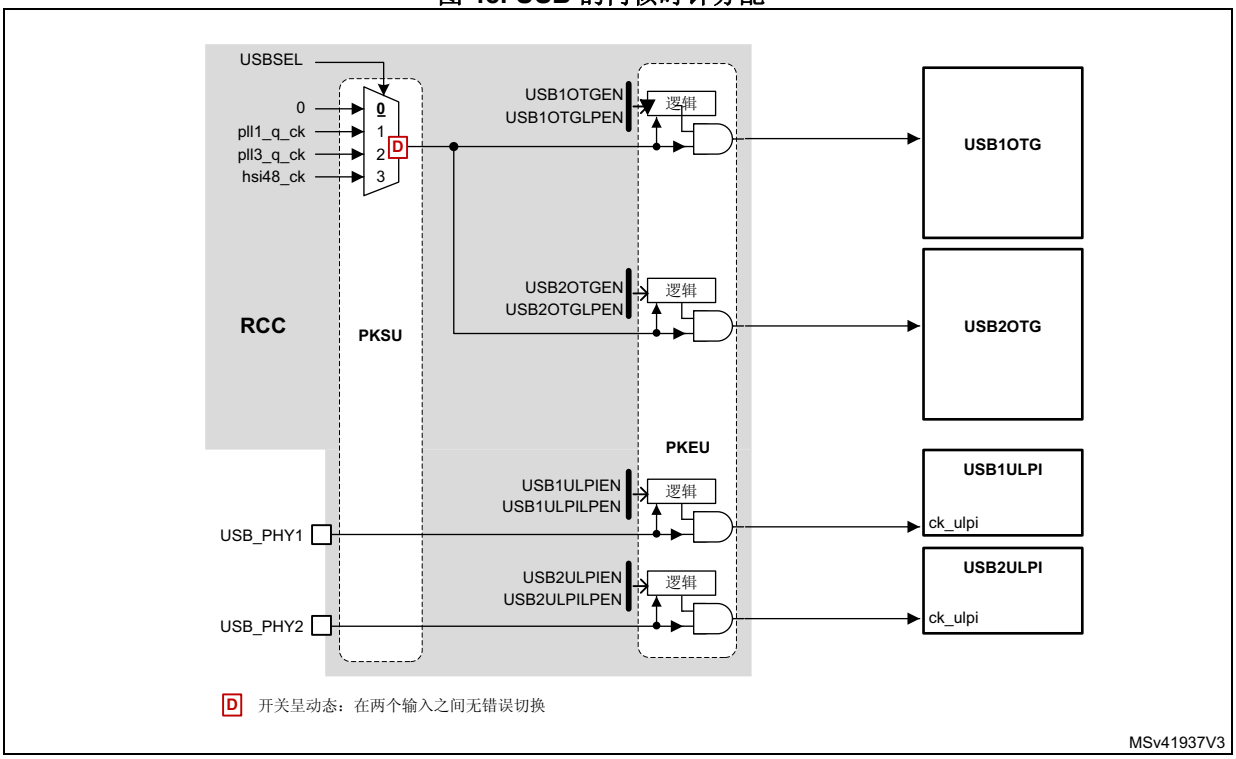


1. **X** 表示系统复位后所选择的复用器输入。
2. 该图未显示总线接口时钟与外设的连接。有关各个使能单元的详细信息，请参见*第 8.5.11 节：外设时钟门控*。

图 48 显示了 USB 模块的时钟分配。USBxULPI 模块从外部 PHY 接收其时钟。

USBxOTG 模块接收用于 USB 通信的时钟，可通过 USBSEL 控制的复用器在各个时钟源之间进行选择。

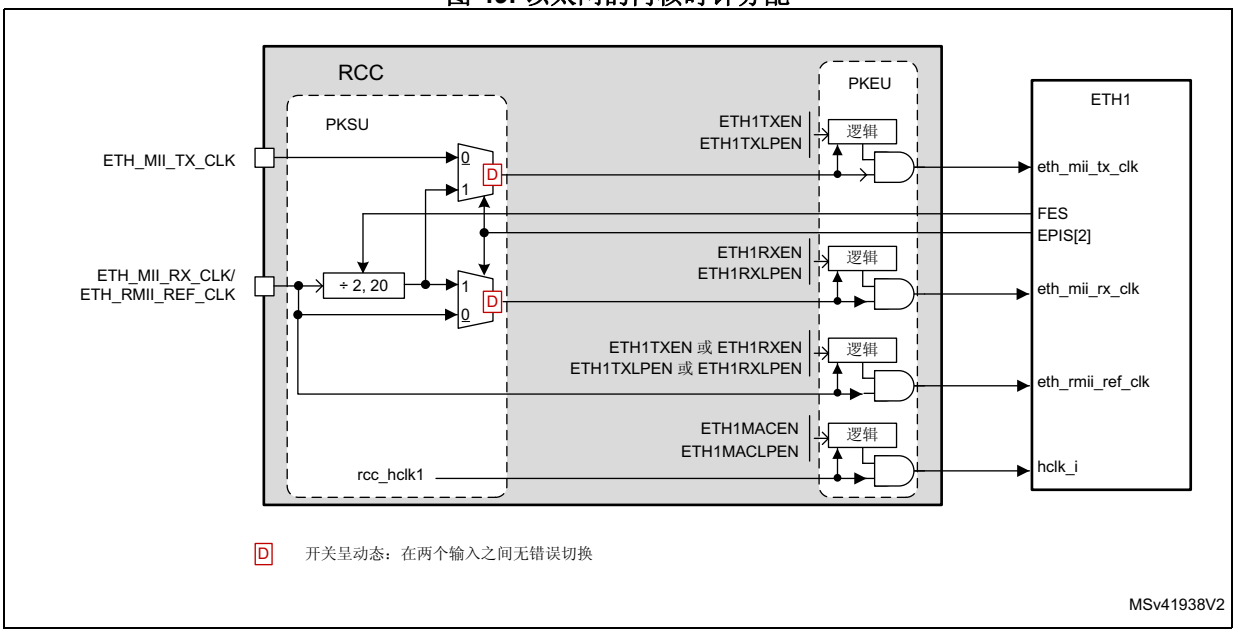
图 48. USB 的内核时钟分配 (2)



1. **X** 表示系统复位后所选择的复用器输入。
2. 该图未显示总线接口时钟与外设的连接。有关各个使能单元的详细信息，请参见*第 8.5.11 节：外设时钟门控*。

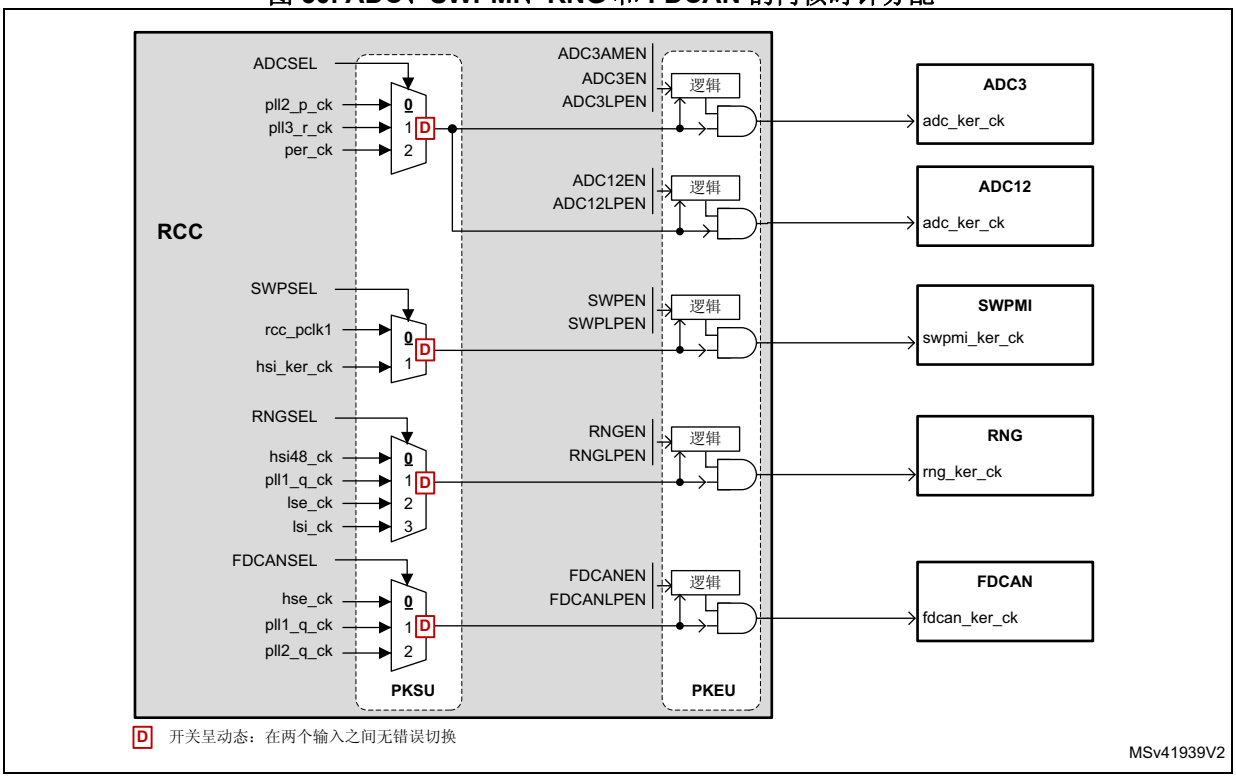
以太网发送和接收时钟应由外部以太网 PHY 提供。RX 和 TX 路径的时钟选择由 SYSCFG 模块进行控制。

图 49. 以太网的内核时钟分配



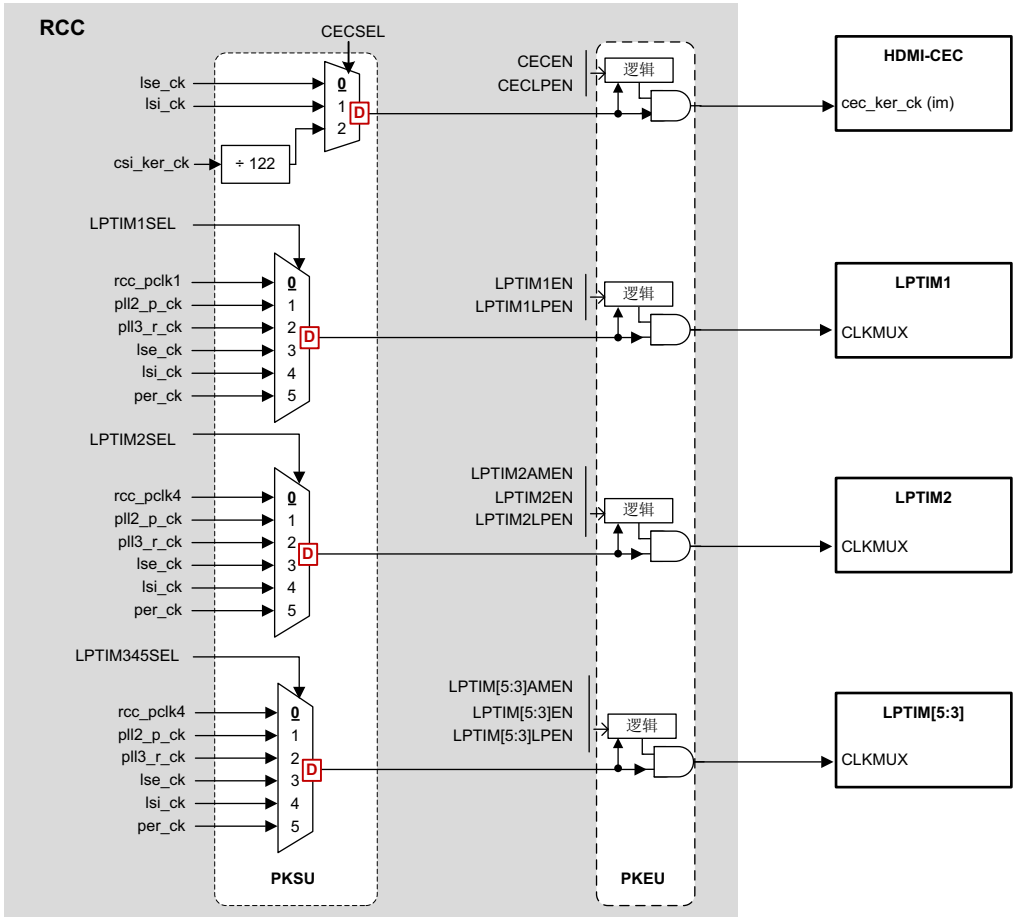
1. **X** 表示系统复位后所选择的复用器输入。
2. 该图未显示总线接口时钟与外设的连接。有关各个使能单元的详细信息，请参见*第 8.5.11 节：外设时钟门控*。

图 50. ADC、SWPMI、RNG 和 FDCAN 的内核时钟分配 (2)



1. **X** 表示系统复位后所选择的复用器输入。
2. 该图未显示总线接口时钟与外设的连接。有关各个使能单元的详细信息，请参见*第 8.5.11 节：外设时钟门控*。

图 51. LPTIM 和 HDMI-CEC 的内核时钟分配 (2)



1. **X** 表示系统复位后所选择的复用器输入
2. 该图未显示总线接口时钟与外设的连接。有关各个使能单元的详细信息，请参见*第 8.5.11 节：外设时钟门控*。

RTC/AWU 时钟

rtc_ck 时钟源可为：

- **hse_1M_ck**（**hse_ck** 经过可编程预分频器分频获得）
- **lse_ck**
- 或 **lsi_ck** 时钟

时钟源通过 *RCC 备份域控制寄存器 (RCC_BDCR)* 中的 RTCSEL[1:0] 位和 *RCC 时钟配置寄存器 (RCC_CFGR)* 中的 RTCPRE[5:0] 位进行选择。

所做的选择只能通过复位备份域的方式修改。

如果选择 LSE 作为 RTC 时钟，则即使在备份或 V_{DD} 电源丢失时 RTC 也仍将正常工作。

LSE 时钟位于备份域中，而其他振荡器则不是。因此：

- 如果选择 LSE 作为 RTC 时钟，则即使 V_{DD} 电源关断，RTC 也仍会继续工作（假设 V_{BAT} 电源保持工作）。
- 如果选择 LSI 作为 RTC 时钟，则在 V_{DD} 电源关断时将无法保证 AWU 状态。
- 如果将 HSE 时钟用作 RTC 时钟，则在 V_{DD} 电源关断或 V_{CORE} 电源关断时，无法保证 RTC 状态。

rtc_ck 时钟通过 *RCC 备份域控制寄存器 (RCC_BDCR)* 中的 RTCEN 位使能。

RTC 总线接口时钟（APB 时钟）通过 RCC_APB4ENR/LPENR 寄存器中的 RTCAPBEN 和 RTCAPBLPEN 位使能。

注： *在 APB 时钟频率低于 RTC 时钟频率的七倍 ($F_{APB} < 7 \times F_{RTCLCK}$) 时读取 RTC 日历寄存器，软件必须读取日历时间寄存器和日期寄存器两次。如果对 RTC_TR 的第二次读取访问得到的结果与第一次相同，则表明数据正确无误。否则必须执行第三次读取访问。*

看门狗时钟

RCC 为电路中的四个看门狗模块提供时钟。独立看门狗 (IWDG1) 连接到 LSI。窗口看门狗 (WWDG1) 连接到 APB 时钟。

如果独立看门狗已通过硬件选项或软件访问的方式启动，则 LSI 将强制打开且不可禁止。在 LSI 振荡器设置延迟之后，时钟将提供给 IWDG。

注意： 在使能 WWDG1 之前，应用程序必须将 WW1RSC 位置“1”。如果 WW1RSC 保持为“0”，则在使能 WWDG1 时，无法保证其行为。WW1RSC 位在 *RCC 全局控制寄存器 (RCC_GCR)* 中。

使用 TIMx 测量时钟频率

大多数时钟源发生器频率均可通过对 TIMx 的输入捕获进行测量。

- 使用 LSE 校准 HSI 或 CSI

  将 LSE 连接到 TIMx 输入捕获的主要目的是为了精确测量 HSI 或 CSI。这需要直接或通过 PLL1 将 HSI 或 CSI 用作系统时钟源。借助 LSE 信号连续边沿之间的系统时钟计数数量，即可对内部时钟周期进行测量。利用 LSE 的高精度（通常为几十 ppm），用户能以同一分辨率测定时钟频率，并可通过对时钟源进行微调来补偿由生产工艺造成的和/或与温度和电压相关的频率偏差。

  其基本原理是基于相对的测量（例如，HSI/LSE 比）：因此，精度与两个时钟源之比紧密相关。比率越大，测量结果越准确。

  HSI 和 CSI 振荡器均设有针对此目的的专用校准位，且支持用户访问（请参见 *RCC 内部时钟源校准寄存器 (RCC_ICSCR)*）。通过 PLLx 使用 HSI 或 CSI 时，还可通过 PLL 的小数分频器对系统时钟进行微调。

- 使用 HSI 校准 LSI

  同时也可测量 LSI 频率：这对于没有晶振的应用场合非常实用。超低功耗 LSI 振荡器具有较大的生产工艺偏差。通过测量该振荡器与 HSI 时钟源的比率，可以借助 HSI 精度测定其频率。通过此测量值可实现更加精确的 RTC 时基超时（当使用 LSI 作为 RTC 时钟源时）和/或实现精度水平可以接受的 IWDG 超时。

8.5.9 常规时钟概念概述

RCC 处理针对系统（D1、D2 和 D3 域）的 CPU、总线接口和外设时钟分配，具体取决于各个功能的工作模式（有关时钟定义的详细信息，请参见*第 8.5.1 节：时钟命名约定*）。

对于各个外设，应用程序可控制其内核及总线接口时钟的激活/禁用。在使用外设之前，CPU 必须先将其使能（通过将 PERxEN 置“1”），并定义该外设是否在 CSleep 模式下保持激活（通过将 PERxLPEN 置“1”）。这被称为到 CPU 的外设“分配”（更多详细信息，请参见*第 8.5.10 节：外设分配*）。

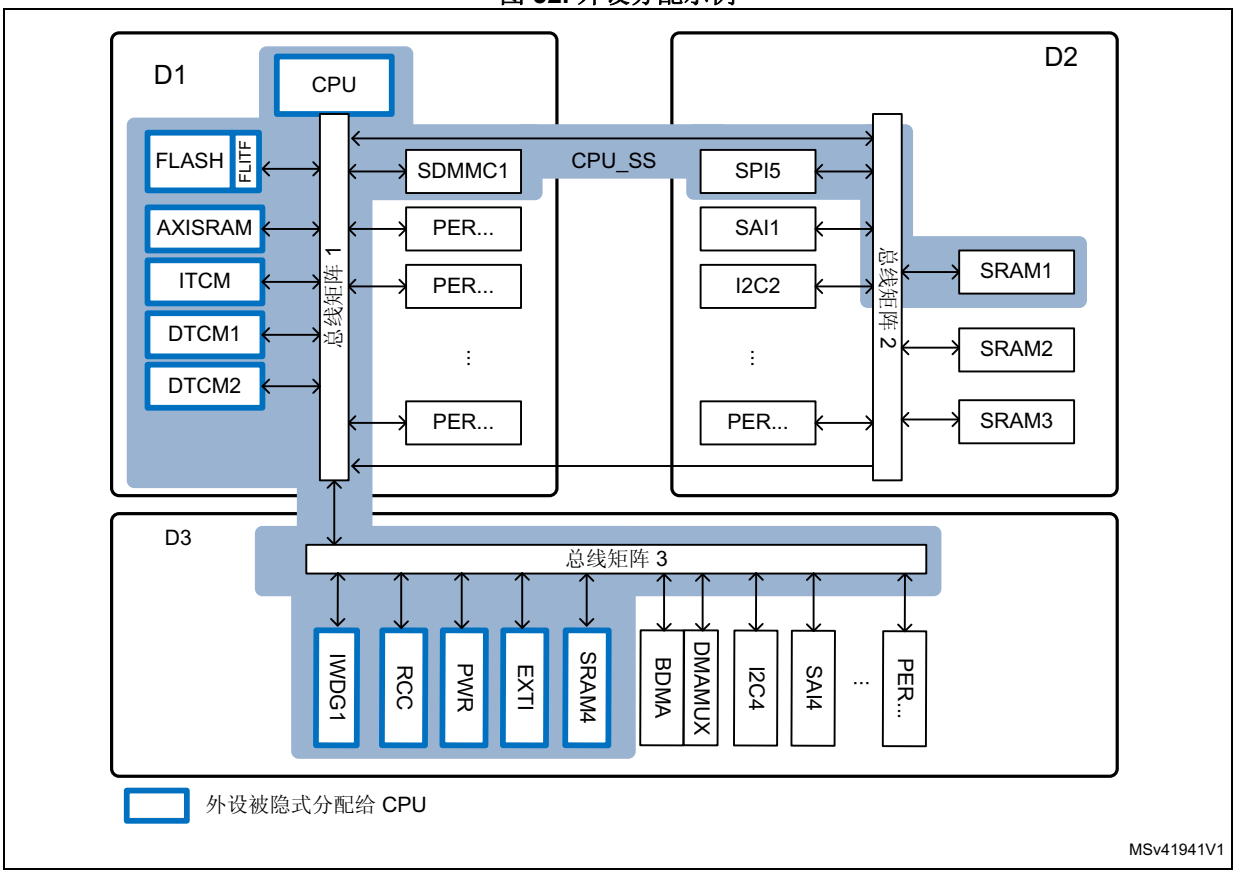
外设分配供 RCC 用于根据 CPU 和域模式自动控制时钟门控，以及供 PWR 用于控制 D1、D2 和 D3 域的电源电压。

图 52 所示为外设分配的示例：

- CPU 使能的 SDMMC1、SPI5 和 SRAM1、AXISRAM、ITCM、DTCM1、DTCM2 以及 SRAM4 被隐式分配给 CPU。包括 CPU、总线矩阵 1/2/3 和所分配外设的组形成了一个子系统 (CPU_SS)。

注： *FLASH、AXISRAM、ITCM、DTCM1、DTCM2、SRAM4、IWGD1、IWGD2、PWR、EXTI 和 RCC 为公用资源，被隐式分配给 CPU。*

图 52. 外设分配示例



当 CPU 进入 CStop 模式时，RCC 会自动禁止 CPU_SS 所有外设的总线接口和内核时钟以及 CPU 时钟。如果 PLL 已使能，则不会由 RCC 禁止，因为 D3 仍处于运行状态。

当 CPU 处于 CStop 模式时，D3 域可保持在 DRun 模式下，D1 和 D2 域处于 DStop 或 DStandby 模式下。可通过将 PWR_CPUCR 寄存器中的 RUN_D3 位置 1 来完成此操作。

- 如果将 RUN_D3 置“1”，则 D3 保持在 DRun 模式下，这与 CPU 模式无关（请参见 *PWR CPU 控制寄存器 (PWR_CPUCR)*）。
- 如果将 RUN_D3 设为“0”，则在 CPU 进入 CStop 模式时，D3 域进入 DStop 或 DStandby 模式（请参见*表 51*）。

请注意，CPU 可通过 *PWR CPU 控制寄存器 (PWR_CPUCR)* 的位 PDDS_D1、PDDS_D2 和 PDDS_D3 控制是否允许 D1、D2 或 D3 域在满足条件时进入 DStandby 模式。

唤醒事件可使 D1、D2 和 D3 域退出 DStandby 或 DStop 模式。

此外，还为 D3 域中的某些外设提供了更多自主性（有关详细信息，请参见*D3 域自主模式*一节）。

D3 域自主模式

自主模式允许向 D3 中的外设提供外设时钟，即使 CPU 处于 CStop 模式亦如此。如果外设已使能其自主位，则会在 CPU 处于 CStop 模式时根据 D3 域状态接收相应的外设时钟：

- 如果 D3 域处于 DRun 模式下，则已激活自主模式的外设会接收其外设时钟，
- 如果 D3 域处于 DStop 模式下，则不会提供外设时钟。

自主模式不会阻止 D3 域进入 DStop 或 DStandby 模式。

自主位在 *RCC D3 自主模式寄存器 (RCC_D3AMR)* 中。

例如，在 SAI4 使用通过 BDMA 接收自外部器件的数据填充 SRAM4 时，CPU 可进入 CStop 模式。达到一定的接收数据量时，可通过唤醒事件激活 CPU。可通过在 D3 保持在 DRun 模式（RUN_D3 置“1”）时将 SAI4、BDMA 和 SRAM4 置于自主模式来完成上述操作。在本示例中，RCC 不会关闭 PLL，因为 D3 域始终处于 DRun 模式。

将自主模式与某些外设（UART、I2C）的功能相结合来自行请求内核时钟，而无需唤醒 CPU，这可以进一步降低功耗。例如，如果系统正通过 I2C4 等待消息，则可将整个系统置于停止模式。如果 I2C4 外设检测到 START 位，则会生成一个“内核时钟请求”。该请求可使能 HSI 或 CSI，并且只会向请求方（在我们的示例中，为 I2C4）提供内核时钟。I2C4 随后会对传入消息进行解码。之后可能会出现以下几种情况：

- 如果消息的器件地址不匹配，则在检测到新 START 条件之前，I2C4 会释放其“内核时钟请求”。
- 如果传入消息的器件地址匹配，则必须将其存储在 D3 本地存储器中。如果地址匹配，则 I2C4 能生成一个唤醒事件以将 D3 域切换到 DRun 模式。随后会通过 BDMA 将消息传送到存储器中，D3 域将返回到 DStop 模式下，且不会激活 CPU。请注意，如果传送到存储器中的数据量达到传送计数，则 BDMA 还会生成一个中断来唤醒 CPU。
- 如果传入消息的器件地址匹配且外设被设置为唤醒 CPU，则 I2C4 会生成一个唤醒事件来激活 CPU。

请参见 EXTI 模块的说明，了解哪个外设能对哪个域执行唤醒事件。

存储器处理

CPU 可访问产品中的所有存储区：

- AXISRAM、ITCM、DTCM1、DTCM2 和 FLASH，
- SRAM1、SRAM2 和 SRAM3，
- SRAM4 和 BKPRAM。

如 *图 52* 所示，FLASH、AXISRAM、SRAM4、ITCM、DTCM1 和 DTCM2 被隐式分配给 CPU。因此，不存在允许 CPU 分配这些存储器的使能位。

如果 CPU 要使用 D2 域中的存储器（SRAM1、SRAM2 和 SRAM3），则必须将其使能。

BKPRAM 具有专用的使能信号，可对总线接口时钟进行门控。CPU 在使用 BKPRAM 之前需要先将其使能。

注： *在 CSleep 模式下，可通过软件（通过 DxSRAMyLPEN 位）停止存储器接口时钟（Flash 和 RAM 接口）。*

有关时钟使能的详细信息，请参见*外设时钟门控*和 *CPU 和总线矩阵时钟门控*部分。

系统状态概述

表 51 给出了 D1、D2 和 D3 域模式的系统状态概述。

- 只要 D3 处于 DRun 模式，系统就会保持在运行模式下。系统运行模式的部分子状态并未在此详细介绍（更多信息，请参见*电源控制 (PWR)*）。
- 当 D1 域处于 Drun 模式时，D2 域可处于 DRun、DStop 或 DStandby 模式。当 D1 域处于 DStop 或 DStandby 模式时，D2 域可不再保持 DRun 模式，它将根据 PDDS_D2 位切换到 DStop 或 DStandby 模式。
- 当 D1 和 D2 处于 DStop/DStandby 模式时，D3 可凭借 PWR_CPUCR 寄存器的 RUN_D3 位保持运行状态或者在自主模式下运行。
- 只要 D3 处于 DStop 模式，系统就会保持在停止模式下。这暗示着，D1 和 D2 处于 DStop 或 DStandby 模式下。一旦 D1 或 D2 退出 DStop 或 DStandby，D3 就会切换到 DRun 模式。
- 只要 D1、D2 和 D3 处于 DStandby 模式下，系统就会保持在待机模式。
- 域状态与 CPU 状态：
  - 当 D1 域处于 DRun 模式时，则意味着已为其总线矩阵提供时钟，且 CPU 处于 CRun 模式。
  - 当 D2 域处于 DRun 模式时，则意味着已为其总线矩阵提供时钟，CPU 处于 CRun 模式，且至少已分配 D2 域的一个外设。
  - 当 D1 域处于 DStop 模式时，则意味着不再为其总线矩阵提供时钟，且 CPU 处于 CStop 模式。
  - 当 D2 域处于 DStop 模式时，则意味着不再为其总线矩阵提供时钟。在以下条件下会出现这种情况：

    - CPU 未分配 D2 域的外设，

    - CPU 已分配 D2 域的外设，但 CPU 处于 CStop 或 Cstandby 模式下，
  - 当域处于 DStandby 模式时，则意味着域及其 CPU 掉电。

表 51. 系统状态概述

系统状态	D1 状态	D2 状态	D3 状态
运行	DRUN	DRun/DStop/DStandby	DRUN
	DStop/DStandby	DStop/DStandby	
Stop	DStop/DStandby	DStop/DStandby	DStop
待机	DStandby	DStandby	DStandby

8.5.10 外设分配

CPU 可分配外设并因此控制其内核和总线接口时钟。

CPU 可通过将以下寄存器中的专用 PERxEN 位置 1 来分配外设：

- RCC_xxxxENR 寄存器或
- RCC_C1_xxxxENR 寄存器。

CPU 处于 CSleep 模式时可通过以下寄存器中的 PERxLPEN 位控制外设时钟门控：

- RCC_xxxxLPENR 寄存器或
- RCC_C1_xxxxLPENR 寄存器。

有关更多信息，请参见*第 8.7.1 节：寄存器映射概述*。

外设分配位（PERxEN 位）供硬件用于为外设提供内核和总线接口时钟。不过，其还可用于将外设与 CPU（CPU 子系统）相连。这样，硬件便能根据 CPU 状态安全地对外设时钟和总线矩阵时钟进行门控。PWR 模块还使用该信息来适当控制域状态。

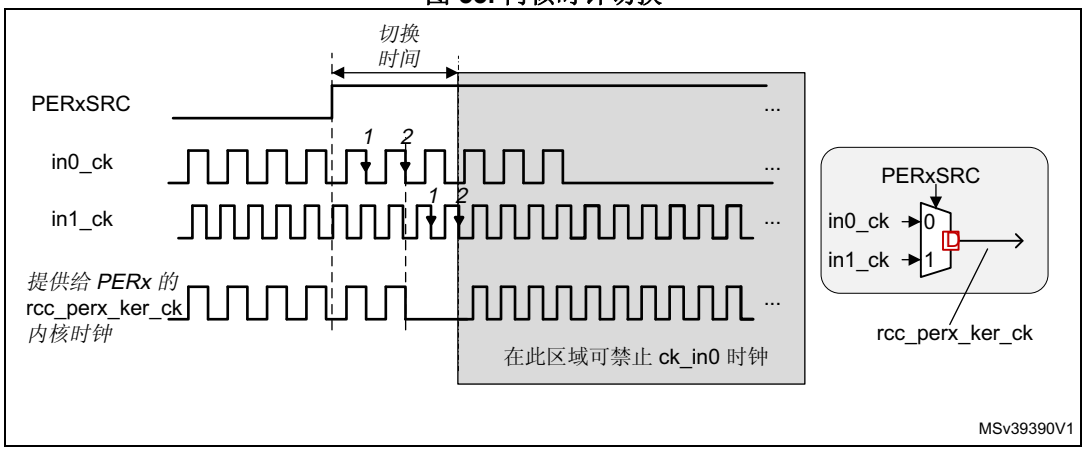
时钟开关和门控

- 时钟切换延迟

  内核时钟开关所选择的输入可动态更改，而不会产生杂波或违反时序。因此，只有在原始输入和新输入端均存在时钟时，才能从原始输入切换到新输入。否则，不会为外设提供时钟。要从这种情况中恢复，用户必须为两个输入提供有效时钟。

  从一个输入切换到另一个输入时，将对为外设提供的内核时钟进行门控，在最坏情况下，会在之前所选时钟的 2 个时钟周期内和新选择时钟的 2 个时钟周期内进行门控。如*图 53* 所示，在切换期间两个输入时钟均应存在。

图 53. 内核时钟切换



- 时钟使能延迟

  同样，为避免产生杂波，时钟门控逻辑会将使能命令（通常来自于内核时钟请求或 PERxEN 位）与所选时钟同步。

  – 在使能命令和时钟的第一个上升沿之间，可能会发生最长两个所使能时钟周期的延迟。使能命令可为 RCC_xxxxENR 寄存器的 PERxEN 位的上升沿，或者是外设所发出的内核时钟请求。

  – 在禁止命令和时钟的最后一个下降沿之间，可能会发生最长 1.5 个所禁止时钟周期的延迟。禁止命令可为 RCC_xxxxENR 寄存器的 PERxEN 位的下降沿，或者是外设所释放的内核时钟请求。

注： *内核时钟和总线接口时钟均会受此再同步延迟的影响。*

此外，如果应用程序首次使能未处于相同域内的外设，则会大幅延长时钟使能延迟。这是因为，外设所在的域可能处于 DStop 或 DStandby 模式下。该域必须先切换到 DRun 模式，然后应用程序才能使用该外设。

例如，如果 CPU 使能 D2 域中的外设，而 D2 域处于 DStop/DStandby 模式，则电源控制器 (PWR) 必须先为 D2 供电，然后 RCC 必须等待来自 PWR 的确认，之后才可使能 D2 域的时钟。为了正确处理这种情况，RCC 和 PWR 模块具有以下四个标志：

– *RCC 源控制寄存器 (RCC_CR)* 中的 D1CKRDY/D2CKRDY

– *PWR CPU 控制寄存器 (PWR_CPUCR)* 中的 SBF_D1 和 SBF_D2。

为避免发生上述问题，请按照以下顺序进行操作：

a) 通过在 RCC_xxxxENR 寄存器中为相应的 PERxEN 位写入“1”来使能外设时钟（即，分配外设），

b) 回读 RCC_xxxxENR 寄存器以确保写入缓冲器中不存在等待处理的前一次写操作。

c) 如果外设位于其他域中，请执行以下两步：

读取 DxCKRDY，直到其置“1”。

向 SBF_Dx 写入零并回读值，以检查外设所在的域是否仍处于 DStandby 模式下。如果相应的位读为“1”，则意味着域仍处于 DStandby 模式下。重复进行该操作，直到 SBF_Dx 等于“0”，然后继续执行其他步骤。

d) 对所使能外设的寄存器执行空读操作。该操作将至少占用 2 个时钟周期，该时间相当于使能命令的最大延迟时间。

e) 随后即可使用外设。

注： *如果总线接口时钟未激活，则不支持对外设寄存器进行读访问或写访问。读访问将返回无效数据。写访问将被忽略，但不会产生总线错误。*

8.5.11 外设时钟门控

如前文所述，每个外设均需一个名为 **rcc_perx_bus_ck**（针对外设“x”）的总线接口时钟。该时钟可为 APB、AHB 或 AXI 时钟，具体取决于与外设相连的总线。

对于用作 D1 域中外设的总线接口的时钟，可以是 **rcc_aclk**、**rcc_hclk3** 或 **rcc_pclk3**，具体取决于与每个外设相连的总线。为简单起见，这类时钟被称为 **rcc_bus_d1_ck**。

同样，名为 **rcc_bus_d2_ck** 的信号表示 **rcc_hclk1**、**rcc_hclk2**、**rcc_pclk1** 或 **rcc_pclk2**，具体取决于连接到 D2 域各个外设的总线。

类似地，信号 **rcc_bus_d3_ck** 表示 D3 中外设的 **rcc_hclk4** 或 **rcc_pclk4**。

某些外设（SAI、UART...）还需要一个专用时钟来用于其通信接口。该时钟通常与总线接口时钟异步。它被称为内核时钟 (**perx_ker_ckreq**)。可根据下文详细介绍的各个条件对两个时钟进行门控。

如*图 54* 所示，每个外设的内核和总线接口时钟的使能都取决于多个输入信号：

- PERxEN 和 PERxLPEN 位

  PERxEN 表示 CPU 的外设使能（分配）位。CPU 可通过 RCC_C1_xxxxENR 或 RCC_xxxxENR 寄存器向这些位写入“1”。
- PERxAMEN 位

  PERxAMEN 位属于 *RCC D3 自主模式寄存器 (RCC_D3AMR)*。
- CPU 状态（**c_sleep** 和 **c_deepsleep** 信号）
- D3 域状态（**d3_deepsleep** 信号）
- 外设自身的内核时钟请求 (**perx_ker_ckreq**)，前提是功能可用。

更多详细信息，请参见*第 8.5.10 节：外设分配*。

图 54. 外设内核时钟使能逻辑详细信息

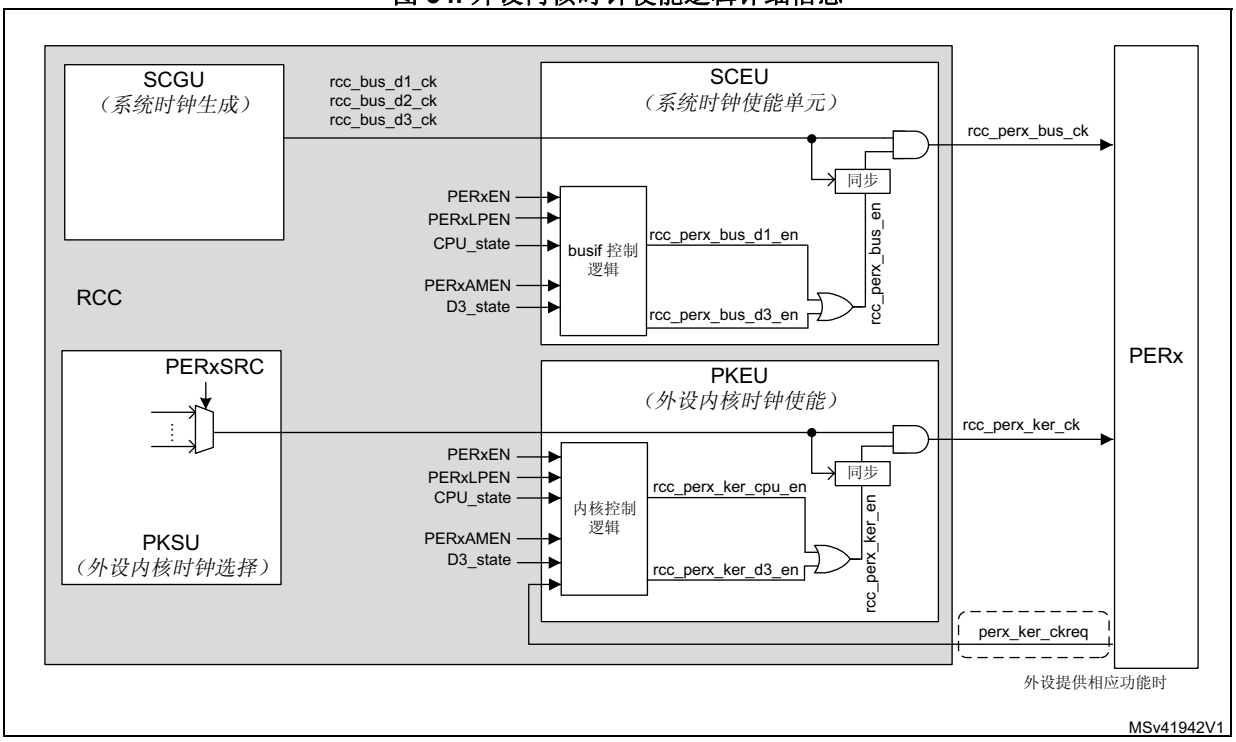


表 52 所示为 D1 或 D2 域中的外设以及 CPU 所分配外设的外设时钟使能逻辑的详细说明。

表 52. D1 和 D2 外设的外设时钟使能

PERxEN	PERxLPEN	PERxSRC	perx_ker_ckreq	CPU 状态	rcc_perx_ker_c_en	rcc_perx_bus_d1_en	注释
0	X	X	X	X	0	0	没有向外设提供时钟，因为 PERxEN = “0”
1	X	X	X	CRun	1	1	向外设提供内核和总线接口时钟，因为 CPU 处于 CRun 模式下且 PERxEN = “1”
1	0	X	X	CSleep	0	0	没有向外设提供时钟，因为 CPU 处于 CSleep 模式下且 PERxLPEN = “0”
1	1	X	X		1	1	向外设提供内核和总线接口时钟，因为 CPU 处于 CSleep 模式下且 PERxLPEN = “1”
1	0	X	X	CStop	0	0	没有向外设提供时钟，因为 PERxLPEN 位被设为“0”。
1	1	非 lsi_ck 且 非 lse_ck 且非 hsi_ker_ck 且非 csi_ker_ck	X		0	0	没有向外设提供时钟，因为 CPU 处于 CStop 模式且未选择 **lse_ck**、**lsi_ck**、**hsi_ker_ck** 或 **csi_ker_ck**。
1	1	lsi_ck 或 lse_ck	X		1 (1)	0	向外设提供内核时钟，因为 PERxEN = PERxLPEN = “1” 且选择了 **lsi_ck** 或 **lse_ck**。 没有提供总线接口时钟，因为 CPU 处于 CStop 模式
1	1	hsi_ker_ck 或 csi_ker_ck	1		1	0	向外设提供内核时钟，因为 **req_ker_perx** = “1”，PERxEN = PERxLPEN =1 且选择了 **hsi_ker_ck** 或 **csi_ker_ck**。 没有提供总线接口时钟，因为 CPU 处于 CStop 模式
1	1	hsi_ker_ck 或 csi_ker_ck	0		0	0	没有向外设提供时钟，因为 CPU 处于 CStop 模式，且不存在等待处理的内核时钟请求

1. 对于 RNG 模块，如果其分配到的 CPU 处于 CStop 模式，则不会传送内核时钟，即使选择的时钟为 lsi_ck 或 lse_ck 亦如此。

总之，满足以下条件时会为 D1 和 D2 域中的外设提供内核时钟：

1. CPU 处于 CRun 模式，且已分配外设。
2. CPU 处于 CSleep 模式，且已分配外设（PERxLPEN = “1”）。
3. CPU 处于 CStop 模式，已通过 PERxLPEN = “1” 分配外设，外设生成一个内核时钟请求，且所选时钟为 **hsi_ker_ck** 或 **csi_ker_ck**。
4. CPU 处于 CStop 模式，已通过 PERxLPEN = “1” 分配外设，外设的内核时钟源为 **lse_ck** 或 **lsi_ck**。

只有在满足条件 1 或 2 时才会向外设提供总线接口时钟。

表 53 给出了 D3 中所有外设的内核时钟使能逻辑的详细说明。

表 53. D3 外设的外设时钟使能

PERxEN	PERxLPEN	PERxAMEN	PERxSRC	perx_ker_ckreq	CPU 状态	D3 状态	rcc_perx_ker_d3_en	rcc_perx_bus_d3_en	注释
0	X	X	X	X	任意	任意	0	0	没有向外设提供时钟，因为 PERxEN = “0”
1	X	X	X	X	CRun	DRUN	1	1	向外设提供内核和总线接口时钟，因为 CPU 处于 CRun 模式下且 PERxEN = “1”
1	0	X	X	X	CSleep		0	0	没有向外设提供时钟，因为 CPU 处于 CSleep 模式下且 PERxLPEN = “0”
1	1	X	X	X			1	1	向外设提供内核和总线接口时钟，因为 CPU 处于 CSleep 模式下且 PERxLPEN = “1”
1	X	0	X	X	CStop		0	0	由于 CPU 处于 CStop 模式，且 PERxEN = “1”，因此将根据 D3 状态和 PERxAMEN 位进行内核时钟门控。 没有向外设提供时钟，因为 PERxAMEN = “0”。
1	X	1	X	X			1	1	提供内核和总线接口时钟，因为即使 CPU 处于 CStop 模式，D3 也仍处于 DRun 模式下，且 PERxEN 和 PERxAMEN 位置 “1”。
1	X	1	非 lse_ck 且 非 lsi_ck	0		DStop	0	0	没有向外设提供时钟，因为 D3 处于 DStop 模式下，**req_ker_perx** = “0”，且未选择 **lse_ck** 或 **lsi_ck**。
1	X	1	非 hsi_ker_ck 且非 csi_ker_ck 且 非 lse_ck 且 非 lsi_ck	1	CStop	DStop	0	0	没有向外设提供时钟，因为即使 **req_ker_perx** = “0”，也不会选择 **lse_ck**、**lsi_ck**、**hsi_ker_ck** 或 **csi_ker_ck**。
1	X	1	hsi_ker_ck 或 csi_ker_ck	1			1	0	向外设提供内核时钟，因为 **req_ker_perx** = “1”，PERxEN = PERxAMEN = “1”，且所选时钟为 **hsi_ker_ck** 或 **csi_ker_ck**。 没有提供总线接口时钟，因为 D3 处于 DStop 模式。
1	X	1	lse_ck 或 lsi_ck	X			1	0	向外设提供内核时钟，因为在 D3 处于停止模式时 PERxEN = PERxAMEN = “1”，且选择了 **lse_ck** 或 **lsi_ck**。 没有提供总线接口时钟，因为 D3 处于 DSTOP 模式。

总之，满足以下条件时会为 D3 的外设提供内核时钟：

1. CPU 处于 CRun 模式，且已分配外设。
2. CPU 处于 CSleep 模式，且已分配外设（PERxLPEN = “1”）。
3. CPU 处于 CStop 模式，已分配外设，且 D3 域处于 DRun 模式（PERxAMEN = “1”）。
4. CPU 处于 CStop 模式，已分配外设，D3 域处于 DStop 模式（PERxAMEN = “1”），外设正在生成一个内核时钟请求，且内核时钟源为 **hsi_ker_ck** 或 **csi_ker_ck**。
5. CPU 处于 CStop 模式，已分配外设，D3 域处于 DStop 模式下（PERxAMEN = “1”），且外设的内核时钟源为 **lse_ck** 或 **lsi_ck**。

只有在满足条件 1、2 或 3 时才会向外设提供总线接口时钟。

注： *当自主位置 1 时，表示相关外设将根据 D3 状态（而非 CPU 的模式）接收内核时钟。*

只有 I2C、U(S)ART 和 LPUART 外设才能请求内核时钟。借助该功能，外设能在保证最佳功耗的条件下传送数据。

专用于 D3 域中的某些外设的自主位支持在不激活 CPU 的情况下，与外部器件之间进行数据传送。

*为在电路处于停止模式时使 LPTIMER 使用 **lse_ck** 或 **lsi_ck**，用户应用必须通过 LPTIMxSEL 字段选择 **lsi_ck** 或 **lse_ck** 输入，并将位 LPTIMxAMEN 和 LPTIMxLPEN 置“1”。*

8.5.12 CPU 和总线矩阵时钟门控

对于每个域，都可以控制 CPU 时钟和总线矩阵时钟的激活/禁用。

有关命名约定的信息，请参见*第 8.5.11 节：外设时钟门控*。

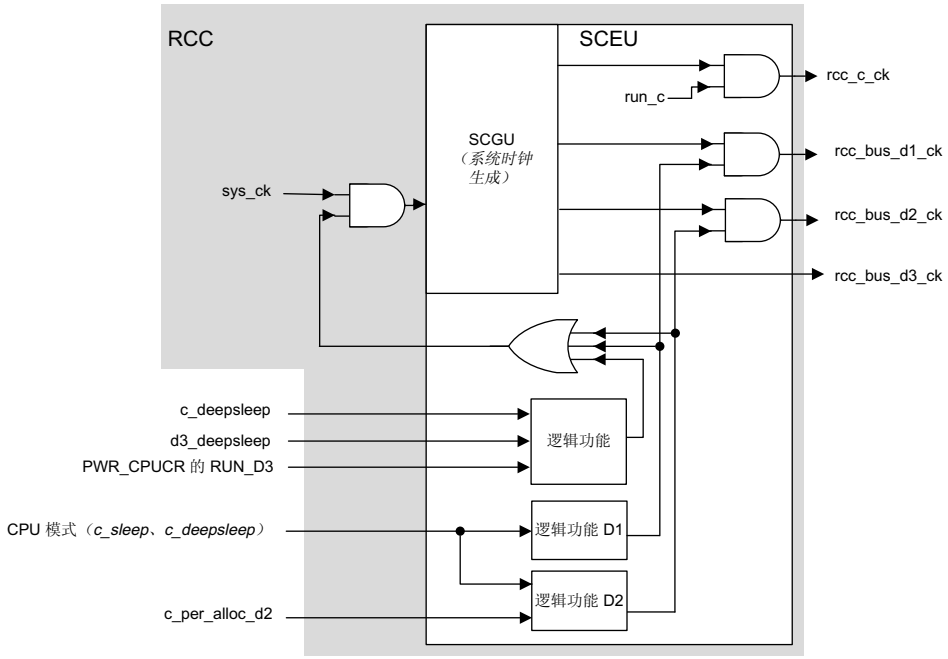
CPU、AHB 和 AXI 桥以及 APB 总线的时钟根据下述规则进行使能：

- 在 CPU 处于 CRun 模式时，使能 CPU 时钟 **rcc_c_ck**。
- 在 CPU 处于 CRun 模式时，使能 AXI 桥时钟。
- 在以下情况下使能 D2 域 AHB 桥时钟：
  - CPU 处于 CRun 模式，或
  - CPU 处于 CSleep 模式，且至少有一个连接到该总线的外设（主器件）的 PERxEN 和 PERxLPEN 均置“1”，或
  - CPU 处于 CSleep 模式，且至少有一个 APB 总线已使能其时钟。
- 在以下情况下使能 D3 域 AHB 桥时钟：
  - CPU 处于 CRun 或 CSleep 模式，或
  - RUN_D3 位置“1”（与 CPU 模式无关），或
  - **d3_deepsleep** 信号未激活 (0)（与 CPU 模式无关）。
- 在以下情况下使能 APB1,2,3 总线：
  - CPU 处于 CRun 模式，或
  - CPU 处于 CSleep 模式，且至少有一个连接到该总线的外设的 PERxEN 和 PERxLPEN 均置 1。
- 在以下情况下使能 APB4 总线：D3 域处于 DRun 模式。

如*图 55* 所示，各个域的内核和总线时钟的使能取决于多个输入信号：

- 来自 CPU 的**c_sleep** 和 **c_deepsleep** 信号，
- **d3_sleepdeep** 信号，
- D2 域中外设的 RCC_xxxxENR.PERxEN 位

图 55. 总线时钟使能逻辑



8.6 RCC 中断

RCC 提供 3 条中断线：

- **rcc_it**：通用中断线，在 PLL 就绪或振荡器就绪时提供相关事件。
- **rcc_hsecss_it**：专用于对 HSE 时钟安全系统进行故障检测的中断线。
- **rcc_lsecss_it**：专用于对 LSE 时钟安全系统进行故障检测的中断线。

除 HSE CSS 故障外，中断使能通过 *RCC 时钟源中断使能寄存器 (RCC_CIER)* 进行控制。使能 HSE CSS 功能时，无法屏蔽中断的生成。

可通过 *RCC 时钟源中断标志寄存器 (RCC_CIFR)* 检查中断标志，并且可通过 *RCC 时钟源中断清零寄存器 (RCC_CICR)* 将这些标志清零。

注： *如果相应的中断使能位未置 1，则中断标志不相关。*

表 54 汇总了中断源及其控制方式。

表 54. 中断源和控制

中断源	说明	中断使能	中断清除操作	中断线
LSIRDYF	LSI 就绪	LSIRDYIE	将 LSIRDYC 位置“1”。	rcc_it
LSERDYF	LSE 就绪	LSERDYIE	将 LSERDYC 位置“1”。	
HSIDRYF	HSI 就绪	HSIDRYIE	将 HSIRDYC 位置“1”。	
HSERDYF	HSE 就绪	HSERDYIE	将 HSERDYC 位置“1”。	
CSIRDYF	CSI 就绪	CSIRDYIE	将 CSIRDYC 位置“1”。	
HSI48RDYF	HSI48 就绪	HSI48RDYIE	将 HSI48RDYC 位置“1”。	
PLL1RDYF	PLL1 就绪	PLL1RDYIE	将 PLL1RDYC 位置“1”。	
PLL2RDYF	PLL2 就绪	PLL2RDYIE	将 PLL2RDYC 位置“1”。	
PLL3RDYF	PLL3 就绪	PLL3RDYIE	将 PLL3RDYC 位置“1”。	
LSECSSF	LSE 时钟安全系统故障	LSECSSFIE (1)	将 LSECSSC 位置“1”。	rcc_lsecss_it
HSECSSF	HSE 时钟安全系统故障	-(2)	将 HSECSSC 位置“1”。	rcc_hsecss_it

1. 为生成中断，还必须使能安全系统功能（LSECSSON = “1”）。
2. 使能安全系统功能（HSECSSON = “1”）时，不能屏蔽该中断。

8.7 RCC 寄存器说明

8.7.1 寄存器映射概述

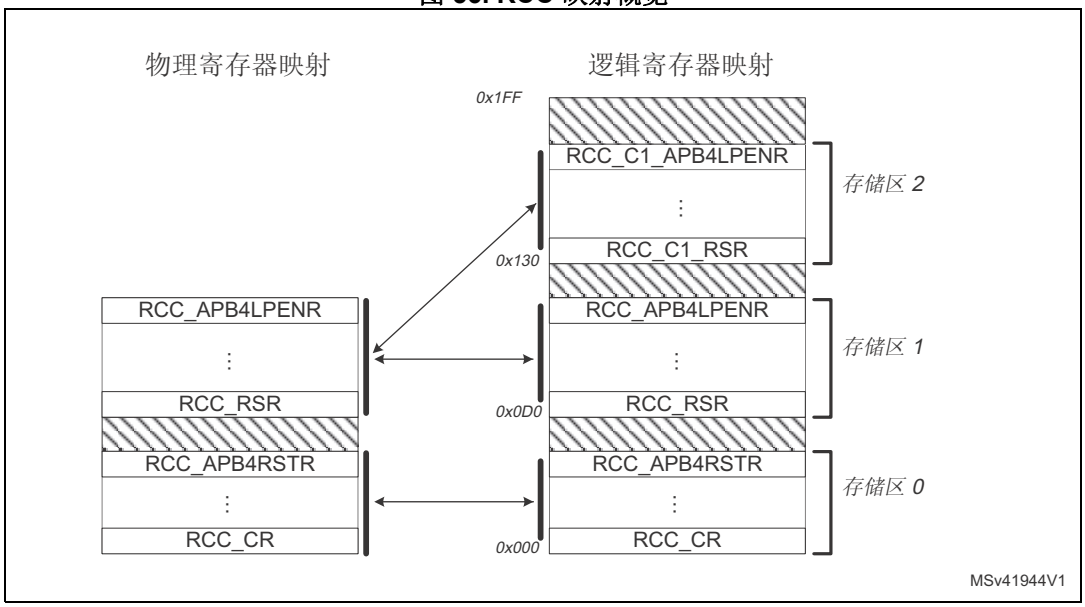
请注意，可在以下两个不同的偏移地址处控制 PERxEN 和 PERxLPEN 位：0x0D0 和 0x130。因此应用程序可使用以下寄存器执行相应控制：

- RCC_xxxxENR 或 RCC_C1_xxxxENR，控制 PERxEN 位
- RCC_xxxxLPENR 或 RCC_C1_xxxxLPENR，控制 PERxLPEN 位
- RCC_RSR 或 RCC_C1_RSR，控制复位标志状态位

该功能可确保与本系列其他产品的兼容性。

图 56 所示为 RCC 映射概览。

图 56. RCC 映射概览



8.7.2 RCC 源控制寄存器 (RCC_CR)

RCC Source Control Register

偏移地址：0x000

复位值：0x0000 0001

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	PLL3RDY	PLL3ON	PLL2RDY	PLL2ON	PLL1RDY	PLL1ON	Res.	Res.	Res.	Res.	HSECSSON	HSEBYP	HSERDY	HSEON
		r	rw	r	rw	r	rw					rs	rw	r	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
D2CKRDY	D1CKRDY	HSI48RDY	HSI48ON	Res.	Res.	CSIKERON	CSIRDY	CSION	Res.	HSIDIVF	HSIDIV[1:0]		HSIRDY	HSIKERON	HSION
r	r	r	rw			rw	r	rw		r	rw	rw	r	rw	rw

位 31:30 保留，必须保持复位值。

位 29 **PLL3RDY**：PLL3 时钟就绪标志 (PLL3 clock ready flag)
由硬件置 1，用以指示 PLL3 已锁定。
0：PLL3 未锁定（复位后的默认值）
1：PLL3 已锁定

位 28 **PLL3ON**：PLL3 使能 (PLL3 enable)
由软件置 1 和清零，用于使能 PLL3。
当进入停机或待机模式时由硬件清零。
0：PLL3 关闭（复位后的默认值）
1：PLL3 开启

位 27 **PLL2RDY**：PLL2 时钟就绪标志 (PLL2 clock ready flag)
由硬件置 1，用以指示 PLL2 已锁定。
0：PLL2 未锁定（复位后的默认值）
1：PLL2 已锁定

位 26 **PLL2ON**：PLL2 使能 (PLL2 enable)
由软件置 1 和清零，用于使能 PLL2。
当进入停机或待机模式时由硬件清零。
0：PLL2 关闭（复位后的默认值）
1：PLL2 开启

位 25 **PLL1RDY**：PLL1 时钟就绪标志 (PLL1 clock ready flag)
由硬件置 1，用以指示 PLL1 已锁定。
0：PLL1 未锁定（复位后的默认值）
1：PLL1 已锁定

位 24 **PLL1ON**：PLL1 使能 (PLL1 enable)
由软件置 1 和清零，用于使能 PLL1。
当进入停机或待机模式时由硬件清零。请注意，如果将 PLL1 输出用作系统时钟，则硬件可防止向该位写入“0”。
0：PLL1 关闭（复位后的默认值）
1：PLL1 开启

位 23:20 保留，必须保持复位值。

位 19 **HSECSSON**：HSE 时钟安全系统使能 (HSE Clock Security System enable)

由软件置 1，用于使能 HSE 上的时钟安全系统。
该位“仅置 1”（由系统复位或在系统进入待机模式时禁止）。
当 HSECSSON 置 1 时，时钟监测器将在 HSE 就绪时由硬件使能，并在检测到振荡器故障时由硬件禁止。
0：HSE 上的时钟安全系统关闭（时钟监测器关闭）（复位后的默认值）
1：HSE 上的时钟安全系统开启（如果 HSE 振荡器稳定，则时钟监测器开启；否则将关闭）。

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

由软件置 1 和清零，用于用外部时钟旁路振荡器。
外部时钟必须通过 HSEON 位使能才能为器件所用。
HSEBYP 只有在 HSE 振荡器已禁止的情况下才可写入。
0：不旁路 HSE 振荡器（复位后的默认值）
1：外部时钟旁路 HSE 振荡器

位 17 **HSERDY**：HSE 时钟就绪标志 (HSE clock ready flag)

由硬件置 1，用以指示 HSE 振荡器已稳定。
0：HSE 时钟未就绪（复位后的默认值）
1：HSE 时钟已就绪

位 16 **HSEON**：HSE 时钟使能 (HSE clock enable)

由软件置 1 和清零。
由硬件清零，用于在进入停止或待机模式时停止 HSE。
如果（通过软件复用）直接将 HSE 用作系统时钟，或将 HSE 选作已使能的 PLL1（将 PLL1ON 位置“1”）的参考时钟，则不能将该位清零。
0：HSE 关闭（复位后的默认值）
1：HSE 开启

位 15 **D2CKRDY**：D2 域时钟就绪标志 (D2 domain clocks ready flag)

由硬件置 1，用于指示 D2 域时钟可用。
0：D2 域时钟不可用（复位后的默认值）
1：D2 域时钟可用

位 14 **D1CKRDY**：D1 域时钟就绪标志 (D1 domain clocks ready flag)

由硬件置 1，用于指示 D1 域时钟（CPU、总线和外设）可用。
0：D1 域时钟不可用（复位后的默认值）
1：D1 域时钟可用

位 13 **HSI48RDY**：HSI48 时钟就绪标志 (HSI48 clock ready flag)

由硬件置 1，用以指示 HSI48 振荡器已稳定。
0：HSI48 时钟未就绪（复位后的默认值）
1：HSI48 时钟已就绪

位 12 **HSI48ON**：HSI48 时钟使能 (HSI48 clock enable)

由软件置 1，在系统进入停止或待机模式时由软件或硬件清零。
0：HSI48 关闭（复位后的默认值）
1：HSI48 开启

位 11:10 保留，必须保持复位值。

位 9 **CSIKERON**：停止模式下的 CSI 时钟使能 (CSI clock enable in Stop mode)

由软件置 1 和复位，即使在停止模式下也可强制 CSI 开启，从而可快速用作某些外设的内核时钟。该位对 CSION 的值没有任何影响。

0：对 CSI 无影响（复位后的默认值）

1：即使在停止模式下也强制 CSI 开启

位 8 **CSIRDY**：CSI 时钟就绪标志 (CSI clock ready flag)

由硬件置 1，用以指示 CSI 振荡器已稳定。仅在由 CSION 使能 RC 时才会激活该位（由 CSIKERON 或外设请求使能 CSI 时不会激活该位）。

0：CSI 时钟未就绪（复位后的默认值）

1：CSI 时钟已就绪

位 7 **CSION**：CSI 时钟使能 (CSI clock enable)

由软件置 1 和复位，用于使能/禁止系统和/或外设的 CSI 时钟。

如果 STOPWUCK =“1”或 STOPKERWUCK =“1”，则由硬件置 1，以在系统退出停止模式时强制 CSI 开启。

如果（通过软件复用）直接将 CSI 用作系统时钟，或将 CSI 选作已使能的 PLL1（将 PLL1ON 位置“1”）的参考时钟，则不能将该位清零。

0：CSI 关闭（复位后的默认值）

1：CSI 开启

位 6 保留，必须保持复位值。

位 5 **HSIDIVF**：HSI 分频器标志 (HSI divider flag)

由硬件置 1 和复位。

由于对 HSIDIV 的写操作不会直接对频率起作用，因此该标志指示的是 HSI 分频器的当前状态。HSIDIVF 会在 HSIDIV 值发生变化时立即变为“0”，并会在输出频率与编程到 HSIDIV 中的值匹配时重新置“1”。

0：新分频比尚未传播到 **hsi(_ker)_ck**（复位后的默认值）

1：**hsi(_ker)_ck** 时钟频率反映新 HSIDIV 值

位 4:3 **HSIDIV[1:0]**：HSI 时钟分频器 (HSI clock divider)

由软件置 1 和复位。

可使用这些位来选择一个分频比，以配置所需 HSI 时钟频率。如果将 HSI 选作至少一个已使能 PLL（PLLxON 位置“1”）的参考时钟，则不能更改 HSIDIV。在这种情况下，将忽略新 HSIDIV 值。

00：采用 1 分频，**hsi(_ker)_ck** = 64 MHz（复位后的默认值）

01：采用 2 分频，**hsi(_ker)_ck** = 32 MHz

10：采用 4 分频，**hsi(_ker)_ck** = 16 MHz

11：采用 8 分频，**hsi(_ker)_ck** = 8 MHz

位 2 **HSIRDY**：HSI 时钟就绪标志 (HSI clock ready flag)

由硬件置 1，用以指示 HSI 振荡器已稳定。

0：HSI 时钟未就绪（复位后的默认值）

1：HSI 时钟已就绪

位 1 **HSIKERON**：停止模式下的高速内部时钟使能 (High Speed Internal clock enable in Stop mode)

由软件置 1 和复位，即使在停止模式下也可强制 HSI 开启，从而可快速用作外设的内核时钟。该位对 HSION 的值没有任何影响。

0：对 HSI 无影响（复位后的默认值）

1：即使在停止模式下也强制 HSI 开启

位 0 **HSION**：高速内部时钟使能 (High Speed Internal clock enable)
由软件置 1 和清零。
如果 STOPWUCK = “0”或 STOPKERWUCK = “0”，则由硬件置 1，以在产品退出停止模式时强制 HSI 开启。
由硬件置 1，用于在产品退出待机模式时或在用作系统时钟源的 HSE 发生故障时强制 HSI 开启。
如果（通过软件复用）直接将 HSI 用作系统时钟，或将 HSI 选作已使能的 PLL1（将 PLL1ON 位置 “1”）的参考时钟，则不能将该位清零。
0：HSI 关闭
1：HSI 开启（复位后的默认值）

8.7.3 RCC 内部时钟源校准寄存器 (RCC_ICSCR)

RCC Internal Clock Source Calibration Register

偏移地址：0x004

复位值：0x4xxx 0xxx

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	CSITRIM[4:0]					CSICAL[7:0]								HSITRIM[5:4]	
	rw					r								rw	

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

位 31 保留，必须保持复位值。

位 30:26 **CSITRIM[4:0]**：CSI 时钟微调 (CSI clock trimming)
由软件置 1，用于调整校准。
CSITRIM 字段与复位阶段加载的工程选项字节 (FLASH_CSI_opt) 相加，以构成校准微调值。
CSICAL = CSITRIM + FLASH_CSI_opt。
注： 字段的复位值为 0x10。

位 25:18 **CSICAL[7:0]**：CSI 时钟校准 (CSI clock calibration)
由硬件通过系统复位 **nreset** 期间的选项字节加载来置 1。
由软件通过微调位 CSITRIM 进行调节。
该字段表示工程选项字节校准值与 CSITRIM 位值之和

位 17:12 **HSITRIM[5:0]**：HSI 时钟微调 (HSI clock trimming)
由软件置 1，用于调整校准。
HSITRIM 字段与复位阶段加载的工程选项字节 (FLASH_HSI_opt) 相加，以构成校准微调值。
HSICAL = HSITRIM + FLASH_HSI_opt。
注： 字段的复位值为 0x20。

位 11:0 **HSICAL[11:0]**：HSI 时钟校准 (HSI clock calibration)
由硬件通过系统复位 **nreset** 期间的选项字节加载来置 1。
由软件通过微调位 HSITRIM 进行调节。
该字段表示工程选项字节校准值与 HSITRIM 位值之和。

8.7.4 RCC 时钟恢复 RC 寄存器 (RCC_CRRCR)

RCC Clock Recovery RC Register

偏移地址：0x008

复位值：0x0000 0xxx

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	HSI48CAL[9:0]									
						r									

位 31:10 保留，必须保持复位值。

位 9:0 **HSI48CAL[9:0]**：内部 RC 48 MHz 时钟校准 (Internal RC 48 MHz clock calibration)
由硬件通过系统复位 **nreset** 期间的选项字节加载来置 1。
只读。

8.7.5 RCC 时钟配置寄存器 (RCC_CFGR)

RCC Clock Configuration Register

偏移地址：0x010

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MCO2[2:0]			MCO2PRE[3:0]				MCO1[2:0]			MCO11PRE[3:0]				Res.	Res.
rw			rw				rw			rw					

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TIMPRE	HRTIMSEL	RTCPRE[5:0]						STOPKERWUCK	STOPWUCK	SWS[2:0]			SW[2:0]		
rw	rw	rw						rw	rw	r			rw		

位 31:29 **MCO2[2:0]**：微控制器时钟输出 2 (Micro-controller clock output 2)
由软件置 1 和清零。时钟源选择可能会造成对 MCO2 的干扰。
强烈建议仅在复位后但在使能外部振荡器和 PLL 之前来配置这些位。
000：选择系统时钟 (**sys_ck**)（复位后的默认值）
001：选择 PLL2 振荡器时钟 (**pll2_p_ck**)
010：选择 HSE 时钟 (**hse_ck**)
011：选择 PLL1 时钟 (**pll1_p_ck**)
100：选择 CSI 时钟 (**csi_ck**)
101：选择 LSI 时钟 (**lsi_ck**)
其它值：保留

位 28:25 **MCO2PRE[3:0]**：MCO2 预分频器 (MCO2 prescaler)
由软件置 1 和清零，用于配置 MCO2 的预分频器。对此预分频器进行修改可能会对 MCO2 造成干扰。强烈建议仅在复位后且在使能外部振荡器和 PLL 之前进行此分频器的更改。
0000：禁止预分频器（复位后的默认值）
0001：1 分频（旁路）
0010：2 分频
0011：3 分频
0100：4 分频
...
1111：15 分频

位 24:22 **MCO1[2:0]**：微控制器时钟输出 1 (Micro-controller clock output 1)
由软件置 1 和清零。时钟源选择可能会造成对 MCO1 的干扰。
强烈建议仅在复位后但在使能外部振荡器和 PLL 之前来配置这些位。
000：选择 HSI 时钟 (**hsi_ck**)（复位后的默认值）
001：选择 LSE 振荡器时钟 (**lse_ck**)
010：选择 HSE 时钟 (**hse_ck**)
011：选择 PLL1 时钟 (**pll1_q_ck**)
100：选择 HSI48 时钟 (**hsi48_ck**)
其它值：保留

位 21:18 **MCO1PRE[3:0]**：MCO1 预分频器 (MCO1 prescaler)

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

0000：禁止预分频器（复位后的默认值）
0001：1 分频（旁路）
0010：2 分频
0011：3 分频
0100：4 分频
...
1111：15 分频

位 17:16 保留，必须保持复位值。

位 15 **TIMPRE**：定时器时钟预分频器选择 (Timers clocks prescaler selection)

此位由软件置 1 和复位，用于控制连接到 APB1 和 APB2 域的所有定时器的时钟频率。

0：如果 D2PPREx 对应于 1 或 2 分频，则定时器内核时钟为 **rcc_hclk1**；否则为 2 x $F_{rcc_pclkx_d2}$（复位后的默认值）
1：如果 D2PPREx 对应于 1、2 或 4 分频，则定时器内核时钟为 **rcc_hclk1**；否则为 4 x $F_{rcc_pclkx_d2}$
请参考*表 47：时钟定时器与 pclk 之比*。

位 14 **HRTIMSEL**：高分辨率定时器时钟预分频器选择 (High Resolution Timer clock prescaler selection)

此位由软件置 1 和复位，用于控制高分辨率定时器 (HRTIM) 的时钟频率。
0：HRTIM 预分频器时钟源与其他定时器的时钟源相同。（复位后的默认值）
1：HRTIM 预分频器时钟源为 CPU 时钟 (**rcc_c_ck**)。

位 13:8 **RTCPRE[5:0]**：适用于 RTC 时钟的 HSE 分频系数 (HSE division factor for RTC clock)

由软件置 1 和清零，用于对 HSE 进行分频，进而为 RTC 生成时钟。
小心：软件必须正确设置这些位，以确保提供给 RTC 的时钟小于 1 MHz。在选择 RTC 时钟源之前必须配置这些位。

000000：无时钟（复位后的默认值）
000001：无时钟
000010：HSE/2
000011：HSE/3
000100：HSE/4
...
111110：HSE/62
111111：HSE/63

位 7 **STOPKERWUCK**：将系统从停止模式唤醒后的内核时钟选择 (Kernel clock selection after a wake up from system Stop)

由软件置 1 和复位，用于选择将系统从停止模式唤醒后的内核时钟。
0：选择 HSI 作为将系统从停止模式唤醒后的时钟（复位后的默认值）
1：选择 CSI 作为将系统从停止模式唤醒后的时钟
详细信息，请参见*第 8.5.7 节：在停止和待机模式下处理时钟发生器*。

位 6 **STOPWUCK**：将系统从停止模式唤醒后的系统时钟选择 (System clock selection after a wake up from system Stop)

由软件置 1 和复位，用于选择将系统从停止模式唤醒后的系统时钟。
所选时钟还用作 HSE 上的时钟安全系统的紧急时钟。
0：选择 HSI 作为将系统从停止模式唤醒后的时钟（复位后的默认值）
1：选择 CSI 作为将系统从停止模式唤醒后的时钟
详细信息，请参见*第 8.5.7 节：在停止和待机模式下处理时钟发生器*。

注意： 时钟安全系统已使能（通过 HSECSSON 位）且系统时钟为 HSE（SWS=“10”）或者请求开启 HSE（SW=“10”）时，不能修改 STOPWUCK。

位 5:3 **SWS[2:0]**：系统时钟切换状态 (System clock switch status)
由硬件置 1 和复位，用于指示用作系统时钟的时钟源。
000：将 HSI 用作系统时钟 (**hsi_ck**)（复位后的默认值）
001：将 CSI 用作系统时钟 (**csi_ck**)
010：将 HSE 用作系统时钟 (**hse_ck**)
011：将 PLL1 用作系统时钟 (**pll1_p_ck**)
其它值：保留

位 2:0 **SW[2:0]**：系统时钟切换 (System clock switch)
由软件置 1 和复位，用于选择系统时钟源 (**sys_ck**)。
由硬件置 1，用于：
– 在系统退出停止模式时强制选择 HSI 或 CSI（具体取决于 STOPWUCK 选择）
– 在 HSE 直接或间接用作系统时钟的情况下发生故障时，强制选择 HSI。
000：将 HSI 选作系统时钟 (**hsi_ck**)（复位后的默认值）
001：将 CSI 选作系统时钟 (**csi_ck**)
010：将 HSE 选作系统时钟 (**hse_ck**)
011：将 PLL1 选作系统时钟 (**pll1_p_ck**)
其它值：保留

8.7.6 RCC 域 1 时钟配置寄存器 (RCC_D1CFGR)

RCC Domain 1 Clock Configuration Register

偏移地址：0x018

复位值：0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	D1CPRE[3:0]				Res.	D1PPRE[2:0]			HPRE[3:0]			
				rw					rw			rw			

位 31:12 保留，必须保持复位值。

位 11:8 **D1CPRE[3:0]**：D1 域内核预分频器 (D1 domain Core prescaler)
由软件置 1 和复位，用于控制 D1 域 CPU 时钟分频系数。
更改此分频比会影响 CPU 时钟以及所有总线矩阵时钟的频率。
时钟按新预分频系数进行分频。在 D1CPRE 更新后，该系数介于 **rcc_pclk[4:1]** 的最慢 APB 时钟的 1 到 16 个周期之间。应用程序可通过回读该寄存器来检查是否将新分频系数考虑在内。
0xxx：sys_ck**sys_ck** 未分频（复位后的默认值）
1000：**sys_ck** 2 分频
1001：**sys_ck** 4 分频
1010：**sys_ck** 8 分频
1011：**sys_ck** 16 分频
1100：**sys_ck** 64 分频
1101：**sys_ck** 128 分频
1110：**sys_ck** 256 分频
1111：**sys_ck** 512 分频

位 7 保留，必须保持复位值。

位 6:4 **D1PPRE[2:0]**：D1 域 APB3 预分频器 (D1 domain APB3 prescaler)

由软件置 1 和复位，用于控制 **rcc_pclk3** 的分频系数。

在 D1PPRE 写入后，时钟将由介于 1 到 16 个 **rcc_hclk3** 周期之间的新预分频系数进行分频。

0xx：**rcc_pclk3** = **rcc_hclk3**（复位后的默认值）

100：**rcc_pclk3** = **rcc_hclk3** / 2

101：**rcc_pclk3** = **rcc_hclk3** / 4

110：**rcc_pclk3** = **rcc_hclk3** / 8

111：**rcc_pclk3** = **rcc_hclk3** / 16

位 3:0 **HPRE[3:0]**：D1 域 AHB 预分频器 (D1 domain AHB prescaler)

由软件置 1 和复位，用于控制 **rcc_hclk3** 和 **rcc_aclk** 的分频系数。更改此分频比会影响所有总线矩阵时钟的频率。

0xxx：**rcc_hclk3** = **sys_d1cpre_ck**（复位后的默认值）

1000：**rcc_hclk3** = **sys_d1cpre_ck** / 2

1001：**rcc_hclk3** = **sys_d1cpre_ck** / 4

1010：**rcc_hclk3** = **sys_d1cpre_ck** / 8

1011：**rcc_hclk3** = **sys_d1cpre_ck** / 16

1100：**rcc_hclk3** = **sys_d1cpre_ck** / 64

1101：**rcc_hclk3** = **sys_d1cpre_ck** / 128

1110：**rcc_hclk3** = **sys_d1cpre_ck** / 256

1111：**rcc_hclk3** = **sys_d1cpre_ck** / 512

*注： 在 HPRE 更新后，时钟将由介于 **rcc_pclk[4:1]** 内最慢 APB 时钟的 1 到 16 个周期之间的新预分频系数进行分频。*

*注： 另请注意，**rcc_hclk3** = **rcc_aclk**。*

注意： 使用电压调节功能时必须谨慎。由于新分频系数存在传播延迟，因此在预分频系数发生更改之后以及在 V_{CORE} 电压下降之前，必须对该寄存器进行读取，以检查是否已将新预分频器值考虑在内。

根据时钟源频率和电压范围，软件应用程序必须在 HPRE 中编程一个合适的值，以确保系统频率不会超出最大频率。

8.7.7 RCC 域 2 时钟配置寄存器 (RCC_D2CFGR)

RCC Domain 2 Clock Configuration Register

偏移地址：0x01C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	D2PPRE2[2:0]			Res.	D2PPRE1[2:0]			Res.	Res.	Res.	Res.
					rw				rw						

位 31:11 保留，必须保持复位值。

位 10:8 **D2PPRE2[2:0]**：D2 域 APB2 预分频器 (D2 domain APB2 prescaler)
由软件置 1 和复位，用于控制 D2 域 APB2 时钟分频系数。
在 D2PPRE2 写入后，时钟将由介于 1 到 16 个 **rcc_hclk1** 周期之间的新预分频系数进行分频。
0xx：**rcc_pclk2 = rcc_hclk1**（复位后的默认值）
100：**rcc_pclk2 = rcc_hclk1** / 2
101：**rcc_pclk2 = rcc_hclk1** / 4
110：**rcc_pclk2 = rcc_hclk1** / 8
111：**rcc_pclk2 = rcc_hclk1** / 16

位 7 保留，必须保持复位值。

位 6:4 **D2PPRE1[2:0]**：D2 域 APB1 预分频器 (D2 domain APB1 prescaler)
由软件置 1 和复位，用于控制 D2 域 APB1 时钟分频系数。
在 D2PPRE1 写入后，时钟将由介于 1 到 16 个 **rcc_hclk1** 周期之间的新预分频系数进行分频。
0xx：**rcc_pclk1 = rcc_hclk1**（复位后的默认值）
100：**rcc_pclk1 = rcc_hclk1** / 2
101：**rcc_pclk1 = rcc_hclk1** / 4
110：**rcc_pclk1 = rcc_hclk1** / 8
111：**rcc_pclk1 = rcc_hclk1** / 16

位 3:0 保留，必须保持复位值。

8.7.8 RCC 域 3 时钟配置寄存器 (RCC_D3CFGR)

RCC Domain 3 Clock Configuration Register

偏移地址：0x020

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	D3PPRE[2:0]			Res.			
									rw						

位 31:7 保留，必须保持复位值。

位 6:4 **D3PPRE[2:0]**：D3 域 APB4 预分频器 (D3 domain APB4 prescaler)

由软件置 1 和复位，用于控制 D3 域 APB4 时钟分频系数。

在 D3PPRE 写入后，时钟将由介于 1 到 16 个 **rcc_hclk4** 周期之间的新预分频系数进行分频。

0xx：**rcc_pclk4 = rcc_hclk4**（复位后的默认值）

100：**rcc_pclk4 = rcc_hclk4** / 2

101：**rcc_pclk4 = rcc_hclk4** / 4

110：**rcc_pclk4 = rcc_hclk4** / 8

111：**rcc_pclk4 = rcc_hclk4** / 16

位 3:0 保留，必须保持复位值。

8.7.9 RCC PLL 时钟源选择寄存器 (RCC_PLLCKSELR)

RCC PLLs Clock Source Selection Register

偏移地址：0x028

复位值：0x0202 0200

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	DIVM3[5:0]						Res.	Res.	DIVM2[5:4]	
						rw								rw	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DIVM2[3:0]				Res.	Res.	DIVM1[5:0]						Res.	Res.	PLLSRC[1:0]	
rw						rw								rw	

位 31:26 保留，必须保持复位值。

位 25:20 **DIVM3[5:0]**：PLL3 的预分频器 (Prescaler for PLL3)

由软件置 1 和清零，用于配置 PLL3 的预分频器。
使能 PLL3（PLL3ON =“1”）时，硬件不允许对该预分频器进行任何修改。
为实现节能，不使用 PLL3 时，必须将 DIVM3 的值设为“0”。
000000：禁止预分频器（复位后的默认值）
000001：1 分频（旁路）
000010：2 分频
000011：3 分频
...
100000：32 分频（复位后的默认值）
...
111111：63 分频

位 19:18 保留，必须保持复位值。

位 17:12 **DIVM2[5:0]**：PLL2 的预分频器 (Prescaler for PLL2)

由软件置 1 和清零，用于配置 PLL2 的预分频器。
使能 PLL2（PLL2ON =“1”）时，硬件不允许对该预分频器进行任何修改。
为实现节能，不使用 PLL2 时，必须将 DIVM2 的值设为“0”。
000000：禁止预分频器
000001：1 分频（旁路）
000010：2 分频
000011：3 分频
...
100000：32 分频（复位后的默认值）
...
111111：63 分频

位 11:10 保留，必须保持复位值。

位 9:4 **DIVM1[5:0]**：PLL1 的预分频器 (Prescaler for PLL1)

由软件置 1 和清零，用于配置 PLL1 的预分频器。

使能 PLL1（PLL1ON =“1”）时，硬件不允许对该预分频器进行任何修改。

为实现节能，不使用 PLL1 时，必须将 DIVM1 的值设为“0”。

000000：禁止预分频器

000001：1 分频（旁路）

000010：2 分频

000011：3 分频

...

100000：32 分频（复位后的默认值）

...

111111：63 分频

位 3:2 保留，必须保持复位值。

位 1:0 **PLLSRC[1:0]**：DIVMx 和 PLL 时钟源选择 (DIVMx and PLLs clock source selection)

由软件置 1 和复位，用于选择 PLL 时钟源。

这些位只能在所有 PLL 均已禁止时写入。

为实现节能，不使用 PLL 时，必须将 PLLSRC 的值设为“11”。

00：将 HSI 选作 PLL 时钟 (**hsi_ck**)（复位后的默认值）

01：将 CSI 选作 PLL 时钟 (**csi_ck**)

10：将 HSE 选作 PLL 时钟 (**hse_ck**)

11：未向 DIVMx 分频器和 PLL 发送任何时钟

8.7.10 RCC PLL 配置寄存器 (RCC_PLLCFGR)

RCC PLLs Configuration Register

偏移地址：0x02C

复位值：0x01FF 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	DIVR3EN	DIVQ3EN	DIVP3EN	DIVR2EN	DIVQ2EN	DIVP2EN	DIVR1EN	DIVQ1EN	DIVP1EN
							rw	rw	rw	rw	rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	PLL3RGE[1:0]		PLL3VCOSEL	PLL3FRACEN	PLL2RGE[1:0]		PLL2VCOSEL	PLL2FRACEN	PLL1RGE[1:0]		PLL1VCOSEL	PLL1FRACEN
				rw		rw	rw	rw		rw	rw	rw		rw	rw

位 31:25 保留，必须保持复位值。

位 24 **DIVR3EN**：PLL3 DIVR 分频器输出使能 (PLL3 DIVR divider output enable)
由软件置 1 和复位，用于使能 PLL3 的 **pll3_r_ck** 输出。
为实现节能，不使用 **pll3_r_ck** 时，必须将 DIVR3EN 和 DIVR3 位设为“0”。
该位只能在 PLL3 禁止（PLL3ON = “0”且 PLL3RDY = “0”）时写入。
0：禁止 **pll3_r_ck** 输出
1：使能 **pll3_r_ck** 输出（复位后的默认值）

位 23 **DIVQ3EN**：PLL3 DIVQ 分频器输出使能 (PLL3 DIVQ divider output enable)
由软件置 1 和复位，用于使能 PLL3 的 **pll3_q_ck** 输出。
为实现节能，不使用 **pll3_r_ck** 时，必须将 DIVR3EN 和 DIVR3 位设为“0”。
该位只能在 PLL3 禁止（PLL3ON = “0”且 PLL3RDY = “0”）时写入。
0：禁止 **pll3_q_ck** 输出
1：使能 **pll3_q_ck** 输出（复位后的默认值）

位 22 **DIVP3EN**：PLL3 DIVP 分频器输出使能 (PLL3 DIVP divider output enable)
由软件置 1 和复位，用于使能 PLL3 的 **pll3_p_ck** 输出。
该位只能在 PLL3 禁止（PLL3ON = “0”且 PLL3RDY = “0”）时写入。
为实现节能，不使用 **pll3_r_ck** 时，必须将 DIVR3EN 和 DIVR3 位设为“0”。
0：禁止 **pll3_p_ck** 输出
1：使能 **pll3_p_ck** 输出（复位后的默认值）

位 21 **DIVR2EN**：PLL2 DIVR 分频器输出使能 (PLL2 DIVR divider output enable)
由软件置 1 和复位，用于使能 PLL2 的 **pll2_r_ck** 输出。
为实现节能，不使用 **pll3_r_ck** 时，必须将 DIVR3EN 和 DIVR3 位设为“0”。
该位只能在 PLL2 禁止（PLL2ON = “0”且 PLL2RDY = “0”）时写入。
0：禁止 **pll2_r_ck** 输出
1：使能 **pll2_r_ck** 输出（复位后的默认值）

位 20 **DIVQ2EN**：PLL2 DIVQ 分频器输出使能 (PLL2 DIVQ divider output enable)
由软件置 1 和复位，用于使能 PLL2 的 **pll2_q_ck** 输出。
为实现节能，不使用 **pll3_r_ck** 时，必须将 DIVR3EN 和 DIVR3 位设为“0”。
该位只能在 PLL2 禁止（PLL2ON = “0”且 PLL2RDY = “0”）时写入。
0：禁止 **pll2_q_ck** 输出
1：使能 **pll2_q_ck** 输出（复位后的默认值）

位 19 **DIVP2EN**：PLL2 DIVP 分频器输出使能 (PLL2 DIVP divider output enable)
由软件置 1 和复位，用于使能 PLL2 的 **pll2_p_ck** 输出。
该位只能在 PLL2 禁止（PLL2ON = “0”且 PLL2RDY = “0”）时写入。
为实现节能，不使用 **pll3_r_ck** 时，必须将 DIVR3EN 和 DIVR3 位设为“0”。
0：禁止 **pll2_p_ck** 输出
1：使能 **pll2_p_ck** 输出（复位后的默认值）

位 18 **DIVR1EN**：PLL1 DIVR 分频器输出使能 (PLL1 DIVR divider output enable)
由软件置 1 和复位，用于使能 PLL1 的 **pll1_r_ck** 输出。
为实现节能，不使用 **pll3_r_ck** 时，必须将 DIVR3EN 和 DIVR3 位设为“0”。
该位只能在 PLL1 禁止（PLL1ON = “0”且 PLL1RDY = “0”）时写入。
0：禁止 **pll1_r_ck** 输出
1：使能 **pll1_r_ck** 输出（复位后的默认值）

位 17 **DIVQ1EN**：PLL1 DIVQ 分频器输出使能 (PLL1 DIVQ divider output enable)
由软件置 1 和复位，用于使能 PLL1 的 **pll1_q_ck** 输出。
为实现节能，不使用 PLL1 的 **pll1_q_ck** 输出时，必须禁止 **pll1_q_ck**。
该位只能在 PLL1 禁止（PLL1ON = “0”且 PLL1RDY = “0”）时写入。
0：禁止 **pll1_q_ck** 输出
1：使能 **pll1_q_ck** 输出（复位后的默认值）

位 16 **DIVP1EN**：PLL1 DIVP 分频器输出使能 (PLL1 DIVP divider output enable)
由软件置 1 和复位，用于使能 PLL1 的 **pll1_p_ck** 输出。
该位只能在 PLL1 禁止（PLL1ON = “0”且 PLL1RDY = “0”）时写入。
为实现节能，不使用 PLL1 的 **pll1_p_ck** 输出时，必须禁止 **pll1_p_ck**。
0：禁止 **pll1_p_ck** 输出
1：使能 **pll1_p_ck** 输出（复位后的默认值）

位 15:12 保留，必须保持复位值。

位 11:10 **PLL3RGE[1:0]**：PLL3 输入频率范围 (PLL3 input frequency range)
由软件置 1 和复位，用于为 PLL3 选用合适的参考频率范围。
这些位必须在使能 PLL3 之前写入。
00：PLL3 输入 (**ref3_ck**) 时钟频率范围介于 1 MHz 到 2 MHz 之间（复位后的默认值）
01：PLL3 输入 (**ref3_ck**) 时钟频率范围介于 2 MHz 到 4 MHz 之间
10：PLL3 输入 (**ref3_ck**) 时钟频率范围介于 4 MHz 到 8 MHz 之间
11：PLL3 输入 (**ref3_ck**) 时钟频率范围介于 8 MHz 到 16 MHz 之间

位 9 **PLL3VCOSEL**：PLL3 VCO 选择 (PLL3 VCO selection)
由软件置 1 和复位，用于为 PLL3 选用合适的 VCO 频率范围。
该位必须在使能 PLL3 之前写入。
0：宽 VCO 范围，192 MHz 到 836 MHz（复位后的默认值）
1：中等 VCO 范围，150 MHz 到 420 MHz

位 8 **PLL3FRACEN**：PLL3 小数锁存使能 (PLL3 fractional latch enable)
由软件置 1 和复位，用于将 FRACN3 的内容锁存到 Sigma-Delta 调制器中。
为将 FRACN3 值锁存到 Sigma-Delta 调制器中，必须将 PLL3FRACEN 设为“0”，然后将其置“1”：从 0 到 1 的转换会将 FRACN3 内容传送到调制器中。更多相关信息，请参见*PLL 初始化阶段*一节。

位 7:6 **PLL2RGE[1:0]**：PLL2 输入频率范围 (PLL2 input frequency range)

由软件置 1 和复位，用于为 PLL2 选用合适的参考频率范围。

这些位必须在使能 PLL2 之前写入。

00：PLL2 输入 (**ref2_ck**) 时钟频率范围介于 1 MHz 到 2 MHz 之间（复位后的默认值）

01：PLL2 输入 (**ref2_ck**) 时钟频率范围介于 2 MHz 到 4 MHz 之间

10：PLL2 输入 (**ref2_ck**) 时钟频率范围介于 4 MHz 到 8 MHz 之间

11：PLL2 输入 (**ref2_ck**) 时钟频率范围介于 8 MHz 到 16 MHz 之间

位 5 **PLL2VCOSEL**：PLL2 VCO 选择 (PLL2 VCO selection)

由软件置 1 和复位，用于为 PLL2 选用合适的 VCO 频率范围。

该位必须在使能 PLL2 之前写入。

0：宽 VCO 范围，192 MHz 到 836 MHz（复位后的默认值）

1：中等 VCO 范围，150 MHz 到 420 MHz

位 4 **PLL2FRACEN**：PLL2 小数锁存使能 (PLL2 fractional latch enable)

由软件置 1 和复位，用于将 FRACN2 的内容锁存到 Sigma-Delta 调制器中。

为将 FRACN2 值锁存到 Sigma-Delta 调制器中，必须将 PLL2FRACEN 设为“0”，然后将其置“1”：从 0 到 1 的转换会将 FRACN2 内容传送到调制器中。更多相关信息，请参见*PLL 初始化阶段*一节。

位 3:2 **PLL1RGE[1:0]**：PLL1 输入频率范围 (PLL1 input frequency range)

由软件置 1 和复位，用于为 PLL1 选用合适的参考频率范围。

该位必须在使能 PLL1 之前写入。

00：PLL1 输入 (**ref1_ck**) 时钟频率范围介于 1 MHz 到 2 MHz 之间（复位后的默认值）

01：PLL1 输入 (**ref1_ck**) 时钟频率范围介于 2 MHz 到 4 MHz 之间

10：PLL1 输入 (**ref1_ck**) 时钟频率范围介于 4 MHz 到 8 MHz 之间

11：PLL1 输入 (**ref1_ck**) 时钟频率范围介于 8 MHz 到 16 MHz 之间

位 1 **PLL1VCOSEL**：PLL1 VCO 选择 (PLL1 VCO selection)

由软件置 1 和复位，用于为 PLL1 选用合适的 VCO 频率范围。

这些位必须在使能 PLL1 之前写入。

0：宽 VCO 范围，192 MHz 到 836 MHz（复位后的默认值）

1：中等 VCO 范围，150 MHz 到 420 MHz

位 0 **PLL1FRACEN**：PLL1 小数锁存使能 (PLL1 fractional latch enable)

由软件置 1 和复位，用于将 FRACN1 的内容锁存到 Sigma-Delta 调制器中。

为将 FRACN1 值锁存到 Sigma-Delta 调制器中，必须将 PLL1FRACEN 设为“0”，然后将其置“1”：从 0 到 1 的转换会将 FRACN1 内容传送到调制器中。更多相关信息，请参见*PLL 初始化阶段*一节。

8.7.11 RCC PLL1 分频器配置寄存器 (RCC_PLL1DIVR)

RCC PLL1 Dividers Configuration Register

偏移地址：0x030

复位值：0x0101 0280

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	DIVR1[6:0]							Res.	DIVQ1[6:0]						
	rw								rw						

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DIVP1[6:0]								DIVN1[8:0]							
rw								rw							

位 31 保留，必须保持复位值。

位 30:24 **DIVR1[6:0]**：PLL1 DIVR 分频系数 (PLL1 DIVR division factor)

由软件置 1 和复位，用于控制 **pll1_r_ck** 时钟的频率。

这些位只能在 PLL1 禁止（PLL1ON = “0”且 PLL1RDY = “0”）时写入。

0000000：**pll1_r_ck = vco1_ck**

0000001：**pll1_r_ck = vco1_ck** / 2（复位后的默认值）

0000010：**pll1_r_ck = vco1_ck** / 3

0000011：**pll1_r_ck = vco1_ck** / 4

...

1111111：**pll1_r_ck = vco1_ck** / 128

位 23 保留，必须保持复位值。

位 22:16 **DIVQ1[6:0]**：PLL1 DIVQ 分频系数 (PLL1 DIVQ division factor)

由软件置 1 和复位，用于控制 **pll1_q_ck** 时钟的频率。

这些位只能在 PLL1 禁止（PLL1ON = “0”且 PLL1RDY = “0”）时写入。

0000000：**pll1_q_ck = vco1_ck**

0000001：**pll1_q_ck = vco1_ck** / 2（复位后的默认值）

0000010：**pll1_q_ck = vco1_ck** / 3

0000011：**pll1_q_ck = vco1_ck** / 4

...

1111111：**pll1_q_ck = vco1_ck** / 128

位 15:9 **DIVP1[6:0]**：PLL1 DIVP 分频系数 (PLL1 DIVP division factor)

由软件置 1 和复位，用于控制 **pll1_p_ck** 时钟的频率。

这些位只能在 PLL1 禁止（PLL1ON = “0”且 PLL1RDY = “0”）时写入。

请注意，不允许使用奇数分频系数。

0000000：不允许

0000001：**pll1_p_ck** = **vco1_ck** / 2（复位后的默认值）

0000010：不允许

0000011：**pll1_p_ck** = **vco1_ck** / 4

...

1111111：**pll1_p_ck** = **vco1_ck** / 128

位 8:0 **DIVN1[8:0]**：PLL1 VCO 的倍频系数 (Multiplication factor for PLL1 VCO)

由软件置 1 和复位，用于控制 VCO 的倍频系数。

这些位只能在 PLL 禁止（PLL1ON = “0”且 PLL1RDY = “0”）时写入。

0x003：DIVN1 = 4

0x004：DIVN1 = 5

0x005：DIVN1 = 6

...

0x080：DIVN1 = 129（复位后的默认值）

...

0x1FF：DIVN1 = 512

其他：错误配置

注意： 软件必须正确设置这些位，以确保 VCO 输出频率介于其有效频率范围内，即：

– 192 MHz 到 836 MHz（PLL1VCOSEL = “0”时）

– 150 MHz 到 420 MHz（PLL1VCOSEL = “1”时）

当小数值 0 已载入 FRACN1 中时，VCO 输出频率 = F_{ref1_ck} x DIVN1，其中：

– DIVN1 介于 4 到 512 之间

– 输入频率 F_{ref1_ck} 介于 1 MHz 到 16 MHz 之间

8.7.12 RCC PLL1 小数分频器寄存器 (RCC_PLL1FRACR)

RCC PLL1 Fractional Divider Register

偏移地址：0x034

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FRACN1[12:0]													Res.	Res.	Res.
rw															

位 31:16 保留，必须保持复位值。

位 15:3 **FRACN1[12:0]**：PLL1 VCO 的倍频系数的小数部分 (Fractional part of the multiplication factor for PLL1 VCO)

由软件置 1 和复位，用于控制 VCO 的倍频系数的小数部分。
这些位可随时写入，进而对 PLL1 VCO 进行动态微调。

注意： 软件必须正确设置这些位，以确保 VCO 输出频率介于其有效频率范围内，即：

- 192 MHz 到 836 MHz（PLL1VCOSEL = “0”时）
- 150 MHz 到 420 MHz（PLL1VCOSEL = “1”时）

VCO 输出频率 = F_{ref1_ck} x (DIVN1 + (FRACN1 / 2^{13}))，其中

- DIVN1 应介于 4 到 512 之间
- FRACN1 可介于 0 到 2^{13}- 1 之间
- 输入频率 F_{ref1_ck} 应介于 1 MHz 到 16 MHz 之间。

如果执意要在已使能 PLL 的情况下实时更改 FRACN 值，应用程序必须按如下过程进行操作：

- 将位 PLL1FRACEN 设为“0”，
- 将新的小数值写入 FRACN1，
- 将位 PLL1FRACEN 置“1”。

位 2:0 保留，必须保持复位值。

8.7.13 RCC PLL2 分频器配置寄存器 (RCC_PLL2DIVR)

RCC PLL2 Dividers Configuration Register

偏移地址：0x038

复位值：0x0101 0280

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	DIVR2[6:0]							Res.	DIVQ2[6:0]						
	rw								rw						

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DIVP2[6:0]								DIVN2[8:0]							
rw								rw							

位 31 保留，必须保持复位值。

位 30:24 **DIVR2[6:0]**：PLL2 DIVR 分频系数 (PLL2 DIVR division factor)

由软件置 1 和复位，用于控制 **pll2_r_ck** 时钟的频率。

这些位只能在 PLL2 禁止（PLL2ON = “0”且 PLL2RDY = “0”）时写入。

0000000：**pll2_r_ck** = **vco2_ck**

0000001：**pll2_r_ck** = **vco2_ck** / 2（复位后的默认值）

0000010：**pll2_r_ck** = **vco2_ck** / 3

0000011：**pll2_r_ck** = **vco2_ck** / 4

...

1111111：**pll2_r_ck** = **vco2_ck** / 128

位 23 保留，必须保持复位值。

位 22:16 **DIVQ2[6:0]**：PLL2 DIVQ 分频系数 (PLL2 DIVQ division factor)

由软件置 1 和复位，用于控制 **pll2_q_ck** 时钟的频率。

这些位只能在 PLL2 禁止（PLL2ON = “0”且 PLL2RDY = “0”）时写入。

0000000：**pll2_q_ck** = **vco2_ck**

0000001：**pll2_q_ck** = **vco2_ck** / 2（复位后的默认值）

0000010：**pll2_q_ck** = **vco2_ck** / 3

0000011：**pll2_q_ck** = **vco2_ck** / 4

...

1111111：**pll2_q_ck** = **vco2_ck** / 128

位 15:9 **DIVP2[6:0]**：PLL2 DIVP 分频系数 (PLL2 DIVP division factor)

由软件置 1 和复位，用于控制 **pll2_p_ck** 时钟的频率。

这些位只能在 PLL2 禁止（PLL2ON =“0”且 PLL2RDY =“0”）时写入。

0000000：**pll2_p_ck** = **vco2_ck**
0000001：**pll2_p_ck** = **vco2_ck** / 2（复位后的默认值）
0000010：**pll2_p_ck** = **vco2_ck** / 3
0000011：**pll2_p_ck** = **vco2_ck** / 4
...
1111111：**pll2_p_ck** = **vco2_ck** / 128

位 8:0 **DIVN2[8:0]**：PLL2 VCO 的倍频系数 (Multiplication factor for PLL2 VCO)

由软件置 1 和复位，用于控制 VCO 的倍频系数。

这些位只能在 PLL 禁止（PLL2ON =“0”且 PLL2RDY =“0”）时写入。

注意： 软件必须正确设置这些位，以确保 VCO 输出频率介于其有效频率范围内，即：

- 192 MHz 到 836 MHz（PLL2VCOSEL =“0”时）
- 150 MHz 到 420 MHz（PLL2VCOSEL =“1”时）

当小数值 0 已载入 FRACN2 中时，VCO 输出频率 = F_{ref2_ck} x DIVN2，其中：

- DIVN2 介于 4 到 512 之间
- 输入频率 F_{ref2_ck} 介于 1 MHz 到 16 MHz 之间

0x003：DIVN2 = 4
0x004：DIVN2 = 5
0x005：DIVN2 = 6
...
0x080：DIVN2 = 129（复位后的默认值）
...
0x1FF：DIVN2 = 512
其他：错误配置

8.7.14 RCC PLL2 小数分频器寄存器 (RCC_PLL2FRACR)

RCC PLL2 Fractional Divider Register

偏移地址：0x03C

复位值：0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FRACN2[12:0]													Res.	Res.	Res.
rw															

位 31:16 保留，必须保持复位值。

位 15:3 **FRACN2[12:0]**：PLL2 VCO 的倍频系数的小数部分 (Fractional part of the multiplication factor for PLL2 VCO)

由软件置 1 和复位，用于控制 VCO 的倍频系数的小数部分。
这些位可随时写入，进而对 PLL2 VCO 进行动态微调。

注意： 软件必须正确设置这些位，以确保 VCO 输出频率介于其有效频率范围内，即：

- 192 MHz 到 836 MHz（PLL2VCOSEL = “0”时）
- 150 MHz 到 420 MHz（PLL2VCOSEL = “1”时）

VCO 输出频率 = F_{ref2_ck} x $\left(DIVN2 + (FRACN2 / 2^{13})\right)$，其中

- DIVN2 应介于 4 到 512 之间
- FRACN2 可介于 0 到 2^{13} - 1 之间
- 输入频率 F_{ref2_ck} 应介于 1 MHz 到 16 MHz 之间

如果执意要在已使能 PLL 的情况下实时更改 FRACN 值，应用程序必须按如下过程进行操作：

- 将位 PLL2FRACEN 设为“0”，
- 将新的小数值写入 FRACN2，
- 将位 PLL2FRACEN 置“1”。

位 2:0 保留，必须保持复位值。

8.7.15 RCC PLL3 分频器配置寄存器 (RCC_PLL3DIVR)

RCC PLL3 Dividers Configuration Register

偏移地址：0x040

复位值：0x0101 0280

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	DIVR3[6:0]							Res.	DIVQ3[6:0]						
	rw								rw						

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DIVP3[6:0]								DIVN3[8:0]							
rw								rw							

位 31 保留，必须保持复位值。

位 30:24 **DIVR3[6:0]**：PLL3 DIVR 分频系数 (PLL3 DIVR division factor)

由软件置 1 和复位，用于控制 **pll3_r_ck** 时钟的频率。

这些位只能在 PLL3 禁止（PLL3ON = “0”且 PLL3RDY = “0”）时写入。

0000000：**pll3_r_ck** = **vco3_ck**

0000001：**pll3_r_ck** = **vco3_ck** / 2（复位后的默认值）

0000010：**pll3_r_ck** = **vco3_ck** / 3

0000011：**pll3_r_ck** = **vco3_ck** / 4

...

1111111：**pll3_r_ck** = **vco3_ck** / 128

位 23 保留，必须保持复位值。

位 22:16 **DIVQ3[6:0]**：PLL3 DIVQ 分频系数 (PLL3 DIVQ division factor)

由软件置 1 和复位，用于控制 **pll3_q_ck** 时钟的频率。

这些位只能在 PLL3 禁止（PLL3ON = “0”且 PLL3RDY = “0”）时写入。

0000000：**pll3_q_ck** = **vco3_ck**

0000001：**pll3_q_ck** = **vco3_ck** / 2（复位后的默认值）

0000010：**pll3_q_ck** = **vco3_ck** / 3

0000011：**pll3_q_ck** = **vco3_ck** / 4

...

1111111：**pll3_q_ck** = **vco3_ck** / 128

位 15:9 **DIVP3[6:0]**：PLL3 DIVP 分频系数 (PLL3 DIVP division factor)

由软件置 1 和复位，用于控制 **pll3_p_ck** 时钟的频率。

这些位只能在 PLL3 禁止（PLL3ON = “0”且 PLL3RDY = “0”）时写入。

0000000：**pll3_p_ck** = **vco3_ck**

0000001：**pll3_p_ck** = **vco3_ck** / 2（复位后的默认值）

0000010：**pll3_p_ck** = **vco3_ck** / 3

0000011：**pll3_p_ck** = **vco3_ck** / 4

...

1111111：pll3_p_ck = **vco3_ck** / 128

位 8:0 **DIVN3[7:0]**：PLL3 VCO 的倍频系数 (Multiplication factor for PLL3 VCO)

由软件置 1 和复位，用于控制 VCO 的倍频系数。

这些位只能在 PLL 禁止（PLL3ON = “0”且 PLL3RDY = “0”）时写入。

注意： 软件必须正确设置这些位，以确保 VCO 输出频率介于其有效频率范围内，即：

- 192 MHz 到 836 MHz（PLL3VCOSEL = “0”时）
- 150 MHz 到 420 MHz（PLL3VCOSEL = “1”时）

当小数值 0 已载入 FRACN3 中时，VCO 输出频率 = F_{ref3_ck} x DIVN3，其中：

- DIVN3 介于 4 到 512 之间
- 输入频率 F_{ref3_ck} 介于 1 MHz 到 16 MHz 之间

0x003：DIVN3 = 4

0x004：DIVN3 = 5

0x005：DIVN3 = 6

...

0x080：DIVN3 = 129（复位后的默认值）

...

0x1FF：DIVN3 = 512

其他：错误配置

8.7.16 RCC PLL3 小数分频器寄存器 (RCC_PLL3FRACR)

RCC PLL3 Fractional Divider Register

偏移地址：0x044

复位值：0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FRACN3[12:0]													Res.	Res.	Res.
rw															

位 31:16 保留，必须保持复位值。

位 15:3 **FRACN3[12:0]**：PLL3 VCO 的倍频系数的小数部分 (Fractional part of the multiplication factor for PLL3 VCO)

由软件置 1 和复位，用于控制 VCO 的倍频系数的小数部分。
这些位可随时写入，进而对 PLL3 VCO 进行动态微调。

注意： 软件必须正确设置这些位，以确保 VCO 输出频率介于其有效频率范围内，即：

- 192 MHz 到 836 MHz（PLL3VCOSEL = “0”时）
- 150 MHz 到 420 MHz（PLL3VCOSEL = “1”时）

VCO 输出频率 = F_{ref3_ck} x (DIVN3 + (FRACN3 / 2^{13}))，其中

- DIVN3 应介于 4 到 512 之间
- FRACN3 可介于 0 到 2^{13} - 1 之间
- 输入频率 F_{ref3_ck} 应介于 1 MHz 到 16 MHz 之间

如果执意要在已使能 PLL 的情况下实时更改 FRACN 值，应用程序必须按如下过程进行操作：

- 将位 PLL1FRACEN 设为“0”，
- 将新的小数值写入 FRACN1，
- 将位 PLL1FRACEN 置“1”。

位 2:0 保留，必须保持复位值。

8.7.17 RCC 域 1 内核时钟配置寄存器 (RCC_D1CCIPR)

RCC Domain 1 Kernel Clock Configuration Register

偏移地址：0x04C

复位值：0x0000 0000

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

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

1. 允许实时更改时钟源，且不会违反时序。不过用户必须确保在切换期间以及在整个转换时间内，之前的时钟源和新时钟源均存在。请参见*时钟开关和门控*一节。

位 31:30 保留，必须保持复位值。

位 29:28 **CKPERSEL[1:0]**：**per_ck** 时钟源选择 (per_ck clock source selection)
00：将 **hsi_ker_ck** 时钟选作 **per_ck** 时钟（复位后的默认值）
01：将 **csi_ker_ck** 时钟选作 **per_ck** 时钟
10：将 **hse_ck** 时钟选作 **per_ck** 时钟
11：保留，禁止 **per_ck** 时钟

位 27:17 保留，必须保持复位值。

位 16 **SDMMCSEL**：SDMMC 内核时钟源选择 (SDMMC kernel clock source selection)
0：将 **pll1_q_ck** 时钟选作内核外设时钟（复位后的默认值）
1：将 **pll2_r_ck** 时钟选作内核外设时钟

位 15:6 保留，必须保持复位值。

位 5:4 **QSPISEL[1:0]**：QUADSPI 内核时钟源选择 (QUADSPI kernel clock source selection)
00：将 **rcc_hclk3** 时钟选作内核外设时钟（复位后的默认值）
01：将 **pll1_q_ck** 时钟选作内核外设时钟
10：将 **pll2_r_ck** 时钟选作内核外设时钟
11：将 **per_ck** 时钟选作内核外设时钟

位 3:2 保留，必须保持复位值。

位 1:0 **FMCSEL[1:0]**：FMC 内核时钟源选择 (FMC kernel clock source selection)
00：将 **rcc_hclk3** 时钟选作内核外设时钟（复位后的默认值）
01：将 **pll1_q_ck** 时钟选作内核外设时钟
10：将 **pll2_r_ck** 时钟选作内核外设时钟
11：将 **per_ck** 时钟选作内核外设时钟

8.7.18 RCC 域 2 内核时钟配置寄存器 (RCC_D2CCIP1R)

RCC Domain 2 Kernel Clock Configuration Register

偏移地址：0x050

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SWPSEL (1)	Res.	FDCANSEL[1:0] (1)		Res.	Res.	Res.	DFSDM1SEL (1)	Res.	Res.	SPDIFSEL[1:0] (1)		Res.	SPI45SEL[2:0] (1)		
rw		rw					rw			rw			rw		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	SPI123SEL[2:0] (1)			Res.	Res.	Res.	SAI23SEL[2:0] (1)			Res.	Res.	Res.	SAI1SEL[2:0] (1)		
	rw						rw						rw		

1. 允许实时更改时钟源，且不会违反时序。不过用户必须确保在切换期间以及在整个转换时间内，之前的时钟源和新时钟源均存在。请参见*时钟开关和门控*一节。

位 31 **SWPSEL**：SWPMI 内核时钟源选择 (SWPMI kernel clock source selection)
由软件置 1 和复位。
0：将 **pclk** 选作 SWPMI 内核时钟（复位后的默认值）
1：将 **hsi_ker_ck** 时钟选作 SWPMI 内核时钟

位 30 保留，必须保持复位值。

位 29:28 **FDCANSEL**：FDCAN 内核时钟源选择 (FDCAN kernel clock source selection)
由软件置 1 和复位。
00：将 **hse_ck** 时钟选作 FDCAN 内核时钟（复位后的默认值）
01：将 **pll1_q_ck** 时钟选作 FDCAN 内核时钟
10：将 **pll2_q_ck** 时钟选作 FDCAN 内核时钟
11：保留，禁止内核时钟

位 27:25 保留，必须保持复位值。

位 24 **DFSDM1SEL**：DFSDM1 内核 **Clk** 时钟源选择 (DFSDM1 kernel Clk clock source selection)
由软件置 1 和复位。
注： 由 SAI1SEL 进行 DFSDM1 Aclk 时钟源选择。
0：将 **rcc_pclk2** 选作 DFSDM1 Clk 内核时钟（复位后的默认值）
1：将 **sys_ck** 时钟选作 DFSDM1 Clk 内核时钟

位 23:22 保留，必须保持复位值。

位 21:20 **SPDIFSEL[1:0]**：SPDIFRX 内核时钟源选择 (SPDIFRX kernel clock source selection)
00：将 **pll1_q_ck** 时钟选作 SPDIFRX 内核时钟（复位后的默认值）
01：将 **pll2_r_ck** 时钟选作 SPDIFRX 内核时钟
10：将 **pll3_r_ck** 时钟选作 SPDIFRX 内核时钟
11：将 **hsi_ker_ck** 时钟选作 SPDIFRX 内核时钟

位 19 保留，必须保持复位值。

位 18:16 **SPI45SEL[2:0]**：SPI4 和 5 内核时钟源选择 (SPI4 and 5 kernel clock source selection)

由软件置 1 和复位。

000：将 APB 时钟选作内核时钟（复位后的默认值）

001：将 **pll2_q_ck** 时钟选作内核时钟

010：将 **pll3_q_ck** 时钟选作内核时钟

011：将 **hsi_ker_ck** 时钟选作内核时钟

100：将 **csi_ker_ck** 时钟选作内核时钟

101：将 **hse_ck** 时钟选作内核时钟

其他：保留，禁止内核时钟

位 15 保留，必须保持复位值。

位 14:12 **SPI123SEL[2:0]**：SPI/I2S1、2 和 3 内核时钟源选择 (SPI/I2S1,2 and 3 kernel clock source selection)

由软件置 1 和复位。

注意： 如果所选时钟为外部时钟，且该时钟已停止，则无法切换到其他时钟。有关更多信息，请参见*时钟开关和门控*一节。

000：将 **pll1_q_ck** 时钟选作 SPI/I2S1、2 和 3 内核时钟（复位后的默认值）

001：将 **pll2_p_ck** 时钟选作 SPI/I2S1、2 和 3 内核时钟

010：将 **pll3_p_ck** 时钟选作 SPI/I2S1、2 和 3 内核时钟

011：将 **I2S_CKIN** 时钟选作 SPI/I2S1、2 和 3 内核时钟

100：将 **per_ck** 时钟选作 SPI/I2S1、2 和 3 内核时钟

其他：保留，禁止内核时钟

注： I2S_CKIN 为来自某个引脚的外部时钟。

位 11:9 保留，必须保持复位值。

位 8:6 **SAI23SEL[2:0]**：SAI2 和 SAI3 内核时钟源选择 (SAI2 and SAI3 kernel clock source selection)

由软件置 1 和复位。

注意： 如果所选时钟为外部时钟，且该时钟已停止，则无法切换到其他时钟。有关更多信息，请参见*时钟开关和门控*一节。

000：将 **pll1_q_ck** 时钟选作 SAI2 和 SAI3 内核时钟（复位后的默认值）

001：将 **pll2_p_ck** 时钟选作 SAI2 和 SAI3 内核时钟

010：将 **pll3_p_ck** 时钟选作 SAI2 和 SAI3 内核时钟

011：将 I2S_CKIN 时钟选作 SAI2 和 SAI3 内核时钟

100：将 **per_ck** 时钟选作 SAI2 和 SAI3 内核时钟

其他：保留，禁止内核时钟

注： I2S_CKIN 为来自某个引脚的外部时钟。

位 5:3 保留，必须保持复位值。

位 2:0 **SAI1SEL[2:0]**：SAI1 和 DFSDM1 内核 **Aclk** 时钟源选择 (SAI1 and DFSDM1 kernel Aclk clock source selection)

由软件置 1 和复位。

注意： 如果所选时钟为外部时钟，且该时钟已停止，则无法切换到其他时钟。有关更多信息，请参见*时钟开关和门控*一节。

注： 由 DFSDM1SEL 进行 DFSDM1 时钟源选择。

000：将 **pll1_q_ck** 时钟选作 SAI1 和 DFSDM1 **Aclk** 内核时钟（复位后的默认值）

001：将 **pll2_p_ck** 时钟选作 SAI1 和 DFSDM1 **Aclk** 内核时钟

010：将 **pll3_p_ck** 时钟选作 SAI1 和 DFSDM1 **Aclk** 内核时钟

011：将 I2S_CKIN 时钟选作 SAI1 和 DFSDM1 **Aclk** 内核时钟

100：将 **per_ck** 时钟选作 SAI1 和 DFSDM1 **Aclk** 内核时钟

其他：保留，禁止内核时钟

注： I2S_CKIN 为来自某个引脚的外部时钟。

8.7.19 RCC 域 2 内核时钟配置寄存器 (RCC_D2CCIP2R)

RCC Domain 2 Kernel Clock Configuration Register

偏移地址：0x054

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	LPTIM1SEL[2:0] (1)			Res.	Res.	Res.	Res.	CECSEL[1:0] (1)		USBSEL[1:0] (1)		Res.	Res.	Res.	Res.
	rw							rw		rw					

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	I2C123SEL[1:0] (1)		Res.	Res.	RNGSEL[1:0] (1)		Res.	Res.	USART16SEL[2:0] (1)			USART234578SEL[2:0] (1)		
		rw				rw				rw			rw		

1. 允许实时更改时钟源，且不会违反时序。不过用户必须确保在切换期间以及在整个转换时间内，之前的时钟源和新时钟源均存在。请参见*时钟开关和门控*一节。

位 31 保留，必须保持复位值。

位 30:28 **LPTIM1SEL[2:0]**：LPTIM1 内核时钟源选择 (LPTIM1 kernel clock source selection)
由软件置 1 和复位。
000：将 **rcc_pclk1** 时钟选作内核外设时钟（复位后的默认值）
001：将 **pll2_p_ck** 时钟选作内核外设时钟
010：将 **pll3_r_ck** 时钟选作内核外设时钟
011：将 **lse_ck** 时钟选作内核外设时钟
100：将 **lsi_ck** 时钟选作内核外设时钟
101：将 **per_ck** 时钟选作内核外设时钟
其他：保留，禁止内核时钟

位 27:24 保留，必须保持复位值。

位 23:22 **CECSEL[1:0]**：HDMI-CEC 内核时钟源选择 (HDMI-CEC kernel clock source selection)
由软件置 1 和复位。
00：将 **lse_ck** 时钟选作内核时钟（复位后的默认值）
01：将 **lsi_ck** 时钟选作内核时钟
10：将 122 分频的 **csi_ker_ck** 选作内核时钟
11：保留，禁止内核时钟

位 21:20 **USBSEL[1:0]**：USBOTG 1 和 2 内核时钟源选择 (USBOTG 1 and 2 kernel clock source selection)
由软件置 1 和复位。
00：禁止内核时钟（复位后的默认值）
01：将 **pll1_q_ck** 时钟选作内核时钟
10：将 **pll3_q_ck** 时钟选作内核时钟
11：将 **hsi48_ck** 时钟选作内核时钟

位 19:14 保留，必须保持复位值。

位 13:12 **I2C123SEL[1:0]**：I2C1,2,3 内核时钟源选择 (I2C1,2,3 kernel clock source selection)
由软件置 1 和复位。
00：将 **rcc_pclk1** 时钟选作内核时钟（复位后的默认值）
01：将 **pll3_r_ck** 时钟选作内核时钟
10：将 **hsi_ker_ck** 时钟选作内核时钟
11：将 **csi_ker_ck** 时钟选作内核时钟

位 11:10 保留，必须保持复位值。

位 9:8 **RNGSEL[1:0]**：RNG 内核时钟源选择 (RNG kernel clock source selection)
由软件置 1 和复位。
00：将 **hsi48_ck** 时钟选作内核时钟（复位后的默认值）
01：将 **pll1_q_ck** 时钟选作内核时钟
10：将 **lse_ck** 时钟选作内核时钟
11：将 **lsi_ck** 时钟选作内核时钟

位 7:6 保留，必须保持复位值。

位 5:3 **USART16SEL[2:0]**：USART1 和 6 内核时钟源选择 (USART1 and 6 kernel clock source selection)
由软件置 1 和复位。
000：将 **rcc_pclk2** 时钟选作内核时钟（复位后的默认值）
001：将 **pll2_q_ck** 时钟选作内核时钟
010：将 **pll3_q_ck** 时钟选作内核时钟
011：将 **hsi_ker_ck** 时钟选作内核时钟
100：将 **csi_ker_ck** 时钟选作内核时钟
101：将 **lse_ck** 时钟选作内核时钟
其他：保留，禁止内核时钟

位 2:0 **USART234578SEL[2:0]**：USART2/3、UART4,5、7/8 (APB1) 内核时钟源选择 (USART2/3, UART4,5, 7/8 (APB1) kernel clock source selection)
由软件置 1 和复位。
000：将 **rcc_pclk1** 时钟选作内核时钟（复位后的默认值）
001：将 **pll2_q_ck** 时钟选作内核时钟
010：将 **pll3_q_ck** 时钟选作内核时钟
011：将 **hsi_ker_ck** 时钟选作内核时钟
100：将 **csi_ker_ck** 时钟选作内核时钟
101：将 **lse_ck** 时钟选作内核时钟
其他：保留，禁止内核时钟

8.7.20 RCC 域 3 内核时钟配置寄存器 (RCC_D3CCIPR)

RCC Domain 3 Kernel Clock Configuration Register

偏移地址：0x058

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	SPI6SEL[2:0] (1)			Res.	SAI4BSEL[2:0] (1)			SAI4ASEL[2:0] (1)			Res.	Res.	Res.	ADCSEL[1:0] (1)	
	rw				rw			rw						rw	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LPTIM345SEL[2:0] (1)			LPTIM2SEL[2:0] (1)			I2C4SEL[1:0] (1)		Res.	Res.	Res.	Res.	Res.	LPUART1SEL[2:0] (1)		
rw			rw			rw							rw		

1. 允许实时更改时钟源，且不会违反时序。不过用户必须确保在切换期间以及在整个转换时间内，之前的时钟源和新时钟源均存在。请参见*时钟开关和门控*一节。

位 31 保留，必须保持复位值。

位 30:28 **SPI6SEL[2:0]**：SPI6 内核时钟源选择 (SPI6 kernel clock source selection)

由软件置 1 和复位。

000：将 **rcc_pclk4** 时钟选作内核外设时钟（复位后的默认值）

001：将 **pll2_q_ck** 时钟选作内核外设时钟

010：将 **pll3_q_ck** 时钟选作内核外设时钟

011：将 **hsi_ker_ck** 时钟选作内核外设时钟

100：将 **csi_ker_ck** 时钟选作内核外设时钟

101：将 **hse_ck** 时钟选作内核外设时钟

其他：保留，禁止内核时钟

位 27 保留，必须保持复位值。

位 26:24 **SAI4BSEL[2:0]**：SAI4 子模块 B 内核时钟源选择 (Sub-Block B of SAI4 kernel clock source selection)

由软件置 1 和复位。

注意： 如果所选时钟为外部时钟，且该时钟已停止，则无法切换到其他时钟。有关更多信息，请参见*时钟开关和门控*一节。

000：将 **pll1_q_ck** 时钟选作内核外设时钟（复位后的默认值）

001：将 **pll2_p_ck** 时钟选作内核外设时钟

010：将 **pll3_p_ck** 时钟选作内核外设时钟

011：将 I2S_CKIN 时钟选作内核外设时钟

100：将 **per_ck** 时钟选作内核外设时钟

其他：保留，禁止内核时钟

注： I2S_CKIN 为来自某个引脚的外部时钟。

位 23:21 **SAI4ASEL[2:0]**：SAI4 子模块 A 内核时钟源选择 (Sub-Block A of SAI4 kernel clock source selection)
由软件置 1 和复位。

注意： 如果所选时钟为外部时钟，且该时钟已停止，则无法切换到其他时钟。有关更多信息，请参见*时钟开关和门控*一节。

000：将 **pll1_q_ck** 时钟选作内核外设时钟（复位后的默认值）
001：将 **pll2_p_ck** 时钟选作内核外设时钟
010：将 **pll3_p_ck** 时钟选作内核外设时钟
011：将 I2S_CKIN 时钟选作内核外设时钟
100：将 **per_ck** 时钟选作内核外设时钟
其他：保留，禁止内核时钟

注： I2S_CKIN 为来自某个引脚的外部时钟。

位 20:18 保留，必须保持复位值。

位 17:16 **ADCSEL[1:0]**：SAR ADC 内核时钟源选择 (SAR ADC kernel clock source selection)
由软件置 1 和复位。
00：将 **pll2_p_ck** 时钟选作内核外设时钟（复位后的默认值）
01：将 **pll3_r_ck** 时钟选作内核外设时钟
10：将 **per_ck** 时钟选作内核外设时钟
其他：保留，禁止内核时钟

位 15:13 **LPTIM345SEL[2:0]**：LPTIM3,4,5 内核时钟源选择 (LPTIM3,4,5 kernel clock source selection)
由软件置 1 和复位。
000：将 **rcc_pclk4** 时钟选作内核外设时钟（复位后的默认值）
001：将 **pll2_p_ck** 时钟选作内核外设时钟
010：将 **pll3_r_ck** 时钟选作内核外设时钟
011：将 **lse_ck** 时钟选作内核外设时钟
100：将 **lsi_ck** 时钟选作内核外设时钟
101：将 **per_ck** 时钟选作内核外设时钟
其他：保留，禁止内核时钟

位 12:10 **LPTIM2SEL[2:0]**：LPTIM2 内核时钟源选择 (LPTIM2 kernel clock source selection)
由软件置 1 和复位。
000：将 **rcc_pclk4** 时钟选作内核外设时钟（复位后的默认值）
001：将 **pll2_p_ck** 时钟选作内核外设时钟
010：将 **pll3_r_ck** 时钟选作内核外设时钟
011：将 **lse_ck** 时钟选作内核外设时钟
100：将 **lsi_ck** 时钟选作内核外设时钟
101：将 **per_ck** 时钟选作内核外设时钟
其他：保留，禁止内核时钟

位 9:8 **I2C4SEL[1:0]**：I2C4 内核时钟源选择 (I2C4 kernel clock source selection)
由软件置 1 和复位。
00：将 **rcc_pclk4** 时钟选作内核外设时钟（复位后的默认值）
01：将 **pll3_r_ck** 时钟选作内核外设时钟
10：将 **hsi_ker_ck** 时钟选作内核外设时钟
11：将 **csi_ker_ck** 时钟选作内核外设时钟

位 7:3 保留，必须保持复位值。

位 2:0 **LPUART1SEL[2:0]**：LPUART1 内核时钟源选择 (LPUART1 kernel clock source selection)
由软件置 1 和复位。
000：将 **rcc_pclk_d3** 时钟选作内核外设时钟（复位后的默认值）
001：将 **pll2_q_ck** 时钟选作内核外设时钟
010：将 **pll3_q_ck** 时钟选作内核外设时钟
011：将 **hsi_ker_ck** 时钟选作内核外设时钟
100：将 **csi_ker_ck** 时钟选作内核外设时钟
101：将 **lse_ck** 时钟选作内核外设时钟
其他：保留，禁止内核时钟

8.7.21 RCC 时钟源中断使能寄存器 (RCC_CIER)

RCC Clock Source Interrupt Enable Register

偏移地址：0x060

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	LSECSSIE	PLL3RDYIE	PLL2RDYIE	PLL1RDYIE	HSI48RDYIE	CSIRDYIE	HSERDYIE	HSIRDYIE	LSERDYIE	LSIRDYIE
						rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:10 保留，必须保持复位值。

位 9 **LSECSSIE**：LSE 时钟安全系统中断使能 (LSE clock security system Interrupt Enable)
由软件置 1 和复位，用于使能/禁止由外部 32 kHz 振荡器上的时钟安全系统引起的中断。
0：禁止 LSE CSS 中断（复位后的默认值）
1：使能 LSE CSS 中断

位 8 **PLL3RDYIE**：PLL3 就绪中断使能 (PLL3 ready Interrupt Enable)
由软件置 1 和复位，用于使能/禁止由 PLL3 锁定引起的中断。
0：禁止 PLL3 锁定中断（复位后的默认值）
1：使能 PLL3 锁定中断

位 7 **PLL2RDYIE**：PLL2 就绪中断使能 (PLL2 ready Interrupt Enable)
由软件置 1 和复位，用于使能/禁止由 PLL2 锁定引起的中断。
0：禁止 PLL2 锁定中断（复位后的默认值）
1：使能 PLL2 锁定中断

位 6 **PLL1RDYIE**：PLL1 就绪中断使能 (PLL1 ready Interrupt Enable)
由软件置 1 和复位，用于使能/禁止由 PLL1 锁定引起的中断。
0：禁止 PLL1 锁定中断（复位后的默认值）
1：使能 PLL1 锁定中断

位 5 **HSI48RDYIE**：HSI48 就绪中断使能 (HSI48 ready Interrupt Enable)
由软件置 1 和复位，用于使能/禁止由 HSI48 振荡器稳定所引起的中断。
0：禁止 HSI48 就绪中断（复位后的默认值）
1：使能 HSI48 就绪中断

位 4 **CSIRDYIE**：CSI 就绪中断使能 (CSI ready Interrupt Enable)
由软件置 1 和复位，用于使能/禁止由 CSI 振荡器稳定所引起的中断。
0：禁止 CSI 就绪中断（复位后的默认值）
1：使能 CSI 就绪中断

位 3 **HSERDYIE**：HSE 就绪中断使能 (HSE ready interrupt enable)
由软件置 1 和复位，用于使能/禁止由 HSE 振荡器稳定所引起的中断。
0：禁止 HSE 就绪中断（复位后的默认值）
1：使能 HSE 就绪中断

位 2 **HSIRDYIE**：HSI 就绪中断使能 (HSI ready interrupt enable)
由软件置 1 和复位，用于使能/禁止由 HSI 振荡器稳定所引起的中断。
0：禁止 HSI 就绪中断（复位后的默认值）
1：使能 HSI 就绪中断

位 1 **LSERDYIE**：LSE 就绪中断使能 (LSE ready interrupt enable)
由软件置 1 和复位，用于使能/禁止由 LSE 振荡器稳定所引起的中断。
0：禁止 LSE 就绪中断（复位后的默认值）
1：使能 LSE 就绪中断

位 0 **LSIRDYIE**：LSI 就绪中断使能 (LSI ready interrupt enable)
由软件置 1 和复位，用于使能/禁止由 LSI 振荡器稳定所引起的中断。
0：禁止 LSI 就绪中断（复位后的默认值）
1：使能 LSI 就绪中断

8.7.22 RCC 时钟源中断标志寄存器 (RCC_CIFR)

RCC Clock Source Interrupt Flag Register

偏移地址：0x64

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	HSECSSF	LSECSSF	PLL3RDYF	PLL2RDYF	PLL1RDYF	HSI48RDYF	CSIRDYF	HSERDYF	HSIRDYF	LSERDYF	LSIRDYF
					r	r	r	r	r	r	r	r	r	r	r

位 31:11 保留，必须保持复位值。

位 10 **HSECSSF**：HSE 时钟安全系统中断标志 (HSE clock security system Interrupt Flag)
通过软件写入 HSECSSC 位进行复位。
如果 HSE 时钟出现故障，则由硬件置 1。
0：未因 HSE 时钟故障而引起时钟安全中断（复位后的默认值）
1：因 HSE 时钟故障而引起时钟安全中断

位 9 **LSECSSF**：LSE 时钟安全系统中断标志 (LSE clock security system Interrupt Flag)
通过软件写入 LSECSSC 位进行复位。
当外部 32 kHz 振荡器中检测到故障且 LSECSSIE 置 1 时由硬件置 1。
0：未在外部 32 kHz 振荡器上检测到故障（复位后的默认值）
1：在外部 32 kHz 振荡器上检测到故障

位 8 **PLL3RDYF**：PLL3 就绪中断标志 (PLL3 ready Interrupt Flag)
通过软件写入 PLL3RDYC 位进行复位。
当 PLL3 锁定并且 PLL3RDYIE 置 1 时由硬件置 1。
0：未因 PLL3 锁定而引起时钟就绪中断（复位后的默认值）
1：因 PLL3 锁定而引起时钟就绪中断

位 7 **PLL2RDYF**：PLL2 就绪中断标志 (PLL2 ready Interrupt Flag)
通过软件写入 PLL2RDYC 位进行复位。
当 PLL2 锁定并且 PLL2RDYIE 置 1 时由硬件置 1。
0：未因 PLL2 锁定而引起时钟就绪中断（复位后的默认值）
1：因 PLL2 锁定而引起时钟就绪中断

位 6 **PLL1RDYF**：PLL1 就绪中断标志 (PLL1 ready Interrupt Flag)
通过软件写入 PLL1RDYC 位进行复位。
当 PLL1 锁定并且 PLL1RDYIE 置 1 时由硬件置 1。
0：未因 PLL1 锁定而引起时钟就绪中断（复位后的默认值）
1：因 PLL1 锁定而引起时钟就绪中断

位 5 **HSI48RDYF**：HSI48 就绪中断标志 (HSI48 ready Interrupt Flag)
通过软件写入 HSI48RDYC 位进行复位。
当 HSI48 时钟稳定且 HSI48RDYIE 置 1 时由硬件置 1。
0：未因 HSI48 振荡器而引起时钟就绪中断（复位后的默认值）
1：因 HSI48 振荡器引起时钟就绪中断

位 4 **CSIRDYF**：CSI 就绪中断标志 (CSI ready Interrupt Flag)
通过软件写入 CSIRDYC 位进行复位。
当 CSI 时钟稳定且 CSIRDYIE 置 1 时由硬件置 1。
0：未因 CSI 而引起时钟就绪中断（复位后的默认值）
1：因 CSI 引起时钟就绪中断

位 3 **HSERDYF**：HSE 就绪中断标志 (HSE ready Interrupt Flag)
通过软件写入 HSERDYC 位进行复位。
当 HSE 时钟稳定且 HSERDYIE 置 1 时由硬件置 1。
0：未因 HSE 而引起时钟就绪中断（复位后的默认值）
1：因 HSE 引起时钟就绪中断

位 2 **HSIRDYF**：HSI 就绪中断标志 (HSI ready Interrupt Flag)
通过软件写入 HSIRDYC 位进行复位。
当 HSI 时钟稳定且 HSIRDYIE 置 1 时由硬件置 1。
0：未因 HSI 而引起时钟就绪中断（复位后的默认值）
1：因 HSI 引起时钟就绪中断

位 1 **LSERDYF**：LSE 就绪中断标志 (LSE ready Interrupt Flag)
通过软件写入 LSERDYC 位进行复位。
当 LSE 时钟稳定且 LSERDYIE 置 1 时由硬件置 1。
0：未因 LSE 而引起时钟就绪中断（复位后的默认值）
1：因 LSE 引起时钟就绪中断

位 0 **LSIRDYF**：LSI 就绪中断标志 (LSI ready Interrupt Flag)
通过软件写入 LSIRDYC 位进行复位。
当 LSI 时钟稳定且 LSIRDYIE 置 1 时由硬件置 1。
0：未因 LSI 而引起时钟就绪中断（复位后的默认值）
1：因 LSI 引起时钟就绪中断

8.7.23 RCC 时钟源中断清零寄存器 (RCC_CICR)

RCC Clock Source Interrupt Clear Register

偏移地址：0x68

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	HSECSSC	LSECSSC	PLL3RDYC	PLL2RDYC	PLL1RDYC	HSI48RDYC	CSIRDYC	HSERDYC	HSIRDYC	LSERDYC	LSIRDYC
					rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1

位 31:11 保留，必须保持复位值。

位 10 **HSECSSC**：HSE 时钟安全系统中断清零 (HSE clock security system Interrupt Clear)
由软件置 1，用于将 HSECSSF 清零。
完成清零后由硬件复位。
0：对 HSECSSF 无影响（复位后的默认值）
1：已将 HSECSSF 清零

位 9 **LSECSSC**：LSE 时钟安全系统中断清零 (LSE clock security system Interrupt Clear)
由软件置 1，用于将 LSECSSF 清零。
完成清零后由硬件复位。
0：对 LSECSSF 无影响（复位后的默认值）
1：已将 LSECSSF 清零

位 8 **PLL3RDYC**：PLL3 就绪中断清零 (PLL3 ready Interrupt Clear)
由软件置 1，用于将 PLL3RDYF 清零。
完成清零后由硬件复位。
0：对 PLL3RDYF 无影响（复位后的默认值）
1：已将 PLL3RDYF 清零

位 7 **PLL2RDYC**：PLL2 就绪中断清零 (PLL2 ready Interrupt Clear)
由软件置 1，用于将 PLL2RDYF 清零。
完成清零后由硬件复位。
0：对 PLL2RDYF 无影响（复位后的默认值）
1：已将 PLL2RDYF 清零

位 6 **PLL1RDYC**：PLL1 就绪中断清零 (PLL1 ready Interrupt Clear)
由软件置 1，用于将 PLL1RDYF 清零。
完成清零后由硬件复位。
0：对 PLL1RDYF 无影响（复位后的默认值）
1：已将 PLL1RDYF 清零

位 5 **HSI48RDYC**：HSI48 就绪中断清零 (HSI48 ready Interrupt Clear)
由软件置 1，用于将 HSI48RDYF 清零。
完成清零后由硬件复位。
0：对 HSI48RDYF 无影响（复位后的默认值）
1：已将 HSI48RDYF 清零

位 4 **CSIRDYC**：CSI 就绪中断清零 (CSI ready Interrupt Clear)
由软件置 1，用于将 CSIRDYF 清零。
完成清零后由硬件复位。
0：对 CSIRDYF 无影响（复位后的默认值）
1：已将 CSIRDYF 清零

位 3 **HSERDYC**：HSE 就绪中断清零 (HSE ready interrupt clear)
由软件置 1，用于将 HSERDYF 清零。
完成清零后由硬件复位。
0：对 HSERDYF 无影响（复位后的默认值）
1：已将 HSERDYF 清零

位 2 **HSIRDYC**：HSI 就绪中断清零 (HSI ready interrupt clear)
由软件置 1，用于将 HSIRDYF 清零。
完成清零后由硬件复位。
0：对 HSIRDYF 无影响（复位后的默认值）
1：已将 HSIRDYF 清零

位 1 **LSERDYC**：LSE 就绪中断清零 (LSE ready interrupt clear)
由软件置 1，用于将 LSERDYF 清零。
完成清零后由硬件复位。
0：对 LSERDYF 无影响（复位后的默认值）
1：已将 LSERDYF 清零

位 0 **LSIRDYC**：LSI 就绪中断清零 (LSI ready interrupt clear)
由软件置 1，用于将 LSIRDYF 清零。
完成清零后由硬件复位。
0：对 LSIRDYF 无影响（复位后的默认值）
1：已将 LSIRDYF 清零

8.7.24 RCC 备份域控制寄存器 (RCC_BDCR)

RCC Backup Domain Control Register

偏移地址：0x070

复位值：0x0000 0000，通过备份域复位进行复位。

访问：0 ≤ 等待周期 ≤ 7，按字、半字和字节访问。连续访问该寄存器时，则插入等待周期。

系统复位后，RCC_BDCR 寄存器受写保护。要修改此寄存器，*PWR 控制寄存器 1 (PWR_CR1)* 中的 DBP 位必须置 “1”。只有备份域复位后， RCC_BDCR 位才能复位 （请参见*第 8.4.6 节：备份域复位*）。其他内部复位或外部复位对这些位不会有任何影响。

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RTCEN	Res.	Res.	Res.	Res.	Res.	RTCSEL[1:0]		Res.	LSECSSD	LSECSSON	LSEDRV[1:0]		LSEBYP	LSERDY	LSEON
rw						rwo			r	rs	rw		rw	r	rw

位 31:17 保留，必须保持复位值。

位 16 **BDRST**：备份域软件复位 (Backup domain software reset)

由软件置 1 和复位。

0：复位未激活（备份域复位后的默认值）

1：复位整个 VSW 域

位 15 **RTCEN**：RTC 时钟使能 (RTC clock enable)

由软件置 1 和复位。

0：禁止 **rtc_ck** 时钟（备份域复位后的默认值）

1：使能 **rtc_ck** 时钟

位 14:10 保留，必须保持复位值。

位 9:8 **RTCSEL[1:0]**：RTC 时钟源选择 (RTC clock source selection)

由软件置 1，用于为 RTC 选择时钟源。这些位只能写入一次（在 LSE 上检测到故障的情况除外）。这些位必须在使能 LSECSSON 之前写入。BDRST 位可用于复位这些位，然后该位可再次写入。

如果将 HSE 选作 RTC 时钟：在系统处于停止模式或在发生引脚复位 (NRST) 时，该时钟将丢失。

00：无时钟（备份域复位后的默认值）

01：将 LSE 时钟用作 RTC 时钟

10：将 LSI 时钟用作 RTC 时钟

11：将以 RTCPRE 值进行分频的 HSE 时钟用作 RTC 时钟

位 7 保留，必须保持复位值。

位 6 **LSECSSD**：LSE 时钟安全系统故障检测 (LSE clock security system failure detection)

由硬件置 1，用于指示外部 32 kHz 振荡器上的时钟安全系统何时检测到故障。

0：未在 32 kHz 振荡器上检测到故障（备份域复位后的默认值）

1：在 32 kHz 振荡器上检测到故障

位 5 **LSECSSON**：LSE 时钟安全系统使能 (LSE clock security system enable)

由软件置 1，用于使能 32 kHz 振荡器上的时钟安全系统。

在 LSE 使能（已使能 LSEON）和就绪（由硬件将 LSERDY 置 1）后以及在选择 RTCSEL 后，LSECSSON 必须使能。

该位一旦使能便不能禁止，但检测到 LSE 故障（LSECSSD =“1”）后除外。此时，软件**必须**禁止 LSECSSON。

0：32 kHz 振荡器上的时钟安全系统关闭（备份域复位后的默认值）

1：32 kHz 振荡器上的时钟安全系统开启

位 4:3 **LSEDRV[1:0]**：LSE 振荡器驱动能力 (LSE oscillator driving capability)

由软件置 1，用于选择 LSE 振荡器的驱动能力。

00：最低驱动（备份域复位后的默认值）

01：中低驱动

10：中高驱动

11：最高驱动

位 2 **LSEBYP**：LSE 振荡器旁路 (LSE oscillator bypass)

由软件置 1 和复位，用于在调试模式下旁路振荡器。在 LSE 使能（通过 LSEON）或就绪（LSERDY =“1”）时，不能写入该位

0：不旁路 LSE 振荡器（备份域复位后的默认值）

1：旁路 LSE 振荡器

位 1 **LSERDY**：LSE 振荡器就绪 (LSE oscillator ready)

由硬件置 1 和复位，用于指示 LSE 已稳定。将 LSEON 设为“0”后，该位需要经过 6 个 **lse_ck** 时钟周期才能变为低电平。

0：LSE 振荡器未就绪（备份域复位后的默认值）

1：LSE 振荡器已就绪

位 0 **LSEON**：LSE 振荡器使能 (LSE oscillator enabled)

由软件置 1 和复位。

0：LSE 振荡器关闭（备份域复位后的默认值）

1：LSE 振荡器开启

8.7.25 RCC 时钟控制和状态寄存器 (RCC_CSR)

RCC Clock Control and Status Register

偏移地址：0x074

复位值：0x0000 0000

访问：0 ≤ 等待周期 ≤ 7，按字、半字和字节访问

连续访问该寄存器时，则插入等待周期。

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

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

位 31:2 保留，必须保持复位值。

位 1 **LSIRDY**：LSI 振荡器就绪 (LSI oscillator ready)

由硬件置 1 和复位，用于指示低速内部 RC 振荡器已稳定。

将 LSION 设为“0”后，该位需要经过 3 个 **lsi_ck** 时钟周期才能变为低电平。

如果存在 LSE 上的时钟安全系统、低速看门狗或 RTC 发出的 LSI 时钟请求，即使 LSION 未使能，该位也可置 1。

0：LSI 时钟未就绪（复位后的默认值）

1：LSI 时钟已就绪

位 0 **LSION**：LSI 振荡器使能 (LSI oscillator enable)

由软件置 1 和复位。

0：LSI 关闭（复位后的默认值）

1：LSI 开启

8.7.26 RCC AHB3 复位寄存器 (RCC_AHB3RSTR)

RCC AHB3 Reset Register

偏移地址：0x07C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CPURST	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SDMMC1RST
rs															rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	QSPIRST	Res.	FMCRST	Res.	Res.	Res.	Res.	Res.	Res.	JPGDECRST	DMA2DRST	Res.	Res.	Res.	MDMARST
	rw		rw							rw	rw				rw

位 31 **CPURST**：CPU 复位 (CPU reset)
由软件置 1，由硬件复位
0：不复位 CPU（复位后的默认值）
1：复位 CPU，位自动零。

位 30:17 保留，必须保持复位值。

位 16 **SDMMC1RST**：SDMMC1 和 SDMMC1 延迟模块复位 (SDMMC1 and SDMMC1 delay block reset)
由软件置 1 和复位。
0：不复位 SDMMC1 和 SDMMC1 延迟模块（复位后的默认值）
1：复位 SDMMC1 和 SDMMC1 延迟模块

位 15 保留，必须保持复位值。

位 14 **QSPIRST**：QUADSPI 和 QUADSPI 延迟模块复位 (QUADSPI and QUADSPI delay block reset)
由软件置 1 和复位。
0：不复位 QUADSPI 和 QUADSPI 延迟模块（复位后的默认值）
1：复位 QUADSPI 和 QUADSPI 延迟模块

位 13 保留，必须保持复位值。

位 12 **FMCRST**：FMC 模块复位 (FMC block reset)
由软件置 1 和复位。
0：不复位 FMC 模块（复位后的默认值）
1：复位 FMC 模块

位 11:6 保留，必须保持复位值。

位 5 **JPGDECRST**：JPGDEC 模块复位 (JPGDEC block reset)
由软件置 1 和复位。
0：不复位 JPGDEC 模块（复位后的默认值）
1：复位 JPGDEC 模块

位 4 **DMA2DRST**：DMA2D 模块复位 (DMA2D block reset)
由软件置 1 和复位。
0：不复位 DMA2D 模块（复位后的默认值）
1：复位 DMA2D 模块

位 3:1 保留，必须保持复位值。

位 0 **MDMARST**：MDMA 模块复位 (MDMA block reset)
由软件置 1 和复位。
0：不复位 MDMA 模块（复位后的默认值）
1：复位 MDMA 模块

8.7.27 RCC AHB1 外设复位寄存器 (RCC_AHB1RSTR)

RCC AHB1 Peripheral Reset Register

偏移地址：0x080

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	USB2OTGRST	Res.	USB1OTGRST	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
				rw		rw									

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ETH1MACRST	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ADC12RST	Res.	Res.	Res.	DMA2RST	DMA1RST
rw										rw				rw	rw

位 31:28 保留，必须保持复位值。

位 27 **USB2OTGRST**：USB2OTG 模块复位 (USB2OTG block reset)
由软件置 1 和复位。
0：不复位 USB2OTG 模块（复位后的默认值）
1：复位 USB2OTG 模块

位 26 保留，必须保持复位值。

位 25 **USB1OTGRST**：USB1OTG 模块复位 (USB1OTG block reset)
由软件置 1 和复位。
0：不复位 USB1OTG 模块（复位后的默认值）
1：复位 USB1OTG 模块

位 24:16 保留，必须保持复位值。

位 15 **ETH1MACRST**：ETH1MAC 模块复位 (ETH1MAC block reset)
由软件置 1 和复位。
0：不复位 ETH1MAC 模块（复位后的默认值）
1：复位 ETH1MAC 模块

位 14:6 保留，必须保持复位值。

位 5 **ADC12RST**：ADC1 和 2 模块复位 (ADC1 and 2 block reset)
由软件置 1 和复位。
0：不复位 ADC1 和 2 模块（复位后的默认值）
1：复位 ADC1 和 2 模块

位 4:2 保留，必须保持复位值。

位 1 **DMA2RST**：DMA2 模块复位 (DMA2 block reset)
由软件置 1 和复位。
0：不复位 DMA2 模块（复位后的默认值）
1：复位 DMA2 模块

位 0 **DMA1RST**：DMA1 模块复位 (DMA1 block reset)
由软件置 1 和复位。
0：不复位 DMA1 模块（复位后的默认值）
1：复位 DMA1 模块

8.7.28 RCC AHB2 外设复位寄存器 (RCC_AHB2RSTR)

RCC AHB2 Peripheral Reset Register

偏移地址：0x084

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	SDMMC2RST	Res.	Res.	RNGRST	HASHRST	CRYPTRST	Res.	Res.	Res.	CAMITFRST
						rw			rw	rw	rw				rw

位 31:10 保留，必须保持复位值。

位 9 **SDMMC2RST**：SDMMC2 和 SDMMC2 延迟模块复位 (SDMMC2 and SDMMC2 Delay block reset)
由软件置 1 和复位。
0：不复位 SDMMC2 和 SDMMC2 延迟模块（复位后的默认值）
1：复位 SDMMC2 和 SDMMC2 延迟模块

位 8:7 保留，必须保持复位值。

位 6 **RNGRST**：随机数发生器模块复位 (Random Number Generator block reset)
由软件置 1 和复位。
0：不复位 RNG 模块（复位后的默认值）
1：复位 RNG 模块

位 5 **HASHRST**：散列模块复位 (Hash block reset)
由软件置 1 和复位。
0：不复位散列模块（复位后的默认值）
1：复位散列模块

位 4 **CRYPTRST**：加密模块复位 (Cryptography block reset)
由软件置 1 和复位。
0：不复位加密模块（复位后的默认值）
1：复位加密模块

位 3:1 保留，必须保持复位值。

位 0 **CAMITFRST**：CAMITF 模块复位 (CAMITF block reset)
由软件置 1 和复位。
0：不复位 CAMITF 模块（复位后的默认值）
1：复位 CAMITF 模块

8.7.29 RCC AHB4 外设复位寄存器 (RCC_AHB4RSTR)

RCC AHB4 Peripheral Reset Register

偏移地址：0x088

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	HSEMRST	ADC3RST	Res.	Res.	BDMARST	Res.	CRCRST	Res.	Res.	Res.
						rw	rw			rw		rw			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	GPIOKRST	GPIOJRST	GPIOIRST	GPIOHRST	GPIOGRST	GPIOFRST	GPIOERST	GPIODRST	GPIOCRST	GPIOBRST	GPIOARST
					rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:26 保留，必须保持复位值。

位 25 **HSEMRST**：HSEM 模块复位 (HSEM block reset)
由软件置 1 和复位。
0：不复位 HSEM 模块（复位后的默认值）
1：复位 HSEM 模块

位 24 **ADC3RST**：ADC3 模块复位 (ADC3 block reset)
由软件置 1 和复位。
0：不复位 ADC3 模块（复位后的默认值）
1：复位 ADC3 模块

位 23:22 保留，必须保持复位值。

位 21 **BDMARST**：BDMA 模块复位 (BDMA block reset)
由软件置 1 和复位。
0：不复位 BDMA 模块（复位后的默认值）
1：复位 BDMA 模块

位 20 保留，必须保持复位值。

位 19 **CRCRST**：CRC 模块复位 (CRC block reset)
由软件置 1 和复位。
0：不复位 CRC 模块（复位后的默认值）
1：复位 CRC 模块

位 18:11 保留，必须保持复位值。

位 10 **GPIOKRST**：GPIOK 模块复位 (GPIOK block reset)
由软件置 1 和复位。
0：不复位 GPIOK 模块（复位后的默认值）
1：复位 GPIOK 模块

位 9 **GPIOJRST**：GPIOJ 模块复位 (GPIOJ block reset)
由软件置 1 和复位。
0：不复位 GPIOJ 模块（复位后的默认值）
1：复位 GPIOJ 模块

位 8 **GPIOIRST**：GPIOI 模块复位 (GPIOI block reset)
由软件置 1 和复位。
0：不复位 GPIOI 模块（复位后的默认值）
1：复位 GPIOI 模块

位 7 **GPIOHRST**：GPIOH 模块复位 (GPIOH block reset)
由软件置 1 和复位。
0：不复位 GPIOH 模块（复位后的默认值）
1：复位 GPIOH 模块

位 6 **GPIOGRST**：GPIOG 模块复位 (GPIOG block reset)
由软件置 1 和复位。
0：不复位 GPIOG 模块（复位后的默认值）
1：复位 GPIOG 模块

位 5 **GPIOFRST**：GPIOF 模块复位 (GPIOF block reset)
由软件置 1 和复位。
0：不复位 GPIOF 模块（复位后的默认值）
1：复位 GPIOF 模块

位 4 **GPIOERST**：GPIOE 模块复位 (GPIOE block reset)
由软件置 1 和复位。
0：不复位 GPIOE 模块（复位后的默认值）
1：复位 GPIOE 模块

位 3 **GPIODRST**：GPIOD 模块复位 (GPIOD block reset)
由软件置 1 和复位。
0：不复位 GPIOD 模块（复位后的默认值）
1：复位 GPIOD 模块

位 2 **GPIOCRST**：GPIOC 模块复位 (GPIOC block reset)
由软件置 1 和复位。
0：不复位 GPIOC 模块（复位后的默认值）
1：复位 GPIOC 模块

位 1 **GPIOBRST**：GPIOB 模块复位 (GPIOB block reset)
由软件置 1 和复位。
0：不复位 GPIOB 模块（复位后的默认值）
1：复位 GPIOB 模块

位 0 **GPIOARST**：GPIOA 模块复位 (GPIOA block reset)
由软件置 1 和复位。
0：不复位 GPIOA 模块（复位后的默认值）
1：复位 GPIOA 模块

8.7.30 RCC APB3 外设复位寄存器 (RCC_APB3RSTR)

RCC APB3 Peripheral Reset Register

偏移地址：0x08C

复位值：0x0000 0000

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

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

位 31:4 保留，必须保持复位值。

位 3 **LTDCRST**：LTDC 模块复位 (LTDC block reset)
由软件置 1 和复位。
0：不复位 LTDC 模块（复位后的默认值）
1：复位 LTDC 模块

位 2:0 保留，必须保持复位值。

8.7.31 RCC APB1 外设复位寄存器 (RCC_APB1LRSTR)

RCC APB1 Peripheral Reset Register

偏移地址：0x090

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
UART8RST	UART7RST	DAC12RST	Res.	HDMICECRST	Res.	Res.	Res.	I2C3RST	I2C2RST	I2C1RST	UART5RST	UART4RST	USART3RST	USART2RST	SPDIFRXRST
rw	rw	rw		rw				rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SPI3RST	SPI2RST	Res.	Res.	Res.	Res.	LPTIM1RST	TIM14RST	TIM13RST	TIM12RST	TIM7RST	TIM6RST	TIM5RST	TIM4RST	TIM3RST	TIM2RST
rw	rw					rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31 **UART8RST**：UART8 模块复位 (UART8 block reset)
由软件置 1 和复位。
0：不复位 UART8 模块（复位后的默认值）
1：复位 UART8 模块

位 30 **UART7RST**：UART7 模块复位 (UART7 block reset)
由软件置 1 和复位。
0：不复位 UART7 模块（复位后的默认值）
1：复位 UART7 模块

位 29 **DAC12RST**：DAC1 和 2 模块复位 (DAC1 and 2 Blocks Reset)
由软件置 1 和复位。
0：不复位 DAC1 和 2 模块（复位后的默认值）
1：复位 DAC1 和 2 模块

位 28 保留，必须保持复位值。

位 27 **HDMICECRST**：HDMI-CEC 模块复位 (HDMI-CEC block reset)
由软件置 1 和复位。
0：不复位 HDMI-CEC 模块（复位后的默认值）
1：复位 HDMI-CEC 模块

位 26:24 保留，必须保持复位值。

位 23 **I2C3RST**：I2C3 模块复位 (I2C3 block reset)
由软件置 1 和复位。
0：不复位 I2C3 模块（复位后的默认值）
1：复位 I2C3 模块

位 22 **I2C2RST**：I2C2 模块复位 (I2C2 block reset)
由软件置 1 和复位。
0：不复位 I2C2 模块（复位后的默认值）
1：复位 I2C2 模块

位 21 **I2C1RST**：I2C1 模块复位 (I2C1 block reset)
由软件置 1 和复位。
0：不复位 I2C1 模块（复位后的默认值）
1：复位 I2C1 模块

位 20 **UART5RST**：UART5 模块复位 (UART5 block reset)
由软件置 1 和复位。
0：不复位 UART5 模块（复位后的默认值）
1：复位 UART5 模块

位 19 **UART4RST**：UART4 模块复位 (UART4 block reset)
由软件置 1 和复位。
0：不复位 UART4 模块（复位后的默认值）
1：复位 UART4 模块

位 18 **USART3RST**：USART3 模块复位 (USART3 block reset)
由软件置 1 和复位。
0：不复位 USART3 模块（复位后的默认值）
1：复位 USART3 模块

位 17 **USART2RST**：USART2 模块复位 (USART2 block reset)
由软件置 1 和复位。
0：不复位 USART2 模块（复位后的默认值）
1：复位 USART2 模块

位 16 **SPDIFRXRST**：SPDIFRX 模块复位 (SPDIFRX block reset)
由软件置 1 和复位。
0：不复位 SPDIFRX 模块（复位后的默认值）
1：复位 SPDIFRX 模块

位 15 **SPI3RST**：SPI3 模块复位 (SPI3 block reset)
由软件置 1 和复位。
0：不复位 SPI3 模块（复位后的默认值）
1：复位 SPI3 模块

位 14 **SPI2RST**：SPI2 模块复位 (SPI2 block reset)
由软件置 1 和复位。
0：不复位 SPI2 模块（复位后的默认值）
1：复位 SPI2 模块

位 13:10 保留，必须保持复位值。

位 9 **LPTIM1RST**：LPTIM1 模块复位 (LPTIM1 block reset)
由软件置 1 和复位。
0：不复位 LPTIM1 模块（复位后的默认值）
1：复位 LPTIM1 模块

位 8 **TIM14RST**：TIM14 模块复位 (TIM14 block reset)
由软件置 1 和复位。
0：不复位 TIM14 模块（复位后的默认值）
1：复位 TIM14 模块

位 7 **TIM13RST**：TIM13 模块复位 (TIM13 block reset)
由软件置 1 和复位。
0：不复位 TIM13 模块（复位后的默认值）
1：复位 TIM13 模块

位 6 **TIM12RST**：TIM12 模块复位 (TIM12 block reset)
由软件置 1 和复位。
0：不复位 TIM12 模块（复位后的默认值）
1：复位 TIM12 模块

位 5 **TIM7RST**：TIM7 模块复位 (TIM7 block reset)
由软件置 1 和复位。
0：不复位 TIM7 模块（复位后的默认值）
1：复位 TIM7 模块

位 4 **TIM6RST**：TIM6 模块复位 (TIM6 block reset)
由软件置 1 和复位。
0：不复位 TIM6 模块（复位后的默认值）
1：复位 TIM6 模块

位 3 **TIM5RST**：TIM5 模块复位 (TIM5 block reset)
由软件置 1 和复位。
0：不复位 TIM5 模块（复位后的默认值）
1：复位 TIM5 模块

位 2 **TIM4RST**：TIM4 模块复位 (TIM4 block reset)
由软件置 1 和复位。
0：不复位 TIM4 模块（复位后的默认值）
1：复位 TIM4 模块

位 1 **TIM3RST**：TIM3 模块复位 (TIM3 block reset)
由软件置 1 和复位。
0：不复位 TIM3 模块（复位后的默认值）
1：复位 TIM3 模块

位 0 **TIM2RST**：TIM2 模块复位 (TIM2 block reset)
由软件置 1 和复位。
0：不复位 TIM2 模块（复位后的默认值）
1：复位 TIM2 模块

8.7.32 RCC APB1 外设复位寄存器 (RCC_APB1HRSTR)

RCC APB1 Peripheral Reset Register

偏移地址：0x094

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	FDCANRST	Res.	Res.	MDIOSRST	OPAMPRST	Res.	SWPRST	CRSRST	Res.
							rw			rw	rw		rw	rw	

位 31:9 保留，必须保持复位值。

位 8 **FDCANRST**：FDCAN 模块复位 (FDCAN block reset)
由软件置 1 和复位。
0：不复位 FDCAN 模块（复位后的默认值）
1：复位 FDCAN 模块

位 7:6 保留，必须保持复位值。

位 5 **MDIOSRST**：MDIOS 模块复位 (MDIOS block reset)
由软件置 1 和复位。
0：不复位 MDIOS 模块（复位后的默认值）
1：复位 MDIOS 模块

位 4 **OPAMPRST**：OPAMP 模块复位 (OPAMP block reset)
由软件置 1 和复位。
0：不复位 OPAMP 模块（复位后的默认值）
1：复位 OPAMP 模块

位 3 保留，必须保持复位值。

位 2 **SWPRST**：SWPMI 模块复位 (SWPMI block reset)
由软件置 1 和复位。
0：不复位 SWPMI 模块（复位后的默认值）
1：复位 SWPMI 模块

位 1 **CRSRST**：时钟恢复系统复位 (Clock Recovery System reset)
由软件置 1 和复位。
0：不复位 CRS（复位后的默认值）
1：复位 CRS

位 0 保留，必须保持复位值。

8.7.33 RCC APB2 外设复位寄存器 (RCC_APB2RSTR)

RCC APB2 Peripheral Reset Register

偏移地址：0x098

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	HRTIMRST	DFSDM1RST	Res.	Res.	Res.	SAI3RST	SAI2RST	SAI1RST	Res.	SPI5RST	Res.	TIM17RST	TIM16RST	TIM15RST
		rw	rw				rw	rw	rw		rw		rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	SPI4RST	SPI1RST	Res.	Res.	Res.	Res.	Res.	Res.	USART6RST	USART1RST	Res.	Res.	TIM8RST	TIM1RST
		rw	rw							rw	rw			rw	rw

位 31:30 保留，必须保持复位值。

位 29 **HRTIMRST**：HRTIM 模块复位 (HRTIM block reset)
由软件置 1 和复位。
0：不复位 HRTIM 模块（复位后的默认值）
1：复位 HRTIM 模块

位 28 **DFSDM1RST**：DFSDM1 模块复位 (DFSDM1 block reset)
由软件置 1 和复位。
0：不复位 DFSDM1 模块（复位后的默认值）
1：复位 DFSDM1 模块

位 27:25 保留，必须保持复位值。

位 24 **SAI3RST**：SAI3 模块复位 (SAI3 block reset)
由软件置 1 和复位。
0：不复位 SAI3 模块（复位后的默认值）
1：复位 SAI3 模块

位 23 **SAI2RST**：SAI2 模块复位 (SAI2 block reset)
由软件置 1 和复位。
0：不复位 SAI2 模块（复位后的默认值）
1：复位 SAI2 模块

位 22 **SAI1RST**：SAI1 模块复位 (SAI1 block reset)
由软件置 1 和复位。
0：不复位 SAI1（复位后的默认值）
1：复位 SAI1

位 21 保留，必须保持复位值。

位 20 **SPI5RST**：SPI5 模块复位 (SPI5 block reset)
由软件置 1 和复位。
0：不复位 SPI5 模块（复位后的默认值）
1：复位 SPI5 模块

位 19 保留，必须保持复位值。

位 18 **TIM17RST**：TIM17 模块复位 (TIM17 block reset)
由软件置 1 和复位。
0：不复位 TIM17 模块（复位后的默认值）
1：复位 TIM17 模块

位 17 **TIM16RST**：TIM16 模块复位 (TIM16 block reset)
由软件置 1 和复位。
0：不复位 TIM16 模块（复位后的默认值）
1：复位 TIM16 模块

位 16 **TIM15RST**：TIM15 模块复位 (TIM15 block reset)
由软件置 1 和复位。
0：不复位 TIM15 模块（复位后的默认值）
1：复位 TIM15 模块

位 15:14 保留，必须保持复位值。

位 13 **SPI4RST**：SPI4 模块复位 (SPI4 block reset)
由软件置 1 和复位。
0：不复位 SPI4 模块（复位后的默认值）
1：复位 SPI4 模块

位 12 **SPI1RST**：SPI1 模块复位 (SPI1 block reset)
由软件置 1 和复位。
0：不复位 SPI1 模块（复位后的默认值）
1：复位 SPI1 模块

位 11:6 保留，必须保持复位值。

位 5 **USART6RST**：USART6 模块复位 (USART6 block reset)
由软件置 1 和复位。
0：不复位 USART6 模块（复位后的默认值）
1：复位 USART6 模块

位 4 **USART1RST**：USART1 模块复位 (USART1 block reset)
由软件置 1 和复位。
0：不复位 USART1 模块（复位后的默认值）
1：复位 USART1 模块

位 3:2 保留，必须保持复位值。

位 1 **TIM8RST**：TIM8 模块复位 (TIM8 block reset)
由软件置 1 和复位。
0：不复位 TIM8 模块（复位后的默认值）
1：复位 TIM8 模块

位 0 **TIM1RST**：TIM1 模块复位 (TIM1 block reset)
由软件置 1 和复位。
0：不复位 TIM1 模块（复位后的默认值）
1：复位 TIM1 模块

8.7.34 RCC APB4 外设复位寄存器 (RCC_APB4RSTR)

RCC APB4 Peripheral Reset Register

偏移地址：0x09C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SAI4RST	Res.	Res.	Res.	Res.	Res.
										rw					

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VREFRST	COMP12RST	Res.	LPTIM5RST	LPTIM4RST	LPTIM3RST	LPTIM2RST	Res.	I2C4RST	Res.	SPI6RST	Res.	LPUART1RST	Res.	SYSCFGRST	Res.
rw	rw		rw	rw	rw	rw		rw		rw		rw		rw	

位 31:26 保留，必须保持复位值。

位 25:22 保留，必须保持复位值。

位 21 **SAI4RST**：SAI4 模块复位 (SAI4 block reset)
由软件置 1 和复位。
0：不复位 SAI4 模块（复位后的默认值）
1：复位 SAI4 模块

位 20:16 保留，必须保持复位值。

位 15 **VREFRST**：VREF 模块复位 (VREF block reset)
由软件置 1 和复位。
0：不复位 VREF 模块（复位后的默认值）
1：复位 VREF 模块

位 14 **COMP12RST**：COMP12 模块复位 (COMP12 Blocks Reset)
由软件置 1 和复位。
0：不复位 COMP 1 和 2 模块（复位后的默认值）
1：复位 COMP 1 和 2 模块

位 13 保留，必须保持复位值。

位 12 **LPTIM5RST**：LPTIM5 模块复位 (LPTIM5 block reset)
由软件置 1 和复位。
0：不复位 LPTIM5 模块（复位后的默认值）
1：复位 LPTIM5 模块

位 11 **LPTIM4RST**：LPTIM4 模块复位 (LPTIM4 block reset)
由软件置 1 和复位。
0：不复位 LPTIM4 模块（复位后的默认值）
1：复位 LPTIM4 模块

位 10 **LPTIM3RST**：LPTIM3 模块复位 (LPTIM3 block reset)
由软件置 1 和复位。
0：不复位 LPTIM3 模块（复位后的默认值）
1：复位 LPTIM3 模块

位 9 **LPTIM2RST**：LPTIM2 模块复位 (LPTIM2 block reset)
由软件置 1 和复位。
0：不复位 LPTIM2 模块（复位后的默认值）
1：复位 LPTIM2 模块

位 8 保留，必须保持复位值。

位 7 **I2C4RST**：I2C4 模块复位 (I2C4 block reset)
由软件置 1 和复位。
0：不复位 I2C4 模块（复位后的默认值）
1：复位 I2C4 模块

位 6 保留，必须保持复位值。

位 5 **SPI6RST**：SPI6 模块复位 (SPI6 block reset)
由软件置 1 和复位。
0：不复位 SPI6 模块（复位后的默认值）
1：复位 SPI6 模块

位 4 保留，必须保持复位值。

位 3 **LPUART1RST**：LPUART1 模块复位 (LPUART1 block reset)
由软件置 1 和复位。
0：不复位 LPUART1 模块（复位后的默认值）
1：复位 LPUART1 模块

位 2 保留，必须保持复位值。

位 1 **SYSCFGRST**：SYSCFG 模块复位 (SYSCFG block reset)
由软件置 1 和复位。
0：不复位 SYSCFG 模块（复位后的默认值）
1：复位 SYSCFG 模块

位 0 保留，必须保持复位值。

8.7.35 RCC 全局控制寄存器 (RCC_GCR)

RCC Global Control Register

偏移地址：0x0A0

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WW1RSC
															rw1

位 31:1 保留，必须保持复位值。

位 0 **WW1RSC**：WWDG1 复位范围控制 (WWDG1 reset scope control)
此位可由软件置 1，但在系统复位期间由硬件清零
为了确保正常工作，在使能 WWDG1 之前，必须将该位置“1”。

8.7.36 RCC D3 自主模式寄存器 (RCC_D3AMR)

RCC D3 Autonomous mode Register

自主模式允许向 D3 中的外设提供外设时钟，即使 CPU 处于 CStop 模式亦如此。如果外设已使能且其自主位已使能，则会在 CPU 处于 CStop 模式时根据 D3 域状态接收相应的外设时钟。

偏移地址：0x0A8

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	SRAM4AMEN	BKPRAMAMEN	Res.	Res.	Res.	ADC3AMEN	Res.	Res.	SAI4AMEN	Res.	CRCAMEN	Res.	Res.	RTCAMEN
		rw	rw				rw			rw		rw			rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VREFAMEN	COMP12AMEN	Res.	LPTIM5AMEN	LPTIM4AMEN	LPTIM3AMEN	LPTIM2AMEN	Res.	I2C4AMEN	Res.	SPI6AMEN	Res.	LPUART1AMEN	Res.	Res.	BDMAAMEN
rw	rw		rw	rw	rw	rw		rw		rw		rw			rw

位 31:30 保留，必须保持复位值。

位 29 **SRAM4AMEN**：SRAM4 自主模式使能 (SRAM4 Autonomous mode enable)
由软件置 1 和复位。
0：CPU 处于 Cstop 模式时禁止 SRAM4 时钟（复位后的默认值）
1：D3 域处于 DRun 模式时使能 SRAM4 外设总线时钟
有关更多信息，请参见*第 8.5.11 节：外设时钟门控*。

位 28 **BKPSRAMAMEN**：备份 RAM 自主模式使能 (Backup RAM Autonomous mode enable)
由软件置 1 和复位。
0：CPU 处于 CStop 模式时禁止备份 RAM 时钟（复位后的默认值）
1：备份 RAM 时钟的使能由 D3 域状态控制
有关更多信息，请参见*第 8.5.11 节：外设时钟门控*。

位 27 保留，必须保持复位值。

位 26:25 保留，必须保持复位值。

位 24 **ADC3AMEN**：ADC3 自主模式使能 (ADC3 Autonomous mode enable)
由软件置 1 和复位。
0：CPU 处于 CStop 模式时禁止 ADC3 外设时钟（复位后的默认值）
1：D3 域处于 DRun 模式时使能 ADC3 外设时钟
有关更多信息，请参见*第 8.5.11 节：外设时钟门控*。

位 23:22 保留，必须保持复位值。

位 21 **SAI4AMEN**：SAI4 自主模式使能 (SAI4 Autonomous mode enable)
由软件置 1 和复位。
0：CPU 处于 CStop 模式时禁止 SAI4 外设时钟（复位后的默认值）
1：D3 域处于 DRun 模式时使能 SAI4 外设时钟
有关更多信息，请参见*第 8.5.11 节：外设时钟门控*。

位 20 保留，必须保持复位值。

位 19 **CRCAMEN**：CRC 自主模式使能 (CRC Autonomous mode enable)
由软件置 1 和复位。
0：CPU 处于 CStop 模式时禁止 CRC 外设时钟（复位后的默认值）
1：D3 域处于 DRun 模式时使能 CRC 外设时钟
有关更多信息，请参见*第 8.5.11 节：外设时钟门控*。

位 18:17 保留，必须保持复位值。

位 16 **RTCAMEN**：RTC 自主模式使能 (RTC Autonomous mode enable)
由软件置 1 和复位。
0：CPU 处于 CStop 模式时禁止 RTC 外设时钟（复位后的默认值）
1：D3 域处于 DRun 模式时使能 RTC 外设时钟
有关更多信息，请参见*第 8.5.11 节：外设时钟门控*。

位 15 **VREFAMEN**：VREF 自主模式使能 (VREF Autonomous mode enable)
由软件置 1 和复位。
0：CPU 处于 CStop 模式时禁止 VREF 外设时钟（复位后的默认值）
1：D3 域处于 DRun 模式时使能 VREF 外设时钟
有关更多信息，请参见*第 8.5.11 节：外设时钟门控*。

位 14 **COMP12AMEN**：COMP12 自主模式使能 (COMP12 Autonomous mode enable)
由软件置 1 和复位。
0：CPU 处于 CStop 模式时禁止 COMP12 外设时钟（复位后的默认值）
1：D3 域处于 DRun 模式时使能 COMP12 外设时钟
有关更多信息，请参见*第 8.5.11 节：外设时钟门控*。

位 13 保留，必须保持复位值。

位 12 **LPTIM5AMEN**：LPTIM5 自主模式使能 (LPTIM5 Autonomous mode enable)
由软件置 1 和复位。
0：CPU 处于 CStop 模式时禁止 LPTIM5 外设时钟（复位后的默认值）
1：D3 域处于 DRun 模式时使能 LPTIM5 外设时钟
有关更多信息，请参见*第 8.5.11 节：外设时钟门控*。

位 11 **LPTIM4AMEN**：LPTIM4 自主模式使能 (LPTIM4 Autonomous mode enable)
由软件置 1 和复位。
0：CPU 处于 CStop 模式时禁止 LPTIM4 外设时钟（复位后的默认值）
1：D3 域处于 DRun 模式时使能 LPTIM4 外设时钟
有关更多信息，请参见*第 8.5.11 节：外设时钟门控*。

位 10 **LPTIM3AMEN**：LPTIM3 自主模式使能 (LPTIM3 Autonomous mode enable)
由软件置 1 和复位。
0：CPU 处于 CStop 模式时禁止 LPTIM3 外设时钟（复位后的默认值）
1：D3 域处于 DRun 模式时使能 LPTIM3 外设时钟
有关更多信息，请参见*第 8.5.11 节：外设时钟门控*。

位 9 **LPTIM2AMEN**：LPTIM2 自主模式使能 (LPTIM2 Autonomous mode enable)
由软件置 1 和复位。
0：CPU 处于 CStop 模式时禁止 LPTIM2 外设时钟（复位后的默认值）
1：D3 域处于 DRun 模式时使能 LPTIM2 外设时钟
有关更多信息，请参见*第 8.5.11 节：外设时钟门控*。

位 8 保留，必须保持复位值。

位 7 **I2C4AMEN**：I2C4 自主模式使能 (I2C4 Autonomous mode enable)
由软件置 1 和复位。
0：CPU 处于 CStop 模式时禁止 I2C4 外设时钟（复位后的默认值）
1：D3 域处于 DRun 模式时使能 I2C4 外设时钟
有关更多信息，请参见*第 8.5.11 节：外设时钟门控*。

位 6 保留，必须保持复位值。

位 5 **SPI6AMEN**：SPI6 自主模式使能 (SPI6 Autonomous mode enable)
由软件置 1 和复位。
0：CPU 处于 CStop 模式时禁止 SPI6 外设时钟（复位后的默认值）
1：D3 域处于 DRun 模式时使能 SPI6 外设时钟
有关更多信息，请参见*第 8.5.11 节：外设时钟门控*。

位 4 保留，必须保持复位值。

位 3 **LPUART1AMEN**：LPUART1 自主模式使能 (LPUART1 Autonomous mode enable)
由软件置 1 和复位。
0：CPU 处于 CStop 模式时禁止 LPUART1 外设时钟（复位后的默认值）
1：D3 域处于 DRun 模式时使能 LPUART1 外设时钟
有关更多信息，请参见*第 8.5.11 节：外设时钟门控*。

位 2:1 保留，必须保持复位值。

位 0 **BDMAAMEN**：BDMA 和 DMAMUX 自主模式使能 (BDMA and DMAMUX Autonomous mode enable)
由软件置 1 和复位。
0：CPU 处于 CStop 模式时禁止 BDMA 和 DMAMUX 外设时钟（复位后的默认值）
1：D3 域处于 DRun 模式时使能 BDMA 和 DMAMUX 外设时钟
有关更多信息，请参见*第 8.5.11 节：外设时钟门控*。

8.7.37 RCC 复位状态寄存器 (RCC_RSR)

RCC Reset Status Register

该寄存器可通过两个不同的偏移地址进行访问。

表 55. RCC_RSR 偏移地址和复位值

寄存器名	偏移地址	复位值
RCC_RSR	0x0D0	0x00FE 0000 (1)
RCC_C1_RSR	0x130	

1. 仅通过上电复位实现复位

访问：0 ≤ 等待周期 ≤ 7，按字、半字和字节访问。连续访问该寄存器时，则插入等待周期。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	LPWRRSTF	Res.	WWDG1RSTF	Res.	IWDG1RSTF	Res.	SFTRSTF	PORRSTF	PINRSTF;	BORRSTF	D2RSTF	D1RSTF	Res.	CPURSTF	RMVF
	r		r		r		r	r	r	r	r	r		r	rw

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

位 31 保留，必须保持复位值。

位 30 **LPWRRSTF**：因非法 D1 DStandby 或 CPU CStop 标志而复位 (Reset due to illegal D1 DStandby or CPU CStop flag) (1)

通过软件写入 RMVF 位进行复位。
D1 域错误地进入 DStandby 模式时或 CPU 错误地进入 CStop 模式时，由硬件置 1。
0：未发生非法复位（上电复位后的默认值）
1：发生非法 D1 DStandby 或 CPU CStop 复位

位 29 保留，必须保持复位值。

位 28 **WWDG1RSTF**：窗口看门狗复位标志 (Window watchdog reset flag)(1)

通过软件写入 RMVF 位进行复位。
发生窗口看门狗复位时，由硬件置 1。
0：WWDG1 未发生窗口看门狗复位（上电复位后的默认值）
1：WWDG1 发生窗口看门狗复位

位 27 保留，必须保持复位值。

位 26 **IWDG1RSTF**：独立看门狗复位标志 (Independent watchdog reset flag)(1)

通过软件写入 RMVF 位进行复位。
发生独立看门狗复位时，由硬件置 1。
0：未发生独立看门狗复位（上电复位后的默认值）
1：发生独立看门狗复位

位 25 保留，必须保持复位值。

位 24 **SFTRSTF**：通过 CPU 实现系统复位的复位标志 (System reset from CPU reset flag) (1)
通过软件写入 RMVF 位进行复位。
系统复位是因 CPU 引起时，由硬件置 1。CPU 可通过写入 CM7 AIRCR 寄存器的 SYSRESETREQ 位生成系统复位。
0：未发生 CPU 软件复位（上电复位后的默认值）
1：CPU 已生成系统复位

位 23 **PORRSTF**：POR/PDR 复位标志 (POR/PDR reset flag) (1)
通过软件写入 RMVF 位进行复位。
发生 POR/PDR 复位时，由硬件置 1。
0：未发生 POR/PDR 复位
1：发生 POR/PDR 复位（上电复位后的默认值）

位 22 **PINRSTF**：引脚复位标志 (NRST) (Pin reset flag (NRST)) (1)
通过软件写入 RMVF 位进行复位。
发生来自引脚的复位时，由硬件置 1。
0：未发生来自引脚的复位
1：发生来自引脚的复位（上电复位后的默认值）

位 21 **BORRSTF**：BOR 复位标志 (BOR reset flag) (1)
通过软件写入 RMVF 位进行复位。
发生 BOR 复位 (**pwr_bor_rst**) 时，由硬件置 1。
0：未发生 BOR 复位
1：发生 BOR 复位（上电复位后的默认值）

位 20 **D2RSTF**：D2 域电源开关复位标志 (D2 domain power switch reset flag) (1)
通过软件写入 RMVF 位进行复位。
D2 域退出 DStandby 时或上电复位后，由硬件置 1。有关详细信息，请参见*表 46*。
0：未发生 D2 域电源开关复位
1：发生 D2 域电源开关 (ePOD2) 复位（上电复位后的默认值）

位 19 **D1RSTF**：D1 域电源开关复位标志 (D1 domain power switch reset flag) (1)
通过软件写入 RMVF 位进行复位。
D1 域退出 DStandby 时或上电复位后，由硬件置 1。有关详细信息，请参见*表 46*。
0：未发生 D1 域电源开关复位
1：发生 D1 域电源开关 (ePOD1) 复位（上电复位后的默认值）

位 18 保留，必须保持复位值。

位 17 **CPURSTF**：CPU 复位标志 (CPU reset flag) (1)
通过软件写入 RMVF 位进行复位。
每次发生 CPU 复位时，均由硬件置 1。
0：未发生 CPU 复位
1：发生 CPU 复位（上电复位后的默认值）

位 16 **RMVF**：清除复位标志 (Remove reset flag)
由软件置 1 和复位，用于对复位标志的值进行复位。
0：未激活对复位标志的复位（上电复位后的默认值）
1：对复位标志的值进行复位

位 15:0 保留，必须保持复位值。

1. 有关标志行为的详细信息，请参见*表 46：复位源标识 (RCC_RSR)*。

8.7.38 RCC AHB3 时钟寄存器 (RCC_AHB3ENR)

RCC AHB3 Clock Register

该寄存器可通过两个不同的偏移地址进行访问。

表 56. RCC_AHB3ENR 偏移地址和复位值

寄存器名	偏移地址	复位值
RCC_AHB3ENR	0x0D4	0x0000 0000
RCC_C1_AHB3ENR	0x134	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SDMMC1EN
															rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	QSPIEN	Res.	FMCEN	Res.	Res.	Res.	Res.	Res.	Res.	JPGDECEN	DMA2DEN	Res.	Res.	Res.	MDMAEN
	rw		rw							rw	rw				rw

位 31:17 保留，必须保持复位值。

位 16 **SDMMC1EN**：SDMMC1 和 SDMMC1 延迟时钟使能 (SDMMC1 and SDMMC1 Delay Clock Enable)
由软件置 1 和复位。
0：禁止 SDMMC1 和 SDMMC1 延迟时钟（复位后的默认值）
1：使能 SDMMC1 和 SDMMC1 延迟时钟

位 15 保留，必须保持复位值。

位 14 **QSPIEN**：QUADSPI 和 QUADSPI 延迟时钟使能 (QUADSPI and QUADSPI Delay Clock Enable)
由软件置 1 和复位。
0：禁止 QUADSPI 和 QUADSPI 延迟时钟（复位后的默认值）
1：使能 QUADSPI 和 QUADSPI 延迟时钟

位 13 保留，必须保持复位值。

位 12 **FMCEN**：FMC 外设时钟使能 (FMC Peripheral Clocks Enable)
由软件置 1 和复位。
0：禁止 FMC 外设时钟（复位后的默认值）
1：使能 FMC 外设时钟
FMC 的外设时钟为：由 FMCSEL 选择并提供给 fmc_ker_ck 输入的内核时钟，以及 **rcc_hclk3** 总线接口时钟。

位 11:6 保留，必须保持复位值。

位 5 **JPGDECEN**：JPGDEC 外设时钟使能 (JPGDEC Peripheral Clock Enable)
由软件置 1 和复位。
0：禁止 JPGDEC 外设时钟（复位后的默认值）
1：使能 JPGDEC 外设时钟

位 4 **DMA2DEN**：DMA2D 外设时钟使能 (DMA2D Peripheral Clock Enable)
由软件置 1 和复位。
0：禁止 DMA2D 外设时钟（复位后的默认值）
1：使能 DMA2D 外设时钟

位 3:1 保留，必须保持复位值。

位 0 **MDMAEN**：MDMA 外设时钟使能 (MDMA Peripheral Clock Enable)
由软件置 1 和复位。
0：禁止 MDMA 外设时钟（复位后的默认值）
1：使能 MDMA 外设时钟

8.7.39 RCC AHB1 时钟寄存器 (RCC_AHB1ENR)

RCC AHB1 Clock Register

该寄存器可通过两个不同的偏移地址进行访问。

表 57. RCC_AHB1ENR 偏移地址和复位值

寄存器名	偏移地址	复位值
RCC_AHB1ENR	0x0D8	0x0000 0000
RCC_C1_AHB1ENR	0x138	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	USB2ULPIEN	USB2OTGEN	USB1ULPIEN	USB1OTGEN	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ETH1RXEN	ETH1TXEN
			rw	rw	rw	rw								rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ETH1MACEN	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ADC12EN	Res.	Res.	Res.	DMA2EN	DMA1EN
rw										rw				rw	rw

位 31:29 保留，必须保持复位值。

位 28 **USB2ULPIEN**：USB_PHY2 时钟使能 (USB_PHY2 Clocks Enable)
由软件置 1 和复位。
0：禁止 USB_PHY2 时钟（复位后的默认值）
1：使能 USB_PHY2 时钟

位 27 **USB2OTGEN**：USB2OTG 外设时钟使能 (USB2OTG Peripheral Clocks Enable)
由软件置 1 和复位。
0：禁止 USB2OTG 外设时钟（复位后的默认值）
1：使能 USB2OTG 外设时钟
USB2OTG 的外设时钟为：由 USBSEL 选择的内核时钟，以及 **rcc_hclk1** 总线接口时钟。

位 26 **USB1ULPIEN**：USB_PHY1 时钟使能 (USB_PHY1 Clocks Enable)
由软件置 1 和复位。
0：禁止 USB1ULPI PHY 时钟（复位后的默认值）
1：使能 USB1ULPI PHY 时钟

位 25 **USB1OTGEN**：USB1OTG 外设时钟使能 (USB1OTG Peripheral Clocks Enable)
由软件置 1 和复位。
0：禁止 USB1OTG 外设时钟（复位后的默认值）
1：使能 USB1OTG 外设时钟
USB1OTG 的外设时钟为：由 USBSEL 选择的内核时钟，以及 **rcc_hclk1** 总线接口时钟。

位 24:18 保留，必须保持复位值。

位 17 **ETH1RXEN**：以太网接收时钟使能 (Ethernet Reception clock enable)
由软件置 1 和复位。
0：禁止以太网接收时钟（复位后的默认值）
1：使能以太网接收时钟

位 16 **ETH1TXEN**：以太网发送时钟使能 (Ethernet Transmission clock enable)
由软件置 1 和复位。
0：禁止以太网发送时钟（复位后的默认值）
1：使能以太网发送时钟

位 15 **ETH1MACEN**：以太网 MAC 总线接口时钟使能 (Ethernet MAC bus interface Clock Enable)
由软件置 1 和复位。
0：禁止以太网 MAC 总线接口时钟（复位后的默认值）
1：使能以太网 MAC 总线接口时钟

位 14:6 保留，必须保持复位值。

位 5 **ADC12EN**：ADC1/2 外设时钟使能 (ADC1/2 Peripheral Clocks Enable)
由软件置 1 和复位。
0：禁止 ADC1 和 2 外设时钟（复位后的默认值）
1：使能 ADC1 和 2 外设时钟
ADC1 和 2 的外设时钟为：由 ADCSEL 选择并提供给 adc_ker_ck 输入的内核时钟，以及 **rcc_hclk1** 总线接口时钟。

位 4:2 保留，必须保持复位值。

位 1 **DMA2EN**：DMA2 时钟使能 (DMA2 clock enable)
由软件置 1 和复位。
0：禁止 DMA2 时钟（复位后的默认值）
1：使能 DMA2 时钟

位 0 **DMA1EN**：DMA1 时钟使能 (DMA1 clock enable)
由软件置 1 和复位。
0：禁止 DMA1 时钟（复位后的默认值）
1：使能 DMA1 时钟

8.7.40 RCC AHB2 时钟寄存器 (RCC_AHB2ENR)

RCC AHB2 Clock Register

该寄存器可通过两个不同的偏移地址进行访问。

表 58. RCC_AHB2ENR 偏移地址和复位值

寄存器名	偏移地址	复位值
RCC_AHB2ENR	0x0DC	0x0000 0000
RCC_C1_AHB2ENR	0x13C	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SRAM3EN	SRAM2EN	SRAM1EN	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
rw	rw	rw													

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	SDMMC2EN	Res.	Res.	RNGEN	HASHEN	CRYPTEN	Res.	Res.	Res.	DCMIEN
						rw			rw	rw	rw				rw

位 31 **SRAM3EN**：SRAM3 模块使能 (SRAM3 block enable)

由软件置 1 和复位。

该位置 1 时，指示 SRAM3 已由 CPU 进行分配。这会导致 D2 域还会考虑到 CPU 工作模式，即在 CPU 处于 CRun 模式时会使 D2 域保持在 DRun 模式下。

0：禁止 SRAM3 接口时钟。（复位后的默认值）

1：使能 SRAM3 接口时钟。

位 30 **SRAM2EN**：SRAM2 模块使能 (SRAM2 block enable)

由软件置 1 和复位。

该位置 1 时，指示 SRAM2 已由 CPU 进行分配。这会导致 D2 域还会考虑到 CPU 工作模式，即在 CPU 处于 CRun 模式时会使 D2 域保持在 DRun 模式下。

0：禁止 SRAM2 接口时钟。（复位后的默认值）

1：使能 SRAM2 接口时钟。

位 29 **SRAM1EN**：SRAM1 模块使能 (SRAM1 block enable)

由软件置 1 和复位。

该位置 1 时，指示 SRAM1 已由 CPU 进行分配。这会导致 D2 域还会考虑到 CPU 工作模式，即在 CPU 处于 CRun 模式时会使 D2 域保持在 DRun 模式下。

0：禁止 SRAM1 接口时钟。（复位后的默认值）

1：使能 SRAM1 接口时钟。

位 28:10 保留，必须保持复位值。

位 9 **SDMMC2EN**：SDMMC2 和 SDMMC2 延迟时钟使能 (SDMMC2 and SDMMC2 delay clock enable)

由软件置 1 和复位。

0：禁止 SDMMC2 和 SDMMC2 延迟时钟（复位后的默认值）

1：使能 SDMMC2 和 SDMMC2 延迟时钟

位 8:7 保留，必须保持复位值。

位 6 **RNGEN**：RNG 外设时钟使能 (RNG peripheral clocks enable)
由软件置 1 和复位。
0：禁止 RNG 外设时钟（复位后的默认值）
1：使能 RNG 外设时钟：
RNG 的外设时钟为：由 RNGSEL 选择并提供给 **rng_ker_ck** 输入的内核时钟，以及 **rcc_hclk2** 总线接口时钟。

位 5 **HASHEN**：HASH 外设时钟使能 (HASH peripheral clock enable)
由软件置 1 和复位。
0：禁止 HASH 外设时钟（复位后的默认值）
1：使能 HASH 外设时钟

位 4 **CRYPTEN**：CRYPT 外设时钟使能 (CRYPT peripheral clock enable)
由软件置 1 和复位。
0：禁止 CRYPT 外设时钟（复位后的默认值）
1：使能 CRYPT 外设时钟

位 3:1 保留，必须保持复位值。

位 0 **DCMIEN**：DCMI 外设时钟使能 (DCMI peripheral clock enable)
由软件置 1 和复位。
0：禁止 DCMI 外设时钟（复位后的默认值）
1：使能 DCMI 外设时钟

8.7.41 RCC AHB4 时钟寄存器 (RCC_AHB4ENR)

RCC AHB4 Clock Register

该寄存器可通过两个不同的偏移地址进行访问。

表 59. RCC_AHB4ENR 偏移地址和复位值

寄存器名	偏移地址	复位值
RCC_AHB4ENR	0x0E0	0x0000 0000
RCC_C1_AHB4ENR	0x140	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	BKPRAMEN	Res.	Res.	HSEMEN	ADC3EN	Res.	Res.	BDMAEN	Res.	CRCEN	Res.	Res.	Res.
			rw			rw	rw			rw		rw			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	GPIOKEN	GPIOJEN	GPIOIEN	GPIOHEN	GPIOGEN	GPIOFEN	GPIOEEN	GPIODEN	GPIOCEN	GPIOBEN	GPIOAEN
					rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:29 保留，必须保持复位值。

位 28 **BKPRAMEN**：备份 RAM 时钟使能 (Backup RAM Clock Enable)
由软件置 1 和复位。
0：禁止备份 RAM 时钟（复位后的默认值）
1：使能备份 RAM 时钟

位 27:26 保留，必须保持复位值。

位 25 **HSEMEN**：HSEM 外设时钟使能 (HSEM peripheral clock enable)
由软件置 1 和复位。
0：禁止 HSEM 外设时钟（复位后的默认值）
1：使能 HSEM 外设时钟

位 24 **ADC3EN**：ADC3 外设时钟使能 (ADC3 Peripheral Clocks Enable)
由软件置 1 和复位。
0：禁止 ADC3 外设时钟（复位后的默认值）
1：使能 ADC3 外设时钟
ADC3 的外设时钟为：由 ADCSEL 选择并提供给 adc_ker_ck 输入的内核时钟，以及 **rcc_hclk4** 总线接口时钟。

位 23:22 保留，必须保持复位值。

位 21 **BDMAEN**：BDMA 和 DMAMUX2 时钟使能 (BDMA and DMAMUX2 Clock Enable)
由软件置 1 和复位。
0：禁止 BDMA 和 DMAMUX2 时钟（复位后的默认值）
1：使能 BDMA 和 DMAMUX2 时钟

位 20 保留，必须保持复位值。

位 19 **CRCEN**：CRC 外设时钟使能 (CRC peripheral clock enable)
由软件置 1 和复位。
0：禁止 CRC 外设时钟（复位后的默认值）
1：使能 CRC 外设时钟

位 18:11 保留，必须保持复位值。

位 10 **GPIOKEN**：GPIOK 外设时钟使能 (GPIOK peripheral clock enable)
由软件置 1 和复位。
0：禁止 GPIOK 外设时钟（复位后的默认值）
1：使能 GPIOK 外设时钟

位 9 **GPIOJEN**：GPIOJ 外设时钟使能 (GPIOJ peripheral clock enable)
由软件置 1 和复位。
0：禁止 GPIOJ 外设时钟（复位后的默认值）
1：使能 GPIOJ 外设时钟

位 8 **GPIOIEN**：GPIOI 外设时钟使能 (GPIOI peripheral clock enable)
由软件置 1 和复位。
0：禁止 GPIOI 外设时钟（复位后的默认值）
1：使能 GPIOI 外设时钟

位 7 **GPIOHEN**：GPIOH 外设时钟使能 (GPIOH peripheral clock enable)
由软件置 1 和复位。
0：禁止 GPIOH 外设时钟（复位后的默认值）
1：使能 GPIOH 外设时钟

位 6 **GPIOGEN**：GPIOG 外设时钟使能 (GPIOG peripheral clock enable)
由软件置 1 和复位。
0：禁止 GPIOG 外设时钟（复位后的默认值）
1：使能 GPIOG 外设时钟

位 5 **GPIOFEN**：GPIOF 外设时钟使能 (GPIOF peripheral clock enable)
由软件置 1 和复位。
0：禁止 GPIOF 外设时钟（复位后的默认值）
1：使能 GPIOF 外设时钟

位 4 **GPIOEEN**：GPIOE 外设时钟使能 (GPIOE peripheral clock enable)
由软件置 1 和复位。
0：禁止 GPIOE 外设时钟（复位后的默认值）
1：使能 GPIOE 外设时钟

位 3 **GPIODEN**：GPIOD 外设时钟使能 (GPIOD peripheral clock enable)
由软件置 1 和复位。
0：禁止 GPIOD 外设时钟（复位后的默认值）
1：使能 GPIOD 外设时钟

位 2 **GPIOCEN**：GPIOC 外设时钟使能 (GPIOC peripheral clock enable)
由软件置 1 和复位。
0：禁止 GPIOC 外设时钟（复位后的默认值）
1：使能 GPIOC 外设时钟

位 1 **GPIOBEN**：GPIOB 外设时钟使能 (GPIOB peripheral clock enable)
由软件置 1 和复位。
0：禁止 GPIOB 外设时钟（复位后的默认值）
1：使能 GPIOB 外设时钟

位 0 **GPIOAEN**：GPIOA 外设时钟使能 (GPIOA peripheral clock enable)
由软件置 1 和复位。
0：禁止 GPIOA 外设时钟（复位后的默认值）
1：使能 GPIOA 外设时钟

8.7.42 RCC APB3 时钟寄存器 (RCC_APB3ENR)

RCC APB3 Clock Register

该寄存器可通过两个不同的偏移地址进行访问。

表 60. RCC_APB3ENR 偏移地址和复位值

寄存器名	偏移地址	复位值
RCC_APB3ENR	0x0E4	0x0000 0000
RCC_C1_APB3ENR	0x144	

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WWDG1EN	Res.	Res.	LTDCEN	Res.	Res.	Res.
									rs			rw			

位 31:7 保留，必须保持复位值。

位 6 **WWDG1EN**：WWDG1 时钟使能 (WWDG1 Clock Enable)
由软件置 1，在发生系统复位时由硬件复位。
请注意，为了确保正常工作，在使能 WWDG1 之前，必须将位 WW1RSC 置“1”。
0：禁止 WWDG1 外设时钟（复位后的默认值）
1：使能 WWDG1 外设时钟

位 5:4 保留，必须保持复位值。

位 3 **LTDCEN**：LTDC 外设时钟使能 (LTDC peripheral clock enable)
向 LTDC 模块提供像素时钟 (**ltdc_ker_ck**)。
由软件置 1 和复位。
0：禁止 LTDC 外设时钟（复位后的默认值）
1：向 LTDC 模块提供 LTDC 外设时钟

位 2:0 保留，必须保持复位值。

8.7.43 RCC APB1 时钟寄存器 (RCC_APB1LENR)

RCC APB1 Clock Register

该寄存器可通过两个不同的偏移地址进行访问。

表 61. RCC_APB1ENR 偏移地址和复位值

寄存器名	偏移地址	复位值
RCC_APB1LENR	0x0E8	0x0000 0000
RCC_C1_APB1LENR	0x148	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
UART8EN	UART7EN	DAC12EN	Res.	HDMICECEN	Res.	Res.	Res.	I2C3EN	I2C2EN	I2C1EN	UART5EN	UART4EN	USART3EN	USART2EN	SPDIFRXEN
rw	rw	rw		rw				rw	rw	rw	rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SPI3EN	SPI2EN	Res.	Res.	Res.	Res.	LPTIM1EN	TIM14EN	TIM13EN	TIM12EN	TIM7EN	TIM6EN	TIM5EN	TIM4EN	TIM3EN	TIM2EN
rw	rw					rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31 **UART8EN**：UART8 外设时钟使能 (UART8 Peripheral Clocks Enable)
由软件置 1 和复位。
0：禁止 UART8 外设时钟（复位后的默认值）
1：使能 UART8 外设时钟
UART8 的外设时钟为：由 USART234578SEL 选择并提供给 usart_ker_ck 输入的内核时钟，以及 **rcc_pclk1** 总线接口时钟。

位 30 **UART7EN**：UART7 外设时钟使能 (UART7 Peripheral Clocks Enable)
由软件置 1 和复位。
0：禁止 UART7 外设时钟（复位后的默认值）
1：使能 UART7 外设时钟
UART7 的外设时钟为：由 USART234578SEL 选择并提供给 usart_ker_ck 输入的内核时钟，以及 **rcc_pclk1** 总线接口时钟。

位 29 **DAC12EN**：DAC 1 和 2 外设时钟使能 (DAC 1 and 2 peripheral clock enable)
由软件置 1 和复位。
0：禁止 DAC 1 和 2 外设时钟（复位后的默认值）
1：使能 DAC 1 和 2 外设时钟

位 28 保留，必须保持复位值。

位 27 **HDMICECEN**：HDMI-CEC 外设时钟使能 (HDMI-CEC peripheral clock enable)
由软件置 1 和复位。
0：禁止 HDMI-CEC 外设时钟（复位后的默认值）
1：使能 HDMI-CEC 外设时钟
HDMI-CEC 的外设时钟为：由 CECSEL 选择并提供给 **cec_ker_ck** 输入的内核时钟，以及 **rcc_pclk1** 总线接口时钟。

位 26:24 保留，必须保持复位值。

位 23 **I2C3EN**：I2C3 外设时钟使能 (I2C3 Peripheral Clocks Enable)
由软件置 1 和复位。
0：禁止 I2C3 外设时钟（复位后的默认值）
1：使能 I2C3 外设时钟
I2C3 的外设时钟为：由 I2C123SEL 选择并提供给 **i2c_ker_ck** 输入的内核时钟，以及 **rcc_pclk1** 总线接口时钟。

位 22 **I2C2EN**：I2C2 外设时钟使能 (I2C2 Peripheral Clocks Enable)
由软件置 1 和复位。
0：禁止 I2C2 外设时钟（复位后的默认值）
1：使能 I2C2 外设时钟
I2C2 的外设时钟为：由 I2C123SEL 选择并提供给 **i2c_ker_ck** 输入的内核时钟，以及 **rcc_pclk1** 总线接口时钟。

位 21 **I2C1EN**：I2C1 外设时钟使能 (I2C1 Peripheral Clocks Enable)
由软件置 1 和复位。
0：禁止 I2C1 外设时钟（复位后的默认值）
1：使能 I2C1 外设时钟
I2C1 的外设时钟为：由 I2C123SEL 选择并提供给 **i2c_ker_ck** 输入的内核时钟，以及 **rcc_pclk1** 总线接口时钟。

位 20 **UART5EN**：UART5 外设时钟使能 (UART5 Peripheral Clocks Enable)
由软件置 1 和复位。
0：禁止 UART5 外设时钟（复位后的默认值）
1：使能 UART5 外设时钟
UART5 的外设时钟为：由 USART234578SEL 选择并提供给 **uart_ker_ck** 输入的内核时钟，以及 **rcc_pclk1** 总线接口时钟。

位 19 **UART4EN**：UART4 外设时钟使能 (UART4 Peripheral Clocks Enable)
由软件置 1 和复位。
0：禁止 UART4 外设时钟（复位后的默认值）
1：使能 UART4 外设时钟
UART4 的外设时钟为：由 USART234578SEL 选择并提供给 **uart_ker_ck** 输入的内核时钟，以及 **rcc_pclk1** 总线接口时钟。

位 18 **USART3EN**：USART3 外设时钟使能 (USART3 Peripheral Clocks Enable)
由软件置 1 和复位。
0：禁止 USART3 外设时钟（复位后的默认值）
1：使能 USART3 外设时钟
USART3 的外设时钟为：由 USART234578SEL 选择并提供给 **usart_ker_ck** 输入的内核时钟，以及 **rcc_pclk1** 总线接口时钟。

位 17 **USART2EN**：USART2 外设时钟使能 (USART2 Peripheral Clocks Enable)
由软件置 1 和复位。
0：禁止 USART2 外设时钟（复位后的默认值）
1：使能 USART2 外设时钟
USART2 的外设时钟为：由 USART234578SEL 选择并提供给 **usart_ker_ck** 输入的内核时钟，以及 **rcc_pclk1** 总线接口时钟。

位 16 **SPDIFRXEN**：SPDIFRX 外设时钟使能 (SPDIFRX Peripheral Clocks Enable)
由软件置 1 和复位。
0：禁止 SPDIFRX 外设时钟（复位后的默认值）
1：使能 SPDIFRX 外设时钟
SPDIFRX 的外设时钟为：由 SPDIFSEL 选择并提供给 **spdifrx_ker_ck** 输入的内核时钟，以及 **rcc_pclk1** 总线接口时钟。

位 15 **SPI3EN**：SPI3 外设时钟使能 (SPI3 Peripheral Clocks Enable)
由软件置 1 和复位。
0：禁止 SPI3 外设时钟（复位后的默认值）
1：使能 SPI3 外设时钟
SPI3 的外设时钟为：由 I2S123SRC 选择并提供给 **spi_ker_ck** 输入的内核时钟，以及 **rcc_pclk1** 总线接口时钟。

位 14 **SPI2EN**：SPI2 外设时钟使能 (SPI2 Peripheral Clocks Enable)
由软件置 1 和复位。
0：禁止 SPI2 外设时钟（复位后的默认值）
1：使能 SPI2 外设时钟
SPI2 的外设时钟为：由 I2S123SRC 选择并提供给 **spi_ker_ck** 输入的内核时钟，以及 **rcc_pclk1** 总线接口时钟。

位 13:10 保留，必须保持复位值。

位 9 **LPTIM1EN**：LPTIM1 外设时钟使能 (LPTIM1 Peripheral Clocks Enable)
由软件置 1 和复位。
0：禁止 LPTIM1 外设时钟（复位后的默认值）
1：使能 LPTIM1 外设时钟
LPTIM1 的外设时钟为：由 LPTIM1SEL 选择并提供给 **lptim_ker_ck** 输入的内核时钟，以及 **rcc_pclk1** 总线接口时钟。

位 8 **TIM14EN**：TIM14 外设时钟使能 (TIM14 peripheral clock enable)
由软件置 1 和复位。
0：禁止 TIM14 外设时钟（复位后的默认值）
1：使能 TIM14 外设时钟

位 7 **TIM13EN**：TIM13 外设时钟使能 (TIM13 peripheral clock enable)
由软件置 1 和复位。
0：禁止 TIM13 外设时钟（复位后的默认值）
1：使能 TIM13 外设时钟

位 6 **TIM12EN**：TIM12 外设时钟使能 (TIM12 peripheral clock enable)
由软件置 1 和复位。
0：禁止 TIM12 外设时钟（复位后的默认值）
1：使能 TIM12 外设时钟

位 5 **TIM7EN**：TIM7 外设时钟使能 (TIM7 peripheral clock enable)
由软件置 1 和复位。
0：禁止 TIM7 外设时钟（复位后的默认值）
1：使能 TIM7 外设时钟

位 4 **TIM6EN**：TIM6 外设时钟使能 (TIM6 peripheral clock enable)
由软件置 1 和复位。
0：禁止 TIM6 外设时钟（复位后的默认值）
1：使能 TIM6 外设时钟

位 3 **TIM5EN**：TIM5 外设时钟使能 (TIM5 peripheral clock enable)
由软件置 1 和复位。
0：禁止 TIM5 外设时钟（复位后的默认值）
1：使能 TIM5 外设时钟

位 2 **TIM4EN**：TIM4 外设时钟使能 (TIM4 peripheral clock enable)
由软件置 1 和复位。
0：禁止 TIM4 外设时钟（复位后的默认值）
1：使能 TIM4 外设时钟

位 1 **TIM3EN**：TIM3 外设时钟使能 (TIM3 peripheral clock enable)
由软件置 1 和复位。
0：禁止 TIM3 外设时钟（复位后的默认值）
1：使能 TIM3 外设时钟

位 0 **TIM2EN**：TIM2 外设时钟使能 (TIM2 peripheral clock enable)
由软件置 1 和复位。
0：禁止 TIM2 外设时钟（复位后的默认值）
1：使能 TIM2 外设时钟

8.7.44 RCC APB1 时钟寄存器 (RCC_APB1HENR)

RCC APB1 Clock Register

该寄存器可通过两个不同的偏移地址进行访问。

表 62. RCC_APB1ENR 偏移地址和复位值

寄存器名	偏移地址	复位值
RCC_APB1HENR	0x0EC	0x0000 0000
RCC_C1_APB1HENR	0x14C	

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	FDCANEN	Res.	Res.	MDIOSEN	OPAMPEN	Res.	SWPEN	CRSEN	Res.
							rw			rw	rw		rw	rw	

位 31:9 保留，必须保持复位值。

位 8 **FDCANEN**：FDCAN 外设时钟使能 (FDCAN Peripheral Clocks Enable)
由软件置 1 和复位。
0：禁止 FDCAN 外设时钟（复位后的默认值）
1：使能 FDCAN 外设时钟
FDCAN 的外设时钟为：由 FDCANSEL 选择并提供给 **fdcan_ker_ck** 输入的内核时钟，以及 **rcc_pclk1** 总线接口时钟。

位 7:6 保留，必须保持复位值。

位 5 **MDIOSEN**：MDIOS 外设时钟使能 (MDIOS peripheral clock enable)
由软件置 1 和复位。
0：禁止 MDIOS 外设时钟（复位后的默认值）
1：使能 MDIOS 外设时钟

位 4 **OPAMPEN**：OPAMP 外设时钟使能 (OPAMP peripheral clock enable)
由软件置 1 和复位。
0：禁止 OPAMP 外设时钟（复位后的默认值）
1：使能 OPAMP 外设时钟

位 3 保留，必须保持复位值。

位 2 **SWPEN**：SWPMI 外设时钟使能 (SWPMI Peripheral Clocks Enable)
由软件置 1 和复位。
0：禁止 SWPMI 外设时钟（复位后的默认值）
1：使能 SWPMI 外设时钟
SWPMI 的外设时钟为：由 SWPSEL 选择并提供给 **swpmi_ker_ck** 输入的内核时钟，以及 **rcc_pclk1** 总线接口时钟。

位 1 **CRSEN**：时钟恢复系统外设时钟使能 (Clock Recovery System peripheral clock enable)
由软件置 1 和复位。
0：禁止 CRS 外设时钟（复位后的默认值）
1：使能 CRS 外设时钟

位 0 保留，必须保持复位值。

8.7.45 RCC APB2 时钟寄存器 (RCC_APB2ENR)

RCC APB2 Clock Register

该寄存器可通过两个不同的偏移地址进行访问。

表 63. RCC_APB2ENR 偏移地址和复位值

寄存器名	偏移地址	复位值
RCC_APB2ENR	0x0F0	0x0000 0000
RCC_C1_APB2ENR	0x150	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	HRTIMEN	DFSDM1EN	Res.	Res.	Res.	SAI3EN	SAI2EN	SAI1EN	Res.	SPI5EN	Res.	TIM17EN	TIM16EN	TIM15EN
		rw	rw				rw	rw	rw		rw		rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	SPI4EN	SPI1EN	Res.	Res.	Res.	Res.	Res.	Res.	USART6EN	USART1EN	Res.	Res.	TIM8EN	TIM1EN
		rw	rw							rw	rw			rw	rw

位 31:30 保留，必须保持复位值。

位 29 **HRTIMEN**：HRTIM 外设时钟使能 (HRTIM peripheral clock enable)
由软件置 1 和复位。
0：禁止 HRTIM 外设时钟（复位后的默认值）
1：使能 HRTIM 外设时钟

位 28 **DFSDM1EN**：DFSDM1 外设时钟使能 (DFSDM1 Peripheral Clocks Enable)
由软件置 1 和复位。
0：禁止 DFSDM1 外设时钟（复位后的默认值）
1：使能 DFSDM1 外设时钟
DFSDM1 的外设时钟为：由 SAI1SEL 和 DFSDM1SEL 选择并分别提供给 **Aclk** 和 **clk** 输入的内核时钟，以及 **rcc_pclk2** 总线接口时钟。

位 27:25 保留，必须保持复位值。

位 24 **SAI3EN**：SAI3 外设时钟使能 (SAI3 Peripheral Clocks Enable)
由软件置 1 和复位。
0：禁止 SAI3 外设时钟（复位后的默认值）
1：使能 SAI3 外设时钟
SAI3 的外设时钟为：由 SAI23SEL 选择并提供给 **sai_a_ker_ck** 和 **sai_b_ker_ck** 输入的内核时钟，以及 **rcc_pclk2** 总线接口时钟。

位 23 **SAI2EN**：SAI2 外设时钟使能 (SAI2 Peripheral Clocks Enable)
由软件置 1 和复位。
0：禁止 SAI2 外设时钟（复位后的默认值）
1：使能 SAI2 外设时钟
SAI2 的外设时钟为：由 SAI23SEL 选择并提供给 **sai_a_ker_ck** 和 **sai_b_ker_ck** 输入的内核时钟，以及 **rcc_pclk2** 总线接口时钟。

位 22 **SAI1EN**：SAI1 外设时钟使能 (SAI1 Peripheral Clocks Enable)
由软件置 1 和复位。
0：禁止 SAI1 外设时钟（复位后的默认值）
1：使能 SAI1 外设时钟
SAI1 的外设时钟为：由 SAI1SEL 选择并提供给 **sai_a_ker_ck** 和 **sai_b_ker_ck** 输入的内核时钟，以及 **rcc_pclk2** 总线接口时钟。

位 21 保留，必须保持复位值。

位 20 **SPI5EN**：SPI5 外设时钟使能 (SPI5 Peripheral Clocks Enable)
由软件置 1 和复位。
0：禁止 SPI5 外设时钟（复位后的默认值）
1：使能 SPI5 外设时钟
SPI5 的外设时钟为：由 SPI45SEL 选择并提供给 **spi_ker_ck** 输入的内核时钟，以及 **rcc_pclk2** 总线接口时钟。

位 19 保留，必须保持复位值。

位 18 **TIM17EN**：TIM17 外设时钟使能 (TIM17 peripheral clock enable)
由软件置 1 和复位。
0：禁止 TIM17 外设时钟（复位后的默认值）
1：使能 TIM17 外设时钟

位 17 **TIM16EN**：TIM16 外设时钟使能 (TIM16 peripheral clock enable)
由软件置 1 和复位。
0：禁止 TIM16 外设时钟（复位后的默认值）
1：使能 TIM16 外设时钟

位 16 **TIM15EN**：TIM15 外设时钟使能 (TIM15 peripheral clock enable)
由软件置 1 和复位。
0：禁止 TIM15 外设时钟（复位后的默认值）
1：使能 TIM15 外设时钟

位 15:14 保留，必须保持复位值。

位 13 **SPI4EN**：SPI4 外设时钟使能 (SPI4 Peripheral Clocks Enable)
由软件置 1 和复位。
0：禁止 SPI4 外设时钟（复位后的默认值）
1：使能 SPI4 外设时钟
SPI4 的外设时钟为：由 SPI45SEL 选择并提供给 **spi_ker_ck** 输入的内核时钟，以及 **rcc_pclk2** 总线接口时钟。

位 12 **SPI1EN**：SPI1 外设时钟使能 (SPI1 Peripheral Clocks Enable)
由软件置 1 和复位。
0：禁止 SPI1 外设时钟（复位后的默认值）
1：使能 SPI1 外设时钟
SPI1 的外设时钟为：由 I2S123SRC 选择并提供给 **spi_ker_ck** 输入的内核时钟，以及 **rcc_pclk2** 总线接口时钟。

位 11:6 保留，必须保持复位值。

位 5 **USART6EN**：USART6 外设时钟使能 (USART6 Peripheral Clocks Enable)
由软件置 1 和复位。
0：禁止 USART6 外设时钟（复位后的默认值）
1：使能 USART6 外设时钟
USART6 的外设时钟为：由 USART16SEL 选择并提供给 **usart_ker_ck** 输入的内核时钟，以及 **rcc_pclk2** 总线接口时钟。

位 4 **USART1EN**：USART1 外设时钟使能 (USART1 Peripheral Clocks Enable)
由软件置 1 和复位。
0：禁止 USART1 外设时钟（复位后的默认值）
1：使能 USART1 外设时钟
USART1 的外设时钟为：由 USART16SEL 选择并提供给 **usart_ker_ck** 输入的内核时钟，以及 **rcc_pclk2** 总线接口时钟。

位 3:2 保留，必须保持复位值。

位 1 **TIM8EN**：TIM8 外设时钟使能 (TIM8 peripheral clock enable)
由软件置 1 和复位。
0：禁止 TIM8 外设时钟（复位后的默认值）
1：使能 TIM8 外设时钟

位 0 **TIM1EN**：TIM1 外设时钟使能 (TIM1 peripheral clock enable)
由软件置 1 和复位。
0：禁止 TIM1 外设时钟（复位后的默认值）
1：使能 TIM1 外设时钟

8.7.46 RCC APB4 时钟寄存器 (RCC_APB4ENR)

RCC APB4 Clock Register

该寄存器可通过两个不同的偏移地址进行访问。

表 64. RCC_APB4ENR 偏移地址和复位值

寄存器名	偏移地址	复位值
RCC_APB4ENR	0x0F4	0x0001 0000
RCC_C1_APB4ENR	0x154	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SAI4EN	Res.	Res.	Res.	Res.	RTCAPBEN
										rw					rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VREFEN	COMP12EN	Res.	LPTIM5EN	LPTIM4EN	LPTIM3EN	LPTIM2EN	Res.	I2C4EN	Res.	SPI6EN	Res.	LPUART1EN	Res.	SYSCFGEN	Res.
rw	rw		rw	rw	rw	rw		rw		rw		rw		rw	

位 31:26 保留，必须保持复位值。

位 25:22 保留，必须保持复位值。

位 21 **SAI4EN**：SAI4 外设时钟使能 (SAI4 Peripheral Clocks Enable)
由软件置 1 和复位。
0：禁止 SAI4 外设时钟（复位后的默认值）
1：使能 SAI4 外设时钟
SAI4 的外设时钟为：由 SAI4ASEL 和 SAI4BSEL 选择并分别提供给 **sai_a_ker_ck** 和 **sai_b_ker_ck** 输入的内核时钟，以及 **rcc_pclk4** 总线接口时钟。

位 20:17 保留，必须保持复位值。

位 16 **RTCAPBEN**：RTC APB 时钟使能 (RTC APB Clock Enable)
由软件置 1 和复位。
0：禁止 RTC (APB) 的寄存器时钟接口
1：使能 RTC (APB) 的寄存器时钟接口（复位后的默认值）

位 15 **VREFEN**：VREF 外设时钟使能 (VREF peripheral clock enable)
由软件置 1 和复位。
0：禁止 VREF 外设时钟（复位后的默认值）
1：使能 VREF 外设时钟

位 14 **COMP12EN**：COMP1/2 外设时钟使能 (COMP1/2 peripheral clock enable)
由软件置 1 和复位。
0：禁止 COMP1/2 外设时钟（复位后的默认值）
1：使能 COMP1/2 外设时钟

位 13 保留，必须保持复位值。

位 12 **LPTIM5EN**：LPTIM5 外设时钟使能 (LPTIM5 Peripheral Clocks Enable)
由软件置 1 和复位。
0：禁止 LPTIM5 外设时钟（复位后的默认值）
1：使能 LPTIM5 外设时钟
LPTIM5 的外设时钟为：由 LPTIM345SEL 选择并提供给 **lptim_ker_ck** 输入的内核时钟，以及 **rcc_pclk4** 总线接口时钟。

位 11 **LPTIM4EN**：LPTIM4 外设时钟使能 (LPTIM4 Peripheral Clocks Enable)
由软件置 1 和复位。
0：禁止 LPTIM4 外设时钟（复位后的默认值）
1：使能 LPTIM4 外设时钟
LPTIM4 的外设时钟为：由 LPTIM345SEL 选择并提供给 **lptim_ker_ck** 输入的内核时钟，以及 **rcc_pclk4** 总线接口时钟。

位 10 **LPTIM3EN**：LPTIM3 外设时钟使能 (LPTIM3 Peripheral Clocks Enable)
由软件置 1 和复位。
0：禁止 LPTIM3 外设时钟（复位后的默认值）
1：使能 LPTIM3 外设时钟
LPTIM3 的外设时钟为：由 LPTIM345SEL 选择并提供给 **lptim_ker_ck** 输入的内核时钟，以及 **rcc_pclk4** 总线接口时钟。

位 9 **LPTIM2EN**：LPTIM2 外设时钟使能 (LPTIM2 Peripheral Clocks Enable)
由软件置 1 和复位。
0：禁止 LPTIM2 外设时钟（复位后的默认值）
1：使能 LPTIM2 外设时钟
LPTIM2 的外设时钟为：由 LPTIM2SEL 选择并提供给 **lptim_ker_ck** 输入的内核时钟，以及 **rcc_pclk4** 总线接口时钟。

位 8 保留，必须保持复位值。

位 7 **I2C4EN**：I2C4 外设时钟使能 (I2C4 Peripheral Clocks Enable)
由软件置 1 和复位。
0：禁止 I2C4 外设时钟（复位后的默认值）
1：使能 I2C4 外设时钟
I2C4 的外设时钟为：由 I2C4SEL 选择并提供给 **i2c_ker_ck** 输入的内核时钟，以及 **rcc_pclk4** 总线接口时钟。

位 6 保留，必须保持复位值。

位 5 **SPI6EN**：SPI6 外设时钟使能 (SPI6 Peripheral Clocks Enable)
由软件置 1 和复位。
0：禁止 SPI6 外设时钟（复位后的默认值）
1：使能 SPI6 外设时钟
SPI6 的外设时钟为：由 SPI6SEL 选择并提供给 **spi_ker_ck** 输入的内核时钟，以及 **rcc_pclk4** 总线接口时钟。

位 4 保留，必须保持复位值。

位 3 **LPUART1EN**：LPUART1 外设时钟使能 (LPUART1 Peripheral Clocks Enable)
由软件置 1 和复位。
0：禁止 LPUART1 外设时钟（复位后的默认值）
1：使能 LPUART1 外设时钟
LPUART1 的外设时钟为：由 LPUART1SEL 选择并提供给 **lpuart_ker_ck** 输入的内核时钟，以及 **rcc_pclk4** 总线接口时钟。

位 2 保留，必须保持复位值。

位 1 **SYSCFGEN**：SYSCFG 外设时钟使能 (SYSCFG peripheral clock enable)

由软件置 1 和复位。

0：禁止 SYSCFG 外设时钟（复位后的默认值）

1：使能 SYSCFG 外设时钟

位 0 保留，必须保持复位值。

8.7.47 RCC AHB3 睡眠时钟寄存器 (RCC_AHB3LPENR)

RCC AHB3 Sleep Clock Register

该寄存器可通过两个不同的偏移地址进行访问。

表 65. RCC_AHB3LPENR 偏移地址和复位值

寄存器名	偏移地址	复位值
RCC_AHB3LPENR	0x0FC	0xF001 5131
RCC_C1_AHB3LPENR	0x15C	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
AXISRAMLPEN	ITCMLPEN	DTCM2LPEN	DTCM1LPEN	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SDMMC1LPEN
rw	rw	rw	rw												rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	QSPILPEN	Res.	FMCLPEN	Res.	Res.	Res.	FLASHLPEN	Res.	Res.	JPGDECLPEN	DMA2DLPEN	Res.	Res.	Res.	MDMALPEN
	rw		rw				rw			rw	rw				rw

位 31 **AXISRAMLPEN**：CSleep 模式期间的 AXISRAM 模块时钟使能 (AXISRAM Block Clock Enable During CSleep mode)

由软件置 1 和复位。

0：CSleep 模式期间禁止 AXISRAM 接口时钟

1：CSleep 模式期间使能 AXISRAM 接口时钟（复位后的默认值）

位 30 **ITCMLPEN**：CSleep 模式期间的 D1ITCM 模块时钟使能 (D1ITCM Block Clock Enable During CSleep mode)

由软件置 1 和复位。

0：CSleep 模式期间禁止 D1 ITCM 接口时钟

1：CSleep 模式期间使能 D1 ITCM 接口时钟（复位后的默认值）

位 29 **DTCM2LPEN**：CSleep 模式期间的 D1 DTCM2 模块时钟使能 (D1 DTCM2 Block Clock Enable During CSleep mode)

由软件置 1 和复位。

0：CSleep 模式期间禁止 D1 DTCM2 接口时钟

1：CSleep 模式期间使能 D1 DTCM2 接口时钟（复位后的默认值）

位 28 **D1DTCM1LPEN**：CSleep 模式期间的 D1DTCM1 模块时钟使能 (D1DTCM1 Block Clock Enable During CSleep mode)

由软件置 1 和复位。

0：CSleep 模式期间禁止 D1DTCM1 接口时钟

1：CSleep 模式期间使能 D1DTCM1 接口时钟（复位后的默认值）

位 27:17 保留，必须保持复位值。

位 16 **SDMMC1LPEN**：CSleep 模式期间的 SDMMC1 和 SDMMC1 延迟时钟使能 (SDMMC1 and SDMMC1 Delay Clock Enable During CSleep Mode)

由软件置 1 和复位。

0：CSleep 模式期间禁止 SDMMC1 和 SDMMC1 延迟时钟

1：CSleep 模式期间使能 SDMMC1 和 SDMMC1 延迟时钟（复位后的默认值）

位 15 保留，必须保持复位值。

位 14 **QSPILPEN**：CSleep 模式期间的 QUADSPI 和 QUADSPI 延迟时钟使能 (QUADSPI and QUADSPI Delay Clock Enable During CSleep Mode)

由软件置 1 和复位。

0：CSleep 模式期间禁止 QUADSPI 和 QUADSPI 延迟时钟

1：CSleep 模式期间使能 QUADSPI 和 QUADSPI 延迟时钟（复位后的默认值）

位 13 保留，必须保持复位值。

位 12 **FMCLPEN**：CSleep 模式期间的 FMC 外设时钟使能 (FMC Peripheral Clocks Enable During CSleep Mode)

由软件置 1 和复位。

0：CSleep 模式期间禁止 FMC 外设时钟

1：CSleep 模式期间使能 FMC 外设时钟（复位后的默认值）：

FMC 的外设时钟为：由 FMCSEL 选择并提供给 **fmc_ker_ck** 输入的内核时钟，以及 **rcc_hclk3** 总线接口时钟。

位 11:9 保留，必须保持复位值。

位 8 **FLITFLPEN**：CSleep 模式期间的 Flash 接口时钟使能 (Flash interface Clock Enable During CSleep Mode)

由软件置 1 和复位。

0：CSleep 模式期间禁止 Flash 接口时钟

1：CSleep 模式期间使能 Flash 接口时钟（复位后的默认值）

位 7:6 保留，必须保持复位值。

位 5 **JPGDECLPEN**：CSleep 模式期间的 JPGDEC 时钟使能 (JPGDEC Clock Enable During CSleep Mode)

由软件置 1 和复位。

0：CSleep 模式期间禁止 JPGDEC 外设时钟

1：CSleep 模式期间使能 JPGDEC 外设时钟（复位后的默认值）

位 4 **DMA2DLPEN**：CSleep 模式期间的 DMA2D 时钟使能 (DMA2D Clock Enable During CSleep Mode)

由软件置 1 和复位。

0：CSleep 模式期间禁止 DMA2D 外设时钟

1：CSleep 模式期间使能 DMA2D 外设时钟（复位后的默认值）

位 3:1 保留，必须保持复位值。

位 0 **MDMALPEN**：CSleep 模式期间的 MDMA 时钟使能 (MDMA Clock Enable During CSleep Mode)

由软件置 1 和复位。

0：CSleep 模式期间禁止 MDMA 外设时钟

1：CSleep 模式期间使能 MDMA 外设时钟（复位后的默认值）

8.7.48 RCC AHB1 睡眠时钟寄存器 (RCC_AHB1LPENR)

RCC AHB1 Sleep Clock Register

该寄存器可通过两个不同的偏移地址进行访问。

表 66. RCC_AHB1LPENR 偏移地址和复位值

寄存器名	偏移地址	复位值
RCC_AHB1LPENR	0x100	0x1E03 C023
RCC_C1_AHB1LPENR	0x160	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	USB2ULPILPEN	USB2OTGLPEN	USB1ULPILPEN	USB1OTGLPEN	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ETH1RXLPEN	ETH1TXLPEN
			rw	rw	rw	rw								rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ETH1MACLPEN	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ADC12LPEN	Res.	Res.	Res.	DMA2LPEN	DMA1LPEN
rw										rw				rw	rw

位 31:29 保留，必须保持复位值。

位 28 **USB2ULPILPEN**：CSleep 模式期间的 USB_PHY2 时钟使能 (USB_PHY2 clocks enable during CSleep mode)

由软件置 1 和复位。

0：CSleep 模式期间禁止 USB_PHY2 时钟

1：CSleep 模式期间使能 USB_PHY2 时钟（复位后的默认值）

位 27 **USB2OTGLPEN**：CSleep 模式期间的 USB2OTG 外设时钟使能 (USB2OTG peripheral clock enable during CSleep mode)

由软件置 1 和复位。

0：CSleep 模式期间禁止 USB2OTG 外设时钟

1：CSleep 模式期间使能 USB2OTG 外设时钟（复位后的默认值）

USB2OTG 的外设时钟为：由 USBSEL 选择的内核时钟，以及 **rcc_hclk1** 总线接口时钟。

位 26 **USB1ULPILPEN**：CSleep 模式期间的 USB_PHY1 时钟使能 (USB_PHY1 clock enable during CSleep mode)

由软件置 1 和复位。

0：CSleep 模式期间禁止 USB_PHY1 外设时钟

1：CSleep 模式期间使能 USB_PHY1 外设时钟（复位后的默认值）

位 25 **USB1OTGLPEN**：CSleep 模式期间的 USB1OTG 外设时钟使能 (USB1OTG peripheral clock enable during CSleep mode)

由软件置 1 和复位。

0：CSleep 模式期间禁止 USB1OTG 外设时钟

1：CSleep 模式期间使能 USB1OTG 外设时钟（复位后的默认值）

USB1OTG 的外设时钟为：由 USBSEL 选择的内核时钟，以及 **rcc_hclk1** 总线接口时钟。

位 24:18 保留，必须保持复位值。

位 17 **ETH1RXLPEN**：CSleep 模式期间的以太网接收时钟使能 (Ethernet Reception Clock Enable During CSleep Mode)

由软件置 1 和复位。
0：CSleep 模式期间禁止以太网接收时钟
1：CSleep 模式期间使能以太网接收时钟（复位后的默认值）

位 16 **ETH1TXLPEN**：CSleep 模式期间的以太网发送时钟使能 (Ethernet Transmission Clock Enable During CSleep Mode)

由软件置 1 和复位。
0：CSleep 模式期间禁止以太网发送时钟
1：CSleep 模式期间使能以太网发送时钟（复位后的默认值）

位 15 **ETH1MACLPEN**：CSleep 模式期间的以太网 MAC 总线接口时钟使能 (Ethernet MAC bus interface Clock Enable During CSleep Mode)

由软件置 1 和复位。
0：CSleep 模式期间禁止以太网 MAC 总线接口时钟
1：CSleep 模式期间使能以太网 MAC 总线接口时钟（复位后的默认值）

位 14:6 保留，必须保持复位值。

位 5 **ADC12LPEN**：CSleep 模式期间的 ADC1/2 外设时钟使能 (ADC1/2 Peripheral Clocks Enable During CSleep Mode)

由软件置 1 和复位。
0：CSleep 模式期间禁止 ADC1/2 外设时钟
1：CSleep 模式期间使能 ADC1/2 外设时钟（复位后的默认值）
ADC1 和 2 的外设时钟为：由 ADCSEL 选择并提供给 **adc_ker_ck** 输入的内核时钟，以及 **rcc_hclk1** 总线接口时钟。

位 4:2 保留，必须保持复位值。

位 1 **DMA2LPEN**：CSleep 模式期间的 DMA2 时钟使能 (DMA2 Clock Enable During CSleep Mode)

由软件置 1 和复位。
0：CSleep 模式期间禁止 DMA2 时钟
1：CSleep 模式期间使能 DMA2 时钟（复位后的默认值）

位 0 **DMA1LPEN**：CSleep 模式期间的 DMA1 时钟使能 (DMA1 Clock Enable During CSleep Mode)

由软件置 1 和复位。
0：CSleep 模式期间禁止 DMA1 时钟
1：CSleep 模式期间使能 DMA1 时钟（复位后的默认值）

8.7.49 RCC AHB2 睡眠时钟寄存器 (RCC_AHB2LPENR)

RCC AHB2 Sleep Clock Register

该寄存器可通过两个不同的偏移地址进行访问。

表 67. RCC_AHB2LPENR 偏移地址和复位值

寄存器名	偏移地址	复位值
RCC_AHB2LPENR	0x104	0xE000 0271
RCC_C1_AHB2LPENR	0x164	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SRAM3LPEN	SRAM2LPEN	SRAM1LPEN	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
rw	rw	rw													

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	SDMMC2LPEN	Res.	Res.	RNGLPEN	HASHLPEN	CRYPTLPEN	Res.	Res.	Res.	CAMITFLPEN
						rw			rw	rw	rw				rw

位 31 **SRAM3LPEN**：CSleep 模式期间的 SRAM3 时钟使能 (SRAM3 Clock Enable During CSleep Mode)
由软件置 1 和复位。
0：CSleep 模式期间禁止 SRAM3 时钟
1：CSleep 模式期间使能 SRAM3 时钟（复位后的默认值）

位 30 **SRAM2LPEN**：CSleep 模式期间的 SRAM2 时钟使能 (SRAM2 Clock Enable During CSleep Mode)
由软件置 1 和复位。
0：CSleep 模式期间禁止 SRAM2 时钟
1：CSleep 模式期间使能 SRAM2 时钟（复位后的默认值）

位 29 **SRAM1LPEN**：CSleep 模式期间的 SRAM1 时钟使能 (SRAM1 Clock Enable During CSleep Mode)
由软件置 1 和复位。
0：CSleep 模式期间禁止 SRAM1 时钟
1：CSleep 模式期间使能 SRAM1 时钟（复位后的默认值）

位 28:10 保留，必须保持复位值。

位 9 **SDMMC2LPEN**：CSleep 模式期间的 SDMMC2 和 SDMMC2 延迟时钟使能 (SDMMC2 and SDMMC2 Delay Clock Enable During CSleep Mode)
由软件置 1 和复位。
0：CSleep 模式期间禁止 SDMMC2 和 SDMMC2 延迟时钟
1：CSleep 模式期间使能 SDMMC2 和 SDMMC2 延迟时钟（复位后的默认值）

位 8:7 保留，必须保持复位值。

位 6 **RNGLPEN**：CSleep 模式期间的 RNG 外设时钟使能 (RNG peripheral clock enable during CSleep mode)

由软件置 1 和复位。
0：CSleep 模式期间禁止 RNG 外设时钟
1：CSleep 模式期间使能 RNG 外设时钟（复位后的默认值）
RNG 的外设时钟为：由 RNGSEL 选择并提供给 **rng_ker_ck** 输入的内核时钟，以及 **rcc_hclk2** 总线接口时钟。

位 5 **HASHLPEN**：CSleep 模式期间的 HASH 外设时钟使能 (HASH peripheral clock enable during CSleep mode)

由软件置 1 和复位。
0：CSleep 模式期间禁止 HASH 外设时钟
1：CSleep 模式期间使能 HASH 外设时钟（复位后的默认值）

位 4 **CRYPTLPEN**：CSleep 模式期间的 CRYPT 外设时钟使能 (CRYPT peripheral clock enable during CSleep mode)

由软件置 1 和复位。
0：CSleep 模式期间禁止 CRYPT 外设时钟
1：CSleep 模式期间使能 CRYPT 外设时钟（复位后的默认值）

位 3:1 保留，必须保持复位值。

位 0 **DCMILPEN**：CSleep 模式期间的 DCMI 外设时钟使能 (DCMI peripheral clock enable during CSleep mode)

由软件置 1 和复位。
0：CSleep 模式期间禁止 DCMI 外设时钟
1：CSleep 模式期间使能 DCMI 外设时钟（复位后的默认值）

8.7.50 RCC AHB4 睡眠时钟寄存器 (RCC_AHB4LPENR)

RCC AHB4 Sleep Clock Register

该寄存器可通过两个不同的偏移地址进行访问。

表 68. RCC_AHB4LPENR 偏移地址和复位值

寄存器名	偏移地址	复位值
RCC_AHB4LPENR	0x108	0x3128 07FF
RCC_C1_AHB4LPENR	0x168	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	SRAM4LPEN	BKPRAMLPEN	Res.	Res.	Res.	ADC3LPEN	Res.	Res.	BDMALPEN	Res.	CRCLPEN	Res.	Res.	Res.
		rw	rw				rw			rw		rw			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	GPIOKLPEN	GPIOJLPEN	GPIOILPEN	GPIOHLPEN	GPIOGLPEN	GPIOFLPEN	GPIOELPEN	GPIODLPEN	GPIOCLPEN	GPIOBLPEN	GPIOALPEN
					rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:30 保留，必须保持复位值。

位 29 **SRAM4LPEN**：CSleep 模式期间的 SRAM4 时钟使能 (SRAM4 Clock Enable During CSleep Mode)
由软件置 1 和复位。
0：CSleep 模式期间禁止 SRAM4 时钟
1：CSleep 模式期间使能 SRAM4 时钟（复位后的默认值）

位 28 **BKPRAMLPEN**：CSleep 模式期间的备份 RAM 时钟使能 (Backup RAM Clock Enable During CSleep Mode)
由软件置 1 和复位。
0：CSleep 模式期间禁止备份 RAM 时钟
1：CSleep 模式期间使能备份 RAM 时钟（复位后的默认值）

位 27:25 保留，必须保持复位值。

位 24 **ADC3LPEN**：CSleep 模式期间的 ADC3 外设时钟使能 (ADC3 Peripheral Clocks Enable During CSleep Mode)
由软件置 1 和复位。
0：CSleep 模式期间禁止 ADC3 外设时钟
1：CSleep 模式期间使能 ADC3 外设时钟（复位后的默认值）
ADC3 的外设时钟为：由 ADCSEL 选择并提供给 adc_ker_ck 输入的内核时钟，以及 **rcc_hclk4** 总线接口时钟。

位 23:22 保留，必须保持复位值。

位 21 **BDMALPEN**：CSleep 模式期间的 BDMA 时钟使能 (BDMA Clock Enable During CSleep Mode)
由软件置 1 和复位。
0：CSleep 模式期间禁止 BDMA 时钟
1：CSleep 模式期间使能 BDMA 时钟（复位后的默认值）

位 20 保留，必须保持复位值。

位 19 **CRCLPEN**：CSleep 模式期间的 CRC 外设时钟使能 (CRC peripheral clock enable during CSleep mode)

由软件置 1 和复位。
0：CSleep 模式期间禁止 CRC 外设时钟
1：CSleep 模式期间使能 CRC 外设时钟（复位后的默认值）

位 18:11 保留，必须保持复位值。

位 10 **GPIOKLPEN**：CSleep 模式期间的 GPIOK 外设时钟使能 (GPIOK peripheral clock enable during CSleep mode)

由软件置 1 和复位。
0：CSleep 模式期间禁止 GPIOK 外设时钟
1：CSleep 模式期间使能 GPIOK 外设时钟（复位后的默认值）

位 9 **GPIOJLPEN**：CSleep 模式期间的 GPIOJ 外设时钟使能 (GPIOJ peripheral clock enable during CSleep mode)

由软件置 1 和复位。
0：CSleep 模式期间禁止 GPIOJ 外设时钟
1：CSleep 模式期间使能 GPIOJ 外设时钟（复位后的默认值）

位 8 **GPIOILPEN**：CSleep 模式期间的 GPIOI 外设时钟使能 (GPIOI peripheral clock enable during CSleep mode)

由软件置 1 和复位。
0：CSleep 模式期间禁止 GPIOI 外设时钟
1：CSleep 模式期间使能 GPIOI 外设时钟（复位后的默认值）

位 7 **GPIOHLPEN**：CSleep 模式期间的 GPIOH 外设时钟使能 (GPIOH peripheral clock enable during CSleep mode)

由软件置 1 和复位。
0：CSleep 模式期间禁止 GPIOH 外设时钟
1：CSleep 模式期间使能 GPIOH 外设时钟（复位后的默认值）

位 6 **GPIOGLPEN**：CSleep 模式期间的 GPIOG 外设时钟使能 (GPIOG peripheral clock enable during CSleep mode)

由软件置 1 和复位。
0：CSleep 模式期间禁止 GPIOG 外设时钟
1：CSleep 模式期间使能 GPIOG 外设时钟（复位后的默认值）

位 5 **GPIOFLPEN**：CSleep 模式期间的 GPIOF 外设时钟使能 (GPIOF peripheral clock enable during CSleep mode)

由软件置 1 和复位。
0：CSleep 模式期间禁止 GPIOF 外设时钟
1：CSleep 模式期间使能 GPIOF 外设时钟（复位后的默认值）

位 4 **GPIOELPEN**：CSleep 模式期间的 GPIOE 外设时钟使能 (GPIOE peripheral clock enable during CSleep mode)

由软件置 1 和复位。
0：CSleep 模式期间禁止 GPIOE 外设时钟
1：CSleep 模式期间使能 GPIOE 外设时钟（复位后的默认值）

位 3 **GPIODLPEN**：CSleep 模式期间的 GPIOD 外设时钟使能 (GPIOD peripheral clock enable during CSleep mode)

由软件置 1 和复位。
0：CSleep 模式期间禁止 GPIOD 外设时钟
1：CSleep 模式期间使能 GPIOD 外设时钟（复位后的默认值）

位 2 **GPIOCLPEN**：CSleep 模式期间的 GPIOC 外设时钟使能 (GPIOC peripheral clock enable during CSleep mode)

由软件置 1 和复位。

0：CSleep 模式期间禁止 GPIOC 外设时钟

1：CSleep 模式期间使能 GPIOC 外设时钟（复位后的默认值）

位 1 **GPIOBLPEN**：CSleep 模式期间的 GPIOB 外设时钟使能 (GPIOB peripheral clock enable during CSleep mode)

由软件置 1 和复位。

0：CSleep 模式期间禁止 GPIOB 外设时钟

1：CSleep 模式期间使能 GPIOB 外设时钟（复位后的默认值）

位 0 **GPIOALPEN**：CSleep 模式期间的 GPIOA 外设时钟使能 (GPIOA peripheral clock enable during CSleep mode)

由软件置 1 和复位。

0：CSleep 模式期间禁止 GPIOA 外设时钟

1：CSleep 模式期间使能 GPIOA 外设时钟（复位后的默认值）

8.7.51 RCC APB3 睡眠时钟寄存器 (RCC_APB3LPENR)

RCC APB3 Sleep Clock Register

该寄存器可通过两个不同的偏移地址进行访问。

表 69. RCC_APB3LPENR 偏移地址和复位值

寄存器名	偏移地址	复位值
RCC_APB3LPENR	0x10C	0x0000 0058
RCC_C1_APB3LPENR	0x16C	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WWDG1LPEN	Res.	Res.	LTDCLPEN	Res.	Res.	Res.
									rw			rw			

位 31:7 保留，必须保持复位值。

位 6 **WWDG1LPEN**：CSleep 模式期间的 WWDG1 时钟使能 (WWDG1 Clock Enable During CSleep Mode)

由软件置 1 和复位。

0：CSleep 模式期间禁止 WWDG1 时钟

1：CSleep 模式期间使能 WWDG1 时钟（复位后的默认值）

位 5:4 保留，必须保持复位值。

位 3 **LTDCLPEN**：CSleep 模式期间的 LTDC 外设时钟使能 (LTDC peripheral clock enable during CSleep mode)

向 LTDC 模块提供像素时钟 (**ltdc_ker_ck**)。
由软件置 1 和复位。
0：CSleep 模式期间禁止 LTDC 时钟
1：CSleep 模式期间向 LTDC 提供 LTDC 时钟（复位后的默认值）

位 2:0 保留，必须保持复位值。

8.7.52 RCC APB1 低位睡眠时钟寄存器 (RCC_APB1LLPENR)

RCC APB1 Low Sleep Clock Register

该寄存器可通过两个不同的偏移地址进行访问。

表 70. RCC_APB1LLPENR 偏移地址和复位值

寄存器名	偏移地址	复位值
RCC_APB1LLPENR	0x110	0xE8FF CBFF
RCC_C1_APB1LLPENR	0x170	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
UART8LPEN	UART7LPEN	DAC12LPEN	Res.	CECLPEN	Res.	Res.	Res.	I2C3LPEN	I2C2LPEN	I2C1LPEN	UART5LPEN	UART4LPEN	USART3LPEN	USART2LPEN	SPDIFRXLPEN
rw	rw	rw		rw				rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SPI3LPEN	SPI2LPEN	Res.	Res.	Res.	Res.	LPTIM1LPEN	TIM14LPEN	TIM13LPEN	TIM12LPEN	TIM7LPEN	TIM6LPEN	TIM5LPEN	TIM4LPEN	TIM3LPEN	TIM2LPEN
rw	rw					rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31 **UART8LPEN**：CSleep 模式期间的 UART8 外设时钟使能 (UART8 Peripheral Clocks Enable During CSleep Mode)

由软件置 1 和复位。
0：CSleep 模式期间禁止 UART8 外设时钟
1：CSleep 模式期间使能 UART8 外设时钟（复位后的默认值）：
USART8 的外设时钟为：由 USART234578SEL 选择并提供给 **usart_ker_ck** 输入的内核时钟，以及 **rcc_pclk1** 总线接口时钟。

位 30 **UART7LPEN**：CSleep 模式期间的 UART7 外设时钟使能 (UART7 Peripheral Clocks Enable During CSleep Mode)

由软件置 1 和复位。
0：CSleep 模式期间禁止 UART7 外设时钟
1：CSleep 模式期间使能 UART7 外设时钟（复位后的默认值）：
UART7 的外设时钟为：由 USART234578SEL 选择并提供给 **usart_ker_ck** 输入的内核时钟，以及 **rcc_pclk1** 总线接口时钟。

位 29 **DAC12LPEN**：CSleep 模式期间的 DAC1/2 外设时钟使能 (DAC1/2 peripheral clock enable during CSleep mode)

由软件置 1 和复位。

0：CSleep 模式期间禁止 DAC1/2 外设时钟

1：CSleep 模式期间使能 DAC1/2 外设时钟（复位后的默认值）

位 28 保留，必须保持复位值。

位 27 **CECLPEN**：CSleep 模式期间的 HDMI-CEC 外设时钟使能 (HDMI-CEC Peripheral Clocks Enable During CSleep Mode)

由软件置 1 和复位。

0：CSleep 模式期间禁止 HDMI-CEC 外设时钟

1：CSleep 模式期间使能 HDMI-CEC 外设时钟（复位后的默认值）

HDMI-CEC 的外设时钟为：由 CECSEL 选择并提供给 **cec_ker_ck** 输入的内核时钟，以及 **rcc_pclk1** 总线接口时钟。

位 26:24 保留，必须保持复位值。

位 23 **I2C3LPEN**：CSleep 模式期间的 I2C3 外设时钟使能 (I2C3 Peripheral Clocks Enable During CSleep Mode)

由软件置 1 和复位。

0：CSleep 模式期间禁止 I2C3 外设时钟

1：CSleep 模式期间使能 I2C3 外设时钟（复位后的默认值）：

I2C3 的外设时钟为：由 I2C123SEL 选择并提供给 **i2c_ker_ck** 输入的内核时钟，以及 **rcc_pclk1** 总线接口时钟。

位 22 **I2C2LPEN**：CSleep 模式期间的 I2C2 外设时钟使能 (I2C2 Peripheral Clocks Enable During CSleep Mode)

由软件置 1 和复位。

0：CSleep 模式期间禁止 I2C2 外设时钟

1：CSleep 模式期间使能 I2C2 外设时钟（复位后的默认值）：

I2C2 的外设时钟为：由 I2C123SEL 选择并提供给 **i2c_ker_ck** 输入的内核时钟，以及 **rcc_pclk1** 总线接口时钟。

位 21 **I2C1LPEN**：CSleep 模式期间的 I2C1 外设时钟使能 (I2C1 Peripheral Clocks Enable During CSleep Mode)

由软件置 1 和复位。

0：CSleep 模式期间禁止 I2C1 外设时钟

1：CSleep 模式期间使能 I2C1 外设时钟（复位后的默认值）：

I2C1 的外设时钟为：由 I2C123SEL 选择并提供给 **i2c_ker_ck** 输入的内核时钟，以及 **rcc_pclk1** 总线接口时钟。

位 20 **UART5LPEN**：CSleep 模式期间的 UART5 外设时钟使能 (UART5 Peripheral Clocks Enable During CSleep Mode)

由软件置 1 和复位。

0：CSleep 模式期间禁止 UART5 外设时钟

1：CSleep 模式期间使能 UART5 外设时钟（复位后的默认值）

UART5 的外设时钟为：由 USART234578SEL 选择并提供给 **uart_ker_ck** 输入的内核时钟，以及 **rcc_pclk1** 总线接口时钟。

位 19 **UART4LPEN**：CSleep 模式期间的 UART4 外设时钟使能 (UART4 Peripheral Clocks Enable During CSleep Mode)

由软件置 1 和复位。

0：CSleep 模式期间禁止 UART4 外设时钟

1：CSleep 模式期间使能 UART4 外设时钟（复位后的默认值）

UART4 的外设时钟为：由 USART234578SEL 选择并提供给 **uart_ker_ck** 输入的内核时钟，以及 **rcc_pclk1** 总线接口时钟。

位 18 **USART3LPEN**：CSleep 模式期间的 USART3 外设时钟使能 (USART3 Peripheral Clocks Enable During CSleep Mode)

由软件置 1 和复位。

0：CSleep 模式期间禁止 USART3 外设时钟

1：CSleep 模式期间使能 USART3 外设时钟（复位后的默认值）：

USART3 的外设时钟为：由 USART234578SEL 选择并提供给 **usart_ker_ck** 输入的内核时钟，以及 **rcc_pclk1** 总线接口时钟。

位 17 **USART2LPEN**：CSleep 模式期间的 USART2 外设时钟使能 (USART2 Peripheral Clocks Enable During CSleep Mode)

由软件置 1 和复位。

0：CSleep 模式期间禁止 USART2 外设时钟

1：CSleep 模式期间使能 USART2 外设时钟（复位后的默认值）

USART2 的外设时钟为：由 USART234578SEL 选择并提供给 **usart_ker_ck** 输入的内核时钟，以及 **rcc_pclk1** 总线接口时钟。

位 16 **SPDIFRXLPEN**：CSleep 模式期间的 SPDIFRX 外设时钟使能 (SPDIFRX Peripheral Clocks Enable During CSleep Mode)

由软件置 1 和复位。

0：CSleep 模式期间禁止 SPDIFRX 外设时钟

1：CSleep 模式期间使能 SPDIFRX 外设时钟（复位后的默认值）

SPDIFRX 的外设时钟为：由 SPDIFSEL 选择并提供给 **spdifrx_ker_ck** 输入的内核时钟，以及 **rcc_pclk1** 总线接口时钟。

位 15 **SPI3LPEN**：CSleep 模式期间的 SPI3 外设时钟使能 (SPI3 Peripheral Clocks Enable During CSleep Mode)

由软件置 1 和复位。

0：CSleep 模式期间禁止 SPI3 外设时钟

1：CSleep 模式期间使能 SPI3 外设时钟（复位后的默认值）

SPI3 的外设时钟为：由 I2S123SRC 选择并提供给 **spi_ker_ck** 输入的内核时钟，以及 **rcc_pclk1** 总线接口时钟。

位 14 **SPI2LPEN**：CSleep 模式期间的 SPI2 外设时钟使能 (SPI2 Peripheral Clocks Enable During CSleep Mode)

由软件置 1 和复位。

0：CSleep 模式期间禁止 SPI2 外设时钟

1：CSleep 模式期间使能 SPI2 外设时钟（复位后的默认值）

SPI2 的外设时钟为：由 I2S123SRC 选择并提供给 **spi_ker_ck** 输入的内核时钟，以及 **rcc_pclk1** 总线接口时钟。

位 13:10 保留，必须保持复位值。

位 9 **LPTIM1LPEN**：CSleep 模式期间的 LPTIM1 外设时钟使能 (LPTIM1 Peripheral Clocks Enable During CSleep Mode)

由软件置 1 和复位。

0：CSleep 模式期间禁止 LPTIM1 外设时钟

1：CSleep 模式期间使能 LPTIM1 外设时钟（复位后的默认值）

LPTIM1 的外设时钟为：由 LPTIM1SEL 选择并提供给 **lptim_ker_ck** 输入的内核时钟，以及 **rcc_pclk1** 总线接口时钟。

位 8 **TIM14LPEN**：CSleep 模式期间的 TIM14 外设时钟使能 (TIM14 peripheral clock enable during CSleep mode)

由软件置 1 和复位。

0：CSleep 模式期间禁止 TIM14 外设时钟

1：CSleep 模式期间使能 TIM14 外设时钟（复位后的默认值）

位 7 **TIM13LPEN**：CSleep 模式期间的 TIM13 外设时钟使能 (TIM13 peripheral clock enable during CSleep mode)

由软件置 1 和复位。
0：CSleep 模式期间禁止 TIM13 外设时钟
1：CSleep 模式期间使能 TIM13 外设时钟（复位后的默认值）

位 6 **TIM12LPEN**：CSleep 模式期间的 TIM12 外设时钟使能 (TIM12 peripheral clock enable during CSleep mode)

由软件置 1 和复位。
0：CSleep 模式期间禁止 TIM12 外设时钟
1：CSleep 模式期间使能 TIM12 外设时钟（复位后的默认值）

位 5 **TIM7LPEN**：CSleep 模式期间的 TIM7 外设时钟使能 (TIM7 peripheral clock enable during CSleep mode)

由软件置 1 和复位。
0：CSleep 模式期间禁止 TIM7 外设时钟
1：CSleep 模式期间使能 TIM7 外设时钟（复位后的默认值）

位 4 **TIM6LPEN**：CSleep 模式期间的 TIM6 外设时钟使能 (TIM6 peripheral clock enable during CSleep mode)

由软件置 1 和复位。
0：CSleep 模式期间禁止 TIM6 外设时钟
1：CSleep 模式期间使能 TIM6 外设时钟（复位后的默认值）

位 3 **TIM5LPEN**：CSleep 模式期间的 TIM5 外设时钟使能 (TIM5 peripheral clock enable during CSleep mode)

由软件置 1 和复位。
0：CSleep 模式期间禁止 TIM5 外设时钟
1：CSleep 模式期间使能 TIM5 外设时钟（复位后的默认值）

位 2 **TIM4LPEN**：CSleep 模式期间的 TIM4 外设时钟使能 (TIM4 peripheral clock enable during CSleep mode)

由软件置 1 和复位。
0：CSleep 模式期间禁止 TIM4 外设时钟
1：CSleep 模式期间使能 TIM4 外设时钟（复位后的默认值）

位 1 **TIM3LPEN**：CSleep 模式期间的 TIM3 外设时钟使能 (TIM3 peripheral clock enable during CSleep mode)

由软件置 1 和复位。
0：CSleep 模式期间禁止 TIM3 外设时钟
1：CSleep 模式期间使能 TIM3 外设时钟（复位后的默认值）

位 0 **TIM2LPEN**：CSleep 模式期间的 TIM2 外设时钟使能 (TIM2 peripheral clock enable during CSleep mode)

由软件置 1 和复位。
0：CSleep 模式期间禁止 TIM2 外设时钟
1：CSleep 模式期间使能 TIM2 外设时钟（复位后的默认值）

8.7.53 RCC APB1 高位睡眠时钟寄存器 (RCC_APB1HLPENR)

RCC APB1 High Sleep Clock Register

该寄存器可通过两个不同的偏移地址进行访问。

表 71. RCC_APB1HLPENR 偏移地址和复位值

寄存器名	偏移地址	复位值
RCC_APB1HLPENR	0x114	0x0000 0136
RCC_C1_APB1HLPENR	0x174	

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	FDCANLPEN	Res.	Res.	MDIOSLPEN	OPAMPLPEN	Res.	SWPLPEN	CRSLPEN	Res.
							rw			rw	rw		rw	rw	

位 31:9 保留，必须保持复位值。

位 8 **FDCANLPEN**：CSleep 模式期间的 FDCAN 外设时钟使能 (FDCAN Peripheral Clocks Enable During CSleep Mode)

由软件置 1 和复位。

0：CSleep 模式期间禁止 FDCAN 外设时钟

1：CSleep 模式期间使能 FDCAN 外设时钟（复位后的默认值）

FDCAN 的外设时钟为：由 FDCANSEL 选择并提供给 **fdcan_ker_ck** 输入的内核时钟，以及 **rcc_pclk1** 总线接口时钟。

位 7:6 保留，必须保持复位值。

位 5 **MDIOSLPEN**：CSleep 模式期间的 MDIOS 外设时钟使能 (MDIOS peripheral clock enable during CSleep mode)

由软件置 1 和复位。

0：CSleep 模式期间禁止 MDIOS 外设时钟

1：CSleep 模式期间使能 MDIOS 外设时钟（复位后的默认值）

位 4 **OPAMPLPEN**：CSleep 模式期间的 OPAMP 外设时钟使能 (OPAMP peripheral clock enable during CSleep mode)

由软件置 1 和复位。

0：CSleep 模式期间禁止 OPAMP 外设时钟

1：CSleep 模式期间使能 OPAMP 外设时钟（复位后的默认值）

位 3 保留，必须保持复位值。

位 2 **SWPLPEN**：CSleep 模式期间的 SWPMI 外设时钟使能 (SWPMI Peripheral Clocks Enable During CSleep Mode)

由软件置 1 和复位。

0：CSleep 模式期间禁止 SWPMI 外设时钟

1：CSleep 模式期间使能 SWPMI 外设时钟（复位后的默认值）

SWPMI 的外设时钟为：由 SWPSEL 选择并提供给 **swpmi_ker_ck** 输入的内核时钟，以及 **rcc_pclk1** 总线接口时钟。

位 1 **CRSLPEN**：CSleep 模式期间的时钟恢复系统外设时钟使能 (Clock Recovery System peripheral clock enable during CSleep mode)

由软件置 1 和复位。

0：CSleep 模式期间禁止 CRS 外设时钟

1：CSleep 模式期间使能 CRS 外设时钟（复位后的默认值）

位 0 保留，必须保持复位值。

8.7.54 RCC APB2 睡眠时钟寄存器 (RCC_APB2LPENR)

RCC APB2 Sleep Clock Register

该寄存器可通过两个不同的偏移地址进行访问。

表 72. RCC_APB2LPENR 偏移地址和复位值

寄存器名	偏移地址	复位值
RCC_APB2LPENR	0x118	0x31D7 3033
RCC_C1_APB2LPENR	0x178	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	HRTIMLPEN	DFSDM1LPEN	Res.	Res.	Res.	SAI3LPEN	SAI2LPEN	SAI1LPEN	Res.	SPI5LPEN	Res.	TIM17LPEN	TIM16LPEN	TIM15LPEN
		rw	rw				rw	rw	rw		rw		rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	SPI4LPEN	SPI1LPEN	Res.	Res.	Res.	Res.	Res.	Res.	USART6LPEN	USART1LPEN	Res.	Res.	TIM8LPEN	TIM1LPEN
		rw	rw							rw	rw			rw	rw

位 31:30 保留，必须保持复位值。

位 29 **HRTIMLPEN**：CSleep 模式期间的 HRTIM 外设时钟使能 (HRTIM peripheral clock enable during CSleep mode)

由软件置 1 和复位。

0：CSleep 模式期间禁止 HRTIM 外设时钟

1：CSleep 模式期间使能 HRTIM 外设时钟（复位后的默认值）

位 28 **DFSDM1LPEN**：CSleep 模式期间的 DFSDM1 外设时钟使能 (DFSDM1 Peripheral Clocks Enable During CSleep Mode)

由软件置 1 和复位。

0：CSleep 模式期间禁止 DFSDM1 外设时钟

1：CSleep 模式期间使能 DFSDM1 外设时钟（复位后的默认值）

DFSDM1 的外设时钟为：由 SAI1SEL 和 DFSDM1SEL 选择并分别提供给 **Aclk** 和 **clk** 输入的内核时钟，以及 **rcc_pclk2** 总线接口时钟。

位 27:25 保留，必须保持复位值。

位 24 **SAI3LPEN**：CSleep 模式期间的 SAI3 外设时钟使能 (SAI3 Peripheral Clocks Enable During CSleep Mode)

由软件置 1 和复位。

0：CSleep 模式期间禁止 SAI3 外设时钟

1：CSleep 模式期间使能 SAI3 外设时钟（复位后的默认值）

SAI3 的外设时钟为：由 SAI23SEL 选择并提供给 **sai_a_ker_ck** 和 **sai_b_ker_ck** 输入的内核时钟，以及 **rcc_pclk2** 总线接口时钟。

位 23 **SAI2LPEN**：CSleep 模式期间的 SAI2 外设时钟使能 (SAI2 Peripheral Clocks Enable During CSleep Mode)

由软件置 1 和复位。

0：CSleep 模式期间禁止 SAI2 外设时钟

1：CSleep 模式期间使能 SAI2 外设时钟（复位后的默认值）

SAI2 的外设时钟为：由 SAI23SEL 选择并提供给 **sai_a_ker_ck** 和 **sai_b_ker_ck** 输入的内核时钟，以及 **rcc_pclk2** 总线接口时钟。

位 22 **SAI1LPEN**：CSleep 模式期间的 SAI1 外设时钟使能 (SAI1 Peripheral Clocks Enable During CSleep Mode)

由软件置 1 和复位。

0：CSleep 模式期间禁止 SAI1 外设时钟

1：CSleep 模式期间使能 SAI1 外设时钟（复位后的默认值）

SAI1 的外设时钟为：由 SAI1SEL 选择并提供给 **sai_a_ker_ck** 和 **sai_b_ker_ck** 输入的内核时钟，以及 **rcc_pclk2** 总线接口时钟。

位 21 保留，必须保持复位值。

位 20 **SPI5LPEN**：CSleep 模式期间的 SPI5 外设时钟使能 (SPI5 Peripheral Clocks Enable During CSleep Mode)

由软件置 1 和复位。

0：CSleep 模式期间禁止 SPI5 外设时钟

1：CSleep 模式期间使能 SPI5 外设时钟（复位后的默认值）

SPI5 的外设时钟为：由 SPI45SEL 选择并提供给 **spi_ker_ck** 输入的内核时钟，以及 **rcc_pclk2** 总线接口时钟。

位 19 保留，必须保持复位值。

位 18 **TIM17LPEN**：CSleep 模式期间的 TIM17 外设时钟使能 (TIM17 peripheral clock enable during CSleep mode)

由软件置 1 和复位。

0：CSleep 模式期间禁止 TIM17 外设时钟

1：CSleep 模式期间使能 TIM17 外设时钟（复位后的默认值）

位 17 **TIM16LPEN**：CSleep 模式期间的 TIM16 外设时钟使能 (TIM16 peripheral clock enable during CSleep mode)

由软件置 1 和复位。

0：CSleep 模式期间禁止 TIM16 外设时钟

1：CSleep 模式期间使能 TIM16 外设时钟（复位后的默认值）

位 16 **TIM15LPEN**：CSleep 模式期间的 TIM15 外设时钟使能 (TIM15 peripheral clock enable during CSleep mode)

由软件置 1 和复位。
0：CSleep 模式期间禁止 TIM15 外设时钟
1：CSleep 模式期间使能 TIM15 外设时钟（复位后的默认值）

位 15:14 保留，必须保持复位值。

位 13 **SPI4LPEN**：CSleep 模式期间的 SPI4 外设时钟使能 (SPI4 Peripheral Clocks Enable During CSleep Mode)

由软件置 1 和复位。
0：CSleep 模式期间禁止 SPI4 外设时钟
1：CSleep 模式期间使能 SPI4 外设时钟（复位后的默认值）
SPI4 的外设时钟为：由 SPI45SEL 选择并提供给 **spi_ker_ck** 输入的内核时钟，以及 **rcc_pclk2** 总线接口时钟。

位 12 **SPI1LPEN**：CSleep 模式期间的 SPI1 外设时钟使能 (SPI1 Peripheral Clocks Enable During CSleep Mode)

由软件置 1 和复位。
0：CSleep 模式期间禁止 SPI1 外设时钟
1：CSleep 模式期间使能 SPI1 外设时钟（复位后的默认值）
SPI1 的外设时钟为：由 I2S123SRC 选择并提供给 **spi_ker_ck** 输入的内核时钟，以及 **rcc_pclk2** 总线接口时钟。

位 11:6 保留，必须保持复位值。

位 5 **USART6LPEN**：CSleep 模式期间的 USART6 外设时钟使能 (USART6 Peripheral Clocks Enable During CSleep Mode)

由软件置 1 和复位。
0：CSleep 模式期间禁止 USART6 外设时钟
1：CSleep 模式期间使能 USART6 外设时钟（复位后的默认值）
USART6 的外设时钟为：由 USART16SEL 选择并提供给 **usart_ker_ck** 输入的内核时钟，以及 rcc_pclk2 总线接口时钟。

位 4 **USART1LPEN**：CSleep 模式期间的 USART1 外设时钟使能 (USART1 Peripheral Clocks Enable During CSleep Mode)

由软件置 1 和复位。
0：CSleep 模式期间禁止 USART1 外设时钟
1：CSleep 模式期间使能 USART1 外设时钟（复位后的默认值）
USART1 的外设时钟为：由 USART16SEL 选择并提供给 **usart_ker_ck** 输入的内核时钟，以及 **rcc_pclk2** 总线接口时钟。

位 3:2 保留，必须保持复位值。

位 1 **TIM8LPEN**：CSleep 模式期间的 TIM8 外设时钟使能 (TIM8 peripheral clock enable during CSleep mode)

由软件置 1 和复位。
0：CSleep 模式期间禁止 TIM8 外设时钟
1：CSleep 模式期间使能 TIM8 外设时钟（复位后的默认值）

位 0 **TIM1LPEN**：CSleep 模式期间的 TIM1 外设时钟使能 (TIM1 peripheral clock enable during CSleep mode)

由软件置 1 和复位。
0：CSleep 模式期间禁止 TIM1 外设时钟
1：CSleep 模式期间使能 TIM1 外设时钟（复位后的默认值）

8.7.55 RCC APB4 睡眠时钟寄存器 (RCC_APB4LPENR)

RCC APB4 Sleep Clock Register

该寄存器可通过两个不同的偏移地址进行访问。

表 73. RCC_APB4LPENR 偏移地址和复位值

寄存器名	偏移地址	复位值
RCC_APB4LPENR	0x11C	0x0421 DEAA
RCC_C1_APB4LPENR	0x17C	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SAI4LPEN	Res.	Res.	Res.	Res.	RTCAPBLPEN
										rw					rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VREFLPEN	COMP12LPEN	Res.	LPTIM5LPEN	LPTIM4LPEN	LPTIM3LPEN	LPTIM2LPEN	Res.	I2C4LPEN	Res.	SPI6LPEN	Res.	LPUART1LPEN	Res.	SYSCFGLPEN	Res.
rw	rw		rw	rw	rw	rw		rw		rw		rw		rw	

位 31:26 保留，必须保持复位值。

位 25:22 保留，必须保持复位值。

位 21 **SAI4LPEN**：CSleep 模式期间的 SAI4 外设时钟使能 (SAI4 Peripheral Clocks Enable During CSleep Mode)

由软件置 1 和复位。

0：CSleep 模式期间禁止 SAI4 外设时钟

1：CSleep 模式期间使能 SAI4 外设时钟（复位后的默认值）

SAI4 的外设时钟为：由 SAI4ASEL 和 SAI4BSEL 选择并分别提供给 **sai_a_ker_ck** 和 **sai_b_ker_ck** 输入的内核时钟，以及 **rcc_pclk4** 总线接口时钟。

位 20:17 保留，必须保持复位值。

位 16 **RTCAPBLPEN**：CSleep 模式期间的 RTC APB 时钟使能 (RTC APB Clock Enable During CSleep Mode)

由软件置 1 和复位。

0：CSleep 模式期间禁止 RTC (APB) 的寄存器时钟接口

1：CSleep 模式期间使能 RTC (APB) 的寄存器时钟接口（复位后的默认值）

位 15 **VREFLPEN**：CSleep 模式期间的 VREF 外设时钟使能 (VREF peripheral clock enable during CSleep mode)

由软件置 1 和复位。

0：CSleep 模式期间禁止 VREF 外设时钟

1：CSleep 模式期间使能 VREF 外设时钟（复位后的默认值）

位 14 **COMP12LPEN**：CSleep 模式期间的 COMP1/2 外设时钟使能 (COMP1/2 peripheral clock enable during CSleep mode)

由软件置 1 和复位。

0：CSleep 模式期间禁止 COMP1/2 外设时钟

1：CSleep 模式期间使能 COMP1/2 外设时钟（复位后的默认值）

位 13 保留，必须保持复位值。

位 12 **LPTIM5LPEN**：CSleep 模式期间的 LPTIM5 外设时钟使能 (LPTIM5 Peripheral Clocks Enable During CSleep Mode)

由软件置 1 和复位。

0：CSleep 模式期间禁止 LPTIM5 外设时钟

1：CSleep 模式期间使能 LPTIM5 外设时钟（复位后的默认值）

LPTIM5 的外设时钟为：由 LPTIM345SEL 选择并提供给 **lptim_ker_ck** 输入的内核时钟，以及 **rcc_pclk4** 总线接口时钟。

位 11 **LPTIM4LPEN**：CSleep 模式期间的 LPTIM4 外设时钟使能 (LPTIM4 Peripheral Clocks Enable During CSleep Mode)

由软件置 1 和复位。

0：CSleep 模式期间禁止 LPTIM4 外设时钟

1：CSleep 模式期间使能 LPTIM4 外设时钟（复位后的默认值）

LPTIM4 的外设时钟为：由 LPTIM345SEL 选择并提供给 **lptim_ker_ck** 输入的内核时钟，以及 **rcc_pclk4** 总线接口时钟。

位 10 **LPTIM3LPEN**：CSleep 模式期间的 LPTIM3 外设时钟使能 (LPTIM3 Peripheral Clocks Enable During CSleep Mode)

由软件置 1 和复位。

0：CSleep 模式期间禁止 LPTIM3 外设时钟

1：CSleep 模式期间使能 LPTIM3 外设时钟（复位后的默认值）

LPTIM3 的外设时钟为：由 LPTIM345SEL 选择并提供给 **lptim_ker_ck** 输入的内核时钟，以及 **rcc_pclk4** 总线接口时钟。

位 9 **LPTIM2LPEN**：CSleep 模式期间的 LPTIM2 外设时钟使能 (LPTIM2 Peripheral Clocks Enable During CSleep Mode)

由软件置 1 和复位。

0：CSleep 模式期间禁止 LPTIM2 外设时钟

1：CSleep 模式期间使能 LPTIM2 外设时钟（复位后的默认值）

LPTIM5 的外设时钟为：由 LPTIM2SEL 选择并提供给 **lptim_ker_ck** 输入的内核时钟，以及 **rcc_pclk4** 总线接口时钟。

位 8 保留，必须保持复位值。

位 7 **I2C4LPEN**：CSleep 模式期间的 I2C4 外设时钟使能 (I2C4 Peripheral Clocks Enable During CSleep Mode)

由软件置 1 和复位。

0：CSleep 模式期间禁止 I2C4 外设时钟

1：CSleep 模式期间使能 I2C4 外设时钟（复位后的默认值）

I2C4 的外设时钟为：由 I2C4SEL 选择并提供给 **i2c_ker_ck** 输入的内核时钟，以及 **rcc_pclk4** 总线接口时钟。

位 6 保留，必须保持复位值。

位 5 **SPI6LPEN**：CSleep 模式期间的 SPI6 外设时钟使能 (SPI6 Peripheral Clocks Enable During CSleep Mode)

由软件置 1 和复位。

0：CSleep 模式期间禁止 SPI6 外设时钟

1：CSleep 模式期间使能 SPI6 外设时钟（复位后的默认值）

SPI6 的外设时钟为：由 SPI6SEL 选择并提供给 **spi_ker_ck** 输入的内核时钟，以及 **rcc_pclk4** 总线接口时钟。

位 4 保留，必须保持复位值。

位 3 **LPUART1LPEN**：CSleep 模式期间的 LPUART1 外设时钟使能 (LPUART1 Peripheral Clocks Enable During CSleep Mode)

由软件置 1 和复位。

0：CSleep 模式期间禁止 LPUART1 外设时钟

1：CSleep 模式期间使能 LPUART1 外设时钟（复位后的默认值）

LPUART1 的外设时钟为：由 LPUART1SEL 选择并提供给 **lpuart_ker_ck** 输入的内核时钟，以及 **rcc_pclk4** 总线接口时钟。

位 2 保留，必须保持复位值。

位 1 **SYSCFGLPEN**：CSleep 模式期间的 SYSCFG 外设时钟使能 (SYSCFG peripheral clock enable during CSleep mode)

由软件置 1 和复位。

0：CSleep 模式期间禁止 SYSCFG 外设时钟

1：CSleep 模式期间使能 SYSCFG 外设时钟（复位后的默认值）

位 0 保留，必须保持复位值。

8.8 RCC 寄存器映射

表 74. RCC 寄存器映射和复位值

偏移	寄存器名	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x000	RCC_CR	Res.	Res.	PLL3RDY	PLL3ON	PLL2RDY	PLL2ON	PLL1RDY	PLL1ON	Res.	Res.	Res.	Res.	HSECSSON	HSEBYP	HSERDY	HSEON	D2CKRDY	D1CKRDY	HSI48RDY	HSI48ON	Res.	Res.	CSIKERON	CSIRDY	CSION	Res.	HSIDIVF	HSIDIV[1:0]		HSIRDY	HSIKERON	HSION
	Reset value			0	0	0	0	0	0					0	0	0	0	0	0	0	0			0	0	0		0	0	0	0	0	1
0x004	RCC_ICSCR	Res.	CSITRIM[4:0]					CSICAL[7:0]								HSITRIM[5:0]						HSICAL[11:0]											
	Reset value		1	0	0	0	0	-	-	-	-	-	-	-	-	1	0	0	0	0	0	-	-	-	-	-	-	-	-	-	-	-	-
0x008	RCC_CRRCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	HSI48CAL[9:0]									
	Reset value																							-	-	-	-	-	-	-	-	-	-
0x00C	reserved	reserved																															
0x010	RCC_CFGR	MCO2[2:0]			MCO2PRE[3:0]				MCO1[2:0]			MCO1PRE[3:0]				Res.	Res.	TIMPRE	HRTIMSEL	RTCPRE[5:0]						STOPKERWUCK	STOPWUCK	SWS[2:0]			SW[2:0]		
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x014	reserved	reserved																															
0x018	RCC_D1CFGR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	D1CPRE[3:0]				Res.	D1PPRE[2:0]			HPRE[3:0]			
	Reset value																					0	0	0	0		0	0	0	0	0	0	0
0x01C	RCC_D2CFGR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	D2PPRE2[2:0]			Res.	D2PPRE1[2:0]			Res.	Res.	Res.	Res.
	Reset value																						0	0	0		0	0	0				
0x020	RCC_D3CFGR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	D3PPRE[2:0]			Res.	Res.	Res.	Res.
	Reset value																										0	0	0				
0x024	reserved	reserved																															
0x028	RCC_ PLLCKSELR	Res.	Res.	Res.	Res.	Res.	Res.	DIVM3[5:0]						Res.	Res.	DIVM2[5:0]						Res.	Res.	DIVM1[5:0]						Res.	Res.	PLLSRC[1:0]	
	Reset value							1	0	0	0	0	0			1	0	0	0	0	0			1	0	0	0	0	0			0	0

表 74.（续）RCC 寄存器映射和复位值

偏移	寄存器名	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x02C	RCC_PLLCFGR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DIVR3EN	DIVQ3EN	DIVP3EN	DIVR2EN	DIVQ2EN	DIVP2EN	DIVR1EN	DIVQ1EN	DIVP1EN	Res.	Res.	Res.	Res.	PLL3RGE[1:0]		PLL3VCOSEL	PLL3FRACEN	PLL2RGE[1:0]		PLL2VCOSEL	PLL2FRACEN	PLL1RGE[1:0]		PLL1VCOSEL	PLL1FRACEN
	Reset value								1	1	1	1	1	1	1	1	1					0	0	0	0	0	0	0	0	0	0	0	0
0x030	RCC_PLL1DIVR	Res.	DIVR1[6:0]							Res.	DIVQ1[6:0]							DIVP1[6:0]							DIVN1[8:0]								
	Reset value		0	0	0	0	0	0	1		0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0
0x034	RCC_PLL1FRACR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FRACN1[12:0]													Res.	Res.	Res.
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0			
0x038	RCC_PLL2DIVR	Res.	DIVR2[6:0]							Res.	DIVQ2[6:0]							DIVP2[6:0]							DIVN2[8:0]								
	Reset value		0	0	0	0	0	0	1		0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0
0x03C	RCC_PLL2FRACR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FRACN2[12:0]													Res.	Res.	Res.
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0			
0x040	RCC_PLL3DIVR	Res.	DIVR3[6:0]							Res.	DIVQ3[6:0]							DIVP3[6:0]							DIVN3[8:0]								
	Reset value		0	0	0	0	0	0	1		0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0
0x044	RCC_PLL3FRACR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FRACN3[12:0]													Res.	Res.	Res.
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0			
0x048	reserved	reserved																															
0x04C	RCC_D1CCIPR	Res.	Res.	CKPERSEL[1:0]		Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SDMMCSEL	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	QSPISEL[1:0]		Res.	Res.	FMCSEL[1:0]	
	Reset value			0	0												0											0	0			0	0
0x050	RCC_D2CCIP1R	SWPSEL	Res.	FDCANSEL[1:0]		Res.	Res.	Res.	DFSDM1SEL	Res.	Res.	SPDIFSEL[1:0]		Res.	SPI45SEL[2:0]			Res.	SPI123SEL[2:0]			Res.	Res.	Res.	SAI23SEL[2:0]			Res.	Res.	Res.	SAI1SEL[2:0]		
	Reset value	0		0	0				0			0	0		0	0	0		0	0	0				0	0	0				0	0	0
0x054	RCC_D2CCIP2R	Res.	LPTIM1SEL[2:0]			Res.	Res.	Res.	Res.	CECSEL[1:0]		USBSEL[1:0]		Res.	Res.	Res.	Res.	Res.	Res.	I2C123SEL[1:0]		Res.	Res.	RNGSEL[1:0]		Res.	Res.	USART16SEL[2:0]			USART234578SEL[2:0]		
	Reset value		0	0	0					0	0	0	0							0	0			0	0			0	0	0	0	0	0

表 74.（续）RCC 寄存器映射和复位值

偏移	寄存器名	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x058	RCC_D3CCIPR	Res.	SPI6SEL[2:0]			Res.	SAI4BSEL[2:0]			SAI4ASEL[2:0]			Res.	Res.	Res.	ADCSEL[1:0]		LPTIM345SEL[2:0]			LPTIM2SEL[2:0]			I2C4SEL[1:0]		Res.	Res.	Res.	Res.	Res.	LPUART1SEL[2:0]		
	Reset value		0	0	0		0	0	0	0	0	0				0	0	0	0	0	0	0	0	0	0						0	0	0
0x05C	**reserved**	reserved																															
0x060	RCC_CIER	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LSECSSIE	PLL3RDYIE	PLL2RDYIE	PLL1RDYIE	HSI48RDYIE	CSIRDYIE	HSERDYIE	HSIRDYIE	LSERDYIE	LSIRDYIE
	Reset value																							0	0	0	0	0	0	0	0	0	0
0x064	RCC_CIFR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	HSECSSF	LSECSSF	PLL3RDYF	PLL2RDYF	PLL1RDYF	HSI48RDYF	CSIRDYF	HSERDYF	HSIRDYF	LSERDYF	LSIRDYF
	Reset value																						0	0	0	0	0	0	0	0	0	0	0
0x068	RCC_CICR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	HSECSSC	LSECSSC	PLL3RDYC	PLL2RDYC	PLL1RDYC	HSI48RDYC	CSIRDYC	HSERDYC	HSIRDYC	LSERDYC	LSIRDYC
	Reset value																						0	0	0	0	0	0	0	0	0	0	0
0x06C	**reserved**	reserved																															
0x070	RCC_BDCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BDRST	RTCEN	Res.	Res.	Res.	Res.	Res.	RTCSEL[1:0]		Res.	LSECSSD	LSECSSON	LSEDRV[1:0]		LSEBYP	LSERDY	LSEON
	Reset value																0	0						0	0		0	0	0	0	0	0	0
0x074	RCC_CSR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LSIRDY	LSION
	Reset value																															0	0
0x078	**reserved**	reserved																															
0x07C	RCC_ AHB3RSTR	CPURST	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SDMMC1RST	Res.	QSPIRST	Res.	FMCRST	Res.	Res.	Res.	Res.	Res.	Res.	JPGDECRST	DMA2DRST	Res.	Res.	Res.	MDMARST
	Reset value	0															0		0		0							0	0				0
0x080	RCC_ AHB1RSTR	Res.	Res.	Res.	Res.	USB2OTGRST	Res.	USB1OTGRST	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ETH1MACRST	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ADC12RST	Res.	Res.	Res.	DMA2RST	DMA1RST
	Reset value					0		0										0										0				0	0

表 74.（续）RCC 寄存器映射和复位值

偏移	寄存器名	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x084	**RCC_ AHB2RSTR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SDMMC2RST	Res.	Res.	RNGRST	HASHRST:	CRYPTRST	Res.	Res.	Res.	CAMITFRST
	Reset value																							0			0	0	0				0
0x088	**RCC_ AHB4RSTR**	Res.	Res.	Res.	Res.	Res.	Res.	HSEMRST	ADC3RST	Res.	Res.	BDMARST	Res.	CRCRST	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	GPIOKRST	GPIOJRST	GPIOIRST	GPIOHRST	GPIOGRST	GPIOFRST	GPIOERST	GPIODRST	GPIOCRST	GPIOBRST	GPIOARST
	Reset value							0	0			0		0									0	0	0	0	0	0	0	0	0	0	0
0x08C	**RCC_ APB3RSTR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LTDCRST	Res.	Res.	Res.
	Reset value																													0			
0x090	**RCC_ APB1LRSTR**	UART8RST	UART7RST	DAC12RST	Res.	HDMICECRST	Res.	Res.	Res.	I2C3RST	I2C2RST	I2C1RST	UART5RST	UART4RST	USART3RST	USART2RST	SPDIFRXRST	SPI3RST	SPI2RST	Res.	Res.	Res.	Res.	LPTIM1RST	TIM14RST	TIM13RST	TIM12RST	TIM7RST	TIM6RST	TIM5RST	TIM4RST	TIM3RST	TIM2RST
	Reset value	0	0	0		0				0	0	0	0	0	0	0	0	0	0					0	0	0	0	0	0	0	0	0	0
0x094	**RCC_ APB1HRSTR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FDCANRST	Res.	Res.	MDIOSRST	OPAMPRST	Res.	RST	CRSRST	Res.
	Reset value																								0			0	0		0	0	
0x098	**RCC_ APB2RSTR**	Res.	Res.	HRTIMRST	DFSDM1RST	Res.	Res.	Res.	SAI3RST	SAI2RST	SAI1RST	Res.	SPI5RST	Res.	TIM17RST	TIM16RST	TIM15RST	Res.	Res.	SPI4RST	SPI1RST	Res.	Res.	Res.	Res.	Res.	Res.	Res.	USART6RST	USART1RST	Res.	Res.	TIM8RST	TIM1RST
	Reset value			0	0				0	0	0		0		0	0	0			0	0							0	0			0	0
0x09C	**RCC_ APB4RSTR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SAI4RST	Res.	Res.	Res.	Res.	VREFRST	COMP12RST	Res.	LPTIM5RST	LPTIM4RST	LPTIM3RST	LPTIM2RST	Res.	Res.	I2C4RST	Res.	SPI6RST	Res.	LPUART1RST	Res.	SYSCFGRST	Res.
	Reset value											0					0	0		0	0	0	0			0		0		0		0	
0x0A0	**RCC_GCR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WW1RSC
	Reset value																																0
0x0A4	**reserved**	reserved																															

表 74.（续）RCC 寄存器映射和复位值

偏移	寄存器名	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x0A8	**RCC_D3AMR**	Res.		SRAM4AMEN	BKPRAMAMEN	Res.	Res.	Res.	ADC3AMEN	Res.	Res.	SAI4AMEN	Res.	CRCAMEN	Res.	Res.	RTCAMEN	VREFAMEN	COMP12AMEN	Res.	LPTIM5AMEN	LPTIM4AMEN	LPTIM3AMEN	LPTIM2AMEN	Res.	I2C4AMEN	Res.	SPI6AMEN	Res.	LPUART1AMEN	Res.	Res.	BDMAAMEN
	Reset value			0	0				0			0		0			0	0	0		0	0	0	0		0		0		0			0
0x0AC to 0x0CC	**reserved**	reserved																															
0x0D0	**RCC_RSR**	Res.	LPWRRSTF	Res.	WWDG1RSTF	Res.	IWDG1RSTF	Res.	SFTRSTF	PORRSTF	PINRSTF;	BORRSTF	D2RSTF	D1RSTF	Res.	CPURSTF	RMVF	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	Reset value		0		0		0		0	1	1	1	1	1		1	0																
0x0D4	**RCC_ AHB3ENR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SDMMC1EN	Res.	QSPIEN	Res.	FMCEN	Res.	Res.	Res.	Res.	Res.	Res.	JPGDECEN	DMA2DEN	Res.	Res.	Res.	MDMAEN
	Reset value																0		0		0							0	0				0
0x0D8	**RCC_ AHB1ENR**	Res.	Res.	Res.	USB2ULPIEN	USB2OTGEN	USB1ULPIEN	USB1OTGEN	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ETH1RXEN	ETH1TXEN	ETH1MACEN	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ADC12EN	Res.	Res.	Res.	DMA2EN	DMA1EN
	Reset value				0	0	0	0								0	0	0										0				0	0
0x0DC	**RCC_ AHB2ENR**	SRAM3EN	SRAM2EN	SRAM1EN	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SDMMC2EN	Res.	Res.	RNGEN	Res.	HASHEN	CRYPTEN	Res.	Res.	Res.	DCMIEN
	Reset value	0	0	0																			0			0	0	0	0				0
0x0E0	**RCC_ AHB4ENR**	Res.	Res.	Res.	BKPRAMEN	Res.	Res.	HSEMEN	ADC3EN	Res.	Res.	BDMAEN	Res.	CRCEN	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	GPIOKEN	GPIOJEN	GPIOIEN	GPIOHEN	GPIOGEN	GPIOFEN	GPIOEEN	GPIODEN	GPIOCEN	GPIOBEN	GPIOAEN
	Reset value				0			0	0			0		0									0	0	0	0	0	0	0	0	0	0	0
0x0E4	**RCC_ APB3ENR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WWDG1EN	Res.	Res.	LTDCEN	Res.	Res.	Res.
	Reset value																										0			0			
0x0E8	**RCC_ APB1LENR**	UART8EN	UART7EN	DAC12EN	Res.	HDMICECEN	Res.	Res.	Res.	I2C3EN	I2C2EN	I2C1EN	UART5EN	UART4EN	USART3EN	USART2EN	SPDIFRXEN	SPI3EN	SPI2EN	Res.	Res.	Res.	Res.	LPTIM1EN	TIM14EN	TIM13EN	TIM12EN	TIM7EN	TIM6EN	TIM5EN	TIM4EN	TIM3EN	TIM2EN
	Reset value	0	0	0		0				0	0	0	0	0	0	0	0	0	0					0	0	0	0	0	0	0	0	0	0

表 74.（续）RCC 寄存器映射和复位值

偏移	寄存器名	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x0EC	RCC_ APB1HENR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FDCANEN	Res.	Res.	MDIOSEN	OPAMPEN	Res.	SWPEN	CRSEN	Res.
	Reset value																								0			0	0		0	0	
0x0F0	RCC_ APB2ENR	Res.	Res.	HRTIMEN	DFSDM1EN	Res.	Res.	Res.	SAI3EN	SAI2EN	SAI1EN	Res.	SPI5EN	Res.	TIM17EN	TIM16EN	TIM15EN	Res.	Res.	SPI4EN	SPI1EN	Res.	Res.	Res.	Res.	Res.	Res.	USART6EN	USART1EN	Res.	Res.	TIM8EN	TIM1EN
	Reset value			0	0				0	0	0		0		0	0	0			0	0							0	0			0	0
0x0F4	RCC_ APB4ENR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SAI4EN	Res.	Res.	Res.	Res.	RTCAPBEN	VREFEN	COMP12EN	Res.	LPTIM5EN	LPTIM4EN	LPTIM3EN	LPTIM2EN	Res.	I2C4EN	reserved	SPI6EN	Res.	LPUART1EN	Res.	SYSCFGEN	Res.
	Reset value											0					1	0	0		0	0	0	0		0	0	0		0		0	
0x0F8	reserved	reserved																															
0x0FC	RCC_ AHB3LPENR	AXISRAMLPEN	ITCMLPEN	DTCM2LPEN	DTCM1LPEN	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SDMMC1LPEN	Res.	QSPILPEN	Res.	FMCLPEN	Res.	Res.	Res.	FLASHLPEN	Res.	Res.	JPGDECLPEN	DMA2DLPEN	Res.	Res.	Res.	MDMALPEN
	Reset value	1	1	1	1												1		1		1				1			1	1				1
0x100	RCC_ AHB1LPENR	Res.	Res.	Res.	USB2ULPILPEN	USB2OTGLPEN	USB1ULPILPEN	USB1OTGLPEN	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ETH1RXLPEN	ETH1TXLPEN	ETH1MACLPEN	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ADC12LPEN	Res.	Res.	Res.	DMA2LPEN	DMA1LPEN
	Reset value				1	1	1	1								1	1	1										1				1	1
0x104	RCC_ AHB2LPENR	SRAM3LPEN	SRAM2LPEN	SRAM1LPEN	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SDMMC2LPEN	Res.	Res.	Res.	RNGLPEN	HASHLPEN	CRYPTLPEN	Res.	Res.	Res.	CAMITFLPEN
	Reset value	1	1	1																			1				1	1	1				1
0x108	RCC_ AHB4LPENR	Res.	Res.	SRAM4LPEN	BKPRAMLPEN	Res.	Res.	Res.	ADC3LPEN	Res.	Res.	BDMALPEN	Res.	CRCLPEN	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	GPIOKLPEN	GPIOJLPEN	GPIOILPEN	GPIOHLPEN	GPIOGLPEN	GPIOFLPEN	GPIOELPEN	GPIODLPEN	GPIOCLPEN	GPIOBLPEN	GPIOALPEN
	Reset value			1	1				1			1		1									1	1	1	1	1	1	1	1	1	1	1
0x10C	RCC_ APB3LPENR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WWDG1LPEN	Res.	Res.	LTDCLPEN	Res.	Res.	Res.
	Reset value																										1			1			

表 74.（续）RCC 寄存器映射和复位值

偏移	寄存器名	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x110	RCC_ APB1LLPENR	UART8LPEN	UART7LPEN	DAC12LPEN	Res.	CECLPEN	Res.	Res.	Res.	I2C3LPEN	I2C2LPEN	I2C1LPEN	UART5LPEN	UART4LPEN	USART3LPEN	USART2LPEN	SPDIFRXLPEN	SPI3LPEN	SPI2LPEN	Res.	Res.	Res.	Res.	LPTIM1LPEN	TIM14LPEN	TIM13LPEN	TIM12LPEN	TIM7LPEN	TIM6LPEN	TIM5LPEN	TIM4LPEN	TIM3LPEN	TIM2LPEN
	Reset value	1	1	1		1				1	1	1	1	1	1	1	1	1	1					1	1	1	1	1	1	1	1	1	1
0x114	RCC_ APB1HLPENR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FDCANLPEN	Res.	Res.	MDIOSLPEN	OPAMPLPEN	Res.	SWPLPEN	CRSLPEN	Res.
	Reset value																								1			1	1		1	1	
0x118	RCC_ APB2LPENR	Res.	Res.	HRTIMLPEN	DFSDM1LPEN	Res.	Res.	Res.	SAI3LPEN	SAI2LPEN	SAI1LPEN	Res.	SPI5LPEN	Res.	TIM17LPEN	TIM16LPEN	TIM15LPEN	Res.	Res.	SPI4LPEN	SPI1LPEN	Res.	Res.	Res.	Res.	Res.	Res.	USART6LPEN	USART1LPEN	Res.	Res.	TIM8LPEN	TIM1LPEN
	Reset value			1	1				1	1	1		1		1	1	1			1	1							1	1			1	1
0x11C	RCC_ APB4LPENR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SAI4LPEN	Res.	Res.	Res.	Res.	RTCAPBLPEN	VREFLPEN	COMP12LPEN	Res.	LPTIM5LPEN	LPTIM4LPEN	LPTIM3LPEN	LPTIM2LPEN	Res.	I2C4LPEN	Res.	SPI6LPEN	Res.	LPUART1LPEN	Res.	SYSCFGLPEN	Res.
	Reset value											1					1	1	1		1	1	1	1		1		1		1		1	
0x120 to 0x130	reserved	reserved																															
0x134	RCC_C1_ AHB3ENR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SDMMC1EN	Res.	QSPIEN	Res.	FMCEN	Res.	Res.	Res.	Res.	Res.	Res.	JPGDECEN	DMA2DEN	Res.	Res.	Res.	MDMAEN
	Reset value																0		0		0							0	0				0
0x138	RCC_C1_ AHB1ENR	Res.	Res.	Res.	USB2ULPIEN	USB2OTGEN	USB1ULPIEN	USB1OTGEN	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ETH1RXEN	ETH1TXEN	ETH1MACEN	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ADC12EN	Res.	Res.	Res.	DMA2EN	DMA1EN
	Reset value				0	0	0	0								0	0	0										0				0	0
0x13C	RCC_C1_ AHB2ENR	SRAM3EN	SRAM2EN	SRAM1EN	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SDMMC2EN	Res.	Res.	RNGEN	HASHEN	CRYPTEN	Res.	Res.	Res.	DCMIEN
	Reset value	0	0	0																			0			0	0	0				0
0x140	RCC_C1_ AHB4ENR	Res.	Res.	Res.	BKPRAMEN	Res.	Res.	HSEMEN	ADC3EN	Res.	Res.	DMA1EN	Res.	CRCEN	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	GPIOKEN	GPIOJEN	GPIOIEN	GPIOHEN	GPIOGEN	GPIOFEN	GPIOEEN	GPIODEN	GPIOCEN	GPIOBEN	GPIOAEN
	Reset value				0			0	0			0		0									0	0	0	0	0	0	0	0	0	0	0

表 74.（续）RCC 寄存器映射和复位值

偏移	寄存器名	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x144	**RCC_C1_ APB3ENR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WWDG1EN	Res.	Res.	LTDCEN	Res.	Res.	Res.
	Reset value																										0			0			
0x148	**RCC_C1_ APB1LENR**	UART8EN	UART7EN	DAC12EN	Res.	HDMICECEN	Res.	Res.	Res.	I2C3EN	I2C2EN	I2C1EN	UART5EN	UART4EN	USART3EN	USART2EN	SPDIFRXEN	SPI3EN	SPI2EN	Res.	Res.	Res.	Res.	LPTIM1EN	TIM14EN	TIM13EN	TIM12EN	TIM7EN	TIM6EN	TIM5EN	TIM4EN	TIM3EN	TIM2EN
	Reset value	0	0	0		0				0	0	0	0	0	0	0	0	0	0					0	0	0	0	0	0	0	0	0	0
0x14C	**RCC_C1_ APB1HENR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FDCANEN	Res.	Res.	MDIOSEN	OPAMPEN	Res.	SWPEN	CRSEN	Res.
	Reset value																								0			0	0		0	0	
0x150	**RCC_C1_ APB2ENR**	Res.	Res.	HRTIMEN	DFSDM1EN	Res.	Res.	Res.	SAI3EN	SAI2EN	SAI1EN	Res.	SPI5EN	Res.	TIM17EN	TIM16EN	TIM15EN	Res.	Res.	SPI4EN	SPI1EN	Res.	Res.	Res.	Res.	Res.	Res.	USART6EN	USART1EN	Res.	Res.	TIM8EN	TIM1EN
	Reset value			0	0				0	0	0		0		0	0	0			0	0							0	0			0	0
0x154	**RCC_C1_ APB4ENR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SAI4EN	Res.	Res.	Res.	Res.	RTCAPBEN	VREFEN	COMP12EN	Res.	LPTIM5EN	LPTIM4EN	LPTIM3EN	LPTIM2EN	Res.	I2C4EN	Res.	SPI6EN	Res.	LPUART1EN	Res.	SYSCFGEN	Res.
	Reset value											0					1	0	0		0	0	0	0		0		0		0		0	
0x158	**reserved**	reserved																															
0x15C	**RCC_C1_ AHB3LPENR**	AXISRAMLPEN	ITCMLPEN	DTCM2LPEN	DTCM1LPEN	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SDMMC1LPEN	Res.	QSPILPEN	Res.	FMCLPEN	Res.	Res.	Res.	FLASHLPEN	Res.	Res.	JPGDECLPEN	DMA2DLPEN	Res.	Res.	Res.	MDMALPEN
	Reset value	1	1	1	1												1		1		1				1			1	1				1
0x160	**RCC_C1_ AHB1LPENR**	Res.	Res.	Res.	USB2ULPILPEN	USB2OTGLPEN	USB1ULPILPEN	USB1OTGLPEN	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ETH1RXLPEN	ETH1TXLPEN	ETH1MACLPEN	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ADC12LPEN	Res.	Res.	Res.	DMA2LPEN	DMA1LPEN
	Reset value				1	1	1	1								1	1	1										1				1	1
0x164	**RCC_C1_ AHB2LPENR**	SRAM3LPEN	SRAM2LPEN	SRAM1LPEN	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SDMMC2LPEN	Res.	Res.	RNGLPEN	HASHLPEN	CRYPTLPEN	Res.	Res.	Res.	CAMITFLPEN
	Reset value	1	1	1																			1			1	1	1				1

表 74.（续）RCC 寄存器映射和复位值

偏移	寄存器名	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x168	**RCC_C1_ AHB4LPENR**	Res.	Res.	SRAM4LPEN	BKPRAMLPEN	Res.	Res.	Res.	ADC3LPEN	Res.	Res.	DMA1LPEN	Res.	CRCLPEN	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	GPIOKLPEN	GPIOJLPEN	GPIOILPEN	GPIOHLPEN	GPIOGLPEN	GPIOFLPEN	GPIOELPEN	GPIODLPEN	GPIOCLPEN	GPIOBLPEN	GPIOALPEN
	Reset value			1	1				1			1		1									1	1	1	1	1	1	1	1	1	1	1
0x16C	**RCC_C1_ APB3LPENR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WWDG1LPEN	Res.	Res.	LTDCLPEN	Res.	Res.	Res.
	Reset value																										1			1			
0x170	**RCC_C1_ APB1LLPENR**	UART8LPEN	UART7LPEN	DAC12LPEN	Res.	CECLPEN	Res.	Res.	Res.	I2C3LPEN	I2C2LPEN	I2C1LPEN	UART5LPEN	UART4LPEN	USART3LPEN	USART2LPEN	SPDIFRXLPEN	SPI3LPEN	SPI2LPEN	Res.	Res.	Res.	LPTIM1LPEN	TIM14LPEN	TIM13LPEN	TIM12LPEN	TIM7LPEN	TIM6LPEN	TIM5LPEN	TIM4LPEN	TIM3LPEN	TIM2LPEN
	Reset value	1	1	1		1				1	1	1	1	1	1	1	1	1	1				1	1	1	1	1	1	1	1	1	1	1
0x174	**RCC_C1_ APB1HLPENR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FDCANLPEN	Res.	Res.	MDIOSLPEN	OPAMPLPEN	Res.	SWPLPEN	CRSLPEN	Res.
	Reset value																								1			1	1		1	1	
0x178	**RCC_C1_ APB2LPENR**	Res.	Res.	HRTIMLPEN	DFSDM1LPEN	Res.	Res.	Res.	SA3LPEN	SAI2LPEN	SAI1LPEN	Res.	SPI5LPEN	Res.	TIM17LPEN	TIM16LPEN	TIM15LPEN	Res.	Res.	SPI4LPEN	SPI1LPEN	Res.	Res.	Res.	Res.	Res.	Res.	USART6LPEN	USART1LPEN	Res.	Res.	TIM8LPEN	TIM1LPEN
	Reset value			1	1				1	1	1		1		1	1	1			1	1							1	1			1	1
0x17C	**RCC_C1_ APB4LPENR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SAI4LPEN	Res.	Res.	Res.	Res.	RTCAPBLPEN	VREFLPEN	COMP12LPEN	Res.	LPTIM5LPEN	LPTIM4LPEN	LPTIM3LPEN	LPTIM2LPEN	Res.	I2C4LPEN	Res.	SPI6LPEN	Res.	LPUART1LPEN	Res.	SYSCFGLPEN	Res.
	Reset value											1					1	1	1		1	1	1	1		1		1		1		1	
0x180 to 0x1FC	**reserved**	reserved																															

有关寄存器边界地址的信息，请参见*第 100 页的第 2.2.2 节*。

9 时钟恢复系统 (CRS)

9.1 前言

时钟恢复系统 (CRS) 是一个作用于内部精细粒度可调 RC 振荡器 HSI48 的高级数字控制器。CRS 提供一种十分强大的方法，可通过与可选同步信号进行比较来评估振荡器输出频率。它能够根据测得的频率误差值自动调整振荡器微调，同时还可以进行手动微调。

CRS 非常适合为 USB 外设提供精密时钟。在这种情况下，同步信号可由 USB 总线上的帧起始 (SOF) 数据包信号提供，USB 主机以 1 ms 的时间间隔精确发送该信号。

同步信号也可由 LSE 振荡器输出提供，或者由用户软件生成。

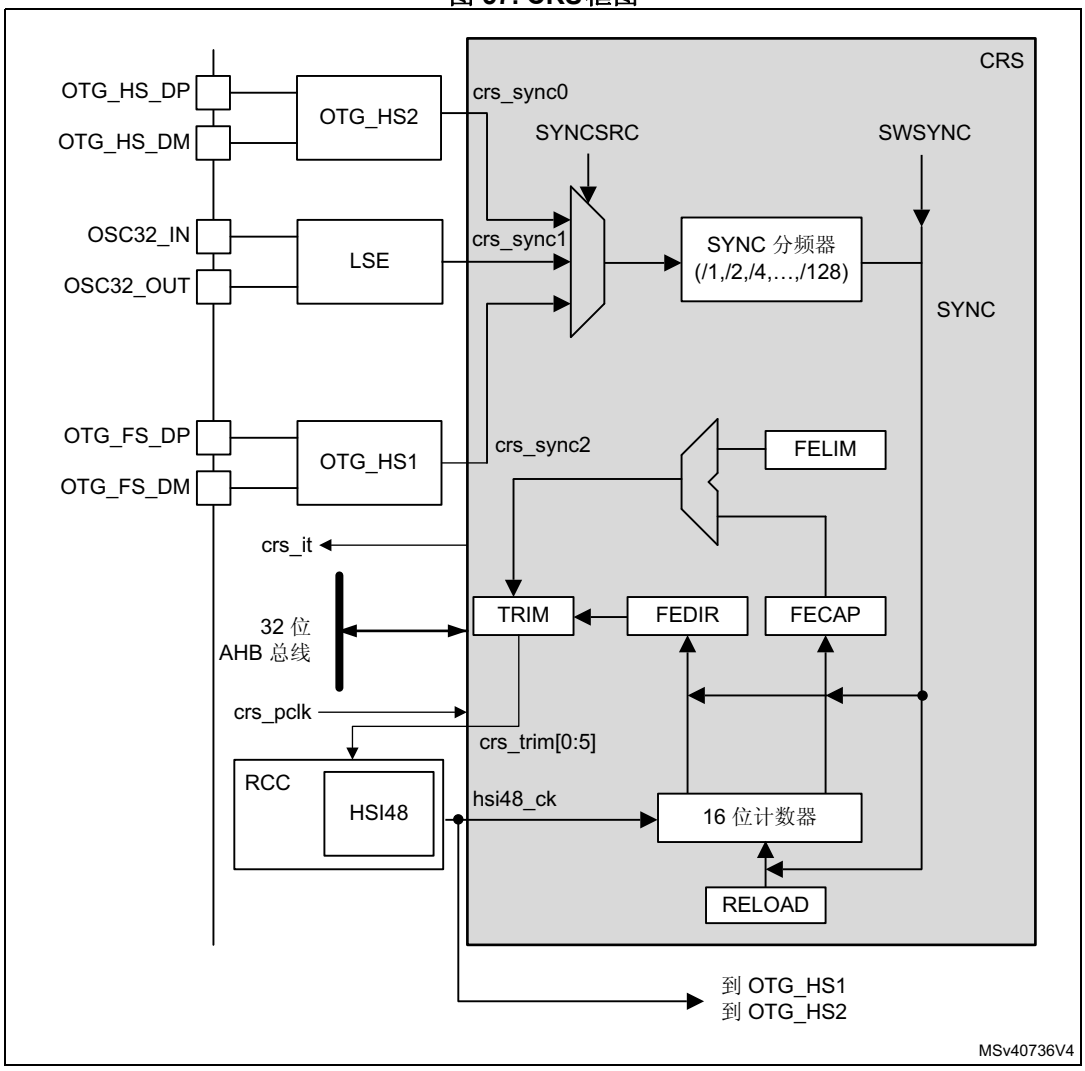
9.2 CRS 主要特性

- 具有可编程预分频器和极性的可选同步源：
  - USB2 SOF 数据包接收
  - LSE 振荡器输出
  - USB1 SOF 数据包接收
- 可由软件生成同步脉冲
- 振荡器自动微调功能，无需 CPU 操作
- 手动控制选项，可加快启动融合
- 16 位频率误差计数器，用于自动误差值捕捉和重载
- 可编程限值，用于自动频率误差值评估和状态报告
- 可屏蔽中断/事件：
  - 预期同步 (ESYNC)
  - 同步正常 (SYNCOK)
  - 同步警告 (SYNCWARN)
  - 同步或微调错误 (ERR)

9.3 CRS 功能说明

9.3.1 CRS框图

图 57. CRS框图



9.4 CRS 内部信号

表 75 列出了 CRS 内部信号。

表 75. CRS 内部输入/输出信号

信号名称	信号类型	说明
crs_it	数字输出	CRS 中断
crs_pclk	数字输入	AHB 总线时钟
hsi48_ck	数字输入	HSI48 振荡器时钟
crs_trim[0:5]	数字输出	HSI48 振荡器平滑微调值
crs_sync0、crs_sync1 和 crs_sync2	数字输入	SYNC 信号源选择（USB2、LSE 或 USB1）

9.4.1 同步输入

CRS 同步 (SYNC) 源可通过 CRS_CFGR 寄存器选择，它可以是来自 LSE 时钟的信号、USB1 SOF 信号或 USB2 SOF 信号。该信号源还具有可配置极性，可随后由可编程二进制预分频器分频，以获得合理频率范围内的同步信号（通常为 1 kHz 左右）。

有关 CRS 同步源配置的更多信息，请参见*第 9.7.2 节：CRS 配置寄存器 (CRS_CFGR)*。

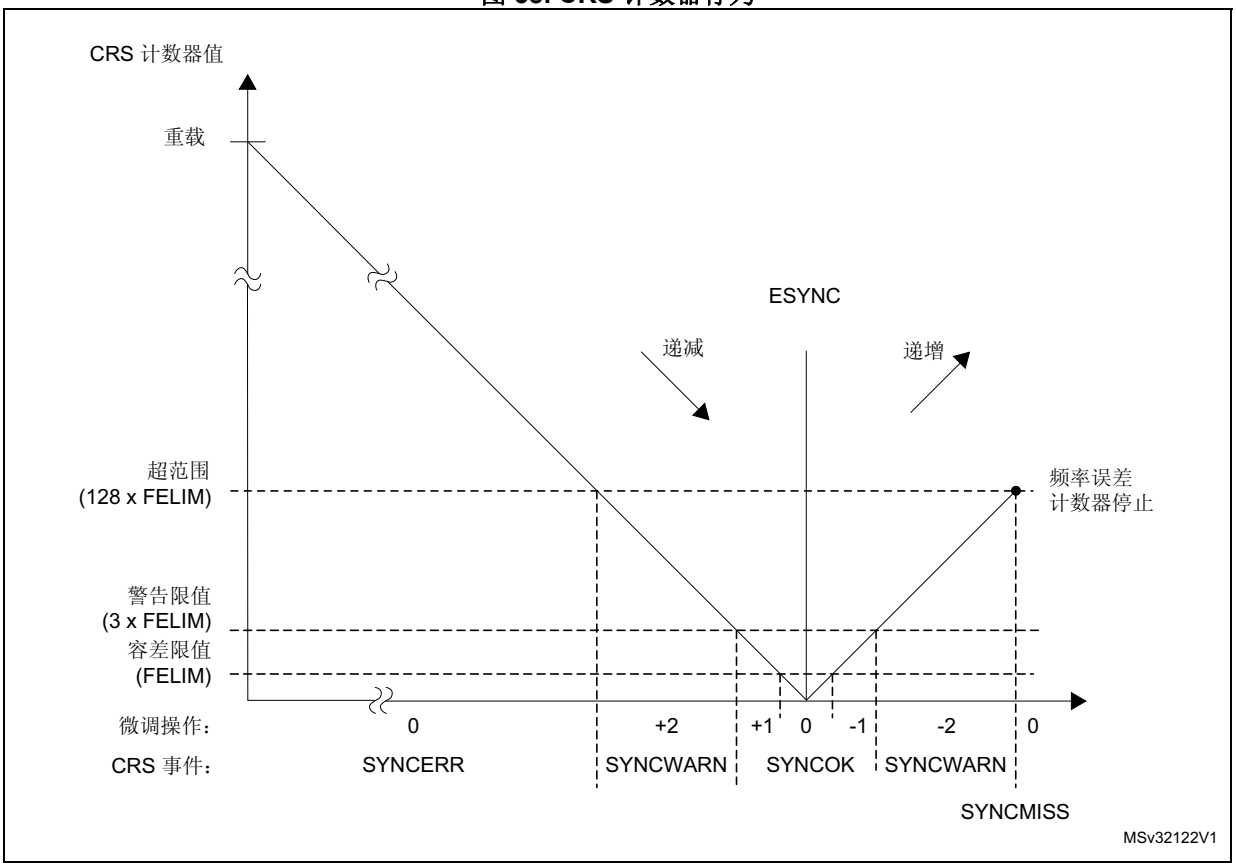
还可通过用软件将 CRS_CR 寄存器中的 SWSYNC 位置 1 的方法来生成同步事件。

9.4.2 频率误差测量

频率误差计数器是一个 16 位递减/递增计数器，会在每个 SYNC 事件发生时重载 RELOAD 值。它开始递减计数，直至达到零值，此时会生成 ESYNC（预期同步）事件。随后它将递增计数到 OUTRANGE 限值，这种情况下最终将停止计数（如果未接收到 SYNC 事件）并生成 SYNCMISS 事件。OUTRANGE 限值定义为频率误差限值（CRS_CFGR 寄存器的 FELIM 字段）乘以 128。

当检测到 SYNC 事件时，频率误差计数器的实际值及其方向存储在 CRS_ISR 寄存器的 FECAP（频率误差捕捉）字段和 FEDIR（频率误差方向）位中。当在递减计数阶段（达到零值之前）期间检测到 SYNC 事件时，意味着实际频率小于目标频率（因此，TRIM 值应递增）；当在递增计数期间检测到 SYNC 事件时，意味着实际频率大于目标频率（因此，TRIM 值应递减）。

图 58. CRS 计数器行为



9.4.3 频率误差评估和自动微调

测得的频率误差通过将频率值与一组限值进行比较进行估算：

- TOLERANCE LIMIT，在 CRS_CFGR 寄存器的 FELIM 字段中直接给出
- WARNING LIMIT，定义为 3 * FELIM 值
- OUTRANGE（误差限值），定义为 128 * FELIM 值

该比较结果用于生成状态指示以及控制自动微调，自动微调通过将 CRS_CR 寄存器中的 AUTOTRIMEN 位置 1 来使能。

- 当频率误差低于容差限值时，意味着 TRIM 字段中的实际微调值是最优值，因此无需任何微调操作。
  - 指示 SYNCOK 状态
  - AUTOTRIM 模式下的 TRIM 值无变化
- 当频率误差低于警告限值但高于或等于容差限值时，意味着需要某种微调操作，但只需一步微调便可达到最优 TRIM 值。
  - 指示 SYNCOK 状态
  - 在 AUTOTRIM 模式下通过一步微调来调整 TRIM 值
- 当频率误差高于或等于警告限值但低于误差限值时，意味着需要更强的微调操作，并且存在下一周期无法达到最优 TRIM 值的风险。
  - 指示 SYNCWARN 状态
  - 在 AUTOTRIM 模式下通过两步微调来调整 TRIM 值
- 当频率误差高于或等于误差限值时，意味着频率超出微调范围。当 SYNC 输入不干净或某个 SYNC 脉冲丢失时（例如，当一个 USB SOF 被破坏时）。也会发生这种情况。
  - 指示 SYNCERR 或 SYNCMISS 状态
  - AUTOTRIM 模式下的 TRIM 值无变化

注： *如果 TRIM 字段的实际值十分接近其限值，则自动微调会强制其上溢或下溢，之后 TRIM 值会恰好设置为限值，此时将指示 TRIMOVF 状态。*

在 AUTOTRIM 模式（CRS_CR 寄存器中的 AUTOTRIMEN 位置 1）下，CRS_CR 的 TRIM 字段由硬件调整并且是只读的。

9.4.4 CRS 初始化和配置

RELOAD 值

RELOAD 值应根据预分频后目标频率与同步源频率之比进行选择。该值随后会减 1，以在零值时达到预期同步。具体公式如下：

$RELOAD = (f_{TARGET} / f_{SYNC}) - 1$

RELOAD 字段的复位值对应于 48 MHz 的目标频率和 1 kHz 的同步信号频率（来自 USB 的 SOF 信号）。

FELIM 值

FELIM 值的选择与 HSI48 振荡器的特性及其典型微调步长紧密相关。最优值对应于微调步长的一半，以 HSI48 振荡器时钟节拍数表示。可使用以下公式：

FELIM = (f_{TARGET} / f_{SYNC}) * STEP[%] / 100% / 2

结果应始终舍入为最接近的整数值以获得最佳微调响应。如果应用中不需要频繁的微调操作，可通过稍微增大 FELIM 值来增加微调滞后。

FELIM 字段的复位值对应于 (f_{TARGET} / f_{SYNC}) = 48000 以及 0.14% 的典型微调步长。

注意： 错误配置 RELOAD 和 FELIM 字段无法实现硬件保护，从而导致不定微调响应。预期工作模式需要正确设置 RELOAD 值（根据同步源频率），该值还大于 128 * FELIM 值（OUTRANGE 限值）。

9.5 CRS 低功耗模式

表 76. 低功耗模式对 CRS 的作用

模式	说明
睡眠	无影响。 CRS 中断可使器件退出睡眠模式。
停止	CRS 寄存器被冻结。 CRS 停止工作，直到退出停止或待机模式且 HSI48 振荡器重启。
待机	

9.6 CRS 中断

表 77. 中断控制位

中断事件	事件标志	启用控制位	清零标志位
预期同步	ESYNCF	ESYNCIE	ESYNCC
同步正常	SYNCOKF	SYNCOKIE	SYNCOKC
同步警告	SYNCWARNF	SYNCWARNIE	SYNCWARNC
同步或微调错误 （TRIMOVF、SYNCMISS 和 SYNCERR）	ERRF	ERRIE	ERRC

9.7 CRS 寄存器

有关寄存器说明中使用的缩写，请参见参考手册中的*第 94 页的第 1.1 节*。

外设寄存器可按字（32 位）进行访问。

9.7.1 CRS 控制寄存器 (CRS_CR)

CRS control register

偏移地址：0x00

复位值：0x0000 2000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	TRIM[5:0]						SWSYNC	AUTOTRIMEN	CEN	Res.	ESYNCIE	ERRIE	SYNCWARNIE	SYNCOKIE
		rw	rw	rw	rw	rw	rw	rt_w	rw	rw		rw	rw	rw	rw

位 31:14 保留，必须保持复位值。

位 13:8 **TRIM[5:0]**：HSI48 振荡器平滑微调 (HSI48 oscillator smooth trimming)

这些位为 HSI48 振荡器提供用户可编程的微调值。可通过编程使其适应电压和温度的差异，使 HSI48 的频率更为准确。

默认值为 32，该值对应于微调间隔的中间值。连续两个 TRIM 步之间的微调步长约为 67 kHz。较高的 TRIM 值对应于较高的输出频率。

当 AUTOTRIMEN 位置 1 时，该字段由硬件控制并且是只读的。

位 7 **SWSYNC**：生成软件 SYNC 事件 (Generate software SYNC event)

此位由软件置 1 以生成软件 SYNC 事件。它由硬件自动清零。

0：不执行任何操作

1：生成软件 SYNC 事件。

位 6 **AUTOTRIMEN**：自动微调使能 (Automatic trimming enable)

此位根据在两个 SYNC 事件间测得的频率误差使能 TRIM 位的自动硬件调整。如果此位置 1，则 TRIM 位是只读的。TRIM 值可通过硬件一次调整一步或两步，具体取决于测得的频率误差值。更多详细信息，请参见*第 9.4.3 节：频率误差评估和自动微调*。

0：禁止自动微调，TRIM 位可由用户调整。

1：使能自动微调，TRIM 位是只读的且由硬件控制。

位 5 **CEN**：频率误差计数器使能 (Frequency error counter enable)

此位为频率误差计数器使能振荡器时钟。

0：禁止频率误差计数器

1：使能频率误差计数器

当此位置 1 时，CRS_CFGR 寄存器被写保护，无法修改。

位 4 保留，必须保持复位值。

位 3 **ESYNCIE**：预期 SYNC 中断使能 (Expected SYNC interrupt enable)
0：禁止预期 SYNC (ESYNCF) 中断
1：使能预期 SYNC (ESYNCF) 中断

位 2 **ERRIE**：同步或微调错误中断使能 (Synchronization or trimming error interrupt enable)
0：禁止同步或微调错误 (ERRF) 中断
1：使能同步或微调错误 (ERRF) 中断

位 1 **SYNCWARNIE**：SYNC 警告中断使能 (SYNC warning interrupt enable)
0：禁止 SYNC 警告 (SYNCWARNF) 中断
1：使能 SYNC 警告 (SYNCWARNF) 中断

位 0 **SYNCOKIE**：SYNC 事件正常中断使能 (SYNC event OK interrupt enable)
0：禁止 SYNC 事件正常 (SYNCOKF) 中断
1：使能 SYNC 事件正常 (SYNCOKF) 中断

9.7.2 CRS 配置寄存器 (CRS_CFGR)

CRS configuration register

只有禁止频率误差计数器时，才可写入该寄存器（CRS_CR 中的 CEN 位清零）。当计数器使能时，该寄存器被写保护。

偏移地址：0x04

复位值：0x2022 BB7F

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

位 31 **SYNCPOL**：SYNC 极性选择 (SYNC polarity selection)
此位由软件置 1 和清零，以选择 SYNC 信号源的输入极性。
0：SYNC 在上升沿有效（默认）
1：SYNC 在下降沿有效

位 30 保留，必须保持复位值。

位 29:28 **SYNCSRC[1:0]**：SYNC 信号源选择 (SYNC signal source selection)
这些位由软件置 1 和清零，以选择 SYNC 信号源。
00：USB2 SOF 选择为 SYNC 信号源
01：LSE 选择为 SYNC 信号源
10：USB1 SOF 选择为 SYNC 信号源（默认）
11：保留

注： *当使用 USB LPM（链路层电源管理）且器件处于休眠模式时，周期性 USB SOF 不会由主机产生。因此，不会向 CRS 提供任何 SYNC 信号来校准运行状态下的 HSI48。要确保从休眠模式唤醒后所需的时钟精度，LSE 应用作 SYNC 信号。*

位 27 保留，必须保持复位值。

位 26:24 **SYNCDIV[2:0]**：SYNC 分频器 (SYNC divider)

这些位由软件置 1 和清零，以控制 SYNC 信号的分频系数。

000：SYNC 不进行分频（默认）

001：SYNC 2 分频

010：SYNC 4 分频

011：SYNC 8 分频

100：SYNC 16 分频

101：SYNC 32 分频

110：SYNC 64 分频

111：SYNC 128 分频

位 23:16 **FELIM[7:0]**：频率误差限值 (Frequency error limit)

FELIM 包含用于评估 CRS_ISR 寄存器的 FECAP[15:0] 位中锁存的已捕捉频率误差值的值。

有关 FECAP 评估的更多详细信息，请参见*第 9.4.3 节：频率误差评估和自动微调*。

位 15:0 **RELOAD[15:0]**：计数器重载值 (Counter reload value)

RELOAD 是每次发生 SYNC 事件时要装载到频率误差计数器中的值。

有关计数器行为的更多详细信息，请参见*第 9.4.2 节：频率误差测量*。

9.7.3 CRS 中断和状态寄存器 (CRS_ISR)

CRS interrupt and status register

偏移地址：0x08

复位值：0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FEDIR	Res.	Res.	Res.	Res.	TRIMOVF	SYNCMISS	SYNCERR	Res.	Res.	Res.	Res.	ESYNCF	ERRF	SYNCWARNF	SYNCOKF
r					r	r	r					r	r	r	r

位 31:16 **FECAP[15:0]**：频率误差捕捉 (Frequency error capture)

FECAP 是最后一次 SYNC 事件发生时锁存的频率误差计数器值。

有关 FECAP 使用的更多详细信息，请参见*第 9.4.3 节：频率误差评估和自动微调*。

位 15 **FEDIR**：频率误差方向 (Frequency error direction)

FEDIR 是最后一次 SYNC 事件发生时锁存的频率误差计数器的计数方向。它显示实际频率低于目标频率还是高于目标频率。

0：递增计数方向，实际频率高于目标频率。

1：递减计数方向，实际频率低于目标频率。

位 14:11 保留，必须保持复位值。

位 10 **TRIMOVF**：微调上溢或下溢 (Trimming overflow or underflow)

当自动微调尝试使 TRIM 值上溢或下溢时，该标志由硬件置 1。如果 CRS_CR 寄存器中的 ERRIE 位置 1，将生成中断。此位用软件清零，方法是将 CRS_ICR 寄存器中的 ERRC 位置 1。

0：未发出微调错误信号

1：已发出微调错误信号

位 9 **SYNCMISS**：SYNC 丢失 (SYNC missed)

当频率误差计数器达到值 FELIM * 128 且未检测到 SYNC 时，该标志由硬件置 1，这意味着 SYNC 脉冲丢失或频率误差过大（内部频率过高）而无法通过调整 TRIM 值来补偿，应采取其它措施。此时，频率误差计数器停止（等待下一个 SYNC），并且会在 CRS_CR 寄存器中的 ERRIE 位置 1 时生成中断。此位用软件清零，方法是将 CRS_ICR 寄存器中的 ERRC 位置 1。

0：未发出 SYNC 丢失错误信号

1：已发出 SYNC 丢失错误信号

位 8 **SYNCERR**：SYNC 错误 (SYNC error)

当 SYNC 脉冲在 ESYNC 事件之前达到且所测得的频率误差大于或等于 FELIM * 128 时，该标志由硬件置 1。这意味着频率误差过大（内部频率过低）而无法通过调整 TRIM 值进行补偿，应采取其它措施。如果 CRS_CR 寄存器中的 ERRIE 位置 1，将生成中断。此位用软件清零，方法是将 CRS_ICR 寄存器中的 ERRC 位置 1。

0：未发出 SYNC 错误信号

1：已发出 SYNC 错误信号

位 7:4 保留，必须保持复位值。

位 3 **ESYNCF**：预期 SYNC 标志 (Expected SYNC flag)

当频率误差计数器达到零值时，此标志由硬件置 1。如果 CRS_CR 寄存器中的 ESYNCIE 位置 1，将生成中断。此位用软件清零，方法是将 CRS_ICR 寄存器中的 ESYNCC 位置 1。

0：未发出预期 SYNC 信号

1：已发出预期 SYNC 信号

位 2 **ERRF**：错误标志 (Error flag)

发生同步或微调错误时，该标志由硬件置 1。它是 TRIMOVF、SYNCMISS 和 SYNCERR 的逻辑或运算的结果。如果 CRS_CR 寄存器中的 ERRIE 位置 1，将生成中断。此位用软件清零，方法是将 CRS_ICR 寄存器中的 ERRC 位置 1，这将清零 TRIMOVF、SYNCMISS 和 SYNCERR 位。

0：未发出同步或微调错误信号

1：已发出同步或微调错误信号

位 1 **SYNCWARNF**：SYNC 警告标志 (SYNC warning flag)

当所测得的频率误差大于或等于 FELIM * 3 但小于 FELIM * 128 时，该标志由硬件置 1。这意味着要补偿频率误差，必须通过两步或多步来调整 TRIM 值。如果 CRS_CR 寄存器中的 SYNCWARNIE 位置 1，将生成中断。此位用软件清零，方法是将 CRS_ICR 寄存器中的 SYNCWARNC 位置 1。

0：未发出 SYNC 警告信号

1：已发出 SYNC 警告信号

位 0 **SYNCOKF**：SYNC 事件正常标志 (SYNC event OK flag)

当所测得的频率误差小于 FELIM * 3 时，该标志由硬件置 1。这意味着无需调整 TRIM 值或只需一步微调便足以补偿频率误差。如果 CRS_CR 寄存器中的 SYNCOKIE 位置 1，将生成中断。此位用软件清零，方法是将 CRS_ICR 寄存器中的 SYNCOKC 位置 1。

0：未发出 SYNC 事件正常信号

1：已发出 SYNC 事件正常信号

9.7.4 CRS 中断标志清零寄存器 (CRS_ICR)

CRS interrupt flag clear register

偏移地址：0x0C

复位值：0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ESYNCC	ERRC	SYNCWARNC	SYNCOKC
												rw	rw	rw	rw

位 31:4 保留，必须保持复位值

位 3 **ESYNCC**：预期 SYNC 清零标志 (Expected SYNC clear flag)

将 1 写入此位时，CRS_ISR 寄存器中 ESYNCF 标志将清零。

位 2 **ERRC**：错误清零标志 (Error clear flag)

将 1 写入此位将清零 TRIMOVF、SYNCMISS 和 SYNCERR 位，进而会清零 CRS_ISR 寄存器中的 ERRF 标志。

位 1 **SYNCWARNC**：SYNC 警告清零标志 (SYNC warning clear flag)

将 1 写入此位时，CRS_ISR 寄存器中 SYNCWARNF 标志将清零。

位 0 **SYNCOKC**：SYNC 事件正常清零标志 (SYNC event OK clear flag)

将 1 写入此位时，CRS_ISR 寄存器中 SYNCOKF 标志将清零。

9.7.5 CRS寄存器映射

表 78. CRS 寄存器映射和复位值

偏移	寄存器	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	**CRS_CR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TRIM[5:0]						SWSYNC	AUTOTRIMEN	CEN	Res.	ESYNCIE	ERRIE	SYNCWARNIE	SYNCOKIE
	Reset value																			1	0	0	0	0	0	0	0	0	0	0	0	0	0
0x04	**CRS_CFGR**	SYNCPOL	Res.	SYNC SRC [1:0]		Res.	SYNC DIV [2:0]			FELIM[7:0]								RELOAD[15:0]															
	Reset value	0		1	0		0	0	0	0	0	1	0	0	0	1	0	1	0	1	1	1	0	1	1	0	1	1	1	1	1	1	1
0x08	**CRS_ISR**	FECAP[15:0]																FEDIR	Res.	Res.	Res.	Res.	TRIMOVF	SYNCMISS	SYNCERR	Res.	Res.	Res.	Res.	ESYNCF	ERRF	SYNCWARNF	SYNCOKF
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0					0	0	0					0	0	0	0
0x0C	**CRS_ICR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ESYNCC	ERRC	SYNCWARNC	SYNCOKC
	Reset value																													0	0	0	0

有关寄存器边界地址的信息，请参见*第 100 页的第 2.2.2 节*。

10 硬件信号量 (HSEM)

10.1 硬件信号量简介

硬件信号量模块提供 32 个（32 位）基于寄存器的信号量。

信号量可以用于确保在内核上运行的不同进程之间的同步。HSEM 提供一个非阻塞机制来以原子方式来锁定信号量。其提供了下列功能：

- 可通过以下 2 种方式锁定信号量：
  - 2 步锁定：将 MasterID 和 ProcessID 写入信号量，然后进行读取检查
  - 1 步锁定：从信号量读取 MasterID
- 当信号量被释放时生成中断
  - 每个信号量可生成一个中断
- 信号量清零保护
  - 只有当 MasterID 与 ProcessID 匹配时，才会将信号量清零
- 根据 MasterID 将全局信号量清零

10.2 硬件信号量的主要特性

HSEM 包括下列特性：

- 32 个（32 位）信号量
- 8 位 ProcessID
- 4 位 MasterID
- 1 条中断线
- 锁定指示

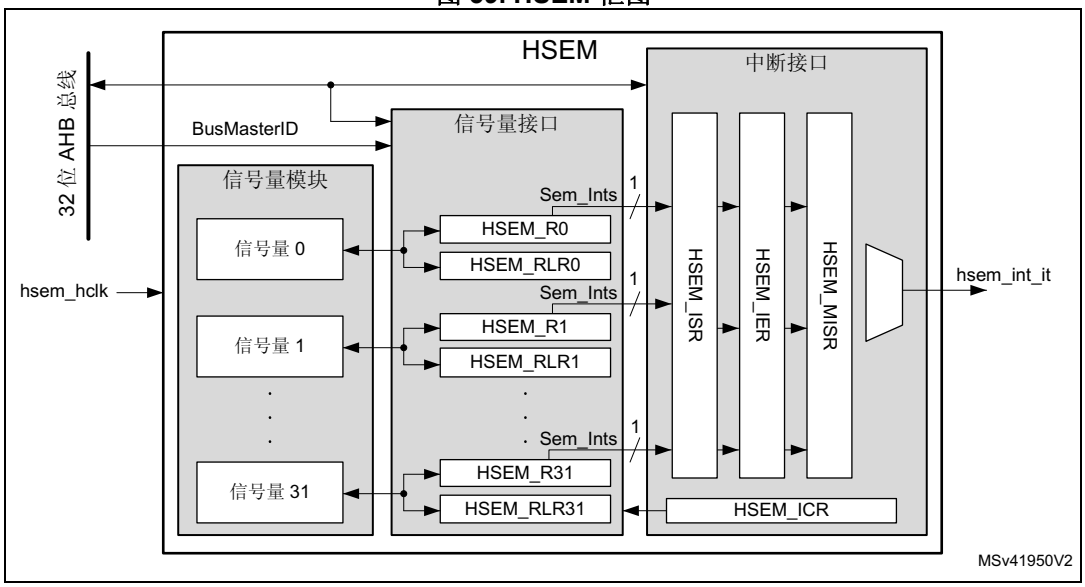
10.3 HSEM 功能说明

10.3.1 HSEM框图

如*图 59* 所示，HSEM 基于以下三个子模块：

- 包含信号量状态和 ID 的信号量模块
- 通过 HSEM_R 和 HSEM_RLR 寄存器对信号量进行 AHB 访问的信号量接口模块
- 通过 HSEM_CnISR、HSEM_CnIER、HSEM_CnMISR 和 HSEM_CnICR 寄存器提供中断控制的中断接口模块

图 59. HSEM 框图



10.3.2 HSEM 内部信号

表 79. HSEM 内部输入/输出信号

信号名称	信号类型	说明
hsem_hclk	数字输入	AHB 时钟
hsem_int_it	数字输出	中断线

10.3.3 HSEM 锁定步骤

共有 2 种锁定步骤

- 2 步（写）锁定
- 1 步（读）锁定

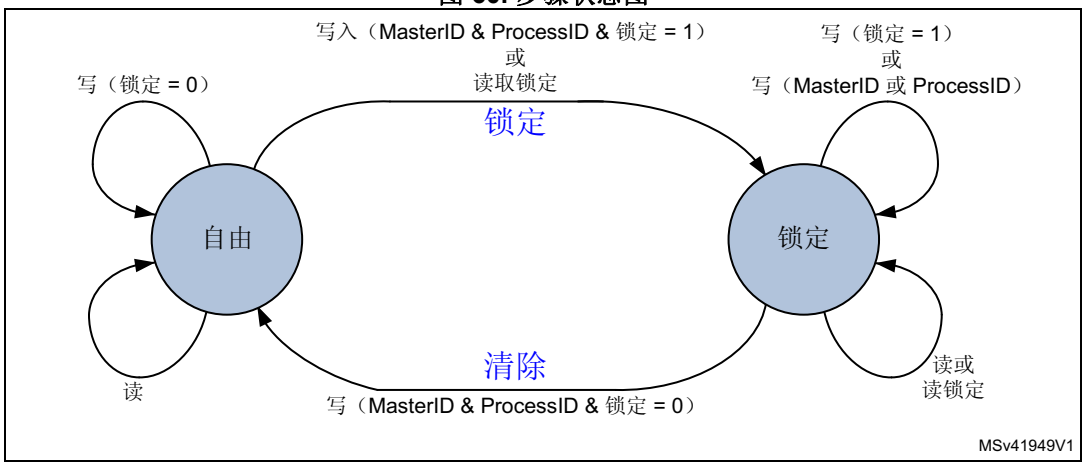
当锁定位为 0 时，信号量空闲，在这种情况下，MasterID 与 ProcessID 也为 0。当锁定位为 1 时，信号量锁定，MasterID 指示是哪一个 AHB 总线主控将其锁定。ProcessID 指示是 AHB 总线主控的哪一个进程锁定了信号量。

当“写”锁定信号量时，MasterID 取自总线主控 ID，ProcessID 取自“写”数据。当“读”锁定信号量时，MasterID 取自主控 ID，ProcessID 为零。1 步（读）锁定不存在 ProcessID。

MasterID 取自 AHB 总线主控 ID。ProcessID 通过相应 AHB 总线主控的固件写入。每个 AHB 总线主控进程必须有唯一的 ProcessID。只有 2 步锁定步骤才存在 ProcessID。

这两种步骤（1 步和 2 步）可以同时使用。

图 60. 步骤状态图



2 步（写）锁定步骤

2 步锁定步骤由两步组成，首先通过“写”以锁定信号量，然后通过“读”HSEM_Rn 寄存器以检查是否成功锁定。

- 将 ProcessID 和 MasterID 以及 LOCK 位 = 1 写入信号量
  （如果在写操作时信号量空闲，则锁定信号量）
- 回读信号量
  （固件检查锁定状态，如果 ProcessID 与 MasterID 匹配，则锁定被确认）
- 否则重试（信号量已由其它 AHB 总线主控或进程锁定）

信号量只能在空闲时才能锁定。

信号量可在 ProcessID 为“0”时锁定。

连续的将锁定位 = 1 写到已锁定信号量的尝试都被忽略。

1 步（读）锁定步骤

1 步步骤由单步组成，通过读取 HSEM_RLRn 寄存器以锁定并检查信号量。

- 读锁定信号量的 MasterID。
- 如果读 MasterID 与 ProcessID 匹配且 ProcessID = 0，则锁定信号量。
  如果 MasterID 与 ProcessID 匹配，但 ProcessID 不为“0”，则意味着来自同一 MasterID 的另一进程使用 2 步（写）步骤锁定了信号量。
- 否则重试（信号量已由其它 AHB 总线主控或进程锁定）

信号量只能在空闲时才能锁定。当读锁定一个空闲信号量时，ProcessID 将为“0”。当读锁定一个锁定信号量时，将返回锁定它的 MasterID 和 ProcessID。所有读锁定（包括锁定信号量的第一次读锁定）都将返回锁定信号量的 MasterID。

如果同一 AHB 总线主控的多个进程使用 1 步步骤，则所有使用同一信号量的进程都将读到相同的状态。如果只有一个进程锁定信号量，则该 AHB 总线主控的所有进程都将读到信号量由其自己和 MasterID 锁定。

10.3.4 HSEM 写/读/读锁定寄存器地址

对于每个信号量，都会提供两个 AHB 寄存器地址，这两个地址位于两个 0x80 存储区中，彼此分开。

在第一个寄存器地址区中，可通过 HSEM_R 寄存器对信号量进行写（锁定/清零）和读操作。

在第二个寄存器地址区中，可通过 HSEM_RLR 寄存器对信号量进行读（锁定）操作。

10.3.5 HSEM 清零步骤

将信号量清零是一个受保护的过程，目的是防止 AHB 总线主控或不具有信号量锁定权限的过程意外将信号量清零。信号量清零步骤由将相应的 MasterID 和 ProcessID 以及锁定位 = 0 写到信号量组成。清零后，信号量锁定位、MasterID 和 ProcessID 均为“0”。

清零后，可能生成一个中断以表示该事件。为此，应使能信号量中断。

清零步骤由一个对信号量 HSEM_R 寄存器的写操作组成。

- 将 ProcessID 和 MasterID 以及锁定位 = 0 写到信号量
- 如果 ProcessID 与 MasterID 匹配，则信号量释放，并且，如果使能了中断，则可能生成一个中断
- 否则，忽略写操作，信号量保持锁定状态并且不会生成任何中断（信号量通过其它 AHB 总线主控或进程锁定）

如果同一 AHB 总线主控的多个过程使用 1 步锁定步骤(ProcessID = 0)，则使用同一信号量的所有进程还将为相应 AHB 总线主控的其它进程清零信号量。

10.3.6 HSEM MasterID 信号量清零

AHB 总线主控锁定的所有信号量可使用 HSEM_CR 寄存器一次全部清零。

清零 AHB 总线主控锁定的全部信号量的步骤如下：

- 写入 MasterID 和正确的键值。具有匹配 MasterID 的所有锁定信号量都将清零（设置为释放），并且可生成中断（使能时）。

该步骤可在 AHB 总线主控非正常工作时使用，在这种情况下，AHB 总线主控可释放锁定的信号量，方法是向 HSEM_CR 寄存器写入 MasterID 和正确的键值。这将清零具有匹配 MasterID 的所有锁定信号量。

可为释放的信号量生成中断。为此，应在 HSEM_CnIER 寄存器中使能信号量中断。

10.3.7 HSEM 中断

hsem_int_it 中断线允许 32 个信号量中的每个信号量生成中断。

该中断线提供以下功能：

- 每个信号量的中断使能
- 每个信号量的中断清零
- 每个信号量的中断状态
- 每个信号量的屏蔽中断状态

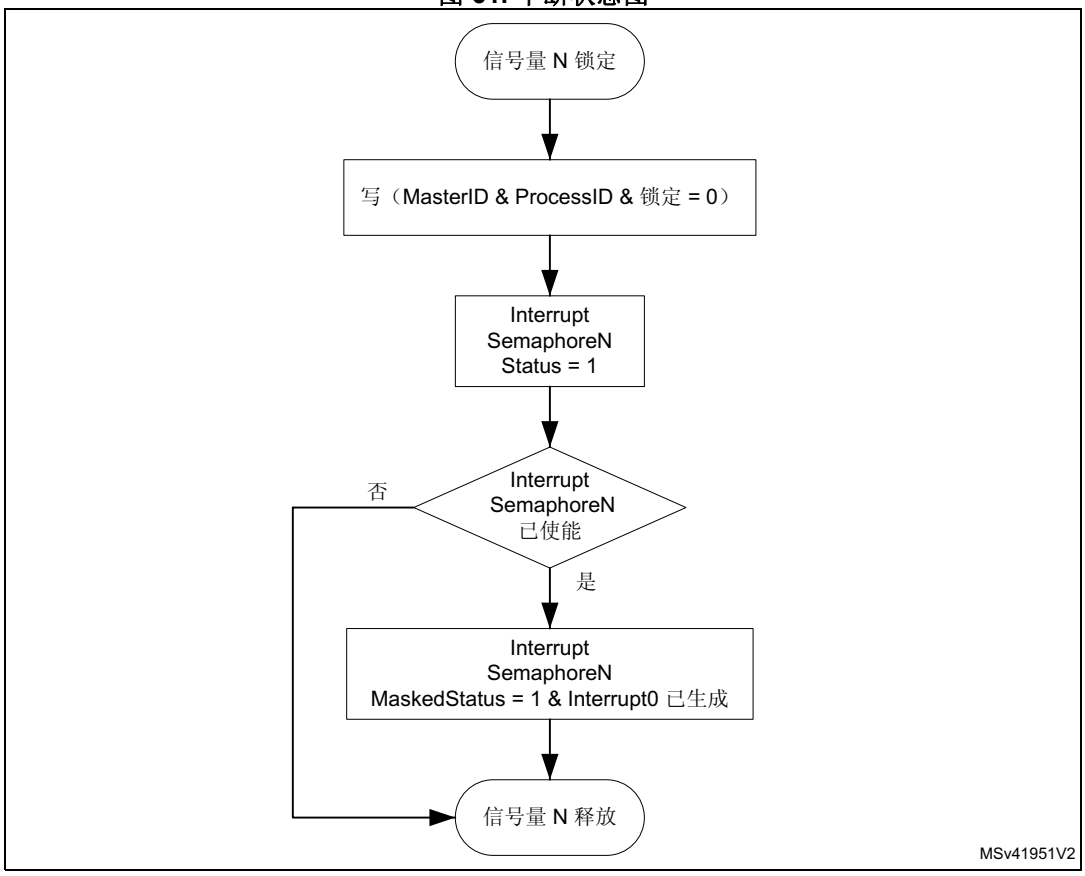
可以通过中断使能 (HSEM_CnIER) 使能信号量影响中断线。已禁止（已屏蔽）信号量中断不会设置该信号量的屏蔽中断状态，也不会在中断线上生成中断。

中断清零 (HSEM_CnICR) 将清零中断线相关信号量的中断状态和屏蔽中断状态。

中断状态 (HSEM_CnISR) 会镜像使能前中断线的信号量中断状态。

屏蔽中断状态 (HSEM_CnMISR) 仅会镜像中断线上已使能信号量中断的信号量中断状态。已使能信号量的所有已屏蔽中断状态均需要清零，才能清零中断线。

图 61. 中断状态图



下文将介绍信号量释放时获得中断的步骤。

尝试锁定信号量 N

- 如果信号量锁定成功，则无需中断。
- 如果信号量锁定失败：
  需在 HSEM_CnICR 中为中断线 hsem_intn_it 清零挂起的信号量 N 的中断状态。
  再次重新尝试锁定信号量 N：
  - 如果信号量锁定成功，则无需中断（在第一次尝试锁定信号量到清零信号量中断状态的过程中，已释放信号量）。
  - 如果信号量锁定失败，需在 HSEM_CnIER 中为中断线 hsem_intnn_it 使能信号量 N 的中断。

在出现信号量 N 释放中断后，尝试锁定信号量 N

- 如果信号量锁定成功：
  需在 HSEM_CnIER 中为中断线 hsem_intn_it 禁止信号量 N 的中断，并在 HSEM_CnICR 中为中断线 hsem_intn_it 清零挂起的信号量 N 的中断状态。
- 如果信号量 N 锁定失败：
  需在 HSEM_CnICR 中为中断线 hsem_intn_it 清零挂起的信号量 N 的中断状态。
  再次尝试锁定信号量 N：
  - 如果信号量锁定成功（在第一次尝试锁定信号量到清零信号量中断状态的过程中，已释放信号量）：
    需在 HSEM_CnIER 中为中断线 hsem_intn_it 禁止信号量中断。
  - 如果信号量锁定失败，需等待信号量释放中断。

注： *中断不会锁定信号量。中断后，AHB 总线主控或过程仍必须执行锁定该步骤来锁定信号量。*

10.3.8 AHB 总线主控 ID 验证

HSEM 仅允许经授权的 AHB 总线主控 ID 锁定和解锁信号量。

- 对信号量 HSEM_Rn 寄存器进行 AHB 总线主控 2 步锁定写入访问时，将通过有效的总线主控 ID 进行核验。
  - 如果通过未经授权的 AHB 总线主控 ID 进行访问，则相关访问将遭到丢弃，不会锁定信号量。
- 对信号量 HSEM_RLRn 寄存器进行 AHB 总线主控 1 步锁定读取访问时，将通过有效的总线主控 ID 进行核验。
  - 如果对 HSEM_RLRn 进行未经授权的 AHB 总线主控 ID 读取访问，则将返回以下读取数据，具体取决于信号量状态：
    - 当信号量释放时，将返回全“0”
    - 当信号量之前已锁定时，将返回 HSEM_RLRn 数据
- 对信号量 HSEM_CR 寄存器进行 MasterID 信号量清零写入访问时，将通过有效的总线主控 ID 进行核验。只有有效的总线主控 ID 可写入 HSEM_CR 寄存器，清零任一该 MasterID 信号量。
  - 如果通过未经授权的 AHB 总线主控 ID 进行访问，则相关访问将遭到丢弃，不会清零 MasterID 信号量。

表 80 详细介绍了总线主控/CPU 与 MASTERID 之间的关系。

表 80. 经授权的 AHB 总线主控 ID

总线主控 0 (CPU)
MASTERID = 3

注：　允许通过未经授权的 AHB 总线主控 ID 对其它寄存器进行访问。

10.4 HSEM寄存器

应按字格式访问寄存器。将忽略字节和半字访问，这些访问不会影响信号量。字节和半字读取访问将始终返回 0。字节和半字访问将不会生成总线错误。

10.4.1 HSEM 寄存器 (HSEM_R0 - HSEM_R31)

HSEM register

偏移地址：0x000 + N x 4（其中 N = 信号量编号 0 - 31）

复位值：0x0000 0000

HSEM_R0 - HSEM_R31 应用于执行 2 步写入锁定和回读。仅允许通过经授权的 AHB 总线主控 ID 进行写入访问。将丢弃通过未经授权的 AHB 总线主控 ID 进行的写入访问。

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

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

位 31 **LOCK**：锁定指示 (Lock indication)。
该位可通过固件写入和读取。
0：写入时释放信号量（仅当 MASTERID 与 PROCID 匹配时），读取时已释放信号量。
1：写入时尝试锁定信号量，读取时已锁定信号量。

位 30:12 保留，必须保持复位值

位 11:8 **MASTERID**：信号量 MasterID (Semaphore MasterID)。
通过固件写入，当信号量释放且 LOCK 位写入 1 时，只有在写入信号量的 AHB 总线主控的总线 ID 匹配时，才会写入 MASTERID。
当信号量清零时（LOCK 位写入 0 且 AHB 总线主控 ID 与 MASTERID 匹配），MasterID 将清零。
当信号量清零时（LOCK 位写入 0 且 AHB 总线主控 ID 与 MASTERID 不匹配），MasterID 将不受影响。
当 LOCK 位已为 1（信号量已锁定）时写入，MasterID 将不受影响。
读取操作将返回存储的 MasterID 值。

位 7:0 **PROCID**：信号量 ProcessID (Semaphore ProcessID)。
通过固件写入，当信号量释放且锁定位写入 1 时，ProcessID 将设置为写入的数据。
当信号量清零（LOCK 位写入 0）时，ProcessID 将清零。
当 LOCK 位已为 1（信号量已锁定）时写入，ProcessID 将不受影响。
读取操作将返回编程的 ProcessID 值。

10.4.2 HSEM 读取锁定寄存器 (HSEM_RLR0 - HSEM_RLR31)

HSEM Read lock register

偏移地址：0x080 + N x 4（其中 N = 信号量编号 0 - 31）

复位值：0x0000 0000

访问与 HSEM_R0 - HSEM_R31 相同的物理位。HSEM_RLR0 - HSEM_RLR31 应用于执行 1 步读取锁定。仅允许通过经授权的 AHB 总线主控 ID 进行读取访问。将丢弃通过未经授权的 AHB 总线主控 ID 进行的读取访问。

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

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

位 31 **LOCK**：锁定指示 (Lock indication)。

该位仅由固件在该地址处读取。通过有效的总线主控 ID 读取时，将始终返回 1。

当信号量释放且固件执行读取操作时，硬件会将信号量设置为锁定状态。

当信号量锁定且固件执行读取操作时，LOCK 位不受影响。

0：信号量释放。

1：信号量锁定。

位 30:12 保留，必须保持复位值

位 11:8 **MASTERID**：信号量 MasterID (Semaphore MasterID)。

该字段仅由固件在该地址处读取。

如果在信号量释放时读取，则硬件会将 MasterID 设置为读取信号量的 AHB 总线主控 ID。将读取锁定信号量的 AHB 总线主控的 MasterID。

如果在信号量锁定时读取，则将返回已锁定信号量的 AHB 总线主控的 MasterID。

位 7:0 **PROCID**：信号量 ProcessID (Semaphore ProcessID)。

该字段仅由固件在该地址处读取。

如果在信号量释放时读取，则将返回 0。

如果在信号量锁定时读取，则将返回已锁定信号量的进程的 ProcessID。

10.4.3 HSEM 中断使能寄存器 (HSEM_CnIER)

HSEM Interrupt enable register

偏移地址：0x100

复位值：0x0000 0000

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

位 31:0 **ISE[31:0]**：中断信号量 n 使能位 (Interrupt semaphore n enable bit)。
通过固件读取和写入该位。
0：禁止（屏蔽）信号量 n 的中断生成。
1：使能（不屏蔽）信号量 n 的中断生成。

10.4.4 HSEM 中断清零寄存器 (HSEM_CnICR)

HSEM Interrupt clear register

偏移地址：0x104

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ISC[31:16]															
wr0	wr0	wr0	wr0	wr0	wr0	wr0	wr0	wr0	wr0	wr0	wr0	wr0	wr0	wr0	wr0
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ISC[15:0]															
wr0	wr0	wr0	wr0	wr0	wr0	wr0	wr0	wr0	wr0	wr0	wr0	wr0	wr0	wr0	wr0

位 31:0 **ISC[31:0]**：中断信号量 n 清零位 (Interrupt semaphore n clear bit)。
该位通过固件写入，始终读为 0。
0：中断信号量 n 的状态和屏蔽状态不受影响。
1：中断信号量 n 的状态和屏蔽状态清零。

10.4.5 HSEM 中断状态寄存器 (HSEM_CnISR)

HSEM Interrupt status register

偏移地址：0x108

复位值：0x0000 0000

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

位 31:0 **ISF[31:0]**：使能（屏蔽）之前的中断信号量 n 状态位 (Interrupt semaphore n status bit before enable (mask))。

该位通过硬件置 1，只能通过固件读取。

该位通过固件写入相应的 HSEM_CnICR 位来清零。

0：中断信号量 n 的状态，中断未挂起。

1：中断信号量 n 的状态，中断挂起。

10.4.6 HSEM 屏蔽中断状态寄存器 (HSEM_CnMISR)

HSEM Masked interrupt status register

偏移地址：0x10C

复位值：0x0000 0000

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

位 31:0 **MISF[31:0]**：使能（屏蔽）之后的屏蔽中断信号量 n 状态位 (masked interrupt semaphore n status bit after enable (mask))。

该位通过硬件置 1，只能通过固件读取。

该位通过固件写入相应的 HSEM_CnICR 位来清零。

0：屏蔽后中断信号量 n 的状态未挂起。

1：屏蔽后中断信号量 n 的状态挂起。

10.4.7 HSEM 清零寄存器 (HSEM_CR)

HSEM Clear register

偏移地址：0x140

复位值：0x0000 0000

仅允许通过经授权的 AHB 总线主控 ID 进行写入访问。将丢弃通过未经授权的 AHB 总线主控 ID 进行的写入访问。

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

位 31:16 **KEY**：信号量清零键 (Semaphore clear Key)。
该位可通过固件写入， 将始终读为 0。
键值与 HSEM_KEYR.KEY 不匹配时，信号量不受影响。
键值与 HSEM_KEYR.KEY 匹配时，与 MASTERID 匹配的所有信号量都将清零为释放状态。

位 15:12 保留，必须保持复位值。

位 11:8 **MASTERID**：要清零的信号量 MasterID (MasterID of semaphores to be cleared)。
该字段可通过固件写入，始终读为 0。
指示在写入 HSEM_CR 时清零信号量的 MasterID。

位 7:0 保留，必须保持复位值

10.4.8 HSEM 中断清零寄存器 (HSEM_KEYR)

HSEM Interrupt clear register

偏移地址：0x144

复位值：0x0000 0000

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

位 31:16 **KEY**：信号量清零键 (Semaphore Clear Key)。
该位可通过固件写入和读取。
清零信号量时要匹配的键值。

位 15:0 保留，必须保持复位值

10.4.9 HSEM寄存器映射

表 81. HSEM 寄存器映射和复位值

偏移	寄存器	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x000	**HSEM_R0**	LOCK	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MASTERID[3:0]				PROCID[7:0]							
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x004	**HSEM_R1**	LOCK	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MASTERID[3:0]				PROCID[7:0]							
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
⋮																																	
0x07C	**HSEM_R31**	LOCK	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MASTERID[3:0]				PROCID[7:0]							
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x080	**HSEM_RLR0**	LOCK	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MASTERID[3:0]				PROCID							
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x084	**HSEM_RLR1**	LOCK	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MASTERID[3:0]				PROCID[7:0]							
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
⋮																																	
0x0FC	**HSEM_RLR31**	LOCK	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MASTERID[3:0]				PROCID[7:0]							
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x100	**HSEM_C1IER**	ISE[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x104	**HSEM_C1ICR**	ISC[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x108	**HSEM_C1ISR**	ISF[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x10C	**HSEM_C1MISR**	MISF[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x140	**HSEM_CR**	KEY																Res.	Res.	Res.	Res.	MASTERID[3:0]				Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x144	**HSEM_KEYR**	KEY																Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

有关寄存器边界地址的信息，请参见*第 100 页的第 2.2.2 节*。

11 通用 I/O (GPIO)

11.1 简介

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

11.2 GPIO 主要特性

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

11.3 GPIO 功能描述

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

- 输入浮空
- 输入上拉
- 输入下拉
- 模拟
- 具有上拉或下拉功能的开漏输出
- 具有上拉或下拉功能的推挽输出
- 具有上拉或下拉功能的复用功能推挽
- 具有上拉或下拉功能的复用功能开漏

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

图 62 和*图 63* 分别显示了标准和 5 V 容忍 I/O 端口位的基本结构。*表 82* 给出了可能的端口位配置方案。

图 62. I/O 端口位的基本结构

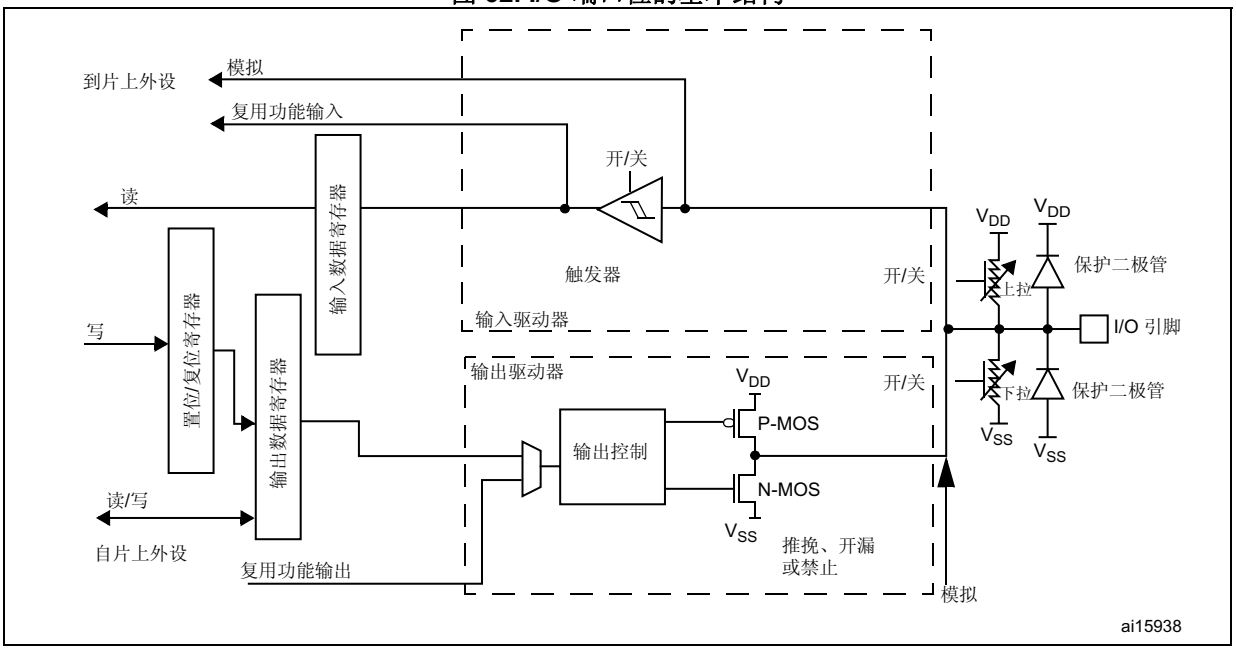
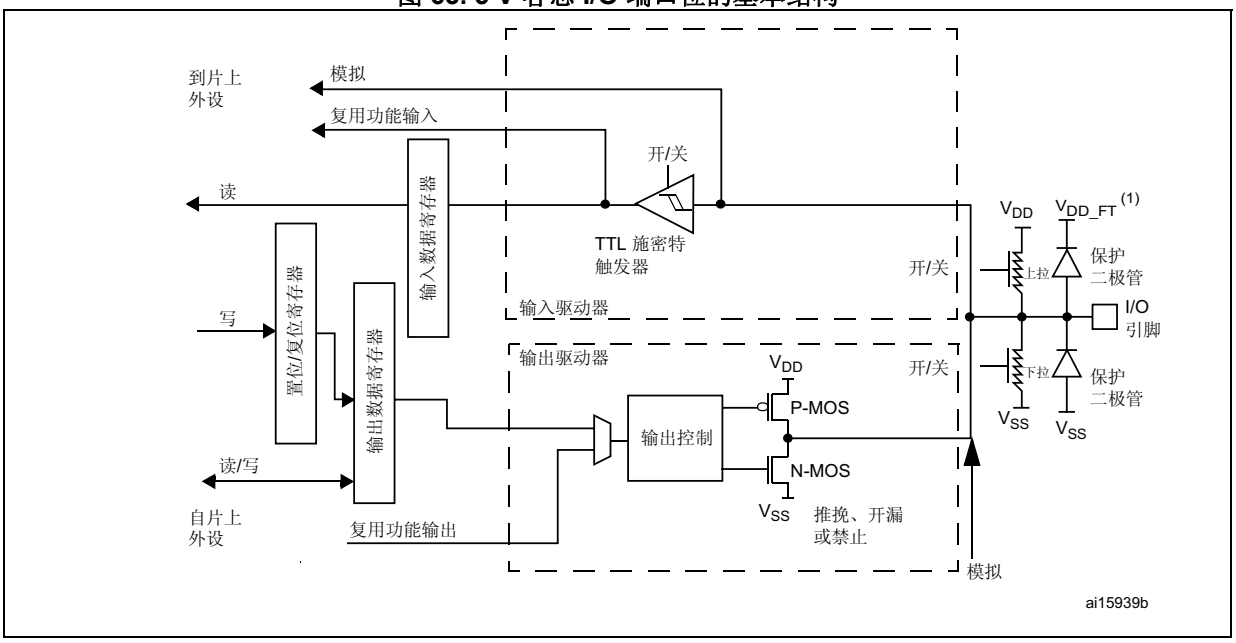


图 63. 5 V 容忍 I/O 端口位的基本结构



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

表 82. 端口位配置表(1)

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

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

11.3.1 通用 I/O (GPIO)

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

复位后，调试引脚处于复用功能上拉/下拉状态：

- PA15：JTDI 处于上拉状态
- PA14：JTCK/SWCLK 处于下拉状态
- PA13：JTMS/SWDAT 处于上拉状态
- PB4：NJTRST 处于上拉状态
- PB3：JTDO 处于浮空状态

当引脚配置为输出后，写入到输出数据寄存器 (GPIOx_ODR) 的值将在 I/O 引脚上输出。可以在推挽模式下或开漏模式下使用输出驱动器（仅驱动低电平，高电平为高阻态）。

输入数据寄存器 (GPIOx_IDR) 每个 AHB 时钟周期捕获一次 I/O 引脚的数据。

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

11.3.2 I/O 引脚复用功能复用器和映射

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

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

- 复位后，复用器选择为复用功能 0 (AF0)。在复用模式下通过 GPIOx_MODER 寄存器配置 I/O。
- 器件数据手册中详细说明了每个引脚的特定复用功能分配。
- Cortex-M7 FPU EVENTOUT 映射到 AF15。

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

要在指定配置下使用 I/O，用户必须按照以下步骤操作：

- **调试功能：**每个器件复位后，立即将这些引脚分配为可由调试主机使用的复用功能引脚。
- **系统功能：**必须将 MCOx 引脚配置为复用功能模式。
- **GPIO：**在 GPIOx_MODER 寄存器中将所需 I/O 配置为输出、输入或模拟通道。
- **外设复用功能：**
  - 在 GPIOx_AFRL 或 GPIOx_AFRH 寄存器中，将 I/O 连接到所需的 AFx。
  - 通过 GPIOx_OTYPER、GPIOx_PUPDR 和 GPIOx_OSPEEDER 寄存器，分别选择类型、上拉/下拉以及输出速度。
  - 在 GPIOx_MODER 寄存器中将所需 I/O 配置为复用功能。

- **其它功能：**
  - 对于 ADC 和 DAC，在 GPIOx_MODER 寄存器中将所需 I/O 配置为模拟模式，并在 ADC 和 DAC 寄存器中配置所需功能。
  - 对于 RTC_OUT、RTC_TS、RTC_TAMPx、WKUPx 和振荡器的其它功能，在相关的 RTC、PWR 和 RCC 寄存器中配置所需功能。这些功能优先于标准 GPIO 寄存器中的配置。有关 RTC 的 I/O 控制的详细信息，请参见*第 1745 页的第 46.3 节：RTC 功能说明*。
- EVENTOUT
  - 配置用于输出内核 EVENTOUT 信号的 I/O 引脚（通过将其连接到 AF15）。

有关复用功能 I/O 引脚映射的详细信息，请参见器件数据手册中的“复用功能映射”表。

11.3.3 I/O 端口控制寄存器

每个 GPIO 端口有 4 个 32 位存储器映射的控制寄存器（GPIOx_MODER、GPIOx_OTYPER、GPIOx_OSPEEDR、GPIOx_PUPDR），可配置多达 16 个 I/O。GPIOx_MODER 寄存器用于选择 I/O 模式（输入、输出、AF、模拟）。GPIOx_OTYPER 和 GPIOx_OSPEEDR 寄存器用于选择输出类型（推挽或开漏）和速度。无论采用哪种 I/O 方向，GPIOx_PUPDR 寄存器都用于选择上拉/下拉。

11.3.4 I/O 端口数据寄存器

每个 GPIO 都具有 2 个 16 位数据寄存器：输入和输出数据寄存器（GPIOx_IDR 和 GPIOx_ODR）。GPIOx_ODR 用于存储待输出数据，可对其进行读/写访问。通过 I/O 输入的数据存储到输入数据寄存器 (GPIOx_IDR) 中，它是一个只读寄存器。

有关寄存器说明的详细信息，请参见*第 11.4.5 节：GPIO 端口输入数据寄存器 (GPIOx_IDR) (x = A..K)*和*第 11.4.6 节：GPIO 端口输出数据寄存器 (GPIOx_ODR) (x = A..K)*。

11.3.5 I/O 数据位操作

置位复位寄存器 (GPIOx_BSRR) 是一个 32 位寄存器，它允许应用程序在输出数据寄存器 (GPIOx_ODR) 中对各个单独的数据位执行置位和复位操作。置位复位寄存器的大小是 GPIOx_ODR 的二倍。

GPIOx_ODR 中的每个数据位对应于 GPIOx_BSRR 中的两个控制位：BS(i) 和 BR(i)。当写入 1 时，BS(i) 位会**置位**对应的 ODR(i) 位。当写入 1 时，BR(i) 位会**复位** ODR(i) 对应的位。

在 GPIOx_BSRR 中向任何位写入 0 都不会对 GPIOx_ODR 中的对应位产生任何影响。如果在 GPIOx_BSRR 中同时尝试对某个位执行置位和复位操作，则置位操作优先。

使用 GPIOx_BSRR 寄存器更改 GPIOx_ODR 中各个位的值是一个“单次”操作，不会锁定 GPIOx_ODR 位。随时都可以直接访问 GPIOx_ODR 位。GPIOx_BSRR 寄存器提供了一种执行原子按位处理的方法。

在对 GPIOx_ODR 进行位操作时，软件无需禁止中断：在一次原子 AHB 写访问中，可以修改一个或多个位。

11.3.6 GPIO 锁定机制

通过将特定的写序列应用到 GPIOx_LCKR 寄存器，可以冻结 GPIO 控制寄存器。冻结的寄存器包括 GPIOx_MODER、GPIOx_OTYPER、GPIOx_OSPEEDR、GPIOx_PUPDR、GPIOx_AFRL 和 GPIOx_AFRH。

要对 GPIOx_LCKR 寄存器执行写操作，必须应用特定的写/读序列。当正确的 LOCK 序列应用到此寄存器的第 16 位后，会使用 LCKR[15:0] 的值来锁定 I/O 的配置（在写序列期间，LCKR[15:0] 的值必须相同）。将 LOCK 序列应用到某个端口位后，在执行下一次 MCU 复位或外设复位之前，将无法对该端口位的值进行修改。每个 GPIOx_LCKR 位都会冻结控制寄存器（GPIOx_MODER、GPIOx_OTYPER、GPIOx_OSPEEDR、GPIOx_PUPDR、GPIOx_AFRL 和 GPIOx_AFRH）中的对应位。

LOCK 序列（参见*第 11.4.8 节：GPIO 端口配置锁定寄存器 (GPIOx_LCKR) (x = A..K)*）只能通过对 GPIOx_LCKR 寄存器进行字（32 位长）访问的方式来执行，因为 GPIOx_LCKR 的第 16 位必须与 [15:0] 位同时置位。

有关详细信息，请参见*第 11.4.8 节：GPIO 端口配置锁定寄存器 (GPIOx_LCKR) (x = A..K)* 中的 LCKR 寄存器说明。

11.3.7 I/O 复用功能输入/输出

有两个寄存器可用来从每个 I/O 可用的复用功能输入/输出中进行选择。借助这些寄存器，用户可根据应用程序的要求将某个复用功能连接到其它某个引脚。

这意味着可使用 GPIOx_AFRL 和 GPIOx_AFRH 复用功能寄存器在每个 GPIO 上复用多个可用的外设功能。这样一来，应用程序可为每个 I/O 选择任何一个可用功能。由于 AF 选择信号由复用功能输入和复用功能输出共用，所以只需为指定 I/O 的复用功能输入/输出选择一个通道即可。

要了解在每个 GPIO 引脚上复用了哪些功能，请参见器件数据手册。

11.3.8 外部中断线/唤醒线

所有端口都具有外部中断功能。要使用外部中断线，必须将端口配置为输入模式。请参见*第 20 节：扩展中断和事件控制器 (EXTI)* 和*第 20.3 节：EXTI 功能说明*。

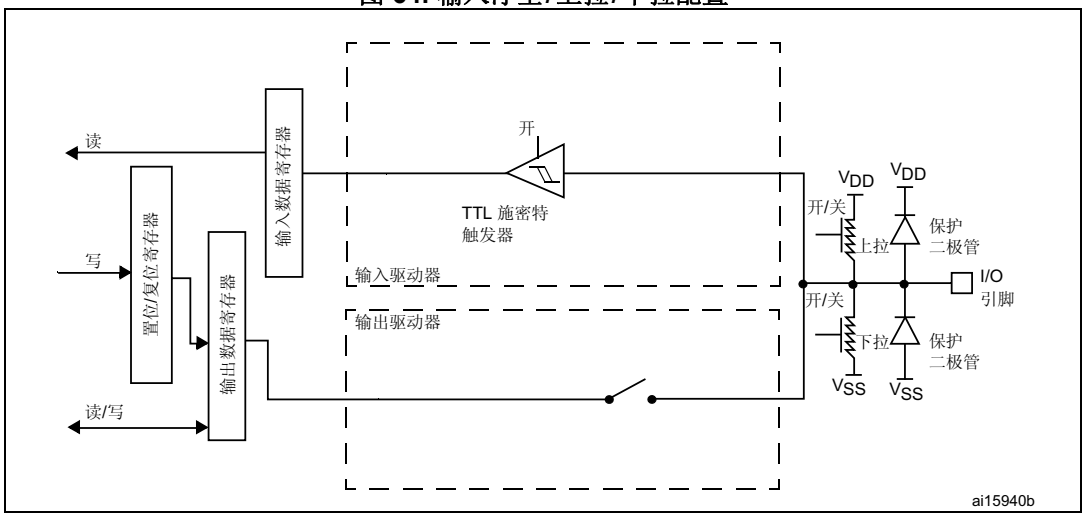
11.3.9 输入配置

对 I/O 端口进行编程作为输入时：

- 输出缓冲器被禁止
- 施密特触发器输入被打开
- 根据 GPIOx_PUPDR 寄存器中的值决定是否打开上拉和下拉电阻
- 输入数据寄存器每个 AHB 时钟周期对 I/O 引脚上的数据进行一次采样
- 对输入数据寄存器的读访问可获取 I/O 状态

图 64 说明了 I/O 端口位的输入配置。

图 64. 输入浮空/上拉/下拉配置



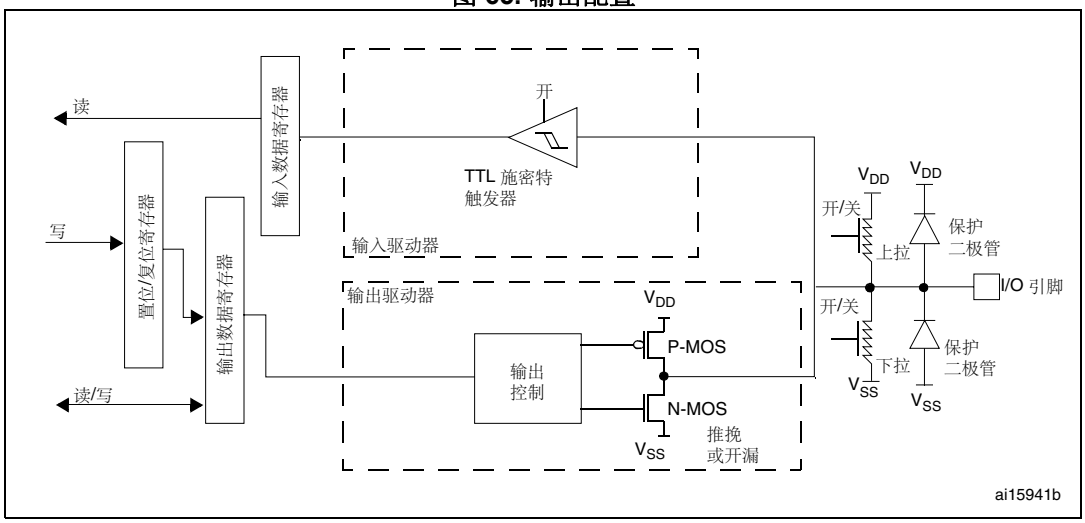
11.3.10 输出配置

对 I/O 端口进行编程作为输出时：

- 输出缓冲器被打开：
  - 开漏模式：输出寄存器中的“0”可激活 N-MOS，而输出寄存器中的“1”会使端口保持高阻态 (Hi-Z)（P-MOS 始终不激活）
  - 推挽模式：输出寄存器中的“0”可激活 N-MOS，而输出寄存器中的“1”可激活 P-MOS
- 施密特触发器输入被打开
- 根据 GPIOx_PUPDR 寄存器中的值决定是否打开上拉和下拉电阻
- 输入数据寄存器每个 AHB 时钟周期对 I/O 引脚上的数据进行一次采样
- 对输入数据寄存器的读访问可获取 I/O 状态
- 对输出数据寄存器的读访问可获取最后的写入值

图 65 说明了 I/O 端口位的输出配置。

图 65. 输出配置



11.3.11 I/O 补偿单元

该单元用于控制 I/O 通信斜率 (t_{fall}/t_{rise})，从而降低 I/O 端口噪声对电源的影响。

该单元分为两个块：

- 第一个块提供当前 PVT 的最佳代码。当 SYSCFG_CCSR 的 READY 标志置 1 时，可读取该块中存储的代码。
- 第二个块控制 I/O 斜率。用户选择要应用的代码并通过软件对其进行编程。

I/O 补偿单元具有 2 个电压范围：1.62 V 到 2.0 V 和 2.7 V 到 3.6 V。

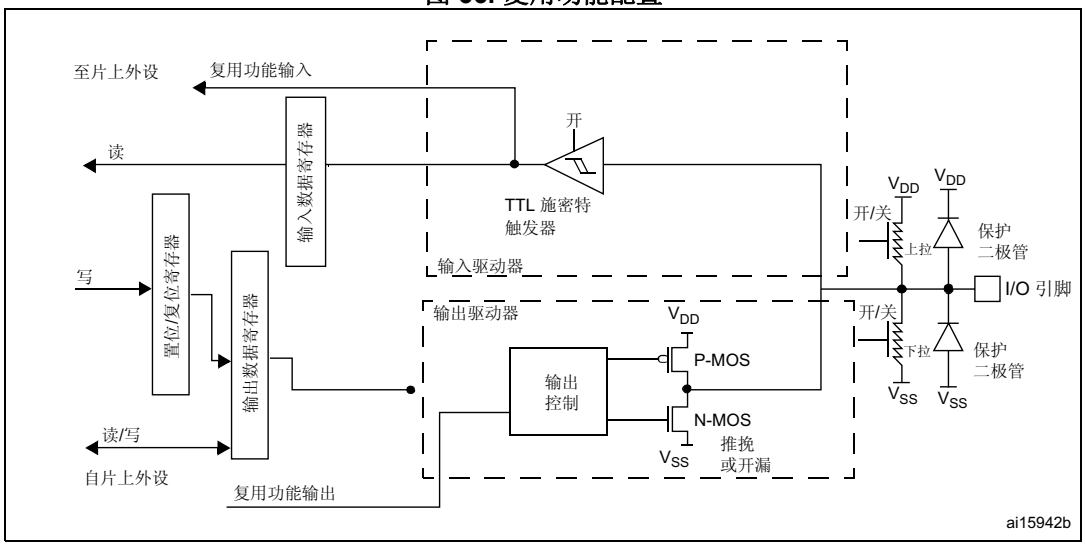
11.3.12 复用功能配置

对 I/O 端口进行编程作为复用功能时：

- 可将输出缓冲器配置为开漏或推挽模式
- 输出缓冲器由来自外设的信号驱动（发送器使能和数据）
- 施密特触发器输入被打开
- 根据 GPIOx_PUPDR 寄存器中的值决定是否打开弱上拉电阻和下拉电阻
- 输入数据寄存器每个 AHB 时钟周期对 I/O 引脚上的数据进行一次采样
- 对输入数据寄存器的读访问可获取 I/O 状态

图 66 说明了 I/O 端口位的复用功能配置。

图 66. 复用功能配置



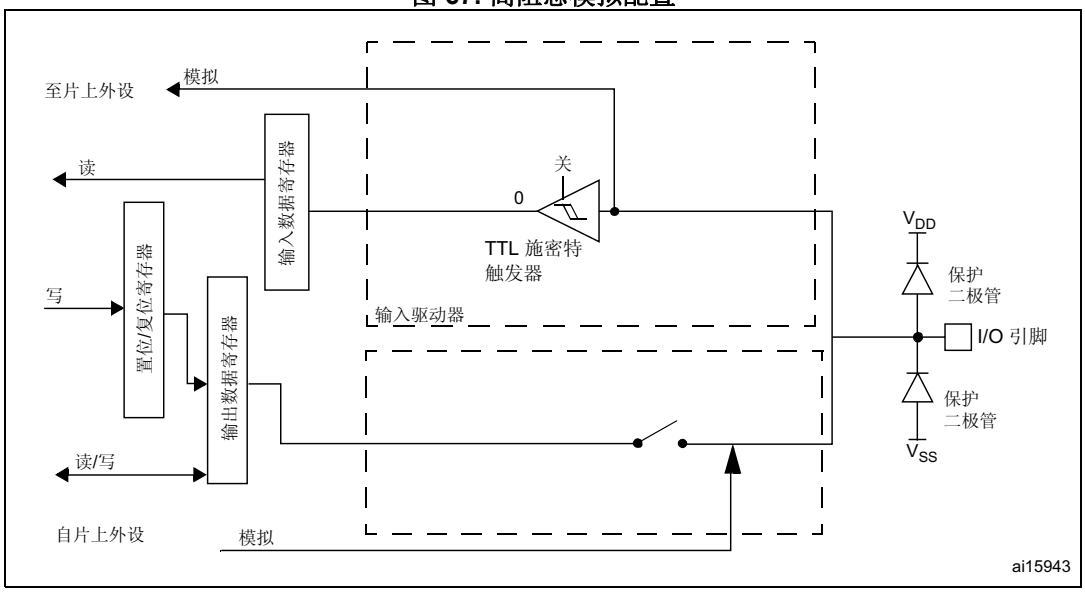
11.3.13 模拟配置

对 I/O 端口进行编程作为模拟配置时：

- 输出缓冲器被禁止
- 施密特触发器输入停用，I/O 引脚的每个模拟输入的功耗变为零。施密特触发器的输出被强制处理为恒定值 (0)
- 弱上拉和下拉电阻被硬件关闭
- 对输入数据寄存器的读访问值为“0”

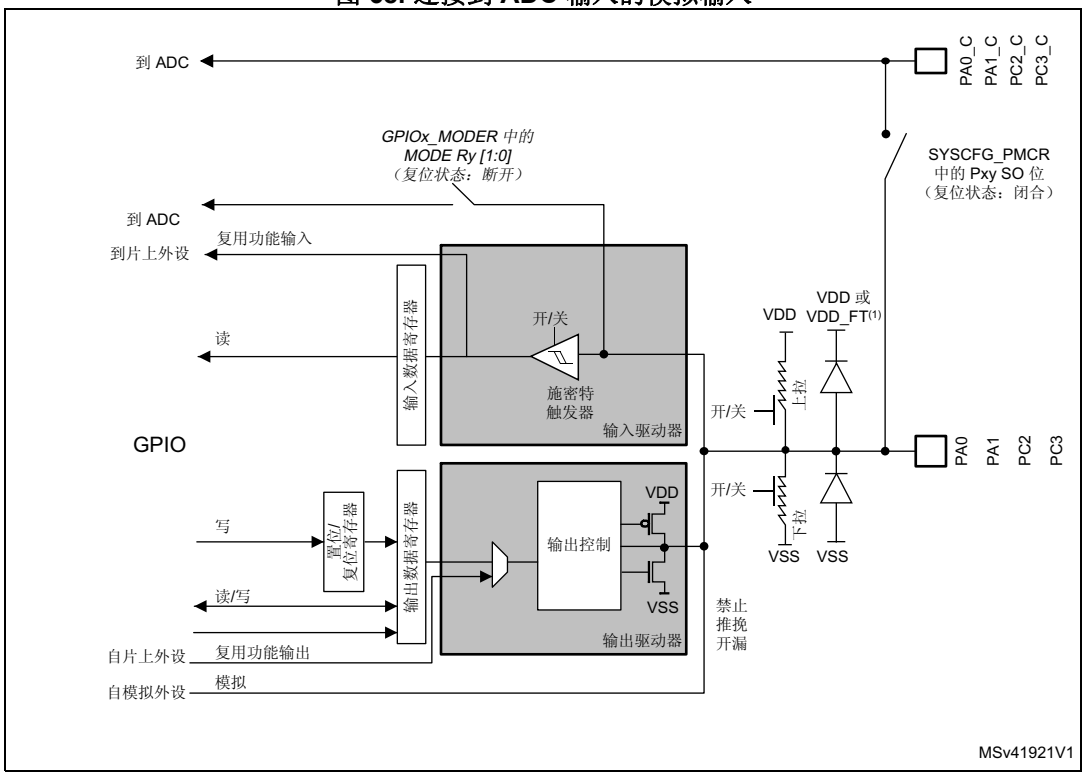
图 67 说明了 I/O 端口位的高阻态模拟输入配置。

图 67. 高阻态模拟配置



某些引脚/焊球直接连接到 PA0_C、PA1_C、PC2_C 和 PC3_C ADC 模拟输入（请参见 *图 68*）：Pxy_C 和 Pxy 引脚/焊球之间存在一条直接路径，通过模拟开关实现（有关如何配置模拟开关的详细信息，请参见*第 12.3.1 节：SYSCFG 外设模式配置寄存器 (SYSCFG_PMCR)*）。

图 68. 连接到 ADC 输入的模拟输入



1. VDD_FT 是和 5V 容忍 I/O 相关的电位， 与 VDD 不同。

11.3.14 将 HSE 或 LSE 振荡器引脚用作 GPIO

当 HSE 或 LSE 振荡器关闭（复位后的默认状态）时，可将相关的振荡器引脚用作常规 GPIO。

当 HSE 或 LSE 振荡器打开（通过设置 RCC_CSR 寄存器中的 HSEON 或 LSEON 位）时，振荡器会控制与其相关联的引脚，这些引脚的 GPIO 配置不起作用。

将振荡器配置为用户外部时钟模式时，仅为时钟输入保留 OSC_IN 或 OSC32_IN 引脚，OSC_OUT 或 OSC32_OUT 引脚仍可用作常规 GPIO。

11.3.15 在备份电源域中使用 GPIO 引脚

当内核电源域掉电时（器件进入待机模式时），PC13/PC14/PC15/PI8 GPIO 功能会丢失。在这种情况下，如果不通过 RTC 配置旁路其 GPIO 配置，这些引脚将被设置为模拟输入模式。

11.4 GPIO 寄存器

本节对 GPIO 寄存器进行了详细介绍。

有关寄存器位、寄存器偏移地址和复位值的汇总，请参见*表 83*。

可按字、半字或字节模式写入外设寄存器。

11.4.1 GPIO 端口模式寄存器 (GPIOx_MODER) (x =A..K)

GPIO port mode register

偏移地址：0x00

复位值：

- 0xABFF FFFF（端口 A）
- 0xFFFF FEBF（端口 B）
- 0xFFFF FFFF（其他端口）

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MODER15[1:0]		MODER14[1:0]		MODER13[1:0]		MODER12[1:0]		MODER11[1:0]		MODER10[1:0]		MODER9[1:0]		MODER8[1:0]	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MODER7[1:0]		MODER6[1:0]		MODER5[1:0]		MODER4[1:0]		MODER3[1:0]		MODER2[1:0]		MODER1[1:0]		MODER0[1:0]	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 2y+1:2y **MODERy[1:0]**：端口 x 配置位 (Port x configuration bits) (y = 0..15)
这些位通过软件写入，用于配置 I/O 模式。
00：输入模式（复位状态）
01：通用输出模式
10：复用功能模式
11：模拟模式

11.4.2 GPIO 端口输出类型寄存器 (GPIOx_OTYPER) (x = A..K)

GPIO port output type register

偏移地址：0x04

复位值：0x0000 0000

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

位 31:16 保留，必须保持复位值。

位 15:0 **OTy**：端口 x 配置位 (Port x configuration bits) (y = 0..15)
这些位通过软件写入，用于配置 I/O 输出类型。
0：推挽输出（复位状态）
1：开漏输出

11.4.3 GPIO 端口输出速度寄存器 (GPIOx_OSPEEDR) (x = A..K)

GPIO port output speed register

偏移地址：0x08

复位值：

- 0x0C00 0000（端口 A）
- 0x0000 00C0（端口 B）
- 0x0000 0000（其它端口）

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
OSPEEDR15 [1:0]		OSPEEDR14 [1:0]		OSPEEDR13 [1:0]		OSPEEDR12 [1:0]		OSPEEDR11 [1:0]		OSPEEDR10 [1:0]		OSPEEDR9 [1:0]		OSPEEDR8 [1:0]	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OSPEEDR7 [1:0]		OSPEEDR6 [1:0]		OSPEEDR5 [1:0]		OSPEEDR4 [1:0]		OSPEEDR3 [1:0]		OSPEEDR2 [1:0]		OSPEEDR1 [1:0]		OSPEEDR0 [1:0]	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 2y+1:2y **OSPEEDRy[1:0]**：端口 x 配置位 (Port x configuration bits) (y = 0..15)

这些位通过软件写入，用于配置 I/O 输出速度。

00：低速

01：中速

10：高速

11：超高速

注：　有关 OSPEEDRy 位以及 V_{DD} 范围和外部负载的值，请参见产品数据手册。

11.4.4 GPIO 端口上拉/下拉寄存器 (GPIOx_PUPDR) (x = A..K)

GPIO port pull-up/pull-down register

偏移地址：0x0C

复位值：

- 0x6400 0000（端口 A）
- 0x0000 0100（端口 B）
- 0x0000 0000（其它端口）

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PUPDR15[1:0]		PUPDR14[1:0]		PUPDR13[1:0]		PUPDR12[1:0]		PUPDR11[1:0]		PUPDR10[1:0]		PUPDR9[1:0]		PUPDR8[1:0]	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PUPDR7[1:0]		PUPDR6[1:0]		PUPDR5[1:0]		PUPDR4[1:0]		PUPDR3[1:0]		PUPDR2[1:0]		PUPDR1[1:0]		PUPDR0[1:0]	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 2y+1:2y **PUPDRy[1:0]**：端口 x 配置位 (Port x configuration bits) (y = 0..15)
这些位通过软件写入，用于配置 I/O 上拉或下拉。
00：无上拉或下拉
01：上拉
10：下拉
11：保留

11.4.5 GPIO 端口输入数据寄存器 (GPIOx_IDR) (x = A..K)

GPIO port input data register

偏移地址：0x10

复位值：0x0000 XXXX（其中 X 表示未定义）

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IDR15	IDR14	IDR13	IDR12	IDR11	IDR10	IDR9	IDR8	IDR7	IDR6	IDR5	IDR4	IDR3	IDR2	IDR1	IDR0
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:16 保留，必须保持复位值。

位 15:0 **IDRy**：端口输入数据位 (Port input data bit) (y = 0..15)
这些位为只读。它们包含相应 I/O 端口的输入值。

11.4.6 GPIO 端口输出数据寄存器 (GPIOx_ODR) (x = A..K)

GPIO port output data register

偏移地址：0x14

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ODR15	ODR14	ODR13	ODR12	ODR11	ODR10	ODR9	ODR8	ODR7	ODR6	ODR5	ODR4	ODR3	ODR2	ODR1	ODR0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:16 保留，必须保持复位值。

位 15:0 **ODRy**：端口输出数据位 (Port output data bit) (y = 0..15)
这些位可通过软件读取和写入。
注：　对于原子置位/复位，通过写入 GPIOx_BSRR 或 GPIOx_BRR 寄存器，可分别置位和/或复位 ODR 位 (x = A..F)。

11.4.7 GPIO 端口置位/复位寄存器 (GPIOx_BSRR) (x = A..K)

GPIO port bit set/reset register

偏移地址：0x18

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
BR15	BR14	BR13	BR12	BR11	BR10	BR9	BR8	BR7	BR6	BR5	BR4	BR3	BR2	BR1	BR0
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BS15	BS14	BS13	BS12	BS11	BS10	BS9	BS8	BS7	BS6	BS5	BS4	BS3	BS2	BS1	BS0
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

位 31:16 **BRy**：端口 x 复位位 y (Port x reset bit y) (y = 0..15)
这些位为只写。读取这些位可返回值 0x0000。
0：不会对相应的 ODRx 位执行任何操作
1：复位相应的 ODRx 位
注： 如果同时对 BSx 和 BRx 置位，则 BSx 的优先级更高。

位 15:0 **BSy**：端口 x 置位位 y (Port x set bit y) (y= 0..15)
这些位为只写。读取这些位可返回值 0x0000。
0：不会对相应的 ODRx 位执行任何操作
1：置位相应的 ODRx 位

11.4.8 GPIO 端口配置锁定寄存器 (GPIOx_LCKR) (x = A..K)

GPIO port configuration lock register

当正确的写序列应用到第 16 位 (LCKK) 时，此寄存器将用于锁定端口位的配置。位 [15:0] 的值用于锁定 GPIO 的配置。在写序列期间，不能更改 LCKR[15:0] 的值。将 LOCK 序列应用到某个端口位后，在执行下一次 MCU 复位或外设复位之前，将无法对该端口位的值进行修改。

注： 可使用特定的写序列对 GPIOx_LCKR 寄存器执行写操作。在此锁定序列期间只允许使用字访问（32 位长）。

每个锁定位冻结一个特定的配置寄存器（控制寄存器和复用功能寄存器）。

偏移地址：0x1C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LCKK
															rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LCK15	LCK14	LCK13	LCK12	LCK11	LCK10	LCK9	LCK8	LCK7	LCK6	LCK5	LCK4	LCK3	LCK2	LCK1	LCK0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:17 保留，必须保持复位值。

位 16 **LCKK**：锁定键 (Lock key)

可随时读取此位。可使用锁定键写序列对其进行修改。

0：端口配置锁定键未激活

1：端口配置锁定键已激活。到下一次 MCU 复位或外设复位前，GPIOx_LCKR 寄存器将一直处于锁定状态。

锁定键写序列：

WR LCKR[16] = ‘1’ + LCKR[15:0]

WR LCKR[16] = ‘0’ + LCKR[15:0]

WR LCKR[16] = ‘1’ + LCKR[15:0]

RD LCKR

RD LCKR[16] = ‘1’（此读操作为可选操作，但它可确认锁定已激活）

注： 在锁定键写序列期间，不能更改 LCK[15:0] 的值。

锁定序列中的任何错误都将中止锁定操作。

在任一端口位上的第一个锁定序列之后，对 LCKK 位的任何读访问都将返回“1”，直到下一次 MCU 复位或外设复位为止。

位 15:0 **LCKy**：端口 x 锁定位 y (Port x lock bit y) (y= 0..15)

这些位都是读/写位，但只能在 LCKK 位等于“0”时执行写操作。

0：端口配置未锁定

1：端口配置已锁定

11.4.9 GPIO 复用功能低位寄存器 (GPIOx_AFRL) (x = A..K)

GPIO alternate function low register

偏移地址：0x20

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
AFR7[3:0]				AFR6[3:0]				AFR5[3:0]				AFR4[3:0]			
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AFR3[3:0]				AFR2[3:0]				AFR1[3:0]				AFR0[3:0]			
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:0 **AFRy[3:0]**：端口 x 引脚 y 的复用功能选择 (Alternate function selection for port x pin y) (y = 0..7)

这些位通过软件写入，用于配置复用功能 I/O。

AFSELy 选择：

0000: AF0
0001: AF1
0010: AF2
0011: AF3
0100: AF4
0101: AF5
0110: AF6
0111: AF7
1000: AF8
1001: AF9
1010: AF10
1011: AF11
1100: AF12
1101: AF13
1110: AF14
1111: AF15

11.4.10 GPIO 复用功能高位寄存器 (GPIOx_AFRH) (x = A..J)

GPIO alternate function high register

偏移地址：0x24

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
AFR15[3:0]				AFR14[3:0]				AFR13[3:0]				AFR12[3:0]			
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AFR11[3:0]				AFR10[3:0]				AFR9[3:0]				AFR8[3:0]			
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:0 **AFRy[3:0]**：端口 x 引脚 y 的复用功能选择 (Alternate function selection for port x pin y) (y = 8..15)

这些位通过软件写入，用于配置复用功能 I/O。

AFSELy 选择：

0000：AF0
0001：AF1
0010：AF2
0011：AF3
0100：AF4
0101：AF5
0110：AF6
0111：AF7
1000：AF8
1001：AF9
1010：AF10
1011：AF11
1100：AF12
1101：AF13
1110：AF14
1111：AF15

11.4.11 GPIO 寄存器映射

下表提供了 GPIO 寄存器映射和复位值。

表 83. GPIO 寄存器映射和复位值

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	**GPIOA_MODER**	MODE15[1:0]		MODE14[1:0]		MODE13[1:0]		MODE12[1:0]		MODE11[1:0]		MODE10[1:0]		MODE9[1:0]		MODE8[1:0]		MODE7[1:0]		MODE6[1:0]		MODE5[1:0]		MODE4[1:0]		MODE3[1:0]		MODE2[1:0]		MODE1[1:0]		MODE0[1:0]	
	Reset value	1	0	1	0	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0x00	**GPIOB_MODER**	MODER15[1:0]		MODER14[1:0]		MODER13[1:0]		MODER12[1:0]		MODER11[1:0]		MODER10[1:0]		MODER9[1:0]		MODER8[1:0]		MODER7[1:0]		MODER6[1:0]		MODER5[1:0]		MODER4[1:0]		MODER3[1:0]		MODER2[1:0]		MODER1[1:0]		MODER0[1:0]	
	Reset value	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	0	1	1	1	1	1	1
0x00	**GPIOx_MODER** (where x = **C..K**)	MODER15[1:0]		MODER14[1:0]		MODER13[1:0]		MODER12[1:0]		MODER11[1:0]		MODER10[1:0]		MODER9[1:0]		MODER8[1:0]		MODER7[1:0]		MODER6[1:0]		MODER5[1:0]		MODER4[1:0]		MODER3[1:0]		MODER2[1:0]		MODEv1[1:0]		MODER0[1:0]	
	Reset value	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0x04	**GPIOx_OTYPER** (where x = **A..K**)	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OT15	OT14	OT13	OT12	OT11	OT10	OT9	OT8	OT7	OT6	OT5	OT4	OT3	OT2	OT1	OT0
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x08	**GPIOA_OSPEEDR**	OSPEEDR15[1:0]		OSPEEDR14[1:0]		OSPEEDR13[1:0]		OSPEEDR12[1:0]		OSPEEDR11[1:0]		OSPEEDR10[1:0]		OSPEEDR9[1:0]		OSPEEDR8[1:0]		OSPEEDR7[1:0]		OSPEEDR6[1:0]		OSPEEDR5[1:0]		OSPEEDR4[1:0]		OSPEEDR3[1:0]		OSPEEDR2[1:0]		OSPEEDR1[1:0]		OSPEEDR0[1:0]	
	Reset value	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x08	**GPIOB_OSPEEDR**	OSPEEDR15[1:0]		OSPEEDR14[1:0]		OSPEEDR13[1:0]		OSPEEDR12[1:0]		OSPEEDR11[1:0]		OSPEEDR10[1:0]		OSPEEDR9[1:0]		OSPEEDR8[1:0]		OSPEEDR7[1:0]		OSPEEDR6[1:0]		OSPEEDR5[1:0]		OSPEEDR4[1:0]		OSPEEDR3[1:0]		OSPEEDR2[1:0]		OSPEEDR1[1:0]		OSPEEDR0[1:0]	
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0
0x08	**GPIOx_OSPEEDR** (where x = **C..K**)	OSPEEDR15[1:0]		OSPEEDR14[1:0]		OSPEEDR13[1:0]		OSPEEDR12[1:0]		OSPEEDR11[1:0]		OSPEEDR10[1:0]		OSPEEDR9[1:0]		OSPEEDR8[1:0]		OSPEEDR7[1:0]		OSPEEDR6[1:0]		OSPEEDR5[1:0]		OSPEEDR4[1:0]		OSPEEDR3[1:0]		OSPEEDR2[1:0]		OSPEEDR1[1:0]		OSPEEDR0[1:0]	
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0C	**GPIOA_PUPDR**	PUPDR15[1:0]		PUPDR14[1:0]		PUPDR13[1:0]		PUPDR12[1:0]		PUPDR11[1:0]		PUPDR10[1:0]		PUPDR9[1:0]		PUPDR8[1:0]		PUPDR7[1:0]		PUPDR6[1:0]		PUPDR5[1:0]		PUPDR4[1:0]		PUPDR3[1:0]		PUPDR2[1:0]		PUPDR1[1:0]		PUPDR0[1:0]	
	Reset value	0	1	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0C	**GPIOB_PUPDR**	PUPDR15[1:0]		PUPDR14[1:0]		PUPDR13[1:0]		PUPDR12[1:0]		PUPDR11[1:0]		PUPDR10[1:0]		PUPDR9[1:0]		PUPDR8[1:0]		PUPDR7[1:0]		PUPDR6[1:0]		PUPDR5[1:0]		PUPDR4[1:0]		PUPDR3[1:0]		PUPDR2[1:0]		PUPDR1[1:0]		PUPDR0[1:0]	
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0

表 83. GPIO 寄存器映射和复位值（续）

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x0C	**GPIOx_PUPDR** (where x = **C..K**)	PUPDR15[1:0]		PUPDR14[1:0]		PUPDR13[1:0]		PUPDR12[1:0]		PUPDR11[1:0]		PUPDR10[1:0]		PUPDR9[1:0]		PUPDR8[1:0]		PUPDR7[1:0]		PUPDR6[1:0]		PUPDR5[1:0]		PUPDR4[1:0]		PUPDR3[1:0]		PUPDR2[1:0]		PUPDR1[1:0]		PUPDR0[1:0]	
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x10	**GPIOx_IDR** (where x = **A..I/J/K**)	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	IDR15	IDR14	IDR13	IDR12	IDR11	IDR10	IDR9	IDR8	IDR7	IDR6	IDR5	IDR4	IDR3	IDR2	IDR1	IDR0
	Reset value																	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
0x14	**GPIOx_ODR** (where x = **A..K**)	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ODR15	ODR14	ODR13	ODR12	ODR11	ODR10	ODR9	ODR8	ODR7	ODR6	ODR5	ODR4	ODR3	ODR2	ODR1	ODR0
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x18	**GPIOx_BSRR** (where x = **A..I/J/K**)	BR15	BR14	BR13	BR12	BR11	BR10	BR9	BR8	BR7	BR6	BR5	BR4	BR3	BR2	BR1	BR0	BS15	BS14	BS13	BS12	BS11	BS10	BS9	BS8	BS7	BS6	BS5	BS4	BS3	BS2	BS1	BS0
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x1C	**GPIOx_LCKR** (where x = **A..K**)	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LCKK	LCK15	LCK14	LCK13	LCK12	LCK11	LCK10	LCK9	LCK8	LCK7	LCK6	LCK5	LCK4	LCK3	LCK2	LCK1	LCK0
	Reset value																0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x20	**GPIOx_AFRL** (where x = **A..K**)	AFR7[3:0]				AFR6[3:0]				AFR5[3:0]				AFR4[3:0]				AFR3[3:0]				AFR2[3:0]				AFR1[3:0]				AFR0[3:0]			
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x24	**GPIOx_AFRH** (where x = **A..K**)	AFR15[3:0]				AFR14[3:0]				AFR13[3:0]				AFR12[3:0]				AFR11[3:0]				AFR10[3:0]				AFR9[3:0]				AFR8[3:0]			
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

有关寄存器边界地址的信息，请参见*第 100 页的第 2.2.2 节*。

12 系统配置控制器 (SYSCFG)

12.1 前言

器件配有一组配置寄存器。本部分的目标是详细介绍系统配置控制器。

12.2 SYSCFG主要特性

系统配置控制器的主要功能如下：

- 模拟开关配置管理
- I2C Fm+ 配置
- 选择以太网 PHY 接口
- 管理 GPIO 的外部中断线连接
- 管理 I/O 补偿单元
- 获取读保护和 Flash 存储区交换信息
- 管理自举序列和自举地址
- 管理 BOR 复位级别
- 管理安全 Flash 和受保护扇区的状态
- 管理 Flash 写保护状态
- 管理 DTCM 安全部分状态
- 管理独立看门狗行为（硬件或软件/冻结）
- 在停止和待机模式下生成复位信号
- 使能/禁止安全模式

12.3 SYSCFG 寄存器说明

12.3.1 SYSCFG 外设模式配置寄存器 (SYSCFG_PMCR)

SYSCFG peripheral mode configuration register

偏移地址：0x04

复位值：0x0X00 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	PC3SO	PC2SO	PA1SO	PA0SO	EPIS[2:0]			Res.	Res.	Res.	Res.	Res.
				rw	rw	rw	rw	rw	rw	rw					

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	BOOSTE	PB9 FMP	PB8 FMP	PB7 FMP	PB6 FMP	I2C4 FMP	I2C3 FMP	I2C2 FMP	I2C1 FMP
							rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:28 保留，必须保持复位值。

位 27 **PC3SO**：PC3 开关断开 (PC3 Switch Open)
该位控制 PC3 和 PC3_C 之间的模拟开关（双引脚）
0：闭合模拟开关（2 个引脚通过模拟开关相连）
1：断开模拟开关（2 个引脚分离）

位 26 **PC2SO**：PC2 开关断开 (PC2 Switch Open)
该位控制 PC2 和 PC2_C 之间的模拟开关（双引脚）
0：闭合模拟开关（2 个引脚通过模拟开关相连）
1：断开模拟开关（2 个引脚分离）

位 25 **PA1SO**：PA1 开关断开 (PA1 Switch Open)
该位控制 PA1 和 PA1_C 之间的模拟开关（双引脚）
0：闭合模拟开关（2 个引脚通过模拟开关相连）
1：断开模拟开关（2 个引脚分离）

位 24 **PA0SO**：PA0 开关断开 (PA0 Switch Open)
该位控制 PA0 和 PA0_C 之间的模拟开关（双引脚）
0：闭合模拟开关（2 个引脚通过模拟开关相连）
1：断开模拟开关（2 个引脚分离）

位 23:21 **EPIS[2:0]**：以太网 PHY 接口选择 (Ethernet PHY interface selection)
这些位选择以太网 PHY 接口。
000：MII
001：保留
010：保留
011：保留
100：RMII
101：保留
110：保留
111：保留

位 20:9 保留，必须保持复位值。

位 8 **BOOSTE**：升压器使能 (Booster Enable)
该位使能升压器，以在电源电压低于 2.7 V 时降低模拟开关的总谐波失真。
激活升压器可在电源电压低于 2.7 V 时保证模拟开关交流性能：在这种情况下，模拟开关性能与全电压范围时的性能相同。
0：禁止升压器
1：使能升压器

位 7 **PB9FMP**：PB(9) Fm+
该位使能 PB(9) 上的 I2C Fm+。
0：禁止 Fm+
1：使能 Fm+

位 6 **PB8FMP**：PB(8) 超快速模式 (PB(8) Fast Mode Plus)
该位使能 PB(8) 上的 I2C Fm+。
0：禁止 Fm+
1：使能 Fm+

位 5 **PB7FMP**：PB(7) 超快速模式 (PB(7) Fast Mode Plus)
该位使能 PB(7) 上的 I2C Fm+。
0：禁止 Fm+
1：使能 Fm+

位 4 **PB6FMP**：PB(6) Fm+
该位使能 PB(6) 上的 I2C Fm+。
0：禁止 Fm+
1：使能 Fm+

位 3 **I2C4FMP**：I2C4 Fm+
该位使能 I2C4 上的 Fm+。
0：禁止 Fm+
1：使能 Fm+

位 2 **I2C3FMP**：I2C3 Fm+
该位使能 I2C3 上的 Fm+。
0：禁止 Fm+
1：使能 Fm+

位 1 **I2C2FMP**：I2C2 Fm+
该位使能 I2C2 上的 Fm+。
0：禁止 Fm+
1：使能 Fm+

位 0 **I2C1FMP**：I2C1 Fm+
该位使能 I2C1 上的 Fm+。
0：禁止 Fm+
1：使能 Fm+

12.3.2 SYSCFG 外部中断配置寄存器 1 (SYSCFG_EXTICR1)

SYSCFG external interrupt configuration register 1

偏移地址：0x08

复位值：0x0000 0000

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

位 31:16 保留，必须保持复位值。

位 15:0 **EXTIx[3:0]**：EXTI x 配置（x = 0 到 3）(EXTI x configuration (x = 0 to 3))
这些位通过软件写入，以选择外部中断/事件检测的 EXTI 输入的源输入。
0000：PA[x] 引脚
0001：PB[x] 引脚
0010：PC[x] 引脚
0011：PD[x] 引脚
0100：PE[x] 引脚
0101：PF[x] 引脚
0110：PG[x] 引脚
0111：PH[x] 引脚
1000：PI[x] 引脚
1001：PJ[x] 引脚
1010：PK[x] 引脚
其他配置：保留

12.3.3 SYSCFG 外部中断配置寄存器 2 (SYSCFG_EXTICR2)

SYSCFG external interrupt configuration register 2

偏移地址：0x0C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EXTI7[3:0]				EXTI6[3:0]				EXTI5[3:0]				EXTI4[3:0]			
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:16 保留，必须保持复位值。

位 15:0 **EXTIx[3:0]**：EXTI x 配置（x = 4 到 7）(EXTI x configuration (x = 4 to 7))
这些位通过软件写入，以选择外部中断/事件检测的 EXTI 输入的源输入。
0000：PA[x] 引脚
0001：PB[x] 引脚
0010：PC[x] 引脚
0011：PD[x] 引脚
0100：PE[x] 引脚
0101：PF[x] 引脚
0110：PG[x] 引脚
0111：PH[x] 引脚
1000：PI[x] 引脚
1001：PJ[x] 引脚
1010：PK[x] 引脚
其他配置：保留

12.3.4 SYSCFG 外部中断配置寄存器 3 (SYSCFG_EXTICR3)

SYSCFG external interrupt configuration register 3

偏移地址：0x10

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EXTI11[3:0]				EXTI10[3:0]				EXTI9[3:0]				EXTI8[3:0]			
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:16 保留，必须保持复位值。

位 15:0 **EXTIx[3:0]**：EXTI x 配置（x = 8 到 11）(EXTI x configuration (x = 8 to 11))

这些位通过软件写入，以选择外部中断/事件检测的 EXTI 输入的源输入。

0000：PA[x] 引脚
0001：PB[x] 引脚
0010：PC[x] 引脚
0011：PD[x] 引脚
0100：PE[x] 引脚
0101：PF[x] 引脚
0110：PG[x] 引脚
0111：PH[x] 引脚
1000：PI[x] 引脚
1001：PJ[x] 引脚
1010：PK[x] 引脚
其他配置：保留

注： PK[11:8] 未使用

12.3.5 SYSCFG 外部中断配置寄存器 4 (SYSCFG_EXTICR4)

SYSCFG external interrupt configuration register 4

偏移地址：0x14

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EXTI15[3:0]				EXTI14[3:0]				EXTI13[3:0]				EXTI12[3:0]			
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:16 保留，必须保持复位值。

位 15:0 **EXTIx[3:0]**：EXTI x 配置（x = 12 到 15）(EXTI x configuration (x = 12 to 15))
这些位通过软件写入，以选择外部中断/事件检测的 EXTI 输入的源输入。
0000：PA[x] 引脚
0001：PB[x] 引脚
0010：PC[x] 引脚
0011：PD[x] 引脚
0100：PE[x] 引脚
0101：PF[x] 引脚
0110：PG[x] 引脚
0111：PH[x] 引脚
1001：PJ[x] 引脚
1010：PK[x] 引脚
其他配置：保留

注： PK[15:12] 未使用

12.3.6 SYSCFG 补偿单元控制/状态寄存器 (SYSCFG_CCCSR)

SYSCFG compensation cell control/status register

偏移地址：0x20

复位值：0x0000 0000

有关 I/O 补偿机制的详细说明，请参见*第 11.3.11 节：I/O 补偿单元*。

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	READY	Res.	Res.	Res.	Res.	Res.	Res.	CS	EN
							r							rw	rw

位 31:17 保留，必须保持复位值。

位 16 **HSLV**：低压时高速 (High-speed at low-voltage)
该位通过软件写入，用于在产品处于低压时优化 I/O 速度。
该位仅在 IO_HSLV 用户选项位置 1 时有效。该位必须仅在产品电源电压低于 2.7 V 时使用。V_{DD} 高于 2.7 V 时将该位置 1 可能会导致破坏性结果。
0：无 I/O 速度优化
1：I/O 速度优化

位 15:9 保留，必须保持复位值。

位 8 **READY**：补偿单元就绪标志 (Compensation cell ready flag)
该位提供补偿单元的状态。
0：I/O 补偿单元未就绪
1：I/O 补偿单元就绪

位 7:2 保留，必须保持复位值。

位 1 **CS**：代码选择 (Code selection)
该位选择 I/O 补偿单元要应用的代码。
0：单元的代码（在 SYSCFG_CCVR 中提供）
1：SYSCFG 补偿单元代码寄存器 (SYSCFG_CCCR) 中的代码

位 0 **EN**：使能 (Enable)
该位使能 I/O 补偿单元。
0：禁止 I/O 补偿单元
1：使能 I/O 补偿单元

12.3.7 SYSCFG 补偿单元值寄存器 (SYSCFG_CCVR)

SYSCFG compensation cell value register

偏移地址：0x24

复位值：0x0000 0000

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

位 31:8 保留，必须保持复位值。

位 7:4 **PCV[3:0]**：PMOS 补偿值 (PMOS compensation value)

该值由单元提供，并可供 CPU 用于计算 PMOS 晶体管的 I/O 补偿单元代码。复位 SYSCFG_CMPCR 的 CS 位时，I/O 补偿单元会应用该代码。

位 3:0 **NCV[3:0]**：NMOS 补偿值 (NMOS compensation value)

该值由单元提供，并可供 CPU 用于计算 NMOS 晶体管的 I/O 补偿单元代码。复位 SYSCFG_CMPCR 的 CS 位时，I/O 补偿单元会应用该代码。

12.3.8 SYSCFG 补偿单元代码寄存器 (SYSCFG_CCCR)

SYSCFG compensation cell code register

偏移地址：0x28

复位值：0x0000 0000

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

位 31:8 保留，必须保持复位值。

位 7:4 **PCC[3:0]**：PMOS 补偿代码 (PMOS compensation code)

这些位通过软件写入，用于定义 PMOS 晶体管的 I/O 补偿单元代码。
将 SYSCFG_CMPCR 的 CS 位置 1 时，I/O 补偿单元会应用该代码。

位 3:0 **NCC[3:0]**：NMOS 补偿代码 (NMOS compensation code)

这些位通过软件写入，用于定义 NMOS 晶体管的 I/O 补偿单元代码。
将 SYSCFG_CCCSR 的 CS 位置 1 时，I/O 补偿单元会应用该代码。

12.3.9 SYSCFG 封装寄存器 (SYSCFG_PKGR)

SYSCFG package register

偏移地址：0x124

复位值：0x000X 000X

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

位 31:4 保留，必须保持复位值。

位 3:0 **PKG[3:0]**：封装 (Package)

这些位指示器件封装。

0000：LQFP100 (STM32H7x3)

0010：TQFP144 (STM32H7x3)

0101：TQFP176/UFBGA176 (STM32H7x3)

1000：LQFP208/TFBGA240 (STM32H7x3)

其他配置：使能所有引脚

12.3.10 SYSCFG 用户寄存器 0 (SYSCFG_UR0)

SYSCFG user register 0

偏移地址：0x300

复位值：0x00XX 000X

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

位 31:24 保留，必须保持复位值。

位 23:16 **RDP[7:0]**：读保护 (Readout protection)
这些位指示读保护级别。
0xAA：级别 0，无保护
0xCC：级别 2（Flash 受读保护，全部调试功能，从 SRAM 自举且边界扫描处于禁止状态）
其他配置：级别 1（Flash 受读保护，调试功能受限且边界扫描处于使能状态）

位 15:1 保留，必须保持复位值。

位 0 **BKS**：存储区域交换 (Bank Swap)
该位指示 Flash 存储区映射。
0：Flash 存储区地址取反
1：Flash 存储区映射到其源地址

12.3.11 SYSCFG 用户寄存器 2 (SYSCFG_UR2)

SYSCFG user register 2

偏移地址：0x308

复位值：0xXXXX 000X

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
BOOT_ADD0[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BORH[1:0]	
														r	r

位 31:16 **BOOT_ADD0[15:0]**：自举地址 0 (Boot Address 0)
这些位定义 BOOT 引脚为低电平时内核自举地址的 MSB。

位 15:2 保留，必须保持复位值。

位 1:0 **BORH[1:0]**：BOR_LVL 欠压复位阈值电压 (BOR_LVL Brownout Reset Threshold Level)
这些位指示欠压复位阈值电压。
0x11：BOR 高复位阈值电压 (2.7 V-3.6 V)
0x10：BOR 中等复位阈值电压 (2.4 V-2.7 V)
0x01：BOR 低复位阈值电压 (2.1 V-2.4 V)
0x00：BOR 关闭复位阈值电压 (2.7 V-3.6 V)

12.3.12 SYSCFG 用户寄存器 3 (SYSCFG_UR3)

SYSCFG user register 3

偏移地址：0x30C

复位值：0xXXXX XXXX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BOOT_ADD1[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:16 保留，必须保持复位值。

位 15:0 **BOOT_ADD1[15:0]**：自举地址 1 (Boot Address 1)

这些位定义 BOOT 引脚为高电平时内核自举地址的 MSB。

12.3.13 SYSCFG 用户寄存器 4 (SYSCFG_UR4)

SYSCFG user register 4

偏移地址：0x310

复位值：0x000X XXXX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MEPAD_1
															r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

位 31:17 保留，必须保持复位值。

位 16 **MEPAD_1**：禁止批量擦除存储区 1 的受保护区域 (Mass Erase Protected Area Disabled for bank 1)

该位指示 Flash 受保护区域（存储区 1）是否受批量擦除影响。

0：批量擦除时，擦除受保护区域。

1：批量擦除时，不擦除受保护区域。

位 15:0 保留，必须保持复位值。

12.3.14 SYSCFG 用户寄存器 5 (SYSCFG_UR5)

SYSCFG user register 5

偏移地址：0x314

复位值：0x00XX 000X

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WRPS_1[7:0]							
								r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MESAD_1
															r

位 31:24 保留，必须保持复位值。

位 23:16 **WRPN_1[7:0]**：Flash 存储区 1 的写保护 (Write protection for flash bank 1)

WRPN[i] 位指示 Flash 存储区 1 的扇区 i 是否受保护。

0：激活扇区 i 的写保护

1：未激活扇区 i 的写保护

位 15:1 保留，必须保持复位值。

位 0 **MESAD_1**：禁止批量擦除存储区 1 的安全区域 (Mass erase secured area disabled for bank 1)

该位指示 Flash 安全区域（存储区 1）是否受批量擦除影响。

0：批量擦除时，擦除安全区域。

1：批量擦除时，不擦除安全区域。

12.3.15 SYSCFG 用户寄存器 6 (SYSCFG_UR6)

SYSCFG user register 6

偏移地址：0x318

复位值：0x0XXX 0XXX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	PA_END_1[11:0]											
				r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	PA_BEG_1[11:0]											
				r	r	r	r	r	r	r	r	r	r	r	r

位 31:28 保留，必须保持复位值。

位 27:16 **PA_END_1[11:0]**：存储区 1 受保护区域的结束地址 (Protected area end address for bank 1)

存储区 1 受保护区域的结束地址。

位 15:12 保留，必须保持复位值。

位 11:0 **PA_BEG_1[11:0]**：存储区 1 受保护区域的起始地址 (Protected area start address for bank 1)

存储区 1 受保护区域的起始地址。

12.3.16 SYSCFG 用户寄存器 7 (SYSCFG_UR7)

SYSCFG user register 7

偏移地址：0x31C

复位值：0x0XXX 0XXX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	SA_END_1[11:0]											
				r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	SA_BEG_1[11:0]											
				r	r	r	r	r	r	r	r	r	r	r	r

位 31:28 保留，必须保持复位值。

位 27:16 **SA_END_1[11:0]**：存储区 1 安全区域的结束地址 (Secured area end address for bank 1)

存储区 1 安全区域的结束地址。

位 15:12 保留，必须保持复位值。

位 11:0 **SA_BEG_1[11:0]**：存储区 1 安全区域的起始地址 (Secured area start address for bank 1)

存储区 1 安全区域的起始地址。

12.3.17 SYSCFG 用户寄存器 8 (SYSCFG_UR8)

SYSCFG user register 8

偏移地址：0x320

复位值：0x000X 000X

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MESAD_2
															r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MEPAD_2
															r

位 31:17 保留，必须保持复位值。

位 16 **MESAD_2**：禁止批量擦除存储区 2 的安全区域 (Mass erase secured area disabled for bank 2)

该位指示 Flash 安全区域（存储区 2）是否受批量擦除影响。
0：批量擦除时，擦除安全区域。
1：批量擦除时，不擦除安全区域。

位 15:1 保留，必须保持复位值。

位 0 **MEPAD_2**：禁止批量擦除存储区 2 的受保护区域 (Mass erase protected area disabled for bank 2)

该位指示 Flash 受保护区域（存储区 2）是否受批量擦除影响。
0：批量擦除时，擦除受保护区域。
1：批量擦除时，不擦除受保护区域。

12.3.18 SYSCFG 用户寄存器 9 (SYSCFG_UR9)

SYSCFG user register 9

偏移地址：0x324

复位值：0x0XXX 00XX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	PA_BEG_2[11:0]											
				r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WRPS_2[7:0]							
								r	r	r	r	r	r	r	r

位 31:28 保留，必须保持复位值。

位 27:16 **PA_BEG_2[11:0]**：存储区 2 受保护区域的起始地址 (Protected area start address for bank 2)

存储区 2 受保护区域的起始地址。

位 15:8 保留，必须保持复位值。

位 7:0 **WRPN_2[7:0]**：Flash 存储区 2 的写保护 (Write protection for flash bank 2)

WRPN[i] 位指示 Flash 存储区 2 的扇区 i 是否受保护。
0：激活扇区 i 的写保护
1：未激活扇区 i 的写保护

12.3.19 SYSCFG 用户寄存器 10 (SYSCFG_UR10)

SYSCFG user register 10

偏移地址：0x328

复位值：0x0XXX 0XXX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	SA_BEG_2[11:0]											
				r	r	r	r	r	r	r	r	r	r	r	r

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	PA_END_2[11:0]											
				r	r	r	r	r	r	r	r	r	r	r	r

位 31:28 保留，必须保持复位值。

位 27:16 **SA_BEG_2[11:0]**：存储区 2 安全区域的起始地址 (Secured area start address for bank 2)
存储区 2 安全区域的起始地址。

位 15:12 保留，必须保持复位值。

位 11:0 **PA_END_2[11:0]**：存储区 2 受保护区域的结束地址 (Protected area end address for bank 2)
存储区 2 受保护区域的结束地址。

12.3.20 SYSCFG 用户寄存器 11 (SYSCFG_UR11)

SYSCFG user register 11

偏移地址：0x32C

复位值：0x000X 0XXX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	IWDG1M
															r

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	SA_END_2[11:0]											
				r	r	r	r	r	r	r	r	r	r	r	r

位 31:17 保留，必须保持复位值。

位 16 **IWDG1M**：独立看门狗 1 模式 (Independent Watchdog 1 mode)
该位指示独立看门狗 1 (IWDG1) 的控制模式。
0：IWDG1 通过软件控制
1：IWDG1 通过硬件控制

位 15:12 保留，必须保持复位值。

位 11:0 **SA_END_2[11:0]**：存储区 2 安全区域的结束地址 (Secured area end address for bank 2)
存储区 2 安全区域的结束地址。

12.3.21 SYSCFG 用户寄存器 12 (SYSCFG_UR12)

SYSCFG user register 12

偏移地址：0x330

复位值：0x000X 000X

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

位 31:17 保留，必须保持复位值。

位 16 **SECURE**：安全模式 (Secure mode)
该位指示安全模式状态。
0：安全模式处于禁止状态
1：安全模式处于使能状态

位 15:0 保留，必须保持复位值。

12.3.22 SYSCFG 用户寄存器 13 (SYSCFG_UR13)

SYSCFG user register 13

偏移地址：0x334

复位值：0x000X 000X

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	D1SBRST
															rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SDRS[1:0]	
														r	r

位 31:17 保留，必须保持复位值。

位 16 **D1SBRST**：D1 待机复位 (D1 Standby reset)
该位指示 D1 域进入 DStandby 模式时是否生成复位信号。
0：进入 D1 待机模式时生成复位信号
1：进入 D1 待机模式时不生成复位信号

位 15:2 保留，必须保持复位值。

位 1:0 **SDRS[1:0]**：安全 DTCM RAM 大小 (Secured DTCM RAM Size)
这些位指示安全 DTCM RAM 大小。
00：2 KB
01：4 KB
10：8 KB
11：16 KB

12.3.23 SYSCFG 用户寄存器 14 (SYSCFG_UR14)

SYSCFG user register 14

偏移地址：0x338

复位值：0x000X 000X

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	D1STPRST
															rw

位 31:16 保留，必须保持复位值。

位 15:1 保留，必须保持复位值。

位 0 **D1STPRST**：D1 停止复位 (D1 Stop Reset)
该位指示 D1 域进入 DStop 模式时是否生成复位信号。
0：进入 D1 停止模式时生成复位信号
1：进入 D1 停止模式时不生成复位信号

12.3.24 SYSCFG 用户寄存器 15 (SYSCFG_UR15)

SYSCFG user register 15

偏移地址：0x33C

复位值：0x000X 000X

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

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

位 31:17 保留，必须保持复位值。

位 16 **FZIWDGSTB**：在待机模式下冻结独立看门狗 (Freeze independent watchdog in Standby mode)

该位指示在待机模式下是否冻结独立看门狗。

0：在待机模式下冻结独立看门狗

1：在待机模式下运行独立看门狗

位 15:0 保留，必须保持复位值。

12.3.25 SYSCFG 用户寄存器 16 (SYSCFG_UR16)

SYSCFG user register 16

偏移地址：0x340

复位值：0x000X 000X

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

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

位 31:17 保留，必须保持复位值。

位 16 **PKP**：私钥编程 (Private key programmed)

该位指示是否已编程器件私钥。

0：未编程私钥

1：已编程私钥

位 15:1 保留，必须保持复位值。

位 0 **FZIWDGSTP**：在停止模式下冻结独立看门狗 (Freeze independent watchdog in Stop mode)

该位指示在停止模式下是否冻结独立看门狗。
0：在停止模式下冻结独立看门狗
1：在停止模式下运行独立看门狗

12.3.26 SYSCFG 用户寄存器 17 (SYSCFG_UR17)

SYSCFG user register 17

偏移地址：0x344

复位值：0x0000 000X

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

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

位 31:1 保留，必须保持复位值。

位 0 **IO_HSLV**：I/O 高速/低压 (I/O high speed / low voltage)

该位指示已设置 IOHSLV 选项位。
0：产品在全电压范围内工作
1：产品在低于 2.7 V 的电压下工作

12.3.27 SYSCFG 寄存器映射

下表列出了 SYSCFG 寄存器映射和复位值。

表 84. SYSCFG 寄存器映射和复位值

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	Reserved	Reserved																															
0x04	SYSCFG_PMCR	Res.	Res.	Res.	Res.	PC3SO	PC2SO	PA1SO	PA0SO	EPIS[2:0]			Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BOOSTE	PB9FMP	PB8FMP	PB7FMP	PB6FMP	I2C4FMP	I2C3FMP	I2C2FMP	I2C1FMP
	Reset value					X	X	X	0	0	0	0													0	0	0	0	0	0	0	0	0
0x08	SYSCFG_EXTICR1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	EXTI3[3:0]				EXTI2[3:0]				EXTI1[3:0]				EXTI0[3:0]			
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0C	SYSCFG_EXTICR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	EXTI7[3:0]				EXTI6[3:0]				EXTI5[3:0]				EXTI4[3:0]			
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x10	SYSCFG_EXTICR3	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	EXTI11[3:0]				EXTI10[3:0]				EXTI9[3:0]				EXTI8[3:0]			
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x14	SYSCFG_EXTICR4	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	EXTI15[3:0]				EXTI14[3:0]				EXTI13[3:0]				EXTI12[3:0]			
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x20	SYSCFG_CCSR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	HSLV	Res.	Res.	Res.	Res.	Res.	Res.	Res.	READY	Res.	Res.	Res.	Res.	Res.	Res.	CS	EN
	Reset value																0								0							0	0
0x24	SYSCFG_CCVR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PCV[3:0]				NCV[3:0]			
	Reset value																									0	0	0	0	0	0	0	0
0x28	SYSCFG_CCCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PCC[3:0]				NCC[3:0]			
	Reset value																									0	0	0	0	0	0	0	0
0x2C - 0x120	Reserved	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	Reset value																																
0x124	SYSCFG_PKGR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PKG[3:0]			
	Reset value																													X	X	X	X
0x128 - 0x2FC	Reserved	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	Reset value																																
0x300	SYSCFG_UR0	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RDP[7:0]								Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BKS
	Reset value									x	x	x	x	x	x	x	x																x
0x304	Reserved	Reserved																															
0x308	SYSCFG_UR2	BOOT_ADD0[15:0]																Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BORH[1:0]	
	Reset value	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X															X	X
0x30C	SYSCFG_UR3	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BOOT_ADD1[15:0]															
	Reset value																	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

表 84. SYSCFG 寄存器映射和复位值（续）

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x310	SYSCFG_UR4	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MEPAD_1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	Reset value																X																
0x314	SYSCFG_UR5	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WRPN_1[7:0]								Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MESAD_1
	Reset value									X	X	X	X	X	X	X	X																X
0x318	SYSCFG_UR6	Res.	Res.	Res.	Res.	PA_END_1[11:0]												Res.	Res.	Res.	Res.	PA_BEG_1[11:0]												
	Reset value					X	X	X	X	X	X	X	X	X	X	X	X					X	X	X	X	X	X	X	X	X	X	X	X
0x31C	SYSCFG_UR7	Res.	Res.	Res.	Res.	SA_END_1[11:0]												Res.	Res.	Res.	Res.	SA_BEG_1[11:0]												
	Reset value					X	X	X	X	X	X	X	X	X	X	X	X					X	X	X	X	X	X	X	X	X	X	X	X
0x320	SYSCFG_UR8	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MESAD_2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MEPAD_2
	Reset value																x																x
0x324	SYSCFG_UR9	Res.	Res.	Res.	Res.	PA_END_2[11:0]												Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WRPN_2[7:0]							
	Reset value					X	X	X	X	X	X	X	X	X	X	X	X									X	X	X	X	X	X	X	X
0x328	SYSCFG_UR10	Res.	Res.	Res.	Res.	SA_BEG_2[11:0]												Res.	Res.	Res.	Res.	PA_END_2[11:0]												
	Reset value					X	X	X	X	X	X	X	X	X	X	X	X					X	X	X	X	X	X	X	X	X	X	X	X
0x32C	SYSCFG_UR11	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	IWDG1M	Res.	Res.	Res.	Res.	SA_END_2[11:0]												
	Reset value																X					X	X	X	X	X	X	X	X	X	X	X	X
0x330	SYSCFG_UR12	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SECURE	RES.	RES.	RES.	RES.	RES.	RES.	RES.	RES.	RES.	RES.	RES.	RES.	RES.	RES.	RES.	Res.
	Reset value																X																X
0x334	SYSCFG_UR13	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	D1SBRST	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SDRS[1:0]	
	Reset value																X															X	X
0x338	SYSCFG_UR14	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RES.	RES.	RES.	RES.	RES.	RES.	RES.	RES.	RES.	RES.	RES.	RES.	RES.	RES.	RES.	D1STPRST
	Reset value																X																X
0x33C	SYSCFG_UR15	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FZIWDGSTB	RES.	RES.	RES.	RES.	RES.	RES.	RES.	RES.	RES.	RES.	RES.	RES.	RES.	RES.	RES.	Res.
	Reset value																X																X

表 84. SYSCFG 寄存器映射和复位值（续）

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x340	SYSCFG_UR16	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PKP	RES.	RES.	RES.	RES.	RES.	RES.	RES.	RES.	RES.	RES.	RES.	RES.	RES.	RES.	RES.	FZIWDGSTP
	Reset value																X																X
0x344	SYSCFG_UR17	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RES.	RES.	RES.	RES.	RES.	RES.	RES.	RES.	RES.	RES.	RES.	RES.	RES.	RES.	RES.	RES.	IO_HSLV
	Reset value																																X

有关寄存器边界地址的信息，请参见*第 100 页的第 2.2.2 节*。

13 块互连

13.1 外设互连

13.1.1 前言

多个外设间有直接连接。

这可以在外设间实现自动通信和同步，从而节省 CPU 资源和降低功耗。

这些硬件连接可消除软件延时、设计可预测的系统并能够减少引脚和 GPIO 的数量。

13.1.2 连接概述

有多种类型的连接。

- **异步连接 (A)**
  源输出信号通过目标时钟采样，从而可能在源输出事件到目标事件检测的延时中引入抖动。
- **同步连接 (S)**
  源和目标同步（它们采用相同的时钟），从源到目标的延时是确定的。不引入抖动。
- **立即连接 (I)**
  源或目标是模拟信号。
- **TIM/HRTIM 输出的刹车/故障连接 (B)**
  源输出信号通过纯组合逻辑路径（无延时）禁止定时器输出。

表 85. 外设互连矩阵（D2 域）(1)(2)

源			目标																										
			D2 域																				D3 域						
			APB1									APB2							AHB1			AHB4	APB4					
			TIM2	TIM3	TIM4	TIM5	TIM12	LPTIM1	DAC	CRS	CAN	TIM1	TIM8	TIM15	TIM16	TIM17	DFSDM1	HRTIM	ADC1	ADC2	以太网	ADC3	LPTIM2	LPTIM3	LPTIM4	LPTIM5	COMP1	COMP2
D2 域	APB1	TIM2	-	S	S	-	-	-	S	-	A	S	S	S	-	-	-	S	S	S	A	S	-	-	-	-	I	I
		TIM3	S	-	S	S	-	-	-	-	A	S		S	-	-	S	S	S	S	A	S	-	-	-	-	I	I
		TIM4	S	S	-	S	S	-	S	-	-	S	S	S	-	-	S	-	S	S	-	S	-	-	-	-	-	-
		TIM5	-	-	-	-	S	-	S	-	-		S		-	-	-	-	-	-	-	S	-	-	-	-	-	-
		TIM6	-	-	-	-			S	-	-	-	-	-	-	-	S	S	S	S	-	S	-	-	-	-	-	-
		TIM7	-	-	-	-			S	-	-	-	-	-	-	-	S	S	-	-	-	-	-	-	-	-	-	-
		TIM13	-	-	-	-	S			-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
		TIM14	-	-	-	-	S			-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
		LPTIM1	-	-	-	-	-	-	A	-	-	-	-	-	-	-	A	A	A	A		A	-	-	-	-	-	-
		SPDIFRX	-	-	-	-	-	-		-	-	-	-	-	-	S	-	-	-	-	-	-	-	-	-	-	-	-
		OPAMP	-	-	-	-	-	-		-	-	-	-	-	-	-	-	A	-	-	-	-	-	-	-	-	-	-
		CAN	-	-	-	A	-	-		-	-	-	-	-	-	-	-	A	-	-	A	-	-	-	-	-	-	-
	APB2	TIM1	S	S	S	S	-	-	S	-	-	-	S	S	-	-	S	S	S	S	-	S	-	-	-	-	I	I
		TIM8	S	-	S	S	-	-	S	-	-	-	-	-	-	-	S		S	S	-	S	-	-	-	-	I	I
		TIM15	-	S	-	-	-	-	S	-	-	S	-	-	-	-		S	S	S	-	S	-	-	-	-	I	I
		TIM16	-	-	-	-	-	-	-	-	-	-	-	S	-	-	S	S	-	-	-	-	-	-	-	-	-	-
		TIM17	-	-	-	-	-	-	-	-	-	-	-	S	-	-		S	-	-	-	-	-	-	-	-	-	-
		SAI1	A	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	A	-	-	-	-
		SAI2	-	-	-	A	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	A	-	-	-
		DFSDM1	-	-	-	-	-	-	-	-	-	B	B	B	B	B	-	-	-	-	-	-	-	-	-	-	-	-
		HRTIM	-	-	-	-	-	-	A	-	A	-	-	-	-	-	S	-	A	A	A	A	-	-	-	-	-	-
	AHB1	ADC1	-	-	-	-	-	-	-	-	-	A	-	-	-	-	-	A	-	-	-	-	-	-	-	-	-	-
		ADC2	-	-	-	-	-	-	-	-	-	-	A	-	-	-	-	A	-	-	-	-	-	-	-	-	-	-
		ETH	A	A	-	-	-	-	-	-	A	-	-	-	-	-	-		-	-	-	-	-	-	-	-	-	-
		USB1	A			A	-	-		A	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
		USB2	A			A	-	-		A	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

1. 表中的字母对应于*第 13.1.2 节：连接概述*中介绍的连接类型。
2. 灰色单元格中的“-”符号表示无互连。

表 86. 外设互连矩阵（D3 域） (1) (2)

源			目标																							
			D2 域																			D3 域 AHB4 APB4				
			APB1								APB2							AHB1								
			TIM2	TIM3	TIM4	TIM5	TIM12	LPTIM1	DAC	CRS	CAN	TIM1	TIM8	TIM15	TIM16	TIM17	DFSDM1	HRTIM	ADC1	ADC2	以太网	ADC3	LPTIM2	LPTIM3	LPTIM4	LPTIM5
D3 域	APB4	EXTI	-	-	-	-	-	-	A	-	-	-	-	-	-	-	A	-	A	A	-	-	-	-	-	-
		LPTIM2	-	-	-	-	-	-	A	-	-	-	-	-	-	-	A	A	A	A	-	A	-	A	A	A
		LPTIM3	-	-	-	-	-	-	-	-	-	-	-	-	-	-	A	-	A	A	-	A	-	-	A	A
		LPTIM4	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	A	-	A
		LPTIM5	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	A	A	-
		COMP1	A	A	-	-	-	A	-	-	-	A/B	A/B	B	B	B		A/B	-	-	-	-	A	-	-	-
		COMP2	A	A	-	-	-	A	-	-	-	A/B	A/B	B	B	B		A/B	-	-	-	-	A	-	-	-
		SAI4	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	A	-	A
		RTC	-	-	-	-	-	A	-	-	-	-	-	-	A	-	-	-	-	-	-	-	A	-	-	-
	AHB4	ADC3	-	-	-	-	-	-	-	-	-	A	A	-	-	-	-	-	-	-	-	-	-	-	-	-
		RCC	A	-	-	-	-	-	-	A	-	-	-	A	A	A	-	-	-	-	-	-	-	-	-	-

1. 表中的字母对应于*第 13.1.2 节：连接概述*中介绍的连接类型。
2. 灰色单元格中的“-”符号表示无互连。

表 87. 外设互连矩阵详细信息(1)

源				目标				类型	备注
域	总线	外设	信号	信号	外设	总线	域		
D2	APB2	TIM1	TRGO	ITR0	TIM2	APB1	D2	S	-
		TIM8	TRGO	ITR1				S	-
	APB1	TIM3	TRGO	ITR2				S	-
		TIM4	TRGO	ITR3				S	-
	AHB1	ETH	PPS	ITR4				S	-
		USB1	SOF	ITR5				S	-
		USB2	SOF	ITR6				S	-
D3	APB4	COMP1	comp1_out	ETR1				I	-
		COMP2	comp2_out	ETR2				I	-
		RCC	lse_ck	ETR3				A	-
D2	APB2	SAI1	SAI1_FS_A	ETR4				A	-
		SAI1	SAI1_FS_B	ETR5				A	-
D3	APB4	COMP1	comp1_out	TI4_1				I	-
		COMP2	comp2_out	TI4_2				I	-
		COMP1 或 COMP2(2)	comp1_out 或 comp2_out	TI4_3				I	-
D2	APB2	TIM1	TRGO	ITR0	TIM3	APB1	D2	S	-
	APB1	TIM2	TRGO	ITR1				S	-
	APB2	TIM15	TRGO	ITR2				S	-
	APB1	TIM4	TRGO	ITR3				S	-
	AHB1	ETH	PPS	ITR4				S	-
D3	APB4	COMP1	comp1_out	ETR1				I	-
		COMP1	comp1_out	TI1_1				I	-
		COMP2	comp2_out	TI1_2				I	-
		COMP1 或 COMP2(2)	comp1_out 或 comp2_out	TI1_3				I	-
D2	APB2	TIM1	TRGO	ITR0	TIM4	APB1	D2	S	-
	APB1	TIM2	TRGO	ITR1				S	-
		TIM3	TRGO	ITR2				S	-
	APB2	TIM8	TRGO	ITR3				S	-

表 87. 外设互连矩阵详细信息(1)（续）

源				目标				类型	备注
域	总线	外设	信号	信号	外设	总线	域		
D2	APB2	TIM1	TRGO	ITR0	TIM5	APB1	D2	S	-
		TIM8	TRGO	ITR1				S	-
	APB1	TIM3	TRGO	ITR2				S	-
		TIM4	TRGO	ITR3				S	-
		CAN	SOC	ITR6				S	-
	AHB1	USB1	SOF	ITR7				S	-
		USB2	SOF	ITR8				S	-
	APB2	SAI2	SAI2_FS_A	ETR1				A	-
		SAI2	SAI2_FS_B	ETR2				A	-
	APB1	CAN	TMP	TI1_1				A	-
		CAN	RTP	TI1_2				A	-
D2	APB1	TIM4	TRGO	ITR0	TIM12	APB1	D2	S	-
		TIM5	TRGO	ITR1				S	-
		TIM13	OC1	ITR2				S	-
		TIM14	OC1	ITR3				S	-
	AHB1	USB1	SOF	crs_sync2	CRS	APB1	D2	A	-
		USB2	SOF	crs_sync0	CRS	APB1	D2	A	-
	AHB2	USB1	SOF	crs_sync2	CRS	APB1	D2	A	-
		USB2	SOF	crs_sync0	CRS	APB1	D2	A	-
D3	AHB4	RCC	lse_ck	crs_sync1	CRS	APB1	D2	A	-

表 87. 外设互连矩阵详细信息(1)（续）

源				目标				类型	备注
域	总线	外设	信号	信号	外设	总线	域		
D2	APB2	TIM15	TRGO	ITR0	TIM1	APB2	D2	S	-
	APB1	TIM2	TRGO	ITR1				S	-
		TIM3	TRGO	ITR2				S	-
		TIM4	TRGO	ITR3				S	-
D3	APB4	COMP1	comp1_out	ETR1				I	-
		COMP2	comp2_out	ETR2				I	-
D2	AHB1	ADC1	adc1_awd1	ETR3				A	-
		ADC1	adc1_awd2	ETR4				A	-
		ADC1	adc1_awd3	ETR5				A	-
D3	AHB4	ADC3	adc3_awd1	ETR6				A	-
		ADC3	adc3_awd2	ETR7				A	-
		ADC3	adc3_awd3	ETR8				A	-
	APB4	COMP1	comp1_out	TI1_1				I	-
		COMP1	comp1_out	BRK_1				B	-
		COMP2	comp2_out	BRK_2				B	-
D2	APB2	DFSDM1	dfsdm1_ break0	BRK_3				B	-
D3	APB4	COMP1	comp1_out	BRK2_1				B	-
		COMP2	comp2_out	BRK2_2				B	-
D2	APB2	DFSDM1	dfsdm1_ break1	BRK2_3				B	-

表 87. 外设互连矩阵详细信息(1)（续）

源				目标				类型	备注
域	总线	外设	信号	信号	外设	总线	域		
D2	APB2	TIM1	TRGO	ITR0	TIM8	APB2	D2	S	-
	APB1	TIM2	TRGO	ITR1				S	-
		TIM4	TRGO	ITR2				S	-
		TIM5	TRGO	ITR3				S	-
D3	APB4	COMP1	comp1_out	ETR1				I	-
		COMP2	comp2_out	ETR2				I	-
D2	AHB1	ADC2	adc2_awd1	ETR3				A	-
		ADC2	adc2_awd2	ETR4				A	-
		ADC2	adc2_awd3	ETR5				A	-
D3	AHB4	ADC3	adc3_awd1	ETR6				A	-
		ADC3	adc3_awd2	ETR7				A	-
		ADC3	adc3_awd3	ETR8				A	-
	APB4	COMP2	comp2_out	TI1_1				I	-
		COMP1	comp1_out	BRK_1				B	-
		COMP2	comp2_out	BRK_2				B	-
D2	APB2	DFSDM1	dfsdm1_break2	BRK_3				B	-
D3	APB4	COMP1	comp1_out	BRK2_1				B	-
		COMP2	comp2_out	BRK2_2				B	-
D2	APB2	DFSDM1	dfsdm1_break3	BRK2_3				B	-

表 87. 外设互连矩阵详细信息(1)（续）

源				目标				类型	备注
域	总线	外设	信号	信号	外设	总线	域		
D2	APB2	TIM1	TRGO	ITR0	TIM15	APB2	D2	S	-
	APB1	TIM3	TRGO	ITR1				S	-
	APB2	TIM16	OC1	ITR2				S	-
		TIM17	OC1	ITR3				S	-
	APB1	TIM2	CH1	TI1_1				A	-
		TIM3	CH1	TI1_2				A	-
		TIM4	CH1	TI1_3				A	-
D3	AHB4	RCC	lse_ck	TI1_4				A	-
		RCC	csi_ck	TI1_5				A	-
		RCC	MCO2	TI1_6				A	-
D2	APB1	TIM2	CH2	TI2_1				A	-
		TIM3	CH2	TI2_2				A	-
		TIM4	CH2	TI2_3				A	-
D3	APB4	COMP1	comp1_out	BRK_1				B	-
		COMP2	comp2_out	BRK_2				B	-
D2	APB2	DFSDM1	dfsdm_ break0	BRK_3				B	-
D3	AHB4	RCC	lsi_ck	TI1_1	TIM16	APB2	D2	A	-
		RCC	lse_ck	TI1_2				A	-
	APB4	RTC	WKUP_IT	TI1_3				A	-
		COMP1	comp1_out	BRK_1				B	-
		COMP2	comp2_out	BRK_2				B	-
D2	APB2	DFSDM1	dfsdm_ break1	BRK_3				B	-
D2	APB1	SPDIFRX	spdifrx_frame_ sync	TI1_1	TIM17	APB2	D2	A	-
D3	AHB4	RCC	HSE_1MHZ	TI1_2				A	-
		RCC	MCO1	TI1_3				A	-
	APB4	COMP1	comp1_out	BRK_1				B	-
		COMP2	comp2_out	BRK_2				B	-
D2	APB2	DFSDM1	dfsdm_ break2	BRK_3				B	-

表 87. 外设互连矩阵详细信息(1)（续）

源				目标				类型	备注
域	总线	外设	信号	信号	外设	总线	域		
D3	APB4	COMP1	comp1_out	hrtim_evt11	HRTIM	APB2	D2	B	-
D2	APB2	TIM1	TRGO	hrtim_evt12				B	-
	AHB1	ADC1	adc1_awd1	hrtim_evt13				B	-
D3	APB4	COMP2	OUT	hrtim_evt21				B	-
D2	APB2	TIM2	TRGO	hrtim_evt22				B	-
	AHB1	ADC1	adc1_awd2	hrtim_evt23				B	-
	NC	NC	NC	hrtim_evt31				B	-
	APB2	TIM3	TRGO	hrtim_evt32				B	-
	AHB1	ADC1	adc1_awd3	hrtim_evt33				B	-
	APB1	OPAMP1	opamp1_out	hrtim_evt41				B	-
		TIM7	TRGO	hrtim_evt42				B	-
	AHB1	ADC2	adc2_awd1	hrtim_evt43				B	-
	NC	NC	NC	hrtim_evt51				B	-
	APB1	LPTIM1	lptim1_out	hrtim_evt52				B	-
	AHB1	ADC2	adc2_awd2	hrtim_evt53				B	-
D3	APB4	COMP1	comp1_out	hrtim_evt61				I	-
D2	APB1	TIM6	TRGO	hrtim_evt62				S	-
	AHB1	ADC2	adc2_awd3	hrtim_evt63				A	-
D3	APB4	COMP2	comp2_out	hrtim_evt71				I	-
D2	APB1	TIM7	TRGO	hrtim_evt72				S	-
	NC	NC	NC	hrtim_evt73				-	-
				hrtim_evt81				-	-
	APB1	TIM6	TRGO	hrtim_evt82				S	-
	APB1	CAN	TTCAN_TMP	hrtim_evt83				A	-
		OPAMP1	opamp1_out	hrtim_evt91				I	-
	APB2	TIM15	TRGO	hrtim_evt92				S	-
	APB1	CAN	TTCAN_RTP	hrtim_evt93				A	-
	NC	NC	NC	hrtim_evt101					-
D3	APB4	LPTIM2	lptim2_out	hrtim_evt102				A	-
D2	APB1	CAN	TTCAN_SOC	hrtim_evt103				A	-

表 87. 外设互连矩阵详细信息(1)（续）

源				目标				类型	备注
域	总线	外设	信号	信号	外设	总线	域		
D3	APB4	COMP1	comp1_out	hrtim_in_flt1	HRTIM	APB2	D2	B	-
		COMP2	comp2_out	hrtim_in_flt2				B	-
D2	APB2	TIM16	OC	hrtim_upd_en1				S	-
		TIM17	OC	hrtim_upd_en2				S	-
	APB1	TIM6	TRGO	hrtim_upd_en3				S	-
		TIM7	TRGO	hrtim_bm_trg				S	-
	APB2	TIM16	OC	hrtim_bm_ck1				S	-
		TIM17	OC	hrtim_bm_ck2				S	-
	APB1	TIM7	TRGO	hrtim_bm_ck3				S	-
D3	APB4	RTC	rtc_alarm_a_evt	lptim1_ext_trg0	LPTIM1	APB1	D2	A	-
		RTC	rtc_alarm_b_evt	lptim1_ext_trg1				A	-
		RTC	rtc_tamp1_evt	lptim1_ext_trg2				A	-
		RTC	rtc_tamp2_evt	lptim1_ext_trg3				A	-
		RTC	rtc_tamp3_evt	lptim1_ext_trg4				A	-
		COMP1	comp1_out	lptim1_ext_trg5				I	-
		COMP2	comp2_out	lptim1_ext_trg6				I	-
		COMP1	comp1_out	lptim1_in1_mux1				I	-
		COMP2	comp2_out	lptim1_in2_mux2				I	-

表 87. 外设互连矩阵详细信息(1)（续）

源				目标				类型	备注
域	总线	外设	信号	信号	外设	总线	域		
D3	APB4	RTC	rtc_alarm_a_evt	lptim2_ext_trg0	LPTIM2	APB4	D3	A	-
		RTC	rtc_alarm_b_evt	lptim2_ext_trg1				A	-
		RTC	rtc_tamp1_evt	lptim2_ext_trg2				A	-
		RTC	rtc_tamp2_evt	lptim2_ext_trg3				A	-
		RTC	rtc_tamp3_evt	lptim2_ext_trg4				A	-
		COMP1	comp1_out	lptim2_ext_trg5				I	-
		COMP2	comp2_out	lptim2_ext_trg6				I	-
		COMP1	comp1_out	lptim2_in1_mux1				I	-
		COMP2	comp2_out	lptim2_in1_mux2				I	-
		COMP1 或 COMP2 (2)	comp1_out 或 comp2_out	lptim2_in1_mux3				I	-
		COMP2	comp2_out	lptim2_in2_mux1				I	-
D3	APB4	LPTIM2	lptim2_out	lptim3_ext_trg0	LPTIM3	APB4	D3	S	如果内核时钟源相同
		NC	NC	lptim3_ext_trg1				-	-
		LPTIM4	lptim4_out	lptim3_ext_trg2				S	如果内核时钟源相同
		LPTIM5	lptim5_out	lptim3_ext_trg3				S	如果内核时钟源相同
D2	APB2	SAI1	SAI1_FS_A	lptim3_ext_trg4				A	-
		SAI1	SAI1_FS_B	lptim3_ext_trg5				A	-
		SAI1	SAI1_FS_A	lptim3_in1_mux1				A	-
		SAI1	SAI1_FS_B	lptim3_in1_mux2				A	-

表 87. 外设互连矩阵详细信息(1)（续）

源				目标				类型	备注
域	总线	外设	信号	信号	外设	总线	域		
D3	APB4	LPTIM2	lptim2_out	lptim4_ext_trg0	LPTIM4	APB4	D3	S	如果内核时钟源相同
		LPTIM3	lptim3_out	lptim4_ext_trg1				S	如果内核时钟源相同
		NC	NC	lptim4_ext_trg2					-
		LPTIM5	lptim5_out	lptim4_ext_trg3				S	如果内核时钟源相同
D2	APB2	SAI2	SAI2_FS_A	lptim4_ext_trg4				A	-
		SAI2	SAI2_FS_B	lptim4_ext_trg5				A	-
D3	APB4	LPTIM2	lptim2_out	lptim5_ext_trg0	LPTIM5	APB4	D3	S	如果内核时钟源相同
		LPTIM3	lptim3_out	lptim5_ext_trg1				S	如果内核时钟源相同
		LPTIM4	lptim4_out	lptim5_ext_trg2				S	如果内核时钟源相同
		SAI4	SAI4_FS_A	lptim5_ext_trg3				A	-
		SAI4	SAI4_FS_B	lptim5_ext_trg4				A	-

表 87. 外设互连矩阵详细信息(1)（续）

源				目标				类型	备注
域	总线	外设	信号	信号	外设	总线	域		
D2	APB2	TIM1	TRGO	dac_ch1/2_trg0	DAC 通道 1/通道 2	APB1	D2	S	-
	APB1	TIM2	TRGO	dac_ch1/2_trg1				S	-
		TIM4	TRGO	dac_ch1/2_trg02				S	-
		TIM5	TRGO	dac_ch1/2_trg3				S	-
		TIM6	TRGO	dac_ch1/2_trg4				S	-
		TIM7	TRGO	dac_ch1/2_trg5				S	-
	APB2	TIM8	TRGO	dac_ch1/2_trg6				S	-
		TIM15	TRGO	dac_ch1/2_trg7				S	-
		HRTIM1	hrtim_dac_trg1	dac_ch1/2_trg8				S	-
			hrtim_dac_trg2	dac_ch1/2_trg9				S	-
	APB1	LPTIM1	lptim1_out	dac_ch1/2_trg10				S	-
		LPTIM2	lptim2_out	dac_ch1/2_trg11				S	-
D3	APB4	SYSCFG	EXTI9	dac_ch1/2_trg12				S	-

表 87. 外设互连矩阵详细信息(1)（续）

源				目标				类型	备注
域	总线	外设	信号	信号	外设	总线	域		
D2	APB2	TIM1	TRGO	TRG0	DFSDM1	APB2	D2	S	-
		TIM1	TRGO2	TRG1				S	-
		TIM8	TRGO	TRG2				S	-
		TIM8	TRGO2	TRG3				S	-
	APB1	TIM3	TRGO	TRG4				S	-
		TIM4	TRGO	TRG5				S	-
	APB2	TIM16	OC1	TRG6				S	-
	APB1	TIM6	TRGO	TRG7				S	-
		TIM7	TRGO	TRG8				S	-
	APB2	HRTIM1	hrtim_adc_trg1	TRG9				S	-
		HRTIM1	hrtim_adc_trg3	TRG10				S	-
D3	APB4	SYSCFG	EXTI11	TRG24				A	-
		SYSCFG	EXTI15	TRG25				A	-
D2	APB1	LPTIM1	lptim1_out	TRG26				A	-
D3	APB4	LPTIM2	lptim2_out	TRG27				A	-
		LPTIM3	lptim3_out	TRG28				A	-
D2	APB2	TIM1	CC1	adc_ext_trg0	ADC1 / ADC2	AHB1	D2	S	-
		TIM1	CC2	adc_ext_trg1				S	-
		TIM1	CC3	adc_ext_trg2				S	-
	APB1	TIM2	CC2	adc_ext_trg3				S	-
		TIM3	TRGO	adc_ext_trg4				S	-
		TIM4	CC4	adc_ext_trg5				S	-
D3	APB4	SYSCFG	EXTI11	adc_ext_trg6				A	-

表 87. 外设互连矩阵详细信息(1)（续）

源				目标				类型	备注
域	总线	外设	信号	信号	外设	总线	域		
D2	APB2	TIM8	TRGO	adc_ext_trg7	ADC1 / ADC2	AHB1	D2	S	-
		TIM8	TRGO2	adc_ext_trg8				S	-
		TIM1	TRGO	adc_ext_trg9				S	-
		TIM1	TRGO2	adc_ext_trg10				S	-
	APB1	TIM2	TRGO	adc_ext_trg11				S	-
		TIM4	TRGO	adc_ext_trg12				S	-
		TIM6	TRGO	adc_ext_trg13				S	-
	APB2	TIM15	TRGO	adc_ext_trg14				S	-
	APB1	TIM3	CC4	adc_ext_trg15				S	-
	APB2	HRTIM1	hrtim_adc_trg1	adc_ext_trg16				A	-
		HRTIM1	hrtim_adc_trg3	adc_ext_trg17				A	-
		LPTIM1	lptim1_out	adc_ext_trg18				A	-
D3	APB4	LPTIM2	lptim2_out	adc_ext_trg19				A	-
		LPTIM3	lptim3_out	adc_ext_trg20				A	-
D2	APB2	TIM1	TRGO	adc_jext_trg0	ADC1 / ADC2	AHB1	D2	S	-
		TIM1	CC4	adc_jext_trg1				S	-
	APB1	TIM2	TRGO	adc_jext_trg2				S	-
		TIM2	CC1	adc_jext_trg3				S	-
		TIM3	CC4	adc_jext_trg4				S	-
		TIM4	TRGO	adc_jext_trg5				S	-

表 87. 外设互连矩阵详细信息(1)（续）

源				目标				类型	备注
域	总线	外设	信号	信号	外设	总线	域		
D3	APB4	SYSCFG	EXTI15	adc_jext_trg6	ADC1 / ADC2	AHB1	D2	A	-
D2	APB2	TIM8	CC4	adc_jext_trg7				S	-
		TIM1	TRGO2	adc_jext_trg8				S	-
		TIM8	TRGO	adc_jext_trg9				S	-
		TIM8	TRGO2	adc_jext_trg10				S	-
	APB1	TIM3	CC3	adc_jext_trg11				S	-
		TIM3	TRGO	adc_jext_trg12				S	-
		TIM3	CC1	adc_jext_trg13				S	-
		TIM6	TRGO	adc_jext_trg14				S	-
	APB2	TIM15	TRGO	adc_jext_trg15				S	-
		HRTIM1	hrtim_adc_trg2	adc_jext_trg16				A	-
		HRTIM1	hrtim_adc_trg4	adc_jext_trg17				A	-
	APB1	LPTIM1	lptim1_out	adc_jext_trg18				A	-
D3	APB4	LPTIM2	lptim2_out	adc_jext_trg19				A	-
		LPTIM3	lptim2_out	adc_jext_trg20				A	-

表 87. 外设互连矩阵详细信息(1)（续）

源				目标				类型	备注
域	总线	外设	信号	信号	外设	总线	域		
D2	APB2	TIM1	CC1	EXT0	ADC3	AHB4	D3	S	-
		TIM1	CC2	EXT1				S	-
		TIM1	CC3	EXT2				S	-
	APB1	TIM2	CC2	EXT3				S	-
		TIM3	TRGO	EXT4				S	-
		TIM4	CC4	EXT5				S	-
D3	APB4	SYSCFG	EXTI11	EXT6				A	-
D2	APB2	TIM8	TRGO	EXT7				S	-
		TIM8	TRGO2	EXT8				S	-
		TIM1	TRGO	EXT9				S	-
		TIM1	TRGO2	EXT10				S	-
	APB1	TIM2	TRGO	EXT11				S	-
		TIM4	TRGO	EXT12				S	-
		TIM6	TRGO	EXT13				S	-
	APB2	TIM15	TRGO	EXT14				S	-
	APB1	TIM3	CC4	EXT15				S	-
	APB2	HRTIM1	hrtim_adc_trg1	EXT16				A	-
		HRTIM1	hrtim_adc_trg3	EXT17				A	-
		LPTIM1	lptim1_out	EXT18				A	-
D3	APB4	LPTIM2	lptim2_out	EXT19				A	-
		LPTIM3	lptim3_out	EXT20				A	-

表 87. 外设互连矩阵详细信息(1)（续）

源				目标				类型	备注
域	总线	外设	信号	信号	外设	总线	域		
D2	APB2	TIM1	TRGO	JEXT0	ADC3	AHB4	D3	SI	-
		TIM1	CC4	JEXT1				S	-
	APB1	TIM2	TRGO	JEXT2				S	-
		TIM2	CC1	JEXT3				S	-
		TIM3	CC4	JEXT4				S	-
		TIM4	TRGO	JEXT5				S	-
D3	APB4	SYSCFG	EXTI15	JEXT6				A	-
D2	APB2	TIM8	CC4	JEXT7				S	-
		TIM1	TRGO2	JEXT8				S	-
		TIM8	TRGO	JEXT9				S	-
		TIM8	TRGO2	JEXT10				S	-
	APB1	TIM3	CC3	JEXT11				S	-
		TIM3	TRGO	JEXT12				S	-
		TIM3	CC1	JEXT13				S	-
		TIM6	TRGO	JEXT14				S	-
	APB2	TIM15	TRGO	JEXT15				S	-
		HRTIM1	hrtim_adc_trg2	JEXT16				A	-
		HRTIM1	hrtim_adc_trg4	JEXT17				A	-
	APB1	LPTIM1	OUT	JEXT18				A	-
D3	APB4	LPTIM2	OUT	JEXT19				A	-
		LPTIM3	OUT	JEXT20				A	-
D2	APB2	TIM1	OC5	comp_blk1	COMP1 / COMP2	APB4	D3	I	-
		TIM1	OC3	comp_blk2				I	-
	APB1	TIM3	OC3	comp_blk3				I	-
		TIM3	OC4	comp_blk4				I	-
	APB2	TIM8	OC5	comp_blk5				I	-
		TIM15	OC1	comp_blk6				I	-

表 87. 外设互连矩阵详细信息(1)（续）

源				目标				类型	备注
域	总线	外设	信号	信号	外设	总线	域		
D2	APB1	TIM2	TRGO	SWT0	FDCAN	APB1	D2	A	-
		TIM3	TRGO	SWT1				A	-
	AHB1	ETH	PPS	SWT2				A	-
	APB2	HRTIM1	hrtim_dac_trg1	SWT3				A	-
	APB1	TIM2	TRGO	EVT0				A	-
		TIM3	TRGO	EVT1				A	-
	AHB1	ETH	PPS	EVT2				A	-
	APB2	HRTIM1	hrtim_dac_trg1	EVT3				A	-
	APB1	TIM2	TRGO	PTP0	ETH	AHB1	D2	A	-
		TIM3	TRGO	PTP1				A	-
	APB2	HRTIM1	hrtim_dac_trg2	PTP2				A	-
	APB1	CAN	TMP	PTP3				A	-

1. 表中的字母对应于*第 13.1.2 节：连接概述*中介绍的连接类型。
2. comp1_out 和 comp2_out 连接到或门的输入。该或门的输出连接到 lptim2_in1_mux3 输入。

13.2 从低功耗模式唤醒

扩展中断和事件控制器模块 (EXTI) 允许从停止模式唤醒系统和/或从 CStop 模式唤醒 CPU。唤醒事件来自外设。

这些事件通过 EXTI 处理为可配置事件 (**C**) 或直接事件 (**D**)。请参见*表 88* 中的*类型*列。更多详细信息，请参见*第 20 节：扩展中断和事件控制器 (EXTI)*。

以下三种类型的外设输出信号与 EXTI 输入事件相连：

- 唤醒信号。这些信号可通过外设生成，而无需任何总线接口时钟，它们在*表 88* 中称为 xxx_wkup。一些外设不具有该功能。
- 中断信号。只有当外设总线接口时钟运行时，才能生成这些信号。这些中断信号通常直接连接到 CPU 的 NVIC。它们称为 xxx_it。
- 信号，即通过外设生成的脉冲。外设生成信号后，外设级不需要任何操作（标志清零）。

每个 EXTI 输入事件有不同的唤醒功能或可能目标（请参见*表 88* 中的*目标*列）：

- CPU 唤醒 (**CPU**)：可使能输入事件来唤醒 CPU。
- 自动运行模式下的 CPU 和 D3 域唤醒 (**ANY**)：可使能输入事件来仅针对自动运行模式阶段唤醒 CPU 或 D3 域。

表 88. EXTI 唤醒输入(1)

源				目标		类型	目标	备注
域	总线	外设	信号	信号	外设			
D3	APB4	SYSCFG	exti0_wkup	WKUP0	EXTI	C	任意	-
			exti1_wkup	WKUP1				-
			exti2_wkup	WKUP2				-
			exti3_wkup	WKUP3				-
			exti4_wkup	WKUP4				-
			exti5_wkup	WKUP5				-
			exti6_wkup	WKUP6				-
			exti7_wkup	WKUP7				-
			exti8_wkup	WKUP8				-
			exti9_wkup	WKUP9				-
			exti10_wkup	WKUP10				-
			exti11_wkup	WKUP11				-
			exti12_wkup	WKUP12				-
			exti13_wkup	WKUP13				-
			exti14_wkup	WKUP14				-
			exti15_wkup	WKUP15				-
D3	AHB4	PWR	pvd_avd_wkup	WKUP16		C	CPU	-
D3	APB4	RTC	ALARMS	WKUP17		D	CPU	-
D3	APB4	RTC	TAMPER TIMESTAMP	WKUP18		C	CPU	-
D3	AHB4	RCC	CSS_LSE					-
D3	APB4	RTC	WKUP	WKUP19		C	任意	-
D3	APB4	COMP1	comp1_out	WKUP20		C	任意	-
D3	APB4	COMP2	comp2_out	WKUP21		C	任意	-
D2	APB1	I2C1	i2c1_wkup	WKUP22		C	CPU	-
D2	APB1	I2C2	i2c2_wkup	WKUP23		D	CPU	-
D2	APB1	I2C3	i2c3_wkup	WKUP24		D	CPU	-
D2	APB1	I2C4	i2c4_wkup	WKUP25		D	任意	-
D2	APB2	USART1	usart1_wkup	WKUP26		D	CPU	-
D2	APB1	USART2	usart2_wkup	WKUP27		D	CPU	-

表 88. EXTI 唤醒输入(1)（续）

源				目标		类型	目标	备注
域	总线	外设	信号	信号	外设			
D2	APB1	USART3	usart3_wkup	WKUP28		D	CPU	-
D2	APB2	USART6	usart6_wkup	WKUP29		D	CPU	-
D2	APB1	UART4	uart4_wkup	WKUP30		D	CPU	-
D2	APB1	UART5	uart5_wkup	WKUP31		D	CPU	-
D2	APB1	UART7	uart7_wkup	WKUP32		D	CPU	-
D2	APB1	UART8	uart8_wkup	WKUP33		D	CPU	-
D3	APB4	LPUART	lpuart_rx_wkup	WKUP34		D	任意	-
D3	APB4	LPUART	lpuart_tx_wkup	WKUP35		D	任意	-
D2	APB2	SPI1	spi1_wkup	WKUP36		D	CPU	-
D2	APB1	SPI2	spi2_wkup	WKUP37		D	CPU	-
D2	APB1	SPI3	spi3_wkup	WKUP38		D	CPU	-
D2	APB2	SPI4	spi4_wkup	WKUP39		D	CPU	-
D2	APB2	SPI5	spi5_wkup	WKUP40		D	CPU	-
D3	APB4	SPI6	spi6_wkup	WKUP41	EXTI	D	任意	-
D2	APB1	MDIOS	mdios_wkup	WKUP42		D	CPU	-
D2	AHB1	USB1	usb1_wkup	WKUP43		D	CPU	-
D2	AHB1	USB2	usb2_wkup	WKUP44		D	CPU	-
-	-	NC	NC	WKUP45		-	-	-
D2	APB1	LPTIM1	lptim1_wkup	WKUP47		D	CPU	-
D3	APB4	LPTIM2	lptim2_wkup	WKUP48		D	任意	-
D3	APB4	LPTIM2	lptim2_out	WKUP49		C	任意	(2)
D3	APB4	LPTIM3	lptim3_wkup	WKUP50		D	任意	-
D3	APB4	LPTIM3	lptim3_out	WKUP51		C	任意	(2)
D3	APB4	LPTIM4	lptim4_wkup	WKUP52		D	任意	-
D3	APB4	LPTIM5	lptim5_wkup	WKUP53		D	任意	-
D2	APB1	SWPMI	swpmi_wkup	WKUP54		D	CPU	-

表 88. EXTI 唤醒输入(1)（续）

源				目标		类型	目标	备注
域	总线	外设	信号	信号	外设			
D3	AHB4	PWR	pwr_wkup1_wkup	WKUP55	EXTI	D	CPU	-
			pwr_wkup2_wkup	WKUP56				-
			pwr_wkup3_wkup	WKUP57				-
			pwr_wkup4_wkup	WKUP58				-
			pwr_wkup5_wkup	WKUP59				-
			pwr_wkup6_wkup	WKUP60				-
D3	AHB4	RCC	rcc_it	WKUP61		D	CPU	-
D3	APB4	I2C4	i2c4_ev_it	WKUP62		D	CPU	(1)
		I2C4	i2c4_err_it	WKUP63		D	CPU	(1)
D3	APB4	LPUART1	lpuart1_it	WKUP64		D	CPU	(1)
D3	APB4	SPI6	spi6_it	WKUP64		D	CPU	(1)
D3	AHB4	BDMA	bdma_ch0_it	WKUP66		D	CPU	(1)
			bdma_ch1_it	WKUP67		D	CPU	(1)
			bdma_ch2_it	WKUP68		D	CPU	(1)
			bdma_ch3_it	WKUP69		D	CPU	(1)
			bdma_ch4_it	WKUP70		D	CPU	(1)
			bdma_ch5_it	WKUP71		D	CPU	(1)
			bdma_ch6_it	WKUP72		D	CPU	(1)
			bdma_ch7_it	WKUP73		D	CPU	(1)
D3	AHB4	DMAMUX2	dmamux2_it	WKUP74			CPU	(1)
D3	AHB4	ADC3	adc3_it	WKUP75		D	CPU	(1)
D3	APB4	SAI4	sai4_gbl_it	WKUP76		D	CPU	(1)
D3	AHB4	HSEM	hsem_int_it	WKUP77		D	CPU	(1)
-	-	NC	NC	WKUP81		-	-	-
D1	APB3	WWDG1	wwdg1_out_rst	WKUP82		C	CPU	(1)
-	-	NC	NC	WKUP83		-	-	-
D1	APB1	CEC	cec_wkup	WKUP85		C	CPU	-
D2	AHB1	ETH	eth	WKUP86		C	CPU	-
D3	AHB4	RCC	hse_css_rcc_wkup	WKUP87		D	CPU	-

1. 源外设需要其总线时钟来生成事件。这是 D3 域中的 PCLK4 或 HCLK4 以及 D1 域中的 PCLK3。
2. 源外设信号未连接到 NVIC。

扩展中断和事件控制器 (EXTI) 模块事件输入能够唤醒自动运行模式下的 D3 域，这些输入具有待处理的请求逻辑，该逻辑可通过 4 个不同的输入源（*表 89*）清零。更多详细信息，请

参见*第 20 节：扩展中断和事件控制器 (EXTI)*。

表 89. EXTI 待处理请求清零输入

源				目标				备注
域	总线	外设	信号	信号	外设	总线	域	
D3	AHB4	DMAMUX2	dmamux2_evt6	PRC0	EXTI	APB4	D3	-
			dmamux2_evt7	PRC1				-
	APB4	LPTIM4	lptim4_out	PRC2				-
		LPTIM5	lptim5_out	PRC3				-

13.3 DMA

在 D1 域中，MDMA 允许存储器传输数据。它可通过软件或硬件触发（请参照*第 13.3.1 节*所介绍的连接）。

D2 域中的 DMA 复用器 (DMAMUX1) 允许将任何外设 DMA 请求映射到 DMA1 或 DMA2 的任何数据流中。此外，DMAMUX 还提供两个其它的功能：

- 可将外设 DMA 请求与定时器、外部引脚或者其它数据流的 DMA 传输完成同步。
- 可通过 DMAMUX1 自身在数据流中生成 DMA 请求。该事件可通过定时器、外部引脚事件或其它流的 DMA 传输完成来触发。生成的 DMA 请求的数量可配置。

第 17 节：DMA 请求复用器 (DMAMUX)、*第 15 节：直接存储器访问控制器（DMA1、DMA2）*和*第 16 节：基本直接存储器访问控制器 (BDMA)* 介绍了 DMAMUX1 和 DMA1/DMA2 上的连接。

D3 域中的 DMA 复用器 (DMAMUX2) 的功能与 DMAMUX1 相同，它连接到基本 DMA (BDMA)。

*第 13.3.3 节：DMAMUX2 和 BDMA（D3 域）*介绍了 DMAMUX2 和 BDMA 上的连接。更多详细信息，请参见*第 13.3.3 节：DMAMUX2 和 BDMA（D3 域）*和*第 16 节：基本直接存储器访问控制器 (BDMA)*。

13.3.1 MDMA（D1 域）

表 90. MDMA

源				目标				备注
域	总线	外设	信号	信号	外设	总线	域	
D2	AHB1	DMA1	dma1_tcif0	mdma_str0	MDMA	AXI	D1	DMA1 数据流 0 传输完成
			dma1_tcif1	mdma_str1				DMA1 数据流 1 传输完成
			dma1_tcif2	mdma_str2				DMA1 数据流 2 传输完成
			dma1_tcif3	mdma_str3				DMA1 数据流 3 传输完成
			dma1_tcif4	mdma_str4				DMA1 数据流 4 传输完成
			dma1_tcif5	mdma_str5				DMA1 数据流 5 传输完成标志
			dma1_tcif6	mdma_str6				DMA1 数据流 6 传输完成
			dma1_tcif7	mdma_str7				DMA1 数据流 7 传输完成
D2	AHB1	DMA2	dma2_tcif0	mdma_str8				DMA2 数据流 0 传输完成
			dma2_tcif1	mdma_str9				DMA2 数据流 1 传输完成
			dma2_tcif2	mdma_str10				DMA2 数据流 2 传输完成
			dma2_tcif3	mdma_str11				DMA2 数据流 3 传输完成
			dma2_tcif4	mdma_str12				DMA2 数据流 4 传输完成
			dma2_tcif5	mdma_str13				DMA2 数据流 5 传输完成
			dma2_tcif6	mdma_str14				DMA2 数据流 6 传输完成
			dma2_tcif7	mdma_str15				DMA2 数据流 7 传输完成
D1	APB3	LTDC	ltdc_li_it	mdma_str16				LTDC 线中断
D1	AHB3	JPEG	jpeg_ift_trg	mdma_str17				JPEG 输入 FIFO 阈值
			jpeg_ifnt_trg	mdma_str18				JPEG 输入 FIFO 未满
			jpeg_oft_trg	mdma_str19				JPEG 输出 FIFO 阈值
			jpeg_ofne_trg	mdma_str20				JPEG 输出 FIFO 非空
			jpeg_oec_trg	mdma_str21				JPEG 转换结束
D1	AHB3	QUADSPI	quadspi_ft_trg	mdma_str22				QUADSPI FIFO 阈值
			quadspi_tc_trg	mdma_str23				QUADSPI 传输完成
D1	AHB3	DMA2D	dma2d_clut_trg	mdma_str24				DMA2D CLUT 传输完成
			dma2d_tc_trg	mdma_str25				DMA2D 传输完成
			dma2d_tw_trg	mdma_str26				DMA2D 传输水印
D1	AHB3	SDMMC1	sdmmc1_dataend_trg	mdma_str29				数据结束

13.3.2 DMAMUX1、DMA1 和 DMA2（D2 域）

表 91. DMAMUX1、DMA1 和 DMA2 连接(1)

源				目标				备注
域	总线	外设	信号	信号	外设	总线	域	
D3	AHB4	dmamux1 内部（请求发生器）		dmamux1_req_in0	DMAMUX1	AHB1	D2	请求
				dmamux1_req_in1				
				dmamux1_req_in2				
				dmamux1_req_in3				
				NC				
				NC				
				NC				
				NC				
D2	AHB1	ADC1	adc1_dma	dmamux1_req_in8				
D2	AHB1	ADC2	adc2_dma	dmamux1_req_in9				
D2	APB2	TIM1	tim1_ch1_dma	dmamux1_req_in10				
			tim1_ch2_dma	dmamux1_req_in11				
			tim1_ch3_dma	dmamux1_req_in12				
			tim1_ch4_dma	dmamux1_req_in13				
			tim1_up_dma	dmamux1_req_in14				
			tim1_trig_dma	dmamux1_req_in15				
			tim1_com_dma	dmamux1_req_in16				
D2	APB1	TIM2	tim2_ch1_dma	dmamux1_req_in17				
			tim2_ch2_dma	dmamux1_req_in18				
			tim2_ch3_dma	dmamux1_req_in19				
			tim2_ch4_dma	dmamux1_req_in20				
			tim2_up_dma	dmamux1_req_in21				
D2	APB1	TIM3	tim3_ch1_dma	dmamux1_req_in22				
			tim3_ch2_dma	dmamux1_req_in23				
			tim3_ch3_dma	dmamux1_req_in24				
			tim3_ch4_dma	dmamux1_req_in25				
			tim3_up_dma	dmamux1_req_in26				
			tim3_trig_dma	dmamux1_req_in27				
D2	APB1	TIM4	tim4_ch1_dma	dmamux1_req_in28				
			tim4_ch2_dma	dmamux1_req_in29				
			tim4_ch3_dma	dmamux1_req_in30				
			tim4_up_dma	dmamux1_req_in31				

表 91. DMAMUX1、DMA1 和 DMA2 连接(1)（续）

源				目标				备注
域	总线	外设	信号	信号	外设	总线	域	
D2	APB1	I2C1	i2c1_rx_dma	dmamux1_req_in32	DMAMUX1	AHB1	D2	请求
			i2c1_tx_dma	dmamux1_req_in33				
D2	APB1	I2C2	i2c2_rx_dma	dmamux1_req_in34				
			i2c2_tx_dma	dmamux1_req_in35				
D2	APB2	SPI1	spi1_rx_dma	dmamux1_req_in36				
			spi1_tx_dma	dmamux1_req_in37				
D2	APB1	SPI2	spi2_rx_dma	dmamux1_req_in38				
			spi2_tx_dma	dmamux1_req_in39				
D2	APB2	USART1	usart1_rx_dma	dmamux1_req_in40				
			usart1_tx_dma	dmamux1_req_in41				
D2	APB1	USART2	usart2_rx_dma	dmamux1_req_in42				
			usart2_tx_dma	dmamux1_req_in43				
D2	APB1	USART3	usart3_rx_dma	dmamux1_req_in44				
			usart3_tx_dma	dmamux1_req_in45				
D2	APB2	TIM8	tim8_ch1_dma	dmamux1_req_in46				
			tim8_ch2_dma	dmamux1_req_in47				
			tim8_ch3_dma	dmamux1_req_in48				
			tim8_ch4_dma	dmamux1_req_in49				
			tim8_up_dma	dmamux1_req_in50				
			tim8_trig_dma	dmamux1_req_in51				
			tim8_com_dma	dmamux1_req_in52				
-	-	NC	NC	NC				
D1	APB1	TIM3	tim5_ch1_dma	dmamux1_req_in54				
			tim5_ch2_dma	dmamux1_req_in55				
			tim5_ch3_dma	dmamux1_req_in56				
			tim5_ch4_dma	dmamux1_req_in57				
			tim5_up_dma	dmamux1_req_in58				
			tim5_trig_dma	dmamux1_req_in59				
D2	APB1	SPI3	spi3_rx_dma	dmamux1_req_in60				
			spi3_tx_dma	dmamux1_req_in61				
D1	APB1	UART4	uart4_rx_dma	dmamux1_req_in62				
			uart4_tx_dma	dmamux1_req_in63				

表 91. DMAMUX1、DMA1 和 DMA2 连接(1)（续）

源				目标				备注
域	总线	外设	信号	信号	外设	总线	域	
D1	APB1	UART5	uart5_rx_dma	dmamux1_req_in64	DMAMUX1	AHB1	D2	请求
			uart5_tx_dma	dmamux1_req_in65				
D2	APB1	DAC1	dac_ch1_dma	dmamux1_req_in66				
D2	APB1	DAC2	dac_ch2_dma	dmamux1_req_in67				
D2	APB1	TIM6	tim6_up_dma	dmamux1_req_in68				
D2	APB1	TIM7	tim7_up_dma	dmamux1_req_in69				
D2	APB2	USART6	usart6_rx_dma	dmamux1_req_in70				
			usart6_tx_dma	dmamux1_req_in71				
D2	APB1	I2C3	i2c3_rx_dma	dmamux1_req_in72				
			i2c3_tx_dma	dmamux1_req_in73				
D2	AHB2	DCMI	dcmi_dma	dmamux1_req_in74				
D2	AHB2	CRYP	cryp_in_dma	dmamux1_req_in75				
			cryp_out_dma	dmamux1_req_in76				
D2	AHB2	HASH	hash_in_dma	dmamux1_req_in77				
D2	APB1	UART7	uart7_rx_dma	dmamux1_req_in78				
			uart7_tx_dma	dmamux1_req_in79				
D2	APB1	UART8	uart8_rx_dma	dmamux1_req_in80				
			uart8_tx_dma	dmamux1_req_in81				
D2	APB2	SPI4	spi4_rx_dma	dmamux1_req_in82				
			spi4_tx_dma	dmamux1_req_in83				
D2	APB2	SPI5	spi5_rx_dma	dmamux1_req_in84				
			spi5_tx_dma	dmamux1_req_in85				
D2	APB2	SAI1	sai1_a_dma	dmamux1_req_in86				
			sai1_b_dma	dmamux1_req_in87				
D2	APB2	SAI2	sai2_a_dma	dmamux1_req_in88				
			sai2_b_dma	dmamux1_req_in89				
D2	APB1	SWPMI	swpmi_rx_dma	dmamux1_req_in90				
			swpmi_tx_dma	dmamux1_req_in91				
D2	APB1	SPDIFRX	spdifrx_dt_dma	dmamux1_req_in92				
			spdifrx_cs_dma	dmamux1_req_in93				

表 91. DMAMUX1、DMA1 和 DMA2 连接(1)（续）

源				目标				备注
域	总线	外设	信号	信号	外设	总线	域	
D2	APB2	HRTIM1	hrtim_dma1	dmamux1_req_in94	DMAMUX1	AHB1	D2	请求
			hrtim_dma2	dmamux1_req_in95				
			hrtim_dma3	dmamux1_req_in96				
			hrtim_dma4	dmamux1_req_in97				
			hrtim_dma5	dmamux1_req_in98				
			hrtim_dma6	dmamux1_req_in99				
D2	APB2	DFSDM1	dfsdm1_dma0	dmamux1_req_in100				
			dfsdm1_dma1	dmamux1_req_in101				
			dfsdm1_dma2	dmamux1_req_in102				
			dfsdm1_dma3	dmamux1_req_in103				
D2	APB2	TIM15	tim15_ch1_dma	dmamux1_req_in104				
			tim15_up_dma	dmamux1_req_in105				
			tim15_trig_dma	dmamux1_req_in106				
			tim15_com_dma	dmamux1_req_in107				
D2	APB2	TIM16	tim16_ch1_dma	dmamux1_req_in108				
			tim16_up_dma	dmamux1_req_in109				
D2	APB2	TIM17	tim17_ch1_mda	dmamux1_req_in110				
			tim17_up_dma	dmamux1_req_in111				
D2	APB2	SAI3	sai3_a_dma	dmamux1_req_in112				
			sai3_b_dma	dmamux1_req_in113				
D3	AHB4	ADC3	adc3_dma	dmamux1_req_in114				
D2	AHB1	DMAMUX1	dmamux1_evt0	dmamux1_gen0	DMAMUX1	AHB1	D2	请求生成
			dmamux1_evt1	dmamux1_gen1				
			dmamux1_evt2	dmamux1_gen2				
D2	APB1	LPTIM1	lptim1_out	dmamux1_gen3				
D2	APB1	LPTIM2	lptim2_out	dmamux1_gen4				
D2	APB1	LPTIM3	lptim3_out	dmamux1_gen5				
D3	APB4	EXTI	exti_exti0_it	dmamux1_gen6				
D2	APB1	TIM12	tim12_trgo	dmamux1_gen7				

表 91. DMAMUX1、DMA1 和 DMA2 连接(1)（续）

源				目标				备注
域	总线	外设	信号	信号	外设	总线	域	
D2	AHB1	DMAMUX1	dmamux1_evt0	dmamux1_trg0	DMAMUX1	AHB1	D2	触发
			dmamux1_evt1	dmamux1_trg1				
			dmamux1_evt2	dmamux1_trg2				
D2	APB1	LPTIM1	lptim1_out	dmamux1_trg3				
D2	APB1	LPTIM2	lptim2_out	dmamux1_trg4				
D2	APB1	LPTIM3	lptim3_out	dmamux1_trg5				
D3	APB4	EXTI	exti_exti0_it	dmamux1_trg6				
D2	APB1	TIM12	tim12_trgo	dmamux1_trg7				
D2	AHB1	DMAMUX1	dmamux1_req_out0	dma1_str0	DMA1	AHB1	D2	请求输出
			dmamux1_req_out1	dma1_str1				
			dmamux1_req_out2	dma1_str2				
			dmamux1_req_out3	dma1_str3				
			dmamux1_req_out4	dma1_str4				
			dmamux1_req_out5	dma1_str5				
			dmamux1_req_out6	dma1_str6				
			dmamux1_req_out7	dma1_str7				
			dmamux1_req_out8	dma2_str0	DMA2	AHB1	D2	
			dmamux1_req_out9	dma2_str1				
			dmamux1_req_out10	dma2_str2				
			dmamux1_req_out11	dma2_str3				
			dmamux1_req_out12	dma2_str4				
			dmamux1_req_out13	dma2_str5				
			dmamux1_req_out14	dma2_str6				
			dmamux1_req_out15	dma2_str7				

1. 灰色单元格中的“-”符号表示无互连。

13.3.3 DMAMUX2 和 BDMA（D3 域）

表 92. DMAMUX2 和 BDMA 连接

源				目标				备注
域	总线	外设	信号	信号	外设	总线	域	
D3	AHB4	dmamux2 内部（请求发生器）		dmamux2_req_in0	DMAMUX2	AHB4	D3	请求
				dmamux2_req_in1				
				dmamux2_req_in2				
				dmamux2_req_in3				
				NC				
				NC				
				NC				
				NC				
D3	APB4	LPUART	dma_rx_lpuart	dmamux2_req_in8				
			dma_tx_lpuart	dmamux2_req_in9				
D3	APB4	SPI6	dma_rx_spi6	dmamux2_req_in10				
			dma_tx_spi6	dmamux2_req_in11				
D2	APB1	I2C4	dma_rx_i2c4	dmamux2_req_in12				
			dma_tx_i2c4	dmamux2_req_in13				
D3	APB4	SAI4	dma_a_sai4	dmamux2_req_in14				
			dma_b_sai4	dmamux2_req_in15				
D3	APB4	ADC3	dma_adc3	dmamux2_req_in16				

表 92. DMAMUX2 和 BDMA 连接（续）

源				目标				备注
域	总线	外设	信号	信号	外设	总线	域	
D3	AHB4	DMAMUX2	dmamux2_evt0	dmamux2_gen0	DMAMUX2	AHB4	D3	请求生成
			dmamux2_evt1	dmamux2_gen1				
			dmamux2_evt2	dmamux2_gen2				
			dmamux2_evt3	dmamux2_gen3				
			dmamux2_evt4	dmamux2_gen4				
			dmamux2_evt5	dmamux2_gen5				
			dmamux2_evt6	dmamux2_gen6				
D3	APB4	EXTI	it_exti_rx_lpuart	dmamux2_gen7				
			it_exti_tx_lpuart	dmamux2_gen8				
			it_exti_wkup_lptim2	dmamux2_gen9				
			it_exti_out_lptim2	dmamux2_gen10				
			it_exti_wkup_lptim3	dmamux2_gen11				
			it_exti_out_lptim3	dmamux2_gen12				
			it_exti_wkup_lptim4	dmamux2_gen13				
			it_exti_wkup_lptim5	dmamux2_gen14				
			it_exti_wkup_i2c4	dmamux2_gen15				
			it_exti_wkup_spi6	dmamux2_gen16				
			it_exti_out_comp1	dmamux2_gen17				
			it_exti_out_comp2	dmamux2_gen18				
			it_exti_wkup_rtc	dmamux2_gen19				
			it_exti_exti0_syscfg	dmamux2_gen20				
			it_exti_exti2_syscfg	dmamux2_gen21				
D3	APB4	I2C4	it_evt_i2c4	dmamux2_gen22				
D3	APB4	SPI6	it_spi6	dmamux2_gen23				
D3	APB4	LPUART	it_tx_lpuart1	dmamux2_gen24				
			it_rx_lpuart1	dmamux2_gen25				
D3	AHB4	ADC3	it_adc3	dmamux2_gen26				
			out_awd1_adc3	dmamux2_gen27				
D3	AHB4	BDMA	it_ch0_bdma	dmamux2_gen28				
			it_ch1_bdma	dmamux2_gen29				

表 92. DMAMUX2 和 BDMA 连接（续）

源				目标				备注
域	总线	外设	信号	信号	外设	总线	域	
D3	AHB4	DMAMUX2	dmamux2_evt0	dmamux2_trg0	DMAMUX2	AHB4	D3	触发
			dmamux2_evt1	dmamux2_trg1				
			dmamux2_evt2	dmamux2_trg2				
			dmamux2_evt3	dmamux2_trg3				
			dmamux2_evt4	dmamux2_trg4				
			dmamux2_evt5	dmamux2_trg5				
D3	APB4	EXTI	it_exti_tx_lpuart1	dmamux2_trg6				
			it_exti_rx_lpuart1	dmamux2_trg7				
			it_exti_out_lptim2	dmamux2_trg8				
			it_exti_out_lptim3	dmamux2_trg9				
			it_exti_wkup_i2c4	dmamux2_trg10				
			it_exti_wkup_spi6	dmamux2_trg11				
			it_exti_out_comp1	dmamux2_trg12				
			it_exti_wkup_rtc	dmamux2_trg13				
			it_exti_exti0_syscfg	dmamux2_trg14				
			it_exti_exti2_syscfg	dmamux2_trg15				
D3	AHB4	DMAMUX2	dmamux1_req_out0	bdma_ch0	BDMA	AHB4	D3	请求输出
			dmamux1_req_out1	bdma_ch1				
			dmamux1_req_out2	bdma_ch2				
			dmamux1_req_out3	bdma_ch3				
			dmamux1_req_out4	bdma_ch4				
			dmamux1_req_out5	bdma_ch5				
			dmamux1_req_out6	bdma_ch6				
			dmamux1_req_out7	bdma_ch7				

14 MDMA 控制器 (MDMA)

14.1 MDMA 简介

主机直接存储器访问 (MDMA) 用于在存储器与存储器之间或外设与存储器之间提供高速数据传输。可以在无需任何 CPU 操作的情况下通过 MDMA 快速移动数据。这样节省的 CPU 资源便可供其他操作使用。

MDMA 控制器提供有一个 AXI 主控总线接口，用于主存储器和外设寄存器访问（系统访问端口）；还提供有一个 AHB 主控总线接口，该接口只适用于 Cortex-M7 TCM 存储器访问（TCM 访问端口）。

MDMA 与标准 DMA 控制器（DMA1 或 DMA2）结合使用。它最多可提供 16 个通道，每个通道都专用于管理来自一个 DMA 数据流存储器缓冲区或其他外设（带集成 FIFO）的存储器访问请求。

14.2 MDMA 主要特性

- AXI/AHB 主控总线架构，一个专用于主存储器/外设访问，一个专用于 Cortex-M7 AHBS 端口（仅限 TCM 访问）。
- 16通道。
- 多达 32 个硬件触发源。
- 每个通道请求均可在任何请求源之间选择。可由软件配置，允许多个外设启动 DMA 请求。在一个块传输结束时，会自动更改触发选择。
- 所有通道均相同，可连接至一个标准 DMA 或一个外设请求（通过数据读/写数据进行确认）系统。
- 各个通道还支持软件触发。
- 一个 256 级存储器缓冲区，被分为两个 128 级先进先出 (FIFO)，将用于针对 1 个或 2 个连续缓冲区临时存储要传输的数据（突发或单次传输模式）。一个 FIFO 将存储要在当前通道块传输期间传输的数据（最大为块传输大小）。另一个 FIFO 可用于要传输的下一个缓冲区，既适用于同一通道也适用于下一通道传输。
- DMA 通道之间的优先级可用软件编程（4 个级别：非常高、高、中、低），在软件优先级相同的情况下可以通过硬件决定优先级（例如，通道 0 的优先级高于通道 1）。
- 独立的源和目标传输宽度可分别设置（字节、半字、字、双字）：源和目标的数据宽度不相等时，MDMA 会适当地封装/解封数据来优化带宽。
- 可分别选择源和目标的数据长度及地址增量。

注： *借助这一分离特性，可在软件层面进行某些更高级的封装/解封操作。例如，2 个 16 位数据块可使用两个 MDMA 通道在目标存储器中彼此交错，该过程的实现仅需编程 2 个通道（以 4 字节步长为增量，且数据大小为 16 位 + 2 个通道间的起始地址移 2 位）。*

- 源和目标的递增、递减或非递增/固定寻址。
- 始终根据小端模式约定进行数据封装/解封：数据实体（双字、字或半字）中的低地址始终包含最低有效字节。这与源和目标的地址递增/递减模式无关。
- 支持递增突发传输。突发大小可由软件进行配置，最大可为 128 字节。对于较大的数据单元，突发长度受限，需遵循最大 128 字节数据突发大小（例如，16x64 位或 32x32 位）。

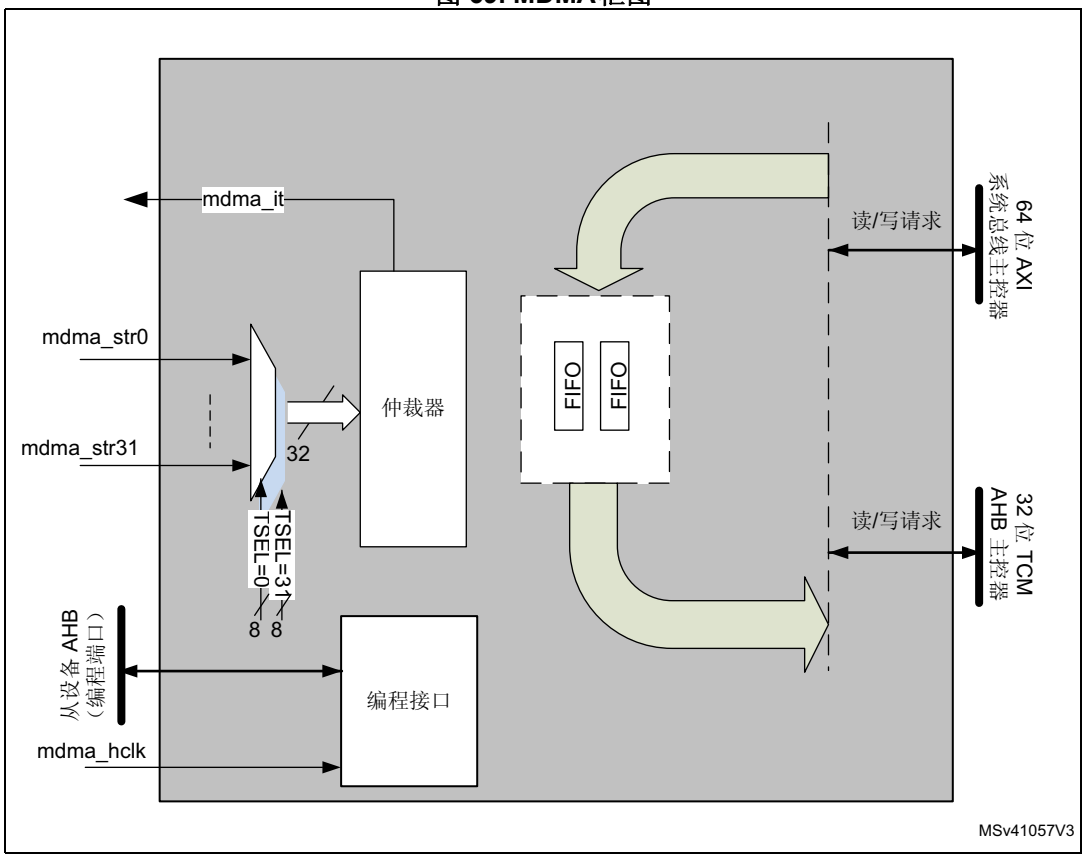
- 对于 TCM 存储器访问，只有在增量和数据大小等于或小于等于 32 位时，才允许突发访问。
- 提供 5 个事件标志（MDMA 通道传输完成、MDMA 块传输完成、MDMA 块重复传输完成、MDMA 缓冲区传输完成、MDMA 传输出错），且这些标志可生成中断。

14.3 MDMA 功能说明

14.3.1 MDMA 框图

图 69 显示了 MDMA 的框图。

图 69. MDMA框图



14.3.2 MDMA 内部信号

表 93 显示了内部 MDMA 信号。

表 93. MDMA 内部输入/输出信号

信号名称	信号类型	说明
mdma_hclk	数字输入	MDMA AHB 时钟
mdma_it	数字输出	MDMA 中断
mdma_str[0:31]	数字输入	MDMA 数据流请求

14.3.3 MDMA 概述

MDMA 控制器执行直接存储器传输：做为一个 AXI/AHB 主设备，它可以控制 AXI/AHB 总线矩阵来发起 AXI/AHB 事务。

它可以执行下列传输：

- 存储器到存储器（软件触发）
- 外设到存储器的传输
- 存储器到外设的传输

对于后两种传输类型，存储器还可替换为存储器映射的外设，其无法控制 MDMA 流。使用这些类型的传输且请求来自标准 DMA（DMA1 或 DMA2）时，外设寄存器访问会替换为该 DMA 所使用的存储器缓冲区的存储器访问。

注： *固定地址模式不能够用于存储器访问。*

源和目标仅由地址指定（存储器映射的外设也如此）。

AHB 从端口用于对 MDMA 控制器进行编程（它支持 8/16/32 位访问）。

单次请求传输的数据数组长度为以下长度之一：

1. 缓冲区传输大小
2. 块大小
3. 重复块
4. 完整通道数据（直至通道的链表指针为空）

通过 TRGM[1:0]（触发模式）选择字段进行大小选择。

用户必须根据可用的数据数组大小（通常在 DMA1/2 存储器缓冲区中）和其他 MDMA 通道的“实时”要求，选择其中的一种（因为 MDMA 缓冲区传输是不会在 MDMA 通道请求之间进行新仲裁的最小数据集合）。

对于每个通道，有三种数据数组长度非常重要：

1. 突发大小：此为可在突发模式下执行的数据传输长度。此突发长度定义了不能被总线仲裁中断并且可阻止其他主设备访问该总线的数据传输最大长度
2. 缓冲区传输大小：此为在检查其他通道的 MDMA 请求之前，要在某个通道中传输的数据数组长度。在该长度范围内的数据传输时，不能够被其他 MDMA 通道请求所中断
3. 块大小：该值有两个可同时使用的含义：
   a) 主要功能：在 MDMA 链表的块结构中说明数据块的长度（对应于链表中的一个条目）
   b) 可选功能：如果 TRGM[1:0] 等于 01，则该长度为在单次 MDMA 请求激活（针对相应通道）时传输的数据数组长度

14.3.4 MDMA 通道

每个 DMA 控制器通道都能够提供源和目标之间的单向传输链路。

每个通道均可执行：

- 单块传输：传输一个块。在块传输结束时，会禁止 DMA 通道，并会生成通道传输结束中断。
- 重复块传输：在禁止相应通道前，传输大量块。
- 链表传输：当前数据块（或重复模式下的最后一个块）的传输完成后，会从存储器加载新块控制结构，并会开始新块传输。

可编程要针对每个请求传输的最小数据量（缓冲区大小，最高 128 字节）。一个块中的总数据大小是可编程的，最大为 64 KB。每次传输后，该值都会递减。此计数器达到 0 时，即表示达到块末尾，并会根据重复传输计数器（针对重复块传输）和/或链表结构值决定下一步操作。

注： *如果块长度不是缓冲区长度的倍数，则块中的最后一次缓冲区传输将变短，其中包含要在当前块中传输的剩余字节。*

如果链路结构地址指向有效存储器地址，则 MDMA 会在该地址处从存储器重载完整通道描述符结构寄存器内容。然后，将基于该信息执行新块传输（在出现下一个 MDMA 通道请求时）。

如果链路结构地址为 0x0，则在当前/重复块传输结束时，将禁止 MDMA 通道，并且会生成通道传输结束中断。

14.3.5 源、目标和传输模式

源传输和目标传输可在整个 4 GB 区域（地址在 0x0000 0000 和 0xFFFF FFFF 之间）寻址外设和存储器。

源/目标地址可固定（例如，FIFO/单个数据寄存器外设）或递增/递减。可在单次访问或突发模式下进行传输（可编程）。

14.3.6 指针更新

根据 MDMA_CxCR 寄存器的 SINC[1:0] 和 DINC[1:0] 位，源和目标存储器指针在每次传输后可以自动递增/递减或保持常量。

通过单个寄存器/FIFO 模式访问外设源或目标数据时，禁止递增模式十分有用。

如果使能了递增/递减模式，则根据在 MDMA_CxCR 寄存器 SINCOS[1:0] 或 DINCOS[1:0] 位中编程的增量大小，下一次数据传输的地址将是前一次传输的地址递增/递减 1、2、4 或 8。

为优化封装操作，可独立编程增量偏移大小和数据大小。

14.3.7 MDMA 缓冲区传输

此为发生 MDMA 请求事件时，在一个通道上传输的数据的最小逻辑量（最多 128 字节）。

MDMA 缓冲区传输包括一系列给定数据量传输（以单次或突发数据传输的形式进行）。要传输的数据项的数目及其宽度（8 位、16 位、32 位或 64 位）可用软件编程。用于数据传输的突发长度也可独立编程。

在产生数据传输请求事件后，DMA/外设会向 MDMA 控制器发送请求信号。MDMA 控制器根据通道优先级处理该请求。

如果掩码地址寄存器已被设置，向掩码寄存器指定的地址写入掩码数据时就会确认这个请求。

如果掩码地址寄存器未设置（值 0x00），则向该外设读取/写入数据即可复位请求。此时，如果由目标外设完成请求，则必须将写入操作设为不可缓冲，以避免出现错误的新 MDMA 请求。

在 MDMA 请求之后当前通道需要传输的数据总量由 TRGM[1:0] 字段决定。

如果 TRGM[1:0] 等于 00，MDMA 会在一个缓冲区的数据传输完成后在同一个通道上等待另一个请求。

注： *在这种情况下，不会再考虑当前激活通道的硬件请求（FIFO 中的数据），直至该通道的写入阶段结束。此时，即使通道在读取阶段结束后仍处于激活状态，另一通道（甚至是优先级较低的通道）也能够开始读取阶段。因此，较低优先级的通道与当前通道的数据传输交错进行。*

如果 TRGM[1:0] 不为 00（需要传输多个缓冲区），则当前通道的 mdma_strx 保持激活（在内部存储），直至完成 TRGM 定义的完整传输（块、重复块或整个通道/链表数据）。不过，传输完单一缓冲区后，MDMA 将进入新仲裁阶段（在新外部请求和内部存储请求之间）。如果不存在更高的其他优先级，则通道请求激活，并且会对同一通道开始新缓冲区传输。

注： *如果 TRGM[1:0] 不为 00，则一次请求后会传输大量数据。但是，由于在每个缓冲区传输后都会进行通道仲裁，因此在较低优先级通道的数据传输将较高级别 MDMA 请求阻塞的时间不会超过一个缓冲区的传输周期。*

14.3.8 请求仲裁

仲裁器根据 MDMA 通道请求优先级对其进行管理。MDMA 空闲以及在各个缓冲区传输结束后，会对所有已使能通道检查所有 MDMA 请求（硬件或软件）。

优先级管理分为两个阶段：

- 软件：每个数据流优先级都可以在 MDMA_CxCR 寄存器中配置。分为四个级别：
  - 非常高优先级
  - 高优先级
  - 中优先级
  - 低优先级
- 硬件：就硬件而言，通道优先级固定。如果两个请求具有相同的软件优先级，则编号低的通道优先于编号高的通道。例如，如果通道 2 和通道 4 的软件优先级相同，则通道 2 的优先级较高。

14.3.9 FIFO

FIFO 结构用于在将源数据写入到目标之前，临时存储这些数据。所有通道共用一个中央 FIFO 结构。

为最大化数据带宽和总线使用率，使用了以下机制，从而可并行执行多个读/写操作。

- 在缓冲区传输期间，如果 FIFO 包含的数据足以用于目标突发传输，则会立即开始写操作。
- 将用于一个缓冲区传输的所有数据读入 FIFO 时，仲裁开始。之后，可将要传输的下一缓冲区数据读入 FIFO。

在缓冲区数据传输期间，如果正在工作的通道因为出错而被禁止，那么内部 FIFO 中的剩余数据将被丢弃。

14.3.10 块传输

块为“连续的”数据数组，最多支持 64 KB，通过连续缓冲区传输实现传输。

每个数据块均由起始地址和块长度进行定义。完成块传输后，可执行以下三种操作之一：

- 块为重复块传输的一部分：重载块长度，以及计算新块起始地址（基于 MDMA_CxBRUR 寄存器中的信息）
- 块为单块或重复块传输中的最后一个块：从存储器加载下一块的信息（使用 MDMA_CxLAR 中的链表地址信息）
- 块为需要针对当前 MDMA 通道传输的最后一个块 (MDMA_CxLAR = 0)：通道被禁止，且不会为该通道接受更多 MDMA 请求

14.3.11 块重复模式

块重复模式允许以源和目标的不同起始地址重复块传输。

重复块模式激活（重复传输计数器非 0）时，当前块传输结束后，将更新块参数（根据 BRSUM/BRDUM 配置重载 BNDT 值和 SAR/DAR 值），且重复传输计数器将减 1。

重复块计数器达到 0 时，会将此最后一个块视为单块传输。

14.3.12 链表模式

在链表模式下，允许从 CxLAR 寄存器的已知地址加载新 MDMA 配置（CxTCR、CxBNDTR、CxSAR、CxDAR、CxBRUR、CxLAR、CxTBR、CxMAR 和 CxMDR 寄存器）。该地址必须对 AXI 系统总线上映射的存储器进行寻址。

在进行完该操作后，通道即准备好接受新请求（如上文的块/重复块模式中所述），或者会继续传输（如果 TRGM[1:0] 等于 11）。

加载 CxTBR 值时，可自动更改触发源。

TRGM[1:0] 等于 11 时，TRGM 和 SWRM 值不能更改。

14.3.13 MDMA 传输完成

以下各种事件均可结束传输过程，并将状态寄存器 (MDMA_CxISR) 中的 CTCIFx 位置 1：

- MDMA_CxBNDTR 计数器达到零，块重复传输计数器为 0 以及链表指针地址为 0
- 传输结束前禁止了通道（通过将 MDMA_CxCR 寄存器中的 EN 位清零），并且所有的剩余数据均已从 FIFO 传输到目标

14.3.14 MDMA 传输暂停

可以随时暂停 MDMA 传输以供稍后重新开始；也可以在 MDMA 传输结束前明确禁止传输。

分为两种情况：

- 通道被禁止，以后不从停止点重新开始。在这种情况下，只需将 MDMA_CxCR 寄存器中的 EN 位清零来禁止通道，除此之外不需要任何其他操作。禁止数据流可能要花费一些时间（需要首先完成正在进行的缓冲区传输）。传输完成中断标志被置 1 表明传输已结束。此时 MDMA_CxCR 中的 EN 位值为 0，这确认了通道的数据传输已中断。MDMA_CxNDTR 寄存器包含通道停止时剩余数据项的数目，这样软件便可以确定通道中断前已传输了多少数据项。
- 在 MDMA_CxBNDTR 寄存器中要传输的剩余字节数达到 0 之前，通道暂停传输。目的是以后通过重新使能通道重新开始传输。通道传输完成中断标志 CTCIF 被置 1 表明传输已结束。如果 MDMA_CxBNDTR、SAR 和 DAR 寄存器未由软件修改，则重新使能通道时会继续进行传输。在重新启动通道前，还必须将 CTCIF 复位。

注： *如果完成的缓冲区是最后一个块，则还会在禁止通道前更新配置寄存器，以便正确准备软重启。*

注： *在重新编程通道前，软件必须等待 CTCIF 寄存器置 1，以确保正在进行的操作均已完成。*

14.3.15 错误管理

MDMA 控制器可以检测到以下错误：

当发生下列情况时，传输错误中断标志 (TEIF) 将置 1：

- MDMA 读或写访问期间发生总线错误
- 地址对齐的位置与数据的大小不匹配
- 块大小不是（源和/或目标）数据大小的倍数：最后一次传输时会激活该错误，错误地址指向最后一次传输（无法完成）

14.4 MDMA 中断

对于每个 MDMA 通道，可在发生以下事件时生成中断：

- 通道传输完成
- 块传输完成
- 块传输重复完成
- 缓冲区传输完成
- 传输错误

可以使用单独的中断使能位以实现灵活性，如*表 94* 所示。

表 94. MDMA 中断请求

中断事件	事件标志	使能控制位
通道传输完成	CTCIF	CTCIE
块传输重复完成	BTRIF	BTRIE
块传输完成	BTIF	BTIE
缓冲区传输完成	TCIF	TCIE
传输错误	TEIF	TEIE

注： *在将使能控制位置 1 前，应将相应的事件标志清零，否则如果该位已置 1，则会立即生成中断。*

注： *如果至少有一个中断标志以及相应的使能控制位置 1，则会在 GISR 中将通道中断位置 1。中断输出也会激活。如果在 NVIC 中使能了相应的中断通道，则会生成中断。*

14.5 MDMA寄存器

可按字/半字或字节格式访问 MDMA 寄存器。

14.5.1 MDMA 全局中断/状态寄存器 (MDMA_GISR0)

MDMA Global Interrupt/Status Register

偏移地址：0x00

复位值：0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GIF15	GIF14	GIF13	GIF12	GIF11	GIF10	GIF9	GIF8	GIF7	GIF6	GIF5	GIF4	GIF3	GIF2	GIF1	GIF0
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:16 保留，读为 0，用于所有未使用的通道。

位 15:0 **GIFx**：通道 x 全局中断标志 (Channel x global interrupt flag) (x=0..15)

此位由硬件置 1 和复位。它是在中断屏蔽寄存器（CTCIEx、BTIEx、BRTIEx、TEIEx）中使能的所有通道 x 中断标志（CTCIF、BTIF、BRTIF、TEIF）的逻辑或运算结果。

0：通道 x 未生成中断

1：通道 x 生成了中断

14.5.2 MDMA 通道 x 中断/状态寄存器 (MDMA_CxISR) (x = 0..15)

MDMA channel x interrupt/status register

偏移地址：0x40 + 0x40 × *通道编号*

复位值：0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TCIF	BTIF	BRTIF	CTCIF	TEIF
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:17 保留，必须保持复位值。

位 16 **CRQA**：通道 x 请求激活标志 (Channel x ReQuest Active flag)

由软件通过向 MDMA_CxCR 寄存器中的 SWRQx 位写入 1 来将该位置 1，以请求 MDMA 传输，且使能通道 x。

通道请求激活且使能通道时，该位还可由硬件置 1。硬件请求在处理前，始终被存储。

通道 x 请求完成时（因当前请求而进行最后一个缓冲区传输的源写入阶段后），该位由硬件清零。

0：对于通道 x，MDMA 传输 mdma_strx 未激活。

1：对于通道 x，MDMA 传输 mdma_strx 激活

该位还会在通道被禁止时（发生传输错误，或通道数据传输结束——重复块 = 0，链表指针为空——或在结束前由软件将通道使能位编程为 0 时）由硬件复位。

位 15:5 保留，必须保持复位值。

位 4 **TCIF**：通道 x 缓冲区传输完成中断标志 (Channel x buffer transfer complete interrupt flag)

此位将由硬件置 1， 由软件清零，软件只需将 1 写入 MDMA_IFCRy 寄存器的相应位。

0：通道 x 上无缓冲区传输完成事件

1：通道 x 上发生缓冲区传输完成事件

传输单缓冲区时 TC 置 1。它将在发出每个通道传输请求时激活。

这可用作调试功能（无中断），指示自上次标志复位起（至少）已生成一个 MDMA 缓冲区传输。

位 3 **BTIF**：通道 x 块传输完成中断标志 (Channel x block transfer complete interrupt flag)

此位将由硬件置 1， 由软件清零，软件只需将 1 写入 MDMA_IFCRy 寄存器的相应位。

0：通道 x 上无块传输完成事件

1：通道 x 上发生块传输完成事件

位 2 **BRTIF**：通道 x 块重复传输完成中断标志 (Channel x block repeat transfer complete interrupt flag)

此位将由硬件置 1， 由软件清零，软件只需将 1 写入 MDMA_IFCRy 寄存器的相应位。

0：通道 x 上无块重复传输完成事件

1：通道 x 上发生块重复传输完成事件

位 1 **CTCIF**：通道 x 通道传输完成中断标志 (Channel x Channel Transfer Complete interrupt flag)

此位将由硬件置 1， 由软件清零，软件只需将 1 写入 MDMA_IFCRy 寄存器的相应位。

0：通道 x 上无通道传输完成事件

1：通道 x 上发生通道传输完成事件

最后一个块已传输且通道已自动禁止时 CTC 置 1。

当通道因将 EN 位写 0 而暂停传输时，CTC 也置 1。

位 0 **TEIF**：通道 x 传输错误中断标志 (Channel x transfer error interrupt flag)

此位将由硬件置 1， 由软件清零，软件只需将 1 写入 MDMA_IFCRy 寄存器的相应位。

0：数据流 x 上无传输错误

1：数据流 x 上发生传输错误

14.5.3 MDMA 通道 x 中断标志清零寄存器 (MDMA_CxIFCR) (x = 0..15)

MDMA channel x interrupt flag clear register

偏移地址：0x44 + 0x40 × *通道编号*

复位值：0x0000 0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CLTCIF	CBTIF	CBRTIF	CCTCIF	CTEIF
r	r	r	r	r	r	r	r	r	r	r	w	w	w	w	w

位 31:5 保留，必须保持复位值。

位 4 **CLTCIF**：通道 x 的清零缓冲区传输完成中断标志 (CLear buffer Transfer Complete Interrupt Flag for channel x)

向该位写入 1 可将 MDMA_ISRy 寄存器中的 TCIF 清零

位 3 **CBTIF**：通道 x 清零块传输完成中断标志 (Channel x Clear block transfer complete interrupt flag)

向该位写入 1 可将 MDMA_ISRy 寄存器中的 BTIF 清零

位 2 **CBRTIF**：通道 x 清零块重复传输完成中断标志 (Channel x clear block repeat transfer complete interrupt flag)

向该位写入 1 可将 MDMA_ISRy 寄存器中的 BRTIF 清零

位 1 **CCTCIF**：通道 x 的清零通道传输完成中断标志 (Clear Channel transfer complete interrupt flag for channel x)

向该位写入 1 可将 MDMA_ISRy 寄存器中的 CTCIF 清零

位 0 **CTEIF**：通道 x 清零传输错误中断标志 (Channel x clear transfer error interrupt flag)

向该位写入 1 可将 MDMA_ISRy 寄存器中的 TEIF 清零

14.5.4 MDMA 通道 x 错误状态寄存器 (MDMA_CxESR) (x = 0..15)

MDMA Channel x error status register

偏移地址：0x48 + 0x40 × *通道编号*

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	BSE	ASE	TEMD	TELD	TED	TEA[6:0]						
				r	r	r	r	r	r	r	r	r	r	r	r

位 31:12 保留，必须保持复位值。

位 11 **BSE**：块大小错误 (Block Size Error)

块大小不是源或目标的数据大小的整数倍时，该位由硬件置 1。TED 将指示问题是出在源上还是出在目标上。

通过向 MDMA_IFCRy 寄存器中的 **CTEIF** 写入 1，此位由软件清零。

0：无块大小错误。

1：编程的块大小不是数据大小的整数倍。

位 10 **ASR**：地址/大小错误 (Address/Size Error)

设置的地址与数据大小不匹配时，该位由硬件置 1。TED 将指示问题是出在源上还是出在目标上。

通过向 MDMA_IFCRy 寄存器中的 **CTEIF** 写入 1，此位由软件清零。

0：无地址/大小错误。

1：设置的地址与数据大小不匹配。

位 9 **TEMD**：传输错误掩码数据 (Transfer Error Mask Data)

写入掩码数据发生传输错误时，该位由硬件置 1。

通过向 MDMA_IFCRy 寄存器中的 **CTEIF** 写入 1，此位由软件清零。

0：无掩码写访问错误。

1：通道上最后一次传输错误与掩码数据的写入有关。

位 8 **TELD**：传输错误链路数据 (Transfer Error Link Data)

如果在读取块链路数据结构时发生传输错误，则该位由硬件置 1。

通过向 MDMA_IFCRy 寄存器中的 **CTEIF** 写入 1，此位由软件清零。

0：无链路数据读访问错误。

1：通道上最后一次传输错误与链路数据结构的读取有关。

位 7 **TED**：传输错误方向 (Transfer Error Direction)

在出现 MDMA 数据传输错误时，该位由硬件置 1 和清零。

0：通道上最后一次传输错误与读访问有关。

1：通道上最后一次传输错误与写访问有关。

位 6:0 **TEA[6:0]**：产生传输错误的地址 (Transfer Error Address)

在出现 MDMA 数据传输错误时，这些位由硬件置 1 和清零。其与 TED 搭配使用。

该字段指示生成传输/访问错误的地址的 7 个 LSBit。

软件可以用这个值来检索失效地址，方法是将该值（截取至缓冲区传输长度大小）与当前 SAR/DAR 值相加。

注：　因存在 FIFO 管理系统，因此当前 SAR/DAR 值反映的并不是数据传输的最后一个地址。SAR/DAR 仅在结束一次缓冲区传输（长度为 TLEN+1 字节）后进行更新。

注：　出现链路数据错误时该位不会置 1。

14.5.5 MDMA 通道 x 控制寄存器 (MDMA_CxCR) (x = 0..15)

MDMA channel x control register

此寄存器用于控制相关通道。

偏移地址：0x4C + 0x40 × *通道编号*

复位值：0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	WEX	HEX	BEX	Res.	Res.	Res.	Res.	PL[1:0]		TCIE	BTIE	BRTIE	CTCIE	TEIE	EN
	rw	rw	rw					rw	rw	rw	rw	rw	rw	rw	rw

位 31:17 保留，必须保持复位值。

位 16 **SWRQ**：软件请求 (Software Request)

向该位写入 1 会将 MDMA_ISRy 寄存器的 CRQA 置 1，这将激活通道 x 的请求

注： 整个 CxCR 寄存器或偏移地址为 0x4E + 0x40 × 通道编号的 8 位/16 位寄存器可用于 SWRQ 激活。

对于软件请求，不会产生一个回应（既没有硬件信号，也没有 CxMAR 写访问）。

位 15 保留，必须保持复位值。

位 14 **WEX**：字的字节顺序交换 (Word Endianess exchange)

此位由软件置 1 和清零。

0：为字保留小端字节顺序

1：在双字中交换字顺序

该位置 1 时，会保留目标双字中的字顺序：较高地址字包含从源的较低地址中读取的数据。

如果目标不是双字，则该位的值无关紧要。

此位受到保护，只有 EN 为 0 时才可以写入。

位 13 **HEX**：半字的字节顺序交换 (Half word Endianess exchange)

此位由软件置 1 和清零。

0：为半字保留小端字节顺序

1：在各个字中交换半字顺序

该位置 1 时，会保留各个目标字中的半字顺序：较高地址半字包含从源的较低地址中读取的数据。

如果目标长度短于字的长度，则该位的值无关紧要。

此位受到保护，只有 EN 为 0 时才可以写入。

位 12 **BEX**：字节的字节顺序交换 (Byte Endianess exchange)

此位由软件置 1 和清零。

0：为字节保留小端字节顺序

1：在各个半字中交换字节顺序

该位置 1 时，会保留各个目标半字中的字节顺序：较高地址字包含从源的较低地址中读取的数据。

如果目标为字节，则该位的值无关紧要。

此位受到保护，只有 EN 为 0 时才可以写入。

位 11:9 保留，必须保持复位值。

位 8 保留，必须保持复位值。

位 7:6 **PL[1:0]**：优先级 (Priority level)
这些位将由软件置 1 和清零。
00：低
01：中
10：高
11：非常高
这些位受到保护，只有 EN 为 0 时才可以写入。

位 5 **TCIE**：缓冲区传输完成中断使能 (buffer Transfer Complete interrupt enable)
此位由软件置 1 和清零。
0：禁止 TC 中断
1：使能 TC 中断

位 4 **BTIE**：块传输中断使能 (Block Transfer interrupt enable)
此位由软件置 1 和清零。
0：禁止 BT 完成中断
1：使能 BT 完成中断

位 3 **BRTIE**：块重复传输中断使能 (Block Repeat transfer interrupt enable)
此位由软件置 1 和清零。
0：禁止 BT 中断
1：使能 BT 中断

位 2 **CTCIE**：通道传输完成中断使能 (Channel Transfer Complete interrupt enable)
此位由软件置 1 和清零。
0：禁止 TC 中断
1：使能 TC 中断

位 1 **TEIE**：传输错误中断使能 (Transfer error interrupt enable)
此位由软件置 1 和清零。
0：禁止 TE 中断
1：使能 TE 中断

位 0 **EN**：通道使能/读作低电平时通道就绪标志 (Channel enable/flag channel ready when read low)
此位由软件置 1 和清零。
0：禁止通道
1：使能通道
在以下情况下，此位可由硬件清零：
– MDMA 传输结束时（准备好配置数据流）
– AHB/AXI 主总线出现传输错误（总线错误/硬性故障）时
– 出现其他错误条件（数据对齐、块/数据大小不兼容）时
该位由软件复位时，将完成正在进行的缓冲区传输（如有）。所有状态/配置寄存器都将保持其当前值。如果在未写入这些寄存器的情况下重新使能通道，则该通道将从中断点继续。
将该位读作 0 时，允许软件对配置寄存器进行编程。将 EN 位读作 1 时，禁止向这些寄存器执行写操作（写操作将被忽略）。

注： 该位由软件复位时，建议等待 CTCIF = 1，以确保在重新编程通道之前，正在进行的缓冲区传输均已完成。

14.5.6 MDMA 通道 x 传输配置寄存器 (MDMA_CxTCR) (x = 0..15)

MDMA channel x Transfer Configuration register

此寄存器用于配置相关通道。

偏移地址：0x50 + 0x40 × *通道编号*

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
BWM	SWRM	TRGM[1:0]		PAM[1:0]		PKE	TLEN[6:0]							DBURST[2:1]	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DBURST[0]	SBURST[2:0]			DINCOS[1:0]		SINCOS[1:0]		DSIZE[1:0]		SSIZE[1:0]		DINC[1:0]		SINC[1:0]	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31 **BWM**：可缓冲的写模式 (Bufferable Write Mode)

此位由软件置 1 和清零。

0：目标写操作不可缓冲。

1：目标写操作可缓冲。

此位受到保护，只有 EN 为 0 时才可以写入。

注： 所有 MDMA 目标访问均不可缓存。

位 30 **SWRM**：软件请求模式 (Software Request Mode)

此位由软件置 1 和清零。如果硬件或软件请求处于激活状态，则位更改会被延迟，直至完成当前传输。

0：采用硬件请求考虑在内：传输的启动由 TRGM 值定义以及由 MDMA ACKx 信号确认。如果 CxMAR 包含有效地址，则还会在 CxMAR 地址处写入 CxMDR 值。

1：硬件请求被忽略。由软件通过向 SWRQ 位写入 1 来触发传输。

此位受到保护，只有 EN 为 0 时才可以写入。

位 29:28 **TRGM[1:0]**：触发模式 (Trigger Mode)

这些位将由软件置 1 和清零。

00：每个 MDMA 请求（软件或硬件）都会触发缓冲区传输

01：每个 MDMA 请求（软件或硬件）都会触发块传输

10：每个 MDMA 请求（软件或硬件）都会触发重复块传输（如果块重复值为 0，则会传输单块

11：每个 MDMA 请求（软件或硬件）都会触发针对相应通道的完整数据（例如链表）传输，直至通道结束传输并被禁止。

注： 如果当前块的 TRGM 为 11，则在当前块末尾载入的所有值均必须通过链表机制保持相同值 (TRGM=11) 以及相同的 SWRM 值，否则结果是不确定的。

这些位受到保护，只有 EN 为 0 时才可以写入。

位 27:26 **PAM[1:0]**：填充/对齐模式 (Padding/Alignement Mode)
这些位将由软件置 1 和清零。

情况 1：源数据大小小于目标数据大小——3 个选项有效。
00：右对齐，用 0 进行填充（默认值）
01：右对齐，符号扩展
10：左对齐（用 0 进行填充）
11：保留

情况 2：源数据大小大于目标数据大小。
00：右对齐——只会将源的 LSB 部分写入到目标地址
10：左对齐——只会将源的 MSB 部分写入到目标地址
其余部分将被丢弃。

PKE = 1 或 DSIZE=SSIZE 时，将忽略这些位。
这些位受到保护，只有 EN 为 0 时才可以写入

位 25 **PKE**：包使能 (Pack Enable)
此位由软件置 1 和清零。
0：按原样将源数据写入到目标中。
如果源大小小于目标大小，则会根据 PAM 值进行填充。
如果源数据大小大于目标数据大小，则会将其截断。将根据 PAM[1:0] 值进行对齐。
1：将源数据封装/解封为目标数据大小。所有数据均右对齐（采用小端模式）。
此位受到保护，只有 EN 为 0 时才可以写入

位 24:18 **TLEN[6:0]**：缓冲区传输长度 (buffer Transfer Length)（字节数 - 1）
这些位将由软件置 1 和清零。
TLEN+1 值表示要在单次传输过程中传输的字节数。
传输长度必须为（源和目标）数据大小的倍数
注： 源/目标大小不同且使用填充/截断时，TLEN+1 指的是源数据数组大小。
这些位受到保护，只有 EN 为 0 时才可以写入
必须对 DBURST 值进行编程，以确保突发大小小于传输大小。

位 17:15 **DBURST[2:0]**：目标突发传输配置 (Destination burst transfer configuration)
这些位将由软件置 1 和清零。
000：单次传输
N：2^N 个节拍的突发
这些位受到保护，只有 EN 为 0 时才可以写入
必须对 DBURST 值进行编程，以确保突发大小小于传输长度。如果无法保证这一点，将无法预测结果。
注： 如果目标总线为 TCM/AHB (DBUS=1) 且 DINCOS=11 或者 DINC=00 或 DINCOS/=DSIZE，则必须将 DBURST 编程为 000（单次传输），否则将无法预测结果。
注： 如果目标总线为系统/AXI 总线 (DBUS=0) 且 DINC=00，则 DBURST 必须为最大值 100（16 个节拍的突发），否则将无法预测结果。

位 14:12 **SBURST[2:0]**：源突发传输配置 (Source burst transfer configuration)

这些位将由软件置 1 和清零。

000：单次传输

N：2^N 个节拍的突发

这些位受到保护，只有 EN 为 0 时才可以写入

必须对 SBURST 值进行编程，以确保突发大小小于传输长度。如果无法保证这一点，将无法预测结果。

注： 如果源总线为 TCM (SBUS=1) 且 SINCOS=11 或者 SINC = 00 或 SINCOS/=SSIZE，则必须将 SBURST 编程为 000（单次传输），否则将无法预测结果。

注： 如果源总线为系统/AXI 总线 (SBUS=0) 且 SINC=00，则 SBURST 必须为最大值 100（16 个节拍的突发），否则将无法预测结果。

位 11:10 **DINCOS[1:0]**：目标增量偏移量 (Destination increment offset size)

这些位将由软件置 1 和清零。

00：字节（8 位）

01：半字（16 位）

10：字（32 位）

11：双字（64 位）

如果位 DINC[1:0] = ‘00’，则这些位没有意义。

这些位受到保护，只有 EN = ‘0’ 时才可以写入。

如果 DINCOS < DSIZE 且 DINC /= 00，将无法预测结果。

如果目标为 AHB 且 DBURST =/ 000，则目标地址必须与 DINCOS 大小一致，否则将无法预测结果。

位 9:8 **SINCOS[1:0]**：源增量偏移量 (Source increment offset size)

这些位将由软件置 1 和清零。

00：字节（8 位）

01：半字（16 位）

10：字（32 位）

11：双字（64 位）

如果位 SINC[1:0] = ‘00’，则这些位没有意义。

这些位受到保护，只有 EN = ‘0’ 时才可以写入。

如果 SINCOS < SSIZE 且 SINC /= 00，将无法预测结果。

如果源为 TCM/AHB 且 SBURST =/ 000，则源地址必须与 SINCOS 大小一致，否则将无法预测结果。

位 7:6 **DSIZE[1:0]**：目标数据大小 (Destination data size)

这些位将由软件置 1 和清零。

00：字节（8 位）

01：半字（16 位）

10：字（32 位）

11：双字（64 位）

这些位受到保护，只有 EN 为 0 时才可以写入。

注： 如果对 TCM 访问/AHB 端口编程值 11，则会发生传输错误（TEIF 位置 1）
如果 DINCOS < DSIZE 且 DINC /= 00，将无法预测结果。

注： 目标为 TCM/AHB 总线 (DBUS=1) 时，禁止 DSIZE = 11（双字）。

位 5:4 **SSIZE[1:0]**：源数据大小 (Source data size)

这些位将由软件置 1 和清零。

00：字节（8 位）

01：半字（16 位）

10：字（32 位）

11：双字（64 位）

这些位受到保护，只有 EN 为 0 时才可以写入

注： 如果对 TCM 访问/AHB 端口编程值 11，则会发生传输错误（TEIF 位置 1）

如果 SINCOS < SSIZE 且 SINC /= 00，将无法预测结果。

注： 源为 TCM/AHB 总线 (SBUS=1) 时，禁止 SSIZE = 11（双字）。

位 3:2 **DINC[1:0]**：目标增量模式 (Destination increment mode)

这些位将由软件置 1 和清零。

00：目标地址指针固定

10：每次数据传输后，目标地址指针递增（增量为 DINCOS 值）

11：每次数据传输后，目标地址指针递减（减量为 DINCOS 值）

这些位受到保护，只有 EN 为 0 时才可以写入

注： 目标为 AHB (DBUS=1) 时，禁止 DINC = 00。

位 1:0 **SINC[1:0]**：源增量模式 (Source increment mode)

这些位将由软件置 1 和清零。

00：源地址指针固定

10：每次数据传输后，源地址指针递增（增量为 SINCOS 值）

11：每次数据传输后，源地址指针递减（减量为 SINCOS 值）

这些位受到保护，只有 EN 为 0 时才可以写入

注： 源为 AHB (SBUS=1) 时，禁止 SINC = 00。

在链表模式下，在块（单块或在重复块传输模式下的最后一个块）末尾，该寄存器将从存储器进行加载（从当前 LAR[31:0] + 0x00 提供的地址进行加载）。

14.5.7 MDMA 通道 x 块数据数寄存器 (MDMA_CxBNDTR) (x = 0..15)

MDMA Channel x block number of data register

偏移地址：0x54 + 0x40 × *通道编号*

复位值：0x0000 0000

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

位 31:20 **BRC[11:0]**：块重复计数 (Block Repeat Count)

该字段包含当前块的重复数（0 到 4095）。使能通道后，此寄存器为只读型，用于指示除当前块之外的剩余块数。每次完成块传输后，此寄存器将递减。

最后一个块传输完成后，该寄存器既可保持为零，也可自动从存储器重载（在链表模式下，即，链路地址有效）。

这些位受到保护，只有 EN 为 0 时才可以写入。

位 19 **BRDUM**：块重复目标地址更新模式 (Block Repeat Destination address Update Mode)
0：块传输结束后，通过将 DUV 与当前 DAR 值（当前目标地址）相加来更新 DAR 寄存器
1：块传输结束后，通过从当前 DAR 值（当前目标地址）中减去 DUV 来更新 DAR 寄存器
这些位受到保护，只有 EN 为 0 时才可以写入。

位 18 **BRSUM**：块重复源地址更新模式 (Block Repeat Source address Update Mode)
0：块传输结束后，通过将 SUV 与当前 SAR 值（当前源地址）相加来更新 SAR 寄存器
1：块传输结束后，通过从当前 SAR 值（当前源地址）中减去 SUV 来更新 SAR 寄存器
这些位受到保护，只有 EN 为 0 时才可以写入。

位 17 保留，必须保持复位值。

位 16:0 **BNDT[16:0]**：块中待传输的数据字节数 (Block Number of data bytes to transfer)
当前块中要传输的字节数（0 到 65536）。使能通道后，此寄存器为只读型，用于指示要传输的剩余数据项数。通道活动期间，该寄存器递减，指示当前块中剩余的数据项数。
块传输完成后，该寄存器可保持为零，或者，如果已将通道配置为块重复模式，则自动重载先前编程的值。
如果该寄存器的值为零，则即使使能数据流，也无法完成任何事务。
这些位受到保护，只有 EN 为 0 时才可以写入。

注： 1：如果 BNDT 值不是 TLEN+1 值的整数倍，则最后一次传输将变短，仅包含块中的剩余数据。

注： 2：块的大小必须是源和目标数据大小的倍数。否则，会出现错误，且不会写入任何数据。

在链表模式下，在块（单块或在重复块传输模式下的最后一个块）末尾，该寄存器将从存储器进行加载（从当前 LAR[31:0] + 0x04 提供的地址进行加载）

14.5.8 MDMA 通道 x 源地址寄存器 (MDMA_CxSAR) (x = 0..15)

MDMA channel x source address register

偏移地址：0x58 + 0x40 × *通道编号*

复位值：0x0000 0000

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

位 31:0 **SAR[31:0]**：源地址 (Source address)
这些位表示从其读取数据或将数据读至的外设数据寄存器的基址。其必须与 SSIZE 对齐（例如，SSIZE=10 时 SAR[1:0] = 00），但可不与 SINCOS 对齐。
源为 TCM/AHB 时，如果地址不与 SINCOS 对齐，则必须将访问编程为单次访问 (SBURST=000)。
这些位受到写保护，只有 DMA_SxCR 寄存器中的 EN 为“0”时才可以写入。
在通道活动期间该寄存器将更新，反映了下次要从中读取数据的当前地址。
如果块重复模式激活，则在完成块传输后，通过将 SAU 值与当前值相加/相减来更新源地址（已在块中的最后一次传输后进行更新）。
链表模式激活时，在块（重复块或非重复块）传输结束后，将从存储器（从地址 LSA + m）加载 SAR 值。

在链表模式下，在块（单块或在重复块传输模式下的最后一个块）末尾，该寄存器将从存储器进行加载（从当前 LAR[31:0] + 0x08 提供的地址进行加载）。

14.5.9 MDMA 通道 x 目标地址寄存器 (MDMA_CxDAR) (x = 0..15)

MDMA channel x destination address register

偏移地址：0x5C + 0x40 × *通道编号*

复位值：0x0000 0000

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

位 31:0 **DAR[31:0]**：目标地址 (Destination address)

要向其中写入数据的目标地址的基址。

这些位受到写保护，只有 DMA_SxCR 寄存器中的 EN 为“0”时才可以写入。

必须与 DSIZE 对齐（例如，DSIZE=01 时 DAR[0] = 0），但可不与 DINCOS 对齐。

目标为 AHB 时，如果地址不与 DINCOS 对齐，则必须将访问编程为单次访问 (DBURST=000)。

在通道活动期间该寄存器将更新，反映了下次要向其中写入数据的当前地址。

如果块重复模式激活，则在完成块传输后，通过将 DAU 值与当前值相加/相减来更新目标地址（已在块中的最后一次传输后进行更新）。

链表模式激活时，在块（重复块或非重复块）传输结束后，将从存储器（从地址 LSA + m）加载 DAR 值。

在链表模式下，在块（单块或在重复块传输模式下的最后一个块）末尾，该寄存器将从存储器进行加载（从当前 LAR[31:0] + 0x0C 提供的地址进行加载）。

14.5.10 MDMA 通道 x 块重复地址更新寄存器 MDMA_CxBRUR (x = 0..15)

MDMA channel x Block Repeat address Update register

偏移地址：0x60 + 0x40 × *通道编号*

复位值：0x0000 0000

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

位 31:16 **DUV[15:0]**：目标地址更新值 (Destination address Update Value)

该值用于在块传输结束后更新（通过加法或减法）当前目标地址。必须为 DSIZE 的整数倍，以便使 DAR 与 DSIZE 对齐（例如，DSIZE=10 时 DAR[1:0] = 00）。

如果该值为 0，下次重复的块传输的目标地址将从下一地址继续。

块重复模式未激活 (BRC=0) 时，将忽略该字段。

这些位受到写保护，只有 MDMA_CxCR 寄存器中的 EN 为“0”时才可以写入。

注： DINC[1:0] = 00 时必须将该字段编程为 0。

位 15:0 **SUV[15:0]**：源地址更新值 (Source address Update Value)

该值用于在块传输结束后更新（通过加法或减法）当前源地址。必须为 SSIZE 的整数倍，以便使 SAR 与 SSIZE 对齐（例如，SSIZE=10 时 SAR[1:0] = 00）。

如果该值为 0，下次重复的块传输的源地址将从下一地址继续。

块重复模式未激活 (BRC=0) 时，将忽略该字段。

这些位受到写保护，只有 MDMA_CxCR 寄存器中的 EN 为“0”时才可以写入。

注： SINC[1:0] = 00 时必须将该字段编程为 0。

在链表模式下，在块（单块或在重复块传输模式下的最后一个块）末尾，该寄存器将从存储器进行加载（从当前 LAR[31:0] + 0x10 提供的地址进行加载）

14.5.11 MDMA 通道 x 链路地址寄存器 (MDMA_CxLAR) (x = 0..15)

MDMA channel x Link Address register

偏移地址：0x64 + 0x40 × *通道编号*

复位值：0x0000 0000

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

位 31:0 **LAR[31:0]**：链路地址寄存器 (Link Address Register)

（重复）块传输结束后，当前通道配置寄存器（CxTCR、CxBNDTR、CxSAR、CxDAR、CxBRUR、CxMAR、CxMDR 和 CxLAR 寄存器本身）使用该地址处的数据结构进行加载。

如果该寄存器的值为 0，则不会进行寄存器更新，通道将被禁止且 CTCIF 将置 1，指示该通道的传输结束。这些位受到写保护，只有 MDMA_CxCR 寄存器中的 EN 为“0”时才可以写入。

通道配置（LAR 地址）必须处于 AXI 地址空间内。

LAR 值必须在双字地址处匹配，即 LAR[2:0] = 0x0

在链表模式下，在块（单块或在重复块传输模式下的最后一个块）末尾，该寄存器将从存储器进行加载（从当前 LAR[31:0] + 0x14 提供的地址进行加载）。

注： 只有在所有寄存器均更新后，才会为下次块结束采用新值。

14.5.12 MDMA 通道 x 触发和总线选择寄存器 (MDMA_CxTBR) (x = 0..15)

MDMA channel x Trigger and Bus selection Register

偏移地址：0x68 + 0x40 × *通道编号*

复位值：0x0000 0000

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

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

位 31:18 保留，必须保持复位值。

位 17 **DBUS**：目标总线选择 (Destination BUS select)

0：将系统/AXI 总线用作通道 x 的目标（写操作）。

1：将 AHB 总线/TCM 用作通道 x 的目标（写操作）。

此位受到保护，只有 EN 为 0 时才可以写入。

位 16 **SBUS**：源总线选择 (Source BUS select)

0：将系统/AXI 总线用作通道 x 的源（读操作）。

1：将 AHB 总线/TCM 用作通道 x 的源（读操作）。

此位受到保护，只有 EN 为 0 时才可以写入。

位 15:6 保留，必须保持复位值。

位 5:0 **TSEL[5:0]**：触发选择 (Trigger selection)

该位域用于为通道 x 选择硬件触发 (RQ) 输入。在具有相同索引值的 ACK 输出上发送 ACK。

如果 SWRM 位置 1（已选择软件请求），则会忽略该位域。

这些位受到写保护，只有 MDMA_CxCR 寄存器中的 EN 为“0”时才可以写入。

注： 如果同一事件触发多个通道（具有相同 TSEL 值），则会并行触发所有这些通道。不过，只有具有最低索引的通道才会确认请求。

在链表模式下，在块（单块或在重复块传输模式下的最后一个块）末尾，该寄存器将从存储器进行加载（从当前 LAR[31:0] + 0x18 提供的地址进行加载）

14.5.13 MDMA 通道 x 掩码地址寄存器 (MDMA_CxMAR) (x = 0..15)

MDMA channel x Mask address register

偏移地址：0x70 + 0x40 × *通道编号*

复位值：0x0000 0000

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

位 31:0 **MAR[31:0]**：掩码地址 (Mask address)

也会对该地址写入 MDR 值。这样可以通过写入中断清零寄存器来清除 DMA2 生成的 RQ 信号。
如果该寄存器的值为 0，则会禁止该功能。这些位受到写保护，只有 MDMA_CxCR 寄存器中的 EN 为“0”时才可以写入。

在链表模式下，在块（单块或在重复块传输模式下的最后一个块）末尾，该寄存器将从存储器进行加载（从当前 LAR[31:0] + 0x20 提供的地址进行加载）

14.5.14 MDMA 通道 x 掩码数据寄存器 (MDMA_CxMDR) (x = 0..15)

MDMA channel x Mask Data register

偏移地址：0x74 + 0x40 × *通道编号*

复位值：0x0000 0000

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

位 31:0 **MDR[31:0]**：掩码数据 (Mask Data)

也会对 MAR 寄存器定义的地址写入 MDR 值。这样可以通过写入中断清零寄存器来清除 DMA2 生成的 RQ 信号。
这些位受到写保护，只有 MDMA_CxCR 寄存器中的 EN 为“0”时才可以写入。

在链表模式下，在块（单块或在重复块传输模式下的最后一个块）末尾，该寄存器将从存储器进行加载（从当前 LAR[31:0] + 0x24 提供的地址进行加载）

14.5.15 MDMA寄存器映射

表 95 汇总了 MDMA 寄存器。

表 95. MDMA 寄存器映射和复位值

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	MDMA_GISR0	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	GIF15	GIF14	GIF13	GIF12	GIF11	GIF10	GIF9	GIF8	GIF7	GIF6	GIF5	GIF4	GIF3	GIF2	GIF1	GIF0
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x04 - 0x3C	Reserved	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res
	Reset value																																
0x40 + 0x40 × 通道编号	MDMA_CxISR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CRQA	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TCIF	BTIF	BRTIF	CTCIF	TEIF
	Reset value																0												0	0	0	0	0
0x44 + 0x40 × 通道编号	MDMA_CxIFCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CLTCIF	CBTIF	CBRTIF	CCTCIF	CTEIF
	Reset value																												0	0	0	0	0
0x48 + 0x40 × 通道编号	MDMA_CxESR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TEMD	TELD	TED	TEA[6:0]						
	Reset value																							0	0	0	0	0	0	0	0	0	0
0x4C + 0x40 × 通道编号	MDMA_CxCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SWRQ	Res.	WEX	HEX	BEX	Res.	Res.	Res.	Res.	PL[1:0]		TCIE	BTIE	BRTIE	CTCIE	TEIE	EN
	Reset value																0		0	0	0					0	0	0	0	0	0	0	0
0x50 + 0x40 × 通道编号	MDMA_CxTCR	BWM	SWRM	TRGM[1:0]		PAM[1:0]		PKE	TLEN[6:0].							DBURST[2:0]		SBURST[2:0]			DINCOS[1:0]		SINCOS[1:0]		DSIZE[1:0]		SSIZE[1:0]		DINC[1:0]		SINC[1:0]	
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x54 + 0x40 × 通道编号	MDMA_CxBNDTR	BRC[11:0]												BRDUM	BRSUM	Res.	BNDT[16:0]																
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x58 + 0x40 × 通道编号	MDMA_CxSAR	SAR[31:0].																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x5C + 0x40 × 通道编号	MDMA_CxDAR	DAR[31:0].																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x60 + 0x40 × 通道编号	MDMA_CxBRUR	DUV[15:0].																SUV[15:0].															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x64 + 0x40 × 通道编号	MDMA_CxLAR	LAR[31:0].																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x68 + 0x40 × 通道编号	MDMA_CxTBR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DBUS	SBUS	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TSEL[5:0]					
	Reset value															0	0											0	0	0	0	0	0
0x6C + 0x40 × 通道编号	Reserved	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res
	Reset value																																
0x70 + 0x40 × 通道编号	MDMA_CxMAR	MAR[31:0].																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

表 95. MDMA 寄存器映射和复位值（续）

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x74 + 0x40 × 通道编号	MDMA_CxMDR	MDR[31:0].																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x74 - 0x7C + 0x40 × 通道编号	Reserved	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	Reset value																																

有关寄存器边界地址的信息，请参见*第 2.2.2 节：存储器映射和寄存器边界地址*。

15 直接存储器访问控制器（DMA1、DMA2）

15.1 DMA 简介

直接存储器访问 (DMA) 用于在外设与存储器之间以及存储器与存储器之间提供高速数据传输。可以在无需任何 CPU 操作的情况下通过 DMA 快速移动数据。这样节省的 CPU 资源可供其它操作使用。

DMA 控制器基于复杂的总线矩阵架构，将功能强大的双 AHB 主总线架构与独立的 FIFO 结合在一起，优化了系统带宽。

两个 DMA 控制器总共有 16 个数据流（每个控制器 8 个），每一个 DMA 控制器都用于管理一个或多个外设的存储器访问请求。每个数据流总共可以有多达 8 个通道（或称请求）。每个通道都有一个仲裁器，用于处理 DMA 请求间的优先级。

15.2 DMA 主要特性

DMA 主要特性是：

- 双 AHB 主总线架构，一个用于存储器访问，另一个用于外设访问
- 仅支持 32 位访问的 AHB 从编程接口
- 每个 DMA 控制器有 8 个数据流，每个数据流有多达 115 个通道（或称请求）
- 每个数据流有四个字深的 32 位先进先出存储器缓冲区 (FIFO)，可用于 FIFO 模式或直接模式：
  - FIFO 模式：可通过软件将阈值级别选取为 FIFO 大小的 1/4、1/2 或 3/4
  - 直接模式

    每个 DMA 请求会立即启动对存储器的传输。当在直接模式（禁止 FIFO）下将 DMA 请求配置为以存储器到外设模式传输数据时，DMA 仅会将一个数据从存储器预加载到内部 FIFO，从而确保一旦外设触发 DMA 请求时则立即传输数据。
- 通过硬件可以将每个数据流配置为：
  - 支持外设到存储器、存储器到外设和存储器到存储器传输的常规通道
  - 在存储器端支持双缓冲的双缓冲区通道
- 8 个数据流中的每一个都连接到专用硬件 DMA 通道（请求）
- DMA 数据流请求之间的优先级可用软件编程（4 个级别：非常高、高、中、低），在软件优先级相同的情况下可以通过硬件决定优先级（例如，请求 0 的优先级高于请求 1）
- 每个数据流还支持通过软件触发的存储器间的传输
- 可通过 DMAMux1 在多达 115 个可能的通道请求中选择每个数据流请求。此选择可由软件配置，允许多个外设发起 DMA 请求
- 要传输的数据项的数目可以由 DMA 控制器或外设管理：
  - DMA 流控制器：要传输的数据项的数目可用软件编程，从 1 至 65535
  - 外设流控制器：要传输的数据项的数目未知并由源或目标外设控制，这些外设通过硬件发出传输结束的信号
- 独立的源和目标传输宽度（字节、半字、字）：源和目标的数据宽度不相等时，DMA 自动封装/解封必要的传输数据来优化带宽。这个特性仅在 FIFO 模式下可用
- 对源和目标的增量或非增量寻址
- 支持 4 个、8 个和 16 个节拍的增量突发传输。突发增量的大小可由软件配置，通常等于外设 FIFO 大小的一半

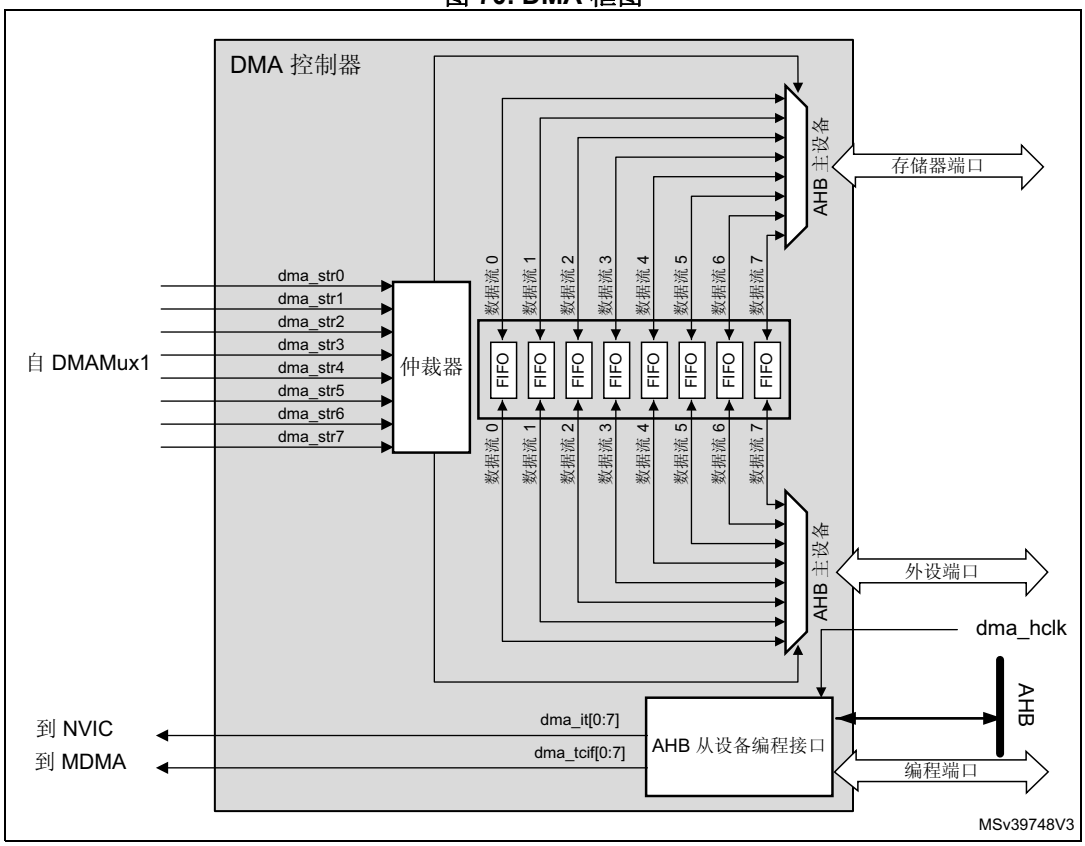
- 每个数据流都支持循环缓冲区管理
- 5 个事件标志（DMA 半传输、DMA 传输完成、DMA 传输错误、DMA FIFO 错误、直接模式错误），进行逻辑或运算，从而产生每个数据流的单个中断请求

15.3 DMA 功能说明

15.3.1 DMA 框图

图 70 显示了 DMA 的框图。

图 70. DMA 框图



15.3.2 DMA 内部信号

表 96 显示了内部 DMA 信号。

表 96. DMA 内部输入/输出信号

信号名称	信号类型	说明
dma_hclk	数字输入	DMA AHB 时钟
dma_it[0:7]	数字输出	DMA 数据流 [0:7] 全局中断
dma_tcif[0:7]	数字输出	MDMA 触发
dma_str[0:7]	数字输入	DMA 数据流 [0:7] 请求

15.3.3 DMA概述

DMA 控制器执行直接存储器传输：做为一个 AHB 主设备，它可以控制 AHB 总线矩阵来发起 AHB 传输。

它可以执行下列传输：

- 外设到存储器的传输
- 存储器到外设的传输
- 存储器到存储器的传输

DMA 控制器提供两个 AHB 主端口：*AHB 存储器端口*（用于连接存储器）和 *AHB 外设端口*（用于连接外设）。但是，要执行存储器到存储器的传输，*AHB 外设端口*必须也能访问存储器。

AHB 从端口用于对 DMA 控制器进行编程（它仅支持 32 位访问）。

15.3.4 DMA 传输

DMA 传输由给定数目的数据传输序列组成。要传输的数据项的数目及其宽度（8 位、16 位或 32 位）可用软件编程。

每个 DMA 传输包含三项操作：

- 通过 DMA_SxPAR 或 DMA_SxM0AR 寄存器寻址，从外设数据寄存器或存储器单元中加载数据
- 通过 DMA_SxPAR 或 DMA_SxM0AR 寄存器寻址，将加载的数据存储到外设数据寄存器或存储器单元
- DMA_SxNDTR 计数器在数据存储结束后递减，该计数器中包含仍需执行的事务数。

在产生事件后，外设会向 DMA 控制器发送请求信号。DMA 控制器根据通道优先级处理该请求。只要 DMA 控制器访问外设，DMA 控制器就会向外设发送确认信号。外设获得 DMA 控制器的确认信号后，便会立即释放其请求。一旦外设使请求失效，DMA 控制器就会释放确认信号。如果有更多请求，外设可以启动下一个事务。

15.3.5 DMA 请求映射

*第 17.3.2 节：DMAMUX1 映射*对映射到外设和 DMA 通道的 DMA 请求进行了介绍。

15.3.6 仲裁器

仲裁器为两个 AHB 主端口（存储器和外设端口）提供基于请求优先级的 8 个 DMA 数据流请求管理，并启动外设/存储器访问序列。

优先级管理分为两个阶段：

- 软件：每个数据流优先级都可以在 DMA_SxCR 寄存器中配置。分为四个级别：
  - 非常高优先级
  - 高优先级
  - 中优先级
  - 低优先级
- 硬件：如果两个请求具有相同的软件优先级，则编号低的数据流优先于编号高的数据流。例如，数据流 2 的优先级高于数据流 4。

15.3.7 DMA 数据流

8 个 DMA 控制器数据流都能够提供源和目标之间的单向传输链路。

每个数据流配置后都可以执行：

- 常规类型事务：存储器到外设、外设到存储器或存储器到存储器的传输。
- 双缓冲区类型事务：使用两个存储器指针的双缓冲区传输（当 DMA 正在进行自/至缓冲区的读/写操作时，应用程序可以进行至/自其它缓冲区的写/读操作）。

要传输的数据量（多达 65535）可以编程，并与连接到外设 AHB 端口的外设（请求 DMA 传输）的源宽度相关。每个事务完成后，包含要传输的数据项总量的寄存器都会递减。

15.3.8 源、目标和传输模式

源传输和目标传输在整个 4 GB 区域（地址在 0x0000 0000 和 0xFFFF FFFF 之间）都可以寻址外设和存储器。

传输方向使用 DMA_SxCR 寄存器中的 DIR[1:0] 位进行配置，有三种可能的传输方向：存储器到外设、外设到存储器或存储器到存储器。*表 97* 介绍了相应的源和目标地址。

表 97. 源和目标地址

DMA_SxCR 寄存器的位 DIR[1:0]	方向	源地址	目标地址
00	外设到存储器	DMA_SxPAR	DMA_SxM0AR
01	存储器到外设	DMA_SxM0AR	DMA_SxPAR
10	存储器到存储器	DMA_SxPAR	DMA_SxM0AR
11	保留	-	-

当数据宽度（在 DMA_SxCR 寄存器的 PSIZE 或 MSIZE 位中编程）分别是半字或字时，写入 DMA_SxPAR 或 DMA_SxM0AR/M1AR 寄存器的外设或存储器地址必须分别在字或半字地址的边界对齐。

外设到存储器模式

图 71 介绍了这种模式。

使能这种模式（将 DMA_SxCR 寄存器中的位 EN 置 1）时，每次产生外设请求，数据流都会启动数据源到 FIFO 的传输。

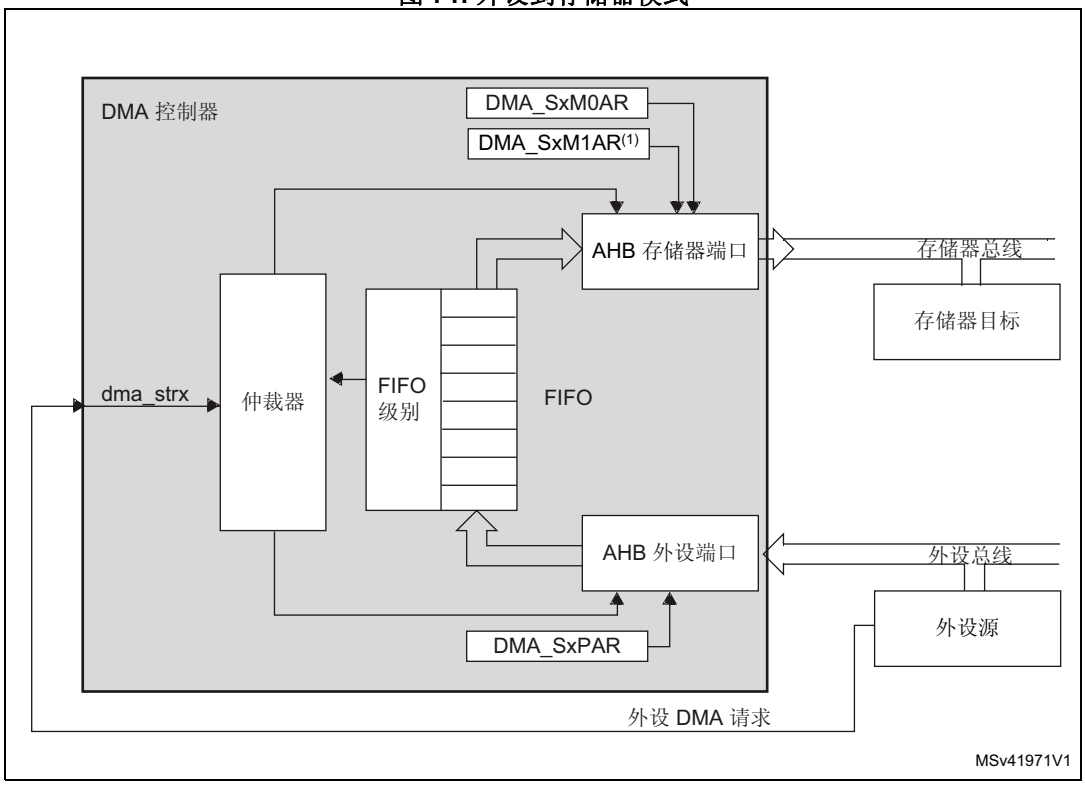
达到 FIFO 的阈值级别时，FIFO 的内容移出并存储到目标中。

如果 DMA_SxNDTR 寄存器达到零、外设请求传输终止（在使用外设流控制器的情况下）或 DMA_SxCR 寄存器中的 EN 位由软件清零，传输即会停止。

在直接模式下（当 DMA_SxFCR 寄存器中的 DMDIS 值为“0”时），不使用 FIFO 的阈值级别控制：每完成一次从外设到 FIFO 的数据传输后，相应的数据立即就会移出并存储到目标中。

只有赢得了数据流的仲裁后，相应数据流才有权访问 AHB 源或目标端口。系统使用在 DMA_SxCR 寄存器 PL[1:0] 位中为每个数据流定义的优先级执行仲裁。

图 71. 外设到存储器模式



1. 用于双缓冲区模式。

存储器到外设模式

图 72 介绍了这种模式。

使能这种模式（将 DMA_SxCR 寄存器中的 EN 位置 1）时，数据流会立即启动传输，从源完全填充 FIFO。

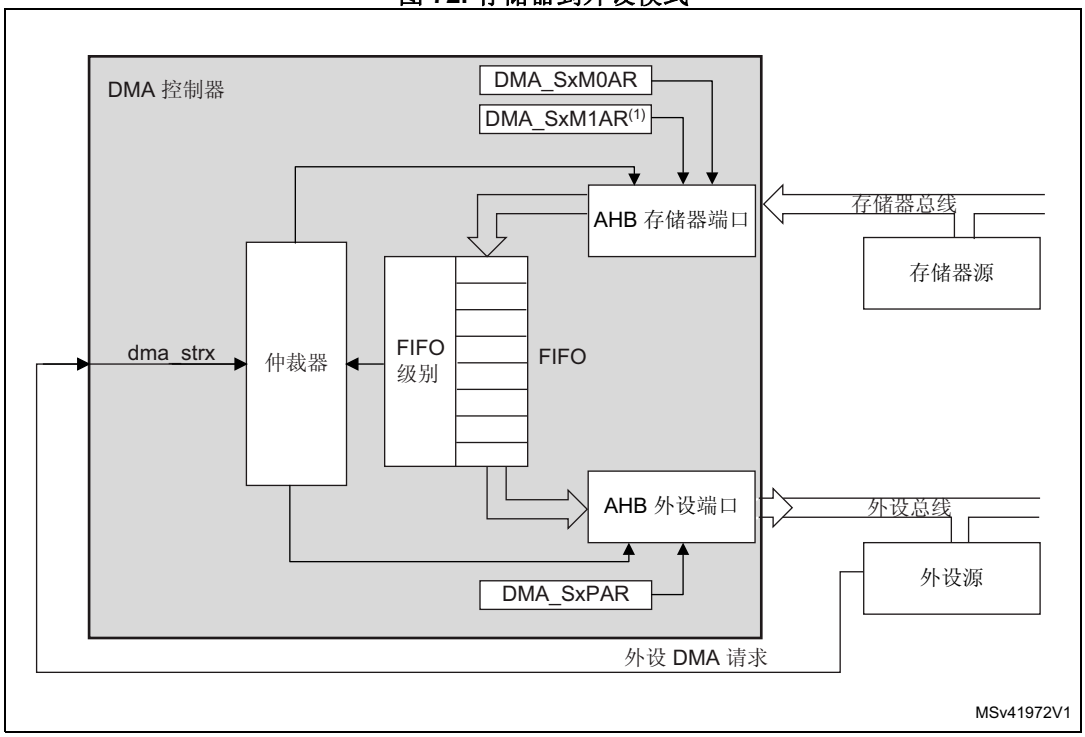
每次发生外设请求，FIFO 的内容都会移出并存储到目标中。当 FIFO 的级别小于或等于预定义的阈值级别时，将使用存储器中的数据完全重载 FIFO。

如果 DMA_SxNDTR 寄存器达到零、外设请求传输终止（在使用外设流控制器的情况下）或 DMA_SxCR 寄存器中的 EN 位由软件清零，传输即会停止。

在直接模式下（当 DMA_SxFCR 寄存器中的 DMDIS 值为“0”时），不使用 FIFO 的阈值级别。一旦使能了数据流，DMA 便会预装载第一个数据，将其传输到内部 FIFO。一旦外设请求数据传输，DMA 便会将预装载的值传输到配置的目标。然后，它会使用要传输的下一个数据再次重载内部空 FIFO。预装载的数据大小为 DMA_SxCR 寄存器中 PSIZE 位域的值。

只有赢得了数据流的仲裁后，相应数据流才有权访问 AHB 源或目标端口。系统使用在 DMA_SxCR 寄存器 PL[1:0] 位中为每个数据流定义的优先级执行仲裁。

图 72. 存储器到外设模式



1. 用于双缓冲区模式。

存储器到存储器模式

DMA 通道在没有外设请求触发的情况下同样可以工作。此为*图 73* 中介绍的存储器到存储器模式。

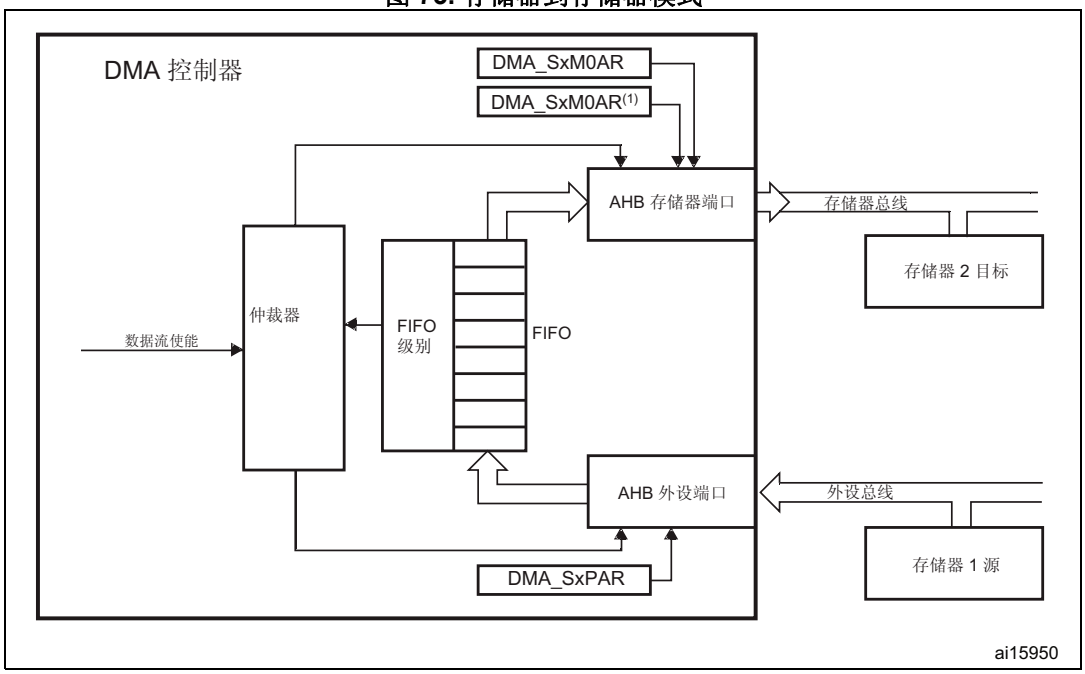
通过将 DMA_SxCR 寄存器中的使能位 (EN) 置 1 来使能数据流时，数据流会立即开始填充 FIFO，直至达到阈值级别。达到阈值级别后，FIFO 的内容便会移出，并存储到目标中。

如果 DMA_SxNDTR 寄存器达到零或 DMA_SxCR 寄存器中的 EN 位由软件清零，传输即会停止。

只有赢得了数据流的仲裁后，相应数据流才有权访问 AHB 源或目标端口。系统使用在 DMA_SxCR 寄存器 PL[1:0] 位中为每个数据流定义的优先级执行仲裁。

注： *使用存储器到存储器模式时，不允许循环模式和直接模式。*

图 73. 存储器到存储器模式



1. 用于双缓冲区模式。

15.3.9 指针递增

根据 DMA_SxCR 寄存器中 PINC 和 MINC 位的状态，外设和存储器指针在每次传输后可以自动向后递增或保持常量。

通过单个寄存器访问外设源或目标数据时，禁止递增模式十分有用。

如果使能了递增模式，则根据在 DMA_SxCR 寄存器 PSIZE 或 MSIZE 位中编程的数据宽度，下一次传输的地址将是前一次传输的地址递增 1（对于字节）、2（对于半字）或 4（对于字）。

为了优化封装操作，可以不管 AHB 外设端口上传输的数据的大小，将外设地址的增量偏移大小固定下来。DMA_SxCR 寄存器中的 PINCOS 位用于将增量偏移大小与外设 AHB 端口或 32 位地址（此时地址递增 4）上的数据大小对齐。PINCOS 位仅对 AHB 外设端口有影响。

如果将 PINCOS 位置 1，则不论 PSIZE 值是多少，下一次传输的地址总是前一次传输的地址递增 4（自动与 32 位地址对齐）。但是，AHB 存储器端口不受此操作影响。

15.3.10 循环模式

循环模式可用于处理循环缓冲区和连续数据流（例如 ADC 扫描模式）。可以使用 DMA_SxCR 寄存器中的 CIRC 位使能此特性。

当激活循环模式时，要传输的数据项的数目在数据流配置阶段自动用设置的初始值进行加载，并继续响应 DMA 请求。

注： *在循环模式下，如果为存储器配置了突发模式，必须遵循下列规则：*

DMA_SxNDTR = ((Mburst 节拍) × (Msize)/(Psize)) 的倍数，其中：

- *(Mburst 节拍) = 4、8 或 16（取决于 DMA_SxCR 寄存器中的 MBURST 位）*
- *((Msize)/(Psize)) = 1、2、4、1/2 或 1/4（Msize 和 Psize 表示 DMA_SxCR 寄存器中的 MSIZE 和 PSIZE 位。它们与字节相关）*
- *DMA_SxNDTR = AHB 外设端口上要传输的数据项的数目*

Mburst 节拍 = 8 (INCR8)，MSIZE = “00”（字节）和 PSIZE = “01”（半字），在此例中：DMA_SxNDTR 必须是 (8 × 1/2 = 4) 的倍数。

如果不遵循此公式，则 DMA 行为和数据完整性得不到保证。

NDTR 还必须是外设突发大小与外设数据大小乘积的倍数，否则会导致错误的 DMA 行为。

15.3.11 双缓冲区模式

此模式可用于所有 DMA1 和 DMA2 数据流。

通过将 DMA_SxCR 寄存器中的 DBM 位置 1，即可使能双缓冲区模式。

除了有两个存储器指针之外，双缓冲区数据流的工作方式与常规（单缓冲区）数据流的一样。使能双缓冲区模式时，将自动使能循环模式（DMA_SxCR 中的 CIRC 位的状态是“无关”），并在每次事务结束时交换存储器指针。

在此模式下，每次事务结束时，DMA 控制器都从一个存储器目标交换为另一个存储器目标。这样，软件在处理一个存储器区域的同时，DMA 传输还可以填充/使用第二个存储器区域。如*表 98：双缓冲区模式下的源和目标地址寄存器 (DBM=1)* 所述，双缓冲区数据流可以双向工作（存储器既可以是源也可以是目标）。

注： *在双缓冲区模式下使能数据流时，可遵循下列条件，实时更新 AHB 存储器的基址（DMA_SxM0AR 或 DMA_SxM1AR）：*

- *当 DMA_SxCR 寄存器中的 CT 位为“0”时，可以写入 DMA_SxM1AR 寄存器。当 CT = “1”时，试图写入此寄存器会将错误标志位 (TEIF) 置 1，并自动禁止数据流。*
- *当 DMA_SxCR 寄存器中的 CT 位为“1”时，可以写入 DMA_SxM0AR 寄存器。当 CT = “0”时，试图写入此寄存器会将错误标志位 (TEIF) 置 1，并自动禁止数据流。*

为避免出现任何错误状态，建议在 TCIF 标志位置位时立即更改基址。因为此时根据上述两个条件之一，目标存储器依据 DMA_SxCR 寄存器中 CT 值的情况，一定已从存储器 0 更改为存储器 1（或从存储器 1 更改为存储器 0）。

对于所有其它模式（双缓冲区模式除外），一旦使能数据流，存储器地址寄存器即被写保护。

表 98. 双缓冲区模式下的源和目标地址寄存器 (DBM=1)

DMA_SxCR 寄存器的位 DIR[1:0]	方向	源地址	目标地址
00	外设到存储器	DMA_SxPAR	DMA_SxM0AR/DMA_SxM1AR
01	存储器到外设	DMA_SxM0AR/DMA_SxM1AR	DMA_SxPAR
10	不允许(1)		
11	保留	-	-

1. 使能双缓冲区模式时，自动使能循环模式。由于存储器到存储器模式与循环模式不兼容，所以当使能双缓冲区模式时，不允许配置存储器到存储器模式。

15.3.12 可编程数据宽度、封装/解封、字节序

要传输的数据项数目必须在使能数据流之前编程到 DMA_SxNDTR（要传输数据项数目位，NDT）中，当流控制器是外设且 DMA_SxCR 中的 PFCTRL 位置为 1 时除外。

当使用内部 FIFO 时，源和目标数据的数据宽度可以通过 DMA_SxCR 寄存器的 PSIZE 和 MSIZE 位（可以是 8、16 或32 位）编程。

当 PSIZE 和 MSIZE 不相等时：

- 在 DMA_SxNDTR 寄存器中配置的要传输的数据项数目的数据宽度等于外设总线的宽度（由 DMA_SxCR 寄存器中的 PSIZE 位配置）。例如，在外设到存储器、存储器到外设或存储器到存储器传输的情况下，如果将 PSIZE[1:0] 位配置为半字，则要传输的字节数等于 2 × NDT。
- DMA 控制器仅按小字节序寻址源和目标。相关内容在*表 99：封装/解封和字节序行为（位 PINC = MINC = 1）*中介绍。

在封装/解封数据的过程中，如果在数据完全封装/解封前中断操作，则有数据损坏的危险。因此，为了确保数据一致性，可将数据流配置成生成突发传输：在这种情况下，属于一个突发的每组传输不可分割（请参见*第 15.3.13 节：单次传输和突发传输*）。

在直接模式下（DMA_SxFCR 寄存器中的 DMDIS = 0），不能进行数据封装/解封。这种情况下，不允许源与目标的传输数据宽度不同，二者必须相等，并由 DMA_SxCR 中的 PSIZE 位定义，MSIZE 位的状态是“无关”。

表 99. 封装/解封和字节序行为（位 PINC = MINC = 1）

AHB 存储器端口宽度	AHB 外设端口宽度	要传输的数据项的数目 (NDT)	存储器传输数目	存储器端口地址/字节通道	外设传输数目	外设端口地址/字节通道	
						PINCOS = 1	PINCOS = 0
8	8	4	1 2 3 4	0x0/B0[7:0] 0x1/B1[7:0] 0x2/B2[7:0] 0x3/B3[7:0]	1 2 3 4	0x0/B0[7:0] 0x4/B1[7:0] 0x8/B2[7:0] 0xC/B3[7:0]	0x0/B0[7:0] 0x1/B1[7:0] 0x2/B2[7:0] 0x3/B3[7:0]
8	16	2	1 2 3 4	0x0/B0[7:0] 0x1/B1[7:0] 0x2/B2[7:0] 0x3/B3[7:0]	1 2	0x0/B1|B0[15:0] 0x4/B3|B2[15:0]	0x0/B1|B0[15:0] 0x2/B3|B2[15:0]
8	32	1	1 2 3 4	0x0/B0[7:0] 0x1/B1[7:0] 0x2/B2[7:0] 0x3/B3[7:0]	1	0x0/B3|B2|B1|B0[31:0]	0x0/B3|B2|B1|B0[31:0]
16	8	4	1 2	0x0/B1|B0[15:0] 0x2/B3|B2[15:0]	1 2 3 4	0x0/B0[7:0] 0x4/B1[7:0] 0x8/B2[7:0] 0xC/B3[7:0]	0x0/B0[7:0] 0x1/B1[7:0] 0x2/B2[7:0] 0x3/B3[7:0]
16	16	2	1 2	0x0/B1|B0[15:0] 0x2/B1|B0[15:0]	1 2	0x0/B1|B0[15:0] 0x4/B3|B2[15:0]	0x0/B1|B0[15:0] 0x2/B3|B2[15:0]
16	32	1	1 2	0x0/B1|B0[15:0] 0x2/B3|B2[15:0]	1	0x0/B3|B2|B1|B0[31:0]	0x0/B3|B2|B1|B0[31:0]
32	8	4	1	0x0/B3|B2|B1|B0[31:0]	1 2 3 4	0x0/B0[7:0] 0x4/B1[7:0] 0x8/B2[7:0] 0xC/B3[7:0]	0x0/B0[7:0] 0x1/B1[7:0] 0x2/B2[7:0] 0x3/B3[7:0]
32	16	2	1	0x0/B3|B2|B1|B0[31:0]	1 2	0x0/B1|B0[15:0] 0x4/B3|B2[15:0]	0x0/B1|B0[15:0] 0x2/B3|B2[15:0]
32	32	1	1	0x0/B3|B2|B1|B0 [31:0]	1	0x0/B3|B2|B1|B0 [31:0]	0x0/B3|B2|B1|B0[31:0]

注： *外设端口可以是源或目标（在存储器到存储器传输的情况下，也可能是存储器源）。*

必须配置 PSIZE、MSIZE 和 NDT[15:0]，以确保最后一次传输的完整性。当外设端口的数据宽度（PSIZE 位）小于存储器端口的数据宽度（MSIZE 位）时，可能会发生数据传输不完整的情况。此限制条件汇总在*表 100* 中。

表 100. PSIZE 与 MSIZE 确定时对 NDT 的限制条件

DMA_SxCR 的 PSIZE[1:0]	DMA_SxCR 的 MSIZE[1:0]	DMA_SxNDTR 的 NDT[15:0]
00（8 位）	01（16 位）	必须是 2 的倍数
00（8 位）	10（32 位）	必须是 4 的倍数
01（16 位）	10（32 位）	必须是 2 的倍数

15.3.13 单次传输和突发传输

DMA 控制器可以产生单次传输或 4 个、8 个和 16 个节拍的增量突发传输。

突发大小通过软件针对两个 AHB 端口独立配置，配置时使用 DMA_SxCR 寄存器中的 MBURST[1:0] 和 PBURST[1:0] 位。

突发大小指示突发中的节拍数，而不是传输的字节数。

为确保数据一致性，形成突发的每一组传输都不可分割：在突发传输序列期间，AHB 传输会锁定，并且 AHB 总线矩阵的仲裁器不解除对 DMA 主总线的授权。

根据单次或突发配置的情况，每个 DMA 请求在 AHB 外设端口上相应地启动不同数量的传输。

- 当 AHB 外设端口被配置为单次传输时，根据 DMA_SxCR 寄存器 PSIZE[1:0] 位的值，每个 DMA 请求产生一次字节、半字或字的数据传输。
- 当 AHB 外设端口被配置为突发传输时，根据 DMA_SxCR 寄存器 PBURST[1:0] 和 PSIZE[1:0] 位的值，每个 DMA 请求相应地生成 4 个、8 个或 16 个节拍的字节、半字或字的传输。

对于需要配置 MBURST 和 MSIZE 位的 AHB 存储器端口，必须考虑与上述相同的内容。

在直接模式下，数据流只能生成单次传输，而 MBURST[1:0] 和 PBURST[1:0] 位由硬件强制配置。

必须选择地址指针（DMA_SxPAR 或 DMA_SxM0AR 寄存器），以确保一个突发块内的所有传输在等于传输大小的地址边界对齐。

选择突发配置必须要遵守 AHB 协议，即突发传输 *不得*越过 1 KB 地址边界，因为可以分配给单个从设备的最小地址空间是 1 KB。这意味着突发块传输不应越过 1 KB 地址边界，否则就会产生一个 AHB 错误，并且 DMA 寄存器不会报告这个错误。

15.3.14 FIFO

FIFO 结构

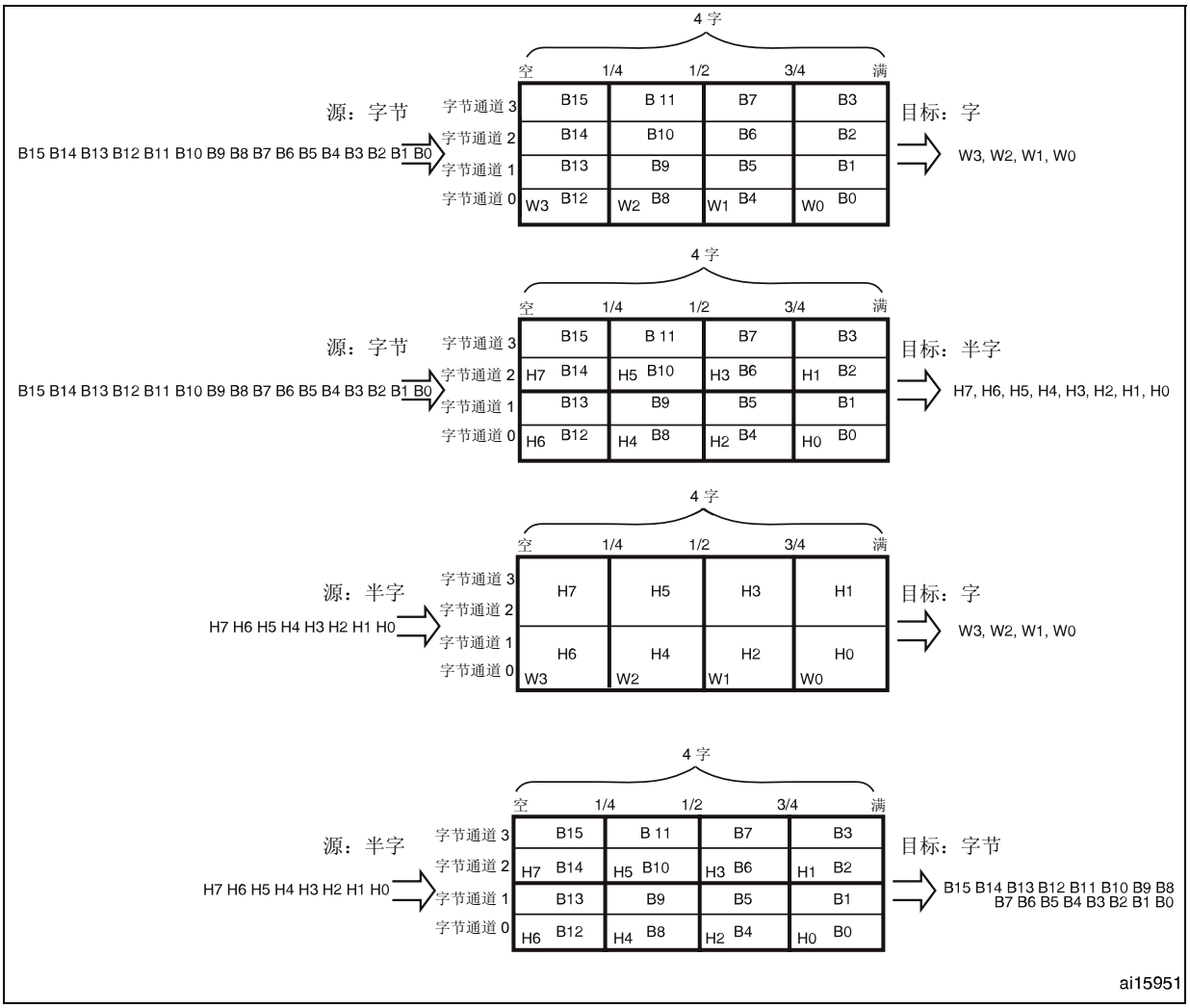
FIFO 用于在源数据传输到目标之前临时存储这些数据。

每个数据流都有一个独立的 4 字 FIFO，阈值级别可由软件配置为 1/4、1/2、3/4 或满。

为了使能 FIFO 阈值级别，必须通过将 DMA_SxFCR 寄存器中的 DMDIS 位置 1 来禁止直接模式。

FIFO 的结构随源与目标数据宽度而不同，相关内容在*图 74：FIFO 结构*中介绍。

图 74. FIFO 结构



FIFO 阈值与突发配置

选择 FIFO 阈值（DMA_SxFCR 寄存器的位 FTH[1:0]）和存储器突发大小（DMA_SxCR 寄存器的 MBURST[1:0] 位）时需要小心：FIFO 阈值指向的内容必须与整数个存储器突发传输完全匹配。如果不是这样，当使能数据流时将生成一个 FIFO 错误（DMA_HISR 或 DMA_LISR 寄存器的标志 FEIFx），然后将自动禁止数据流。允许的和禁止的配置在*表 101* 中介绍。禁止的配置在表中以灰色突出显示。

表 101. FIFO 阈值配置

MSIZE	FIFO 级别	MBURST = INCR4	MBURST = INCR8	MBURST = INCR16
字节	1/4	4 个节拍的 1 次突发	禁止	禁止
	1/2	4 个节拍的 2 次突发	8 个节拍的 1 次突发	
	3/4	4 个节拍的 3 次突发	禁止	
	满	4 个节拍的 4 次突发	8 个节拍的 2 次突发	16 个节拍的 1 次突发
半字	1/4	禁止	禁止	禁止
	1/2	4 个节拍的 1 次突发		
	3/4	禁止		
	满	4 个节拍的 2 次突发	8 个节拍的 1 次突发	
字	1/4	禁止	禁止	
	1/2			
	3/4			
	满	4 个节拍的 1 次突发		

所有这些情况下，突发大小与数据大小的乘积不得超过 FIFO 大小（数据大小可以为：1（字节）、2（半字）或 4（字））。

如果发生下列一种情况，会导致 DMA 传输结束时出现不完整的突发传输：

- 对于 AHB 外设端口配置：数据项总数（在 DMA_SxNDTR 寄存器中设置）不是突发大小与数据大小乘积的倍数。
- 对于 AHB 存储器端口配置：要传输到存储器的 FIFO 中的剩余数据项的数目不是突发大小与数据大小乘积的倍数。

在这些情况下，即使在 DMA 数据流配置期间请求突发事务，要传输的剩余数据也将由 DMA 在单独模式下管理。

注： *当在外设 AHB 端口上请求突发传输并且使用 FIFO（DMA_SxCR 寄存器中 DMDIS = 1）时，必须根据 DMA 数据流方向遵守下列规则来避免出现上溢或下溢情况：*

如果 (PBURST × PSIZE) = FIFO_SIZE（4 字），则当 PSIZE = 1、2 或 4，PBURST = 4、8 或 16 时，禁止 FIFO_Threshold = 3/4。

此规则将确保一次释放足够的 FIFO 空间来处理外设的请求。

FIFO 刷新

当复位 DMA_SxCR 寄存器中的 EN 位来禁止数据流，以及配置数据流来管理外设到存储器或存储器到存储器的传输时，可以刷新 FIFO。因为如果禁止数据流时仍有某些数据存留在 FIFO 中，DMA 控制器会将剩余的数据继续传输到目标（即使已经有效禁止了数据流）。刷新完成时，会将 DMA_LISR 或 DMA_HISR 寄存器中的传输完成状态位 (TCIFx) 置 1。

在这种情况下，剩余数据计数器 DMA_SxNDTR 保持的值指示在目标存储器现有多少可用数据项。

请注意在 FIFO 刷新操作期间，如果 FIFO 中要传输到存储器的剩余数据项的数目（以字节为单位）小于存储器数据宽度（例如在 MSIZE 配置为字时 FIFO 中为 2 个字节），则会使用在 DMA_SxCR 寄存器中的 MSIZE 位设置的数据宽度发送数据。这意味着将使用非预期值写入存储器。软件可以读取 DMA_SxNDTR 寄存器来确定包含良好数据的存储器区域（起始地址和最后地址）。

如果 FIFO 中剩余数据项的数目小于突发大小，并且数据流通过将 DMA_SxCR 寄存器 MBURST 位置 1 进行配置来管理在 AHB 存储器端口上的突发传输，则使用单次传输来完成 FIFO 的刷新。

直接模式

默认情况下，FIFO 以直接模式操作（将 DMA_SxFCR 中的 DMDIS 位置 1），不使用 FIFO 阈值级别。如果在每次 DMA 请求之后，系统需要至/自存储器的立即和单独传输，这种模式非常有用。

当在直接模式（禁止 FIFO）下将 DMA 配置为以存储器到外设模式传输数据时，DMA 会将一个数据从存储器预加载到内部 FIFO，从而确保一旦外设触发 DMA 请求时则立即传输数据。

为了避免 FIFO 饱和，建议使用高优先级配置相应的数据流。

该模式仅限以下方式的传输：

- 源和目标传输宽度相等，并均由 DMA_SxCR 中的 PSIZE[1:0] 位定义（MSIZE[1:0] 位的状态是“无关”）
- 不可能进行突发传输（DMA_SxCR 中的 PBURST[1:0] 和 MBURST[1:0] 位的状态是“无关”）

当实现存储器到存储器传输时不得使用直接模式。

15.3.15 DMA 传输完成

以下各种事件均可以结束传输过程，并将 DMA_LISR 或 DMA_HISR 状态寄存器中的 TCIFx 位置 1：

- 在 DMA 流控制器模式下：
  - 在存储器到外设模式下，DMA_SxNDTR 计数器已达到零
  - 传输结束前禁止了数据流（通过将 DMA_SxCR 寄存器中的 EN 位清零），并且（当传输是外设到存储器或存储器到存储器）所有的剩余数据均已从 FIFO 刷新到存储器

- 在外设流控制器模式下：
  - 已从外设生成最后的外部突发请求或单独请求，并当 DMA 在外设到存储器模式下工作时，剩余数据已从 FIFO 传输到存储器
  - 数据流由软件禁止，并当 DMA 在外设到存储器模式下工作时，剩余数据已从 FIFO 传输到存储器

注： *仅在外设到存储器模式下，传输的完成取决于 FIFO 中要传输到存储器的剩余数据。这种情况不适用于存储器到外设模式。*

如果是在非循环模式下配置数据流，传输结束后（即要传输的数据数目达到零），除非软件重新对数据流编程并重新使能数据流（通过将 DMA_SxCR 寄存器中的 EN 位置 1），否则 DMA 即会停止传输（通过硬件将 DMA_SxCR 寄存器中的 EN 位清零）并且不再响应任何 DMA 请求。

15.3.16 DMA 传输暂停

可以随时暂停 DMA 传输以供稍后重新开始；也可以在 DMA 传输结束前明确禁止传输。

分为两种情况：

- 数据流禁止传输，以后不从停止点重新开始。这种情况下，只需将 DMA_SxCR 寄存器中的 EN 位清零来禁止数据流，除此之外不需要任何其他操作。禁止数据流可能要花费一些时间（需要首先完成正在进行的传输）。需要将传输完成中断标志（DMA_LISR 或 DMA_HISR 寄存器中的 TCIF）置 1 来指示传输结束。当前 DMA_SxCR 中的 EN 位的值是“0”，借此确认数据流已经终止传输。DMA_SxNDTR 寄存器包含数据流停止时剩余数据项的数目，这样软件便可以确定数据流中断前已传输了多少数据项。
- 数据流在 DMA_SxNDTR 寄存器中要传输的剩余数据项数目达到 0 之前暂停传输。目的是以后通过重新使能数据流重新开始传输。为了在传输停止点重新开始传输，软件必须在通过写入 DMA_SxCR 寄存器中的 EN 位（然后检查确认该位为‘0’）禁止数据流之后，首先读取 DMA_SxNDTR 寄存器来了解已经收集的数据项的数目。然后：
  - 必须更新外设和/或存储器地址以调整地址指针
  - 必须使用要传输的剩余数据项的数目（禁止数据流时读取的值）更新 SxNDTR 寄存器
  - 然后可以重新使能数据流，从停止点重新开始传输

注： *请注意，传输完成中断标志（DMA_LISR 或 DMA_HISR 中的 TCIF）置 1 将指示因数据流中断而结束传输。*

15.3.17 流控制器

控制要传输的数据数目的实体称为流控制器。此流控制器使用 DMA_SxCR 寄存器中的 PFCTRL 位针对每个数据流独立配置。

流控制器可以是：

- DMA 控制器：在这种情况下，要传输的数据项的数目在使能 DMA 数据流之前由软件编程到 DMA_SxNDTR 寄存器。
- 外设源或目标：当要传输的数据项的数目未知时属于这种情况。当所传输的是最后的数据时，外设通过硬件向 DMA 控制器发出指示。仅限能够发出传输结束信号的外设支持此功能，也就是：

当外设流控制器用于给定数据流时，写入 DMA_SxNDTR 的值对 DMA 传输没有作用。实际上，不论写入什么值，一旦使能数据流，硬件即会将该值强制置为 0xFFFF 来执行下列方案：

- 预期的数据流中断：DMA_SxCR 寄存器中的 EN 位由软件重置为 0，以在外设发送最后的数据硬件信号（单独或突发）之前停止数据流。这样，在外设到存储器 DMA 传输的情况下，数据流即会关闭并触发 FIFO 刷新。状态寄存器中相应数据流的 TCIFx 标志置 1 以指示 DMA 完成传输。要了解 DMA 传输期间传输的数据项的数目，请读取 DMA_SxNDTR 寄存器并应用下列公式：
  - 传输的数据数目 = 0xFFFF – DMA_SxNDTR
- 因接收到最后的数据硬件信号而引起的正常数据流中断：当外设请求最后的传输（单独或突发）并当此传输完成时，自动中断数据流。相应流的 TCIFx 标志在状态寄存器中置 1 以指示 DMA 传输完成。要了解传输的数据项的数目，请读取 DMA_SxNDTR 寄存器并应用与上面相同的公式。
- DMA_SxNDTR 寄存器达到 0：状态寄存器中相应数据流的 TCIFx 标志置 1 以指示强制的 DMA 传输完成。即使尚未置位最后的数据硬件信号（单独或突发），也会自动关闭数据流。已传输的数据不会丢失。这意味着即使在外设流控制模式下，DMA 在单独的事务中最多处理 65535 个数据项。

注： *当在存储器到存储器模式下配置时，DMA 始终是流控制器，而 PFCTRL 位由硬件强制置为 0。*

在外设流控制器模式下禁止循环模式。

15.3.18 可能的 DMA 配置汇总

表 102 汇总了各种可能的 DMA 配置。禁止的配置在表中以灰色突出显示。

表 102. 可能的 DMA 配置

DMA 传输模式	源	目标	流控制器	循环模式	传输类型	直接模式	双缓冲区模式
外设到存储器	AHB 外设端口	AHB 存储器端口	DMA	允许	单独	允许	允许
					突发	禁止	
			外设	禁止	单独	允许	禁止
					突发	禁止	
存储器到外设	AHB 存储器端口	AHB 外设端口	DMA	允许	单独	允许	允许
					突发	禁止	
			外设	禁止	单独	允许	禁止
					突发	禁止	
存储器到存储器	AHB 外设端口	AHB 存储器端口	仅 DMA	禁止	单独	禁止	禁止
					突发		

15.3.19 流配置过程

配置 DMA 数据流 x（其中 x 是数据流编号）时应遵守下面的顺序：

1. 如果使能了数据流，通过重置 DMA_SxCR 寄存器中的 EN 位将其禁止，然后读取此位以确认没有正在进行的数据流操作。将此位写为 0 不会立即生效，因为实际上只有所有当前传输都已完成时才会将其写为 0。当所读取 EN 位的值为 0 时，才表示可以配置数据流。因此在开始任何数据流配置之前，需要等待 EN 位置 0。应将先前的数据块 DMA 传输中在状态寄存器（DMA_LISR 和 DMA_HISR）中置 1 的所有数据流专用的位置 0，然后才可重新使能数据流。
2. 在 DMA_SxPAR 寄存器中设置外设端口寄存器地址。外设事件发生后，数据会从此地址移动到外设端口或从外设端口移动到此地址。
3. 在 DMA_SxMA0R 寄存器（在双缓冲区模式的情况下还有 DMA_SxMA1R 寄存器）中设置存储器地址。外设事件发生后，将从此存储器读取数据或将数据写入此存储器。
4. 在 DMA_SxNDTR 寄存器中配置要传输的数据项的总数。每出现一次外设事件或每出现一个节拍的突发传输，该值都会递减。
5. 使用 DMAMux1 将 DMA 请求线接到 DMA 通道。
6. 如果外设用作流控制器而且支持此功能，请将 DMA_SxCR 寄存器中的 PFCTRL 位置 1。
7. 使用 DMA_SxCR 寄存器中的 PL[1:0] 位配置数据流优先级。
8. 配置 FIFO 的使用情况（使能或禁止，发送和接收阈值）
9. 配置数据传输方向、外设和存储器增量/固定模式、单独或突发事务、外设和存储器数据宽度、循环模式、双缓冲区模式和传输完成一半和/或全部完成，和/或 DMA_SxCR 寄存器中错误的中断。
10. 通过将 DMA_SxCR 寄存器中的 EN 位置 1 激活数据流。

一旦使能了流，即可响应连接到数据流的外设发出的任何 DMA 请求。

一旦在 AHB 目标端口上传输了一半数据，传输一半标志 (HTIF) 便会置 1，如果传输一半中断使能位 (HTIE) 置 1，还会生成中断。传输结束时，传输完成标志 (TCIF) 便会置 1，如果传输完成中断使能位 (TCIE) 置 1，还会生成中断。

警告： **要关闭连接到 DMA 数据流请求的外设，必须首先关闭外设连接的 DMA 数据流，然后等待 EN 位 = 0。只有这样才能安全地禁止外设。**

15.3.20 错误管理

DMA 控制器可以检测到以下错误：

- **传输错误**：当发生下列情况时，传输错误中断标志 (TEIFx) 将置 1：
  - DMA 读或写访问期间发生总线错误
  - 软件请求在双缓冲区模式下写访问存储器地址寄存器，但是，已使能数据流，并且当前目标存储器是受写入存储器地址寄存器操作影响的存储器（请参见*第 15.3.11 节：双缓冲区模式*）
- **FIFO 错误**：如果发生下列情况，FIFO 错误中断标志 (FEIFx) 将置 1：
  - 检测到 FIFO 下溢情况
  - 检测到 FIFO 上溢情况（在存储器到存储器模式下由 DMA 内部管理请求和传输，所以在此模式下不检测溢出情况）
  - 当 FIFO 阈值级别与存储器突发大小不兼容时使能流（请参见*表 101：FIFO 阈值配置*）
- **直接模式错误**：只有当在直接模式下工作并且已将 DMA_SxCR 寄存器中的 MINC 位清零时，才能在外设到存储器模式下将直接模式错误中断标志 (DMEIFx) 置 1。当在先前数据未完全传输到存储器（因为存储器总线未得到授权）的情况下发生 DMA 请求时，该标志将置 1。在这种情况下，该标志指示有两个数据项相继传输到相同的目标地址，如果目标不能管理这种情况，会发生问题。

在直接模式下，如果出现下列条件，FIFO 错误标志也会置 1：

- 在外设到存储器模式下，如果未对存储器总线授权支持多个外设请求，FIFO 可能饱和（上溢）
- 在存储器到外设模式下，如果在外设请求发生前存储器总线未得到授权，则可能发生下溢的情况

如果由于突发大小与 FIFO 阈值级别之间的不兼容而引起 TEIFx 或 FEIFx 标志置 1，则硬件自动将相应数据流配置寄存器 (DMA_SxCR) 中的 EN 位清零，从而禁止错误的数据流。

如果由于上溢或下溢情况引起 DMEIFx 或 FEIFx 标志置 1，则不会自动禁止错误的数据流，而是由软件决定是否通过重置 DMA_SxCR 寄存器中的 EN 位禁止数据流。这是因为当发生这种错误时没有数据丢失。

当 DMA_LISR 或 DMA_HISR 寄存器中数据流的错误中断标志（TEIF、FEIF、DMEIF）置 1 时，如果 DMA_SxCR 或 DMA_SxFCR 寄存器中相应的中断使能位（TEIE、FEIE、DMIE）也置 1，则会生成一个中断。

注： *当 FIFO 上溢或下溢的情况发生时，因为直到上溢或下溢的情况被清除，数据流才会确认外设请求，所以数据不会丢失。如果此确认过程花费过多时间，则外设本身会检测到其内部缓冲器上溢或下溢的情况，数据可能丢失。*

15.4 DMA 中断

对于每个 DMA 数据流，可在发生以下事件时产生中断：

- 达到半传输
- 传输完成
- 传输错误
- FIFO 错误（上溢、下溢或 FIFO 级别错误）
- 直接模式错误

可以使用单独的中断使能位以实现灵活性，如*表 103* 所示。

表 103. DMA 中断请求

中断事件	事件标志	使能控制位
半传输	HTIF	HTIE
传输完成	TCIF	TCIE
传输错误	TEIF	TEIE
FIFO 上溢/下溢	FEIF	FEIE
直接模式错误	DMEIF	DMEIE

注： *在将使能控制位置‘1’前，应将相应的事件标志清零，否则会立即产生中断。*

15.5 DMA 寄存器

DMA 寄存器必须按字（32 位）进行访问。

15.5.1 DMA 低中断状态寄存器 (DMA_LISR)

DMA low interrupt status register

偏移地址：0x00

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	TCIF3	HTIF3	TEIF3	DMEIF3	Res.	FEIF3	TCIF2	HTIF2	TEIF2	DMEIF2	Res.	FEIF2
r	r	r	r	r	r	r	r		r	r	r	r	r		r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	TCIF1	HTIF1	TEIF1	DMEIF1	Res.	FEIF1	TCIF0	HTIF0	TEIF0	DMEIF0	Res.	FEIF0
r	r	r	r	r	r	r	r		r	r	r	r	r		r

位 31:28、15:12 保留，必须保持复位值。

位 27、21、11、5 **TCIFx**：数据流 x 传输完成中断标志 (Stream x transfer complete interrupt flag) (x = 3..0)
此位将由硬件置 1，由软件清零，软件只需将 1 写入 DMA_LIFCR 寄存器的相应位。
0：数据流 x 上无传输完成事件
1：数据流 x 上发生传输完成事件

位 26、20、10、4 **HTIFx**：数据流 x 半传输中断标志 (Stream x half transfer interrupt flag) (x=3..0)
此位将由硬件置 1，由软件清零，软件只需将 1 写入 DMA_LIFCR 寄存器的相应位。
0：数据流 x 上无半传输事件
1：数据流 x 上发生半传输事件

位 25、19、9、3 **TEIFx**：数据流 x 传输错误中断标志 (Stream x transfer error interrupt flag) (x=3..0)
此位将由硬件置 1，由软件清零，软件只需将 1 写入 DMA_LIFCR 寄存器的相应位。
0：数据流 x 上无传输错误
1：数据流 x 上发生传输错误

位 24、18、8、2 **DMEIFx**：数据流 x 直接模式错误中断标志 (Stream x direct mode error interrupt flag) (x=3..0)
此位将由硬件置 1，由软件清零，软件只需将 1 写入 DMA_LIFCR 寄存器的相应位。
0：数据流 x 上无直接模式错误
1：数据流 x 上发生直接模式错误

位 23、17、7、1 保留，必须保持复位值。

位 22、16、6、0 **FEIFx**：数据流 x FIFO 错误中断标志 (Stream x FIFO error interrupt flag) (x=3..0)
此位将由硬件置 1，由软件清零，软件只需将 1 写入 DMA_LIFCR 寄存器的相应位。
0：数据流 x 上无 FIFO 错误事件
1：数据流 x 上发生 FIFO 错误事件

15.5.2 DMA 高中断状态寄存器 (DMA_HISR)

DMA high interrupt status register

偏移地址：0x04

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	TCIF7	HTIF7	TEIF7	DMEIF7	Res.	FEIF7	TCIF6	HTIF6	TEIF6	DMEIF6	Res.	FEIF6
				r	r	r	r		r	r	r	r	r		r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	TCIF5	HTIF5	TEIF5	DMEIF5	Res.	FEIF5	TCIF4	HTIF4	TEIF4	DMEIF4	Res.	FEIF4
				r	r	r	r		r	r	r	r	r		r

位 31:28、15:12 保留，必须保持复位值。

位 27、21、11、5 **TCIFx**：数据流 x 传输完成中断标志 (Stream x transfer complete interrupt flag) (x=7..4)
此位将由硬件置 1，由软件清零，软件只需将 1 写入 DMA_HIFCR 寄存器的相应位。
0：数据流 x 上无传输完成事件
1：数据流 x 上发生传输完成事件

位 26、20、10、4 **HTIFx**：数据流 x 半传输中断标志 (Stream x half transfer interrupt flag) (x=7..4)
此位将由硬件置 1，由软件清零，软件只需将 1 写入 DMA_HIFCR 寄存器的相应位。
0：数据流 x 上无半传输事件
1：数据流 x 上发生半传输事件

位 25、19、9、3 **TEIFx**：数据流 x 传输错误中断标志 (Stream x transfer error interrupt flag) (x=7..4)
此位将由硬件置 1，由软件清零，软件只需将 1 写入 DMA_HIFCR 寄存器的相应位。
0：数据流 x 上无传输错误
1：数据流 x 上发生传输错误

位 24、18、8、2 **DMEIFx**：数据流 x 直接模式错误中断标志 (Stream x direct mode error interrupt flag) (x=7..4)
此位将由硬件置 1，由软件清零，软件只需将 1 写入 DMA_HIFCR 寄存器的相应位。
0：数据流 x 上无直接模式错误
1：数据流 x 上发生直接模式错误

位 23、17、7、1 保留，必须保持复位值。

位 22、16、6、0 **FEIFx**：数据流 x FIFO 错误中断标志 (Stream x FIFO error interrupt flag) (x=7..4)
此位将由硬件置 1，由软件清零，软件只需将 1 写入 DMA_HIFCR 寄存器的相应位。
0：数据流 x 上无 FIFO 错误事件
1：数据流 x 上发生 FIFO 错误事件

15.5.3 DMA 低中断标志清零寄存器 (DMA_LIFCR)

DMA low interrupt flag clear register

偏移地址：0x08

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	CTCIF3	CHTIF3	CTEIF3	CDMEIF3	Res.	CFEIF3	CTCIF2	CHTIF2	CTEIF2	CDMEIF2	Res.	CFEIF2
				w	w	w	w		w	w	w	w	w		w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	CTCIF1	CHTIF1	CTEIF1	CDMEIF1	Res.	CFEIF1	CTCIF0	CHTIF0	CTEIF0	CDMEIF0	Res.	CFEIF0
				w	w	w	w		w	w	w	w	w		w

位 31:28、15:12 保留，必须保持复位值。

位 27、21、11、5 **CTCIFx**：数据流 x 传输完成中断标志清零 (Stream x clear transfer complete interrupt flag) (x = 3..0)

将 1 写入此位时，DMA_LISR 寄存器中相应的 TCIFx 标志将清零

位 26、20、10、4 **CHTIFx**：数据流 x 半传输中断标志清零 (Stream x clear half transfer interrupt flag) (x = 3..0)

将 1 写入此位时，DMA_LISR 寄存器中相应的 HTIFx 标志将清零

位 25、19、9、3 **CTEIFx**：数据流 x 传输错误中断标志清零 (Stream x clear transfer error interrupt flag) (x = 3..0)

将 1 写入此位时，DMA_LISR 寄存器中相应的 TEIFx 标志将清零

位 24、18、8、2 **CDMEIFx**：数据流 x 直接模式错误中断标志清零 (Stream x clear direct mode error interrupt flag) (x = 3..0)

将 1 写入此位时，DMA_LISR 寄存器中相应的 DMEIFx 标志将清零

位 23、17、7、1 保留，必须保持复位值。

位 22、16、6、0 **CFEIFx**：数据流 x FIFO 错误中断标志清零 (Stream x clear FIFO error interrupt flag) (x = 3..0)

将 1 写入此位时，DMA_LISR 寄存器中相应的 CFEIFx 标志将清零

15.5.4 DMA 高中断标志清零寄存器 (DMA_HIFCR)

DMA high interrupt flag clear register

偏移地址：0x0C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	CTCIF7	CHTIF7	CTEIF7	CDMEIF7	Res.	CFEIF7	CTCIF6	CHTIF6	CTEIF6	CDMEIF6	Res.	CFEIF6
				w	w	w	w		w	w	w	w	w		w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	CTCIF5	CHTIF5	CTEIF5	CDMEIF5	Res.	CFEIF5	CTCIF4	CHTIF4	CTEIF4	CDMEIF4	Res.	CFEIF4
				w	w	w	w		w	w	w	w	w		w

位 31:28、15:12 保留，必须保持复位值。

位 27、21、11、5 **CTCIFx**：数据流 x 传输完成中断标志清零 (Stream x clear transfer complete interrupt flag) (x = 7..4)

将 1 写入此位时，DMA_HISR 寄存器中相应的 TCIFx 标志将清零

位 26、20、10、4 **CHTIFx**：数据流 x 半传输中断标志清零 (Stream x clear half transfer interrupt flag) (x = 7..4)

将 1 写入此位时，DMA_HISR 寄存器中相应的 HTIFx 标志将清零

位 25、19、9、3 **CTEIFx**：数据流 x 传输错误中断标志清零 (Stream x clear transfer error interrupt flag) (x = 7..4)

将 1 写入此位时，DMA_HISR 寄存器中相应的 TEIFx 标志将清零

位 24、18、8、2 **CDMEIFx**：数据流 x 直接模式错误中断标志清零 (Stream x clear direct mode error interrupt flag) (x = 7..4)

将 1 写入此位时，DMA_HISR 寄存器中相应的 DMEIFx 标志将清零

位 23、17、7、1 保留，必须保持复位值。

位 22、16、6、0 **CFEIFx**：数据流 x FIFO 错误中断标志清零 (Stream x clear FIFO error interrupt flag) (x = 7..4)

将 1 写入此位时，DMA_HISR 寄存器中相应的 CFEIFx 标志将清零

15.5.5 DMA 数据流 x 配置寄存器 (DMA_SxCR) (x = 0..7)

DMA stream x configuration register

此寄存器用于配置相关数据流。

偏移地址：0x10 + 0x18 × *数据流编号*

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	MBURST [1:0]		PBURST[1:0]		Res.	CT	DBM	PL[1:0]	
							rw	rw	rw	rw		rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PINCOS	MSIZE[1:0]		PSIZE[1:0]		MINC	PINC	CIRC	DIR[1:0]		PFCTRL	TCIE	HTIE	TEIE	DMEIE	EN
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:25 保留，必须保持复位值。

位 24:23 **MBURST**：存储器突发传输配置 (Memory burst transfer configuration)

这些位将由软件置 1 和清零。
00：单次传输
01：INCR4（4 个节拍的增量突发传输）
10：INCR8（8 个节拍的增量突发传输）
11：INCR16（16 个节拍的增量突发传输）
这些位受到保护，只有 EN 为“0”时才可以写入
在直接模式中，当位 EN =“1”时，这些位由硬件强制置为 0x0。

位 22:21 **PBURST[1:0]**：外设突发传输配置 (Peripheral burst transfer configuration)

这些位将由软件置 1 和清零。
00：单次传输
01：INCR4（4 个节拍的增量突发传输）
10：INCR8（8 个节拍的增量突发传输）
11：INCR16（16 个节拍的增量突发传输）
这些位受到保护，只有 EN 为“0”时才可以写入
在直接模式下，这些位由硬件强制置为 0x0。

位 20 保留，必须保持复位值。

位 19 **CT**：当前目标（仅在双缓冲区模式下）(Current target (only in double buffer mode))
此位由硬件置 1 和清零，也可由软件写入。
0：当前目标存储器为存储器 0（使用 DMA_SxM0AR 指针寻址）
1：当前目标存储器为存储器 1（使用 DMA_SxM1AR 指针寻址）
只有 EN 为“0”时，此位才可以写入，以指示第一次传输的目标存储区。在使能数据流后，此位相当于一个状态标志，用于指示作为当前目标的存储区。

位 18 **DBM**：双缓冲区模式 (Double buffer mode)
此位由软件置 1 和清零。
0：传输结束时不切换缓冲区
1：DMA 传输结束时切换目标存储区
此位受到保护，只有 EN 为“0”时才可以写入。

位 17:16 **PL[1:0]**：优先级 (Priority level)
这些位将由软件置 1 和清零。
00：低
01：中
10：高
11：非常高
这些位受到保护，只有 EN 为“0”时才可以写入。

位 15 **PINCOS**：外设增量偏移量 (Peripheral increment offset size)
此位由软件置 1 和清零
0：用于计算外设地址的偏移量与 PSIZE 相关
1：用于计算外设地址的偏移量固定为 4（32 位对齐）。
如果位 PINC =“0”，则此位没有意义。
此位受到保护，只有 EN 为“0”时才可以写入。
如果选择直接模式或者 PBURST 不等于“00”，则当使能数据流（位 EN =“1”）时，此位由硬件强制置为低电平。

位 14:13 **MSIZE[1:0]**：存储器数据大小 (Memory data size)
这些位将由软件置 1 和清零。
00：字节（8 位）
01：半字（16 位）
10：字（32 位）
11：保留
这些位受到保护，只有 EN 为“0”时才可以写入。
在直接模式下，当位 EN =“1”时，MSIZE 位由硬件强制置为与 PSIZE 相同的值。

位 12:11 **PSIZE[1:0]**：外设数据大小 (Peripheral data size)
这些位将由软件置 1 和清零。
00：字节（8 位）
01：半字（16 位）
10：字（32 位）
11：保留
这些位受到保护，只有 EN 为“0”时才可以写入

位 10 **MINC**：存储器递增模式 (Memory increment mode)
此位由软件置 1 和清零。
0：存储器地址指针固定
1：每次数据传输后，存储器地址指针递增（增量为 MSIZE 值）
此位受到保护，只有 EN 为“0”时才可以写入。

位 9 **PINC**：外设递增模式 (Peripheral increment mode)
此位由软件置 1 和清零。
0：外设地址指针固定
1：每次数据传输后，外设地址指针递增（增量为 PSIZE 值）
此位受到保护，只有 EN 为“0”时才可以写入。

位 8 **CIRC**：循环模式 (Circular mode)
此位由软件置 1 和清零，并可由硬件清零。
0：禁止循环模式
1：使能循环模式
如果外设为流控制器（位 PFCTRL=1）且使能数据流（位 EN=1），此位由硬件自动强制清零。
如果 DBM 位置 1，当使能数据流（位 EN =“1”）时，此位由硬件自动强制置 1。

位 7:6 **DIR[1:0]**：数据传输方向 (Data transfer direction)
这些位将由软件置 1 和清零。
00：外设到存储器
01：存储器到外设
10：存储器到存储器
11：保留
这些位受到保护，只有 EN 为“0”时才可以写入。

位 5 **PFCTRL**：外设流控制器 (Peripheral flow controller)
此位由软件置 1 和清零。
0：DMA 是流控制器
1：外设是流控制器
此位受到保护，只有 EN 为“0”时才可以写入。
选择存储器到存储器模式（位 DIR[1:0]=10）后，此位由硬件自动强制清零。

位 4 **TCIE**：传输完成中断使能 (Transfer complete interrupt enable)
此位由软件置 1 和清零。
0：禁止 TC 中断
1：使能 TC 中断

位 3 **HTIE**：半传输中断使能 (Half transfer interrupt enable)
此位由软件置 1 和清零。
0：禁止 HT 中断
1：使能 HT 中断

位 2 **TEIE**：传输错误中断使能 (Transfer error interrupt enable)
此位由软件置 1 和清零。
0：禁止 TE 中断
1：使能 TE 中断

位 1 **DMEIE**：直接模式错误中断使能 (Direct mode error interrupt enable)
此位由软件置 1 和清零。
0：禁止 DME 中断
1：使能 DME 中断

位 0 **EN**：数据流使能/读作低电平时数据流就绪标志 (Stream enable / flag stream ready when read low)

此位由软件置 1 和清零。

0：禁止数据流

1：使能数据流

以下情况下，此位可由硬件清零：

– DMA 传输结束时（准备好配置数据流）

– AHB 主总线出现传输错误时

– 存储器 AHB 端口上的 FIFO 阈值与突发大小不兼容时

此位读作 0 时，软件可以对配置和 FIFO 位寄存器编程。EN 位读作 1 时，禁止向这些寄存器执行写操作。

注： 将 EN 位置“1”以启动新传输之前，DMA_LISR 或 DMA_HISR 寄存器中与数据流相对应的事件标志必须清零。

15.5.6 DMA 数据流 x 数据项数寄存器 (DMA_SxNDTR) (x = 0..7)

DMA stream x number of data register

偏移地址：0x14 + 0x18 × *数据流编号*

复位值：0x0000 0000

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

位 31:16 保留，必须保持复位值。

位 15:0 **NDT[15:0]**：要传输的数据项数目 (Number of data items to transfer)

要传输的数据项数目（0 到 65535）。只有在禁止数据流时，才能向此寄存器执行写操作。使能数据流后，此寄存器为只读，用于指示要传输的剩余数据项数。每次 DMA 传输后，此寄存器将递减。

传输完成后，此寄存器保持为零（数据流处于正常模式时），或者在以下情况下自动以先前编程的值重载：

– 以循环模式配置数据流时

– 通过将 EN 位置“1”来重新使能数据流时

如果该寄存器的值为零，则即使使能数据流，也无法完成任何事务。

15.5.7 DMA 数据流 x 外设地址寄存器 (DMA_SxPAR) (x = 0..7)

DMA stream x peripheral address register

偏移地址：0x18 + 0x18 × *数据流编号*

复位值：0x0000 0000

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

位 31:0 **PAR[31:0]**：外设地址 (Peripheral address)

读/写数据的外设数据寄存器的基址。

这些位受到写保护，只有 DMA_SxCR 寄存器中的 EN 为“0”时才可以写入。

15.5.8 DMA 数据流 x 存储器 0 地址寄存器 (DMA_SxM0AR) (x = 0..7)

DMA stream x memory 0 address register

偏移地址：0x1C + 0x18 × *数据流编号*

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
M0A[31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
M0A[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:0 **M0A[31:0]**：存储器 0 地址 (Memory 0 address)

读/写数据的存储区 0 的基址。

这些位受到写保护，只有在以下情况下才可以写入：

- 禁止数据流（DMA_SxCR 寄存器中的位 EN=“0”）或
- 使能数据流（DMA_SxCR 寄存器中的 EN=“1”）并且 DMA_SxCR 寄存器中的位 CT =“1”（在双缓冲区模式下）。

15.5.9 DMA 数据流 x 存储器 1 地址寄存器 (DMA_SxM1AR) (x = 0..7)

DMA stream x memory 1 address register

偏移地址：0x20 + 0x18 × *数据流编号*

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
M1A[31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
M1A[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:0 **M1A[31:0]**：存储器 1 地址（用于双缓冲区模式）(Memory 1 address (used in case of Double buffer mode))

读/写数据的存储区 1 的基址。

此寄存器仅用于双缓冲区模式。

这些位受到写保护，只有在以下情况下才可以写入：

- 禁止数据流（DMA_SxCR 寄存器中的位 EN=“0”）或
- 使能数据流（DMA_SxCR 寄存器中的 EN=“1”）并且 DMA_SxCR 寄存器中的位 CT =“0”。

15.5.10 DMA 数据流 x FIFO 控制寄存器 (DMA_SxFCR) (x = 0..7)

DMA stream x FIFO control register

偏移地址：0x24 + 0x24 × *数据流编号*

复位值：0x0000 0021

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

位 31:8 保留，必须保持复位值。

位 7 **FEIE**：FIFO 错误中断使能 (FIFO error interrupt enable)

此位由软件置 1 和清零。

0：禁止 FE 中断

1：使能 FE 中断

位 6 保留，必须保持复位值。

位 5:3 **FS[2:0]**：FIFO 状态 (FIFO status)

这些位为只读。

000：0 < fifo_level < 1/4

001：1/4 ≤ fifo_level < 1/2

010：1/2 ≤ fifo_level < 3/4

011：3/4 ≤ fifo_level < 满

100：FIFO 为空

101：FIFO 已满

其它：无意义

在直接模式（DMDIS 位为零）下，这些位无意义。

位 2 **DMDIS**：直接模式禁止 (Direct mode disable)

此位由软件置 1 和清零。它可由硬件置 1。

0：使能直接模式

1：禁止直接模式

此位受到保护，只有 EN 为“0”时才可以写入。

如果选择存储器到存储器模式（DMA_SxCR 中的 DIR 位为“10”），并且 DMA_SxCR 寄存器中的 EN 位为“1”，则此位由硬件置 1，因为在存储器到存储器配置不能使用直接模式。

位 1:0 **FTH[1:0]**：FIFO 阈值选择 (FIFO threshold selection)

这些位将由软件置 1 和清零。

00：FIFO 容量的 1/4

01：FIFO 容量的 1/2

10：FIFO 容量的 3/4

11：FIFO 完整容量

在直接模式（DMDIS 值为零）下，不使用这些位。

这些位受到保护，只有 EN 为“0”时才可以写入。

15.5.11 DMA 寄存器映射

表 104 汇总了 DMA 寄存器。

表 104. DMA 寄存器映射和复位值

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x0000	**DMA_LISR**	Res.	Res.	Res.	Res.	TCIF3	HTIF3	TEIF3	DMEIF3	Res.	FEIF3	TCIF2	HTIF2	TEIF2	DMEIF2	Res.	FEIF2	Res.	Res.	Res.	Res.	TCIF1	HTIF1	TEIF1	DMEIF1	Res.	FEIF1	TCIF0	HTIF0	TEIF0	DMEIF0	Reserved	FEIF0
	Reset value					0	0	0	0		0	0	0	0	0		0					0	0	0	0		0	0	0	0	0		0
0x0004	**DMA_HISR**	Res	Res	Res	Res	TCIF7	HTIF7	TEIF7	DMEIF7	Res	FEIF7	TCIF6	HTIF6	TEIF6	DMEIF6	Res	FEIF6	Res	Res	Res	Res	TCIF5	HTIF5	TEIF5	DMEIF5	Res	FEIF5	TCIF4	HTIF4	TEIF4	DMEIF4	Reserved	FEIF4
	Reset value					0	0	0	0		0	0	0	0	0		0					0	0	0	0		0	0	0	0	0		0
0x0008	**DMA_LIFCR**	Res	Res	Res	Res	CTCIF3	CHTIF3	TEIF3	CDMEIF3	Reserved	CFEIF3	CTCIF2	CHTIF2	CTEIF2	CDMEIF2	Res	CFEIF2	Res	Res	Res	Res	CTCIF1	CHTIF1	CTEIF1	CDMEIF1	Res	CFEIF1	CTCIF0	CHTIF0	CTEIF0	CDMEIF0	Reserved	CFEIF0
	Reset value					0	0	0	0		0	0	0	0	0		0					0	0	0	0		0	0	0	0	0		0
0x000C	**DMA_HIFCR**	Res	Res	Res	Res	CTCIF7	CHTIF7	CTEIF7	CDMEIF7	Reserved	CFEIF7	CTCIF6	CHTIF6	CTEIF6	CDMEIF6	Res	CFEIF6	Res	Res	Res	Res	CTCIF5	CHTIF5	CTEIF5	CDMEIF5	Res	CFEIF5	CTCIF4	CHTIF4	CTEIF4	CDMEIF4	Reserved	CFEIF4
	Reset value	Res	Res	Res	Res	0	0	0	0		0	0	0	0	0	Res	0					0	0	0	0		0	0	0	0	0		0
0x0010	**DMA_S0CR**	Res	Res	Res	Res	Res	Res	Res	MBURST[1:0]		PBURST[1:0]		Res	CT	DBM	PL[1:0]		PINCOS	MSIZE[1:0]		PSIZE[1:0]		MINC	PINC	CIRC	DIR[1:0]		PFCTRL	TCIE	HTIE	TEIE	DMEIE	EN
	Reset value								0	0	0	0		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0014	**DMA_S0NDTR**	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	NDT[15:.]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0018	**DMA_S0PAR**	PA[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x001C	**DMA_S0M0AR**	M0A[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0020	**DMA_S0M1AR**	M1A[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0024	**DMA_S0FCR**	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	FEIE	Res	FS[2:0]			DMDIS	FTH [1:0]	
	Reset value																									0		1	0	0	0	0	1
0x002C	**DMA_S1NDTR**	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	NDT[15:.]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0030	**DMA_S1PAR**	PA[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0034	**DMA_S1M0AR**	M0A[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

表 104. DMA 寄存器映射和复位值（续）

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x0038	**DMA_S1M1AR**	M1A[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x003C	**DMA_S1FCR**	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	FEIE	Res	FS[2:0]			DMDIS	FTH [1:0]	
	Reset value																									0		1	0	0	0	0	1
0x0044	**DMA_S2NDTR**	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	NDT[15:.]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0048	**DMA_S2PAR**	PA[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x004C	**DMA_S2M0AR**	M0A[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0050	**DMA_S2M1AR**	M1A[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0054	**DMA_S2FCR**	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	FEIE	Res	FS[2:0]			DMDIS	FTH [1:0]	
	Reset value																									0		1	0	0	0	0	1
0x005C	**DMA_S3NDTR**	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	NDT[15:.]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0060	**DMA_S3PAR**	PA[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0064	**DMA_S3M0AR**	M0A[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0068	**DMA_S3M1AR**	M1A[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x006C	**DMA_S3FCR**	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	FEIE	Res	FS[2:0]			DMDIS	FTH [1:0]	
	Reset value																									0		1	0	0	0	0	1
0x0074	**DMA_S4NDTR**	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	NDT[15:.]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0078	**DMA_S4PAR**	PA[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x007C	**DMA_S4M0AR**	M0A[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0080	**DMA_S4M1AR**	M1A[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0084	**DMA_S4FCR**	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	FEIE	Res	FS[2:0]			DMDIS	FTH [1:0]	
	Reset value																									0		1	0	0	0	0	1

表 104. DMA 寄存器映射和复位值（续）

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x008C	**DMA_S5NDTR**	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	NDT[15:.]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0090	**DMA_S5PAR**	PA[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0094	**DMA_S5M0AR**	M0A[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0098	**DMA_S5M1AR**	M1A[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x009C	**DMA_S5FCR**	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	FEIE	Res	FS[2:0]			DMDIS	FTH [1:0]	
	Reset value																									0		1	0	0	0	0	1
0x00A4	**DMA_S6NDTR**	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	NDT[15:.]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x00A8	**DMA_S6PAR**	PA[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x00AC	**DMA_S6M0AR**	M0A[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x00B0	**DMA_S6M1AR**	M1A[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x00B4	**DMA_S6FCR**	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	FEIE	Res	FS[2:0]			DMDIS	FTH [1:0]	
	Reset value																									0		1	0	0	0	0	1
0x00BC	**DMA_S7NDTR**	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	NDT[15:.]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x00C0	**DMA_S7PAR**	PA[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x00C4	**DMA_S7M0AR**	M0A[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x00C8	**DMA_S7M1AR**	M1A[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x00CC	**DMA_S7FCR**	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	FEIE	Res	FS[2:0]			DMDIS	FTH [1:0]	
	Reset value																									0		1	0	0	0	0	1

有关寄存器边界地址的信息，请参见*第 100 页的第 2.2.2 节*。

16 基本直接存储器访问控制器 (BDMA)

16.1 前言

基本直接存储器访问 (BDMA) 控制器用于在外设与存储器之间以及存储器与存储器之间提供高速数据传输。可以在无需任何 CPU 操作的情况下通过 DMA 快速移动数据。这样节省的 CPU 资源可供其它操作使用。

BDMA 控制器总共有 8 个通道，每个通道都专门管理来自一个或多个外设的存储器访问请求。每个通道都有一个仲裁器，用于处理 DMA 请求间的优先级。

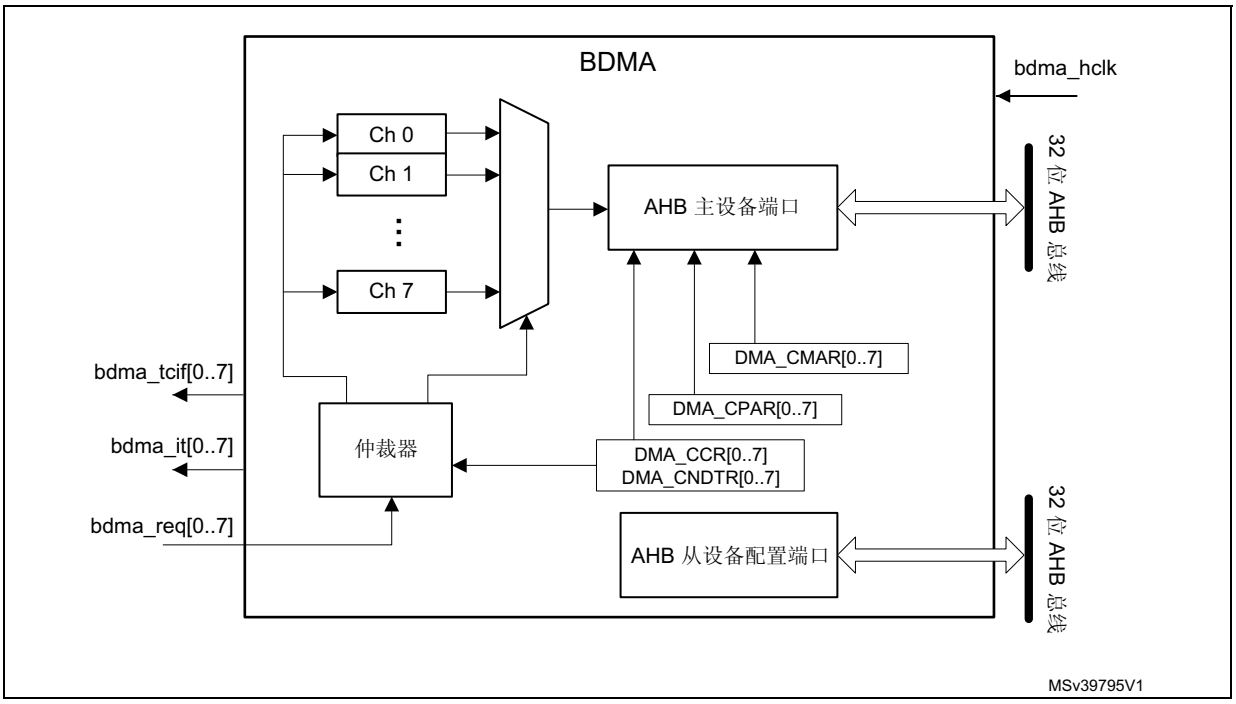
16.2 BDMA 主要特性

- 8 个可单独配置的通道（请求）
- 每个通道都与专用的硬件 DMA 请求相连，同时还支持软件触发。此配置由软件执行
- BDMA 控制器的通道请求的优先级可用软件编程（4 个级别：非常高、高、中和低），在软件优先级相同的情况下可以通过硬件决定优先级（例如，请求 1 的优先级高于请求 2）
- 独立的源和目标传输大小（字节、半字、字），模拟打包和解包。源/目标地址必须与数据大小匹配
- 支持循环缓冲区管理
- 3 个事件标志（DMA 半传输、DMA 传输完成和 DMA 传输错误），在单个中断请求中针对每个通道一起进行逻辑或运算
- 存储器到存储器的传输
- 外设到存储器、存储器到外设以及外设到外设的传输
- 访问作为源和目标的 D3 域存储器和外设
- 可编程的待传输数据数目：最大 65535

16.3 BDMA 功能说明

框图如下图所示。

图 75. BDMA框图



BDMA 控制器通过与其他系统主设备共用系统总线来执行直接存储器传输。当 DMA 与 CPU 访问目标（存储器或外设）相同时，DMA 请求会将 CPU 对系统总线的访问停止数个总线周期。总线矩阵执行循环调度，从而确保 CPU 至少可使用一半的系统总线带宽（无论对于存储器还是外设）。

16.3.1 BDMA 传输

在产生事件后，外设会向 BDMA 控制器发送请求信号。BDMA 控制器根据通道优先级处理该请求。只要 BDMA 控制器访问外设，BDMA 控制器就会向外设发送确认信号。外设获得 BDMA 控制器的确认信号后，便会立即释放其请求。一旦外设使请求失效，DMA 控制器就会释放确认信号。如果有更多请求，外设可以启动下一个事务。

概括起来，每个 BDMA 传输都包含三项操作：

- 从外设数据寄存器或通过内部当前外设/存储器地址寄存器寻址的存储器单元中加载数据。用于首次传输的起始地址是在 BDMA_CPARx 或 BDMA_CMARx 寄存器中编程的外设/存储器基址。
- 将加载的数据存储到外设数据寄存器或通过内部当前外设/存储器地址寄存器寻址的存储器单元。用于首次传输的起始地址是在 BDMA_CPARx 或 BDMA_CMARx 寄存器中编程的外设/存储器基址。
- BDMA_CNDTRx 寄存器在传输后递减，该寄存器中包含仍需执行的传输数量。

16.3.2 仲裁器

仲裁器根据通道请求的优先级管理通道请求，并启动外设/存储器访问序列。

优先级管理分为两个阶段：

- 软件：每个通道优先级都可以在 BDMA_CCRx 寄存器中配置。分为四个级别：
  - 非常高优先级
  - 高优先级
  - 中优先级
  - 低优先级
- 硬件：如果两个请求具有相同的软件优先级，则编号低的通道优先于编号高的通道。例如，通道 2 的优先级高于通道 4。

16.3.3 BDMA通道

各个通道均能处理外设寄存器（位于固定地址中）与存储器单元之间的 DMA 传输。待传输数据的数目可编程（最大 65535）。每个事务完成后，包含要传输的数据项总数的寄存器都会递减。

数据大小可编程

外设和存储器的传输数据大小可通过 BDMA_CCRx 寄存器的 PSIZE 位和 MSIZE 位完全进行编程。

指针递增

根据 BDMA_CCRx 寄存器的 PINC 位和 MINC 位的状态，可以选择使外设和存储器指针在每次传输后自动向后递增。如果使能了递增模式，则根据所选的数据大小，下次传输的地址即为前一次传输的地址加上 1、2 或 4。首次传输的地址可在 BDMA_CPARx/BDMA_CMARx 寄存器中进行编程。在传输过程中，这些寄存器将保持初始编程的值。软件无法获得当前传输的地址（位于当前内部外设或存储器地址寄存器中）。

如果将通道配置为非循环模式，则在最后一次传输完成后（即待传输的数据项数目达到零后），将不处理任何 DMA 请求。若需要将待传输数据项的新数目重新加载至 BDMA_CNDTRx 寄存器中，则必须禁止 DMA 通道。

注： *如果禁止 DMA 通道，DMA 寄存器不会被复位。DMA 通道寄存器（BDMA_CCRx、BDMA_CPARx 和 BDMA_CMARx）仍保留在通道配置阶段的设置值。*

在循环模式下，最后一次传输完成后，BDMA_CNDTRx 寄存器将自动重新加载初始编程值。当前的内部地址寄存器重新加载 BDMA_CPARx/BDMA_CMARx 寄存器中的基址值。

通道配置流程

配置 DMA 通道 x（其中 x 为通道编号）时应遵循下列顺序：

1. 在 BDMA_CPARx 寄存器中设置外设寄存器地址。外设事件发生后，数据会从此地址移动到存储器或从存储器移动到此地址。
2. 设置 BDMA_CMARx 寄存器中的存储器地址。外设事件发生后，将从此存储器读取数据或将数据写入此存储器。
3. 在 BDMA_CNDTRx 寄存器中配置待传输的总数据数。在每次外设事件发生后，此值都会递减。
4. 使用 BDMA_CCRx 寄存器中的 PL[1:0] 位来配置通道优先级。
5. 在 BDMA_CCRx 寄存器中配置数据传输方向、循环模式、外设和存储器递增模式、外设和存储器数据大小以及中断方式（数据传输一半产生中断还是数据完全传输后产生中断）
6. 将 BDMA_CCRx 寄存器中的 ENABLE 位置 1 以激活通道。

通道使能后，即可响应与该通道相连的外设所发出的所有 DMA 请求。

在数据传输一半后，传输一半标志 (HTIF) 便会置 1，如果传输一半中断使能位 (HTIE) 置 1，还会生成中断。传输结束时，传输完成标志 (TCIF) 便会置 1，如果传输完成中断使能位 (TCIE) 置 1，还会生成中断。

循环模式

循环模式可用于处理循环缓冲区和连续数据流（例如 ADC 扫描模式）。可以使用 BDMA_CCRx 寄存器中的 CIRC 位使能此特性。如果循环模式已激活，带传输数据的数目将自动加载为在通道配置阶段设置的初始值，并继续响应 DMA 请求。

存储器到存储器模式

DMA 通道在没有外设请求触发的情况下同样可以工作。此模式称为“存储器到存储器”模式。

如果 BDMA_CCRx 寄存器中的 MEM2MEM 位置 1，则只要通过软件将 BDMA_CCRx 寄存器中的使能位 (EN) 置 1，通道就会启动传输。BDMA_CNDTRx 寄存器达到零后，传输停止。存储器到存储器模式不能与循环模式同时使用。

16.3.4 可编程数据宽度、数据对齐和字节存储次序

如果 PSIZE 与 MSIZE 不相等，则 DMA 将按照*表 105：可编程的数据宽度、字节存储次序（位 PINC = MINC = 1 时）*所述方式进行数据对齐。

表 105. 可编程的数据宽度、字节存储次序（位 PINC = MINC = 1 时）

源端口宽度	目标端口宽度	要传输的数据项的数目 (NDT)	源内容：地址或/数据	传输操作	目标内容：地址/数据
8	8	4	@0x0/B0 @0x1/B1 @0x2/B2 @0x3/B3	1：读取 B0[7:0] @0x0，然后写入 B0[7:0] @0x0 2：读取 B1[7:0] @0x1，然后写入 B1[7:0] @0x1 3：读取 B2[7:0] @0x2，然后写入 B2[7:0] @0x2 4：读取 B3[7:0] @0x3，然后写入 B3[7:0] @0x3	@0x0/B0 @0x1/B1 @0x2/B2 @0x3/B3
8	16	4	@0x0/B0 @0x1/B1 @0x2/B2 @0x3/B3	1：读取 B0[7:0] @0x0，然后写入 00B0[15:0] @0x0 2：读取 B1[7:0] @0x1，然后写入 00B1[15:0] @0x2 3：读取 B2[7:0] @0x2，然后写入 00B2[15:0] @0x4 4：读取 B3[7:0] @0x3，然后写入 00B3[15:0] @0x6	@0x0/00B0 @0x2/00B1 @0x4/00B2 @0x6/00B3
8	32	4	@0x0/B0 @0x1/B1 @0x2/B2 @0x3/B3	1：读取 B0[7:0] @0x0，然后写入 000000B0[31:0] @0x0 2：读取 B1[7:0] @0x1，然后写入 000000B1[31:0] @0x4 3：读取 B2[7:0] @0x2，然后写入 000000B2[31:0] @0x8 4：读取 B3[7:0] @0x3，然后写入 000000B3[31:0] @0xC	@0x0/000000B0 @0x4/000000B1 @0x8/000000B2 @0xC/000000B3
16	8	4	@0x0/B1B0 @0x2/B3B2 @0x4/B5B4 @0x6/B7B6	1：读取 B1B0[15:0] @0x0，然后写入 B0[7:0] @0x0 2：读取 B3B2[15:0] @0x2，然后写入 B2[7:0] @0x1 3：读取 B5B4[15:0] @0x4，然后写入 B4[7:0] @0x2 4：读取 B7B6[15:0] @0x6，然后写入 B6[7:0] @0x3	@0x0/B0 @0x1/B2 @0x2/B4 @0x3/B6
16	16	4	@0x0/B1B0 @0x2/B3B2 @0x4/B5B4 @0x6/B7B6	1：读取 B1B0[15:0] @0x0，然后写入 B1B0[15:0] @0x0 2：读取 B3B2[15:0] @0x2，然后写入 B3B2[15:0] @0x2 3：读取 B5B4[15:0] @0x4，然后写入 B5B4[15:0] @0x4 4：读取 B7B6[15:0] @0x6，然后写入 B7B6[15:0] @0x6	@0x0/B1B0 @0x2/B3B2 @0x4/B5B4 @0x6/B7B6
16	32	4	@0x0/B1B0 @0x2/B3B2 @0x4/B5B4 @0x6/B7B6	1：读取 B1B0[15:0] @0x0，然后写入 0000B1B0[31:0] @0x0 2：读取 B3B2[15:0] @0x2，然后写入 0000B3B2[31:0] @0x4 3：读取 B5B4[15:0] @0x4，然后写入 0000B5B4[31:0] @0x8 4：读取 B7B6[15:0] @0x6，然后写入 0000B7B6[31:0] @0xC	@0x0/0000B1B0 @0x4/0000B3B2 @0x8/0000B5B4 @0xC/0000B7B6
32	8	4	@0x0/B3B2B1B0 @0x4/B7B6B5B4 @0x8/BBBAB9B8 @0xC/BFBEBDBC	1：读取 B3B2B1B0[31:0] @0x0，然后写入 B0[7:0] @0x0 2：读取 B7B6B5B4[31:0] @0x4，然后写入 B4[7:0] @0x1 3：读取 BBBAB9B8[31:0] @0x8，然后写入 B8[7:0] @0x2 4：读取 BFBEBDBC[31:0] @0xC，然后写入 BC[7:0] @0x3	@0x0/B0 @0x1/B4 @0x2/B8 @0x3/BC
32	16	4	@0x0/B3B2B1B0 @0x4/B7B6B5B4 @0x8/BBBAB9B8 @0xC/BFBEBDBC	1：读取 B3B2B1B0[31:0] @0x0，然后写入 B1B0[15:0] @0x0 2：读取 B7B6B5B4[31:0] @0x4，然后写入 B5B4[15:0] @0x2 3：读取 BBBAB9B8[31:0] @0x8，然后写入 B9B8[15:0] @0x4 4：读取 BFBEBDBC[31:0] @0xC，然后写入 BDBC[15:0] @0x6	@0x0/B1B0 @0x2/B5B4 @0x4/B9B8 @0x6/BDBC
32	32	4	@0x0/B3B2B1B0 @0x4/B7B6B5B4 @0x8/BBBAB9B8 @0xC/BFBEBDBC	1：读取 B3B2B1B0[31:0] @0x0，然后写入 B3B2B1B0[31:0] @0x0 2：读取 B7B6B5B4[31:0] @0x4，然后写入 B7B6B5B4[31:0] @0x4 3：读取 BBBAB9B8[31:0] @0x8，然后写入 BBBAB9B8[31:0] @0x8 4：读取 BFBEBDBC[31:0] @0xC，然后写入 BFBEBDBC[31:0] @0xC	@0x0/B3B2B1B0 @0x4/B7B6B5B4 @0x8/BBBAB9B8 @0xC/BFBEBDBC

解决 AHB 外设无法支持字节或半字写操作的问题

如果 DMA 初始化了 AHB 字节或半字写操作，则会将数据复制到 HWDATA[31:0] 总线的未使用数据线上。因此，若所用的 AHB 从外设不支持字节或半字写操作（即外设未使用 HSIZE）*且*不会产生任何错误，则 DMA 会对 HWDATA 的 32 个位执行写操作，如下列两个示例所示：

- 要写入半字“0xABCD”，则 DMA 会将 HWDATA 总线设为“0xABCDABCD”，同时将 HSIZE 设为 HalfWord
- 要写入字节“0xAB”，则 DMA 会将 HWDATA 总线设为“0xABABABAB”，同时将 HSIZE 设为 Byte

假设 AHB/APB 桥为 AHB 32 位从外设，且该外设未设置数据的 HSIZE，则任何 AHB 字节或半字操作都将转换为 32 位 APB 操作，转换方式如下所示：

- 将 AHB 字节写操作转化为 APB 字写操作，如向 0x0（或 0x1、0x2、0x3）写入数据“0xB0”转化为向 0x0 写入数据 “0xB0B0B0B0”
- 将 AHB 半字写操作转化为 APB 字写操作，如向 0x0（或 0x2）写入数据“0xB1B0”转化为向 0x0 写入数据“0xB1B0B1B0”

例如，如果要对 APB 备份寄存器（与 32 位地址边界对齐的16 位寄存器）执行写操作，则软件必须将存储器源大小 (MSIZE) 配置为“16 位”，同时将外设目标大小 (PSIZE) 配置为“32 位”。

16.3.5 错误管理

当对保留的地址空间执行读取操作时，将生成 DMA 传输错误。若在 DMA 读或写访问过程中生成了 DMA 传输错误，则会将相应的通道配置寄存器 (BDMA_CCRx) 的 EN 位进行硬件清零，从而自动禁止出错的通道。BDMA_IFR 寄存器中该通道的传输错误中断标志 (TEIF) 将置 1，如果 BDMA_CCRx 寄存器中的传输错误中断使能位 (TEIE) 置 1，则生成中断。

16.3.6 BDMA 中断

对于每个 DMA 通道，在发生“半传输”、“传输完成”或“传输错误”时都可以产生中断。可以使用单独的中断使能位以提高灵活性。

表 106. BDMA 中断请求

中断事件	事件标志	使能控制位
半传输	HTIF	HTIE
传输完成	TCIF	TCIE
传输错误	TEIF	TEIE

*第 17.3.3 节：DMAMUX2 映射*对 BDMA 请求映射进行了介绍。

16.4 BDMA 寄存器

有关寄存器说明中使用的缩写，请参见*第 94 页的第 1.1 节*。

外设寄存器可支持字节（8 位）、半字（16 位）或字（32 位）访问。

16.4.1 DMA 中断状态寄存器 (BDMA_ISR)

DMA interrupt status register

偏移地址：0x00

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TEIF8	HTIF8	TCIF8	GIF8	TEIF7	HTIF7	TCIF7	GIF7	TEIF6	HTIF6	TCIF6	GIF6	TEIF5	HTIF5	TCIF5	GIF5
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TEIF4	HTIF4	TCIF4	GIF4	TEIF3	HTIF3	TCIF3	GIF3	TEIF2	HTIF2	TCIF2	GIF2	TEIF1	HTIF1	TCIF1	GIF1
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31、27、23、19、15、11、7、3 **TEIFx**：通道 x 传输错误标志 (Channel x transfer error flag) (x = 1..8)
此位将由硬件置 1，由软件清零，软件只需将 1 写入 BDMA_IFCR 寄存器的相应位。
0：通道 x 上无传输错误 (TE)
1：通道 x 上出现传输错误 (TE)

位 30、26、22、18、14、10、6、2 **HTIFx**：通道 x 半传输标志 (Channel x half transfer flag) (x = 1..8)
此位将由硬件置 1，由软件清零，软件只需将 1 写入 BDMA_IFCR 寄存器的相应位。
0：通道 x 上无半传输 (HT) 事件
1：通道 x 上发生半传输 (HT) 事件

位 29、25、21、17、13、9、5、1 **TCIFx**：通道 x 传输完成标志 (Channel x transfer complete flag) (x = 1..8)
此位将由硬件置 1，由软件清零，软件只需将 1 写入 BDMA_IFCR 寄存器的相应位。
0：通道 x 上无传输完成 (TC) 事件
1：通道 x 上发生传输完成 (TC) 事件

位 28、24、20、16、12、8、4、0 **GIFx**：通道 x 全局中断标志 (Channel x global interrupt flag) (x = 1..8)
此位将由硬件置 1，由软件清零，软件只需将 1 写入 BDMA_IFCR 寄存器的相应位。
0：通道 x 上无 TE、HT 或 TC 事件
1：通道 x 上发生 TE、HT 或 TC 事件

16.4.2 DMA 中断标志清零寄存器 (BDMA_IFCR)

DMA interrupt flag clear register

偏移地址：0x04

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CTEIF8	CHTIF8	CTCIF8	CGIF8	CTEIF7	CHTIF7	CTCIF7	CGIF7	CTEIF6	CHTIF6	CTCIF6	CGIF6	CTEIF5	CHTIF5	CTCIF5	CGIF5
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CTEIF4	CHTIF4	CTCIF4	CGIF4	CTEIF3	CHTIF3	CTCIF3	CGIF3	CTEIF2	CHTIF2	CTCIF2	CGIF2	CTEIF1	CHTIF1	CTCIF1	CGIF1
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

位 31、27、23、19、15、11、7、3 **CTEIFx**：通道 x 传输错误清零 (Channel x transfer error clear) (x = 1..8)
该位由软件置 1，
0：无影响
1：清除 BDMA_ISR 寄存器中相应的 TEIF 标志

位 30、26、22、18、14、10、6、2 **CHTIFx**：通道 x 半传输清零 (Channel x half transfer clear) (x = 1..8)
该位由软件置 1，
0：无影响
1：清除 BDMA_ISR 寄存器中相应的 HTIF 标志

位 29、25、21、17、13、9、5、1 **CTCIFx**：通道 x 传输完成清零 (Channel x transfer complete clear) (x = 1..8)
该位由软件置 1，
0：无影响
1：清除 BDMA_ISR 寄存器中相应的 TCIF 标志

位 28、24、20、16、12、8、4、0 **CGIFx**：通道 x 全局中断清零 (Channel x global interrupt clear) (x = 1..8)
该位由软件置 1，
0：无影响
1：清除 BDMA_ISR 寄存器中的 GIF、TEIF、HTIF 和 TCIF 标志

16.4.3 DMA 通道 x 配置寄存器 (BDMA_CCRx)（x = 1..8，其中 x 表示通道编号）

DMA channel x configuration register

偏移地址：0x08 + 0d20 × (通道编号 – 1)

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	MEM2MEM	PL[1:0]		MSIZE[1:0]		PSIZE[1:0]		MINC	PINC	CIRC	DIR	TEIE	HTIE	TCIE	EN
	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:15 保留，必须保持复位值。

位 14 **MEM2MEM**：存储器到存储器模式 (Memory to memory mode)
此位由软件置 1 和清零。
0：禁止存储器到存储器模式
1：使能存储器到存储器模式

位 13:12 **PL[1:0]**：通道优先级 (Channel priority level)
这些位将由软件置 1 和清零。
00：低
01：中
10：高
11：非常高

位 11:10 **MSIZE[1:0]**：存储器大小 (Memory size)
这些位将由软件置 1 和清零。
00：8 位
01：16 位
10：32 位
11：保留

位 9:8 **PSIZE[1:0]**：外设大小 (Peripheral size)
这些位将由软件置 1 和清零。
00：8 位
01：16 位
10：32 位
11：保留

位 7 **MINC**：存储器递增模式 (Memory increment mode)
此位由软件置 1 和清零。
0：禁止存储器递增模式
1：使能存储器递增模式

位 6 **PINC**：外设递增模式 (Peripheral increment mode)
此位由软件置 1 和清零。
0：禁止外设递增模式
1：使能外设递增模式

位 5 **CIRC**：循环模式 (Circular mode)
此位由软件置 1 和清零。
0：禁止循环模式
1：使能循环模式

位 4 **DIR**：数据传输方向 (Data transfer direction)
此位由软件置 1 和清零。
0：从外设读取
1：从存储器读取

位 3 **TEIE**：传输错误中断使能 (Transfer error interrupt enable)
此位由软件置 1 和清零。
0：禁止 TE 中断
1：使能 TE 中断

位 2 **HTIE**：半传输中断使能 (Half transfer interrupt enable)
此位由软件置 1 和清零。
0：禁止 HT 中断
1：使能 HT 中断

位 1 **TCIE**：传输完成中断使能 (Transfer complete interrupt enable)
此位由软件置 1 和清零。
0：禁止 TC 中断
1：使能 TC 中断

位 0 **EN**：通道使能 (Channel enable)
此位由软件置 1 和清零。
0：禁止通道
1：使能通道

16.4.4 DMA 通道 x 数据数寄存器 (BDMA_CNDTRx)（x = 1..8，其中 x 表示通道编号）

DMA channel x number of data register

偏移地址：0x0C + 0d20 × (通道编号 – 1)

复位值：0x0000 0000

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

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

位 31:16 保留，必须保持复位值。

位 15:0 **NDT[15:0]**：待传输的数据数 (Number of data to transfer)

待传输的数据数（0 到 65535）。只有在通道禁止时才能写入此寄存器。通道使能后，该寄存器为只读，用于指示待传输的剩余字节数。每次 DMA 传输后，此寄存器将递减。

传输完成后，该寄存器的值可以保持为零，若已将通道配置为循环模式，则自动重新加载先前编程的值。

若该寄存器的值为零，则无论通道是否使能，都将不会处理任何事务。

16.4.5 DMA 通道 x 外设地址寄存器 (BDMA_CPARx)（x = 1..8，其中 x 表示通道编号）

DMA channel x peripheral address register

偏移地址：0x10 + 0d20 × (通道编号 – 1)

复位值：0x0000 0000

若通道已使能，则 *不得* 对该寄存器执行写操作。

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

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

位 31:0 **PA[31:0]**：外设地址 (Peripheral address)

读/写数据的外设数据寄存器的基址。

若 PSIZE 为 01（16 位），则忽略 PA[0] 位。访问将自动对齐到半字地址。

若 PSIZE 为 10（32 位），则忽略 PA[1:0]。访问将自动对齐到字地址。

16.4.6 DMA 通道 x 存储器地址寄存器 (BDMA_CMARx)（x = 1..8，其中 x 表示通道编号）

DMA channel x memory address register

偏移地址：0x14 + 0d20 × (通道编号 – 1)

复位值：0x0000 0000

若通道已使能，则*不得*对该寄存器执行写操作。

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

位 31:0 **MA[31: 0]**：存储器地址 (Memory Address)

读/写数据的存储区基址。

若 MSIZE 为 01（16 位），则忽略 MA[0] 位。访问将自动对齐到半字地址。

若 MSIZE 为 10（32 位），则忽略 MA[1:0]。访问将自动对齐到字地址。

16.4.7 BDMA寄存器映射

下表列出了 DMA 寄存器映射和复位值。

表 107. BDMA 寄存器映射和复位值

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	**BDMA_ISR**	TEIF8	HTIF8	TCIF8	GIF8	TEIF7	HTIF7	TCIF7	GIF7	TEIF6	HTIF6	TCIF6	GIF6	TEIF5	HTIF5	TCIF5	GIF5	TEIF4	HTIF4	TCIF4	GIF4	TEIF3	HTIF3	TCIF3	GIF3	TEIF2	HTIF2	TCIF2	GIF2	TEIF1	HTIF1	TCIF1	GIF1
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x04	**BDMA_IFCR**	CTEIF8	CHTIF8	CTCIF8	CGIF8	CTEIF7	CHTIF7	CTCIF7	CGIF7	CTEIF6	CHTIF6	CTCIF6	CGIF6	CTEIF5	CHTIF5	CTCIF5	CGIF5	CTEIF4	CHTIF4	CTCIF4	CGIF4	CTEIF3	CHTIF3	CTCIF3	CGIF3	CTEIF2	CHTIF2	CTCIF2	CGIF2	CTEIF1	CHTIF1	CTCIF1	CGIF1
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x08	**BDMA_CCR1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MEM2MEM	PL [1:0]		MSIZE [1:0]		PSIZE [1:0]		MINC	PINC	CIRC	DIR	TEIE	HTIE	TCIE	EN
	Reset value																		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0C	**BDMA_CNDTR1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	NDT[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x10	**BDMA_CPAR1**	PA[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x14	**BDMA_CMAR1**	MA[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x18	Reserved	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
0x1C	**BDMA_CCR2**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MEM2MEM	PL [1:0]		MSIZE [1:0]		PSIZE [1:0]		MINC	PINC	CIRC	DIR	TEIE	HTIE	TCIE	EN
	Reset value																		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x20	**BDMA_CNDTR2**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	NDT[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x24	**BDMA_CPAR2**	PA[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x28	**BDMA_CMAR2**	MA[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x2C	Reserved	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
0x30	**BDMA_CCR3**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MEM2MEM	PL [1:0]		MSIZE [1:0]		PSIZE [1:0]		MINC	PINC	CIRC	DIR	TEIE	HTIE	TCIE	EN
	Reset value																		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x34	**BDMA_CNDTR3**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	NDT[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x38	**BDMA_CPAR3**	PA[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x3C	**BDMA_CMAR3**	MA[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

表 107. BDMA 寄存器映射和复位值（续）

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x40	Reserved	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
0x44	**BDMA_CCR4**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MEM2MEM	PL [1:0]		MSIZE [1:0]		PSIZE [1:0]		MINC	PINC	CIRC	DIR	TEIE	HTIE	TCIE	EN
	Reset value																		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x48	**BDMA_CNDTR4**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	NDT[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x4C	**BDMA_CPAR4**	PA[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x50	**BDMA_CMAR4**	MA[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x54	Reserved	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
0x58	**BDMA_CCR5**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MEM2MEM	PL [1:0]		MSIZE [1:0]		PSIZE [1:0]		MINC	PINC	CIRC	DIR	TEIE	HTIE	TCIE	EN
	Reset value																		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x5C	**BDMA_CNDTR5**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	NDT[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x60	**BDMA_CPAR5**	PA[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x64	**BDMA_CMAR5**	MA[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x68	Reserved	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
0x6C	**BDMA_CCR6**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MEM2MEM	PL [1:0]		MSIZE [1:0]		PSIZE [1:0]		MINC	PINC	CIRC	DIR	TEIE	HTIE	TCIE	EN
	Reset value																		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x70	**BDMA_CNDTR6**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	NDT[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x74	**BDMA_CPAR6**	PA[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x78	**BDMA_CMAR6**	MA[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x7C	Reserved	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
0x80	**BDMA_CCR7**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MEM2MEM	PL [1:0]		MSIZE [1:0]		PSIZE [1:0]		MINC	PINC	CIRC	DIR	TEIE	HTIE	TCIE	EN
	Reset value																		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x84	**BDMA_CNDTR7**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	NDT[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

表 107. BDMA 寄存器映射和复位值（续）

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x88	**BDMA_CPAR7**	PA[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x8C	**BDMA_CMAR7**	MA[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x90	Reserved	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
0x94	**BDMA_CCR8**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MEM2MEM	PL [1:0]		MSIZE [1:0]		PSIZE [1:0]		MINC	PINC	CIRC	DIR	TEIE	HTIE	TCIE	EN
	Reset value																		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x98	**BDMA_CNDTR8**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	NDT[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x9C	**BDMA_CPAR8**	PA[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0xA0	**BDMA_CMAR8**	MA[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

有关寄存器边界地址的信息，请参见*第 100 页的第 2.2.2 节*。

17 DMA 请求复用器 (DMAMUX)

17.1 简介

外设通过设置其 DMA 请求信号来指示存在 DMA 传输请求。DMA 控制器会处理 DMA 请求并生成 DMA 确认信号，而且相应的 DMA 请求信号也将变为无效，但在此之前 DMA 请求一直处于挂起状态。

在本文档中，DMA 请求/确认协议所需的控制信号组并未明确列出和说明，而是称作 DMA 请求线。

DMAMUX 请求复用器可在产品的外设和 DMA 控制器之间重新配置（路由）DMA 请求线。该路由功能通过可编程的多通道 DMA 请求线复用器来确保实现。每条通道可不受限制地选择一个 DMA 请求线，或者按照与来自其 DMAMUX 同步输入的事件相同步的方式选择一个 DMA 请求线。此外，DMAMUX 还可用作其输入触发信号的可编程事件的 DMA 请求发生器。

*第 17.3.1 节*规定了 DMAMUX 实例的数量及其主要特性。

DMAMUX 请求复用器输入到外设 DMA 请求线和到 DMAMUX 请求发生器输出的分配、DMAMUX 请求复用器输出到 DMA 控制器通道的分配以及 DMAMUX 同步和触发输入到内部和外部信号的分配取决于产品实现，具体在*第 17.6 节*中进行了介绍。

17.2 DMAMUX 主要特性

- 多达 16 通道可编程 DMA 请求线复用器输出
- 多达 8 通道 DMA 请求发生器
- 多达 32 个 DMA 请求发生器的触发输入
- 多达 16 个同步输入
- 每个 DMA 请求发生器通道均具有：
  - DMA 请求触发输入选择器
  - DMA 请求计数器
  - 所选 DMA 请求触发输入的事件溢出标志
- 每个 DMA 请求线复用器通道输出均具有：
  - 多达 107 个来自外设的输入 DMA 请求线
  - 一个 DMA 请求线输出
  - 同步输入选择器
  - DMA 请求计数器
  - 所选同步输入的事件溢出标志
  - 一个事件输出，用于 DMA 请求链接

17.3 DMAMUX 实现

17.3.1 DMAMUX1 和 DMAMUX2 实例化

产品集成 DMA 请求复用器的两个实例：

- DMAMUX1，用于 DMA1 和 DMA2（D2 域）
- DMAMUX2，用于 BDMA（D3 域）

DMAMUX1 和 DMAMUX2 通过下表中列出的硬件配置参数进行实例化。

表 108. DMAMUX1 和 DMAMUX2 实例化

特性	DMAMUX1	DMAMUX2
DMAMUX 输出请求通道数	16	8
DMAMUX 请求发生器通道数	8	8
DMAMUX 请求触发输入数	8	32
DMAMUX 同步输入数	8	16
DMAMUX 外设请求输入数	107	12

17.3.2 DMAMUX1 映射

资源到 DMAMUX1 的映射通过硬接线实现。

DMAMUX1 与 D2 域中的 DMA1 和 DMA2 配合使用：

- DMAMUX1 通道 0 到 7 与 DMA1 通道 0 到 7 相连
- DMAMUX1 通道 8 到 15 与 DMA2 通道 0 到 7 相连

表 109. DMAMUX1：复用器输入到资源的分配

DMA 请求 MUX 输入	资源	DMA 请求 MUX 输入	资源	DMA 请求 MUX 输入	资源
1	dmamux1_req_gen0	40	SPI2_TX	79	UART7_RX
2	dmamux1_req_gen1	41	USART1_RX	80	UART7_TX
3	dmamux1_req_gen2	42	USART1_TX	81	UART8_RX
4	dmamux1_req_gen3	43	USART2_RX	82	UART8_TX
5	dmamux1_req_gen4	44	USART2_TX	83	SPI4_RX
6	dmamux1_req_gen5	45	USART3_RX	84	SPI4_TX
7	dmamux1_req_gen6	46	USART3_TX	85	SPI5_RX
8	dmamux1_req_gen7	47	TIM8_CH1	86	SPI5_TX
9	ADC1	48	TIM8_CH2	87	SAI1_A
10	ADC2	49	TIM8_CH3	88	SAI1_B
11	TIM1_CH1	50	TIM8_CH4	89	SAI2_A
12	TIM1_CH2	51	TIM8_UP	90	SAI2_B
13	TIM1_CH3	52	TIM8_TRIG	91	SWPMI_RX

表 109. DMAMUX1：复用器输入到资源的分配（续）

DMA 请求 MUX 输入	资源	DMA 请求 MUX 输入	资源	DMA 请求 MUX 输入	资源
14	TIM1_CH4	53	TIM8_COM	92	SWPMI_TX
15	TIM1_UP	54	保留	93	SPDIFRX_DT
16	TIM1_TRIG	55	TIM5_CH1	94	SPDIFRX_CS
17	TIM1_COM	56	TIM5_CH2	95	HR_REQ(1)
18	TIM2_CH1	57	TIM5_CH3	96	HR_REQ(2)
19	TIM2_CH2	58	TIM5_CH4	97	HR_REQ(3)
20	TIM2_CH3	59	TIM5_UP	98	HR_REQ(4)
21	TIM2_CH4	60	TIM5_TRIG	99	HR_REQ(5)
22	TIM2_UP	61	SPI3_RX	100	HR_REQ(6)
23	TIM3_CH1	62	SPI3_TX	101	dfsdm1_dma0
24	TIM3_CH2	63	UART4_RX	102	dfsdm1_dma1
25	TIM3_CH3	64	UART4_TX	103	dfsdm1_dma2
26	TIM3_CH4	65	USART5_RX	104	dfsdm1_dma3
27	TIM3_UP	66	UART5_TX	105	TIM15_CH1
28	TIM3_TRIG	67	DAC1	106	TIM15_UP
29	TIM4_CH1	68	DAC2	107	TIM15_TRIG
30	TIM4_CH2	69	TIM6_UP	108	TIM15_COM
31	TIM4_CH3	70	TIM7_UP	109	TIM16_CH1
32	TIM4_UP	71	USART6_RX	110	TIM16_UP
33	I2C1_RX	72	USART6_TX	111	TIM17_CH1
34	I2C1_TX	73	I2C3_RX	112	TIM17_UP
35	I2C2_RX	74	I2C3_TX	113	SAI3_A
36	I2C2_TX	75	DCMI	114	SAI3_B
37	SPI1_RX	76	CRYP_IN	115	ADC3
38	SPI1_TX	77	CRYP_OUT	-	-
39	SPI2_RX	78	HASH_IN	-	-

表 110. DMAMUX1：触发输入到资源的分配

触发输入	资源	触发输入	资源
0	dmamux1_evt0	4	LPTIMER2_out
1	dmamux1_evt1	5	LPTIMER3_out
2	dmamux1_evt2	6	extit0
3	LPTIMER1_out	7	TIM12_TRGO

表 111. DMAMUX1：同步输入到资源的分配

SYNC 输入	资源	SYNC 输入	资源
0	dmamux1_evt0	4	LPTIMER2_out
1	dmamux1_evt1	5	LPTIMER3_out
2	dmamux1_evt2	6	extit0
3	LPTIMER1_out	7	TIM12_TRGO

17.3.3 DMAMUX2 映射

DMAMUX2 通道 0 到 7 与 BDMA 通道 0 到 7 相连。

表 112. DMAMUX2：复用器输入到资源的分配

DMA 请求 MUX 输入	资源	DMA 请求 MUX 输入	资源
1	dmamux2_req_gen0	11	SPI6_RX
2	dmamux2_req_gen1	12	SPI6_TX
3	dmamux2_req_gen2	13	I2C4_RX
4	dmamux2_req_gen3	14	I2C4_TX
5	dmamux2_req_gen4	15	SAI4_A
6	dmamux2_req_gen5	16	SAI4_B
7	dmamux2_req_gen6	17	ADC3_REQ
8	dmamux2_req_gen7	18	保留
9	LP UART1_RX	19	保留
10	LP UART1_TX	20	保留

表 113. DMAMUX2：触发输入到资源的分配

触发输入	资源	触发输入	资源
0	dmamux2_evt0	16	Spi6_it_async
1	dmamux2_evt1	17	Comp1_out
2	dmamux2_evt2	18	Comp2_out
3	dmamux2_evt3	19	RTC_wkup
4	dmamux2_evt4	20	Syscfg_exti0_mux
5	dmamux2_evt5	21	Syscfg_exti2_mux
6	dmamux2_evt6	22	I2c4_it_event
7	Lpuart1_it_R_WUP_ASYNC	23	Spi6_it
8	Lpuart1_it_T_WUP_ASYNC	24	Lpuart1_it_T
9	Lptim2_ait	25	Lpuart1_it_R
10	Lptim2_out	26	ADC3_it
11	Lptim3_ait	27	ADC3_AWD1_out

表 113. DMAMUX2：触发输入到资源的分配（续）

触发输入	资源	触发输入	资源
12	Lptim3_out	28	DMA1_D3_ch0_it
13	Lptim4_ait	29	DMA1_D3_ch1_it
14	Lptim5_ait	30	保留
15	I2c4_it_async	31	保留

表 114. DMAMUX2：同步输入到资源的分配

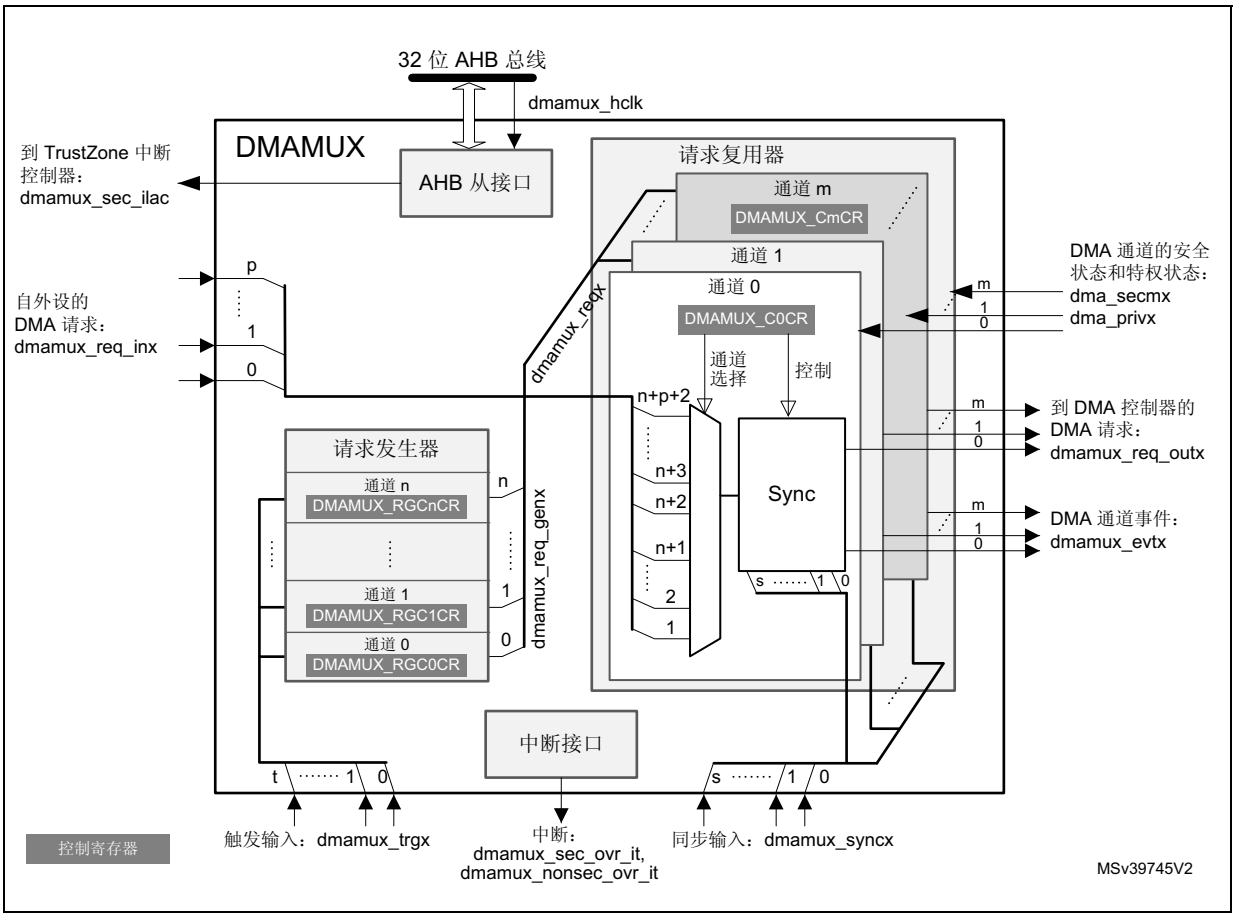
同步输入	资源	同步输入	资源
0	dmamux2_evt0	8	Lptim2_out
1	dmamux2_evt1	9	Lptim3_out
2	dmamux2_evt2	10	I2c4_it_async
3	dmamux2_evt3	11	Spi6_it_async
4	dmamux2_evt4	12	Comp1_out
5	dmamux2_evt5	13	RTC_wkup
6	Lpuart1_it_R_WUP_ASYNC	14	Syscfg_exti0_mux
7	Lpuart1_it_T_WUP_ASYNC	15	Syscfg_exti2_mux

17.4 DMAMUX 功能说明

17.4.1 DMAMUX 框图

图 76 显示了 DMAMUX 框图。

图 76. DMAMUX 框图



DMAMUX 具有两个主要子模块：请求线复用器和请求线发生器。

该实现可：

- 分配 DMAMUX 请求复用器子模块输入 (dmamux_reqx)，这些输入来自外设 (dmamux_req_inx) 和 DMAMUX 请求发生器子模块的通道 (dmamux_req_genx)
- 将 DMAMUX 请求输出分配到 DMA 控制器的通道 (dmamux_req_outx)
- 将内部或外部信号分配到 DMA 请求触发输入 (dmamux_trgx)
- 将内部或外部信号分配到同步输入 (dmamux_syncx)

17.4.2 DMAMUX 信号

表 115 列出了 DMAMUX 信号。

表 115. DMAMUX 信号

信号名称	说明
dmamux_hclk	DMAMUX AHB 时钟
dmamux_req_inx	DMAMUX DMA 请求线输入（来自外设）
dmamux_trgx	DMAMUX DMA 请求触发输入（到请求发生器子模块）
dmamux_req_genx	DMAMUX 请求发生器子模块通道输出
dmamux_reqx	DMAMUX 请求复用器子模块输入（来自外设请求和请求发生器通道）
dmamux_syncx	DMAMUX 同步输入（到请求复用器子模块）
dmamux_req_outx	DMAMUX 请求输出（到 DMA 控制器）
dmamux_evtx	DMAMUX 事件输出
dmamux_ovr_it	DMAMUX 溢出中断

17.4.3 DMAMUX 通道

DMAMUX 通道是一个可能包括额外的 DMAMUX 请求发生器通道的 DMAMUX 请求复用器通道，具体取决于所选的请求复用器输入。

DMAMUX 请求复用器通道专用于连接到 DMA 控制器的一个通道。

通道配置流程

请遵循以下顺序来配置 DMAMUX x 通道和相关的 DMA 通道 y：

1. 对 DMA 通道 y 进行完整的设置和配置，但不要使能通道 y。
2. 对相关 DMAMUX y 通道进行完整的设置和配置。
3. 最后，将 DMA y 通道寄存器中的 EN 位置 1 以激活 DMA 通道。

17.4.4 DMAMUX 请求线复用器

DMAMUX 请求复用器及其多条通道可确保 DMA 请求/确认控制信号（称为 DMA 请求线）的实际路由。

每个 DMA 请求线并行连接到 DMAMUX请求线复用器的所有通道。

DMA 请求来自外设或 DMAMUX 请求发生器。

DMAMUX 请求线复用器通道 x 按照 DMAMUX_CxCR 寄存器中的 8 位 DMAREQ_ID 字段的配置来选择 DMA 请求线编号。

注： *字段 DMAREQ_ID 为空值表示未选择任何 DMA 请求线。*

不得为不同的 x 和 y DMAMUX 请求复用器通道编程相同的非空 DMA_REQ_ID 值（通过 DMAMUX_CxCR 和 DMAMUX CyCR）。

不允许为 DMAMUX 请求线复用器的两个不同通道配置相同的非零 DMAREQ_ID。

除了 DMA 请求选择外，还可根据需要配置和使能同步模式和/或事件生成。

同步模式和通道事件生成

每个 DMAMUX 请求线复用器通道 x 可单独同步，方法是将 DMAMUX_CxCR 寄存器中的同步使能 (SE) 位置 1。

DMAMUX 具有多个同步输入。同步输入并行连接到请求复用器的所有通道。

同步输入通过给定通道 x 的 DMAMUX_CxCR 寄存器中的 5 位 SYNC_ID 字段选择。

当某个通道处于此同步模式时，如果通过 DMAMUX_CxCR 寄存器的 SPOL[1:0] 字段检测到所选输入同步信号上的可编程上升沿/下降沿，则所选输入 DMA 请求线信号将被传送到复用器通道输出。

此外，DMAMUX 请求复用器内部有一个可编程 DMA 请求计数器，可用于实现通道请求输出生成功能以及事件生成功能。通道 x 输出上的事件生成功能通过 DMAMUX_CxCR 寄存器的 EGE 位（事件生成使能）使能。

如以下两个图所示，检测到同步输入的边沿后，所选输入 DMA 请求线将连接到 DMAMUX 复用器通道 x 输出。此后，所选 DMAMUX 请求线上每处理一个 DMA 请求（例如，当请求信号取消激活时）都会使 DMA 请求计数器减 1。下溢时，DMA 请求计数器将自动装入 DMAMUX_CxCR 寄存器的 NBREQ 字段中的值，输入 DMA 请求线将与复用器通道 x 输出断开。

因此，检测到同步事件后传送到复用器通道 x 输出的 DMA 请求数等于 NBREQ 字段中的值加 1。

注： *只能通过软件在相应复用器通道 x 的同步使能位 SE 和事件生成使能位 EGE 均禁止时写入 NBREQ 字段的值。*

如果 EGE 使能，当复用器通道的 DMA 请求计数器自动重新装入已编程 NBREQ 字段的值时，复用器通道会生成一个通道事件（即，一个 AHB 时钟周期的脉冲）。

图 77. DMAMUX 请求线复用器通道的同步模式

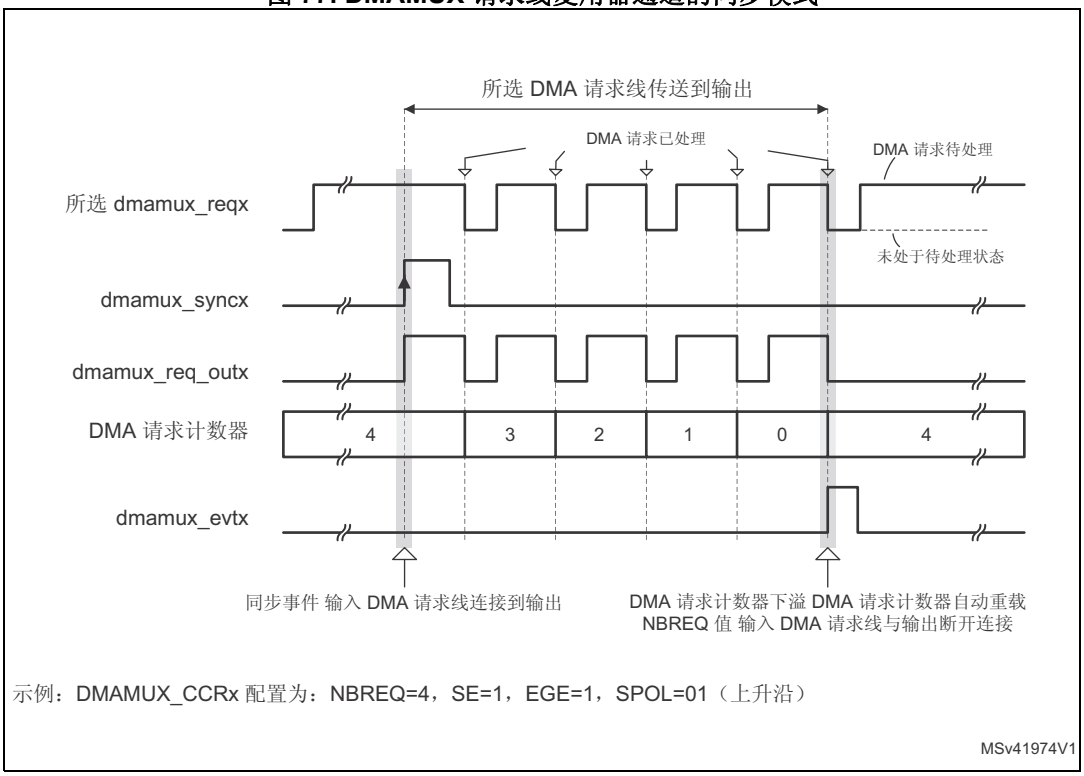
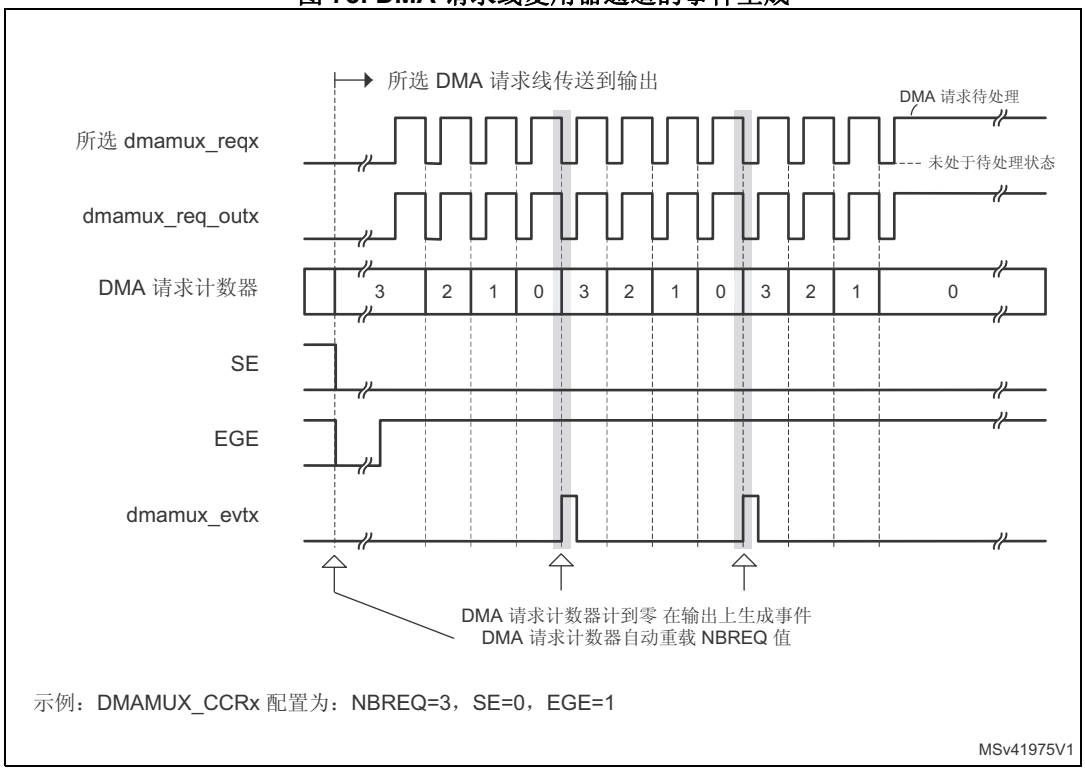


图 78. DMA 请求线复用器通道的事件生成



注： *如果边沿后的状态保持稳定的时间持续 2 个 AHB 时钟周期以上，则会检测到同步事件（边沿）。*

写入 DMAMUX_CxCR 寄存器后，同步事件将被屏蔽 3 个 AHB 时钟周期。

同步溢出和中断

如果在 DMA 请求计数器的值小于复用器通道 x 已编程的 NBREQ 字段值时发生新的同步事件，则 DMAMUX_CSR 状态寄存器中的同步溢出标志位 SOFx 将置 1。

注： *DMA 控制器的相关通道使用结束后，应禁止请求复用器通道 x 同步 (DMAMUX_CxCR.SE=0)。否则，在新检测到的同步事件后，由于未从 DMA 控制器收到 DMA 确认（即 无已处理请求），将会发生同步溢出。*

溢出标志 SOFx 的复位方式是将 DMAMUX_CFR 寄存器中的相关清除同步溢出标志位 CSOFx 置 1。

如果 DMAMUX_CxCR 寄存器中的同步溢出中断使能位 SOIE 置 1，则将同步溢出标志置 1 会产生中断。

17.4.5 DMAMUX 请求发生器

DMAMUX 请求发生器会在其 DMA 请求触发输入上出现触发事件后产生 DMA 请求。

DMAMUX 请求发生器有多个通道。DMA 请求触发输入并行连接到所有通道。

DMAMUX 请求发生器通道的输出是 DMAMUX 请求线复用器的输入。

每个 DMAMUX 请求发生器通道 x 在相应的 DMAMUX_RGxCR 寄存器中都有一个使能位 GE（发生器使能）。

DMAMUX 请求发生器通道 x 的 DMA 请求触发输入通过相应 DMAMUX_RGxCR 寄存器中的 SIG_ID（触发信号 ID）字段选择。

DMA 请求触发输入上的触发事件可以是上升沿、下降沿或任一边沿。有效边沿通过相应 DMAMUX_RGxCR 寄存器中的 GPOL（发生器极性）字段选择。

触发事件后，相应发生器通道开始在其输出上生成 DMA 请求。每个已处理 DMA 请求（即，当请求信号取消激活时）会使 DMAMUX 请求发生器内置的 DMA 请求计数器减 1。下溢时，DMA 请求计数器将自动装入相应 DMAMUX_RGxCR 寄存器的 GNBREQ 字段中的值，请求发生器通道将停止生成 DMA 请求。

因此，触发事件后生成的 DMA 请求的数量为 GNBREQ+1。

只要相应通道禁止（即，DMAMUX_RGxCR.GE 位为低电平），DMA 请求计数器便会保持 GNBREQ 字段的值。

注： *只能通过软件在相应发生器通道 x 的使能位 GE 被禁止时写入 GNBREQ 字段的值。*

如果边沿后的状态保持稳定的时间持续 2 个 AHB 时钟周期以上，则会检测到触发事件（边沿）。

写入 DMAMUX_RGxCR 寄存器后，触发事件将被屏蔽 3 个 AHB 时钟周期。

触发溢出和中断

如果在 DMAMUX 请求发生器计数器的值小于相应请求发生器通道 x 已编程的 GNBREQ 字段值时发生新的 DMA 请求触发事件，并且请求发生器通道 x 通过 GE 使能，则状态 DMAMUX_RGSR 寄存器中的请求触发事件溢出标志位 OFx 将通过硬件置为有效。

注： *DMA 控制器的相关通道使用结束后，应禁止请求发生器通道 x (DMAMUX_RGxCR.GE=0)。否则，在新检测到的触发事件后，由于未从 DMA 收到确认（即 无已处理请求），将会发生触发溢出。*

溢出标志 OFx 的复位方式是将 DMAMUX_RGCFR 寄存器中的相关清除溢出标志位 COFx 置 1。

如果 DMAMUX_RGxCR 寄存器中的 DMA 请求触发事件溢出中断使能位 OIE 置 1，则将 DMAMUX 请求触发溢出标志置 1 会产生中断。

17.5 DMAMUX 中断

发生以下事件时会产生中断：

- 每个 DMA 请求线复用器通道发生同步事件溢出
- 每个 DMA 请求发生器通道发生触发事件溢出

对任一事件，每个通道的中断使能、状态和清除标志寄存器位均可单独使用。

表 116. DMAMUX 中断

中断信号	中断事件	事件标志	清除位	使能位
dmamuxovr_it	DMAMUX 请求线复用器的通道 x 上发生同步事件溢出	SOFx	CSOFx	SOIE
	DMAMUX 请求发生器的通道 x 上发生触发事件溢出	OFx	COFx	OIE

17.6 DMAMUX1 寄存器

有关 DMAMUX1 和 DMAMUX2 基址的信息，请参见包括寄存器边界地址的表格。

DMAMUX 寄存器支持（8 位）字节、（16 位）半字或（32 位）字访问。地址应根据使用数据的大小进行对齐。

17.6.1 DMAMUX1 请求线复用器通道 x 配置寄存器 (DMAMUX1_CxCR)

DMAMUX1 request line multiplexer channel x configuration register

偏移地址：0x04 * x（x = 0 到 15）

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	SYNC_ID[4:0]					NBREQ[4:0]					SPOL[1:0]		SE
			rw					rw					rw		rw

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

位 31:29 保留，必须保持复位值

位 28:24 **SYNC_ID[4:0]**：同步标识 (Synchronization identification)
选择同步输入。（请参见*表 111：DMAMUX1：同步输入到资源的分配*）

位 23:19 **NBREQ[4:0]**：要转发的 DMA 请求数减 1 (Number of DMA requests minus 1 to forward)
定义在同步事件后要转发到 DMA 控制器的 DMA 请求的数量，和/或生成输出事件前的 DMA 请求的数量。
只有当 SE 和 EGE 位均为低电平时，才能写入该字段。

位 18:17 **SPOL[1:0]**：同步极性 (Synchronization polarity)
定义所选同步输入的边沿极性：
00：无事件。即，无同步和检测。
01：上升沿
10：下降沿
11：上升沿和下降沿

位 16 **SE**：同步使能 (Synchronization enable)
0：禁止同步
1：使能同步

位 15:10 保留，必须保持复位值

位 9 **EGE**：事件生成使能 (Event generation enable)
0：禁止事件生成
1：使能事件生成

位 8 **SOIE**：同步溢出中断使能 (Synchronization overrun interrupt enable)
0：禁止中断
1：使能中断

位 7:0 **DMAREQ_ID[7:0]**：DMA 请求标识 (DMA request identification)
选择输入 DMA 请求。（请参见表：DMAMUX - 复用器输入到资源的分配）

17.6.2 DMAMUX2 请求线复用器通道 x 配置寄存器 (DMAMUX2_CxCR)

DMAMUX2 request line multiplexer channel x configuration register

偏移地址：0x04 * x，其中 x = 0 到 7

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	SYNC_ID[4:0]					NBREQ[4:0]					SPOL[1:0]		SE
			rw					rw					rw		rw

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

位 31:29 保留，必须保持复位值

位 28:24 **SYNC_ID[4:0]**：同步标识 (Synchronization identification)
选择同步输入。（请参见表：DMAMUX - 同步输入到资源的分配）

位 23:19 **NBREQ[4:0]**：要转发的 DMA 请求数减 1 (Number of DMA requests minus 1 to forward)
定义在同步事件后要转发到 DMA 控制器的 DMA 请求的数量，和/或生成输出事件前的 DMA 请求的数量。
只有当 SE 和 EGE 位均为低电平时，才能写入该字段。

位 18:17 **SPOL[1:0]**：同步极性 (Synchronization polarity)
定义所选同步输入的边沿极性：
00：无事件。即，无同步和检测。
01：上升沿
10：下降沿
11：上升沿和下降沿

位 16 **SE**：同步使能 (Synchronization enable)
0：禁止同步
1：使能同步

位 15:10 保留，必须保持复位值

位 9 **EGE**：事件生成使能 (Event generation enable)
0：禁止事件生成
1：使能事件生成

位 8 **SOIE**：同步溢出中断使能 (Synchronization overrun interrupt enable)
0：禁止中断
1：使能中断

位 7:0 **DMAREQ_ID[7:0]**：DMA 请求标识 (DMA request identification)
选择输入 DMA 请求。（请参见表：DMAMUX - 复用器输入到资源的分配）

17.6.3 DMAMUX1 请求线复用器中断通道状态寄存器 (DMAMUX1_CSR)

DMAMUX1 request line multiplexer interrupt channel status register

偏移地址：0x080

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOF15	SOF14	SOF13	SOF12	SOF11	SOF10	SOF9	SOF8	SOF7	SOF6	SOF5	SOF4	SOF3	SOF2	SOF1	SOF0
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:16 保留，必须保持复位值。

位 15:0 **SOF[15:0]**：同步溢出事件标志 (Synchronization overrun event flag)

当 DMA 请求线复用器通道 x 上发生同步事件且 DMA 请求计数器的值小于 NBREQ 时，该标志将置 1。

该标志的清零方式是向 DMAMUX_CFR 寄存器中的相应 CSOFx 位写入 1。对于 DMAMUX2_CFR，位 15:8 保留。

17.6.4 DMAMUX2 请求线复用器中断通道状态寄存器 (DMAMUX2_CSR)

DMAMUX2 request line multiplexer interrupt channel status register

偏移地址：0x080

复位值：0x0000 0000

根据目标 DMAMUX2 请求线复用器通道 x 的安全模式，基于所连接的 DMA 控制器通道 y 的安全控制位，以及考虑到 DMAMUX2 x 通道输出连接到 DMA 的 y 通道，该寄存器应通过非安全读取或安全读取操作以位级访问（请参见 DMAMXUX2 映射实现部分）。

根据目标 DMAMUX2 请求线复用器通道 x 的特权模式，基于所连接的 DMA 控制器通道 y 的特权控制位，以及考虑到 DMAMUX2 x 通道输出连接到 DMA 的 y 通道，该寄存器应通过非特权读取或特权读取操作以位级访问（请参见 DMAMXUX2 映射实现部分）。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SOF7	SOF6	SOF5	SOF4	SOF3	SOF2	SOF1	SOF0
								r	r	r	r	r	r	r	r

位 31:8 保留，必须保持复位值。

位 7:0 **SOF[7:0]**：同步溢出事件标志 (Synchronization overrun event flag)

当 DMA 请求线复用器通道 x 上发生同步事件且 DMA 请求计数器的值小于 NBREQ 时，该标志将置 1。

该标志的清零方式是向 DMAMUX2_CFR 寄存器中的相应 CSOFx 位写入 1。

17.6.5 DMAMUX1 请求线复用器中断清除标志寄存器 (DMAMUX1_CFR)

DMAMUX1 request line multiplexer interrupt clear flag register

偏移地址：0x084

复位值：0x0000 0000

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

位 31:16 保留，必须保持复位值。

位 15:0 **CSOF[15:0]**：清除同步溢出事件标志 (Clear synchronization overrun event flag)
将 1 写入每个位时，DMAMUX_CSR 寄存器中相应的溢出标志 SOFx 将清零。

17.6.6 DMAMUX2 请求线复用器中断清除标志寄存器 (DMAMUX2_CFR)

DMAMUX2 request line multiplexer interrupt clear flag register

偏移地址：0x084

复位值：0x0000 0000

根据目标 DMAMUX 请求线复用器通道 x 的安全模式，基于所连接的 DMA 控制器通道 y 的安全控制位，以及考虑到 DMAMUX x 通道输出连接到 DMA 的 y 通道，该寄存器应通过非安全写入或安全写入操作以位级写入（请参见 DMAMXUX 映射实现部分）。

根据目标 DMAMUX 请求线复用器通道 x 的特权模式，基于所连接的 DMA 控制器通道 y 的特权控制位，以及考虑到 DMAMUX x 通道输出连接到 DMA 的 y 通道，该寄存器应通过非特权写入或特权写入操作以位级写入（请参见 DMAMXUX 映射实现部分）。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CSOF6	CSOF6	CSOF5	CSOF4	CSOF3	CSOF2	CSOF1	CSOF0
								w	w	w	w	w	w	w	w

位 31:8 保留，必须保持复位值。

位 7:0 **CSOF[7:0]**：清除同步溢出事件标志 (Clear synchronization overrun event flag)
将 1 写入每个位时，DMAMUX2_CSR 寄存器中相应的溢出标志 SOFx 将清零。

17.6.7 DMAMUX1 请求发生器通道 x 配置寄存器 (DMAMUX1_RGxCR)

DMAMUX1 request generator channel x configuration register

偏移地址：0x100 + 0x04 * (x-0)（x = 0 到 7）

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	GNBREQ[4:0]					GPOL[1:0]		GE
								rw					rw		rw

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

位 31:24 保留，必须保持复位值。

位 23:19 **GNBREQ[4:0]**：要生成的 DMA 请求数（减 1）(Number of DMA requests to be generated (minus 1))

定义触发事件后生成的 DMA 请求的数量。生成的 DMA 请求的实际数量为 GNBREQ+1。

注： 只有在 GE 位禁止时才能写入此字段。

位 18:17 **GPOL[1:0]**：DMA 请求发生器触发极性 (DMA request generator trigger polarity)

定义所选触发输入的边沿极性

00：无事件。即，无触发检测和生成。

01：上升沿

10：下降沿

11：上升沿和下降沿

位 16 **GE**：DMA 请求发生器通道 x 使能 (DMA request generator channel x enable)

0：禁止 DMA 请求发生器通道 x

1：使能 DMA 请求发生器通道 x

位 15:9 保留，必须保持复位值。

位 8 **OIE**：触发溢出中断使能 (Trigger overrun interrupt enable)

0：禁止在出现触发溢出事件时产生中断。

1：使能在出现触发溢出事件时产生中断。

位 7:5 保留，必须保持复位值。

位 4:0 **SIG_ID[4:0]**：信号标识 (Signal identification)

选择用于 DMA 请求发生器的通道 x 的 DMA 请求触发输入

17.6.8 DMAMUX2 请求发生器通道 x 配置寄存器 (DMAMUX2_RGxCR)

DMAMUX2 request generator channel x configuration register

偏移地址：0x100 + 0x04 * (x-0)（x = 0 到 7）

复位值：0x0000 0000

根据所分配的目标 DMAMUX 请求线复用器通道 y 的安全模式，并且考虑到 DMAMUX 请求发生器 x 通道输出通过 DMAMUX 请求线通道的 y 通道来选择，该寄存器应通过非安全写入或安全写入操作来写入（请参见 DMAMUX2_CyCR.DMAREQ_ID[7:0] 和 DMAMUX 映射实现部分）。

根据所分配的目标 DMAMUX 请求线复用器通道 y 的特权模式，并且考虑到 DMAMUX 请求发生器 x 通道输出通过 DMAMUX 请求线通道的 y 通道来选择，该寄存器应通过非特权写入或特权写入操作来写入（请参见 DMAMUX2_CyCR.DMAREQ_ID[7:0] 和 DMAMXUX 映射实现部分）。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	GNBREQ[4:0]					GPOL[1:0]		GE
								rw					rw		rw

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

位 31:24 保留，必须保持复位值。

位 23:19 **GNBREQ[4:0]**：要生成的 DMA 请求数（减 1）(Number of DMA requests to be generated (minus 1))

定义触发事件后生成的 DMA 请求的数量。生成的 DMA 请求的实际数量为 GNBREQ+1。

注： 只有在 GE 位禁止时才能写入此字段。

位 18:17 **GPOL[1:0]**：DMA 请求发生器触发极性 (DMA request generator trigger polarity)

定义所选触发输入的边沿极性

00：无事件。即，无触发检测和生成。

01：上升沿

10：下降沿

11：上升沿和下降沿

位 16 **GE**：DMA 请求发生器通道 x 使能 (DMA request generator channel x enable)

0：禁止 DMA 请求发生器通道 x

1：使能 DMA 请求发生器通道 x

位 15:9 保留，必须保持复位值。

位 8 **OIE**：触发溢出中断使能 (Trigger overrun interrupt enable)

0：禁止在出现触发溢出事件时产生中断。

1：使能在出现触发溢出事件时产生中断。

位 7:5 保留，必须保持复位值。

位 4:0 **SIG_ID[4:0]**：信号标识 (Signal identification)

选择用于 DMA 请求发生器的通道 x 的 DMA 请求触发输入

17.6.9 DMAMUX1 请求发生器中断状态寄存器 (DMAMUX1_RGSR)

DMAMUX1 request generator interrupt status register

偏移地址：0x140

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OF7	OF6	OF5	OF4	OF3	OF2	OF1	OF0
								r	r	r	r	r	r	r	r

位 31:8 保留，必须保持复位值。

位 7:0 **OF[7:0]**：触发溢出事件标志 (Trigger overrun event flag)
当 DMA 请求发生器通道 x 上发生触发事件且 DMA 请求发生器计数器的值小于 GNBREQ 时，该标志将置 1。
该标志的清零方式是向 DMAMUX_RGCFR 寄存器中的相应 COFx 位写入 1。

17.6.10 DMAMUX2 请求发生器中断状态寄存器 (DMAMUX2_RGSR)

DMAMUX2 request generator interrupt status register

偏移地址：0x140

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OF7	OF6	OF5	OF4	OF3	OF2	OF1	OF0
								r	r	r	r	r	r	r	r

位 31:8 保留，必须保持复位值。

位 7:0 **OF[7:0]**：触发溢出事件标志 (Trigger overrun event flag)
当 DMA 请求发生器通道 x 上发生触发事件且 DMA 请求发生器计数器的值小于 GNBREQ 时，该标志将置 1。
该标志的清零方式是向 DMAMUX2_RGCFR 寄存器中的相应 COFx 位写入 1。

17.6.11 DMAMUX1 请求发生器中断清除标志寄存器 (DMAMUX1_RGCFR)

DMAMUX1 request generator interrupt clear flag register

偏移地址：0x144

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	COF7	COF6	COF5	COF4	COF3	COF2	COF1	COF0
								w	w	w	w	w	w	w	w

位 31:8 保留，必须保持复位值。

位 7:0 **COF[7:0]**：清除触发溢出事件标志 (Clear trigger overrun event flag)
将 1 写入每个位时，DMAMUX_RGSR 寄存器中相应的溢出标志 OFx 将清零。

17.6.12 DMAMUX2 请求发生器中断清除标志寄存器 (DMAMUX2_RGCFR)

DMAMUX2 request generator interrupt clear flag register

偏移地址：0x144

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	COF7	COF6	COF5	COF4	COF3	COF2	COF1	COF0
								r	r	r	r	r	r	r	r

位 31:8 保留，必须保持复位值。

位 7:0 **COF[7:0]**：清除触发溢出事件标志 (Clear trigger overrun event flag)
将 1 写入每个位时，DMAMUX2_RGSR 寄存器中相应的溢出标志 OFx 将清零。

17.6.13 DMAMUX 寄存器映射

下表汇总了 DMAMUX 寄存器和复位值。有关 DMAMUX 寄存器的基本地址，请参见寄存器边界地址表。

表 117. DMAMUX 寄存器映射和复位值

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x000	DMAMUX_C0CR	Res.	Res.	Res.	SYNC_ID[4:0]					NBREQ[4:0]					SPOL		SE	Res.	Res.	Res.	Res.	Res.	Res.	EGE	SOIE	DMAREQ_ID[7:0]							
	Reset value				0	0	0	0	0	0	0	0	0	0	0	0	0							0	0	0	0	0	0	0	0	0	0
0x004	DMAMUX_C1CR	Res.	Res.	Res.	SYNC_ID[4:0]					NBREQ[4:0]					SPOL		SE	Res.	Res.	Res.	Res.	Res.	Res.	EGE	SOIE	DMAREQ_ID[7:0]							
	Reset value				0	0	0	0	0	0	0	0	0	0	0	0	0							0	0	0	0	0	0	0	0	0	0
0x008	DMAMUX_C2CR	Res.	Res.	Res.	SYNC_ID[4:0]					NBREQ[4:0]					SPOL		SE	Res.	Res.	Res.	Res.	Res.	Res.	EGE	SOIE	DMAREQ_ID[7:0]							
	Reset value				0	0	0	0	0	0	0	0	0	0	0	0	0							0	0	0	0	0	0	0	0	0	0
0x00C	DMAMUX_C3CR	Res.	Res.	Res.	SYNC_ID[4:0]					NBREQ[4:0]					SPOL		SE	Res.	Res.	Res.	Res.	Res.	Res.	EGE	SOIE	DMAREQ_ID[7:0]							
	Reset value				0	0	0	0	0	0	0	0	0	0	0	0	0							0	0	0	0	0	0	0	0	0	0
0x010	DMAMUX_C4CR	Res.	Res.	Res.	SYNC_ID[4:0]					NBREQ[4:0]					SPOL		SE	Res.	Res.	Res.	Res.	Res.	Res.	EGE	SOIE	DMAREQ_ID[7:0]							
	Reset value				0	0	0	0	0	0	0	0	0	0	0	0	0							0	0	0	0	0	0	0	0	0	0
0x014	DMAMUX_C5CR	Res.	Res.	Res.	SYNC_ID[4:0]					NBREQ[4:0]					SPOL		SE	Res.	Res.	Res.	Res.	Res.	Res.	EGE	SOIE	DMAREQ_ID[7:0]							
	Reset value				0	0	0	0	0	0	0	0	0	0	0	0	0							0	0	0	0	0	0	0	0	0	0
0x018	DMAMUX_C6CR	Res.	Res.	Res.	SYNC_ID[4:0]					NBREQ[4:0]					SPOL		SE	Res.	Res.	Res.	Res.	Res.	Res.	EGE	SOIE	DMAREQ_ID[7:0]							
	Reset value				0	0	0	0	0	0	0	0	0	0	0	0	0							0	0	0	0	0	0	0	0	0	0
0x01C	DMAMUX_C7CR	Res.	Res.	Res.	SYNC_ID[4:0]					NBREQ[4:0]					SPOL		SE	Res.	Res.	Res.	Res.	Res.	Res.	EGE	SOIE	DMAREQ_ID[7:0]							
	Reset value				0	0	0	0	0	0	0	0	0	0	0	0	0							0	0	0	0	0	0	0	0	0	0
0x020(1)	DMAMUX_C8CR	Res.	Res.	Res.	SYNC_ID[4:0]					NBREQ[4:0]					SPOL		SE	Res.	Res.	Res.	Res.	Res.	Res.	EGE	SOIE	DMAREQ_ID[7:0]							
	Reset value				0	0	0	0	0	0	0	0	0	0	0	0	0							0	0	0	0	0	0	0	0	0	0
0x024(1)	DMAMUX_C9CR	Res.	Res.	Res.	SYNC_ID[4:0]					NBREQ[4:0]					SPOL		SE	Res.	Res.	Res.	Res.	Res.	Res.	EGE	SOIE	DMAREQ_ID[7:0]							
	Reset value				0	0	0	0	0	0	0	0	0	0	0	0	0							0	0	0	0	0	0	0	0	0	0
0x028(1)	DMAMUX_C10CR	Res.	Res.	Res.	SYNC_ID[4:0]					NBREQ[4:0]					SPOL		SE	Res.	Res.	Res.	Res.	Res.	Res.	EGE	SOIE	DMAREQ_ID[7:0]							
	Reset value				0	0	0	0	0	0	0	0	0	0	0	0	0							0	0	0	0	0	0	0	0	0	0
0x02C(1)	DMAMUX_C11CR	Res.	Res.	Res.	SYNC_ID[4:0]					NBREQ[4:0]					SPOL		SE	Res.	Res.	Res.	Res.	Res.	Res.	EGE	SOIE	DMAREQ_ID[7:0]							
	Reset value				0	0	0	0	0	0	0	0	0	0	0	0	0							0	0	0	0	0	0	0	0	0	0
0x030(1)	DMAMUX_C12CR	Res.	Res.	Res.	SYNC_ID[4:0]					NBREQ[4:0]					SPOL		SE	Res.	Res.	Res.	Res.	Res.	Res.	EGE	SOIE	DMAREQ_ID[7:0]							
	Reset value				0	0	0	0	0	0	0	0	0	0	0	0	0							0	0	0	0	0	0	0	0	0	0
0x034(1)	DMAMUX_C13CR	Res.	Res.	Res.	SYNC_ID[4:0]					NBREQ[4:0]					SPOL		SE	Res.	Res.	Res.	Res.	Res.	Res.	EGE	SOIE	DMAREQ_ID[7:0]							
	Reset value				0	0	0	0	0	0	0	0	0	0	0	0	0							0	0	0	0	0	0	0	0	0	0
0x038(1)	DMAMUX_C14CR	Res.	Res.	Res.	SYNC_ID[4:0]					NBREQ[4:0]					SPOL		SE	Res.	Res.	Res.	Res.	Res.	Res.	EGE	SOIE	DMAREQ_ID[7:0]							
	Reset value				0	0	0	0	0	0	0	0	0	0	0	0	0							0	0	0	0	0	0	0	0	0	0
0x03C(1)	DMAMUX_C15CR	Res.	Res.	Res.	SYNC_ID[4:0]					NBREQ[4:0]					SPOL		SE	Res.	Res.	Res.	Res.	Res.	Res.	EGE	SOIE	DMAREQ_ID[7:0]							
	Reset value				0	0	0	0	0	0	0	0	0	0	0	0	0							0	0	0	0	0	0	0	0	0	0
0x040 - 0x07C	Reserved	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RES.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

表 117. DMAMUX 寄存器映射和复位值（续）

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x080	DMAMUX_CSR(2)	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SOF15	SOF14	SOF13	SOF12	SOF11	SOF10	SOF9	SOF8	SOF7	SOF6	SOF5	SOF4	SOF3	SOF2	SOF1	SOF0
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x084	DMAMUX_CFR(2)	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CSOF15	CSOF14	CSOF13	CSOF12	CSOF11	CSOF10	CSOF9	CSOF8	CSOF7	CSOF6	CSOF5	CSOF4	CSOF3	CSOF2	CSOF1	CSOF0
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x088 - 0x0FC	Reserved	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
0x100	DMAMUX_RG0CR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	GNBREQ[4:0]					GPOL	GE	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OIE	Res.	Res.	Res.	SIG_ID[4:0]				
	Reset value									0	0	0	0	0	0	0	0								0				0	0	0	0	0
0x104	DMAMUX_RG1CR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	GNBREQ[4:0]					GPOL	GE	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OIE	Res.	Res.	Res.	SIG_ID[4:0]				
	Reset value									0	0	0	0	0	0	0	0								0				0	0	0	0	0
0x108	DMAMUX_RG2CR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	GNBREQ[4:0]					GPOL	GE	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OIE	Res.	Res.	Res.	SIG_ID[4:0]				
	Reset value									0	0	0	0	0	0	0	0								0				0	0	0	0	0
0x10C	DMAMUX_RG3CR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	GNBREQ[4:0]					GPOL	GE	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OIE	Res.	Res.	Res.	SIG_ID[4:0]				
	Reset value									0	0	0	0	0	0	0	0								0				0	0	0	0	0
0x140	DMAMUX_RGSR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OF15	OF14	OF13	OF12	OF11	OF10	OF9	OF8	OF7	OF6	OF5	OF4	OF3	OF2	OF1	OF0
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x144	DMAMUX_RGCFR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	COF15	COF14	COF13	COF12	COF11	COF10	COF9	COF8	COF7	COF6	COF5	COF4	COF3	COF2	COF1	COF0
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x140	DMAMUX_RGSR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OF7	OF6	OF5	OF4	OF3	OF2	OF1	OF0
	Reset value																									0	0	0	0	0	0	0	0
0x144	DMAMUX_RGCFR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	COF7	COF6	COF5	COF4	COF3	COF2	COF1	COF0
	Reset value																									0	0	0	0	0	0	0	0
0x148 - 0x3FC	Reserved	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

1. 仅适用于 DMAMUX1。对于 DMAMUX2，该字保留。
2. 对于 DMAMUX2，位 15:8 保留。

18 Chrom-Art Accelerator™ 控制器 (DMA2D)

18.1 DMA2D 简介

Chrom-Art Accelerator™ (DMA2D) 是专用于图像处理的专业 DMA。它可以执行下列操作：

- 用特定颜色填充目标图像的一部分或全部
- 将源图像的一部分（或全部）复制到目标图像的一部分（或全部）中
- 通过像素格式转换将源图像的一部分或全部复制到目标图像的一部分或全部中
- 将像素格式不同的两个源图像部分和/或全部混合，再将结果复制到颜色格式不同的部分或整个目标图像中。

在索引颜色模式或直接颜色模式下，支持所有经典颜色编码方案，并支持每像素 4 位到最高 32 位（其中包括基于模块的 YCbCr，用于处理 JPEG 解码器输出）。DMA2D 自身具有专门的 CLUT（颜色查找表）存储器。

18.2 DMA2D 的主要特性

DMA2D 的主要特性有：

- 采用单 AXI 主设备总线架构
- AHB 从设备编程接口支持 8/16/32 位访问（32 位的 CLUT 访问除外）
- 用户可编程工作区大小
- 用户可编程源区域和目标区域的偏移
- 用户可编程整个存储空间的源地址和目标地址
- 最多支持 2 个源的混合操作
- Alpha 值可修改（源值、固定值或调制的值）
- 用户可编程源颜色格式和目标的颜色格式
- 采用间接或直接颜色编码时，支持多达 11 种颜色格式，且支持每像素 4 位到最高 32 位
- 支持基于块 (8x8) 的 YCbCr，色度子采样系数包括 4:4:4、4:2:2 和 4:2:0 三种
- 间接颜色模式下使用 2 个内部存储器存储 CLUT
- 通过 CPU 自动加载 CLUT 或对 CLUT 进行编程
- 用户可编程 CLUT 大小
- 使用内部定时器控制 AXI 带宽
- 支持 4 种工作模式：寄存器到存储器、存储器到存储器、存储器到存储器且支持像素格式转换和存储器到存储器且支持像素格式转换和混合
- 可使用固定颜色进行区域填充
- 可从一个区域复制到另一个区域
- 在源图像和目标图像之间进行复制时进行像素格式转换
- 支持从颜色格式不同的两幅源图像复制并混合
- 可中止并挂起 DMA2D 操作
- 支持在传输用户可编程的目标行时生成水印中断
- 支持发生总线错误或访问冲突时生成中断
- 支持处理完成时生成中断

18.3 DMA2D 功能说明

18.3.1 概述

DMA2D 控制器执行直接存储器传输。作为一个 AXI 主设备，它可以控制 AXI 总线矩阵来启动 AXI 事务。

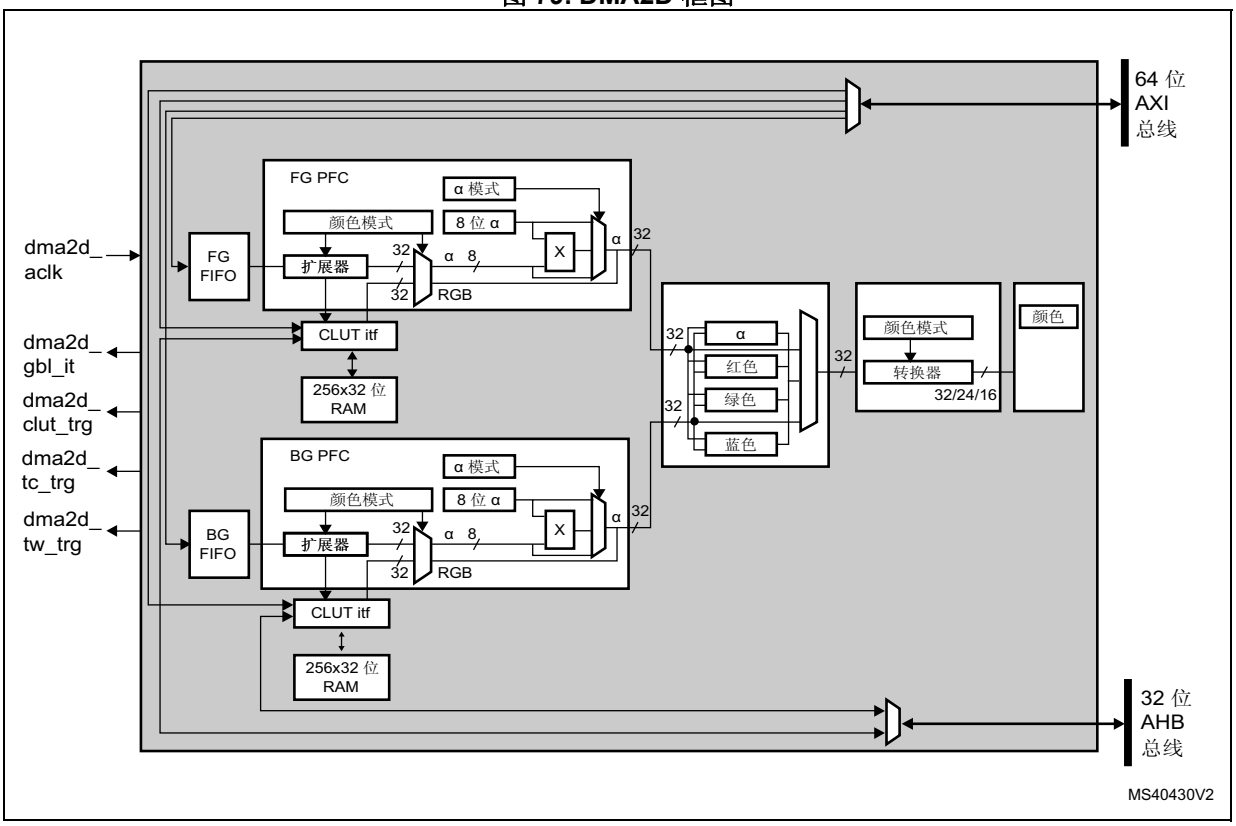
DMA2D 可在以下四种模式下工作：

- 寄存器到存储器
- 存储器到存储器
- 存储器到存储器并执行像素格式转换
- 存储器到存储器并执行像素格式转换和混合

AHB 从设备端口用于编程 DMA2D 控制器。

DMA2D 的框图如*图 79：DMA2D 框图*所示。

图 79. DMA2D 框图



18.4 DMA2D 引脚和内部信号

表 118 列出了 DMA2D 内部信号。

表 118. DMA2D 内部输入/输出信号

信号名称	信号类型	说明
dma2d_aclk	数字输入	32 位 AXI 总线时钟
dma2d_gbl_it	数字输出	DMA2D 全局中断
dma2d_clut_trg	数字输出	CLUT 传输完成（到 MDMA）
dma2d_tc_trg	数字输出	传输完成（到 MDMA）
dma2d_tw_trg	数字输出	传输水印（到 MDMA）

18.4.1 DMA2D 控制

通过 DMA2D 控制寄存器 (DMA2D_CR) 配置 DMA2D 控制器，允许选择：

用户应用可以执行下列操作：

- 选择工作模式
- 使能/禁止 DMA2D 中断
- 启动/挂起/中止进行中的数据传输

18.4.2 DMA2D 前景层 FIFO 和背景层 FIFO

DMA2D 前景层 (FG) FG FIFO 和背景层 (BG) FIFO 获取要复制和/或处理的输入数据。

这些 FIFO 根据相应像素格式转换器 (PFC) 中定义的颜色格式获取像素。

通过如下一组寄存器对它们进行编程：

- DMA2D 前景层存储器地址寄存器 (DMA2D_FGMAR)
- DMA2D 前景层偏移寄存器 (DMA2D_FGOR)
- DMA2D 背景层存储器地址寄存器 (DMA2D_BGMAR)
- DMA2D 背景层偏移寄存器 (DMA2D_BGBOR)
- DMA2D 行数寄存器（行数和每行像素数）(DMA2D_NLR)

DMA2D 在寄存器到存储器模式下工作时，不激活任何 FIFO。

DMA2D 在存储器到存储器模式下工作时（无像素格式转换和混合操作），仅激活 FG FIFO，并将其用作缓冲区。

DMA2D 在存储器到存储器模式下工作时并支持像素格式转换时（无混合操作），不会激活 BG FIFO。

18.4.3 DMA2D 前景层和背景层像素格式转换器 (PFC)

DMA2D 前景层和背景层像素格式转换器 (PFC) 执行像素格式转换，以生成每像素 32 位的值。PFC 还能够修改 alpha 通道。

转换器在第一阶段转换颜色格式。前景层像素和背景层像素的原始颜色格式分别通过 DMA2D_FGPFCCR 和 DMA2D_BGPFCCR 的 CM[3:0] 位来配置。

*表 119：输入时支持的颜色模式*给出了支持的输入格式。

表 119. 输入时支持的颜色模式

CM[3:0]	颜色模式
0000	ARGB8888
0001	RGB888
0010	RGB565
0011	ARGB1555
0100	ARGB4444
0101	L8
0110	AL44
0111	AL88
1000	L4
1001	A8
1010	A4
1011	YCbCr（仅用于前景层）

颜色格式的编码方式如下：

- Alpha 值字段：透明
  0xFF 值对应不透明像素，0x00 对应透明像素。
- R 字段代表红色
- G 字段代表绿色
- B 字段代表蓝色
- L 字段：亮度
  该字段是 CLUT 的索引，用于检索三个/四个 RGB/ARGB 分量。

如果原始格式为直接颜色模式，则通过将 MSB 复制到 LSB 扩展为每通道 8 位。这可以确保转换具有良好的线性。

如果原始格式不包括 alpha 通道，则会自动将 alpha 值设为 0xFF（不透明）。

如果原始格式为间接颜色模式，则需要使用 CLUT，并且每个像素格式转换器与一个 256 个 32 位 条目的 CLUT 相关联。

对于特定的 alpha 模式 A4 和 A8，既不存储颜色信息，也不编制索引。用于生成图像的颜色是固定的，并且在 DMA2D_FGCOLR 寄存器中定义前景层像素的颜色，在 DMA2D_BGCOLR 寄存器中定义背景层像素的颜色。

系统存储器中的字段顺序如*表 120：存储器中的数据顺序*所示。

表 120. 存储器中的数据顺序

颜色模式	@ + 3	@ + 2	@ + 1	@ + 0
ARGB8888	A_0[7:0]	R_0[7:0]	G_0[7:0]	B_0[7:0]
RGB888	B_1[7:0]	R_0[7:0]	G_0[7:0]	B_0[7:0]
	G_2[7:0]	B_2[7:0]	R_1[7:0]	G_1[7:0]
	R_3[7:0]	G_3[7:0]	B_3[7:0]	R_2[7:0]
RGB565	R_1[4:0]G_1[5:3]	G_1[2:0]B_1[4:0]	R_0[4:0]G_0[5:3]	G_0[2:0]B_0[4:0]
ARGB1555	A_1[0]R_1[4:0]G_1[4:3]	G_1[2:0]B_1[4:0]	A_0[0]R_0[4:0]G_0[4:3]	G_0[2:0]B_0[4:0]
ARGB4444	A_1[3:0]R_1[3:0]	G_1[3:0]B_1[3:0]	A_0[3:0]R_0[3:0]	G_0[3:0]B_0[3:0]
L8	L_3[7:0]	L_2[7:0]	L_1[7:0]	L_0[7:0]
AL44	A_3[3:0]L_3[3:0]	A_2[3:0]L_2[3:0]	A_1[3:0]L_1[3:0]	A_0[3:0]L_0[3:0]
AL88	A_1[7:0]	L_1[7:0]	A_0[7:0]	L_0[7:0]
L4	L_7[3:0]L_6[3:0]	L_5[3:0]L_4[3:0]	L_3[3:0]L_2[3:0]	L_1[3:0]L_0[3:0]
A8	A_3[7:0]	A_2[7:0]	A_1[7:0]	A_0[7:0]
A4	A_7[3:0]A_6[3:0]	A_5[3:0]A_4[3:0]	A_3[3:0]A_2[3:0]	A_1[3:0]A_0[3:0]

通过 ARGB8888 模式支持按 32 位对齐 24 位 RGB888。

生成 32 位值后，即可根据 DMA2D_FGPFCCR/DMA2D_BGPFCCR 寄存器的 AM[1:0] 字段修改 alpha 通道，如*表 121：Alpha 模式配置*所示。

Alpha 通道可以：

- 保持不变（不做修改）
- 替换为 DMA2D_FGPFCCR/DMA2D_BGPFCCR 的 ALPHA[7:0] 值
- 或替换为原始 alpha 值与 DMA2D_FGPFCCR/DMA2D_BGPFCCR 的 ALPHA [7:0] 值除以 255 所得商的乘积。

表 121. Alpha 模式配置

AM[1:0]	Alpha 模式
00	不做修改
01	替换为 DMA2D_xxPFCCR 中的值
10	替换为原始值与 DMA2D_xxPFCCR 中的值/255 所得商的乘积
11	保留

注： *要支持备用格式，可通过将 DMA2D_FGPFCCR/DMA2D_BGPFCCR 寄存器的 AI 位置 1 将传入的 alpha 值取反。这同样适用于 DMA2D_FGPFCCR/DMA2D_BGPFCCR 和 CLUT 中存储的 Alpha 值。*

此外，还可通过将 DMA2D_FGPFCCR/DMA2D_BGPFCCR 寄存器的 RBS 位置 1 来交换 R 字段和 B 字段。这同样适用于 CLUT 和 DMA2D_FGCOLR/DMA2D_BGCOLR 寄存器中使用的 RGB 顺序。

18.4.4 DMA2D 前景层 FIFO 和背景层 CLUT 接口

CLUT 接口可管理对 CLUT 存储器的访问以及 CLUT 的自动加载。

支持如下三种访问：

- PFC 在像素格式转换期间读取 CLUT
- CPU 对 CLUT 进行数据的读取或写入时，通过 AHB 从设备端口访问 CLUT
- 执行自动加载 CLUT 时，通过 AXI 主设备端口进行 CLUT 写入

可通过两种不同方法执行 CLUT 存储器加载：

- 自动加载

  加载 CLUT 时应遵守以下顺序：

  a) 将 CLUT 地址编程到 DMA2D_FGCMAR 寄存器（前景层 CLUT）或 DMA2D_BGCMAR 寄存器（背景层 CLUT）。

  b) 将 CLUT 大小编程到 DMA2D_FGPFCCR 寄存器（前景层 CLUT）或 MA2D_BGPFCCR 寄存器（背景层 CLUT）的 S[7:0] 字段。

  c) 将 MA2D_FGPFCCR 寄存器（前景层 CLUT）或 DMA2D_BGPFCCR 寄存器（背景层 CLUT）的 START 位置 1 以启动传输。自动加载过程期间，不可通过 CPU 访问 CLUT。如果出现冲突，若 DMA2D_CR 中 CAEIE 被置 1，则发生 CLUT 访问错误中断。

- 手动加载

  应用程序必须通过 DMA2D AHB 从设备端口手动编程本地 CLUT 存储器映射到的 CLUT。前景层 CLUT 从偏移地址 0x0400 开始，背景层 CLUT 从偏移地址 0x0800 开始。

CLUT 格式可以是 24 位或 32 位。通过 DMA2D_FGPFCCR 寄存器（前景层 CLUT）或 DMA2D_BGPFCCR 寄存器（背景层 CLUT）的 CCM 位配置格式，如*表 122：支持的 CLUT 颜色模式*所示。

表 122. 支持的 CLUT 颜色模式

CCM	CLUT 颜色模式
0	32 位 ARGB8888
1	24 位 RGB888

*表 123：存储器中的 CLUT 数据顺序*给出了系统存储器中 CLUT 数据的组织方式。

表 123. 存储器中的 CLUT 数据顺序

CLUT 颜色模式	@ + 3	@ + 2	@ + 1	@ + 0
ARGB8888	A_0[7:0]	R_0[7:0]	G_0[7:0]	B_0[7:0]
RGB888	B_1[7:0]	R_0[7:0]	G_0[7:0]	B_0[7:0]
	G_2[7:0]	B_2[7:0]	R_1[7:0]	G_1[7:0]
	R_3[7:0]	G_3[7:0]	B_3[7:0]	R_2[7:0]

18.4.5 DMA2D 混合器

DMA2D 混合器成对混合源像素以计算结果像素。

混合将按以下公式执行：

$$\text{其中 } \alpha_{Mult} = \frac{\alpha_{FG} \cdot \alpha_{BG}}{255}$$

$$\alpha_{OUT} = \alpha_{FG} + \alpha_{BG} - \alpha_{Mult}$$

$$C_{OUT} = \frac{C_{FG} \cdot \alpha_{FG} + C_{BG} \cdot \alpha_{BG} - C_{BG} \cdot \alpha_{Mult}}{\alpha_{OUT}} \quad \text{其中 C = R 或 G 或 B}$$

商将向下取整

混合器不需要任何配置寄存器。是否使用混合器取决于 DMA2D_CR 寄存器的 MODE[1:0] 字段中定义的 DMA2D 工作模式。

18.4.6 DMA2D 输出 PFC

输出 PFC 将像素格式从 32 位转换为指定的输出格式，输出格式在 DMA2D 输出像素格式转换器配置寄存器 (DMA2D_OPFCCR) 的 CM[2:0] 字段中定义。

*表 124：输出时支持的颜色模式*给出了支持的输出格式。

表 124. 输出时支持的颜色模式

CM[2:0]	颜色模式
000	ARGB8888
001	RGB888
010	RGB565
011	ARGB1555
100	ARGB4444

注： *要支持备用格式，可通过将 DMA2D_OPFCCR 寄存器的 AI 位置 1 将计算的 alpha 值取反。这同样适用于 DMA2D_OCOLR 中使用的 Alpha 值。*

此外，还可通过将 DMA2D_OPFCCR 寄存器的 RBS 位置 1 来交换 R 字段和 B 字段。这同样适用于 DMA2D_OCOLR 中使用的 RGB 顺序。

18.4.7 DMA2D 输出 FIFO

输出 FIFO 根据输出 PFC 中定义的颜色格式对像素进行编程。

通过如下一组寄存器定义目标区域：

- DMA2D 输出存储器地址寄存器 (DMA2D_OMAR)
- DMA2D 输出偏移寄存器 (DMA2D_OOR)
- DMA2D 行数寄存器（行数和每行像素数）(DMA2D_NLR)

如果 DMA2D 在寄存器到存储器模式下工作，则配置的输出矩形将以 DMA2D 输出颜色寄存器 (DMA2D_OCOLR) 中指定的颜色填充，该寄存器中包含固定的 32 位、24 位或 16 位值。通过 DMA2D_OPFCCR 寄存器的 CM[2:0] 字段选择格式。

将按照*表 125：存储器中的数据顺序*中定义的顺序在存储器中存储数据

表 125. 存储器中的数据顺序

颜色模式	@ + 3	@ + 2	@ + 1	@ + 0
ARGB8888	A_0[7:0]	R_0[7:0]	G_0[7:0]	B_0[7:0]
RGB888	B_1[7:0]	R_0[7:0]	G_0[7:0]	B_0[7:0]
	G_2[7:0]	B_2[7:0]	R_1[7:0]	G_1[7:0]
	R_3[7:0]	G_3[7:0]	B_3[7:0]	R_2[7:0]
RGB565	R_1[4:0]G_1[5:3]	G_1[2:0]B_1[4:0]	R_0[4:0]G_0[5:3]	G_0[2:0]B_0[4:0]
ARGB1555	A_1[0]R_1[4:0]G_1[4:3]	G_1[2:0]B_1[4:0]	A_0[0]R_0[4:0]G_0[4:3]	G_0[2:0]B_0[4:0]
ARGB4444	A_1[3:0]R_1[3:0]	G_1[3:0]B_1[3:0]	A_0[3:0]R_0[3:0]	G_0[3:0]B_0[3:0]

通过 ARGB8888 模式支持按 32 位对齐 RGB888。

18.4.8 DMA2D AXI 主设备端口定时器

AXI 主设备端口内嵌一个 8 位定时器，以便可选择限制交叉开关矩阵的带宽。

此定时器由 AXI 时钟驱动，对两个连续访问之间的死区进行计数。这样可限制带宽的使用。

通过 AXI 主设备端口定时器配置寄存器 (DMA2D_AMPTCR) 配置定时器使能和死区值。

18.4.9 DMA2D 事务

DMA2D 事务由给定数目的数据传输序列组成。可通过软件对数据数目和宽度进行编程。

每个 DMA2D 数据传输最多需要 4 个步骤：

1. 从 DMA2D_FGMAR 寄存器寻址的存储单元加载数据并按照 DMA2D_FGCR 中的定义进行像素格式转换。
2. 从 DMA2D_BGMAR 寄存器寻址的存储单元加载数据并按照 DMA2D_BGCR 中的定义进行像素格式转换。
3. 根据对 alpha 值进行 PFC 操作所得到的 alpha 通道，将所有检索到的像素混合。
4. 根据 DMA2D_OCR 寄存器对合成像素进行像素格式转换，然后将数据编程到通过 DMA2D_OMAR 寄存器寻址的存储单元。

18.4.10 DMA2D 配置

源和目标数据传输在整个 4 GB 区域（地址范围在 0x0000 0000 和 0xFFFF FFFF 之间）都可以寻址外设和存储器。

DMA2D 可在以下四种模式下工作，通过 DMA2D_CR 寄存器的 MODE[1:0] 位选择工作模式：

- 寄存器到存储器
- 存储器到存储器
- 存储器到存储器并执行 PFC
- 存储器到存储器并执行 PFC 和混合

寄存器到存储器

寄存器到存储器模式用于以预定义颜色填充用户自定义区域。

颜色格式在 DMA2D_OPFCCR 中设置。

DMA2D 不从任何源获取数据。它只将 DMA2D_OCOLR 寄存器中定义的颜色写入通过 DMA2D_OMAR 寻址以及 DMA2D_NLR 和 DMA2D_OOR 定义的区域。

存储器到存储器

在存储器到存储器模式下，DMA2D 不执行任何图形数据转换。前景层输入 FIFO 充当缓冲区，数据从 DMA2D_FGMAR 中定义的源存储单元传输到 DMA2D_OMAR 寻址的目标存储单元。

DMA2D_FGPFCCR 寄存器的 CM[3:0] 位中编程的颜色模式决定输入和输出的每像素位数。

对于要传输的区域大小，源区域大小由 DMA2D_NLR 和 DMA2D_FGOR 寄存器定义，目标区域大小则由 DMA2D_NLR 和 DMA2D_OOR 寄存器定义。

存储器到存储器并执行 PFC

此模式下，DMA2D 对源数据执行像素格式转换并将结果存储在目标存储单元。

对于要传输的区域大小，源区域大小由 DMA2D_NLR 和 DMA2D_FGOR 寄存器定义，目标区域大小则由 DMA2D_NLR 和 DMA2D_OOR 寄存器定义。

从 DMA2D_FGMAR 寄存器定义的位置获取数据，并由前景层 PFC 进行处理。原始像素格式通过 DMA2D_FGPFCCR 寄存器配置。

如果原始像素格式是直接颜色模式，则所有颜色通道都扩展到 8 位。

如果像素格式是间接颜色模式，则必须将相关 CLUT 加载到 CLUT 存储器中。

CLUT 加载可按如下顺序自动完成：

1. 在 DMA2D_FGCMAR 中设置 CLUT 地址。
2. 在 DMA2D_FGPFCCR 寄存器的 CS[7:0] 位设置 CLUT 大小。
3. 在 DMA2D_FGPFCCR 寄存器的 CCM 位设置 CLUT 格式（24 或 32 位）。
4. 将 DMA2D_FGPFCCR 寄存器的 START 位置 1 启动 CLUT 加载。

CLUT 加载完成时，DMA2D_ISR 寄存器将置位 CTCIF 标志；如果 DMA2D_CR 中的 CTCIE 位置 1，还将产生中断。CLUT 自动加载过程无法与传统的 DMA2D 传输同时进行。

CLUT 还可由 CPU 填充，或由任意其它主设备通过 AHB 端口填充。在 DMA2D 传输进行期间和使用 CLUT（间接颜色格式）时无法访问 CLUT。

在颜色转换执行期间，可根据 DMA2D_FGPFCCR 寄存器中编程的值添加或更改 alpha 值。如果原始图像没有 alpha 通道，则会自动添加一个默认的 alpha 值 0xFF 以获得完全不透明的像素。可根据 DMA2D_FGPFCCR 寄存器的 AM[1:0] 位修改 alpha 值：

- 保持不变。
- 替换为 DMA2D_FGPFCCR 寄存器的 ALPHA[7:0] 值中定义的值。
- 替换为原始值与 DMA2D_FGPFCCR 寄存器的 ALPHA[7:0] 值除以 255 所得商的乘积。

结果得到的 32 位数据由 OUT PFC 编码成 DMA2D_OPFCCR 寄存器的 CM[2:0] 字段所指定的格式。输出像素格式不可是间接模式，原因是输出时不支持 CLUT 生成过程。

数据经处理后，将写入 DMA2D_OMAR 寻址的目标存储单元。

存储器到存储器并执行 PFC 和混合

此模式下，将在前景层 FIFO 和背景层 FIFO（分别在 DMA2D_FGMAR 和 DMA2D_BGMAR 中定义）获取 2 个源图像。

必须按存储器到存储器模式中所述配置两个像素格式转换器。由于这两个像素格式转换器各自独立且自身具有 CLUT 存储器，因此其配置可以不同。

在每个像素都通过相应的 PFC 转换为 32 位后，将根据以下公式进行混合：

$$\text{其中 } \alpha_{Mult} = \frac{\alpha_{FG} \cdot \alpha_{BG}}{255}$$

$$\alpha_{OUT} = \alpha_{FG} + \alpha_{BG} - \alpha_{Mult}$$

$$C_{OUT} = \frac{C_{FG}\alpha_{FG} + C_{BG}\alpha_{BG} - C_{BG}\alpha_{Mult}}{\alpha_{OUT}} \quad \text{其中 C = R 或 G 或 B}$$

商将向下取整

输出 PFC 将根据指定的输出格式对得到的 32 位像素值进行编码，并且编码数据将写入 DMA2D_OMAR 寻址的目标存储单元。

配置错误检测

DMA2D 将在每次执行传输前检查配置是否正确。开始新的传输/自动加载时，如果检测到配置错误，硬件将设置配置错误中断标志。如果 DMA2D_CR 寄存器的 CEIE 位置 1，还将产生中断。

可检测到的错误配置如下：

- 前景层 CLUT 自动加载：DMA2D_FGCMAR 的 MA 位与 DMA2D_FGPFCCR 的 CCM 位不匹配
- 背景层 CLUT 自动加载：DMA2D_BGCMAR 的 MA 位与 DMA2D_BGPFCCR 的 CCM 位不匹配
- 存储器传输（寄存器到存储器模式除外）：DMA2D_FGMAR 的 MA 位与 DMA2D_FGPFCCR 的 CM 位不匹配
- 存储器传输（寄存器到存储器模式除外）：DMA2D_FGPFCCR 中的 CM 位无效
- 存储器传输（寄存器到存储器模式除外）：DMA2D_NLR 的 PL 位为奇，而 DMA2D_FGPFCCR 的 CM 位为 A4 或 L4

- 存储器传输（寄存器到存储器模式除外）：DMA2D_FGOR 中的 LO 位为奇，而 DMA2D_FGPFCCR 的 CM 位为 A4 或 L4
- 存储器传输（仅限混合模式）：DMA2D_BGMAR 中的 MA 位与 DMA2D_BGPFCCR 的 CM 位不一致
- 存储器传输：DMA2D_BGPFCCR 的 CM 无效（仅限混合模式）
- 存储器传输（仅限混合模式）：DMA2D_NLR 的 PL 位为奇，而 DMA2D_BGPFCCR 的 CM 位为 A4 或 L4
- 存储器传输 （仅限混合模式）：DMA2D_BGOR 的 LO 位为奇，而 DMA2D_BGPFCCR 的 CM 位为 A4 或 L4
- 存储器传输（存储器到存储器模式除外）：DMA2D_OMAR 的 MA 位与 DMA2D_OPFCCR 的 CM 位不匹配
- 存储器传输（存储器到存储器模式除外）：DMA2D_OPFCCR 中的 CM 位无效
- 存储器传输：DMA2D_NLR 中的 NL 位 = 0
- 存储器传输：DMA2D_NLR 中的 PL 位 = 0
- YCbCr 格式：当 CLUT 加载开始将 DMA2D_FGPFCCR 的 START 位置 1 时
- YCbCr 格式：当选择存储器到存储器模式时
- YCbCr 格式：当选择 YCbCR4:4:4 且像素 (PL) 数与行偏移 LO 之和不是 8 个像素的倍数时
- YCbCr 格式：当选择 YCbCr4:2:2 或 YCbCr4:2:0 且像素 (PL) 数与行偏移 LO 之和不是 16 个像素的倍数时

18.4.11 支持 YCbCr

DMA2D 前景平面可支持将基于 8x8 块的 YCbCr 作为 JPEG 解码器的输出，并支持使用不同的色度子采样系数：

存储器构成遵循标准 JFIF 规则：

- 三个颜色分量中的每一个都必须按 8 位编码
- 每个分量必须按 8x8（64 字节）的块（称为 MCU）排列

根据色度子采样系数，MCU 必须按*表 126：存储器中的 MCU 顺序*所述在存储器中排列。

表 126. 存储器中的 MCU 顺序

子采样	@	@ + 64	@ + 128	@+192	@+256	@ + 320
4:4:4	Y1	CB1	CR1	Y_2	Cb_2	CR2
4:2:2	Y1	Y_2	Cb_{12}	Cr_{12}	Y_3	Y_4
4:2:0	Y1	Y_2	Y_3	Y_4	Cb_{1234}	Cr_{1234}

色度子采样系数通过 DMA2D_FGPFCCR 寄存器的 CSS 字段配置。

DMA2D 采用在 YCbCr 颜色模式下配置的前景启动后，前 2 个色度 MCU 会装入前景 CLUT。色度 MCU 装入后，DMA2D 会针对经典颜色模式执行 Y MCU 加载。

18.4.12 DMA2D 传输控制（启动、挂起、中止和完成）

配置 DMA2D 后，通过将 DMA2D_CR 寄存器的 START 位置 1 可启动传输。传输完成时，START 位自动复位且 DMA2D_ISR 寄存器置位 TCIF 标志。如果 DMA2D_CR 寄存器中的 TCIE 位置 1，还将产生中断。

用户应用程序随时都可通过将 DMA2D_CR 寄存器的 SUSP 位置 1 来挂起 DMA2D。随即可通过将 DMA2D_CR 寄存器的 ABORT 位置 1 中止事务，或通过将 DMA2D_CR 寄存器的 SUSP 位复位重新启动事务。

用户应用程序随时都可通过将 DMA2D_CR 寄存器的 ABORT 位置 1 来中止处理中的事务。在这种情况下，不会置位 TCIF 标志。

还可通过 DMA2D_FGPFCCR 和 DMA2D_BGPFCCR 寄存器自身的 START 位中止或挂起 CLUT 自动传输过程。

18.4.13 水印

可对水印编程，以在指定行的最后一个像素写入目标存储区域时产生中断。

行号在 DMA2D_LWR 寄存器的 LW[15:0] 字段中定义。

在该行的最后一个像素传输完成时，DMA2D_ISR 将置位 TWIF 标志；在 DMA2D_CR 的 TWIE 位置 1 的情况下，还将产生中断。

18.4.14 错误管理

可触发两种错误：

- AXI 主设备端口错误，通过 DMA2D_ISR 寄存器的 TEIF 标志指示。
- CLUT 访问引发的冲突（CPU 在 CLUT 加载或 DMA2D 传输执行期间尝试访问 CLUT），通过 DMA2D_ISR 寄存器的 CAEIF 标志指示。

这两个标志都与其在 DMA2D_CR 寄存器中的中断使能标志（控制在需要时产生中断的 TEIE 和 CAEIE）相关。

18.4.15 AXI 死区

要限制 AXI 带宽的使用，可在两个连续的 AXI 访问之间编程一个死区。

可通过将 DMA2D_AMTCR 寄存器中的 EN 位置 1 使能此特性。

死区值存储在 DMA2D_AMTCR 寄存器的 DT[7:0] 字段中。该值表示 AXI 总线上两个连续事务之间允许占用的最少周期数。

在 DMA2D 运行过程中对死区值所做的更新将在下一次 AXI 传输时生效。

18.5 DMA2D 中断

发生如下事件时可生成中断：

- 配置错误
- CLUT 传输完成
- CLUT 访问错误
- 到达传输水印
- 传输完成
- 传输错误

可以使用单独的中断使能位以提高灵活性。

表 127. DMA2D 中断请求

中断事件	事件标志	使能控制位
配置错误	CEIF	CEIE
CLUT 传输完成	CTCIF	CTCIE
CLUT 访问错误	CAEIF	CAEIE
传输水印	TWF	TWIE
传输完成	TCIF	TCIE
传输错误	TEIF	TEIE

18.6 DMA2D 寄存器

18.6.1 DMA2D 控制寄存器 (DMA2D_CR)

DMA2D control register

偏移地址：0x0000

复位值：0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	CEIE	CTCIE	CAEIE	TWIE	TCIE	TEIE	Res.	Res.	Res.	Res.	Res.	ABORT	SUSP	START
		rw	rw	rw	rw	rw	rw						rs	rw	rs

位 31:18 保留，必须保持复位值

位 17:16 **MODE**：DMA2D 模式 (DMA2D mode)
此位由软件置 1 和清零。无法在传输进行时修改此位。
00：存储器到存储器（仅限 FG 获取）
01：存储器到存储器并执行 PFC（仅限 FG PFC 激活时的 FG 获取）
10：存储器到存储器并执行混合（执行 PFC 和混合时的 FG 和 BG 获取）
11：寄存器到存储器（无 FG 和 BG，仅输出阶段激活）

位 15:14 保留，必须保持复位值

位 13 **CEIE**：配置错误中断使能 (Configuration Error Interrupt Enable)
此位由软件置 1 和清零。
0：CE 中断禁止
1：CE 中断使能

位 12 **CTCIE**：CLUT 传输完成中断使能 (CLUT transfer complete interrupt enable)
此位由软件置 1 和清零。
0：CTC 中断禁止
1：CTC 中断使能

位 11 **CAEIE**：CLUT 访问错误中断使能 (CLUT access error interrupt enable)
此位由软件置 1 和清零。
0：CAE 中断禁止
1：CAE 中断使能

位 10 **TWIE**：传输水印中断使能 (Transfer watermark interrupt enable)
此位由软件置 1 和清零。
0：TW 中断禁止
1：TW 中断使能

位 9 **TCIE**：传输完成中断使能 (Transfer complete interrupt enable)
此位由软件置 1 和清零。
0：TC 中断禁止
1：TC 中断使能

位 8 **TEIE**：传输错误中断使能 (Transfer error interrupt enable)

此位由软件置 1 和清零。

0：TE 中断禁止

1：TE 中断使能

位 7:3 保留，必须保持复位值

位 2 **ABORT**：中止 (Abort)

此位可用于中止当前传输。此位可通过软件置 1 并在 START 位复位时由硬件自动复位。

0：不请求传输中止

1：请求传输中止

位 1 **SUSP**：挂起 (Suspend)

此位可用于挂起当前传输。此位由软件置 1 和复位。此位在 START 位复位时由硬件自动复位。

0：传输不挂起

1：传输挂起

位 0 **START**：启动 (Start)

此位可用于根据各种配置寄存器中加载的参数启动 DMA2D。在下列情况下将自动复位此位：

- 传输结束时
- 通过用户应用程序将 DMA2D_CR 中的 ABORT 位置 1 中止数据传输时
- 数据传输出错时
- 因配置错误或已经在进行其它传输操作（CLUT 自动加载）导致数据传输未启动时

18.6.2 DMA2D 中断状态寄存器 (DMA2D_ISR)

DMA2D Interrupt Status Register

偏移地址：0x0004

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CEIF	CTCIF	CAEIF	TWIF	TCIF	TEIF
										r	r	r	r	r	r

位 31:6 保留，必须保持复位值

位 5 **CEIF**：配置错误中断标志 (Configuration error interrupt flag)
DMA2D_CR、DMA2DFGPFCCR 或 DMA2D_BGPFCCR 的 START 位置 1 以及编程了错误的配置时，此位置 1。

位 4 **CTCIF**：CLUT 传输完成中断标志 (CLUT transfer complete interrupt flag)
完成将 CLUT 从系统存储区复制到 DMA2D 内部存储器时，此位置 1。

位 3 **CAEIF**：CLUT 访问错误中断标志 (CLUT access error interrupt flag)
在从系统存储器自动将 CLUT 复制到 DMA2D 内部存储器期间，CPU 若访问 CLUT，此位将置 1。

位 2 **TWIF**：传输水印中断标志 (Transfer watermark interrupt flag)
带水印行的最后一个像素完成传输时，此位置 1。

位 1 **TCIF**：传输完成中断标志 (Transfer complete interrupt flag)
DMA2D 传输操作完成（仅限数据传输）时此位置 1。

位 0 **TEIF**：传输错误中断标志 (Transfer error interrupt flag)
DMA 传输期间（数据传输或 CLUT 自动加载）出错时此位置 1。

18.6.3 DMA2D 中断标志清零寄存器 (DMA2D_IFCR)

DMA2D interrupt flag clear register

偏移地址：0x0008

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CCEIF	CCTCIF	CAECIF	CTWIF	CTCIF	CTEIF
										rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1

位 31:6 保留，必须保持复位值

位 5 **CCEIF**：清除配置错误中断标志 (Clear configuration error interrupt flag)
将此位编程为 1 可清除 DMA2D_ISR 寄存器中的 CEIF 标志

位 4 **CCTCIF**：清除 CLUT 传输完成中断标志 (Clear CLUT transfer complete interrupt flag)
将此位编程为 1 可清除 DMA2D_ISR 寄存器中的 CTCIF 标志

位 3 **CAECIF**：清除 CLUT 访问错误中断标志 (Clear CLUT access error interrupt flag)
将此位编程为 1 可清除 DMA2D_ISR 寄存器中的 CAEIF 标志

位 2 **CTWIF**：清除传输水印中断标志 (Clear transfer watermark interrupt flag)
将此位编程为 1 可清除 DMA2D_ISR 寄存器中的 TWIF 标志

位 1 **CTCIF**：清除传输完成中断标志 (Clear transfer complete interrupt flag)
将此位编程为 1 可清除 DMA2D_ISR 寄存器中的 TCIF 标志

位 0 **CTEIF**：清除传输错误中断标志 (Clear Transfer error interrupt flag)
将此位编程为 1 可清除 DMA2D_ISR 寄存器中的 TEIF 标志

18.6.4 DMA2D 前景层存储器地址寄存器 (DMA2D_FGMAR)

DMA2D foreground memory address register

偏移地址：0x000C

复位值：0x0000 0000

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

位 31:0 **MA[31:0]**：存储器地址 (Memory Address)

前景层图像所用数据的地址。只有在禁止数据传输的情况下才能写入该寄存器。数据传输一旦启动，此寄存器即变为只读。

地址对齐必须与所选图像格式相匹配，例如每像素 32 位格式必须为 32 位对齐，每像素 16 位格式必须为 16 位对齐，而每像素 4 位格式必须为 8 位对齐。

18.6.5 DMA2D 前景层偏移寄存器 (DMA2D_FGOR)

DMA2D foreground offset register

偏移地址：0x0010

复位值：0x0000 0000

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

位 31:14 保留，必须保持复位值

位 13:0 **LO[13:0]**：行偏移 (Line Offset)

用于前景层图像的行偏移（以像素表示）。此值用于生成地址。行偏移将添加到各行末尾，用于确定下一行的起始地址。

只有在禁止数据传输的情况下才能写入这些位。数据传输一旦启动，这些位将变为只读。

如果图像格式为每像素 4 位，则行偏移值必须为偶数。

18.6.6 DMA2D 背景层存储器地址寄存器 (DMA2D_BGMAR)

DMA2D background memory address register

偏移地址：0x0014

复位值：0x0000 0000

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

位 31:0 **MA[31:0]**：存储器地址 (Memory Address)

背景层图像所用数据的地址。只有在禁止数据传输的情况下才能写入该寄存器。数据传输一旦启动，此寄存器即变为只读。

地址对齐必须与所选图像格式相匹配，例如每像素 32 位格式必须为 32 位对齐，每像素 16 位格式必须为 16 位对齐，而每像素 4 位格式必须为 8 位对齐。

18.6.7 DMA2D 背景层偏移寄存器 (DMA2D_BGOR)

DMA2D background offset register

偏移地址：0x0018

复位值：0x0000 0000

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

位 31:14 保留，必须保持复位值

位 13:0 **LO[13:0]**：行偏移 (Line Offset)

用于背景层图像的行偏移（以像素表示）。此值用于生成地址。行偏移将添加到各行末尾，用于确定下一行的起始地址。

只有在禁止数据传输的情况下才能写入这些位。数据传输一旦启动，这些位将变为只读。

如果图像格式为每像素 4 位，则行偏移值必须为偶数。

18.6.8 DMA2D 前景层 PFC 控制寄存器 (DMA2D_FGPFCCR)

DMA2D foreground PFC control register

偏移地址：0x001C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ALPHA[7:0]								Res.	Res.	RBS	AI	CSS[1:0]		AM[1:0]	
rw	rw	rw	rw	rw	rw	rw	rw			rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CS[7:0]								Res.	Res.	START	CCM	CM[3:0]			
rw	rw	rw	rw	rw	rw	rw	rw			rc_w1	rw	rw	rw	rw	rw

位 31:24 **ALPHA[7:0]**：Alpha 值 (Alpha value)

这些位定义固定的 alpha 通道值，该值可替代原始的 alpha 值或与原始的 alpha 值相乘，具体取决于通过 AM[1:0] 位选择的 alpha 模式。

只有在禁止数据传输的情况下才能写入这些位。传输一旦启动，这些位将变为只读。

位 23:22 保留，必须保持复位值

位 21 **RBS**：红蓝交换 (Red Blue Swap)

此位可交换 R 和 B，从而支持 BGR 或 ABGR 颜色格式。传输一旦启动，此位将变为只读。

0：常规模式（RGB 或 ARGB）

1：交换模式（BGR 或 ABGR）

位 20 **AI**：将 Alpha 取反 (Alpha Inverted)

此位用于将 alpha 值取反。传输一旦启动，此位将变为只读。

0：常规 Alpha

1：取反 Alpha

位 19:18 **CSS[1:0]**：色度子采样 (Chroma Sub-Sampling)

这些位用于为 YCbCr 颜色模式定义色度子采样模式。传输一旦启动，这些位将变为只读。

00：4:4:4（无色度子采样）

01：4:2:2

10：4:2:0

其它：无意义

位 17:16 **AM[1:0]**：Alpha 模式 (Alpha mode)

这些位用于选择将用于前景层图像的 alpha 通道值。只有在禁止数据传输的情况下才能写入这些位。传输一旦启动，这些位将变为只读。

00：不修改前景层图像的 alpha 通道值

01：原始前景层图像的 alpha 通道值替换为 ALPHA[7:0]

10：原始前景层图像的 alpha 通道值替换为 ALPHA[7:0] 与原始 alpha 通道值的乘积

其它配置无意义

位 15:8 **CS[7:0]**：CLUT 大小 (CLUT size)

这些位定义前景层图像所用的 CLUT 的大小。CLUT 传输一旦启动，此字段将变为只读。CLUT 条目数等于 CS[7:0] + 1。

位 7:6 保留，必须保持复位值

位 5 **START**：启动 (Start)

可将此位置 1 以启动 CLUT 的自动加载过程。该位在以下情况下自动复位：

- 传输结束时
- 通过用户应用程序将 DMA2D_CR 中的 ABORT 位置 1 中止传输时
- 传输出错时
- 因配置错误或已经在进行其它传输操作（数据传输或自动背景层 CLUT 传输）导致传输未启动时

位 4 **CCM**：CLUT 颜色模式 (CLUT color mode)

此位定义 CLUT 的颜色格式。只有在禁止数据传输的情况下才能写入该位。CLUT 传输一旦启动，此位将变为只读。

0：ARGB8888
1：RGB888

位 3:0 **CM[3:0]**：颜色模式 (Color mode)

这些位定义前景层图像的颜色格式。只有在禁止数据传输的情况下才能写入这些位。传输一旦启动，这些位将变为只读。

0000：ARGB8888
0001：RGB888
0010：RGB565
0011：ARGB1555
0100：ARGB4444
0101：L8
0110：AL44
0111：AL88
1000：L4
1001：A8
1010：A4
1011：YCbCr
其它：无意义

18.6.9 DMA2D 前景层颜色寄存器 (DMA2D_FGCOLR)

DMA2D foreground color register

偏移地址：0x0020

复位值：0x0000 0000

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

位 31:24 保留，必须保持复位值

位 23:16 **RED[7:0]**：红色值 (Red value)

这些位定义前景层图像的 A4 或 A8 模式的红色值。只有在禁止数据传输的情况下才能写入这些位。传输一旦启动，这些位将变为只读。

位 15:8 **GREEN[7:0]**：绿色值 (Green value)

这些位定义前景层图像的 A4 或 A8 模式的绿色值。只有在禁止数据传输的情况下才能写入这些位。传输一旦启动，这些位将变为只读。

位 7:0 **BLUE[7:0]**：蓝色值 (Blue value)

这些位定义前景层图像的 A4 或 A8 模式的蓝色值。只有在禁止数据传输的情况下才能写入这些位。传输一旦启动，这些位将变为只读。

18.6.10 DMA2D 背景层 PFC 控制寄存器 (DMA2D_BGPFCCR)

DMA2D background PFC control register

偏移地址：0x0024

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ALPHA[7:0]								Res.	Res.	RBS	AI	Res.	Res.	AM[1:0]	
rw	rw	rw	rw	rw	rw	rw	rw			rw	rw			rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CS[7:0]								Res.	Res.	START	CCM	CM[3:0]			
rw	rw	rw	rw	rw	rw	rw	rw			rc_w1	rw	rw	rw	rw	rw

位 31:24 **ALPHA[7:0]**：Alpha 值 (Alpha value)

这些位定义固定的 alpha 通道值，该值可替代原始的 alpha 值或与原始的 alpha 值相乘，具体取决于通过 AM[1:0] 位选择的 alpha 模式。只有在禁止数据传输的情况下才能写入这些位。传输一旦启动，这些位将变为只读。

位 23:22 保留，必须保持复位值

位 21 **RBS**：红蓝交换 (Red Blue Swap)

此位可交换 R 和 B，从而支持 BGR 或 ABGR 颜色格式。传输一旦启动，此位将变为只读。

0：常规模式（RGB 或 ARGB）

1：交换模式（BGR 或 ABGR）

位 20 **AI**：将 Alpha 取反 (Alpha Inverted)

此位用于将 alpha 值取反。传输一旦启动，此位将变为只读。

0：常规 Alpha

1：取反 Alpha

位 19:18 保留，必须保持复位值

位 17:16 **AM[1:0]**：Alpha 模式 (Alpha mode)

这些位定义将用于背景层图像的 alpha 通道值。只有在禁止数据传输的情况下才能写入这些位。传输一旦启动，这些位将变为只读。

00：不修改前景层图像的 alpha 通道值

01：原始背景层图像的 alpha 通道值替换为 ALPHA[7:0]

10：原始背景层图像的 alpha 通道值替换为 ALPHA[7:0] 与原始 alpha 通道值的乘积

其它：无意义

位 15:8 **CS[7:0]**：CLUT 大小 (CLUT size)

这些位定义背景层图像所用的 CLUT 的大小。CLUT 传输一旦启动，此字段将变为只读。CLUT 条目数等于 CS[7:0] + 1。

位 7:6 保留，必须保持复位值

位 5 **START**：启动 (Start)

可将此位置 1 以启动 CLUT 的自动加载过程。该位在以下情况下自动复位：

- 传输结束时
- 通过用户应用程序将 DMA2D_CR 中的 ABORT 位置 1 中止传输时
- 传输出错时
- 因配置错误或已经在进行其它传输操作（数据传输或自动背景层 CLUT 传输）导致传输未启动时

位 4 **CCM**：CLUT 颜色模式 (CLUT Color mode)

这些位定义 CLUT 的颜色格式。只有在禁止传输的情况下才能写入该寄存器。CLUT 传输一旦启动，此位将变为只读。

0：ARGB8888

1：RGB888

位 3:0 **CM[3:0]**：颜色模式 (Color mode)

这些位定义前景层图像的颜色格式。只有在禁止数据传输的情况下才能写入这些位。传输一旦启动，这些位将变为只读。

0000：ARGB8888

0001：RGB888

0010：RGB565

0011：ARGB1555

0100：ARGB4444

0101：L8

0110：AL44

0111：AL88

1000：L4

1001：A8

1010：A4

其它：无意义

18.6.11 DMA2D 背景层颜色寄存器 (DMA2D_BGCOLR)

DMA2D background color register

偏移地址：0x0028

复位值：0x0000 0000

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

位 31:24 保留，必须保持复位值

位 23:16 **RED[7:0]**：红色值 (Red value)

这些位定义背景层图像的 A4 或 A8 模式的红色值。只有在禁止数据传输的情况下才能写入这些位。传输一旦启动，这些位将变为只读。

位 15:8 **GREEN[7:0]**：绿色值 (Green value)

这些位定义背景层图像的 A4 或 A8 模式的绿色值。只有在禁止数据传输的情况下才能写入这些位。传输一旦启动，这些位将变为只读。

位 7:0 **BLUE[7:0]**：蓝色值 (Blue value)

这些位定义背景层图像的 A4 或 A8 模式的蓝色值。只有在禁止数据传输的情况下才能写入这些位。传输一旦启动，这些位将变为只读。

18.6.12 DMA2D 前景层 CLUT 存储器地址寄存器 (DMA2D_FGCMAR)

DMA2D foreground CLUT memory address register

偏移地址：0x002C

复位值：0x0000 0000

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

位 31:0 **MA[31:0]**：存储器地址 (Memory Address)

专用于前景层图像的 CLUT 地址所使用的数据地址。只有不存在进行中传输的情况下才能写入该寄存器。CLUT 传输一旦启动，此寄存器将变为只读。

如果前景层 CLUT 格式是 32 位，则地址必须是 32 位对齐。

18.6.13 DMA2D 背景层 CLUT 存储器地址寄存器 (DMA2D_BGCMAR)

DMA2D background CLUT memory address register

偏移地址：0x0030

复位值：0x0000 0000

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

位 31:0 **MA[31:0]**：存储器地址 (Memory Address)

专用于背景层图像的 CLUT 地址所使用的数据地址。只有不存在进行中传输的情况下才能写入该寄存器。CLUT 传输一旦启动，此寄存器将变为只读。

如果背景层 CLUT 格式是 32 位，则地址必须是 32 位对齐。

18.6.14 DMA2D 输出 PFC 控制寄存器 (DMA2D_OPFCCR)

DMA2D output PFC control register

偏移地址：0x0034

复位值：0x0000 0000

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

位 31:22 保留，必须保持复位值

位 21 **RBS**：红蓝交换 (Red Blue Swap)

此位可交换 R 和 B，从而支持 BGR 或 ABGR 颜色格式。传输一旦启动，此位将变为只读。

0：常规模式（RGB 或 ARGB）

1：交换模式（BGR 或 ABGR）

位 20 **AI**：将 Alpha 取反 (Alpha Inverted)

此位用于将 alpha 值取反。传输一旦启动，此位将变为只读。

0：常规 Alpha

1：取反 Alpha

位 19:3 保留，必须保持复位值

位 2:0 **CM[2:0]**：颜色模式 (Color mode)

这些位定义背景层图像的颜色格式。只有在禁止数据传输的情况下才能写入这些位。传输一旦启动，这些位将变为只读。

000：ARGB8888

001：RGB888

010：RGB565

011：ARGB1555

100：ARGB4444

其它：无意义

18.6.15 DMA2D 输出颜色寄存器 (DMA2D_OCOLR)

DMA2D output color register

偏移地址：0x0038

复位值：0x0000 0000

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

位 31:24 **ALPHA[7:0]**：Alpha 通道值 (Alpha Channel Value)

这些位定义输出颜色的 alpha 通道。只有在禁止数据传输的情况下才能写入这些位。传输一旦启动，这些位将变为只读。

位 23:16 **RED[7:0]**：红色值 (Red value)

这些位定义输出图像的红色值。只有在禁止数据传输的情况下才能写入这些位。传输一旦启动，这些位将变为只读。

位 15:8 **GREEN[7:0]**：绿色值 (Green value)

这些位定义输出图像的绿色值。只有在禁止数据传输的情况下才能写入这些位。传输一旦启动，这些位将变为只读。

位 7:0 **BLUE[7:0]**：蓝色值 (Blue value)

这些位定义输出图像的蓝色值。只有在禁止数据传输的情况下才能写入这些位。传输一旦启动，这些位将变为只读。

18.6.16 DMA2D 输出存储器地址寄存器 (DMA2D_OMAR)

DMA2D output memory address register

偏移地址：0x003C

复位值：0x0000 0000

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

位 31:0 **MA[31:0]**：存储器地址 (Memory Address)

输出 FIFO 所用数据的地址。只有在禁止数据传输的情况下才能写入这些位。传输一旦启动，这些位将变为只读。

地址对齐必须与所选图像格式相匹配，例如每像素 32 位格式必须为 32 位对齐，每像素 16 位格式必须为 16 位对齐。

18.6.17 DMA2D 输出偏移寄存器 (DMA2D_OOR)

DMA2D output offset register

偏移地址：0x0040

复位值：0x0000 0000

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

位 31:14 保留，必须保持复位值

位 13:0 **LO[13:0]**：行偏移 (Line Offset)

用于输出的行偏移（以像素表示）。此值用于生成地址。行偏移将添加到各行末尾，用于确定下一行的起始地址。只有在禁止数据传输的情况下才能写入这些位。传输一旦启动，这些位将变为只读。

18.6.18 DMA2D 行数寄存器 (DMA2D_NLR)

DMA2D number of line register

偏移地址：0x0044

复位值：0x0000 0000

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

位 31:30 保留，必须保持复位值

位 29:16 **PL[13:0]**：每行像素数 (Pixel per lines)

待传输区域的每行像素数。只有在禁止数据传输的情况下才能写入这些位。传输一旦启动，这些位将变为只读。

如果任何一个输入图像格式为每像素 4 位，则每行像素数必须为偶数。

位 15:0 **NL[15:0]**：行数 (Number of lines)

待传输区域的行数。只有在禁止数据传输的情况下才能写入这些位。传输一旦启动，这些位将变为只读。

18.6.19 DMA2D 行水印寄存器 (DMA2D_LWR)

DMA2D line watermark register

偏移地址：0x0048

复位值：0x0000 0000

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

位 31:16 保留，必须保持复位值

位 15:0 **LW[15:0]**：行水印 (Line watermark)
这些位可用以配置可产生中断的行水印。
在带水印行的最后一个像素传输完成时产生中断。只有在禁止数据传输的情况下才能写入这些位。传输一旦启动，这些位将变为只读。

18.6.20 DMA2D AXI 主设备定时器配置寄存器 (DMA2D_AMTCR)

DMA2D AXI master timer configuration register

偏移地址：0x004C

复位值：0x0000 0000

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

位 31:16 保留，必须保持复位值

位 15:8 **DT[7:0]**：死区 (Dead Time)
在 AXI 主设备端口上两个连续访问之间所插入的死区值，以 AXI 时钟周期数表示。这些位表示两个连续 AXI 访问之间允许占用的最少周期数。

位 7:1 保留，必须保持复位值

位 0 **EN**：使能 (Enable)
使能死区功能。

18.6.21 DMA2D 寄存器映射

下表对 DMA2D 寄存器进行了汇总。有关 DMA2D 寄存器基地址的信息，请参见*第 100 页的第 2.2.2 节*。

表 128. DMA2D 寄存器映射和复位值

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x0000	DMA2D_CR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MODE[1:0]		Res.	Res.	Res.	CEIE	CTCIE	CAEIE	TWIE	TCIE	TEIE	Res.	Res.	Res.	Res.	ABORT	SUSP	START
	Reset value															0	0				0	0	0	0	0	0					0	0	0
0x0004	DMA2D_ISR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CEIF	CTCIF	CAEIF	TWIF	TCIF	TEIF
	Reset value																											0	0	0	0	0	0
0x0008	DMA2D_IFCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CCEIF	CCTCIF	CAECIF	CTWIF	CTCIF	CTEIF
	Reset value																											0	0	0	0	0	0
0x000C	DMA2D_FGMAR	MA[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0010	DMA2D_FGOR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LO[13:0]													
	Reset value																			0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0014	DMA2D_BGMAR	MA[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0018	DMA2D_BGOR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LO[13:0]													
	Reset value																			0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x001C	DMA2D_FGPFCCR	ALPHA[7:0]								Res.	Res.	RBS	AI	CSS[1:0]		AM[1:0]		CS[7:0]								Res.	Res.	START	CCM	CM[3:0]			
	Reset value	0	0	0	0	0	0	0	0			0	0	0	0	0	0	0	0	0	0	0	0	0	0			0	0	0	0	0	0
0x0020	DMA2D_FGCOLR	APLHA[7:0]								RED[7:0]								GREEN[7:0]								BLUE[7:0]					
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0024	DMA2D_BGPFCCR	ALPHA[7:0]								Res.	Res.	RBS	AI	Res.	Res.	AM[1:0]		CS[7:0]								Res.	Res.	START	CCM	CM[3:0]			
	Reset value	0	0	0	0	0	0	0	0			0	0			0	0	0	0	0	0	0	0	0	0			0	0	0	0	0	0
0x0028	DMA2D_BGCOLR	APLHA[7:0]								RED[7:0]								GREEN[7:0]								BLUE[7:0]					
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x002C	DMA2D_FGCMAR	MA[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0030	DMA2D_BGCMAR	MA[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0034	DMA2D_OPFCCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RBS	AI	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CM[2:0]		
	Reset value											0	0																		0	0	0

表 128. DMA2D 寄存器映射和复位值（续）

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x0038	DMA2D_OCOLR	APLHA[7:0]								RED[7:0]								GREEN[7:0]								BLUE[7:0]							
		Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RED[4:0]					GREEN[6:0]						BLUE[4:0]				
		Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	A	RED[4:0]				GREEN[4:0]						BLUE[4:0]				
		Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ALPHA[3:0]				RED[3:0]				GREEN[3:0]				BLUE[3:0]			
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x003C	DMA2D_OMAR	MA[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0040	DMA2D_OOR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LO[13:0]													
	Reset value																			0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0044	DMA2D_NLR	Res.	Res.	PL[13:0]														NL[15:0]															
	Reset value			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0048	DMA2D_LWR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LW[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x004C	DMA2D_AMTCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DT[7:0]								Res.	Res.	Res.	Res.	Res.	Res.	Res.	EN
	Reset value																	0	0	0	0	0	0	0	0								0
0x0050-Ox03FF		Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
0x0400-0x07FF	DMA2D_FGCLUT	APLHA[7:0][255:0]								RED[7:0][255:0]								GREEN[7:0][255:0]								BLUE[7:0][255:0]							
	Reset value	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
0x0800-0x0BFF	DMA2D_BGCLUT	APLHA[7:0][255:0]								RED[7:0][255:0]								GREEN[7:0][255:0]								BLUE[7:0][255:0]							
	Reset value	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

有关寄存器边界地址的信息，请参见*第 100 页的第 2.2.2 节*。

19 嵌套向量中断控制器

19.1 NVIC 特性

NVIC 包括下列特性：

- 具有多达 150 个用于 STM32H7xxx 的可屏蔽中断通道（不包括带 FPU 的 Cortex®-M7 的 16 根中断线）
- 16 个可编程优先级（使用了 4 位中断优先级）
- 低延迟异常和中断处理
- 电源管理控制
- 系统控制寄存器的实现

嵌套向量中断控制器 (NVIC) 和处理器内核接口紧密配合，可以实现低延迟的中断处理和晚到中断的高效处理。

包括内核异常在内的所有中断均通过 NVIC 进行管理。

关于异常和 NVIC 编程的更多信息，请参见 PM0253 Cortex®-M7 编程手册。

19.1.1 SysTick 校准值寄存器

SysTick 校准值设置为 18750。当 SysTick 时钟设置为 18.75 MHz（HCLK/8，HCLK 设为 150 MHz），会产生 1 ms 时间基准。

19.1.2 中断和异常向量

与 NVIC 连接的异常向量如下：reset、NMI、HardFault、MemManage、Bus Fault、UsageFault、SVCall、 DebugMonitor、 PendSV 和 SysTick。

表 129. NVIC(1)

信号	优先级	NVIC 位置	缩略语	说明	偏移地址
-	-	-	-	保留	0x0000 0000
-	-3	-	Reset	复位	0x0000 0004
-	-2	-	NMI	不可屏蔽中断。 RCC 时钟安全系统 (CSS) 连接到 NMI 向量。	0x0000 0008
-	-1	-	HardFault	所有类型的错误	0x0000 000C
-	0	-	MemManage	存储器管理	0x0000 0010
-	1	-	BusFault	预取故障，存储器访问故障	0x0000 0014
-	2	-	UsageFault	未定义的指令或非法状态	0x0000 0018
-	-	-	-	保留	0x0000 001C-0x0000 002B

表 129. NVIC(1)（续）

信号	优先级	NVIC 位置	缩略语	说明	偏移地址
-	3	-	SVCall	通过 SWI 指令调用的系统服务	0x0000 002C
-	4	-	DebugMonitor	调试监控器	0x0000 0030
-	-	-	-	保留	0x0000 0034
-	5	-	PendSV	可挂起的系统服务	0x0000 0038
-	6	-	SysTick	系统节拍定时器	0x0000 003C
wwdg1_it	7	0	WWDG1	窗口看门狗中断	0x0000 0040
exti_pwr_pvd_wkup	8	1	PVD_PVM	连接到 EXTI 线的可编程电压检测 (PVD) 中断	0x0000 0044
exti_tamp_rtc_wkup	9	2	RTC_TAMP_STAMP_CSS_LSE	RTC 入侵、时间戳	0x0000 0048
lsecss_rcc_it				CSS (LSE)	
exti_wkup_rtc_wkup	10	3	RTC_WKUP	连接到 EXTI 线的 RTC 唤醒中断	0x0000 004C
flash_it	11	4	FLASH	Flash存储器全局中断	0x0000 0050
rcc_it	12	5	RCC	RCC 全局中断	0x0000 0054
exti_exti0_wkup	13	6	EXTI0	EXTI 线 0 中断	0x0000 0058
exti_exti1_wkup	14	7	EXTI1	EXTI 线 1 中断	0x0000 005C
exti_exti2_wkup	15	8	EXTI2	EXTI 线 2 中断	0x0000 0060
exti_exti3_wkup	16	9	EXTI3	EXTI 线 3 中断	0x0000 0064
exti_exti4_wkup	17	10	EXTI4	EXTI 线 4 中断	0x0000 0068
dma1_it0	18	11	DMA_STR0	DMA1 流 0 全局中断	0x0000 006C
dma1_it1	19	12	DMA_STR1	DMA1 流 1 全局中断	0x0000 0070
dma1_it2	20	13	DMA_STR2	DMA1 流 2 全局中断	0x0000 0074
dma1_it3	21	14	DMA_STR3	DMA1 流 3 全局中断	0x0000 0078
dma1_it4	22	15	DMA_STR4	DMA1 流 4 全局中断	0x0000 007C
dma1_it5	23	16	DMA_STR5	DMA1 流 5 全局中断	0x0000 0080
dma1_it6	24	17	DMA_STR6	DMA1 流 6 全局中断	0x0000 0084
adc1_it	25	18	ADC1_2	ADC1 和 ADC2 全局中断	0x0000 0088
adc2_it					

表 129. NVIC(1)（续）

信号	优先级	NVIC 位置	缩略语	说明	偏移地址
ttfdcan_intr0_it	26	19	FDCAN1_IT0	FDCAN1 中断 0	0x0000 008C
fdcan_intr0_it	27	20	FDCAN2_IT0	FDCAN2 中断 0	0x0000 0090
ttfdcan_intr1_it	28	21	FDCAN1_IT1	FDCAN1 中断 1	0x0000 0094
fdcan_intr1_it	29	22	FDCAN2_IT1	FDCAN2 中断 1	0x0000 0098
exti_exti5_wkup	30	23	EXTI9_5	EXTI 线 [9:5] 中断	0x0000 009C
exti_exti6_wkup					
exti_exti7_wkup					
exti_exti8_wkup					
exti_exti9_wkup					
tim1_brk_it	31	24	TIM1_BRK	TIM1 断路中断	0x0000 00A0
tim1_upd_it	32	25	TIM1_UP	TIM1 更新中断	0x0000 00A4
tim1_trg_it	33	26	TIM1_TRG_COM	TIM1 触发和通信中断	0x0000 00A8
tim1_cc_it	34	27	TIM_CC	TIM1 捕获/比较中断	0x0000 00AC
tim2_it	35	28	TIM2	TIM2 全局中断	0x0000 00B0
tim3_it	36	29	TIM3	TIM3 全局中断	0x0000 00B4
tim4_it	37	30	TIM4	TIM4 全局中断	0x0000 00B8
i2c1_ev_it	38	31	I2C1_EV	I2C1 事件中断	0x0000 00BC
exti_i2c1_ev_wkup					
i2c1_err_it	39	32	I2C1_ER	I2C1 错误中断	0x0000 00C0
i2c2_ev_it	40	33	I2C2_EV	I2C2 事件中断	0x0000 00C4
exti_i2c2_ev_wkup					
i2c2_err_it	41	34	I2C2_ER	I2C2 错误中断	0x0000 00C8
spi1_it	42	35	SPI1	SPI1 全局中断	0x0000 00CC
exti_spi1_it					
spi2_it	43	36	SPI2	SPI2 全局中断	0x0000 00D0
exti_spi2_it					
usart1_gbl_it	44	37	USART1	USART1 全局中断	0x0000 00D4
exti_usart1_wkup					
usart2_gbl_it	45	38	USART2	USART2 全局中断	0x0000 00D8
exti_usart2_wkup					
usart3_gbl_it	46	39	USART3	USART3 全局中断	0x0000 00DC
exti_usart3_wkup					

表 129. NVIC(1)（续）

信号	优先级	NVIC 位置	缩略语	说明	偏移地址
exti_exti10_it	47	40	EXTI15_10	EXTI 线 [15:10] 中断	0x0000 00E0
exti_exti11_wkup					
exti_exti12_wkup					
exti_exti13_wkup					
exti_exti14_wkup					
exti_exti15_wkup					
exti_rtc_al	48	41	RTC_ALARM	连接到 EXTI 线的 RTC 闹钟（A 和 B）中断	0x0000 00E4
-	49	42	-	-	0x0000 00E8
tim8_brk_it	50	43	TIM8_BRK_TIM12	TIM8 断路和 TIM12 全局中断	0x0000 00EC
tim12_gbl_it					
tim8_upd_it	51	44	TIM8_UP_TIM13	TIM8 更新和 TIM13 全局中断	0x0000 00F0
tim13_gbl_it					
tim8_trg_it	52	45	TIM8_TRG_COM _TIM14	TIM8 触发/通信和 TIM14 全局中断	0x0000 00F4
tim14_gbl_it					
tim8_cc_it	53	46	TIM8_CC	TIM8 捕获/比较中断	0x0000 00F8
dma1_it7	54	47	DMA1_STR7	DMA1 流 7 全局中断	0x0000 00FC
fmc_gbl_it	55	48	FMC	FMC 全局中断	0x0000 0100
sdmmc_gbl_it	56	49	SDMMC1	SDMMC 全局中断	0x0000 0104
tim5_gbl_it	57	50	TIM5	TIM5 全局中断	0x0000 0108
spi3_it	58	51	SPI3	SPI3 全局中断	0x0000 010C
exti_spi3_wkup					
uart4_gbl_it	59	52	UART4	UART4 全局中断	0x0000 0110
exti_uart4_wkup					
uart5_gbl_it	60	53	UART5	UART5 全局中断	0x0000 0114
exti_uart5_wkup					
tim6_gbl_it	61	54	TIM6_DAC	TIM6 全局中断	0x0000 0118
dac_unr_it				DAC 下溢错误中断	
tim7_gbl_it	62	55	TIM7	TIM7 全局中断	0x0000 011C
dma2_it0	63	56	DMA2_STR0	DMA2 流 0 中断	0x0000 0120
dma2_it1	64	57	DMA2_STR1	DMA2 流 1 中断	0x0000 0124
dma2_it2	65	58	DMA2_STR2	DMA2 流 2 中断	0x0000 0128

表 129. NVIC(1)（续）

信号	优先级	NVIC 位置	缩略语	说明	偏移地址
dma2_it3	66	59	DMA2_STR3	DMA2 流 3 中断	0x0000 012C
dma2_it4	67	60	DMA2_STR4	DMA2 流 4 中断	0x0000 0130
eth_sbd_intr_it	68	61	ETH	以太网全局中断	0x0000 0134
exti_eth_wkup	69	62	ETH_WKUP	连接到 EXTI 线的以太网唤醒中断	0x0000 0138
can_cal_it	70	63	FDCAN_CAL	CAN2 TX 中断	0x0000 013C
NC	71	64	-	-	0x0000 0140
NC	72	65	-	-	0x0000 0144
NC	73	66	-	-	0x0000 0148
NC	74	67	-	-	0x0000 014C
dma2_it5	75	68	DMA2_STR5	DMA2 流 5 中断	0x0000 0150
dma2_it6	76	69	DMA2_STR6	DMA2 流 6 中断	0x0000 0154
dma2_it7	77	70	DMA2_STR7	DMA2 流 7 中断	0x0000 0158
usart6_gbl_it	78	71	USART6	USART6 全局中断	0x0000 015C
exti_usart6_wkup				USART6 唤醒中断	
i2c3_ev_it	79	72	I2C3_EV	I2C3 事件中断	0x0000 0160
exti_i2c3_ev_wkup					
i2c3_err_it	80	73	I2C3_ER	I2C3 错误中断	0x0000 0164
usb1_out_it	81	74	OTG_HS_EP1_OUT	OTG_HS 输出全局中断	0x0000 0168
usb1_in_it	82	75	OTG_HS_EP1_IN	OTG_HS 输入全局中断	0x0000 016C
exti_usb1_wkup	83	76	OTG_HS_WKUP	OTG_HS 唤醒中断	0x0000 0170
usb1_gbl_it	84	77	OTG_HS	OTG_HS 全局中断	0x0000 0174
dcmi_it	85	78	DCMI	DCMI 全局中断	0x0000 0178
cryp_it	86	79	CRYP	CRYP 全局中断	0x0000 017C
hash_rng_it	87	80	HASH_RNG	HASH 和 RNG 全局中断	0x0000 0180
cpu_fpu_it	88	81	FPU	CPU FPU	0x0000 0184
uart7_gbl_it	89	82	UART7	UART7 全局中断	0x0000 0188
exti_uart7_wkup					
uart8_gbl_it	90	83	UART8	UART8 全局中断	0x0000 018C
exti_uart8_wkup					
spi4_it	91	84	SPI4	SPI4 全局中断	0x0000 0190
exti_spi4_wkup					

表 129. NVIC(1)（续）

信号	优先级	NVIC 位置	缩略语	说明	偏移地址
spi5_it	92	85	SPI5	SPI5 全局中断	0x0000 0194
exti_spi5_wkup					
spi6_it	93	86	SPI6	SPI6 全局中断	0x0000 0198
exti_spi6_wkup					
sai1_it	94	87	SAI1	SAI1 全局中断	0x0000 019C
ltdc_it	95	88	LTDC	LCD-TFT 全局中断	0x0000 01A0
ltdc_err_it	96	89	LTDC_ER	LCD-TFT 错误中断	0x0000 01A4
dma2d_gbl_it	97	90	DMA2D	DMA2D 全局中断	0x0000 01A8
-	98	91	SAI2	SAI2 全局中断	0x0000 01AC
-	99	92	QUADSPI	QuadSPI 全局中断	0x0000 01B0
lptim1_it	100	93	LPTIM1	LPTIM1 全局中断	0x0000 01B4
exti_lptim_wkup					
cec_it	101	94	CEC	HDMI-CEC 全局中断	0x0000 01B8
exti_cec_it					
i2c4_ev_it	102	95	I2C4_EV	I2C4 事件中断	0x0000 01BC
exti_i2c4_ev_it					
i2c4_err_it	103	96	I2C4_ER	I2C4 错误中断	0x0000 01C0
-	104	97	SPDIF	SPDIFRX 全局中断	0x0000 01C4
usb2_out_it	105	98	OTG_FS_EP1_OUT	OTG_FS 输出全局中断	0x0000 01C8
usb2_in_it	106	99	OTG_FS_EP1_IN	OTG_FS 输入全局中断	0x0000 01CC
exti_usb2_wkup	107	100	OTG_FS_WKUP	OTG_FS 唤醒	0x0000 01D0
usb2_gbl_it	108	101	OTG_FS	OTG_FS 全局中断	0x0000 01D4
dmamux1_ovr_it	109	102	DMAMUX1_OV	DMAMUX1 溢出中断	0x0000 01D8
hrtim1_mst_it	110	103	HRTIM1_MST	HRTIM1 主定时器中断	0x0000 01DC
hrtim1_tima_it	111	104	HRTIM1_TIMA	HRTIM1 定时器 A 中断	0x0000 01E0
hrtim1_timb_it	112	105	HRTIM_TIMB	HRTIM1 定时器 B 中断	0x0000 01E4
hrtim1_timc_it	113	106	HRTIM1_TIMC	HRTIM1 定时器 C 中断	0x0000 01E8
hrtim1_timd_it	114	107	HRTIM1_TIMD	HRTIM1 定时器 D 中断	0x0000 01EC
hrtim1_time_it	115	108	HRTIM_TIME	HRTIM1 定时器 E 中断	0x0000 01F0
hrtim1_fault_it	116	109	HRTIM1_FLT	HRTIM1 故障中断	0x0000 01F4
dfsdm1_it0	117	110	DFSDM1_FLT0	DFSDM1 滤波器 0 中断	0x0000 01F8
dfsdm1_it1	118	111	DFSDM1_FLT1	DFSDM1 滤波器 1 中断	0x0000 01FC
dfsdm1_it2	119	112	DFSDM1_FLT2	DFSDM1 滤波器 2 中断	0x0000 0200

表 129. NVIC(1)（续）

信号	优先级	NVIC 位置	缩略语	说明	偏移地址
dfsdm1_it3	120	113	DFSDM1_FLT3	DFSDM1 滤波器 3 中断	0x0000 0204
sai3_gbl_it_it	121	114	SAI3	SAI3 全局中断	0x0000 0208
swpmi_gbl_it	122	115	SWPMI1	SWPMI 全局中断	0x0000 020C
exti_swpmi_wup				SWPMI 唤醒	
tim15_gbl_it	123	116	TIM15	TIM15 全局中断	0x0000 0210
tim16_gbl_it	124	117	TIM16	TIM16 全局中断	0x0000 0214
tim17_gbl_it	125	118	TIM17	TIM17 全局中断	0x0000 0218
-	126	119	MDIOS_WKUP	MDIOS 唤醒	0x0000 021C
mdios_it	127	120	MDIOS	MDIOS 全局中断	0x0000 0220
-	128	121	JPEG	JPEG 全局中断	0x0000 0224
mdma_it	129	122	MDMA	MDMA	0x0000 0228
-	131	124	SDMMC	SDMMC 全局中断	0x0000 0230
hsem_it	132	125	HSEM0	HSEM 全局中断 1	0x0000 0234
-	133	-	-	-	0x0000 0238
-	134	127	ADC3	ADC3 全局中断	0x0000 023C
-	135	128	DMAMUX2_OVR	DMAMUX2 溢出中断	0x0000 0240
bdma_ch0_it	136	129	BDMA_CH1	BDMA 通道 1 中断	0x0000 0244
bdma_ch1_it	137	130	BDMA_CH2	BDMA 通道 2 中断	0x0000 0248
bdma_ch2_it	138	131	BDMA_CH3	BDMA 通道 3 中断	0x0000 024C
bdma_ch3_it	139	132	BDMA_CH4	BDMA 通道 4 中断	0x0000 0250
bdma_ch4_it	140	133	BDMA_CH5	BDMA 通道 5 中断	0x0000 0254
bdma_ch5_it	141	134	BDMA_CH6	BDMA 通道 6 中断	0x0000 0258
bdma_ch6_it	142	135	BDMA_CH7	BDMA 通道 7 中断	0x0000 025C
bdma_ch7_it	143	136	BDMA_CH8	BDMA 通道 8 中断	0x0000 0260
comp_gbl_it	144	137	COMP	COMP1 和 COMP2 全局中断	0x0000 0264
exti_comp1_wkup					
exti_comp2_wkup					
lptim2_it	145	138	LPTIM2	LPTIM2 定时器中断	0x0000 0268
exti_lptim2_wkup					
lptim3_it	146	139	LPTIM3	LPTIM2 定时器中断	0x0000 026C
exti_lptim3_wkup					
lptim4_it	147	140	LPTIM4	LPTIM2 定时器中断	0x0000 0270
exti_lptim4_wkup					

表 129. NVIC(1)（续）

信号	优先级	NVIC 位置	缩略语	说明	偏移地址
lptim5_it	148	141	LPTIM5	LPTIM2 定时器中断	0x0000 0274
exti_lptim5_wkup					
lpuart_gbl_it	149	142	LPUART	LPUART 全局中断	0x0000 0278
exti_lpuart_rx_it					
exti_lpuart_tx_it					
exti_d1_wwdg1_wkup	150	143	WWDG1_RST	窗口看门狗中断	0x0000 027C
crs_it	151	144	CRS	时钟恢复系统全局中断	0x0000 0280
-	152	145	-	-	0x0000 0284
-	153	146	SAI4	SAI4 全局中断	0x0000 0288
-	154	147	-	-	0x0000 028C
-	155	148	-	-	0x0000 0290
exti_wkup1_wkup	156	149	WKUP	WKUP1 到 WKUP6 引脚	0x0000 0294
exti_wkup2_wkup					
exti_wkup3_wkup					
exti_wkup4_wkup					
exti_wkup5_wkup					
exti_wkup6_wkup					

1. 当不同的信号连接到同一 NVIC 中断线时，它们会进行逻辑或运算。

20 扩展中断和事件控制器 (EXTI)

扩展中断和事件控制器 (EXTI) 通过可配置的事件输入和直接事件输入来管理唤醒。它可以针对电源控制提供唤醒请求、针对 CPU NVIC 和 D3 域 DMAMUX2 生成中断请求，以及针对 CPU 事件输入生成事件。

EXTI 唤醒请求可让系统从停止模式唤醒，以及让 CPU 从 CStop 模式唤醒。

此外，还可以在运行模式下生成中断请求和事件请求。

20.1 EXTI 主要特性

EXTI 的主要特性如下：

- 所有事件输入均可让 CPU 唤醒以及生成 CPU 中断和/或 CPU 事件
- 某些事件输入允许用户唤醒自主运行模式下的 D3 域，以及针对 D3 域（即 DMAMUX2）生成中断

异步事件输入分为 2 组：

- 可配置事件（来自能够生成脉冲的 I/O 或外设的信号），这类事件具有以下特性：
  - 可选择的有效触发边沿
  - 中断挂起状态寄存器位
  - 单独的中断和事件生成屏蔽
  - 支持软件触发
  - 可配置系统 D3 域唤醒事件包含一个 D3 挂起屏蔽和状态寄存器，并且可能包含一个 D3 中断信号。
- 直接事件（来自其他外设的中断和唤醒源，需要在外设中清除），这类事件具有以下特性
  - 固定上升沿有效触发
  - EXTI 中无中断挂起状态寄存器位（中断挂起状态由生成事件的外设提供）
  - 单独的中断和事件生成屏蔽
  - 不支持软件触发
  - 直接系统 D3 域唤醒事件包含一个 D3 挂起屏蔽和状态寄存器，并且可能包含一个 D3 中断信号

20.2 EXTI 框图

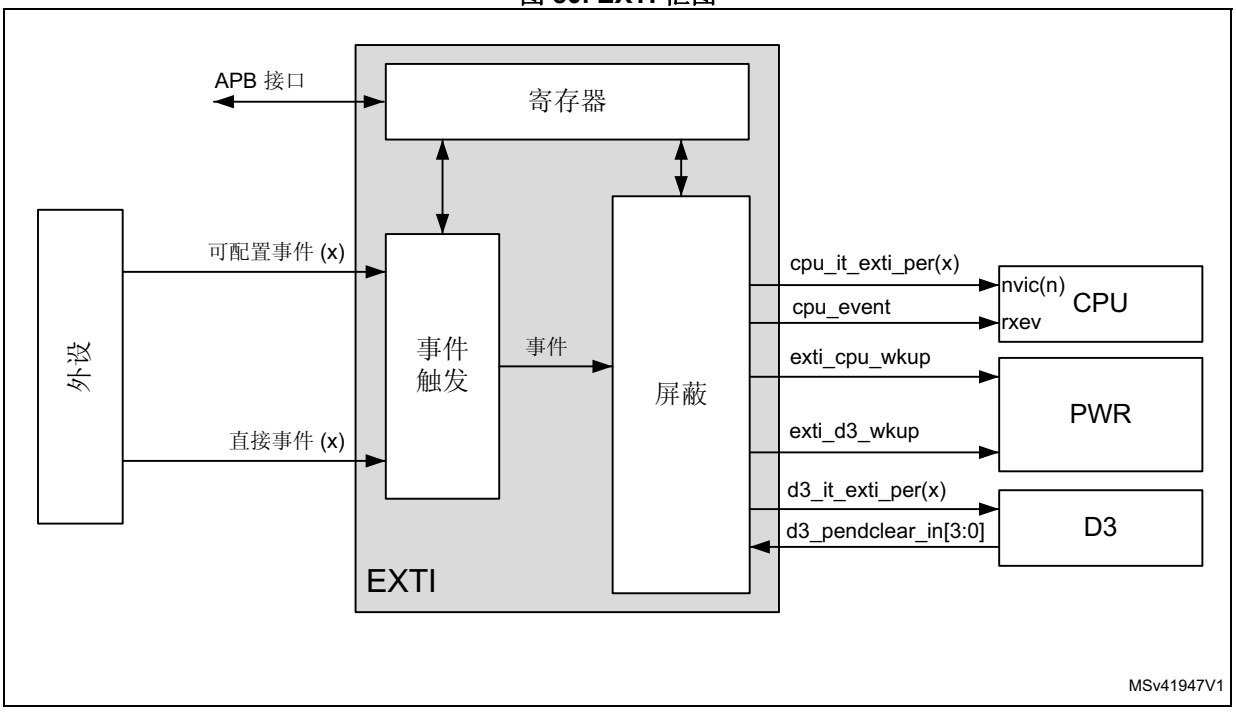
如*图 80* 所示，EXTI 包含一个通过 APB 接口访问的寄存器模块、一个事件输入触发模块和一个屏蔽模块。

寄存器模块包含所有 EXTI 寄存器。

事件输入触发模块提供事件输入边沿触发逻辑。

屏蔽模块为不同的唤醒、中断和事件输出及其屏蔽功能提供事件输入分配。

图 80. EXTI 框图



20.2.1 外设、CPU 和 D3 域之间的 EXTI 连接

对于能够在系统处于停止模式或 CPU 处于 CStop 模式时生成唤醒事件的外设，将连接至 EXTI 可配置事件输入或直接事件输入：

- 生成脉冲的外设信号连接至 EXTI 可配置事件输入。对于这类事件，EXTI 提供的 CPU 状态挂起位必须清零。
- 对于必须在外设中清除的外设中断和唤醒源，将连接至 EXTI 直接事件输入。EXTI 中无 CPU 状态挂起位。中断或唤醒由外设中的 CPU 清除。

对于能够唤醒自主运行模式下的 D3 的事件输入，将收到 D3 域挂起请求函数，该请求必须清除。此清除请求由挂起清除选择所选择的信号进行处理。

CPU 中断连接至相应的 CPU NVIC，类似地，CPU 事件连接至 CPU rxev 输入。

EXTI 唤醒信号连接至 PWR 模块，用于唤醒 D3 域和/或 CPU。

D3 域中断允许系统触发用于 D3 域自主运行模式工作的事件。

20.3 EXTI 功能说明

根据 EXTI 事件输入类型和唤醒目标，将使用不同的逻辑实现。适用的功能通过寄存器位进行控制：

- 有效触发边沿使能，具体包括 *EXTI 上升沿触发选择寄存器 (EXTI_RTSR1)*、*EXTI 上升沿触发选择寄存器 (EXTI_RTSR2)*、*EXTI 上升沿触发选择寄存器 (EXTI_RTSR3)* 和 *EXTI 下降沿触发选择寄存器 (EXTI_FTSR1)*、*EXTI 下降沿触发选择寄存器 (EXTI_FTSR2)*、*EXTI 下降沿触发选择寄存器 (EXTI_FTSR3)*
- 软件触发，具体包括 *EXTI 软件中断事件寄存器 (EXTI_SWIER1)*、*EXTI 软件中断事件寄存器 (EXTI_SWIER2)*、*EXTI 软件中断事件寄存器 (EXTI_SWIER3)*
- CPU 中断使能，具体包括 *EXTI 中断屏蔽寄存器 (EXTI_CPUIMR1)*、*EXTI 中断屏蔽寄存器 (EXTI_CPUIMR2)*、*EXTI 中断屏蔽寄存器 (EXTI_CPUIMR3)*
- CPU 事件使能，具体包括 *EXTI 事件屏蔽寄存器 (EXTI_CPUEMR1)*、*EXTI 事件屏蔽寄存器 (EXTI_CPUEMR2)*、*EXTI 事件屏蔽寄存器 (EXTI_CPUEMR3)*
- D3 域唤醒挂起，具体包括 *EXTI D3 挂起屏蔽寄存器 (EXTI_D3PMR1)*、*EXTI D3 挂起屏蔽寄存器 (EXTI_D3PMR2)*、*EXTI D3 挂起屏蔽寄存器 (EXTI_D3PMR3)*

表 130. EXTI 事件输入配置和寄存器控制(1)

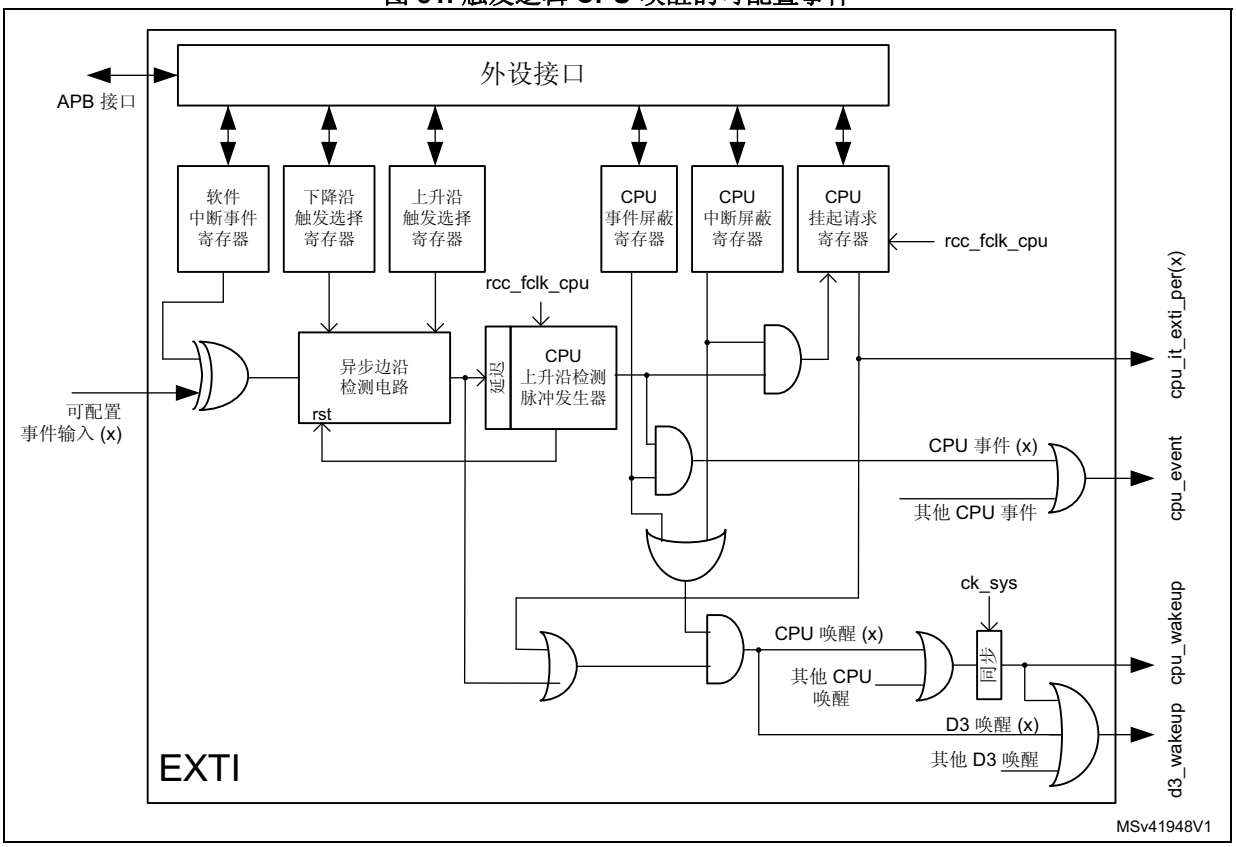
事件输入类型	唤醒目标	逻辑实现	EXTI_RTSR	EXTI_FTSR	EXTI_SWIER	EXTI_CPUIMR	EXTI_CPUEMR	EXTI_D3PMR
可配置	CPU	可配置事件输入，CPU 唤醒逻辑	X	X	X	X	X	-
	任意(2)	可配置事件输入，任意唤醒逻辑						X
直接	CPU	直接事件输入，CPU 唤醒逻辑	-	-	-	X	X	-
	任意(2)	直接事件输入，任意唤醒逻辑						X

1. X 表示功能可用。
2. 唤醒自主运行模式下的 D3 域和/或 CPU。

20.3.1 EXTI 可配置事件输入 CPU 唤醒

图 82 所示为始终将唤醒 CPU 的可配置事件输入的相关逻辑详图。

图 81. 触发逻辑 CPU 唤醒的可配置事件



软件中断事件寄存器允许系统通过软件触发可配置事件，具体方法是写入 *EXTI 软件中断事件寄存器 (EXTI_SWIER1)*、*EXTI 软件中断事件寄存器 (EXTI_SWIER2)* 或 *EXTI 软件中断事件寄存器 (EXTI_SWIER3)* 寄存器位。

上升沿 *EXTI 上升沿触发选择寄存器 (EXTI_RTSR1)*、*EXTI 上升沿触发选择寄存器 (EXTI_RTSR2)*、*EXTI 上升沿触发选择寄存器 (EXTI_RTSR3)* 和下降沿 *EXTI 下降沿触发选择寄存器 (EXTI_FTSR1)*、*EXTI 下降沿触发选择寄存器 (EXTI_FTSR2)*、*EXTI 下降沿触发选择寄存器 (EXTI_FTSR3)* 选择寄存器允许系统使能和选择可配置事件有效触发边沿或两种边沿。

器件具有专用中断屏蔽寄存器（*EXTI 中断屏蔽寄存器 (EXTI_CPUIMR1)*、*EXTI 中断屏蔽寄存器 (EXTI_CPUIMR2)* 和 *EXTI 中断屏蔽寄存器 (EXTI_CPUIMR3)*）以及可配置事件挂起请求寄存器（*EXTI 挂起寄存器 (EXTI_CPUPR1)*、*EXTI 挂起寄存器 (EXTI_CPUPR2)* 和 *EXTI 挂起寄存器 (EXTI_CPUPR3)*。CPU 挂起寄存器只会针对未屏蔽的 CPU 中断而置位。每个事件均为 CPU NVIC 提供单独的 CPU 中断。可配置事件中断需要由软件在 EXTI_CPUPR 寄存器中进行确认。

器件具有专用事件屏蔽寄存器，即 *EXTI 事件屏蔽寄存器 (EXTI_CPUEMR1)*、*EXTI 事件屏蔽寄存器 (EXTI_CPUEMR2)* 和 *EXTI 事件屏蔽寄存器 (EXTI_CPUEMR3)*。使能的事件随即会在 CPU 上生成一个事件。CPU 的所有事件均在一个 CPU 事件信号中进行逻辑或运算。CPU 挂起寄存器 (EXTI_CPUPR) 不会针对未屏蔽的 CPU 事件而置位。

使能 CPU 中断或 CPU 事件时，异步边沿检测电路由时钟延迟和上升沿检测脉冲发生器复位。这可以确保 CPU 时钟在异步边沿检测电路复位之前唤醒。

注： *只有在 CPU 唤醒时，才会将检测到的可配置事件（由 CPU 使能）清除。*

20.3.2 EXTI 可配置事件输入任意唤醒

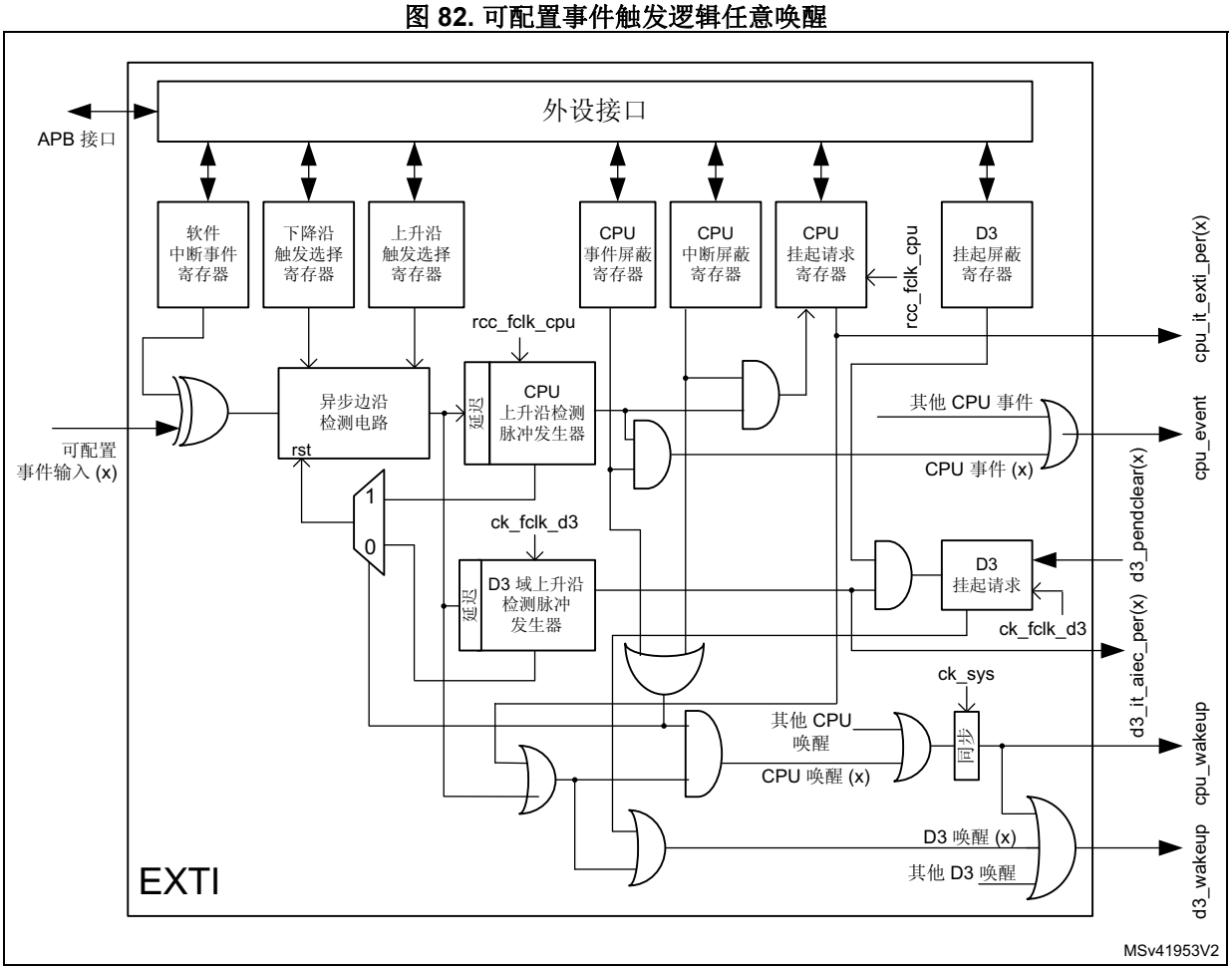
图 82 所示为能够唤醒自主运行模式下的 D3 域和/或 CPU（“任意”目标）的可配置事件输入的相关逻辑详图。其功能与可配置事件输入 CPU 唤醒的功能基本相同，只是新增了单独唤醒 D3 域的功能。

禁止所有 CPU 中断和 CPU 事件时，异步边沿检测电路由 D3 域时钟延迟和上升沿检测脉冲发生器复位。这可以确保 D3 域时钟在异步边沿检测电路复位之前唤醒。

表 131. 可配置事件输入异步边沿检测器复位

EXTI_C1IMR	EXTI_C1EMR	异步边沿检测器复位源
均为 0		D3 域时钟上升沿检测脉冲发生器
至少一个为 1		CPU 时钟上升沿检测脉冲发生器

图 82. 可配置事件触发逻辑任意唤醒



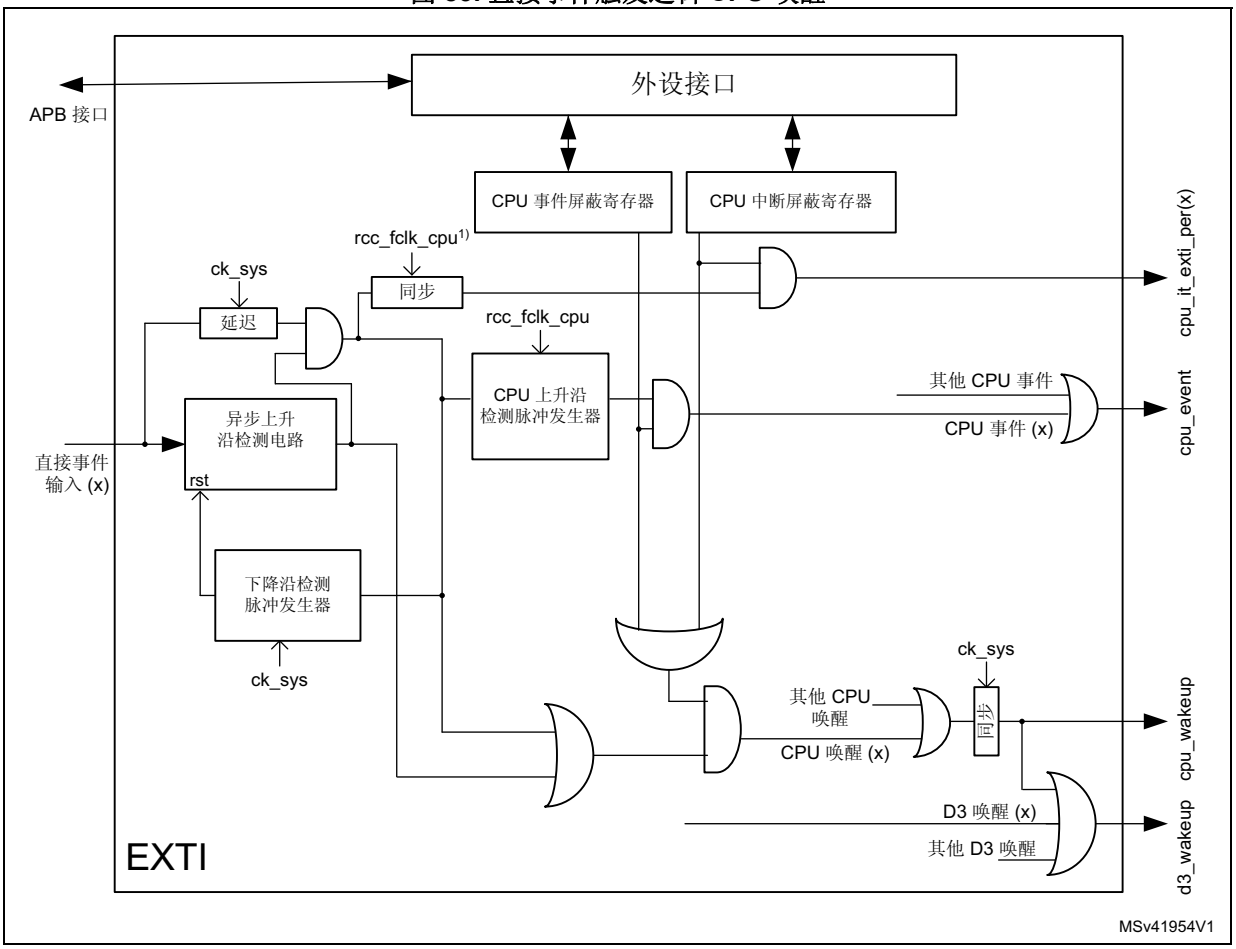
“任意”目标的事件触发逻辑额外配有 D3 挂起屏蔽寄存器 *EXTI D3 挂起屏蔽寄存器 (EXTI_D3PMR1)*、*EXTI D3 挂起屏蔽寄存器 (EXTI_D3PMR2)*、*EXTI D3 挂起屏蔽寄存器 (EXTI_D3PMR3)* 和 D3 挂起请求逻辑。D3 挂起请求逻辑只会针对未屏蔽的 D3 挂起事件而置位。D3 挂起请求逻辑使 D3 域保持在运行模式下，直到所选 D3 域挂起清除源将 D3 挂起请求逻辑清除。

20.3.3 EXTI 直接事件输入 CPU 唤醒

图 83 所示为唤醒 CPU 的直接事件输入的相关逻辑详图。

直接事件仅具有 CPU 中断使能和 CPU 事件使能功能。

图 83. 直接事件触发逻辑 CPU 唤醒

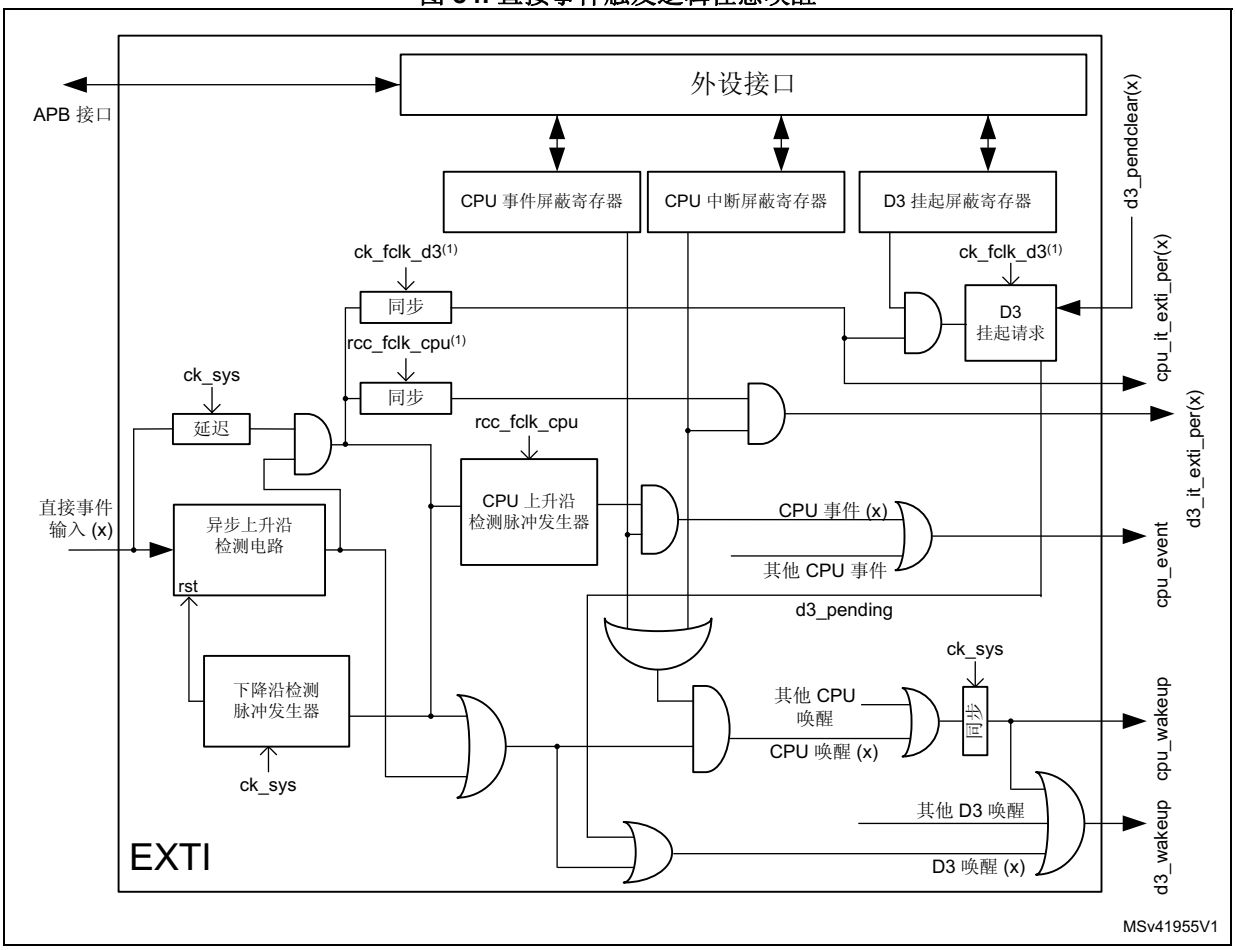


1. 用于异步直接事件输入（外设唤醒信号）的 CPU 中断与 CPU 时钟同步。在异步边沿检测之后，同步直接事件输入（外设中断信号）直接发送至 CPU 中断，无需重新同步。

20.3.4 EXTI 直接事件输入任意唤醒

图 84 所示为能够唤醒自主运行模式下的 D3 域和/或 CPU（“任意”目标）的直接事件输入的相关逻辑详图。其功能与直接事件输入 CPU 唤醒的功能基本相同，只是新增了单独唤醒 D3 域的功能。

图 84. 直接事件触发逻辑任意唤醒



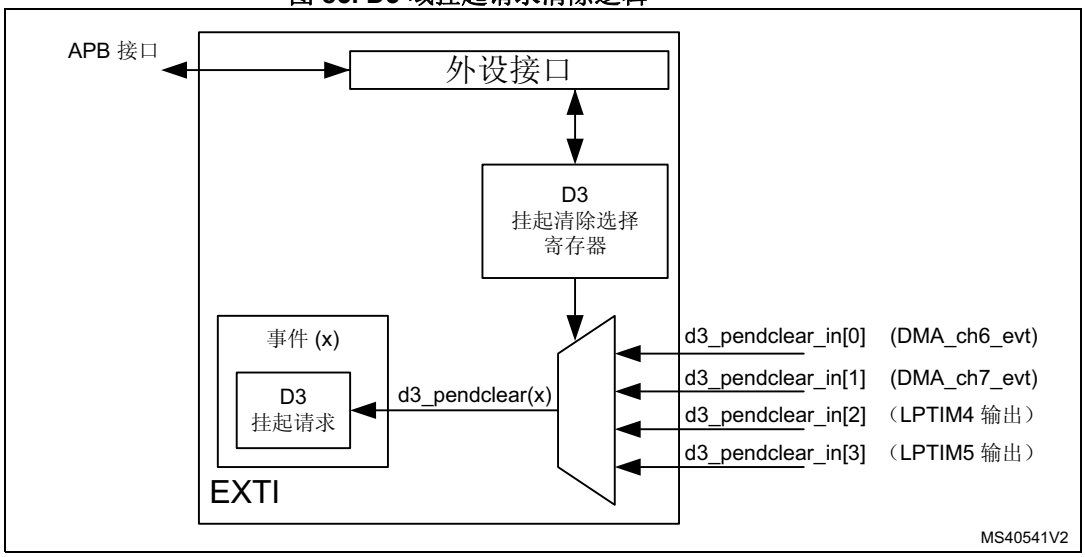
1. 用于异步直接事件输入（外设唤醒信号）的 CPU 中断和 D3 域中断分别与 CPU 时钟和 D3 域时钟同步。在异步边沿检测之后，同步直接事件输入（外设中断信号）直接发送至 CPU 中断和 D3 域中断，无需在 EXTI 中重新同步。

20.3.5 EXTI D3 挂起请求清除选择

对于能够唤醒自主运行模式下的 D3 域的事件输入，其具有的 D3 挂起请求逻辑可通过选定的 D3 挂起清除源来清除。对于每个 D3 挂起请求，可从四个不同的输入中选择 D3 域挂起清除源。

图 85 所示为选择 D3 挂起清除源的逻辑的详图。

图 85. D3 域挂起请求清除逻辑



D3 挂起请求清除选择寄存器 *EXTI D3 挂起清除选择寄存器低位字 (EXTI_D3PCR1L)*、*EXTI D3 挂起清除选择寄存器高位字 (EXTI_D3PCR1H)*、*EXTI D3 挂起清除选择寄存器低位字 (EXTI_D3PCR2L)*、*EXTI D3 挂起清除选择寄存器高位字 (EXTI_D3PCR2H)*、*EXTI D3 挂起清除选择寄存器低位字 (EXTI_D3PCR3L)* 和 *EXTI D3 挂起清除选择寄存器高位字 (EXTI_D3PCR3H)* 允许系统选择用于复位 D3 挂起请求的源。

20.4 EXTI 事件输入映射

对于十六个 GPIO 事件输入，必须在 SYSCFG 寄存器 SYSCFG_EXTICRn 中选择相关 IOPORT 引脚。来自各个 IOPORT 的相同引脚会映射到相应的 EXTI 事件输入。

表 132 对各个事件输入的唤醒功能进行了详细介绍。事件输入可唤醒 CPU，对于“任意”目标，还可唤醒自主运行模式下的 D3 域。

连接至 NVIC 列给出了与 CPU NVIC 相连的 EXTI 事件输入。对于未连接至 NVIC 的 EXTI 事件，外设中断将直接连接至 NVIC（同时还与 EXTI 相连）。

所有 EXTI 事件输入均进行逻辑或运算，并连接至 CPU 事件输入 (rxev)。

表 132. EXTI 事件输入映射

事件输入	源	事件输入类型	唤醒目标	连接至 NVIC
0 - 15	EXTI[15:0]	可配置	任意	有
16	PVD 和 AVD(1)	可配置	仅限 CPU	有
17	RTC 闹钟	可配置	仅限 CPU	有
18	RTC 入侵、RTC 时间戳、RCC LSECSS(2)	可配置	仅限 CPU	有
19	RTC 唤醒定时器	可配置	任意	有
20	COMP1	可配置	任意	有
21	COMP2	可配置	任意	有
22	I2C1 唤醒	直接	仅限 CPU	有

表 132. EXTI 事件输入映射（续）

事件输入	源	事件输入类型	唤醒目标	连接至 NVIC
23	I2C2 唤醒	直接	仅限 CPU	有
24	I2C3 唤醒	直接	仅限 CPU	有
25	I2C4 唤醒	直接	任意	有
26	USART1 唤醒	直接	仅限 CPU	有
27	USART2 唤醒	直接	仅限 CPU	有
28	USART3 唤醒	直接	仅限 CPU	有
29	USART6 唤醒	直接	仅限 CPU	有
30	UART4 唤醒	直接	仅限 CPU	有
31	UART5 唤醒	直接	仅限 CPU	有
32	UART7 唤醒	直接	仅限 CPU	有
33	UART8 唤醒	直接	仅限 CPU	有
34	LPUART1 RX 唤醒	直接	任意	有
35	LPUART1 TX 唤醒	直接	任意	有
36	SPI1 唤醒	直接	仅限 CPU	有
37	SPI2 唤醒	直接	仅限 CPU	有
38	SPI3 唤醒	直接	仅限 CPU	有
39	SPI4 唤醒	直接	仅限 CPU	有
40	SPI5 唤醒	直接	仅限 CPU	有
41	SPI6 唤醒	直接	任意	有
42	MDIO 唤醒	直接	仅限 CPU	有
43	USB1 唤醒	直接	仅限 CPU	有
44	USB2 唤醒	直接	仅限 CPU	有
45	保留	-	-	-
46	保留	-	-	-
47	LPTIM1 唤醒	直接	仅限 CPU	有
48	LPTIM2 唤醒	直接	任意	有
49	LPTIM2 输出	可配置	任意	无(3)
50	LPTIM3 唤醒	直接	任意	有
51	LPTIM3 输出	可配置	任意	无(3)
52	LPTIM4 唤醒	直接	任意	有
53	LPTIM5 唤醒	直接	任意	有
54	SWPMI 唤醒	直接	仅限 CPU	有
55	WKUP1	直接	仅限 CPU	有
56	WKUP2	直接	仅限 CPU	有
57	WKUP3	直接	仅限 CPU	有

表 132. EXTI 事件输入映射（续）

事件输入	源	事件输入类型	唤醒目标	连接至 NVIC
58	WKUP4	直接	仅限 CPU	有
59	WKUP5	直接	仅限 CPU	有
60	WKUP6	直接	仅限 CPU	有
61	RCC 中断	直接	仅限 CPU	无(4)
62	I2C4 事件中断	直接	仅限 CPU	无(4)
63	I2C4 错误中断	直接	仅限 CPU	无(4)
64	LPUART1 全局中断	直接	仅限 CPU	无(4)
65	SPI6 中断	直接	仅限 CPU	无(4)
66	BDMA CH0 中断	直接	仅限 CPU	无(4)
67	BDMA CH1 中断	直接	仅限 CPU	无(4)
68	BDMA CH2 中断	直接	仅限 CPU	无(4)
69	BDMA CH3 中断	直接	仅限 CPU	无(4)
70	BDMA CH4 中断	直接	仅限 CPU	无(4)
71	BDMA CH5 中断	直接	仅限 CPU	无(4)
72	BDMA CH6 中断	直接	仅限 CPU	无(4)
73	BDMA CH7 中断	直接	仅限 CPU	无(4)
74	DMAMUX2 中断	直接	仅限 CPU	无(4)
75	ADC3 中断	直接	仅限 CPU	无(4)
76	SAI4 中断	直接	仅限 CPU	无(4)
77	保留	-	-	-
78	保留	-	-	-
79	保留	-	-	-
80	保留	-	-	-
81	保留	-	-	-
82	保留	-	-	-
83	保留	-	-	-
84	保留	-	-	-
85	HDMI-CEC 唤醒	可配置	仅限 CPU	有
86	ETHERNET 唤醒	可配置	仅限 CPU	有
87	HSECSS 中断	直接	仅限 CPU	无(4)
88	保留	-	-	-

1. PVD 和 AVD 信号在同一 EXTI 事件输入上进行逻辑或运算。
2. RTC 入侵、RTC 时间戳和 RCC LSECSS 信号在同一 EXTI 事件输入上进行逻辑或运算。
3. 在 CPU NVIC 上不可用，仅适用于系统唤醒或 CPU 事件输入 (rxev)。
4. 在 CPU NVIC 上可用，直接由外设提供

20.5 EXTI 功能行为

在生成事件的相应外设中使能直接事件输入。通过使能至少一个触发沿来使能可配置事件。

在停止模式下，事件始终会唤醒 D3 域。在系统运行和停止模式下，事件始终会生成一个相关 D3 域中断。只有在事件相关的 CPU 中断未屏蔽和/或 CPU 事件未屏蔽时，事件才会唤醒 CPU。

表 133. 屏蔽功能

CPU		可配置事件输入（EXTI_CPUPR 的 PRx 位）	CPU			D3 域唤醒
中断使能（EXTI_CPUIMR 的 MRx 位）	事件使能（EXTI_CPUEMR 的 MRx 位）		中断	Event	唤醒	
0	0	无	已屏蔽	已屏蔽	已屏蔽	是(1)/已屏蔽(2)
0	1	无	已屏蔽	有	有	有
1	0	状态已锁存	有	已屏蔽	有	有
1	1	状态已锁存	有	有	有	有

1. 仅限允许系统唤醒自主运行模式下的 D3 域（任意目标）的事件输入。
2. 适用于将始终唤醒 CPU 的事件输入。

对于可配置事件输入，当事件输入上出现使能的边沿时，会生成一个事件请求。相关 CPU 中断未屏蔽时，EXTI_CPUPR 的相应挂起 PRx 位会置 1，CPU 中断信号会激活。EXTI_CPUPR PRx 挂起位应通过软件写入“1”来清零。这会清除 CPU 中断。

对于直接事件输入，当在相关外设中使能时，只会在上升沿生成事件请求。不存在相应的 CPU 挂起位。当相关 CPU 中断未被屏蔽时，会激活相应的 CPU 中断信号。

要生成事件，CPU 事件必须处于未屏蔽状态。当事件输入上出现使能的边沿时，会生成 CPU 事件脉冲。不存在 CPU 事件挂起位。

CPU 中断和 CPU 事件可在同一事件输入上使能。它们会触发相同的事件输入条件。

对于可配置事件输入，可通过软件在软件中断/事件寄存器 EXTI_SWIER 中写入“1”来生成事件输入请求。

如果事件输入已使能且 CPU 中断和/或 CPU 事件未被屏蔽，则事件输入还会在 CPU 唤醒信号紧接着生成一个 D3 域唤醒信号。

某些事件输入能够唤醒自主运行模式下的 D3 域，在这种情况下，CPU 中断和 CPU 事件会被屏蔽，进而阻止 CPU 被唤醒。支持两种 D3 域自主运行模式唤醒机制：

- 无挂起 的 D3 域唤醒 (EXTI_D3PMR = 0)
  - 对于可配置事件输入，该机制会在延迟 + 上升沿检测脉冲发生器之后自动唤醒 D3 域以及清除 D3 域唤醒信号。
  - 对于直接事件输入，该机制会在直接事件输入信号清除后唤醒 D3 域以及清除 D3 域唤醒信号。
- 存在挂起 的 D3 域唤醒 (EXTI_D3PMR = 1)
  - 对于可配置事件输入，该机制会在延迟 + 上升沿检测脉冲发生器之后清除 D3 挂起请求时唤醒 D3 域以及清除 D3 域唤醒信号。
  - 对于直接事件输入，该机制会在直接事件输入信号清除后以及 D3 挂起请求清除时，唤醒 D3 域以及清除 D3 域唤醒信号。

20.5.1 EXTI CPU 中断程序

- 通过将 EXTI_CPUIMR 寄存器的相应屏蔽位置 1 对事件输入中断取消屏蔽。
- 对于可配置事件输入，通过将 EXTI_RTSR 和 EXTI_FTSR 寄存器中的一个或两个相应触发边沿使能位置 1 来使能事件输入。
- 使能 CPU NVIC 中的相关中断源或使用 SEVONPEND，以便 CPU 在 WFI/WFE 指令后可检测到来自 CPU 中断信号的中断。
  - 对于可配置事件输入，需要将相关 EXTI 挂起位清零。

20.5.2 EXTI CPU 事件程序

- 通过将 EXTI_CPUEMR 寄存器的相应屏蔽位置 1 对事件输入取消屏蔽。
- 对于可配置事件输入，通过将 EXTI_RTSR 和 EXTI_FTSR 寄存器中的一个或两个相应触发边沿使能位置 1 来使能事件输入。
- CPU 会在 WFE 指令后检测到 CPU 事件信号。
  - 对于可配置事件输入，不存在要清零的 EXTI 挂起位。

20.5.3 EXTI CPU 唤醒程序

- 通过将 EXTI_CPUIMR 和/或 EXTI_CPUEMR 寄存器中的至少一个相应屏蔽位置 1，以对事件输入取消屏蔽。CPU 唤醒与未屏蔽的 CPU 中断和/或 CPU 事件同时生成。
- 对于可配置事件输入，通过将 EXTI_RTSR 和 EXTI_FTSR 寄存器中的一个或两个相应触发边沿使能位置 1 来使能事件输入。
- 直接事件将自动生成 CPU 唤醒。

20.5.4 自主运行模式下的 EXTI D3 域唤醒程序

- 通过将 EXTI_CPUIMR 和/或 EXTI_CPUEMR 寄存器中的两个相应屏蔽位清零，以屏蔽用于唤醒 CPU 的事件输入。
- 对于可配置事件输入，通过将 EXTI_RTSR 和 EXTI_FTSR 寄存器中的一个或两个相应触发边沿使能位置 1 来使能事件输入。
- 直接事件将自动生成 D3 域唤醒。
- 在 EXTI_D3PMR 中选择 D3 域唤醒机制。
  - 如果选择无挂起 的 D3 域唤醒 (EXTI_PMR = 0)，则在清除事件输入后会自动清除唤醒。
  - 如果选择存在挂起 的 D3 域唤醒 (EXTI_PMR = 1)，则需要由所选 D3 域挂起清除源来清除唤醒。
    此外，也可通过 FW 将相关 EXTI_D3PMR 寄存器位清零来清除挂起的 D3 域唤醒信号。
- 在 D3 域唤醒后，会生成 D3 域中断。
  - 可配置事件输入将在 D3 域中断时生成脉冲。
  - 直接事件输入将激活 D3 域中断，直到在外设中清除事件输入。

20.5.5 EXTI 软件中断/事件触发程序

所有可配置事件输入均可通过软件中断/事件寄存器触发（相关 CPU 中断和/或 CPU 事件应通过其相应的程序进行使能）。

- 通过将 EXTI_RTSR 和/或 EXTI_FTSR 寄存器中的至少一个相应边沿触发位置 1 来使能事件输入。
- 通过将 EXTI_CPUIMR 和/或 EXTI_CPUEMR 寄存器中的至少一个相应屏蔽位置 1，以对软件中断/事件触发取消屏蔽。
- 通过向 EXTI_SWIER 寄存器中的相应位写入“1”来触发软件中断/事件。
- 可通过将 EXTI_RTSR 和 EXTI_FTSR 寄存器位清零来禁止事件输入。

注： *可配置事件输入上的边沿也会触发中断/事件。*

软件触发可用于将 D3 挂起请求逻辑置 1，从而使 D3 域保持在运行模式下，直到 D3 挂起请求逻辑清零。

20.6 EXTI 寄存器说明

每个寄存器仅支持 32 位（字）访问， 不支持按字节或半字进行读写操作。

20.6.1 EXTI 上升沿触发选择寄存器 (EXTI_RTSR1)

EXTI rising trigger selection register

偏移地址：0x00
复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TR21	TR20	TR19	TR18	TR17	TR16
										rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TR15	TR14	TR13	TR12	TR11	TR10	TR9	TR8	TR7	TR6	TR5	TR4	TR3	TR2	TR1	TR0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:22 保留，必须保持复位值。

位 21:0 **TRx**：可配置事件输入 x 的上升沿触发事件配置位 (Rising trigger event configuration bit of Configurable Event input x)(1)

0：禁止输入线上升沿触发（事件和中断）
1：允许输入线上升沿触发（事件和中断）

1. 可配置事件输入为边沿触发，在这些输入上不能出现毛刺信号。
如果在对寄存器执行写操作时可配置事件输入上出现上升沿，则相关挂起位不会被置 1。
在同一可配置事件输入上，可同时设置上升沿和下降沿触发。在这种情况下，两种边沿均会生成触发信号。

20.6.2 EXTI 下降沿触发选择寄存器 (EXTI_FTSR1)

EXTI falling trigger selection register

偏移地址：0x04
复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TR21	TR20	TR19	TR18	TR17	TR16
										rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TR15	TR14	TR13	TR12	TR11	TR10	TR9	TR8	TR7	TR6	TR5	TR4	TR3	TR2	TR1	TR0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:22 保留，必须保持复位值。

位 21:0 **TRx**：可配置事件输入 x 的下降沿触发事件配置位 (Falling trigger event configuration bit of Configurable Event input x)(1)

0：禁止输入线下降沿触发（事件和中断）
1：允许输入线下降沿触发（事件和中断）。

1. 可配置事件输入为边沿触发，在这些输入上不能出现毛刺信号。
如果在对寄存器执行写操作时可配置事件输入上出现下降沿，则相关挂起位不会被置 1。
在同一可配置事件输入上，可同时设置上升沿和下降沿触发。在这种情况下，两种边沿均会生成触发信号。

20.6.3 EXTI 软件中断事件寄存器 (EXTI_SWIER1)

EXTI software interrupt event register

偏移地址：0x08
复位值：0x0000 0000

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

位 31:22 保留，必须保持复位值。

位 21:0 **SWIERx**：线 x 上的软件中断 (Software Interrupt on line x)
被读取时将始终返回 0。

0：写入 0 无影响。
1：向该位写入 1 会触发线 x 上的事件。此位由硬件自动清零。

20.6.4 EXTI D3 挂起屏蔽寄存器 (EXTI_D3PMR1)

EXTI D3 pending mask register

偏移地址：0x0C
复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	MR25	Res.	Res.	Res.	MR21	MR20	MR19	Res.	Res.	Res.
						rw				rw	rw	rw			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MR15	MR14	MR13	MR12	MR11	MR10	MR9	MR8	MR7	MR6	MR5	MR4	MR3	MR2	MR1	MR0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:26 保留，必须保持复位值。

位 25 **MRx**：事件输入 x 上的 D3 挂起屏蔽 (D3 Pending Mask on Event input x)
0：屏蔽来自线 x 的 D3 挂起请求。向该位写入 0 也会清除 D3 挂起请求。
1：取消屏蔽来自线 x 的 D3 挂起请求。D3 域挂起信号在触发时会使 D3 域唤醒保持激活，直至其被清除。

位 24:22 保留，必须保持复位值。

位 21:19 **MRx**：事件输入 x 上的 D3 挂起屏蔽 (D3 Pending Mask on Event input x)
0：屏蔽来自线 x 的 D3 挂起请求。向该位写入 0 也会清除 D3 挂起请求。
1：取消屏蔽来自线 x 的 D3 挂起请求。D3 域挂起信号在触发时会使 D3 域唤醒保持激活，直至其被清除。

位 18:16 保留，必须保持复位值。

位 15:0 **MRx**：事件输入 x 上的 D3 挂起屏蔽 (D3 Pending Mask on Event input x)
0：屏蔽来自线 x 的 D3 挂起请求。向该位写入 0 也会清除 D3 挂起请求。
1：取消屏蔽来自线 x 的 D3 挂起请求。D3 域挂起信号在触发时会使 D3 域唤醒保持激活，直至其被清除。

20.6.5 EXTI D3 挂起清除选择寄存器低位字 (EXTI_D3PCR1L)

EXTI D3 pending clear selection register low

偏移地址：0x10
复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PCS15		PCS14		PCS13		PCS12		PCS11		PCS10		PCS9		PCS8	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PCS7		PCS6		PCS5		PCS4		PCS3		PCS2		PCS1		PCS0	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:0 **PCSx**：事件输入 x = truncate (n/2) 上的 D3 挂起请求清除输入信号选择 (D3 Pending request clear input signal selection on Event input x = truncate (n/2))

00：选择 DMA ch6 事件作为 D3 域挂起清除源
01：选择 DMA ch7 事件作为 D3 域挂起清除源
10：选择 LPTIM4 输出作为 D3 域挂起清除源
11：选择 LPTIM5 输出作为 D3 域挂起清除源

20.6.6 EXTI D3 挂起清除选择寄存器高位字 (EXTI_D3PCR1H)

EXTI D3 pending clear selection register high

偏移地址：0x14
复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PCS25		Res.	Res.
												rw	rw		
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	PCS21		PCS20		PCS19		Res.	Res.	Res.	Res.	Res.	Res.
				rw	rw	rw	rw	rw	rw						

位 31:20 保留，必须保持复位值。

位 19:18 **PCSx**：事件输入 x = truncate ((n+32)/2) 上的 D3 挂起请求清除输入信号选择 (D3 Pending request clear input signal selection on Event input x = truncate ((n+32)/2))

00：选择 DMA ch6 事件作为 D3 域挂起清除源
01：选择 DMA ch7 事件作为 D3 域挂起清除源
10：选择 LPTIM4 输出作为 D3 域挂起清除源
11：选择 LPTIM5 输出作为 D3 域挂起清除源

位 17:12 保留，必须保持复位值。

位 11:6 **PCSx**：事件输入 x = truncate ((n+32)/2) 上的 D3 挂起请求清除输入信号选择 (D3 Pending request clear input signal selection on Event input x = truncate ((n+32)/2))

00：选择 DMA ch6 事件作为 D3 域挂起清除源
01：选择 DMA ch7 事件作为 D3 域挂起清除源
10：选择 LPTIM4 输出作为 D3 域挂起清除源
11：选择 LPTIM5 输出作为 D3 域挂起清除源

位 5:0 保留，必须保持复位值。

20.6.7 EXTI 上升沿触发选择寄存器 (EXTI_RTSR2)

EXTI rising trigger selection register

偏移地址：0x20
复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TR51	Res.	TR49	Res.
												rw		rw	

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

位 31:20 保留，必须保持复位值。

位 19 **TRx**：可配置事件输入 x+32 的上升沿触发事件配置位 (Rising trigger event configuration bit of Configurable Event input x+32)(1)

0：禁止输入线上升沿触发（事件和中断）
1：允许输入线上升沿触发（事件和中断）

位 18 保留，必须保持复位值。

位 17 **TRx**：可配置事件输入 x+32 的上升沿触发事件配置位 (Rising trigger event configuration bit of Configurable Event input x+32)(1)

0：禁止输入线上升沿触发（事件和中断）
1：允许输入线上升沿触发（事件和中断）

位 16:0 保留，必须保持复位值。

1. 可配置的事件输入为边沿触发，在这些输入上不能出现毛刺信号。
如果在对寄存器执行写操作时可配置事件输入上出现上升沿，则相关挂起位不会被置 1。
在同一可配置事件输入上，可同时设置上升沿和下降沿触发。在这种情况下，两种边沿均会生成触发信号。

20.6.8 EXTI 下降沿触发选择寄存器 (EXTI_FTSR2)

EXTI falling trigger selection register

偏移地址：0x24
复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TR51	Res.	TR49	Res.
												rw		rw	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

位 31:20 保留，必须保持复位值。

位 19 **TRx**：可配置事件输入 x+32 的下降沿触发事件配置位 (Falling trigger event configuration bit of Configurable Event input x+32)(1)

0：禁止输入线下降沿触发（事件和中断）
1：允许输入线下降沿触发（事件和中断）。

位 18 保留，必须保持复位值。

位 17 **TRx**：可配置事件输入 x+32 的下降沿触发事件配置位 (Falling trigger event configuration bit of Configurable Event input x+32)(1)

0：禁止输入线下降沿触发（事件和中断）
1：允许输入线下降沿触发（事件和中断）。

位 16:0 保留，必须保持复位值。

1. 可配置的事件输入为边沿触发，在这些输入上不能出现毛刺信号。
如果在对寄存器执行写操作时可配置事件输入上出现下降沿，则相关挂起位不会被置 1。
在同一可配置事件输入上，可同时设置上升沿和下降沿触发。在这种情况下，两种边沿均会生成触发信号。

20.6.9 EXTI 软件中断事件寄存器 (EXTI_SWIER2)

EXTI software interrupt event register

偏移地址：0x28
复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SWIER 51	Res.	SWIER 49	Res.
												rw		rw	

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

位 31:20 保留，必须保持复位值。

位 19 **SWIERx**：线 x+32 上的软件中断 (Software interrupt on line x+32)
被读取时将始终返回 0。
0：写入 0 无影响。
1：向该位写入 1 会触发线 x 上的事件。此位由硬件自动清零。

位 18 保留，必须保持复位值。

位 17 **SWIERx**：线 x+32 上的软件中断 (Software interrupt on line x+32)
被读取时将始终返回 0。
0：写入 0 无影响。
1：向该位写入 1 会触发线 x 上的事件。此位由硬件自动清零。

位 16:0 保留，必须保持复位值。

20.6.10 EXTI D3 挂起屏蔽寄存器 (EXTI_D3PMR2)

EXTI D3 pending mask register

偏移地址：0x2C
复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MR53	MR52	MR51	MR50	MR49	MR48
										rw	rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	MR41	Res.	Res.	Res.	Res.	Res.	MR35	MR34	Res.	Res.
						rw						rw	rw		

位 31:22 保留，必须保持复位值。

位 21:16 **MRx**：事件输入 x+32 上的 D3 挂起屏蔽 (D3 Pending Mask on Event input x+32)
0：屏蔽来自线 x+32 的 D3 挂起请求。向该位写入 0 也会清除 D3 挂起请求。
1：取消屏蔽来自线 x+32 的 D3 挂起请求。D3 域挂起信号在触发时会使 D3 域唤醒保持激活，直至其被清除。

位 15:10 保留，必须保持复位值。

位 9 **MRx**：事件输入 x+32 上的 D3 挂起屏蔽 (D3 Pending Mask on Event input x+32)
0：屏蔽来自线 x+32 的 D3 挂起请求。向该位写入 0 也会清除 D3 挂起请求。
1：取消屏蔽来自线 x+32 的 D3 挂起请求。D3 域挂起信号在触发时会使 D3 域唤醒保持激活，直至其被清除。

位 8:4 保留，必须保持复位值。

位 3:2 **MRx**：事件输入 x+32 上的 D3 挂起屏蔽 (D3 Pending Mask on Event input x+32)
0：屏蔽来自线 x+32 的 D3 挂起请求。向该位写入 0 也会清除 D3 挂起请求。
1：取消屏蔽来自线 x+32 的 D3 挂起请求。D3 域挂起信号在触发时会使 D3 域唤醒保持激活，直至其被清除。

位 1:0 保留，必须保持复位值。

20.6.11 EXTI D3 挂起清除选择寄存器低位字 (EXTI_D3PCR2L)

EXTI D3 pending clear selection register low

偏移地址：0x30
复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PCS41		Res.	Res.
												rw	rw		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PCS35		PCS34		Res.	Res.	Res.	Res.
								rw	rw	rw	rw				

位 31:20 保留，必须保持复位值。

位 19:18 **PCSx**：事件输入 x = truncate ((n+64)/2) 上的 D3 挂起请求清除输入信号选择 (D3 Pending request clear input signal selection on Event input x = truncate ((n+64)/2)
00：选择 DMA ch6 事件作为 D3 域挂起清除源
01：选择 DMA ch7 事件作为 D3 域挂起清除源
10：选择 LPTIM4 输出作为 D3 域挂起清除源
11：选择 LPTIM5 输出作为 D3 域挂起清除源

位 17:8 保留，必须保持复位值。

位 7:4 **PCSx**：事件输入 x = truncate ((n+64)/2) 上的 D3 挂起请求清除输入信号选择 (D3 Pending request clear input signal selection on Event input x= truncate ((n+64)/2))
00：选择 DMA ch6 事件作为 D3 域挂起清除源
01：选择 DMA ch7 事件作为 D3 域挂起清除源
10：选择 LPTIM4 输出作为 D3 域挂起清除源
11：选择 LPTIM5 输出作为 D3 域挂起清除源

位 3:0 保留，必须保持复位值。

20.6.12 EXTI D3 挂起清除选择寄存器高位字 (EXTI_D3PCR2H)

EXTI D3 pending clear selection register high

偏移地址：0x34
复位值：0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	PCS53		PCS52		PCS51		PCS50		PCS49		PCS48	
				rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:12 保留，必须保持复位值。

位 11:0 **PCSx**：事件输入 x = truncate ((n+96)/2) 上的 D3 挂起请求清除输入信号选择 (D3 Pending request clear input signal selection on Event input x= truncate ((n+96)/2))

00：选择 DMA ch6 事件作为 D3 域挂起清除源
01：选择 DMA ch7 事件作为 D3 域挂起清除源
10：选择 LPTIM4 输出作为 D3 域挂起清除源
11：选择 LPTIM5 输出作为 D3 域挂起清除源

20.6.13 EXTI 上升沿触发选择寄存器 (EXTI_RTSR3)

EXTI rising trigger selection register

偏移地址：0x40
复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TR86	TR85	TR84	Res.	TR82	Res.	Res.
									rw	rw	rw		rw		

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

位 31:23 保留，必须保持复位值。

位 22:20 **TRx**：可配置事件输入 x+64 的上升沿触发事件配置位 (Rising trigger event configuration bit of Configurable Event input x+64)(1)

0：禁止输入线上升沿触发（事件和中断）
1：允许输入线上升沿触发（事件和中断）

位 19 保留，必须保持复位值。

位 18 **TRx**：可配置事件输入 x+64 的上升沿触发事件配置位 (Rising trigger event configuration bit of Configurable Event input x+64)(1)

0：禁止输入线上升沿触发（事件和中断）
1：允许输入线上升沿触发（事件和中断）

位 17:0 保留，必须保持复位值。

1. 可配置的事件输入为边沿触发，在这些输入上不能出现毛刺信号。
如果在对寄存器执行写操作时可配置事件输入上出现上升沿，则相关挂起位不会被置 1。
在同一可配置事件输入上，可同时设置上升沿和下降沿触发。在这种情况下，两种边沿均会生成触发信号。

20.6.14 EXTI 下降沿触发选择寄存器 (EXTI_FTSR3)

EXTI falling trigger selection register

偏移地址：0x44
复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TR86	TR85	TR84	Res.	TR82	Res.	Res.
									rw	rw	rw		rw		
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

位 31:23 保留，必须保持复位值。

位 22:20 **TRx**：可配置事件输入 x+64 的下降沿触发事件配置位 (Falling trigger event configuration bit of Configurable Event input x+64)(1)

0：禁止输入线下降沿触发（事件和中断）
1：允许输入线下降沿触发（事件和中断）

位 19 保留，必须保持复位值。

位 18 **TRx**：可配置事件输入 x+64 的下降沿触发事件配置位 (Falling trigger event configuration bit of Configurable Event input x+64)(1)

0：禁止输入线下降沿触发（事件和中断）
1：允许输入线下降沿触发（事件和中断）

位 17:0 保留，必须保持复位值。

1. 可配置的事件输入为边沿触发，在这些输入上不能出现毛刺信号。
如果在对寄存器执行写操作时可配置事件输入上出现下降沿，则相关挂起位不会被置 1。
在同一可配置事件输入上，可同时设置上升沿和下降沿触发。在这种情况下，两种边沿均会生成触发信号。

20.6.15 EXTI 软件中断事件寄存器 (EXTI_SWIER3)

EXTI software interrupt event register

偏移地址：0x48
复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SWIER86	SWIER85	SWIER84	Res.	SWIER82	Res.	Res.
									rw	rw	rw		rw		

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

位 31:23 保留，必须保持复位值。

位 22:20 **SWIERx**：线 x+64 上的软件中断 (Software interrupt on line x+64)
被读取时将始终返回 0。
0：写入 0 无影响。
1：向该位写入 1 会触发线 x 上的事件。此位由硬件自动清零。

位 19 保留，必须保持复位值。

位 18 **SWIERx**：线 x+64 上的软件中断 (Software interrupt on line x+64)
被读取时将始终返回 0。
0：写入 0 无影响。
1：向该位写入 1 会触发线 x 上的事件。此位由硬件自动清零。

位 17:0 保留，必须保持复位值。

20.6.16 EXTI D3 挂起屏蔽寄存器 (EXTI_D3PMR3)

EXTI D3 pending mask register

偏移地址：0x4C
复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	MR88	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
							rw								

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

位 31:25 保留，必须保持复位值。

位 24 **MRx**：事件输入 x+64 上的 D3 挂起屏蔽 (D3 Pending Mask on Event input x+64)
0：屏蔽来自线 x+64 的 D3 挂起请求。向该位写入 0 也会清除 D3 挂起请求。
1：取消屏蔽来自线 x+64 的 D3 挂起请求。D3 域挂起信号在触发时会使 D3 域唤醒保持激活，直至其被清除。

位 23:0 保留，必须保持复位值。

20.6.17 EXTI D3 挂起清除选择寄存器低位字 (EXTI_D3PCR3L)

EXTI D3 pending clear selection register low

偏移地址：0x50
复位值：0x0000 0000

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

位 31:0 保留，必须保持复位值。

20.6.18 EXTI D3 挂起清除选择寄存器高位字 (EXTI_D3PCR3H)

EXTI D3 pending clear selection register high

偏移地址：0x54
复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PCS88	
														rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

位 31:18 保留，必须保持复位值。

位 17:16 **PCSx**：事件输入 x = truncate ((n+160)/2) 上的 D3 挂起请求清除输入信号选择 (D3 Pending request clear input signal selection on Event input x= truncate ((n+160)/2))

00：选择 DMA ch6 事件作为 D3 域挂起清除源
01：选择 DMA ch7 事件作为 D3 域挂起清除源
10：选择 LPTIM4 输出作为 D3 域挂起清除源
11：选择 LPTIM5 输出作为 D3 域挂起清除源

位 15:0 保留，必须保持复位值。

20.6.19 EXTI 中断屏蔽寄存器 (EXTI_CPUIMR1)

EXTI interrupt mask register

偏移地址：0x80

复位值：0xFFC0 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MR31	MR30	MR29	MR28	MR27	MR26	MR25	MR24	MR23	MR22	MR21	MR20	MR19	MR18	MR17	MR16
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MR15	MR14	MR13	MR12	MR11	MR10	MR9	MR8	MR7	MR6	MR5	MR4	MR3	MR2	MR1	MR0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:22 **MRx**：直接事件输入 x 上的 CPU 中断屏蔽 (CPU interrupt Mask on Direct Event input x)(1)

0：屏蔽来自 x 线的中断请求
1：取消屏蔽来自线 x 的中断请求

位 21:0 **MRx**：可配置事件输入 x 上的 CPU 中断屏蔽 (CPU interrupt Mask on Configurable Event input x) (2)

0：屏蔽来自 x 线的中断请求
1：取消屏蔽来自线 x 的中断请求

1. 直接事件输入的复位值置“1”，以在默认情况下使能中断。
2. 可配置事件输入的复位值置“0”，以在默认情况下禁止中断。

20.6.20 EXTI 事件屏蔽寄存器 (EXTI_CPUEMR1)

EXTI event mask register

偏移地址：0x84

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MR31	MR30	MR29	MR28	MR27	MR26	MR25	MR24	MR23	MR22	MR21	MR20	MR19	MR18	MR17	MR16
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MR15	MR14	MR13	MR12	MR11	MR10	MR9	MR8	MR7	MR6	MR5	MR4	MR3	MR2	MR1	MR0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:0 **MRx**：事件输入 x 上的 CPU 事件屏蔽 (CPU Event mask on Event input x)

0：屏蔽来自 x 线的事件请求
1：取消屏蔽来自线 x 的事件请求

20.6.21 EXTI 挂起寄存器 (EXTI_CPUPR1)

EXTI pending register

偏移地址：0x88

复位值：未定义

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PR21	PR20	PR19	PR18	PR17	PR16
										rc1	rc1	rc1	rc1	rc1	rc1
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PR15	PR14	PR13	PR12	PR11	PR10	PR9	PR8	PR7	PR6	PR5	PR4	PR3	PR2	PR1	PR0
rc1	rc1	rc1	rc1	rc1	rc1	rc1	rc1	rc1	rc1	rc1	rc1	rc1	rc1	rc1	rc1

位 31:22 保留，必须保持复位值。

位 21:0 **PRx**：可配置事件输入 x 挂起位 (Configurable event inputs x Pending bit)

0：未发生触发请求

1：发生了选择的触发请求

当在外部中断线上发生了选择的边沿事件，该位被置“1”。向此位写入 1 可将其清除，也可以通过更改边沿检测器的敏感性将其清除。

20.6.22 EXTI 中断屏蔽寄存器 (EXTI_CPUIMR2)

EXTI interrupt mask register

偏移地址：0x90

复位值：0xFFF5 FFFF

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MR63	MR62	MR61	MR60	MR59	MR58	MR57	MR56	MR55	MR54	MR53	MR52	MR51	MR50	MR49	MR48
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MR47	MR46	Res.	MR44	MR43	MR42	MR41	MR40	MR39	MR38	MR37	MR36	MR35	MR34	MR33	MR32
rw	rw	1	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:20 **MRx**：直接事件输入 x+32 上的 CPU 中断屏蔽 (CPU Interrupt Mask on Direct Event input x+32)(1)

0：屏蔽来自 x 线的中断请求

1：取消屏蔽来自线 x 的中断请求

位 19 **MRx**：可配置事件输入 x+32 上的 CPU 中断屏蔽 (CPU interrupt Mask on Configurable Event input x+32) (2)

0：屏蔽来自 x 线的中断请求

1：取消屏蔽来自线 x 的中断请求

位 18 **MRx**：直接事件输入 x+32 上的 CPU 中断屏蔽 (CPU Interrupt Mask on Direct Event input x+32)(1)

0：屏蔽来自 x 线的中断请求

1：取消屏蔽来自线 x 的中断请求

位 17 **MRx**：可配置事件输入 x+32 上的 CPU 中断屏蔽 (CPU interrupt Mask on Configurable Event input x+32)(2)

0：屏蔽来自 x 线的中断请求
1：取消屏蔽来自线 x 的中断请求

位 16:14 **MRx**：直接事件输入 x+32 上的 CPU 中断屏蔽 (CPU Interrupt Mask on Direct Event input x+32)(1)

0：屏蔽来自 x 线的中断请求
1：取消屏蔽来自线 x 的中断请求

位 13 保留，必须保持复位值 (1)。

位 12:0 **MRx**：直接事件输入 x+32 上的 CPU 中断屏蔽 (CPU Interrupt Mask on Direct Event input x+32)(1)

0：屏蔽来自 x 线的中断请求
1：取消屏蔽来自线 x 的中断请求

1. 直接事件输入的复位值置“1”，以在默认情况下使能中断。
2. 可配置事件输入的复位值置“0”，以在默认情况下禁止中断。

20.6.23 EXTI 事件屏蔽寄存器 (EXTI_CPUEMR2)

EXTI event mask register

偏移地址：0x94

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MR63	MR62	MR61	MR60	MR59	MR58	MR57	MR56	MR55	MR54	MR53	MR52	MR51	MR50	MR49	MR48
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MR47	MR46	Res.	MR44	MR43	MR42	MR41	MR40	MR39	MR38	MR37	MR36	MR35	MR34	MR33	MR32
rw	rw	0	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:14 **MRx**：事件输入 x+32 上的 CPU 事件屏蔽 (CPU Event mask on Event input x+32)

0：屏蔽来自 x 线的事件请求
1：取消屏蔽来自线 x 的事件请求

位 13 保留，必须保持复位值。

位 12:0 **MRx**：事件输入 x+32 上的 CPU 事件屏蔽 (CPU Event mask on Event input x+32)

0：屏蔽来自 x 线的事件请求
1：取消屏蔽来自线 x 的事件请求

20.6.24 EXTI 挂起寄存器 (EXTI_CPUPR2)

EXTI pending register

偏移地址：0x98

复位值：未定义

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PR51	Res.	PR49	Res.
												rc1		rc1	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

位 31:20 保留，必须保持复位值。

位 19 **PRx**：可配置事件输入 x+32 挂起位 (Configurable event inputs x+32 Pending bit)

0：未发生触发请求
1：发生了选择的触发请求
当在外部中断线上发生了选择的边沿事件，该位被置“1”。向此位写入 1 可将其清除，也可以通过更改边沿检测器的敏感性将其清除。

位 18 保留，必须保持复位值。

位 17 **PRx**：可配置事件输入 x+32 挂起位 (Configurable event inputs x+32 Pending bit)

0：未发生触发请求
1：发生了选择的触发请求
当在外部中断线上发生了选择的边沿事件，该位被置“1”。向此位写入 1 可将其清除，也可以通过更改边沿检测器的敏感性将其清除。

位 16:0 保留，必须保持复位值。

20.6.25 EXTI 中断屏蔽寄存器 (EXTI_CPUIMR3)

EXTI interrupt mask register

偏移地址：0xA0

复位值：0x018B FFFF

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	MR88	MR87	MR86	MR85	MR84	Res.	MR82	Res.	MR80
							rw	rw	rw	rw	rw		rw		rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MR79	MR78	MR77	MR76	MR75	MR74	MR73	MR72	MR71	MR70	MR69	MR68	MR67	MR66	MR65	MR64
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:25 保留，必须保持复位值。

位 24:23 **MRx**：直接事件输入 x+64 上的 CPU 中断屏蔽 (CPU Interrupt Mask on Direct Event input x+64) (1)

0：屏蔽来自 x 线的中断请求
1：取消屏蔽来自线 x 的中断请求

位 22:20 **MRx**：可配置事件输入 x+64 上的 CPU 中断屏蔽 (CPU interrupt Mask on Configurable Event input x+64) (2)

0：屏蔽来自 x 线的中断请求
1：取消屏蔽来自线 x 的中断请求

位 19 保留，必须保持复位值 (1)。

位 18 **MRx**：可配置事件输入 x+64 上的 CPU 中断屏蔽 (CPU interrupt Mask on Configurable Event input x+64)(2)

0：屏蔽来自 x 线的中断请求
1：取消屏蔽来自线 x 的中断请求

位 17 保留，必须保持复位值 (1)。

位 16:0 **MRx**：直接事件输入 x+64 上的 CPU 中断屏蔽 (CPU Interrupt Mask on Direct Event input x+64)(1)

0：屏蔽来自 x 线的中断请求
1：取消屏蔽来自线 x 的中断请求

1. 直接事件输入的复位值置“1”，以在默认情况下使能中断。
2. 可配置事件输入的复位值置“0”，以在默认情况下禁止中断。

20.6.26 EXTI 事件屏蔽寄存器 (EXTI_CPUEMR3)

EXTI event mask register

偏移地址：0xA4

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	MR88	MR87	MR86	MR85	MR84	Res.	MR82	Res.	MR80
							rw	rw	rw	rw	rw		rw		rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MR79	MR78	MR77	MR76	MR75	MR74	MR73	MR72	MR71	MR70	MR69	MR68	MR67	MR66	MR65	MR64
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:25 保留，必须保持复位值。

位 24:0 **MRx**：事件输入 x+64 上的 CPU 事件屏蔽 (CPU Event mask on Event input x+64)

0：屏蔽来自 x 线的事件请求
1：取消屏蔽来自线 x 的事件请求

20.6.27 EXTI 挂起寄存器 (EXTI_CPUPR3)

EXTI pending register

偏移地址：0xA8

复位值：未定义

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PR86	PR85	PR84	Res.	PR82	Res.	Res.
									rc1	rc1	rc1		rc1		
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

位 31:23 保留，必须保持复位值。

位 22:20 **PRx**：可配置事件输入 x+64 挂起位 (Configurable event inputs x+64 Pending bit)

0：未发生触发请求
1：发生了选择的触发请求
当在外部中断线上发生了选择的边沿事件，该位被置“1”。向此位写入 1 可将其清除，也可以通过更改边沿检测器的敏感性将其清除。

位 19 保留，必须保持复位值。

位 18 **PRx**：可配置事件输入 x+64 挂起位 (Configurable event inputs x+64 Pending bit)

0：未发生触发请求
1：发生了选择的触发请求
当在外部中断线上发生了选择的边沿事件，该位被置“1”。向此位写入 1 可将其清除，也可以通过更改边沿检测器的敏感性将其清除。

位 17:0 保留，必须保持复位值。

20.6.28 EXTI 寄存器映射

下表列出了 EXTI 寄存器映射和复位值。

表 134. 异步中断/事件控制器寄存器映射和复位值

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	**EXTI_RTSR1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TR[21:0]																					
	Reset value											0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x04	**EXTI_FTSR1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TR[21:0]																					
	Reset value											0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x08	**EXTI_SWIER1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SWIER[21:0]																					
	Reset value											0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0C	**EXTI_D3PMR1**	Res.	Res.	Res.	Res.	Res.	Res.	MR[25]	Res.	Res.	Res.	MR[21:19]			Res.	Res.	Res.	MR[15:0]															
	Reset value							0				0	0	0				0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x10	**EXTI_D3PCR1L**	PCSI[15]		PCSI[14]		PCSI[13]		PCSI[12]		PCSI[11]		PCSI[10]		PCSI[9]		PCSI[8]		PCSI[7]		PCSI[6]		PCSI[5]		PCSI[4]		PCSI[3]		PCSI[2]		PCSI[1]		PCSI[0]	
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x14	**EXTI_D3PCR1H**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PCSI[25]		Res.	Res.	Res.	Res.	Res.	Res.	PCSI[21]		PCSI[20]		PCSI[19]		Res.	Res.	Res.	Res.	Res.	Res.
	Reset value													0	0							0	0	0	0	0	0						
0x20	**EXTI_RTSR2**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TR[51]	Res.	TR[49]	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	Reset value													0		0																	
0x24	**EXTI_FTSR2**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TR[51]	Res.	TR[49]	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	Reset value													0		0																	
0x28	**EXTI_SWIER2**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SWIER[51]	Res.	SWIER[49]	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	Reset value													0		0																	
0x2C	**EXTI_D3PMR2**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MR[53:48]						Res.	Res.	Res.	Res.	Res.	Res.	MR[41]	Res.	Res.	Res.	Res.	Res.	MR[35]	MR[34]	Res.	Res.
	Reset value											0	0	0	0	0	0							0						0	0		
0x30	**EXTI_D3PCR2L**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PCSI[41]		Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PCSI[35]		PCSI[34]		Res.	Res.	Res.	Res.
	Reset value													0	0											0	0	0	0				
0x34	**EXTI_D3PCR2H**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PCSI[53]		PCSI[52]		PCSI[51]		PCSI[50]		PCSI[49]		PCSI[48]	
	Reset value																					0	0	0	0	0	0	0	0	0	0	0	0
0x40	**EXTI_RTSR3**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TR[86]	TR[85]	TR[84]	Res.	TR[82]	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	Reset value										0	0	0		0																		

表 134. 异步中断/事件控制器寄存器映射和复位值

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x44	**EXTI_FTSR3**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TR[86]	TR[85]	TR[84]	Res.	TR[82]	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	Reset value										0	0	0		0																		
0x48	**EXTI_SWIER3**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SWIER[86]	SWIER[85]	SWIER[84]	Res.	SWIER[82]	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	Reset value										0	0	0		0																		
0x4C	**EXTI_D3PMR3**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MR[88]	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	Reset value								0																								
0x50	**EXTI_D3PCR3L**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	Reset value																																
0x54	**EXTI_D3PCR3H**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PCS[88]		Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	Reset value															0	0																
0x58-0x7C	Reserved																																
0x80	**EXTI_CPUIMR1**	MR[31:22]										MR[21:0]																					
	Reset value	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x84	**EXTI_CPUEMR1**	MR[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x88	**EXTI_CPUPR1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PR[21:0]																					
	Reset value											0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x90	**EXTI_CPUIMR2**	MR[63:52]												MR[51]	MR[50]	MR[49]	MR[48:46]			Res.	MR[44:32]												
	Reset value	1	1	1	1	1	1	1	1	1	1	1	1	0	1	0	1	1	1		1	1	1	1	1	1	1	1	1	1	1	1	1
0x94	**EXTI_CPUEMR2**	MR[63:46]																		Res.	MR[44:32]												
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		0	0	0	0	0	0	0	0	0	0	0	0	0
0x98	**EXTI_CPUPR2**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PR[51]	Res.	PR[49]	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	Reset value													0		0																	
0xA0	**EXTI_CPUIMR3**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MR[88]	MR[87]	MR[86]	MR[85]	MR[84]	Res.	MR[82]	Res.	MR[80:64]																
	Reset value								1	1	0	0	0		0		1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0xA4	**EXTI_CPUEMR3**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MR[88:84]					Res.	MR[82]	Res.	MR[80:64]																
	Reset value								0	0	0	0	0		0		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

表 134. 异步中断/事件控制器寄存器映射和复位值

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0xA8	**EXTI_CPUPR3**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PR[86]	PR[85]	PR[84]	Res.	PR[82]	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	Reset value										0	0	0		0																		
0xAC-0xBC	Reserved	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

有关寄存器边界地址的信息，请参见*第 100 页的第 2.2.2 节*。

21 循环冗余校验计算单元 (CRC)

21.1 简介

CRC（循环冗余校验）计算单元使用多项式发生器从一个 8 位/16 位/32 位的数据字中产生 CRC 码。

在众多的应用中，基于 CRC 的技术还常用来验证数据传输或存储的完整性。根据功能安全标准的规定，这些技术提供了验证 Flash 完整性的方法。CRC 计算单元有助于在运行期间计算软件的签名，并将该签名与链接时生成并存储在指定存储单元的参考签名加以比较。

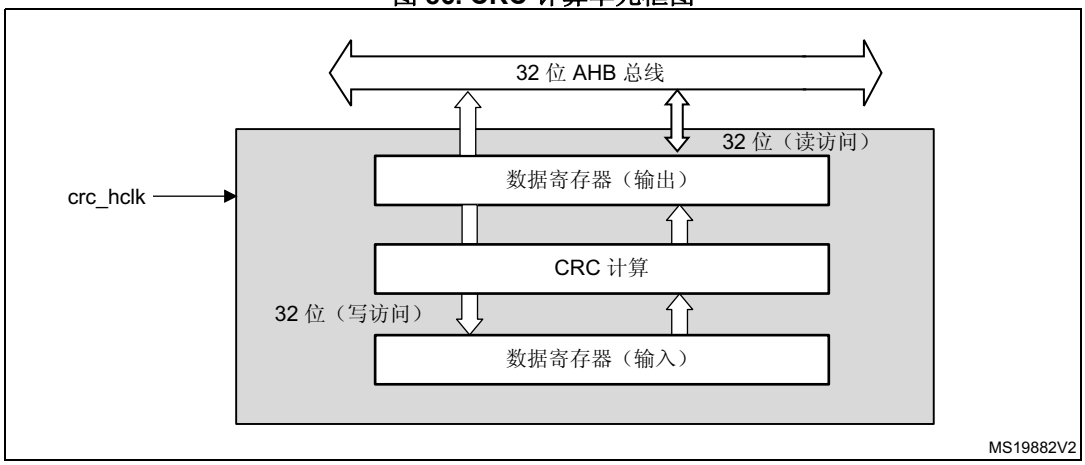
21.2 CRC 主要特性

- 使用 CRC-32 （以太网）多项式：0x4C11DB7
  $X^{32} + X^{26} + X^{23} + X^{22} + X^{16} + X^{12} + X^{11} + X^{10} + X^8 + X^7 + X^5 + X^4 + X^2 + X + 1$
- 或者，使用位数可编程的 （7 位、 8 位、 16 位和 32 位）的完全可编程多项式
- 处理 8 位、16 位、32 位数据大小
- 可编程 CRC 初始值
- 单输入/输出 32 位数据寄存器
- 输入缓冲器可避免计算期间发生总线阻塞
- 对于 32 位数据大小，CRC 计算在 4 个 AHB 时钟周期 (HCLK) 内完成
- 8 位通用寄存器 （可用于临时存储）
- I/O 数据的可逆性选项

21.3 CRC 功能说明

21.3.1 CRC 框图

图 86. CRC 计算单元框图



21.3.2 CRC 内部信号

表 135. CRC 内部输入/输出信号

信号名称	信号类型	说明
crc_hclk	数字输入	AHB时钟

21.3.3 CRC 操作

CRC 计算单元具有单个 32 位读/写数据寄存器 (CRC_DR)。它用于输入新数据（写访问）和保存之前 CRC 计算的结果（读访问）。

对数据寄存器的每个写操作都会对之前的 CRC 值（存储在 CRC_DR 中）和新值再做一次 CRC 计算。CRC 计算针对整个 32 位数据字或逐个字节完成，具体取决于数据的写入格式。

CRC_DR 寄存器可按字、右对齐半字和右对齐字节进行访问。对于其它寄存器，只允许进行 32 位访问。

计算时间取决于数据宽度：

- 32 位数据需要 4 个 AHB 时钟周期
- 16 位数据需要 2 个 AHB 时钟周期
- 8 位数据需要 1 个 AHB 时钟周期

输入缓冲器中可立即写入第二个数据，无需因之前的 CRC 计算而等待任何等待状态。

可动态调整数据大小，从而能最大程度地减少给定字节数的写访问次数。例如，对 5 个字节进行 CRC 计算时，可先写入字，然后写入字节。

输入数据的顺序可反转，以管理各种数据存放方式（大端/小端）。可对 8 位、16 位和 32 位数据执行反转操作，具体取决于 CRC_CR 寄存器中的 REV_IN[1:0] 位。

例如，输入数据 0x1A2B3C4D 在 CRC 计算中用作：

按字节执行位反转的 0x58D43CB2

按半字执行位反转的 0xD458B23C

按全字执行位反转的 0xB23CD458

通过将 CRC_CR 寄存器中 REV_OUT 位置 1 也可以将输出数据反转。

该操作按位进行：例如，输出数据 0x11223344 将转换为 0x22CC4488。

使用 CRC_CR 寄存器中的 RESET 控制位可将 CRC 计算器初始化为可编程值（默认值为 0xFFFFFFFF）。

可使用 CRC_INIT 寄存器对 CRC 初始值进行编程。对 CRC_INIT 寄存器进行写访问时会自动初始化 CRC_DR 寄存器。

CRC_IDR 寄存器可用于保存与 CRC 计算相关的临时值。它不受 CRC_CR 寄存器中的 RESET 位影响。

多项式可编程

多项式系数可完全通过 CRC_POL 寄存器进行编程，通过编程 CRC_CR 寄存器中的 POLYSIZE[1:0] 位可将多项式大小配置为 7 位、8 位、16 位或 32 位。不支持偶数多项式。

如果 CRC 数据小于 32 位，可从 CRC_DR 寄存器的最低有效位读取 CRC 的值。

为实现可靠的 CRC 计算，CRC 计算期间不能实时更改多项式的值或大小。因此，如果正在执行 CRC 计算，则在更改多项式前，应用程序必须先复位，或者先执行 CRC_DR 读操作。

默认的多项式值为 CRC-32（以太网）多项式：0x4C11DB7。

21.4 CRC 寄存器

21.4.1 数据寄存器 (CRC_DR)

Data register

偏移地址：0x00

复位值：0xFFFF FFFF

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

位 31:0 **DR[31:0]**：数据寄存器位 (Data register bits)

该寄存器用于向 CRC 计算器写入新数据。

读取寄存器时可读出之前的 CRC 计算结果。

如果数据大小小于 32 位，则最低有效位可用于写入/读取正确值。

21.4.2 独立数据寄存器 (CRC_IDR)

Independent data register

偏移地址：0x04

复位值：0x0000 0000

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

位 31:0 **IDR[31:0]**：通用的 32 位数据寄存器位 (General-purpose 32-bit data register bits)

这些位可用作四个字节的临时存储单元。

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

21.4.3 控制寄存器 (CRC_CR)

Control register

偏移地址：0x08

复位值：0x0000 0000

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

位 31:8 保留，必须保持清零。

位 7 **REV_OUT**：反转输出数据 (Reverse output data)

该位用于控制输出数据位顺序的反转。

0：不影响位顺序

1：位反转输出格式

位 6:5 **REV_IN[1:0]**：反转输入数据 (Reverse input data)

这些位用于控制输入数据位顺序的反转。

00：不影响位顺序

01：按字节执行位反转

10：按半字执行位反转

11：按字执行位反转

位 4:3 **POLYSIZE[1:0]**：多项式大小 (Polynomial size)

这些位用于控制多项式的大小。

00：32 位多项式

01：16 位多项式

10：8 位多项式

11：7 位多项式

位 2:1 保留，必须保持清零。

位 0 **RESET**：RESET 位

此位由软件置 1，用于复位 CRC 计算单元并将数据寄存器设置为存储在 CRC_INIT 寄存器中的值。此位只能置 1，将由硬件自动进行清零。

21.4.4 CRC 初始值 (CRC_INIT)

Initial CRC value

偏移地址：0x10

复位值：0xFFFF FFFF

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

位 31:0 **CRC_INIT**：*可编程 CRC 初始值 (Programmable initial CRC value)*
此寄存器用于写入 CRC 初始值。

21.4.5 CRC 多项式 (CRC_POL)

CRC polynomial

偏移地址：0x14

复位值：0x04C11DB7

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

位 31:0 **POL[31:0]**：可编程多项式 (Programmable polynomial)
此寄存器用于写入要用于 CRC 计算的多项式系数。
如果多项式大小小于 32 位，则必须使用最低有效位编程正确值。

21.4.6 CRC 寄存器映射

表 136. CRC 寄存器映射和复位值

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	**CRC_DR**	DR[31:0]																															
	Reset value	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0x04	**CRC_IDR**	IDR[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x08	**CRC_CR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REV_OUT	REV_IN[1:0]		POLYSIZE[1:0]		Res.	Res.	RESET
	Reset value																									0	0	0	0	0			0
0x10	**CRC_INIT**	CRC_INIT[31:0]																															
	Reset value	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0x14	**CRC_POL**	Polynomial coefficients																															
	Reset value	0x04C11DB7																															

有关寄存器边界地址的信息，请参见*第 100 页的第 2.2.2 节*。

22 灵活存储控制器（FMC）

可变存储控制器 (FMC) 包括以下三个存储控制器：

- NOR/PSRAM 存储控制器
- NAND 存储控制器
- 同步 DRAM (SDRAM/Mobile LPSDR SDRAM) 控制器

22.1 FMC 主要特性

FMC 功能块可连接：同步 / 异步静态存储器、SDRAM 存储器和 NAND Flash。其主要用途有：

- 将 AXI 数据通信事务转换为适当的外部器件协议
- 满足外部存储器器件的访问时间要求

所有外部存储器共享地址、数据和控制信号，但有各自的片选信号。FMC 一次只能访问一个外部器件。

FMC 控制器的主要特性如下：

- 连接外部静态存储器，包括：
  - 静态随机访问存储器 (SRAM)
  - NOR Flash/OneNAND Flash
  - PSRAM（4 个存储区域）
  - 带有硬件 ECC 的 NAND Flash 存储器，可检查多达 8 KB 的数据
- 连接同步 DRAM (SDRAM/Mobile LPSDR SDRAM) 存储器
- 支持突发模式，能够更快速地访问同步器件（如 NOR Flash、PSRAM 和 SDRAM）
- 可编程连续时钟输出以支持异步和同步访问
- 具有 8 位、16 位或 32 位宽的数据总线
- 每个存储区域有独立的片选控制
- 每个存储区域可独立配置
- 写使能和字节通道选择输出，可配合 PSRAM、SRAM 和 SDRAM 器件使用
- 外部异步等待控制
- 16 x 32 位深度写 FIFO

  写 FIFO 由所有存储控制器所共用，包括：
  - 写数据 FIFO，用于储存将写入存储器的数据
  - 写地址 FIFO，用于存储地址（最多 28 位）以及数据大小（最多 2 位）。在突发模式下工作时，将仅存储起始地址，但越过页边界时除外（适用于 PSRAM 和 SDRAM）。在此情况下，突发传输将分成两个 FIFO 条目
- SDRAM 控制器具有可缓存的 6 x 64 位深度读 FIFO（6 x 14 位地址标记）

启动时，必须通过用户应用程序对 FMC 引脚进行配置。应用程序未使用的 FMC I/O 引脚可用于其它用途。

定义外部器件类型和其特性的 FMC 寄存器在启动时进行设置，并且在下次上电或复位前保持不变。但是，只有几个位可实时更改。

- FMC_PCR 寄存器中的 ECCEN 和 PBEN 位。
- FMC_SR 寄存器中的 IFS、IRS 和 ILS 位。
- FMC_SDCMR 寄存器中的 MODE[2:0]、CTB1/CTB2、NRFS 和 MRD 位。
- FMC_SDRTR 寄存器中的 REIE 和 CRE 位。

当 FMC 使能时，按照以下序列修改部分参数：

1. 首先禁止 FMC 控制器，防止在寄存器修改期间对存储控制器进行任何进一步的访问。
2. 更新全部所需配置。
3. 再次使能 FMC 控制器。

当使用 SDRAM 控制器时，如果必须在初始化阶段之后修改 SDCLK 时钟速率或刷新速率，请务必遵循以下程序：

1. 将 SDRAM 器件置于自刷新模式。
2. 通过复位 FMC_BCR1 寄存器中的 FMCEN 禁止 FMC 控制器。
3. 更新所需参数。
4. 在所有参数更新完毕后，使能 FMC 控制器。
5. 之后，发送时钟配置使能命令以退出自刷新模式。

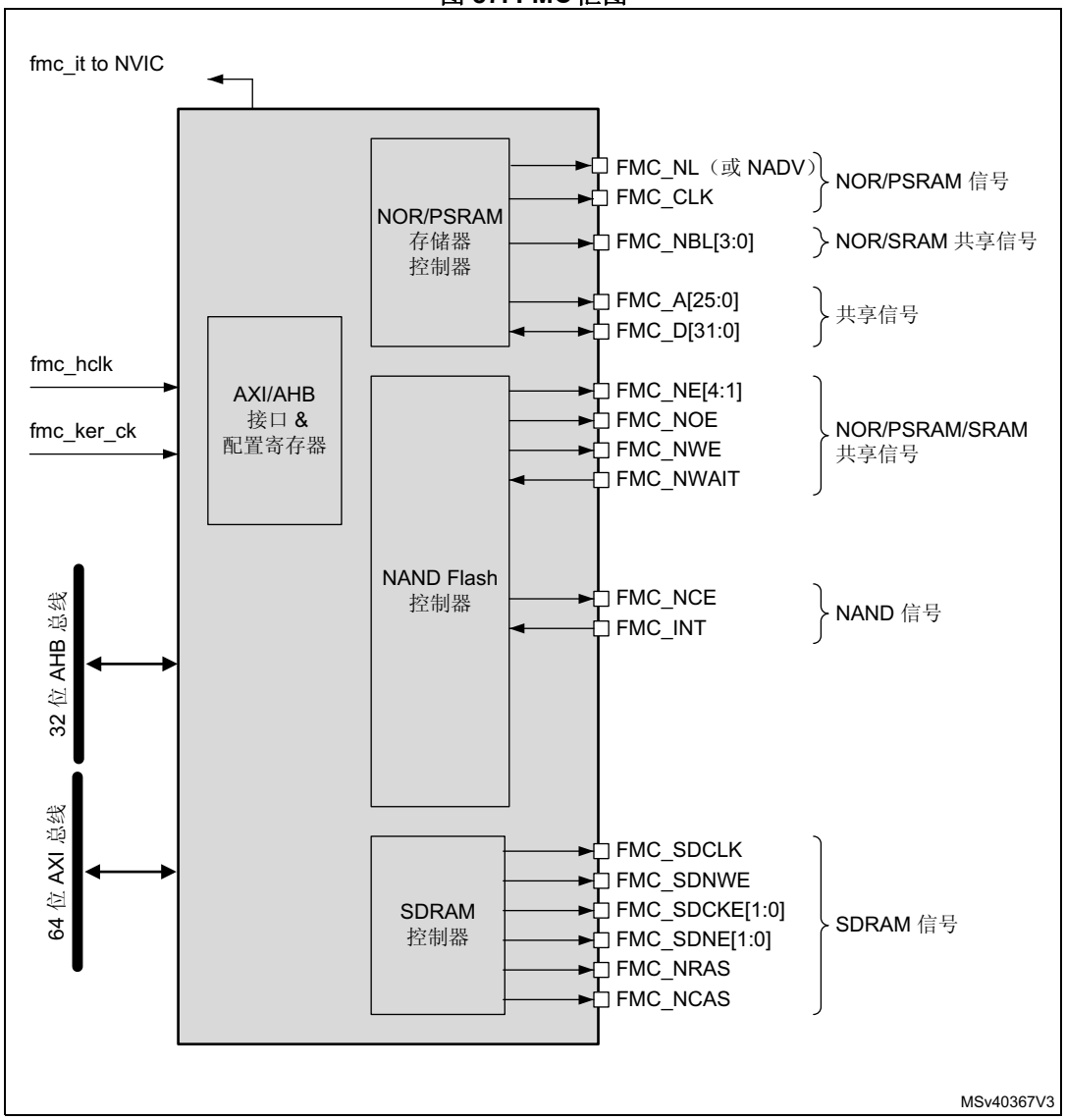
22.2 FMC 框图

FMC 包含以下主要模块：

- NOR Flash/PSRAM/SRAM 控制器
- NAND 控制器
- SDRAM 控制器
- AXI 接口
- AHB 接口（包括 FMC 配置寄存器）

框图如下图所示。

图 87. FMC框图



22.3 FMC 内部信号

表 137 列出了 FMC 内部信号。 FMC 引脚 （即外部信号）在*第 22.7.1 节：外部存储器接口信号*中进行了说明。

表 137. FMC 引脚

名称	信号类型	说明
fmc_it	数字输出	FMC 中断
fmc_ker_ck	数字输入	FMC 内核时钟
fmc_hclk	数字输入	FMC 接口时钟

22.4 AHB 接口

AHB 从接口使得内部 CPU 能够配置 FMC 寄存器。

AHB 时钟 (fmc_hclk) 是 FMC 寄存器访问的参考时钟。

22.5 AXI接口

CPU 和其它 AXI 总线主设备可通过该 AXI 从设备接口访问外部存储器。

AXI 事务会转换为外部器件协议。由于 AXI 数据总线为 64 位宽，因此 AXI 事务可根据数据大小访问拆分成几个连续的 32 位、16 位或 8 位访问。除非在使能扩展模式时在 D 模式下进行访问，否则 FMC 片选 (FMC_NEx) 在连续两次访问之间不会翻转。

当满足下列其中一个条件时，FMC 将生成 AXI 从错误：

- 读取或写入未使能的 FMC 存储区域（存储区域 1 至 4）。
- 在 FMC_BCRx 寄存器中的 FACCEN 位复位时读取或写入 NOR Flash 存储区域。
- 写入受写保护的 SDRAM 存储区域（SDRAM_SDCRx 寄存器中 WP 位）。
- 违反 SDRAM 地址范围（访问保留的地址范围）。
- 试图在 SDRAM 存储区域尚未完成初始化的情况下对其进行读取/写入访问。

在 BMAP [1:0] 位配置之后，如果 FMC 存储区域基址不支持 ADDR[31:28] 地址位，FMC 会产生 AXI 解码器错误。

FMC 控制器的内核时钟是异步 fmc_ker_ck 时钟（关于 fmc_ker_ck 时钟源的选择，请参见*第 8 节：复位和时钟控制 (RCC)*）。

22.5.1 支持的存储器和事务

通用事务规则

所请求的 AXI 事务数据宽度可以是 8、16 或 32 位，但访问的外部器件具有固定的数据宽度。这可能会导致不一致的数据宽度。

因此，可遵循一些简单的事务规则，具体取决于 AXI 事务大小与存储器数据大小的关系：

- AXI 事务大小与存储器数据大小相等可防止问题的发生。
- AXI 事务数据宽度大于存储器宽度：

  在此情况下，FMC 会将 AXI 事务分为多个较小的连续存储器访问，以符合外部数据宽度。
- AXI 事务数据宽度小于存储器宽度：

  传送可能一致，也可能不一致，具体取决于外部器件的类型：
  - 访问具有字节选择功能的器件（SRAM、ROM、PSRAM 和 SDRAM）

    在此情况下，FMC 允许读/写事务并通过其字节选择通道 NBL[3:0] 访问恰当的数据。

    通过 NBL[3:0] 寻址要写入的字节。

    读取所有存储器字节（NBL[3:0] 在读取事务期间保持为低电平），并丢弃无用的字节。
  - 访问不具有字节选择功能的器件（NOR 和 NAND Flash）

    当请求对 16 位宽的 Flash 存储器进行字节访问时会发生此情形。由于无法在字节模式下访问器件（只能向 Flash 读取或写入 16 位字），因此允许读事务而不允许写事务（控制器会读取全部 16 位存储器字，但只使用所需字节）。

NOR Flash/PSRAM 和 SDRAM 的回卷支持

由于并非所有主器件都能发出回卷事务，因此同步存储器必须按未定义长度的线性突发模式进行配置。

如果主器件生成回卷事务：

- 读操作将分为两个线性突发事务。
- 如果使能写 FIFO，写操作将分为两个线性突发事务；如果禁止写 FIFO，写操作将分为多个线性突发事务。

配置寄存器

可通过一组寄存器配置 FMC。有关 NOR Flash/PSRAM 控制寄存器的详细说明，请参见*第 22.7.6 节*。有关 NAND Flash 寄存器的详细说明，请参见*第 22.8.7 节*，有关 SDRAM 控制寄存器的详细说明，请参见*第 22.9.5 节*。

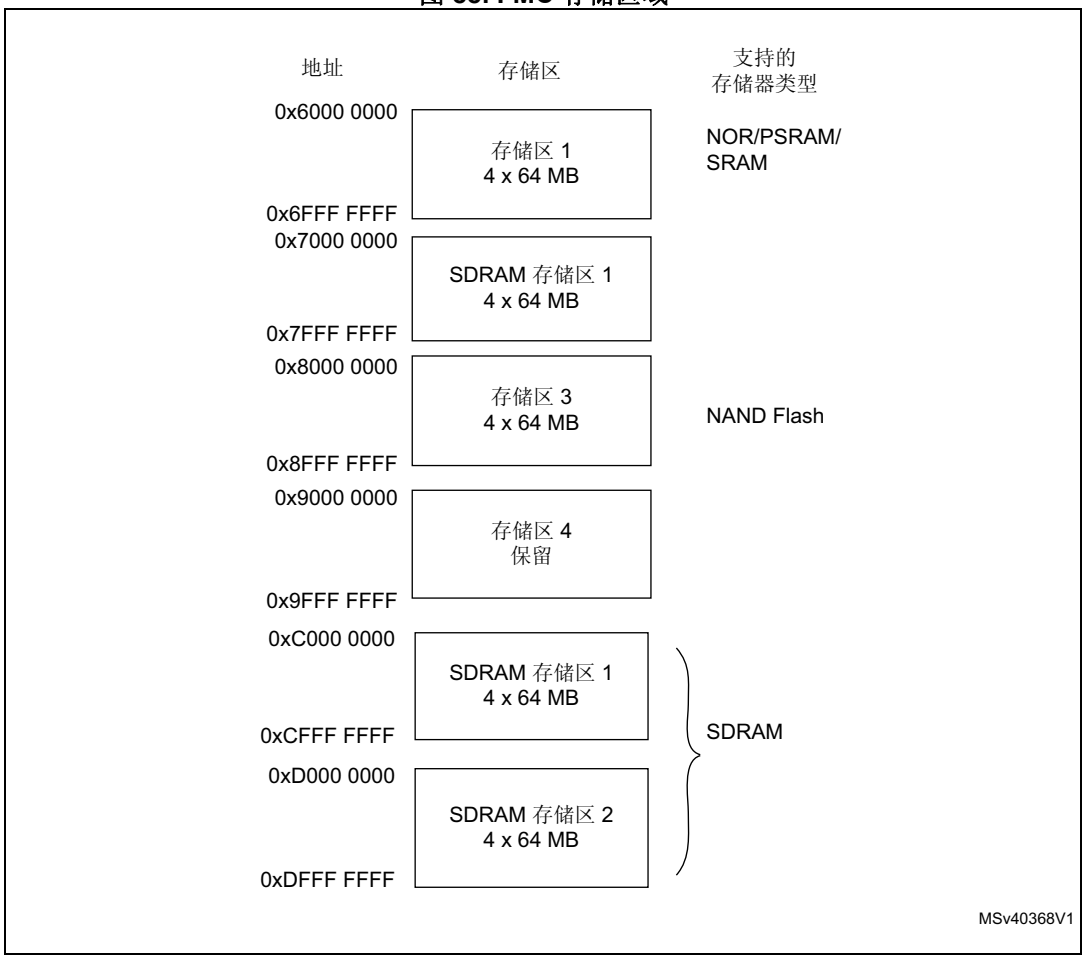
22.6 外部器件地址映射

从 FMC 的角度，外部存储器被划分为固定大小的存储区域，每个存储区域的大小为 256 MB（请参见*图 88*）：

- 存储区域 1 可连接多达 4 个 NOR Flash 或 PSRAM 设备。此存储区域被划分为如下 4 个 NOR/PSRAM 子区域，带 4 个专用片选信号：
  - 存储区域 1 NOR/PSRAM 1
  - 存储区域 1 NOR/PSRAM 2
  - 存储区域 1 NOR/PSRAM 3
  - 存储区域 1 NOR/PSRAM 4
- 存储区域 2 用于 SDRAM 器件，具体是 SDRAM 存储区域 1 还是 SDRAM 存储区域 2 取决于 BMAP 位配置。
- 存储区域 3 用于连接 NAND Flash 器件。此空间的 MPU 存储器特性必须通过软件重新配置到器件中。
- 存储区域 4 和 5 用于连接 SDRAM 器件（每个存储区域 1 个器件）。

对于每个存储区域，所要使用的存储器类型可由用户应用程序通过配置寄存器配置。

图 88. FMC 存储区域



FMC 存储区域映射可通过 FMC_BCR1 寄存器中的 BMAP[1:0] 位修改。*表 138* 显示了用于交换 NOR/PSRAM 存储区域与 SDRAM 存储区域或者重映射 SDRAM 存储区域 2 的配置，从而允许在两个不同的地址映射中访问 SDRAM 存储区域。

表 138. FMC 存储区域映射选项

起始地址 - 结束地址	BMAP[1:0]=00 （默认映射）	BMAP[1:0]=01 交换 NOR/PSRAM 与 SDRAM 存储区域	BMAP[1:0]=10 重映射 SDRAM 存储区域 2
0x6000 0000 - 0x6FFF FFFF	NOR/PSRAM 存储区域	SDRAM 存储区域 1	NOR/PSRAM 存储区域
0x7000 0000 - 0x7FFF FFFF	SDRAM 存储区域 1	SDRAM 存储区域 2	SDRAM 存储区域 2
0x8000 0000 - 0x8FFF FFFF	NAND 存储区域	NAND 存储区域	NAND 存储区域
0x9000 0000 - 0x9FFF FFFF	保留	保留	保留
0xC000 0000 - 0xCFFF FFFF	SDRAM 存储区域 1	NOR/PSRAM 存储区域	SDRAM 存储区域 1
0xD000 0000 - 0xDFFF FFFF	SDRAM 存储区域 2	SDRAM 存储区域 2	SDRAM 存储区域 2

22.6.1 NOR/PSRAM 地址映射

ADDR[27:26] 位用于从*表 139* 中所示的四个存储区域之中选择其中一个存储区域。

表 139. NOR/PSRAM 存储区域选择

ADDR[27:26](1)	选择的存储区域
00	存储区域 1 NOR/PSRAM 1
01	存储区域 1 NOR/PSRAM 2
10	存储区域 1 NOR/PSRAM 3
11	存储区域 1 NOR/PSRAM 4

1. ADDR 是内部地址线，但也会参与对外部存储器的寻址。

ADDR[25:0] 位包含外部存储器地址。由于 ADDR 为字节地址，而存储器按字寻址，所以根据存储器数据宽度不同，实际向存储器发送的地址也将有所不同，如下表所示。

表 140. NOR/PSRAM 外部存储器地址

存储器宽度(1)	向存储器发出的数据地址	最大存储器容量（位）
8 位	ADDR[25:0]	64 MB x 8 = 512 Mb
16 位	ADDR[25:1] >> 1	64 MB/2 x 16 = 512 Mb
32 位	ADDR[25:2] >> 2	64 MB/4 x 32 = 512 Mb

1. 如果外部存储器的宽度为 16 位，FMC 将使用内部的 ADDR[25:1] 地址来作为对外部存储器的寻址地址 FMC_A[24:0]。如果存储器宽度为 32 位，FMC 将使用内部的 ADDR[25:2] 地址进行外部寻址。
无论外部存储器的宽度是多少，FMC_A[0] 都应连接到外部存储器地址 A[0]。

22.6.2 NAND Flash 地址映射

NAND 存储区域分为几个存储器区域，如*表 141* 所示。

表 141. NAND 存储映射和时序寄存器

起始地址	结束地址	FMCBank	存储空间	时序寄存器
0x8800 0000	0x8BFF FFFF	存储区域 3 - NAND Flash	特性区	FMC_PATT (0x8C)
0x8000 0000	0x83FF FFFF		通用区	FMC_PMEM (0x88)

对于 NAND Flash 存储器，通用区和特性区存储空间分为三个部分，均位于低位 256 KB 中（见下面的*表 142*）：

- 数据区域（通用/特性存储空间中的第一个 64 KB）
- 命令区域（通用/特性存储空间中的第二个 64 KB）
- 地址区域（通用/特性存储空间中的下一个 128 KB）

表 142. NAND 存储区域选择

部分名称	ADDR[17:16]	地址范围
地址区域	1X	0x020000-0x03FFFF
命令区域	01	0x010000-0x01FFFF
数据区域	00	0x000000-0x0FFFF

应用程序软件使用这 3 个区域来访问 NAND Flash 存储器：

- **向 NAND Flash 存储器发送命令**：软件可以向命令区域中的任意存储器位置写入命令值。
- **指定读取或写入的 NAND Flash 地址**，软件必须向地址区域中的任意存储单元写入地址值。由于地址的长度可以是 4 或 5 个字节（具体取决于实际存储器大小），要指定完整的地址，需要对地址区域执行多个连续写入操作。
- **读取或写入数据**，软件将从数据区域中的任意存储单元读取数据，或者向其中写入数据。

由于 NAND Flash 存储器会自动递增地址，所以在访问连续存储器位置时，无需递增数据区域的地址。

22.6.3 SDRAM 地址映射

表 143 显示了两个可用的 SDRAM 存储区域。

表 143. SDRAM 存储区域选择

选择的存储区域	控制寄存器	时序寄存器
SDRAM 存储区域 1	FMC_SDCR1	FMC_SDTR1
SDRAM 存储区域 2	FMC_SDCR2	FMC_SDTR2

表 144 显示了 13 位行和 11 位列配置的 SDRAM 映射。

表 144. SDRAM 地址映射

存储器宽度(1)	内部存储区域	行地址	列地址(2)	最大存储器容量（MB）
8 位	ADDR[25:24]	ADDR[23:11]	ADDR[10:0]	64 MB： 4 x 8K x 2K
16 位	ADDR[26:25]	ADDR[24:12]	ADDR[11:1]	128 MB： 4 x 8K x 2K x 2
32 位	ADDR[27:26]	ADDR[25:13]	ADDR[12:2]	256 MB： 4 x 8K x 2K x 4

1. 连接 16 位存储器时，FMC 内部使用 ADDR[11:1] 内部地址线进行外部寻址。连接 32 位存储器时，FMC 内部使用 ADDR[12:2] 地址线进行外部寻址。无论外部存储器的宽度是多少，FMC_A[0] 都必须连接到外部存储器地址 A[0]。
2. 不支持 AutoPrecharge。FMC_A[10] 必须连接到外部存储器地址 A[10]，但始终为低电平。

ADDR[27:0] 位将转换为外部 SDRAM 地址，具体取决于 SDRAM 控制器配置：

- 数据大小：8、16 或 32 位
- 行大小：11、12 或 13 位
- 列大小：8、9、10 或 11 位
- 内部存储区域数量：两个或四个内部存储区域

下表显示了不同 SDRAM 控制器配置的 SDRAM 地址映射。

表 145. 数据总线宽度为 8 位时的 SDRAM 地址映射(1)(2)

行大小配置	ADDR（内部地址线）																										
	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
11 位行大小配置	Res.							Bank [1:0]		Row[10:0]											Column[7:0]							
	Res.						Bank [1:0]			Row[10:0]											Column[8:0]							
	Res.					Bank [1:0]				Row[10:0]											Column[9:0]							
	Res.				Bank [1:0]					Row[10:0]											Column[10:0]							
12 位行大小配置	Res.						Bank [1:0]			Row[11:0]											Column[7:0]							
	Res.					Bank [1:0]				Row[11:0]											Column[8:0]							
	Res.				Bank [1:0]					Row[11:0]											Column[9:0]							
	Res.			Bank [1:0]						Row[11:0]											Column[10:0]							
13 位行大小配置	Res.					Bank [1:0]				Row[12:0]											Column[7:0]							
	Res.				Bank [1:0]					Row[12:0]											Column[8:0]							
	Res.			Bank [1:0]						Row[12:0]											Column[9:0]							
	Res.		Bank [1:0]							Row[12:0]											Column[10:0]							

1. BANK[1:0] 为存储区域地址 BA[1:0]。当仅使用 2 个内部存储区域时，BA1 必须始终设置为“0”。
2. 访问保留的 (Res.) 地址范围会生成 AXI 从错误。

表 146. 数据总线宽度为 16 位时的 SDRAM 地址映射(1)(2)

行大小配置	ADDR（地址线）																										
	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
11 位行大小配置	Res.						Bank [1:0]		Row[10:0]											Column[7:0]								BM0(3)
	Res.						Bank [1:0]		Row[10:0]											Column[8:0]								BM0
	Res.						Bank [1:0]		Row[10:0]											Column[9:0]								BM0
	Res.						Bank [1:0]		Row[10:0]											Column[10:0]								BM0
12 位行大小配置	Res.						Bank [1:0]		Row[11:0]											Column[7:0]								BM0
	Res.						Bank [1:0]		Row[11:0]											Column[8:0]								BM0
	Res.						Bank [1:0]		Row[11:0]											Column[9:0]								BM0
	Res.						Bank [1:0]		Row[11:0]											Column[10:0]								BM0
13 位行大小配置	Res.						Bank [1:0]		Row[12:0]											Column[7:0]								BM0
	Res.						Bank [1:0]		Row[12:0]											Column[8:0]								BM0
	Res.						Bank [1:0]		Row[12:0]											Column[9:0]								BM0
	Res.						Bank [1:0]		Row[12:0]											Column[10:0]								BM0

1. BANK[1:0] 为存储区域地址 BA[1:0]。当仅使用 2 个内部存储区域时，BA1 必须始终设置为“0”。
2. 访问保留的空间 (Res.) 会生成 AXI 从错误。
3. BM0：是 16 位访问的字节屏蔽。

表 147. 32 位数据总线宽度时的 SDRAM 地址映射(1)(2)

行大小配置	ADDR（地址线）																										
	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
11 位行大小配置	Res.					Bank [1:0]		Row[10:0]											Column[7:0]								BM[1:0] (3)	
	Res.						Bank [1:0]		Row[10:0]										Column[8:0]								BM[1:0]	
	Res.							Bank [1:0]		Row[10:0]									Column[9:0]								BM[1:0]	
	Res.								Bank [1:0]		Row[10:0]								Column[10:0]								BM[1:0]	
12 位行大小配置	Res.						Bank [1:0]		Row[11:0]										Column[7:0]								BM[1:0]	
	Res.							Bank [1:0]		Row[11:0]									Column[8:0]								BM[1:0]	
	Res.								Bank [1:0]		Row[11:0]								Column[9:0]								BM[1:0]	
	Res.									Bank [1:0]		Row[11:0]							Column[10:0]								BM[1:0]	
13 位行大小配置	Res.							Bank [1:0]		Row[12:0]									Column[7:0]								BM[1:0]	
	Res.								Bank [1:0]		Row[12:0]								Column[8:0]								BM[1:0]	
	Res.									Bank [1:0]		Row[12:0]							Column[9:0]								BM[1:0]	
	Bank [1:0]										Row[12:0]								Column[10:0]								BM[1:0]	

1. BANK[1:0] 为存储区域地址 BA[1:0]。当仅使用 2 个内部存储区域时，BA1 必须始终设置为“0”。
2. 访问保留的空间 (Res.) 会生成 AXI 从错误。
3. BM[1:0]是 32 位访问的字节屏蔽。

22.7 NOR Flash/PSRAM 控制器

FMC 会生成适当的信号时序，以驱动以下类型的存储器：

- 异步 SRAM 和 ROM
  - 8 位
  - 16 位
  - 32 位
- PSRAM (Cellular RAM)
  - 异步模式
  - 用于同步访问的突发模式，具有在跨越 CRAM 1.5 的边界页时用于拆分突发访问的可配置选项
  - 复用或非复用
- NOR Flash
  - 异步模式
  - 同步访问的突发模式
  - 复用或非复用

FMC 会为每个存储区域输出唯一的片选信号 NE[4:1]。所有其它信号（地址、数据和控制）均为共享信号。

FMC 通过可编程时序支持多种器件，其中：

- 等待周期可编程（最多 15 个时钟周期）
- 总线周转周期可编程（最多 15 个时钟周期）
- 输出使能和写入使能延迟可编程（最多 15 个时钟周期）
- 独立的读和写时序和协议，以支持各种存储器和时序
- 可编程连续时钟 (FMC_CLK) 输出

FMC 输出时钟（FMC _CLK）是 fmc_ker_ck 时钟的约数。该时钟可传送到选定的外部器件，根据 FMC_BCR1 寄存器中 CCKEN 位FMC的配置，可决定只在同步访问期间传送还是同步异步访问期间都传送：

- 如果 CCLKEN 位置 0，则 FMC 仅在同步访问（读/写事务）期间生成时钟 (FMC_CLK)。
- 如果 CCLKEN 位置 1，则 FMC 将在异步和同步访问期间生成连续时钟。要生成 FMC_CLK 连续时钟，存储区域 1 还必须配置为支持同步模式（请参见*第 22.7.6 节：NOR/PSRAM 控制寄存器*）。由于所有同步存储器均使用同一个时钟，因此在生成连续输出时钟和执行同步访问时，AXI 数据大小必须与存储器数据宽度 (MWID) 相同，否则 FMC_CLK 频率将根据 AXI 数据事务发生变化（有关 FMC_CLK 分频比公式的信息，请参见*第 22.7.5 节：同步事务*）。

每个存储区域的大小固定，均为 64 MB。每个存储区域都通过专用的寄存器配置（请参见*第 22.7.6 节：NOR/PSRAM 控制寄存器*）。

存储器的可编程参数包括访问时间（请参见*表 148*）和对等待管理的支持（用于在突发模式下访问 NOR Flash 和 PSRAM）。

表 148. NOR/PSRAM 的可编程访问参数

参数	功能	访问模式	单位	最小值	最大值
地址建立	地址建立阶段的持续时间	异步	FMC 时钟周期 (fmc_ker_ck)	0	15
地址保持	地址保持阶段的持续时间	异步，复用 I/O	FMC 时钟周期 (fmc_ker_ck)	1	15
数据建立	数据建立阶段的持续时间	异步	FMC 时钟周期 (fmc_ker_ck)	1	256
总线周转	总线周转阶段的持续时间	异步和同步读取	FMC 时钟周期 (fmc_ker_ck)	0	15
时钟分频比	构建一个存储器时钟周期 (CLK) 所需的 FMC 时钟周期 (fmc_ker_ck) 数量	同步	FMC 时钟周期 (fmc_ker_ck)	2	16
数据延迟	在发出突发的第一个数据前向存储器发出的时钟周期数量	同步	存储器时钟周期 (fmc_ker_ck)	2	17

22.7.1 外部存储器接口信号

表 149、*表 150* 和*表 151* 列出了通常用于连接 NOR Flash、SRAM 和 PSRAM 的信号。

注： *前缀“N”标识低电平有效的信号。*

NOR Flash，非复用 I/O

表 149. 非复用 I/O NOR Flash

FMC信号名称	I/O	功能
CLK	O	时钟（用于同步访问）
A[25:0]	O	地址总线
D[31:0]	I/O	双向数据总线
NE[x]	O	片选，x = 1..4
NOE	O	输出使能
NWE	O	写入使能
NL(= NADV)	O	锁存使能（对于部分 NOR Flash 器件，此信号也称为地址有效 (NADV)）
NWAIT	I	FMC 的 NOR Flash 等待输入信号

最大容量为 512 Mb（26 个地址线）。

NOR Flash，16 位复用 I/O

表 150. 16 位复用 I/O NOR Flash

FMC信号名称	I/O	功能
CLK	O	时钟（用于同步访问）
A[25:16]	O	地址总线
AD[15:0]	I/O	16 位复用，双向地址/数据总线 （16 位地址 A[15:0] 和数据 D[15:0] 在数据总线上复用）
NE[x]	O	片选，x = 1..4
NOE	O	输出使能
NWE	O	写入使能
NL(= NADV)	O	锁存使能 （对于部分 NOR Flash 器件，此信号也称为地址有效 (NADV)）
NWAIT	I	FMC 的 NOR Flash 等待输入信号

最大容量为 512 Mb。

PSRAM/SRAM，非复用 I/O

表 151. 非复用 I/O PSRAM/SRAM

FMC信号名称	I/O	功能
CLK	O	时钟（仅用于 PSRAM 同步访问）
A[25:0]	O	地址总线
D[31:0]	I/O	数据双向总线
NE[x]	O	片选，x = 1..4 （在 PSRAM 应用中被称作 NCE（Cellular RAM，即 CRAM））
NOE	O	输出使能
NWE	O	写入使能
NL(= NADV)	O	仅用于 PSRAM 输入的地址有效信号（存储器信号名称：NADV）
NWAIT	I	FMC 的 PSRAM 等待输入信号
NBL[3:0]	O	字节通道输出。字节 0 到字节 3 控制（高字节和低字节使能）

最大容量为 512 Mb。

PSRAM，16 位复用 I/O

表 152. 16 位复用 I/O PSRAM

FMC信号名称	I/O	功能
CLK	O	时钟（用于同步访问）
A[25:16]	O	地址总线
AD[15:0]	I/O	16 位复用，双向地址/数据总线 （16 位地址 A[15:0] 和数据 D[15:0] 在数据总线上复用）
NE[x]	O	片选，x = 1..4 （在 PSRAM 应用中被称作 NCE（Cellular RAM，即 CRAM））
NOE	O	输出使能
NWE	O	写入使能
NL(= NADV)	O	用于 PSRAM 输入的地址有效信号（存储器信号名称：NADV）
NWAIT	I	FMC 的 PSRAM 等待输入信号
NBL[1:0]	O	字节通道输出。字节 0 和字节 1 控制（高字节和低字节使能）

最大容量为 512 Mb（26 个地址线）。

22.7.2 支持的存储器和事务

下面的*表 153* 显示的是当 NOR Flash、PSRAM 和 SRAM 的存储器数据总线宽度为 16 位时所支持的设备、访问模式和事务的示例。本示例中 FMC 不允许（或不支持）的事务以灰色显示。

表 153. NOR Flash/PSRAM：支持的存储器和事务示例(1)

设备	模式	R/W	AXI 数据大小	存储器数据大小	是否允许	注释
NOR Flash（复用 I/O 和非复用 I/O）	异步	R	8	16	是	
	异步	W	8	16	否	
	异步	R	16	16	是	
	异步	W	16	16	是	
	异步	R	32	16	是	分为 2 次 FMC 访问
	异步	W	32	16	是	分为 2 次 FMC 访问
	异步	R	64	16	是	分为 4 次 FMC 访问
	异步	W	64	16	是	分为 4 次 FMC 访问
	异步页	R	-	16	否	不支持该模式
	同步	R	8	16	否	
	同步	R	16	16	是	
	同步	R	32/64	16	是	
PSRAM（复用 I/O 和非复用 I/O）	异步	R	8	16	是	
	异步	W	8	16	是	使用字节通道 NBL[1:0]
	异步	R	16	16	是	
	异步	W	16	16	是	
	异步	R	32	16	是	分为 2 次 FMC 访问
	异步	W	32	16	是	分为 2 次 FMC 访问
	异步	R	64	16	是	分为 4 次 FMC 访问
	异步	W	64	16	是	分为 4 次 FMC 访问
	异步页	R	-	16	否	不支持该模式
	同步	R	8	16	否	
	同步	R	16	16	是	
	同步	R	32/64	16	是	
	同步	W	8	16	是	使用字节通道 NBL[1:0]
	同步	W	16/32/64	16	是	

表 153. NOR Flash/PSRAM：支持的存储器和事务示例(1)（续）

设备	模式	R/W	AXI 数据大小	存储器数据大小	是否允许	注释
SRAM 和 ROM	异步	R	8/16	16	是	
	异步	W	8/16	16	是	使用字节通道 NBL[1:0]
	异步	R	32	16	是	分为 2 次 FMC 访问
	异步	W	32	16	是	分为 2 次 FMC 访问，使用字节通道 NBL[1:0]
	异步	R	64	16	是	分为 4 次 FMC 访问
	异步	W	64	16	是	分为 4 次 FMC 访问，使用字节通道 NBL[1:0]

1. NBL[1:0] 同样通过 AXI 写入选通驱动。

22.7.3 通用时序规则

信号同步执行如下：

- 所有的控制器输出信号均在 fmc_ker_ck 时钟的上升沿变化。
- 在同步读取和写入模式下，全部输出信号均在 fmc_ker_ck 时钟的上升沿变化。无论 CLKDIV 值为何，所有输出均会按以下方式变化：
  - NOEL/NWEL/NEL/NADVL/NADVH/NBLL/ 地址有效输出在 FMC_CLK 时钟的下降沿变化。
  - NOEH/NWEH/NEH/NOEH/NBLH/ 地址有效输出在 FMC_CLK 时钟的上升沿变化。

22.7.4 NOR Flash/PSRAM 控制器异步事务

静态存储器（NOR Flash、PSRAM、SRAM）上的异步事务执行如下：

- 信号通过内部时钟进行同步。不会将此时钟发送到存储器
- FMC 总是会先对数据进行采样，而后再禁止片选信号。这样可以确保符合存储器数据保持时序的要求（数据转换的最小芯片使能高电平通常为 0 ns）
- 如果使能扩展模式（FMC_BCRx 寄存器中的 EXTMOD 位置 1），则最多可提供四种扩展模式（A、B、C 和 D）。可以混合使用 A、B、C 和 D 模式来进行读取和写入操作。例如，可以在模式 A 下执行读取操作，而在模式 B 下执行写入操作。
- 如果禁用扩展模式（FMC_BCRx 寄存器中的 EXTMOD 位复位），则 FMC 可以在模式 1 或模式 2 下运行，如下所述：
  - 当选择 SRAM/PSRAM 存储器类型时，模式 1 为默认模式（FMC_BCRx 寄存器中 MTYP = 0x0 或 0x01）。
  - 当选择 NOR 存储器类型时，模式 2 为默认模式（FMC_BCRx 寄存器中 MTYP = 0x10）。

模式 1 - SRAM/PSRAM (CRAM)

下图显示了所支持模式的读取和写入事务以及所需的 FMC _BCRx 和 FMC_BTRx/FMC_BWTRx 寄存器配置。

图 89. 模式 1 读取访问波形

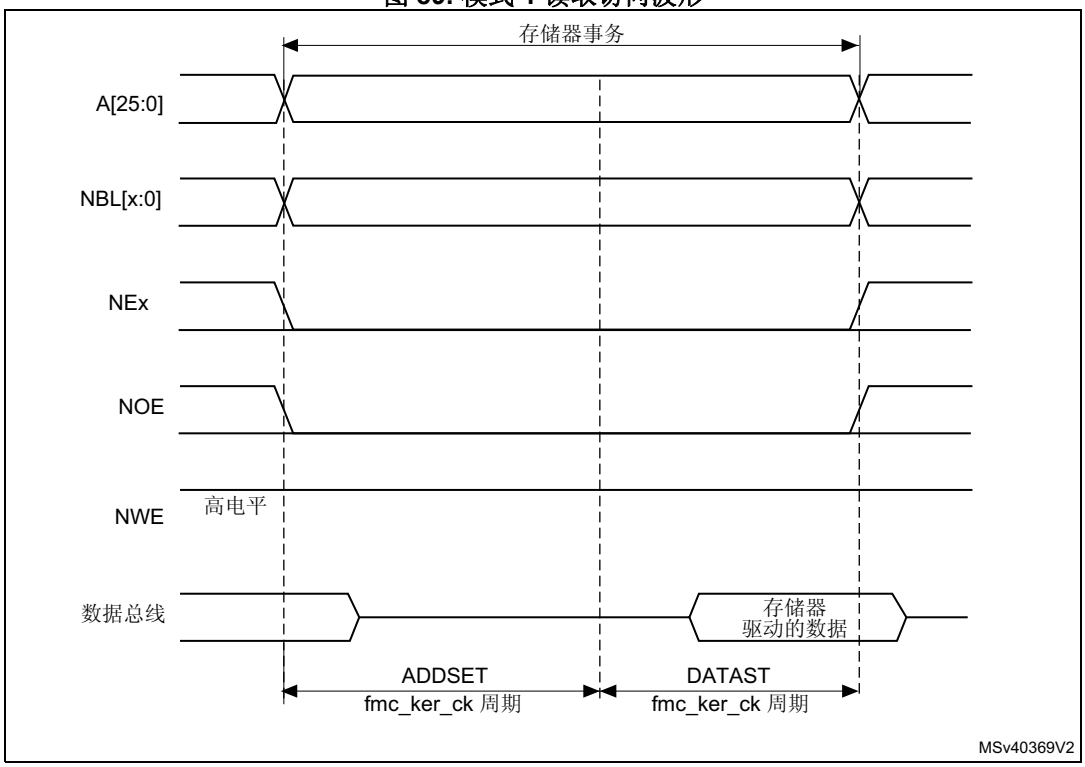
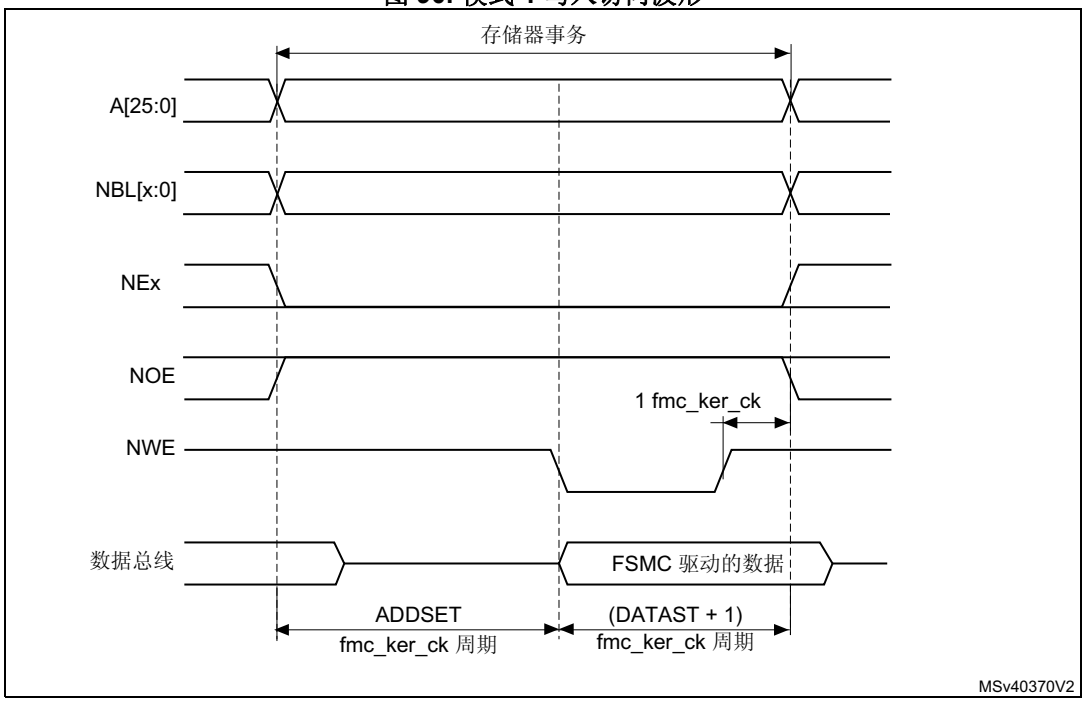


图 90. 模式 1 写入访问波形



位于写入事务末尾的一个 fmc_ker_ck 周期有助于确保 NWE 上升沿之后的地址和数据保持时间。由于存在此 fmc_ker_ck 周期，DATAST 值必须大于零 (DATAST > 0)。

表 154. FMC_BCRx 位域

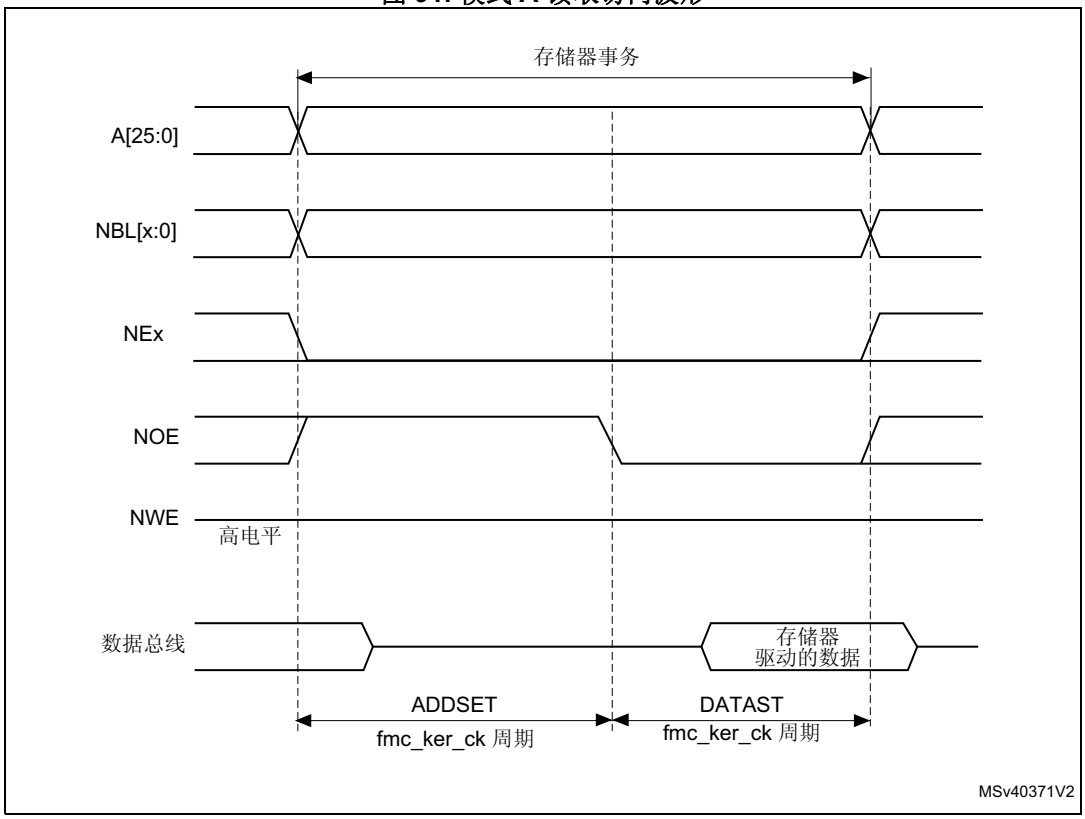
位号	位名	要设置的值
31	FMCEN	0x0
30-26	保留	0x000
25-24	BMAP	根据需要进行设置
23-22	保留	0x000
21	WFDIS	根据需要进行设置
20	CCLKEN	根据需要进行设置
19	CBURSTRW	0x0（对异步模式没有影响）
18:16	CPSIZE	0x0（对异步模式没有影响）
15	ASYNCWAIT	如果存储器支持该特性，则置为 1。否则，保持为 0。
14	EXTMOD	0x0
13	WAITEN	0x0（对异步模式没有影响）
12	WREN	根据需要进行设置
11	WAITCFG	无关
10	保留	0x0
9	WAITPOL	仅当位 15 为 1 时才有意义
8	BURSTEN	0x0
7	保留	0x1
6	FACCEN	无关
5-4	MWID	根据需要进行设置
3-2	MTYP	根据需要进行设置，0x2 除外 (NOR Flash)
1	MUXE	0x0
0	MBKEN	0x1

表 155. FMC_BTRx 位域

位号	位名	要设置的值
31:30	保留	0x0
29-28	ACCMOD	无关
27-24	DATLAT	无关
23-20	CLKDIV	无关
19-16	BUSTURN	NEx 变为高电平到 NEx 变为低电平之间的时间 (BUSTURN fmc_ker_ck)
15-8	DATAST	第二个访问阶段的持续时间（写入访问为 DATAST+1 个 fmc_ker_ck 周期，读取访问为 DATAST 个 fmc_ker_ck 周期）。
7-4	ADDHLD	无关
3-0	ADDSET	第一个访问阶段的持续时间（ADDSET 个 fmc_ker_ck 周期）。ADDSET 最小值为 0。

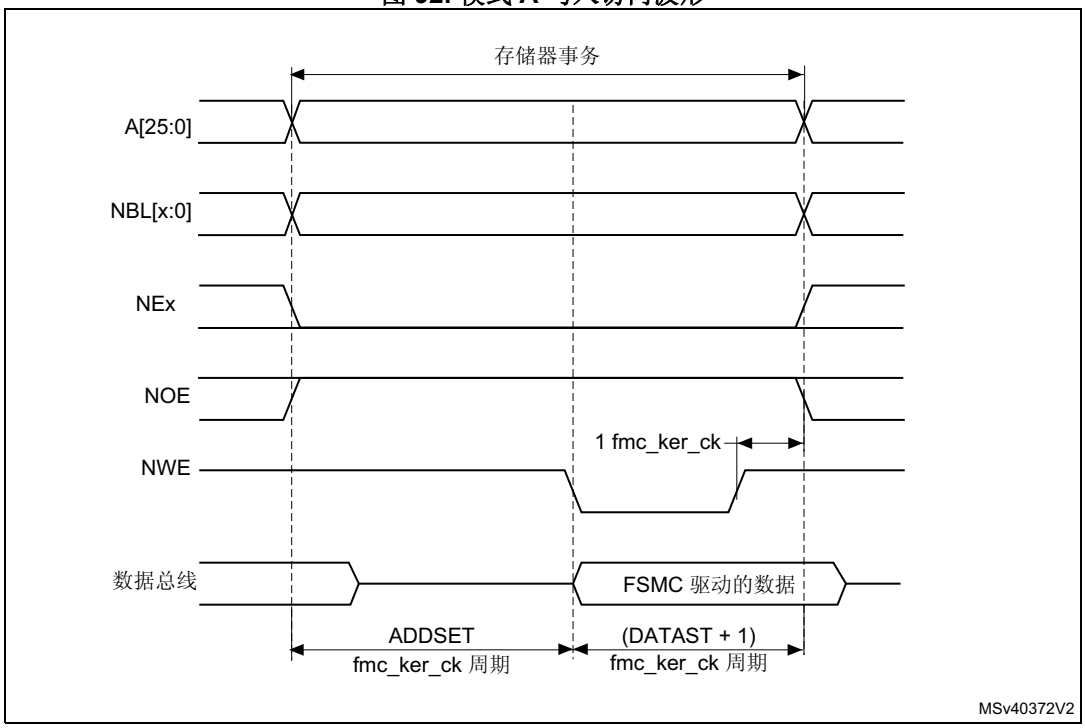
模式 A - SRAM/PSRAM (CRAM) OE 切换

图 91. 模式 A 读取访问波形



1. NBL[3:0] 在进行读取访问时为低电平

图 92. 模式 A 写入访问波形



与模式 1 的不同之处在于 NOE 的切换与独立的读取和写入时序。

表 156. FMC_BCRx 位域

位号	位名	要设置的值
31	FMCEN	0x0
30-26	保留	0x000
25-24	BMAP	根据需要进行设置
23-22	保留	0x000
21	WFDIS	根据需要进行设置
20	CCLKEN	根据需要进行设置
19	CBURSTRW	0x0（对异步模式没有影响）
18:16	CPSIZE	0x0（对异步模式没有影响）
15	ASYNCWAIT	如果存储器支持该特性，则置为 1。否则，保持为 0。
14	EXTMOD	0x1
13	WAITEN	0x0（对异步模式没有影响）
12	WREN	根据需要进行设置
11	WAITCFG	无关
10	保留	0x0

表 156. FMC_BCRx 位域（续）

位号	位名	要设置的值
9	WAITPOL	仅当位 15 为 1 时才有意义
8	BURSTEN	0x0
7	保留	0x1
6	FACCEN	无关
5-4	MWID	根据需要进行设置
3-2	MTYP	根据需要进行设置，0x2 除外 (NOR Flash)
1	MUXEN	0x0
0	MBKEN	0x1

表 157. FMC_BTRx 位域

位号	位名	要设置的值
31:30	保留	0x0
29-28	ACCMOD	0x0
27-24	DATLAT	无关
23-20	CLKDIV	无关
19-16	BUSTURN	NEx 变为高电平到 NEx 变为低电平之间的时间 (BUSTURN fmc_ker_ck)
15-8	DATAST	读取访问第二个阶段的持续时间（DATAST 个 fmc_ker_ck 周期）。
7-4	ADDHLD	无关
3-0	ADDSET	读取访问第一个阶段的持续时间（ADDSET 个 fmc_ker_ck 周期）。ADDSET 最小值为 0。

表 158. FMC_BWTRx 位域

位号	位名	要设置的值
31:30	保留	0x0
29-28	ACCMOD	0x0
27-24	DATLAT	无关
23-20	CLKDIV	无关
19-16	BUSTURN	NEx 变为高电平到 NEx 变为低电平之间的时间 (BUSTURN fmc_ker_ck)
15-8	DATAST	写入访问第二个阶段的持续时间（DATAST 个 fmc_ker_ck 周期）。
7-4	ADDHLD	无关
3-0	ADDSET	写入访问第一个阶段的持续时间（ADDSET 个 fmc_ker_ck 周期）。ADDSET 最小值为 0。

模式 2/B - NOR Flash

图 93. 模式 2 和模式 B 读取访问波形

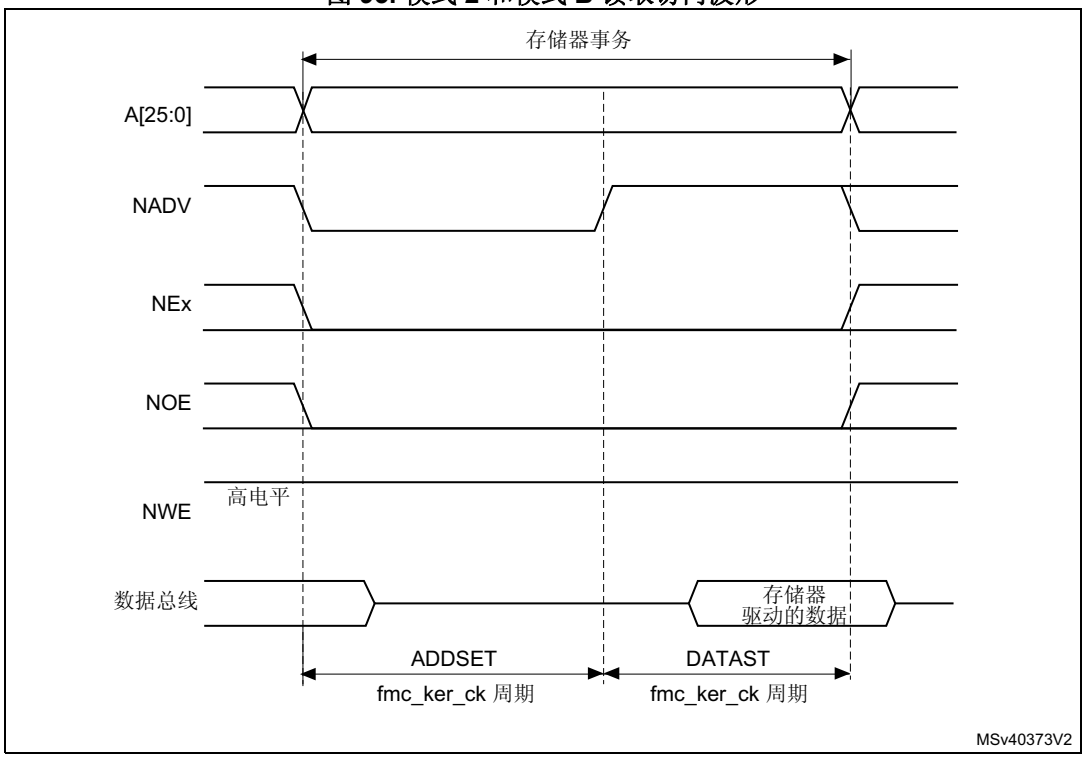


图 94. 模式 2 写入访问波形

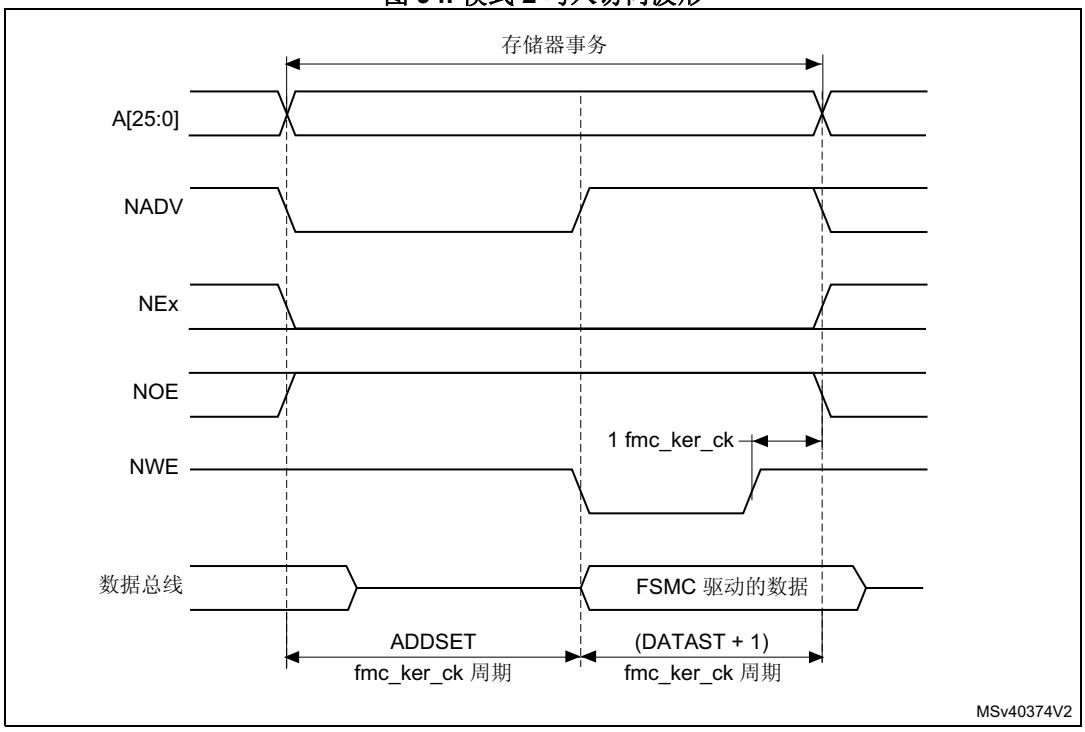
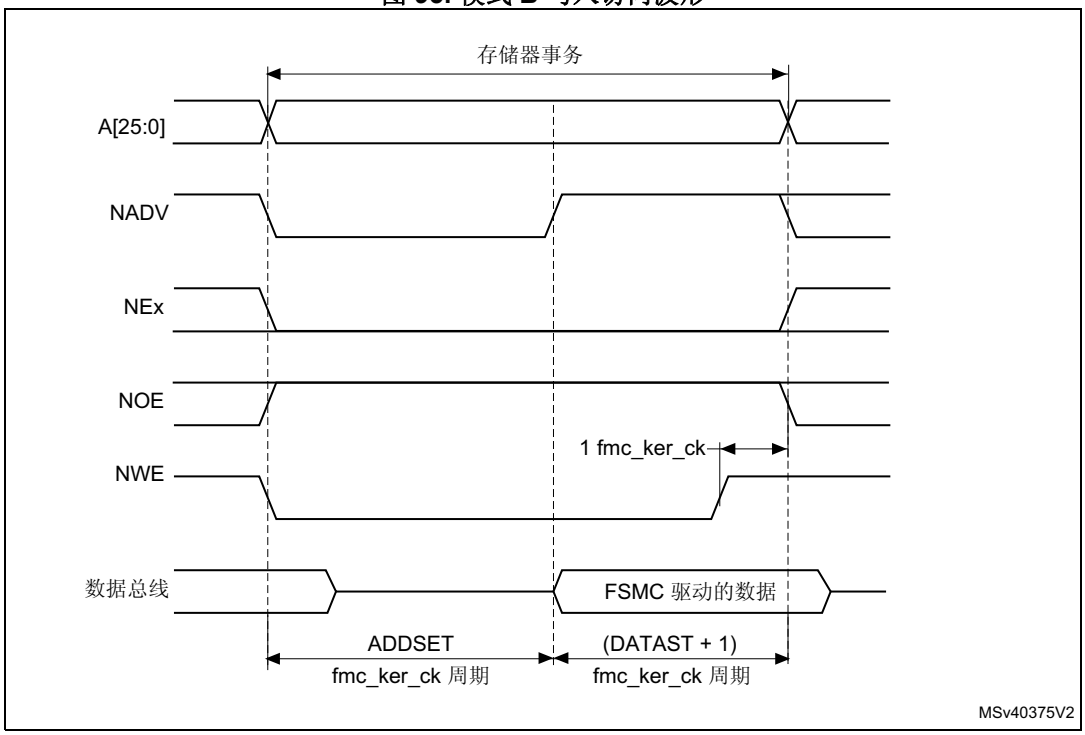


图 95. 模式 B 写入访问波形



与模式 1 的不同之处在于设置扩展模式（模式 B）时的 NWE 切换与独立的读取和写入时序。

表 159. FMC_BCRx 位域

位号	位名	要设置的值
31	FMCEN	0x0
30-26	保留	0x000
25-24	BMAP	根据需要进行设置
23-22	保留	0x000
21	WFDIS	根据需要进行设置
20	CCLKEN	根据需要进行设置
19	CBURSTRW	0x0（对异步模式没有影响）
18:16	CPSIZE	0x0（对异步模式没有影响）
15	ASYNCWAIT	如果存储器支持该特性，则置为 1。否则，保持为 0。
14	EXTMOD	模式 B 为 0x1，模式 2 为 0x0
13	WAITEN	0x0（对异步模式没有影响）
12	WREN	根据需要进行设置
11	WAITCFG	无关
10	保留	0x0

表 159. FMC_BCRx 位域（续）

位号	位名	要设置的值
9	WAITPOL	仅当位 15 为 1 时才有意义
8	BURSTEN	0x0
7	保留	0x1
6	FACCEN	0x1
5-4	MWID	根据需要进行设置
3-2	MTYP	0x2 (NOR Flash)
1	MUXEN	0x0
0	MBKEN	0x1

表 160. FMC_BTRx 位域

位号	位名	要设置的值
31-30	保留	0x0
29-28	ACCMOD	设置了扩展模式时为 0x1
27-24	DATLAT	无关
23-20	CLKDIV	无关
19-16	BUSTURN	NEx 变为高电平到 NEx 变为低电平之间的时间 (BUSTURN fmc_ker_ck)
15-8	DATAST	读取访问第二个阶段的持续时间（DATAST 个 fmc_ker_ck 周期）。
7-4	ADDHLD	无关
3-0	ADDSET	读取访问第一个阶段的持续时间（ADDSET 个 fmc_ker_ck 周期）。ADDSET 最小值为 0。

表 161. FMC_BWTRx 位域

位号	位名	要设置的值
31-30	保留	0x0
29-28	ACCMOD	设置了扩展模式时为 0x1
27-24	DATLAT	无关
23-20	CLKDIV	无关
19-16	BUSTURN	NEx 变为高电平到 NEx 变为低电平之间的时间 (BUSTURN fmc_ker_ck)
15-8	DATAST	写入访问第二个阶段的持续时间（DATAST 个 fmc_ker_ck 周期）。
7-4	ADDHLD	无关
3-0	ADDSET	写入访问第一个阶段的持续时间（ADDSET 个 fmc_ker_ck 周期）。ADDSET 最小值为 0。

注：　仅当设置了扩展模式（模式 B）时，FMC_BWTRx 寄存器才有效，否则其内容均为“无关”。

模式 C - NOR Flash - OE 切换

图 96. 模式 C 读取访问波形

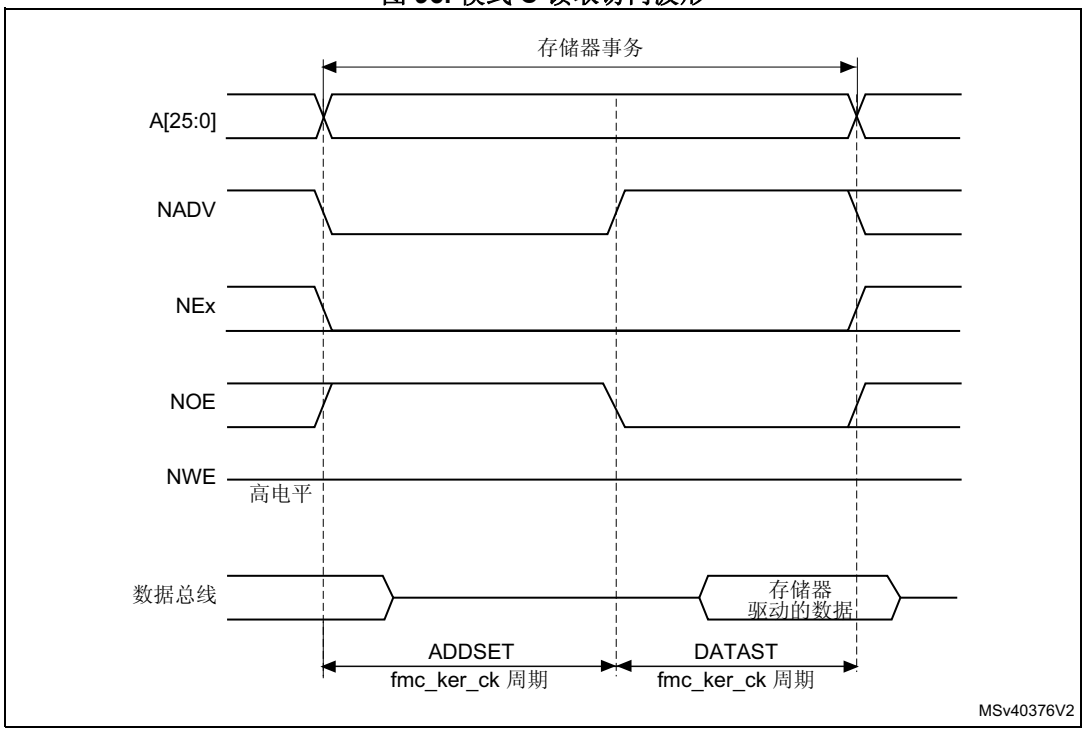
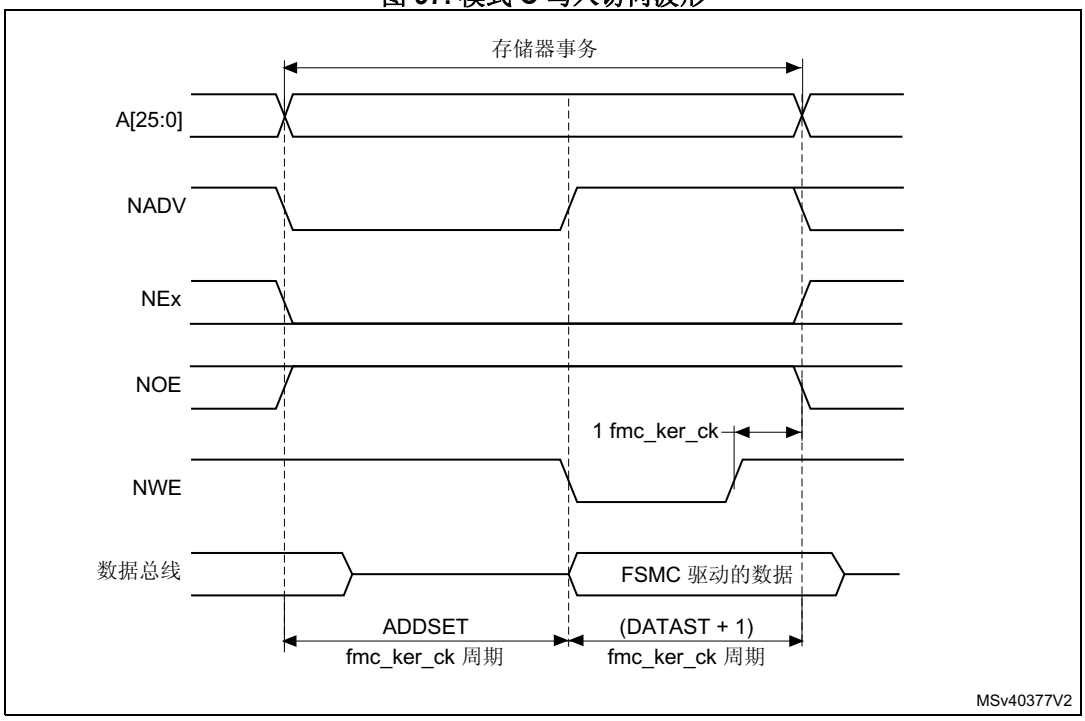


图 97. 模式 C 写入访问波形



与模式 1 的不同之处在于 NOE 的切换与独立的读取和写入时序。

表 162. FMC_BCRx 位域

位号	位名	要设置的值
31	FMCEN	0x0
30-26	保留	0x000
25-24	BMAP	根据需要进行设置
23-22	保留	0x000
21	WFDIS	根据需要进行设置
20	CCLKEN	根据需要进行设置
19	CBURSTRW	0x0（对异步模式没有影响）
18:16	CPSIZE	0x0（对异步模式没有影响）
15	ASYNCWAIT	如果存储器支持该特性，则置为 1。否则，保持为 0。
14	EXTMOD	0x1
13	WAITEN	0x0（对异步模式没有影响）
12	WREN	根据需要进行设置
11	WAITCFG	无关
10	保留	0x0
9	WAITPOL	仅当位 15 为 1 时才有意义
8	BURSTEN	0x0
7	保留	0x1
6	FACCEN	0x1
5-4	MWID	根据需要进行设置
3-2	MTYP	0x02 (NOR Flash)
1	MUXEN	0x0
0	MBKEN	0x1

表 163. FMC_BTRx 位域

位号	位名	要设置的值
31:30	保留	0x0
29-28	ACCMOD	0x2
27-24	DATLAT	0x0
23-20	CLKDIV	0x0
19-16	BUSTURN	NEx 变为高电平到 NEx 变为低电平之间的时间 (BUSTURN fmc_ker_ck)
15-8	DATAST	读取访问第二个阶段的持续时间（DATAST 个 fmc_ker_ck 周期）。
7-4	ADDHLD	无关
3-0	ADDSET	读取访问第一个阶段的持续时间（ADDSET 个 fmc_ker_ck 周期）。 ADDSET 最小值为 0。

表 164. FMC_BWTRx 位域

位号	位名	要设置的值
31:30	保留	0x0
29-28	ACCMOD	0x2
27-24	DATLAT	无关
23-20	CLKDIV	无关
19-16	BUSTURN	NEx 变为高电平到 NEx 变为低电平之间的时间 (BUSTURN fmc_ker_ck)
15-8	DATAST	写入访问第二个阶段的持续时间（DATAST 个 fmc_ker_ck 周期）。
7-4	ADDHLD	无关
3-0	ADDSET	写入访问第一个阶段的持续时间（ADDSET 个 fmc_ker_ck 周期）。 ADDSET 最小值为 0。

模式 D - 扩展地址异步访问

图 98. 模式 D 读取访问波形

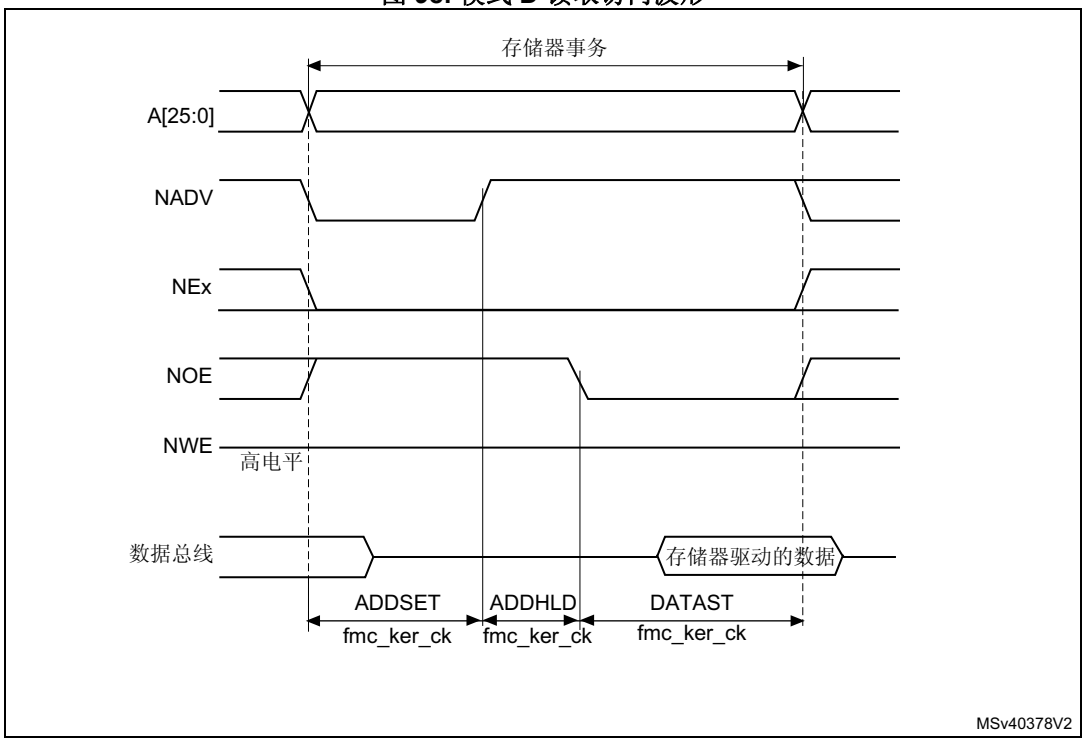
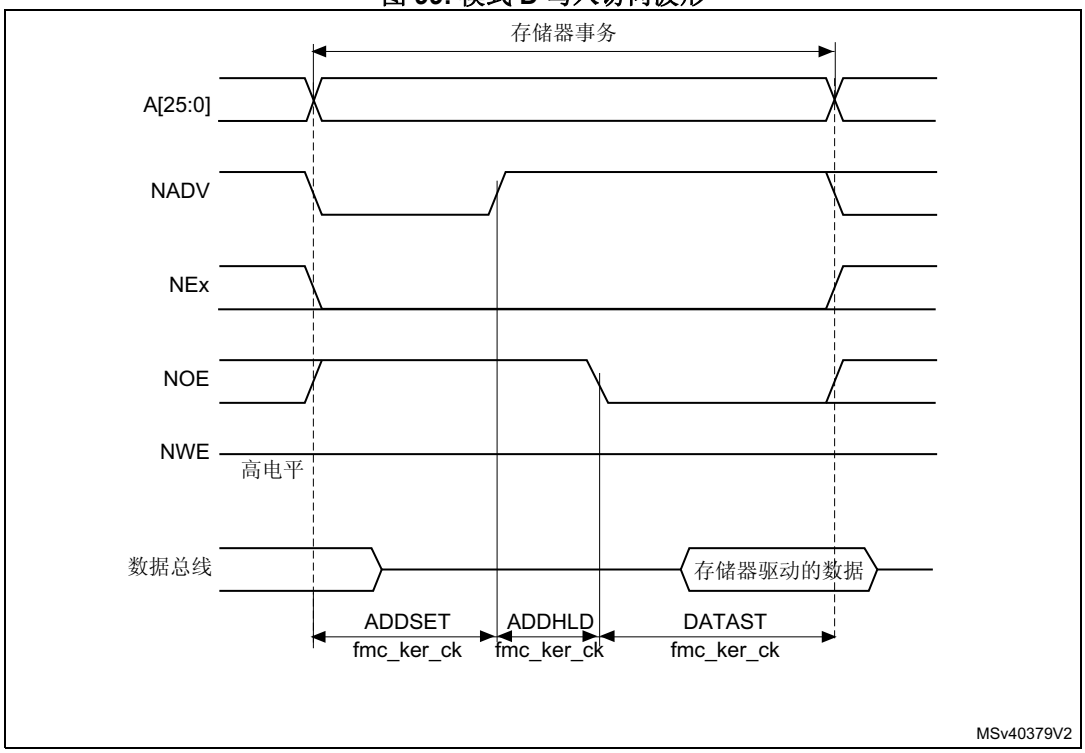


图 99. 模式 D 写入访问波形



与模式 1 的不同之处在于 NADV 变化后 NOE 的切换与独立的读取和写入时序。

表 165. FMC_BCRx 位域

位号	位名	要设置的值
31	FMCEN	0x0
30-26	保留	0x000
25-24	BMAP	根据需要进行设置
23-22	保留	0x000
21	WFDIS	根据需要进行设置
20	CCLKEN	根据需要进行设置
19	CBURSTRW	0x0（对异步模式没有影响）
18:16	CPSIZE	0x0（对异步模式没有影响）
15	ASYNCWAIT	如果存储器支持该特性，则置为 1。否则，保持为 0。
14	EXTMOD	0x1
13	WAITEN	0x0（对异步模式没有影响）
12	WREN	根据需要进行设置
11	WAITCFG	无关
10	保留	0x0
9	WAITPOL	仅当位 15 为 1 时才有意义
8	BURSTEN	0x0
7	保留	0x1
6	FACCEN	根据存储器支持情况进行设置
5-4	MWID	根据需要进行设置
3-2	MTYP	根据需要进行设置
1	MUXEN	0x0
0	MBKEN	0x1

表 166. FMC_BTRx 位域

位号	位名	要设置的值
31:30	保留	0x0
29-28	ACCMOD	0x3
27-24	DATLAT	无关
23-20	CLKDIV	无关
19-16	BUSTURN	NEx 变为高电平到 NEx 变为低电平之间的时间 (BUSTURN fmc_ker_ck)
15-8	DATAST	读取访问第二个阶段的持续时间（DATAST 个 fmc_ker_ck 周期）。
7-4	ADDHLD	读取访问中间阶段的持续时间（ADDHLD 个 fmc_ker_ck 周期）。
3-0	ADDSET	读取访问第一个阶段的持续时间（ADDSET 个 fmc_ker_ck 周期）。ADDSET 的最小值为 1。

表 167. FMC_BWTRx 位域

位号	位名	要设置的值
31:30	保留	0x0
29-28	ACCMOD	0x3
27-24	DATLAT	无关
23-20	CLKDIV	无关
19-16	BUSTURN	NEx 变为高电平到 NEx 变为低电平之间的时间 (BUSTURN fmc_ker_ck)
15-8	DATAST	写入访问第二个阶段的持续时间（DATAST + 1 个 fmc_ker_ck 周期）
7-4	ADDHLD	写入访问中间阶段的持续时间（ADDHLD 个 fmc_ker_ck 周期）
3-0	ADDSET	写入访问第一个阶段的持续时间（ADDSET 个 fmc_ker_ck 周期）。ADDSET 的最小值为 1。

复用模式 - 复用异步访问 NOR Flash

图 100. 复用读取访问波形

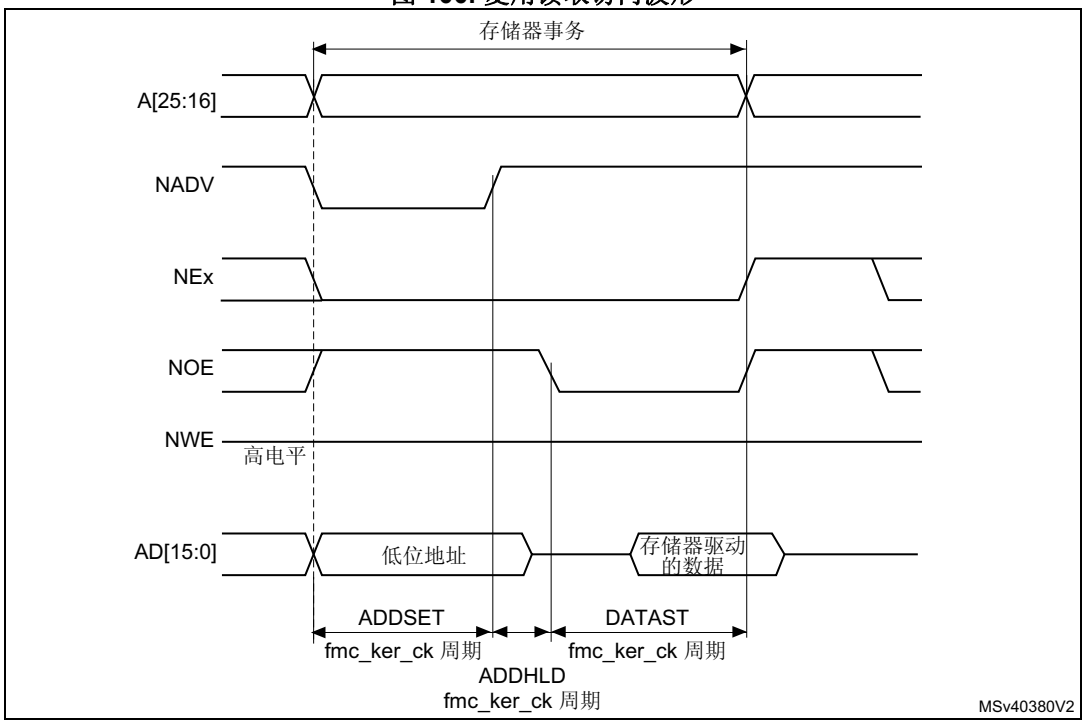
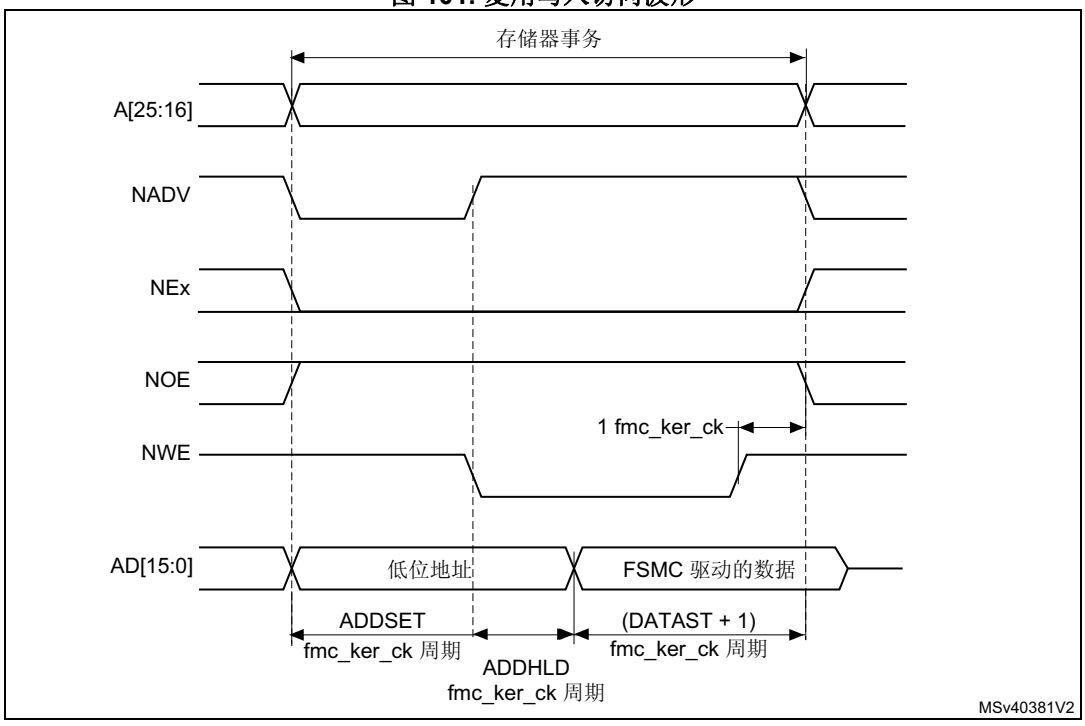


图 101. 复用写入访问波形



与模式 D 的不同之处在于在数据总线上驱动低地址字节。

表 168. FMC_BCRx 位域

位号	位名	要设置的值
31	FMCEN	0x0
30-26	保留	0x000
25-24	BMAP	根据需要进行设置
23-22	保留	0x000
21	WFDIS	根据需要进行设置
20	CCLKEN	根据需要进行设置
19	CBURSTRW	0x0（对异步模式没有影响）
18:16	CPSIZE	0x0（对异步模式没有影响）
15	ASYNCWAIT	如果存储器支持该特性，则置为 1。否则，保持为 0。
14	EXTMOD	0x0
13	WAITEN	0x0（对异步模式没有影响）
12	WREN	根据需要进行设置
11	WAITCFG	无关
10	保留	0x0
9	WAITPOL	仅当位 15 为 1 时才有意义
8	BURSTEN	0x0
7	保留	0x1
6	FACCEN	0x1
5-4	MWID	根据需要进行设置
3-2	MTYP	0x2 (NOR Flash)
1	MUXEN	0x1
0	MBKEN	0x1

表 169. FMC_BTRx 位域

位号	位名	要设置的值
31:30	保留	0x0
29-28	ACCMOD	0x0
27-24	DATLAT	无关
23-20	CLKDIV	无关
19-16	BUSTURN	NEx 变为高电平到 NEx 变为低电平之间的时间 (BUSTURN fmc_ker_ck)
15-8	DATAST	第二个访问阶段的持续时间（读取访问为 DATAST 个 fmc_ker_ck 周期，写入访问为 DATAST+1 个 fmc_ker_ck 周期）。
7-4	ADDHLD	访问中间阶段的持续时间（ADDHLD 个 fmc_ker_ck 周期）。
3-0	ADDSET	第一个访问阶段的持续时间（ADDSET 个 fmc_ker_ck 周期）。 ADDSET 的最小值为 1。

异步访问中的 WAIT 管理

如果异步存储器发出 WAIT 信号，指示尚未准备好接受或提供数据，则 FMC_BCRx 寄存器中的 ASYNCWAIT 位必须置 1。

如果 WAIT 信号处于有效状态（电平高低取决于 WAITPOL 位），则由 DATAST 位控制的第二个访问阶段（数据建立阶段）将延长，直到 WAIT 变为无效状态。与数据建立阶段不同，由 ADDSET 和 ADDHLD 位控制的第一个访问阶段（地址建立和地址保持阶段）对 WAIT 不敏感，因此第一个访问阶段不会延长。

必须配置数据建立阶段，以便在存储器事务结束前 4 个 fmc_ker_ck 周期检测到 WAIT。必须考虑以下情况：

1. 存储器发出的 WAIT 信号和 NOE/NWE 信号对齐：

$$DATAST \geq (4 \times FMC_CLK) + max_wait_assertion_time$$

2. 存储器发出的 WAIT 信号和 NEx 对齐（或者 NOE/NWE 信号不翻转）：

   如果

$$max_wait_assertion_time > address_phase + hold_phase$$

   那么：

$$DATAST \geq (4 \times FMC_CLK) + (max_wait_assertion_time - address_phase - hold_phase)$$

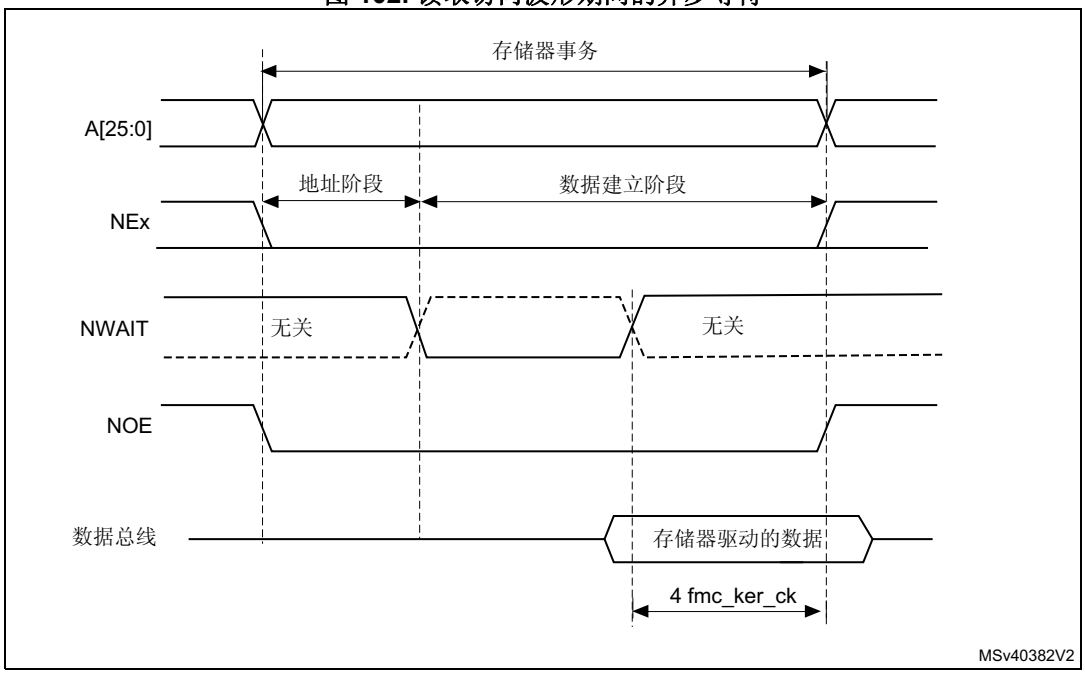
   否则

$$DATAST \geq (4 \times FMC_CLK)$$

   其中，max_wait_assertion_time 是在 NEx/NOE/NWE 变为低电平后存储器使能 WAIT 信号所花费的最长时间。

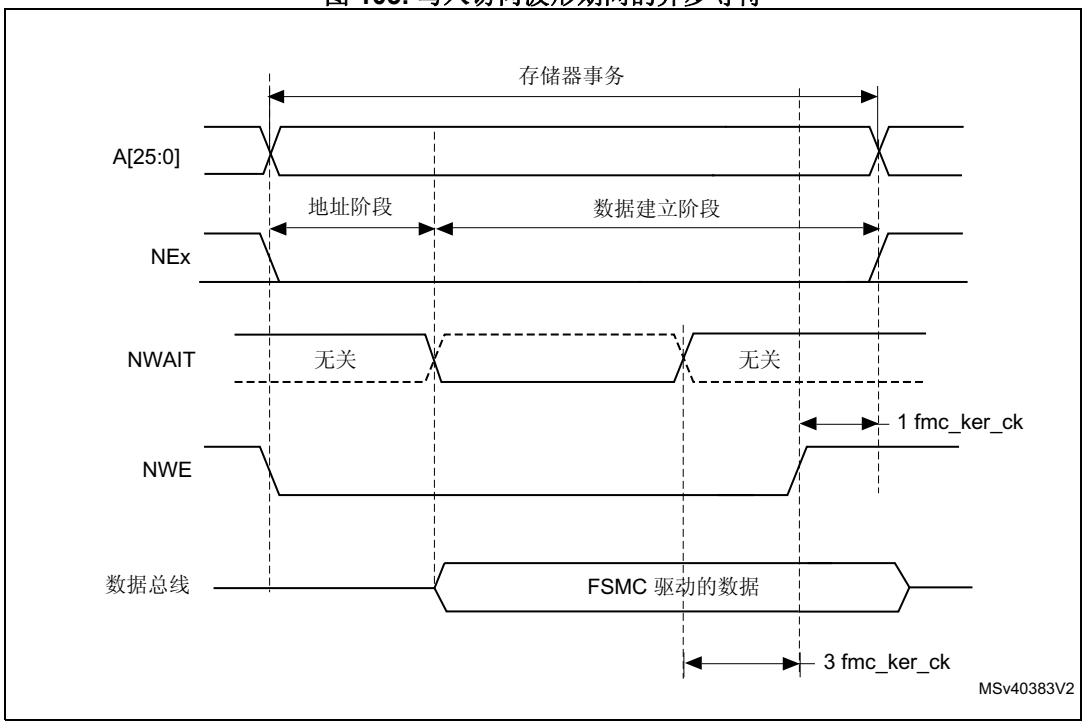
图 102 和*图 103* 显示了异步存储器释放 WAIT 之后，在存储器访问阶段增加的 fmc_ker_ck 时钟周期的个数（与上述情况无关）。

图 102. 读取访问波形期间的异步等待



1. NWAIT 极性取决于 FMC_BCRx 寄存器中的 WAITPOL 位设置。

图 103. 写入访问波形期间的异步等待



1. NWAIT 极性取决于 FMC_BCRx 寄存器中的 WAITPOL 位设置。

22.7.5 同步事务

存储器时钟 FMC_CLK 是 fmc_ker_ck 的约数。它取决于 CLKDIV 的值和 MWID/AXI 数据大小，如以下公式所示：

$$\text{FMC_CLK 分频比} = \max(\text{CLKDIV} + 1, \text{MWID(AXI 数据大小)})$$

MWID 为 16 位或 8 位时，FMC_CLK 分频比始终由 CLKDIV 的设定值定义。

MWID 为 32 位时，FMC_CLK 分频比还取决于 AXI 数据大小。

示例：

- CLKDIV = 1，MWID = 32 位，AXI 数据大小 = 8 位时，FMC_CLK = fmc_ker_ck/4。
- CLKDIV = 1，MWID = 16 位，AXI 数据大小 = 8 位时，FMC_CLK = fmc_ker_ck/2。

NOR Flash 指定了从 NADV 使能到 FMC_CLK 高电平的最短时间。为了符合这一限制，FMC 不会在同步访问的第一个内部时钟周期内（NADV 使能之前）将时钟发到存储器。这样可以确保存储器时钟的上升沿出现在 NADV 低脉冲的中间。

对于必须配置为同步模式的一些 PSRAM 存储器，在 BCR 寄存器写入期间，存储器属性空间必须配置为器件或强序模式。一旦配置了 PSRAM BCR 寄存器，PSRAM 地址空间的存储器属性即可编程为可缓存。

数据延迟与 NOR 延迟

数据延迟是对数据进行采样之前需要等待的周期数。DATLAT 的值必须与 NOR FLASH 配置寄存器中指定的延迟值一致。当数据延迟计数中 NADV 为低电平时，FMC 不会计入时钟周期。

注意： 一些 NOR Flash 将 NADV 低电平周期计入数据延迟计数，这样 NOR Flash 延迟和 FMC DATLAT 参数之间的确切关系可以是以下任一种：

- NOR Flash 延迟 = (DATLAT + 2) 个 FMC_CLK 时钟周期
- NOR Flash 延迟 = (DATLAT + 3) 个 FMC_CLK 时钟周期

近来有一些存储器会在延迟阶段使能 NWAIT。在这种情况下，可以将 DATLAT 设置为最小值。然后，FMC 会对数据进行采样，并且等待足够长的时间来评估数据是否有效。这样，FMC 就能检测到存储器存在延迟的时间，从而处理真实数据。

其他存储器不会在延迟期间使能 NWAIT。在这种情况下，必须正确设置 FMC 和存储器的延迟，否则可能会将无效数据误用为有效数据，或者在存储器访问初始阶段丢失有效数据。

单次突发传输

当所选存储区域配置为同步突发模式时，例如，如果向 16 位存储器请求了一个单次突发事务，则 FMC 会执行长度为 1 的突发事务（如果 AXI 传输为 16 位）或者长度为 2 的突发事务（如果 AXI 传输为 32 位），然后在最后一个数据选通时禁止片选信号。

与异步读取操作相比，就周期而言这并不是最有效的传输方法。但是，随机异步读取需要先重新编程存储器访问模式，这样总时间会更长。

越过 Cellular RAM 1.5 的边界页

Cellular RAM 1.5 不允许突发访问越过页边界。根据存储器页大小配置 FMC_BCR1 寄存器中的 CPSIZE 位来达到存储器页大小时，FMC 控制器会自动分离突发访问。

等待管理

对于同步 NOR Flash，会在配置的延迟周期（相当于 (DATLAT+2) 个 FMC_CLK 时钟周期）之后对 NWAIT 进行评估。

如果 NWAIT 有效（WAITPOL = 0 时为低电平，WAITPOL = 1 时为高电平），会插入等待状态，直到 NWAIT 无效（WAITPOL = 0 时为高电平，WAITPOL = 1 时为低电平）。

当 NWAIT 无效时，数据将立即（位 WAITCFG = 1）或在下一个时钟边沿（位 WAITCFG = 0）被视为有效。

通过 NWAIT 信号插入等待周期期间，控制器会继续将时钟脉冲发送到存储器，保持片选和输出使能信号有效。但不将数据视为有效。

突发模式下，NOR Flash NWAIT 信号有两种时序配置：

- Flash 在等待周期之前一个数据周期发出 NWAIT 信号（复位后的默认值）
- Flash 在等待周期期间发出 NWAIT 信号

FMC 支持这两种 NOR Flash 等待周期配置，通过 FMC_BCRx 寄存器的 WAITCFG 位 (x = 0..3) 针对每个片选进行配置。

图 104. 等待配置波形

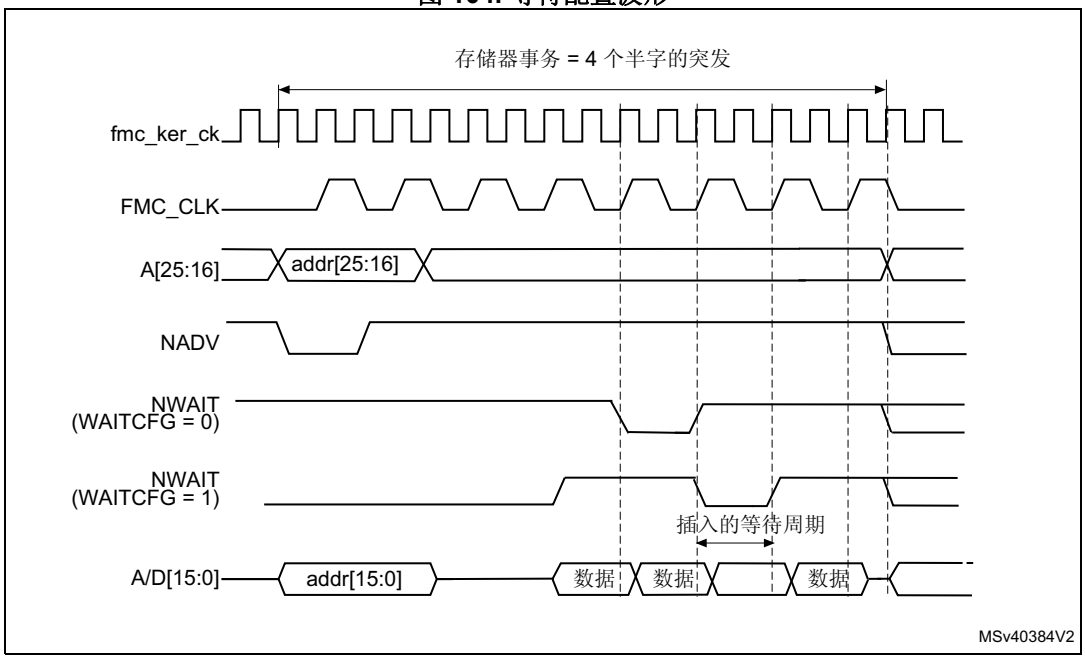
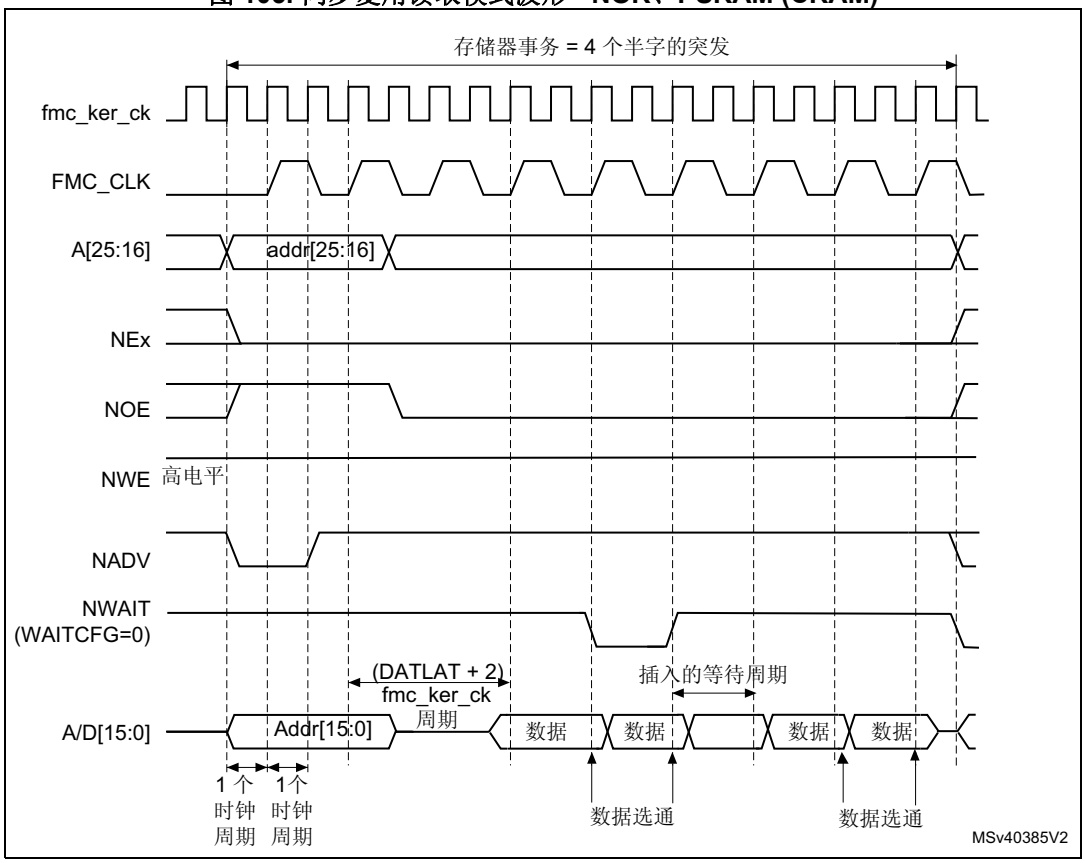


图 105. 同步复用读取模式波形 - NOR、PSRAM (CRAM)



1. 字节通道输出 NBL 未显示，它们对于 NOR 访问保持高电平，对于 PSRAM (CRAM) 访问则保持低电平。

表 170. FMC_BCRx 位域

位号	位名	要设置的值
31	MC	0x1
30-26	保留	0x000
25-24	BMAP	根据需要进行设置
23-22	保留	0x000
21	WFDIS	根据需要进行设置
20	CCLKEN	根据需要进行设置
19	CBURSTRW	对同步读取没有影响
18:16	CPSIZE	根据需要进行设置（使用 CRAM 1.5 时为 0x1）
15	ASYNCWAIT	0x0
14	EXTMOD	0x0
13	WAITEN	在存储器支持该特性的情况下置位为 1，否则保持为 0
12	WREN	对同步读取没有影响
11	WAITCFG	是否置位视存储器情况而定
10	保留	0x0
9	WAITPOL	是否置位视存储器情况而定
8	BURSTEN	0x1
7	保留	0x1
6	FACCEN	在存储器支持的情况下置位 (NOR Flash)
5-4	MWID	根据需要进行设置
3-2	MTYP	0x1 或 0x2
1	MUXEN	根据需要进行设置
0	MBKEN	0x1

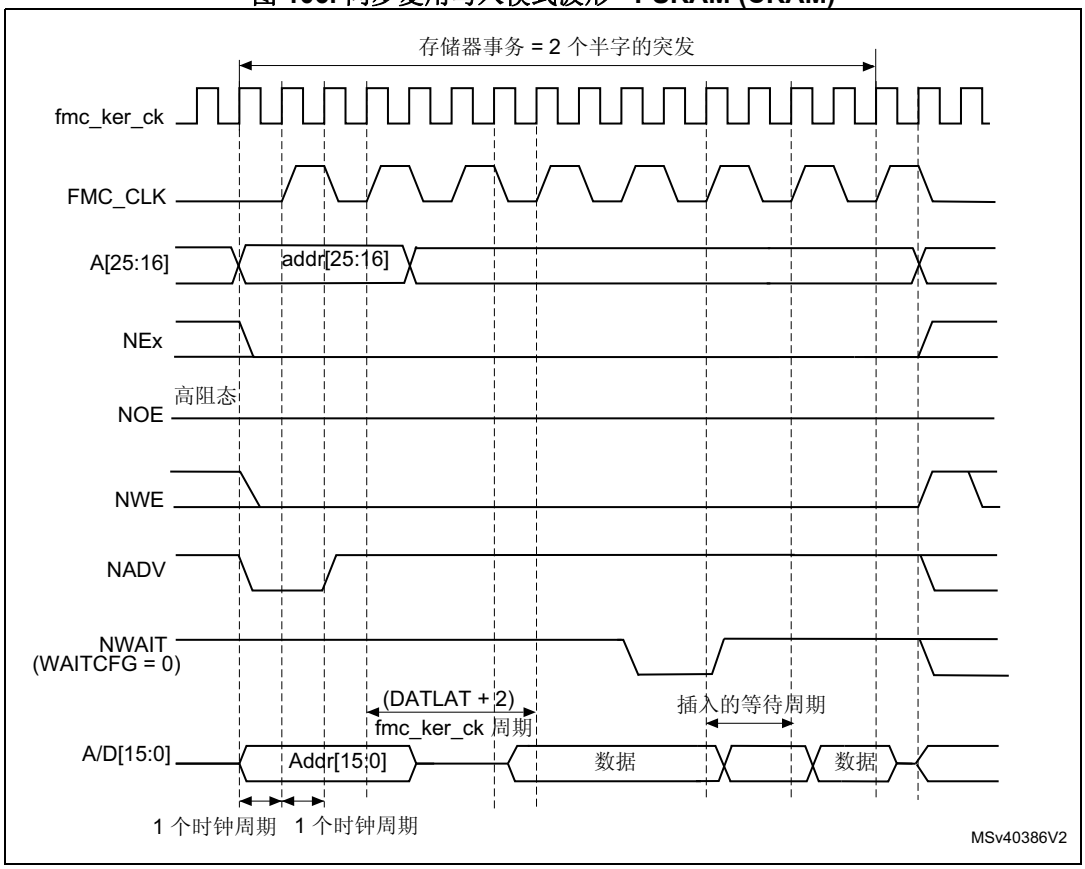
表 171. FMC_BTRx 位域

位号	位名	要设置的值
31:30	保留	0x0
29:28	ACCMOD	0x0
27-24	DATLAT	数据延迟
27-24	DATLAT	数据延迟
23-20	CLKDIV	0x0，使 CLK = fmc_ker_ck（不支持） 0x1，使 CLK = 2 × fmc_ker_ck ..
19-16	BUSTURN	NEx 变为高电平到 NEx 变为低电平之间的时间 (BUSTURN fmc_ker_ck)
15-8	DATAST	无关

表 171. FMC_BTRx 位域（续）

位号	位名	要设置的值
7-4	ADDHLD	无关
3-0	ADDSET	无关

图 106. 同步复用写入模式波形 - PSRAM (CRAM)



1. 存储器必须提前一个周期发出 NWAIT 信号，相应地 WAITCFG 必须编程为 0。
2. 字节通道 (NBL) 输出未显示，当 NEx 有效时它们保持低电平。

表 172. FMC_BCRx 位域

位号	位名	要设置的值
31	FMCEN	0x0
30-26	保留	0x000
25-24	BMAP	根据需要进行设置
23-22	保留	0x000
21	WFDIS	根据需要进行设置
20	CCLKEN	根据需要进行设置
19	CBURSTRW	对同步读取没有影响

表 172. FMC_BCRx 位域（续）

位号	位名	要设置的值
18:16	CPSIZE	根据需要进行设置（使用 CRAM 1.5 时为 0x1）
15	ASYNCWAIT	0x0
14	EXTMOD	0x0
13	WAITEN	在存储器支持该特性的情况下置位为 1，否则保持为 0。
12	WREN	0x1
11	WAITCFG	0x0
10	保留	0x0
9	WAITPOL	是否置位视存储器情况而定
8	BURSTEN	对同步写入没有影响
7	保留	0x1
6	FACCEN	根据存储器支持情况进行设置
5-4	MWID	根据需要进行设置
3-2	MTYP	0x1
1	MUXEN	根据需要进行设置
0	MBKEN	0x1

表 173. FMC_BTRx 位域

位号	位名	要设置的值
31-30	保留	0x0
29:28	ACCMOD	0x0
27-24	DATLAT	数据延迟
23-20	CLKDIV	0x0，使 CLK = fmc_ker_ck（不支持） 0x1，使 CLK = 2 × fmc_ker_ck
19-16	BUSTURN	NEx 变为高电平到 NEx 变为低电平之间的时间 (BUSTURN fmc_ker_ck)
15-8	DATAST	无关
7-4	ADDHLD	无关
3-0	ADDSET	无关

22.7.6 NOR/PSRAM 控制寄存器

SRAM/NOR-Flash 片选控制寄存器 1..4 (FMC_BCR1..4)

SRAM/NOR-Flash chip-select control registers 1..4

偏移地址：8 * (x – 1)，x = 1...4

复位值：对于存储区域 1 为 0x0000 30DB，对于存储区域 2 到 4 为 0x0000 30D2

该寄存器包含每个存储区域的控制信息，用于 SRAM、PSRAM 和 NOR Flash。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FMCEN	Res.	Res.	Res.	Res.	Res.	BMAP[1:0]		Res.	Res.	WFDIS	CCLKEN	CBURSTRW	CPSIZE[2:0]			ASYNCWAIT	EXTMOD	WAITEN	WREN	WAITCFG	Res.	WAITPOL	BURSTEN	Res.	FACCEN	MWID		MTYP		MUXEN	MBKEN
rw						rw				rw	rw	rw	rw			rw	rw	rw	rw	rw		rw	rw		rw	rw	rw	rw	rw	rw	rw

位 31 **FMCEN**：FMC 控制器使能 (FMC controller Enable)

该位用于使能/禁止 FMC 控制器。

0：禁止 FMC 控制器

1：使能 FMC 控制器

注： FMC_BCR2..4 寄存器的 FMCEN 位为“无关”位。只能通过 FMC_BCR1 寄存器使能。

位 30:26 保留，必须保持复位值

位 25:24 **BMAP[1:0]**：FMC 存储区域映射 (FMC bank mapping)

这些位用于重映射 SDRAM 存储区域 2 或者交换 FMC NOR/PSRAM 存储区域和 SDRAM 存储区域。请参见表 10。

01：默认映射（请参见图 2 和表 10）。

01：交换 NOR/PSRAM 存储区域和 SDRAM 存储区域 1/存储区域 2。

10：SDRAM 存储区域 2 重映射在 FMC 存储区域 2 上，并且仍然可在默认映射下访问。

11：保留。

注： FMC_BCR2..4 寄存器的 BMAP 位为“无关”位。只能通过 FMC_BCR1 寄存器使能。

位 23:22 保留，必须保持复位值

位 21 **WFDIS**：写 FIFO 禁止 (Write FIFO Disable)

此位可禁止 FMC 控制器使用的写 FIFO。

0：使能写 FIFO（复位后的默认值）

1：禁止写 FIFO

注： FMC_BCR2..4 寄存器的 WFDIS 位为“无关”位。只能通过 FMC_BCR1 寄存器使能。

位 20 **CCLKEN**：连续时钟使能 (Continuous Clock Enable)。

该位可使能向外部存储器器件输出 FMC_CLK 时钟。

0：仅在同步存储器访问（读/写事务）期间生成 FMC_CLK。FMC_CLK 时钟分频比由 FMC_BCRx 寄存器中配置的 CLKDIV 值指定（复位后的默认值）。

1：异步和同步访问期间连续生成 FMC_CLK。CCLKEN 置 1 将激活 FMC_CLK 时钟。

注： *FMC_BCR2..4 寄存器的 CCLKEN 位为“无关”位。只能通过 FMC_BCR1 寄存器使能。存储区域 1 必须配置为支持同步模式才能生成 FMC_CLK 连续时钟。*

如果 CCLKEN 位置 1，则 FMC_CLK 时钟分频比将由 FMC_BTR1 寄存器的 CLKDIV 值指定。FMC_BWTR1 寄存器中的 CLKDIV 为无关位。

如果采用同步模式且 CCLKEN 位置 1，则与存储区域 1 之外的其它存储区域相连的同步存储器将共用同一个时钟源（FMC_BTR2..4 寄存器和 FMC_BWTR2..4 寄存器中的 CLKDIV 值对其它存储区域不起作用。）

位 19 **CBURSTRW**：突发写使能 (Write burst enable)。

PSRAM (CRAM) 在突发模式下工作时，该位可使能同步写访问。同步读取访问的使能位为 FMC_BCRx 寄存器中的 BURSTEN 位。

0：始终在异步模式下写入

1：在同步模式下写入

位 18:16 **CPSIZE[2:0]**：CRAM 页大小 (CRAM page size)。

这些位用于 Cellular RAM 1.5，Cellular RAM 1.5 不允许突发访问越过页面之间的地址边界。如果配置这些位，FMC 控制器会在达到存储器页大小后自动分离突发访问（请参见存储器数据手册了解页大小）。

000：越过页边界后不进行突发分离（复位后的默认值）

001：128 字节

010：256 字节

100：1024 字节

其他配置：保留

位 15 **ASYNCWAIT**：异步传输期间的等待信号 (Wait signal during asynchronous transfers)

该位可使能/禁止 FMC 使用等待信号，即使在异步协议期间也有效。

0：运行异步协议时不考虑 NWAIT 信号（复位后的默认值）

1：运行异步协议时考虑 NWAIT 信号

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

FMC 可对 FMC_BWTR 寄存器中异步访问的写入时序进行配置，此配置由 EXTMOD 位使能，进而使读取和写入操作采用不同时序。

0：不考虑 FMC_BWTR 寄存器中的值（复位后的默认值）

1：考虑 FMC_BWTR 寄存器中的值

注： *如果禁用扩展模式，FMC 可以在模式 1 或模式 2 下运行，如下所述：*

- *当选择 SRAM/PSRAM 存储器类型时，模式 1 为默认模式（MTYP = 0x0 或 0x01）*
- *当选择 NOR 存储器类型时，模式 2 为默认模式（MTYP = 0x10）*

位 13 **WAITEN**：等待使能位 (Wait enable bit)。

该位可使能/禁止在同步模式下访问存储器时通过 NWAIT 信号插入等待周期。

0：禁止 NWAIT 信号（不考虑其电平，不在配置过的 Flash 延迟周期后插入等待周期）

1：使能 NWAIT 信号（考虑其电平，如果使能，在配置过的延迟周期后插入等待周期）（复位后的默认值）

位 12 **WREN**：写入使能位 (Write enable bit)。

该位指示 FMC 是否使能/禁止在存储区域内写入：

0：FMC 禁止在存储区域内写入，如果进行写操作将报告 AXI 从错误，

1：FMC 使能在存储区域内写入（复位后的默认值）。

位 11 **WAITCFG**：等待时序配置 (Wait timing configuration)。

NWAIT 信号指示存储器中的数据是否有效，或者在同步模式下访问存储器时是否必须插入等待周期。该配置位决定存储器是在等待周期之前的一个时钟周期还是等待周期期间使能 NWAIT：

0：NWAIT 信号在等待周期之前的一个数据周期有效（复位后的默认值），

1：NWAIT 信号在等待周期期间有效（不适用于 PSRAM）。

位 10 保留，必须保持复位值

位 9 **WAITPOL**：等待信号极性位 (Wait signal polarity bit)。

该位定义同步或异步模式下使用的存储器的等待信号极性：

0：NWAIT 低电平有效（复位后的默认值），

1：NWAIT 高电平有效。

位 8 **BURSTEN**：突发使能位 (Burst enable bit)。

该位可使能/禁止同步读取访问。该位仅对突发模式下工作的同步存储器有效：

0：禁止突发模式（复位后的默认值）。在异步模式下进行读取访问。

1：使能突发模式。在同步模式下进行读取访问。

位 7 保留，必须保持复位值

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

该位使能 NOR Flash 访问操作。

0：禁止相应的 NOR Flash 访问

1：使能相应的 NOR Flash 访问（复位后的默认值）

位 5:4 **MWID[1:0]**：存储器数据总线宽度 (Memory data bus width)

定义外部存储器器件宽度，对所有类型的存储器均有效。

00：8 位

01：16 位（复位后的默认值）

10：32 位

11：保留

位 3:2 **MTYP[1:0]**：存储器类型 (Memory type)

这些位定义与相应存储区域相连的外部存储器类型：

00：SRAM（对于存储区域 2...4，复位后的默认值）

01：PSRAM (CRAM)

10：NOR Flash/OneNAND Flash（对于存储区域 1，复位后的默认值）

11：保留

位 1 **MUXEN**：地址/数据复用使能位 (Address/data multiplexing enable bit)。

该位置 1 时，地址和数据值在数据总线上复用，仅对 NOR 和 PSRAM 存储器有效：

0：地址/数据非复用

1：地址/数据在数据总线上复用（复位后的默认值）

位 0 **MBKEN**：存储区域使能位 (Memory bank enable bit)。

该位使能存储区域。复位后使能存储区域 1，其它存储区域均禁止。访问禁止的存储区域会引起 AXI 总线上的错误。

0：禁止相应的存储区域

1：使能相应的存储区域

SRAM/NOR-Flash 片选时序寄存器 1..4 (FMC_BTR1..4)

SRAM/NOR-Flash chip-select timing registers 1..4

偏移地址：0x04 + 8 * (x – 1)，x = 1..4

复位值：0x0FFF FFFF

该寄存器包含每个存储区域的控制信息，用于 SRAM、PSRAM 和 NOR Flash。如果 FMC_BCRx 寄存器中的 EXTMOD 位置 1，该寄存器将和另外一个寄存器配合来配置写入和读取访问，也就是说有 2 个寄存器可用：一个用于配置读取访问（此寄存器），另一个用于配置写入访问（FMC_BWTRx 寄存器）。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	ACCMOD		DATLAT				CLKDIV				BUSTURN			
		rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATAST								ADDHLD				ADDSET			
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:30 保留，必须保持复位值

位 29:28 **ACCMOD[1:0]**：访问模式 (Access mode)

这些位指定异步访问模式，如时序图所示。仅当 FMC_BCRx 寄存器中的 EXTMOD 位置 1 时，这些位才有效。

00：访问模式 A

01：访问模式 B

10：访问模式 C

11：访问模式 D

位 27:24 **DATLAT[3:0]**：同步存储器的数据延迟 (Data latency for synchronous memory)（请参见位说明下的注释）

对于使能了读/写突发模式（BURSTEN/CBURSTRW 位置 1）的同步访问，这些位定义读写首个数据前要发送给存储器的存储器时钟周期数 (+2)：

该时序参数以 FMC_CLK 周期而非 fmc_ker_ck 周期表示。

该值与异步访问模式无关。

0000：首次突发访问时，2 个 FMC_CLK 时钟周期的数据延迟

1111：首次突发访问时，17 个 FMC_CLK 时钟周期的数据延迟（复位后的默认值）

位 23:20 **CLKDIV[3:0]**：FMC_CLK 信号的时钟分频比 (Clock divide ratio (for FMC_CLK signal))

这些位定义 FMC_CLK 时钟输出信号的周期，以 fmc_ker_ck 周期数表示：

0000：保留

0001：FMC_CLK 周期 = 2 × fmc_ker_ck 周期

0010：FMC_CLK 周期 = 3 × fmc_ker_ck 周期

1111：FMC_CLK 周期 = 16 × fmc_ker_ck 周期（复位后的默认值）

在异步 NOR Flash、SRAM 或 PSRAM 访问模式下，该值为无关值。

注：　有关 FMC_CLK 分频比公式的信息，请参见第 22.7.5 节：同步事务

位 19:16 **BUSTURN**：总线周转阶段的持续时间 (Bus turnaround phase duration)

通过软件写入这些位可在写入-读取（和读取-写入）事务的结尾添加延迟。该延迟可以匹配连续事务之间的最短时间（tEHEL 由 NEx 高电平变为 NEx 低电平）以及存储器在读取访问后释放数据总线所需的最长时间 (tEHQZ)。编程的总线周转延迟插入到对静态存储区域的异步读取（复用模式或 D 模式）/写入事务与任何其他异步/同步读取/写入事务之间。对于读取事务，存储区域可以相同也可以不同；对于写入事务，存储区域可以不同（复用模式和 D 模式除外）。

在某些情况下，无论编程的 BUSTRUN 值如何，总线周转延迟固定如下：

- 除了复用模式和 D 模式之外，总线周转延迟不会插入到对同一静态存储区域的两个连续异步写入传输之间。
- 下列情况下的总线周转延迟为 1 个 FMC 时钟周期：
  - 对同一静态存储区域的两次连续的异步读取操作之间（复用模式和 D 模式除外）。
  - 对任何静态存储区域/动态存储区域的异步读取操作与异步/同步写入操作之间（复用模式和 D 模式除外）。
  - 异步（模式 1、2、A、B 或 C）读取操作与对另一个静态存储区域的读取操作之间。
- 下列情况下的总线周转延迟为 2 个 FMC 时钟周期：
  - 对同一存储区域的两次连续的同步写入操作之间（在突发或单次模式下）。
  - 对静态存储区域的同步写入（突发或单次）访问与异步写入 / 读取传输之间（在读取操作的情况下，存储区域可以相同也可以不同）。
  - 对另一个静态存储区域的任何同步 / 异步读取或写入操作之后的两次连续的同步读取操作之间（在突发或单次模式下）。
- 下列情况下的总线周转延迟为 3 个 FMC 时钟周期：
  - 对不同静态存储区域的两次连续的同步写入操作之间（在突发或单次模式下）。
  - 对同一或不同存储区域的同步写入访问（在突发或单次模式下）与同步读取访问之间。

0000：BUSTURN 阶段的持续时间 = 增加 0 个 fmc_ker_ck 时钟周期
...
1111：BUSTURN 阶段的持续时间 = 增加 15 个 fmc_ker_ck 时钟周期（复位后的默认值）

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

通过软件写入这些位可定义数据阶段的持续时间（请参见*图 89* 到*图 101*），适用于异步访问模式：
0000 0000：保留
0000 0001：DATAST 阶段的持续时间 = 1 × fmc_ker_ck 时钟周期
0000 0010：DATAST 阶段的持续时间 = 2 × fmc_ker_ck 时钟周期
...
1111 1111：DATAST 阶段的持续时间 = 255 × fmc_ker_ck 时钟周期（复位后的默认值）
有关每种存储器类型和访问模式数据阶段持续时间的信息，请参见相应图片（*图 89* 到*图 101*）。
例如：在模式 1、写入访问以及 DATAST=1 条件下，数据阶段的持续时间 = DATAST+1 = 1 个 fmc_ker_ck 时钟周期。

注：　在同步访问模式下，该值为无关值。

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

通过软件写入这些位可定义*地址保持*阶段的持续时间（请参见*图 89* 到*图 101*），适用于模式 D 或复用访问：

0000：保留

0001：ADDHLD 阶段的持续时间 = 1 × fmc_ker_ck 时钟周期

0010：ADDHLD 阶段的持续时间 = 2 × fmc_ker_ck 时钟周期

...

1111：ADDHLD 阶段的持续时间 = 15 × fmc_ker_ck 时钟周期（复位后的默认值）

有关每种访问模式地址保持阶段持续时间的信息，请参见相应图片（*图 89* 到*图 101*）。

注：　在同步访问模式下，该值不使用，因为地址保持阶段的持续时间始终是 1 个存储器时钟周期。

位 3:0 **ADDSET**：地址设置阶段的持续时间 (Address setup phase duration)

通过软件写入这些位可定义*地址设置*阶段的持续时间（请参见*图 89* 到*图 101*），适用于 SRAM、ROM 和异步 NOR Flash 访问模式：

0000：ADDSET 阶段的持续时间 = 0 × fmc_ker_ck 时钟周期

...

1111：ADDSET 阶段的持续时间 = 15 × fmc_ker_ck 时钟周期（复位后的默认值）

有关每种访问模式地址设置阶段持续时间的信息，请参见相应图片（请参见*图 89* 到*图 101*）。

注：　在同步访问模式下，该值为无关值。

复用模式或模式 D 下，ADDSET 的最小值为 1。

注：　PSRAM (CRAM) 由于内部刷新而导致数据延时时间长度不确定。因此，这些存储器会在整个延迟阶段发送 NWAIT 信号，以便按照需要延长延迟。

对于 PSRAM (CRAM)，字段 DATLAT 必须设置为 0，这样 FMC 会立即退出延迟阶段，开始对存储器中的 NWAIT 采样，然后在存储器准备就绪后开始读取或写入。

此方法也适用于最新一代的同步 Flash，同早期 Flash 不同的是，此类 Flash 会发送 NWAIT 信号（检查所用的具体 Flash 数据表）。

SRAM/NOR-Flash 写入时序寄存器 1..4 (FMC_BWTR1..4)

SRAM/NOR-Flash write timing registers 1..4

偏移地址：0x104 + 8 * (x – 1)，x = 1...4

复位值：0x0FFF FFFF

此寄存器包含每个存储区域的控制信息，用于 SRAM、PSRAM 和 NOR Flash。当 FMC_BCRx 寄存器中的 EXTMOD 位置 1 时，该寄存器将处于有效状态，可以进行写入访问。

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

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

位 31:30 保留，必须保持复位值

位 29:28 **ACCMOD**：访问模式 (Access mode)。

这些位指定异步访问模式，如下一个时序图所示。仅当 FMC_BCRx 寄存器中的 EXTMOD 位置 1 时，这些位才有效。

00：访问模式 A

01：访问模式 B

10：访问模式 C

11：访问模式 D

位 27:20 保留，必须保持复位值

位 19:16 **BUSTURN**：总线周转阶段的持续时间 (Bus turnaround phase duration)

通过软件写入这些位可在写入事务结束时增加一个延迟，进而匹配两个连续事务之间的最小时间（t_{EHEL} 由 ENx 高电平变为 ENx 低电平）：

(BUSTRUN + 1) fmc_ker_ck 周期 ≥ $t_{EHELmin}$。

编程的总线周转延迟插入到对静态存储区域的异步写入传输与任何其他异步/同步读取或写入传输之间。对于读取事务，存储区域可以相同也可以不同；对于写入事务，存储区域可以不同（复用模式和 D 模式除外）。

在某些情况下，无论编程的 BUSTRUN 值如何，总线周转延迟固定如下：

- 除了复用模式和 D 模式之外，总线周转延迟不会插入到对同一静态存储区域的两个连续异步写入传输之间。
- 下列情况下的总线周转延迟为 2 个 FMC 时钟周期：
  - 对同一存储区域的两次连续的同步写入操作之间（在突发或单次模式下）
  - 对静态存储区域的同步写入传输（在突发或单次模式下）与异步写入/读取传输之间。
- 下列情况下的总线周转延迟为 3 个 FMC 时钟周期：
  - 对不同静态存储区域的两次连续的同步写入操作之间（在突发或单次模式下）。
  - 对同一或不同存储区域的同步写入传输（在突发或单次模式下）与同步读取传输之间。

0000：BUSTURN 阶段的持续时间 = 增加 0 个 fmc_ker_ck 时钟周期

...

1111：BUSTURN 阶段的持续时间 = 增加 15 个 fmc_ker_ck 时钟周期（复位后的默认值）

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

通过软件写入这些位可定义数据阶段的持续时间（请参见*图 89* 到*图 101*），适用于异步 SRAM、PSRAM 和 NOR Flash 访问模式：

0000 0000：保留

0000 0001：DATAST 阶段的持续时间 = 1 × fmc_ker_ck 时钟周期

0000 0010：DATAST 阶段的持续时间 = 2 × fmc_ker_ck 时钟周期

...

1111 1111：DATAST 阶段的持续时间 = 255 × fmc_ker_ck 时钟周期（复位后的默认值）

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

通过软件写入这些位可定义*地址保持*阶段的持续时间（请参见*图 89* 到*图 101*），适用于异步复用访问：

0000：保留

0001：ADDHLD 阶段的持续时间 = 1 × fmc_ker_ck 时钟周期

0010：ADDHLD 阶段的持续时间 = 2 × fmc_ker_ck 时钟周期

...

1111：ADDHLD 阶段的持续时间 = 15 × fmc_ker_ck 时钟周期（复位后的默认值）

注：　在同步 NOR Flash 访问模式下，该值不使用，因为地址保持阶段的持续时间始终是 1 个 Flash 时钟周期。

位 3:0 **ADDSET**：地址建立阶段的持续时间 (Address setup phase duration)。

通过软件写入这些位可定义以 fmc_ker_ck 周期表示的*地址建立*阶段持续时间（请参见*图 89* 到*图 101*），适用于异步访问模式：

0000：ADDSET 阶段的持续时间 = 0 × fmc_ker_ck 时钟周期

...

1111：ADDSET 阶段的持续时间 = 15 × fmc_ker_ck 时钟周期（复位后的默认值）

注：　在同步访问模式下，该值不使用，因为地址建立阶段的持续时间始终是 1 个 Flash 时钟周期。复用模式下，ADDSET 的最小值为 1。

22.8 NAND Flash 控制器

FMC 会生成适当的信号时序，以驱动 8 位和 16 位 NAND Flash。

NAND 存储区域通过专用的寄存器配置（*第 22.8.7 节*）。可编程的存储器参数包括访问时序（如*表 174* 所示）和 ECC 配置。

表 174. 可编程的 NAND Flash 访问参数

参数	功能	访问模式	单位	最小值	最大值
存储器建立时间	命令使能前地址建立所需时钟周期 (fmc_ker_ck) 数	读/写	AHB 时钟周期 (fmc_ker_ck)	1	255
存储器等待	命令使能的最小持续时间（按 fmc_ker_ck 时钟周期计）	读/写	AHB 时钟周期 (fmc_ker_ck)	2	255
存储器保持	命令禁止后，必须保持地址（如果进行了写访问还需保持数据）的时钟周期 (fmc_ker_ck) 数	读/写	AHB 时钟周期 (fmc_ker_ck)	1	254
存储器数据总线高阻态	开始进行写访问后，数据总线保持高阻状态期间的时钟周期 (fmc_ker_ck) 数	写	AHB 时钟周期 (fmc_ker_ck)	0	254

22.8.1 外部存储器接口信号

下表列出了通常用于连接 NAND Flash 的信号。

注： *前缀“N”标识低电平有效的信号。*

8 位 NAND Flash

表 175. 8 位 NAND Flash

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

由于 FMC 能够管理足够多的地址周期，因此理论上不存在容量限制。

16 位 NAND Flash

表 176. 16 位 NAND Flash

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

注： *由于 FMC 能够管理足够多的地址周期，因此理论上不存在容量限制。*

22.8.2 NAND Flash 支持的存储器和事务

表 177 介绍了所支持的设备、访问模式和事务。NAND Flash 控制器不允许（或不支持）的事务以灰色显示。

表 177. 支持的存储器和事务

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

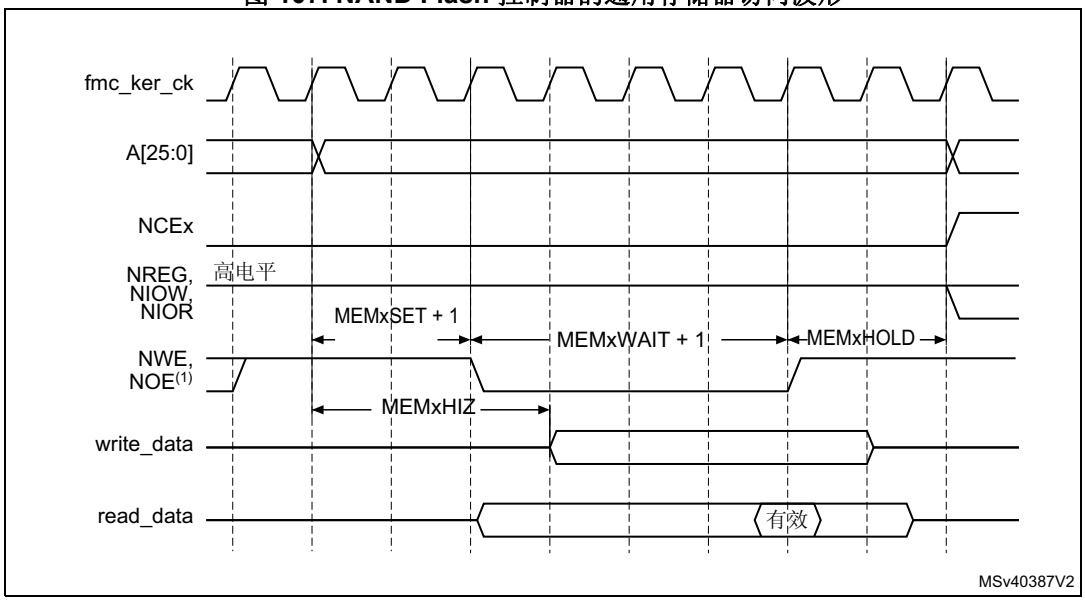
22.8.3 NAND Flash 的时序图

NAND Flash 存储区域通过以下一组寄存器进行管理：

- 控制寄存器：FMC_PCR
- 中断状态寄存器：FMC_SR
- ECC 寄存器：FMC_ECCR
- 通用存储器空间的时序寄存器：FMC_PMEM
- 特性存储器空间的时序寄存器：FMC_PATT

每个时序配置寄存器包含四个参数，其中三个参数用于定义 NAND Flash 的三个访问阶段的 fmc_ker_ck 周期数，另一个参数用于定义进行写访问时开始驱动数据总线的时序。*图 107* 介绍了通用存储器访问的时序参数定义，特性存储器空间的访问时序与此类似。

图 107. NAND Flash 控制器的通用存储器访问波形



1. 进行写访问期间 NOE 保持高电平（无效）。进行读访问期间 NWE 保持高电平（无效）。
2. 对于写访问，保持阶段延迟为 (MEMHOLD) 个 fmc_ker_ck 周期，而对于读访问则为 (MEMHOLD + 1) 个 fmc_ker_ck 周期。

22.8.4 NAND Flash 操作

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

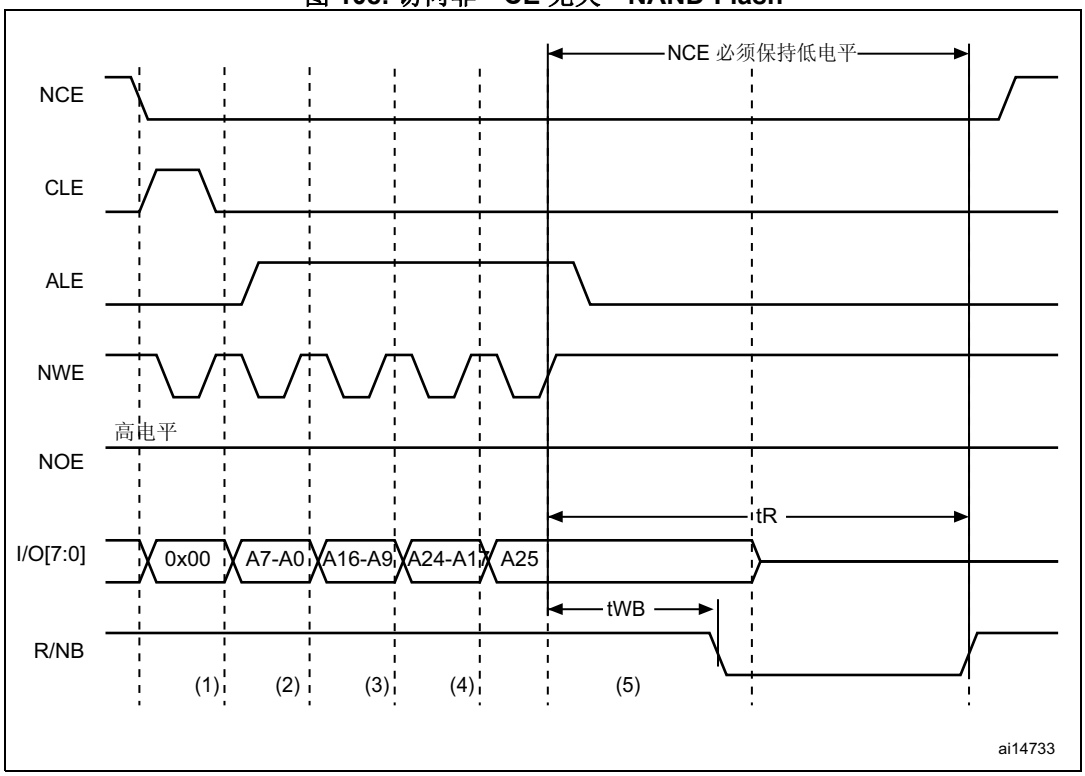
从 NAND Flash 设备进行的典型页读取操作需要执行以下步骤：

1. 根据 NAND Flash 的特性（PWID 位指示 NAND Flash 数据总线宽度，根据需要 PWAITEN = 0 或 1；有关时序配置，请参见*第 22.6.2 节：NAND Flash 地址映射*），配置 FMC_PCR 和 FMC_PMEM（对于某些设备，配置 FMC_PATT，请参见*第 22.8.5 节：NAND Flash 预等待功能*）寄存器，进而编程和使能相应的存储区域。
2. CPU 向通用存储器空间执行字节写操作，此时数据字节等于一个 Flash 命令字节（例如 Samsung NAND Flash 设备的字节为 0x00）。在写入选通期间（NWE 上为低电平脉冲），NAND Flash 的 LE 输入有效，因此将已写入的字节视为 NAND Flash 的一个命令。该命令由存储器设备锁存后，随后进行页读取操作时，不需要再次写入该命令。
3. 通过在通用存储器空间或特性空间写入 STARTAD[7:0]、STARTAD[16:9]、STARTAD[24:17] 以及 STARTAD[25]（针对 64 Mb x 8 位的 NAND Flash）这 4 个字节（对于容量较小的设备写入 3 个字节），CPU 能够发送读操作的起始地址 (STARTAD)。在写入选通期间（NWE 上为低电平脉冲），NAND Flash 设备的 ALE 输入有效，因此将已写入的字节视为读操作的起始地址。借助特性存储器空间，可使用 FMC 的另一种不同时序配置，实现某些 NAND Flash 所需的预等待功能（有关详细信息 FMC，请参见*第 22.8.5 节：NAND Flash 预等待功能*）。
4. 控制器在对同一个或另一个存储区域进行新访问之前，等待 NAND Flash 准备好（R/NB 信号处于高电平）。等待期间，控制器保持 NCE 信号有效（低电平）。
5. 然后 CPU 能够从通用存储器空间执行字节读操作，进而按字节读取 NAND Flash 页（数据字段 + 备用字段）。
6. 可读取下一个 NAND Flash 页，而无需任何 CPU 命令或地址写操作。可采用以下三种不同的方式实现：
   - 执行步骤 5 中介绍的操作
   - 通过重新开始步骤 3 中的操作随机访问一个新地址
   - 通过重新开始步骤 2 向 NAND Flash 设备发送新命令

22.8.5 NAND Flash 预等待功能

一些 NAND Flash 设备需要在写入地址的最后一部分后，控制器等待 R/NB 信号变为低电平。（见*图 108*）。

图 108. 访问非“CE 无关”NAND-Flash



1. CPU 在地址 0x7001 0000 处写入字节 0x00。
2. CPU 在地址 0x7002 0000 处写入字节 A7~A0。
3. CPU 在地址 0x7002 0000 处写入字节 A16~A9。
4. CPU 在地址 0x7002 0000 处写入字节 A24~A17。
5. CPU 在地址 0x8802 0000 处写入字节 A25：FMC 通过 FMC_PATT2 时序定义执行写访问，其中 ATTHOLD ≥ 7（假设 (7+1) × fmc_ker_ck = 112 ns > t_{WB} 最大值）。这可确保 NCE 保持低电平，直到 R/NB 再次变为低电平后变为高电平（只有 NCE 信号对之有作用的 NAND Flash 才需要此功能）。

当需要此功能时，可通过编程 MEMHOLD 值确保满足 t_{WB} 时序。然而 CPU 对 NAND Flash 进行的所有读取访问均会经过 (MEMHOLD + 1) 个 fmc_ker_ck 周期的保持延迟，而所有写入访问均会经过 (MEMHOLD) 个 fmc_ker_ck 周期延迟（该延迟插入在 NWE 信号的上升沿与下一访问之间）。

要克服该时序限制，可使用特性存储器空间，将其时序寄存器编程为满足 t_{WB} 时序的 ATTHOLD 值并将 MEMHOLD 保持为其最小值。然后，CPU 必须使用通用存储器空间进行所有的 NAND Flash 读和写访问，向 NAND Flash 设备写入最后一个地址字节时除外，此时 CPU 必须对特性存储器空间执行写操作。

22.8.6 纠错码 (ECC) 计算 (NAND Flash)

FMC 控制器包括纠错码计算硬件模块。该模块可在软件处理 ECC 时减少主机 CPU 工作负载。ECC 模块与 NAND 存储区域相关联。

对 NAND Flash 执行读取或写入操作时，相应每 256、512、1 024、2 048、4 096 或 8 192 个字节，FMC 中使用的 ECC 算法可修正 1 位错误并且检测出 2 位错误。该操作基于 BCH8 编码算法，并且包括计算行和列奇偶校验。

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

ECC 按如下说明操作：

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

主机 CPU 对 NAND Flash 完成所需字节数的读取/写入操作后，必须读取 FMC_ECCR 寄存器，才能检索计算出的值。读取后，应通过将 ECCEN 位复位为零来将这些寄存器清零。要计算新的数据块，必须将 FMC_PCR 寄存器中的 ECCEN 位置 1。

按如下序列执行 ECC 计算：

1. 使能 FMC_PCR 寄存器中的 ECCEN 位。
2. 将数据写入 NAND Flash 页。在写入 NAND 页期间，ECC 模块将计算 ECC 值。
3. 等待 ECC 代码就绪（FIFO 为空）。
4. 读取 FMC_ECCR 寄存器中所提供的 ECC 值，并将其存储到变量。
5. 将 FMC_PCR 寄存器中的 ECCEN 位清零后使能，然后从 NAND 页回读写入的数据。在读取 NAND 页期间，ECC 模块将计算 ECC 值。
6. 读取 FMC_ECCR 寄存器中所提供的新 ECC 值。
7. 如果两次读取的 ECC 值相同，则无需校正，否则说明存在 ECC 错误，并且软件校正例程将返回有关该错误是否能够得到校正的信息。

22.8.7 NAND Flash 控制器寄存器

NAND Flash 控制寄存器 (FMC_PCR)

NAND Flash control registers

偏移地址：0x80

复位值：0x0000 0018

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ECCPS			TAR				TCLR				Res.	Res.	ECCEN	PWID		Res.	PBKEN	PWAITEN	Res.
												rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw			rw	rw	rw		rw	rw	

位 31:20 保留，必须保持复位值

位 19:17 **ECCPS**：ECC 页大小 (ECC page size)。

这些位定义扩展 ECC 的页大小：

000：256 字节

001：512 字节

010：1024 字节

011：2048 字节

100：4096 字节

101：8192 字节

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

这些位以 fmc_ker_ck 时钟周期数设置从 ALE 低电平到 RE 低电平的时间。

时间是 t_ar = (TAR + SET + 2) × $t_{fmc_ker_ck}$，其中 $t_{fmc_ker_ck}$ 是 FMC 时钟周期

0000：1 个 fmc_ker_ck 周期（默认）

1111：16 个 fmc_ker_ck 周期

注： 根据寻址空间，SET 为 MEMSET 或 ATTSET。

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

这些位以 fmc_ker_ck 时钟周期数设置从 CLE 低电平到 RE 低电平的时间。该时间可根据以下公式得出：

t_clr = (TCLR + SET + 2) × $t_{fmc_ker_ck}$，其中 $t_{fmc_ker_ck}$ 是 fmc_ker_ck 时钟周期

0000：1 个 fmc_ker_ck 周期（默认）

1111：16 个 fmc_ker_ck 周期

注： 根据寻址空间，SET 为 MEMSET 或 ATTSET。

位 8:7 保留，必须保持复位值

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

0：禁止和复位 ECC 逻辑（复位后为默认值），

1：使能 ECC 逻辑。

位 5:4 **PWID**：数据总线宽度 (Data bus width)。

这些位定义外部存储器件宽度。

00：8 位

01：16 位（复位后的默认值）。

10：保留。

11：保留。

位 3 保留，必须保持复位值

位 2 **PBKEN**：NAND Flash 存储区域使能位 (NAND Flash memory bank enable bit)。
该位使能存储区域。访问禁止的存储区域会引起 AXI 总线上的错误
0：禁止相应的存储区域（复位后为默认值）
1：使能相应的存储区域

位 1 **PWAITEN**：等待特性使能位 (Wait feature enable bit)。
该位使能 NAND Flash 存储区域的等待特性：
0：禁用
1：使能

位 0 保留，必须保持复位值

FIFO 状态和中断寄存器 (FMC_SR)

FIFO status and interrupt register

偏移地址：0x84

复位值：0x0000 0040

该寄存器包含有关 FIFO 状态和中断的信息。FMC 具有一个 FIFO，当向存储器执行写入操作以传输多达 16 字的数据时，使用该 FIFO。

当 FMC 将其 FIFO 的内容移入存储器时，此寄存器用于快速写入 FIFO，然后释放 AXI 总线供 FMC 以外的外设的事务使用。这些寄存器位中有一位用来指示 FIFO 的状态，供 ECC 使用。

在将数据写入存储器时计算 ECC。因此，为了读取正确的 ECC，软件必须等到 FIFO 为空。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FEMPT	IFEN	ILEN	IREN	IFS	ILS	IRS
									r	rw	rw	rw	rw	rw	rw

位 31:7 保留，必须保持复位值

位 6 **FEMPT**：FIFO 为空 (FIFO empty)。
该位是提供 FIFO 状态的只读位
0：FIFO 非空
1：FIFO 为空

位 5 **IFEN**：中断下降沿检测使能位 (Interrupt falling edge detection enable bit)
0：中断下降沿检测请求禁止
1：中断下降沿检测请求使能

位 4 **ILEN**：中断高电平检测使能位 (Interrupt high-level detection enable bit)
0：中断高电平检测请求禁止
1：中断高电平检测请求使能

位 3 **IREN**：中断上升沿检测使能位 (Interrupt rising edge detection enable bit)
0：中断上升沿检测请求禁止
1：中断上升沿检测请求使能

位 2 **IFS**：中断下降沿状态 (Interrupt falling edge status)
此标志由硬件置 1，由软件复位。
0：未出现中断下降沿
1：出现中断下降沿
注： 如果该位由软件写入 1，则将被置 1。

位 1 **ILS**：中断高电平状态 (Interrupt high-level status)
此标志由硬件置 1，由软件复位。
0：未出现中断高电平
1：出现中断高电平

位 0 **IRS**：中断上升沿状态 (Interrupt rising edge status)
此标志由硬件置 1，由软件复位。
0：未出现中断上升沿
1：出现中断上升沿
注： 如果该位由软件写入 1，则将被置 1。

通用存储器空间时序寄存器 2..4 (FMC_PMEM)

Common memory space timing register 2..4

偏移地址：地址：0x88

复位值：0xFCFC FCFC

FMC_PMEM 读/写寄存器包含 NAND Flash 存储区域的时序信息。该信息用于访问 NAND Flash 的通用存储空间来实现命令、地址写访问和数据读/写访问。

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

位 31:24 **MEMHIZ**：通用存储器 x 数据总线高阻态时间 (Common memory x databus Hi-Z time)
这些位定义在对 NAND Flash 的通用存储空间开始执行写访问之后，数据总线保持高阻态所持续的 fmc_ker_ck 时钟周期数。仅对写入事务有效：
0000 0000：0 个 fmc_ker_ck 周期
1111 1110：254 个 fmc_ker_ck 周期
1111 1111：保留。

位 23:16 **MEMHOLD**：通用存储器保持时间 (Common memory hold time)
这些位针对在 NAND Flash 的通用存储空间执行的读或写访问，定义禁止命令（NWE、NOE）之后保持地址（和写访问数据）的写入访问 fmc_ker_ck 时钟周期数和读取访问 fmc_ker_ck+1 时钟周期数：
0000 0000：保留。
0000 0001：写入访问 1 个 fmc_ker_ck 周期/读取访问 3 个 fmc_ker_ck 周期
1111 1110：写入访问 254 个 fmc_ker_ck 周期/读取访问 257 个 fmc_ker_ck 周期
1111 1111：保留。

位 15:8 **MEMWAIT**：通用存储器等待时间 (Common memory wait time)

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

0000 0000：保留

0000 0001：个 fmc_ker_ck 周期（+ 禁止 NWAIT 时引入的等待周期）

1111 1110：255 个 fmc_ker_ck 周期（+ 禁止 NWAIT 时引入的等待周期）

1111 1111：保留。

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

这些位针对在 NAND Flash 的通用存储空间执行的读或写访问，定义使能命令（NWE，NOE）前建立地址所需的 fmc_ker_ck (+1) 时钟周期数：

0000 0000：fmc_ker_ck 周期

1111 1110：255 个 fmc_ker_ck 周期

1111 1111：保留。

特性存储器空间时序寄存器 (FMC_PATT)

Attribute memory space timing registers

偏移地址：0x8C

复位值：0xFCFC FCFC

FMC_PATT 读/写寄存器包含 NAND Flash 存储区域的时序信息。当对 NAND Flash 的特性存储空间进行 8 位访问时，如果最后一次地址写访问的时序必须与先前访问的时序不同，则使用该寄存器（有关就绪/繁忙管理的信息，请参见*第 22.8.5 节：NAND Flash 预等待功能*）。

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

位 31:24 **ATTHIZx**：特性存储器数据总线高阻态时间 (Attribute memory data bus Hi-Z time)

这些位定义在存储区上对 NAND Flash 的特性存储空间开始执行写访问之后，数据总线保持高阻态所持续的 fmc_ker_ck 时钟周期数。仅对写入事务有效：

0000 0000：0 个 fmc_ker_ck 周期

1111 1110：254 个 fmc_ker_ck 周期

1111 1111：保留。

位 23:16 **ATTHOLD**：特性存储器保持时间 (Attribute memory hold time)

这些位针对在 NAND Flash 的特性存储空间执行的读或写访问，定义禁止命令（NWE、NOE）之后保持地址（和写访问数据）的 fmc_ker_ck 时钟周期数：

0000 0000：保留

0000 0001：1 个 fmc_ker_ck 周期

1111 1110：254 个 fmc_ker_ck 周期

1111 1111：保留。

位 15:8 **ATTWAIT**：特性存储器等待时间 (Attribute memory wait time)

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

0000 0000：保留

0000 0001：2 个 fmc_ker_ck 周期（+ 禁止 NWAIT 时引入的等待周期）

1111 1110：255 个 fmc_ker_ck 周期（+ 禁止 NWAIT 时引入的等待周期）

1111 1111：保留。

位 7:0 **ATTSET**：特性存储器建立时间 (Attribute memory setup time)

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

0000 0000：1 个 fmc_ker_ck 周期

1111 1110：255 个 fmc_ker_ck 周期

1111 1111：保留。

ECC 结果寄存器 (FMC_ECCR)

ECC result registers

偏移地址：0x94

复位值：0x0000 0000

该寄存器包含由 FMC NAND 控制器的 ECC 计算模块计算所得的当前纠错码值。当 CPU 从 NAND Flash 页的正确地址读取/写入数据时（请参见 *第 22.8.6 节：纠错码 (ECC) 计算 (NAND Flash)*），ECC 计算模块会自动处理读取自/写入 NAND Flash 的数据。读取 X 个字节后（依据 FMC_PCR 寄存器的 ECCPS 字段），CPU 必须从 FMC_ECC 寄存器读取计算所得的 ECC 值。然后验证这些计算的奇偶校验数据是否与备用区记录的奇偶校验数据相同，从而确定该页是否有效，如不同则进行校正。FMC_ECCR 寄存器被读取后应通过将 ECCEN 位置 0 进行清零。如要计算新的数据块，则必须将 ECCEN 位置 1。

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

位 31:0 **ECC[31:0]**：ECC 结果 (ECC result)

该字段包含由 ECC 计算逻辑计算所得的值。*表 178* 介绍了这些位域的内容。

表 178. ECC 结果相关位

ECCPS[2:0]	以字节为单位的页大小	ECC 位
000	256	ECC[21:0]
001	512	ECC[23:0]
010	1024	ECC[25:0]
011	2048	ECC[27:0]
100	4096	ECC[29:0]
101	8192	ECC[31:0]

22.9 SDRAM 控制器

22.9.1 SDRAM 控制器主要特性

SDRAM 控制器的主要特性如下：

- 两个 SDRAM 存储区域，可独立配置
- 8 位、16 位和 32 位数据总线宽度
- 13 位地址行，11 位地址列，4 个内部存储区域：4x16Mx32bit (256 MB)、4x16Mx16bit (128 MB)、4x16Mx8bit (64 MB)
- 支持字、半字和字节访问
- SDRAM 时钟可以是 fmc_ker_ck/2 或 fmc_ker_ck/3
- 自动进行行和存储区域边界管理
- 多存储区域乒乓访问
- 可编程时序参数
- 支持自动刷新操作，可编程刷新速率
- 自刷新模式
- 掉电模式
- 通过软件进行 SDRAM 上电初始化
- CAS 延迟 1,2,3
- 读 FIFO 可缓存，支持 6 行 x 32 位深度（6 x14 位地址标记）

22.9.2 SDRAM 外部存储器接口信号

启动时，必须通过用户应用程序对用于连接 FMC SDRAM 控制器与外部 SDRAM 设备的 SDRAM I/O 引脚进行配置。应用程序未使用的 SDRAM 控制器 I/O 引脚可用于其它用途。

表 179. SDRAM 信号

SDRAM 信号	I/O 类型	说明	复用功能
SDCLK	O	SDRAM 时钟	-
SDCKE[1:0]	O	SDCKE0：SDRAM 存储区域 1 时钟使能 SDCKE1：SDRAM 存储区域 2 时钟使能	-
SDNE[1:0]	O	SDNE0：SDRAM 存储区域 1 芯片使能 SDNE1：SDRAM 存储区域 2 芯片使能	-
A[12:0]	O	地址	FMC_A[12:0]
D[31:0]	I/O	双向数据总线	FMC_D[31:0]
BA[1:0]	O	存储区域地址	FMC_A[15:14]
NRAS	O	行地址选通	-
NCAS	O	列地址选通	-
SDNWE	O	写入使能	-
NBL[3:0]	O	写访问的输出字节屏蔽 （存储器信号名称：DQM[3:0]）	FMC_NBL[3:0]

22.9.3 SDRAM 控制器功能说明

所有 SDRAM 控制器输出（信号、地址和数据）在存储器时钟 (FMC_SDCLK) 的下降沿上变化。

SDRAM 初始化

初始化序列通过软件进行管理。如果使用了两个存储区域，则必须将 FMC_SDCMR 寄存器中的目标存储区域位 CTB1 和 CTB2 置 1，同时为存储区域 1 和存储区域 2 生成初始化序列：

1. 将存储器件的特性编程到 FMC_SDCRx 寄存器中。SDRAM 时钟频率、RBURST 和 RPIPE 特性必须编程到 FMC_SDCR1 寄存器中。
2. 将存储器设备的时序编程到 FMC_SDTRx 寄存器中。TRP 和 TRC 时序必须编程到 FMC_SDTR1 寄存器中。
3. 将 MODE 位置为“001”并配置 FMC_SDCMR 寄存器中的目标存储区域位（CTB1 和/或 CTB2）以开始为存储器提供时钟信号（SDCKE 驱动为高电平）。
4. 等待指定延迟周期。典型延迟为 100 µs（有关上电后所需延迟的信息，请参见 SDRAM 数据手册）。
5. 将 MODE 位置为“010”并配置 FMC_SDCMR 寄存器中的目标存储区域位（CTB1 和/或 CTB2）以发送“全部预充电”命令。
6. 将 MODE 位置为“011”并配置 FMC_SDCMR 寄存器中的目标存储区位（CTB1 和/或 CTB2）和连续自动刷新命令 (NRFS) 的数量。请参见 SDRAM 数据手册了解应发出的自动刷新命令个数。通常为 8 个。
7. 配置 MRD 字段，将 MODE 位置为“100”并配置 FMC_SDCMR 寄存器中的目标存储区域位（CTB1 和/或 CTB2）以发送“加载模式寄存器”命令并对 SDRAM 设备进行编程。尤其突发长度 (BL) 必须置“1”且必须选择 CAS 延迟。如果两个 SDRAM 存储区域的模式寄存器不同，则此步骤必须重复两次，每个存储区域各一次且目标存储区域位相应置 1。对于移动 SDRAM 器件，MRD 字段还用于配置扩展模式寄存器，同时发出加载模式寄存器命令。
8. 编程 FMC_SDRTR 寄存器中的刷新速率

   刷新速率对应于刷新周期之间的延迟。其值必须与 SDRAM 设备相适应。

在这一阶段，SDRAM 设备已做好接受命令的准备。如果进行 SDRAM 访问期间发生系统复位，则数据总线仍可能由 SDRAM 设备驱动。因此，必须在复位后重新初始化 SDRAM 设备，NOR Flash/PSRAM/SRAM 或 NAND Flash 控制器才能发送新的访问命令。

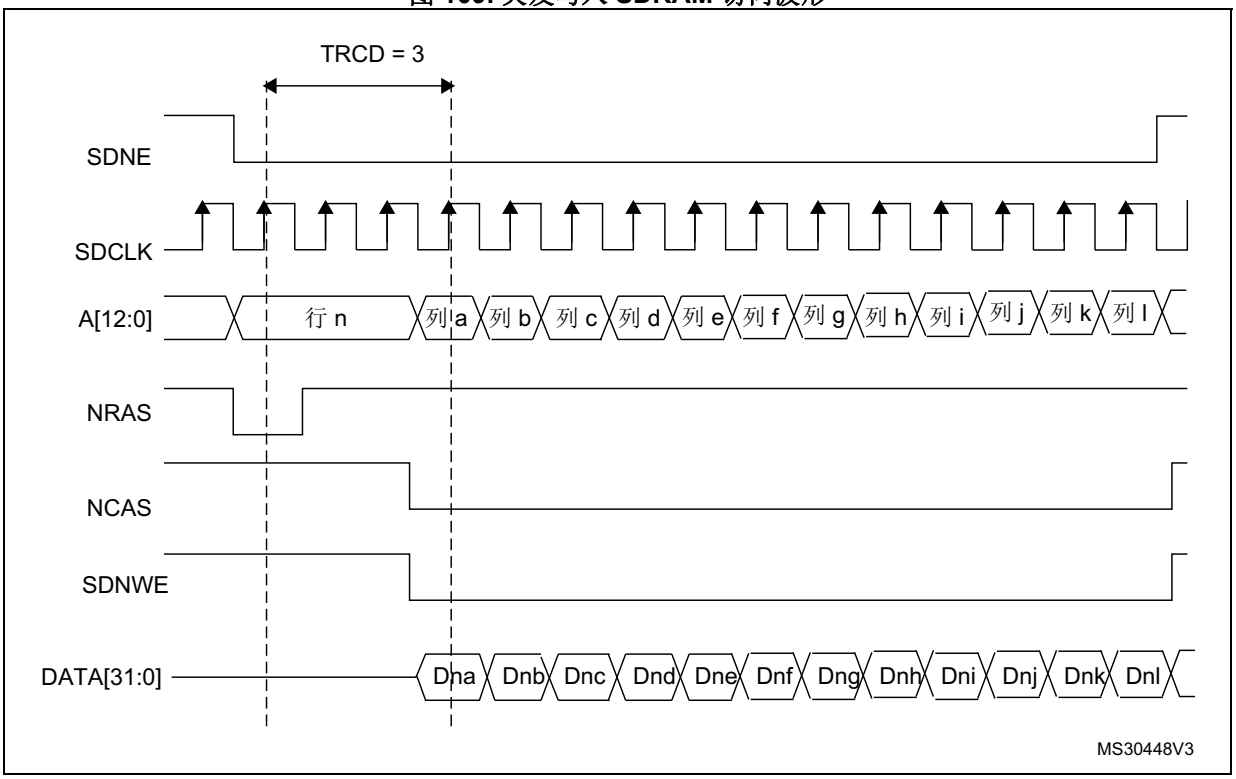
注： *如果有两个 SDRAM 设备连接到 FMC，对命令模式寄存器同时访问这两个器件（加载模式寄存器命令）的情况，将按 FMC_SDTR1 寄存器中为 SDRAM 存储区域 1 配置的时序参数（TMRD 和 TRAS 时序）发出访问命令。*

SDRAM 控制器写周期

SDRAM 控制器可接收单次的和突发的写请求，并将其视为单次存储器访问。在这两种情况下，SDRAM 控制器都会跟踪各存储区域的有效行，以能够对不同的存储区域进行连续的写访问（多存储区域乒乓访问）。

执行任何写访问前，必须将 FMC_SDCRx 寄存器中的 WP 位清零，禁止 SDRAM 存储区域的写保护。

图 109. 突发写入 SDRAM 访问波形



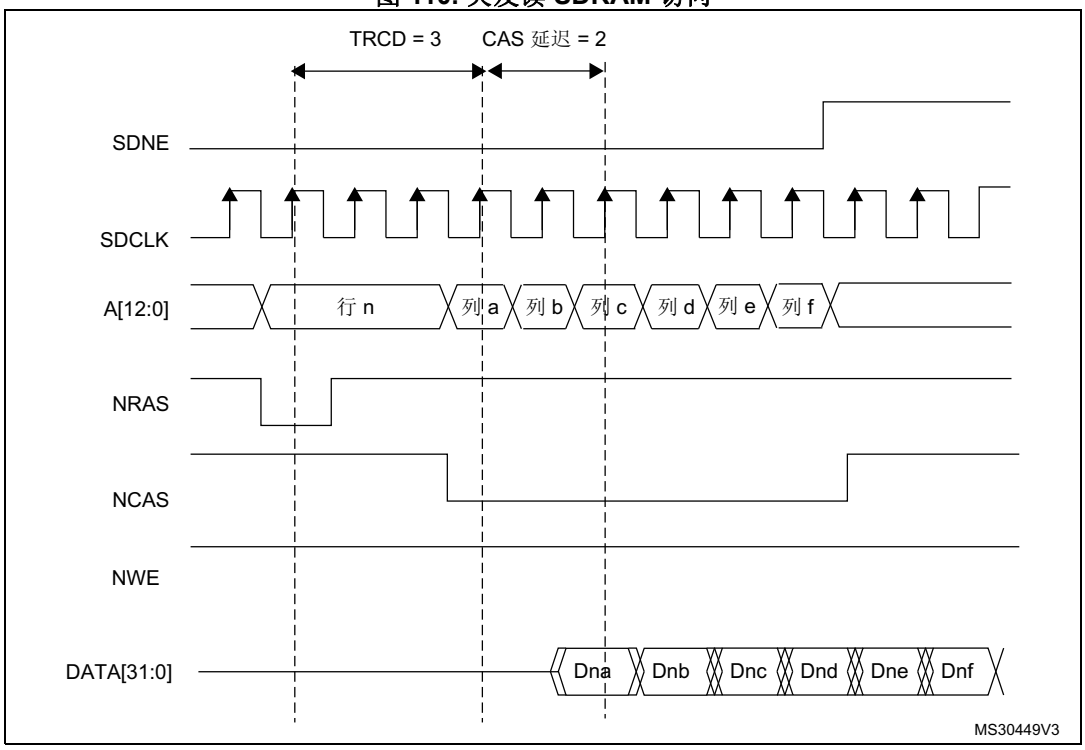
SDRAM 控制器始终会检查下一个访问。

- 如果下一个访问发生在同一行或在其它有效行，则直接执行写操作。
- 如果下一个访问指向一个无效行，则 SDRAM 控制器将生成预充电命令、激活该新行并初始化写命令。

SDRAM 控制器读周期

SDRAM 控制器可接收单次的和突发的读请求，并将其视为单次存储器访问。在这两种情况下，SDRAM 控制器都会跟踪各存储区域的有效行，以能够对不同的存储区域进行连续的读访问（多存储区域乒乓访问）。

图 110. 突发读 SDRAM 访问



FMC SDRAM 控制器具有可缓存的读取 FIFO（6 行 x 32 位），用于存储在 CAS 延迟周期（最多 3 个存储器时钟周期，在 FMC_SDCRx 中配置）和 RPIPE 延迟（设为 2 个 fmc_ker_ck 时钟周期时，在 FMC_SDCR1 中配置）期间提前读取的数据，依据的公式如下：CAS Latency + 1 + (RPIPE DIV2)。必须将 FMC_SDCR1 寄存器中的 RBURST 位置 1 才能接受下一个读访问。

示例

- CAS=3，RPIPE= 2xfmc_ker_ck。这种情况下，FIFO 中将存储 5 个未提交的数据（CAS 延迟期间读取的 4 个数据和 RPIPE 延迟期间读取的 1 个数据）
- CAS=3，RPIPE= 1xfmc_ker_ck。这种情况下，FIFO 中将存储 4 个未提交的数据（CAS 延迟期间读取的 4 个数据）

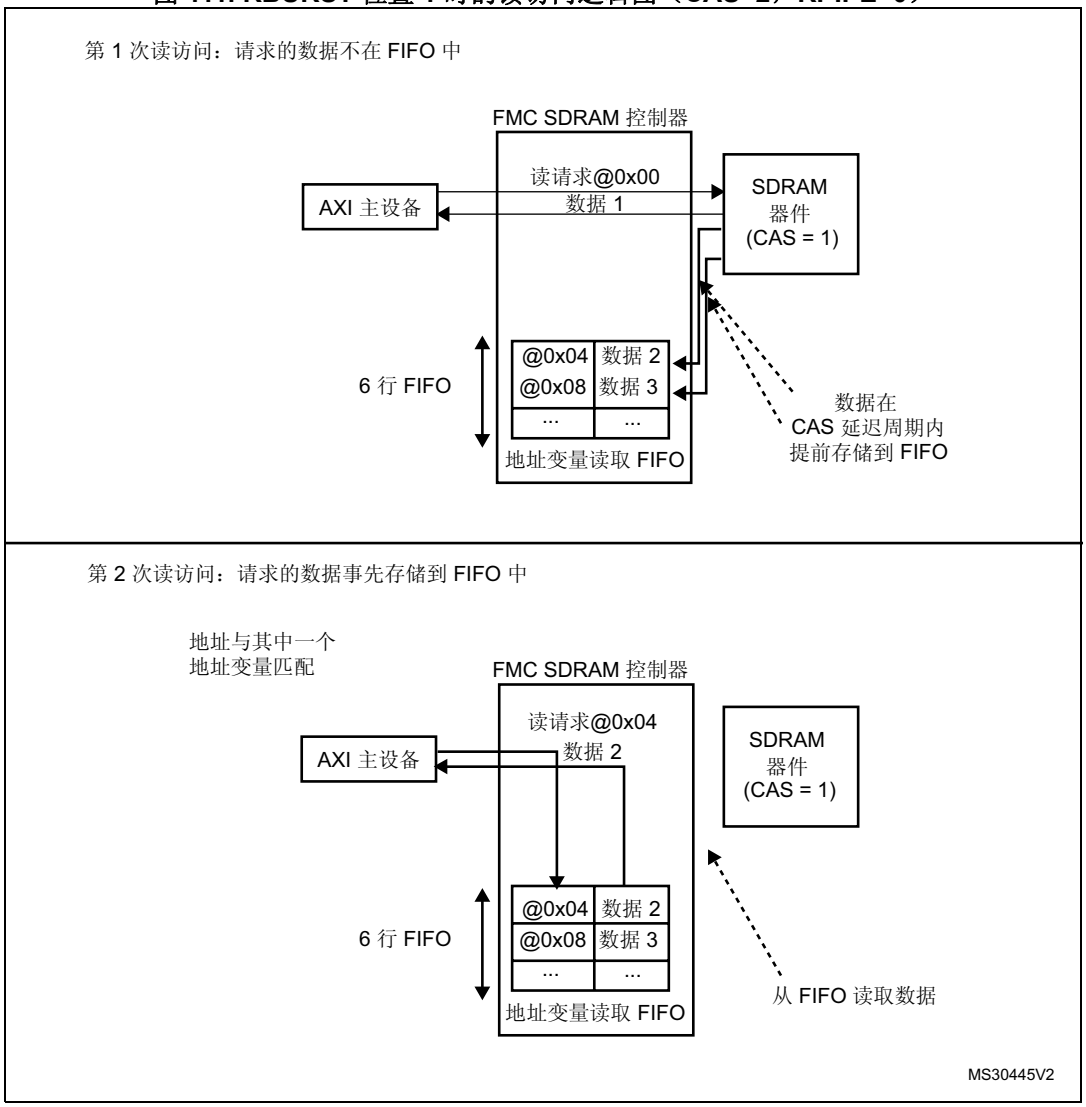
读 FIFO 的每行都具有 14 位地址标记用于标识自身内容：11 位用于表示列地址，2 位用于表示选择内部存储区域和有效行，1 位用于选择 SDRAM 设备

在突发读取事务期间，如果提前到达行末尾，则提前读取的数据（未提交）不存储到读 FIFO。对于单次读访问，数据将正确存储到读 FIFO。

每当出现读请求时，SDRAM 控制器将检查：

- 地址与地址标记之一是否匹配，如果找到匹配，则直接从 FIFO 读取数据并清空相应地址标记/行内容，同时压缩 FIFO 中的剩余数据以避免空行。
- 否则，向存储器发送新的读命令并用新数据更新 FIFO。如果 FIFO 已满，则较早的数据将丢失。

图 111. RBURST 位置 1 时的读访问逻辑图（CAS=2，RPIPE=0）



读访问或预充电命令期间，读 FIFO 将刷新并做好填充新数据的准备。

接到第一个读请求后，如果当前访问还未进行到行边界，则 SDRAM 控制器将在 CAS 延迟周期和 RPIPE 延迟（如果已配置）期间接受下一个读访问。这将通过递增存储器地址来实现。必须满足以下条件：

- FMC_SDCR1 寄存器中的 RBURST 控制位必须置“1”。

地址管理取决于下一个 AXI 请求：

- 下一个请求是连续的（突发访问）

  这种情况下，SDRAM 控制器将递增地址。
- 下一个请求不连续
  - 如果新的读请求指向与上一请求相同的行或另一个有效行，则新地址将传送给存储器，同时主设备在 CAS 延迟周期停止工作，等待从存储器获取新数据。
  - 如果新的读请求指向无效行，则 SDRAM 控制器生成预充电命令、激活该新行并初始化读命令。

如果 RBURST 位置 0，则不使用读 FIFO。

行和存储区域边界管理

当读/写访问跨越了行边界时，如果下一个读/写访问是连续的并且当前访问已执行到行边界，则 SDRAM 控制器将执行以下操作：

1. 对有效行进行预充电
2. 激活新行
3. 启动读/写命令

对于各种列和数据总线宽度配置，都支持在行边界自动激活下一行。

SDRAM 控制器可根据需要在以下命令之间插入附加时钟周期：

- 在预充电和激活命令之间插入以匹配 TRP 参数（仅当下一个访问指向同一存储区域中的其它行时）
- 在激活和读命令之间插入以匹配 TRCD 参数

这些参数在 FMC_SDTRx 寄存器中定义。

有关跨越行边界读取和突发写访问的信息，请参见*图 109* 和*图 110*。

图 112. 跨行边界的读访问

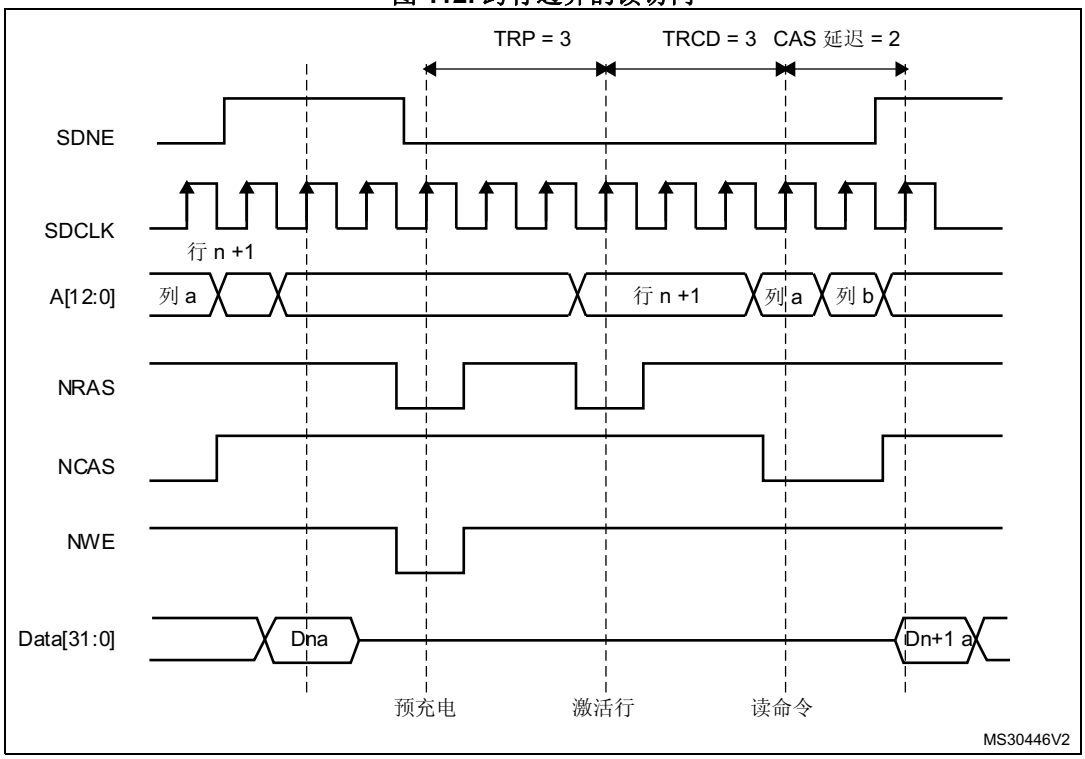
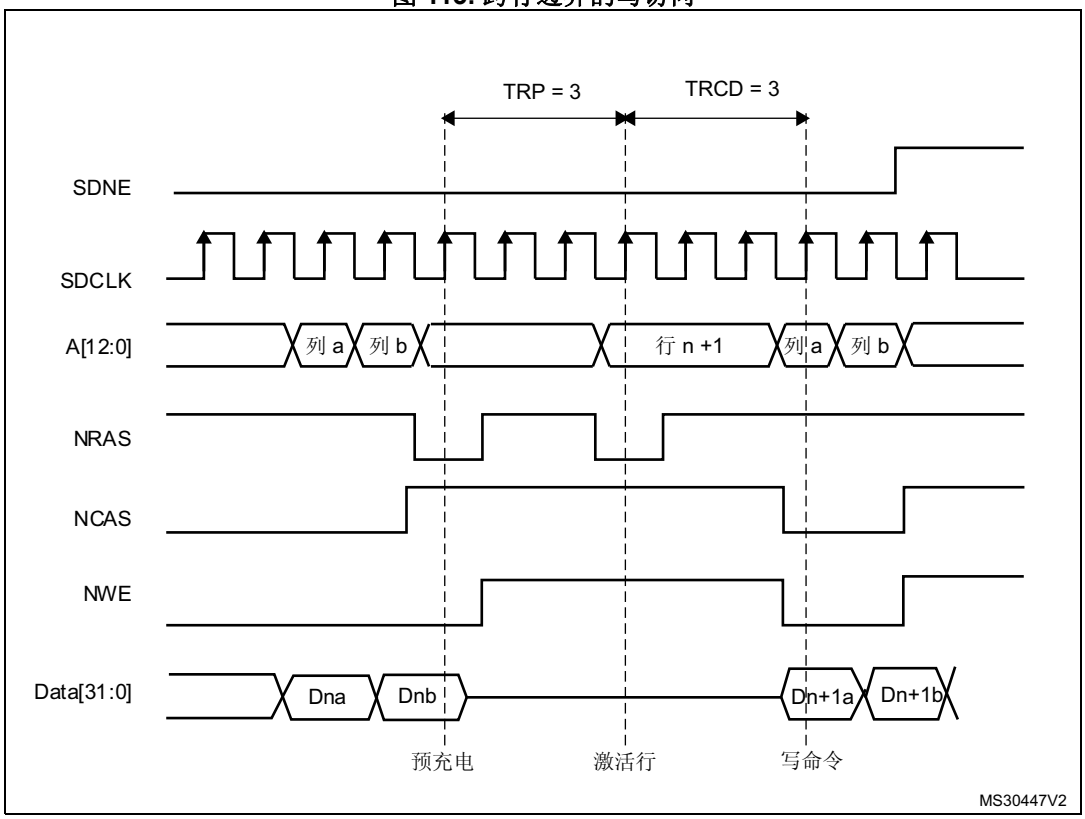


图 113. 跨行边界的写访问



如果下一个访问是连续的并且当前访问跨越了存储区域边界，则 SDRAM 控制器将激活下一个存储区域的第一个行并发出新的读/写命令。可能存在以下两种情况：

- 如果当前存储区域不是最后一个存储区域，则必须对新存储区域中的激活行进行预充电。对于各种行/列和数据总线宽度配置，都支持在存储区域边界自动激活下一行。
- 如果当前存储区域是最后一个存储区域，则仅在对 13 位行、11 位列、4 个内部存储区域和 32 位数据总线的 SDRAM 设备寻址时，支持自动激活下一行。否则，将违反 SDRAM 地址范围并生成 AXI 从错误。
- 对于 13 位行地址、11 位列地址、4 个内部存储区域和总线宽度为 32 位的 SDRAM 存储器，SDRAM 控制器将通过第二个 SDRAM 设备继续进行读/写操作（假设第二个 SDRAM 设备已初始化）：
  a) SDRAM 控制器将激活第一行（在对有效行预充电之后，假设第一个内部存储区域中存在有效行）并发出新的读/写命令。
  b) 如果第一行已激活，则 SDRAM 控制器将仅发出读/写命令。

SDRAM 控制器刷新周期

自动刷新命令用于刷新 SDRAM 设备的内容。SDRAM 控制器会定期发送自动刷新命令。它使用一个内部计数器装载 FMC_SDRTR 寄存器中的 COUNT 值。该值定义刷新周期之间的存储器时钟周期个数（刷新速率）。该计数器的值达到零时将生成一个内部脉冲。

如果存在进行中的存储器访问，则会延迟自动刷新请求。不过，在存储器访问和自动刷新请求同时出现时，则优先处理自动刷新请求。

如果在自动刷新期间访问存储器，则会缓存访问请求并在自动刷新完成后进行处理。

如果在上一个自动刷新请求尚未完成的情况下又出现了新的自动刷新请求，则状态寄存器中的 RE（刷新错误）位将置 1。该位如果已使能（REIE = “1”），将生成中断。

如果 SDRAM 的行不是空闲状态（并非所有行都已关闭），则 SDRAM 控制器将生成 PALL（全部预充电）命令，然后再进行自动刷新。

如果由 FMC_SDCMR 命令模式寄存器（模式位 = “011”）生成自动刷新命令，则必须先发出 PALL 命令（模式位 = “010”）。

22.9.4 低功耗模式

可使用两种低功耗模式：

- 自刷新模式
  由 SDRAM 设备自身执行自动刷新循环以保留数据，无需外部时钟。
- 掉电模式
  由 SDRAM 控制器执行自动刷新循环。

自刷新模式

通过将 MODE 位置为“101”并配置 FMC_SDCMR 寄存器中的目标存储区域位（CTB1 和/或 CTB2）来选择该模式。

SDRAM 时钟在 TRAS 延迟后停止运行，而内部刷新定时器只有在满足以下条件之一时才停止计数：

- 向两个设备都发出了自刷新命令。
- 其中一个设备未激活（SDRAM 存储区域未初始化）。

进入自刷新模式前，SDRAM 控制器会自动发送 PALL 命令。

如果写数据 FIFO 非空，则所有数据都将在自刷新模式激活前发送到存储器并且 BUSY 状态标志保持置 1。

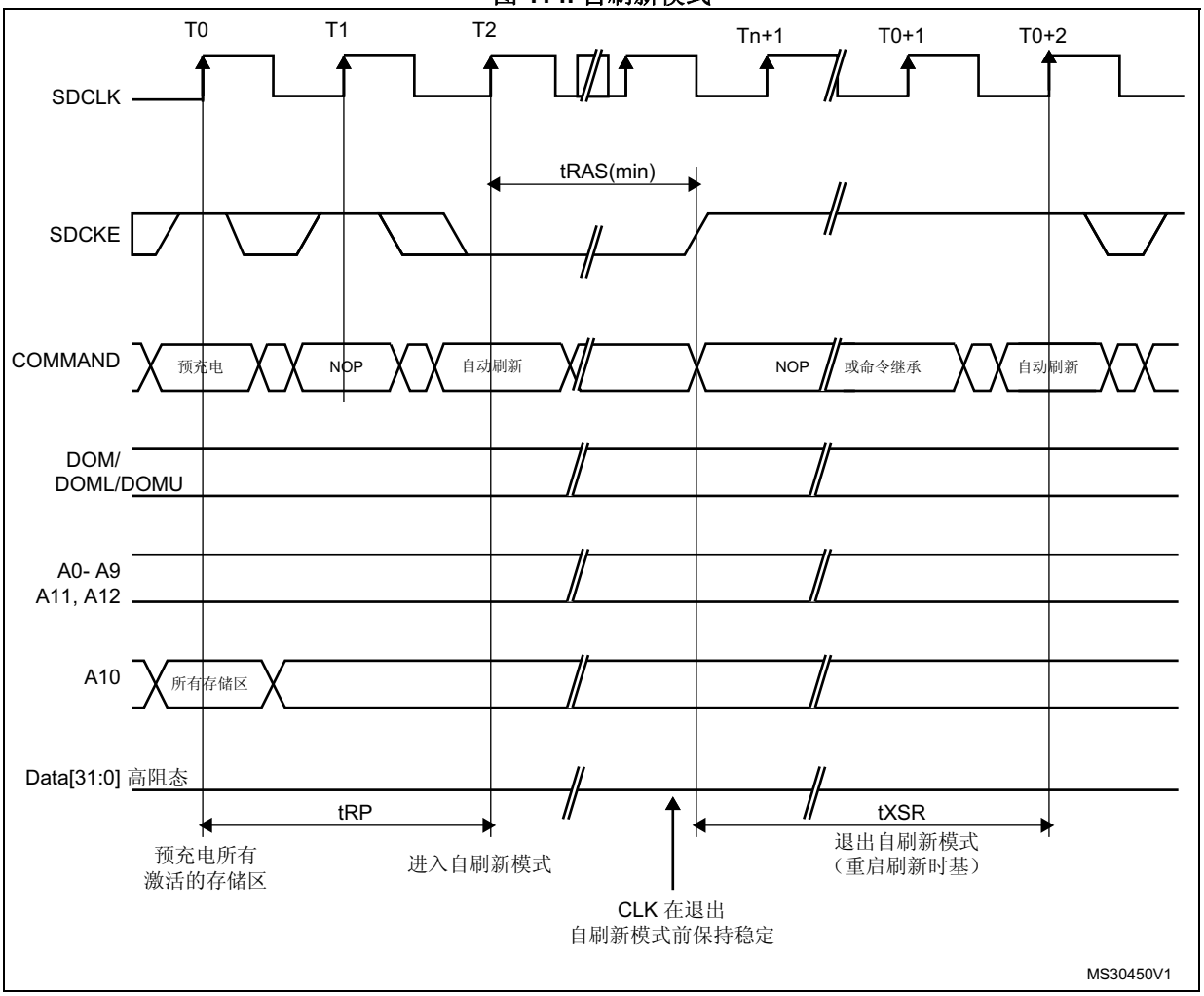
在自刷新模式下，除保持低电平的 SDCKE 外，SDRAM 设备的所有输入都无效。

SDRAM 设备必须处于自刷新模式最短为 TRAS 时间，且能够在更长的时间内始终处于自刷新模式。为保证这一最短时长，在自刷新激活后的 TRAS 延迟期间，BUSY 状态标志将保持高电平。

SDRAM 控制器会在有 SDRAM 设备被选定后立即生成一个命令序列以退出自刷新模式。存储器访问完成后，选定的设备将保持正常模式。

要退出自刷新模式，必须将 MODE 位置为“000”（正常模式）并配置 FMC_SDCMR 寄存器中的目标存储区域位（CTB1 和/或 CTB2）。

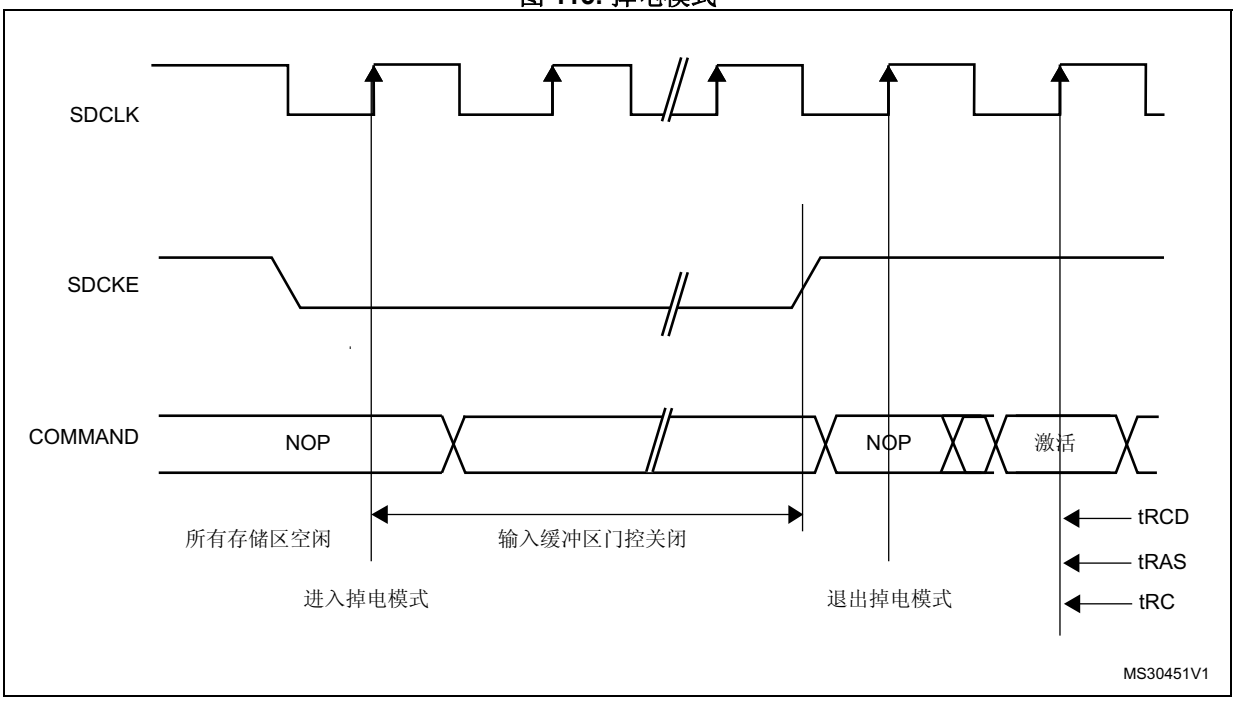
图 114. 自刷新模式



掉电模式

通过将 MODE 位置为“110”并配置 FMC_SDCMR 寄存器中的目标存储区域位（CTB1 和/或 CTB2）来选择该模式。

图 115. 掉电模式



如果写数据 FIFO 非空，则所有数据都将在掉电模式激活前发送到存储器。

SDRAM 控制器会在有 SDRAM 设备被选定后立即退出掉电模式。存储器访问完成后，选定的 SDRAM 设备将保持正常模式。

在掉电模式期间，将禁用 SDRAM 设备的所有输入/输出缓冲区，只有保持低电平的 SDCKE 除外。

SDRAM 设备保持掉电模式的时间不会长于刷新周期，并且自身无法执行自刷新循环。因此，SDRAM 控制器通过以下操作执行刷新：

1. 退出掉电模式并将 SDCKE 驱动为高电平
2. 生成 PALL 命令（前提是在掉电模式下存在激活行）
3. 生成自动刷新命令
4. 再次将 SDCKE 驱动为低电平以返回掉电模式

要退出掉电模式，必须将 MODE 位置为“000”（正常模式）并配置 FMC_SDCMR 寄存器中的目标存储区域位（CTB1 和/或 CTB2）。

22.9.5 SDRAM 控制寄存器

SDRAM 控制寄存器 1，2（FMC_SDCR1，FMC_SDCR2）

SDRAM Control registers 1,2

偏移地址：0x140+ 4* (x – 1)，x = 1,2

复位值：0x0000 02D0

此寄存器包含每个 SDRAM 存储区域的控制参数

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	15
Res.	RPIPE[1:0]		RBURST	SDCLK[1:0]		WP	CAS[1:0]		NB	MWID[1:0]		NR[1:0]		NC51:0]		Res.
	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	

位 31:15 保留，必须保持复位值

位 14:13 **RPIPE[1:0]**：读管道 (Read pipe)

这些位可定义在 CAS 延时后延后多少个 fmc_ker_ck 时钟周期读取数据。

00：无 fmc_ker_ck 时钟周期延迟

01：一个 fmc_ker_ck 时钟周期延迟

10：两个 fmc_ker_ck 时钟周期延迟

11：保留。

注： FMC_SDCR2 寄存器中的相应位为只读位。

位 12 **RBURST**：突发读 (Burst read)

此位可使能突发读模式。SDRAM 控制器预期在 CAS 延迟期间接受下一个读命令并将数据存储在读 FIFO 中。

0：不将单次读请求作为突发请求管理

1：始终将单次读请求作为突发请求管理

注： FMC_SDCR2 寄存器中的相应位为只读位。

位 11:10 **SDCLK[1:0]**：SDRAM 时钟配置 (SDRAM clock configuration)

这些位用于定义两个 SDRAM 存储区域的 SDRAM 时钟周期以及在更改频率前禁止时钟。此时，必须重新初始化 SDRAM。

00：禁止 SDCLK 时钟

01：保留

10：SDCLK 周期 = 2 个 fmc_ker_ck 周期

11：SDCLK 周期 = 3 个 fmc_ker_ck 周期

注： FMC_SDCR2 寄存器中的相应位为只读位。

位 9 **WP**：写保护 (Write protection)

该位可使能对 SDRAM 存储区域的写模式访问。

0：允许写访问

1：忽略写访问

位 8:7 **CAS[1:0]**：CAS 延迟 (CAS Latency)

该位可设置 SDRAM CAS 延迟，按存储器时钟周期计

00：保留。

01：1 个周期

10：2 个周期

11：3 个周期

位 6 **NB**：内部存储区域数量 (Number of internal banks)

该位可设置内部存储区域数量。

0：2 个内部存储区域

1：4 个内部存储区域

位 5:4 **MWID[1:0]**：存储器数据总线宽度 (Memory data bus width)。

这些位定义存储器件宽度。

00：8 位

01：16 位

10：32 位

11：保留。

位 3:2 **NR[1:0]**：行地址位数 (Number of row address bits)

这些位定义行地址的位数。

00：11 位

01：12 位

10：13 位

11：保留。

位 1:0 **NC[1:0]**：列地址位数 (Number of column address bits)

这些位定义列地址的位数。

00：8 位

01：9 位

10：10 位

11：11 位

注： *修改 RBURST 或 RPIPE 设置或者禁止 SDCLK 时钟之前，用户必须先发送 PALL 命令以确保先完成正在进行的操作。*

SDRAM 时序寄存器 1，2（FMC_SDTR1，FMC_SDTR2）

SDRAM Timing registers 1,2

偏移地址：0x148 + 4 * (x – 1)，x = 1，2

复位值：0x0FFF FFFF

此寄存器包含每个 SDRAM 存储区域的时序参数

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	TRCD				TRP				TWR			
				rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TRC				TRAS				TXSR				TMRD			
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:28 保留，必须保持复位值

位 27:24 **TRCD[3:0]**：行到列延迟 (Row to column delay)

这些位定义激活命令与读/写命令之间的延迟，按存储器时钟周期数计。

0000：1 个周期

0001：2 个周期

....

1111：16 个周期

位 23:20 **TRP[3:0]**：行预充电延迟 (Row precharge delay)

这些位定义预充电命令与其它命令之间的延迟，按存储器时钟周期数计。仅在 FMC_SDTR1 寄存器中配置 TRP 时序。如果使用了两个 SDRAM 设备，则必须使用最慢设备的时序配置 TRP。

0000：1 个周期

0001：2 个周期

....

1111：16 个周期

注： FMC_SDTR2 寄存器中的相应位为无关位。

位 19:16 **TWR[3:0]**：恢复延迟 (Recovery delay)

这些位定义写命令和预充电命令之间的延迟，按存储器时钟周期数计。

0000：1 个周期

0001：2 个周期

....

1111：16 个周期

注： TWR 必须设置成与 SDRAM 数据手册中定义的写恢复时间 (t_{WR}) 相匹配，保证：

TWR ³ TRAS - TRCD，并且 TWR ³ TRC - TRCD - TRP

例如：TRAS= 4 个周期，TRCD= 2 个周期。因此，TWR >= 2 个周期。TWR 必须设置为 0x1。

如果使用了两个 SDRAM 设备，则必须为 FMC_SDTR1 和 FMC_SDTR2 配置相同的 TWR 时序（对应于较慢的 SDRAM 设备）。

位 15:12 **TRC[3:0]**：行循环延迟 (Row cycle delay)

这些位定义刷新命令和激活命令之间的延迟，以及两个相邻刷新命令之间的延迟，以存储器时钟周期数表示。仅在 FMC_SDTR1 寄存器中配置 TRC 时序。如果使用了两个 SDRAM 设备，则必须使用最慢设备的时序配置 TRC。

0000：1 个周期

0001：2 个周期

....

1111：16 个周期

注： TRC 必须与 SDRAM 设备数据手册中定义的 TRC 和 TRFC（自动刷新周期）时序相匹配。

注： FMC_SDTR2 寄存器中的相应位为无关位。

位 11:8 **TRAS[3:0]**：自刷新时间 (Self refresh time)

这些位定义最短的自刷新周期，按存储器时钟周期数计。

0000：1 个周期

0001：2 个周期

....

1111：16 个周期

位 7:4 **TXSR[3:0]**：退出自刷新延迟 (Exit Self-refresh delay)

这些位定义从发出自刷新命令到发出激活命令之间的延迟，按存储器时钟周期数计。

0000：1 个周期

0001：2 个周期

....

1111：16 个周期

注： 如果使用了两个 SDRAM 设备，则必须为 FMC_SDTR1 和 FMC_SDTR2 配置相同的 TXSR 时序（对应于较慢的 SDRAM 设备）。

位 3:0 **TMRD[3:0]**：加载模式寄存器到激活 (Load Mode Register to Active)

这些位定义加载模式寄存器命令和激活或刷新命令之间的延迟，按存储器时钟周期计。
0000：1 个周期
0001：2 个周期
....
1111：16 个周期

注： *如果连接了两个 SDRAM 设备，对于命令模式寄存器同时访问这两个设备（加载模式寄存器命令）的情况，将按 FMC_SDTR1 寄存器中为存储区域 1 配置的时序参数（TMRD 和 TRAS 时序）发出访问命令。*

仅在 FMC_SDTR1 寄存器中配置 TRP 和 TRC 时序。如果使用了两个 SDRAM 设备，则必须使用最慢设备的时序配置 TRP 和 TRC 时序。

SDRAM 命令模式寄存器 (FMC_SDCMR)

SDRAM Command Mode register

地址偏移：0x150

复位值：0x0000 0000

该寄存器包含访问 SDRAM 设备时所发出的命令。该寄存器用于初始化 SDRAM 设备、激活自刷新模式和掉电模式。写入 MODE 字段后，将根据 CTB1 和 CTB2 命令位向单个或全部两个 SDRAM 存储区域发送命令。该寄存器为两个 SDRAM 存储区域所共用。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MRD						
									rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MRD							NRFS				CTB1	CTB2	MODE		
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:23 保留，必须保持复位值

位 22:9 **MRD[13:0]**：模式寄存器定义 (Mode Register definition)

此 14 位域定义 SDRAM 模式寄存器内容。可通过加载模式寄存器命令对模式寄存器进行编程。MRD [13:0] 位还用于编程 Mobile SDRAM 的扩展模式寄存器。

位 8:5 **NRFS[3:0]**：自刷新次数 (Number of Auto-refresh)

这些位定义 MODE =“011”时所发出的连续自刷新命令个数。
0000：1 个自刷新周期
0001：2 个自刷新周期
....
1110：15 个自刷新周期
1111：16 个自刷新周期

位 4 **CTB1**：命令目标存储区域 1 (Command Target Bank 1)

该位指示是否向 SDRAM 存储器 1 发送命令。
0：命令未发送到 SDRAM 存储区域 1
1：命令已发送到 SDRAM 存储区域 1

位 3 **CTB2**：命令目标存储区域 2 (Command Target Bank 2)

该位指示是否向 SDRAM 存储器 2 发送命令。

0：命令未发送到 SDRAM 存储区域 2

1：命令已发送到 SDRAM 存储区域 2

位 2:0 **MODE[2:0]**：命令模式 (Command mode)

这些位定义发送到 SDRAM 设备的命令。

000：正常模式

001：时钟配置使能

010：PALL（“预充电所有存储区域”）命令

011：自刷新命令

100：加载模式寄存器

101：自刷新命令

110：掉电命令

111：保留

注：　命令发出后，至少一个命令目标存储区域位（CTB1 或 CTB2）必须置 1，否则该命令将被忽略。

注：　如果使用两个 SDRAM 存储区域，则必须通过将 CTB1 和 CTB2 位置 1 来向两个器件同时发送自刷新和 PALL 命令，否则该命令将被忽略。

注：　如果只使用一个 SDRAM 存储区域，并通过将其关联的 CTB 位置 1 来发出命令，则未使用存储区域的另一个 CTB 位必须保持为 0。

SDRAM 刷新定时器寄存器 (FMC_SDRTR)

SDRAM Refresh Timer register

偏移地址：0x154

复位值：0x0000 0000

该寄存器通过配置刷新定时器计数值来设置刷新循环之间的刷新速率，按 SDCLK 时钟周期数计。

$$\text{刷新速率} = (\text{COUNT} + 1) \times \text{SDRAM 时钟频率}$$

$$\text{COUNT} = ((\text{SDRAM 刷新周期})/(\text{行数})) - 20$$

示例

$$\text{刷新速率} = 64\ \text{ms}/(8196\text{行}) = 7.81\mu\text{s}$$

其中 64 ms 是 SDRAM 的刷新周期。

$$7.81\mu\text{s} \times 60\text{MHz} = 468.6$$

如果在接受读请求后出现内部刷新请求，则必须将刷新速率增加 20 个 SDRAM 时钟周期（如上所示）以获得重充足的裕量。其对应的 COUNT 值为“0000111000000”(448)。

该 13 位域将加载到使用 SDRAM 时钟递减的定时器。此定时器在计数到零时生成刷新脉冲。COUNT 值必须设置为至少 41 个 SDRAM 时钟周期。

一旦完成对 FMC_SDRTR 寄存器的编程，定时器就开始计数。如果寄存器中编程的值为“0”，则不会执行刷新。切不可在初始化后重新编程该寄存器以避免刷新速率被修改。

每当生成刷新脉冲时，都会重新将该 13 位 COUNT 字段加载到计数器中。

如果存在进行中的存储器访问，则会延迟自动刷新请求。不过，在存储器访问和自动刷新请求同时出现时，则优先处理自动刷新请求。如果在刷新期间访问存储器，则会缓存访问请求并在刷新完成后进行处理。

此寄存器为 SDRAM 存储区域 1 和存储区域 2 所共用。

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	REIE	COUNT													CRE
	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	w

位 31:15 保留，必须保持复位值

位 14 **REIE**：RES 中断使能 (RES Interrupt Enable)

0：禁止中断

1：RE = 1 时生成中断

位 13:1 **COUNT[12:0]**：刷新定时器计数 (Refresh Timer Count)

该 13 位域定义 SDRAM 设备的刷新速率，以存储器时钟周期数表示。该字段必须设置为至少 41 个 SDRAM 时钟周期。

刷新速率 = (COUNT + 1) x SDRAM 频率时钟

COUNT =（SDRAM 刷新周期/行数）- 20

位 0 **CRE**：清除刷新错误标志 (Clear Refresh error flag)

该位用于清除状态寄存器中的刷新错误标志 (RE)。

0：无影响

1：清除刷新错误标志

注： *所编程的 COUNT 值不可等于以下时序之和：TWR+TRP+TRC+TRCD+4 个存储器时钟周期。*

SDRAM 状态寄存器 (FMC_SDSR)

SDRAM Status register

偏移地址：0x158

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MODES2		MODES1		RE
											r	r	r	r	r

位 31:5 保留，必须保持复位值

位 4:3 **MODES2**：存储区域 2 的状态模式 (Status Mode for Bank 2)

这些位定义 SDRAM 存储区域 2 的状态模式。

00：正常模式

01：自刷新模式

10：掉电模式

位 2:1 **MODES1**：存储区域 1 的状态模式 (Status Mode for Bank 1)

这些位定义 SDRAM 存储区域 1 的状态模式。

00：正常模式

01：自刷新模式

10：掉电模式

位 0 **RE**：刷新错误标志 (Refresh error flag)

0：未检测到刷新错误

1：检测到刷新错误

在 REIE = 1 且 RE = 1 时会生成中断

22.10 FMC 寄存器映射

下表对 FMC 寄存器进行了汇总。

表 180. FMC 寄存器映射

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	FMC_BCR1	FMCEN	Res.	Res.	Res.	Res.	Res.	BMAP[1:0]		Res.	Res.	WFDIS	CCLKEN	CBURSTRW	CPSIZE[2:0]			ASYNCWAIT	EXTMOD	WAITEN	WREN	WAITCFG	Res.	WAITPOL	BURSTEN	Res.	FACCEN	MWID		MTYP		MUXEN	MBKEN
0x08	**FMC_BCR2**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CBURSTRW	CPSIZE [2:0]			ASYNCWAIT	EXTMOD	WAITEN	WREN	WAITCFG	Res.	WAITPOL	BURSTEN	Res.	FACCEN	MWID [1:0]		MTYP [1:0]		MUXEN	MBKEN
	Reset value													0	0	0	0	0	0	1	1	0		0	0		1	0	1	0	0	1	0
0x10	**FMC_BCR3**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CBURSTRW	CPSIZE [2:0]			ASYNCWAIT	EXTMOD	WAITEN	WREN	WAITCFG	Res.	WAITPOL	BURSTEN	Res.	FACCEN	MWID [1:0]		MTYP [1:0]		MUXEN	MBKEN
	Reset value													0	0	0	0	0	0	1	1	0		0	0		1	0	1	0	0	1	0
0x18	**FMC_BCR4**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CBURSTRW	CPSIZE [2:0]			ASYNCWAIT	EXTMOD	WAITEN	WREN	WAITCFG	Res.	WAITPOL	BURSTEN	Res.	FACCEN	MWID [1:0]		MTYP [1:0]		MUXEN	MBKEN
	Reset value													0	0	0	0	0	0	1	1	0		0	0		1	0	1	0	0	1	0
0x04	**FMC_BTR1**	Res.	Res.	ACCMOD[1:0]		DATLAT[3:0]				CLKDIV[3:0]				BUSTURN[3:0]				DATAST[7:0]									ADDHLD[3:0]			ADDSET[3:0]			
	Reset value			0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0x0C	**FMC_BTR2**	Res.	Res.	ACCMOD[1:0]		DATLAT[3:0]				CLKDIV[3:0]				BUSTURN[3:0]				DATAST[7:0]									ADDHLD[3:0]			ADDSET[3:0]			
	Reset value			0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0x14	**FMC_BTR3**	Res.	Res.	ACCMOD[1:0]		DATLAT[3:0]				CLKDIV[3:0]				BUSTURN[3:0]				DATAST[7:0]									ADDHLD[3:0]			ADDSET[3:0]			
	Reset value			0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0x1C	**FMC_BTR4**	Res.	Res.	ACCMOD[1:0]		DATLAT[3:0]				CLKDIV[3:0]				BUSTURN[3:0]				DATAST[7:0]									ADDHLD[3:0]			ADDSET[3:0]			
	Reset value			0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0x104	**FMC_BWTR1**	Res.	Res.	ACCMOD[1:0]		Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BUSTURN[3:0]				DATAST[7:0]									ADDHLD[3:0]			ADDSET[3:0]			
	Reset value			0	0									1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0x10C	**FMC_BWTR2**	Res.	Res.	ACCMOD[1:0]		Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BUSTURN[3:0]				DATAST[7:0]									ADDHLD[3:0]			ADDSET[3:0]			
	Reset value			0	0									1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0x114	**FMC_BWTR3**	Res.	Res.	ACCMOD[1:0]		Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BUSTURN[3:0]				DATAST[7:0]									ADDHLD[3:0]			ADDSET[3:0]			
	Reset value			0	0									1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0x11C	**FMC_BWTR4**	Res.	Res.	ACCMOD[1:0]		Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BUSTURN[3:0]				DATAST[7:0]									ADDHLD[3:0]			ADDSET[3:0]			
	Reset value			0	0									1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0x80	**FMC_PCR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ECCPS [2:0]				TAR[3:0]				TCLR[3:0]				Res.	Res.	ECCEN	PWID [1:0]		Res.	PBKEN	PWAITEN	Res.
	Reset value													0	0	0	0	0	0	0	0	0	0	0	0			0	0	1	1	0	0	
0x84	**FMC_SR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FEMPT	IFEN	ILEN	IREN	IFS	ILS	IRS
	Reset value																										1	0	0	0	0	0	0

表 180. FMC 寄存器映射（续）

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x88	**FMC_PMEM**	MEMHIZx[7:0]								MEMHOLDx[7:0]								MEMWAITx[7:0]								MEMSETx[7:0]							
	Reset value	1	1	1	1	1	1	0	0	1	1	1	1	1	1	0	0	1	1	1	1	1	1	0	0	1	1	1	1	1	1	0	0
0x8C	**FMC_PATT**	ATTHIZ[7:0]								ATTHOLD[7:0]								ATTWAIT[7:0]								ATTSET[7:0]							
	Reset value	1	1	1	1	1	1	0	0	1	1	1	1	1	1	0	0	1	1	1	1	1	1	0	0	1	1	1	1	1	1	0	0
0x94	**FMC_ECCR**	ECC[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x140	**FMC_SDCR1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RPIPE[1:0]		RBURST	SDCLK[1:0]		WP	CAS[1:0]		NB	MWID[1:0]		NR[1:0]		NC	
	Reset value																		0	0	0	1	1	0	1	0	0	1	0	0	0	0	0
0x144	**FMC_SDCR2**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RPIPE[1:0]		RBURST	SDCLK[1:0]		WP	CAS[1:0]		NB	MWID[1:0]		NR[1:0]		NC	
	Reset value																		0	0	0	1	1	0	1	0	0	1	0	0	0	0	0
0x148	**FMC_SDTR1**	Res.	Res.	Res.	Res.	TRCD[3:0]				TRP[3:0]				TWR[3:0]				TRC[3:0]				TRAS[3:0]				TXSR[3:0]				TMRD[3:0]			
	Reset value					1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0x14C	**FMC_SDTR2**	Res.	Res.	Res.	Res.	TRCD[3:0]				TRP[3:0]				TWR[3:0]				TRC[3:0]				TRAS[3:0]				TXSR[3:0]				TMRD[3:0]			
	Reset value					1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0x150	**FMC_SDCMR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MRD														NRFS[3:0]				CTB1	CTB2	MODE[2:0]		
	Reset value											0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x154	**FMC_SDRTR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REIE	COUNT[12:0]													CRE
	Reset value																		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x158	**FMC_SDSR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MODES2[1:0]		MODES1[1:0]		RE
	Reset value																												0	0	0	0	0

有关寄存器边界地址的信息，请参见*第 100 页的第 2.2.2 节*。

23 QuadSPI接口（QUADSPI）

23.1 简介

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

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

采用双闪存模式时，将同时访问两个 Quad-SPI FLASH，吞吐量和容量均可提高二倍。

23.2 QUADSPI 主要特性

- 三种功能模式：间接模式、状态轮询模式和内存映射模式
- 双闪存模式，通过并行访问两个 FLASH，可同时发送/接收 8 位数据
- 支持 SDR 和 DDR 模式
- 针对间接模式和内存映射模式，完全可编程操作码
- 针对间接模式和内存映射模式，完全可编程帧格式
- 集成 FIFO，用于发送和接收
- 允许 8、16 和 32 位数据访问
- 在达到 FIFO 阈值和传输完成时生成 MDMA 触发信号
- 在达到 FIFO 阈值、超时、操作完成以及发生访问错误时产生中断

23.3 QUADSPI 功能说明

23.3.1 QUADSPI 框图

图 116. QUADSPI 功能框图（双闪存模式禁止）

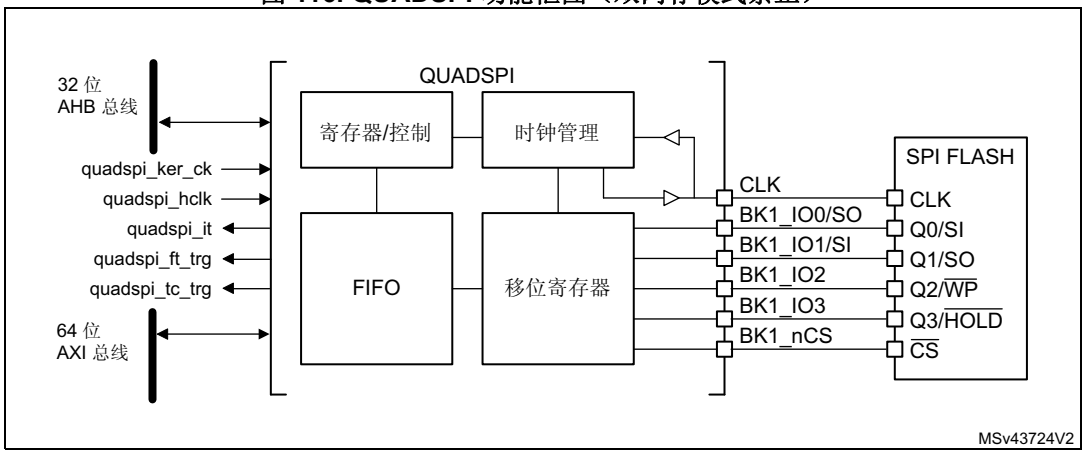
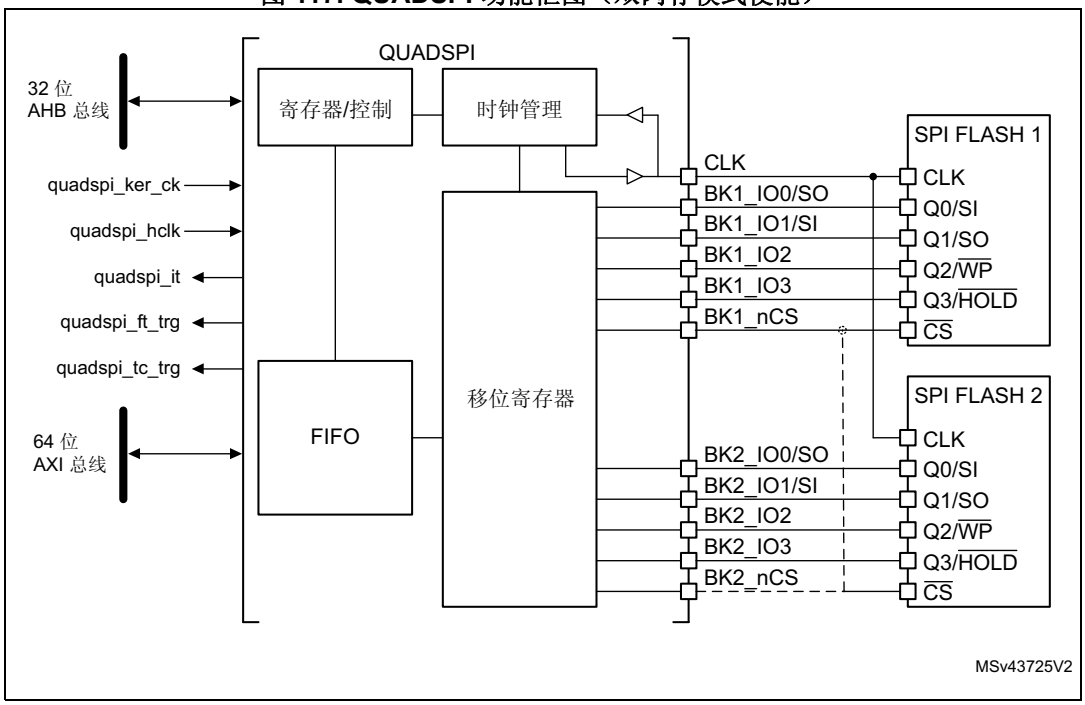


图 117. QUADSPI 功能框图（双闪存模式使能）



23.3.2 QUADSPI 引脚和内部信号

表 181 列出了 QUADSPI 内部信号。

表 181. QUADSPI 内部信号

信号名称	信号类型	说明
quadspi_ker_ck	数字输入	QUADSPI 内核时钟
quadspi_hclk	数字输入	QUADSPI 寄存器接口时钟
quadspi_it	数字输出	QUADSPI 全局中断
quadspi_ft_trg	数字输出	MDMA 的 QUADSPI FIFO 阈值触发信号
quadspi_tc_trg	数字输出	MDMA 的 QUADSPI 传输完成触发信号

表 182 列出了 QUADSPI 引脚，双闪存模式下使用 6 个信号连接单个 FLASH，使用 10 到 11 个信号连接两个 FLASH（FLASH 1 和 FLASH 2）。

表 182. QUADSPI 引脚

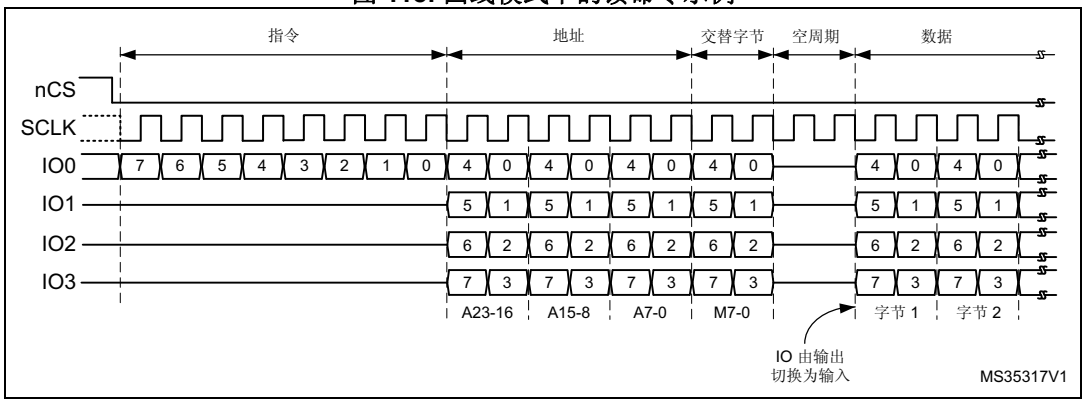
信号名称	信号类型	说明
CLK	数字输出	FLASH 1 和 FLASH 2 的时钟
BK1_IO0/SO	数字输入/输出	在双线/四线模式中为双向 IO，单线模式中为串行输出，适用于 FLASH 1
BK1_IO1/SI	数字输入/输出	在双线/四线模式中为双向 IO，单线模式中为串行输入，适用于 FLASH 1
BK1_IO2	数字输入/输出	在四线模式中为双向 IO，适用于 FLASH 1
BK1_IO3	数字输入/输出	在四线模式中为双向 IO，适用于 FLASH 1
BK2_IO0/SO	数字输入/输出	在双线/四线模式中为双向 IO，单线模式中为串行输出，适用于 FLASH 2
BK2_IO1/SI	数字输入/输出	在双线/四线模式中为双向 IO，单线模式中为串行输入，适用于 FLASH 2
BK2_IO2	数字输入/输出	在四线模式中为双向 IO，适用于 FLASH 2
BK2_IO3	数字输入/输出	在四线模式中为双向 IO，适用于 FLASH 2
BK1_nCS	数字输出	片选（低电平有效），适用于 FLASH 1。如果 QUADSPI 始终在双闪存模式下工作，则其也可用于 FLASH 2。
BK2_nCS	数字输出	片选（低电平有效），适用于 FLASH 2。如果 QUADSPI 始终在双闪存模式下工作，则其也可用于 FLASH 1。

23.3.3 QUADSPI 命令序列

QUADSPI 通过命令与 FLASH 通信 每条命令包括指令、地址、交替字节、空指令和数据这五个阶段 任一阶段均可跳过，但至少要包含指令、地址、交替字节或数据阶段之一。

nCS 在每条指令开始前下降，在每条指令完成后再次上升。

图 118. 四线模式下的读命令示例



指令阶段

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

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

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

地址阶段

在地址阶段，将 1-4 字节发送到 FLASH，指示操作地址。待发送的地址字节数在 QUADSPI_CCR[13:12] 寄存器的 ADSIZE[1:0] 字段中进行配置。在间接模式和自动轮询模式下，待发送的地址字节在 QUADSPI_AR 寄存器的 ADDRESS[31:0] 中指定。在内存映射模式下，则通过 AXI（来自于 Cortex® 或 DMA）直接给出地址。

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

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

交替字节阶段

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

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

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

交替字节阶段存在仅需发送单个半字节而不是一个全字节的情况，比如采用双线模式并且仅使用两个周期发送交替字节时。在这种情况下，固件可采用四线模式 (ABMODE = 11) 并发送一个字节，方法是 ALTERNATE 的位 7 和 3 置“1”（IO3 保持高电平）且位 6 和 2 置“0”（IO2 线保持低电平）。此时，半字节的高 2 位存放在 ALTERNATE 的位 4:3，低 2 位存放在位 1 和 0 中。例如，如果半字节 2 (0010) 通过 IO0/IO1 发送，则 ALTERNATE 应设置为 0x8A (1000_1010)。

空指令周期阶段

在空指令周期阶段，给定的 1-31 个周期内不发送或接收任何数据，目的是当采用更高的时钟频率时，给 FLASH 留出准备数据阶段的时间。这一阶段中给定的周期数在 QUADSPI_CCR[22:18] 寄存器的 DCYC[4:0] 字段中指定。在 SDR 和 DDR 模式下，持续时间被指定为一定个数的全时钟周期。

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

空指令周期阶段的操作模式由 DMODE 确定。

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

数据阶段

在数据阶段，可从 FLASH 接收或向其发送任意数量的字节。

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

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

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

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

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

23.3.4 QUADSPI 信号接口协议模式

单线 SPI 模式

传统 SPI 模式允许串行发送/接收单独的 1 位。在此模式下，数据通过 SO 信号（其 I/O 与 IO0 共享）发送到 FLASH。从 FLASH 接收到的数据通过 SI（其 I/O 与 IO1 共享）送达。

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

在每个已配置为单线模式的阶段中：

- IO0 (SO) 处于输出模式
- IO1 (SI) 处于输入模式（高阻抗）
- IO2 处于输出模式并强制置“0”（以禁止“写保护”功能）
- IO3 处于输出模式并强制置“1”（以禁止“保持”功能）

若 DMODE = 01，这对于空指令阶段也同样如此。

双线 SPI 模式

在双线模式下，通过 IO0/IO1 信号同时发送/接收两位。

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

在每个已配置为双线模式的阶段中：

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

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

四线 SPI 模式

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

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

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

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

IO2 和 IO3 仅用于 Quad SPI 模式 如果未配置任何阶段使用四线 SPI 模式，即使 QUADSPI 激活，对应 IO2 和 IO3 的引脚也可用于其他功能。

SDR 模式

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

在 SDR 模式下，当 QUADSPI 驱动 IO0/SO、IO1、IO2、IO3 信号时，这些信号仅在 CLK 的下降沿发生转变。

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

DDR 模式

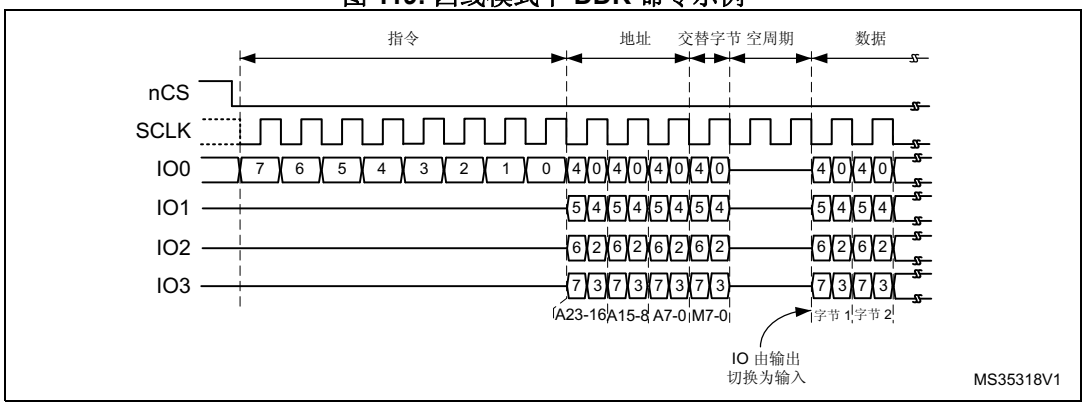
若 DDRM 位 (QUADSPI_CCR[31]) 置 1，则 QUADSPI 在双倍数据速率 (DDR) 模式下工作。

在 DDR 模式下，当 QUADSPI 在地址/交替字节/数据阶段驱动 IO0/SO、IO1、IO2、IO3 信号时，将在 CLK 的每个上升沿和下降沿发送 1 位。

指令阶段不受 DDRM 的影响。始终通过 CLK 的下降沿发送指令。

在 DDR 模式下接收数据时，QUADSPI 假定 FLASH 通过 CLK 的上升沿和下降沿均发送数据。若 DDRM = 1，固件必须清零 SSHIFT 位（QUADSPI_CR 的位 4）。因此，在半个 CLK 周期后（下一个反向边沿）对信号采样。

图 119. 四线模式下 DDR 命令示例



双闪存模式

若 DFM 位（QUADSPI_CR 的位 6）为 1，QUADSPI 处于双闪存模式。QUADSPI 使用两个外部四线 SPI FLASH（FLASH 1 和 FLASH 2），在每个周期中发送/接收 8 位（在 DDR 模式下为 16 位），能够有效地将吞吐量和容量扩大一倍。

每个 FLASH 使用同一个 CLK 并可选择使用同一个 nCS 信号，但其 IO0、IO1、IO2 和 IO3 信号是各自独立的。

双闪存模式可与单比特模式、双比特模式以及四比特模式结合使用，也可与 SDR 或 DDR 模式相结合。

FLASH 的大小在 FSIZE[4:0] (QUADSPI_DCR[20:16]) 中指定，指定的值应能够反映 FLASH 的总容量，即单个组件容量的 2 倍。

如果地址 X 为偶数，QUADSPI 赋给地址 X 的字节是存放于 FLASH 1 的地址 X/2 中的字节，QUADSPI 赋给地址 X+1 的字节是存放于 FLASH 2 的地址 X/2 中的字节。也就是说，偶地址中的字节存储于 FLASH 1，奇地址中的字节存储于 FLASH 2。

在双闪存模式下读取 FLASH 状态寄存器时，需要读取的字节数是单闪存模式下的 2 倍。这意味着在状态寄存器获取指令到达后，如果每个 FLASH 给出 8 个有效位，则 QUADSPI 必须配置为 2 个字节（16 位）的数据长度，它将从每个 FLASH 接收 1 个字节。如果每个 FLASH 给出一个 16 位的状态，则 QUADSPI 必须配置为读取 4 字节，以在双闪存模式下可获取两个 FLASH 的所有状态位 。结果（在数据寄存器中）的最低有效字节是 FLASH 1 状态寄存器的最低有效字节，而下一个字节是 FLASH 2 状态寄存器的最低有效字节。数据寄存器的第三个字节是 FLASH 1 的第二个字节，第四个字节是 FLASH 2 的第二个字节（FLASH 具有 16 位状态寄存器时）。

偶数个字节必须始终在双闪存模式下访问。因此，若 DRM = 1，则数据长度字段 (QUADSPI_DLR[0]) 的位 0 始终保持为 1。

在双闪存模式下，FLASH 1 接口信号的行为基本上与正常模式下相同。在指令、地址、交替字节以及空指令周期阶段，FLASH 2 接口信号具有与 FLASH 1 接口信号完全相同的波形。也就是说，每个 FLASH 总是接收相同的指令与地址。然后，在数据阶段，BK1_IOx 和 BK2_IOx 总线并行传输数据，但发送到 FLASH 1（或从其接收）的数据与 FLASH 2 中的不同。

23.3.5 QUADSPI 间接模式

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

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

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

读取/写入的字节数在数据长度寄存器 (QUADSPI_DLR) 中指定。如果 QUADSPI_DLR = 0xFFFF_FFFF（全为“1”），则数据长度视为未定义，QUADSPI 将继续传输数据，直到到达（由 FSIZE 定义的）FLASH 的结尾。如果不传输任何字节，DMODE (QUADSPI_CCR[25:24]) 应设置为 00。

如果 QUADSPI_DLR = 0xFFFF_FFFF 并且 FSIZE = 0x1F（最大值指示一个 4GB 的 FLASH），在此特殊情况下，传输将无限继续下去，仅在出现终止请求或 QUADSPI 被禁止后停止。在读取最后一个存储器地址后（地址为 0xFFFF_FFFF），将从地址 = 0x0000_0000 开始继续读取。

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

触发命令启动

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

1. 对 INSTRUCTION[7:0] (QUADSPI_CCR) 执行写入操作，如果没有地址是必需的（当 ADMODE = 00）并且不需要固件提供数据（当 FMODE = 01 或 DMODE = 00）
2. 对 ADDRESS[31:0] (QUADSPI_AR) 执行写入操作，如果地址是必需的（当 ADMODE = 00）并且不需要固件提供数据（当 FMODE = 01 或 DMODE = 00）
3. 对 DATA[31:0] (QUADSPI_DR) 执行写入操作，如果地址是必需的（当 ADMODE != 00）并且需要固件提供数据（当 FMODE = 00 并且 DMODE != 00）

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

如果命令启动，BUSY 位（QUADSPI_SR 的位 5）将自动置 1。

FIFO 和数据管理

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

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

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

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

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

在间接模式下，当 FIFO 已满，QUADSPI 将暂时停止从 Flash 读取字节以避免上溢。请注意，只有在 FIFO 中的 4 个字节为空 (FLEVEL ≤ 11) 时才会重新开始读取 FLASH。因此，若 FTHRES ≥ 13，应用程序必须读取足够的字节以确保 QUADSPI 再次从 FLASH 检索数据。否则，只要 11 < FLEVEL < FTHRES，FTF 标志将保持为“0”。

23.3.6 QUADSPI 状态标志轮询模式

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

对 FLASH 的访问最初与在间接读取模式下相同：如果不需要地址 (AMODE = 00)，则在写入 QUADSPI_CCR 时即开始访问。否则，如果需要地址，则在写入 QUADSPI_AR 时开始第一次访问。BUSY 在此时变为高电平，即使在周期性访问期间也保持不变。

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

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

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

如果 SMIE = 1，则在 SMF 置 1 时调用一个中断。

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

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

23.3.7 QUADSPI 内存映射模式

在配置为内存映射模式时，外部 SPI 器件被视为是内部存储器。

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

即使 FLASH 容量更大，寻址空间也无法超过 256MB。

如果访问的地址超出 FSIZE 定义的范围但仍在 256 MB 范围内，则生成总线错误。此错误的影响具体取决于尝试进行访问的总线主器件：

- 如果为 Cortex® CPU，则会在使能总线故障时发生总线故障异常，在禁止总线故障时发生硬性故障 (hard fault) 异常。
- 如果为 DMA，则生成 DMA 传输错误，并自动禁用相应的 DMA 通道。

支持字节、半字和字访问类型。

支持芯片内执行 (XIP) 操作，QUADSPI 接受下一个微控制器访问并提前加载后面地址中的字节。如果之后访问的是连续地址，由于值已经预取，访问将更快完成。

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

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

23.3.8 QUADSPI 自由运行时钟模式

当配置为自由运行时钟模式时，QUADSPI 外设不断输出时钟以进行测试和校准。

QUADSPI 通信配置寄存器 (QUADSPI_CCR) 中的自由运行时钟模式位 (FRCM) 置 1 后立即进入自由运行时钟模式。该模式的退出方式是将 QUADSPI 控制寄存器 (QUADSPI_CR) 的 ABORT 位置 1。

当 QUADSPI 以自由运行时钟模式运行时：

- 时钟连续运行，
- nCS 保持高电平（取消选择外部设备），
- 释放数据线（高阻态），
- QUADSPI 状态寄存器 (QUADSPI_SR) 的 BUSY 标志置 1。

23.3.9 QUADSPI FLASH 配置

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

FSIZE[4:0] 字段使用下面的公式定义外部存储器的大小：

FLASH 中的字节数 = $2^{[FSIZE+1]}$

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

如果 DFM = 1，FSIZE 表示两个 FLASH 容量的总和。

QUADSPI 连续执行两条命令时，它在两条命令之间将片选信号 (nCS) 置为高电平默认仅一个 CLK 周期时长。如果 FLASH 需要命令之间的时间更长，可使用片选高电平时间 (CSHT) 字段指定 nCS 必须保持高电平的最少 CLK 周期数（最大为 8）。

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

23.3.10 QUADSPI 延迟数据采样

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

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

DDR 模式下不支持时钟移位：若 DDRM 位置 1，SSHIFT 位必须清零。

23.3.11 QUADSPI 配置

QUADSPI 配置分两个阶段

- QUADSPI IP 配置
- QUADSPI FLASH 配置

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

QUADSPI IP 配置。

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

DDR 模式可通过 DDRM 位进行设置。使能该模式后，在每个时钟的上升沿和下降沿都会发送地址和交替字节以及发送/接收数据。无论 DDRM 位如何设置，都将在 SDR 模式下发送指令。

生成 MDMA 触发信号或生成中断的 FIFO 电平在 FTHRES 位中进行编程。

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

双闪存模式可通过将 DFM 置 1 来激活。

QUADSPI FLASH 配置

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

23.3.12 QUADSPI 的用法

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

间接模式的操作步骤

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

QUADSPI 用于间接模式时，采用以下方式构建帧：

1. 在 QUADSPI_DLR 中指定待读取或写入的字节数
2. 在 QUADSPI_CCR 中指定帧格式、模式和指令代码
3. 在 QUADSPI_ABR 中指定要在地址阶段后立即发送的可选交替字节
4. 在 QUADSPI_CR 中指定工作模式。
5. 在 QUADSPI_AR 中指定目标地址。
6. 通过 QUADSPI_DR 从 FIFO 读取数据/向 FIFO 写入数据

在写入控制寄存器 (QUADSPI_CR) 时，用户可指定以下设置：

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

在写入通信配置寄存器 (QUADSPI_CCR) 时，用户指定以下参数：

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

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

在需要地址（ADMODE 不为 00），但无需写入数据寄存器（FMODE = 01 或 DMODE = 00）时，通过写入 QUADSPI_AR 更新地址后，命令序列便立即启动。

在数据传输（FMODE = 00 并且 DMODE ! = 00）中，通过 QUADSPI_DR 写入 FIFO 触发通信启动。

状态标志轮询模式

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

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

在 QUADSPI_PISR 寄存器中指定周期性。

在检索到状态数据后，可在内部进行处理，以达到以下目的：

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

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

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

在任何情况下，最新的检索值都在 QUADSPI_DR 中可用。

内存映射模式

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

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

当主器件访问存储器映射空间时,将发送已编程的指令和帧。

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

数据长度寄存器 (QUADSPI_DLR) 在内存映射模式中无意义。

23.3.13 指令仅发送一次

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

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

在 IMODE = 00（无指令）时，SIOO 不起作用。

23.3.14 QUADSPI 差错管理

在以下情况下可能产生错误：

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

23.3.15 QUADSPI 的繁忙位和中止功能

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

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

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

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

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

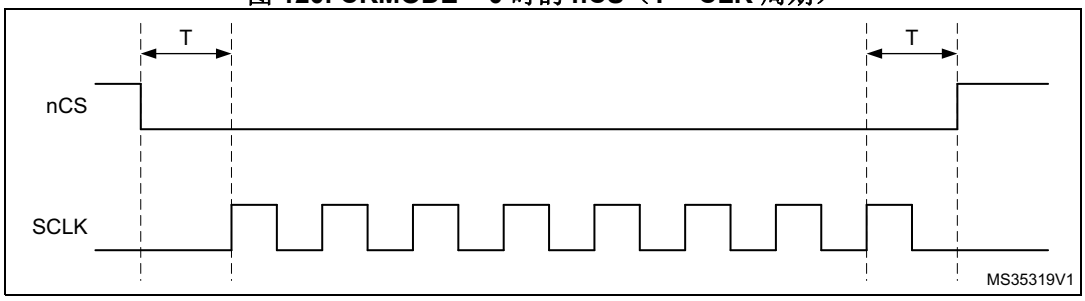
注： *如果中止对状态寄存器的写入操作，有些 FLASH 可能发生错误行为。*

23.3.16 nCS 行为

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

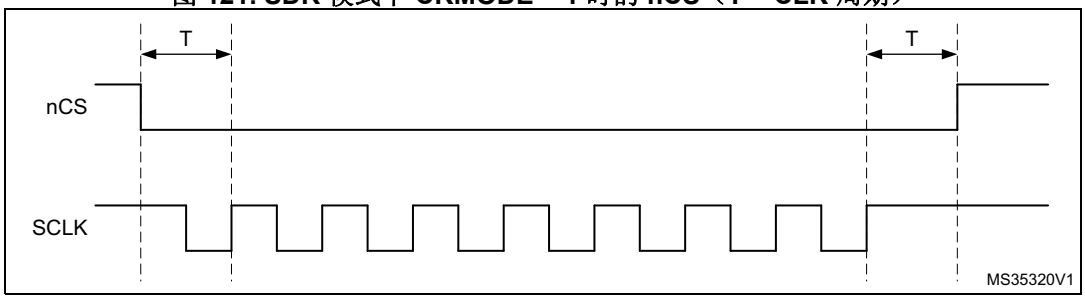
当 CKMODE = 0（“模式 0”，在未进行任何操作时 CLK 保持低电平）时，nCS 在操作首次升高 CLK 边沿时的一个 CLK 周期前降至低电平，在操作最后一次升高 CLK 边沿时的一个 CLK 周期后升至高电平，如*图 120* 所示。

图 120. CKMODE = 0 时的 nCS（T = CLK 周期）



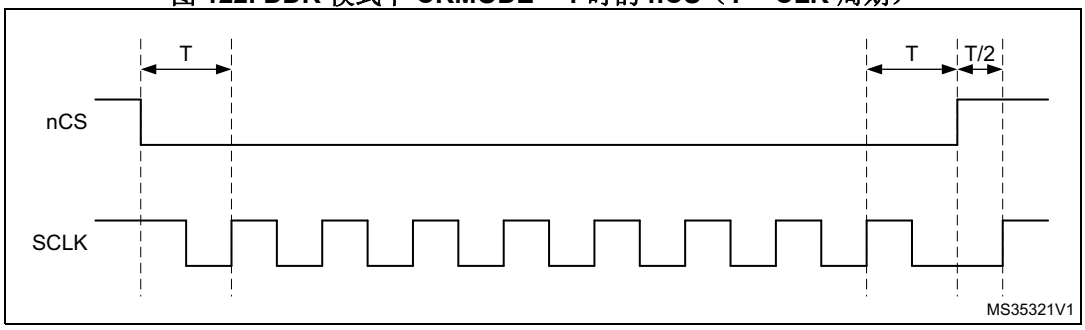
当 CKMODE = 1（“模式 3”，在未进行任何操作时 CLK 升至高电平）且 DDRM = 0（SDR 模式）时，nCS 仍在操作首次升高 CLK 边沿时的一个 CLK 周期前降至低电平，在操作最后一次升高 CLK 边沿时的一个 CLK 周期后升至高电平，如*图 121* 所示。

图 121. SDR 模式下 CKMODE = 1 时的 nCS（T = CLK 周期）



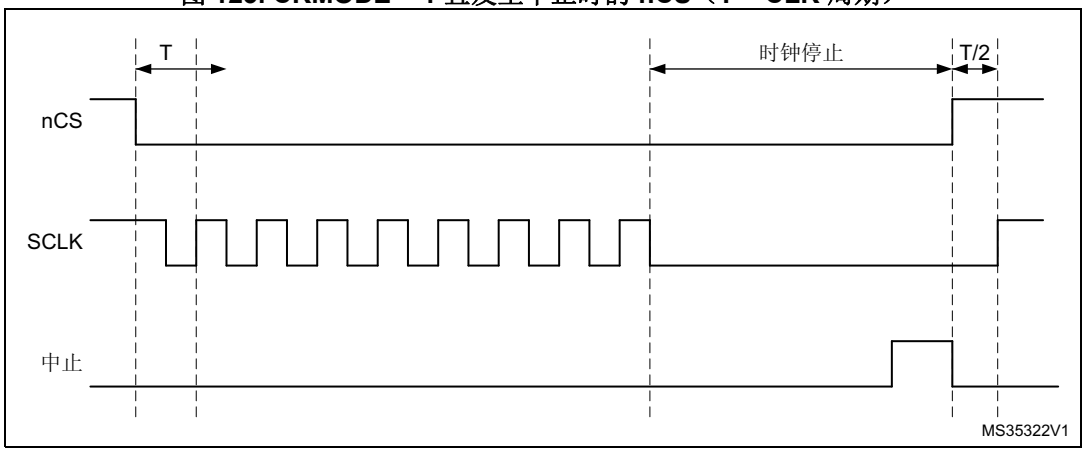
当 CKMODE = 1（模式 3）且 DDRM = 1（DDR 模式）时，nCS 在操作首次升高 CLK 边沿时的一个 CLK 周期前降至低电平，在操作最后一次升高 CLK 边沿时的一个 CLK 周期后升至高电平，如*图 122* 所示。由于 DDR 操作必须伴随下降沿完成，当 nCS 变为高电平时，CLK 为低电平并在经过半个周期后恢复高电平。

图 122. DDR 模式下 CKMODE = 1 时的 nCS（T = CLK 周期）



若 FIFO 在读取操作中保持写满状态或在写入操作中保持为空，则在固件干预 FIFO 前，操作停止并且 CLK 保持低电平。若操作停止时发生中止，则 nCS 在请求中止后即升至高电平，CLK 则在半个周期后升至高电平，如*图 123* 所示。

图 123. CKMODE = 1 且发生中止时的 nCS（T = CLK 周期）



不处于双闪存模式 (DFM = 0) 时，仅访问 FLASH 1，因此 BK2_nCS 保持高电平。在双闪存模式下，BK2_nCS 与 BK1_nCS 的行为完全相同。因此，如果存在 FLASH 2 并且应用程序始终处于双闪存模式，则 FLASH 2 可使用 BK1_nCS，而 BK2_nCS 引脚输出可用于其他功能。

23.4 QUADSPI 中断

发生如下事件时可生成中断：

- 超时
- 状态匹配
- FIFO 阈值
- 传输完成
- 传输错误

可以使用单独的中断使能位以提高灵活性。

表 183. QUADSPI 中断请求

中断事件	事件标志	使能控制位
超时	TOF	TOIE
状态匹配	SMF	SMIE
FIFO 阈值	FTF	FTIE
传输完成	TCF	TCIE
传输错误	TEF	TEIE

23.5 QUADSPI 寄存器

23.5.1 QUADSPI 控制寄存器 (QUADSPI_CR)

QUADSPI control register

偏移地址：0x0000

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PRESCALER								PMM	APMS	Res.	TOIE	SMIE	FTIE	TCIE	TEIE
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw		rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	FTHRES					FSEL	DFM	Res.	SSHIFT	TCEN	Res.	ABORT	EN
			rw	rw	rw	rw	rw	rw	rw		rw	rw		rw	w1s

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

该字段定义基于 quadspi_ker_ck 时钟生成 CLK 所用的分频系数（值 + 1）。

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

1：$F_{CLK} = F_{quadspi_ker_ck}/2$

2：$F_{CLK} = F_{quadspi_ker_ck}/3$

...

255：$F_{CLK} = F_{quadspi_ker_ck}/256$

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

仅可在 BUSY = 0 时修改该字段。

位 23 **PMM**：轮询匹配模式 (Polling match mode)

该位指示在自动轮询模式期间用来确定是否“匹配”的方法。

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

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

仅可在 BUSY = 0 时修改该位。

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

该位确定在匹配后自动轮询是否停止。

0：仅通过中止或禁用 QUADSPI 停止自动轮询模式。

1：发生匹配时，自动轮询模式停止。

仅可在 BUSY = 0 时修改该位。

位 21 保留，必须保持复位值。

位 20 **TOIE**：TimeOut 中断使能 (TimeOut interrupt enable)

该位使能 TimeOut 中断。

0：禁止中断

1：使能中断

位 19 **SMIE**：状态匹配中断使能 (Status match interrupt enable)

该位使能状态匹配中断。

0：禁止中断

1：使能中断

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

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

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

位 15:13 保留，必须保持复位值。

位 12:8 **FTHRES[4:0]**：FIFO 阈值级别 (FIFO threshold level)
定义在间接模式下 FIFO 中将导致 FIFO 阈值标志（FTF，QUADSPI_SR[2]）置 1 的字节数阈值。
在间接写入模式下 (FMODE = 00)：
0：如果 FIFO 中存在 1 个或更多空闲字节可供写入，则 FTF 置 1
1：如果 FIFO 中存在 2 个或更多空闲字节可供写入，则 FTF 置 1
...
31：如果 FIFO 中存在 32 个空闲字节可供写入，则 FTF 置 1
在间接读取模式下 (FMODE = 01)：
0：如果 FIFO 中存在 1 个或更多有效字节可供读取，则 FTF 置 1
1：如果 FIFO 中存在 2 个或更多有效字节可供读取，则 FTF 置 1
...
31：如果 FIFO 中存在 32 个有效字节可供读取，则 FTF 置 1

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

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

位 5 保留，必须保持复位值。

位 4 **SSHIFT**：采样移位 (Sample shift)

默认情况下，QUADSPI 在 FLASH 驱动数据后过半个 CLK 周期开始采集数据。使用该位，可考虑外部信号延迟，推迟数据采集。

0：不发生移位

1：移位半个周期

在 DDR 模式下 (DDRM = 1)，固件必须确保 SSHIFT = 0。

仅可在 BUSY = 0 时修改该字段。

位 3 **TCEN**：超时计数器使能 (Timeout counter enable)

该位仅在内存映射模式 (FMODE = 11) 下有效。激活该位后，如果在一段时间（通过 TIMEOUT[15:0] (QUADSPI_LPTR) 定义）内一直没有进行访问，将释放片选 (nCS)（从而降低功耗）。

使能超时计数器。

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

0：禁止超时计数器，在内存映射模式中进行访问后，片选 (nCS) 保持激活。

1：使能超时计数器，在内存映射模式下，FLASH 持续不活动 TIMEOUT[15:0] 个周期后释放片选 (nCS)。

仅可在 BUSY = 0 时修改该位。

位 2 保留

位 1 **ABORT**：中止请求 (Abort request)

该位中止执行中的命令序列。在中止完成时自动复位。

该位可停止当前的传输。

在轮询模式或内存映射模式下，该位也用以复位 APM 位或 DM 位。

0：不请求中止

1：请求中止

位 0 **EN**：使能 (Enable)

使能 QUADSPI。

0：禁止 QUADSPI

1：使能 QUADSPI

23.5.2 QUADSPI 器件配置寄存器 (QUADSPI_DCR)

QUADSPI device configuration register

偏移地址：0x0004

复位值：0x0000 0000

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

位 31:21 保留，必须保持复位值。

位 20:16 **FSIZE[4:0]**：FLASH 大小 (FLASH memory size)

该字段使用下面的公式定义了外部存储器的大小：

FLASH 中的字节数 = $2^{[FSIZE+1]}$

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

如果 DFM = 1，FSIZE 表示两个 FLASH 容量的总和。

仅可在 BUSY = 0 时修改该字段。

位 15:11 保留，必须保持复位值。

位 10:8 **CSHT[2:0]**：片选高电平时间 (Chip select high time)

CSHT+1 定义片选 (nCS) 在发送至 FLASH 的命令之间必须保持高电平的最少 CLK 周期数。

0：nCS 在 FLASH 命令之间保持高电平至少 1 个周期

1：nCS 在 FLASH 命令之间保持高电平至少 2 个周期

...

7：nCS 在 FLASH 命令之间保持高电平至少 8 个周期

仅可在 BUSY = 0 时修改该字段。

位 7:1 保留，必须保持复位值。

位 0 **CKMODE**：模式 0/模式 3 (Mode 0 / mode 3)

该位指示 CLK 在命令之间（nCS = 1 时）的电平。

0：nCS 为高电平（片选释放）时，CLK 必须保持低电平。这称为模式 0。

1：nCS 为高电平（片选释放）时，CLK 必须保持高电平。这称为模式 3。

仅可在 BUSY = 0 时修改该字段。

23.5.3 QUADSPI 状态寄存器 (QUADSPI_SR)

QUADSPI status register

偏移地址：0x0008

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	FLEVEL[5:0]						Res.	Res.	BUSY	TOF	SMF	FTF	TCF	TEF
		r	r	r	r	r	r			r	r	r	r	r	r

位 31:14 保留，必须保持复位值。

位 13:8 **FLEVEL[5:0]**：FIFO 级别 (FIFO level)

该字段给出 FIFO 中的有效字节数。FIFO 为空时 FLEVEL = 0，写满时 FLEVEL = 32。在内存映射模式和自动状态轮询模式下，FLEVEL 为零。

位 7:6 保留，必须保持复位值。

位 5 **BUSY**：忙 (Busy)

操作进行时，该位置 1。在对 FLASH 的操作完成并且 FIFO 为空时，该位自动清零。

位 4 **TOF**：超时标志 (Timeout flag)

发生超时时该位置 1。向 CTOF 写入 1 可将该位清零。

位 3 **SMF**：状态匹配标志 (Status match flag)

在自动轮询模式下，若未屏蔽的接收数据与匹配寄存器 (QUADSPI_PSMAR) 中的对应位相匹配，则该位置 1。向 CSMF 写入 1 可将该位清零。

位 2 **FTF**：FIFO 阈值标志 (FIFO threshold flag)

在间接模式下，若达到 FIFO 阈值，或从 FLASH 读取完成后，FIFO 中留有数据时，该位置 1。只要阈值条件不再为“真”，该位就自动清零。

在自动轮询模式下，每次读取状态寄存器时，该位即置 1；读取数据寄存器时，该位清零。

位 1 **TCF**：传输完成标志 (Transfer complete flag)

在间接模式下，当传输的数据数量达到编程设定值，或在任何模式下传输中止时，该位置 1。向 CTCF 写入 1 时，该位清零。

位 0 **TEF**：传输错误标志 (Transfer error flag)

在间接模式下访问无效地址时，该位置 1。向 CTEF 写入 1 可将该位清零。

23.5.4 QUADSPI 标志清零寄存器 (QUADSPI_FCR)

QUADSPI flag clear register

偏移地址：0x000C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CTOF	CSMF	Res.	CTCF	CTEF
											w1o	w1o		w1o	w1o

位 31:5 保留，必须保持复位值。

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

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

位 2 保留，必须保持复位值。

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

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

23.5.5 QUADSPI 数据长度寄存器 (QUADSPI_DLR)

QUADSPI data length register

偏移地址：0x0010

复位值：0x0000 0000

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

位 31:0 **DL[31:0]**：数据长度 (Data length)

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

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

0x0000_0000：传输 1 个字节

0x0000_0001：传输 2 个字节

0x0000_0002：传输 3 个字节

0x0000_0003：传输 4 个字节

...

0xFFFF_FFFD：传输 4,294,967,294 (4G-2) 个字节

0xFFFF_FFFE：传输 4,294,967,295 (4G-1) 个字节

0xFFFF_FFFF：未定义长度 -- 传输所有字节直到到达由 FSIZE 定义的 FLASH 的结尾。如果 FSIZE = 0x1F，则读取无限继续下去。

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

该字段在内存映射模式 (FMODE = 10) 下不起作用

仅可在 BUSY = 0 时写入该字段。

23.5.6 QUADSPI 通信配置寄存器 (QUADSPI_CCR)

QUADSPI communication configuration register

偏移地址：0x0014

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DDRM	DHHC	FRCM	SIOO	FMODE[1:0]		DMODE		Res.	DCYC[4:0]					ABSIZE	
rw	rw	rw	rw	rw	rw	rw	rw		rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ABMODE		ADSIZE		ADMODE		IMODE		INSTRUCTION[7:0]							
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

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

该位为地址、交替字节和数据阶段设置 DDR 模式：

0：禁止 DDR 模式

1：使能 DDR 模式

仅可在 BUSY = 0 时写入该字段。

位 30 **DHHC**：DDR 保持 (DDR hold)

在 DDR 模式下，延迟 1/4 个 QUADSPI 输出时钟周期再输出数据：

0：使用模拟延迟来延迟数据输出。

1：延迟 1/4 个 QUADSPI 输出时钟周期再输出数据。

该特性仅在 DDR 模式下激活。

仅可在 BUSY = 0 时写入该字段。

位 29 **FRCM**：自由运行时钟模式 (Free Running Clock Mode)

该位置 1 时，无论 FMODE 位如何，QUADSPI 外设均进入自由运行时钟模式。

0：正常模式

1：自由运行时钟模式

仅可在 BUSY = 0 时写入该位。

位 28 **SIOO**：仅发送指令一次模式 (Send instruction only once mode)
请参见*第 780 页的第 23.3.13 节：指令仅发送一次*。IMODE = 00 时，该位不起作用。
0：在每个事务中发送指令
1：仅为第一条命令发送指令
仅可在 BUSY = 0 时写入该字段。

位27:26 **FMODE[1:0]**：功能模式 (Functional mode)
该字段定义 QUADSPI 操作的功能模式：
00：间接写入模式
01：间接读取模式
10：自动轮询模式
11：内存映射模式
仅可在 BUSY = 0 时写入该字段。

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

位 23 保留，必须保持复位值。

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

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

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

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

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

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

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

23.5.7 QUADSPI 地址寄存器 (QUADSPI_AR)

QUADSPI address register

偏移地址：0x0018

复位值：0x0000 0000

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

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

23.5.8 QUADSPI 交替字节寄存器 (QUADSPI_ABR)

QUADSPI alternate bytes registers

偏移地址：0x001C

复位值：0x0000 0000

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

位 31:0 **ALTERNATE[31:0]**：交替字节 (Alternate Bytes)

指定要在地址后立即发送到外部 SPI 设备的可选数据。

仅可在 BUSY = 0 时写入该字段。

23.5.9 QUADSPI 数据寄存器 (QUADSPI_DR)

QUADSPI data register

偏移地址：0x0020

复位值：0x0000 0000

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

位 31:0 **DATA[31:0]**：数据 (Data)

指定要与外部 SPI 设备交换的数据。

在间接写入模式下，写入该寄存器的数据在数据阶段发送到 FLASH，在此之前则存储于 FIFO。如果 FIFO 太满，将暂停写入，直到 FIFO 具有足够的空间接受要写入的数据才继续。

在间接模式下，读取该寄存器可获得（通过 FIFO）已从 FLASH 接收的数据。如果 FIFO 所含字节数比读取操作要求的字节数少并且 BUSY=1，将暂停读取，直到足够的数据出现或传输完成（不分先后）才继续。

在自动轮询模式下，该寄存器包含最后从 FLASH 读取的数据（未进行屏蔽）。

支持对该寄存器进行字、半字以及字节访问。在间接写入模式下，字节写入将在 FIFO 中增加 1 个字节，半字写入增加 2 个，而字写入则增加 4 个。类似地，在间接读取模式下，字节读取将擦除 FIFO 中的 1 个字节，半字读取擦除 2 个，而字读取则擦除 4 个。间接模式下的访问必须与此寄存器的最低位对齐：字节读取必须读取 DATA[7:0] 而半字读取必须读取 DATA[15:0]。

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

QUADSPI polling status mask register

偏移地址：0x0024

复位值：0x0000 0000

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

位 31:0 **MASK[31:0]**：状态屏蔽 (Status mask)

对在轮询模式下接收的状态字节进行屏蔽

对于位 n：

0：屏蔽在自动轮询模式下所接收数据的位 n，在匹配逻辑中不考虑其值

1：不屏蔽在自动轮询模式下所接收数据的位 n，在匹配逻辑中考虑其值

仅可在 BUSY = 0 时写入该字段。

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

QUADSPI polling status match register

偏移地址：0x0028

复位值：0x0000 0000

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

位 31:0 **MATCH[31:0]**：状态匹配 (Status match)

该值将与屏蔽状态寄存器比较以进行匹配

仅可在 BUSY = 0 时写入该字段。

23.5.12 QUADSPI 轮询间隔寄存器 (QUADSPI _PIR)

QUADSPI polling interval register

偏移地址：0x002C

复位值：0x0000 0000

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

位 31:16 保留，必须保持复位值。

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

23.5.13 QUADSPI 低功耗超时寄存器 (QUADSPI_LPTR)

QUADSPI low-power timeout register

偏移地址：0x0030

复位值：0x0000 0000

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

位 31:16 保留，必须保持复位值。

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

23.5.14 QUADSPI 寄存器映射

表 184. QUADSPI 寄存器映射和复位值

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x0000	**QUADSPI_CR**	PRESCALER[7:0]								PMM	APMS	Res.	TOIE	SMIE	FTIE	TCIE	TEIE	Res.	Res.	Res.	FTHRES [4:0]					FSEL	DFM	Res.	SSHIFT	TCEN	Res.	ABORT	EN
	Reset value	0	0	0	0	0	0	0	0	0	0		0	0	0	0	0				0	0	0	0	0	0	0		0	0		0	0
0x0004	**QUADSPI_DCR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FSIZE[4:0]					Res.	Res.	Res.	Res.	Res.	CSHT			Res.	Res.	Res.	Res.	Res.	Res.	Res.	CKMODE
	Reset value												0	0	0	0	0						0	0	0								0
0x0008	**QUADSPI_SR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FLEVEL[6:0]							Res.	Res.	BUSY	TOF	SMF	FTF	TCF	TEF
	Reset value																		0	0	0	0	0	0	0			0	0	0	0	0	0
0x000C	**QUADSPI_FCR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CTOF	CSMF	Res.	CTCF	CTEF
	Reset value																												0	0		0	0
0x0010	**QUADSPI_DLR**	DL[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0014	**QUADSPI_CCR**	DDRM	DHHC	FRCM	SIOO	FMODE[1:0]		DMODE[1:0]		Res.	DCYC[4:0]					ABSIZE[1:0]		ABMODE[1:0]		ADSIZE[1:0]		ADMODE[1:0]		IMODE[1:0]		INSTRUCTION[7:0]							
	Reset value	0	0	0	0	0	0	0	0		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0018	**QUADSPI_AR**	ADDRESS[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x001C	**QUADSPI_ABR**	ALTERNATE[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0020	**QUADSPI_DR**	DATA[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0024	**QUADSPI_ PSMKR**	MASK[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0028	**QUADSPI_ PSMAR**	MATCH[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x002C	**QUADSPI_PIR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	INTERVAL[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0030	**QUADSPI_ LPTR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TIMEOUT[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

有关寄存器边界地址的信息，请参见*第 2.2.2 节*。

24 延迟模块 (DLYB)

24.1 简介

延迟模块 (DLYB) 用于生成输出时钟，其与输入时钟存在相位偏移。首先必须通过用户应用程序对输出时钟的相位进行编程。然后，输出时钟用于对由另一个外设 （例如 SDMMC 或 QUADSPI 接口）接收的数据进行计时。

该延迟与电压和温度相关，可能需要重新配置应用程序和重新确定输出时钟与接收数据之间的相位关系。

24.2 DLYB主要特性

延迟模块具有以下特性：

- 输入时钟频率范围为 25 MHz 到 208 MHz
- 多达 12 个过采样相位

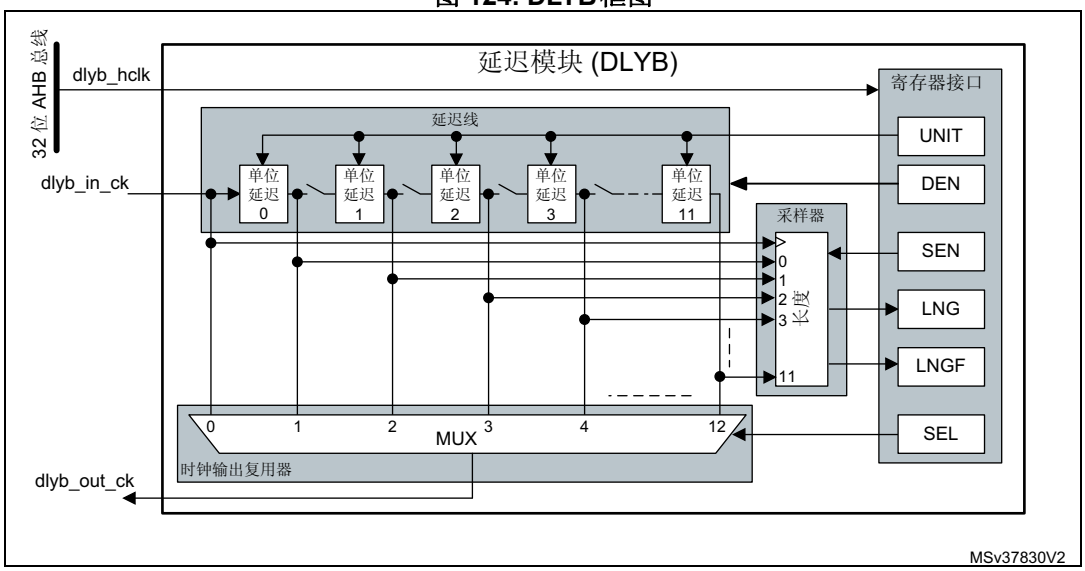
24.3 DLYB 功能说明

24.3.1 DLYB 框图

延迟模块包括以下子模块：

- 寄存器接口模块，用来提供对延迟模块寄存器的 AHB 访问
- 延迟线，用来支持单位延迟
- 延迟线长度采样
- 输出时钟选择复用器

图 124. DLYB框图



24.3.2 DLYB 引脚和内部信号

表 185 列出了 DLYB 内部信号。

表 185. DLYB 内部输入/输出信号

信号名称	信号类型	说明
dlyb_hclk	数字输入	延迟模块寄存器接口时钟
dlyb_in_ck	数字输入	延迟模块输入时钟
dlyb_out_ck	数字输出	延迟模块输出时钟

24.3.3 概述

可通过将 DLYB_CR 寄存器 DEN 位置 1 来使能延迟模块（请参见*第 24.4.1 节：DLYB 控制寄存器 (DLYB_CR)*）。可通过 DLYB_CR 寄存器 SEN 位来使能长度采样器。

当使能延迟模块时，可通过 DLYB_CFGR 寄存器 UNIT 位来定义将延迟增加一个单位延迟（请参见*第 24.4.2 节：DLYB 配置寄存器 (DLYB_CFGR)*）。请注意，只有在禁止输出时钟 (SEN = 1) 时，才能对 UNIT 位进行编程。

当使能延迟模块时，可通过 DLYB_CFGR 寄存器 SEL 位来选择输出时钟相位。请注意，只有在禁止输出时钟 (SEN = 1) 时，才能对 SEL 位进行编程。

在对输出时钟进行移相之前，应将延迟线长度配置为一个输入时钟周期。通过 SEN 位可以使能长度采样器，长度采样器可以访问 DLYB_CFGR 寄存器延迟线长度（LNG 位）和长度有效标志（LNGF），因此可以通过使能长度采样器来配置延迟线长度。

一旦完成延迟线长度配置，就可以通过输出时钟复用器选择经过移相的输出时钟。此操作通过 SEL 位完成。当 SEN 设置为 0 时，只有选定相位的输出时钟可供使用。

*表 186*对延迟模块控制进行了总结。

表 186. 延迟模块控制

DEN	SEN	单位	SEL	LNG	LNGF	输出时钟
0	0	无关	无关	无关	无关	使能（= 输入时钟）
x	1	单位延迟	输出时钟相位	长度	长度标志	禁止
1	0	单位延迟(1)	输出时钟相位(2)	无关	无关	使能 （= 选定相位的输出时钟）

1. 只有当 SEN = 1 时，才可以更改单位延迟。
2. 只有当 SEN = 1 时，才可以更改输出时钟相位。

24.3.4 延迟线长度配置程序

LNG 位用于确定延迟线长度，其与输入时钟周期相关。配置的延迟线长度应当覆盖一个完整的输入时钟周期。

要将延迟线长度配置为一个输入时钟周期，请按照以下顺序进行：

1. 通过将 DEN 位设置为 1 来使能延迟模块。
2. 通过将 SEN 位设置为 1 来使能长度采样。

3. 通过将 SEL 位设置为 12 来使能所有延迟单元。
4. 对于 UNIT = 0 到 127（必须重复此步骤，直到配置了延迟线长度）：
   a) 更新 UNIT 值并等待长度标志 LNGF 设置为 1。
   b) 读取 LNG 位。
   如果 (LNG[10:0] > 0) 且（LNG[11] 或 LNG[10] = 0），则将延迟线长度配置为一个输入时钟周期。
5. 确定需要多少个单位延迟 (N) 才能涵盖一个输入时钟周期。
   – 对于 N = 10 至 0：
   如果 LNG [N] = 1，则涵盖输入时钟周期的单位延迟数量 = N。
6. 通过将 SEN 清零可禁止长度采样。

24.3.5 输出时钟相位配置程序

当将延迟线长度配置为一个输入时钟周期时，可以在涵盖一个输入时钟周期的单位延迟之间选择输出时钟相位。

请按照以下步骤选择输出时钟相位：

1. 通过将 SEN 位设置为 1 来禁止输出时钟并使能访问相位选择 SEL 位。
2. 使用期望的输出时钟相位值对 SEL 位进行编程。
3. 通过将 SEN 清零来使能所选相位的输出时钟。

24.4 DLYB 寄存器

可以对所有寄存器进行字、半字和字节访问。

24.4.1 DLYB 控制寄存器 (DLYB_CR)

DLYB control register

偏移地址：0x000

复位值：0x0000 0000

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

位 31:2 保留，必须保持复位值

位 1 **SEN**：采样器长度使能位 (Sampler length enable bit)
0：禁止采样器长度和寄存器访问 UNIT 和 SEL，使能输出时钟。
1：使能采样器长度和寄存器访问 UNIT 和 SEL，禁止输出时钟。

位 0 **DEN**：延迟模块使能位 (Delay block enable bit)
0：禁止延迟模块。
1：使能延迟模块。

24.4.2 DLYB 配置寄存器 (DLYB_CFGR)

DLYB configuration register

偏移地址：0x004

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LNGF	Res.	Res.	Res.	LNG											
r				r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	UNIT							Res.	Res.	Res.	Res.	SEL			
	rw	rw	rw	rw	rw							rw	rw	rw	rw

位 31 **LNGF**：长度有效标志 (Length valid flag)
该标志指示在 UNIT 位更改之后，LNG 位中包含的延迟线长度值是否有效。
0：LNG 中的长度值无效。
1：LNG 中的长度值有效。

位 30:28 保留，必须保持复位值

位 27:16 **LNG**：延迟线长度值 (Delay line length value)
这些位反映了在输入时钟的上升沿采样的 12 个单位延迟值。
此值仅在 LNGF = 1 时有效。

位 15 保留，必须保持复位值

位 14:8 **UNIT**：定义一个单位延迟单元的延迟 (Delay Defines the delay of a Unit delay cell)
这些位只能在 SEN = 1 时写入。
单位延迟 = 初始延迟 + UNIT x 延迟步长。

位 7:4 保留，必须保持复位值

位 3:0 **SEL**：选择输出时钟的相位 (Select the phase for the Output clock)
这些位只能在 SEN = 1 时写入。
输出时钟相位 = 输入时钟 + SEL x 单位延迟。

24.4.3 DLYB 寄存器映射

表 187. DLYB 寄存器映射和复位值

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x000	**DLYB_CR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SEN	DEN
	Reset value																															0	0
0x004	**DLYB_CFGR**	LNGF	Res.	Res.	Res.	LNG												Res.	UNIT							Res.	Res.	Res.	Res.	SEL			
	Reset value	0				0	0	0	0	0	0	0	0	0	0	0	0		0	0	0	0	0	0	0					0	0	0	0

有关寄存器边界地址的信息，请参见*第 100 页的第 2.2.2 节*。

25 模数转换器 (ADC)

25.1 简介

本部分介绍多达 3 个 ADC 的实现

- ADC1 与 ADC2 紧密耦合，可在双重模式下运行（ADC1 为主器件）。
- ADC3 单独进行实例化。

每个 ADC 由 16 位逐次逼近模数转换器组成。

每个 ADC 的复用通道多达 20 条。各种不同通道的 A/D 转换可在单次、连续、扫描或不连续采样模式下进行。ADC 的结果存储在一个左对齐或右对齐的 32 位数据寄存器中。

ADC 映射到 AHB 总线，从而可实现快速数据处理。

ADC 具有模拟看门狗特性，允许应用检测输入电压是否超过了用户自定义的阈值上限或下限。

内置硬件过采样器，可提高模拟性能，同时还能减轻 CPU 进行相关计算的负担。

采用了高效的低功耗模式，在低频下可实现极低的功耗。

25.2 ADC 主要特性

- 高性能特性
  - 多达 2 个 ADC，可在双重模式下运行
  - 可配置 16 位、14 位、12 位、10 位或 8 位分辨率
  - ADC 转换时间与 AHB 总线时钟频率无关
  - 可通过降低分辨率来缩短转换时间
  - 可管理单端输入或差分输入（可按通道进行编程）
  - AHB 从总线接口，可实现快速数据处理
  - 自校准（偏移校准和线性度校准）
  - 可独立设置各通道采样时间
  - 多达四条注入通道（对常规通道或注入通道的模拟输入分配完全可配置）
  - 硬件辅助准备注入通道的上下文，从而实现快速上下文切换
  - 数据对齐以保持内置数据一致性
  - 数据可由 GP-DMA 管理，基于 FIFO 实现常规通道转换
  - 数据可连接到 DFSDM 进行后期处理
  - 4 个专用数据寄存器供注入通道使用
- 过采样器
  - 32 位数据寄存器
  - 过采样率可在 2x 到 1024x 之间进行调整
  - 可编程数据右移和左移

- 低功耗特性
  - 速度自适应低功耗模式，可降低 ADC 在低频下工作时的功耗
  - 可在实现慢速总线频率应用的同时保持最佳 ADC 性能
  - 提供自动控制，可避免 ADC 在低 AHB 总线时钟频率应用中溢出（自动延迟模式）
- 每个 ADC 都有外部模拟输入通道
  - 专用 GPIO 焊盘有多达 6 条快速通道
  - 专用 GPIO 焊盘有多达 14 条慢速通道
- 此外，还有五条内部专用通道
  - 内部参考电压 (V_{REFINT})，连接到 ADC3
  - 内部温度传感器 (V_{SENSE})，连接到 ADC3
  - V_{BAT} 监测通道 ($V_{BAT}/4$)，连接到 ADC3
  - 内部 DAC 通道 1 和通道 2，连接到 ADC2
- 可通过以下方式启动转换过程：
  - 通过软件启动常规转换和注入转换
  - 通过极性可配置的硬件触发器（内部定时器事件或 GPIO 输入事件）启动常规转换和注入转换
- 转换模式
  - 每个 ADC 均可转换单条通道，也可扫描一系列通道
  - 单次模式会在每次触发时对选定的输入执行一次转换
  - 连续模式可连续转换选定的输入
  - 不连续采样模式
- ADC1 和 ADC2 的双重 ADC 模式
- 在 ADC 准备就绪、采样结束、转换结束（常规转换或注入转换）、序列转换结束（常规转换或注入转换）、模拟看门狗 1/2/3 事件或溢出事件时生成中断
- 每个 ADC 有 3 个模拟看门狗
- ADC 输入范围：$V_{REF-} \leq V_{IN} \leq V_{REF+}$

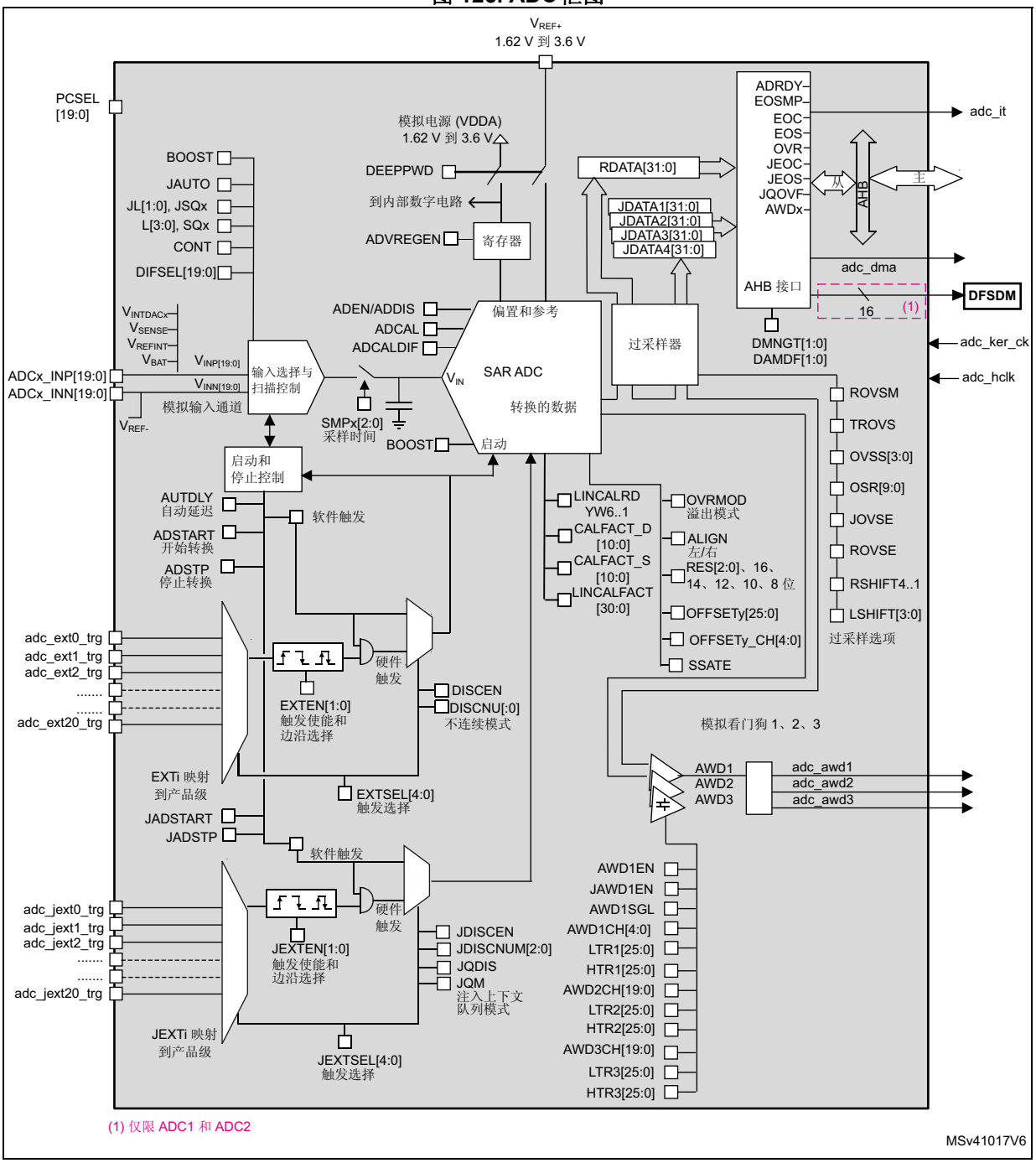
图 125 给出了一个 ADC 的框图。

25.3 ADC 功能说明

25.3.1 ADC框图

图 125 给出了 ADC 框图，*表 189* 列出了 ADC 的引脚说明。

图 125. ADC框图



25.3.2 ADC 引脚和内部信号

表 188. ADC 内部输入/输出信号

内部信号名称	信号类型	说明
adc_ext_trg[20:0]	输入	共有多达 21 个外部触发输入用于常规转换（可连接至片上定时器）。 这些输入由主 ADC 和从 ADC 共享。
adc_jext_trg[20:0]	输入	共有多达 21 个外部触发输入用于注入转换（可连接至片上定时器）。 这些输入由主 ADC 和从 ADC 共享。
adc_awd1 adc_awd2 adc_awd3	输出	内部模拟看门狗输出信号，连接至片上定时器。（x = 模拟看门狗编号 1、2、3）
V_{SENSE}	模拟输入	内部温度传感器的输出电压
V_{REFINT}	模拟输入	内部参考电压的输出电压
V_{BAT}	模拟输入	外部电池电压
adc_it	输出	ADC 中断
adc_hclk	输入	AHB时钟
adc_ker_ck	输入	ADC 内核时钟
adc_dma	输出	ADC DMA 请求

表 189. ADC 输入/输出引脚

名称	信号类型	注释
V_{REF+}	正模拟参考电压输入	ADC 高/正参考电压， 1.62 V ≤ V_{REF+} ≤ V_{DDA}
V_{DDA}	模拟电源输入	模拟电源等于 V_{DDA}： 1.62 V ≤ V_{DDA} ≤ 3.6 V
V_{REF-}	负模拟参考电压输入	ADC 低/负参考电压， V_{REF-} = V_{SSA}
V_{SSA}	模拟电源地输入	模拟电源地电压等于 V_{SS}
V_{INP}[19:0]	每个 ADC 的正输入模拟通道	连接到外部通道：ADC_INP*i* 或内部通道。
V_{INN}[19:0]	每个 ADC 的负输入模拟通道	连接到 V_{REF-} 或外部通道：ADC_INN*i*
ADC_INP[19:0]	外部模拟输入信号	多达 20 个模拟输入通道： – ADC_INP[0:5] 快速通道 – ADC_INP[6:19] 慢速通道
ADC_INN[19:0]		多达 20 个模拟输入通道： – ADC_INN[0:5] 快速通道 – ADC_INN[6:19] 慢速通道
PCSEL[19:0]	输出，通道预选控制信号	连接到 GPIO 预先选择通道

25.3.3 时钟

双时钟域架构

双时钟域架构意味着 ADC 时钟独立于 AHB 总线时钟。

三个 ADC 的输入时钟相同，可从两个不同的时钟源中选择（请参见*图 126：ADC 时钟方案*）：

a) ADC 时钟可以是名为 adc_ker_ck 的特定时钟源，该时钟源独立于 APB 时钟，并与 AHB 时钟异步。

该时钟可在 RCC 中配置（有关如何生成 ADC 时钟 (adc_ker_ck) 专用时钟的更多信息，请参见 RCC 部分）。

要选择此时钟方案，必须将 ADCx_CCR 寄存器的 CKMODE[1:0] 位复位。

b) ADC 时钟可由 ADC 总线接口的 AHB 时钟除以一个可编程的因数（1、2 或 4）来提供。在这种模式下，可选择可编程的分频系数（根据位 CKMODE[3:0]，选择 1、2 或 4）。

要选择此时钟方案，ADCx_CCR 寄存器的 CKMODE[1:0] 位不得为“00”。

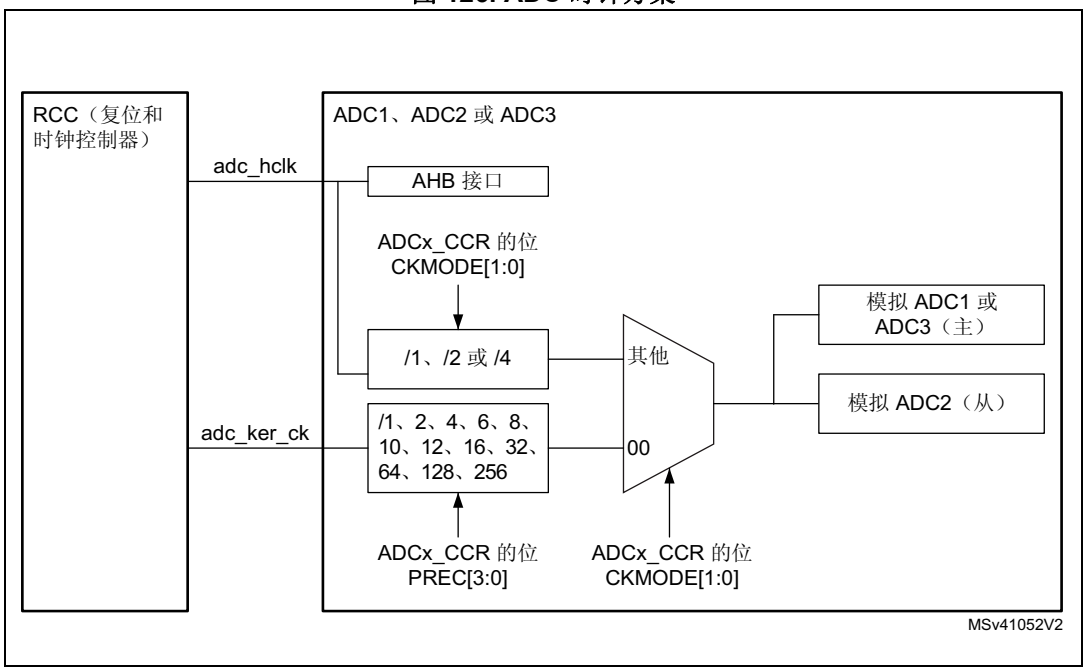
注： *对于选项 b)，仅当 AHB 预分频器设为 1（RCC_CFGR 寄存器中的 HPRE[3:0] = 0xxx）时，才能使用预分频系数 1 (CKMODE[1:0]=01)。*

选项 a) 的优点在于无论选择哪种 AHB 时钟方案，都可以达到最大 ADC 时钟频率。使用通过 ADCx_CCR 寄存器中的 PRESC[3:0] 位配置的预分频器时，ADC 时钟最后会除以以下比率：1、2、4、6、8、10、12、16、32、64、128、256。

选项 b) 的优势在于绕过了时钟域重新同步。如果 ADC 由定时器触发，并且应用要求 ADC 精确触发（不存在任何不确定性），可使用此选项（否则，重新同步两个时钟域会为触发时刻带来不确定性）。

通过 CKMODE[1:0] 位配置的时钟必须符合产品数据手册中指定的工作频率。

图 126. ADC 时钟方案



1. 请参见 RCC 部分了解 adc_hclk 和 adc_ker_ck 的生成方式。

ADC 时钟与 AHB 时钟的时钟比例限制

通常来讲，ADC 时钟与 AHB 时钟之比没有限制，但一些注入通道已编程的情况除外，此时，ADC 时钟与 AHB 时钟之比必须遵循以下规定：

- 如果所有通道的分辨率均为 16 位、14 位、12 位或 10 位，则 $F_{HCLK} >= F_{ADC} / 4$
- 如果一些通道的分辨率为 8 位（且所有通道的分辨率均不低于 8 位），则 $F_{HCLK} >= F_{ADC} / 3$

BOOST 位控制

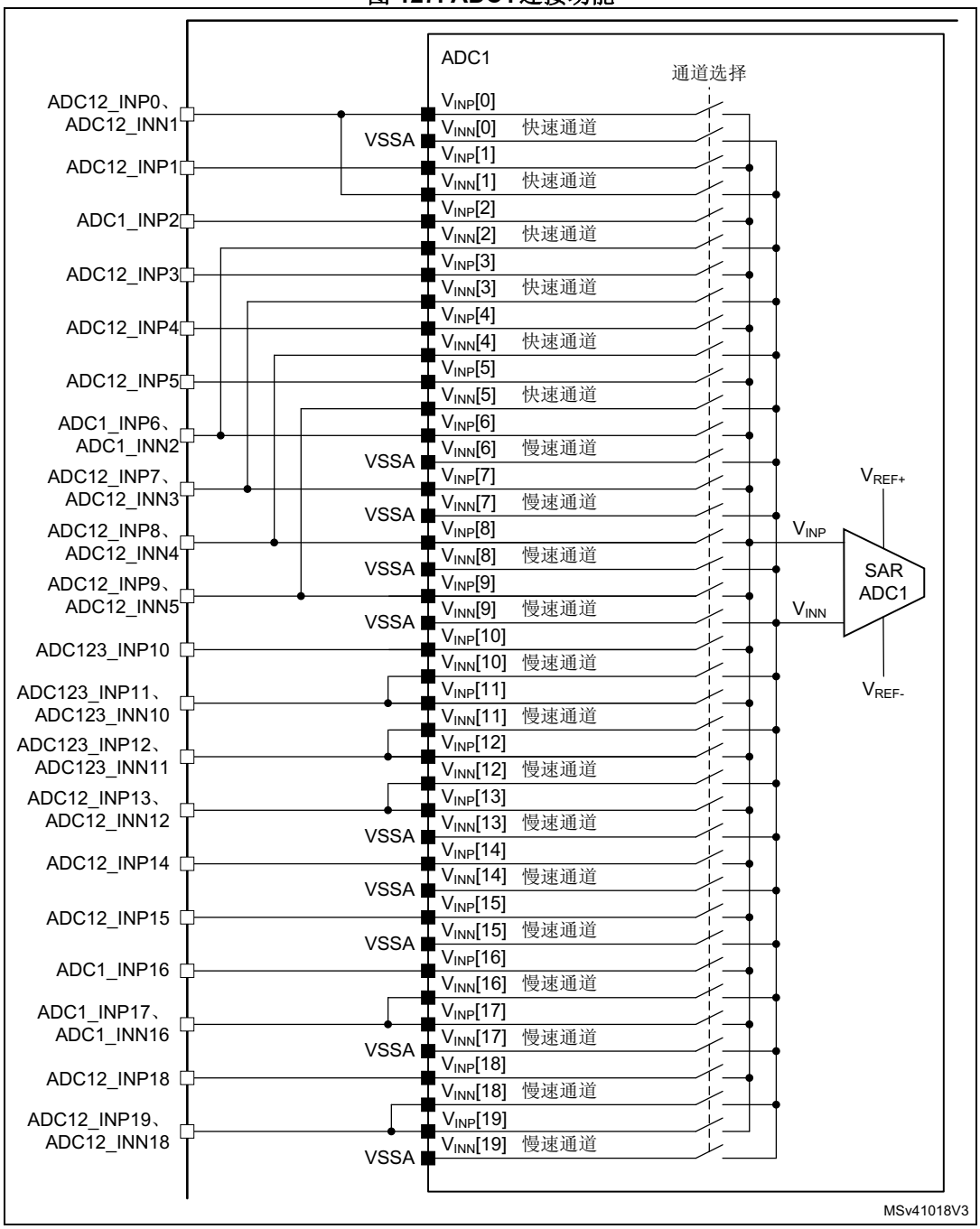
ADCx_CR 寄存器中有一个 ADC 升压控制位 BOOST。

如果 ADC 时钟大于 20 MHz，此位必须置 1。如果 ADC 时钟小于 20 MHz，此位可清零，以达到节能目的。

25.3.4 ADC1/2/3 连接

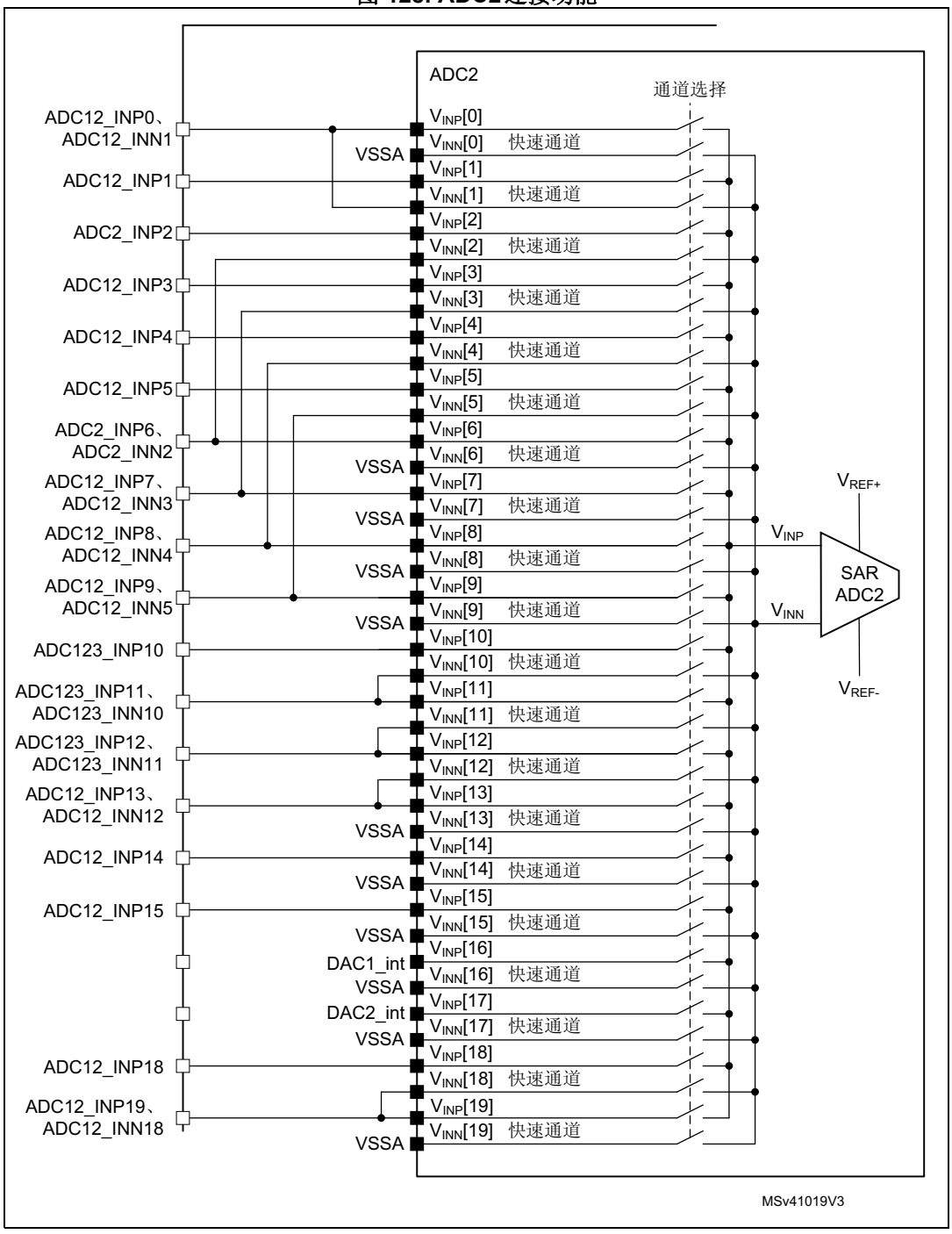
ADC1 和 ADC2 紧密耦合并共用某些外部通道，如下图所示。ADC3 单独进行实例化，但会与 ADC1 和 ADC2 共享一些输入。

图 127. ADC1连接功能



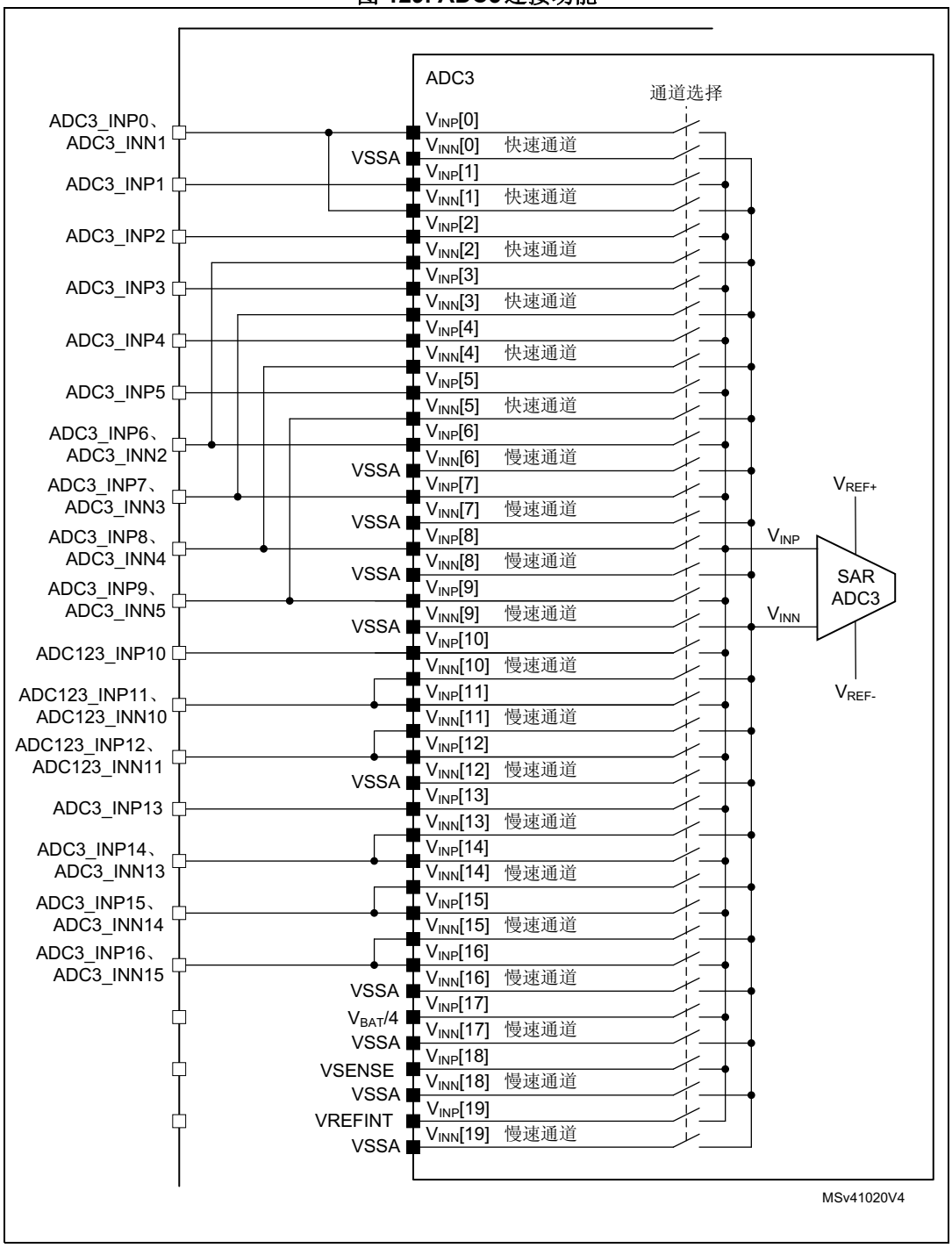
1. ADCx_INNy 信号只能在相应的 ADC 输入通道配置为差分模式时使用。

图 128. ADC2连接功能



1. ADCx_INNy 信号只能在相应的 ADC 输入通道配置为差分模式时使用。

图 129. ADC3连接功能



1. ADCx_INNy 信号只能在相应的 ADC 输入通道配置为差分模式时使用。

25.3.5 从设备 AHB 接口

ADC 利用 AHB 从设备端口实现控制/状态寄存器访问和数据访问。AHB 接口的特性如下：

- 字（32 位）访问
- 单周期响应
- 以零等待状态响应对寄存器的所有读/写访问

AHB 从设备接口不支持分离/重试请求，且绝不会生成 AHB 错误。

25.3.6 ADC 深度掉电模式 (DEEPPWD) 和 ADC 稳压器 (ADVREGEN)

ADC 默认处于深度掉电模式，在该模式下，其电源会在内部切断以降低泄漏电流（ADCx_CR 寄存器中的 DEEPPWD 位的复位状态为 1）。

要开始 ADC 操作，需要先将 DEEPPWD 位清零，使 ADC 退出深度掉电模式。

然后，必须将 ADCx_CR 寄存器中的 ADVREGEN 位置 1，以使能 ADC 内部稳压器。软件必须在 ADC 稳压器的启动时间 ($T_{ADCVREG_STUP}$) 内等待，之后才能开始校准或使能 ADC。该延迟必须通过软件实现。

有关 ADC 稳压器的启动时间，请参见器件数据手册中的 $T_{ADCVREG_STUP}$ 参数。

ADC 操作完成后，可禁止 ADC (ADEN=0)。此外，还可通过禁止 ADC 稳压器实现节能。这可通过向 ADVREGEN 位写入 0 来实现。

之后，为了通过降低泄漏电流的方式进一步实现节能，还可以将 ADCx_CR 寄存器中的 DEEPPWD 位置 1，以使 ADC 再次进入深度掉电模式。进入 STOP 模式之前，这一点特别值得关注。

注： *写入 DEEPPWD=1 会自动禁止 ADC 稳压器，并且 ADVREGEN 位会自动清零。*

注： *如果内部稳压器已禁止 (ADVREGEN=0)，则会保持内部模拟校准。*

在 ADC 深度掉电模式下 (DEEPPWD=1)，内部模拟校准会丢失，需要重新启动校准或重新应用之前保存的校准系数（请参见*第 25.3.8 节：校准（ADCAL、ADCALDIF、ADCALLIN、ADCx_CALFACT）*）。

25.3.7 单端通道和差分输入通道

可通过写入 ADCx_DIFSELDIFSEL 寄存器中的 [19:0] 位将通道配置为单端输入或差分输入。必须在禁止 ADC (ADEN=0) 时写入此配置。

在单端输入模式下，要为通道“i”转换的模拟电压是外部电压 $V_{INP[i]}$（正输入）与 V_{REF-}（负输入）之差。

在差分输入模式下，要为通道“i”转换的模拟电压是外部电压 $V_{INP[i]}$（正输入）与 $V_{INN[i]}$（负输入）之差。

差分模式的输出数据是无符号数据。当 $V_{INP[i]}$ 为 VREF-、$V_{INN[i]}$ 为 VREF+ 时，输出数据为 0x0000（16 位分辨率模式）；当 $V_{INP[i]}$ 为 VREF+、$V_{INN[i]}$ 为 VREF- 时，输出数据为 0xFFFF。

$$\text{转换后的值} = \frac{ADC_Full_Scale}{2} \times \left[1 + \frac{V_{INP} - V_{INN}}{V_{REF+}}\right]$$

当 ADC 配置为差分模式时，两路输入的偏置电压均应为 V_{REF+}/2。

输入信号应为差分信号（共模电压应固定）。

有关各 ADC 输入通道连接方式的完整说明，请参见*图 127：ADC1 连接功能*到*图 129：ADC3 连接功能*。

注意： 将通道“i”配置为差分输入模式时，其负输入电压连接至 $V_{INN[i]}$。因此，连接至 $V_{INN[i]}$ 的通道“*i+n*”不应同时由其他 ADC 进行转换。ADC1 和 ADC2 会共享某些通道，这样会使通道在其他 ADC 上不可用。

25.3.8 校准（ADCAL、ADCALDIF、ADCALLIN、ADCx_CALFACT）

每个 ADC 均提供有自动校准过程，用于驱动包括 ADC 上电/掉电序列在内的所有校准序列。在校准过程中，ADC 会计算校准系数（11 位偏移值或 160 位线性度值），并会在下一次掉电之前在自身内部应用该值。在校准过程中，应用不得使用 ADC，必须等待至校准完成。

执行任何 ADC 操作之前都必须进行校准。校准可消除各芯片间的系统误差，并可补偿偏移和线性度偏差。

单端输入转换与差分输入转换应用的偏移校准系数有所不同：

- 启动要为单端输入转换应用的校准之前，写入 ADCALDIF=0。
- 启动要为差分输入转换应用的校准之前，写入 ADCALDIF=1。

无论采用单端配置还是差分配置，都只能进行一次线性度校准。

- 发起将同时运行偏移校准和线性度校准的校准之前，写入 ADCALLIN=1。
- 发起只运行偏移校准而不运行线性度校准的校准之前，写入 ADCALLIN=0。

随后，通过软件将 ADCAL 位置 1 发起校准。仅当 ADC 禁止 (ADEN=0) 后，才能发起校准。ADCAL 位在所有校准序列过程中保持为 1。校准完成后，此位会立即由硬件清零。此时，相关校准系数会存储在模拟 ADC 内部，同时还会存储在 ADCx_CALFACT 寄存器的 CALFACT_S[10:0] 或 CALFACT_D[10:0] 位中（具体取决于是单端输入校准还是差分输入校准）。可通过将 ADCx_CALFACT2 寄存器的 ADEN 位置 1 的方式访问 16 位线性度校准系数。

如果禁止 ADC (ADEN=0)，则会保留内部模拟校准。但如果长时间禁止 ADC，建议在重新使能 ADC 之前运行新的校准周期。

每次 ADC 掉电时，内部模拟校准都会丢失（例如，产品进入待机模式或 VBAT 模式时）。这种情况下，为了避免浪费时间重新校准 ADC，可将校准系数重新写入 ADCx_CALFACT 和 ADCx_CALFACT2 寄存器，但前提是软件之前保存了上次校准时得出的校准系数。

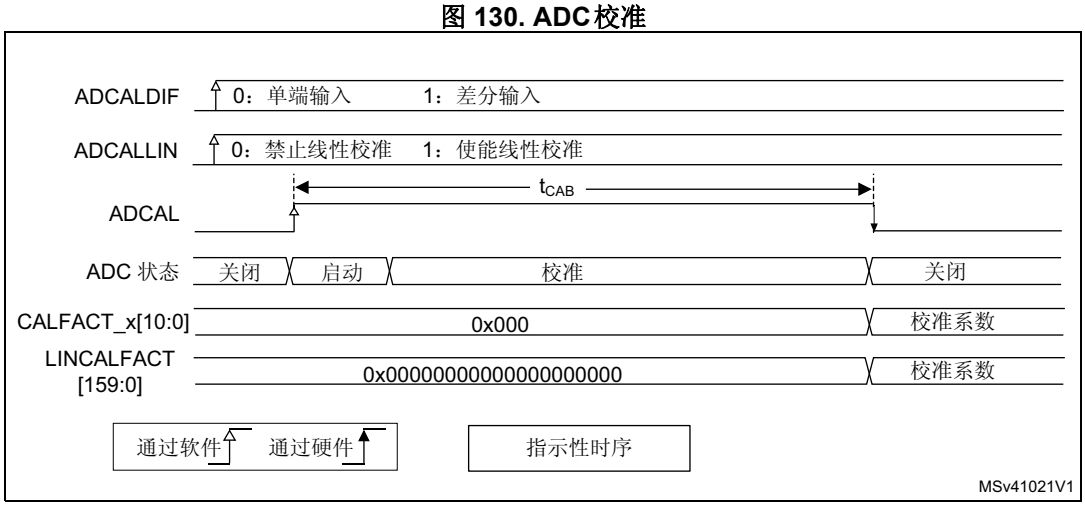
如果 ADC 已使能但未进行转换（ADEN=1、ADSTART=0 且 JADSTART=0），可写入校准系数。随后，下次转换启动时，校准系数会自动添加到模拟 ADC 中。这一载入过程是透明的，不会对转换的启动造成延迟。建议在 V_{REF+} 电压变化超过 10% 时重新进行校准。

线性校准需要 131072 个 ADC 时钟周期，偏移校准需要 520 个 ADC 时钟周期。

ADC 校准的软件流程

1. 确保 DEEPPWD=0、ADVREGEN=1，并检查 ADC 稳压器启动时间是否已过。
2. 确保 ADEN=0。
3. 设置 ADCALDIF=0（单端输入）或 ADCALDIF=1（差分输入），选择此校准的输入模式。设置 ADCALLIN=1（使能）或 ADCALLIN=0（禁止），以选择使能还是禁止线性度校准。
4. 将 ADCAL 置 1。
5. 等待 ADCAL=0。
6. 可从 ADCx_CALFACT 寄存器读取偏移校准系数。
7. 可从 ADCx_CALFACT2 寄存器读取线性度校准系数，具体流程请参见*线性度校准读取流程*一节（必须先将 ADEN 置 1 才能访问 ADCx_CALFACT2 寄存器）。

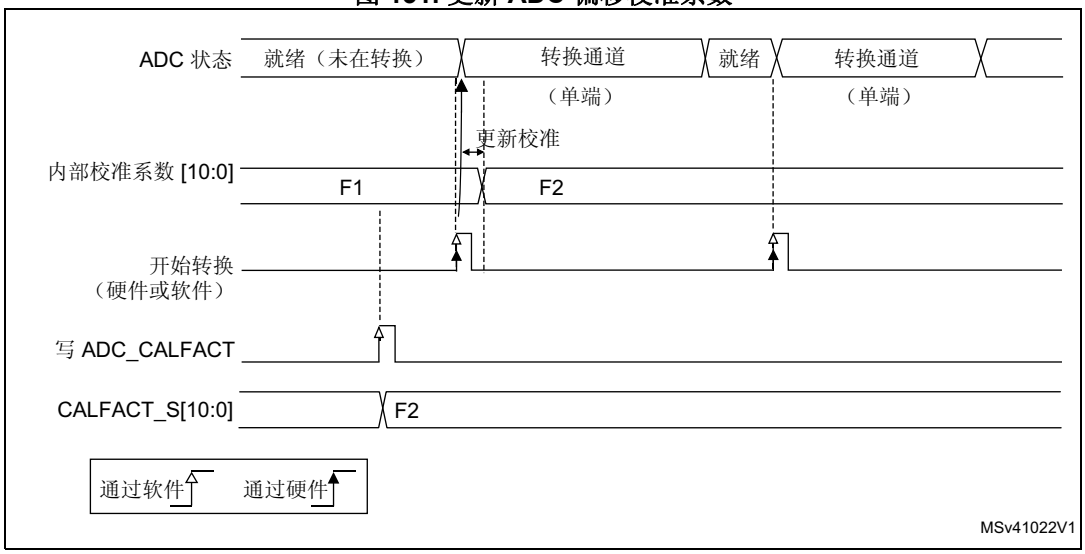
图 130. ADC校准



将校准系数重新注入 ADC 的软件流程

1. 确保 ADEN=1、ADSTART=0 且 JADSTART=0（ADC 已使能，且未进行任何转换）。
2. 将新的偏移校准系数写入 CALFACT_S 和 CALFACT_D。
3. 将新的线性度校准系数写入 LINCALFACT 位，具体流程请参见*线性度校准写入流程*一节。
4. 启动转换时，仅当内部模拟校准系数与 CALFACT_S 位（单端输入通道）或 CALFACT_D 位（差分输入通道）中存储的值不同时，校准系数才会注入到模拟 ADC。

图 131. 更新 ADC 偏移校准系数

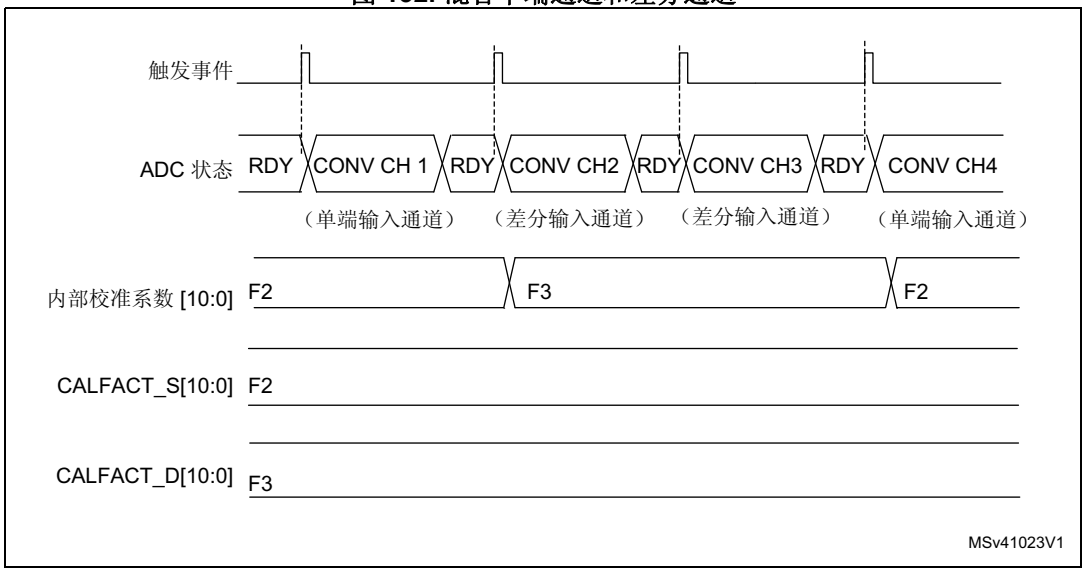


单个 ADC 转换单端模拟输入和差分模拟输入

如果 ADC 既要转换差分输入，也要转换单端输入，则必须执行两次校准，两次校准分别在 ADCALDIF=0 和 ADCALDIF=1 的情况下执行。操作流程如下：

1. 禁止 ADC。
2. 在单端输入模式 (ADCALDIF=0) 和线性度校准使能 (ADCALLIN=1) 条件下校准 ADC。此操作会更新寄存器 CALFACT_S[10:0] 和 LINCALFACT[159:0]。
3. 在差分输入模式 (ADCALDIF=1) 和线性度校准禁止 (ADCALLIN=0) 条件下校准 ADC。此操作会更新寄存器 CALFACT_D[10:0]。
4. 使能 ADC、配置通道并启动转换。每次从单端输入通道切换为差分输入通道（或进行相反切换）时，校准都将自动注入到模拟 ADC 中。

图 132. 混合单端通道和差分通道



线性度校准读取流程

完成 ADCALLIN=1 条件下的校准后（ADCAL 位由硬件清零），可使用 ADCx_CALFACT2 30 位寄存器来读取 160 位线性度校准系数（需要进行 6 次读访问）。

校准完成后，ADCx_CR 中的六个控制/状态位 LINCALRDYW1..6 会置 1。如果 ADEN 置 1，将上述其中一位清零会将线性度系数部分传送到 ADCx_CALFACT2 寄存器的 LINCALFACT[29:0] 中。当 ADCx_CALFACT2 寄存器可读取时，此位将由硬件复位（软件必须轮询此位，直至此位清零）。整个流程如下：

1. 确保 DEEPPWD=0、ADVREGEN=1，并确保 ADC 稳压器启动时间已过。
2. 将 ADEN 置 1 并等待，直至 ADRDY = 1。
3. 将 LINCALRDYW6 位（线性度校准就绪字 6）清零。
4. 轮询 LINCALRDYW6 位，直至返回值为零，指示线性度校准位 [159:150] 在 ADCx_CALFACT2[29:0] 中可用。
5. 读取 ADCx_CALFACT2[29:0]。
6. 将 LINCALRDYW5 位清零。
7. 轮询 LINCALRDYW5 位，直至返回值为零，指示线性度校准位 [149:120] 在 ADCx_CALFACT2[29:0] 中可用。
8. 读取 ADCx_CALFACT2[29:0]。
9. 将 LINCALRDYW4 位清零。
10. 轮询 LINCALRDYW4 位，直至返回值为零，指示线性度校准位 [119:90] 在 ADCx_CALFACT2[29:0] 中可用。
11. 读取 ADCx_CALFACT2[29:0]。
12. 将 LINCALRDYW3 位清零。
13. 轮询 LINCALRDYW3 位，直至返回值为零，指示线性度校准位 [89:60] 在 ADCx_CALFACT2[29:0] 中可用。
14. 读取 ADCx_CALFACT2[29:0]。
15. 将 LINCALRDYW2 位清零。
16. 轮询 LINCALRDYW2 位，直至返回值为零，指示线性度校准位 [59:30] 在 ADCx_CALFACT2[29:0] 中可用。
17. 读取 ADCx_CALFACT2[29:0]。
18. 将 LINCALRDYW1 位清零。
19. 轮询 LINCALRDYW1 位，直至返回值为零，指示线性度校准位 [29:0] 在 ADCx_CALFACT2[29:0] 中可用。
20. 读取 ADCx_CALFACT2[29:0]。

注： *软件一次只能翻转一个 LINCALRDYWx 位（其他位保持不变），否则会导致操作异常。*

仅当 ADEN=1、ADSTART=0 且 JADSTART=0（ADC 已使能，当前未进行任何转换）时，才能通过软件对 LINCALRDYW1..6 位执行写操作来访问线性度校准系数。

线性度校准写入流程

当校准尚未完成或者新线性度校准系数已重写时，ADCx_CR 中的六个控制/状态位 LINCALRDYW1..6 会复位。可以直接强制设置校准系数，也可以执行以下流程重新注入校准系数：

1. 确保 DEEPPWD=0、ADVREGEN=1，并确保 ADC 稳压器启动时间已过。
2. 将 ADEN 置 1 并等待，直至 ADRDY = 1。
3. 将之前保存的线性度校准系数位 [159:150] 写入 ADCx_CALFACT2[9:0]。
4. 将 LINCALRDYW6 位置 1。
5. 轮询 LINCALRDYW6 位，直至返回值为 1，指示线性度校准位 [159:150] 已有效写入。
6. 将之前保存的线性度校准系数位 [149:120] 写入 ADCx_CALFACT2[29:0]。
7. 将 LINCALRDYW5 位置 1。
8. 轮询 LINCALRDYW5 位，直至返回值为 1，指示线性度校准位 [149:120] 已有效写入。
9. 将之前保存的线性度校准系数位 [119:90] 写入 ADCx_CALFACT2[29:0]。
10. 将 LINCALRDYW4 位置 1。
11. 轮询 LINCALRDYW4 位，直至返回值为 1，指示线性度校准位 [119:90] 已有效写入。
12. 将之前保存的线性度校准系数位 [89:60] 写入 ADCx_CALFACT2[29:0]。
13. 将 LINCALRDYW3 位置 1。
14. 轮询 LINCALRDYW3 位，直至返回值为 1，指示线性度校准位 [89:60] 已有效写入。
15. 将之前保存的线性度校准系数位 [59:30] 写入 ADCx_CALFACT2[29:0]。
16. 将 LINCALRDYW2 位置 1。
17. 轮询 LINCALRDYW2 位，直至返回值为 1，指示线性度校准位 [59:30] 已有效写入。
18. 将之前保存的线性度校准系数位 [29:0] 写入 ADCx_CALFACT2[29:0]。
19. 将 LINCALRDYW1 位置 1。
20. 轮询 LINCALRDYW1 位，直至返回值为 1，指示线性度校准位 [29:0] 已有效写入。

注： *软件一次只能翻转一个 LINCALRDYWx 位（其他位保持不变），否则会导致操作异常。*

仅当 ADEN=1、ADSTART=0 且 JADSTART=0（ADC 已使能，当前未进行任何转换）时，才允许通过软件对 LINCALRDYW1..6 位执行写操作来更新线性度校准系数。

25.3.9 ADC 开关控制（ADEN、ADDIS、ADRDY）

首先按照*第 25.3.6 节：ADC 深度掉电模式 (DEEPPWD) 和 ADC 稳压器 (ADVREGEN)*中的流程执行操作。

DEEPPWD=0 且 ADVREGEN=1 后，可使能 ADC，并且 ADC 在开始精确转换之前需要 t_{STAB} 的稳定时间，如*图 133* 所示。以下两个控制位可使能或禁止 ADC：

- 将 ADEN 置 1 可使能 ADC。ADC 准备就绪后，ADRDY 标志会立即置 1。
- 将 ADDIS 置 1 可禁止 ADC。随后，模拟 ADC 被完全禁止后，ADEN 位和 ADDIS 位会自动由硬件清零。

随后可通过将 ADSTART 置 1（请参见*第 25.3.19 节：外部触发转换和触发极性（EXTSEL、EXTEN、JEXTSEL、JEXTEN）*）开始进行常规转换，如果触发器已使能，也可在发生外部触发事件时开始进行常规转换。

可通过将 JADSTART 置 1 开始进行注入转换，如果注入触发器已使能，也可在发生外部注入触发事件时开始进行注入转换。

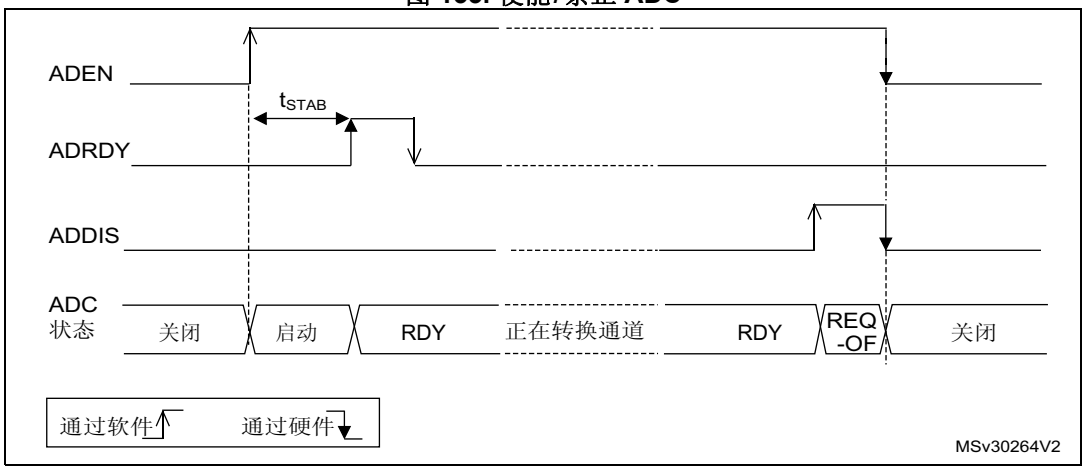
通过软件使能 ADC 的流程

1. 向 ADCx_ISR 寄存器中的 ADRDY 位写入“1”，将其清零。
2. 将 ADEN 置 1。
3. 等待，直至 ADRDY=1（ADRDY 会在 ADC 启动时间后置 1）。这可以使用关联中断来实现（将 ADRDYIE 置 1）。
4. 向 ADCx_ISR 寄存器中的 ADRDY 位写入“1”，将其清零（可选）。

通过软件禁止 ADC 的流程

1. 检查 ADSTART 和 JADSTART 是否均为 0，以确保当前未执行任何转换。如有需要，可将 ADSTP 置 1，并将 JADSTP 置 1，随后等待至 ADSTP=0 且 JADSTP=0，以停止任何正在进行的常规转换和注入转换。
2. 将 ADDIS 置 1。
3. 如果应用要求，可等待 ADEN=0，直至模拟 ADC 已有效禁止（ADEN=0 后，ADDIS 将自动复位）。

图 133. 使能/禁止 ADC



25.3.10 写入 ADC 控制位时的限制

仅当 ADC 已禁止（ADEN 必须等于 0）时，软件才可通过写入 RCC 控制位的方式配置和使能 ADC 时钟（请参见 RCC 部分）、ADCx_DIFSEL 寄存器中的控制位 DIFSEL、ADCx_CCR 寄存器、以及 ADCx_CR 寄存器中的控制位 ADCAL 和 ADEN。

只有在 ADC 已使能、并且没有待处理的禁止 ADC 的请求（ADEN 必须等于 1 且 ADDIS 必须等于 0）时，才允许软件向 ADCx_CR 寄存器的控制位 ADSTART、JADSTART 和 ADDIS 执行写操作。

对于 ADCx_CFGR、ADCx_SMPRy、ADCx_TRy、ADCx_SQRy、ADCx_JDRy、ADCx_OFRy 和 ADCx_IER 寄存器的所有其他控制位：

- 对于与常规转换配置相关的控制位，仅当 ADC 已使能 (ADEN=1) 且未进行常规转换（ADSTART 必须等于 0）时，才允许软件对这些位执行写操作。
- 对于与注入转换配置相关的控制位，仅当 ADC 已使能 (ADEN=1) 且未进行注入转换（JADSTART 必须等于 0）时，才允许软件对这些位执行写操作。

仅当 ADC 已使能（最终会进行转换）且没有待处理的禁止 ADC 的请求（ADSTART 或 JADSTART 必须等于 1 且 ADDIS 必须等于 0）时，软件才可以对 ADCx_CR 寄存器中的 ADSTP 或 JADSTP 控制位执行写操作。

如果 ADC 已使能 (ADEN=1)，软件可随时对 ADCx_JSQR 寄存器执行写操作。

注： *没有硬件保护可以防止执行此类禁止的写操作，ADC 操作可能进入未知状态。要从此状态恢复，必须禁止 ADC（将 ADEN 清零，并且将 ADCx_CR 寄存器的所有位都清零）。*

25.3.11 通道选择（SQRx、JSQRx）

每个 ADC 的复用通道多达20条：

- 6 路来自模拟焊盘和 GPIO 焊盘的快速模拟输入 (ADCx_INP/INN[0..5])
- 多达 14 路来自 GPIO 焊盘的慢速模拟输入 (ADCx_INP/INN[6..19])
- ADC 连接至 5 路内部模拟输入：
  - 内部温度传感器 (V_{SENSE}) 连接到 ADC3_INP/INN18
  - 内部参考电压 (V_{REFINT}) 连接到 ADC3_INP/INN19
  - V_{BAT} 监测通道 (V_{BAT}/4) 连接到 ADC3_INP/INN17
  - DAC 内部通道 1，连接到 ADC2_INP/INN16
  - DAC 内部通道 2，连接到 ADC2_INP/INN17

可以将转换分为两组：常规转换和注入转换。每个组包含一个转换序列，该序列可按任意顺序在任意通道上完成。例如，可按以下顺序对序列进行转换：ADCx_INP/INN3、ADCx_INP/INN8、ADCx_INP/INN2、ADCx_INP/INN2、ADCx_INP/INN2、ADCx_INP/INN0、ADCx_INP/INN2、ADCx_INP/INN2、ADCx_INP/INN15。

- 一个**常规转换组**最多由 16 个转换构成。必须在 ADCx_SQRy 寄存器中选择转换序列的常规通道及其顺序。常规转换组中的转换总数必须写入 ADCx_SQR1 寄存器中的 L[3:0] 位。
- 一个**注入转换组**最多由 4 个转换构成。必须在 ADCx_JSQR 寄存器中选择转换序列的注入通道及其顺序。注入转换组中的转换总数必须写入 ADCx_JSQR 寄存器中的 L[1:0] 位。

不得在可能常规转换时对 ADCx_SQRy 寄存器进行修改。因此，必须先写入 ADSTP=1 停止 ADC 常规转换（请参见*第 25.3.18 节：停止正在进行的转换（ADSTP、JADSTP）*）。

进行注入转换时，可实时修改 ADCx_JSQR 寄存器。请参见第 25.3.22 节：注入转换的上下文队列。

温度传感器、V_{REFINT} 和 V_{BAT} 内部通道

温度传感器 V_{SENSE} 连接到通道 ADC3 VINP[18]。

内部参考电压 V_{REFINT} 连接到 ADC3 V_{INP}[19]。

V_{BAT} 通道连接到通道 ADC3 V_{INP}[17]。

注： *要对其中一条内部模拟通道进行转换，必须先对 ADCx_CCR 寄存器中的 VREFEN、VSENSEEN 或 VBATEN 位进行编程，以使能相应的模拟源。*

25.3.12 通道预选寄存器 (ADCx_PCSEL)

对于每条通过 SQRx 或 JSQRx 选择的通道，必须先对相应的 ADCx_PCSEL 位进行配置。

该 ADCx_PCSEL 位控制集成到 IO 中的传输门。ADC 输入 MUX 会根据 SQRx 和 JSQRx 以非常快的速度选择 ADC 输入，而集成到 IO 中的传输门的响应速度不会像 ADC 复用器那样快。为了避免传输门对 IO 的控制出现延迟，必须预选输入通道，这些通道将在 SQRx 和 JSQRx 中选择。

输入通道是根据每路 ADC 输入的 $V_{INP[i]}$ 选择的。如果 ADC1 要将 ADC123_INP2($V_{INP[2]}$) 转换为差分模式，则还需要在 ADCx_PCSEL 中选择 ADC123_INP6($VI_{NP[6]}$)。

一些 ADC 输入连接至 ADCx 的多个 $V_{INP[i]}$。这些输入会与 ADCx_PCSEL 寄存器位进行或运算。

25.3.13 可独立设置各通道采样时间（SMPR1、SMPR2）

开始转换之前，ADC 必须在待测量电压源与 ADC 内置采样电容之间建立直接连接。该采样时间必须足以使输入电压源为嵌入式电容充电至输入电压水平。

对各通道进行采样时可以使用不同的采样时间，采样时间可通过 ADCx_SMPR1 和 ADCx_SMPR2 寄存器中的 SMP[2:0] 位编程，可选采样时间值如下：

- SMP = 000：1.5 个 ADC 时钟周期
- SMP = 001：2.5 个 ADC 时钟周期
- SMP = 010：8.5 个 ADC 时钟周期
- SMP = 011：16.5 个 ADC 时钟周期
- SMP = 100：32.5 个 ADC 时钟周期
- SMP = 101：64.5 个 ADC 时钟周期
- SMP = 110：387.5 个 ADC 时钟周期
- SMP = 111：810.5 个 ADC 时钟周期

总转换时间的计算公式如下：

T_{CONV} = 采样时间 + 7.5 个 ADC 时钟周期

示例：

如果 $F_{adc_ker_ck}$ = 24 MHz，采样时间为 1.5 个 ADC 时钟周期（14 位模式）：

T_{CONV} = (1.5 + 7.5) 个 ADC 时钟周期 = 9 个 ADC 时钟周期 = 0.375 µs（对于快速通道的 14 位模式）

ADC 通过将状态位 EOSMP 置 1 来指示采样阶段结束（仅限常规转换）。

快速通道和慢速通道的采样时间限制

对于每条通道，必须对 SMP[2:0] 位进行编程，以符合数据手册 ADC 特性部分规定的最短采样时间要求。

I/O 模拟开关升压器

当 V_{DDA} 电压过低时，I/O 模拟开关电阻会增大，此时需要相应地调整采样时间（请参见数据手册中的电气特性）。可通过 SYSCFG_CFGR1 寄存器中的 BOOSTE 位使能内部升压器，从而在低 V_{DDA} 条件下最大限度减小该电阻值。

25.3.14 单次转换模式 (CONT=0)

在单次转换模式下，ADC 会将通道的所有转换执行一次。CONT 位为 0 时，可通过以下方式启动此模式：

- 将 ADCx_CR 寄存器中的 ADSTART 位置 1（如果是常规通道，需要选择软件触发）
- 将 ADCx_CR 寄存器中的 JADSTART 位置 1（如果是注入通道，需要选择软件触发）
- 外部硬件触发事件（适用于常规通道或注入通道）
  触发外部事件之前，ADSTART 位或 JADSTART 位必须置 1。

在常规序列中，每次转换完成后：

- 转换数据存储在 32 位 ADCx_DR 寄存器中
- EOC（常规转换结束）标志置 1
- EOCIE 位置 1 时将产生中断

在注入序列中，每次转换完成后：

- 转换数据存储在四个 32 位 ADC_JDR1 寄存器的其中一个寄存器中
- JEOC（注入转换结束）标志置 1
- JEOCIE 位置 1 时将产生中断

常规序列完成后：

- EOS（常规序列结束）标志置 1
- EOSIE 位置 1 时将产生中断

注入序列完成后：

- JEOS（注入序列结束）标志置 1
- JEOSIE 位置 1 时将产生中断

随后，ADC 会停止工作，直至发生新的外部常规或注入触发，或者 ADSTART 或 JADSTART 位再次置 1。

注： *要转换单个通道，可将序列长度编程为 1。*

25.3.15 连续转换模式 (CONT=1)

该模式仅适用于常规通道。

在连续转换模式下，如果发生软件或硬件常规触发事件，ADC 会将通道的所有常规转换执行一次，随后会自动重启并持续执行序列的每个转换。CONT 位为 1 时，可通过外部触发或将 ADCx_CR 寄存器中的 ADSTART 位置 1 来启动此模式。

在常规序列中，每次转换完成后：

- 转换数据存储在 32 位 ADCx_DR 寄存器中
- EOC（转换结束）标志置 1
- EOCIE 位置 1 时将产生中断

转换序列完成后：

- EOS（序列结束）标志置 1
- EOSIE 位置 1 时将产生中断

随后，会立即重启新序列，ADC 会继续重复执行转换序列。

注： *要转换单个通道，可将序列长度编程为 1。*

不能同时使能不连续模式和连续模式：禁止同时将 DISCEN 和 CONT 位置 1。

注入通道不能连续转换，唯一例外的是，在连续转换模式下（使用 JAUTO 位）注入通道配置为在常规通道后的自动转换，请参见自动注入模式一节。

25.3.16 开始转换（ADSTART、JADSTART）

软件通过将 ADSTART 置 1 的方式开始进行 ADC 常规转换。

ADSTART 置 1 后，会开始进行转换：

- 立即开始转换：EXTEN = 0x0 时（软件触发）
- 在所选常规硬件触发的下一有效边沿开始转换：EXTEN /= 0x0 时

软件通过将 JADSTART 置 1 的方式开始进行 ADC 注入转换。

JADSTART 置 1 后，会开始进行转换：

- 立即开始转换：JEXTEN = 0x0 时（软件触发）
- 在所选注入硬件触发的下一有效边沿开始转换：JEXTEN /= 0x0 时

注： *在自动注入模式下 (JAUTO=1)，使用 ADSTART 位开始常规转换，然后再进行自动注入转换（JADSTART 必须保持清零）。*

ADSTART 位和 JADSTART 位还提供当前是否正在进行 ADC 操作的信息。可以在 ADSTART=0 且 JADSTART=0（指示 ADC 处于空闲状态）时重新配置 ADC。

ADSTART 通过硬件清零：

- 在使用软件触发的单次模式下（CONT=0，EXTEN=0x0）
  - 只要转换序列结束 (EOS=1) 就清零
- 在使用软件触发的不连续模式下（CONT=0、DISCEN=1、EXTEN=0x0）
  - 转换结束时 (EOC=1) 清零
- 在所有其他情况下（CONT=x、EXTEN=x）
  - 执行由软件调用的 ADSTP 程序之后清零

注： *在连续模式下 (CONT=1)，由于序列会自动重新启动，因此，当 EOS 置 1 时，ADSTART 位不会通过硬件清零。*

如果在单次模式下选择了硬件触发（CONT=0 且 EXTEN /=0x00），当 EOS 置 1 时，ADSTART 位不会通过硬件清零，借此软件无需再次为下一个硬件触发事件复位 ADSTART。这样可确保不会错过任何后续的硬件触发。

JADSTART 通过硬件清零：

- 在使用注入触发的单次模式下 (JEXTEN=0x0)
  - 如果 JDISCEN=1，只要注入转换序列结束（JEOS 置 1）或子组处理结束就清零。
- 在所有情况下 (JEXTEN=x)
  - 执行由软件调用的 JADSTP 程序之后清零。

注： *选择软件触发时，如果 EOC 标志仍为高电平，则不应将 ADSTART 位置 1。*

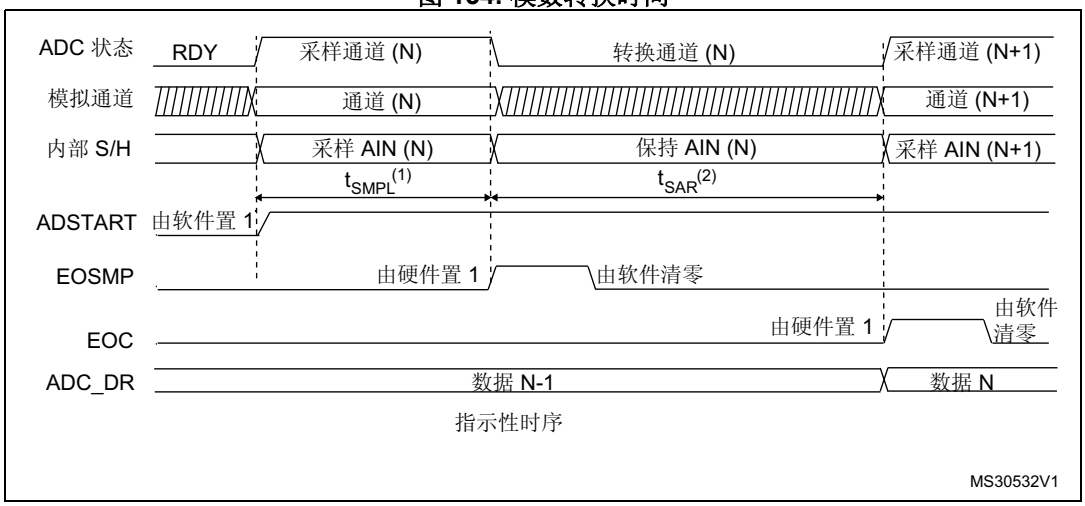
25.3.17 时序

从转换开始到转换结束所经过的时间是配置的采样时间与逐次逼近时间（具体取决于数据分辨率）的总和：

$T_{CONV} = T_{SMPL} + T_{SAR} = [1.5\ _{|min} + 7.5\ _{|14\ 位}] \times T_{adc_ker_ck}$

$T_{CONV} = T_{SMPL} + T_{SAR} = 62.5\ ns\ _{|min} + 312.5\ ns\ _{|14\ 位} = 375.0\ ns$（适用于 $F_{adc_ker_ck} = 24\ MHz$）

图 134. 模数转换时间



1. T_{SMPL} 取决于 SMP[2:0]
2. T_{SAR} 取决于 RES[2:0]

25.3.18 停止正在进行的转换（ADSTP、JADSTP）

软件决定是否停止转换，要停止正在进行的常规转换，应将 ADSTP 置 1；要停止正在进行的注入转换，应将 JADSTP 置 1 。

停止转换将复位正在进行的 ADC 操作。随后可重新配置 ADC（例如：更改通道选择或触发），为新操作做好准备。

请注意，不能在常规转换仍在执行时停止注入转换，反之亦然。这样便可在常规转换仍在进行时重新配置注入转换序列和触发（反之亦然）。

如果 ADSTP 位由软件置 1，则会中止任何正在进行的常规转换，并会丢弃部分转换结果（ADCx_DR 寄存器不会更新为当前转换结果）。

如果 JADSTP 位由软件置 1，则会中止任何正在进行的注入转换，并会丢弃部分转换结果（ADCx_JDRy 寄存器不会更新为当前转换结果）。扫描序列也会中止并会复位（这意味着重启 ADC 将重新开始新的序列）。

该程序执行完毕后，ADSTP/ADSTART 位（常规转换时）或 ADSTP/JADSTART 位（注入转换时）会由硬件清零，软件必须轮询 ADSTART（或 JADSTART）直至其复位，然后才能判定 ADC 已完全停止运行。

注： *在自动注入模式下 (JAUTO=1)，将 ADSTP 位置 1 会中止常规转换和注入转换（不得使用 JADSTP）。*

图 135. 停止正在进行的常规转换

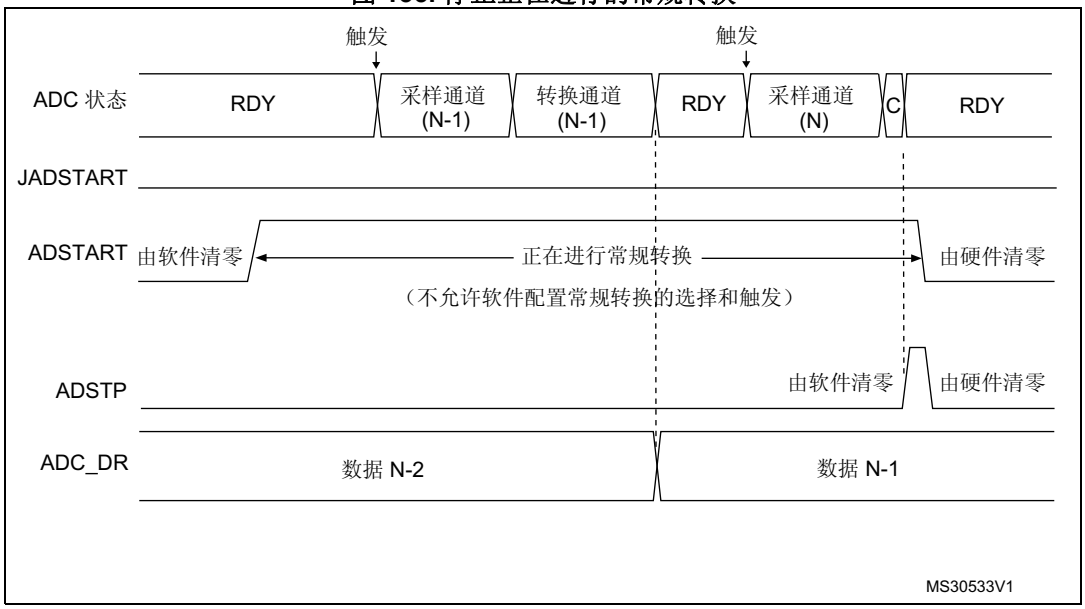
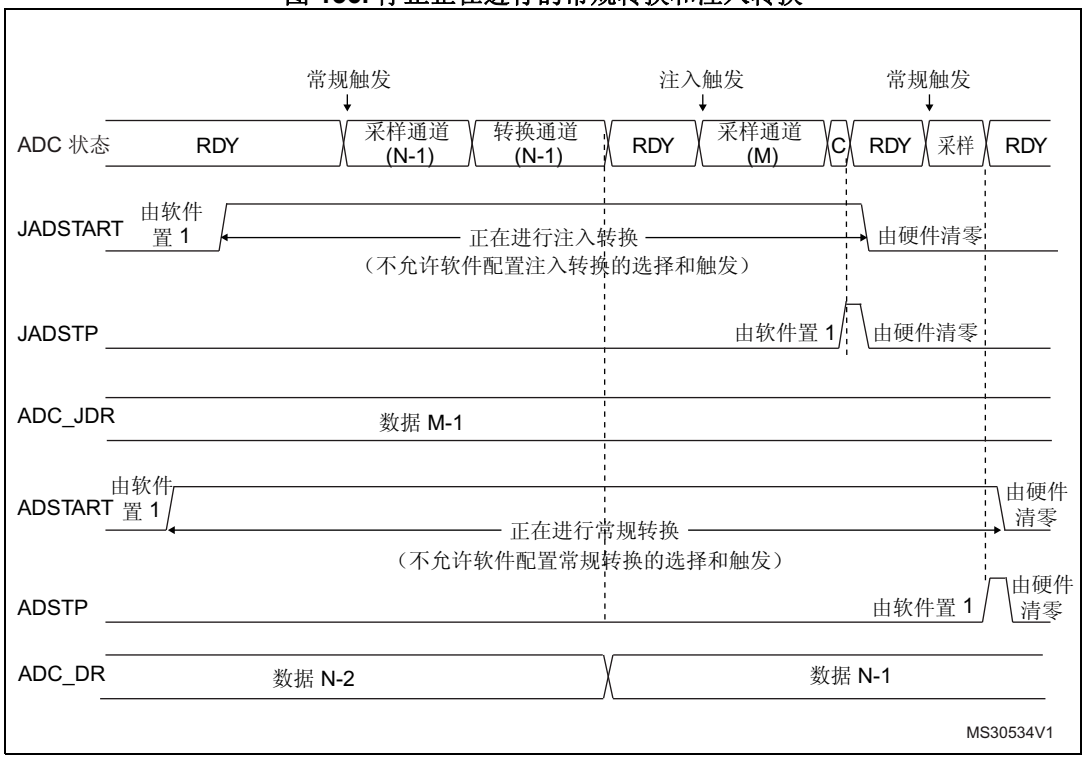


图 136. 停止正在进行的常规转换和注入转换



25.3.19 外部触发转换和触发极性（EXTSEL、EXTEN、JEXTSEL、JEXTEN）

可通过软件或外部事件（例如定时器捕获、输入引脚）触发转换或转换序列。如果 EXTEN[1:0] 控制位（对于常规转换）或 JEXTEN[1:0] 位（对于注入转换）不等于“0b00”，则外部事件能够以所选极性触发转换。

注入队列使能时（位 JQDIS=0），不能实现注入软件触发。

软件将 ADSTART 位置 1 后，常规触发选择有效；软件将 JADSTART 位置 1 后，注入触发选择有效。

在转换进行时发生的硬件触发会被忽略。

- 如果位 ADSTART=0，则会忽略发生的任何常规硬件触发。
- 如果位 JADSTART=0，则会忽略发生的任何注入硬件触发。

表 190 提供了 EXTEN[1:0] 和 JEXTEN[1:0] 值与触发极性之间的对应关系。

表 190. 为常规外部触发配置触发极性

EXTEN[1:0]	源
00	禁止硬件触发检测，使能软件触发检测
01	在上升沿执行硬件触发检测
10	在下降沿执行硬件触发检测
11	在上升沿和下降沿均执行硬件触发检测

注： *不能实时更改常规触发的极性。*

表 191. 为注入外部触发配置触发极性

JEXTEN[1:0]	源
00	– 如果 JQDIS=1（队列禁止）：禁止硬件触发检测，使能软件触发检测 – 如果 JQDIS=0（队列使能）：同时禁止硬件和软件触发检测
01	在上升沿执行硬件触发检测
10	在下降沿执行硬件触发检测
11	在上升沿和下降沿均执行硬件触发检测

注： *如果队列已使能 (JQDIS=0)，可预计并实时更改注入触发的极性。请参见第 25.3.22 节：注入转换的上下文队列。*

EXTSEL[4:0] 和 JEXTSEL[4:0] 控制位用于从 21 个可能事件中选择可触发常规组转换和注入组转换的事件。

可通过注入触发中断常规组转换。

注： *不能实时更改常规触发选择。*
可以预计并实时更改常规触发选择。请参见第 829 页的第 25.3.22 节：注入转换的上下文队列。

每个主 ADC 都与其从 ADC 共享相同的输入触发，如图 137 所述。

图 137. 触发由主 ADC 和从 ADC 共享

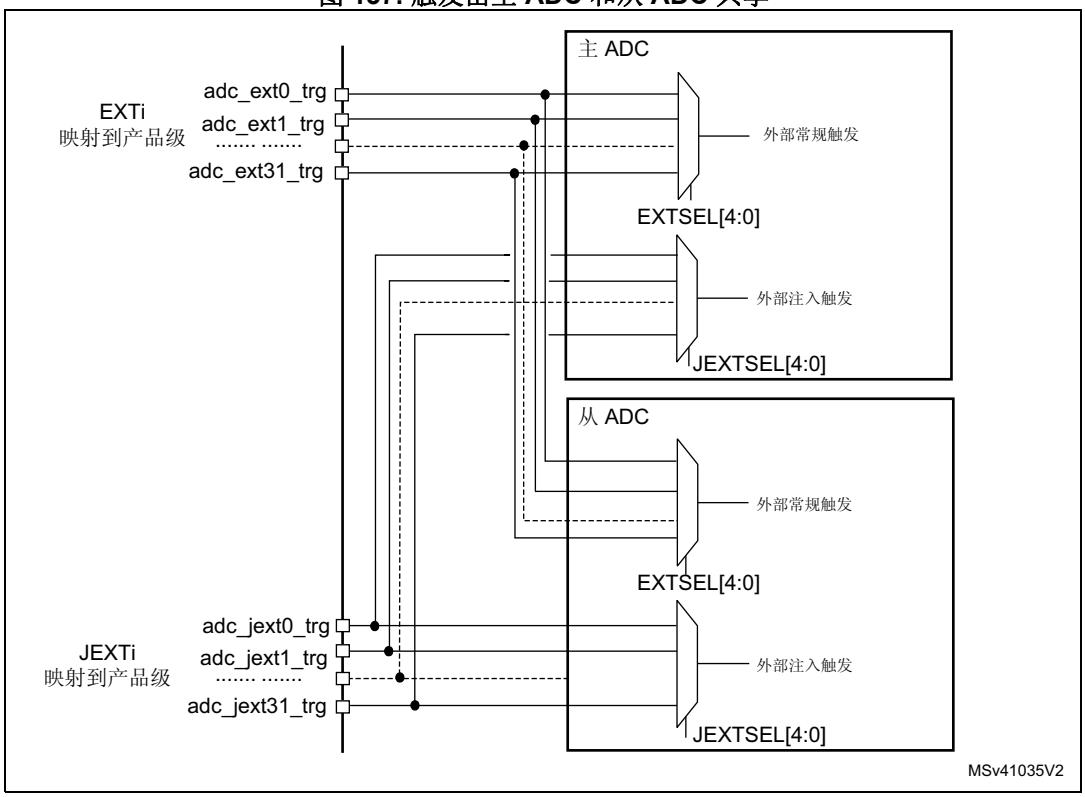


表 192 到*表 193* 列出了用于常规转换和注入转换的三个 ADC 的所有可能存在的外部触发。

表 192. ADC1、ADC2 和 ADC3 - 常规通道的外部触发

名称	源	类型	EXTSEL[4:0]
adc_ext_trg0	TIM1_CC1 事件	片上定时器的内部信号	00000
adc_ext_trg1	TIM1_CC2 事件	片上定时器的内部信号	00001
adc_ext_trg2	TIM1_CC3 事件	片上定时器的内部信号	00010
adc_ext_trg3	TIM2_CC2 事件	片上定时器的内部信号	00011
adc_ext_trg4	TIM3_TRGO事件	片上定时器的内部信号	00100
adc_ext_trg5	TIM4_CC4 事件	片上定时器的内部信号	00101
adc_ext_trg6	EXTI线 11	外部引脚	00110
adc_ext_trg7	TIM8_TRGO 事件	片上定时器的内部信号	00111
adc_ext_trg8	TIM8_TRGO2 事件	片上定时器的内部信号	01000
adc_ext_trg9	TIM1_TRGO 事件	片上定时器的内部信号	01001
adc_ext_trg10	TIM1_TRGO2 事件	片上定时器的内部信号	01010
adc_ext_trg11	TIM2_TRGO 事件	片上定时器的内部信号	01011
adc_ext_trg12	TIM4_TRGO 事件	片上定时器的内部信号	01100
adc_ext_trg13	TIM6_TRGO 事件	片上定时器的内部信号	01101
adc_ext_trg14	TIM15_TRGO 事件	片上定时器的内部信号	01110
adc_ext_trg15	TIM3_CC4 事件	片上定时器的内部信号	01111
adc_ext_trg16	HRTIM1_ADCTRG1 事件	片上定时器的内部信号	10000
adc_ext_trg17	HRTIM1_ADCTRG3 事件	片上定时器的内部信号	10001
adc_ext_trg18	LPTIM1_OUT 事件	片上定时器的内部信号	10010
adc_ext_trg19	LPTIM2_OUT 事件	片上定时器的内部信号	10011
adc_ext_trg20	LPTIM3_OUT 事件	片上定时器的内部信号	10100
adc_ext_trg21	保留	-	10101
adc_ext_trg22	保留	-	10110
adc_ext_trg23	保留	-	10111
adc_ext_trg24	保留	-	11000
adc_ext_trg25	保留	-	11001
adc_ext_trg26	保留	-	11010
adc_ext_trg27	保留	-	11011
adc_ext_trg28	保留	-	11100
adc_ext_trg29	保留	-	11101
adc_ext_trg30	保留	-	11110
adc_ext_trg31	保留	-	11111

表 193. ADC1、ADC2 和 ADC3 - 注入通道的外部触发

名称	源	类型	EXTSEL[4:0]
adc_ext_trg0	TIM1_TRGO 事件	片上定时器的内部信号	00000
adc_ext_trg1	TIM1_CC4 事件	片上定时器的内部信号	00001
adc_jext_trg2	TIM2_TRGO 事件	片上定时器的内部信号	00010
adc_jext_trg3	TIM2_CC1 事件	片上定时器的内部信号	00011
adc_jext_trg4	TIM3_CC4 事件	片上定时器的内部信号	00100
adc_jext_trg5	TIM4_TRGO 事件	片上定时器的内部信号	00101
adc_jext_trg6	EXTI线15	外部引脚	00110
adc_jext_trg7	TIM8_CC4 事件	片上定时器的内部信号	00111
adc_jext_trg8	TIM1_TRGO2 事件	片上定时器的内部信号	01000
adc_jext_trg9	TIM8_TRGO 事件	片上定时器的内部信号	01001
adc_jext_trg10	TIM8_TRGO2 事件	片上定时器的内部信号	01010
adc_jext_trg11	TIM3_CC3 事件	片上定时器的内部信号	01011
adc_jext_trg12	TIM3_TRGO事件	片上定时器的内部信号	01100
adc_jext_trg13	TIM3_CC1 事件	片上定时器的内部信号	01101
adc_jext_trg14	TIM6_TRGO 事件	片上定时器的内部信号	01110
adc_jext_trg15	TIM15_TRGO 事件	片上定时器的内部信号	01111
adc_jext_trg16	HRTIM1_ADCTRG2 事件	片上定时器的内部信号	10000
adc_jext_trg17	HRTIM1_ADCTRG4 事件	片上定时器的内部信号	10001
adc_jext_trg18	LPTIM1_OUT 事件	片上定时器的内部信号	10010
adc_jext_trg19	LPTIM2_OUT 事件	片上定时器的内部信号	10011
adc_jext_trg20	LPTIM3_OUT 事件	片上定时器的内部信号	10100
adc_jext_trg21	保留	-	10101
adc_jext_trg22	保留	-	10110
adc_jext_trg23	保留	-	10111
adc_jext_trg24	保留	-	11000
adc_jext_trg25	保留	-	11001
adc_jext_trg26	保留	-	11010
adc_jext_trg27	保留	-	11011
adc_jext_trg28	保留	-	11100
adc_jext_trg29	保留	-	11101
adc_jext_trg30	保留	-	11110
adc_jext_trg31	保留	-	11111

25.3.20 注入通道管理

触发注入模式

要使用触发注入，必须将 ADCx_CFGR 寄存器中的 JAUTO 位清零。

1. 通过外部触发或将 ADCx_CR 寄存器中的 ADSTART 位置 1 来启动常规通道组转换。
2. 如果在常规通道组转换期间出现外部注入触发或者 ADCx_CR 寄存器中的 JADSTART 位置 1，则当前的转换会复位，并会启动注入通道序列切换（所有注入通道都会转换一次）。
3. 然后，常规通道组的常规转换会从上次中断的常规转换处恢复。
4. 如果在注入转换期间出现常规事件，注入转换不会中断，但在注入序列结束时会执行常规序列。*图 138* 显示了相应的时序图。

注： *使用触发注入时，必须确保触发事件之间的间隔长于注入序列。例如，如果序列长度为 20 个 ADC 时钟周期（即，采样时间为 1.5 个时钟周期的两次转换），则触发事件的最小间隔不能小于 21 个 ADC 时钟周期。*

自动注入模式

如果将 ADCx_CFGR 寄存器中的 JAUTO 位置 1，则注入组中的通道会在常规组通道之后自动转换。这可用于转换最多由 20 个转换构成的序列，这些转换在 ADCx_SQRy 和 ADCx_JSQR 寄存器中编程。

在该模式下，必须将 ADCx_CR 寄存器中的 ADSTART 位置 1 以开始常规转换，然后再进行注入转换（JADSTART 必须保持清零）。将 ADSTP 位置 1 会中止常规转换和注入转换（不得使用 JADSTP 位）。

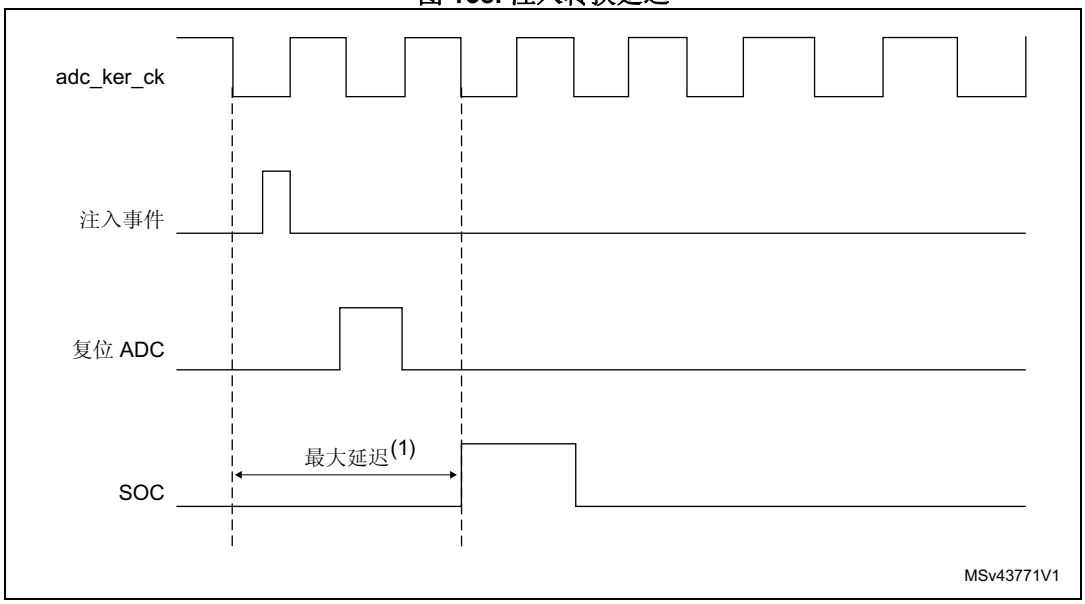
在此模式下，必须禁止注入通道上的外部触发。

如果 CONT 位和 JAUTO 位均已置 1，则在转换常规通道之后会继续转换注入通道。

注： *不能同时使用自动注入和不连续采样模式。*

如果在 JAUTO 模式下使用 DMA 导出常规定序器的数据，需要将其设定为循环模式（将 DMA_CCRx 寄存器中的 CIRC 位置 1）。如果 CIRC 位复位（单发模式），JAUTO 序列将在出现 DMA 传输完成事件时停止。

图 138. 注入转换延迟



1. 有关延迟的最大值，请参见器件数据手册中的电气特性部分。

25.3.21 不连续模式（DISCEN、DISCNUM、JDISCEN）

常规组模式

可将 ADCx_CFGR 寄存器中的 DISCEN 位置 1 来使能此模式。

该模式用于转换含有 n (n ≤ 8) 个转换的短序列（子组），该短序列是在 ADCx_SQRy 寄存器中选择的转换序列的一部分。可通过写入 ADCx_CFGR 寄存器中的 DISCNUM[2:0] 位来指定 n 的值。

出现外部触发时，将启动在 ADCx_SQR 寄存器中选择的接下来 n 个转换，直到序列中的所有转换均完成为止。通过 ADCx_SQR1 寄存器中的 L[3:0] 位定义总序列长度。

示例：

- DISCEN=1，n = 3，要转换的通道 = 1、2、3、6、7、8、9、10、11
  - 第一次触发：转换的通道为 1、2、3（每次转换时都生成 EOC 事件）。
  - 第二次触发：转换的通道为 6、7、8（每次转换时都生成 EOC 事件）。
  - 第三次触发：转换的通道为 9、10、11（每次转换时都生成 EOC 事件），并会在通道 11 转换完成后生成 EOS 事件。
  - 第四次触发：转换的通道为 1、2、3（每次转换时都生成 EOC 事件）。
  - ...
- DISCEN=0，要转换的通道 = 1、2、3、6、7、8、9、10、11
  - 第一次触发：转换整个序列：通道 1、然后是通道 2、3、6、7、8、9、10 和 11。每次转换都会生成 EOC 事件，最后一次转换还会生成 EOS 事件。
  - 所有后续触发事件都将重启整个序列。

注： *在不连续模式下转换常规组时，不会出现翻转（序列最后一个子组的转换次数少于 n 次）。转换完所有子组后，下一个触发信号将启动第一个子组的转换。在上述示例中，第四次触发重新转换了第一个子组中的通道 1、2 和 3。*

不能同时使能不连续模式和连续模式。如果同时使能两种模式（即 DISCEN=1、CONT=1），ADC 会认定连续模式已禁止并继续执行相关操作。

注入组模式

可将 ADCx_CFGR 寄存器中的 JDISCEN 位置 1 来使能此模式。在出现外部注入触发事件之后，该模式会逐通道转换在 ADCx_JSQR 寄存器中选择的序列，相当于不连续模式下常规通道“n”固定为 1 的情况。

出现外部触发时，将启动在 ADCx_JSQR 寄存器中选择的下一个通道转换，直到序列中的所有转换均完成为止。通过 ADCx_JSQR 寄存器中的 JL[1:0] 位定义总序列长度。

示例：

- JDISCEN=1，要转换的通道 = 1、2、3
  - 第一次触发：转换通道 1（生成 JEOC 事件）
  - 第二次触发：转换通道 2（生成 JEOC 事件）
  - 第三次触发：转换通道 3 并生成 JEOC 事件和 JEOS 事件
  - ...

注： *转换完所有注入通道后，下一个触发信号将启动第一个注入通道的转换。在上述示例中，第 4 次触发重新转换了第 1 个注入通道。*

不能同时使用自动注入模式和不连续模式：当 JAUTO 置 1 时，DISCEN 和 JDISCEN 位必须通过软件保持清零状态。

25.3.22 注入转换的上下文队列

实现上下文队列可为下一个注入转换序列准备多达 2 个上下文。必须将 ADCx_CFGR 寄存器的 JQDIS 位复位才能使能此功能。上下文队列使能时，只能进行硬件触发转换。

该上下文包括：

- 注入触发的配置（ADCx_JSQR 寄存器中的 JEXTEN[1:0] 位和 JEXTSEL[4:0] 位）
- 注入序列的定义（ADCx_JSQR 寄存器中的 JSQx[4:0] 位和 JL[1:0] 位）

上下文的所有参数都会在 ADCx_JSQR 这一寄存器中定义，该寄存器会实现一个双缓冲区队列，可缓冲多达 2 组参数。

- JSQR 寄存器可随时写入，正在进行注入转换时也不例外。
- 每个写入到 JSQR 寄存器中的数据均会存储在上下文队列中。
- 队列开始时为空，对 JSQR 寄存器进行的第一次写访问时会立即更改上下文，随即 ADC 会准备好接收注入触发。
- 注入序列完成后，队列会被占用，上下文会根据队列中存储的后续 JSQR 参数进行更改。这一新的上下文会用于下一个注入转换序列。
- 如果在队列已满的情况下向 JSQR 寄存器执行写操作，会发生队列溢出。这种溢出情况会通过 JQOVF 标志置为有效来指示。发生溢出时，会忽略造成溢出的 JSQR 寄存器写访问，并且上下文队列保持不变。如果 JQOVFIE 位置 1，可产生中断。
- 队列变空时可能执行两种操作，具体取决于寄存器 ADCx_CFGR 的控制位 JQM 的值。
  - 如果 JQM=0，队列刚好在使能 ADC 后为空，但在随后的运行操作期间绝不会变空：队列始终保留上一个有效的上下文，并会根据上一个有效的上下文处理后续的有效注入序列启动。
  - 如果 JQM=1，队列会在注入序列结束后或队列被清空时变空。这种情况下，队列中不存在任何上下文，硬件触发也会被禁止。因此会忽略后续的所有硬件注入触发，直至软件重新向 JSQR 寄存器写入新的注入上下文。

- 读取 JSQR 寄存器会返回当前有效的 JSQR 上下文。如果 JSQR 上下文为空，JSQR 的读出值为 0x0000。
- 如果通过将 JADSTP 置 1 的方式停止注入转换、或者通过将 ADDIS 置 1 的方式禁止 ADC，队列会被清空。
  - 如果 JQM=0，队列会保留上一个有效的上下文。
  - 如果 JQM=1，队列会变空，并会忽略触发。

注： *如果配置为不连续模式（位 JDISCEN=1），只有注入序列的最后一次触发会更改上下文并占用队列。第一次触发仅会占用队列，但其他触发仍为有效触发，如下文中的不连续模式示例所示（两个上下文的长度均为 3）：*

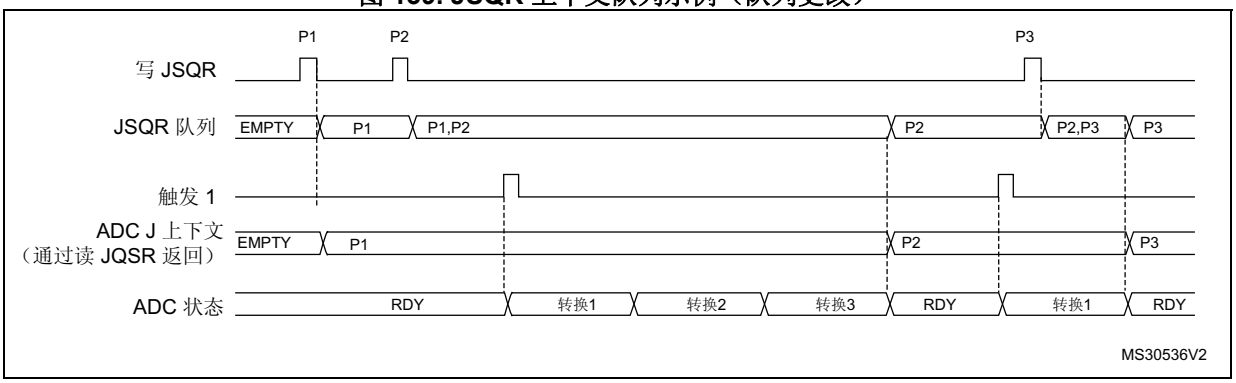
- *第一次触发，不连续。序列 1：上下文 1 已占用，已执行第一次转换*
- *第二次触发，不连续。序列 1：第二次转换。*
- *第三次触发，不连续。序列 1：第三次转换。*
- *第四次触发，不连续。序列 2：上下文 2 已占用，已执行第一次转换。*
- *第五次触发，不连续。序列 2：第二次转换。*
- *第六次触发，不连续。序列 2：第三次转换。*

注： *上下文队列使能时（位 JQDIS=0），仅可使用硬件触发。*

更改触发或序列上下文时的操作

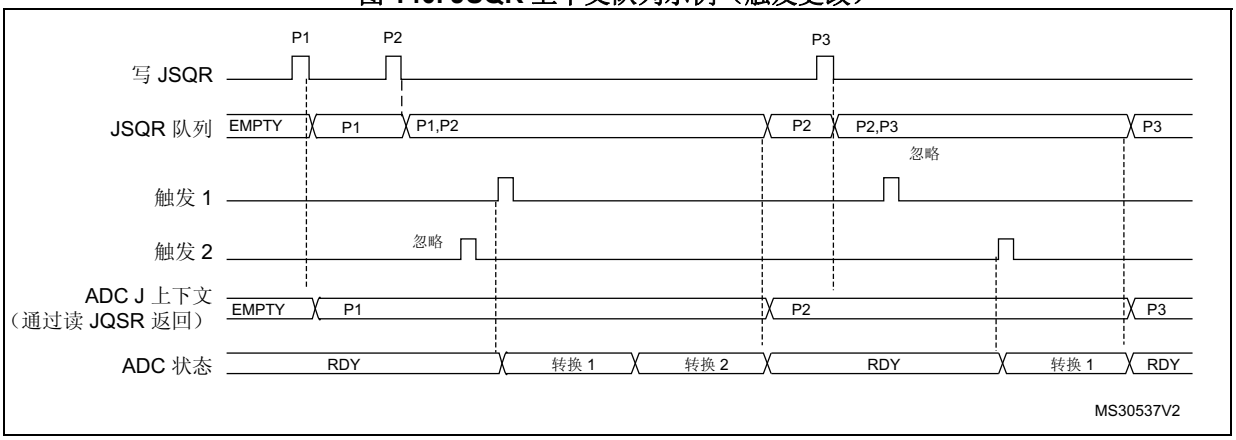
图 139 和*图 140* 显示了更改序列或触发时上下文队列的操作。

图 139. JSQR 上下文队列示例（队列更改）



1. 参数
   P1：3 个转换组成的序列，硬件触发 1
   P2：1 个转换组成的序列，硬件触发 1
   P3：4 个转换组成的序列，硬件触发 1

图 140. JSQR 上下文队列示例（触发更改）

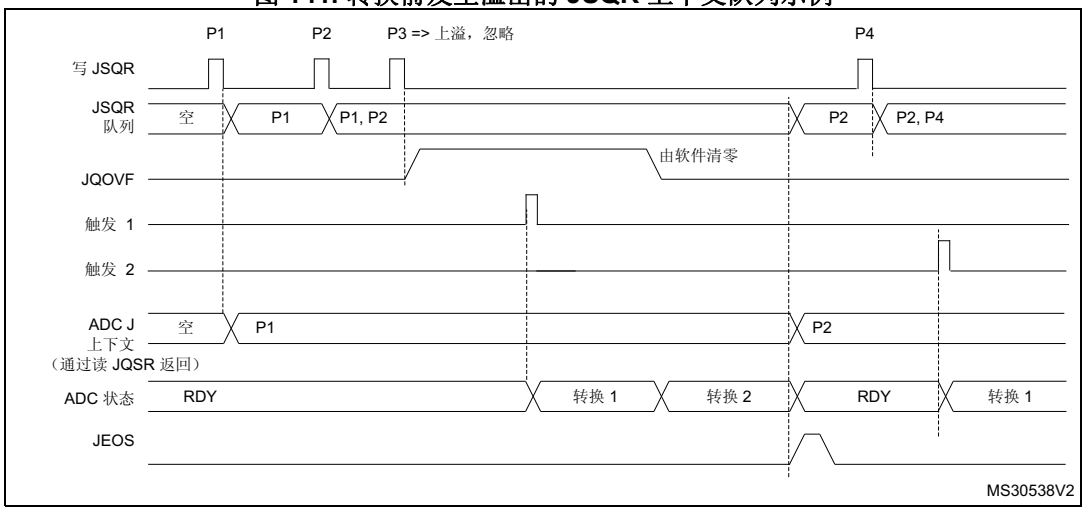


1. 参数
   P1：2 个转换组成的序列，硬件触发 1
   P2：1 个转换组成的序列，硬件触发 2
   P3：4 个转换组成的序列，硬件触发 1

上下文队列：发生队列溢出时的操作

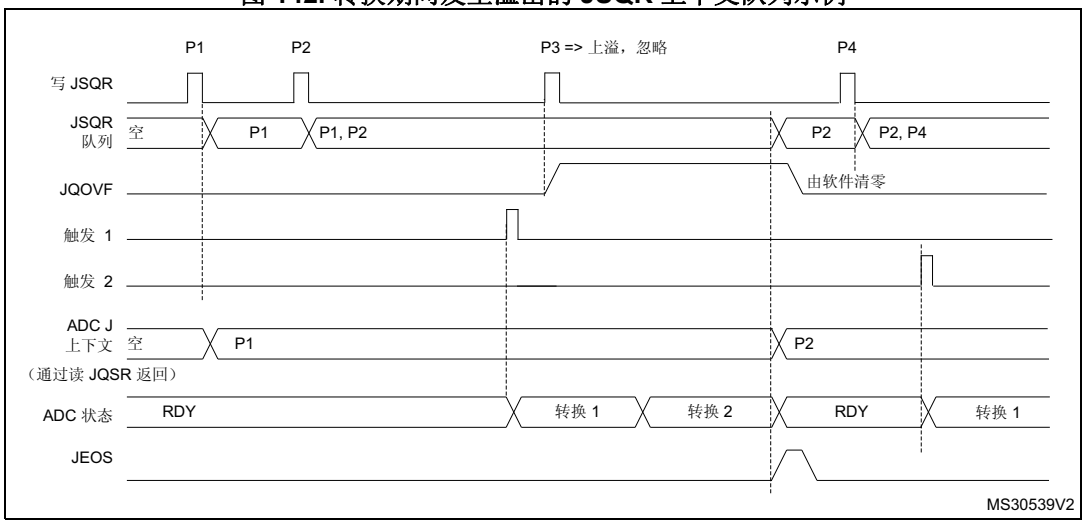
图 141 和*图 142* 显示了转换之前或之后发生溢出时上下文队列的操作。

图 141. 转换前发生溢出的 JSQR 上下文队列示例



1. 参数
   P1：2 个转换组成的序列，硬件触发 1
   P2：1 个转换组成的序列，硬件触发 2
   P3：3 个转换组成的序列，硬件触发 1
   P4：4 个转换组成的序列，硬件触发 1

图 142. 转换期间发生溢出的 JSQR 上下文队列示例



1. 参数
   P1：2 个转换组成的序列，硬件触发 1
   P2：1 个转换组成的序列，硬件触发 2
   P3：3 个转换组成的序列，硬件触发 1
   P4：4 个转换组成的序列，硬件触发 1

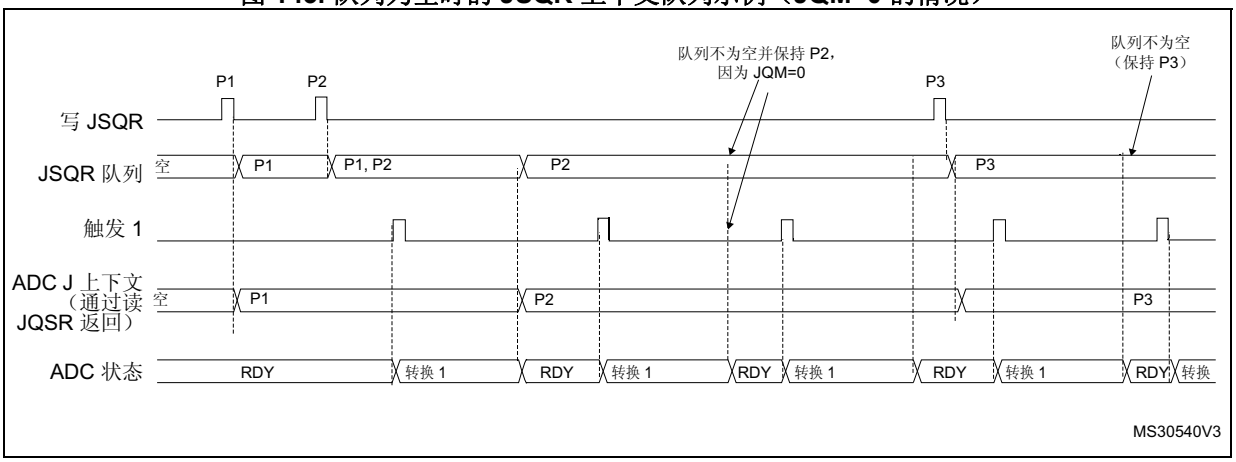
建议按照下述方法管理队列溢出：

- 将每个 P 上下文写入 JSQR 寄存器后，通过标志 JQOVF 指示写操作是否已被忽略（可生成中断）。
- 为避免队列溢出，仅在上一个上下文 P2 的 JEOS 标志置 1 后写入第三个上下文 (P3)。这样可确保上一个上下文已被占用且队列未满。

上下文队列：队列变空时的操作

图 143 和*图 144* 显示了在 JQM=0 或 JQM=1 这两种情况下队列变空时上下文队列的操作。

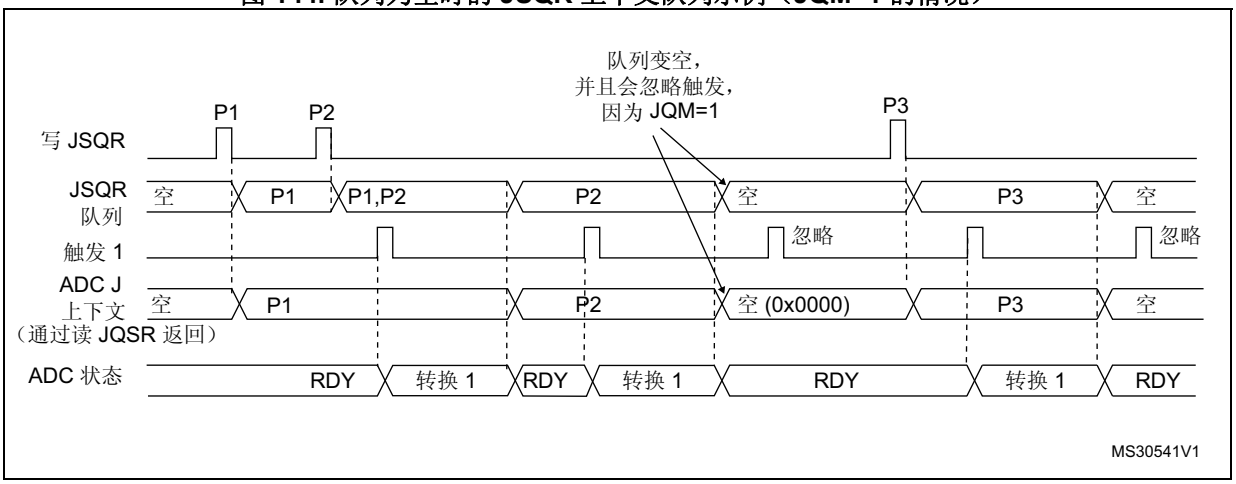
图 143. 队列为空时的 JSQR 上下文队列示例（JQM=0 的情况）



1. 参数
   P1：1 个转换组成的序列，硬件触发 1
   P2：1 个转换组成的序列，硬件触发 1
   P3：1 个转换组成的序列，硬件触发 1

注： *写入 P3 时，上下文会立即改变。但由于内部重新同步的原因，会存在一定的延迟，如果刚好在写入 P3 之后或之前发生触发，可能会发生启动的转换被视为上下文 P2 的情况。为了避免出现这种情况，用户必须确保写入立即应用的新上下文时没有发生 ADC 触发。*

图 144. 队列为空时的 JSQR 上下文队列示例（JQM=1 的情况）

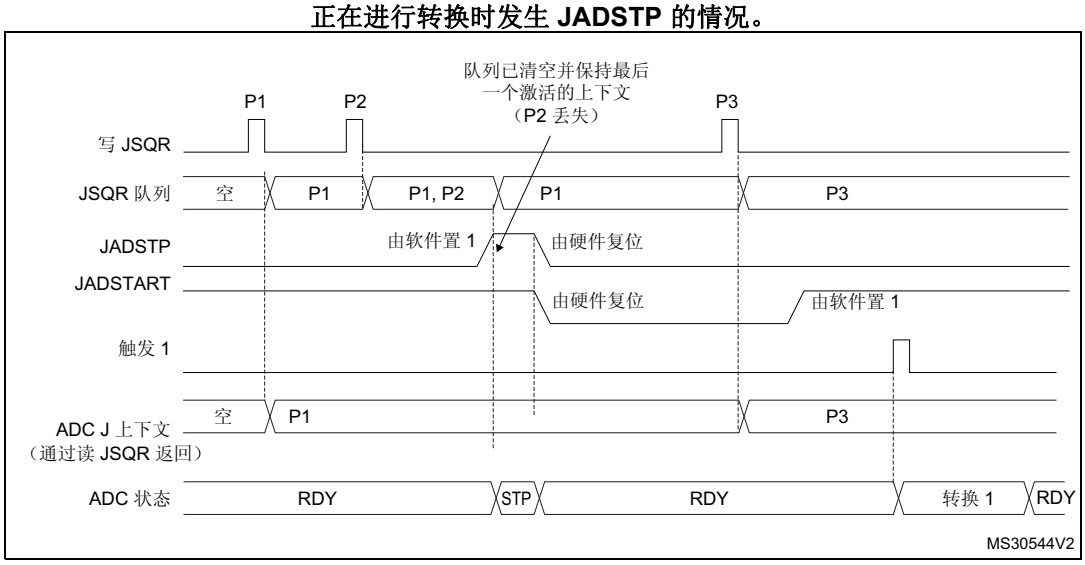


1. 参数
   P1：1 个转换组成的序列，硬件触发 1
   P2：1 个转换组成的序列，硬件触发 1
   P3：1 个转换组成的序列，硬件触发 1

清空上下文队列

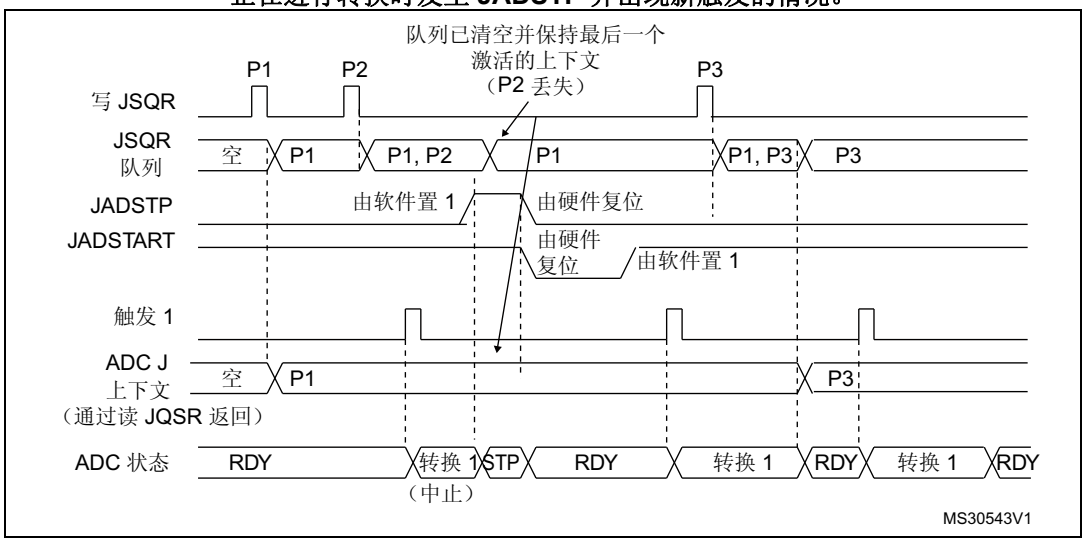
下图显示的是各种情况下清空队列时上下文队列的操作。

图 145. 通过将 JADSTP 置 1 的方式清空 JSQR 上下文队列 (JQM=0)。正在进行转换时发生 JADSTP 的情况。



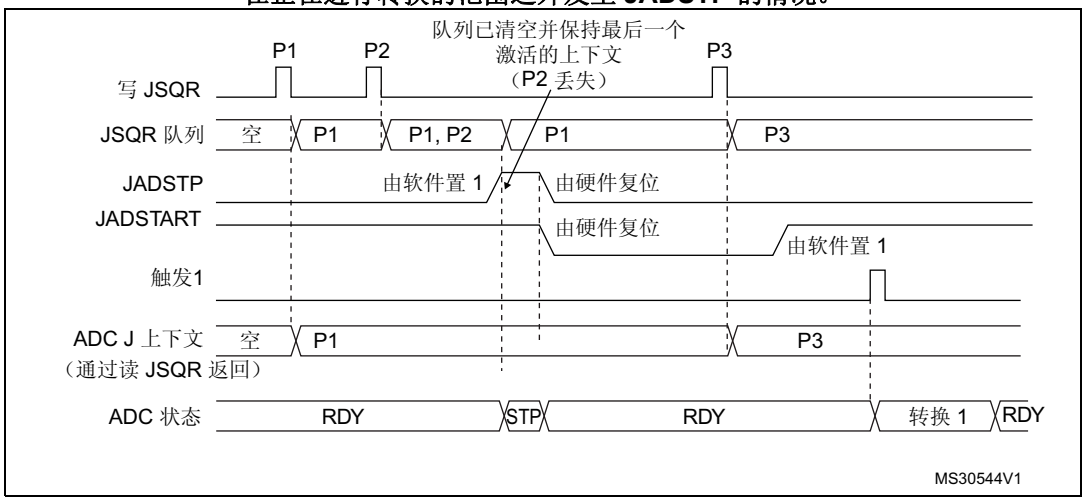
1. 参数
   P1：1 个转换组成的序列，硬件触发 1
   P2：1 个转换组成的序列，硬件触发 1
   P3：1 个转换组成的序列，硬件触发 1

图 146. 通过将 JADSTP 置 1 的方式清空 JSQR 上下文队列 (JQM=0)。正在进行转换时发生 JADSTP 并出现新触发的情况。



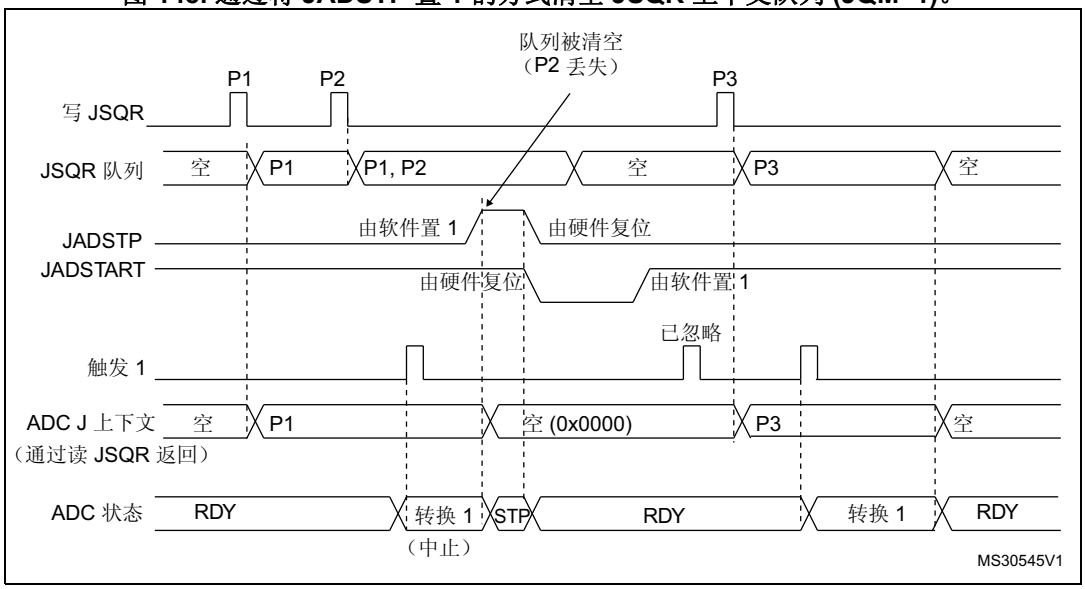
1. 参数
   P1：1 个转换组成的序列，硬件触发 1
   P2：1 个转换组成的序列，硬件触发 1
   P3：1 个转换组成的序列，硬件触发 1

图 147. 通过将 JADSTP 置 1 的方式清空 JSQR 上下文队列 (JQM=0)。在正在进行转换的范围之外发生 JADSTP 的情况。



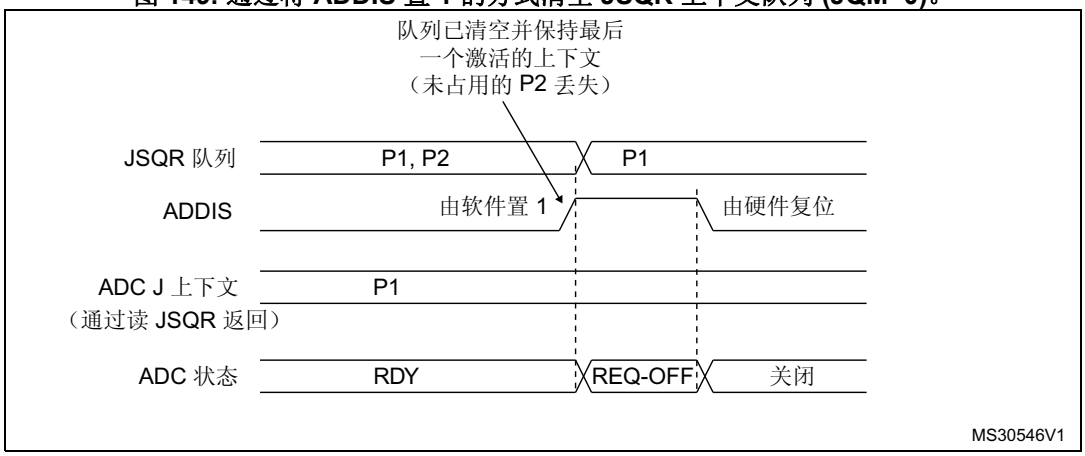
1. 参数
   P1：1 个转换组成的序列，硬件触发 1
   P2：1 个转换组成的序列，硬件触发 1
   P3：1 个转换组成的序列，硬件触发 1

图 148. 通过将 JADSTP 置 1 的方式清空 JSQR 上下文队列 (JQM=1)。



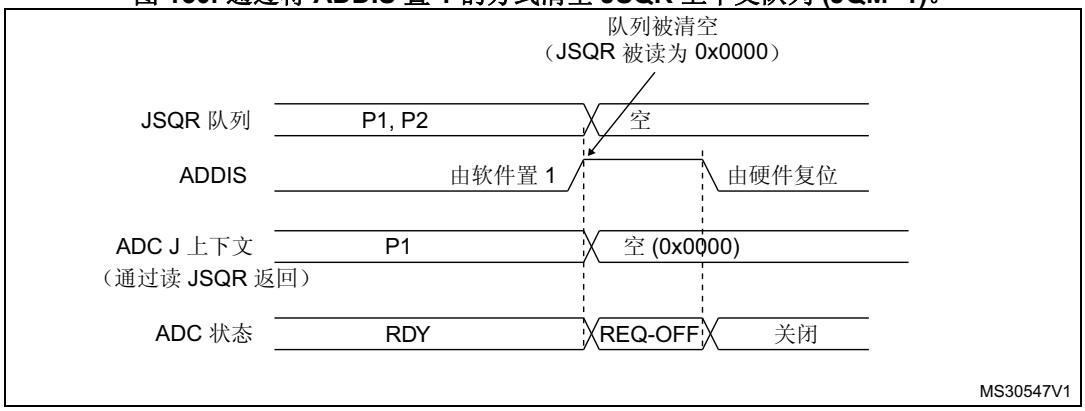
1. 参数
   P1：1 个转换组成的序列，硬件触发 1
   P2：1 个转换组成的序列，硬件触发 1
   P3：1 个转换组成的序列，硬件触发 1

图 149. 通过将 ADDIS 置 1 的方式清空 JSQR 上下文队列 (JQM=0)。



1. 参数
   P1：1 个转换组成的序列，硬件触发 1
   P2：1 个转换组成的序列，硬件触发 1
   P3：1 个转换组成的序列，硬件触发 1

图 150. 通过将 ADDIS 置 1 的方式清空 JSQR 上下文队列 (JQM=1)。



1. 参数
   P1：1 个转换组成的序列，硬件触发 1
   P2：1 个转换组成的序列，硬件触发 1
   P3：1 个转换组成的序列，硬件触发 1

上下文队列：在队列为空时启动 ADC

要在队列为空时启动 ADC 操作，必须按照以下程序进行操作，以免在 ADC 初始化时无法获悉第一个上下文。此程序仅在 JQM 位复位的情况下适用：

5. 写入空 JSQR，JEXTEN 不等于 0（否则会触发软件转换）
6. 将 JADSTART 置 1 (Set NAK)
7. 将 JADSTP 置 1 (Set NAK)
8. 等待，直至 JADSTART 复位
9. 将 JADSTART 置 1 (Set NAK)

禁止队列

可通过将 ADCx_CFGR 寄存器中的 JQDIS 位置 1 的方式禁止队列。

上下文队列：ADCx_JSQR 寄存器编程

当注入转换上下文队列使能时 (JQDIS=0)，必须通过一次寄存器写访问编程 ADCx_JSQR。由于 JL[1:0] 寄存器定义了注入序列数目，因此必须同时写入对应的 JSQ1 到 JSQ4。如果在注入转换开始前对 ADCx_JSQR 进行重新编程，那么重新编程的数据会置于队列中。上下文队列为空时，ADCx_JSQR 的读出值为 0x0000。寄存器访问不应使用“读-修改-写”序列。

如果在已有 2 个上下文排队的情况下对 ADCx_JSQR 进行编程，JQOVF 标志将置 1，并会产生中断。

25.3.23 可编程分辨率 (RES) - 快速转换模式

可通过降低 ADC 分辨率来执行快速转换。

通过对控制位 RES[1:0] 进行编程，可将分辨率配置为 16 位、14 位、12 位、10 位、8 位。*图 155*、*图 156*、*图 157* 和 *图 158* 显示了转换结果格式与分辨率和数据对齐方式的对应关系。

对于不要求使用高数据精度的应用，分辨率越低，转换时间越短，降低分辨率可缩短逐次逼近步骤所需的转换时间，如 *表 194* 所示。

表 194. T_{SAR} 与分辨率的对应关系

RES	T_{SAR} ADC 时钟周期	T_{SAR} (ns)，F_{ADC}=24 MHz	T_{ADC}（ADC 时钟周期）（采样时间 = 1.5 个 ADC 时钟周期）	T_{ADC} (ns)，F_{ADC}=24 MHz
16	8.5 个 ADC 时钟周期	354.2	10 个 ADC 时钟周期	416.7
14	7.5 个 ADC 时钟周期	312.5	9 个 ADC 时钟周期	375
12	6.5 个 ADC 时钟周期	270.8	8 个 ADC 时钟周期	333.3
10	5.5 个 ADC 时钟周期	229.2	7 个 ADC 时钟周期	291.7
8	4.5 个 ADC 时钟周期	187.5	6 个 ADC 时钟周期	250.0

25.3.24 转换结束、采样阶段结束（EOC、JEOC、EOSMP）

每次出现常规转换结束 (EOC) 事件和注入转换 (JEOC) 事件时，ADC 都会通知应用。

新的常规转换数据出现在 ADCx_DR 寄存器中后，ADC 会立即将 EOC 标志置 1。如果 EOCIE 位置 1，可产生中断。EOC 标志可通过由软件向其写入 1 或读取 ADCx_DR 的方式来清零。

新的注入转换数据出现在 ADCx_JDRy 寄存器中后，ADC 会立即将 JEOC 标志置 1。如果 JEOCIE 位置 1，可产生中断。JEOC 标志可通过由软件向其写入 1 或读取相应 ADCx_JDRy 寄存器的方式来清零。

ADC 还通过将状态位 EOSMP 置 1 来指示采样阶段结束（仅限常规转换）。EOSMP 标志可通过由软件向其写入 1 的方式来清零。如果 EOSMPIE 位置 1，可产生中断。

25.3.25 转换序列结束（EOS、JEOS）

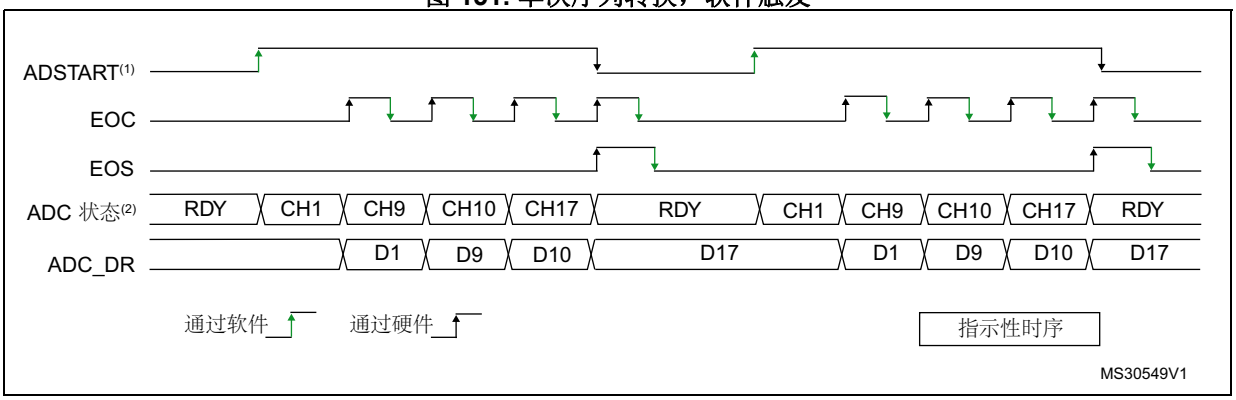
每次出现常规序列结束 (EOS) 事件和注入序列结束 (JEOS) 事件时，ADC 都会通知应用。

常规转换序列的上一数据出现在 ADCx_DR 寄存器中时，ADC 会立即将 EOS 标志置 1。如果 EOSIE 位置 1，可产生中断。EOS 标志可通过由软件向其写入 1 的方式来清零。

注入转换序列的上一数据完成时，ADC 会立即将 JEOS 标志置 1。如果 JEOSIE 位置 1，可产生中断。JEOS 标志可通过由软件向其写入 1 的方式来清零。

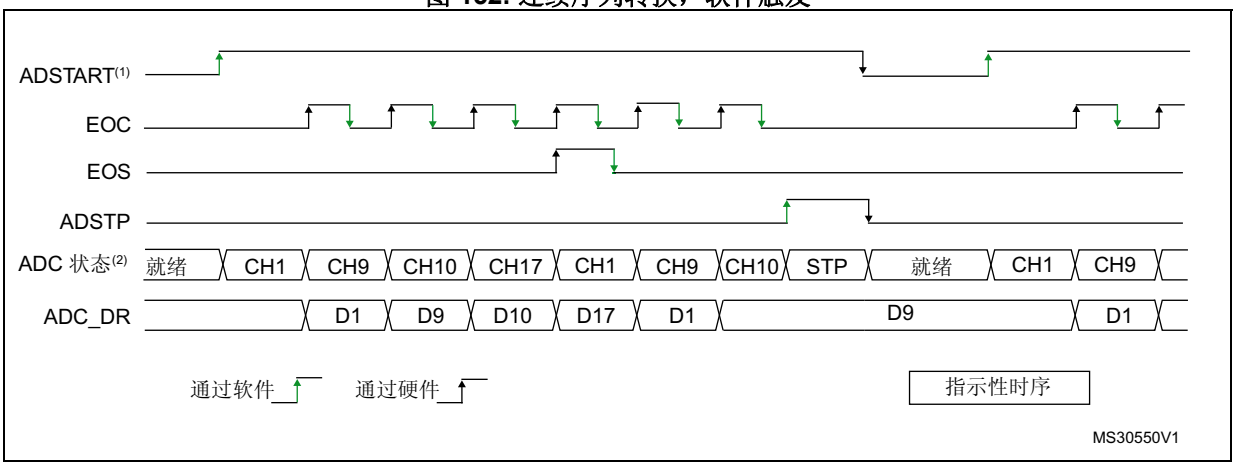
25.3.26 时序图示例（单次模式/连续模式，硬件/软件触发）

图 151. 单次序列转换，软件触发



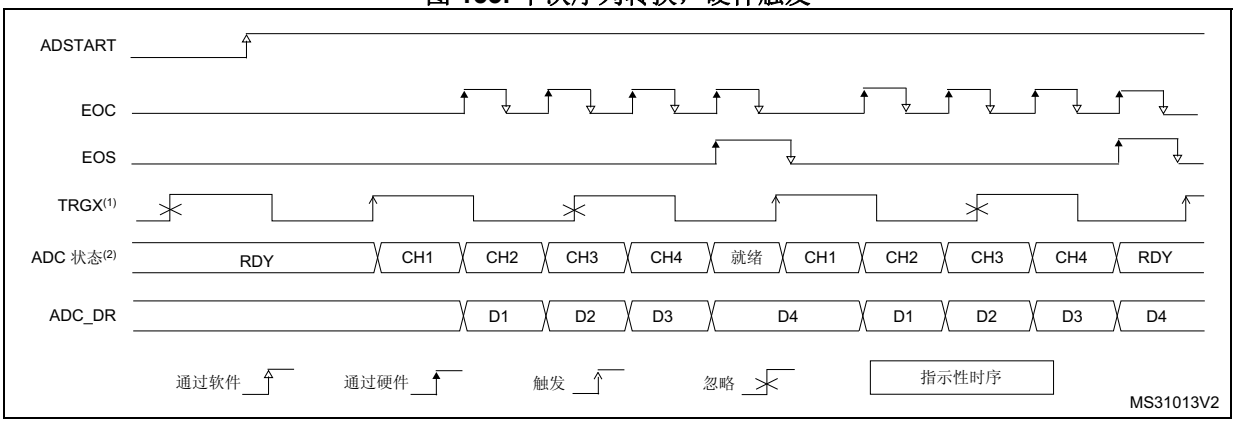
1. EXTEN=0x0，CONT=0
2. 所选通道 = 1、9、10、17；AUTDLY=0。

图 152. 连续序列转换，软件触发



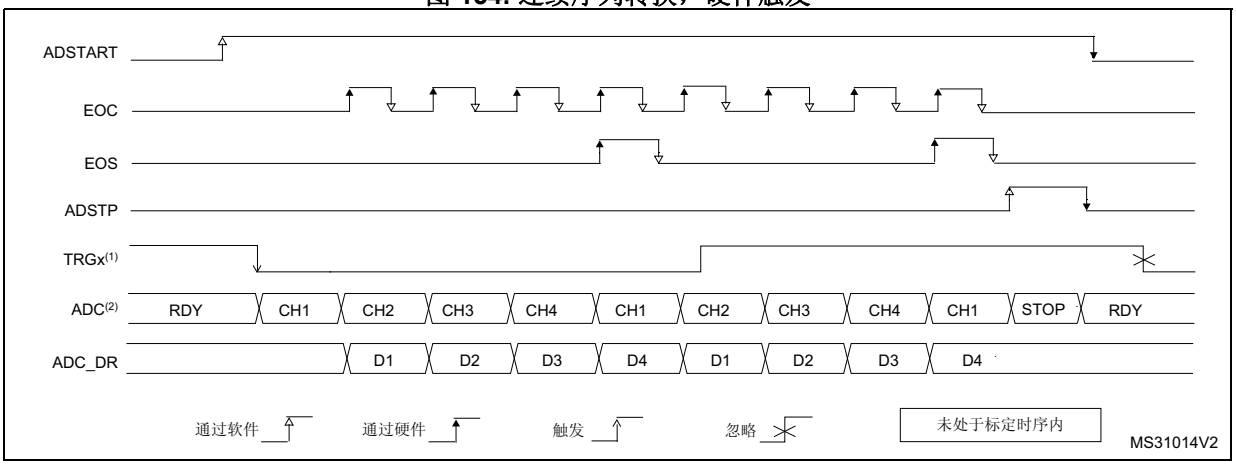
1. EXTEN=0x0，CONT=1
2. 所选通道 = 1、9、10、17；AUTDLY=0。

图 153. 单次序列转换，硬件触发



1. 选择 TRGx（过频）作为触发源，EXTEN = 01，CONT = 0
2. 所选通道 = 1、2、3、4；AUTDLY=0。

图 154. 连续序列转换，硬件触发



1. 选择 TRGx 作为触发源，EXTEN = 10，CONT = 1
2. 所选通道 = 1、2、3、4；AUTDLY=0。

25.3.27 数据管理

数据寄存器、数据对齐和偏移（ADCx_DR、ADCx_JDRy、OFFSETy、OFFSETy_CH、OVSS、LSHIFT、RSHIFT、SSATE）

数据和对齐

每次常规转换通道结束时（发生 EOC 事件时），转换后数据的结果都会存储在宽度为 32 位的 ADCx_DR 数据寄存器中。

每次注入转换通道结束时（发生 JEOC 事件时），转换后数据的结果都会存储在宽度为 32 位的相应 ADCx_JDRy 数据寄存器中。

ADCx_CFGR2 寄存器中的 OVSS[3:0] 和 LSHIFT[3:0] 位域用于选择转换后存储的数据的对齐方式。可选择左对齐和右对齐两种方式，如*图 155*、*图 156*、*图 157* 和*图 158* 所示。

注： *数据可在正常模式和过采样模式下重新对齐。*

偏移

通过在 ADCx_OFRy 寄存器的 OFFSETy[25:0] 位域中编程一个非零值，可对通道应用偏移 y（y=1、2、3、4）。应用偏移的通道会编程到 ADCx_OFRy 寄存器的 OFFSETy_CH[4:0] 位中。这种情况下，转换后的值会减去写入到 OFFSETy[25:0] 位中的用户自定义偏移。25:0].结果可能是负值，因此读取的数据为有符号值，SEXT 位代表扩展符号值。

偏移值应小于最大转换值（例如，16 位模式的最大偏移值为 0xFFFF）。

过采样模式下也支持偏移校准。对于过采样模式，会在应用 OVSS 右移之前减去偏移。

表 195 介绍了如何对模拟看门狗 1、2、3 的所有可能分辨率进行比较。

表 195. 偏移计算与数据分辨率

分辨率（位 RES[2:0]）	原始转换数据与偏移相减：		结果	注释
	原始转换数据，左对齐	偏移		
000：16 位	DATA[15:0]	OFFSET[25:0]	有符号 27 位数据	-
001：14 位	DATA[15:2]，00	OFFSET[25:0]	有符号 27 位数据	用户必须将 OFFSET[1:0] 配置为 00
010：12 位	DATA[15:4]，0000	OFFSET[25:0]	有符号 27 位数据	用户必须将 OFFSET[3:0] 配置为 0000
011：10 位	DATA[15:6]，000000	OFFSET[25:0]	有符号 27 位数据	用户必须将 OFFSET[5:0] 配置为 000000
100：8 位	DATA[15:8]，000000	OFFSET[25:0]	有符号 27 位数据	用户必须将 OFFSET[7:0] 配置为 00000000

从对应于通道“i”的 ADCx_DR（常规通道）或 ADCx_JDRy（注入通道，y=1、2、3、4）读取数据时：

- 如果相应通道的其中一个偏移已使能（位 OFFSETy_EN=1），则读取的数据为有符号数据。
- 如果此通道的四个偏移均未使能，则读取的数据为无符号数据。

图 155、*图 156*、*图 157* 和 *图 158* 介绍了有符号数据和无符号数据的对齐方式以及相应的 OVSS 和 LSHIFT 值。

图 155. 右对齐（偏移禁止，无符号值）

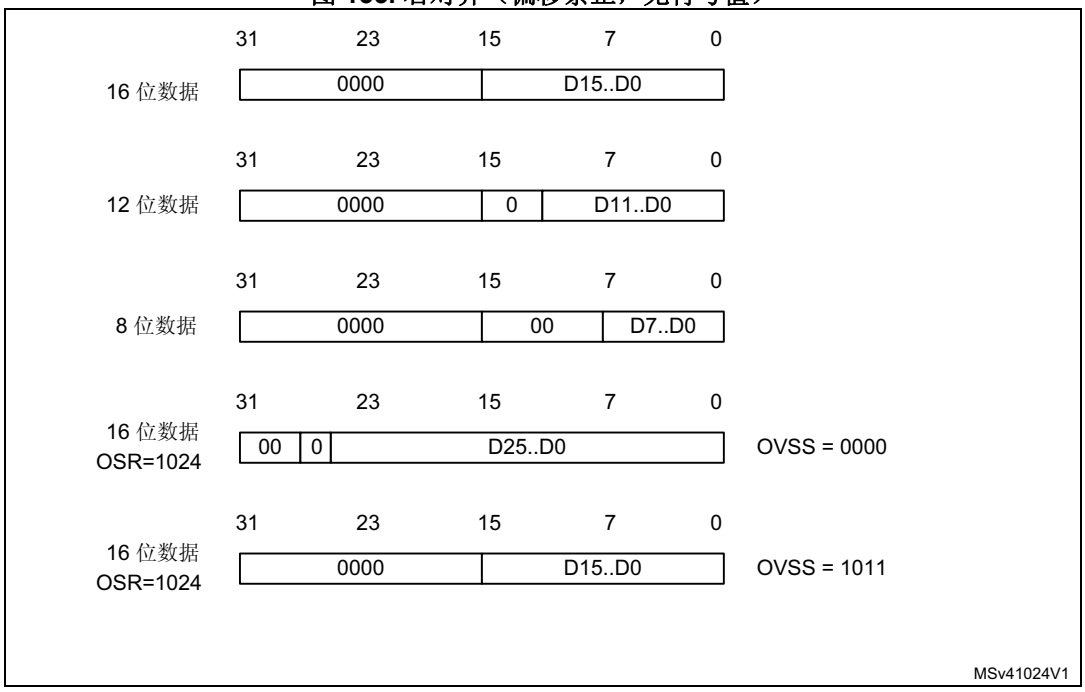


图 156. 右对齐（偏移使能，有符号值）

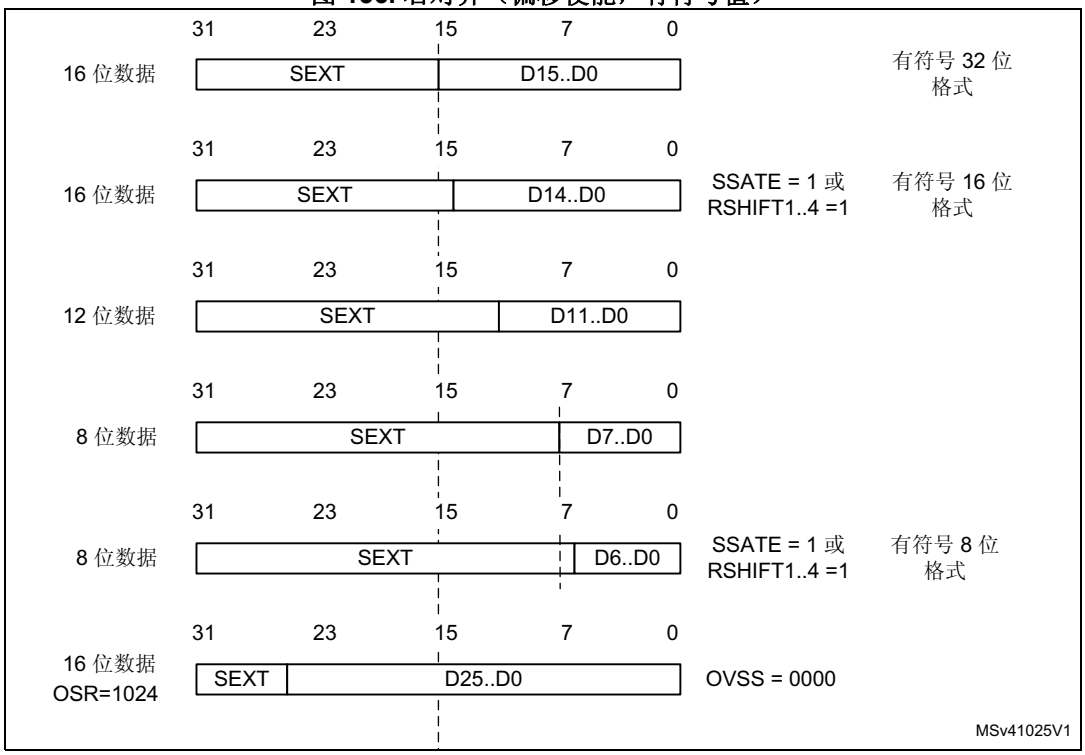


图 157. 左对齐（偏移禁止，无符号值）

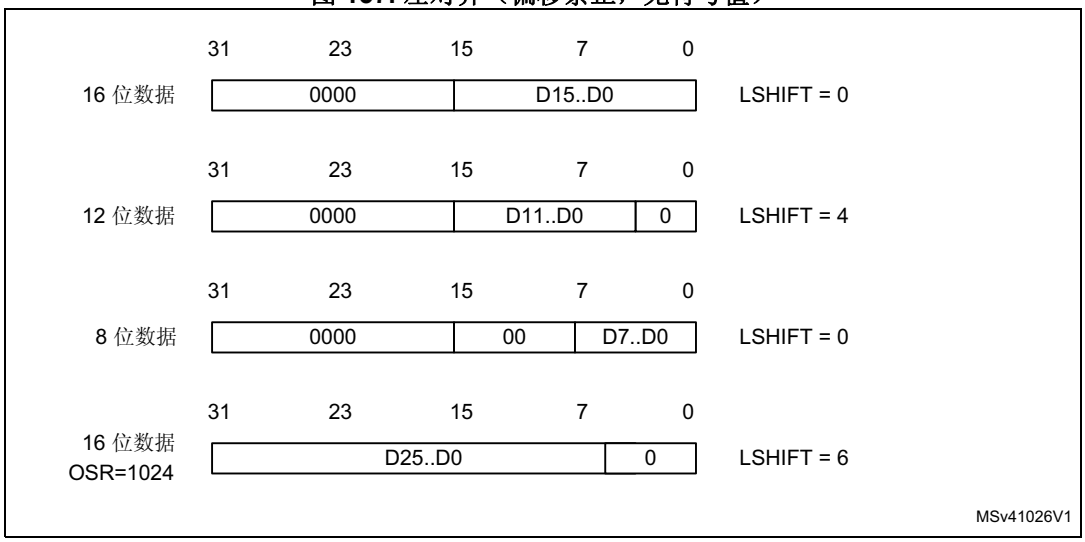
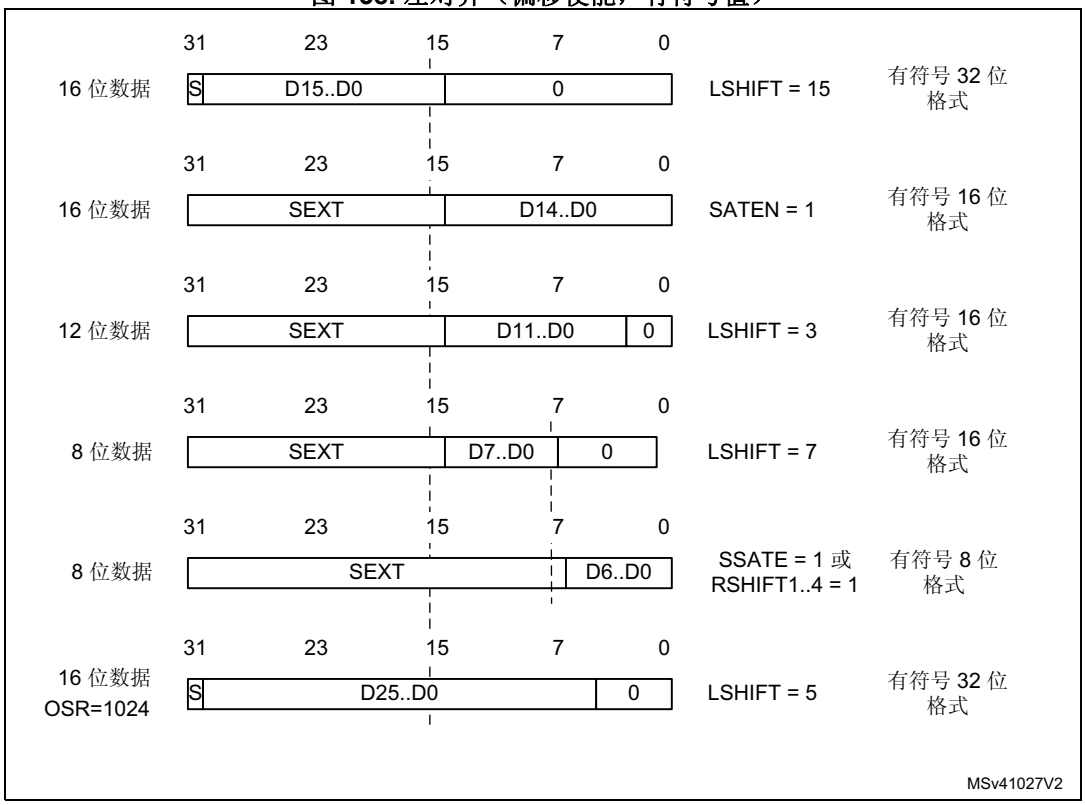


图 158. 左对齐（偏移使能，有符号值）



16 位和 8 位有符号格式管理：RSHIFTx、SSATE

偏移校准符号会扩展数据格式，例如，将无符号 16 位转换扩展为 17 位有符号格式。

可通过三种方式对 8 位和 16 位转换结果进行格式化。

对于偏移校准通道 1 到 4，可通过 ADCx_CFGR2 寄存器中的 RSHIFT1..4 位使结果右移 1 位，成为标准的 8 位或 16 位格式。

另一种方法是使结果饱和，变为 16 位和 8 位有符号格式，但仅适用于以下情况：
RES[2:0] = 000（16 位格式）和 RES[2:0] = 100（8 位格式）。

此模式可使用 ADCx_OFRy 寄存器中的 SSATE 位来使能。

下表汇总了 16 位格式的 3 种可用用例。

表 196. 16 位数据格式

SSATE	RSHIFTx	格式	数据范围 （偏移 = 0x8000）
0	0	符号扩展 17 位有效数据 SEXT[31:16] DATA[15:0]	0x00007FFF - 0x FFFF8000
0	1	符号扩展右移 16 位有效数据 SEXT[31:15] DATA[14:0]	0x3FFF - 0xC000
1	0	符号扩展饱和 16 位有效数据 SEXT[31:15] DATA[14:0]	7FFF - 0x8000
1	1	保留	-

表 197 列出了数字示例以及 3 个不同的偏移值。

表 197. 16 位格式的数字示例（粗体代表饱和）

原始转换结果	偏移值	结果 SSATE = 0 RSHIFT = 0	结果 SSATE = 0 RSHIFT = 1	结果 SSATE = 1 RSHIFT = 0
0xFFFF	0x8000	0x0000 7FFF	3FFF	7FFF
0x8000		0x0000 0000	0	0
0x0000		0xFFFF 8000	C000	8000
0xFFFF	0x8020	0x0000 7FDF	3FEF	7FDF
0x8000		0xFFE0 FFFF	FFF0	FFE0
0x0000		0xFFFF 7FE0	BFF0	8000
0xFFFF	0x7FF0	0x0000 800F	4007	7FFF
0x8000		0x0000 0010	8	0010
0x0000		0xFFFF 8010	C008	8010

过采样模式激活时，不支持 SSATE 和 RSHIFT1..4 位。

ADC 溢出（OVR、OVRMOD）

如果常规转换后的数据未在新转换数据可用之前（由 CPU 或 DMA）读取，会由溢出标志 (OVR) 指示缓冲区溢出事件。

如果新转换完成时 EOC 标志仍为 1，则 OVR 标志会置 1。如果 OVRIE 位置 1，可产生中断。

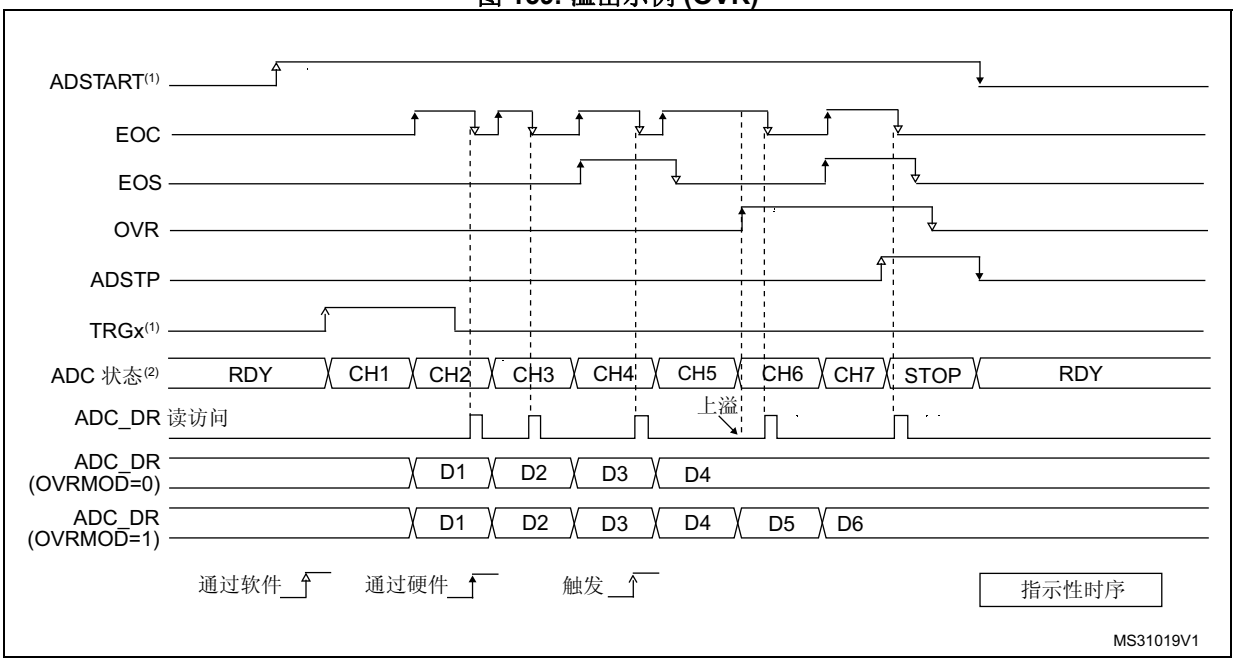
如果发生溢出情况，ADC 仍会保持工作状态并可继续进行转换，除非通过软件将 ADSTP 位置 1，从而停止并复位序列。

OVR 标志可通过由软件向其写入 1 的方式来清零。

可对控制位 OVRMOD 进行编程，从而配置发生溢出事件时是要保留数据还是要覆盖数据：

- OVRMOD=0：溢出事件会保留数据寄存器的数据，防止其被覆盖：会保留原数据，并会丢弃新的转换结果。如果 OVR 保持为 1，可继续进行转换，但还是会丢弃结果数据。
- OVRMOD=1：数据寄存器会由上一次转换结果覆盖，之前未读取的数据会丢失。如果 OVR 保持为 1，可继续正常进行转换，并且 ADCx_DR 寄存器将始终包含最新的转换数据。

图 159. 溢出示例 (OVR)



注： *由于四条注入通道均有专用的数据寄存器，因此不会对注入通道进行溢出检测。*

在不使用 DMA 的情况下管理转换序列

如果转换过程足够慢，则可使用软件来处理转换序列。在这种情况下，软件必须使用 EOC 标志及其相关中断来处理各个数据。每当转换结束时，EOC 都会置 1，并且可以读取 ADCx_DR 寄存器。OVRMOD 应配置为 0，以便将溢出事件作为错误进行管理。

在不使用 DMA 且不发生溢出的情况下管理转换

ADC 在转换一个或多个通道时不是每次都读取数据的情况下，这可能会很有用（例如，存在模拟看门狗时）。在这种情况下，OVRMOD 位必须配置为 1，OVR 标志应被软件忽略。溢出事件不会阻止 ADC 继续进行转换，ADCx_DR 寄存器始终包含最新的转换。

使用 DMA 管理转换

由于转换得出的通道值会存储到唯一的数据寄存器中，因此，对于多个通道的转换，使用 DMA 非常有帮助。这样可以避免丢失在下一次写入之前还未被读出的 ADCx_DR 寄存器中的数据。

DMA 模式使能时（单 ADC 模式下 ADCx_CFGR 寄存器中的 DMNGT 位 = 01 或 11，或者双重 ADC 模式下 MDMA 不为 0b00），会在每次通道转换后生成 DMA 请求。这样便可将转换的数据从 ADCx_DR 寄存器传输到用软件选择的目标位置。

尽管如此，如果因 DMA 无法及时处理 DMA 传输请求而发生溢出 (OVR=1)，ADC 会停止生成 DMA 请求，新转换对应的数据不会通过 DMA 进行传输。这意味着可将传输到 RAM 的所有数据都视为有效数据。

根据 OVRMOD 位的配置，可保留或覆盖数据（请参见 *ADC 溢出（OVR、OVRMOD）*一节）。

DMA 传输请求会禁止，直至软件将 OVR 位清零。

根据应用用途的不同，推荐使用两种不同的 DMA 模式（在单 ADC 模式下，使用 ADCx_CFGR 寄存器中的 DMNGT 位配置；在双重 ADC 模式下，使用 ADCx_CCR 寄存器中的 DAMDF 位配置）：

- DMA 单次模式（DMNGT 位 = 01）。
  如果通过编程将 DMA 设置为传输固定数目的数据，应选择此模式。
- DMA 循环模式（DMNGT 位 = 11）
  如果在循环模式下对 DMA 进行编程，应选择此模式。

DMA 单次模式 (DMNGT=01)

在该模式下，每次出现新的转换数据时，ADC 都会生成 DMA 传输请求，DMA 到达最后一个 DMA 传输操作时（发生 DMA_EOT 中断，请参见 DMA 一段），即使转换已再次开始，ADC 也会停止生成 DMA 请求。

DMA 传输完成后（在 DMA 控制器中配置的所有传输操作均已完成后）：

- ADC 数据寄存器的内容会冻结。
- 任何正在进行的转换都会中止，其部分结果会被丢弃。
- 不会将任何新的 DMA 请求发送到 DMA 控制器。如果仍存在已开始的转换，这样可避免生成溢出错误。
- 扫描序列会停止并复位。
- DMA 会停止。

DMA 循环模式 (DMNGT=11)

在该模式下，每次数据寄存器中出现新的转换数据时，ADC 都会生成 DMA 传输请求，即使 DMA 已到达最后一次 DMA 传输操作也不例外。这样可将处于循环模式的 DMA 配置为处理连续的模拟输入数据流。

基于 FIFO 的 DMA

输出数据寄存器包括 8 级 FIFO，支持同时生成两个不同的 DMA 请求。当有 1 个数据可用时，会生成“SREQ 单次请求”；当有 4 个数据可用时，会生成“BREQ 突发请求”。DMA2 可编程为单次传输模式或增量突发模式（4 个节拍），DMA2 会根据具体模式选择正确的请求线。更多信息，请参见 DMA2 一章。

25.3.28 使用 DFSDM 管理转换

ADC 转换结果可直接传送到具有 ΣΔ 调制器的数字滤波器 (DFSDM)。

这种情况下，DMNGT[1:0] 位必须设置为 10。

ADC 会将常规数据寄存器数据的 16 个最低有效位传输到 DFSDM 中，传输生效后，会立即复位 EOC 标志。

数据必须为 16 位有符号格式：

ADCx_DR[31:16] = 无关位

ADCx_DR[15] = 符号

ADCx_DR[14:0] = 数据

任何超出 16 位有符号格式的值将会被截断。

25.3.29 动态低功耗特性

自动延迟转换模式 (AUTDLY)

ADC 会执行由 AUTDLY 配置位控制的自动延迟转换模式。自动延迟转换可用于简化软件，并可优化采用低频时钟的应用（此类应用可能存在 ADC 溢出的风险）的性能。

AUTDLY=1 时，仅当同一组内所有之前的数据已处理完毕时，才能开始新的转换：

- 对于常规转换：ADCx_DR 寄存器已读取或者 EOC 位已清零时（请参见*图 160*）。
- 对于注入转换：JEOS 位已清零时（请参见*图 161*）。

通过这种方式，可自动调整 ADC 的速度，使其适应系统读取数据的速度。

延时插入到每次常规转换后（无论 DISCEN=0 还是 1）和每个注入转换序列后（无论 JDISCEN=0 还是 1）。

注： *不会在注入转换序列之间插入延时，但会在最后一个序列之后插入延时。*

转换过程中，此延时期间发生的硬件触发事件（针对同一组转换）会被忽略。

注： *如果延时期间发生软件触发，则不会被忽略，仍可在该延时时间内通过将 ADSTART 或 JADSTART 位置 1 的方式重新启动转换；启动新转换之前，会由软件读取数据。*

不会在不同组的转换之间插入延时（常规转换后即进行注入转换，反之亦然）：

- 如果在常规转换的自动延时期间发生注入触发，注入转换会立即开始（请参见*图 161*）。
- 注入序列完成后，ADC 会等待上一常规转换的延时（如果未结束），然后才会启动新的常规转换（请参见*图 163*）。

自动注入模式 (JAUTO=1) 下的操作略有不同，仅当上一注入转换序列的自动延时已结束时（JEOS 已清零时），才能开始进行新的常规转换。这是为了确保软件可在启动新序列之前读取给定序列的所有数据（请参见*图 164*）。

要在连续自动注入和自动延时组合模式下停止转换（JAUTO=1、CONT=1 且 AUTDLY=1），请按照以下程序进行操作：

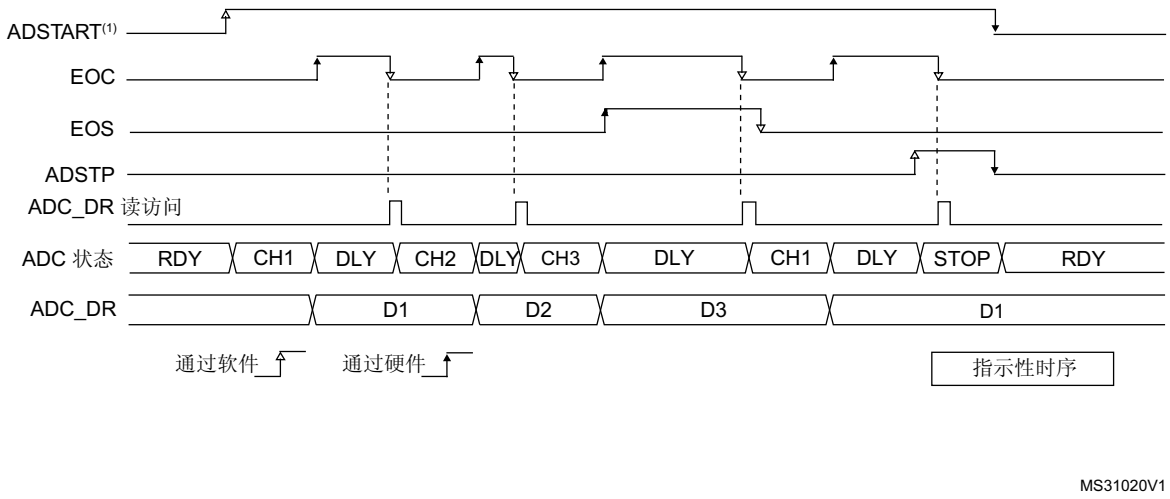
1. 等待，直至 JEOS=1（不会重新启动任何转换）
2. 将 JEOS 清零 (Clear NAK)
3. 将 ADSTP 置 1
4. 读取常规数据。

否则，如果 JEOS 在 ADSTP 已置 1 后清零，可能重新启动新的常规序列。

在 AUTDLY 模式下，如果在正在进行的常规序列期间或序列的最后一次常规转换之后的延时期间发生硬件常规触发事件，则会忽略该触发事件。但是，如果触发事件发生在该延时之后，即使发生在延时之后的注入序列期间，也会将该触发事件视为待处理事件。随后，会在注入序列的延时结束时开始转换。

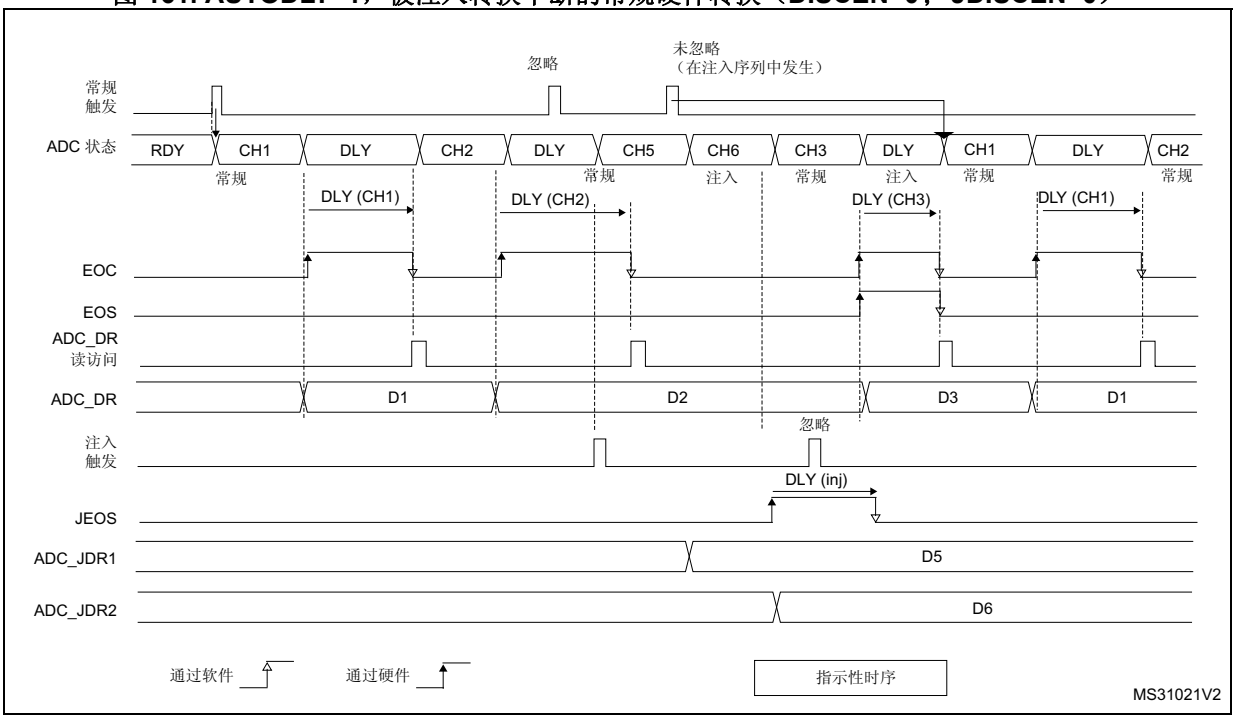
在 AUTDLY 模式下，如果在正在进行的注入序列期间或序列的最后一次注入转换之后的延时期间发生硬件注入触发事件，则会忽略该触发事件。

图 160. AUTODLY=1，连续模式下的常规转换，软件触发



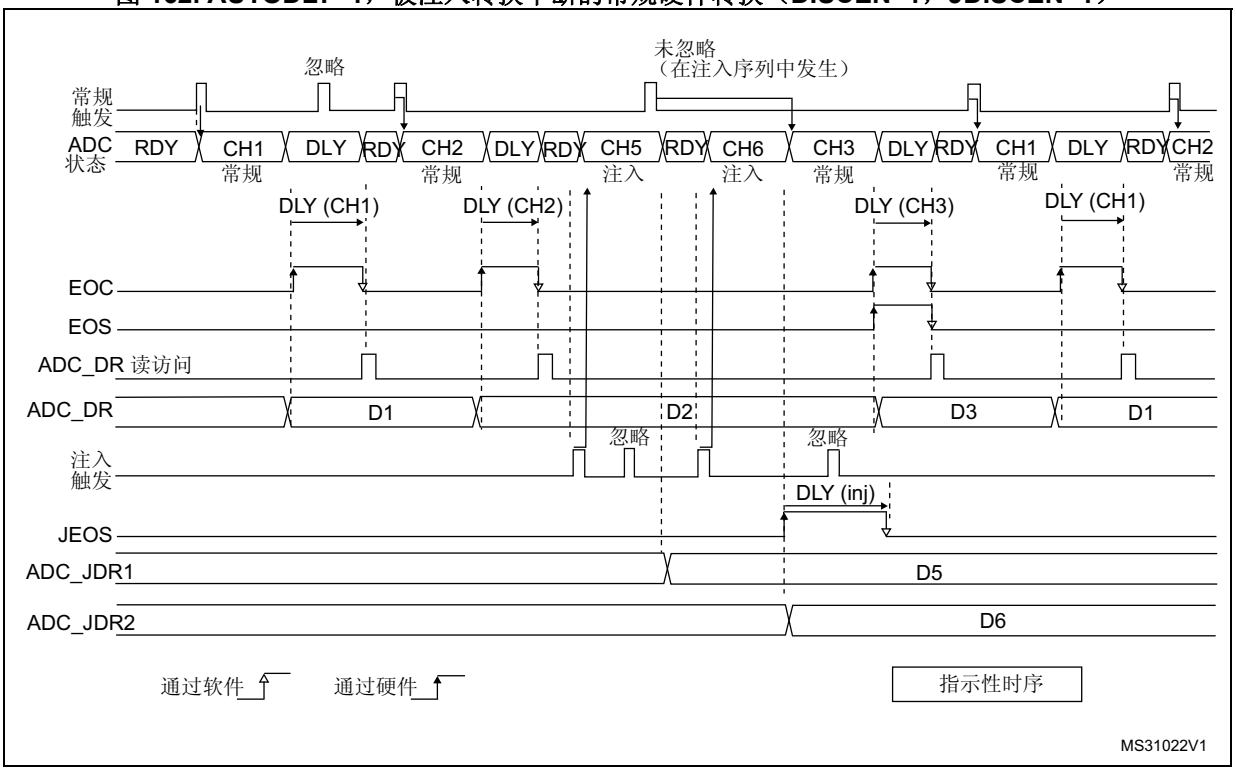
1. AUTDLY=1
2. 常规配置：EXTEN=0x0（软件触发），CONT=1，CHANNELS = 1、2、3
3. 注入配置禁止

图 161. AUTODLY=1，被注入转换中断的常规硬件转换（DISCEN=0；JDISCEN=0）



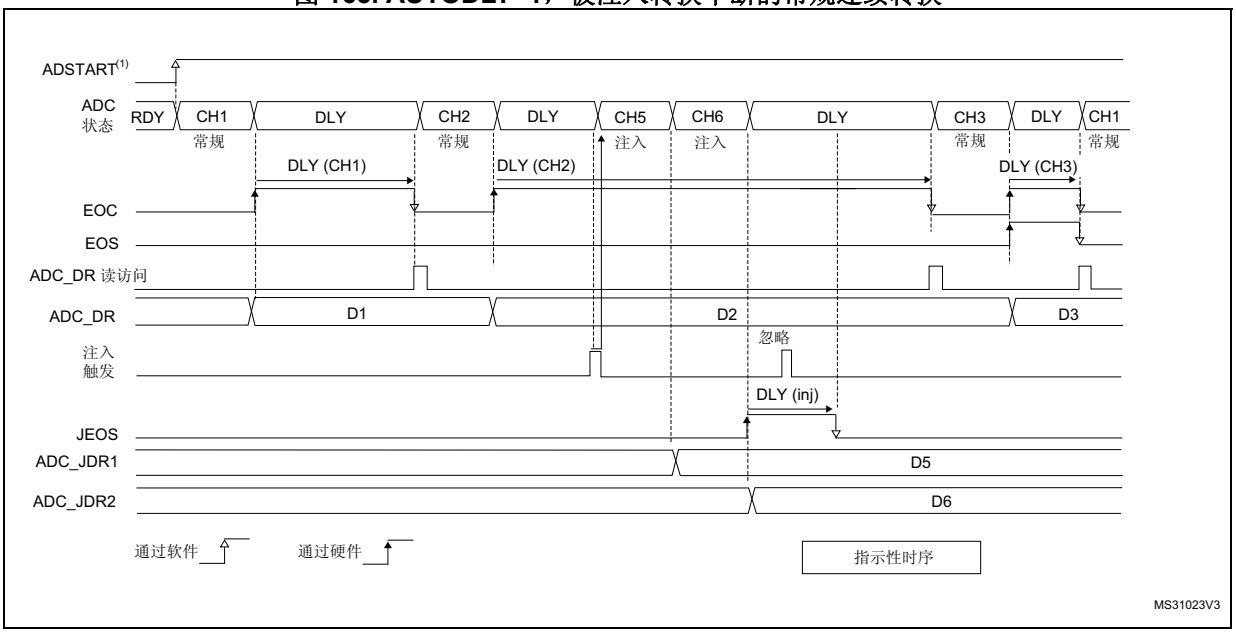
1. AUTDLY=1
2. 常规配置：EXTEN=0x1（硬件触发），CONT=0，DISCEN=0，CHANNELS = 1、2、3
3. 注入配置：JEXTEN=0x1（硬件触发），JDISCEN=0，CHANNELS = 5、6

图 162. AUTODLY=1，被注入转换中断的常规硬件转换（DISCEN=1，JDISCEN=1）



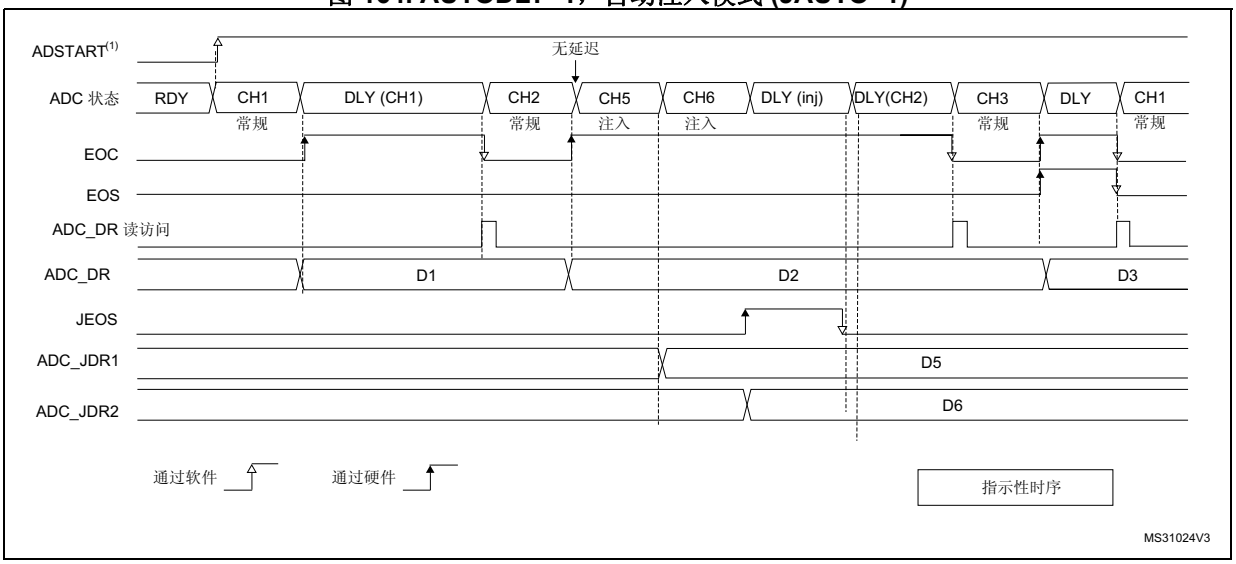
1. AUTDLY=1
2. 常规配置：EXTEN=0x1（硬件触发），CONT=0，DISCEN=1，DISCNUM=1、CHANNELS = 1、2、3。
3. 注入配置：JEXTEN=1x1（硬件触发），JDISCEN=0，CHANNELS = 5、6

图 163. AUTODLY=1，被注入转换中断的常规连续转换



1. AUTDLY=1
2. 常规配置：EXTEN=0x0（软件触发），CONT=0，DISCEN=0，CHANNELS = 1、2、3
3. 注入配置：JEXTEN=0x1（硬件触发），JDISCEN=0，CHANNELS = 5、6

图 164. AUTODLY=1，自动注入模式 (JAUTO=1)

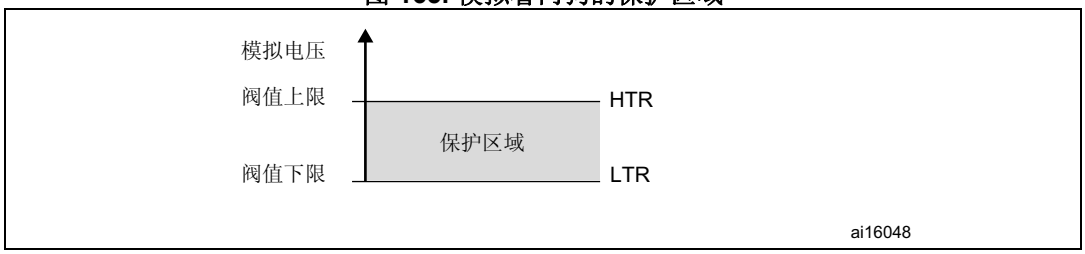


1. AUTDLY=1
2. 常规配置：EXTEN=0x0（软件触发），CONT=0，DISCEN=0，CHANNELS = 1、2
3. 注入配置：JAUTO=1，CHANNELS = 5、6

25.3.30 模拟窗口看门狗（AWD1EN、JAWD1EN、AWD1SGL、AWD1CH、AWD2CH、AWD3CH、AWD_HTRy、AWD_LTRy、AWDy）

三个 AWD 模拟看门狗会监测一些通道是否保持在配置的电压范围（窗口）内。

图 165. 模拟看门狗的保护区域



AWDx 标志和中断

可通过将 ADCx_IER 寄存器（x=1、2、3）中的 AWDyIE 置 1 的方式分别为 3 个模拟看门狗使能中断。

AWDy（y=1、2、3）标志可通过由软件向其写入 1 的方式来清零。

在对齐之前，会将 ADC 转换结果与阈值上限和下限进行比较。

模拟看门狗 1 说明

将 ADCx_CFGR 寄存器中的 AWD1EN 位置 1，可使能 AWD 模拟看门狗 1。该看门狗监测一条已选通道或所有已使能通道(1) 是否仍在配置的电压范围（窗口）内。

表 198 介绍了应如何配置 ADCx_CFGRy 寄存器才能在一条或多个通道上使能模拟看门狗。

表 198. 模拟看门狗通道选择

模拟看门狗保护的通道	AWD1SGL 位	AWD1EN 位	JAWD1EN 位
无	x	0	0
所有注入通道	0	0	1
所有常规通道	0	1	0
所有常规通道和注入通道	0	1	1
单个(1)注入通道	1	0	1
单个(1)常规通道	1	1	0
单个(1)常规通道或注入通道	1	1	1

1. 通过 AWDyCH[4:0] 位选择 此外，还必须将通道编程为在合适的常规序列或注入序列中进行转换。

如果 ADC 转换的模拟电压低于阈值下限或高于阈值上限，则 AWD1 模拟看门狗状态位会置 1。

这些阈值编程到模拟看门狗 1 的 ADCx_HTR1 寄存器的 HTR1[25:0] 位和 ADCx_LTR1 寄存器的 LTR1[25:0] 位。

阈值最高可达到 26 位（16 位分辨率，过采样，OSR=1024）。

如果转换的数据分辨率小于 16 位（取决于 RES[2:0] 位），编程阈值的 LSB 必须保持清零状态，并且会在内部对完整的 16 位转换数据进行比较（左对齐到半字边界）。

表 199 介绍了如何对模拟看门狗 1、2、3 的所有可能的分辨率进行比较。

表 199. 模拟看门狗 1、2、3 比较

分辨率（位 RES[2:0]）	模拟看门狗比较对象：		注释
	原始转换数据，左对齐(1)	阈值	
000：16 位	DATA[15:0]	LTR1[25:0] 和 HTR1[25:0]	-
001：14 位	DATA[15:2]，00	LTR1[25:0] 和 HTR1[25:0]	用户必须将 LTR1[1:0] 和 HTR1[1:0] 配置为 00
010：12 位	DATA[15:4]，0000	LTR1[25:0] 和 HTR1[25:0]	用户必须将 LTR1[3:0] 和 HTR1[3:0] 配置为 0000
011：10 位	DATA[15:6]，000000	LTR1[25:0] 和 HTR1[25:0]	用户必须将 LTR1[5:0] 和 HTR1[5:0] 配置为 000000
100：8 位	DATA[15:8]，00000000	LTR1[25:0] 和 HTR1[25:0]	用户必须将 LTR1[7:0] 和 HTR1[7:0] 配置为 00000000

1. 进行任何对齐计算以及应用任何偏移之前，会对原始转换数据进行看门狗比较（进行比较的数据是无符号数据）。

模拟看门狗 2 和 3 说明

第二个和第三个模拟看门狗更加灵活，可通过编程 AWDCHy[19:0]（y=2、3）中的相应位来保护多条已选通道。

AWDCHy[19:0]（y=2、3）的任何位置 1 时，会使能相应的看门狗。

阈值最高可达到 26 位（16 位分辨率，过采样，OSR=1024），通过 ADCx_HTR2、ADCx_LTR2、ADCx_LTR3 和 ADCx_HTR3 寄存器进行编程。

如果转换的数据分辨率小于 16 位（取决于 RES[2:0] 位），编程阈值的 LSB 必须保持清零状态，并且会在内部对完整的 16 位转换数据进行比较（左对齐到半字边界）。

ADCx_AWDy_OUT 信号输出生成

每个模拟看门狗都关联到一个内部硬件信号 ADCx_AWDy_OUT（x = ADC 编号，y = 看门狗编号），该信号直接连接到一些片上定时器的 ETR 输入（外部触发）。请参见片上定时器部分以了解如何选择 ADCx_AWDy_OUT 信号作为 ETR。

当关联的模拟看门狗使能时，ADCx_AWDy_OUT 会激活：

- 当受保护的转换超出编程阈值时，ADCx_AWDy_OUT 会置 1。
- 在编程阈值范围内的下一受保护转换结束后，ADCx_AWDy_OUT 会复位（如果下一受保护转换仍超出编程阈值范围，该位仍保持置 1）。
- 禁止 ADC 时（将 ADDIS 置 1 时），ADCx_AWDy_OUT 也保持复位状态。请注意，停止常规转换或注入转换（将 ADSTP 或 JADSTP 置 1）对 ADCy_AWDx_OUT 的生成没有任何影响。

注： *AWDx 标志由硬件置 1，并由软件复位：AWDy 标志对 ADCx_AWDy_OUT 的生成没有影响（例如：如果软件未将 AWDx 标志清零，即 AWDx 标志会保持为 1，此时 ADCy_AWDy_OUT 仍可翻转）。*

图 166. ADCy_AWDx_OUT 信号生成（所有常规通道上）

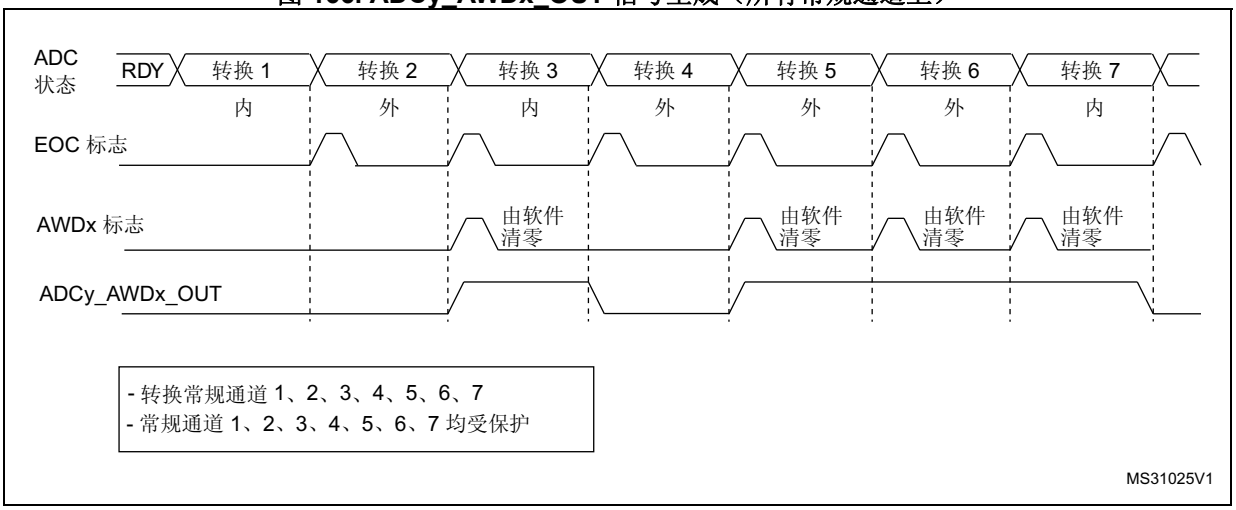


图 167. ADCy_AWDx_OUT 信号生成（AWDx 标志未通过软件清零）

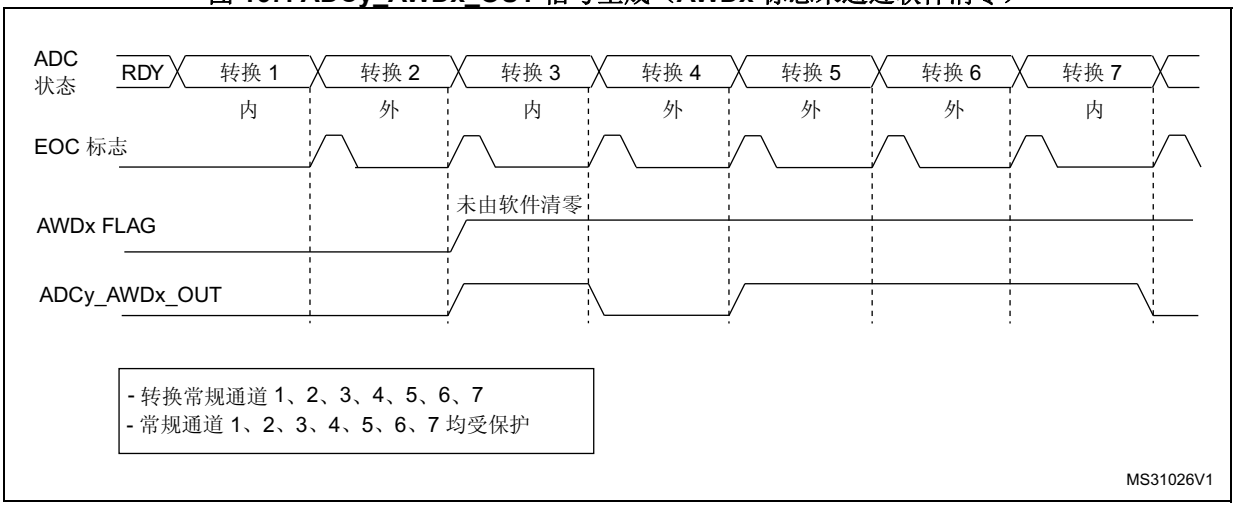


图 168. ADCy_AWDx_OUT 信号生成（单条常规通道上）

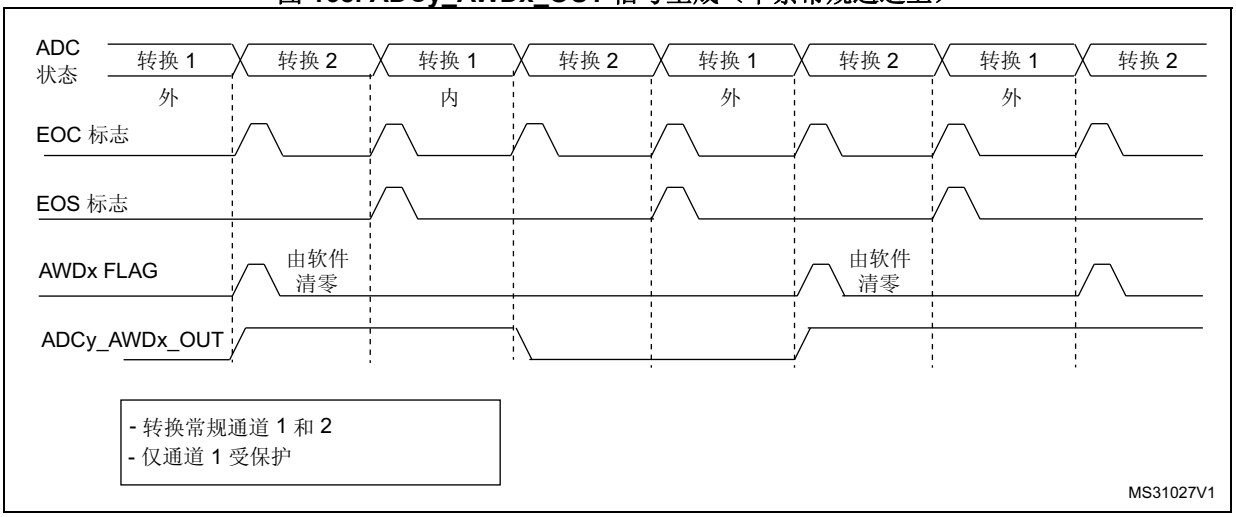
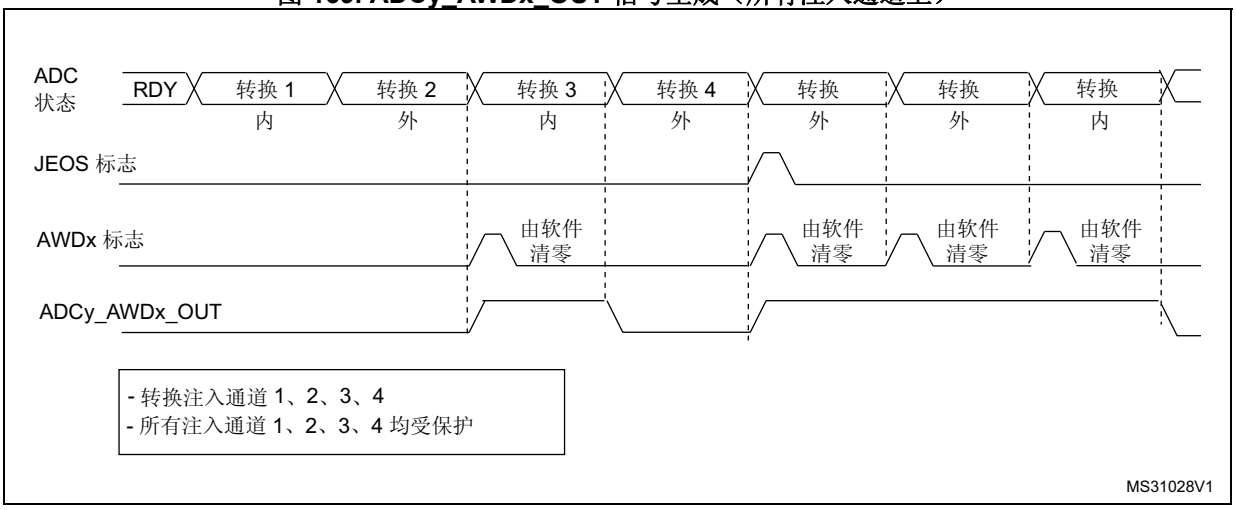


图 169. ADCy_AWDx_OUT 信号生成（所有注入通道上）



25.3.31 过采样器

过采样单元会进行数据预处理，以减轻 CPU 的负担。过采样单元还能处理多个转换，并计算多个转换结果的平均值，得到数据宽度增大（高达 26 位）的单个数据（16 位值，OSR = 1024）。

它提供的结果采用以下形式，其中的 N 和 M 可以进行调整：

$$\text{结果} = \frac{1}{M} \times \sum_{n=0}^{n=N-1} \text{转换}(t_n)$$

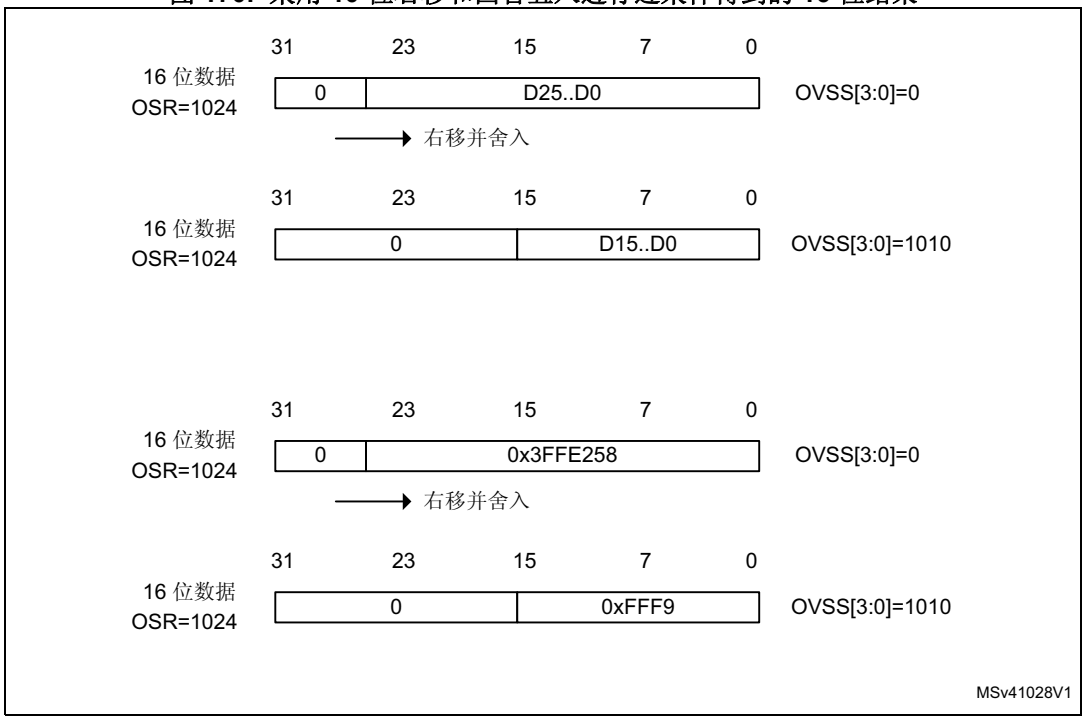
允许通过硬件执行以下功能：计算平均值、降低数据速率、改进 SNR 以及基本滤波。

过采样率 N 通过 ADCx_CFGR2 寄存器中的 OSR[9:0] 位进行定义，范围为 2x 到 1024x。分频系数 M 通过向右移位来实现（最多可移 10 位），并且通过 ADCx_CFGR2 寄存器中的 OVSS[3:0] 位定义。

求和单元可得出多达 26 位（1024 x 16 位结果）的结果，结果可左移或右移。如果选择右移，则会使用移位后剩下的最低有效位四舍五入为最接近的数值，然后将得到的结果传输到 ADCx_DR 数据寄存器中。

图 170 介绍了从原始的 26 位累加数据到最终 16 位结果的数值处理过程。

图 170. 采用 10 位右移和四舍五入进行过采样得到的 16 位结果



过采样模式下的转换时序不会发生变化：在整个过采样序列中，采样时间保持不变。每完成 N 次转换都会提供新数据，等效延迟等于 N x T_{CONV} = N x (t_{SMPL} + t_{SAR})。各标志的置位情况如下：

- 每个采样阶段后都会将采样阶段结束标志 (EOSMP) 置 1
- 如果过采样结果可用，每完成 N 次转换都会发生转换结束事件 (EOC)
- 过采样数据序列完成后（即 N x 序列长度次转换之后），会发生序列结束事件 (EOS)

过采样时支持单 ADC 工作模式

在过采样模式下，大部分 ADC 工作模式都会保留：

- 单次转换或连续模式转换
- 可由软件或触发器启动 ADC 转换
- ADC 在转换过程中停止（中止）
- 通过 CPU 或 DMA 在支持溢出检测的情况下读取数据
- 低功耗模式 (AUTDLY)
- 可编程分辨率：在这种情况下，会按照与 16 位转换相同的方式对分辨率降低的转换值（根据 ADCx_CFGR 寄存器中的 RES[2:0] 位）进行累加、截断、四舍五入和移位。

注： *处理过采样数据时，不可使用对齐模式。ADCx_CFGR1 中的 ALIGN 位会被忽略，且数据始终采用右对齐格式。*

注： *过采样模式下不支持偏移校准。如果 ROVSE 和/或 JOVSE 位置 1，ADCx_OFRy 寄存器中 OFFSETy_EN 位的值会被忽略（视为复位）。*

模拟看门狗

模拟看门狗功能会保留（AWDSGL 位和 AWDEN 位），但存在以下区别：

- 会忽略 RES[2:0] 位，始终会使用完整的 26 位值 HTRx[25:0] 和 LTRx[25:0] 进行比较。
- 在移位之前会对过采样累加值进行比较。

注： *移位位数较大时必须多加留意，因为这样会缩小比较范围。例如，如果过采样结果移了 4 位，得到 12 位右对齐数据，那么只能对 8 个数据位执行有效的模拟看门狗比较。比较操作会在 ADCx_DR[11:4] 与 HT[0:7]/LT[0:7] 之间进行，并且 HT[11:8]/LT[11:8] 必须保持复位状态。*

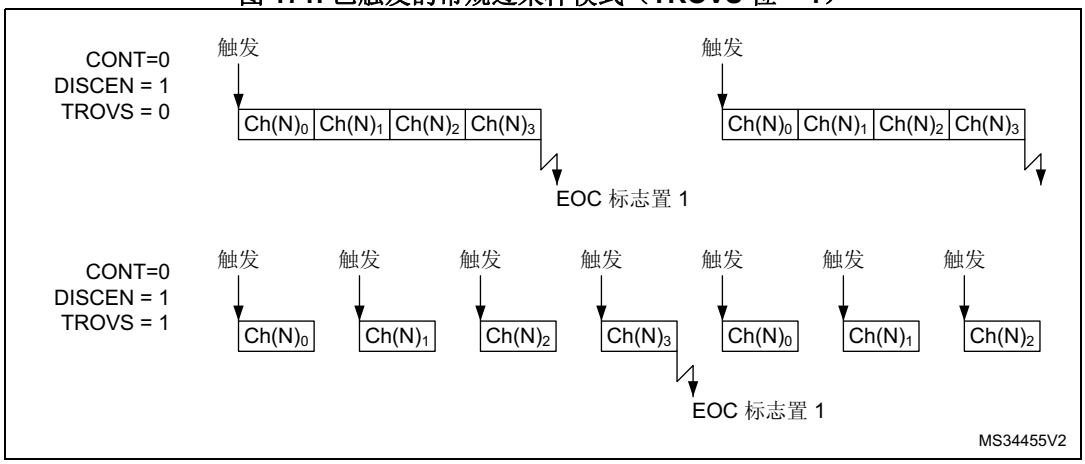
触发模式

平均值计算单元还可用于基本滤波，虽然它不是非常强大的滤波器（衰减缓慢、停止频段衰减受限），但可用作陷波滤波器，用于抑制恒定的寄生频率（通常来自电源或切换模式电源）。为此，可使用 ADCx_CFGR2 中的 TROVS 位使能特定的不连续模式，以获得由用户定义、且与转换时间本身无关的过采样频率。

图 171 显示了在不连续模式下如何响应触发从而开始转换。

如果 TROVS 位置 1，则会忽略 DISCEN 位的内容，并将该位视为 1。

图 171. 已触发的常规过采样模式（TROVS 位 = 1）



过采样时的注入和常规定序器管理

在过采样模式下，注入定序器和常规定序器可以执行不同操作。如果两个定序器必须同时使用，则可为它们使能过采样并设定一些限制条件（与唯一的累加单元相关）。

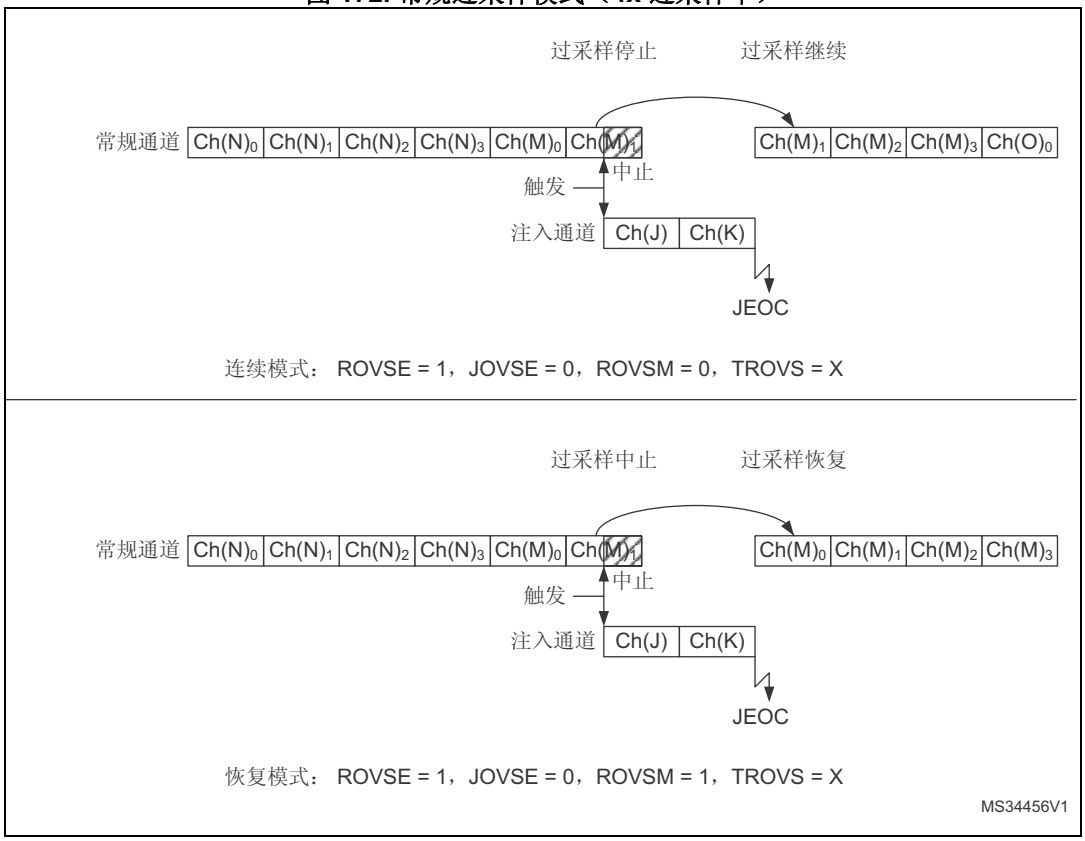
仅对常规通道进行过采样

常规过采样模式位 ROVSM 定义了常规过采样序列在被注入转换中断的情况下如何恢复。

- 在连续模式下，会从上一有效数据开始重新累加（在由于注入触发而发出的转换中止请求之前）。这样可确保在任何注入频率下均可而完成过采样（假设触发之间至少可完成一次常规转换）；
- 在恢复模式下，会从 0 开始重新累加（会忽略之前的转换结果）。该模式可确保所有用于过采样的数据在单个时隙内进行了背靠背转换。需要注意的是，注入触发周期必须超过过采样时长。如果该条件未得到满足，将无法完成过采样，常规定序器将被禁用。

图 172 以 4x 过采样率为例进行了说明。

图 172. 常规过采样模式（4x 过采样率）



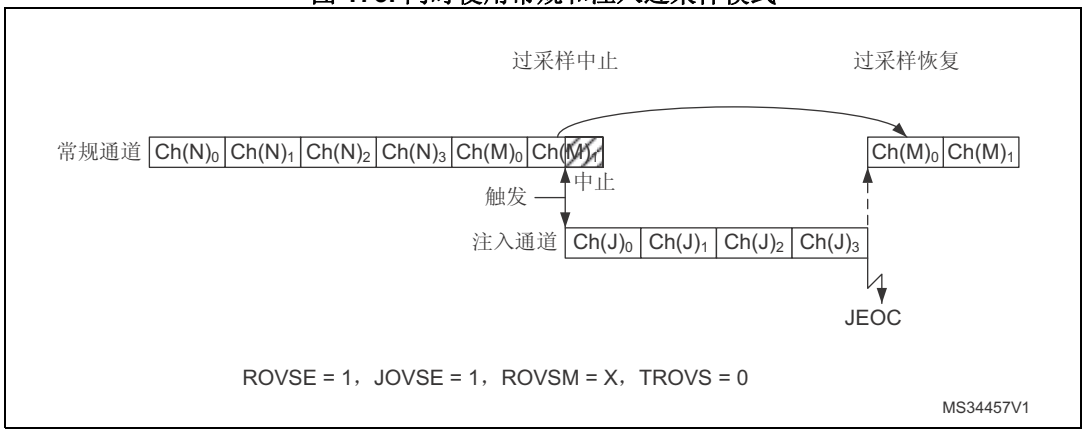
仅对注入通道进行过采样

注入过采样模式位 JOVSE 专为注入定序器中的转换使能过采样。

对常规通道和注入通道进行过采样

可以将 ROVSE 和 JOVSE 位都置 1。在这种情况下，常规过采样模式会强制进入恢复模式（忽略 ROVSM 位），如*图 173* 所示。

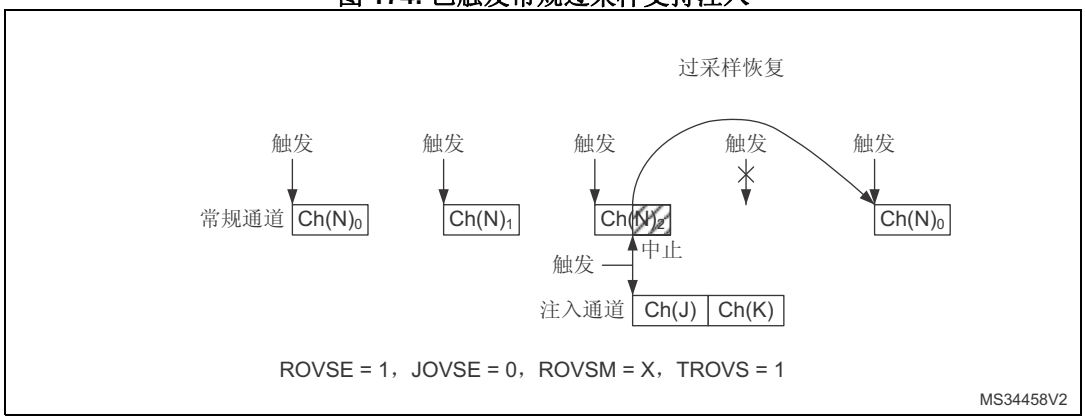
图 173. 同时使用常规和注入过采样模式



已触发常规过采样支持注入转换

已触发常规模式下支持注入转换。在这种情况下，必须禁止注入过采样模式，并且会忽略 ROVSM 位（强制进入恢复模式）。JOVSE 位必须复位。具体操作如*图 174* 所示。

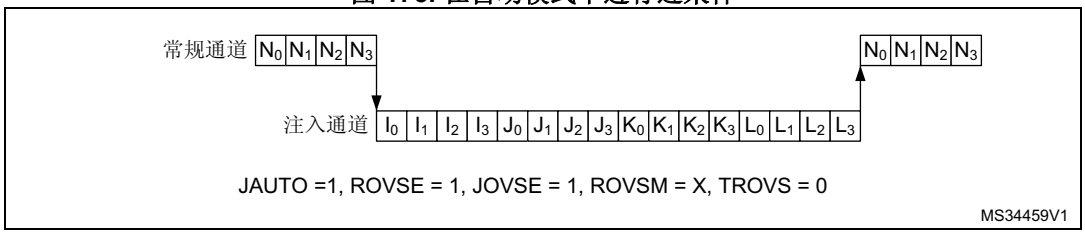
图 174. 已触发常规过采样支持注入



自动注入模式

可以对自动注入序列进行过采样，并将所有转换结果存储在寄存器中以保存 DMA 资源。使用该模式的前提条件是常规和注入过采样均激活：JAUTO = 1、ROVSE = 1 且 JOVSE = 1，不支持其他组合。在自动注入模式下，会忽略 ROVSM 位。*图 175* 显示了转换的排序方式。

图 175. 在自动模式下进行过采样



此外，还可以使用 TROVS 位使能触发模式。在这种情况下，ADC 必须进行如下配置：JAUTO=1、DISCEN=0、JDISCEN=0、ROVSE=1、JOVSE=1 且 TROVSE=1。

过采样时支持双重 ADC 模式

对于注入同步模式和常规同步模式，可在以双重 ADC 配置运行时使能过采样。在这种情况下，必须将两个 ADC 编程为使用完全相同的设置（包括过采样）。

使能常规过采样或注入过采样时（ROVSE = 1 或 JOVSE = 1），不支持所有其他双重 ADC 模式。

组合模式汇总

表 200 汇总了所有组合，包括不支持的模式。

表 200. 过采样器工作模式汇总

常规过采样 ROVSE	注入过采样 JOVSE	过采样器模式 ROVSM 0 = 连续 1 = 恢复	触发常规模式 TROVS	注释
1	0	0	0	常规连续模式
1	0	0	1	不支持
1	0	1	0	常规恢复模式
1	0	1	1	触发常规恢复模式
1	1	0	X	不支持
1	1	1	0	注入和常规恢复模式
1	1	1	1	不支持
0	1	X	X	注入过采样

25.3.32 双重 ADC 模式

如果器件配有两个或多个 ADC，可使用双重 ADC 模式（请参见*图 176*）：

- ADC1 与 ADC2 可在双重模式下共同使用（ADC1 为主器件）

在双重 ADC 模式下，通过 ADCx 主器件到 ADC 从器件的交替触发或同时触发来启动转换，具体取决于 ADCx_CCR 寄存器中的 DUAL[4:0] 位所选的模式。

可实现以下四种模式：

- 注入同步模式
- 常规同步模式
- 交替模式
- 交替触发模式

也可按以下方式组合使用这些模式：

- 注入同步模式 + 常规同步模式
- 常规同步模式 + 交替触发模式
- 注入同步模式 + 交替模式

在双重 ADC 模式下（ADCx_CCR 寄存器中的 DUAL[4:0] 位不等于零时），ADCx_CFGR 寄存器的 CONT、AUTDLY、DISCEN、DISCNUM[2:0]、JDISCEN、JQM、JAUTO 位会在主 ADC 与从 ADC 之间共享：从 ADC 中的位始终与主 ADC 的对应位相等。

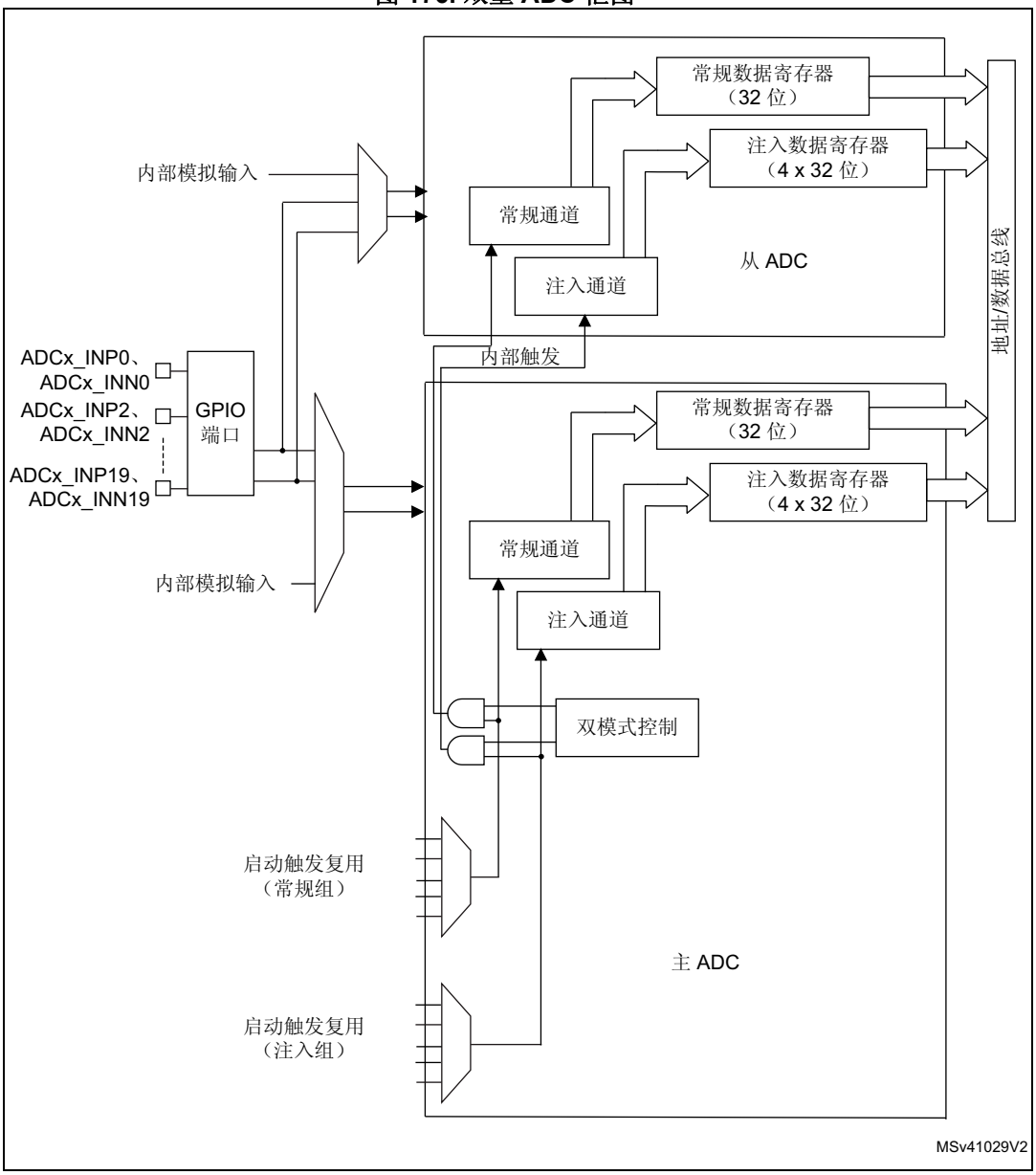
要在双重模式下开始转换，用户必须仅对主 ADC 的 EXTEN、EXTSEL、JEXTEN、JEXTSEL 位进行编程，以配置软件或硬件触发以及常规或注入触发。（从 ADC 的 EXTEN[1:0] 和 JEXTEN[1:0] 位为无关位）。

在常规同步或交替模式下：用户将主 ADC 的 ADSTART 位或 ADSTP 位置 1 后，从 ADC 的相应位也会自动置 1。 但从 ADC 的 ADSTART 位或 ADSTP 位不需要与主 ADC 位同时清零。

在注入同步或交替触发模式下：用户将主 ADC 的 JADSTART 位或 JADSTP 位置 1 后，从 ADC 的相应位也会自动置 1。 但从 ADC 的 JADSTART 位或 JADSTP 位不需要与主 ADC 位同时清零。

在双重模式下，可通过读取 ADC 通用数据寄存器 (ADCx_CDR) 的方式同时读取主 ADC 和从 ADC 的已转换数据。此外，还可以通过读取双重模式状态寄存器 (ADCx_CSR) 的方式同时读取主 ADC 和从 ADC 的状态位。

图 176. 双重 ADC 框图(1)



1. 从 ADC 上也存在外部触发，但该图中未予以显示。
2. ADC 通用数据寄存器 (ADCx_CDR) 包含主 ADC 和从 ADC 的常规转换数据。

注入同步模式

通过将 DUAL[4:0] 位编程为 00101 来选择此模式。

此模式可转换注入通道组。外部触发源来自主 ADC 的注入组多路复用器（通过 ADCx_JSQR 寄存器中的 JEXTSEL[4:0] 位选择）。

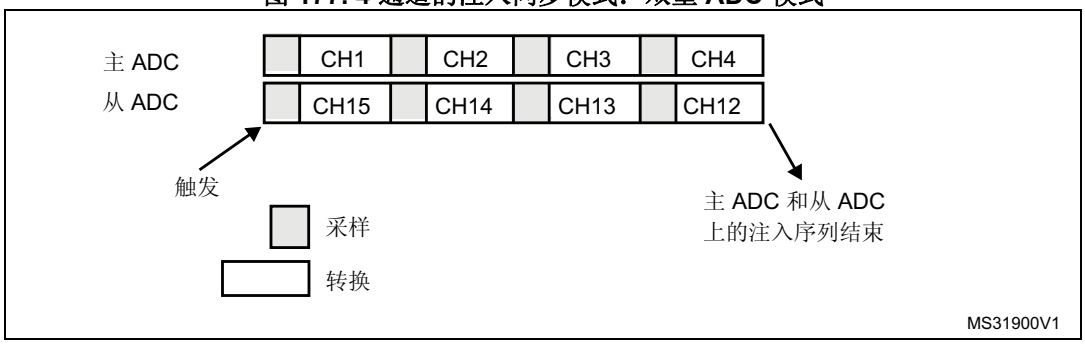
注： *不要在两个 ADC 上转换同一通道（转换同一通道时，不允许两个 ADC 采样时间重叠）。*

在同步模式下，转换的序列长度必须相同，并且在序列中，主器件和从器件的第 N 次转换必须配置为采用相同的采样时间。

常规转换可在一个或所有 ADC 上执行。这种情况下，它们彼此之间都是独立的，而且会在出现注入事件时中断。它们会在注入转换组结束时恢复转换。

- 当主 ADC 上出现注入转换序列结束 (JEOS) 事件时，已转换数据会存储到主 ADCx_JDRy 寄存器中，并会产生 JEOS 中断（若使能）。
- 当从 ADC 上出现注入转换序列结束 (JEOS) 事件时，已转换数据会存储到从 ADCx_JDRy 寄存器中，并会产生 JEOS 中断（若使能）。
- 如果主注入序列的持续时间与从注入序列的持续时间相等（如*图 177* 所示），软件可以只使能两个 JEOS 中断中的一个（例如：主 JEOS）并读取两者的已转换数据（从主 ADCx_JDRy 和从 ADCx_JDRy 寄存器中读取）。

图 177. 4 通道的注入同步模式：双重 ADC 模式



如果 JDISCEN=1，注入序列的每个同步转换都需要在出现一个注入触发事件后才能进行。

该模式可与 AUTDLY 模式结合使用：

- 转换的同步注入序列结束后，仅当主 ADC 和从 ADC 的 JEOS 位均已清零时，才会接受新的注入触发事件（延时阶段）。如果在进行的注入序列中以及关联的延时阶段出现新的注入触发事件，则会被忽略。
- 主 ADC 的常规转换序列结束后，仅当已读取主数据寄存器 (ADCx_DR) 时，才会接受主 ADC 的新常规触发事件。如果在进行的常规序列中以及关联的延时阶段出现关于主 ADC 的新常规触发事件，则会被忽略。
  在从 ADC 上发生的常规序列也是如此。

支持独立注入的常规同步模式

通过将 DUAL[4:0] 位编程为 00110 来选择此模式。

此模式可用于常规通道组。外部触发源来自主 ADC 的常规组多路复用器（通过 ADCx_CFGR 寄存器中的 EXTSEL[4:0] 位选择）。同时触发可用于从 ADC。

在该模式下，支持独立注入转换。注入请求（主 ADC 或从 ADC 上）将中止当前的同步转换，并在注入转换结束后重新开始该同步转换。

注： *不要在两个 ADC 上转换同一通道（转换同一通道时，不允许两个 ADC 采样时间重叠）。*

在常规同步模式下，转换的序列长度必须相同，并且在序列中，主器件和从器件的第 N 次转换必须配置为采用相同的采样时间。

当软件可读取数据时，会通过中断的方式获得通知：

- 每次主 ADC 上出现转换结束事件时 (EOC)，会产生主 EOC 中断（若使能 EOCIE），软件可读取主 ADC 的 ADCx_DR。
- 每次从 ADC 上出现转换结束事件时 (EOC)，会产生从 EOC 中断（若使能 EOCIE），软件可读取从 ADC 的 ADCx_DR。
- 如果主常规序列的持续时间与从常规序列的持续时间相等（如*图 178* 所示），软件可以只使能两个 EOC 中断中的一个（例如：主 EOC）并从通用数据寄存器 (ADCx_CDR) 中读取两者的已转换数据。

此外，还可以使用 DMA 读取常规数据，具体通过以下两种方法来实现：

- 使用两条 DMA 通道（一条用于主 ADC，另一条用于从 ADC）。在这种情况下，必须使 DAMDF[1:0] 位保持清零。
  - 将 DMA 主 ADC 通道配置为读取主 ADC 的 ADCx_DR。每次主 ADC 上出现 EOC 事件时，都会生成 DMA 请求。
  - 将 DMA 从 ADC 通道配置为读取从 ADC 的 ADCx_DR。每次从 ADC 上出现 EOC 事件时，都会生成 DMA 请求。
- 配置双重 ADC 模式数据格式 DAMDF[1:0] 位，留下一条 DMA 通道用于其他用途：
  - 配置 DAMDF[1:0]=0b10 或 0b11（具体取决于分辨率）。
  - 使用一条 DMA 通道（主 ADC 的一条通道）。将 DMA 主 ADC 通道配置为读取通用 ADC 寄存器 (ADCx_CDR)。
  - 每次同时出现主 EOC 事件和从 EOC 事件时，都会生成单次 DMA 请求。此时，从 ADC 转换的数据会出现在 ADCx_CDR 32 位寄存器的高位半字中，主 ADC 转换的数据会出现在 ADCx_CCR 寄存器的低位半字中。
  - DMA 读取 ADCx_CCR 寄存器时，两个 ECO 标志都会清零。

注： *DAMDF[1:0]=0b10 或 0b11 时，用户必须将主序列和从序列中的转换次数编程为相同值。否则，多余的转换将不会生成 DMA 请求。*

图 178. 16 通道的常规同步模式：双重 ADC 模式

如果 DISCEN=1，常规序列的每“n”个同步转换都需要在出现一次常规触发事件后才能进行（“n”由 DISCNUM 定义）。

该模式可与 AUTDLY 模式结合使用：

- 序列的同步转换结束后，仅当通用数据寄存器 ADCx_CDR（或主 ADC 的常规数据寄存器）已被读取时，才会开始序列中的下一个转换（延时阶段）。
- 转换的同步常规序列结束后，仅当通用数据寄存器 (ADCx_CDR) 已被读取时，才会接受新的常规触发事件（延时阶段）。如果在进行的常规序列中以及关联的延时阶段出现新的常规触发事件，则会被忽略。

可以使用 DMA 在常规同步与 AUTDLY 组合模式下处理数据，前提是使用多重 DMA 模式：DAMDF 位必须设为 0b10 或 0b11。

如果常规同步模式与 AUTDLY 模式相结合使用，用户必须确保：

- 主序列中的转换次数必须与从序列中的转换次数相等。
- 对于序列的每个同步转换，从 ADC 的转换长度要小于主 ADC 的转换长度。请注意，序列长度取决于要转换的通道数目以及每条通道的采样时间和分辨率。

注： *仅当只编程了常规通道时，才可以将常规同步模式与 AUTDLY 模式结合使用：禁止在该组合模式下对注入通道进行编程。*

支持独立注入的交替模式

通过将 DUAL[4:0] 位编程为 00111 来选择此模式。

此模式只能用于常规组（通常为一个通道）。外部触发源来自主 ADC 的常规通道多路复用器。

出现外部触发之后：

- 主 ADC 立即启动。
- 从 ADC 在主 ADC 采样阶段完成后的多个-ADC 时钟周期延时后启动。

交替模式下 2 个转换之间的最小延迟通过 ADCx_CCR 寄存器中的 DELAY 位进行配置。该延时会在主转换的采样阶段结束后开始计时。这样一来，如果某个 ADC 的互补 ADC 仍在对其输入进行采样，则该 ADC 无法启动转换（在给定时间内，只有一个 ADC 能够对输入信号采样）。

- 可能的最小延时为 1，用以确保主 ADC 采样阶段模拟开关断开与从 ADC 采样阶段模拟开关闭合之间至少有一个周期时间。
- 最大延时与所选分辨率对应的周期数相等。但用户必须正确计算该延时，以确保在某个 ADC 仍在对其输入进行采样时不会有其他 ADC 开始转换。

如果主 ADC 和从 ADC 上的 CONT 位均置 1，则这两个 ADC 所选常规通道会连续进行转换。

每次从 ADC 上出现转换结束事件时 (EOC)，都会在软件能够读取数据时以中断方式通知软件，会产生从 EOC 中断和主 EOC 中断（若使能 EOCIE），软件可读取从/主 ADC 的 ADCx_DR。

注： *可以仅使能从 ADC 的 EOC 中断并读取通用数据寄存器 (ADCx_CDR)。但在这种情况下，用户必须确保转换持续时间兼容，以保证在序列内，主转换之后始终会进行从转换，之后才会重新开始新的主转换。建议使用 MDMA 模式。*

此外，还可以通过 DMA 传输常规数据。在这种情况下，不能在每个 ADC 上使用单次 DMA 请求，务必使用 MDMA 模式，方法如下：

- 配置 DAMDF[1:0]=0b10 或 0b11（具体取决于分辨率）。
- 使用一条 DMA 通道（主 ADC 的一条通道）。将 DMA 主 ADC 通道配置为读取通用 ADC 寄存器 (ADCx_CDR)。
- 每次同时出现主 EOC 事件和从 EOC 事件时，都会生成单次 DMA 请求。此时，从 ADC 转换的数据会出现在 ADCx_CDR 32 位寄存器的高位半字中，主 ADC 转换的数据会出现在 ADCx_CCR 寄存器的低位半字中。
- DMA 读取 ADCx_CCR 寄存器时，两个 ECO 标志都会清零。

图 179. 连续转换模式下 1 通道的交替模式：双重 ADC 模式

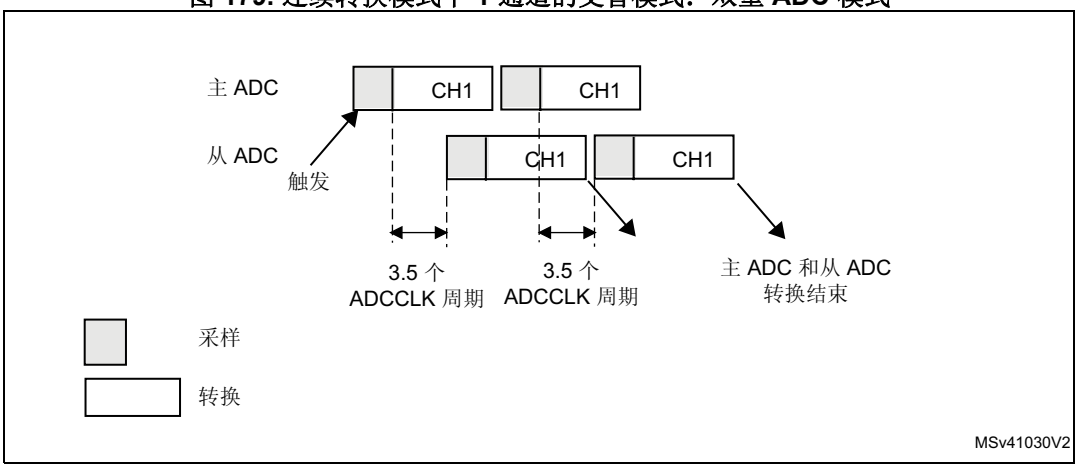
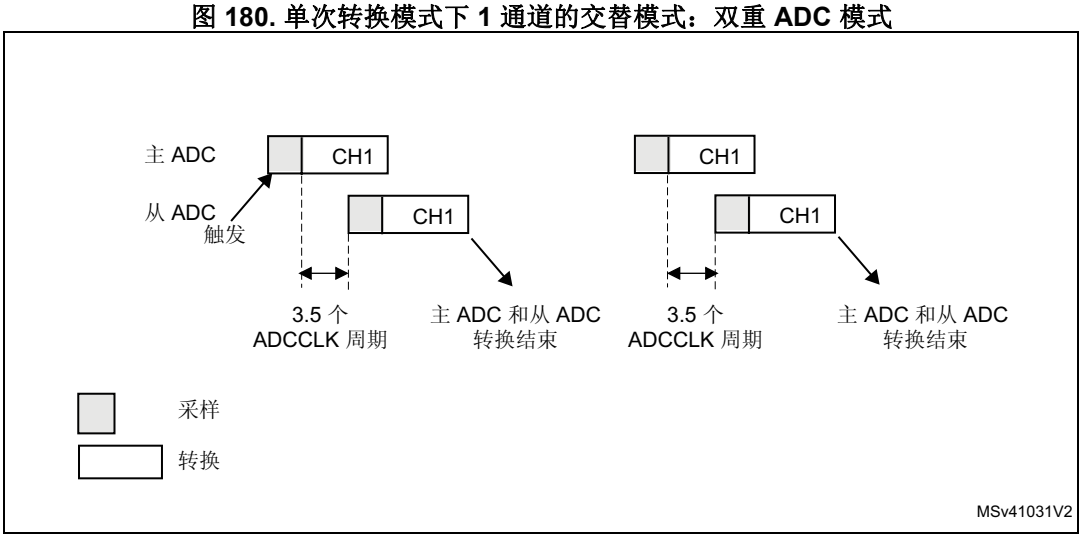


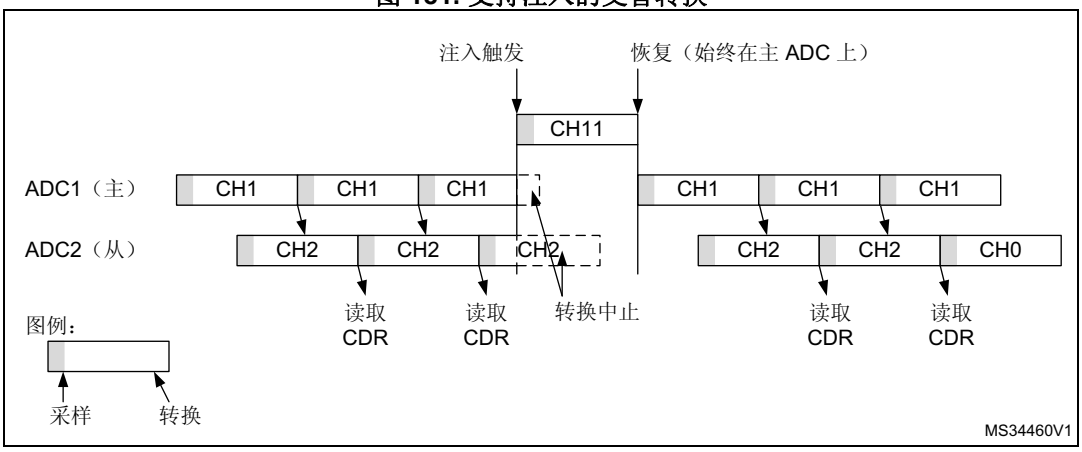
图 180. 单次转换模式下 1 通道的交替模式：双重 ADC 模式



如果 DISCEN=1，常规序列的每“n”（“n”由 DISCNUM 定义）个同步转换需要在出现一个常规触发事件后才能进行。

在该模式下，支持注入转换。注入完成后（在主 ADC 或从 ADC 上），主常规转换和从常规转换都会中止，序列会从主 ADC 重启（请参见下文中的*图 181*）。

图 181. 支持注入的交替转换



交替触发模式

通过将 DUAL[4:0] 位编程为 01001 来选择此模式。

此模式只能用于注入组。外部触发源来自主 ADC 的注入组多路复用器。

仅当选择了硬件触发时，才能使用此模式：JEXTEN 不得为 0x0。

禁止注入不连续模式（两个 ADC 的 JDISCEN 均为 0）

1. 发生第一次触发时，将转换组中主 ADC 的所有注入通道。
2. 发生第二次触发时，将转换组中从 ADC 的所有注入通道。
3. 以此类推。

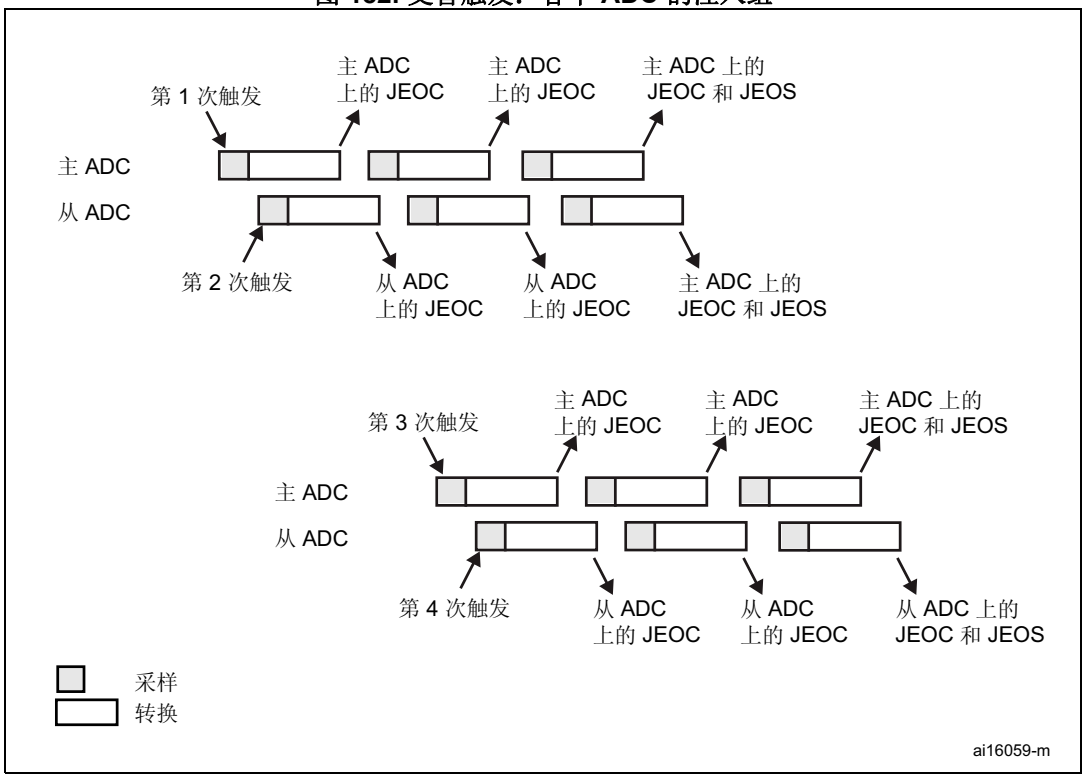
当组中主 ADC 的所有通道都转换完成后，会生成一个 JEOC 中断（如果已使能）。

当组中从 ADC 的所有通道都转换完成后，会生成一个 JEOC 中断（如果已使能）。

每次注入转换后也会生成 JEOC 中断（如果已使能）。

如果在组中的所有注入通道都完成转换后出现另一个外部触发，则可通过转换组中主 ADC 的注入通道来重新启动交替触发过程。

图 182. 交替触发：各个 ADC 的注入组



注： *常规转换可在一个或所有 ADC 上使能。在这种情况下，常规转换彼此之间是独立的。当 ADC 必须执行注入转换时，会中断常规转换。它会在注入转换完成后恢复。*

两次触发事件之间的时间间隔必须大于或等于 1 个 ADC 时钟周期。同一个 ADC 上可启动转换的两个触发事件之间的最小时间间隔与单个 ADC 模式下相同。

使能注入不连续模式（两个 ADC 的 JDISCEN 均为 1）

如果使能主 ADC 和从 ADC 的注入不连续模式：

- 发生第一次触发时，将转换主 ADC 的第一条注入通道。
- 发生第二次触发时，将转换从 ADC 的第一条注入通道。
- 以此类推。

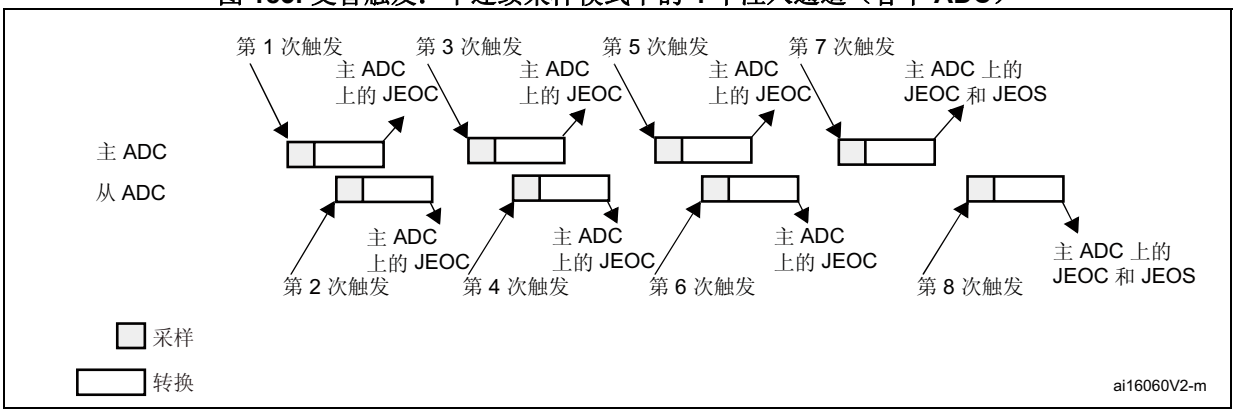
当组中主 ADC 的所有通道都转换完成后，会生成一个 JEOC 中断（如果已使能）。

当组中从 ADC 的所有通道都转换完成后，会生成一个 JEOC 中断（如果已使能）。

每次注入转换后也会生成 JEOC 中断（如果已使能）。

如果注入组中的所有通道都完成转换后出现另一个外部触发，则会重新启动交替触发过程。

图 183. 交替触发：不连续采样模式下的 4 个注入通道（各个 ADC）



混合型常规/注入同步模式

通过将 DUAL[4:0] 位编程为 00001 来选择此模式。

可以中断常规组的同步转换，然后开始注入组的同步转换。

注： *转换的序列长度必须相同，并且在给定序列中，主模式和从模式的第 N 次转换必须配置为采用相同的采样时间，或必须确保触发之间的间隔长于 2 个序列的长转换时间。如果不遵循上述条件，当序列较长的 ADC 完成上一次转换时，序列较短的 ADC 可能重新开始转换。*

常规同步 + 交替触发组合模式

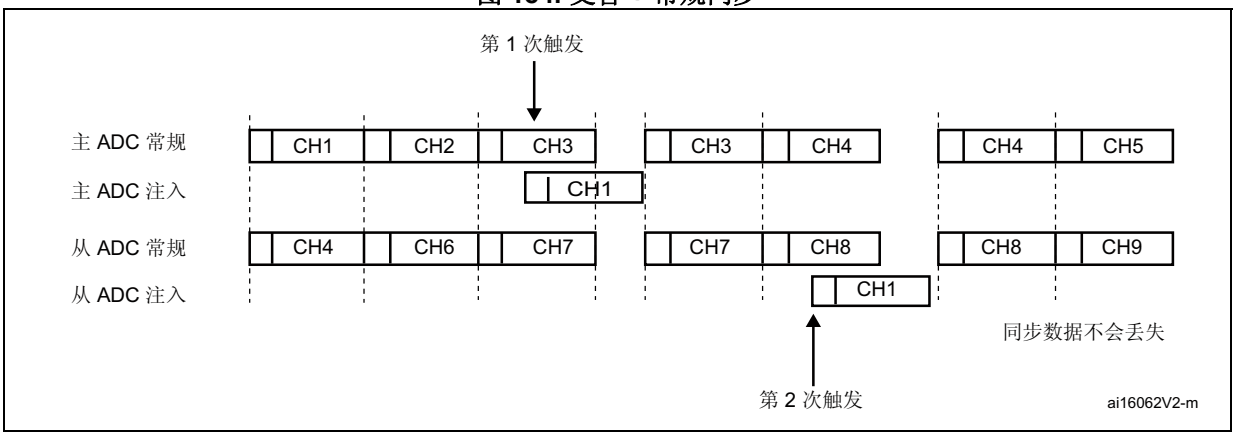
通过将 DUAL[4:0] 位编程为 00010 来选择此模式。

可以中断常规组的同步转换，然后开始注入组的交替触发转换。*图 184* 说明了交替触发模式中断同步常规转换的行为。

注入事件后立即开始注入交替转换。当常规转换处于运行状态时，为确保在注入转换后实现同步，所有的（主/从）ADC 常规转换均将停止，并会在注入转换结束时得以恢复运行。

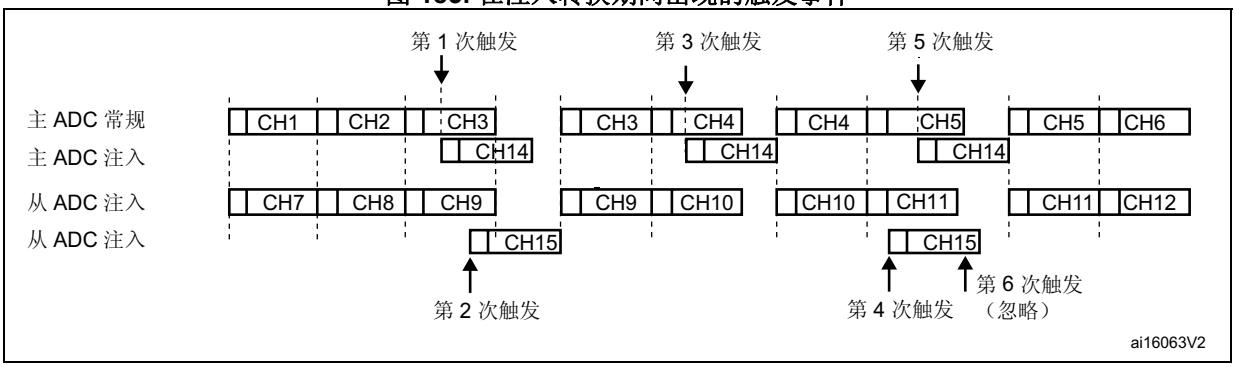
注： *转换的序列长度必须相同，并且在给定序列中，主模式和从模式的第 N 次转换必须配置为采用相同的采样时间，或必须确保触发之间的间隔长于 2 个序列的长转换时间。如果不遵循上述条件，，当序列较长的 ADC 完成上一次转换时，序列较短的 ADC 可能重新开始转换。*

图 184. 交替 + 常规同步



如果在已导致常规转换中断的注入转换期间出现交替触发，则会得到处理。*图 185* 显示了这种情况下的操作（请注意，由于与第 6 次触发相关联的交替转换未完成，因此会忽略第 6 次触发）。

图 185. 在注入转换期间出现的触发事件



注入同步 + 交替组合模式

通过将 DUAL[4:0] 位编程为 00011 来选择此模式。

可以通过同步注入事件中断交替转换。

在这种情况下，交替转换会立即中断，同步注入转换会开始。注入序列结束时，会恢复交替转换。交替常规转换恢复后，执行的第一个常规转换始终是主 ADC 的常规转换。*图 186*、*图 187* 和 *图 188* 对相关操作进行了举例说明。

注意： 在该模式下，必须使用通用数据寄存器以单次读访问的形式读取常规数据。但是，主从数据的一致性无法得到保证。

图 186. 单条交替通道 CH0，注入序列 CH11 和 CH12

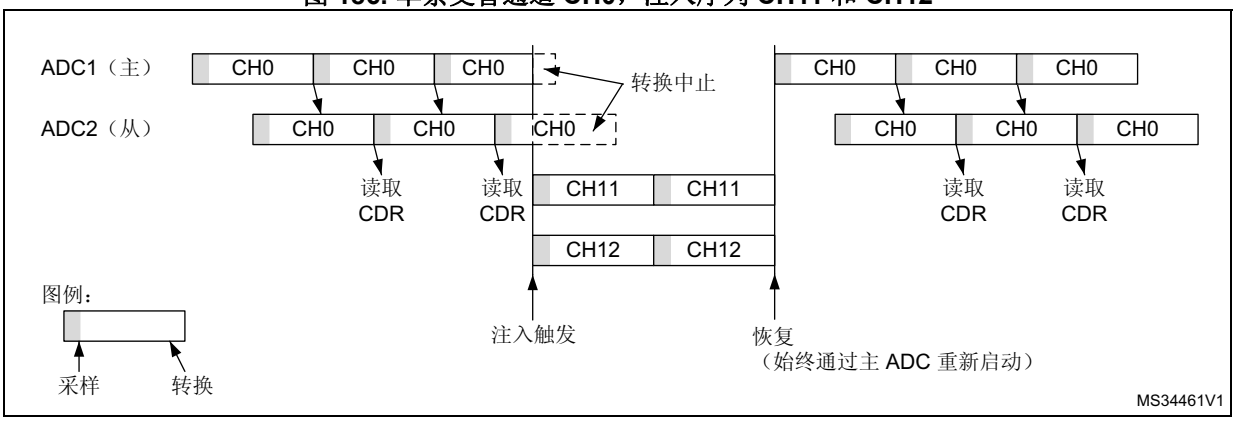


图 187. 两条交替通道（CH1、CH2），注入序列 CH11 和 CH12 - 情况 1：先中断主 ADC

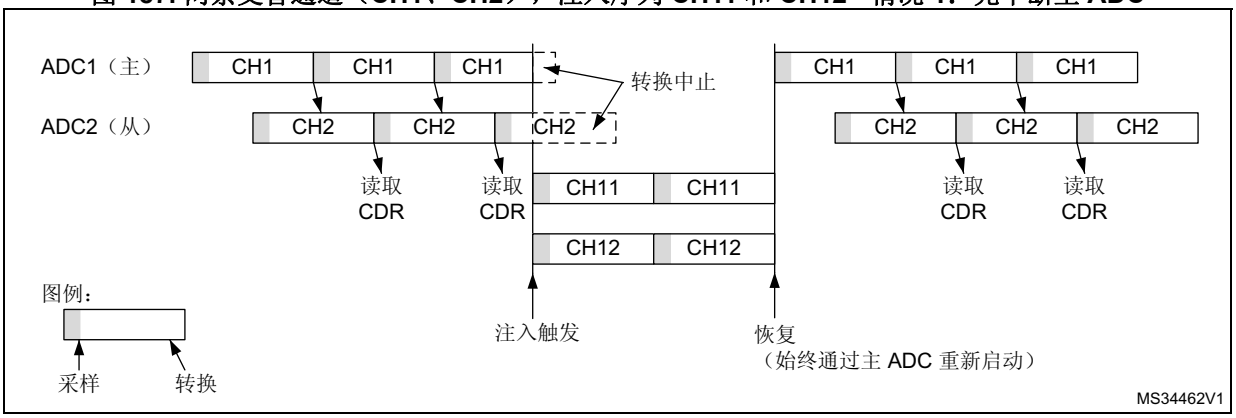
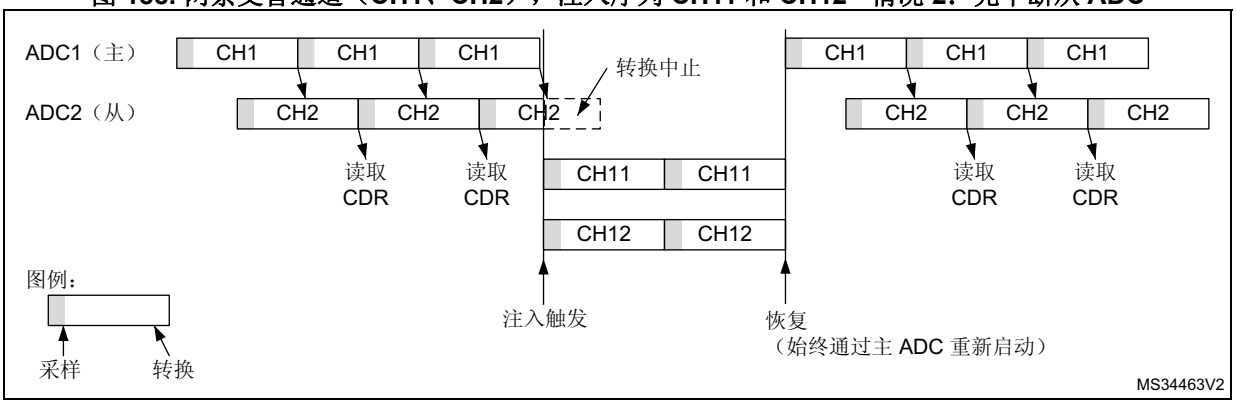


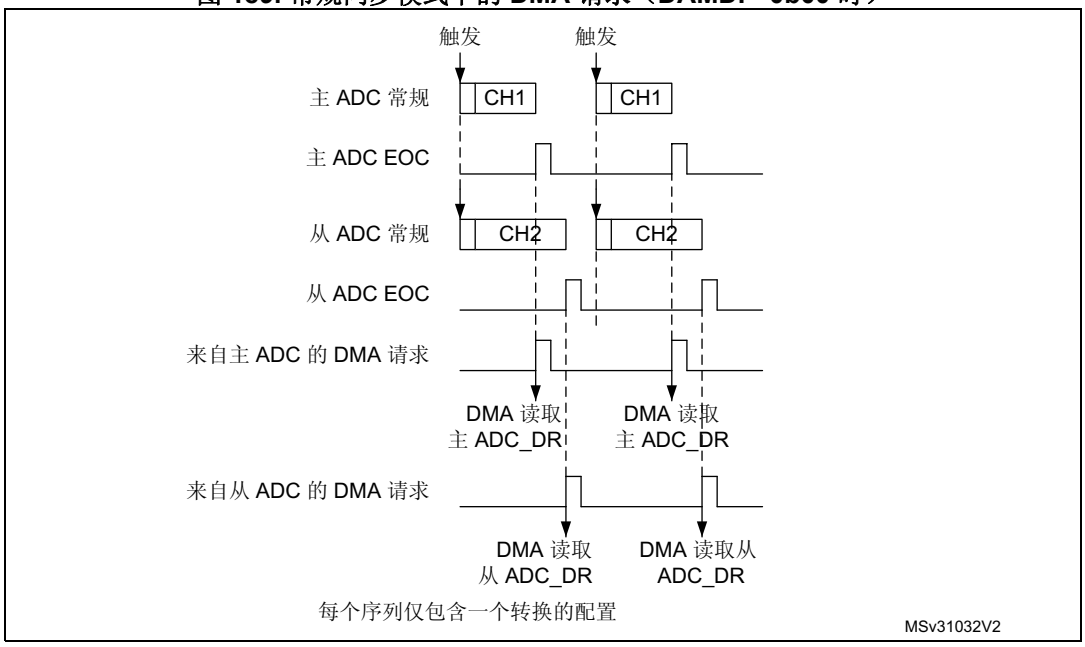
图 188. 两条交替通道（CH1、CH2），注入序列 CH11 和 CH12 - 情况 2：先中断从 ADC



双重 ADC 模式下的 DMA 请求

在所有双重 ADC 模式下，可以使用两条 DMA 通道（一条用于主 ADC，一条用于从 ADC）来传输数据，与在单模式下一样（请参见*图 189：常规同步模式下的 DMA 请求（DAMDF=0b00 时）*）。

图 189. 常规同步模式下的 DMA 请求（DAMDF=0b00 时）



在同步常规模式和交替模式下，还可以保留一条 DMA 通道而使用单条 DMA 通道传输两者的数据。为此，必须配置 ADCx_CCR 寄存器中的 DAMDF 位：

- **DAMDF=0b10，32 位格式**：出现主 EOC 事件或从 EOC 事件时，会交替生成单次 DMA 请求。此时，数据项会交替出现在 ADCx_CDR2 32 位寄存器中。当分辨率高于 16 位时，该模式用于交替模式和常规同步模式。

  示例：

  双重交替模式：每当有新的 32 位数据可用时，就会生成一个 DMA 请求：

  第一个 DMA 请求：ADCx_CDR2[31:0] = MST_ADCx_DR[31:0]

  第二个 DMA 请求：ADCx_CDR2[31:0] = SLV_ADCx_DR[31:0]

- **DAMDF=0b10，16 位格式**：每当同时出现主 EOC 事件和从 EOC 事件时，都会生成单次 DMA 请求。此时有两个数据项可用，32 位寄存器 ADCx_CDR 包含的两个半字代表两个经过 ADC 转换的数据项。从 ADC 数据占用高位半字，主 ADC 数据占用低位半字。当分辨率介于 10 位和 16 位之间时，该模式用于交替模式和常规同步模式。如果主转换器或从转换器中的值大于 16 位，则会被截断为 16 个最低有效位。

  示例：

  双重交替模式：每当有 2 个数据项可用时，就会生成一个 DMA 请求：

  第一个 DMA 请求：ADCx_CDR[31:0] = SLV_ADCx_DR[15:0] | MST_ADCx_DR[15:0]

  第二个 DMA 请求：ADCx_CDR[31:0] = SLV_ADCx_DR[15:0] | MST_ADCx_DR[15:0]

图 190. 常规同步模式下的 DMA 请求（DAMDF=0b10 时）

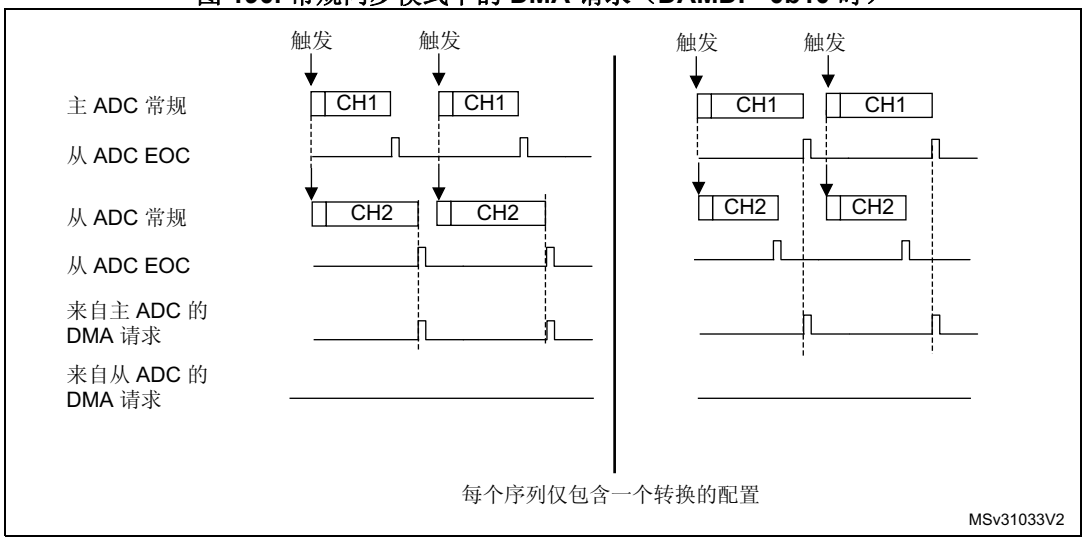
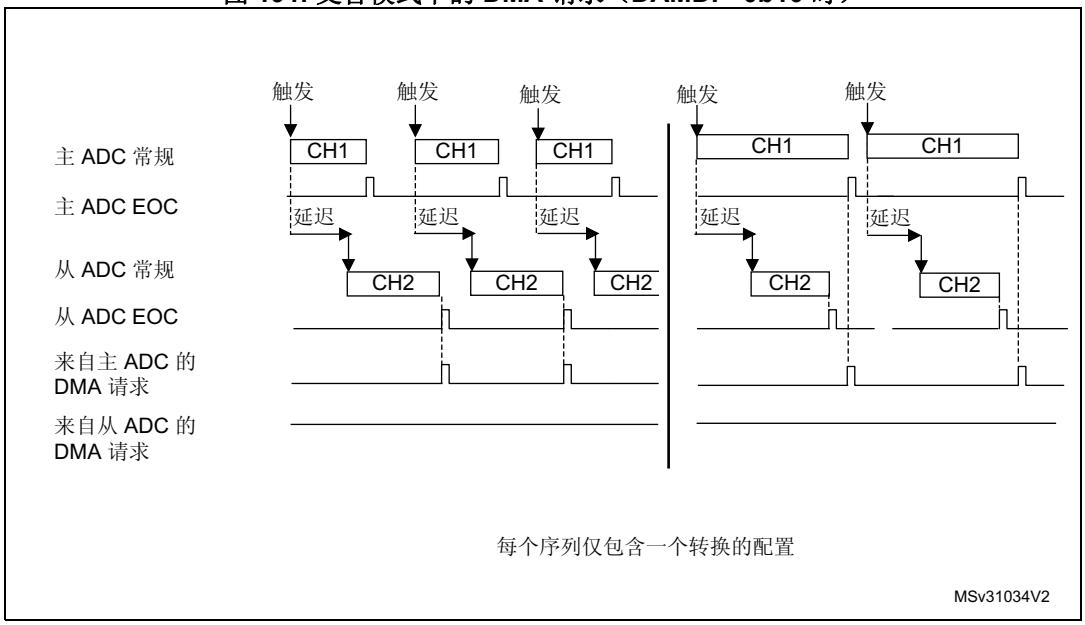


图 191. 交替模式下的 DMA 请求（DAMDF=0b10 时）



注： *使用多重 ADC 模式时，用户必须正确配置主转换与从转换的持续时间，以便在开始新转换之前生成 DMA 请求并用于读取两者（主转换和从转换）的数据。*

- **DAMDF=0b11**：该模式与 DAMDF=0b10 时相似。唯一的区别是：在这种模式下，每发送一个 DMA 请求（两个数据项可用），就会以半字的形式传输表示两个 ADC 转换数据项的两个字节。

  当结果为 8 位时，该模式用于交替模式和常规同步模式。当有 4 个新的 8 位值可用时，会发出新的 DMA 请求。

  示例：

  双重交替模式：每当有 4 个数据项可用时，就会生成一个 DMA 请求（t0、t1... 对应于连续的采样点）

  第一个 DMA 请求：

  $ADCx_CDR[7:0] = MST_ADCx_DR[7:0]_{t0}$

  $ADCx_CDR[15:8] = SLV_ADCx_DR[7:0]_{t0}$

  $ADCx_CDR[23:16] = MST_ADCx_DR[7:0]_{t1}$

  $ADCx_CDR[31:24] = SLV_ADCx_DR[7:0]_{t1}$

  第二个 DMA 请求：

  $ADCx_CDR[7:0] = MST_ADCx_DR[7:0]_{t2}$

  $ADCx_CDR[15:8] = SLV_ADCx_DR[7:0]_{t2}$

  $ADCx_CDR[23:16] = MST_ADCx_DR[7:0]_{t3}$

  $ADCx_CDR[31:24] = SLV_ADCx_DR[7:0]_{t3}$

溢出检测：

在双重 ADC 模式下（DUAL[4:0] 不等于 b00000 时），如果在其中一个 ADC 上检测到溢出，将不再发出 DMA 请求，以确保传输到 RAM 的所有数据均有效（无论 DAMDF 采用何种配置，都具有此特性）。对于与某个 ADC 对应的 EOC 位，有时可能会因为此 ADC 的数据寄存器包含有效数据而保持置 1。

选择多重 ADC 模式时的 DMA 单次模式/DMA 循环模式

如果选择 DAMDF 模式（0b10 或 0b11），还必须将主 ADC 的 ADCx_CCR 寄存器中的 DMNGT[1:0] 位配置为 0b10，以在 DMA 单次模式和 DMA 循环模式之间进行选择，请参见 *使用 DMA 管理转换*一节。

停止双重 ADC 模式下的转换

用户必须将主 ADC 的控制位 ADSTP/JADSTP 置 1，以停止双重 ADC 模式下两个 ADC 的转换。从 ADC 的另一控制位 ADSTP 在双重 ADC 模式下不起作用。

两个 ADC 均有效停止后，主 ADC 和从 ADC 的 ADSTART/JADSTART 均会通过硬件清零。

双重 ADC 交替模式下的 DFSDM 模式

在双重 ADC 交替模式下，ADC 转换结果可直接传送到具有 ΣΔ 调制器的数字滤波器 (DFSDM)。

此模式通过将主 ADC 的 ADCx_CFGR 寄存器中的 DMNGT[1:0] 位设为 0b10 来使能。

ADC 将常规数据寄存器的 16 个最低有效位交替地从主转换器和从转换器传送至 DFSDM 的一条通道。

数据必须为 16 位有符号格式：

ADCx_DR[31:16] = 0x0000

ADCx_DR[15] = 符号

ADCx_DR[14:0] = 数据

如果任何转换器中存在超出 16 位有符号格式的值，则会被截断。

双重 ADC 同步模式下的 DFSDM 模式

在双重 ADC 同步模式下使用 DFSDM 无需双重模式，因为转换数据将由每个单独的通道处理。具有相同触发源的单一模式会导致使用 DFSDM 接口进行同步转换。

25.3.33 温度传感器

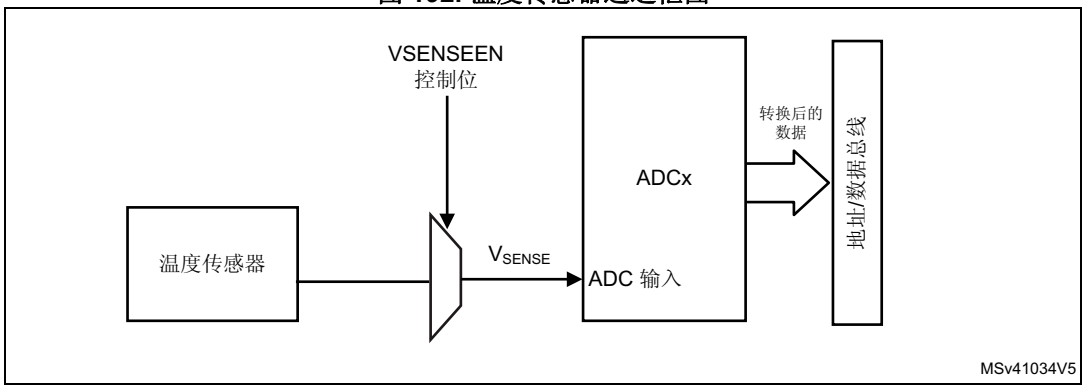
温度传感器可测量器件的结温 (T_J)，温度范围为 -40 °C 到 125 °C。

温度传感器内部连接到 ADC3 VINP[18] 输入通道，该通道用于将传感器输出电压转换为数字值。温度传感器模拟引脚的采样时间必须大于产品数据手册中指定的稳定时间。

不使用时可将传感器置于掉电模式。

图 192 显示了温度传感器框图。

图 192. 温度传感器通道框图



注： *必须将 VSENSEEN 位置 1 才能使能内部通道 ADC3 VINP[18] 的转换（温度传感器，V_{SENSE}）。*

读取温度

要使用传感器，请执行以下操作：

1. 选择 ADC3 VINP[18] 输入通道（使用合适的采样时间）。
2. 设定合适的采样时间（请参见器件数据手册中的电气特性部分）。
3. 在 ADCx_CCR 寄存器中将 VSENSEEN 位置 1，以便将温度传感器从掉电模式中唤醒。

4. 开始 ADC 转换。
5. 读取 ADC 数据寄存器中生成的 V_{SENSE} 数据
6. 使用以下公式计算实际温度：

$$\text{温度}(\,^{\circ}\text{C}) = \frac{110\,^{\circ}\text{C} - 30\,^{\circ}\text{C}}{\text{TS_CAL2} - \text{TS_CAL1}} \times (\text{TS_DATA} - \text{TS_CAL1}) + 30\,^{\circ}\text{C}$$

其中：

- TS_CAL2 是在 110°C 下获得的温度传感器校准值
- TS_CAL1 是在 30°C 下获得的温度传感器校准值
- TS_DATA 是由 ADC 转换得到的实际温度传感器输出值

  更多关于 TS_CAL1 和 TS_CAL2 校准点的信息，请参见器件数据手册

注： *传感器从掉电模式中唤醒需要一个启动时间，启动时间过后其才能正确输出 V_{SENSE}。ADC 在上电后同样需要一个启动时间，因此，为尽可能减少延迟间，应同时将 ADEN 和 SENSEEN 位置 1。*

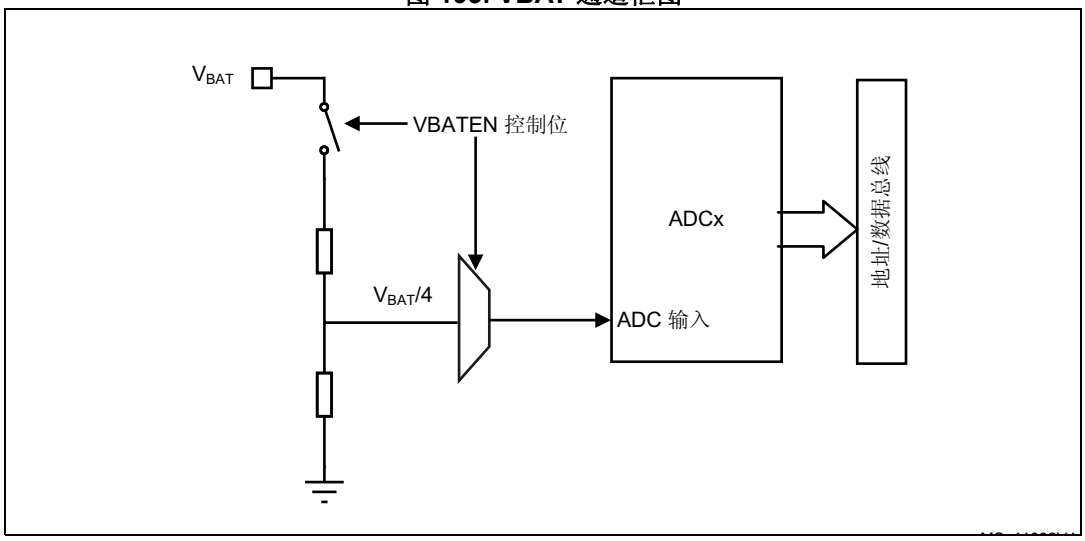
25.3.34 V_{BAT} 电源监测

ADCx_CCR 寄存器中的 VBATEN 位用于切换到电池电压。由于 V_{BAT} 电压可能高于 V_{DDA}，因此 V_{BAT} 引脚需要从内部连接到桥接分配器（除以4），以确保 ADC 正确运行。VBATEN 位置 1 时，会自动使能此桥，以将 V_{BAT}/4 连接到 ADC3 VINP[17] 输入通道。因此，转换出的数字值为 V_{BAT} 电压的四分之一。为防止电池出现意外的电能消耗，建议仅在必要时为 ADC 转换使能桥接分配器。

有关转换 V_{BAT}/4 电压时要应用的采样时间值，请参见器件数据手册的电气特性部分。

图 193 显示了 V_{BAT} 传感特性的框图。

图 193. VBAT 通道框图



注： *必须将 VBATEN 位置 1 才能使能内部通道 ADC3 VINP[17](V_{BAT}/4) 的转换。*

25.3.35 监测内部参考电压

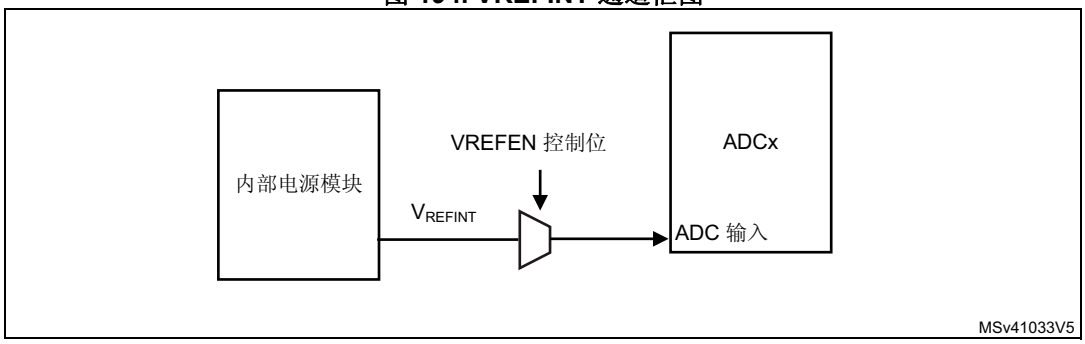
可以通过监测内部参考电压来获得用于评估 ADC V_{REF+} 电压的参考值。

内部参考电压在内部连接到输入通道 ADC3 VINP[19]。

该通道的采样时间必须大于产品数据手册中指定的稳定时间。

图 193 显示了 V_{REFINT} 传感特性的框图。

图 194. VREFINT 通道框图



注： *必须将 ADCx_CCR 寄存器中的 VREFEN 位置 1 才能使能内部通道 ADC3_IN19 (V_{REFINT}) 的转换。*

使用内部参考电压计算实际的 V_{DDA} 电压

施加给微控制器的 V_{DDA} 电源电压可能会有变化，或无法获得准确值。在制造过程中由 ADC 在 V_{DDA} = 3.3 V 的条件下获得的内置内部参考电压 (V_{REFINT}) 及其校准数据可用于评估实际的 V_{DDA} 电压。

以下公式可求得为器件供电的实际的 V_{DDA} 电压：

$$V_{DDA} = 3.3\ V \times VREFINT_CAL / VREFINT_DATA$$

其中：

- VREFINT_CAL 是 VREFINT 校准值
- VREFINT_DATA 是由 ADC 转换得到的实际 VREFINT 输出值

将电源相关的 ADC 测量值转换为绝对电压值

ADC 用于提供对应于模拟电源与施加给转换通道的电压之比的数字值。对于大部分应用用例，需要将该比值转换成与 V_{DDA} 无关的电压。对于 V_{DDA} 已知、ADC 转换值进行了右对齐的应用，可使用以下公式得到该绝对值：

$$V_{CHANNELx} = \frac{V_{DDA}}{FULL_SCALE} \times ADCx_DATA$$

对于 V_{DDA} 值未知的应用，必须使用内部参考电压，V_{DDA} 可替换为*使用内部参考电压计算实际的 V_{DDA} 电压*一节部分提供的表达式，从而得出以下公式：

$$V_{CHANNELx} = \frac{3.3\ V \times VREFINT_CAL \times ADCx_DATA}{VREFINT_DATA \times FULL_SCALE}$$

其中：

- VREFINT_CAL 是 VREFINT 校准值
- ADCx_DATA 是由 ADC 在通道 x 上测得的值（右对齐）
- VREFINT_DATA 是由 ADC 转换得到的实际 VREFINT 输出值
- FULL_SCALE 是 ADC 输出的最大数字值。例如，如果分辨率为 16 位，该值为 2^{16}- 1 = 65535，如果分辨率为 8 位，该值为 2^8 - 1 = 255。

注： *如果执行 ADC 测量时使用的是输出格式而非 16 位右对齐格式，那么必须先将所有参数转换为兼容格式，然后再进行计算。*

25.4 ADC 中断

对于每个 ADC，可在下列情况下产生中断：

- ADC 就绪后，ADC 上电（ADRDY 标志）
- 常规组的任何转换结束时（EOC 标志）
- 常规组的转换序列结束时（EOS 标志）
- 注入组的任何转换结束时（JEOC 标志）
- 注入组的转换序列结束时（JEOS 标志）
- 发生模拟看门狗检测时（AWD1、AWD2 和 AWD3 标志）
- 采样阶段结束时（EOSMP 标志）
- 发生数据溢出时（OVR 标志）
- 注入序列上下文队列溢出时（JQOVF 标志）

可以使用单独的中断使能位以提高灵活性。

表 201. 每个 ADC 的 ADC 中断

中断事件	事件标志	使能控制位
ADC 就绪	ADRDY	ADRDYIE
结束常规组的转换	EOC	EOCIE
常规组的转换序列结束	EOS	EOSIE
注入组的转换结束	JEOC	JEOCIE
注入组的转换序列结束	JEOS	JEOSIE
模拟看门狗 1 状态位置 1	AWD1	AWD1IE
模拟看门狗 2 状态位置 1	AWD2	AWD2IE
模拟看门狗 3 状态位置 1	AWD3	AWD3IE
采样阶段结束	EOSMP	EOSMPIE
上溢	OVR	OVRIE
注入上下文队列溢出	JQOVF	JQOVFIE

25.5 ADC 寄存器（每个 ADC）

有关寄存器说明中使用的缩写，请参见*第 94 页的第 1.1 节*。

25.5.1 ADC x 中断和状态寄存器 (ADCx_ISR)（x=1 到 3）

ADC x interrupt and status register

偏移地址：0x00

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	JQOVF	AWD3	AWD2	AWD1	JEOS	JEOC	OVR	EOS	EOC	EOSMP	ADRDY
					rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1

位 31:11 保留，必须保持复位值。

位 10 **JQOVF**：注入上下文队列溢出 (Injected context queue overflow)
当注入上下文队列溢出时，该位会由硬件置 1。通过软件写入 1 可将该位清零。更多信息，请参见*第 25.3.22 节：注入转换的上下文队列*。
0：未发生注入上下文队列溢出（或标志事件已通过软件确认并清零）
1：发生注入上下文队列溢出

位 9 **AWD3**：模拟看门狗 3 标志 (Analog watchdog 3 flag)
当转换电压超过在 ADCx_TR3 寄存器的 LT3[7:0] 和 HT3[7:0] 字段中编程的值时，硬件会将该位置 1。通过软件写入 1 可将该位清零。
0：未发生模拟看门狗 3 事件（或标志事件已通过软件确认并清零）
1：发生模拟看门狗 3 事件

位 8 **AWD2**：模拟看门狗 2 标志 (Analog watchdog 2 flag)
当转换电压超过在 ADCx_TR2 寄存器的 LT2[7:0] 和 HT2[7:0] 字段中编程的值时，硬件会将该位置 1。通过软件写入 1 可将该位清零。
0：未发生模拟看门狗 2 事件（或标志事件已通过软件确认并清零）
1：发生模拟看门狗 2 事件

位 7 **AWD1**：模拟看门狗 1 标志 (Analog watchdog 1 flag)
当转换电压超过在 ADCx_TR1 寄存器的 LT1[11:0] 和 HT1[11:0] 字段中编程的值时，硬件会将该位置 1。但需要通过软件清零。向其中写入 1。
0：未发生模拟看门狗 1 事件（或标志事件已通过软件确认并清零）
1：发生模拟看门狗 1 事件

位 6 **JEOS**：注入通道序列结束标志 (Injected channel end of sequence flag)
组内所有注入通道转换结束时，硬件将该位置 1。通过软件写入 1 可将该位清零。
0：注入转换序列未完成（或标志事件已通过软件确认并清零）
1：注入转换已完成

位 5 **JEOC**：注入通道转换结束标志 (Injected channel end of conversion flag)

当通道的每次注入转换结束，新数据出现在相应的 ADCx_JDRy 寄存器时，会通过硬件将该位置 1。通过软件向该位写入 1，或读取相应的 ADCx_JDRy 寄存器都可将该位清零。

0：注入通道转换未完成（或标志事件已通过软件确认并清零）

1：注入通道转换已完成

位 4 **OVR**：ADC 溢出 (ADC overrun)

该位在常规通道上发生溢出事件时由硬件置 1，这意味着在 EOC 标志已置 1 时，新转换已完成。通过软件写入 1 可将该位清零。

0：未发生溢出事件（或标志事件已通过软件确认并清零）

1：发生溢出

位 3 **EOS**：常规序列结束标志 (End of regular sequence flag)

常规通道序列转换结束后，硬件将该位置 1。通过软件写入 1 可将该位清零。

0：常规转换序列未完成（或标志事件已通过软件确认并清零）

1：常规转换序列已完成

位 2 **EOC**：转换结束标志 (End of conversion flag)

当通道的每次常规转换结束，新数据出现在 ADCx_DR 寄存器时，会通过硬件将该位置 1。通过软件向该位写入 1，或读取 ADCx_DR 寄存器都可将该位清零。

0：常规通道转换未完成（或标志事件已通过软件确认并清零）

1：常规通道转换已完成

位 1 **EOSMP**：采样结束标志 (End of sampling flag)

在任何通道（仅限常规通道）的转换过程中，当采样阶段结束时，会通过硬件将该位置 1。

0：采样阶段未结束（或标志事件已通过软件确认并清零）

1：采样阶段已结束

位 0 **ADRDY**：ADC READY

ADC 使能后（位 ADEN=1）以及 ADC 达到准备好接收转换请求的状态时，会通过硬件将该位置 1。通过软件写入 1 可将该位清零。

0：ADC 未准备好开始转换（或标志事件已通过软件确认并清零）

1：ADC 已准备好开始转换

25.5.2 ADC x 中断使能寄存器 (ADCx_IER)（x=1 到 3）

ADC x interrupt enable register

偏移地址：0x04

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	JQOVFIE	AWD3IE	AWD2IE	AWD1IE	JEOSIE	JEOCIE	OVRIE	EOSIE	EOCIE	EOSMPIE	ADRDYIE
					rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:11 保留，必须保持复位值。

位 10 **JQOVFIE**：注入上下文队列溢出中断使能 (Injected context queue overflow interrupt enable)

此位由软件置 1 和清零，用于使能/禁止注入上下文队列溢出中断。

0：禁止注入上下文队列溢出中断。

1：使能注入上下文队列溢出中断 JQOVF 位置 1 时产生中断。

注： 仅当 JADSTART=0 时（这可确保当前未进行任何注入转换），才允许通过软件对此位执行写操作。

位 9 **AWD3IE**：模拟看门狗 3 中断使能 (Analog watchdog 3 interrupt enable)

此位由软件置 1 和清零，用于使能/禁止模拟看门狗 2 中断。

0：禁止模拟看门狗 3 中断

1：使能模拟看门狗 3 中断

注： 仅当 ADSTART=0 且 JADSTART=0 时（这可确保当前未进行任何转换），才允许通过软件对此位执行写操作。

位 8 **AWD2IE**：模拟看门狗 2 中断使能 (Analog watchdog 2 interrupt enable)

此位由软件置 1 和清零，用于使能/禁止模拟看门狗 2 中断。

0：禁止模拟看门狗 2 中断

1：使能模拟看门狗 2 中断

注： 仅当 ADSTART=0 且 JADSTART=0 时（这可确保当前未进行任何转换），才允许通过软件对此位执行写操作。

位 7 **AWD1IE**：模拟看门狗 1 中断使能 (Analog watchdog 1 interrupt enable)

此位由软件置 1 和清零，用于使能/禁止模拟看门狗 1 中断。

0：禁止模拟看门狗 1 中断

1：使能模拟看门狗 1 中断

注： 仅当 ADSTART=0 且 JADSTART=0 时（这可确保当前未进行任何转换），才允许通过软件对此位执行写操作。

位 6 **JEOSIE**：注入转换序列结束中断使能 (End of injected sequence of conversions interrupt enable)

此位由软件置 1 和清零，用于使能/禁止注入转换序列结束中断。

0：禁止 JEOS 中断

1：使能 JEOS 中断。JEOS 位置 1 时产生中断。

注： 仅当 JADSTART=0 时（这可确保当前未进行任何注入转换），才允许通过软件对此位执行写操作。

位 5 **JEOCIE**：注入转换结束中断使能 (End of injected conversion interrupt enable)

此位由软件置 1 和清零，用于使能/禁止注入转换结束中断。

0：禁止 JEOC 中断

1：使能 JEOC 中断 JEOC 位置 1 时产生中断。

注： 仅当 JADSTART=0 时（这可确保当前未进行任何常规转换），才允许通过软件对此位执行写操作。

位 4 **OVRIE**：溢出中断使能 (overrun interrupt enable)

此位由软件置 1 和清零，用于使能/禁止常规转换的溢出中断。

0：禁止溢出中断

1：使能溢出中断。OVR 位置 1 时产生中断。

注： 仅当 ADSTART=0 时（这可确保当前未进行任何常规转换），才允许通过软件对此位执行写操作。

位 3 **EOSIE**：常规转换序列结束中断使能 (End of regular sequence of conversions interrupt enable)

此位由软件置 1 和清零，用于使能/禁止常规转换序列结束中断。

0：禁止 EOS 中断

1：使能 EOS 中断。EOS 位置 1 时产生中断。

注： 仅当 ADSTART=0 时（这可确保当前未进行任何常规转换），才允许通过软件对此位执行写操作。

位 2 **EOCIE**：常规转换结束中断使能 (End of regular conversion interrupt enable)

此位由软件置 1 和清零，用于使能/禁止常规转换结束中断。

0：禁止 EOC 中断

1：使能 EOC 中断 EOC 位置 1 时产生中断。

注： 仅当 ADSTART=0 时（这可确保当前未进行任何常规转换），才允许通过软件对此位执行写操作。

位 1 **EOSMPIE**：常规转换采样结束中断使能 (End of sampling flag interrupt enable for regular conversions)

此位由软件置 1 和清零，用于使能/禁止常规转换采样阶段结束中断。

0：禁止 EOSMP 中断。

1：使能 EOSMP 中断。EOSMP 位置 1 时产生中断。

注： 仅当 ADSTART=0 时（这可确保当前未进行任何常规转换），才允许通过软件对此位执行写操作。

位 0 **ADRDYIE**：ADC 就绪中断使能 (ADC ready interrupt enable)

此位由软件置 1 和清零，用于使能/禁止 ADC 就绪中断。

0：禁止 ADRDY 中断

1：使能 ADRDY 中断。ADRDY 位置 1 时产生中断。

注： 仅当 ADSTART=0 且 JADSTART=0 时（这可确保当前未进行任何转换），才允许通过软件对此位执行写操作。

25.5.3 ADC x 控制寄存器 (ADCx_CR)（x=1 到 3）

ADC x control register

偏移地址：0x08

复位值：0x2000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ADCAL	ADCALDIF	DEEPPWD	ADVREGEN	LINCALRDYW6	LINCALRDYW5	LINCALRDYW4	LINCALRDYW3	LINCALRDYW2	LINCALRDYW1	Res.	Res.	Res.	Res.	Res.	ADCALLIN
rs	rw	rw	rw	rw	rw	rw	rw	rw	rw						rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	BOOST	Res.	Res.	JADSTP	ADSTP	JADSTART	ADSTART	ADDIS	ADEN
							rw			rs	rs	rs	rs	rs	rs

位 31 **ADCAL**：ADC 校准 (ADC calibration)

此位由软件置 1，用于开始 ADC 校准。先对 ADCALDIF 位编程，以确定此校准应用于单端输入模式还是差分输入模式。

校准完成后，该位由硬件清零。

0：校准已完成

1：写入 1 可校准 ADC。读取值为 1 表示正在进行校准。

注： 仅当 ADEN=0 时，才允许通过软件将 ADCAL 置 1 的方式启动校准。

注： 仅当 ADEN=1、ADSTART=0 且 JADSTART=0（ADC 已使能，当前未进行任何转换）时，才允许通过软件对 ADCx_CALFACT 执行写操作来更新校准系数。

位 30 **ADCALDIF**：校准的输入模式 (Input mode for calibration)

此位由软件置 1 和清零，用于为校准配置单端输入模式或差分输入模式。

0：写入 ADCAL 将在单端输入模式下启动校准。

1：写入 ADCAL 将在差分输入模式下启动校准。

注： 仅当 ADC 已禁止、且当前未进行校准时（ADCAL=0、JADSTART=0、JADSTP=0、ADSTART=0、ADSTP=0、 ADDIS=0 且 ADEN=0），才允许通过软件对此位执行写操作。

位 29 **DEEPPWD**：深度掉电使能 (Deep-power-down enable)

此位由软件置 1 和清零，用于使 ADC 进入深度掉电模式。

0：ADC 未处于深度掉电模式

1：ADC 处于深度掉电模式（默认复位状态）

注： 仅当 ADC 已禁止时（ADCAL=0、JADSTART=0、JADSTP=0、ADSTART=0、ADSTP=0、ADDIS=0 且 ADEN=0），才允许通过软件对此位执行写操作。

位 28 **ADVREGEN**：ADC 稳压器使能 (ADC voltage regulator enable)

此位由软件置 1，用于使能 ADC 稳压器。

执行启动校准或使能 ADC 等操作之前，必须先使能 ADC 稳压器，并且软件必须等待稳压器启动时间。

0：禁止 ADC 稳压器

1：使能 ADC 稳压器

更多关于 ADC 稳压器使能和禁止序列的详细信息，请参见*第 25.3.6 节：ADC 深度掉电模式 (DEEPPWD) 和 ADC 稳压器 (ADVREGEN)*。

仅当 ADC 已禁止时（ADCAL=0、JADSTART=0、ADSTART=0、ADSTP=0、ADDIS=0 且 ADEN=0），才能通过软件对该位域进行编程。

位 27 **LINCALRDYW6**：线性度校准就绪字 6 (Linearity calibration ready Word 6)

该控制/状态位允许读取/写入第 6 个线性度校准系数。

线性度校准完成后，此位会置 1。如果该位清零，会开始将线性度系数 6 传输到 ADCx_CALFACT2 寄存器的 LINCALFACT[29:0] 中。当 ADCx_CALFACT2 寄存器可读取时，此位将由硬件复位（软件必须轮询此位，直至此位清零）。

LINCALRDYW6 位复位时，可将新的线性度系数 6 的值写入 ADCx_CALFACT2 寄存器的 LINCALFACT[29:0] 中。如果该位置 1，线性度系数 6 会更新，更新完毕后，该位会立即由硬件有效置 1（软件必须轮询该位，直至该位置 1，指示写入有效）。

注： ADCx_CALFACT2[29:10] *包含 0。ADCx_CALFACT2[9:0] 对应于线性度校准系数位 [159:150]。*

注： *仅当 LINCALRDYW5、LINCALRDYW4、LINCALRDYW3、LINCALRDYW2 和 LINCALRDYW1 位保持不变时，才允许通过软件切换该位，详细信息请参见 25.3.8：校准（ADCAL、ADCALDIF、ADCALLIN、ADCx_CALFACT）。*

仅当 ADEN=1、ADSTART=0 且 JADSTART=0（ADC 已使能，当前未进行任何转换）时，才允许通过软件对 LINCALRDYWx 执行写操作来更新线性度校准系数。

位 26 **LINCALRDYW5**：线性度校准就绪字 5 (Linearity calibration ready Word 5)

请参见 LINCALRDYW6 说明。

注： *ADCx_CALFACT2[29:0] 对应于线性度校准系数位 [149:120]。*

仅当 LINCALRDYW6、LINCALRDYW5、LINCALRDYW3、LINCALRDYW2 和 LINCALRDYW1 位保持不变时，才允许通过软件切换该位。

位 25 **LINCALRDYW4**：线性度校准就绪字 4 (Linearity calibration ready Word 4)

请参见 LINCALRDYW6 说明。

注： *ADCx_CALFACT2[29:0] 对应于线性度校准系数位 [119:90]。*

仅当 LINCALRDYW6、LINCALRDYW5、LINCALRDYW3、LINCALRDYW2 和 LINCALRDYW1 位保持不变时，才允许通过软件切换该位。

位 24 **LINCALRDYW3**：线性度校准就绪字 3 (Linearity calibration ready Word 3)

请参见 LINCALRDYW6 说明。

注： *ADCx_CALFACT2[29:0] 对应于线性度校准系数位 [89:60]。*

仅当 LINCALRDYW6、LINCALRDYW5、LINCALRDYW4、LINCALRDYW2 和 LINCALRDYW1 位保持不变时，才允许通过软件切换该位。

位 23 **LINCALRDYW2**：线性度校准就绪字 2 (Linearity calibration ready Word 2)

请参见 LINCALRDYW6 说明。

注： *ADCx_CALFACT2[29:0] 对应于线性度校准系数位 [59:30]。*

仅当 LINCALRDYW6、LINCALRDYW5、LINCALRDYW4、LINCALRDYW3 和 LINCALRDYW1 位保持不变时，才允许通过软件切换该位。

位 22 **LINCALRDYW1**：线性度校准就绪字 1 (Linearity calibration ready Word 1)

请参见 LINCALRDYW6 说明。

注： *ADCx_CALFACT2[29:0] 对应于线性度校准系数位 [29:0]。*

注： *仅当 LINCALRDYW6、LINCALRDYW5、LINCALRDYW4、LINCALRDYW3 和 LINCALRDYW2 位保持不变时，才允许通过软件切换该位。*

位 21:17 保留，必须保持复位值。

位 16 **ADCALLIN**：线性度校准 (Linearity calibration)

此位由软件置 1 和清零，用于使能线性度校准。

0：写入 ADCAL 将启动校准，但不进行线性度校准。

1：写入 ADCAL 将启动校准，并会进行线性度校准。

注： 仅当 ADC 已禁止、且当前未进行校准时（ADCAL=0、JADSTART=0、JADSTP=0、ADSTART=0、ADSTP=0、ADDIS=0 且 ADEN=0），才允许通过软件对此位执行写操作。

位 15:9 保留，必须保持复位值。

位 8 **BOOST**：升压模式控制 (Boost mode control)

此位由软件置 1 和清零，用于使能/禁止升压模式。

0：升压模式关闭。当 ADC 时钟 < 20 MHz 时使用，以在较低时钟频率下实现节能。

1：升压模式开启。ADC 时钟 > 20 MHz 时必须使用。

注： 仅当 ADSTART=0 且 JADSTART=0 时（这可确保当前未进行任何转换），才允许通过软件对此位执行写操作。

如果使能了双重模式（ADCx_CCR 寄存器的 DAMDF 位不等于零），则从 ADC 的 BOOST 位不再可写，其内容与主 ADC 的 AUTDLY 位相等。

位 7:6 保留，必须保持复位值。

位 5 **JADSTP**：ADC 停止注入转换命令 (ADC stop of injected conversion command)

该位由软件置 1，用于停止和丢弃正在进行的注入转换（JADSTP 命令）。

当转换已有效丢弃、并且可重新配置 ADC 注入序列和触发时，会通过硬件将该位清零。随后，ADC 会准备好接收新的开始注入转换命令（JADSTART 命令）。

0：当前未执行 ADC 停止注入转换命令

1：写入 1 可停止正在进行的注入转换。读取值为 1 表示正在执行 ADSTP 命令。

注： 仅当 JADSTART=1 且 ADDIS=0 时（ADC 已使能、最终会进行注入转换、并且没有任何待处理的禁止 ADC 的请求），才允许通过软件将 JADSTP 置 1。

在自动注入模式下 (JAUTO=1)，将 ADSTP 位置 1 会中止常规转换和注入转换（不要使用 JADSTP）

位 4 **ADSTP**：ADC 停止常规转换命令 (ADC stop of regular conversion command)

该位由软件置 1，用于停止和丢弃正在进行的常规转换（ADSTP 命令）。

当转换已有效丢弃、并且可重新配置 ADC 常规序列和触发时，会通过硬件将该位清零。随后，ADC 会准备好接收新的开始常规转换命令（ADSTART 命令）。

0：当前未执行 ADC 停止常规转换命令

1：写入 1 可停止正在进行的常规转换。读取值为 1 表示正在执行 ADSTP 命令。

注： 仅当 ADSTART=1 且 ADDIS=0 时（ADC 已使能、最终会进行常规转换、并且没有任何待处理的禁止 ADC 的请求），才允许通过软件将 ADSTP 置 1。

在自动注入模式下 (JAUTO=1)，将 ADSTP 位置 1 会中止常规转换和注入转换（不要使用 JADSTP）。

在双重 ADC 常规同步模式和交错模式下，必须使用主 ADC 的 ADSTP 位停止常规转换。另一 ADSTP 位不起作用。

位 3 **JADSTART**：ADC 开始注入转换 (ADC start of injected conversion)

此位由软件置 1，用于开始 ADC 的注入通道转换。根据配置位 JEXTEN 的值，可以立即开始转换（软件触发配置），也可以在发生注入硬件触发事件后开始转换（硬件触发配置）。

该位通过硬件清零：

– 在单次转换模式下，如果选择了软件触发 (JEXTSEL=0x0)：出现注入转换序列结束 (JEOS) 标志时清零。

– 在所有情况下：执行完 JADSTP 命令后，由硬件将 JADSTP 位清零的同时清零。

0：当前未进行 ADC 注入转换。

1：写入 1 可开始注入转换。读取值为 1 表示 ADC 正在运行，最终会转换注入通道。

注： *仅当 ADEN=1 且 ADDIS=0 时（ADC 已使能，并且没有任何待处理的禁止 ADC 的请求），才允许通过软件将 JADSTART 置 1。*

在自动注入模式下 (JAUTO=1)，通过将 ADSTART 位置 1 开始常规转换和自动注入转换（JADSTART 必须保持清零）

位 2 **ADSTART**：ADC 开始常规转换 (ADC start of regular conversion)

此位由软件置 1，用于开始 ADC 的常规通道转换。根据配置位 EXTEN 的值，可以立即开始转换（软件触发配置），也可以在发生常规硬件触发事件后开始转换（硬件触发配置）。

该位通过硬件清零：

– 在单次转换模式下（CONT=0、DISCEN=0），如果选择了软件触发 (EXTEN=0x0)：出现常规转换序列结束 (EOS) 标志时清零。

– 在不连续转换模式下（CONT=0、DISCEN=1），如果选择了软件触发 (EXTEN=0x0)：出现转换结束 (EOC) 标志时清零。

– 在所有其他情况下：执行完 ADSTP 命令后，由硬件将 ADSTP 位清零的同时清零。

0：当前未进行 ADC 常规转换。

1：写入 1 可开始常规转换。读取值为 1 表示 ADC 正在运行，最终会转换常规通道。

注： *仅当 ADEN=1 且 ADDIS=0 时（ADC 已使能，并且没有任何待处理的禁止 ADC 的请求），才允许通过软件将 ADSTART 置 1。*

在自动注入模式下 (JAUTO=1)，通过将 ADSTART 位置 1 开始常规转换和自动注入转换（JADSTART 必须保持清零）

位 1 **ADDIS**：ADC 禁止命令 (ADC disable command)

该位由软件置 1，用于禁止 ADC（ADDIS 命令）并使其进入掉电状态（OFF 状态）。

ADC 已有效禁止后，会立即通过硬件将该位清零（此时也会通过硬件将 ADEN 清零）。

0：当前未执行 ADDIS 命令。

1：写入 1 可禁止 ADC。读取值为 1 表示正在执行 ADDIS 命令。

注： *仅当 ADEN=1、DSTART=0 且 JADSTART=0 时（这可确保当前未进行任何转换），才允许通过软件将 ADDIS 置 1。*

位 0 **ADEN**：ADC 使能控制 (ADC enable control)

该位由软件置 1，用于使能 ADC。ADRDY 标志置 1 后，ADC 将立即准备好运行。

如果 ADC 已禁止，则执行 ADDIS 命令后，将通过硬件对该位清零。

0：禁止 ADC（OFF 状态）。

1：写入 1 来使能 ADC。

注： *仅当 ADCx_CR 寄存器的所有位均为 0 时（ADCAL=0、JADSTART=0、ADSTART=0、ADSTP=0、ADDIS=0 且 ADEN=0，但 ADVREGEN 位除外，此位必须为 1），才允许通过软件将 ADEN 置 1（并且软件必须等待稳压器的启动时间）。*

25.5.4 ADC x 配置寄存器 (ADCx_CFGR)（x=1 到 3）

ADC x configuration register

偏移地址：0x0C

复位值：0x8000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
JQDIS	AWD1CH[4:0]					JAUTO	JAWD1EN	AWD1EN	AWD1SGL	JQM	JDISCEN	DISCNUM[2:0]			DISCEN
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	AUTDLY	CONT	OVRMOD	EXTEN[1:0]		EXTSEL[4:0]					RES[2:0]			DMNGT[1:0]	
	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31 **JQDIS**：注入队列禁止 (Injected Queue disable)

这些位由软件置 1 和清零，用于禁止注入队列机制：

0：使能注入队列

1：禁止注入队列

注： *仅当 ADSTART=0 且 JADSTART=0 时（这可确保当前未进行常规转换、也未进行注入转换），才允许通过软件对此位执行写操作。*

将 JQDIS 位置 1 或复位会导致注入队列被清空，JSQR 寄存器也会被清空。

位 30:26 **AWD1CH[4:0]**：模拟看门狗 1 通道选择 (Analog watchdog 1 channel selection)

这些位将由软件置 1 和清零。它们用于选择由模拟看门狗监控的输入通道。

00000：通过 AWD1 监控 ADC 模拟输入通道 0

00001：通过 AWD1 监控 ADC 模拟输入通道 1

.....

10010：通过 AWD1 监控 ADC 模拟输入通道 19

其他：保留，不得使用

注： *通过 AWD1CH 选择的通道必须也在 SQRi 或 JSQRi 寄存器中选择。*

仅当 ADSTART=0 且 JADSTART=0 时（这可确保当前未进行任何转换），才允许通过软件对这些位执行写操作。

位 25 **JAUTO**：注入组自动转换 (Automatic injected group conversion)

通过软件将该位置 1 和清零可在常规组转换后分别使能/禁止注入组自动转换。

0：禁止注入组自动转换

1：使能注入组自动转换

注： *仅当 ADSTART=0 且 JADSTART=0 时（这可确保当前未进行常规转换、也未进行注入转换），才允许通过软件对此位执行写操作。*

如果使能了双重模式（ADCx_CCR 寄存器的 DAMDF 位不等于零），则从 ADC 的 JAUTO 位不再可写，其内容与主 ADC 的 JAUTO 位相等。

位 24 **JAWD1EN**：注入通道上的模拟看门狗 1 使能 (Analog watchdog 1 enable on injected channels)

此位由软件置 1 和清零

0：在注入通道上禁止模拟看门狗 1

1：在注入通道上使能模拟看门狗 1

注： *仅当 JADSTART=0 时（这可确保当前未进行任何注入转换），才允许通过软件对此位执行写操作。*

位 23 **AWD1EN**：常规通道上的模拟看门狗 1 使能 (Analog watchdog 1 enable on regular channels)

此位由软件置 1 和清零

0：在常规通道上禁止模拟看门狗 1

1：在常规通道上使能模拟看门狗 1

注： 仅当 ADSTART=0 时（这可确保当前未进行任何常规转换），才允许通过软件对此位执行写操作。

位 22 **AWD1SGL**：在单一通道或所有通道上使能看门狗 1 (Enable the watchdog 1 on a single channel or on all channels)

此位由软件置 1 和清零，用于在通过 AWD1CH[4:0] 位确定的通道或所有通道上使能模拟看门狗。

0：在所有通道上使能模拟看门狗 1

1：在单一通道上使能模拟看门狗 1

注： 仅当 ADSTART=0 且 JADSTART=0 时（这可确保当前未进行任何转换），才允许通过软件对这些位执行写操作。

位 21 **JQM**：JSQR 队列模式 (JSQR queue mode)

此位由软件置 1 和清零。

此位定义空队列的管理方式。

0：JSQR 模式 0：队列从不为空，并会保留上一次写入 JSQR 的配置。

1：JSQR 模式 1：队列可以为空，队列为空时，会在上一个有效注入序列完成后立即在内部禁止注入序列的软件和硬件触发。

更多信息，请参见*第 25.3.22 节：注入转换的上下文队列*。

注： 仅当 JADSTART=0 时（这可确保当前未进行任何注入转换），才允许通过软件对此位执行写操作。

如果使能了双重模式（ADCx_CCR 寄存器的 DAMDF 位不等于零），则从 ADC 的 JQM 位不再可写，其内容与主 ADC 的 JQM 位相等。

位 20 **JDISCEN**：注入通道的不连续采样模式 (Discontinuous mode on injected channels)

通过软件将该位置 1 和清零可使能/禁止注入通道的不连续采样模式。

0：禁止注入通道的不连续采样模式

1：使能注入通道的不连续采样模式

注： 仅当 JADSTART=0 时（这可确保当前未进行任何注入转换），才允许通过软件对此位执行写操作。

不能同时使用自动注入模式和不连续模式：当 JAUTO 置 1 时，DISCEN 和 JDISCEN 位必须通过软件保持清零状态。

如果使能了双重模式（ADCx_CCR 寄存器的 DAMDF 位不等于零），则从 ADC 的 JDISCEN 位不再可写，其内容与主 ADC 的 JDISCEN 位相等。

位 19:17 **DISCNUM[2:0]**：不连续采样模式通道计数 (Discontinuous mode channel count)

软件将写入这些位，用于定义在接收到外部触发后于不连续采样模式下转换的常规通道数。

000：1 个通道

001：2 个通道

...

111：8 个通道

注： 仅当 ADSTART=0 时（这可确保当前未进行任何常规转换），才允许通过软件对这些位执行写操作。

如果使能了双重模式（ADCx_CCR 寄存器的 DAMDF 位不等于零），则从 ADC 的 DISCNUM[2:0] 位不再可写，其内容与主 ADC 的 DISCNUM[2:0] 位相等。

位 16 **DISCEN**：常规通道的不连续模式 (Discontinuous mode for regular channels)

此位由软件置 1 和清零，用于使能/禁止常规通道的不连续模式。

0：禁止常规通道的不连续模式

1：使能常规通道的不连续模式

注： *不能同时使能不连续模式和连续模式：禁止同时将 DISCEN 和 CONT 位置 1。*

不能同时使用自动注入模式和不连续模式：当 JAUTO 置 1 时，DISCEN 和 JDISCEN 位必须通过软件保持清零状态。

仅当 ADSTART=0 时（这可确保当前未进行任何常规转换），才允许通过软件对此位执行写操作。

如果使能了双重模式（ADCx_CCR 寄存器的 DAMDF 位不等于零），则从 ADC 的 DISCEN 位不再可写，其内容与主 ADC 的 DISCEN 位相等。

位 15 保留，必须保持复位值。

位 14 **AUTDLY**：延迟转换模式 (Delayed conversion mode)

此位由软件置 1 和清零，用于使能/禁止自动延迟转换模式。

0：自动延迟转换模式关闭

1：自动延迟转换模式开启

注： *仅当 ADSTART=0 且 JADSTART=0 时（这可确保当前未进行任何转换），才允许通过软件对此位执行写操作。*

如果使能了双重模式（ADCx_CCR 寄存器的 DAMDF 位不等于零），则从 ADC 的 AUTDLY 位不再可写，其内容与主 ADC 的 AUTDLY 位相等。

位 13 **CONT**：常规转换的单次/连续转换模式 (Single / continuous conversion mode for regular conversions)

此位由软件置 1 和清零。该位置 1 时，常规转换将持续进行，直到该位清零。

0：单次转换模式

1：连续转换模式

注： *不能同时使能不连续模式和连续模式：禁止同时将 DISCEN 和 CONT 位置 1。*

仅当 ADSTART=0 时（这可确保当前未进行任何常规转换），才允许通过软件对此位执行写操作。

如果使能了双重模式（ADCx_CCR 寄存器的 DAMDF 位不等于零），则从 ADC 的 CONT 位不再可写，其内容与主 ADC 的 CONT 位相等。

位 12 **OVRMOD**：溢出模式 (Overrun Mode)

该位由软件置 1 和清零，用于配置数据溢出的管理方式。

0：如果检测到溢出，ADCx_DR 寄存器会保留原有数据。

1：如果检测到溢出，ADCx_DR 寄存器会被上一转换结果覆盖。

注： *仅当 ADSTART=0 时（这可确保当前未进行任何常规转换），才允许通过软件对此位执行写操作。*

位 11:10 **EXTEN[1:0]**：常规通道的外部触发使能和极性选择 (External trigger enable and polarity selection for regular channels)

通过软件将这些位置 1 和清零可选择外部触发极性和使能常规组的触发。

00：禁止硬件触发检测（可通过软件启动转换）

01：在上升沿执行硬件触发检测

10：在下降沿执行硬件触发检测

11：在上升沿和下降沿都执行硬件触发检测

注： *仅当 ADSTART=0 时（这可确保当前未进行任何常规转换），才允许通过软件对这些位执行写操作。*

位 9:5 **EXTSEL[4:0]**：常规组的外部触发选择 (External trigger selection for regular group)

这些位可选择用于触发常规组转换的外部事件。

00000：事件 0
00001：事件 1
00010：事件 2
00011：事件 3
00100：事件 4
00101：事件 5
00110：事件 6
00111：事件 7
...
11111：事件 31

注： 仅当 ADSTART=0 时（这可确保当前未进行任何常规转换），才允许通过软件对这些位执行写操作。

位 4:2 **RES[2:0]**：数据分辨率 (Data resolution)

通过软件写入这些位可选择转换的分辨率。

000：16 位
001：14 位
010：12 位
011：10 位
100：8 位
所有其他代码保留

注： 仅当 ADSTART=0 且 JADSTART=0 时（这可确保当前未进行任何转换），才允许通过软件对这些位执行写操作。

位 1:0 **DMNGT[1:0]**：数据管理配置 (Data Management configuration)

此位由软件置 1 和清零，用于选择 ADC 接口输出数据的管理方式。

00：常规转换数据仅存储在 DR 中
01：选择 DMA 单次模式
10：选择 DFSDM 模式。
11：选择 DMA 循环模式

注： 仅当 ADSTART=0 且 JADSTART=0 时（这可确保当前未进行任何转换），才允许通过软件对此位执行写操作。

在双重 ADC 模式下，此位无意义，会由 ADCx_CCR 寄存器的控制位 DAMDF 代替。

25.5.5 ADC x 配置寄存器 2 (ADCx_CFGR2)（x=1 到 3）

ADC x configuration register 2

偏移地址：0x10

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LSHIFT[3:0]				Res.	Res.	OSR[9:0]									
rw	rw	rw	rw			rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	RSHIFT4	RSHIFT3	RSHIFT2	RSHIFT1	ROVSM	TROVS	OVSS[3:0]				Res.	Res.	Res.	JOVSE	ROVSE
	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw				rw	rw

位 31:28 **LSHIFT[3:0]**：左移位数 (Left shift factor)

此位域由软件置 1 和清零，用于定义应用到最终结果（无论是否进行过采样）的左移位数。

0000：不左移
0001：左移 1 位
0010：左移 2 位
0011：左移 3 位
0100：左移 4 位
0101：左移 5 位
0110：左移 6 位
0111：左移 7 位
1000：左移 8 位
1001：左移 9 位
1010：左移 10 位
1011：左移 11 位
1100：左移 12 位
1101：左移 13 位
1101：左移 14 位
1111：左移 15 位

注： 仅当 ADSTART=0 时（这可确保当前未进行任何转换），才允许通过软件对此位执行写操作。

位 27:26 保留，必须保持复位值。

位 25:16 **OSR[9:0]**：过采样率 (Oversampling ratio)

此位域由软件置 1 和清零，用于定义过采样率。

0：1x（不进行过采样）
1：2x
2：3x
...
1023：1024x

注： 仅当 ADSTART=0 时（这可确保当前未进行任何转换），才允许通过软件对此位执行写操作。

位 15 保留，必须保持复位值。

位 14 **RSHIFT4**：偏移 4 校准后右移数据 (Right-shift data after Offset 4 correction)

请参见 RSHIFT1 说明

位 13 **RSHIFT3**：偏移 3 校准后右移数据 (Right-shift data after Offset 3 correction)
请参见 RSHIFT1 说明

位 12 **RSHIFT2**：偏移 2 校准后右移数据 (Right-shift data after Offset 2 correction)
请参见 RSHIFT1 说明

位 11 **RSHIFT1**：偏移 1 校准后右移数据 (Right-shift data after Offset 1 correction)
此位域由软件置 1 和清零，用于在偏移 1 校准后右移 1 位数据。此位仅用于 8 位和 16 位数据格式（详细信息请参见*数据寄存器、数据对齐和偏移（ADCx_DR、ADCx_JDRy、OFFSETy、OFFSETy_CH、OVSS、LSHIFT、RSHIFT、SSATE）*）。
0：禁止右移
1：数据右移 1 位

位 10 **ROVSM**：常规过采样模式 (Regular Oversampling mode)
此位由软件置 1 和清零，用于选择常规过采样模式。
0：连续模式：如果触发注入转换，过采样会暂时停止，并会在注入序列完成后继续（注入序列过程中会保留过采样缓冲区）
1：恢复模式：如果触发注入转换，当前过采样会中止，并会在注入序列完成后从头开始进行过采样（注入序列开始后，过采样缓冲区会清零）
注：　仅当 ADSTART=0 时（这可确保当前未进行任何转换），才允许通过软件对此位执行写操作。

位 9 **TROVS**：已触发常规过采样 (Triggered Regular Oversampling)
此位由软件置 1 和清零，以使能已触发过采样
0：会在触发后连续完成某一通道的所有过采样转换
1：某一通道的每个过采样转换都需要重新触发
注：　仅当 ADSTART=0 时（这可确保当前未进行任何转换），才允许通过软件对此位执行写操作。

位 8:5 **OVSS[3:0]**：过采样右移 (Oversampling right shift)
此位域由软件置 1 和清零，用于定义应用到原始过采样结果的右移位数。
0000：不右移
0001：右移 1 位
0010：右移 2 位
0011：右移 3 位
0100：右移 4 位
0101：右移 5 位
0110：右移 6 位
0111：右移 7 位
1000：右移 8 位
1001：右移 9 位
1010：右移 10 位
1011：右移 11 位
其他代码保留
注：　仅当 ADSTART=0 时（这可确保当前未进行任何转换），才允许通过软件对这些位执行写操作。

位 4:2 保留，必须保持复位值。

位 1 **JOVSE**：注入过采样使能 (Injected Oversampling Enable)

此位由软件置 1 和清零，用于使能注入过采样。

0：禁止注入过采样

1：使能注入过采样

注： 仅当 ADSTART=0 且 JADSTART=0 时（这可确保当前未进行任何转换），才允许通过软件对此位执行写操作

位 0 **ROVSE**：常规过采样使能 (Regular Oversampling Enable)

此位由软件置 1 和清零，用于使能常规过采样。

0：禁止常规过采样

1：使能常规过采样

注： 仅当 ADSTART=0 且 JADSTART=0 时（这可确保当前未进行任何转换），才允许通过软件对此位执行写操作

25.5.6 ADC x 采样时间寄存器 1 (ADCx_SMPR1)（x=1 到 3）

ADC x sample time register 1

偏移地址：0x14

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	SMP9[2:0]			SMP8[2:0]			SMP7[2:0]			SMP6[2:0]			SMP5[2:1]	
		rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMP5[0]	SMP4[2:0]			SMP3[2:0]			SMP2[2:0]			SMP1[2:0]			SMP0[2:0]		
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:30 保留，必须保持复位值。

位 29:0 **SMPx[2:0]**：通道 x 采样时间选择 (Channel x sampling time selection)

通过软件写入这些位可分别为各个通道选择采样时间。在采样周期期间，通道选择位必须保持不变。

000：1.5 个 ADC 时钟周期

001：2.5 个 ADC 时钟周期

010：8.5 个 ADC 时钟周期

011：16.5 个 ADC 时钟周期

100：32.5 个 ADC 时钟周期

101：64.5 个 ADC 时钟周期

110：387.5 个 ADC 时钟周期

111：810.5 个 ADC 时钟周期

注： 仅当 ADSTART=0 且 JADSTART=0 时（这可确保当前未进行任何转换），才允许通过软件对这些位执行写操作。

25.5.7 ADC x 采样时间寄存器 2 (ADCx_SMPR2)（x=1 到 3）

ADC x sample time register 2

偏移地址：0x18

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	SMP19[2:0]			SMP18[2:0]			SMP17[2:0]			SMP16[2:0]			SMP15[2:1]	
		rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMP15[0]	SMP14[2:0]			SMP13[2:0]			SMP12[2:0]			SMP11[2:0]			SMP10[2:0]		
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:30 保留，必须保持复位值。

位 29:0 **SMPx[2:0]：**通道 x 采样时间选择 (Channel x sampling time selection)

通过软件写入这些位可分别为各个通道选择采样时间。在采样周期期间，通道选择位必须保持不变。

000：1.5 个 ADC 时钟周期
001：2.5 个 ADC 时钟周期
010：8.5 个 ADC 时钟周期
011：16.5 个 ADC 时钟周期
100：32.5 个 ADC 时钟周期
101：64.5 个 ADC 时钟周期
110：387.5 个 ADC 时钟周期
111：810.5 个 ADC 时钟周期

注： 仅当 ADSTART=0 且 JADSTART=0 时（这可确保当前未进行任何转换），才允许通过软件对这些位执行写操作。

25.5.8 ADC x 通道预选寄存器 (ADCx_PCSEL)（x=1 到 3）

ADC x channel preselection register

偏移地址：0x1C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PCSEL19	PCSEL18	PCSEL17	PCSEL16
												rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PCSEL15	PCSEL14	PCSEL13	PCSEL12	PCSEL11	PCSEL10	PCSEL9	PCSEL8	PCSEL7	PCSEL6	PCSEL5	PCSEL4	PCSEL3	PCSEL2	PCSEL1	PCSEL0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:20 保留，必须保持复位值。

位 19:0 PCSELx：通道 x ($V_{INP[i]}$) 预选 (Channel x (VINP[i]) pre selection)

这些位由软件写入，用于预选 IO 实例中要转换的输入通道。

0：未预选通道 x (Vinp x) 进行转换，该通道的 ADC 转换结果显示的结果不正确。

1：预选输入通道 x (Vinp x) 进行转换

注： 仅当 ADSTART=0 且 JADSTART=0 时（这可确保当前未进行任何转换），才允许通过软件对这些位执行写操作。

25.5.9 ADC x 看门狗阈值寄存器 1 (ADCx_LTR1)（x=1 到 3）

ADC x watchdog threshold register 1

偏移地址：0x20

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	LTR1[25:16]									
						rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LTR1[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:26 保留，必须保持复位值。

位 25:0 **LTR1[25:0]**：模拟看门狗 1 阈值下限 (Analog watchdog 1 lower threshold)

这些位由软件写入，用于定义模拟看门狗 1 的阈值下限。

请参见*第 25.3.30 节：模拟窗口看门狗（AWD1EN、JAWD1EN、AWD1SGL、AWD1CH、AWD2CH、AWD3CH、AWD_HTRy、AWD_LTRy、AWDy）*。

注： 仅当 ADSTART=0 且 JADSTART=0 时（这可确保当前未进行任何转换），才允许通过软件对这些位执行写操作。

25.5.10 ADC x 看门狗阈值寄存器 1 (ADCx_HTR1)（x=1 到 3）

ADC x watchdog threshold register 1

偏移地址：0x24

复位值：0x03FF FFFF

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	HTR1[25:16]									
						rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HTR1[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:26 保留，必须保持复位值。

位 25:0 **HTR1[25:0]**：模拟看门狗 1 阈值上限 (Analog watchdog 1 higher threshold)

这些位由软件写入，用于定义模拟看门狗的阈值上限。

请参见*第 25.3.30 节：模拟窗口看门狗（AWD1EN、JAWD1EN、AWD1SGL、AWD1CH、AWD2CH、AWD3CH、AWD_HTRy、AWD_LTRy、AWDy）*。

注： 仅当 ADSTART=0 且 JADSTART=0 时（这可确保当前未进行任何转换），才允许通过软件对这些位执行写操作。

25.5.11 ADC x 常规序列寄存器 1 (ADCx_SQR1)（x=1 到 3）

ADC x regular sequence register 1

偏移地址：0x30

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	SQ4[4:0]					Res.	SQ3[4:0]					Res.	SQ2[4]
			rw	rw	rw	rw	rw		rw	rw	rw	rw	rw		rw

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

位 31:29 保留，必须保持复位值。

位 28:24 **SQ4[4:0]**：常规序列中的第四次转换 (4th conversion in regular sequence)
通过软件写入这些位，并将通道编号 (0..19) 分配为常规转换序列中的第四次转换。

位 23 保留，必须保持复位值。

位 22:18 **SQ3[4:0]**：常规序列中的第三次转换 (3th conversion in regular sequence)
通过软件写入这些位，并将通道编号 (0..19) 分配为常规转换序列中的第三次转换。
注： 仅当 ADSTART=0 时（这可确保当前未进行任何常规转换），才允许通过软件对这些位执行写操作。

位 17 保留，必须保持复位值。

位 16:12 **SQ2[4:0]**：常规序列中的第二次转换 (2th conversion in regular sequence)
通过软件写入这些位，并将通道编号 (0..19) 分配为常规转换序列中的第二次转换。
注： 仅当 ADSTART=0 时（这可确保当前未进行任何常规转换），才允许通过软件对这些位执行写操作。

位 11 保留，必须保持复位值。

位 10:6 **SQ1[4:0]**：常规序列中的第一次转换 (1th conversion in regular sequence)
通过软件写入这些位，并将通道编号 (0..19) 分配为常规转换序列中的第一次转换。
注： 仅当 ADSTART=0 时（这可确保当前未进行任何常规转换），才允许通过软件对这些位执行写操作。

位 5:4 保留，必须保持复位值。

位 3:0 **L[3:0]**：常规通道序列长度 (Regular channel sequence length)
通过软件写入这些位可定义常规通道转换序列中的转换总数。
0000：1 次转换
0001：2 次转换
...
1111：16 次转换
注： 仅当 ADSTART=0 时（这可确保当前未进行任何常规转换），才允许通过软件对这些位执行写操作。

25.5.12 ADC x 常规序列寄存器 2 (ADCx_SQR2)（x=1 到 3）

ADC x regular sequence register 2

偏移地址：0x34

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	SQ9[4:0]					Res.	SQ8[4:0]					Res.	SQ7[4]
			rw	rw	rw	rw	rw		rw	rw	rw	rw	rw		rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SQ7[3:0]				Res.	SQ6[4:0]					Res.	SQ5[4:0]				
rw	rw	rw	rw		rw	rw	rw	rw	rw		rw	rw	rw	rw	rw

位 31:29 保留，必须保持复位值。

位 28:24 **SQ9[4:0]**：常规序列中的第九次转换 (9th conversion in regular sequence)

通过软件写入这些位，并将通道编号 (0..19) 分配为常规转换序列中的第九次转换。

注： 仅当 ADSTART=0 时（这可确保当前未进行任何常规转换），才允许通过软件对这些位执行写操作。

位 23 保留，必须保持复位值。

位 22:18 **SQ8[4:0]**：常规序列中的第八次转换 (8th conversion in regular sequence)

通过软件写入这些位，并将通道编号 (0..19) 分配为常规转换序列中的第八次转换。

注： 仅当 ADSTART=0 时（这可确保当前未进行任何常规转换），才允许通过软件对这些位执行写操作。

位 17 保留，必须保持复位值。

位 16:12 **SQ7[4:0]**：常规序列中的第七次转换 (7th conversion in regular sequence)

通过软件写入这些位，并将通道编号 (0..19) 分配为常规转换序列中的第七次转换。

注： 仅当 ADSTART=0 时（这可确保当前未进行任何常规转换），才允许通过软件对这些位执行写操作。

位 11 保留，必须保持复位值。

位 10:6 **SQ6[4:0]**：常规序列中的第六次转换 (6th conversion in regular sequence)

通过软件写入这些位，并将通道编号 (0..19) 分配为常规转换序列中的第六次转换。

注： 仅当 ADSTART=0 时（这可确保当前未进行任何常规转换），才允许通过软件对这些位执行写操作。

位 5 保留，必须保持复位值。

位 4:0 **SQ5[4:0]**：常规序列中的第五次转换 (5th conversion in regular sequence)

通过软件写入这些位，并将通道编号 (0..19) 分配为常规转换序列中的第五次转换。

注： 仅当 ADSTART=0 时（这可确保当前未进行任何常规转换），才允许通过软件对这些位执行写操作。

25.5.13 ADC x 常规序列寄存器 3 (ADCx_SQR3)（x=1 到 3）

ADC x regular sequence register 3

偏移地址：0x38

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	SQ14[4:0]					Res.	SQ13[4:0]					Res.	SQ12[4]
			rw	rw	rw	rw	rw		rw	rw	rw	rw	rw		rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SQ12[3:0]				Res.	SQ11[4:0]					Res.	SQ10[4:0]				
rw	rw	rw	rw		rw	rw	rw	rw	rw		rw	rw	rw	rw	rw

位 31:29 保留，必须保持复位值。

位 28:24 **SQ14[4:0]**：常规序列中的第十四次转换 (14th conversion in regular sequence)

通过软件写入这些位，并将通道编号 (0..19) 分配为常规转换序列中的第十四次转换。

注： 仅当 ADSTART=0 时（这可确保当前未进行任何常规转换），才允许通过软件对这些位执行写操作。

位 23 保留，必须保持复位值。

位 22:18 **SQ13[4:0]**：常规序列中的第十三次转换 (13th conversion in regular sequence)

通过软件写入这些位，并将通道编号 (0..19) 分配为常规转换序列中的第十三次转换。

注： 仅当 ADSTART=0 时（这可确保当前未进行任何常规转换），才允许通过软件对这些位执行写操作。

位 17 保留，必须保持复位值。

位 16:12 **SQ12[4:0]**：常规序列中的第十二次转换 (12th conversion in regular sequence)

通过软件写入这些位，并将通道编号 (0..19) 分配为常规转换序列中的第十二次转换。

注： 仅当 ADSTART=0 时（这可确保当前未进行任何常规转换），才允许通过软件对这些位执行写操作。

位 11 保留，必须保持复位值。

位 10:6 **SQ11[4:0]**：常规序列中的第十一次转换 (11th conversion in regular sequence)

通过软件写入这些位，并将通道编号 (0..19) 分配为常规转换序列中的第十一次转换。

注： 仅当 ADSTART=0 时（这可确保当前未进行任何常规转换），才允许通过软件对这些位执行写操作。

位 5 保留，必须保持复位值。

位 4:0 **SQ10[4:0]**：常规序列中的第十次转换 (10th conversion in regular sequence)

通过软件写入这些位，并将通道编号 (0..19) 分配为常规转换序列中的第十次转换。

注： 仅当 ADSTART=0 时（这可确保当前未进行任何常规转换），才允许通过软件对这些位执行写操作。

25.5.14 ADC x 常规序列寄存器 4 (ADCx_SQR4)（x=1 到 3）

ADC x regular sequence register 4

偏移地址：0x3C

复位值：0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	SQ16[4:0]					Res.	SQ15[4:0]				
					rw	rw	rw	rw	rw		rw	rw	rw	rw	rw

位 31:11 保留，必须保持复位值。

位 10:6 **SQ16[4:0]**：常规序列中的第十六次转换 (16th conversion in regular sequence)
通过软件写入这些位，并将通道编号 (0..19) 分配为常规转换序列中的第十六次转换。
注： 仅当 ADSTART=0 时（这可确保当前未进行任何常规转换），才允许通过软件对这些位执行写操作。

位 5 保留，必须保持复位值。

位 4:0 **SQ15[4:0]**：常规序列中的第十五次转换 (15th conversion in regular sequence)
通过软件写入这些位，并将通道编号 (0..19) 分配为常规转换序列中的第十五次转换。
注： 仅当 ADSTART=0 时（这可确保当前未进行任何常规转换），才允许通过软件对这些位执行写操作。

25.5.15 ADC x 常规数据寄存器 (ADCx_DR)（x=1 到 3）

ADC x regular Data Register

偏移地址：0x40

复位值：0x0000 0000

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

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

位 31:0 **RDATA[31:0]**：已转换常规数据 (Regular Data converted)
这些位为只读。其中包含上一转换常规通道的转换结果。数据有左对齐和右对齐两种方式，如*第 25.3.27 节：数据管理*所述。

25.5.16 ADC x 注入序列寄存器 (ADCx_JSQR)（x=1 到 3）

ADC x injected sequence register

偏移地址：0x4C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
JSQ4[4:0]					Res.	JSQ3[4:0]					Res.	JSQ2[4:1]			
rw	rw	rw	rw	rw		rw	rw	rw	rw	rw		rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
JSQ2[0]	Res.	JSQ1[4:0]					JEXTEN[1:0]		JEXTSEL[4:0]					JL[1:0]	
rw		rw	rw	rw	rw	rw	rw		rw	rw	rw	rw	rw	rw	rw

位 31:27 **JSQ4[4:0]**：注入序列中的第四次转换 (4th conversion in the injected sequence)
通过软件写入这些位，并将通道编号 (0..19) 分配为注入转换序列中的第四次转换。
注： ADC 使能后 (ADEN=1)，允许随时通过软件写入这些位。

位 26 保留，必须保持复位值。

位 25:21 **JSQ3[4:0]**：注入序列中的第三次转换 (3rd conversion in the injected sequence)
通过软件写入这些位，并将通道编号 (0..19) 分配为注入转换序列中的第三次转换。
注： ADC 使能后 (ADEN=1)，允许随时通过软件写入这些位。

位 20 保留，必须保持复位值。

位 19:15 **JSQ2[4:0]**：注入序列中的第二次转换 (2nd conversion in the injected sequence)
通过软件写入这些位，并将通道编号 (0..19) 分配为注入转换序列中的第二次转换。
注： ADC 使能后 (ADEN=1)，允许随时通过软件写入这些位。

位 14 保留，必须保持复位值。

位 13:9 **JSQ1[4:0]**：注入序列中的第一次转换 (1st conversion in the injected sequence)
通过软件写入这些位，并将通道编号 (0..19) 分配为注入转换序列中的第一次转换。
注： ADC 使能后 (ADEN=1)，允许随时通过软件写入这些位。

位 8:7 **JEXTEN[1:0]**：注入通道的外部触发使能和极性选择 (External Trigger Enable and Polarity Selection for injected channels)

通过软件将这些位置 1 和清零可选择外部触发极性和使能注入组的触发。
00：如果 JQDIS=0（队列使能），同时禁止硬件和软件触发检测
00：如果 JQDIS=1（队列禁止），禁止硬件触发检测（可通过软件启动转换）
01：在上升沿执行硬件触发检测
10：在下降沿执行硬件触发检测
11：在上升沿和下降沿都执行硬件触发检测

注： ADC 使能后 (ADEN=1)，允许随时通过软件写入这些位。
如果 JQM=1 且上下文队列为空，则会在内部禁止注入序列的软件和硬件触发（请参见第 25.3.22 节：注入转换的上下文队列）

位 6:2 **JEXTSEL[4:0]**：注入组的外部触发选择 (External Trigger Selection for injected group)

这些位可选择用于触发注入组转换的外部事件。
00000：事件 0
00001：事件 1
00010：事件 2
00011：事件 3
00100：事件 4
00101：事件 5
00110：事件 6
00111：事件 7
...
11111：事件 31：

注： ADC 使能后 (ADEN=1)，允许随时通过软件写入这些位。

位 1:0 **JL[1:0]**：注入通道序列长度 (Injected channel sequence length)

通过软件写入这些位可定义注入通道转换序列中的转换总数。
00：1 次转换
01：2 次转换
10：3 次转换
11：4 次转换

注： ADC 使能后 (ADEN=1)，允许随时通过软件写入这些位。

25.5.17 ADC x 偏移寄存器 (ADCx_OFRy)（x=1 到 3）

ADC x offset register

偏移地址：0x60 + 0x04 * (y-1)，y= 1 到 4

复位值：0x0000 0000

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

位 31 **SSATE**：有符号饱和使能 (Signed saturation Enable)

此位通过软件写入，用于使能或禁止有符号饱和特性。

仅可为 8 位和 16 位数据格式使能此位（详细信息请参见 *数据寄存器、数据对齐和偏移（ADCx_DR、ADCx_JDRy、OFFSETy、OFFSETy_CH、OVSS、LSHIFT、RSHIFT、SSATE）*）。

0：减去偏移，保持数据完整性，并扩展结果大小（9 位和 17 位有符号格式）。

1：减去偏移，使结果饱和以保持结果大小。

注：　仅当 ADSTART=0 且 JADSTART=0 时（这可确保当前未进行任何转换），才允许通过软件对此位执行写操作。

位 30:26 **OFFSETy_CH[4:0]**：数据偏移 y 的通道选择 (Channel selection for the Data offset y)

这些位通过软件写入，用于定义编程到 OFFSETy[25:0] 位中的偏移将应用于的通道。

注：　仅当 ADSTART=0 且 JADSTART=0 时（这可确保当前未进行任何转换），才允许通过软件对这些位执行写操作。

位 25:0 **OFFSETy[25:0]**：编程到 OFFSETy_CH[4:0] 位中的通道对应的数据偏移 y (Data offset y for the channel programmed into bits OFFSETy_CH[4:0])

这些位通过软件写入，用于定义在转换通道（可以是常规通道或注入通道）时从原始转换数据中减去的偏移 y。应用数据偏移 y 的通道必须在 OFFSETy_CH[4:0] 位中编程。转换结果可从 ADCx_DR（常规转换）或 ADCx_JDRyi 寄存器（注入转换）中读取。

当 OFFSETy[25:0] 位域复位时，会禁止偏移补偿。

注：　仅当 ADSTART=0 且 JADSTART=0 时（这可确保当前未进行任何转换），才允许通过软件对这些位执行写操作。

如果多个偏移 (OFFSETy) 指向同一通道，相减时只会考虑使用 x 值最小的偏移。
例如：如果 OFFSET1_CH[4:0]=4 且 OFFSET2_CH[4:0]=4，转换通道 4 时会减去 OFFSET1[25:0]。

25.5.18 ADC x 注入数据寄存器 (ADCx_JDRy)（x=1 到 3）

ADC x injected data register

偏移地址：0x80 + 0x04 * (y-1)，y= 1 到 4

复位值：0x0000 0000

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

位 31:0 **JDATA[31:0]**：注入数据 (Injected data)

这些位为只读。它们包括来自注入通道 y 的转换结果。数据有左对齐和右对齐两种方式，如 *第 25.3.27 节：数据管理* 所述。

25.5.19 ADC x 模拟看门狗 2 配置寄存器 (ADCx_AWD2CR)（x=1 到 3）

ADC x Analog Watchdog 2 Configuration Register

偏移地址：0xA0

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	AWD2CH[19:16]			
												rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AWD2CH[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:20 保留，必须保持复位值。

位 19:0 **AWD2CH[19:0]**：模拟看门狗 2 通道选择 (Analog watchdog 2 channel selection)

这些位将由软件置 1 和清零。它们用于使能和选择由模拟看门狗 2 监控的输入通道。

AWD2CH[i] = 0：不通过 AWD2 监控 ADC 模拟输入通道 i

AWD2CH[i] = 1：通过 AWD2 监控 ADC 模拟输入通道 i

当 AWD2CH[19:0] = 000..0 时，会禁止模拟看门狗 2

注： *通过 AWD2CH 选择的通道必须也在 SQRi 或 JSQRi 寄存器中选择。*

仅当 ADSTART=0 且 JADSTART=0 时（这可确保当前未进行任何转换），才允许通过软件对这些位执行写操作。

25.5.20 ADC x 模拟看门狗 3 配置寄存器 (ADCx_AWD3CR)（x=1 到 3）

ADC x Analog Watchdog 3 Configuration Register

偏移地址：0xA4

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	AWD3CH[19:16]			
												rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AWD3CH[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:20 保留，必须保持复位值。

位 19:0 **AWD3CH[19:0]**：模拟看门狗 3 通道选择 (Analog watchdog 3 channel selection)

这些位将由软件置 1 和清零。它们用于使能和选择由模拟看门狗 3 监控的输入通道。

AWD3CH[i] = 0：不通过 AWD3 监控 ADC 模拟输入通道 i

AWD3CH[i] = 1：通过 AWD3 监控 ADC 模拟输入通道 i

当 AWD3CH[19:0] = 000..0 时，会禁止模拟看门狗 3

注： 通过 AWD3CH 选择的通道必须也在 SQRi 或 JSQRi 寄存器中选择。

仅当 ADSTART=0 且 JADSTART=0 时（这可确保当前未进行任何转换），才允许通过软件对这些位执行写操作。

25.5.21 ADC x 看门狗阈值下限寄存器 2 (ADCx_LTR2)（x=1 到 3）

ADC x watchdog lower threshold register 2

偏移地址：0xB0

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	LTR2[25:16]									
						rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LTR2[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:26 保留，必须保持复位值。

位 25:0 **LTR2[25:0]**：模拟看门狗 3 阈值下限 (Analog watchdog 3 lower threshold)

这些位通过软件写入，用于为模拟看门狗 2 定义阈值下限。

请参见*第 25.3.30 节：模拟窗口看门狗（AWD1EN、JAWD1EN、AWD1SGL、AWD1CH、AWD2CH、AWD3CH、AWD_HTRy、AWD_LTRy、AWDy）*。

注： 仅当 ADSTART=0 且 JADSTART=0 时（这可确保当前未进行任何转换），才允许通过软件对这些位执行写操作。

25.5.22 ADC x 看门狗阈值上限寄存器 2 (ADCx_HTR2)（x=1 到 3）

ADC x watchdog higher threshold register 2

偏移地址：0xB4

复位值：0x03FF FFFF

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	HTR2[25:16]									
						rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HTR2[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:26 保留，必须保持复位值。

位 25:0 **HTR2[25:0]**：模拟看门狗 2 阈值上限 (Analog watchdog 2 higher threshold)
这些位通过软件写入，用于为模拟看门狗 2 定义阈值上限。
请参见*第 25.3.30 节：模拟窗口看门狗（AWD1EN、JAWD1EN、AWD1SGL、AWD1CH、AWD2CH、AWD3CH、AWD_HTRy、AWD_LTRy、AWDy）*。

注： 仅当 ADSTART=0 且 JADSTART=0 时（这可确保当前未进行任何转换），才允许通过软件对这些位执行写操作。

25.5.23 ADC x 看门狗阈值下限寄存器 3 (ADCx_LTR3)（x=1 到 3）

ADC x watchdog lower threshold register 3

偏移地址：0xB8

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	LTR3[25:16]									
						rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LTR3[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:26 保留，必须保持复位值。

位 25:0 **LTR3[25:0]**：模拟看门狗 3 阈值下限 (Analog watchdog 3 lower threshold)
这些位由软件写入，用于定义模拟看门狗 3 的阈值下限。
请参见*第 25.3.30 节：模拟窗口看门狗（AWD1EN、JAWD1EN、AWD1SGL、AWD1CH、AWD2CH、AWD3CH、AWD_HTRy、AWD_LTRy、AWDy）*。

注： 仅当 ADSTART=0 且 JADSTART=0 时（这可确保当前未进行任何转换），才允许通过软件对这些位执行写操作。

25.5.24 ADC x 看门狗阈值上限寄存器 3 (ADCx_HTR3)（x=1 到 3）

ADC x watchdog higher threshold register 3

偏移地址：0xBC

复位值：0x03FF FFFF

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	HTR3[25:16]									
						rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HTR3[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:26 保留，必须保持复位值。

位 25:0 **HTR3[25:0]**：模拟看门狗 3 阈值上限 (Analog watchdog 3 higher threshold)
这些位通过软件写入，用于定义模拟看门狗 3 的阈值上限。
请参见*第 25.3.30 节：模拟窗口看门狗（AWD1EN、JAWD1EN、AWD1SGL、AWD1CH、AWD2CH、AWD3CH、AWD_HTRy、AWD_LTRy、AWDy）*。

注： 仅当 ADSTART=0 且 JADSTART=0 时（这可确保当前未进行任何转换），才允许通过软件对这些位执行写操作。

25.5.25 ADC x 差分模式选择寄存器 (ADCx_DIFSEL)（x=1 到 3）

ADC x Differential Mode Selection register

偏移地址：0xC0

复位值：0x0000 0000

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

位 31:20 保留，必须保持复位值。

位 19:0 **DIFSEL[19:0]**：通道 19 到 0 的差分模式 (Differential mode for channels 19 to 0)
这些位将由软件置 1 和清零。它们用于选择将通道配置为单端模式或差分模式。
DIFSEL[i] = 0：将 ADC 模拟输入通道 i 配置为单端模式
DIFSEL[i] = 1：将 ADC 模拟输入通道 i 配置为差分模式

注： 仅当 ADC 已禁止时（ADCAL=0、JADSTART=0、JADSTP=0、ADSTART=0、ADSTP=0、ADDIS=0 且 ADEN=0），才允许通过软件对这些位执行写操作。

25.5.26 ADC x 校准系数寄存器 (ADCx_CALFACT)（x=1 到 3）

ADC x Calibration Factors register

偏移地址：0xC4

复位值：0x0000 0000

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

位 31:27 保留，必须保持复位值。

位 26:16 **CALFACT_D[10:0]**：差分模式下的校准系数 (Calibration Factors in differential mode)

这些位可由硬件或软件写入。

差分输入校准完成后，会立即由硬件更新为校准系数。

软件可向这些位写入新的校准系数。如果新校准系数不同于当前存储于模拟 ADC 中的校准系数，启动新的差分校准后，会立即应用新校准系数。

注： 仅当 ADEN=1、ADSTART=0 且 JADSTART=0 时（ADC 已使能、当前未执行任何校准和转换），才允许通过软件对这些位执行写操作。

位 15:11 保留，必须保持复位值。

位 10:0 **CALFACT_S[10:0]**：单端模式下的校准系数 (Calibration Factors In Single-Ended mode)

这些位可由硬件或软件写入。

单端输入校准完成后，会立即由硬件更新为校准系数。

软件可向这些位写入新的校准系数。如果新校准系数不同于当前存储于模拟 ADC 中的校准系数，启动新的单端校准后，会立即应用新校准系数。

注： 仅当 ADEN=1、ADSTART=0 且 JADSTART=0 时（ADC 已使能、当前未执行任何校准和转换），才允许通过软件对这些位执行写操作。

25.5.27 ADC x 校准系数寄存器 2 (ADCx_CALFACT2)（x=1 到 3）

ADC x Calibration Factor register 2

偏移地址：0xC8

复位值：0x0000 0000

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

位 31:30 保留，必须保持复位值。

位 29:0 **LINCALFACT[29:0]**：线性度校准系数 (Linearity Calibration Factor)

这些位可由硬件或软件写入。

这些位存储 160 位线性度校准系数中的 30 位。

单端输入校准完成后，会立即由硬件更新为校准系数。

软件可向这些位写入新的校准系数。如果新校准系数不同于当前存储于模拟 ADC 中的校准系数，启动新的单端校准后，会立即应用新校准系数。

注： 仅当 ADEN=1、ADSTART=0 且 JADSTART=0 时（ADC 已使能、当前未执行任何校准和转换），才允许通过软件对这些位执行写操作。

25.6 ADC 寄存器

这些寄存器定义主 ADC 和从 ADC 共用的控制和状态寄存器：

25.6.1 ADC x 通用状态寄存器 (ADCx_CSR)（x=12 或 3）

ADC x common status register

偏移地址：0x00（该偏移地址与主 ADC 基址 + 0x300 相关）

复位值：0x0000 0000

该寄存器可提供不同 ADC 的状态位图像。但是，它为只读形式且不允许将不同的状态位清零。必须在对应的 ADCx_ISR 寄存器中将其写为 0，才能将各个状态位清零。

ADC1 和 ADC2 由相同的接口控制，而 ADC3 单独控制。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	JQOVF_SLV	AWD3_SLV	AWD2_SLV	AWD1_SLV	JEOS_SLV	JEOC_SLV	OVR_SLV	EOS_SLV	EOC_SLV	EOSMP_SLV	ADRDY_SLV
					r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	JQOVF_MST	AWD3_MST	AWD2_MST	AWD1_MST	JEOS_MST	JEOC_MST	OVR_MST	EOS_MST	EOC_MST	EOSMP_MST	ADRDY_MST
					r	r	r	r	r	r	r	r	r	r	r

位 31:27 保留，必须保持复位值。

位 26 **JQOVF_SLV**：从 ADC 的注入上下文队列溢出标志 (Injected Context Queue Overflow flag of the slave ADC)
该位是相应 ADCx+1_ISR 寄存器中 JQOVF 位的副本。

位 25 **AWD3_SLV**：从 ADC 的模拟看门狗 3 标志 (Analog watchdog 3 flag of the slave ADC)
该位是相应 ADCx+1_ISR 寄存器中 AWD3 位的副本。

位 24 **AWD2_SLV**：从 ADC 的模拟看门狗 2 标志 (Analog watchdog 2 flag of the slave ADC)
该位是相应 ADCx+1_ISR 寄存器中 AWD2 位的副本。

位 23 **AWD1_SLV**：从 ADC 的模拟看门狗 1 标志 (Analog watchdog 1 flag of the slave ADC)
该位是相应 ADCx+1_ISR 寄存器中 AWD1 位的副本。

位 22 **JEOS_SLV**：从 ADC 注入序列结束标志 (End of injected sequence flag of the slave ADC)
该位是相应 ADCx+1_ISR 寄存器中 JEOS 位的副本。

位 21 **JEOC_SLV**：从 ADC 注入转换结束标志 (End of injected conversion flag of the slave ADC)
该位是相应 ADCx+1_ISR 寄存器中 JEOC 位的副本。

位 20 **OVR_SLV**：从 ADC 的溢出标志 (Overrun flag of the slave ADC)
该位是相应 ADCx+1_ISR 寄存器中 OVR 位的副本。

位 19 **EOS_SLV**：从 ADC 常规序列结束标志 (End of regular sequence flag of the slave ADC)
该位是相应 ADCx+1_ISR 寄存器中 EOS 位的副本。

位 18 **EOC_SLV**：从 ADC 常规转换结束标志 (End of regular conversion of the slave ADC)
该位是相应 ADCx+1_ISR 寄存器中 EOC 位的副本。

位 17 **EOSMP_SLV**：从 ADC 采样阶段结束标志 (End of Sampling phase flag of the slave ADC)
该位是相应 ADCx+1_ISR 寄存器中 EOSMP2 位的副本。

位 16 **ADRDY_SLV**：从 ADC 就绪 (Slave ADC ready)
该位是相应 ADCx+1_ISR 寄存器中 ADRDY 位的副本。

位 15:11 保留，必须保持复位值。

位 10 **JQOVF_MST**：主 ADC 的注入上下文队列溢出标志 (Injected Context Queue Overflow flag of the master ADC)
该位是相应 ADCx_ISR 寄存器中 JQOVF 位的副本。

位 9 **AWD3_MST**：主 ADC 的模拟看门狗 3 标志 (Analog watchdog 3 flag of the master ADC)
该位是相应 ADCx_ISR 寄存器中 AWD3 位的副本。

位 8 **AWD2_MST**：主 ADC 的模拟看门狗 2 标志 (Analog watchdog 2 flag of the master ADC)
该位是相应 ADCx_ISR 寄存器中 AWD2 位的副本。

位 7 **AWD1_MST**：主 ADC 的模拟看门狗 1 标志 (Analog watchdog 1 flag of the master ADC)
该位是相应 ADCx_ISR 寄存器中 AWD1 位的副本。

位 6 **JEOS_MST**：主 ADC 注入序列结束标志 (End of injected sequence flag of the master ADC)
该位是相应 ADCx_ISR 寄存器中 JEOS 位的副本。

位 5 **JEOC_MST**：主 ADC 注入转换结束标志 (End of injected conversion flag of the master ADC)
该位是相应 ADCx_ISR 寄存器中 JEOC 位的副本。

位 4 **OVR_MST**：主 ADC 的溢出标志 (Overrun flag of the master ADC)
该位是相应 ADCx_ISR 寄存器中 OVR 位的副本。

位 3 **EOS_MST**：主 ADC 常规序列结束标志 (End of regular sequence flag of the master ADC)
该位是相应 ADCx_ISR 寄存器中 EOS 位的副本。

位 2 **EOC_MST**：主 ADC 常规转换结束标志 (End of regular conversion of the master ADC)
该位是相应 ADCx_ISR 寄存器中 EOC 位的副本。

位 1 **EOSMP_MST**：主 ADC 采样阶段结束标志 (End of Sampling phase flag of the master ADC)
该位是相应 ADCx_ISR 寄存器中 EOSMP 位的副本。

位 0 **ADRDY_MST**：主 ADC 就绪 (Master ADC ready)
该位是相应 ADCx_ISR 寄存器中 ADRDY 位的副本。

25.6.2 ADC x 通用控制寄存器 (ADCx_CCR)（x=12 或 3）

ADC x common control register

偏移地址：0x08（该偏移地址与主 ADC 基址 + 0x300 相关）

复位值：0x0000 0000

ADC1 和 ADC2 由相同的接口控制，而 ADC3 单独控制。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	VBATEN	VSENSEEN	VREFEN	PRESC[3:0]				CKMODE[1:0]	
							rw	rw	rw	rw	rw	rw	rw	rw	rw

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

位 31:25 保留，必须保持复位值。

位 24 **VBATEN**：VBAT 使能 (VBAT enable)

此位由软件置 1 和清零，用于控制 V_{BAT} 通道。

0：禁止 V_{BAT} 通道

1：使能 V_{BAT} 通道

注： 仅当 ADC 已禁止时（ADCAL=0、JADSTART=0、ADSTART=0、ADSTP=0、ADDIS=0 且 ADEN=0），才允许通过软件对此位执行写操作。

位 23 **VSENSEEN**：温度传感器电压使能 (Temperature sensor voltage enable)

此位由软件置 1 和清零，用于控制 V_{SENSE} 通道。

0：禁止温度传感器通道

1：使能温度传感器通道

注： 仅当 ADC 已禁止时（ADCAL=0、JADSTART=0、ADSTART=0、ADSTP=0、ADDIS=0 且 ADEN=0），才允许通过软件对此位执行写操作。

位 22 **VREFEN**：V_{REFINT} 使能 (V_{REFINT} enable)

通过软件将该位置 1 和清零可使能/禁止 VREFINT 通道。

0：禁止 V_{REFINT} 通道

1：使能 V_{REFINT} 通道

注： 仅当 ADC 已禁止时（ADCAL=0、JADSTART=0、ADSTART=0、ADSTP=0、ADDIS=0 且 ADEN=0），才允许通过软件对此位执行写操作。

位 21:18 **PRESC[3:0]**：ADC 预分频系数 (ADC prescaler)

这些位由软件置 1 和清零，用于选择 ADC 的时钟频率。该时钟为所有 ADC 所共用。

0000：输入 ADC 时钟未分频
0001：输入 ADC 时钟 2 分频
0010：输入 ADC 时钟 4 分频
0011：输入 ADC 时钟 6 分频
0100：输入 ADC 时钟 8 分频
0101：输入 ADC 时钟 10 分频
0110：输入 ADC 时钟 12 分频
0111：输入 ADC 时钟 16 分频
1000：输入 ADC 时钟 32 分频
1001：输入 ADC 时钟 64 分频
1010：输入 ADC 时钟 128 分频
1011：输入 ADC 时钟 256 分频
其它：保留

注：　仅当 ADC 已禁止时（ADCAL=0、JADSTART=0、ADSTART=0、ADSTP=0、ADDIS=0 且 ADEN=0），才允许通过软件对这些位执行写操作。仅当 CKMODE[1:0] = 0b00 时，才会应用 ADC 预分频值。

位 17:16 **CKMODE[1:0]**：ADC 时钟模式 (ADC clock mode)

这些位由软件置 1 和清零，用于定义 ADC 时钟方案（主 ADC 和从 ADC 共用）：

00：CK_ADCx (x=1 到 3)（异步时钟模式），在产品级生成（请参见*复位和时钟控制 (RCC) 部分*）。

01：adc_hclk/1（同步时钟模式）。仅当 AHB 时钟预分频器设为 1（RCC_CFGR 寄存器中的 HPRE[3:0] = 0xxx）且系统时钟占空比为 50% 时，才必须使能此配置。

10：adc_hclk/2（同步时钟模式）

11：adc_hclk/4（同步时钟模式）

在所有同步时钟模式下，从定时器触发到转换开始的延迟过程中，不存在抖动。

注：　仅当 ADC 已禁止时（ADCAL=0、JADSTART=0、ADSTART=0、ADSTP=0、ADDIS=0 且 ADEN=0），才允许通过软件对这些位执行写操作。

位 15:14 **DAMDF[1:0]**：双重 ADC 模式数据格式 (Dual ADC Mode Data Format)

此位域由软件置 1 和清零。它指定通用该数据寄存器 ADCx_CDR 中的数据格式。

00：双重 ADC 模式，无数据封装（未使用 ADCx_CDR 和 ADCx_CDR2 寄存器）。
01：保留
10：数据格式化模式，用于 32 位到 10 位分辨率
11：数据格式化模式，用于 8 位分辨率

注：　仅当 ADSTART=0 时（这可确保当前未进行任何常规转换），才允许通过软件对这些位执行写操作。

位 13:12 保留，必须保持复位值。

位 11:8 **DELAY**：2 个采样阶段之间的延迟 (Delay between 2 sampling phases)

这些位将由软件置 1 和清零。这些位在双重交错模式下使用。有关 ADC 分辨率与 DELAY 位值的关系，请参见*表 202*。

注： 仅当 ADC 已禁止时（ADCAL=0、JADSTART=0、ADSTART=0、ADSTP=0、ADDIS=0 且 ADEN=0），才允许通过软件对这些位执行写操作。

位 7:5 保留，必须保持复位值。

位 4:0 **DUAL[4:0]**：双重 ADC 模式选择 (Dual ADC mode selection)

通过软件写入这些位可选择操作模式。

所有 ADC 均独立：

00000：独立模式

00001 到 01001：双重模式，主 ADC 与从 ADC 一起工作

00001：常规同步 + 注入同步组合模式

00010：常规同步 + 交替触发组合模式

00011：交错 + 注入同步组合模式

00100：保留

00101：仅注入同步模式

00110：仅常规同步模式

00111：仅交错模式

01001：仅交替触发模式

其它所有组合均需保留且不允许编程

注： 仅当 ADC 已禁止时（ADCAL=0、JADSTART=0、ADSTART=0、ADSTP=0、ADDIS=0 且 ADEN=0），才允许通过软件对这些位执行写操作。

表 202. DELAY 位与 ADC 分辨率的关系

DELAY 位	16位分辨率	14位分辨率	12位分辨率	10位分辨率	8位分辨率
0000	1.5 * $T_{adc_ker_ck}$	1.5 * $T_{adc_ker_ck}$	1.5 * $T_{adc_ker_ck}$	1.5 * $T_{adc_ker_ck}$	1.5 * $T_{adc_ker_ck}$
0001	2.5 * $T_{adc_ker_ck}$	2.5 * $T_{adc_ker_ck}$	2.5 * $T_{adc_ker_ck}$	2.5 * $T_{adc_ker_ck}$	2.5 * $T_{adc_ker_ck}$
0010	3.5 * $T_{adc_ker_ck}$	3.5 * $T_{adc_ker_ck}$	3.5 * $T_{adc_ker_ck}$	3.5 * $T_{adc_ker_ck}$	3.5 * $T_{adc_ker_ck}$
0011	4.5 * $T_{adc_ker_ck}$	4.5 * $T_{adc_ker_ck}$	4.5 * $T_{adc_ker_ck}$	4.5 * $T_{adc_ker_ck}$	4.5 * $T_{adc_ker_ck}$
0100	5.5 * $T_{adc_ker_ck}$	5.5 * $T_{adc_ker_ck}$	5.5 * $T_{adc_ker_ck}$	5.5 * $T_{adc_ker_ck}$	4.5 * $T_{adc_ker_ck}$
0101	6.5 * $T_{adc_ker_ck}$	6.5 * $T_{adc_ker_ck}$	6.5 * $T_{adc_ker_ck}$	5.5 * $T_{adc_ker_ck}$	4.5 * $T_{adc_ker_ck}$
0110	7.5 * $T_{adc_ker_ck}$	7.5 * $T_{adc_ker_ck}$	6.5 * $T_{adc_ker_c}$	5.5 * $T_{adc_ker_ck}$	4.5 * $T_{adc_ker_ck}$
0111	8.5 * $T_{adc_ker_ck}$	7.5 * $T_{adc_ker_ck}$	6.5 * $T_{adc_ker_ck}$	5.5 * $T_{adc_ker_ck}$	4.5 * $T_{adc_ker_ck}$
1000	8.5 * $T_{adc_ker_ck}$	7.5 * $T_{adc_ker_ck}$	6.5 * $T_{adc_ker_ck}$	5.5 * $T_{adc_ker_ck}$	4.5 * $T_{adc_ker_ck}$
其它值：保留	-	-	-	-	-

25.6.3 适用于双重模式的 ADC x 通用常规数据寄存器 (ADCx_CDR)（x=12 或 3）

ADC x common regular data register for dual mode

偏移地址：0x0C（该偏移地址与 ADC 基址 + 0x300 相关）

复位值：0x0000 0000

ADC1 和 ADC2 由相同的接口控制，而 ADC3 单独控制。

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

位 31:16 **RDATA_SLV[15:0]**：从 ADC 的常规数据 (Regular data of the slave ADC)

在双重模式下，这些位包含从 ADC 的常规数据。请参见*第 25.3.32 节：双重 ADC 模式*。

按*数据寄存器、数据对齐和偏移（ADCx_DR、ADCx_JDRy、OFFSETy、OFFSETy_CH、OVSS、LSHIFT、RSHIFT、SSATE）*一节所述应用数据对齐。

位 15:0 **RDATA_MST[15:0]**：主 ADC 的常规数据 (Regular data of the master ADC)。

在双重模式下，这些位包含主 ADC 的常规数据。请参见*第 25.3.32 节：双重 ADC 模式*。

按*数据寄存器、数据对齐和偏移（ADCx_DR、ADCx_JDRy、OFFSETy、OFFSETy_CH、OVSS、LSHIFT、RSHIFT、SSATE）*一节所述应用数据对齐。

在 MDMA=0b11 模式下，位 15:8 包含 SLV_ADCx_DR[7:0]，位 7:0 包含 MST_ADCx_DR[7:0]。

25.6.4 适用于 32 位双重模式的 ADC x 通用常规数据寄存器 (ADCx_CDR2)（x=12 或 3）

ADC x common regular data register for 32-bit dual mode

偏移地址：0x10（该偏移地址与 ADC 基址 + 0x300 相关）

复位值：0x0000 0000

ADC1 和 ADC2 由相同的接口控制，而 ADC3 单独控制。

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

位 31:0 **RDATA_ALT[31:0]**：主/从交替 ADC 的常规数据 (Regular data of the master/slave alternated ADC)

在双重模式下，这些位交替包含主 ADC 和从 ADC 的常规 32 位数据。请参见*第 25.3.32 节：双重 ADC 模式*。

按*数据寄存器、数据对齐和偏移（ADCx_DR、ADCx_JDRy、OFFSETy、OFFSETy_CH、OVSS、LSHIFT、RSHIFT、SSATE）*一节所述应用数据对齐。

25.6.5 ADC 寄存器映射

下表对 ADC 寄存器进行了汇总。

表 203. ADC 全局寄存器映射

偏移	寄存器
0x000 - 0x0D0	主 ADC1
0x0D4 - 0x0FC	保留
0x100 - 0x1D0	从ADC2
0x1D4 - 0x2FC	保留
0x300 - 0x310	主 ADC 和从 ADC 通用寄存器（ADC12 或 ADC3）

表 204. 每个 ADC 的 ADC 寄存器映射和复位值（主 ADC 偏移=0x000， 从 ADC 偏移=0x100）

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	**ADCx_ISR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	JQOVF	AWD3	AWD2	AWD1	JEOS	JEOC	OVR	EOS	EOC	EOSMP	ADRDY
	Reset value																						0	0	0	0	0	0	0	0	0	0	0
0x04	**ADCx_IER**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	JQOVFIE	AWD3IE	AWD2IE	AWD1IE	JEOSIE	JEOCIE	OVRIE	EOSIE	EOCIE	EOSMPIE	ADRDYIE
	Reset value																						0	0	0	0	0	0	0	0	0	0	0
0x08	**ADCx_CR**	ADCAL	ADCALDIF	DEEPPWD	ADVREGEN	LINCALRDYW6	LINCALRDYW5	LINCALRDYW4	LINCALRDYW3	LINCALRDYW2	LINCALRDYW1	Res.	Res.	Res.	Res.	Res.	ADCALLIN	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BOOST	Res.	Res.	JADSTP	ADSTP	JADSTART	ADSTART	ADDIS	ADEN
	Reset value	0	0	1	0	0	0	0	0	0	0						0							0			0	0	0	0	0	0
0x0C	**ADCx_CFGR**	JQDIS	AWD1CH[4:0]					JAUTO	JAWD1EN	AWD1EN	AWD1SGL	JQM	JDISCEN	DISCNUM [2:0]			DISCEN	Res.	AUTDLY	CONT	OVRMOD	EXTEN[1:0]		EXTSEL [4:0]					RES [2:0]			DMN GT [1:0]	
	Reset value	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x10	**ADCx_CFGR2**	Res.	Res.	Res.	Res.	Res.	Res.	OSR[9:0]										Res.	RSHIFT4	RSHIFT3	RSHIFT2	RSHIFT1	ROVSM	TROVS	OVSS[3:0]				Res.	Res.	Res.	JOVSE	ROVSE
	Reset value							0	0	0	0	0	0	0	0	0	0		0	0	0	0	0	0	0	0	0	0				0	0
0x14	**ADCx_SMPR1**	res.	Res.	SMP9[2:0]			SMP8[2:0]			SMP7[2:0]			SMP6[2:0]			SMP5[2:0]			SMP4[2:0]			SMP3[2:0]			SMP2[2:0]			SMP1[2:0]			SMP0[2:0]	
	Reset value			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x18	**ADCx_SMPR2**	Res.	Res.	Res.	Res.	Res.	SMP18 [2:0]			SMP17 [2:0]			SMP16 [2:0]			SMP15 [2:0]			SMP14 [2:0]			SMP13 [2:0]			SMP12 [2:0]			SMP11 [2:0]			SMP10 [2:0]	
	Reset value						0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x1C	**ADCx_PCSEL**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PCSEL19	PCSEL18	PCSEL17	PCSEL16	PCSEL15	PCSEL14	PCSEL13	PCSEL12	PCSEL11	PCSEL10	PCSEL9	PCSEL8	PCSEL7	PCSEL6	PCSEL5	PCSEL4	PCSEL3	PCSEL2	PCSEL1	PCSEL0
	Reset value													0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x20	**ADCx_LTR1**	Res.	Res.	Res.	Res.	Res.	Res.	LTR1[25:0]																									
	Reset value							0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x24	**ADCx_HTR1**	Res.	Res.	Res.	Res.	Res.	Res.	HTR1[25:0]																									
	Reset value							1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

表 204. 每个 ADC 的 ADC 寄存器映射和复位值（主 ADC 偏移=0x000，从 ADC 偏移=0x100）（续）

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x28	Reserved	Res.																															
0x2C	Reserved	Res.																															
0x30	**ADCx_SQR1**	Res.	Res.	Res.	SQ4[4:0]					Res.	SQ3[4:0]					Res.	SQ2[4:0]					Res.	SQ1[4:0]					Res.	Res.	L[3:0]			
	Reset value				0	0	0	0	0		0	0	0	0	0		0	0	0	0	0		0	0	0	0	0			0	0	0	0
0x34	**ADCx_SQR2**	Res.	Res.	Res.	SQ9[4:0]					Res.	SQ8[4:0]					Res.	SQ7[4:0]					Res.	SQ6[4:0]					Res.	SQ5[4:0]				
	Reset value				0	0	0	0	0		0	0	0	0	0		0	0	0	0	0		0	0	0	0	0		0	0	0	0	0	
0x38	**ADCx_SQR3**	Res.	Res.	Res.	SQ14[4:0]					Res.	SQ13[4:0]					Res.	SQ12[4:0]					Res.	SQ11[4:0]					Res.	SQ10[4:0]				
	Reset value				0	0	0	0	0		0	0	0	0	0		0	0	0	0	0		0	0	0	0	0		0	0	0	0	0	
0x3C	**ADCx_SQR4**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SQ16[4:0]					Res.	SQ15[4:0]				
	Reset value																						0	0	0	0	0		0	0	0	0	0
0x40	**ADCx_DR**	RDATA[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x44-0x48	Reserved	Res.																															
0x4C	**ADCx_JSQR**	JSQ4[4:0]					Res.	JSQ3[4:0]					Res.	JSQ2[4:0]				Res.	JSQ1[4:0]					JEXTEN[1:0]		JEXTSEL[4:0]					JL[1:0]		
	Reset value	0	0	0	0	0		0	0	0	0	0		0	0	0	0	0		0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x50-0x5C	Reserved	Res.																															
0x60	**ADCx_OFR1**	SSATE	OFFSET1_CH[4:0]						OFFSET1[25:0]																								
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x64	**ADCx_OFR2**	SSATE	OFFSET2_CH[4:0]						OFFSET2[25:0]																								
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x68	**ADCx_OFR3**	SSATE	OFFSET3_CH[4:0]						OFFSET3[25:0]																								
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x6C	**ADCx_OFR4**	SSATE	OFFSET4_CH[4:0]						OFFSET4[25:0]																								
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x70 0x7C	Reserved	Res.																															
0x80	**ADCx_JDR1**	JDATA1[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

表 204. 每个 ADC 的 ADC 寄存器映射和复位值（主 ADC 偏移=0x000，从 ADC 偏移=0x100）（续）

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x84	**ADCx_JDR2**	JDATA2[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x88	**ADCx_JDR3**	JDATA3[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x8C	**ADCx_JDR4**	JDATA4[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x90-0x9C	Reserved	Res.																															
0xA0	**ADCx_AWD2CR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	AWD2CH[19:0]																			
	Reset value													0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0xA4	**ADCx_AWD3CR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	AWD3CH[19:0]																			
	Reset value													0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0xA8-0xAC	Reserved	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
0xB0	**ADCx_LTR2**	Res.	Res.	Res.	Res.	Res.	Res.	LTR2[25:0]																									
	Reset value							0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0xB4	**ADCx_HTR2**	Res.	Res.	Res.	Res.	Res.	Res.	HTR2[25:0]																									
	Reset value							1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0xB8	**ADCx_LTR3**	Res.	Res.	Res.	Res.	Res.	Res.	LTR3[25:0]																									
	Reset value							0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0xBC	**ADCx_HTR3**	Res.	Res.	Res.	Res.	Res.	Res.	HTR3[25:0]																									
	Reset value							1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0xC0	**ADCx_DIFSEL**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DIFSEL[19:0]																			
	Reset value													0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0xC4	**ADCx_CALFACT**	Res.	Res.	Res.	Res.	Res.	CALFACT_D[10:0]											Res.	Res.	Res.	Res.	Res.	CALFACT_S[10:0]										
	Reset value						0	0	0	0	0	0	0	0	0	0	0						0	0	0	0	0	0	0	0	0	0	0
0xC8	**ADCx_ CALFACT2**	Res.	Res.	LINCALFACT[29:0]																													
	Reset value			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0xCC	Reserved	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
0x0D	Reserved																																

表 205. ADC 寄存器映射和复位值（主 ADC 和从 ADC 通用寄存器，偏移=0x300）

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	**ADCx_CSR**	Res.	Res.	Res.	Res.	Res.	JQOVF_SLV	AWD3_SLV	AWD2_SLV	AWD1_SLV	JEOS_SLV	JEOC_SLV	OVR_SLV	EOS_SLV	EOC_SLV	EOSMP_SLV	ADRDY_SLV	Res.	Res.	Res.	Res.	Res.	JQOVF_MST	AWD3_MST	AWD2_MST	AWD1_MST	JEOS_MST	JEOC_MST	OVR_MST	EOS_MST	EOC_MST	EOSMP_MST	ADRDY_MST
		slave ADC2																					master ADC1										
	Reset value						0	0	0	0	0	0	0	0	0	0	0						0	0	0	0	0	0	0	0	0	0	0
0x04	Reserved	Res.																															
0x08	**ADCx_CCR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	VBATEN	VSENSEEN	VREFEN	PRESC[3:0]				CKMODE[1:0]		DAMDF[1:0]		DMACFG	Res.	DELAY[3:0]				Res.	Res.	Res.	DUAL[4:0]				
	Reset value								0	0	0	0	0	0	0	0	0	0	0	0		0	0	0	0				0	0	0	0	0
0x0C	**ADCx_CDR**	RDATA_SLV[15:0]																RDATA_MST[15:0]															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x10	**ADCx_CDR2**	RDATA_ALT[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

有关寄存器边界地址的信息，请参见*第 2.2.2 节：存储器映射和寄存器边界地址*。

26 数模转换器 (DAC)

26.1 简介

DAC 模块是 12 位电压输出数模转换器。DAC 可以按 8 位或 12 位模式进行配置，并且可与 DMA 控制器配合使用。在 12 位模式下，数据可以采用左对齐或右对齐。DAC 有两个输出通道，每个通道各有一个转换器。在 DAC 双通道模式下，每个通道可以单独进行转换；当两个通道组合在一起同步执行更新操作时，也可以同时进行转换。可通过一个输入参考电压引脚 V_{REF+}（与其他模拟外设共享）来提高分辨率。还可以在同一输入上设置内部参考。请参见电压参考缓冲器 *(VREFBUF)* 一节。

如果 DAC 输出与输出焊盘断开连接并连接到片上外设，DAC_OUTx 引脚可用作通用输入/输出 (GPIO)。可选择使能 DAC 输出缓冲器，从而支持高驱动输出电流。可分别对每条 DAC 输出通道应用校准。DAC 输出通道支持低功耗模式，即采样和保持模式。

26.2 DAC 主要特性

DAC 的主要特性如下（请参见*图 195：DAC 通道框图*）

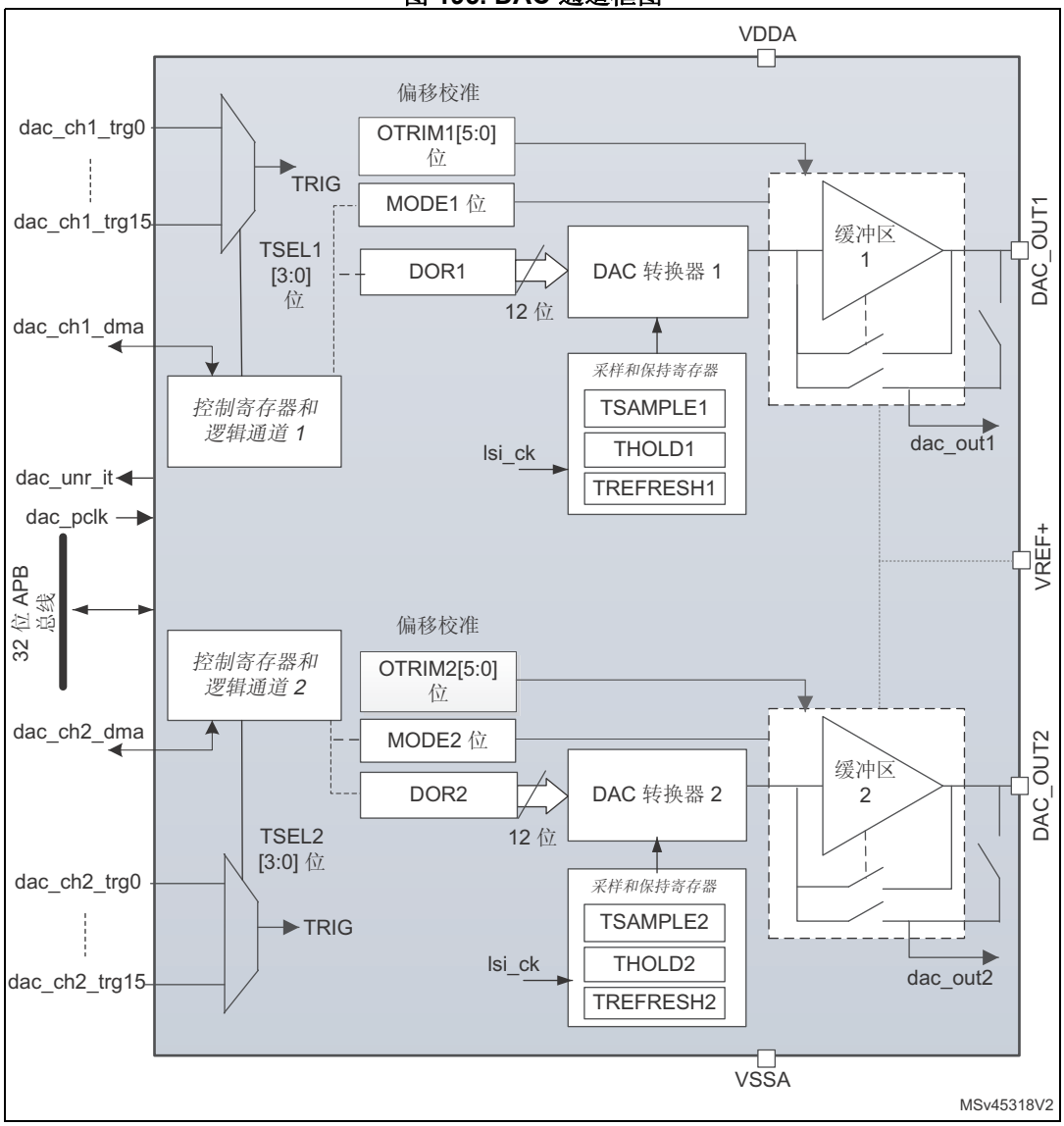
- 两个 DAC 接口，各自最多对应两个输出通道
- 12 位模式下数据采用左对齐或右对齐
- 同步更新功能
- 噪声波和三角波生成
- DAC 双通道单独或同时转换
- 每条通道的 DMA 功能，包括 DMA 下溢错误检测
- 通过外部触发信号进行转换
- DAC 输出通道缓冲/非缓冲模式
- 缓冲器偏移校准
- 每路 DAC 输出均可与 DAC_OUTx 输出引脚断开连接
- DAC 输出可与片上外设连接
- 可在停止模式下通过采样和保持模式实现低功耗运行
- 来自 V_{REF+} 引脚的输入参考电压或内部 VREFBUF 参考电压

图 195 所示为 DAC 通道的框图，*表 206* 则给出了引脚说明。

26.3 DAC 功能说明

26.3.1 DAC框图

图 195. DAC 通道框图



1. 输出模式控制器在正常模式（采用缓冲/非缓冲配置）与采样和保持模式之间切换。

26.3.2 DAC 引脚和内部信号

DAC 包含：

- 多达两条输出通道
- DAC_OUTx 可与输出引脚断开连接并用作普通 GPIO
- dac_outx 可使用与片上外设（如比较器和 OPAMP）的内部引脚连接。
- DAC 输出通道（缓冲/非缓冲）
- 使用 LSI 时钟源在停止模式下运行以实现静态转换的采样和保持模块及其寄存器

DAC 包含多达两条独立的输出通道。每条输出通道均可连接到片上外设，如 COMP、OPAMP 和 ADC。在这种情况下，DAC 输出通道可与 DAC_OUTx 输出引脚断开连接，相应的 GPIO 可用于其他用途。

DAC 输出可缓冲、也可以不缓冲。采样和保持模块及其关联寄存器可在停止模式下使用 LSI 时钟源运行。

表 206. DAC 输入/输出引脚

引脚名称	信号类型	注释
V_{REF+}	正模拟参考电压输入	DAC 高/正参考电压，$V_{REF+} \leq V_{DDAmax}$（请参见数据手册）
VDDA	模拟电源输入	模拟电源
VSSA	模拟电源地输入	模拟电源地
DAC_OUTx	模拟输出信号	DAC 通道 x 模拟输出

表 207. DAC 内部输入/输出信号

内部信号名称	信号类型	说明
dac_ch1_dma	双向	DACx 通道 1 DMA 请求
dac_ch2_dma	双向	DACx 通道 2 DMA 请求
dac_ch1_trg[0:15]	输入	DACx 通道 1
dac_ch2_trg[0:15]	输入	DACx 通道 2
dac_unr_it	输出	DACx 下溢中断
dac_pclk	输入	DACx 外设时钟
dac_out1	模拟输出	片上外设的 DACx 通道 1 输出
dac_out2	模拟输出	片上外设的 DACx 通道 2 输出

26.3.3 DAC 通道使能

将 DACx_CR 寄存器中的相应 ENx 位置 1，即可使能对应 DAC 通道。经过一段启动时间 t_{WAKEUP} 后，DAC 通道被真正使能。

注： *ENx 位只会使能模拟 DAC 通道 x。即使 ENx 位复位，DAC 通道 x 数字接口仍处于使能状态。*

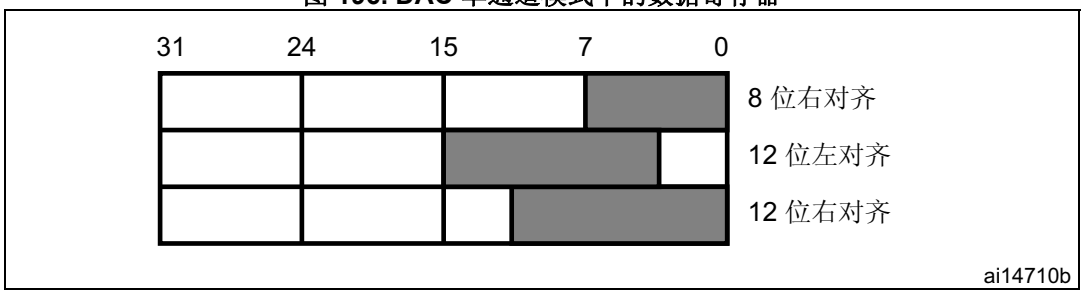
26.3.4 DAC 数据格式

根据所选配置模式，数据必须按如下方式写入指定寄存器：

- 对于 DAC 单通道 x，有三种可能的方式：
  - 8 位右对齐：软件必须将数据加载到 DACx_DHR8Ry [7:0] 位（存储到 DHRy[11:4] 位）
  - 12 位左对齐：软件必须将数据加载到 DACx_DHR12Ly [15:4] 位（存储到 DHRy[11:0] 位）
  - 12 位右对齐：软件必须将数据加载到 DACx_DHR12Ry [11:0] 位（存储到 DHRx[11:0] 位）

根据加载的 DACx_DHRyyyw 寄存器，用户写入的数据将移位并存储到相应的 DHRx（数据保持寄存器 x，即内部非存储器映射寄存器）。之后，DHRx 寄存器将被自动加载，或者通过软件或外部事件触发加载到 DORx 寄存器。

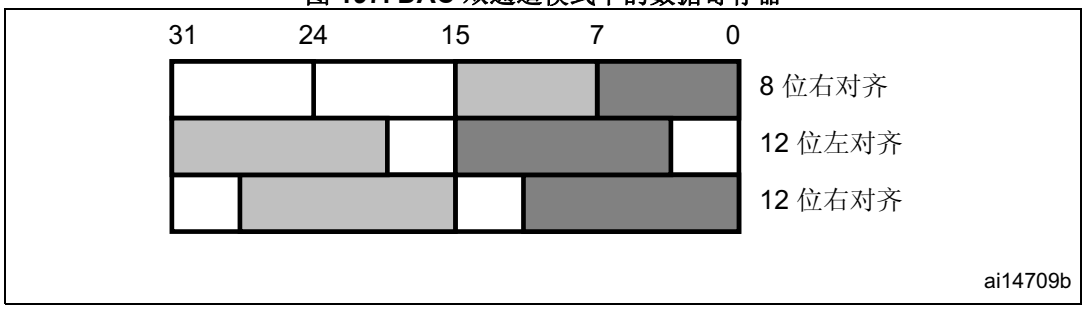
图 196. DAC 单通道模式下的数据寄存器



- 对于 DAC 双通道（可用时），有三种可能的方式：
  - 8 位右对齐：将 DAC 通道 1 的数据加载到 DACx_DHR8RD [7:0] 位（存储到 DHR1[11:4] 位），将 DAC 通道 2 的数据加载到 DACx_DHR8RD [15:8] 位（存储到 DHR2[11:4] 位）
  - 12 位左对齐：将 DAC 通道 1 的数据加载到 DACx_DHR12LD [15:4] 位（存储到 DHR1[11:0] 位），将 DAC 通道 2 的数据加载到 DACx_DHR12LD [31:20] 位（存储到 DHR2[11:0] 位）
  - 12 位右对齐：将 DAC 通道 1 的数据加载到 DACx_DHR12RD [11:0] 位（存储到 DHR1[11:0] 位），将 DAC 通道 2 的数据加载到 DACx_DHR12RD [27:16] 位（存储到 DHR2[11:0] 位）

根据加载的 DACx_DHRyyyD 寄存器，用户写入的数据将移位并存储到 DHR1 和 DHR2（数据保持寄存器，即内部非存储器映射寄存器）。之后，DHR1 和 DHR2 寄存器将被自动加载，或者通过软件或外部事件触发分别被加载到 DOR1 和 DOR2 寄存器。

图 197. DAC 双通道模式下的数据寄存器



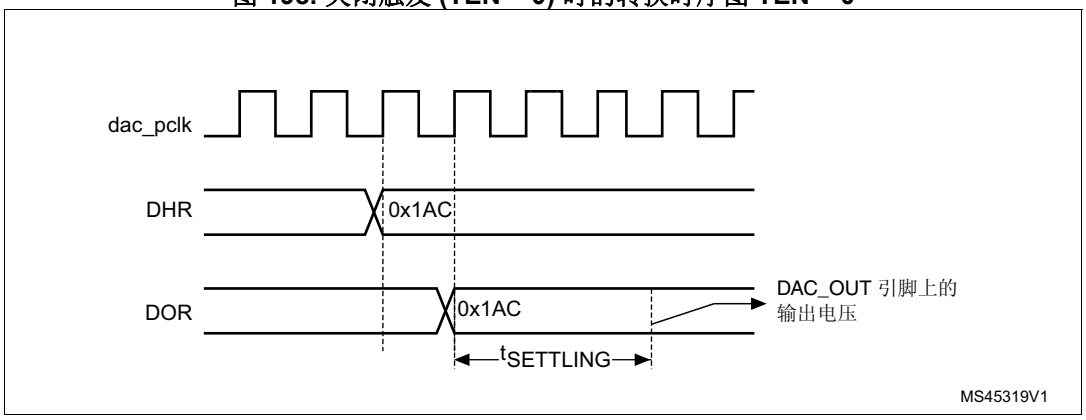
26.3.5 DAC 转换

DACx_DORy 无法直接写入，任何数据都必须通过加载 DACx_DHRy 寄存器（写入 DACx_DHR8Ry、DACx_DHR12Ly、DACx_DHR12Ry、DACx_DHR8RD、DACx_DHR12RD 或 DACx_DHR12LD）才能传输到 DAC 通道 x。

如果未选择硬件触发（DACx_CR 寄存器中的 TENx 位复位），那么经过一个 dac_pclk 时钟周期后，DACx_DHRy 寄存器中存储的数据将自动转移到 DACx_DORy 寄存器。但是，如果选择硬件触发（置位 DACx_CR 寄存器中的 TENx 位）且触发条件到来，将在三个 dac_pclk 时钟周期后进行转移。

当 DACx_DORy 加载了 DACx_DHRy 内容时，模拟输出电压将在一段时间 $t_{SETTLING}$ 后可用，具体时间取决于电源电压和模拟输出负载。

图 198. 关闭触发 (TEN = 0) 时的转换时序图 TEN = 0



26.3.6 DAC 输出电压

经过线性转换后，数字输入会转换为 0 到 V_{REF+} 之间的输出电压。

各 DAC 通道引脚的模拟输出电压通过以下公式确定：

$$DACoutput = V_{REF} \times \frac{DOR}{4096}$$

26.3.7 DAC 触发选择

如果 TENx 控制位置 1，可通过外部事件（定时计数器、外部中断线）触发转换。TSELx[3:0] 控制位将决定通过 16 个可能事件中的哪一个来触发转换，如*第 26.6.1 节：DAC x 控制寄存器 (DACx_CR) （x=1 到 2）*中的 TSEL1[3:0] 和 TSEL2[3:0] 位所示。

每当 DAC 接口在所选触发源上检测到上升沿时（请参见下表），DACx_DHRy 寄存器中存储的最后一个数据即会传输到 DACx_DORy 寄存器中。发生触发后再经过三个 dac_pclk 周期，DACx_DORy 寄存器将会得到更新。

如果选择软件触发，一旦 SWTRIG 位置 1，转换即会开始。DACx_DHRy 寄存器内容加载到 DACx_DORy 寄存器中后，SWTRIG 即由硬件复位。

注： *ENx 位置 1 时，无法更改 TSELx[3:0] 位。*

如果选择软件触发，DACx_DHRy 寄存器的内容只需一个 dac_pclk 时钟周期即可转移到 DACx_DORy 寄存器。

表 208. DAC 触发选择

源	类型	TSELx[3:0]
SWTRIG	软件控制位	0000
TIM1_TRGO	片上定时器的内部信号	0001
TIM2_TRGO	片上定时器的内部信号	0010
TIM4_TRGO	片上定时器的内部信号	0011
TIM5_TRGO	片上定时器的内部信号	0100
TIM6_TRGO	片上定时器的内部信号	0101
TIM7_TRGO	片上定时器的内部信号	0110
TIM8_TRGO	片上定时器的内部信号	0111
TIM15_TRGO	片上定时器的内部信号	1000
HRTIM1_DACTRG1	片上定时器的内部信号	1001
HRTIM1_DACTRG2	片上定时器的内部信号	1010
LPTIM1_OUT	片上定时器的内部信号	1011
LPTIM2_OUT	片上定时器的内部信号	1100
EXTI9	外部引脚	1101
保留	-	1110
保留	-	1111

26.3.8 DMA 请求

每个 DAC 通道都具有 DMA 功能。两个 DMA 通道用于处理 DAC 通道的 DMA 请求。

当 DMAENx 位置 1 时，如果发生外部触发（而不是软件触发），则传输结束后 DACx_DHRy 寄存器的值会转移到 DACx_DORy 寄存器，并且会产生 DMA 请求。

在双通道模式下，如果两个 DMAENx 位均置 1，则将产生两个 DMA 请求。如果只需要一个 DMA 请求，应仅将相应 DMAENx 位置 1。这样，应用程序可以在双通道模式下通过一个 DMA 请求和一个特定 DMA 通道来管理两个 DAC 通道。

由于 DACx_DHRy 已在 DMA 请求之前向 DACx_DORy 传输数据，必须在发生第一个触发事件之前将第一个数据写入 DACx_DHRy。

DMA 下溢

DAC DMA 请求没有缓冲队列。这样，如果第二个外部触发到达时尚未收到第一个外部触发的应答，将不会发出新的请求，并且 DACx_SR 寄存器中的 DAM 通道下溢标志 DMAUDRx 将置 1，以报告这一错误状况。DAC 通道仍将继续转换旧有数据。

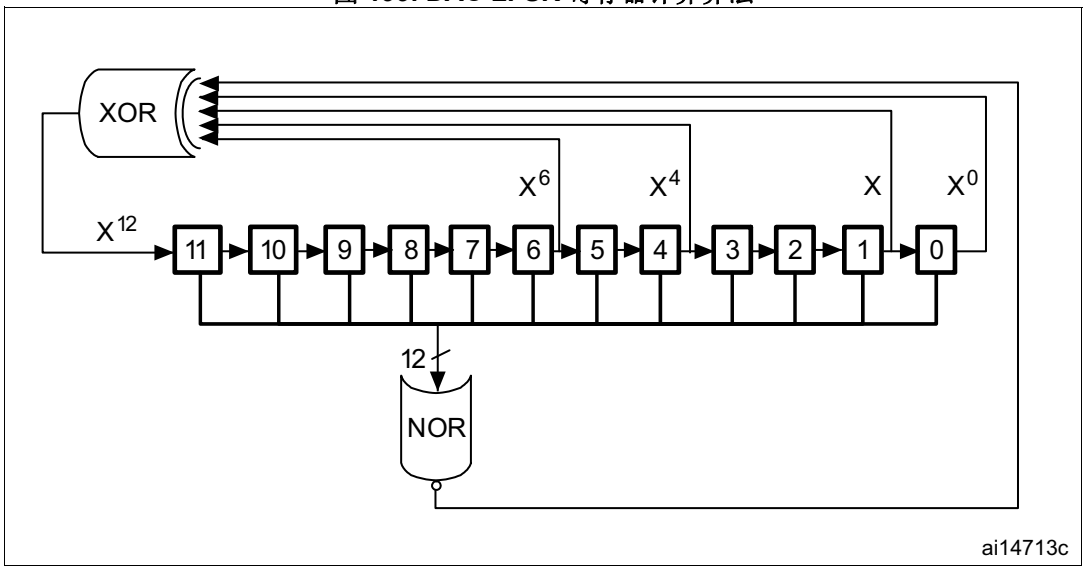
软件应通过写入 1 来将 DMAUDRx 标志清零，将所用 DMA 数据流的 DMAEN 位清零，并重新初始化 DMA 和 DAC 通道，以便正确地重新开始 DMA 传输。软件应修改 DAC 触发转换频率或减轻 DMA 工作负载，以避免再次发生 DMA 下溢。最后，可通过使能 DMA 数据传输和转换触发来继续完成 DAC 转换。

对于各 DAC 通道，如果使能 DACx_CR 寄存器中相应的 DMAUDRIEx 位，还将产生中断。

26.3.9 生成噪声

为了生成可变振幅的伪噪声，可使用 LFSR（线性反馈移位寄存器）。将 WAVEx[1:0] 置为“01”即可选择生成噪声。LFSR 中的预加载值为 0xAAA。在每次发生触发事件后，经过三个 dac_pclk 时钟周期，该寄存器会依照特定的计算算法完成更新。

图 199. DAC LFSR 寄存器计算算法

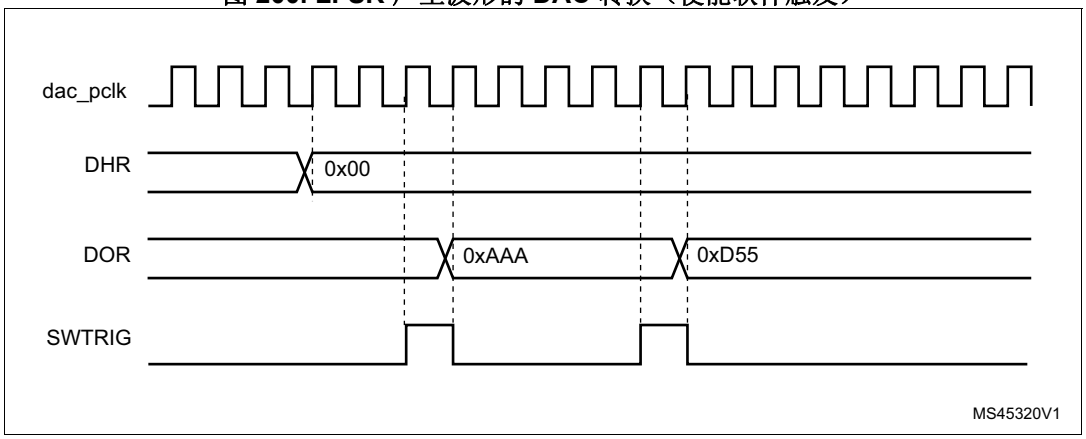


LFSR 值可以通过 DACx_CR 寄存器中的 MAMPx[3:0] 位来部分或完全屏蔽，在不发生溢出的情况下，该值将与 DACx_DHRy 的内容相加，然后转移到 DACx_DORy 寄存器中。

如果 LFSR 为 0x0000，将向其注入“1”（防锁定机制）。

可以通过复位 WAVEx[1:0] 位来将 LFSR 波形产生功能关闭。

图 200. LFSR 产生波形的 DAC 转换（使能软件触发）



注： *要生成三角波，必须通过将 DACx_CR 寄存器中的 TENx 位置 1 来使能 DAC 触发。*

26.3.10 生成三角波

可以在直流电流或慢变信号上叠加一个小幅三角波。将 WAVEx[1:0] 置为“10”即可选择 DAC 生成三角波。振幅通过 DACx_CR 寄存器中的 MAMPx[3:0] 位进行配置。每次发生触发事件后，经过三个 dac_pclk 时钟周期，内部三角波计数器将会递增。在不发生溢出的情况下，该计数器的值将与 DACx_DHRy 寄存器内容相加，所得总和将传输到 DACx_DORy 寄存器中。只要小于 MAMPx[3:0] 位定义的最大振幅，三角波计数器就会一直递增。一旦达到配置的振幅，计数器将递减至零，然后再递增，以此类推。

可以通过复位 WAVEx[1:0] 位来将三角波产生功能关闭。

图 201. 生成 DAC 三角波

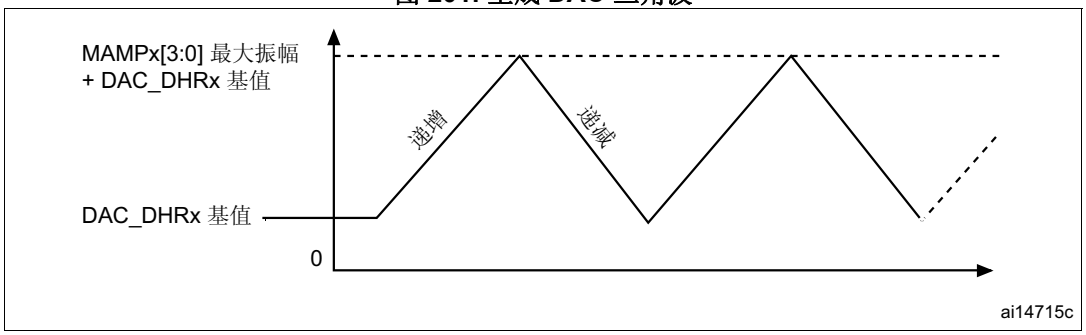
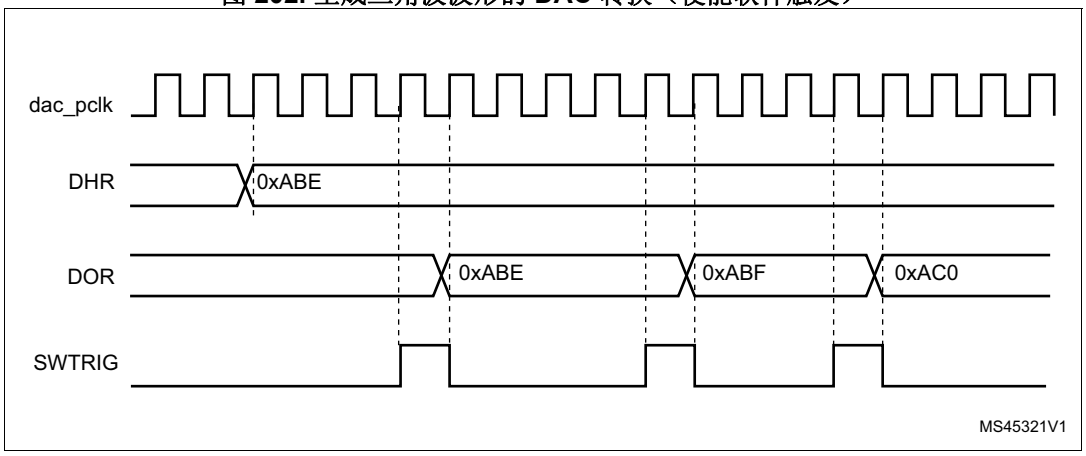


图 202. 生成三角波波形的 DAC 转换（使能软件触发）



注： *要生成三角波，必须通过将 DACx_CR 寄存器中的 TENx 位置 1 来使能 DAC 触发。MAMPx[3:0] 位必须在使能 DAC 之前进行配置，否则将无法更改。*

26.3.11 DAC 通道模式

每条 DAC 通道均可配置为正常模式或采样和保持模式。可使能输出缓冲器，以实现高驱动能力。使能输出缓冲器之前，需要校准电压偏移。此项校准是在出厂时执行的（复位后加载），可在应用运行期间通过软件进行调整。

正常模式

在正常模式下，可通过更改缓冲器状态以及更改 DAC_OUTx 引脚互连实现四种组合。

要使能输出缓冲器，DACx_MCR 寄存器中的 MODEx[2:0] 位应满足：

- 000：DAC 连接到外部引脚
- 001：DAC 连接到外部引脚以及片上外设

要禁止输出缓冲器，DACx_MCR 寄存器中的 MODEx[2:0] 位应满足：

- 010：DAC 连接到外部引脚
- 011：DAC 连接到片上外设

采样和保持模式

在采样和保持模式下，DAC 内核在触发转换后对数据进行转换，然后使电容上保持转换后的电压。不进行转换时，DAC 内核和缓冲器在两次采样之间是完全关闭的，DAC 输出处于三态，因此可降低整体功耗。每次新转换之前都需要一个新的稳定周期（$T_{stab-BON}$ 或 $T_{stab-BOFF}$，具体视缓冲器状态而定）。

在该模式下，DAC 内核以及所有相应的逻辑和寄存器除了受 dac_pclk 时钟驱动外，还受低速时钟 (LSI) 驱动，从而可在深度低功耗模式（如停止模式）下使用 DAC 通道。

采样/保持模式运行可分为 3 个阶段：

1. 采样阶段：采样/保持元件充电到所需电压。充电时间取决于电容值（内部还是外部，由用户选择）。采样时间是通过 DACx_SHSRy 寄存器中的 TSAMx[9:0] 位配置的。对 TSAMx[9:0] 位执行写操作期间，DACx_SR 寄存器中的 BWSTx 位会置 1，从而可在两个时钟域（APB 和低速时钟）之间实现同步，并允许软件在 DAC 通道运行期间更改采样阶段的值。
2. 保持阶段：DAC 输出通道处于三态，DAC 内核和缓冲器关闭，以降低电流消耗。保持时间是通过 DACx_SHHR 寄存器中的 THOLDx[9:0] 位配置的。
3. 刷新阶段：刷新时间是通过 DACx_SHRR 寄存器中的 TREFx[7:0] 位配置的。

以上三个阶段的时间以 LSI 时钟为单位。举例来说，假设选择了 LSI ~32 KHz，要配置 350 µs 的采样时间，2 ms 的保持时间和 100 µs 的刷新时间：

采样阶段需要 12 个周期：SAMx[9:0] = 11，

保持阶段需要 62 个周期：THOLDx[9:0] = 62，

刷新阶段需要 4 个周期：TREFx[7:0] = 4。

本例中，与正常模式相比，功耗几乎降低了 15 倍。

下表列出了计算正确的采样时间和刷新时间的公式，保持时间取决于漏电流。

表 209. 采样时间和刷新时间

缓冲器状态	$t_{sampling\ (1)(3)}$	$t_{refresh\ (2)(3)}$
启用	$T_{stab-BON} + (10*R_{BON}*C_{load})$	$T_{stab-BON} + (R_{BON}*C_{load})*ln(2*Nlsb)$
禁用	$T_{stab-BOFF} + (10*R_{BOFF}*Cload)$	$T_{stab-BOFF} + (R_{BOFF}*Cload)*ln(2*Nlsb)$

注： *上述公式中，要实现 12 位分辨率，稳定为理想代码值（½ LSB 或精度）需要 10 个时间常数。要实现 8 位分辨率，稳定时间为 7 个时间常数。*

*保持阶段的容许压降“Vd”通过电容以输出漏电流放电后的 LSB 数表示。稳定回理想值（½ LSB 误差精度）需要 ln(2*Nlsb) 个 DAC 时间常数。*

参数 $T_{stab-BON}$、$T_{stab-BOFF}$、R_{BON} 和 R_{BOFF} 在数据手册中作出了说明。

输出缓冲器开启时的采样和刷新时间计算示例

注： *下例中使用的数值仅用于说明目的。有关产品数据，请参见产品数据手册。*

C_{load} = 100 nF

V_{DDA} = 3.0 V

采样阶段：

$t_{sampling}$ = 7 µs + (10 * 2000 * 100 * 10^{-9}) = 2.007 ms

（其中，$T_{stab-BON}$ = 7 µs，RBON = 2 kΩ）

刷新阶段：

$t_{refresh}$ = 7 µs + (2000 * 100 * 10^{-9}) * ln(2*10) = 606.1 µs

（其中，N_{lsb} = 10，保持阶段会减少 10 个 LSB）

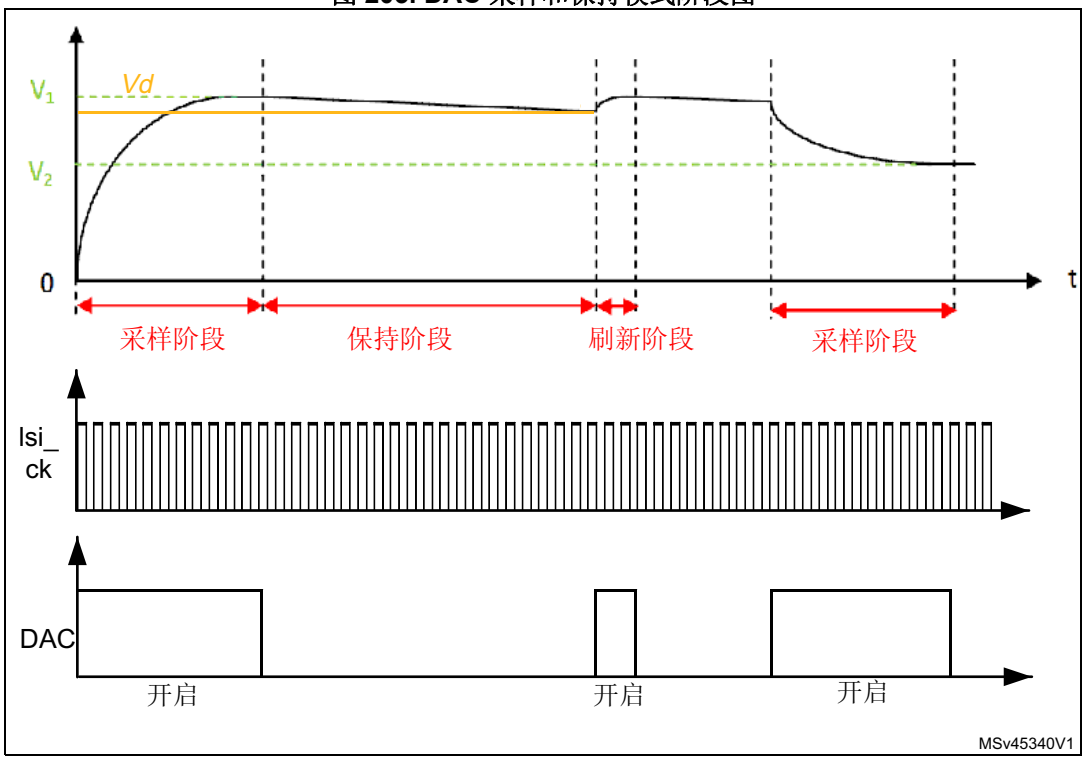
保持阶段：

$D_v = i_{leak} * t_{hold} / C_{load}$ = 0.0073 V（3 V 时 10 个 12 位 LSB）

i_{leak} = 150 nA（所有温度范围内 IO 泄漏的最坏情况）

t_{hold} = 0.0073 * 100 * 10^{-9} / (150 * 10^{-9}) = 4.867 ms

图 203. DAC 采样和保持模式阶段图



与正常模式相似，采样和保持模式也具有不同配置。

要使能输出缓冲器，DACx_MCR 寄存器中的 MODEx[2:0] 位应满足：

- 100：DAC 连接到外部引脚
- 101：DAC 连接到外部引脚以及片上外设

要禁止输出缓冲器，DACx_MCR 寄存器中的 MODEx[2:0] 位应满足：

- 110：DAC 连接到外部引脚以及片上外设
- 111：DAC 连接到片上外设

DACx_MCR 寄存器中的 MODEx[2:0] 位等于 111 时。内部电容“$C_{loadint}$”将保持 DAC 内核的电压输出，随后会将该电压驱动到片上外设。

所有采样和保持阶段均可中断，DACx_DHRy 中的任何更改都会立即触发新的采样阶段。

表 210. 通道输出模式汇总

MODEx[2:0]			模式	缓冲器：	输出连接
0	0	0	正常模式	启用	连接到外部引脚
0	0	1			连接到外部引脚以及片上外设（例如，比较器）
0	1	0		输出关闭	连接到外部引脚
0	1	1			连接到片上外设（例如，比较器）
1	0	0	采样和保持模式	启用	连接到外部引脚
1	0	1			连接到外部引脚以及片上外设（例如，比较器）
1	1	0		输出关闭	连接到外部引脚以及片上外设（例如，比较器）
1	1	1			连接到片上外设（例如，比较器）

26.3.12 DAC 通道缓冲器校准

N 位数模转换器 (DAC) 的传递函数为：

$$V_{out} = ((D/2^{N-1}) \times G \times V_{ref}) + V_{OS}$$

其中，V_{OUT} 为模拟输出，D 为数字输入，G 为增益，V_{ref} 为标称满量程电压，V_{os} 为偏移电压。对于理想的 DAC 通道，G = 1 且 V_{os} = 0。

由于输出缓冲器所具有的特性，各部分的电压偏移可能有所不同，并会在模拟输出上引入绝对偏移误差。为补偿 V_{os}，需要通过调整技术进行校准。

仅当 DAC 通道 x 在缓冲器使能的条件下运行时，校准才有效（MODEx[2:0] = 000b、001b、100b 或 101b）。如果在缓冲器关闭的条件下应用于其他模式，校准无效。校准过程中：

- 缓冲器输出将与引脚内部/外部连接断开，并会进入三态模式 (HiZ)。
- 缓冲器将作为比较器来检测中间代码值 0x800，并通过内部桥将其与 VREF+/2 信号进行比较，然后根据比较结果（CAL_FLAGx 位）将其输出信号切换为 0 或 1

提供以下两种校准技术：

- 出厂调整（始终使能）

  DAC 缓冲器偏移是在出厂时进行调整的。DACx_CCR 寄存器中 OTRIMx[4:0] 位的默认值是出厂调整值，会在 DAC 数字接口复位时载入。
- 用户调整

  如果运行条件不同于标称出厂调整条件，特别是 V_{DD}/V_{DDA} 电压、温度、VREF+ 值发生变化时，可进行用户调整，并且用户可在应用过程中随时通过软件进行调整。

注： *更多关于标称出厂调整条件的详细信息，请参见数据手册。*

另外，如果 V_{DD}/V_{DDA} 移除（例如器件进入 STANDBY 或 VBAT 模式），则需要进行校准。

执行用户调整校准的步骤如下：

1. 如果 DAC 通道激活，可向 DACx_CR 中的 ENx 位写入 0 来禁止通道。
2. 写入 DACx_MCR 寄存器，选择使能了缓冲器的模式（MODEx[2:0] = 000b、001b、100b 或 101b），
3. 将 DACx_CR 寄存器中的 CENx 位置 1，开始进行 DAC 通道 x 校准，
4. 应用调整算法：
   a) 向 OTRIMx[4:0] 位写入以 00000b 开头的代码
   b) 等待 $t_{OFFTRIMmax}$ 延时
   c) 检查 DACx_SR 中的 CAL_FLAGx 位是否已置 1
   d) 如果 CAL_FLAGx 已置 1，调整代码 OTRIMx[4:0] 已找到，并将在运行过程中用来补偿输出值，否则会使 OTRIMx[4:0] 递增并重复执行 (a) 到 (d) 的步骤。

软件算法可使用逐次逼近法或二分法较快地计算和设置 OTRIMx[4:0] 位的内容，

CAL_FLAGx 位的换向/切换指示偏移已正确补偿，相应的调整代码必须保留在 DACx_CCR 寄存器的 OTRIMx[4:0] 位中。

注： *为了获得正确的值，对 DACx_SR 寄存器中的 OTRIMx[4:0] 位进行的写操作与对 CAL_FLAGx 位进行的读操作之间必须保留 $t_{OFFTRIMmax}$ 延时。此参数在数据手册的电气特性部分进行了说明。*

如果 V_{DD}/V_{DDA}、VREF+ 和温度条件在器件运行期间不发生更改，同时器件更为频繁地进入待机和 VBAT 模式，软件可将第一次用户校准时找到的 OTRIMx[4:0] 位存储在 flash 或备用寄存器中。然后在器件电源再次恢复时直接加载/写入这些位，从而无需等待新的校准时间。

当 CENx 位置 1 时，不允许将 ENx 位置 1。

26.3.13 DAC 双通道转换（如果可用）

为了在同时需要两个 DAC 通道的应用中有效利用总线带宽，DAC 模块有三个双寄存器可供操作：DHR8RD、DHR12RD 和 DHR12LD。这样，只需一个寄存器访问即可同时驱动两个 DAC 通道。要生成相应波形，无需访问 DHRxxxD 寄存器。因此，可单独也可同时使用两个输出通道。

通过两个 DAC 通道和这三个双寄存器可以实现 11 种转换模式。但如果需要，所有这些转换模式也都可以通过单独的 DHRx 寄存器来实现。

下面几段内容将介绍所有这些模式。

独立触发（不产生波形）

要将 DAC 配置为此转换模式，需要遵循以下顺序：

1. 将两个 DAC 通道触发使能位 TEN1 和 TEN2 置 1。
2. 将 TSEL1[3:0] 和 TSEL2[3:0] 设置为不同的值，以配置不同的触发源。
3. 将 DAC 双通道数据加载到所需 DHR 寄存器（DACx_DHR12RD、DACx_DHR12LD 或 DACx_DHR8RD）。

DAC 通道 1 触发信号到达时，DHR1 寄存器的内容转移到 DACx_DOR1（三个 dac_pclk 时钟周期之后）。

DAC 通道 2 触发信号到达时，DHR2 寄存器的内容转移到 DACx_DOR2（三个 dac_pclk 时钟周期之后）。

独立触发（生成单个 LFSR）

要将 DAC 配置为此转换模式，需要遵循以下顺序：

1. 将两个 DAC 通道触发使能位 TEN1 和 TEN2 置 1。
2. 将 TSEL1[3:0] 和 TSEL2[3:0] 设置为不同的值，以配置不同的触发源。
3. 将两个 DAC 通道的 WAVEx[1:0] 设置为 01，并在 MAMPx[3:0] 位中配置相同的 LFSR 掩码值。
4. 将 DAC 双通道数据加载到所需 DHR 寄存器（DHR12RD、DHR12LD 或 DHR8RD）。

DAC 通道 1 触发信号到达时，LFSR1 计数器内容（使用相同的掩码）与 DHR1 寄存器内容相加，所得总和转移到 DACx_DOR1 中（三个 dac_pclk 时钟周期之后）。LFSR1 计数器随即更新。

DAC 通道 2 触发信号到达时，LFSR2 计数器内容（使用相同的掩码）与 DHR2 寄存器内容相加，所得总和转移到 DACx_DOR2 中（三个 dac_pclk 时钟周期之后）。LFSR2 计数器随即更新。

独立触发（生成不同 LFSR）

要将 DAC 配置为此转换模式，需要遵循以下顺序：

1. 将两个 DAC 通道触发使能位 TEN1 和 TEN2 置 1。
2. 将 TSEL1[3:0] 和 TSEL2[3:0] 设置为不同的值，以配置不同的触发源。
3. 将两个 DAC 通道的 WAVEx[1:0] 设置为 01，并在 MAMP1[3:0] 和 MAMP2[3:0] 位中设置不同的 LFSR 掩码值。
4. 将 DAC 双通道数据加载到所需 DHR 寄存器（DACx_DHR12RD、DACx_DHR12LD 或 DACx_DHR8RD）。

DAC 通道 1 触发信号到达时，LFSR1 计数器内容（使用 MAMP1[3:0] 配置的掩码）与 DHR1 寄存器内容相加，所得总和转移到 DACx_DOR1 中（三个 dac_pclk 时钟周期之后）。LFSR1 计数器随即更新。

DAC 通道 2 触发信号到达时，LFSR2 计数器内容（使用 MAMP2[3:0] 配置的掩码）与 DHR2 寄存器内容相加，所得总和转移到 DACx_DOR2 中（三个 dac_pclk 时钟周期之后）。LFSR2 计数器随即更新。

独立触发（生成单个三角波）

要将 DAC 配置为此转换模式，需要遵循以下顺序：

1. 将两个 DAC 通道触发使能位 TEN1 和 TEN2 置 1。
2. 将 TSEL1[3:0] 和 TSEL2[3:0] 设置为不同的值，以配置不同的触发源。
3. 将两个 DAC 通道的 WAVEx[1:0] 设置为 1x，并在 MAMPx[3:0] 位中配置相同的最大振幅值。
4. 将 DAC 双通道数据加载到所需 DHR 寄存器（DACx_DHR12RD、DACx_DHR12LD 或 DACx_DHR8RD）。

DAC 通道 1 触发信号到达时，DAC 通道 1 三角波计数器内容（使用相同的三角波振幅）与 DHR1 寄存器内容相加，所得总和转移到 DACx_DOR1 中（三个 dac_pclk 时钟周期之后）。DAC 通道 1 三角波计数器随即更新。

DAC 通道 2 触发信号到达时，DAC 通道 2 三角波计数器内容（使用相同的三角波振幅）与 DHR2 寄存器内容相加，所得总和转移到 DACx_DOR2 中（三个 dac_pclk 时钟周期之后）。DAC 通道 2 三角波计数器随即更新。

独立触发（生成不同三角波）

要将 DAC 配置为此转换模式，需要遵循以下顺序：

1. 将两个 DAC 通道触发使能位 TEN1 和 TEN2 置 1。
2. 将 TSEL1[3:0] 和 TSEL2[3:0] 设置为不同的值，以配置不同的触发源。
3. 将两个 DAC 通道的 WAVEx[1:0] 设置为 1x，并在 MAMP1[3:0] 和 MAMP2[3:0] 位中设置不同的最大振幅值。
4. 将 DAC 双通道数据加载到所需 DHR 寄存器（DACx_DHR12RD、DACx_DHR12LD 或 DACx_DHR8RD）。

DAC 通道 1 触发信号到达时，DAC 通道 1 三角波计数器内容（使用 MAMP1[3:0] 配置的三角波振幅）与 DHR1 寄存器内容相加，所得总和转移到 DACx_DOR1 中（三个 dac_pclk 时钟周期之后）。DAC 通道 1 三角波计数器随即更新。

DAC 通道 2 触发信号到达时，DAC 通道 2 三角波计数器内容（使用 MAMP2[3:0] 配置的三角波振幅）与 DHR2 寄存器内容相加，所得总和转移到 DACx_DOR2 中（三个 dac_pclk 时钟周期之后）。DAC 通道 2 三角波计数器随即更新。

同步软件启动

要将 DAC 配置为此转换模式，需要遵循以下顺序：

- 将 DAC 双通道数据加载到所需 DHR 寄存器（DACx_DHR12RD、DACx_DHR12LD 或 DACx_DHR8RD）

在此配置中，DHR1 和 DHR2 寄存器内容会在一个 dac_pclk 时钟周期后分别转移到 DACx_DOR1 和 DACx_DOR2 中。

同步触发（不产生波形）

要将 DAC 配置为此转换模式，需要遵循以下顺序：

- 将两个 DAC 通道触发使能位 TEN1 和 TEN2 置 1
- 将 TSEL1[3:0] 和 TSEL2[3:0] 设置为相同的值，以便为两个 DAC 通道配置相同的触发源
- 将 DAC 双通道数据加载到所需 DHR 寄存器（DACx_DHR12RD、DACx_DHR12LD 或 DACx_DHR8RD）

当触发信号到达时，DHR1 和 DHR2 寄存器内容将分别转移到 DACx_DOR1 和 DACx_DOR2 中（三个 dac_pclk 时钟周期之后）。

同步触发（生成单个 LFSR）

要将 DAC 配置为此转换模式，需要遵循以下顺序：

- 将两个 DAC 通道触发使能位 TEN1 和 TEN2 置 1
- 将 TSEL1[3:0] 和 TSEL2[3:0] 设置为相同的值，以便为两个 DAC 通道配置相同的触发源
- 将两个 DAC 通道的 WAVEx[1:0] 设置为 01，并在 MAMPx[3:0] 位中配置相同的 LFSR 掩码值
- 将 DAC 双通道数据加载到所需 DHR 寄存器（DHR12RD、DHR12LD 或 DHR8RD）

触发信号到达时，LFSR1 计数器内容（使用相同的掩码）与 DHR1 寄存器内容相加，所得总和转移到 DACx_DOR1 中（三个 dac_pclk 时钟周期之后）。LFSR1 计数器随即更新。同时，LFSR2 计数器内容（使用相同的掩码）与 DHR2 寄存器内容相加，所得总和转移到 DACx_DOR2 中（三个 dac_pclk 时钟周期之后）。LFSR2 计数器随即更新。

同步触发（生成不同 LFSR）

要将 DAC 配置为此转换模式，需要遵循以下顺序：

- 将两个 DAC 通道触发使能位 TEN1 和 TEN2 置 1
- 将 TSEL1[3:0] 和 TSEL2[3:0] 设置为相同的值，以便为两个 DAC 通道配置相同的触发源
- 将两个 DAC 通道的 WAVEx[1:0] 设置为 01，并在 MAMP1[3:0] 和 MAMP2[3:0] 位中设置不同的 LFSR 掩码值
- 将 DAC 双通道数据加载到所需 DHR 寄存器（DACx_DHR12RD、DACx_DHR12LD 或 DACx_DHR8RD）

触发信号到达时，LFSR1 计数器内容（使用 MAMP1[3:0] 配置的掩码）与 DHR1 寄存器内容相加，所得总和转移到 DACx_DOR1 中（三个 dac_pclk 时钟周期之后）。LFSR1 计数器随即更新。
同时，LFSR2 计数器内容（使用 MAMP2[3:0] 配置的掩码）与 DHR2 寄存器内容相加，所得总和转移到 DACx_DOR2 中（三个 dac_pclk 时钟周期之后）。LFSR2 计数器随即更新。

同步触发（生成单个三角波）

要将 DAC 配置为此转换模式，需要遵循以下顺序：

- 将两个 DAC 通道触发使能位 TEN1 和 TEN2 置 1
- 将 TSEL1[3:0] 和 TSEL2[3:0] 设置为相同的值，以便为两个 DAC 通道配置相同的触发源
- 将两个 DAC 通道的 WAVEx[1:0] 设置为 1x，并在 MAMPx[3:0] 位中配置相同的最大振幅值
- 将 DAC 双通道数据加载到所需 DHR 寄存器（DACx_DHR12RD、DACx_DHR12LD 或 DACx_DHR8RD）

触发信号到达时，DAC 通道 1 三角波计数器内容（使用相同的三角波振幅）与 DHR1 寄存器内容相加，所得总和转移到 DACx_DOR1 中（三个 dac_pclk 时钟周期之后）。DAC 通道 1 三角波计数器随即更新。

同时，DAC 通道 2 三角波计数器内容（使用相同的三角波振幅）与 DHR2 寄存器内容相加，所得总和转移到 DACx_DOR2 中（三个 dac_pclk 时钟周期之后）。DAC 通道 2 三角波计数器随即更新。

同步触发（生成不同三角波）

要将 DAC 配置为此转换模式，需要遵循以下顺序：

- 将两个 DAC 通道触发使能位 TEN1 和 TEN2 置 1
- 将 TSEL1[3:0] 和 TSEL2[3:0] 设置为相同的值，以便为两个 DAC 通道配置相同的触发源
- 将两个 DAC 通道的 WAVEx[1:0] 设置为 1x，并在 MAMP1[3:0] 和 MAMP2[3:0] 位中设置不同的最大振幅值
- 将 DAC 双通道数据加载到所需 DHR 寄存器（DACx_DHR12RD、DACx_DHR12LD 或 DACx_DHR8RD）

触发信号到达时，DAC 通道 1 三角波计数器内容（使用 MAMP1[3:0] 配置的三角波振幅）与 DHR1 寄存器内容相加，所得总和转移到 DACx_DOR1 中（三个 APB 时钟周期之后）。DAC 通道 1 三角波计数器随即更新。

同时，DAC 通道 2 三角波计数器内容（使用 MAMP2[3:0] 配置的三角波振幅）与 DHR2 寄存器内容相加，所得总和转移到 DACx_DOR2 中（三个 dac_pclk 时钟周期之后）。DAC 通道 2 三角波计数器随即更新。

26.4 DAC 低功耗模式

表 211. DAC 低功耗模式的作用

模式	说明
Sleep	不起作用，DAC 可与 DMA 结合使用
Stop	如果通过 lsi_ck 时钟选择了采样和保持模式，则 DAC 将以静态输出值继续工作
待机	DAC 外设掉电，退出待机模式后必须重新初始化。

26.5 DAC 中断

表 212. DAC 中断

中断事件	事件标志	使能控制位
DMA 下溢	DMAUDRx	DMAUDRIEx

26.6 DAC 寄存器

有关寄存器说明中使用的缩写，请参见*第 94 页的第 1 节*。

外设寄存器必须按字（32 位）进行访问。

26.6.1 DAC x 控制寄存器 (DACx_CR)（x=1 到 2）

DAC x control register

偏移地址：0x00

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	CEN2	DMAUDRIE2	DMAEN2	MAMP2[3:0]				WAVE2[1:0]		TSEL23	TSEL22	TSEL21	TSEL20	TEN2	EN2
	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	CEN1	DMAUDRIE1	DMAEN1	MAMP1[3:0]				WAVE1[1:0]		TSEL13	TSEL12	TSEL11	TSEL10	TEN1	EN1
	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31 保留，必须保持复位值。

位 30 **CEN2**：DAC 通道 2 校准使能 (DAC Channel 2 calibration enable)
此位由软件置 1 和清零，用于使能/禁止 DAC 通道 2 校准，仅当 DACx_CR 中的位 EN2=0 时才可写入（仅当 DAC 通道已禁止时才可以进入/退出校准模式），否则会忽略写操作。
0：DAC 通道 2 处于正常运行模式
1：DAC 通道 2 处于校准模式

位 29 **DMAUDRIE2**：DAC 通道 2 DMA 下溢中断使能 (DAC channel2 DMA underrun interrupt enable)
此位由软件置 1 和清零。
0：禁止 DAC 通道 2 DMA 下溢中断
1：使能 DAC 通道 2 DMA 下溢中断

位 28 **DMAEN2**：DAC 通道 2 DMA 使能 (DAC channel2 DMA enable)
此位由软件置 1 和清零。
0：禁止 DAC 通道 2 DMA 模式
1：使能 DAC 通道 2 DMA 模式

位 27:24 **MAMP2[3:0]**：DAC 通道 2 掩码/振幅选择器 (DAC channel2 mask/amplitude selector)

这些位由软件写入，用于在生成噪声波模式下选择掩码，或者在生成三角波模式下选择振幅。

0000：不屏蔽 LFSR 的位 0/三角波振幅等于 1

0001：不屏蔽 LFSR 的位 [1:0]/三角波振幅等于 3

0010：不屏蔽 LFSR 的位 [2:0]/三角波振幅等于 7

0011：不屏蔽 LFSR 的位 [3:0]/三角波振幅等于 15

0100：不屏蔽 LFSR 的位 [4:0]/三角波振幅等于 31

0101：不屏蔽 LFSR 的位 [5:0]/三角波振幅等于 63

0110：不屏蔽 LFSR 的位 [6:0]/三角波振幅等于 127

0111：不屏蔽 LFSR 的位 [7:0]/三角波振幅等于 255

1000：不屏蔽 LFSR 的位 [8:0]/三角波振幅等于 511

1001：不屏蔽 LFSR 的位 [9:0]/三角波振幅等于 1023

1010：不屏蔽 LFSR 的位 [10:0]/三角波振幅等于 2047

≥ 1011：不屏蔽 LFSR 的位 [11:0]/三角波振幅等于 4095

位 23:22 **WAVE2[1:0]**：DAC 通道 2 噪声/三角波生成使能 (DAC channel2 noise/triangle wave generation enable)

这些位由软件置 1 或清零。

00：禁止生成波

01：使能生成噪声波

1x：使能生成三角波

注： 只在位 TEN2 = 1（使能 DAC 通道 2 触发）时使用

位 21:18 **TSEL2[3:0]**：DAC 通道 2 触发器选择 (DAC channel2 trigger selection)

这些位用于选择 DAC 通道 2 的外部触发事件

有关触发配置和映射的详细信息，请参见触发选择表。

注： 只在位 TEN2 = 1（使能 DAC 通道 2 触发）时使用。

位 17 **TEN2**：DAC 通道 2 触发使能 (DAC channel2 trigger enable)

此位由软件置 1 和清零，以使能/禁止 DAC 通道 2 触发

0：禁止 DAC 通道 2 触发，写入 DACx_DHR2 寄存器的数据在一个 dac_pclk 时钟周期之后转移到 DACx_DOR2 寄存器

1：使能 DAC 通道 2 触发，DACx_DHR2 寄存器的数据在三个 dac_pclk 时钟周期之后转移到 DACx_DOR2 寄存器

注： 如果选择软件触发，DACx_DHR2 寄存器的内容只需一个 dac_pclk 时钟周期即可转移到 DACx_DOR2 寄存器。

位 16 **EN2**：DAC 通道 2 使能 (DAC channel2 enable)

此位由软件置 1 和清零，以使能/禁止 DAC 通道 2 。

0：禁止 DAC 通道 2

1：使能 DAC 通道 2

位 15 保留，必须保持复位值。

位 14 **CEN1**：DAC 通道 1 校准使能 (DAC Channel 1 calibration enable)

此位由软件置 1 和清零，用于使能/禁止 DAC 通道 1 校准，仅当 DACx_CR 中的位 EN1=0 时才可写入（仅当 DAC 通道已禁止时才可以进入/退出校准模式），否则会忽略写操作。

0：DAC 通道 1 处于正常运行模式

1：DAC 通道 1 处于校准模式

位 13 **DMAUDRIE1**：DAC 通道 1 DMA 下溢中断使能 (DAC channel1 DMA Underrun Interrupt enable)

此位由软件置 1 和清零。

0：禁止 DAC 通道 1 DMA 下溢中断

1：使能 DAC 通道 1 DMA 下溢中断

位 12 **DMAEN1**：DAC 通道 1 DMA 使能 (DAC channel1 DMA enable)

此位由软件置 1 和清零。

0：禁止 DAC 通道 1 DMA 模式

1：使能 DAC 通道 1 DMA 模式

位 11:8 **MAMP1[3:0]**：DAC 通道 1 掩码/振幅选择器 (DAC channel1 mask/amplitude selector)

这些位由软件写入，用于在生成噪声波模式下选择掩码，或者在生成三角波模式下选择振幅。

0000：不屏蔽 LFSR 的位 0/三角波振幅等于 1

0001：不屏蔽 LFSR 的位 [1:0]/三角波振幅等于 3

0010：不屏蔽 LFSR 的位 [2:0]/三角波振幅等于 7

0011：不屏蔽 LFSR 的位 [3:0]/三角波振幅等于 15

0100：不屏蔽 LFSR 的位 [4:0]/三角波振幅等于 31

0101：不屏蔽 LFSR 的位 [5:0]/三角波振幅等于 63

0110：不屏蔽 LFSR 的位 [6:0]/三角波振幅等于 127

0111：不屏蔽 LFSR 的位 [7:0]/三角波振幅等于 255

1000：不屏蔽 LFSR 的位 [8:0]/三角波振幅等于 511

1001：不屏蔽 LFSR 的位 [9:0]/三角波振幅等于 1023

1010：不屏蔽 LFSR 的位 [10:0]/三角波振幅等于 2047

≥ 1011：不屏蔽 LFSR 的位 [11:0]/三角波振幅等于 4095

位 7:6 **WAVE1[1:0]**：DAC 通道 1 噪声/三角波生成使能 (DAC channel1 noise/triangle wave generation enable)

这些位将由软件置 1 和清零。

00：禁止生成波

01：使能生成噪声波

1x：使能生成三角波

只在位 TEN1 = 1（使能 DAC 通道 1 触发）时使用。

位 5:2 **TSEL1[3:0]**：DAC 通道 1 触发器选择 (DAC channel1 trigger selection)

这些位用于选择 DAC 通道 1 的外部触发事件。

有关触发配置和映射的详细信息，请参见触发选择表。

注： 只在位 TEN1 = 1（使能 DAC 通道 1 触发）时使用。

位 1 **TEN1**：DAC 通道 1 触发使能 (DAC channel1 trigger enable)

此位由软件置 1 和清零，以使能/禁止 DAC 通道 1 触发。

0：禁止 DAC 通道 1 触发，写入 DACx_DHR1 寄存器的数据在一个 dac_pclk 时钟周期之后转移到 DACx_DOR1 寄存器

1：使能 DAC 通道 1 触发，DACx_DHR1 寄存器的数据在三个 dac_pclk 时钟周期之后转移到 DACx_DOR1 寄存器

*注： 如果选择软件触发，*DACx_DHR1 *寄存器的内容只需一个 dac_pclk 时钟周期即可转移到 DACx_DOR1 寄存器。*

位 0 **EN1**：DAC 通道 1 使能 (DAC channel1 enable)

此位由软件置 1 和清零，以使能/禁止 DAC 通道 1 。

0：禁止 DAC 通道 1

1：使能 DAC 通道 1

26.6.2 DAC x 软件触发寄存器 (DACx_SWTRGR)（x=1 到 2）

DAC x software trigger register

偏移地址：0x04
复位值：0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SWTRIG2	SWTRIG1
														w	w

位 31:2 保留，必须保持复位值。

位 1 **SWTRIG2**：DAC 通道 2 软件触发 (DAC channel2 software trigger)
该位由软件置 1，用于在软件触发模式下触发 DAC。
0：不触发
1：触发
注： 一旦 DACx_DHR2 寄存器值加载到 DACx_DOR2 寄存器中，该位即会由硬件清零（一个 dac_pclk 时钟周期之后）。

位 0 **SWTRIG1**：DAC 通道 1 软件触发 (DAC channel1 software trigger)
该位由软件置 1，用于在软件触发模式下触发 DAC。
0：不触发
1：触发
注： 一旦 DACx_DHR1 寄存器值加载到 DACx_DOR1 寄存器中，该位即会由硬件清零（一个 dac_pclk 时钟周期之后）。

26.6.3 DAC x 通道 1 12 位右对齐数据保持寄存器 (DACx_DHR12R1)（x = 1 到 2）

DAC x channel1 12-bit right-aligned data holding register

偏移地址：0x08
复位值：0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	DACC1DHR[11:0]											
				rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:12 保留，必须保持复位值。

位 11:0 **DACC1DHR[11:0]**：DAC 通道 1 12 位右对齐数据 (DAC channel1 12-bit right-aligned data)
这些位通过软件写入， 用于指定 DAC 通道 1 的 12 位数据。

26.6.4 DAC x 通道 1 12 位左对齐数据保持寄存器 (DACx_DHR12L1)（x=1 到 2）

DAC x channel1 12-bit left aligned data holding register

偏移地址：0x0C

复位值：0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DACC1DHR[11:0]												Res.	Res.	Res.	Res.
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw				

位 31:16 保留，必须保持复位值。

位 15:4 **DACC1DHR[11:0]**：DAC 通道 1 12 位左对齐数据 (DAC channel1 12-bit left-aligned data)
这些位通过软件写入，
用于指定 DAC 通道 1 的 12 位数据。

位 3:0 保留，必须保持复位值。

26.6.5 DAC x 通道 1 8 位右对齐数据保持寄存器 (DACx_DHR8R1)（x=1 到 2）

DAC x channel1 8-bit right aligned data holding register

偏移地址：0x10

复位值：0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DACC1DHR[7:0]							
								rw	rw	rw	rw	rw	rw	rw	rw

位 31:8 保留，必须保持复位值。

位 7:0 **DACC1DHR[7:0]**：DAC 通道 1 8 位右对齐数据 (DAC channel1 8-bit right-aligned data)
这些位通过软件写入， 用于指定 DAC 通道 1 的 8 位数据。

26.6.6 DAC x 通道 2 12 位右对齐数据保持寄存器 (DACx_DHR12R2)（x=1 到 2）

DAC x channel2 12-bit right aligned data holding register

偏移地址：0x14

复位值：0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	DACC2DHR[11:0]											
				rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:12 保留，必须保持复位值。

位 11:0 **DACC2DHR[11:0]**：DAC 通道 2 12 位右对齐数据 (DAC channel2 12-bit right-aligned data)
这些位通过软件写入， 用于指定 DAC 通道 2 的 12 位数据。

26.6.7 DAC x 通道 2 12 位左对齐数据保持寄存器 (DACx_DHR12L2)（x=1 到 2）

DAC x channel2 12-bit left aligned data holding register

偏移地址：0x18

复位值：0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DACC2DHR[11:0]												Res.	Res.	Res.	Res.
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw				

位 31:16 保留，必须保持复位值。

位 15:4 **DACC2DHR[11:0]**：DAC 通道 2 12 位左对齐数据 (DAC channel2 12-bit left-aligned data)
这些位由软件写入，用于为 DAC 通道 2 指定 12 位数据。

位 3:0 保留，必须保持复位值。

26.6.8 DAC x 通道 2 8 位右对齐数据保持寄存器 (DACx_DHR8R2)（x=1 到 2）

DAC x channel2 8-bit right-aligned data holding register

偏移地址：0x1C

复位值：0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DACC2DHR[7:0]							
								rw	rw	rw	rw	rw	rw	rw	rw

位 31:8 保留，必须保持复位值。

位 7:0 **DACC2DHR[7:0]**：DAC 通道 2 8 位右对齐数据 (DAC channel2 8-bit right-aligned data)
这些位由软件写入，用于为 DAC 通道 2 指定 8 位数据。

26.6.9 双 DAC x 12 位右对齐数据保持寄存器 (DACx_DHR12RD)（x=1 到 2）

Dual DAC x 12-bit right-aligned data holding register

偏移地址：0x20

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	DACC2DHR[11:0]											
				rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	DACC1DHR[11:0]											
				rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:28 保留，必须保持复位值。

位 27:16 **DACC2DHR[11:0]**：DAC 通道 2 12 位右对齐数据 (DAC channel2 12-bit right-aligned data)
这些位由软件写入，用于为 DAC 通道 2 指定 12 位数据。

位 15:12 保留，必须保持复位值。

位 11:0 **DACC1DHR[11:0]**：DAC 通道 1 12 位右对齐数据 (DAC channel1 12-bit right-aligned data)
这些位由软件写入，用于为 DAC 通道 1 指定 12 位数据。

26.6.10 双 DAC x 12 位左对齐数据保持寄存器(DACx_DHR12LD)（x=1 到 2）

Dual DAC x 12-bit left aligned data holding register

偏移地址：0x24

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DACC2DHR[11:0]												Res.	Res.	Res.	Res.
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw				
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DACC1DHR[11:0]												Res.	Res.	Res.	Res.
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw				

位 31:20 **DACC2DHR[11:0]**：DAC 通道 2 12 位左对齐数据 (DAC channel2 12-bit left-aligned data)
这些位由软件写入，用于为 DAC 通道 2 指定 12 位数据。

位 19:16 保留，必须保持复位值。

位 15:4 **DACC1DHR[11:0]**：DAC 通道 1 12 位左对齐数据 (DAC channel1 12-bit left-aligned data)
这些位由软件写入，用于为 DAC 通道 1 指定 12 位数据。

位 3:0 保留，必须保持复位值。

26.6.11 双 DAC x 8 位右对齐数据保持寄存器(DACx_DHR8RD)（x=1 到 2）

Dual DAC x 8-bit right aligned data holding register

偏移地址：0x28

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DACC2DHR[7:0]								DACC1DHR[7:0]							
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:16 保留，必须保持复位值。

位 15:8 **DACC2DHR[7:0]**：DAC 通道 2 8 位右对齐数据 (DAC channel2 8-bit right-aligned data)
这些位由软件写入，用于为 DAC 通道 2 指定 8 位数据。

位 7:0 **DACC1DHR[7:0]**：DAC 通道 1 8 位右对齐数据 (DAC channel1 8-bit right-aligned data)
这些位由软件写入，用于为 DAC 通道 1 指定 8 位数据。

26.6.12 DAC x 通道 1 数据输出寄存器 (DACx_DOR1)（x=1 到 2）

DAC x channel1 data output register

偏移地址：0x2C

复位值：0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	DACC1DOR[11:0]											
				r	r	r	r	r	r	r	r	r	r	r	r

位 31:12 保留，必须保持复位值。

位 11:0 **DACC1DOR[11:0]**：DAC 通道 1 数据输出 (DAC channel1 data output)
这些位为只读，其中包含 DAC 通道 1 的数据输出。

26.6.13 DAC x 通道 2 数据输出寄存器 (DACx_DOR2)（x=1 到 2）

DAC x channel2 data output register

偏移地址：0x30
复位值：0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	DACC2DOR[11:0]											
				r	r	r	r	r	r	r	r	r	r	r	r

位 31:12 保留，必须保持复位值。

位 11:0 **DACC2DOR[11:0]**：DAC 通道 2 数据输出 (DAC channel2 data output)
这些位为只读，其中包含 DAC 通道 2 的数据输出。

26.6.14 DAC x 状态寄存器 (DACx_SR)（x=1 到 2）

DAC x status register

偏移地址：0x34

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
BWST2	CAL_FLAG2	DMAUDR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
r	r	rc_w1													
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BWST1	CAL_FLAG1	DMAUDR1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
r	r	rc_w1													

位 31 **BWST2**：DAC 通道 2 写入采样时间忙标志 (DAC Channel 2 busy writing sample time flag)
此位会在采样和保持模式使能后由系统置 1， 每次软件写入 DACx_SHSR2 寄存器时，此位都会置 1，对 DACx_SHSR2 的写操作完成后，此位通过硬件清零。（需要约 3 个 LSI 同步周期）。
0：目前未对 DACx_SHSR2 进行写操作：DACx_SHSR2 可写入
1：目前正在对 DACx_SHSR2 进行写操作：DACx_SHSR2 不可写入

位 30 **CAL_FLAG2**：DAC 通道 2 校准偏移状态 (DAC Channel 2 calibration offset status)
此位通过硬件置 1 和清零。
0：校准调整值小于偏移校正值
1：校准调整值等于或大于偏移校正值

位 29 **DMAUDR2**：DAC 通道 2 DMA 下溢标志 (DAC channel2 DMA underrun flag)
此位由硬件置 1，由软件清零（写入 1）。
0：DAC 通道 2 未发生 DMA 下溢错误状况
1：DAC 通道 2 发生 DMA 下溢错误状况（当前所选触发源以高于 DMA 服务能力的频率驱动 DAC 通道 2 转换）

位 28 保留，必须保持复位值。

位 27 保留，必须保持复位值。

位 26:16 保留，必须保持复位值。

位 15 **BWST1**：DAC 通道 1 写入采样时间忙标志 (DAC Channel 1 busy writing sample time flag)
此位会在采样和保持模式使能后由系统置 1，每次软件写入 DACx_SHSR1 寄存器时，此位都会置 1，对 DACx_SHSR1 的写操作完成后，此位通过硬件清零。（需要约 3 个 LSI 同步周期）。
0：目前未对 DACx_SHSR1 进行写操作：DACx_SHSR1 可写入
1：目前正在对 DACx_SHSR1 进行写操作：DACx_SHSR1 不可写入

位 14 **CAL_FLAG1**：DAC 通道 1 校准偏移状态 (DAC Channel 1 calibration offset status)
此位通过硬件置 1 和清零。
0：校准调整值小于偏移校正值
1：校准调整值等于或大于偏移校正值

位 13 **DMAUDR1**：DAC 通道 1 DMA 下溢标志 (DAC channel1 DMA underrun flag)
此位由硬件置 1，由软件清零（写入 1）。
0：DAC 通道 1 未发生 DMA 下溢错误状况
1：DAC 通道 1 发生 DMA 下溢错误状况（当前所选触发源以高于 DMA 服务能力的频率驱动 DAC 通道 1 转换）

位 12 保留，必须保持复位值。

位 11 保留，必须保持复位值。

位 10:0 保留，必须保持复位值。

26.6.15 DAC x 校准控制寄存器 (DACx_CCR)（x=1 到 2）

DAC x calibration control register

偏移地址：0x38

复位值：0x00XX 00XX

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OTRIM2[4:0]				
											rw				

15	14	13	12	11	10			7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OTRIM1[4:0]				
											rw				

位 31:21 保留，必须保持复位值。

位 20:16 **OTRIM2[4:0]**：DAC 通道 2 偏移调整值 (DAC Channel 2 offset trimming value)

位 15:5 保留，必须保持复位值。

位 4:0 **OTRIM1[4:0]**：DAC 通道 1 偏移调整值 (DAC Channel 1 offset trimming value)

26.6.16 DAC x 模式控制寄存器 (DACx_MCR)（x=1 到 2）

DAC x mode control register

偏移地址：0x3C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MODE2[2:0]		
													rw		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MODE1[2:0]		
													rw		

位 31:26 保留，必须保持复位值。

位 25 保留，必须保持复位值。

位 24 保留，必须保持复位值。

位 23:19 保留，必须保持复位值。

位 18:16 **MODE2[2:0]**：DAC 通道 2 模式 (DAC Channel 2 mode)

仅当 DAC 已禁止且不处于校准模式时（DACx_CR 寄存器中的位 EN2 = 0 且位 CEN2 = 0），才可写入这些位。如果 EN2=1 或 CEN2 =1，则会忽略写操作。这些位可由软件置 1 和清零，用于选择 DAC 通道 2 模式。

– DAC 通道 2 处于正常模式

000：DAC 通道 2 连接到外部引脚且使能了缓冲器
001：DAC 通道 2 连接到外部引脚以及片上外设且使能了缓冲器
010：DAC 通道 2 连接到外部引脚且禁止了缓冲器
011：DAC 通道 2 连接到片上外设且禁止了缓冲器

– DAC 通道 2 处于采样和保持模式

100：DAC 通道 2 连接到外部引脚且使能了缓冲器
101：DAC 通道 2 连接到外部引脚以及片上外设且使能了缓冲器
110：DAC 通道 2 连接到外部引脚以及片上外设且禁止了缓冲器
111：DAC 通道 2 连接到片上外设且禁止了缓冲器

注： 仅可在 EN2 = 0 时修改该寄存器。

位 15:14 保留，必须保持复位值。

位 13:10 保留，必须保持复位值。

位 9 保留，必须保持复位值。

位 8 保留，必须保持复位值。

位 7:3 保留，必须保持复位值。

位 2:0 **MODE1[2:0]**：DAC 通道 1 模式 (DAC Channel 1 mode)

仅当 DAC 已禁止且不处于校准模式时（DACx_CR 寄存器中的位 EN1 = 0 且位 CEN1 = 0），才可写入这些位。如果 EN1=1 或 CEN1 =1，则会忽略写操作。这些位可由软件置 1 和清零，用于选择 DAC 通道 1 模式。

– DAC 通道 1 处于正常模式

000：DAC 通道 1 连接到外部引脚且使能了缓冲器
001：DAC 通道 1 连接到外部引脚以及片上外设且使能了缓冲器
010：DAC 通道 1 连接到外部引脚且禁止了缓冲器
011：DAC 通道 1 连接到片上外设且禁止了缓冲器

– DAC 通道 1 处于采样和保持模式

100：DAC 通道 1 连接到外部引脚且使能了缓冲器
101：DAC 通道 1 连接到外部引脚以及片上外设且使能了缓冲器
110：DAC 通道 1 连接到外部引脚以及片上外设且禁止了缓冲器
111：DAC 通道 1 连接到片上外设且禁止了缓冲器

注： 仅可在 EN1 = 0 时修改该寄存器。

26.6.17 DACx 采样和保持采样时间寄存器 1 (DACx_SHSR1) （x=1 到 2）

DAC x Sample and Hold sample time register 1

偏移地址：0x40

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	TSAMPLE1[9:0]									
						rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:10 保留，必须保持复位值。

位 9:0 **TSAMPLE1[9:0]**：DAC 通道 1 采样时间 (DAC Channel 1 sample Time)（仅在采样和保持模式下有效）

当 DAC 通道 1 已禁止或者处于正常运行期间时，可写入这些位。对于后一种情况，仅当 DACx_SCR 寄存器的 BWSTx 为低电平时，才可进行写操作，如果 BWSTx=1，会忽略写操作。

注： *它代表执行采样阶段的 LSI 时钟数。采样时间* = (TSAMPLE1[9:0] + 1) x *LSI 时钟周期。*

26.6.18 DACx 采样和保持采样时间寄存器 2 (DACx_SHSR2) （x=1 到 2）

DAC x Sample and Hold sample time register 2

偏移地址：0x44

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	TSAMPLE2[9:0]									
						rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:10 保留，必须保持复位值。

位 9:0 **TSAMPLE2[9:0]**：DAC 通道 2 采样时间 (DAC Channel 2 sample Time)（仅在采样和保持模式下有效）

当 DAC 通道 2 已禁止或者处于正常运行期间时，可写入这些位。对于后一种情况，仅当 DACx_SR 寄存器的 BWSTx 为低电平时，才可进行写操作，如果 BWSTx=1，会忽略写操作。

注： *它代表执行采样阶段的 LSI 时钟数。采样时间* = (TSAMPLE1[9:0] + 1) x *LSI 时钟周期。*

26.6.19 DAC x 采样和保持保持时间寄存器 (DACx_SHHR)（x=1 到 2）

DAC x Sample and Hold hold time register

偏移地址：0x48

复位值：0x0001 0001

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	THOLD2[9:0]									
						rw									
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	THOLD1[9:0]									
						rw									

位 31:26 保留，必须保持复位值。

位 25:16 **THOLD2[9:0]**：DAC 通道 2 保持时间 (DAC Channel 2 hold time)（仅在采样和保持模式下有效）。

保持时间 = (THOLD[9:0]) x LSI 时钟周期

注： 仅可在 EN2 = 0 时修改该寄存器。

位[15:10] 保留，必须保持复位值。

位 9:0 **THOLD1[9:0]**：DAC 通道 1 保持时间 (DAC Channel 1 hold Time)（仅在采样和保持模式下有效）

保持时间 = (THOLD[9:0]) x LSI 时钟周期

注： 仅可在 EN2 = 0 时修改该寄存器。

注： 仅当 DAC 通道已禁止且处于正常运行模式时（DACx_CR 寄存器中的位 ENx=0 且位 CEN2x=0），才可写入这些位。如果 ENx=1 或 CENx =1，则会忽略写操作。

26.6.20 DAC x 采样和保持刷新时间寄存器 (DACx_SHRR)（x=1 到 2）

DAC x Sample and Hold refresh time register

偏移地址：0x4C

复位值：0x0001 0001

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TREFRESH2[7:0]							
								rw							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TREFRESH1[7:0]							
								rw							

位 31:24 保留，必须保持复位值。

位 23:16 **TREFRESH2[7:0]**：DAC 通道 2 刷新时间 (DAC Channel 2 refresh Time)（仅在采样和保持模式下有效）

刷新时间 = (TREFRESH[7:0]) x LSI 时钟周期

注： 仅可在 EN2 = 0 时修改该寄存器。

位 15:8 保留，必须保持复位值。

位 7:0 **TREFRESH1[7:0]**：DAC 通道 1 刷新时间 (DAC Channel 1 refresh Time)（仅在采样和保持模式下有效）

刷新时间 = (TREFRESH[7:0]) x LSI 时钟周期

注： 仅可在 EN2 = 0 时修改该寄存器。

注： 仅当 DAC 通道已禁止且处于正常运行模式时（DACx_CR 寄存器中的位 ENx=0 且位 CEN2x=0），才可写入这些位。如果 ENx=1 或 CENx =1，则会忽略写操作。

26.6.21 DAC 寄存器映射

表 213 汇总了 DAC 寄存器。

表 213. DAC 寄存器映射和复位值

偏移	寄存器名	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	**DACx_CR**	Res.	CEN2	DMAUDRIE2	DMAEN2	MAMP2[3:0]				WAVE2[2:0]		TSEL23	TSEL22	TSEL21	TSEL20	TEN2	EN2	Res.	CEN1	DMAUDRIE1	DMAEN1	MAMP1[3:0]				WAVE1[2:0]		TSEL13	TSEL12	TSEL11	TSEL10	TEN1	EN1
	Reset value		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x04	**DACx_ SWTRGR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SWTRIG2	SWTRIG1
	Reset value																															0	0
0x08	**DACx_ DHR12R1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DACC1DHR[11:0]											
	Reset value																					0	0	0	0	0	0	0	0	0	0	0	0
0x0C	**DACx_ DHR12L1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DACC1DHR[11:0]												Res.	Res.	Res.	Res.
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0				
0x10	**DACx_ DHR8R1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DACC1DHR[7:0]							
	Reset value																									0	0	0	0	0	0	0	0
0x14	**DACx_ DHR12R2**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DACC2DHR[11:0]											
	Reset value																					0	0	0	0	0	0	0	0	0	0	0	0
0x18	**DACx_ DHR12L2**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DACC2DHR[11:0]												Res.	Res.	Res.	Res.
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0				
0x1C	**DACx_ DHR8R2**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DACC2DHR[7:0]							
	Reset value																									0	0	0	0	0	0	0	0
0x20	**DACx_ DHR12RD**	Res.	Res.	Res.	Res.	DACC2DHR[11:0]												Res.	Res.	Res.	Res.	DACC1DHR[11:0]											
	Reset value					0	0	0	0	0	0	0	0	0	0	0	0					0	0	0	0	0	0	0	0	0	0	0	0
0x24	**DACx_ DHR12LD**	DACC2DHR[11:0]												Res.	Res.	Res.	Res.	DACC1DHR[11:0]												Res.	Res.	Res.	Res.
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0					0	0	0	0	0	0	0	0	0	0	0	0				
0x28	**DACx_ DHR8RD**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DACC2DHR[7:0]								DACC1DHR[7:0]							
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x2C	**DACx_ DOR1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DACC1DOR[11:0]											
	Reset value																					0	0	0	0	0	0	0	0	0	0	0	0
0x30	**DACx_ DOR2**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DACC2DOR[11:0]											
	Reset value																					0	0	0	0	0	0	0	0	0	0	0	0

表 213. DAC 寄存器映射和复位值（续）

偏移	寄存器名	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x34	**DACx_SR**	BWST2	CAL_FLAG2	DMAUDR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BWST1	CAL_FLAG1	DMAUDR1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	Reset value	0	0	0														0	0	0													
0x38	**DACx_CCR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OTRIM2[4:0]					Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OTRIM1[4:0]				
	Reset value												X	X	X	X	X												X	X	X	X	X
0x3C	**DACx_MCR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MODE2 [2:0]			Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MODE1 [2:0]		
	Reset value														0	0	0														0	0	0
0x40	**DACx_ SHSR1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TSAMPLE1[9:0]									
	Reset value																							0	0	0	0	0	0	0	0	0	0
0x44	**DACx_ SHSR2**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TSAMPLE2[9:0]									
	Reset value																							0	0	0	0	0	0	0	0	0	0
0x48	**DACx_ SHHR**	Res.	Res.	Res.	Res.	Res.	Res.	THOLD2[9:0]										Res.	Res.	Res.	Res.	Res.	Res.	THOLD1[9:0]									
	Reset value							0	0	0	0	0	0	0	0	0	1							0	0	0	0	0	0	0	0	0	1
0x4C	**DACx_ SHRR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TREFRESH2[7:0]								Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TREFRESH1[7:0]							
	Reset value									0	0	0	0	0	0	0	1									0	0	0	0	0	0	0	1

有关寄存器边界地址的信息，请参见*第 2.2.2 节：存储器映射和寄存器边界地址*。

27 电压参考缓冲器 (VREFBUF)

27.1 简介

STM32H7x3 器件内置有电压参考缓冲器，既可用作 ADC 和 DAC 的参考电压，也可通过 VREF+ 引脚用作外部元件的参考电压。

27.2 VREFBUF 功能说明

内部电压参考缓冲器支持四种电压值(2)，可利用 VREFBUF_CSR 寄存器中的 VRS 位进行配置：

- VRS=000：大约为 2.5 V。
- VRS=001：大约为 2.048 V。
- VRS=010：大约为 1.8 V。
- VRS=011：大约为 1.5 V。

内部参考电压配置为四种不同模式，具体取决于 ENVR 和 HIZ 位的配置。下表列出了这些模式：

表 214. VREF 缓冲器模式

ENVR	(HiZ)	VREF 缓冲器配置
0	0	VREFBUF 缓冲器关闭 – V_{REF+} 引脚下拉到 V_{SSA}
0	1	外部参考电压模式（默认值）： – VREFBUF 缓冲器关闭 – V_{REF+} 引脚输入模式
1	0	内部参考电压模式： – VREFBUF 缓冲器开启 – V_{REF+} 引脚连接到 VREFBUF 缓冲器输出
1	1	保持模式： – VREFBUF 缓冲器关闭 – V_{REF+} 引脚悬空。借助外部电容来保持电压 – 禁止 VRR 检测，VRR 位保持最后一个状态

通过将 VREFBUF_CSR 寄存器中的 ENVR 位置 1 并将 HIZ 位清零使能 VREFBUF 后，用户必须等待至 VRR 位置 1，指示参考电压输出已达到预期值。

2. 最小 V_{DDA} 电压取决于 VRS 设置，请参见产品数据手册。

27.3 VREFBUF 寄存器

27.3.1 VREFBUF 控制和状态寄存器 (VREFBUF_CSR)

VREFBUF control and status register

偏移地址：0x00

复位值：0x0000 0002

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res	Res	Res	Res	Res	Res	Res	Res	Res	VRS[2:0]			VRR	Res	(HiZ)	ENVR
									rw	rw	rw	r		rw	rw

位 31:7 保留，必须保持复位值。

位 6:4 **VRS[2:0]**：参考电压调节 (Voltage reference scale)
这些位选择由电压参考缓冲器生成的值。
000：参考电压设为 2.5 V
001：参考电压设为 2.048 V
010：参考电压设为 1.8 V
011：参考电压设为 1.5 V
其它：保留

位 3 **VRR**：电压参考缓冲器就绪 (Voltage reference buffer ready)
0：电压参考缓冲器输出未就绪。
1：电压参考缓冲器输出已达到请求值。

位 2 保留，必须保持复位值。

位 1 **HIZ**：高阻态模式 (High impedance mode)
此位控制模拟开关是否连接到 V_{REF+} 引脚。
0：V_{REF+} 从内部连接到电压参考缓冲器输出。
1：V_{REF+} 引脚为高阻态。
关于 ENVR 位配置及模式说明，请参见*表 214：VREF 缓冲器模式*。

位 0 **ENVR**：电压参考缓冲器模式使能 (Voltage reference buffer mode enable)
此位用于使能电压参考缓冲器模式。
0：禁止内部参考电压模式（外部参考电压模式）。
1：使能内部参考电压模式（参考缓冲器使能或保持模式）。

27.3.2 VREFBUF 校准控制寄存器 (VREFBUF_CCR)

VREFBUF calibration control register

偏移地址：0x04

复位值：0x0000 00XX

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

位 31:6 保留，必须保持复位值。

位 5:0 **TRIM[5:0]**：微调代码 (Trimming code)

在生产测试期间，使用 Flash 中存储的的微调值进行复位后，这些位将自动初始化。写入这些位可调整内部参考缓冲器电压。

27.3.3 VREFBUF 寄存器映射

下表列出了 VREFBUF 寄存器映射和复位值。

表 215. VREFBUF 寄存器映射和复位值

偏移	寄存器	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	**VREFBUF_CSR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.		VRS[2:0]		VRR	Res.	HIZ	ENVR
	Reset value																										0	0	0	0		1	0
0x04	**VREFBUF_CCR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TRIM[5:0]					
	Reset value																											x	x	x	x	x	x

有关寄存器边界地址的信息，请参见*第 100 页的第 2.2.2 节*。

28 比较器 (COMP)

28.1 简介

器件内置两个超低功耗比较器通道（COMP1 和 COMP2）。他们可用于各种功能，包括：

- 在模拟信号的触发下从低功耗模式唤醒
- 模拟信号调理
- 与定时器的 PWM 输出结合使用时，构成逐周期电流控制环路

28.2 COMP 主要特性

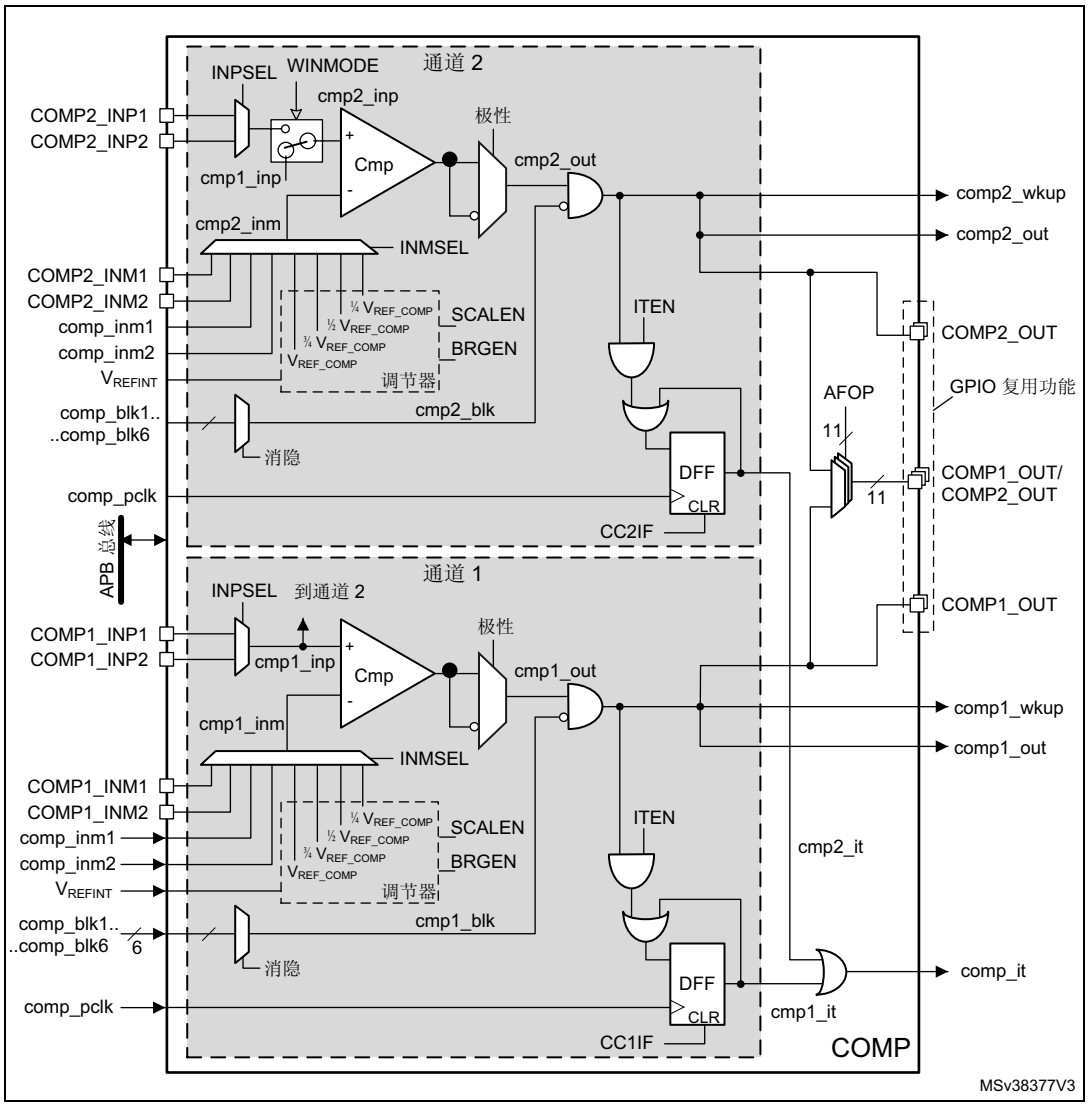
- 可选反相模拟输入：
  - I/O 引脚（因通道而异）
  - DAC 通道 1 和通道 2 输出
  - 通过调节器（缓冲分压器）提供的内部参考电压和三个因数分压值（1/4、1/2、3/4）
- 每个通道有两个 I/O 引脚可选作非反相模拟输入
- 可编程迟滞
- 可编程速度/功耗
- 将输出映射到 I/O
- 将输出重定向到用于触发以下事件的定时器输入：
  - 捕捉事件
  - OCREF_CLR 事件（用于逐周期电流控制）
  - 断路事件（用于快速 PWM 关断）
- 消隐比较器输出
- 窗口比较器
- 可生成中断，用于使器件从睡眠模式和停止模式唤醒（通过 EXIT 控制器）
- 将中断输出定向到 CPU

28.3 COMP 功能说明

28.3.1 COMP框图

比较器的框图如*图 204：比较器功能框图*所示。

图 204. 比较器功能框图



28.3.2 COMP 引脚和内部信号

用作比较器输入的 I/O 必须在 GPIO 寄存器中配置为模拟模式。

比较器输出可通过其复用功能连接到 I/O。请参见产品数据手册。

输出也可以在内部重定向到用于以下用途的各种定时器输入：

- 使用 BKIN 和 BKIN2 输入紧急关断 PWM 信号
- 使用定时器的 ETR 输入进行逐周期电流控制
- 用于时序测量的输入捕捉

比较器输出可以在内部同时连接到 I/O 引脚。

表 216. COMP 输入/输出内部信号

信号名称	信号类型	说明
comp_inm1	模拟输入	两个 COMP 通道的反相输入源：DAC 通道 1
comp_inm2	模拟输入	两个 COMP 通道的反相输入源：DAC 通道 2
comp_blk1	数字输入	两个 COMP 通道的消隐输入源：TIM1 OC5
comp_blk2	数字输入	两个 COMP 通道的消隐输入源：TIM2 OC3
comp_blk3	数字输入	两个 COMP 通道的消隐输入源：TIM3 OC3
comp_blk4	数字输入	两个 COMP 通道的消隐输入源：TIM3 OC4
comp_blk5	数字输入	两个 COMP 通道的消隐输入源：TIM8 OC5
comp_blk6	数字输入	两个 COMP 通道的消隐输入源：TIM15 OC1
comp_pclk	数字输入	两个 COMP 通道的 APB 时钟
comp1_wkup	数字输出	COMP 通道 1 唤醒输出
comp1_out	数字输出	COMP 通道 1 输出
comp2_wkup	数字输出	COMP 通道 2 唤醒输出
comp2_out	数字输出	COMP 通道 2 输出
comp_it	数字输出	COMP 中断输出

表 217. COMP 输入/输出引脚

信号名称	信号类型	说明
COMP1_INM1	模拟输入	COMP 通道 1 反相输入源 1 (PB1)
COMP1_INM2	模拟输入	COMP 通道 1 反相输入源 2 (PC4)
COMP1_INP1	模拟输入	COMP 通道 1 非反相输入源 1 (PB0)
COMP1_INP2	模拟输入	COMP 通道 1 非反相输入源 2 (PB2)
COMP2_INM1	模拟输入	COMP 通道 2 反相输入源 1 (PE10)
COMP2_INM2	模拟输入	COMP 通道 2 反相输入源 2 (PE7)
COMP2_INP1	模拟输入	COMP 通道 2 非反相输入源 1 (PE9)
COMP2_INP2	模拟输入	COMP 通道 2 非反相输入源 2 (PE11)
COMP1_OUT	数字输出	COMP 通道 1 输出：请参见*第 28.3.8 节：GPIO 上的比较器输出*。
COMP2_OUT	数字输出	COMP 通道 2 输出：请参见*第 28.3.8 节：GPIO 上的比较器输出*。

28.3.3 COMP 复位和时钟

时钟控制器提供的时钟 comp_pclk 与 APB 时钟同步。

注： ***重要提示：****极性选择逻辑和到端口的输出重定向独立于 APB 时钟。因此，即使在停止模式下比较器仍能正常工作。连接至 CPU 的 NVIC 的中断线需要 APB 时钟 (comp_pclk) 才能工作。如果没有 APB 时钟，则无法生成中断信号 comp_it。*

28.3.4 比较器锁定机制

这两个比较器可用于过流或热保护等安全用途。对于具有特定功能安全要求的应用，可对比较器配置进行保护，以防发生意外修改（例如，当程序计数器损坏时）。

为此，可以对比较器配置寄存器进行写保护（只读）。

一旦比较器通道配置完成，其 LOCK 位设置为 1。从而使得只能读取比较器通道的整个寄存器组以及通用 COMP_OR 寄存器，包括 LOCK 位。

只能通过 MCU 复位来移除写保护。

COMP_OR 寄存器由 COMP_CFGR1 或 COMP_CFGR2 的 LOCK 位进行锁定。

28.3.5 窗口比较器

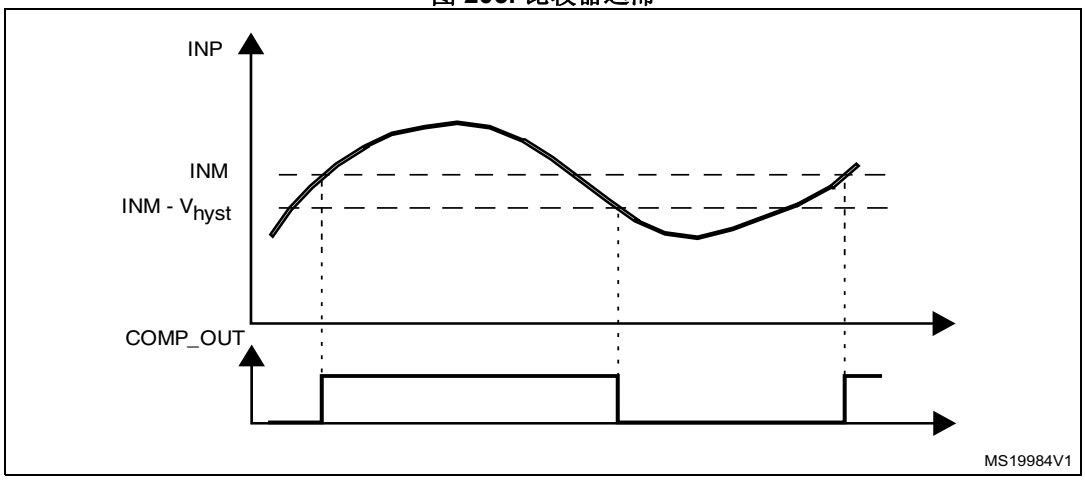
窗口比较器用于监视模拟电压并检查其是否处于阈值上下限所定义的特定电压范围内。

窗口比较器需要两个 COMP 通道。受监视的模拟电压连接至其非反相 (+) 输入，阈值上下限电压分别连接至各比较器的反相 (-) 输入。通过使能 WINMODE 位，可使 COMP 通道 2 的非反相输入在内部与 COMP 通道 1 的非反相输入相连。这可以节省 COMP 通道 2 的输入引脚以用于其他用途。请参见*图 204：比较器功能框图*。

28.3.6 迟滞

比较器具有可编程迟滞，可在有噪声信号时避免发生意外输出转换。迟滞可在不需要时（例如，退出低功耗模式时）禁止，以便使用外部组件强制迟滞值。

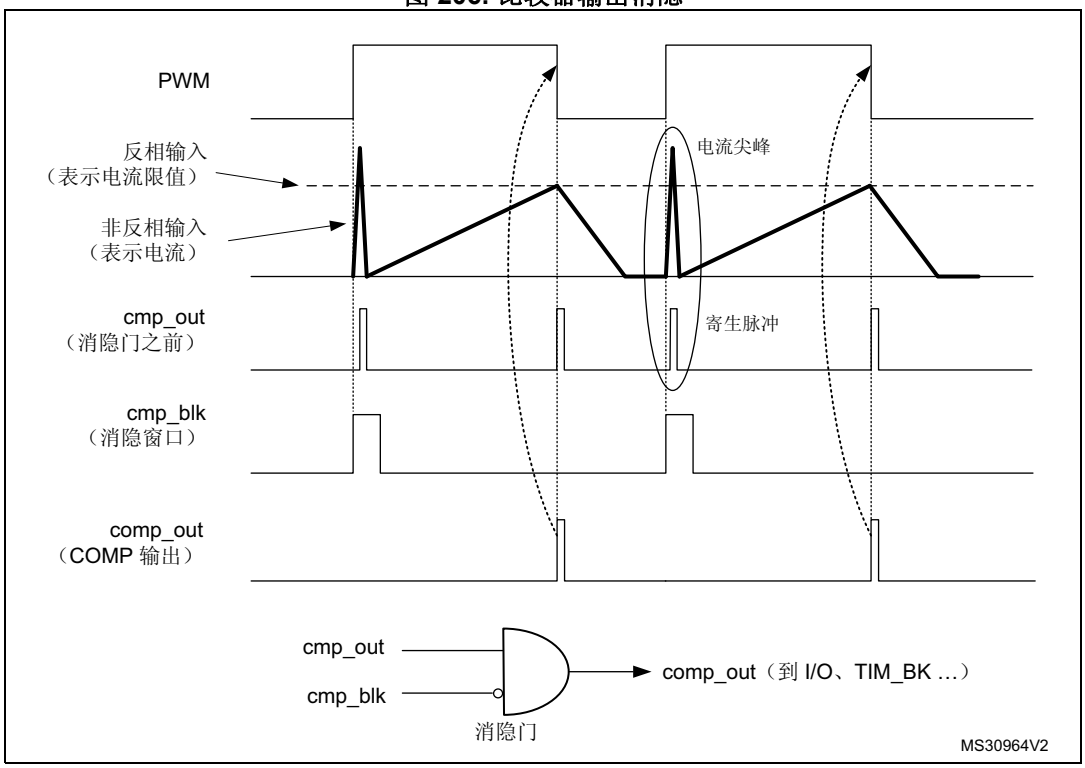
图 205. 比较器迟滞



28.3.7 比较器输出消隐功能

消隐功能的目的是防止电流调节在 PWM 周期开始处出现短暂电流尖峰（通常为功率开关反向并联二极管中的恢复电流）时发生跳闸。该功能使用通过定时器输出比较信号定义的消隐窗口。有关可选消隐信号，请参见寄存器说明。消隐信号对内部比较器输出进行门控，以便能使 comp_out 免受因电流尖峰而导致的寄生脉冲的干扰，如*图 206* 所示（图中未显示 COMP 通道编号）。

图 206. 比较器输出消隐



28.3.8 GPIO 上的比较器输出

比较器通道的 COMP1_OUT 和 COMP2_OUT 输出通过 COMP_OR 寄存器的 AFOP 字段（位 [10:0]）以及通过 GPIO 复用功能映射到 GPIO。

表 218. COMP1_OUT 的 GPIO 分配

COMP1_OUT	复用功能
PC5	AF13
PE12	AF13
PA6	AF10、AF12（可用作定时器刹车输入）
PA8	AF12（可用作定时器刹车输入）
PB12	AF13（可用作定时器刹车输入）
PE6	AF11（可用作定时器刹车输入）
PE15	AF13（可用作定时器刹车输入）
PG2	AF11（可用作定时器刹车输入）
PG3	AF11（可用作定时器刹车输入）
PG4	AF11（可用作定时器刹车输入）
PI1	AF11（可用作定时器刹车输入）
PI4	AF11（可用作定时器刹车输入）
PK2	AF10、AF11（可用作定时器刹车输入）

表 219. COMP2_OUT 的 GPIO 分配

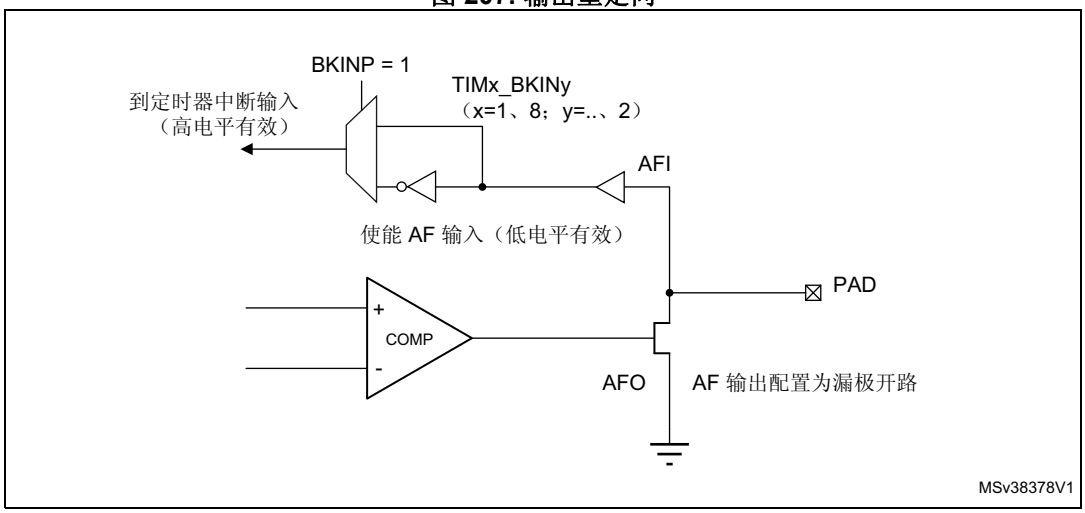
COMP2_OUT	复用功能
PE8	AF13
PE13	AF13
PA6	AF10、AF12（可用作定时器刹车输入）
PA8	AF12（可用作定时器刹车输入）
PB12	AF13（可用作定时器刹车输入）
PE6	AF11（可用作定时器刹车输入）
PE15	AF13（可用作定时器刹车输入）
PG2	AF11（可用作定时器刹车输入）
PG3	AF11（可用作定时器刹车输入）
PG4	AF11（可用作定时器刹车输入）
PI1	AF11（可用作定时器刹车输入）
PI4	AF11（可用作定时器刹车输入）
PK2	AF10、AF11（可用作定时器刹车输入）

两种比较器通道输出的 GPIO 分配都必须在锁定任一通道的寄存器之前完成，因为在锁定任一比较器通道的寄存器时会锁定通用 COMP_OR 寄存器。

28.3.9 比较器输出重定向

任一 COMP 通道的输出均可重定向到定时器刹车输入（TIMx_BKIN 或 TIMx_BKIN2），如 *图 207* 所示。为此，可将 COMP 通道输出连接至其中一个 GPIO（可在复用功能中进行编程），作为定时器刹车输入。见 *表 218* 和 *表 219*。所选 GPIO 必须被设置为开漏模式。COMP 输出通过 GPIO 传递至定时器刹车输入。通过上拉电阻，所选 GPIO 可用作定时器刹车输入（与比较器输出进行逻辑 OR 运算）。

图 207. 输出重定向



28.3.10 COMP 功耗和速度模式

对于给定的应用，可调节 COMP 通道功耗与传播延迟以获得最佳平衡。

COMP_CFGRx 寄存器的位 PWRMODE[1:0] 可编程为如下值：

00：高速/全功耗
01：中速/中等功耗
10：中速/中等功耗
11：极低速/超低功耗

28.4 COMP 低功耗模式

表 220. 低功耗模式下的比较器行为

模式	说明
Sleep	对比较器无影响。 比较器中断可使器件退出睡眠模式。
Stop	对比较器无影响。 比较器中断可使器件退出停止模式。

注： *当内部参考电压关闭时，不能使用比较器将器件从休眠或停止模式退出。*

28.5 COMP 中断

可通过两种方式将比较器用作中断源。

比较器输出从内部连接到扩展中断和事件控制器。每个比较器都具有其各自的 EXTI 线，能够产生中断或事件，可使器件退出低功耗模式。

比较器还提供有连接至 CPU 的 NVIC 的中断线。在 CPU 激活时使用该功能处理低延迟中断。这需要 APB 时钟处于运行状态。

28.5.1 通过 EXTI 模块实现的中断

更多详细信息，请参见中断和事件部分。

通过 EXTI 模块实现 COMPx 中断的程序：

1. 将 EXTI 线（用于接收 comp_wkup 信号）配置为中断模式，选择上升沿、下降沿或任一边沿有效，然后使能 EXTI 线。
2. 配置并使能映射到相应 EXTI 线的 NVIC IRQ 通道。
3. 使能 COMPx。

表 221. 中断控制位

中断事件	事件标志	使能控制位	退出睡眠模式	从停止模式退出	退出待机模式
comp1_wkup	通过 EXTI	通过 EXTI	是	是	N/A
comp2_whup	通过 EXTI	通过 EXTI	是	是	N/A

28.5.2 通过 CPU 的 NVIC 实现中断

通过 CPU 的 NVIC 实现 COMPx 中断的程序：

1. 配置并使能映射到 comp_it 线的 NVIC IRQ 通道。
2. 在 COMP_CFGRx 中配置并使能 ITEN。
3. 使能 COMPx。

表 222. 中断控制位

中断事件	中断标志	使能控制位	中断清除位	退出睡眠模式	从停止模式退出
comp_it	C1IF 输入	COMP_CFGR1 中的 ITEN	CC1IF	是（使用 APB 时钟）	否
comp_it	C2IF 输入	COMP_CFGR2 中的 ITEN	CC2IF	是（使用 APB 时钟）	否

注：　要使用该中断，需要使能 APB 时钟。如果未使能该时钟，则不会生成中断。

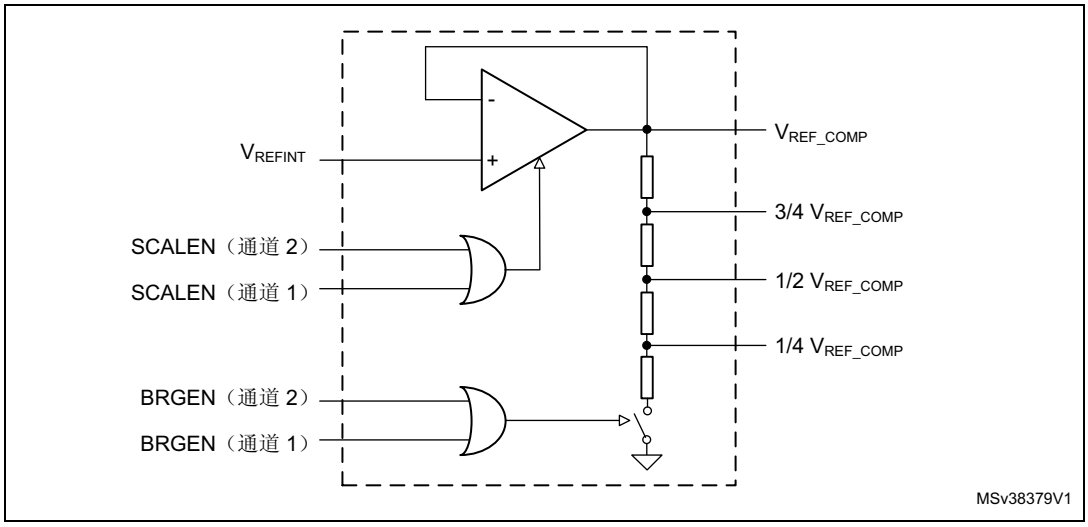
28.6 SCALER 功能

调节器模块用于为各个比较器输入提供不同的参考电压。其基于一个驱动电阻桥的放大器。放大器输入连接至内部参考电压。

放大器和电阻桥可单独使能。放大器由 COMP_CFGRx 寄存器的 SCALEN 位使能。电阻桥由 COMP_CFGRx 寄存器的 BRGEN 位使能。

如果未使用电阻分压，则可断开电阻桥，以降低功耗。断开后，1/4 V_{REF_COMP}、1/2 V_{REF_COMP} 和 3/4 V_{REF_COMP} 值等于 V_{REF_COMP}。

图 208. 框图



28.7 COMP 寄存器

28.7.1 比较器状态寄存器 (COMP_SR)

Comparator status register

COMP_SR 为比较器状态寄存器。

偏移地址：0x00

系统复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	C2IF	C1IF
														r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	C2VAL	C1VAL
														r	r

位 31:18 保留，必须保持复位值。

位 17 **C2IF**：COMP 通道 2 中断标志 (COMP channel 2 Interrupt Flag)
当 COMP 通道 2 输出置 1 时，该位由硬件置 1。
该位由软件通过向 COMP_ICFR 寄存器的 CC2IF 位写入 1 来清零。

位 16 **C1IF**：COMP 通道 1 中断标志 (COMP channel 1 Interrupt Flag)
当 COMP 通道 1 输出置 1 时，该位由硬件置 1。
该位由软件通过向 COMP_ICFR 寄存器的 CC1IF 位写入 1 来清零。

位 15:2 保留，必须保持复位值。

位 1 **C2VAL**：COMP 通道 2 输出状态位 (COMP channel 2 output status bit)
此位为只读。它反映当前 COMP 通道 2 输出的状态（考虑了 POLARITY 和 BLANKING 位的影响）。

位 0 **C1VAL**：COMP 通道 1 输出状态位 (COMP channel 1 output status bit)
此位为只读。它反映当前 COMP 通道 1 输出的状态（考虑了 POLARITY 和 BLANKING 位的影响）。

28.7.2 比较器中断清除标志寄存器 (COMP_ICFR)

Comparator interrupt clear flag register

COMP_ICFR 为比较器中断清除标志寄存器。

偏移地址：0x00

系统复位值：0x0000 0004

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CC2IF	CC1IF
														w1o	w1o

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

位 31:18 保留，必须保持复位值。

位 17 **CC2IF**：清除 COMP 通道 2 中断标志 (Clear COMP channel 2 Interrupt Flag)
写入 1 可将 COMP_SR 寄存器中的 C2IF 标志清零

位 16 **CC1IF**：清除 COMP 通道 1 中断标志 (Clear COMP channel 1 Interrupt Flag)
写入 1 可将 COMP_SR 寄存器中的 C1IF 标志清零

位 15:0 保留，必须保持复位值。

28.7.3 比较器选项寄存器 (COMP_OR)

Comparator option register

COMP_OR 为比较器选项寄存器。

偏移地址：0x08

系统复位值：0x0000 0000

当 OR_CFG = 0 时

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OR15	OR14	OR13	OR12	OR11	AFOP										
rw	rw	rw	rw	rw	rw										

位 31:16 保留，必须保持复位值。

位 15:11 **OR**：选项寄存器 (Option Register)（当 OR_CFG = 0 时）

位 10:0 **AFOP[10:0]**：选择输出端口复用功能源 (Selection of source for alternate function of output ports)

该字段的各个位由软件置 1 和清零（仅限 LOCK 未置 1 的情况）。
输出端口 (GPIO) 对应关系：

位 10	位 9	位 8	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
PK2	PI4	PI1	PG4	PG3	PG2	PE15	PE6	PB12	PA8	PA6

对于各个位：
0：为相应 GPIO 的复用功能选择 COMP1_OUT。
1：为相应 GPIO 的复用功能选择 COMP2_OUT。

当 OR_CFG = 1 时

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
OR31	OR30	OR29	OR28	OR27	OR26	OR25	OR24	OR23	OR22	OR21	OR20	OR19	OR18	OR17	OR16
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OR15	OR14	OR13	OR12	OR11	AFOP										
rw	rw	rw	rw	rw	rw										

位 31:11 **OR**：选项寄存器 (Option Register)（当 OR_CFG = 1 时）

位 10:0 **AFOP[10:0]**：选择输出端口复用功能源 (Selection of source for alternate function of output ports)

该字段的各个位由软件置 1 和清零（仅限 LOCK 未置 1 的情况）。

输出端口 (GPIO) 对应关系：

位 10	位 9	位 8	位 7	位 6	位 5	位 4	位 3	位 2	位 1	位 0
PK2	PI4	PI1	PG4	PG3	PG2	PE15	PE6	PB12	PA8	PA6

对于各个位：

0：为相应 GPIO 的复用功能选择 COMP1_OUT。

1：为相应 GPIO 的复用功能选择 COMP2_OUT。

28.7.4 比较器配置寄存器 1 (COMP_CFGR1)

Comparator configuration register 1

COMP_CFGR1 为 COMP 通道 1 配置寄存器。

偏移地址：0x0C

系统复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LOCK	Res.	Res.	Res.	BLANKING[3:0]				Res.	Res.	Res.	INPSEL	Res.	INMSEL[2:0]		
rw				rw							rw		rw		
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	PWRMODE[1:0]		Res.	Res.	HYST[1:0]		Res.	ITEN	Res.	Res.	POLARITY	SCALEN	BRGEN	EN
		rw				rw			rw			rw	rw	rw	rw

位 31 **LOCK**：锁定位 (Lock bit)

此位由软件置 1，通过硬件系统复位清零。它可以锁定 COMP 通道 1 配置寄存器 COMP_CFGR1[31:0] 和 COMP_OR 寄存器的全部内容

0：可以读/写 COMP_CFGR1[31:0] 寄存器

1：只能读取 COMP_CFGR1[31:0] 和 COMP_OR 寄存器

位 30:28 保留，必须保持复位值。

位 27:24 **BLANKING[3:0]**：COMP 通道 1 消隐源选择位 (COMP channel 1 blanking source selection bits)
该字段的各个位由软件置 1 和清零（仅限 LOCK 未置 1 的情况）。
此字段选择 COMP 通道 1 输出消隐输入源：
0000：无消隐
0001：comp_blk1
0010：comp_blk2
0011：comp_blk3
0100：comp_blk4
0101：comp_blk5
0110：comp_blk6
所有其它值：保留

位 23:21 保留，必须保持复位值。

位 20 **INPSEL**：COMP 通道 1 非反相输入选择位 (COMP channel 1 non-inverting input selection bit)
此位由软件置 1 和清零（仅限 LOCK 未置 1 的情况）。
0：COMP1_INP1 (PB0)
1：COMP1_INP2 (PB2)

位 19 保留，必须保持复位值。

位 18:16 **INMSEL[2:0]**：COMP 通道 1 反相输入选择字段 (COMP channel 1 inverting input selection field)
这些位由软件置 1 和清零（仅限 LOCK 未置 1 的情况）。他们选择哪个输入连接到 COMP 通道 1 的负输入。
000 = 1/4 V_{REF_COMP}
001 = 1/2 V_{REF_COMP}
010 = 3/4 V_{REF_COMP}
011 = V_{REF_COMP}
100 = comp_inm1（DAC 通道 1 输出）
101 = comp_inm2（DAC 通道 2 输出）
110 = COMP1_INM1 (PB1)
111 = COMP1_INM2 (PC4)

位 15:14 保留，必须保持复位值。

位 13:12 **PWRMODE[1:0]**：COMP 通道 1 的功耗模式 (Power Mode of the COMP channel 1)
这些位由软件置 1 和清零（仅限 LOCK 未置 1 的情况）。他们控制 COMP 通道 1 的功耗/速度。
00：高速/全功耗
01：中速/中等功耗
10：中速/中等功耗
11：极低速/超低功耗

位 11:10 保留，必须保持复位值。

位 9:8 **HYST[1:0]**：COMP 通道 1 迟滞选择位 (COMP channel 1 hysteresis selection bits)
这些位由软件置 1 和清零（仅限 LOCK 未置 1 的情况）。他们选择 COMP 通道 1 的迟滞电压。
00：无迟滞
01：低迟滞
10：中等迟滞
11：高迟滞

位 7 保留，必须保持复位值。

位 6 **ITEN**：COMP 通道 1 中断使能 (COMP channel 1 interrupt enable)
此位由软件置 1 和清零（仅限 LOCK 未置 1 的情况）。该位使能 COMP 通道 1 生成中断。
0：禁止 COMP 通道 1 生成中断
1：使能 COMP 通道 1 生成中断

位 5:4 保留，必须保持复位值。

位 3 **POLARITY**：COMP 通道 1 极性选择位 (COMP channel 1 polarity selection bit)
此位由软件置 1 和清零（仅限 LOCK 未置 1 的情况）。它反转 COMP 通道 1 极性。
0：不反转 COMP 通道 1 输出
1：反转 COMP 通道 1 输出

位 2 **SCALEN**：电压调节器使能位 (Voltage scaler enable bit)
此位由软件置 1 和清零（仅限 LOCK 未置 1 的情况）。该位使能 COMP 通道的 V_{REFINT} 调节器。
0：禁止 V_{REFINT} 调节器（COMP_CFGR2 寄存器的 SCALEN 位也为低电平时）
1：使能 V_{REFINT} 调节器

位 1 **BRGEN**：调节器电阻桥使能 (Scaler bridge enable)
此位由软件置 1 和清零（仅限 LOCK 未置 1 的情况）。该位可使能调节器电阻桥。
0：禁止调节器电阻桥（COMP_CFGR2 寄存器的 BRGEN 位也为低电平时）
1：使能调节器电阻桥
如果 SCALEN 置 1 且 BRGEN 复位，则全部四个调节器输出都会提供相同值 V_{REF_COMP}（近似于 V_{REFINT}）。
如果 SCALEN 和 BRGEN 均置 1，则四个调节器输出会分别提供 V_{REF_COMP}、3/4 V_{REF_COMP}、1/2 V_{REF_COMP} 和 1/4 V_{REF_COMP} 值。

位 0 **EN**：COMP 通道 1 使能位 (COMP channel 1 enable bit)
此位由软件置 1 和清零（仅限 LOCK 未置 1 的情况）。它使能 COMP 通道 1。
0：禁止
1：使能

28.7.5 比较器配置寄存器 2 (COMP_CFGR2)

Comparator configuration register 2

COMP_CFGR2 为 COMP 通道 2 配置寄存器。

偏移地址：0x10

系统复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LOCK	Res.	Res.	Res.	BLANKING[3:0]				Res.	Res.	Res.	INPSEL	Res.	INMSEL[2:0]		
rw				rw							rw		rw		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	PWRMODE[1:0]		Res.	Res.	HYST[1:0]		Res.	ITEN	Res.	WINMODE	POLARITY	SCALEN	BRGEN	EN
		rw				rw			rw		rw	rw	rw	rw	rw

位 31 **LOCK**：锁定位 (Lock bit)
此位由软件置 1，通过硬件系统复位清零。它可以锁定 COMP 通道 2 配置寄存器 COMP_CFGR2[31:0] 和 COMP_OR 寄存器的全部内容
0：可以读/写 COMP_CFGR2[31:0] 寄存器
1：只能读取 COMP_CFGR2[31:0] 和 COMP_OR 寄存器

位 30:28 保留，必须保持复位值。

位 27:24 **BLANKING[3:0]**：COMP 通道 2 消隐源选择位 (COMP channel 2 blanking source selection bits)

这些位由软件置 1 和清零（仅限 LOCK 未置 1 的情况）。这些位选择哪个定时器输出控制 COMP 通道 2 输出消隐。

0000：无消隐
0001：选择 TIM1 OC5 作为消隐源
0010：选择 TIM2 OC3 作为消隐源
0011：选择 TIM3 OC3 作为消隐源
0100：选择 TIM3 OC4 作为消隐源
0101：选择 TIM8 OC5 作为消隐源
0110：选择 TIM15 OC1 作为消隐源
所有其它值：保留

位 23:21 保留，必须保持复位值。

位 20 **INPSEL**：COMP 通道 2 非反相输入选择位 (COMP channel 2 non-inverting input selection bit)

此位由软件置 1 和清零（仅限 LOCK 未置 1 的情况）。

0：COMP2_INP1 (PE9)
1：COMP2_INP2 (PE11)

位 19 保留，必须保持复位值。

位 18:16 **INMSEL[2:0]**：COMP 通道 2 反相输入选择字段 (COMP channel 2 inverting input selection field)

这些位由软件置 1 和清零（仅限 LOCK 未置 1 的情况）。他们选择哪个输入连接到 COMP 通道 2 的负输入。

000 = 1/4 V_{REF_COMP}
001 = 1/2 V_{REF_COMP}
010 = 3/4 V_{REF_COMP}
011 = V_{REF_COMP}
100 = comp_inm1（DAC 通道 1 输出）
101 = comp_inm2（DAC 通道 2 输出）
110 = COMP2_INM1 (PE10)
111 = COMP2_INM2 (PE7)

位 15:14 保留，必须保持复位值。

位 13:12 **PWRMODE[1:0]**：COMP 通道 2 的功耗模式 (Power Mode of the COMP channel 2)

这些位由软件置 1 和清零（仅限 LOCK 未置 1 的情况）。他们控制 COMP 通道 2 的功耗/速度。

00：高速/全功耗
01：中速/中等功耗
10：中速/中等功耗
11：极低速/超低功耗

位 11:10 保留，必须保持复位值。

位 9:8 **HYST[1:0]**：COMP 通道 2 迟滞选择位 (COMP channel 2 hysteresis selection bits)

这些位由软件置 1 和清零（仅限 LOCK 未置 1 的情况）。他们选择 COMP 通道 2 的迟滞电压。

00：无迟滞
01：低迟滞
10：中等迟滞
11：高迟滞

位 7 保留，必须保持复位值。

位 6 **ITEN**：COMP 通道 2 中断使能 (COMP channel 2 interrupt enable)
此位由软件置 1 和清零（仅限 LOCK 未置 1 的情况）。该位使能 COMP 通道 2 生成中断。
0：禁止 COMP 通道 2 生成中断
1：使能 COMP 通道 2 生成中断

位 5 保留，必须保持复位值。

位 4 **WINMODE**：窗口比较器模式选择位 (Window comparator mode selection bit)
此位由软件置 1 和清零（仅限 LOCK 未置 1 的情况）。该位选择比较器的窗口模式。如果该位置 1，则 COMP 通道 2 的非反相输入连接至 COMP 通道 1 的非反相输入。
COMP 通道 2 的非反相输入的连接对象取决于该位值：
0：连接至 COMP2_INP 输入选择器
1：连接至 COMP 通道 1 的非反相输入 comp1_inp

位 3 **POLARITY**：COMP 通道 2 极性选择位 (COMP channel 2 polarity selection bit)
此位由软件置 1 和清零（仅限 LOCK 未置 1 的情况）。它反转 COMP 通道 2 极性。
0：不反转 COMP 通道 2 输出
1：反转 COMP 通道 2 输出

位 2 **SCALEN**：电压调节器使能位 (Voltage scaler enable bit)
此位由软件置 1 和清零（仅限 LOCK 未置 1 的情况）。该位使能 COMP 通道的 V_{REFINT} 调节器。
0：V_{REFINT} 调节器已禁止（COMP_CFGR1 寄存器的 SCALEN 位也为低电平时）
1：使能 V_{REFINT} 调节器

位 1 **BRGEN**：调节器电阻桥使能 (Scaler bridge enable)
此位由软件置 1 和清零（仅限 LOCK 未置 1 的情况）。该位可使能调节器电阻桥。
0：调节器电阻桥已禁止（COMP_CFGR1 寄存器的 BRGEN 位也为低电平时）
1：使能调节器电阻桥
如果 SCALEN 置 1 且 BRGEN 复位，则全部四个调节器输出都会提供相同值 V_{REF_COMP}（近似于 V_{REFINT}）。
如果 SCALEN 和 BRGEN 均置 1，则四个调节器输出会分别提供 V_{REF_COMP}、3/4 V_{REF_COMP}、1/2 V_{REF_COMP} 和 1/4 V_{REF_COMP} 值。

位 0 **EN**：COMP 通道 2 使能位 (COMP channel 2 enable bit)
此位由软件置 1 和清零（仅限 LOCK 未置 1 的情况）。它使能 COMP 通道 2。
0：禁止
1：使能

28.7.6 COMP寄存器映射

下表对比较器寄存器进行了汇总。

表 223. COMP 寄存器映射和复位值

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	**COMP_SR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	C2IF	C1IF	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	C2VAL	C1VAL
	Reset value															0	0															0	0
0x04	**COMP_ICFR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CC2IF	CC1IF	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	Reset value															0	0																
0x08	**COMP_OR (OR_CFG=0)**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OR15	OR14	OR13	OR12	OR11	AFOP										
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x08	**COMP_OR (OR_CFG=1)**	OR31	OR30	OR29	OR28	OR27	OR26	OR25	OR24	OR23	OR22	OR21	OR20	OR19	OR18	OR17	OR16	OR15	OR14	OR13	OR12	OR11	AFOP										
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0C	**COMP_CFG R1**	LOCK	Res.	Res.	Res.	BLANKING				Res.	Res.	Res.	INPSEL	Res.	INMSEL			Res.	Res.	PWRMODE		Res.	Res.	HYST		Res.	ITEN	Res.	Res.	POLARITY	SCALEN	BRGEN	EN
	Reset value	0				0	0	0	0				0		0	0	0			0	0			0	0		0			0	0	0	0
0x10	**COMP_CFG R2**	LOCK	Res.	Res.	Res.	BLANKING				Res.	Res.	Res.	INPSEL	Res.	INMSEL			Res.	Res.	PWRMODE		Res.	Res.	HYST		Res.	ITEN	Res.	WINMODE	POLARITY	SCALEN	BRGEN	EN
	Reset value	0				0	0	0	0				0		0	0	0			0	0			0	0		0		0	0	0	0	0

有关寄存器边界地址的信息，请参见*第 2.2.2 节：存储器映射和寄存器边界地址*。

29 运算放大器（OPAMP）

29.1 简介

器件内置两个运算放大器，每个运算放大器具有两个输入和一个输出。这三个 I/O 可连接到外部引脚，因此能实现任意类型的外部互连。可以将（这两个）运算放大器在内部配置为一个跟随器和一个放大器，放大器具有从 2 到 16 的非反相增益或从 -1 到 -15 的反相增益。

正输入可连接到内部 DAC。

其中一个输出可连接到内部 ADC。

29.2 OPAMP 主要特性

- 轨到轨输入和输出电压范围
- 低输入偏置电流（低至 1 nA）
- 低输入失调电压（校准后为 1.5 mV，出厂校准时为 10 mV）
- 7 MHz 增益带宽积
- 高速模式提供更高的压摆率

注： *有关详细的 OPAMP 特性，请参见产品数据手册。*

29.3 OPAMP 功能说明

OPAMP 有多种模式。

每个 OPAMP 均可单独使能，如果被禁止，则输出呈高阻态。

OPAMP 使能后，其既可以处于校准模式，校准模式下 OPAMP 的所有输入都断开，也可以处于功能模式。

功能模式有两种，高速模式和正常模式。在功能模式下，OPAMP 的输入和输出的连接如*第 29.3.3 节：信号走线*中所述。

29.3.1 OPAMP 复位和时钟

访问寄存器时，需要运算放大器时钟。如果应用程序无需对这些寄存器进行读或写访问，则可使用外设时钟使能寄存器关闭此时钟（请参见*第 8.7.43 节：RCC APB1 时钟寄存器 (RCC_APB1LENR)* 中的 OPAMPEN 位）。

OPAEN 位使能和禁止 OPAMP 工作。应先更改 OPAMP 寄存器配置，然后再使能 OPAEN 位，以避免产生错误的输出。

如果不再需要运算放大器的输出，则可禁止运算放大器以节省功率。禁止 OPAMP 时，之前设置的所有配置（包括校准）保持。

29.3.2 初始配置

运算放大器的默认配置为功能模式，在该模式下，三个输入/输出连接到外部引脚。在默认模式下，运算放大器使用出厂微调值进行偏移校准。有关出厂微调的条件，请参见数据手册的电气特性部分，通常温度为 30 °C，电压为 3 V。可对微调值进行调节，有关更改微调值的信息，请参见*第 29.3.5 节：校准*。默认配置使用正常模式，该模式提供标准性能。可以将 OPAHSM 位置 1，从而将运算放大器切换至高速模式，以达到更高的压摆率。数据手册的*电气特性*部分对正常和高速模式特性进行了定义。

一旦 OPAMPx_CSR 寄存器中的 OPAEN 位置 1，运算放大器即可工作。两个输入引脚和一个输出引脚可按照*第 29.3.3 节：信号走线*中的定义进行连接，并且可对默认连接设置进行更改。

注： *必须在相应 GPIOx_MODER 寄存器中将输入和输出引脚配置为模拟模式（默认状态）。*

29.3.3 信号走线

运算放大器引脚走线由 OPAMPx_CSR 寄存器确定。

下表介绍了两个运算放大器（OPAMP1 和 OPAMP2）的引脚连接。

表 224. 运算放大器连接情形

信号	引脚	Internal	注释
OPAMP1_VINM	PC5(INM0) PA7(INM1)	ADC1_IN8 ADC2_IN8 OPAMP1_VOUT 或 PGA	通过 PGA_GAIN 和 VM_SEL 位控制。
OPAMP1_VINP	PB0	dac_out1 ADC1_IN9 ADC2_IN9 COMP1_INP	通过 VP_SEL 位控制。
OPAMP1_VOUT	PC4	ADC1_IN4 ADC2_IN4 COMP1_INM7	在 OPAMP 使能时连接该引脚。ADC 输入通过 ADC 控制。
OPAMP2_VINM	PE8(INM0) PG1(INM1)	OPAMP2_VOUT 或 PGA	通过 PGA_GAIN 和 VM_SEL 位控制。
OPAMP2_VINP	PE9	dac_out2 COMP2_INP	通过 VP_SEL 位控制
OPAMP2_VOUT	PE7	COMP2_INM7	-

29.3.4 OPAMP 模式

运算放大器的输入和输出都可在引脚上访问。放大器可用于多种配置环境：

- 独立模式（外部增益设置模式）
- 跟随器配置模式
- PGA模式

注： *放大器输出引脚直接连接到输出引脚的焊盘，以最大限度地降低输出阻抗。如果使能了放大器，则它的输出引脚不能用作通用 I/O，即使放大器被配置为一个 PGA 并仅连接到内部通道也不行。*

输入信号的阻抗必须保持在不能使输入泄漏产生明显伪信号（由于源极阻降而引起）的水平以下。有关更多详细信息，请参见数据手册的电气特性部分。

独立模式（外部增益设置模式）

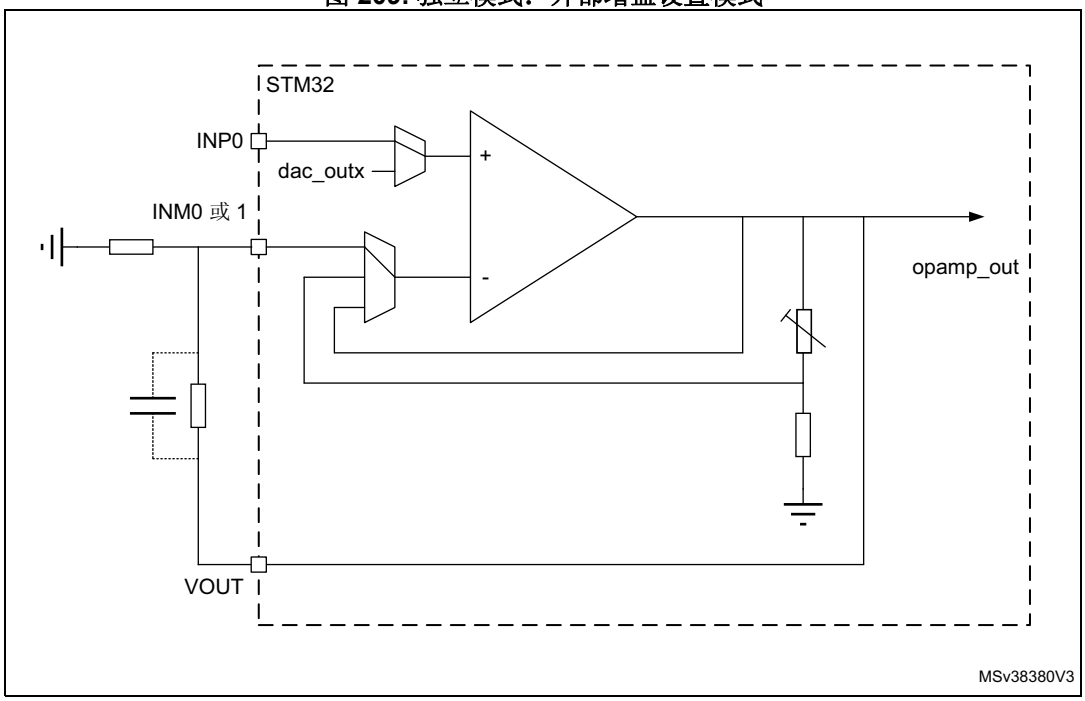
下面介绍在独立模式下使用 OPAMP 的程序。

假设 OPAMPx_CSR 配置为默认值，GPIOx_MODER 配置为默认状态。只要 OPAEN 位置 1，两个输入引脚和一个输出引脚都连接到运算放大器。

此默认配置使用工厂微调值，并在正常模式下运行（达到最高性能）。可按如下方式更改 OPAMP 的行为：

- 可以将 OPAHSM 设置为“运算放大器高速”模式，以达到高压摆率。
- 可以将 USERTRIM 置 1，以修改输入偏移的微调值。

图 209. 独立模式：外部增益设置模式



跟随器配置模式

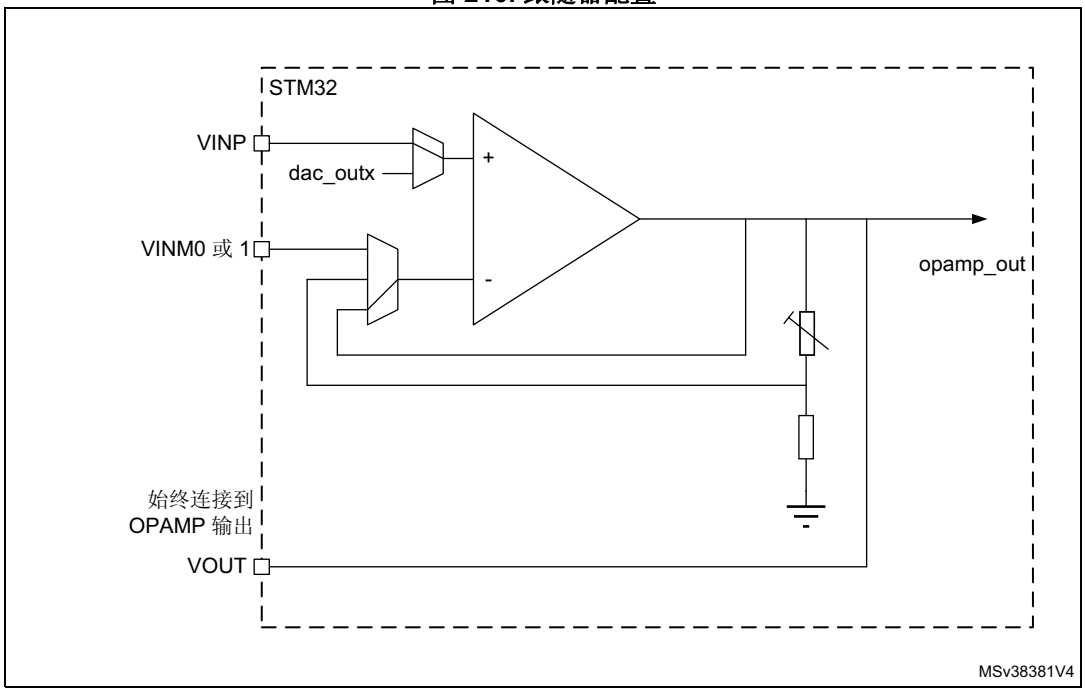
下面介绍在跟随器模式下使用 OPAMP 的程序。

- 将 VM_SEL 位配置为“opamp_out 连接到 OPAMPx_VINM 输入”，11
- 将 VP_SEL 位配置为“GPIO 连接到 OPAMPx_VINP”，00
- 一旦 OPAEN 位置 1，OPAMPx_VINP 引脚上的电压就会被缓冲到 OPAMPx_VOUT 引脚。

注： *对应于 OPAMPx_VINM 的引脚可供另外使用。*

OPAMP1 输出上的信号也视为 ADC 输入。因此，如果输入信号频率与运算放大器增益带宽积规范相符，则配置为跟随器模式的 OPAMP 可用于对输入信号进行阻抗调整，然后再将这些信号馈送至 ADC 输入。

图 210. 跟随器配置



可编程增益放大器模式

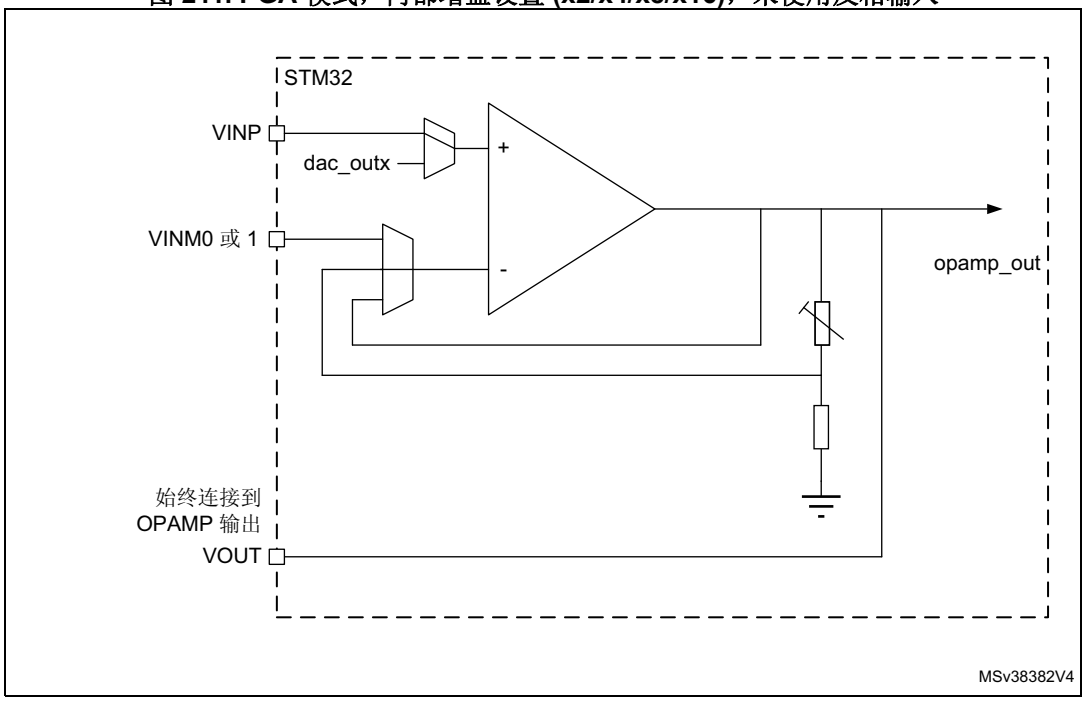
下面介绍将 OPAMP 用作可编程增益放大器的步骤。

- 将 VM_SEL 位配置为“反馈电阻连接到 OPAMPx_VINM 输入”，10
- 将 PGA_GAIN 位配置为“内部增益 2、4、8 或 16”，0000 到 0011
- 将 VP_SEL 位配置为“GPIO 连接到 OPAMPx_VINP”，00

  一旦 OPAEN 位置 1，OPAMPx_VINP 引脚上的电压就会放大所选增益的倍数，并可在 OPAMPx_VOUT 引脚上测得放大后的电压。

注： *为避免饱和，输入电压应保持低于：V_{DDA} 除以所选增益。*

图 211. PGA 模式，内部增益设置 (x2/x4/x8/x16)，未使用反相输入



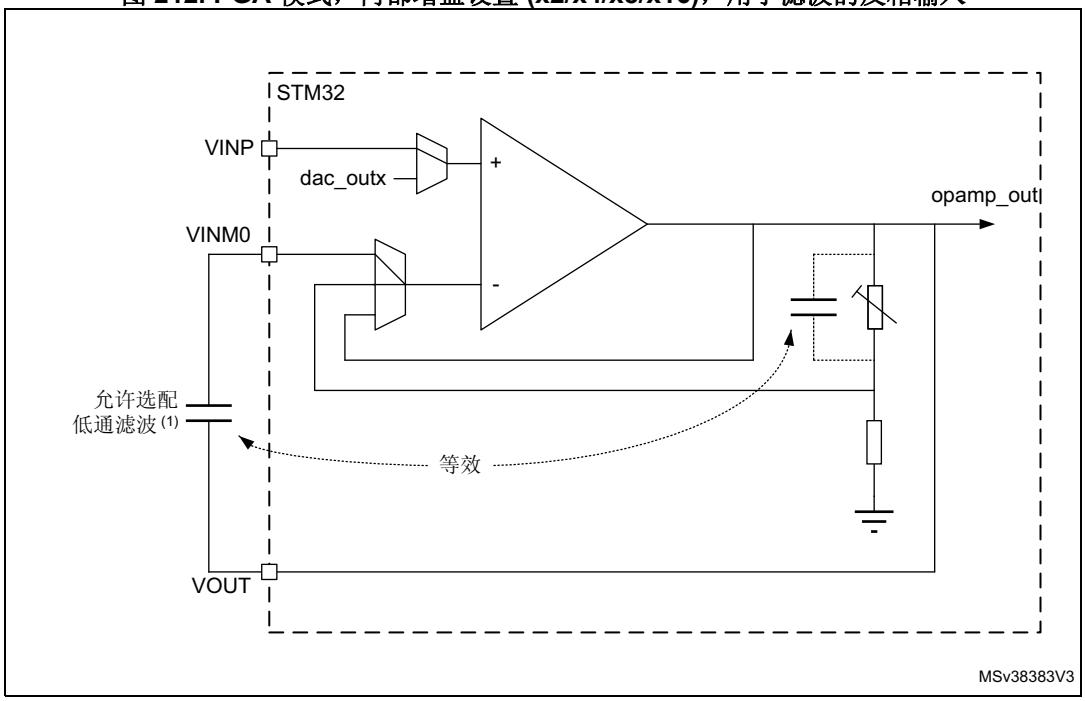
使用外部滤波的可编程增益放大器模式

下面介绍使用 OPAMP 和外部滤波来放大输入信号幅值的步骤。

- 将 VM_SEL 位配置为“反馈电阻连接到 OPAMPx_VINM 输入”，10
- 将 PGA_GAIN 位配置为“内部增益 2、4、8 或 16，在 INM0 上使用滤波”，0100 到 0111
- 将 VP_SEL 位配置为“GPIO 连接到 OPAMPx_VINP”。

  INM 上的任何外部连接均可与内部 PGA 并联使用，例如，可在 opamp_out 和 INM 之间连接一个电容来实现滤波目的（有关 PGA 电阻网络中使用的电阻值，请参见数据手册）。

图 212. PGA 模式，内部增益设置 (x2/x4/x8/x16)，用于滤波的反相输入



1. 增益取决于截止频率。

可编程增益放大器，使用外部偏压的非反相模式或反相模式

下面介绍使用 OPAMP 和偏置电压（针对非反相模式，反相模式无需偏置电压）来放大输入信号幅值的步骤。

- 将 VM_SEL 位配置为“反馈电阻连接到 OPAMPx_VINM 输入”，10
- 将 PGA_GAIN 位配置为“反相增益 = -1、-3、-7、-15/ 非反相增益 = 2、4、8、16，使用 INM0）”，1000 到 1011
- 将 VP_SEL 位配置为“GPIO 连接到 OPAMPx_VINP”。

图 213. PGA 模式，非反相增益设置 (x2/x4/x8/x16) 或反相增益设置 (x-1/x-3/x-7/x-15)

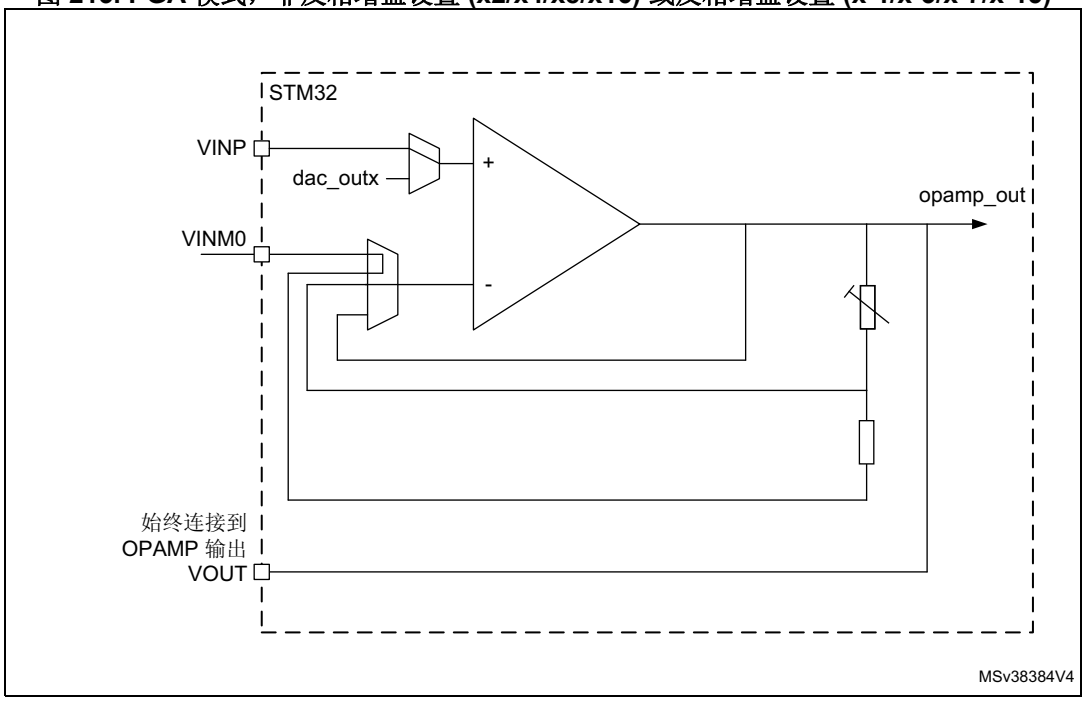
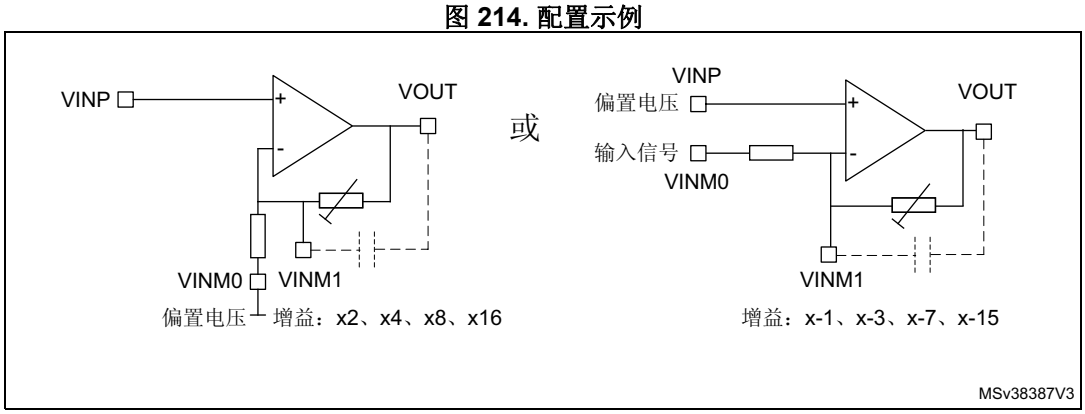


图 214. 配置示例



可编程增益放大器，使用外部偏压的非反相模式或使用滤波的反相模式

下面介绍使用 OPAMP 和偏置电压（针对非反相模式）或滤波（针对反相模式）来放大输入信号幅值的步骤。

- 将 VM_SEL 位配置为“反馈电阻连接到 OPAMPx_VINM 输入”，10
- 将 PGA_GAIN 位配置为“反相增益 = -1、-3、-7、-15/ 非反相增益 = 2、4、8、16，使用 INM0 和用于滤波的 INM1 节点”，1100 到 1111
- 将 VP_SEL 位配置为“GPIO 连接到 OPAMPx_VINP”。

  VM1 上的任何外部连接均可与内部 PGA 并联使用，例如，可在 opamp_out 和 VM1 之间连接一个电容来实现滤波目的（有关 PGA 电阻网络中使用的电阻值，请参见数据手册）。

图 215. PGA 模式，非反相增益设置 (x2/x4/x8/x16)
或使用滤波的反相增益设置 (x-1/x-3/x-7/x-15)

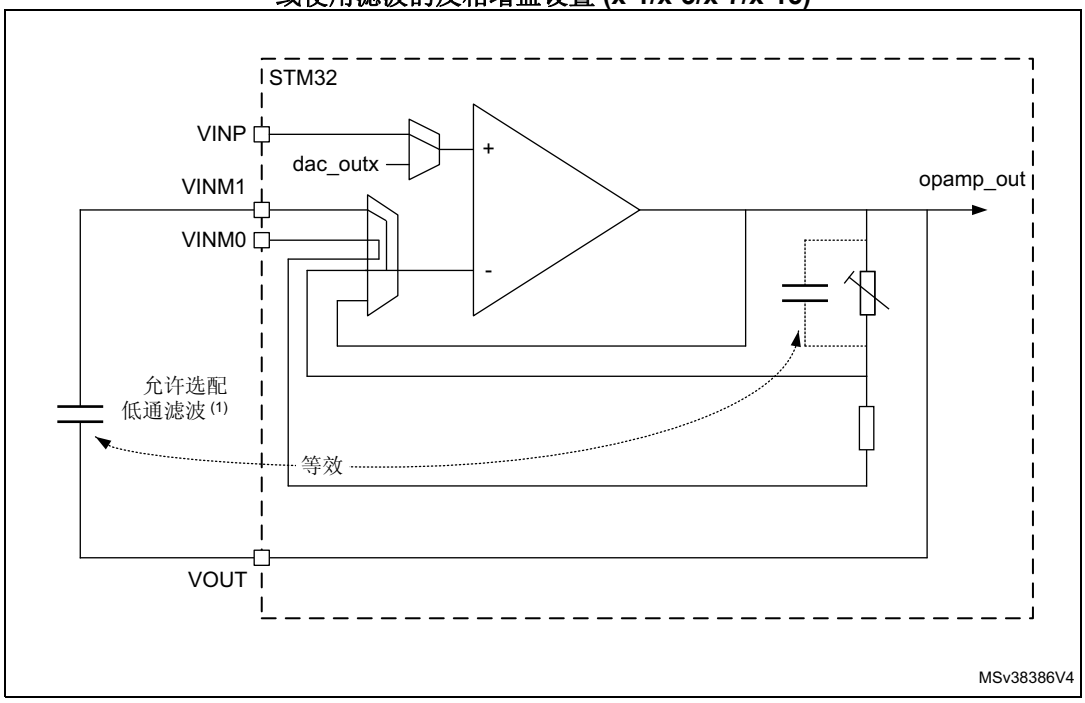
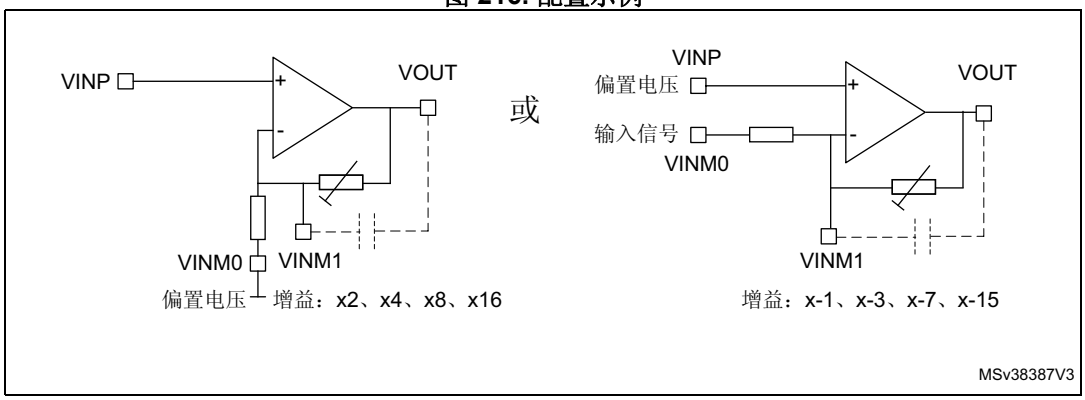


图 216. 配置示例



29.3.5 校准

OPAMP 接口连续将调整的偏移值发送至运算放大器。启动时，使用预设的“出厂”微调值初始化了微调值。

每个运算放大器均可由用户进行微调。对于正常模式和高速模式，特定寄存器允许有不同的微调值。

校准的目的在于尽可能地减小 OPAMP 输入失调电压。在稳定的电压和温度条件下，校准电路可将输入失调电压减至 +/-1.5 mV 以下。

对于每个运算放大器和每种模式，需要对两个微调值进行微调，一个值用于 NMOS 差分对，另一个值用于 PMOS 差分对。

对于每个运算放大器，有两个寄存器可用于微调偏移值，一个用于正常模式 (OPAMPx_OTR)，另一个用于高速模式 (OPAMPx_HSOTR)。每个寄存器都包含 5 bits 用于 P 差分对微调和 5 bits 用于 N 差分对微调。这些均为“用户”微调的值。

用户可使用 OPAMPx_CSR 寄存器的 USERTRIM 位从“出厂”值切换到“用户”微调的值。该位在启动时复位，因此默认情况下，“出厂”值被应用于 OPAMP 选项寄存器。

用户可在校准或功能模式下轻松更改微调值。

通常在将 CALON 位设置为 1 来初始化校准操作之后，才配置偏移微调寄存器。当 CALON = 1 时，运算放大器的输入从工作环境中断开。

- 将 CALSEL 置为 01 可初始化 P 差分对的偏移校准（使用低参考电压）。
- 将 CALSEL 复位为 11 可初始化 N 差分对的偏移校准（使用高参考电压）。

当 CALON = 1 时，CALOUT 位将反映 CALSEL 和 OPAHSM 所选择的微调值的影响。当我们选择 CALSEL=11 且 OPAHSM=0 时，软件应将 OPAMP 控制寄存器中的 TRIMOFFSETN 位从 0x00 增至第一个导致 OPAMP 寄存器的 CALOUT 位从 1 变为 0 的值。如果 CALOUT 位复位，则偏移已正确校准，并且必须存储相应的微调值。在更改微调值之后，CALOUT 标志最多需要 1 ms 的时间达到稳定状态（请参见数据手册电气特性部分的 $t_{OFFTRIM}max$ 延迟规范）。

注： *微调值越接近于最佳微调值，稳定所用的时间就越长（但在任何情况下，最长稳定时间都低于 1 ms）。*

表 225. 工作模式和校准

模式	控制位				输出	
	OPAEN	OPAHSM	CALON	CALSEL	V_{OUT}	CALOUT 标志
正常工作模式	1	0	0	X	模拟	0
高速模式	1	1	0	X	模拟	0
掉电	0	X	X	X	Z	0
正常模式的偏移校准 N 差分	1	0	1	11	模拟	X
正常模式的偏移校准 P 差分	1	0	1	01	模拟	X
高速模式的偏移校准 N 差分	1	1	1	11	模拟	X
高速模式的偏移校准 P 差分	1	1	1	01	模拟	X

校准程序

以下为对任一运算放大器执行完全校准的步骤：

1. 将 OPAMPx_CSR 的 OPAEN 位置 1，以使能运算放大器。
2. 将 OPAMPx_CSR 寄存器中的 USERTRIM 位置 1。
3. 选择一种校准模式（请参见*表 225：工作模式和校准*）。步骤 3 到 4 必须重复 4 次。首先我们选择
   – 正常模式和 N 差分对
   上述校准模式对应于 OPAMPx_CSR 寄存器中的 OPAHSM = 0，CALSEL = 11。
4. 从 00000b 开始递增 OPAMPx_OTR 的 TRIMOFFSETN[4:0]，直至 OPAMPx_CSR 的 CALOUT 变为 0。

注： *在写入 OPAMPx_OTR 寄存器和读取 CALOUT 值之间，确保等待数据手册电气特性部分中指定的 $t_{OFFTRIM}max$ 延迟，以获得正确的 CALOUT 值。*

CALOUT 反转即表示，偏移已正确补偿，必须在 OPAMPx_OTR 寄存器中保存相应的微调值。

针对以下模式重复步骤 3 到 4：

- 正常模式和 P 差分对，CALSEL = 01
- 高速模式和 N 差分对
- 高速模式和 P 差分对

如果未使用某种模式，则无需执行相应的校准。

所有运算放大器均可同时校准。

注： *在整个校准阶段，运算放大器输出的外部连接不能上拉或下拉高于 500 µA 的电流。*

29.4 OPAMP 低功耗模式

表 226. 低功耗模式对 OPAMP 的影响

模式	说明
休眠	无影响。
D2（停止）	无影响，OPAMP 寄存器内容保持不变。
待机	OPAMP 寄存器掉电，退出待机模式后必须重新初始化。

29.5 OPAMP PGA 增益

将 OPAMP 配置为 PGA 模式时，可对非反相模式选择增益 x2、x4、x8 或 x16，对反相模式选择增益 x-1、x-3、x-7 或 x-15。

将 OPAMP 配置为非反相模式时，可通过产品数据手册获取增益误差。将其配置为反相模式时，增益系数不仅由片上反馈电阻决定，而且与信号源阻抗有关。如果相比于 PGA 的输入反馈阻抗，信号源输出阻抗不可忽略不计，则会产生增益误差。请参见产品数据手册中的 PGA 阻抗值。

29.6 OPAMP 寄存器

29.6.1 OPAMP1 控制/状态寄存器 (OPAMP1_CSR)

OPAMP1 control/status register

地址：0x00

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	CAL OUT	TST REF	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	USER TRIM	PGA_GAIN	
	r	rw											rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PGA_GAIN		CALSEL		CALON	Res.	Res.	OPA HSM	Res.	VM_SEL		Res.	VP_SEL		FORCE _VP	OPAEN
rw	rw	rw	rw	rw			rw		rw	rw		rw	rw	rw	rw

位 31 保留，必须保持复位值。

位 30 **CALOUT**：运算放大器校准输出 (Operational amplifier calibration output)
OPAMP 输出状态标志。在校准模式下，OPAMP 用作比较器。
0：非反相 < 反相
1：非反相 > 反相

位 29 **TSTREF**：OPAMP 校准参考电压输出控制 (OPAMP calibration reference voltage output control)
（为测试保留）
0：未输出 OPAMP 的 INTVREF
1：输出 OPAMP 的 INTVREF

位 28:19 保留，必须保持复位值。

位 18 **USERTRIM**：用户微调使能 (User trimming enable)
该位用于从“出厂”AOP 偏移微调值切换至“用户”AOP 偏移微调值。
对于正常模式和高速模式，该位均有效。
0：使用“出厂”微调代码
1：使用“用户”微调代码

位 17:14 **PGA_GAIN**：运算放大器可编程放大器增益值 (Operational amplifier Programmable amplifier gain value)

0000：非反相内部增益 2，参考 VREF
0001：非反相内部增益 4，参考 VREF
0010：非反相内部增益 8，参考 VREF
0011：非反相内部增益 16，参考 VREF
0100：非反相内部增益 2，在 INM0 上使用滤波，参考 VREF
0101：非反相内部增益 4，在 INM0 上使用滤波，参考 VREF
0110：非反相内部增益 8，在 INM0 上使用滤波，参考 VREF
0111：非反相内部增益 16，在 INM0 上使用滤波，参考 VREF
1000：反相增益 = -1/同相增益 = 2，INM0 节点用于输入或偏置
1001：反相增益 = -3/同相增益 = 4，INM0 节点用于输入或偏置
1010：反相增益 = -7/同相增益 = 8，INM0 节点用于输入或偏置
1011：反相增益 = -15/同相增益 = 16，INM0 节点用于输入或偏置
1100：反相增益 = -1/同相增益 = 2，INM0 节点用于输入或偏置，INM1 节点用于滤波
1101：反相增益 = -3/同相增益 = 4，INM0 节点用于输入或偏置，INM1 节点用于滤波
1110：反相增益 = -7/同相增益 = 8，INM0 节点用于输入或偏置，INM1 节点用于滤波
1111：反相增益 = -15/同相增益 = 16，INM0 节点用于输入或偏置，INM1 节点用于滤波

位 13:12 **CALSEL**：校准选择 (Calibration selection)

用于选择 CALON = 1 或 FORCE_VP = 1 时生成内部参考电压所用的偏移校准总线。

00：对 OPAMP 输入应用 0.033*VDDA
01：对 OPAMP 输入应用 0.1*VDDA（用于 PMOS 校准）
10：对 OPAMP 输入应用 0.5*VDDA
11：对 OPAMP 输入应用 0.9*VDDA（用于 NMOS 校准）

位 11 **CALON**：校准模式使能 (Calibration mode enabled)

0：正常模式
1：校准模式（通过硬件打开所有开关）

位 10:9 保留，必须保持复位值。

位 8 **OPAHSM**：运算放大器高速模式 (Operational amplifier high-speed mode)

要更改此配置，必须禁止运算放大器。

0：运算放大器处于正常模式
1：运算放大器处于高速模式

位 7 保留，必须保持复位值。

位 6:5 **VM_SEL**：反相输入选择 (Inverting input selection)

只有在 OPAMODE = 00、01、10 或 11 时才会使用这些位。

00：INM0 连接到 OPAMP INM 输入
01：INM1 连接到 OPAMP INM 输入
10：反馈电阻连接到 OPAMP INM 输入（PGA 模式），反相输入选择取决于 PGA_GAIN 设置
11：opamp_out 连接到 OPAMP INM 输入（跟随器模式）

位 4 保留，必须保持复位值。

位 3:2 **VP_SEL**：非反相输入选择 (Non inverted input selection)
00：GPIO 连接到 OPAMPx_VINP
01：dac_outx 连接到 OPAMPx_VINP
10：保留
11：保留

位 1 **FORCE_VP**：强制内部 VP 参考 (Force internal reference on VP)（为测试保留）
0：正常工作模式。非反相输入连接到输入。
1：校准验证模式：非反相输入连接到校准参考电压。

位 0 **OPAEN**：运算放大器使能 (Operational amplifier Enable)
0：运算放大器已禁止
1：运算放大器已使能

29.6.2 正常模式下的 OPAMP1 微调寄存器 (OPAMP1_OTR)

OPAMP1 trimming register in normal mode

地址：0x04

复位值：0x0000 XXXX（出厂微调值）

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

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

位 31:13 保留，必须保持复位值。

位 12:8 **TRIMOFFSETP[4:0]**：PMOS 差分对的微调 (Trim for PMOS differential pairs)

位 7:5 保留，必须保持复位值。

位 4:0 **TRIMOFFSETN[4:0]**：NMOS 差分对的微调 (Trim for NMOS differential pairs)

29.6.3 高速模式下的 OPAMP1 微调寄存器 (OPAMP1_HSOTR)

OPAMP1 trimming register in high-speed mode

地址：0x08

复位值：0x0000 XXXX（出厂微调值）

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

位 31:13 保留，必须保持复位值。

位 12:8 **TRIMHSOFFSETP[4:0]**：PMOS 差分对的高速模式微调 (High-speed mode trim for PMOS differential pairs)

位 7:5 保留，必须保持复位值。

位 4:0 **TRIMHSOFFSETN[4:0]**：NMOS 差分对的高速模式微调 (High-speed mode trim for NMOS differential pairs)

29.6.4 OPAMP 选项寄存器(OPAMP_OR)

OPAMP option register

地址：0x0C

复位值：0x0000 0000

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

位 31:0 保留，必须保持复位值。

29.6.5 OPAMP2 控制/状态寄存器 (OPAMP2_CSR)

OPAMP2 control/status register

地址：0x10

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	CAL OUT	TST REF	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	USER TRIM	PGA_GAIN	
	r	rw											rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PGA_GAIN		CALSEL		CALON	OPA MODE		OPA HSM	Res.	VM_SEL		Res.	VP_SEL		FORCE _VP	OPAEN
rw	rw	rw	rw	rw	rw	rw	rw		rw	rw		rw	rw	rw	rw

位 31 保留，必须保持复位值。

位 30 **CALOUT**：运算放大器校准输出 (Operational amplifier calibration output)
OPAMP 输出状态标志。在校准模式下，OPAMP 用作比较器。
0：非反相 < 反相
1：非反相 > 反相

位 29 **TSTREF**：OPAMP 校准参考电压输出控制 (OPAMP calibration reference voltage output control)（为测试保留）
0：未输出 OPAMP 的 INTVREF
1：输出 OPAMP 的 INTVREF

位 28:19 保留，必须保持复位值。

位 18 **USERTRIM**：用户微调使能 (User trimming enable)
该位用于从“出厂”AOP 偏移微调值切换至“用户”AOP 偏移微调值。
对于正常模式和高速模式，该位均有效。
0：使用“出厂”微调代码
1：使用“用户”微调代码

位 17:14 **PGA_GAIN**：运算放大器可编程放大器增益值 (Operational amplifier Programmable amplifier gain value)
0000：非反相内部增益 2，参考 VREF
0001：非反相内部增益 4，参考 VREF
0010：非反相内部增益 8，参考 VREF
0011：非反相内部增益 16，参考 VREF
0100：非反相内部增益 2，在 INM0 上使用滤波，参考 VREF
0101：非反相内部增益 4，在 INM0 上使用滤波，参考 VREF
0110：非反相内部增益 8，在 INM0 上使用滤波，参考 VREF
0111：非反相内部增益 16，在 INM0 上使用滤波，参考 VREF
1000：反相增益 = -1/同相增益 = 2，INM0 节点用于输入或偏置
1001：反相增益 = -3/同相增益 = 4，INM0 节点用于输入或偏置
1010：反相增益 = -7/同相增益 = 8，INM0 节点用于输入或偏置
1011：反相增益 = -15/同相增益 = 16，INM0 节点用于输入或偏置
1100：反相增益 = -1/同相增益 = 2，INM0 节点用于输入或偏置，INM1 节点用于滤波
1101：反相增益 = -3/同相增益 = 4，INM0 节点用于输入或偏置，INM1 节点用于滤波
1110：反相增益 = -7/同相增益 = 8，INM0 节点用于输入或偏置，INM1 节点用于滤波
1111：反相增益 = -15/同相增益 = 16，INM0 节点用于输入或偏置，INM1 节点用于滤波

位 13:12 **CALSEL**：校准选择 (Calibration selection)
用于选择 CALON = 1 或 FORCE_VP = 1 时生成内部参考电压所用的偏移校准总线。
00：对 OPAMP 输入应用 0.033*VDDA
01：对 OPAMP 输入应用 0.1*VDDA（用于 PMOS 校准）
10：对 OPAMP 输入应用 0.5*VDDA
11：对 OPAMP 输入应用 0.9*VDDA（用于 NMOS 校准）

位 11 **CALON**：校准模式使能 (Calibration mode enabled)
0：正常模式
1：校准模式（通过硬件打开所有开关）

位 10:9 保留，必须保持复位值。

位 8 **OPAHSM**：运算放大器高速模式 (Operational amplifier high-speed mode)
要更改此配置，必须禁止运算放大器。
0：运算放大器处于正常模式
1：运算放大器处于高速模式

位 7 保留，必须保持复位值。

位 6:5 **VM_SEL**：反相输入选择 (Inverting input selection)
只有在 OPAMODE = 00、01、10 或 11 时才会使用这些位。
00：INM0 连接到 OPAMP INM 输入
01：INM1 连接到 OPAMP INM 输入
10：反馈电阻连接到 OPAMP INM 输入（PGA 模式），反相输入选择取决于 PGA_GAIN 设置
11：opamp_out 连接到 OPAMP INM 输入（跟随器模式）

位 4 保留，必须保持复位值。

位 3:2 **VP_SEL**：非反相输入选择 (Non inverted input selection)
00：GPIO 连接到 OPAMPx_VINP
01：DAC 连接到 OPAMPx_VINP
10：保留
11：保留

位 1 **FORCE_VP**：强制内部 VP 参考 (Force internal reference on VP)（为测试保留）
0：正常工作模式。非反相输入连接到输入。
1：校准验证模式：非反相输入连接到校准参考电压。

位 0 **OPAEN**：运算放大器使能 (Operational amplifier Enable)
0：运算放大器已禁止
1：运算放大器已使能

29.6.6 正常模式下的 OPAMP2 微调寄存器 (OPAMP2_OTR)

OPAMP2 trimming register in normal mode

地址：0x14

复位值：0x0000 XXXX（出厂微调值）

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

位 31:13 保留，必须保持复位值。

位 12:8 **TRIMOFFSETP[4:0]**：PMOS 差分对的微调 (Trim for PMOS differential pairs)

位 7:5 保留，必须保持复位值。

位 4:0 **TRIMOFFSETN[4:0]**：NMOS 差分对的微调 (Trim for NMOS differential pairs)

29.6.7 高速模式下的 OPAMP2 微调寄存器(OPAMP2_HSOTR)

OPAMP2 trimming register in high-speed mode

地址：0x18

复位值：0x0000 XXXX（出厂微调值）

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

位 31:13 保留，必须保持复位值。

位 12:8 **TRIMHSOFFSETP[4:0]**：PMOS 差分对的高速模式微调 (High-speed mode trim for PMOS differential pairs)

位 7:5 保留，必须保持复位值。

位 4:0 **TRIMHSOFFSETN[4:0]**：NMOS 差分对的高速模式微调 (High-speed mode trim for NMOS differential pairs)

29.6.8 OPAMP寄存器映射

表 227. OPAMP 寄存器映射和复位值

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	**OPAMP1_CSR**	Res.	CALOUT	TSTREF	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	USERTRIM	PGA_GAIN				CALSEL		CALON	Res.	Res.	OPAHSM	Res.	VM_SEL		Res.	VP_SEL		FORCE_VP	OPAEN
	Reset value		0	0											0	0	0	0	0	0	0	0			0		0	0		0	0	0	0
0x04	**OPAMP1_OTR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TRIM OFFSETP[4:0]					Res.	Res.	Res.	TRIM OFFSETN[4:0]				
	Reset value																				(1)								(1)				
0x08	**OPAMP1_ HSOTR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TRIMHS OFFSETP[4:0]					Res.	Res.	Res.	TRIMHS OFFSETN[4:0]				
	Reset value																				(1)								(1)				
0x0C	**OPAMP_OR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	Reset value																																
0x10	**OPAMP2_CSR**	Res.	CALOUT	TSTREF	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	USERTRIM	PGA_GAIN				CALSEL		CALON	Res.	Res.	OPAHSM	Res.	VM_SEL		Res.	VP_SEL		FORCE_VP	OPAEN
	Reset value		0	0											0	0	0	0	0	0	0	0			0		0	0		0	0	0	0
0x14	**OPAMP2_OTR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TRIM OFFSETP[4:0]					Res.	Res.	Res.	TRIM OFFSETN[4:0]				
	Reset value																				(1)								(1)				
0x18	**OPAMP2_ HSOTR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TRIMHS OFFSETP[4:0]					Res.	Res.	Res.	TRIMHS OFFSETN[4:0]				
	Reset value																				(1)								(1)				

1. 出厂微调值。

有关寄存器边界地址的信息，请参见*第 2.2.2 节：存储器映射和寄存器边界地址*。

30 数字滤波器，用于 ΣΔ 调制器（DFSDM）

30.1 简介

ΣΔ 调制器数字滤波器 (DFSDM) 是一种高性能模块，专用于将外部 ΣΔ 调制器连接到微控制器。它具有多达 8 个外部数字串行接口（通道）和多达 4 个数字滤波器，具有灵活的 ΣΔ 流数字处理选项，可提供高达 24 位 ADC 最终分辨率。DFSDM 还特有并行数据流输入，输入可选择来自内部 ADC 外设或来自微控制器存储器。

外部 ΣΔ 调制器将其模拟输入的模拟值转换为数字数据流。此数字数据流通过串行接口发送到 DFSDM 输入通道。DFSDM 支持多种标准来连接各种 ΣΔ 调制器输出：SPI 接口和曼彻斯特编码单线接口（都具有可调参数）。DFSDM 模块支持连接多达 8 个复用输入数字串行通道，这些通道可与多达 4 个 DFSDM 模块共享。DFSDM 模块还支持来自多达 8 个内部 16 位数据通道（可选择来自内部 ADC 或来自微控制器存储器）的并行数据输入。

DFSDM 将输入数据流转换为最终数字数据字，该数字字表示 ΣΔ 调制器模拟输入上的模拟输入值。该转换基于以下可配置的数字处理而完成：对输入串行数据流进行数字滤波和抽取。

转换速度和分辨率可根据以下可配置的数字处理参数进行调整：滤波器类型、滤波器阶数、滤波器长度和积分器长度。最大输出数据分辨率高达 24 位。有两种转换模式：单次转换模式和连续转换模式。数据可通过 DMA 自动存储在系统 RAM 缓冲区中，因此降低了软件开销。

可使用灵活的定时器触发系统来控制启动 DFSDM 转换。此定时控制可触发即时转换或延时转换，延时时间可以编程。

DFSDM 具有模拟看门狗功能。模拟看门狗可以分配给任何输入通道数据流或最终输出数据。模拟看门狗本身可以对输入数据流进行数字滤波，以达到被监视数据的所需速度和分辨率。

为了检测控制应用中的短路情况，提供了一个短路检测器。该模块监视各个输入通道数据流是否在定义的持续时间内保持稳定（即输入数据流中有若干 0 或 1）。

极值检测器模块监视最终输出数据并存储最大和最小输出数据值。存储的极值可通过软件重新启动。

支持两种功率模式：正常模式和停止模式。

30.2 DFSDM 主要特性

- 多达 8 个复用输入数字串行通道：
  - 可配置的 SPI 接口，用于连接各种 ΣΔ 调制器
  - 支持可配置的曼彻斯特编码单线接口
  - 用于 ΣΔ 调制器的时钟输出
- 来自多达 8 个内部数字并行通道的可选输入：
  - 分辨率高达 16 位的输入
  - 内部源：ADC 数据或存储器（CPU/DMA 写）数据流
- 可调节的数字信号处理：
  - $Sinc^x$ 滤波器：滤波器阶数/类型 (1..5)，过采样率（高达 1..1024）
  - 积分器：过采样率 (1..256)
- 高达 24 位的输出数据分辨率：
  - 最终数据的右移位器（0..31 位）
- 有符号输出数据格式
- 自动数据偏移校正（偏移值由用户存储在寄存器中）
- 连续或单次转换
- 可通过以下途径同步启动转换：
  - 软件触发
  - 内部定时器
  - 外部事件
  - 使用第一个 DFSDM 滤波器 (DFSDM_FLT0) 同步启动转换
- 模拟看门狗功能：
  - 下限和上限数据阈值寄存器
  - 自带可配置的 $Sinc^x$ 数字滤波器（阶数 = 1..3，过采样率 = 1..32）
  - 输入来自输出数据寄存器或来自一个或多个输入数字串行通道
  - 独立于标准转换的连续监视
- 短路检测器，用于检测饱和的模拟输入值（下限和上限）：
  - 高达 8 位计数器，用于检测输入数据流中 1..256 个连续的 0 或 1
  - 连续监视各个通道（8 个串行通道收发器输出）
- 当发生模拟看门狗事件或短路检测器事件时，生成断路
- 极值检测器：
  - 存储最小和最大输出数据值
  - 由软件更新
- DMA 可用于读取转换数据
- 中断：结束转换，溢出，模拟看门狗，短路，通道时钟缺失
- “常规”或“注入”转换：
  - “常规”转换可随时进行请求，即使在连续模式下亦如此，且不会对“注入”转换的时序产生影响

30.3 DFSDM 实现

本部分介绍在 DFSDMx 中实现的配置。

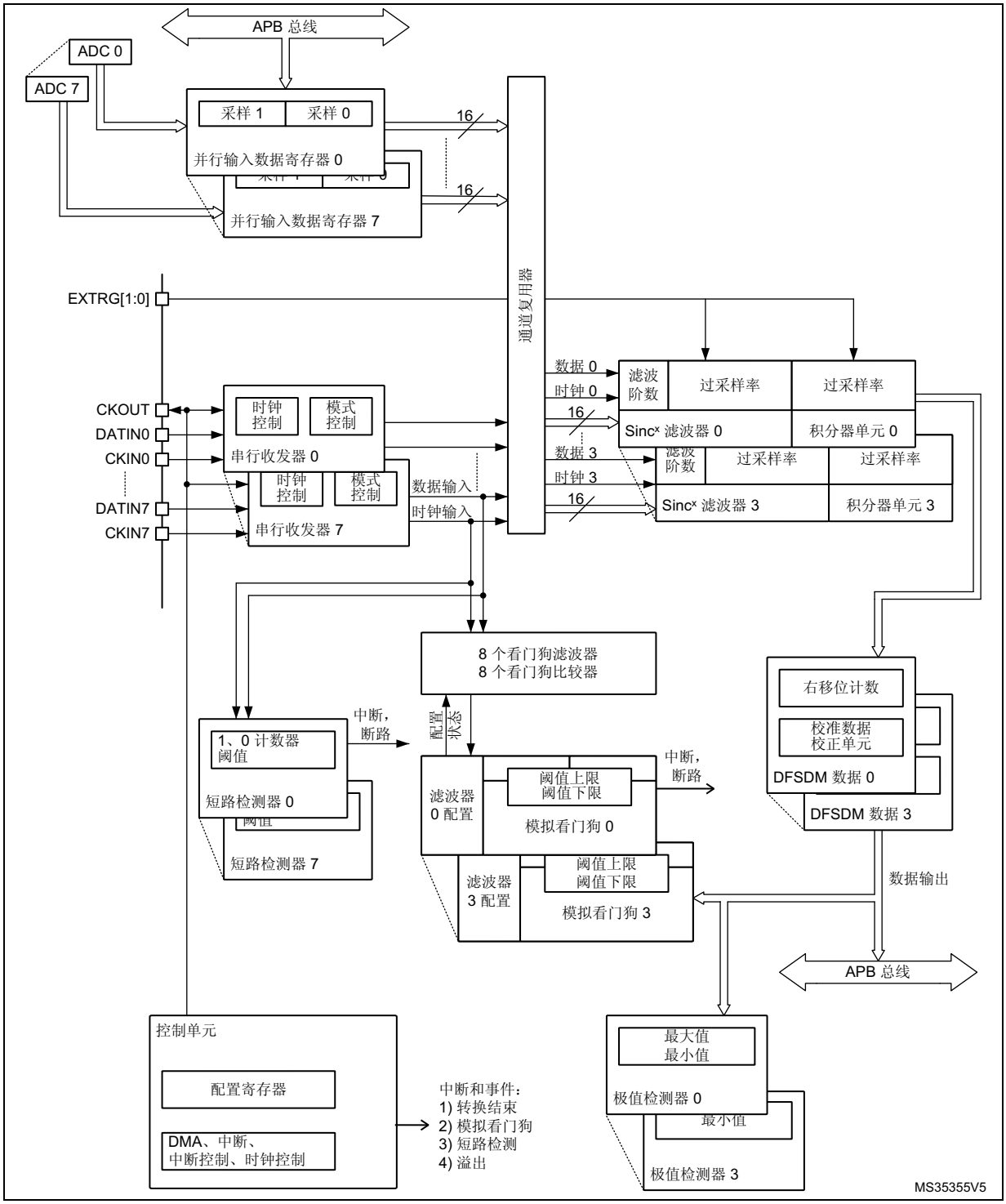
表 228. DFSDM1 实现

DFSDM 功能	DFSDM1
通道数	8
滤波器数	4
来自内部 ADC 的输入	X
支持的触发源	32
脉冲跳过器	-
ID 寄存器支持	-

30.4 DFSDM 功能说明

30.4.1 DFSDM框图

图 217. 单个 DFSDM 框图



1. 本示例显示了 4 个 DFSDM 滤波器和 8 个输入通道（最大 配置）。

30.4.2 DFSDM 引脚和内部信号

表 229. DFSDM 外部引脚

名称	信号类型	注释
VDD	电源	数字电源
VSS	电源	数字接地电源。
CKIN[7:0]	时钟输入	从外部 ΣΔ 调制器提供的时钟信号。FT 输入
DATIN[7:0]	数据输入	从外部 ΣΔ 调制器提供的数据信号。FT 输入
CKOUT	时钟输出	时钟输出，用于为外部 ΣΔ 调制器提供时钟信号。
EXTRG[1:0]	外部触发信号	来自两个 EXTI 信号的输入触发，用于启动模拟转换（来自 GPIO：EXTI11 和 EXTI15）。

表 230. DFSDM 内部信号

名称	信号类型	注释
dfsdm_jtrg[31:0]	内部/外部触发信号	来自内部/外部触发源的输入触发，用于启动模拟转换（来自内部源：同步输入，来自外部源：使用时钟同步的异步输入）。详细信息，请参见*表 231*。
dfsdm_break[3:0]	断路信号输出	由模拟看门狗或短路检测器触发的断路信号事件
dfsdm_dma[3:0]	DMA 请求信号	来自各个 DFSDM_FLTx (x=0..3) 的 DMA 请求信号：结束注入转换事件。
dfsdm_it[3:0]	中断请求信号	各个 DFSDM_FLTx (x=0..3) 的中断信号
dfsdm_dat_adc[15:0]	ADC 输入数据	多达 4 个内部 ADC 数据总线用作并行输入。

表 231. DFSDM 触发器连接

触发器名称	触发源
dfsdm_jtrg0	TIM1_TRGO
dfsdm_jtrg1	TIM1_TRGO2
dfsdm_jtrg2	TIM8_TRGO
dfsdm_jtrg3	TIM8_TRGO2
dfsdm_jtrg4	TIM3_TRGO
dfsdm_jtrg5	TIM4_TRGO
dfsdm_jtrg6	TIM16_OC1
dfsdm_jtrg7	TIM6_TRGO
dfsdm_jtrg8	TIM7_TRGO
dfsdm_jtrg9	HRTIM1_ADCTRG1

表 231. DFSDM 触发器连接（续）

触发器名称	触发源
dfsdm_jtrg10	HRTIM1_ADCTRG3
dfsdm_jtrg[23:11]	保留
dfsdm_jtrg24	EXTI11
dfsdm_jtrg25	EXTI15
dfsdm_jtrg26	LPTIMER1
dfsdm_jtrg27	LPTIMER2
dfsdm_jtrg28	LPTIMER3
dfsdm_jtrg[31:29]	保留

表 232. DFSDM 断路连接

断路名称	断路目标
dfsdm_break[0]	TIM15 断路
dfsdm_break[1]	TIM16 断路 2
dfsdm_break[2]	TIM1/TIM17/TIM8 断路
dfsdm_break[3]	TIM1/TIM8 断路 2

30.4.3 DFSDM 复位和时钟

DFSDM 开关控制

通过将 DFSDM_CH0CFGR1 寄存器的 DFSDMEN 置 1，全局使能 DFSDM 接口。DFSDM 被全局使能后，所有输入通道 y (y=0..7) 和数字滤波器 DFSDM_FLTx (x=0..3) 会在其使能位（DFSDM_CHyCFGR1 中的通道使能位 CHEN 和 DFSDM_FLTxCR1 中的 DFSDM_FLTx 使能位 DFEN）置 1 时开始工作。

数字滤波器 x DFSDM_FLTx (x=0..3) 通过将 DFSDM_FLTxCR1 寄存器的 DFEN 置 1 来使能。使能 DFSDM_FLTx (DFEN=1) 后，$Sinc^x$ 数字滤波器单元和积分器单元会重新初始化。

通过将 DFEN 清零，正在进行的所有转换都会立即停止，DFSDM_FLTx 会置于停止模式。除 DFSDM_FLTxAWSR 和 DFSDM_FLTxISR（均被复位）外，所有寄存器设置保持不变。

通过将 DFSDM_CHyCFGR1 寄存器的 CHEN 置 1 使能通道 y (y=0..7)。通道使能后，会从外部 ΣΔ 调制器或并行内部数据源（ADC 或存储器的 CPU/DMA 写操作）接收串行数据。

在停止系统时钟，使器件进入停止模式之前，必须全局禁止 DFSDM（通过在 DFSDM_CH0CFGR1 中设置 DFSDMEN = 0）。

DFSDM 时钟

内部 DFSDM 时钟 $f_{DFSDMCLK}$ 用于驱动通道收发器、数字处理模块（数字滤波器、积分器）和后续附加模块（模拟看门狗、短路检测器、极值检测器、控制模块），此时钟由 RCC 模块生成，源自系统时钟 SYSCLK（最大 可达 f_{SYSCLK} = MHz）或外设时钟 PCLK2（请参见中的 DFSDMSEL 位说明）。DFSDM 时钟在停止模式下自动停止（对于所有 DFSDM_FLTx，x=0..3，DFEN = 0 时）。

DFSDM 串行通道收发器可以接收外部串行时钟，以采样外部串行数据流。如果使用标准 SPI 编码，内部 DFSDM 时钟必须至少比外部串行时钟快 4 倍；如果使用曼彻斯特编码，则须比外部串行时钟快 6 倍。

DFSDM 可提供一个外部输出时钟信号，以驱动外部 ΣΔ 调制器时钟输入。它通过 CKOUT 引脚提供。该输出时钟信号必须在给定器件手册中的指定范围之内，可以通过分频器取自 DFSDM 时钟，也可以取自音频时钟（请参见 DFSDM_CH0CFGR1 寄存器 CKOUTSRC 位），分频器可以编程，范围为 2 - 256（DFSDM_CH0CFGR1 寄存器中的 CKOUTDIV）。音频时钟源为 SAI1 时钟，可通过 RCC 配置中 SAI1SEL[1:0] 字段进行选择（请参见）。

30.4.4 串行通道收发器

有 8 个复用串行数据通道，可以通过各个滤波器、模拟看门狗或者短路检测器将其选择用于转换。这些串行收发器接收来自外部 ΣΔ 调制器的数据流。数据流能以 SPI 格式或曼彻斯特编码格式进行发送（请参见 DFSDM_CHyCFGR1 寄存器的 SITP[1:0] 位）。
通过将 DFSDM_CHyCFGR1 寄存器的 CHEN 置 1 可以使能通道。

通道输入选择

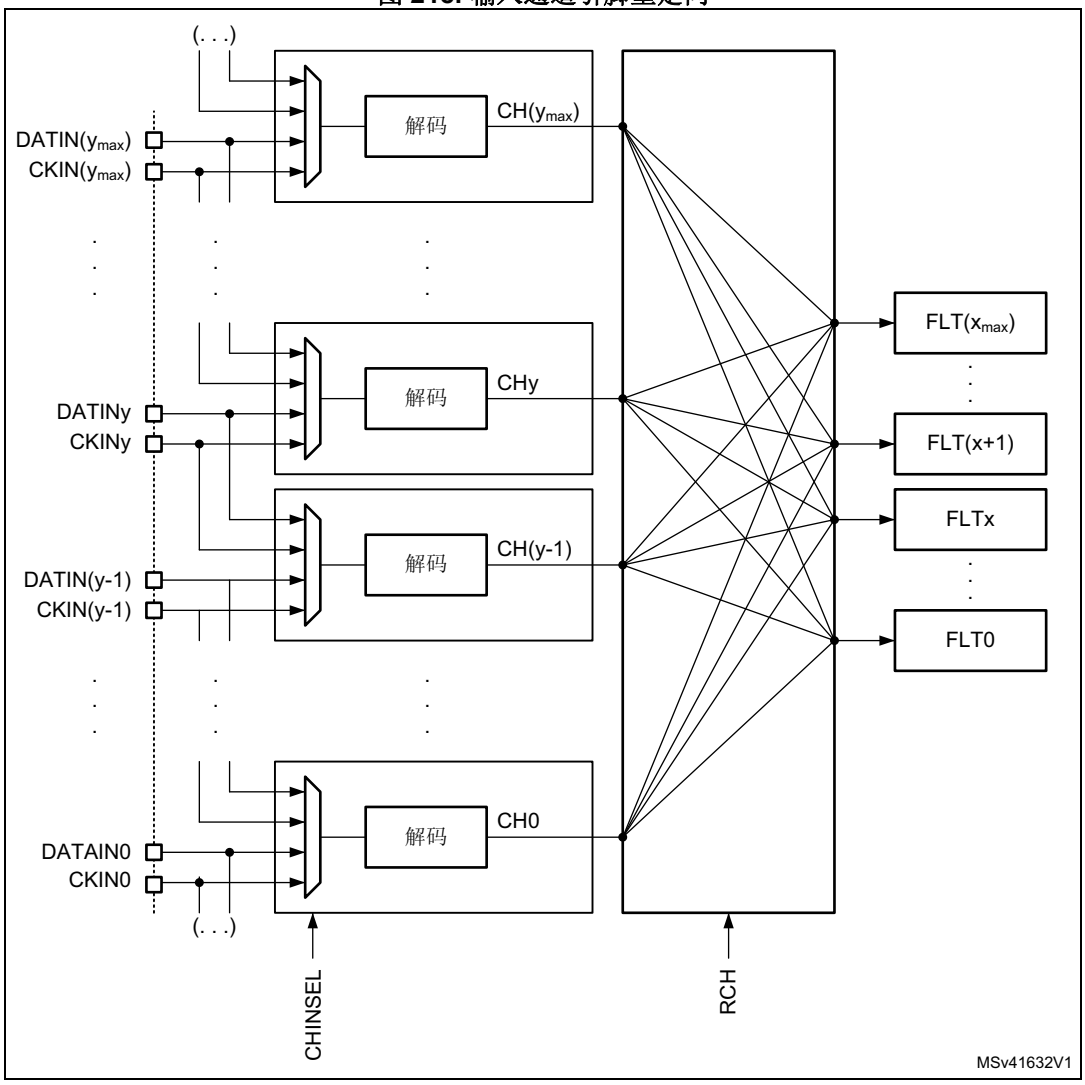
DATINy 和 CKINy 引脚的串行输入（数据和时钟信号）可从后续通道引脚重定向。此串行输入通道重定向可通过 DFSDM_CHyCFGR1 寄存器中的 CHINSEL 位设置。

通道重定向可用于从 PDM（脉冲密度调制）立体声麦克风中采集音频数据。PDM 立体声麦克风包含一个数据和一个时钟信号。数据信号为左右音频通道提供信息（时钟上升沿采样用于左通道，时钟下降沿采样用于右通道）。

PDM 麦克风输入串行通道的配置：

- PDM 麦克风信号（数据、时钟）将连接到 DFSDM 输入串行通道 y（DATINy，CKOUT）引脚。
- 将通道 y 进行以下配置：CHINSEL = 0（输入来自给定通道引脚：DATINy 和 CKINy）。
- 将通道 (y-1)（模8）进行以下配置：CHINSEL = 1（输入来自后续通道 ((y-1)+1) 引脚：DATINy 和 CKINy）。
- 通道 y：SITP[1:0] = 0（上升沿选通数据）=> 通道 y 的左音频通道。
- 通道 (y-1)：SITP[1:0] = 1（下降沿选通数据）=> 通道 y-1 的右音频通道。
- 两个 DFSDM 滤波器将分配至通道 y 和通道 (y-1)（用于对 PDM 麦克风的左右通道进行滤波）。

图 218. 输入通道引脚重定向



输出时钟生成

可在 CKOUT 引脚上提供一个时钟信号，来驱动外部 ΣΔ 调制器时钟输入。此 CKOUT 信号的频率通过预分频器（请参见 DFSDM_CH0CFGR1 寄存器 CKOUTDIV 位）取自 DFSDM 时钟或音频时钟（请参见 DFSDM_CH0CFGR1 寄存器 CKOUTSRC 位）。如果输出时钟停止，则 CKOUT 信号会被置于低电平状态（通过在 DFSDM_CHyCFGR1 寄存器中设置 CKOUTDIV = 0 或 在 DFSDM_CH0CFGR1 寄存器中设置 DFSDMEN = 0，可以停止输出时钟）。在以下时间执行输出时钟停止：

- DFSDMEN 清零后 4 个系统时钟（当 CKOUTSRC =0 时）
- DFSDMEN 清零后 1 个系统时钟和 3 个音频时钟（CKOUTSRC = 1 时）

更改 CKOUTSRC 之前，软件必须等待 CKOUT 停止，以避免在 CKOUT 引脚上产生毛刺信号。输出时钟信号频率必须介于 0 - 20 MHz 范围内。

SPI 数据输入格式操作

在 SPI 格式下，数据流通过数据和时钟信号以串行格式进行发送。数据信号始终由 DATINy 引脚提供。时钟信号可通过 CKINy 引脚从外部提供，也可以通过取自 CKOUT 信号源的信号从内部提供。

如果选择外部时钟源 (SPICKSEL[1:0] = 0)，则根据 （DFSDM_CHyCFGR1 寄存器中的）SITP[1:0] 位设置在（CKINy 引脚的）时钟上升沿或下降沿采样（DATINy 引脚上的）数据信号。

内部时钟源 —— 请参见 DFSDM_CHyCFGR1 寄存器中的 SPICKSEL[1:0]：

- CKOUT 信号：
  - 用于连接外部 ΣΔ 调制器，调制器直接使用其时钟输入（来自 CKOUT）来生成输出串行通信时钟。
  - 采样点：上升沿/下降沿，具体取决于 SITP[1:0] 设置。
- CKOUT/2 信号（在 CKOUT 上升沿生成）：
  - 用于连接外部 ΣΔ 调制器，调制器将时钟输入（来自 CKOUT）除以 2，来生成输出串行通信时钟（该输出时钟变化在各个时钟输入上升沿有效）。
  - 采样点：每第二个 CKOUT 下降沿。
- CKOUT/2 信号（在 CKOUT 下降沿生成）：
  - 用于连接外部 ΣΔ 调制器，调制器将时钟输入（来自 CKOUT）除以 2，来生成输出串行通信时钟（该输出时钟变化在各个时钟输入下降沿有效）。
  - 采样点：每第二个 CKOUT 上升沿。

注： *只有在外部 ΣΔ 调制器将 CKOUT 信号用作时钟输入（以实现同步时钟和数据操作）时，才能使用内部时钟源。*

使用内部时钟源可省去 CKINy 引脚连接（CKINy 引脚可用于其他用途）。

如果采用 SPI 编码，时钟源信号频率必须介于 0 - 20 MHz 范围内，且小于 $f_{DFSDMCLK}/4$。

曼彻斯特编码数据输入格式操作

如果采用曼彻斯特编码格式，数据流仅通过 DATINy 引脚以串行格式进行发送。经过曼彻斯特解码之后，数据和时钟信号从串行流中恢复。可以有两种曼彻斯特编码设置（请参见 DFSDM_CHyCFGR1 寄存器 SITP[1:0] 位）：

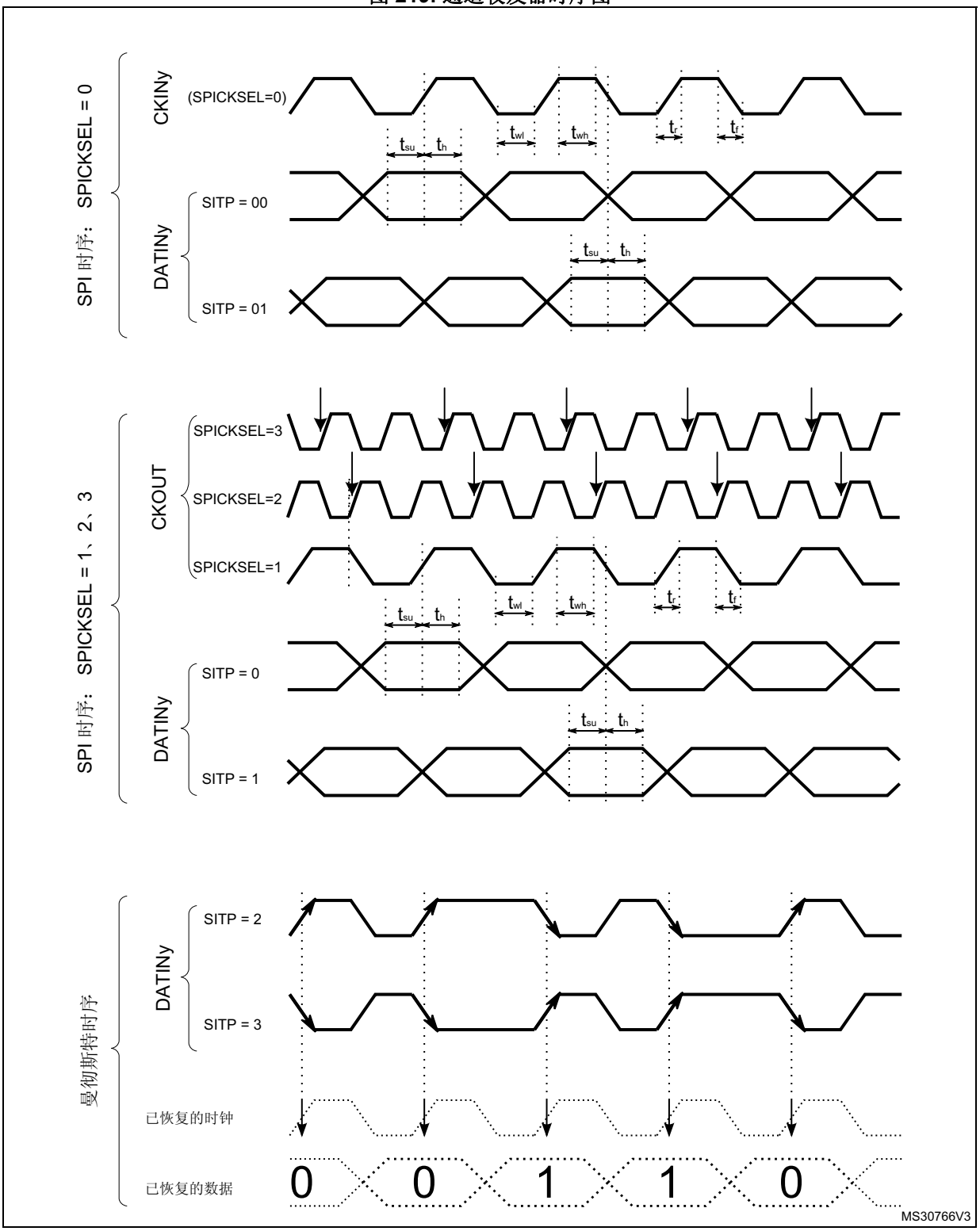
- 信号上升沿 = 逻辑 0；信号下降沿 = 逻辑 1
- 信号上升沿 = 逻辑 1；信号下降沿 = 逻辑 0

如果采用曼彻斯特编码，则恢复后的时钟信号频率必须介于 0 - 10 MHz 范围内，且小于 $f_{DFSDMCLK}/6$。

为了正确接收曼彻斯特编码数据，必须根据预期的曼彻斯特数据速率对（DFSDM_CH0CFGR1 寄存器中的）CKOUTDIV 分频器进行设置，设置公式为：

$$((CKOUTDIV + 1) \times T_{SYSCLK}) < T_{Manchester\ clock} < (2 \times CKOUTDIV \times T_{SYSCLK})$$

图 219. 通道收发器时序图



时钟缺失检测

可检查通道串行时钟输入的时钟缺失/存在情况，以确保转换和错误报告正常运行。可通过 DFSDM_CHyCFGR1 寄存器 CKABEN 位在各个输入通道 y 上使能或禁止时钟缺失检测。如果使能了给定通道的时钟缺失检测，则会对其连续执行时钟缺失检测。如果出现输入时钟错误，则时钟缺失标志会置 1 (CKABF[y] = 1)，并会调用中断（CKABIE =1 时）（请参见 DFSDM_FLT0ISR 寄存器的 CKABF[7:0] 和 DFSDM_CHyCFGR1 的 CKABEN）。时钟缺失标志清零（通过 DFSDM_FLT0ICR 寄存器的 CLRCKABF）后，会刷新时钟缺失标志。相应通道 y 禁止时，则时钟缺失状态位 CKABF[y] 还会由硬件置 1（如果 CHEN[y] = 0，则 CKABF[y] 保持在置 1 状态）。

如果发生了时钟缺失事件，则数据转换（和/或模拟看门狗和短路检测器）会提供错误的数据。当报告了时钟缺失时，用户应管理此事件并丢弃给定数据。

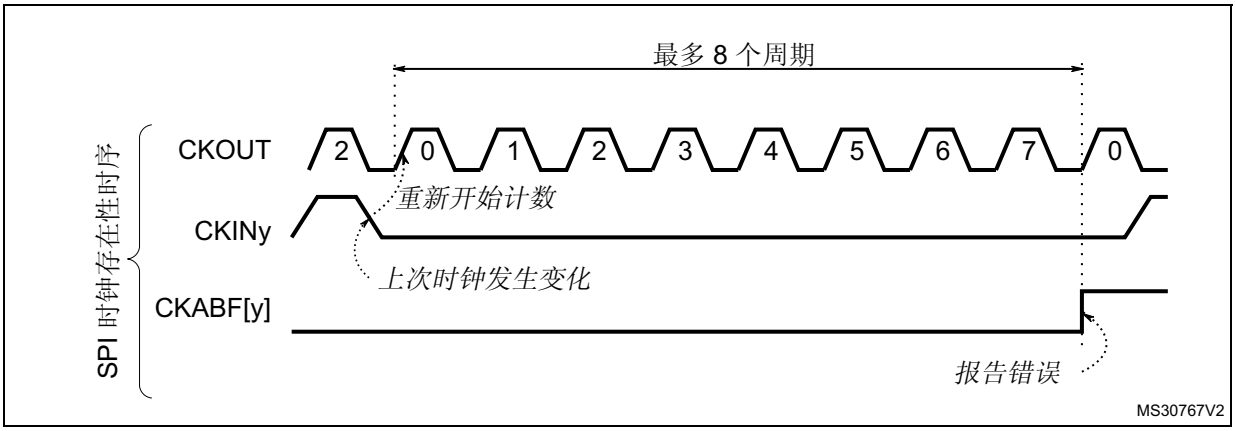
只有在将系统时钟用于 CKOUT 信号（DFSDM_CH0CFGR1 寄存器中的 CKOUTSRC = 0）时，时钟缺失功能才可用。

当收发器尚未被同步时，时钟缺失标志会置 1，且无法由（DFSDM_FLT0ICR 寄存器中）CLRCKABF[y] 位清零。与时钟缺失检测功能相关的软件序列应：

- 通过 CHEN = 1 使能给定通道。
- 尝试将时钟缺失标志清零 （通过 CLRCKABF = 1），直至时钟缺失标志已实际清零 (CKABF = 0)。此时，收发器被同步 （信号时钟有效），能够接收数据。
- 使能时钟缺失功能 CKABEN = 1 且相关中断 CKABIE = 1，以检测 SPI 时钟是否丢失或曼彻斯特数据边沿是否缺失。

如果使用 SPI 数据格式，将基于外部输入时钟与输出时钟生成（CKOUT 信号）的比较进行时钟缺失检测。输入通道中的外部输入时钟信号必须每经过 CKOUT 信号（由 DFSDM_CH0CFGR1 寄存器 CKOUTDIV 字段控制）的 8 个信号周期更改一次。

图 220. SPI 时钟缺失时序图



如果使用曼彻斯特数据格式，则时钟缺失表示无法从曼彻斯特编码信号中恢复时钟。为了能够正常恢复时钟，首先必须接收带 1 - 0 或 0 - 1 转换的数据（有关曼彻斯特同步，请参见 *图 222*）。

如果使用曼彻斯特编码格式，则时钟缺失检测（在首次成功同步之后）是基于编码串行数据输入信号与输出时钟生成（CKOUT 信号）之间的变化比较。在 CKOUT 信号（由 DFSDM_CH0CFGR1 寄存器 CKOUTDIV 位控制）的两个周期期间，DATINy 引脚电平必须发生变化。该条件还定义了能正确恢复曼彻斯特编码的数据和时钟信号的最小数据速率。

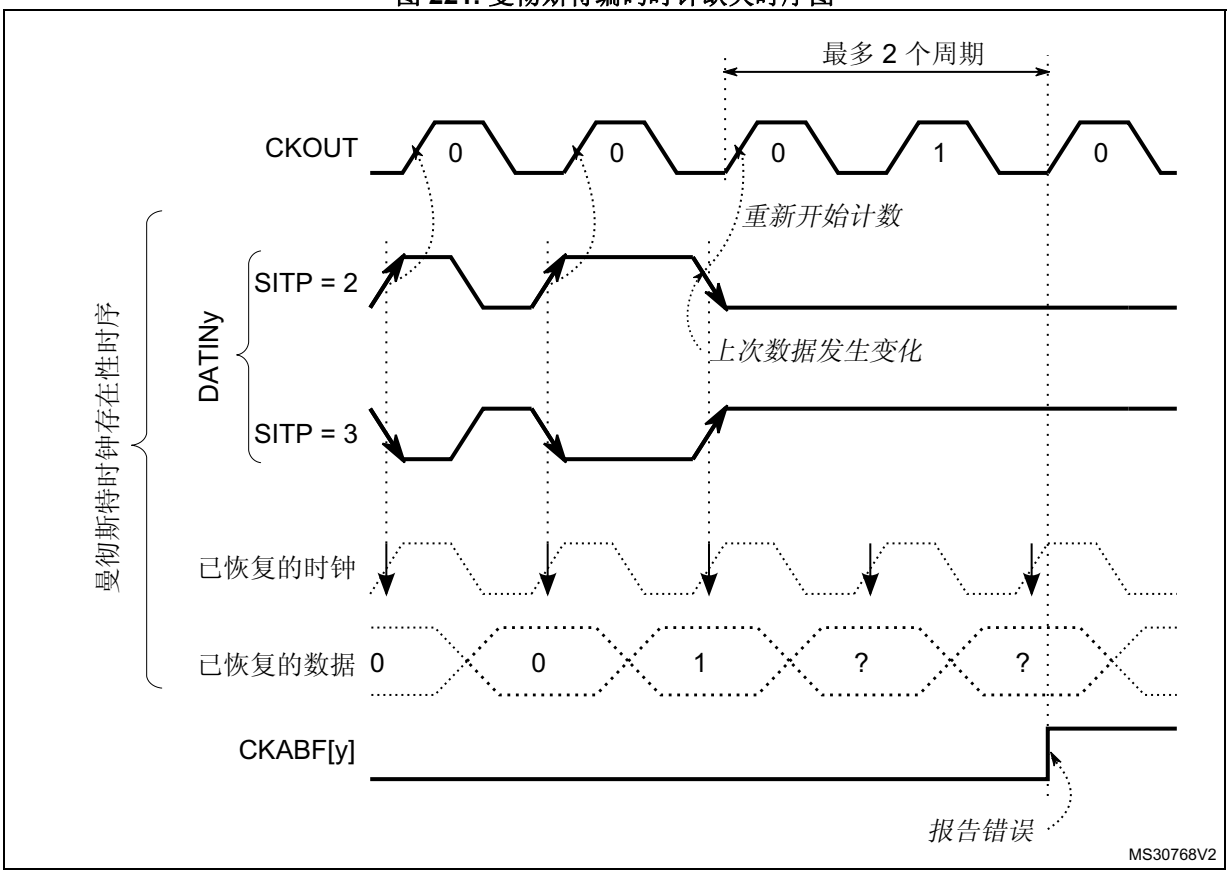
曼彻斯特编码数据的最大数据速率必须小于 CKOUT 信号。

因此，为正确接收曼彻斯特编码数据，必须根据以下公式设置 CKOUTDIV 分频器：

$$((CKOUTDIV + 1) \times T_{SYSCLK}) < T_{Manchester\ clock} < (2 \times CKOUTDIV \times T_{SYSCLK})$$

如果出现输入时钟恢复错误，则时钟缺失标志会置 1 (CKABF[y] = 1)，并会调用中断（CKABIE=1 时）（请参见 DFSDM_FLT0ISR 寄存器的 CKABF[7:0] 和 DFSDM_CHyCFGR1 的 CKABEN）。时钟缺失标志清零 （通过 DFSDM_FLT0ICR 寄存器的 CLRCKABF）后，会刷新时钟缺失标志。

图 221. 曼彻斯特编码时钟缺失时序图



曼彻斯特/SPI 编码同步

必须在使能通道（DFSDM_CHyCFGR1 寄存器 CHEN = 1）后，首次同步曼彻斯特编码数据流。当接收到 0 - 1 或 1 - 0 数据转换（以便能检测有效数据边沿）时，同步结束。在通过 DFSDM_FLT0ICR 的 CLRCKABF [y] 将给定通道的 CKABF[y] 清零之后，可以按照以下详细描述的软件序列通过轮询 CKABF [y] = 0 来检查同步是否结束：

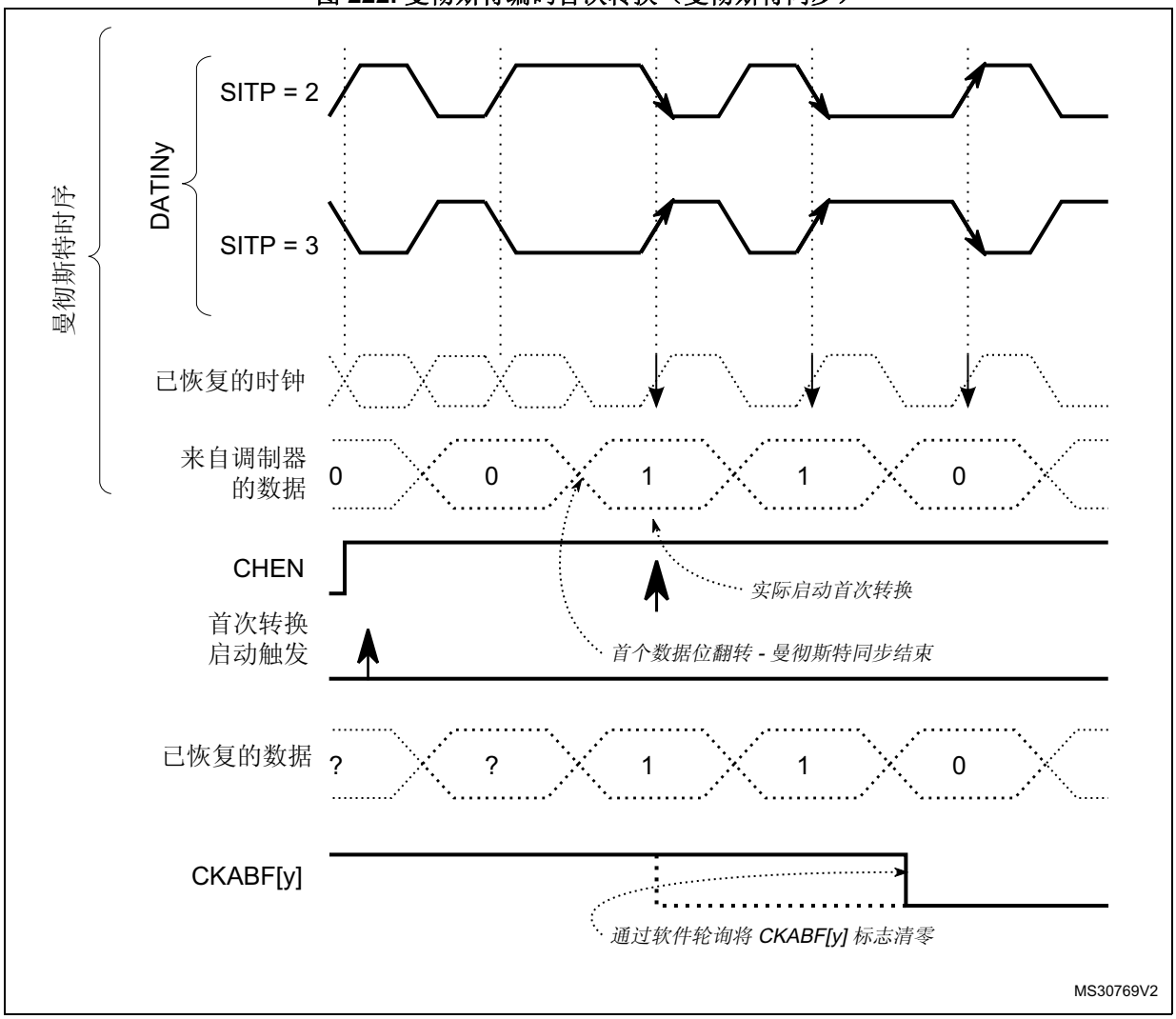
通过将 CLRCKABF[y] 位置 1 将 CKABF[y] 标志清零。如果通道 y 尚未同步，则硬件会立即将 CKABF[y] 标志置 1。软件随后将回读 CKABF[y] 标志，如果该标志置 1，则会再次通过将 CLRCKABF[y] 位置 1 来将其清零。此软件序列（CKABF[y] 标志的轮询）将继续进行，直至 CKABF[y] 标志置 1（表示曼彻斯特码流已同步）。为了能够同步/接收曼彻斯特编码数据，必须根据预期的曼彻斯特数据速率对（DFSDM_CH0CFGR1 寄存器中的）CKOUTDIV 分频器进行设置，以下为设置公式：

$$((CKOUTDIV + 1) \times T_{SYSCLK}) < T_{Manchester\ clock} < (2 \times CKOUTDIV \times T_{SYSCLK})$$

SPI 码流在第一次检测到时钟输入信号（有效的上升沿/下降沿）后同步。

注： *当收发器尚未被同步时，时钟缺失标志会置 1，且无法由（DFSDM_FLT0ICR 寄存器中）CLRCKABF[y] 位清零。*

图 222. 曼彻斯特编码首次转换（曼彻斯特同步）



外部串行时钟频率测量

通过测量通道串行时钟输入频率可提供来自外部 ΣΔ 调制器的实时数据速率，这对应用目的很重要。

外部串行时钟输入频率可以通过在一个转换持续时间内对 DFSDM 时钟（$f_{DFSDMCLK}$）进行计数的定时器来测量。计数在转换触发（常规或注入）后的第一个输入数据时钟处开始，并在转换结束（转换结束标志置 1）前的最后一个输入数据时钟处结束。转换完成（JEOCF =1 或 REOCF = 1）时，会在寄存器 DFSDM_FLTxCNVTIMR 的计数器 CNVCNT[27:0] 中更新每次转换的时间（第一个串行采样和最后一个串行采样之间的时间）。然后用户可以根据数字滤波器设置（FORD、FOSR、IOSR、FAST）计算数据速率。只有在滤波器被旁路（FOSR = 0，只有积分器有效，DFSDM_FLTxCNVTIMR 寄存器 CNVCNT[27:0] = 0）时，才会停止外部串行频率测量。

对于并行数据输入（*第 30.4.6 节：并行数据输入*），测量的频率为一次转换期间内的平均输入数据速率。

注： *转换被中断（例如，通过禁止/使能所选通道）时，还会在 CNVCNT[27:0] 中对中断时间进行计数。因此，为得到正确的转换时间结果，建议不要中断转换。*

转换时间：

FAST = 0 时的注入转换或常规转换（或者 FAST=1 时的首次转换）：

对于 $Sinc^x$ 滤波器 (x =1..5)：

$t = CNVCNT/f_{DFSDMCLK} = [F_{OSR} * (I_{OSR}-1 + F_{ORD}) + F_{ORD}] / f_{CKIN}$

对于 FastSinc 滤波器：

$t = CNVCNT/f_{DFSDMCLK} = [F_{OSR} * (I_{OSR}-1 + 4) + 2] / f_{CKIN}$

FAST = 1 时的常规转换（首次转换除外）：

对于 $Sinc^x$ 和 FastSinc 滤波器：

$t = CNVCNT/f_{DFSDMCLK} = [F_{OSR} * I_{OSR}] / f_{CKIN}$

如果 F_{OSR} = FOSR[9:0] + 1 = 1（滤波器旁路，仅积分器有效），则：

$t = I_{OSR} / f_{CKIN}$（... 但是 CNVCNT = 0）

其中：

- f_{CKIN} 为（给定通道 CKINy 引脚上的）通道输入时钟频率或（并行数据输入时的）输入数据速率
- F_{OSR} 为滤波器过采样率：F_{OSR} = FOSR[9:0] + 1（请参见 DFSDM_FLTxFCR 寄存器）
- I_{OSR} 为积分器过采样率：I_{OSR} = IOSR[7:0] + 1（请参见 DFSDM_FLTxFCR 寄存器）
- F_{ORD} 为滤波器阶数：F_{ORD} = FORD[2:0]（请参见 DFSDM_FLTxFCR 寄存器）

通道偏移设置

每个通道都有自身的偏移设置（在寄存器中），该偏移值最终会从给定通道的各个转换结果（注入或常规）中减去。在数据右移位后执行偏移校正。偏移值以 24 位有符号值的形式存储在 DFSDM_CHyCFGR2 寄存器的 OFFSET[23:0] 字段中。

数据右移位

要使结果与 24 位值对齐，每个通道均定义了一系列右移位，这些右移位将应用于给定通道的各个转换结果（注入或常规）。右移位数存储在 DFSDM_CHyCFGR2 寄存器的 DTRBS[4:0] 位中。

右移位将结果四舍五入为最接近的整数值。移位结果的符号保留，以便得到有效 24 位有符号格式的结果数据。

30.4.5 配置输入串行接口

必须为输入串行接口配置以下参数：

- **输出时钟预分频器**。提供了可编程预分频器，用来从 DFSDM 时钟生成输出时钟 (2 - 256)。它由 DFSDM_CH0CFGR1 寄存器中的 CKOUTDIV[7:0] 位定义。
- **串行接口类型和输入时钟相位。**选择 SPI 或曼彻斯特编码以及输入时钟的采样边沿。它由 DFSDM_CHyCFGR1 寄存器中的 SITP [1:0] 位定义。
- **输入时钟源。**来自 CKINy 引脚的外部源或来自 CKOUT 引脚的内部源。它由 DFSDM_CHyCFGR1 寄存器中的 SPICKSEL[1:0] 字段定义。
- **最终数据右移位。**定义最终数据右移位，以将结果与 24 位值对齐。它由 DFSDM_CHyCFGR2 寄存器中的 DTRBS[4:0] 定义。
- **每个通道的通道偏移。**定义给定串行通道的模拟偏移（所连接外部 ΣΔ 调制器的偏移）。它由 DFSDM_CHyCFGR2 寄存器中的 OFFSET[23:0] 位定义。
- **每个通道的短路检测器和时钟缺失使能。**用于在 DFSDM_CHyCFGR1 寄存器中使能或禁止给定串行通道的短路检测器（通过 SCDEN 位）和时钟缺失检测（通过 CKABEN 位）。
- **模拟看门狗滤波器和短路检测器阈值设置。**用于配置通道模拟看门狗滤波器参数和通道短路检测器参数。在 DFSDM_CHyAWSCDR 寄存器中定义这些配置。

30.4.6 并行数据输入

每个输入通道均为 16 位并行数据输入提供一个寄存器（除了串行数据输入外）。每个 16 位并行输入只能源自内部数据源：

- 内部 ADC 结果(3)
- 直接 CPU/DMA 写操作

通过 DFSDM_CHyCFGR1 寄存器的 DATMPX[1:0] 字段选择是对给定通道使用串行还是并行数据输入。DATMPX[1:0] 中还定义了并行数据源：内部 ADC(3) 或直接由 CPU/DMA 进行的写操作。

每个通道均包含一个 32 位数据输入寄存器 DFSDM_CHyDATINR，在该寄存器中可写入 16 位数据。数据采用 16 位有符号格式。这些数据还可用作接受 16 位并行数据的数字滤波器的输入。

如果选择串行数据输入 (DATMPX[1:0] = 0)，则 DFSDM_CHyDATINR 寄存器被写保护。

来自内部 ADC 的输入(3)

对于并行 ADC 数据输入 (DATMPX[1:0]=1)，ADC[y+1] 结果被分配至通道 y 输入（ADC1 填充 DFSDM_CHDATIN0R 寄存器，ADC2 填充 DFSDM_CHDATIN1R 寄存器，...，ADC8 填充 DFSDM_CHDATIN7R 寄存器）。来自 ADC[y+1] 的转换结束事件会导致更新通道 y 的数据（来自 ADC[y+1] 的并行数据被用作数字滤波器的下一个采样）。转换结束事件发生时，来自 ADC[y+1] 的数据被写入 DFSDM_CHyDATINR 寄存器（INDAT0[15:0] 字段）。

数据封装模式设置（DFSDM_CHyCFGR1 寄存器中的 DATPACK[1:0]）对 ADC 数据输入无影响。

注： *ADC 扩展规范：将内部 ADC 配置为交错模式时（例如，ADC1 与 ADC2 交错 - 请参见 ADC 规范），来自 ADC1 或 ADC2 的每个结果均会传送至同一 16 位总线（即 ADC1 的总线），该总线连至 DFSDM 通道 0（固定连接）。因此，DFSDM 通道 0 中将存在双倍输入数据速率（偶数采样来自 ADC1，奇数采样来自 ADC2）。与 ADC2 相关的通道 1 将处于空闲状态。*

3. 仅 ADC1 和 ADC2。

来自存储器的输入（直接由 CPU/DMA 写入）

对于由 CPU 或 DMA (DATMPX[1:0] = 2) 直接写入到 DFSDM_CHyDATINR 寄存器中的数据，可以将其用作数据输入，以处理来自存储器或外设的数字数据流。

数据可由 CPU 或 DMA 写入到 DFSDM_CHyDATINR 寄存器中：

1. **CPU 数据写入：**

   输入数据直接由 CPU 写入到 DFSDM_CHyDATINR 寄存器中。

2. **DMA 数据写入：**

   DMA 应配置为存储器到存储器传输模式，以将存储器缓冲区中的数据传输至 DFSDM_CHyDATINR 寄存器中。目标存储器地址为 DFSDM_CHyDATINR 寄存器的地址。数据以 DMA 传输速度从存储器传输至 DFSDM 并行输入。

   此 DMA 不同于读取 DFSDM 转换结果所使用的 DMA。这两个 DMA 可同时使用，第一个 DMA（配置为存储器到存储器传输）用于输入数据写入，第二个 DMA（配置为外设到存储器传输）用于数据结果读取。

对 DFSDM_CHyDATINR 的访问可为 16 位或 32 位宽，允许分别在一次写操作中加载一个或两个采样。32 位输入数据寄存器 (DFSDM_CHyDATINR) 可用一个或两个 16 位数据采样进行填充，具体取决于在 DFSDM_CHyCFGR1 寄存器的 DATPACK[1:0] 字段中定义的数据封装操作模式：

1. **标准模式 (DATPACK[1:0] = 0)：**

   只有一个样本存储在 DFSDM_CHyDATINR 寄存器的 INDAT0[15:0] 字段中，其用作通道 y 的输入数据。高 16 位 (INDAT1[15:0]) 被忽略，且被写保护。数字滤波器必须执行一次输入采样（通过 INDAT0[15:0]），以便清空由 CPU/DMA 填充后的数据寄存器。该模式与对 DFSDM_CHyDATINR 寄存器的 16 位 CPU/DMA 访问结合使用，以在每次写操作时加载一个采样。

2. **交错模式 (DATPACK[1:0] = 1)：**

   DFSDM_CHyDATINR 寄存器被用作两个采样缓冲区。第一个采样存储在 INDAT0[15:0] 中，第二个采样存储在 INDAT1[15:0] 中。数字滤波器必须通过通道 y 执行两次输入采样，以清空 DFSDM_CHyDATINR 寄存器。该模式与对 DFSDM_CHyDATINR 寄存器的 32 位 CPU/DMA 访问结合使用，以在每次写操作时加载两个采样。

3. **双通道模式 (DATPACK[1:0] = 2)：**

   将两个采样写入到 DFSDM_CHyDATINR 寄存器中。INDAT0[15:0] 中的数据用于通道 y，INDAT1[15:0] 中的数据用于通道 y+1。INDAT1[15:0] 中的数据会被自动复制到下一 (y+1) 通道数据寄存器 DFSDM_CH[y+1]DATINR) 的 INDAT0[15:0] 中。数字滤波器必须执行两次采样（一次来自通道 y，另一次来自通道 (y+1)），以清空 DFSDM_CHyDATINR 寄存器。

   只能针对偶数通道数（y = 0、2、4、6）进行双通道模式设置 (DATPACK[1:0] = 2)。如果将奇数通道（y = 1、3、5、7）设为双通道模式，则对于该通道，INDAT0[15:0] 和 INDAT1[15:0] 部分均被写保护。如果偶数通道被设为双通道模式，则下一奇数通道必须设置成标准模式 (DATPACK[1:0] = 0)，以便与偶数通道正确配合。

有关 DFSDM_CHyDATINR 寄存器数据模式和通道数据样本分配，请参见*图 223*。

图 223. DFSDM_CHyDATINR 寄存器操作模式和分配

首先使能所选输入通道（通道 y），以便进行数据采集（启动通道 y 转换），然后必须向 DFSDM_CHyDATINR 寄存器进行写操作，以加载一个或两个采样。否则，在下次处理时会丢失写入的数据。

例如：对于单次转换和交错模式，在启动单次转换前，不要开始向 DFSDM_CHyDATINR 写入成对数据样本（开始转换前 DFSDM_CHyDATINR 中存在的数据均会被丢弃）。

30.4.7 通道选择

存在 8 个复用通道，可选择用于注入通道组转换和/或常规通道转换。

注入通道组可以选择 8 个通道中的任一通道或其全部。DFSDM_FLTxJCHGR 寄存器中的 JCHG[7:0] 选择注入组通道，其中 JCHG[y] =1 表示已选择通道 y。

注入转换可工作在扫描模式 (JSCAN = 1) 或单个模式 (JSCAN = 0) 下。在扫描模式下，每个所选通道都会依次转换。最低通道（通道 0，如果已选择）最先转换，紧接着转换下一个较高通道，直至 JCHG[7:0] 选择的所有通道均被转换。在单个模式 (JSCAN=0) 下，只会转换所选的一个通道，通道选择会移至下一通道。JSCAN = 0 时，对 JCHG[7:0] 进行写入操作会将通道选择复位为选择的最低通道。

注入转换可由软件或触发器启动。此类转换从不会被常规转换中断。

常规通道只选择 8 个通道的其中一个。DFSDM_FLTxCR1 寄存器中的 RCH[2:0] 指示所选通道。

常规转换只能由软件启动（不能由触发器启动）。请求注入转换时，会暂时中断连续常规转换序列。

对已禁止通道（DFSDM_CHyCFGR1 寄存器中的 CHEN = 0）执行转换会导致转换将永远不会结束，这是因为未提供输入数据的原因（无时钟信号）。在这种情况下，需要使能给定通道（DFSDM_CHyCFGR1 寄存器中的 CHEN = 1），或停止转换（通过 DFSDM_FLTxCR1 寄存器的 DFEN = 0）。

30.4.8 数字滤波器配置

DFSDM 包含 $Sinc^x$ 类型数字滤波器实现。此 $Sinc^x$ 滤波器执行输入数字数据流滤波，这会导致减小输出数据速率（抽取）以及增大输出数据分辨率。可对 $Sinc^x$ 数字滤波器进行配置，以达到所需输出数据速率和所需输出数据分辨率。可配置参数有：

- 滤波器阶数/类型：（请参见 DFSDM_FLTxFCR 寄存器中的 FORD[2:0] 位）：
  - FastSinc
  - $Sinc^1$
  - $Sinc^2$
  - $Sinc^3$
  - $Sinc^4$
  - $Sinc^5$
- 滤波器过采样/抽取率：（请参见 DFSDM_FLTxFCR 寄存器中的 FOSR[9:0] 位）：
  - FOSR = 1-1024 —— 适用于 FastSinc 滤波器 和 $Sinc^x$ 滤波器 x = F_{ORD} = 1..3
  - FOSR = 1-215 —— 适用于 $Sinc^x$ 滤波器 x = F_{ORD} = 4
  - FOSR = 1-73 —— 适用于 $Sinc^x$ 滤波器 x = F_{ORD} = 5

滤波器支持以下传递函数（H 域中的冲激响应）：

- $Sinc^x$ 滤波器类型：
$$H(z) = \left(\frac{1 - z^{-FOSR}}{1 - z^{-1}}\right)^x$$

- FastSinc 滤波器类型：
$$H(z) = \left(\frac{1 - z^{-FOSR}}{1 - z^{-1}}\right)^2 \cdot (1 + z^{-(2 \cdot FOSR)})$$

图 224. 示例：$Sinc^3$ 滤波器响应

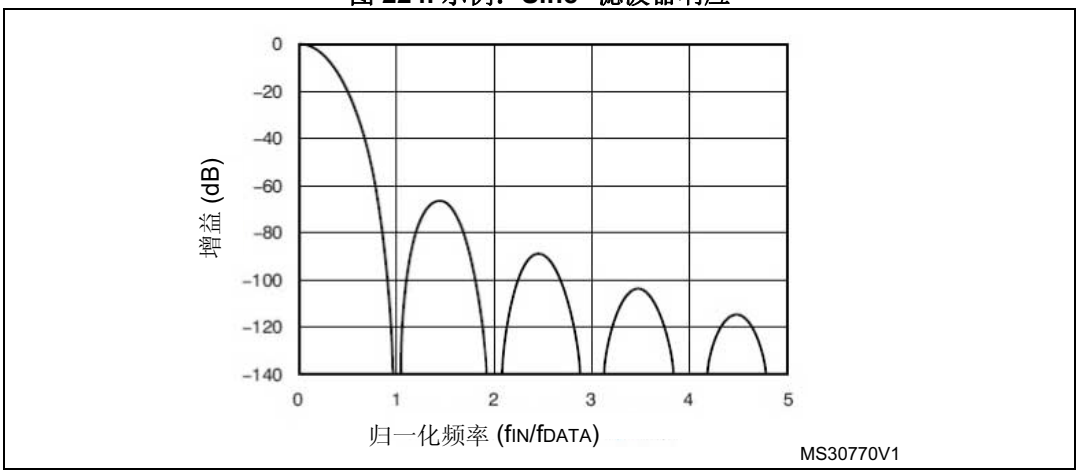


表 233. 滤波器最大输出分辨率（来自滤波器输出的峰值数据值），基于某些 FOSR 值

FOSR	Sinc^1	Sinc^2	FastSinc	Sinc^3	Sinc^4	Sinc^5
x	+/- x	+/- x^2	+/- $2x^2$	+/- x^3	+/- x^4	+/- x^5
4	+/- 4	+/- 16	+/- 32	+/- 64	+/- 256	+/- 1024
8	+/- 8	+/- 64	+/- 128	+/- 512	+/- 4096	-
32	+/- 32	+/- 1024	+/- 2048	+/- 32768	+/- 1048576	+/- 33554432
64	+/- 64	+/- 4096	+/- 8192	+/- 262144	+/- 16777216	+/- 1073741824
128	+/- 128	+/- 16384	+/- 32768	+/- 2097152	+/- 268435456	在满量程输入的条件下，结果会溢出（> 32 位有符号整数值）
256	+/- 256	+/- 65536	+/- 131072	+/- 16777216		
1024	+/- 1024	+/- 1048576	+/- 2097152	+/- 1073741824		

有关 Sinc 滤波器类型属性和使用的更多信息，建议了解下数字滤波器的理论知识（可从 Internet 上下载更多资源）。

30.4.9 积分器单元

积分器会对来自数字滤波器的数据进一步提升抽取率和分辨率。对于来自滤波器的给定数量的数据采样，积分器对其进行简单的求和。

积分器过采样率参数定义了将多少个数据的和作为积分器的一个数据输出。IOSR 可设置为介于 1-256 范围内（请参见 DFSDM_FLTxFCR 寄存器中的 IOSR[7:0] 位说明）。

表 234. 积分器最大输出分辨率（来自积分器输出的峰值数据值），基于某些 IOSR 值，FOSR = 256 以及 Sinc^3 滤波器类型（最大数据）

IOSR	Sinc^1	Sinc^2	FastSinc	Sinc^3	Sinc^4	Sinc^5
x	+/- FOSR. x	+/- $FOSR^2$. x	+/- 2.$FOSR^2$. x	+/- $FOSR^3$. x	+/- $FOSR^4$. x	+/- $FOSR^5$. x
4	-	-	-	+/- 67 108 864	-	-
32	-	-	-	+/- 536 870 912	-	-
128	-	-	-	+/- 2 147 483 648	-	-
256	-	-	-	+/- 2^{32}	-	-

30.4.10 模拟看门狗

模拟看门狗用于在达到或超出给定阈值上限或者达到或低于给定阈值下限时，触发外部信号（断路或中断）。随即可调用中断/事件/断路生成。

每个模拟看门狗都会根据（DFSDM_FLTxCR1 寄存器中）AWFSEL 位设置监视串行数据接收器输出（在每个通道上模拟看门狗滤波器之后）或数据输出寄存器（当前注入或常规转换结果）。是否需要通过模拟看门狗 x 监视的输入通道可以通过 DFSDM_FLTxCR2 寄存器中的 AWDCH[7:0] 选择。

输入通道上的模拟看门狗转换与标准转换无关。在标准转换模式下，模拟看门狗在每个输入通道上使用自己的滤波器和信号处理，与注入或常规主转换无关。在连续转换模式下，对所选输入通道执行模拟看门狗转换，以便在注入或常规主转换暂停（RCIP = 0，JCIP = 0）时也仍能对通道进行监视。

阈值上限和阈值下限寄存器与给定数据值（通过 DFSDM_FLTxAWHTR 寄存器的 AWHT[23:0] 位和 DFSDM_FLTxAWLTR 寄存器的 AWLT[23:0] 位进行设置）进行比较。

可以选择 2 种情况将阈值寄存器与数据值进行比较：

- 选择 1：在这种情况下，输入数据取自最终输出数据寄存器 (AWFSEL = 0)。该选择具有以下特性：
  - 高输入数据分辨率（高达 24 位）
  - 响应时间慢——不适用于过流检测等快速响应应用
  - 为了进行比较，最终数据需要经过移位和偏移数据校正
  - 最终数据仅在执行常规或注入主转换后才可用
  - 可在并行输入数据源（DFSDM_CHyCFGR1 寄存器中的 DATMPX[1:0] ≠ 0）情况下使用
- 选择 2：在这种情况下，输入数据取自任一串行数据接收器输出 (AWFSEL = 1)。该选择具有以下特性：
  - 输入串行数据通过专用模拟看门狗 $Sinc^x$ 通道滤波器进行处理，滤波器的过采样率 (1..32) 和滤波器阶数 (1..3) 可以配置（请参见 DFSDM_CHyAWSCDR 寄存器中的 AWFOSR[4:0] 和 AWFORD[1:0] 位设置）
  - 较低分辨率（最高为 16 位）
  - 响应时间快——适用于需要快速响应的应用，如过流/过压检测
  - 数据以连续模式提供，与常规或注入主转换活动无关

进行输入通道监视 (AWFSEL = 1) 时，用于跟阈值进行比较的数据取自 AWDCH[7:0] 字段（DFSDM_FLTxCR2 寄存器）所选的通道。将每个所选通道滤波器结果与一个阈值对 (AWHT[23:0] / AWLT[23:0]) 进行比较。此时，只有高 16 位 (AWHT[23:8] / AWLT[23:8]) 会定义与模拟看门狗滤波器输出进行比较的 16 位阈值，因为来自模拟看门狗滤波器的数据的分辨率最高为 16 位。在这种情况下 (AWFSEL=1)，不对 AWHT[7:0] / AWLT[7:0] 位进行比较。

在 DFSDM_CHyAWSCDR 寄存器中设置各个输入通道的模拟看门狗滤波器配置的参数（滤波器阶数 AWFORD[1:0] 和滤波器过采样率 AWFOSR[4:0]）。

每个输入通道都配有自身的比较器，用于比较模拟看门狗数据 （来自模拟看门狗滤波器）和模拟看门狗阈值 (AWHT/AWLT)。选择多个通道（DFSDM_FLTxCR2 寄存器的字段 AWDCH[7:0]）时，可同时接收多个比较请求。在这种情况下，首先管理编号最小的通道请求，然后继续管理编号较大的所选通道。对于每个通道，结果均可记录在单独的标志（DFSDM_FLTxAWSR 寄存器的字段 AWHTF[7:0] 和字段 AWLTF[7:0]）中。每个通道请求均在 8 个 DFSDM 时钟周期内执行。因此，各个通道的带宽被限制为 8 个 DFSDM 时钟周期 （AWDCH[7:0] = 0xFF 时）。因为最大输入通道采样时钟频率为 DFSDM 时钟频率的四分之一，因此在该输入时钟速度下，不能对模拟看门狗配置 AWFOSR = 0 （模拟看门狗滤波器被旁路）。因此，用户必须根据输入采样时钟速度和 DFSDM 频率正确配置监视的通道数和模拟看门狗滤波器参数。

给定通道 y 的模拟看门狗滤波器数据可通过固件从 DFSDM_CHyWDATR 寄存器 WDATA[15:0] 字段中读取。如果 DFSDM_CHyCFGR1 寄存器中的 CHEN = 1，则该模拟看门狗滤波器数据被连续转换，其数据速率由模拟看门狗滤波器设置和通道输入时钟频率给定。

模拟看门狗滤波器转换的工作方式类似于常规的快速连续转换（无积分器）。模拟看门狗滤波器输出（通道输入时钟频率为 f_{CKIN}）一个结果所需的串行采样数量：

首次转换：

对于 $Sinc^x$ 滤波器 (x = 1..5)：采样数 = [F_{OSR} * F_{ORD} + F_{ORD} + 1]

对于 FastSinc 滤波器：采样数 = [F_{OSR} * 4 + 2 + 1]

后续转换：

对于 $Sinc^x$ 和 FastSinc 滤波器：采样数 = [FOSR * IOSR]

其中：

F_{OSR} .....滤波器过采样率：F_{OSR} = AWFOSR[4:0] + 1（请参见 DFSDM_CHyAWSCDR 寄存器）

F_{ORD} .....滤波器阶数：F_{ORD} = AWFORD[1:0]（请参见 DFSDM_CHyAWSCDR 寄存器）

对于输出数据寄存器监视 (AWFSEL = 0)，在完成最终数据的右移位和偏移校正（请参见 DFSDM_CHyCFGR2 寄存器的 OFFSET[23:0] 和 DTRBS[4:0] 字段）后进行比较。在由（DFSDM_FLTxCR2 寄存器中）AWDCH[7:0] 字段所选通道的每次注入或常规转换结束后，进行比较。

模拟看门狗事件的状态反映在 DFSDM_FLTxAWSR 寄存器中，该寄存器中锁定了给定事件。AWHTF[y] = 1 标志指示在通道 y 上是否超出了 AWHT[23:0] 值。AWLTF[y] = 1 标志指示在通道 y 上是否超出了 AWLT[23:0] 值。对于 DFSDM_FLTxAWSR 寄存器中锁存的事件，将通过向 DFSDM_FLTxAWCFR 寄存器中的相应清零位 CLRAWHTF[y] 或 CLRAWLTF[y] 写入“1”来清除。

模拟看门狗的全局状态通过 DFSDM_FLTxISR 寄存器中的 AWDF 标志位来反映（其用于快速检测中断源）。AWDF = 1 表示至少发生了一个看门狗事件（至少对于一个通道，AWHTF[y] = 1 或 AWLTF[y] = 1）。所有 AWHTF[7:0] 和 AWLTF[7:0] 均清零时，AWDF 位才清零。

可将模拟看门狗事件分配至断路输出信号。可将四个断路输出分配至超出阈值上限或低于阈值下限的事件 (dfsdm_break[3:0])。通过 DFSDM_FLTxAWHTR 和 DFSDM_FLTxAWLTR 寄存器中的 BKAWH[3:0] 和 BKAWL[3:0] 字段将断路信号分配至给定模拟看门狗事件。

30.4.11 短路检测器

使用短路检测器的目的在于，当模拟信号达到饱和值（超出满量程范围）并保持给定的时间时，短路检测器可以在极快的响应时间内发出信号。该特性可检测短路或断路故障（例如过流或过压）。可调用中断/事件/断路生成。

短路检测器的输入数据取自通道收发器输出。

各个输入通道上都有一个递增计数器，用于对串行数据接收器输出上的连续 0 或 1 进行计数。如果接收的数据流发生变化 （数据信号从 1 到 0 或从 0 到 1），则会重新启动计数器。如果该计数器达到短路阈值寄存器值 （DFSDM_CHyAWSCDR 寄存器中的 SCDT[7:0] 位），则会调用短路事件。每个输入通道都配有短路检测器。通过将 SCDEN 位（DFSDM_CHyCFGR1 寄存器中）置 1，可选择对任一通道进行连续监视，每个通道都有其自己的短路检测器设置（阈值位 SCDT[7:0]、状态位 SCDF[7:0]、状态清零位 CLRSCDF[7:0]） 。禁止相应通道 y (CHEN[y] = 0) 时，状态标志 SCDF[y] 还可由硬件清零。

对于各个通道，可将短路检测器事件分配至断路输出信号 dfsdm_break[3:0]。可将四个断路输出分配至短路检测器事件。通过 DFSDM_CHyAWSCDR 寄存器中的 BKSCD[3:0] 字段将断路信号分配至给定通道短路检测器事件。

如果选择了并行输入数据通道（DFSDM_CHyCFGR1 寄存器中的 DATMPX[1:0] ≠ 0），则无法使用短路检测器。

总共有四个断路输出（与模拟看门狗功能共享）。

30.4.12 极值检测器

极值检测器用于采集最终输出数据字的最小值和最大值（峰峰值）。

如果输出数据字高于极值检测器最大值寄存器（DFSDM_FLTxEXMAX 寄存器中的EXMAX[23:0]位）中存储的值，则该寄存器将用当前输出数据字值进行更新，且存储其数据的通道位于（DFSDM_FLTxEXMAX 寄存器的） EXMAXCH[2:0] 位中。

如果输出数据字低于极值检测器最小值寄存器（DFSDM_FLTxEXMIN 寄存器中的EXMIN[23:0]位）中存储的值，则该寄存器将用当前输出数据字值进行更新，且存储其数据的通道位于（DFSDM_FLTxEXMIN 寄存器的）EXMINCH[2:0] 位中。

最小值和最大值寄存器值可由软件（通过读取给定的 DFSDM_FLTxEXMAX 或 DFSDM_FLTxEXMIN 寄存器）进行刷新。刷新后，极值检测器最小值数据寄存器 DFSDM_FLTxEXMIN 用 0x7FFFFF（最大正值）进行填充，极值检测器最大值寄存器 DFSDM_FLTxEXMAX 用 0x800000（最小负值）进行填充。

在完成右移位和偏移数据校正之后，极值检测器才进行比较。对于每个极值检测器，在（DFSDM_FLTxCR2 寄存器中的）EXCH[7:0] 位中选择计算极值时所涉及的输入通道。

30.4.13 数据单元模块

数据单元模块是整个处理路径的最后一个模块：外部 ΣΔ 调制器 —— 串行收发器 —— Sinc 滤波器 —— 积分器 —— 数据单元块。

输出数据速率取决于串行数据流速率以及滤波器和积分器设置。最大输出数据速率为：

$$\text{数据速率}[\text{采样数/秒}] = \frac{f_{CKIN}}{F_{OSR} \cdot (I_{OSR} - 1 + F_{ORD}) + (F_{ORD} + 1)} \quad \text{...FAST = 0，Sincx 滤波器}$$

$$\text{数据速率}[\text{采样数/秒}] = \frac{f_{CKIN}}{F_{OSR} \cdot (I_{OSR} - 1 + 4) + (2 + 1)} \quad \text{...FAST = 0，FastSinc 滤波器}$$

或

$$\text{数据速率}[\text{采样数/秒}] = \frac{f_{CKIN}}{F_{OSR} \cdot I_{OSR}} \quad \text{...FAST = 1}$$

并行数据输入时的最大输出数据速率：

$$\text{数据速率}[\text{采样数/秒}] = \frac{f_{DATAIN_RATE}}{F_{OSR} \cdot (I_{OSR} - 1 + F_{ORD}) + (F_{ORD} + 1)} \quad \text{...FAST = 0，Sincx 滤波器}$$

或

$$\text{数据速率}[\text{采样数/秒}] = \frac{f_{DATAIN_RATE}}{F_{OSR} \cdot (I_{OSR} - 1 + 4) + (2 + 1)} \quad \text{...FAST = 0，FastSinc 滤波器}$$

或

$$\text{数据速率}[\text{采样数/秒}] = \frac{f_{DATAIN_RATE}}{F_{OSR} \cdot I_{OSR}} \quad \text{...FAST= 1 或任一旁路情形 } (F_{OSR} = 1)$$

其中：f_{DATAIN_RATE}...ADC 或 CPU/DMA 的输入数据速率

在该模块中执行最终数据的右移位，因为最终数据宽度为 24 位，而来自处理路径的数据最高可为 32 位。对于每个所选输入通道，此右移位均可在 0-31 位的范围内进行配置（请参见 DFSDM_CHyCFGR2 寄存器的 DTRBS[4:0] 位）。右移位将结果四舍五入为最接近的整数值。移位结果的符号保留，以便得到有效 24 位有符号格式的结果数据。

接下来对结果进行偏移校正。从给定通道的输出数据中减去偏移校正值（存储在寄存器 DFSDM_CHyCFGR2 中的 OFFSET[23:0]）。OFFSET[23:0] 字段中的数据由软件通过相应的校准程序进行设置。

因数字处理过程中的所有操作均在 32 位有符号寄存器上执行，因此为保证结果不会发生溢出，必须满足以下条件：

$FOSR^{FORD} \cdot IOSR <= 2^{31}$ ... 适用于 $Sinc^x$ 滤波器 (x = 1..5)
$2 \cdot FOSR^2 \cdot IOSR <= 2^{31}$ ... 适用于 FastSinc 滤波器

注： *滤波器和积分器旁路时（IOSR[7:0] = 0，FOSR[9:0] = 0），输入数据速率 (f_{DATAIN_RATE}) 必须被限制为能够读取所有输出数据：*
$f_{DATAIN_RATE} \leq f_{APB}$
其中，f_{APB} 为 DFSDM 外设连接的总线频率。

30.4.14 有符号数据格式

每个 DFSDM 输入串行通道均可连接至一个外部 ΣΔ 调制器。一个外部 ΣΔ 调制器可配有 2 个差分输入（正负），这两个输入可用于差分或单端信号测量。

始终假定 ΣΔ 调制器输出采用有符号格式（来自 ΣΔ 调制器的 0/1 数据流表示值 -1 和 +1）。

寄存器的有符号数据格式：在最终输出数据、模拟看门狗、极值检测器和偏移校正的寄存器中的数据是有符号格式的。输出数据字的 msb 表示值的符号（二进制补码格式）。

30.4.15 启动转换

注入转换可使用以下方法启动：

- 软件：向 DFSDM_FLTxCR1 寄存器中的 JSWSTART 写入“1”。
- 触发：JEXTSEL[4:0] 在 JEXTEN 激活触发时选择触发信号，同时选择有效边沿（请参见 DFSDM_FLTxCR1 寄存器）。
- 如果 JSYNC = 1，则使用 DFSDM_FLT0 同步启动：对于 DFSDM_FLTx (x > 0)，注入转换会在 DFSDM_FLT0 中自动启动；注入转换由软件启动（DFSDM_FLT0CR2 寄存器中的 JSWSTART = 1）。DFSDM_FLTx (x > 0) 中的每个注入转换都始终根据其本地配置设置（JSCAN 和 JCHG 等）执行。

如果使能扫描转换（位 JSCAN = 1），则每次触发注入转换时，注入组（DFSDM_FLTxJCHGR 寄存器中的 JCHG[7:0] 位）的全部所选通道将从最低通道（通道 0，如果已选择）开始，被依次转换。

如果禁止扫描转换（位 JSCAN = 0），则每次触发注入转换时，只会转换注入组（DFSDM_FLTxJCHGR 寄存器中的 JCHG[7:0] 位）的所选通道之一，通道选择随即会移至下一通道。JSCAN = 0 时，对 JCHG[7:0] 位进行写操作会将通道选择置为选择的最低注入通道。

在给定时间内只能进行一次注入转换。因此，如果已发出但尚未完成某个注入转换请求，则会忽略启动注入转换的任何请求。

常规转换可使用以下方法启动：

- 软件：向 DFSDM_FLTxCR1 寄存器中的 RSWSTART 写入“1”。
- 如果 RSYNC = 1，则使用 DFSDM_FLT0 同步启动：对于 DFSDM_FLTx (x > 0)，常规转换会在 DFSDM_FLT0 中自动启动；常规转换由软件启动（DFSDM_FLT0CR2 寄存器中的 RSWSTART = 1）。DFSDM_FLTx (x > 0) 中的每个常规转换都始终根据其本地配置设置（RCONT 和 RCH 等）执行。

在给定时间内只能有一个常规转换处于待处理或正在执行状态。因此，如果已发出但尚未完成某个常规转换请求，则会忽略启动常规转换的任何请求。如果常规转换被注入转换中断，或者常规转换在进行注入转换的过程中启动，则常规转换处于待处理状态。此待处理状态的常规转换随即被延迟，并在完成所有注入转换时执行。延迟的常规转换通过 DFSDM_FLTxRDATAR 寄存器的 RPEND 位进行表示。

30.4.16 连续和快速连续模式

如果将 DFSDM_FLTxCR1 寄存器中的 RCONT 置 1，则会在连续模式下执行常规转换。RCONT = 1 表示向 RSWSTART 写入“1”后，重复转换 RCH[2:0] 所选的通道。

可通过向 RCONT 写入“0”停止在连续模式下执行的常规转换。将 RCONT 清零后，会立即停止正在进行的转换。

在连续模式下，可通过将 DFSDM_FLTxCR1 寄存器的 FAST 位置 1 来提高数据速率。此时，如果对一个通道进行连续转换，则滤波器无需用新的刷新数据重新填充，因为滤波器内部数据有效，这些数据是先前通过对连续数据进行采样而得。速度提升与所选的滤波器阶数有关。通过 RSWSTART = 1 启动连续转换后，将充分进行快速模式 (FAST = 1) 下的首次转换（与 FAST = 0 时相同），之后会在较短的间隔内完成各个后续转换。

连续模式下的转换时间：

FAST = 0 时（或 FAST = 1 时的快速转换）：

对于 $Sinc^x$ 滤波器：

$$t = CNVCNT/f_{DFSDMCLK} = [F_{OSR} * (I_{OSR}-1 + F_{ORD}) + F_{ORD}] / f_{CKIN}$$

对于 FastSinc 滤波器：

$$t = CNVCNT/f_{DFSDMCLK} = [F_{OSR} * (I_{OSR}-1 + 4) + 2] / f_{CKIN}$$

FAST = 1 时（首次转换除外）：

对于 $Sinc^x$ 和 FastSinc 滤波器：

$$t = CNVCNT/f_{DFSDMCLK} = [F_{OSR} * I_{OSR}] / f_{CKIN}$$

F_{OSR} = FOSR[9:0] + 1 = 1 时（滤波器旁路，仅积分器有效）：

$$t = I_{OSR} / f_{CKIN}$$（... 但是 CNVCNT = 0）

连续模式不适用于注入转换。注入转换可通过定时器触发进行启动，以便使用精确时序仿真连续模式。

如果常规连续转换正在进行 (RCONT = 1)，并且如果对请求常规连续转换 (RCONT = 1) 的 DFSDM_FLTxCR1 寄存器进行写访问，则会从下一转换周期重新启动常规连续转换（类似于新的常规连续转换应用于新通道选择，即使 DFSDM_FLTxCR1 寄存器中无更改，亦如此）。

30.4.17 请求优先级

注入转换的优先级高于常规转换。正在进行的常规转换会立即被注入转换请求中断；在注入转换完成后将重新启动此常规转换。

如果某个注入转换处于待处理状态或正在进行，则不能启动其他注入转换：只要（DFSDM_FLTxISR 寄存器中）位 JCIP 为“1”，就会忽略启动注入转换（通过 JSWSTART 或触发）的请求。

同理，如果某个常规转换处于待处理状态或正在进行，则不能启动其他常规转换：只要（DFSDM_FLTxISR 寄存器中）位 RCIP 为“1”，就会忽略启动常规转换（使用 RSWSTART）的请求。

不过，如果在进行某个常规转换时请求注入转换，则常规转换会立即停止，并会启动注入转换。常规转换之后会重新启动，并且此延迟的重新启动会通过位 RPEND 进行表示。

注入转换的优先级高于常规转换，因为注入转换可暂时中断连续的常规转换序列。完成注入转换序列时，如果 RCONT 仍置 1 则会重新启动连续常规转换（RPEND 位将发出基于首次常规转换结果的延迟启动信号）。

通过对 DFSDM 进行相同的写操作启动相关操作时，或者，如果在其他操作结束时有多个操作处于待处理状态，则优先级也至关重要。例如，假设注入转换正在进行 (JCIP = 1)，则对 DFSDM_FLTxCR1 进行单次写操作将向 RSWSTART 写入“1”，以请求常规转换。完成注入序列时，优先级指示接下来执行常规转换，并且通过 RPEND 位指示相应的延迟启动。

30.4.18 在运行模式下的功耗优化

为降低功耗，DFSDM 滤波器和积分器会在未用于转换时自动进入空闲状态（RCIP = 0，JCIP = 0）。

30.5 DFSDM 中断

为提升 CPU 性能，已实现了一组与 CPU 事件发生相关的中断：

- 注入转换结束中断：
  - 由 DFSDM_FLTxCR2 寄存器中的 JEOCIE 位使能
  - 由 DFSDM_FLTxISR 寄存器中的 JEOCF 位指示
  - 通过读取 DFSDM_FLTxJDATAR 寄存器（注入数据）清除
  - 指示哪一个通道发生了转换结束，并在 DFSDM_FLTxJDATAR 寄存器的 JDATACH[2:0] 位中进行报告
- 常规转换结束中断：
  - 由 DFSDM_FLTxCR2 寄存器中的 REOCIE 位使能
  - 由 DFSDM_FLTxISR 寄存器中的 REOCF 位指示
  - 通过读取 DFSDM_FLTxRDATAR 寄存器（常规数据）清除
  - 指示哪一个通道发生了转换结束，并在 DFSDM_FLTxRDATAR 寄存器的 RDATACH[2:0] 位中进行报告

- 注入转换数据溢出中断：
  - 当注入的转换数据未从 DFSDM_FLTxJDATAR 寄存器中读取 （通过 CPU 或 DMA），并被新注入转换覆盖时，会发生此中断
  - 由 DFSDM_FLTxCR2 寄存器中的 JOVRIE 位使能
  - 由 DFSDM_FLTxISR 寄存器中的 JOVRF 位指示
  - 通过向 DFSDM_FLTxICR 寄存器中的 CLRJOVRF 位写入“1”进行清除
- 常规转换数据溢出中断：
  - 当常规转换数据未从 DFSDM_FLTxRDATAR 寄存器中读取（通过 CPU 或 DMA），并被新常规转换覆盖时，会发生此中断
  - 由 DFSDM_FLTxCR2 寄存器中的 ROVRIE 位使能
  - 由 DFSDM_FLTxISR 寄存器中的 ROVRF 位指示
  - 通过向 DFSDM_FLTxICR 寄存器中的 CLRROVRF 位写入“1”进行清除
- 模拟看门狗中断：
  - 当转换数据（输出数据或模拟看门狗滤波器中的数据——具体取决于 DFSDM_FLTxCR1 寄存器中的 AWFSEL 位设置）超出 DFSDM_FLTxAWHTR/ DFSDM_FLTxAWLTR 寄存器中的阈值上限或低于相应阈值下限时，会发生此中断
  - 通过（所选通道 AWDCH[7:0] 的）DFSDM_FLTxCR2 寄存器 AWDIE 位使能
  - 由 DFSDM_FLTxISR 寄存器中的 AWDF 位指示
  - 通过 DFSDM_FLTxAWSR 寄存器中的 AWHTF[7:0] 和 AWLTF[7:0] 字段单独指示模拟看门狗阈值上限或下限错误
  - 通过向 DFSDM_FLTxAWCFR 寄存器中的相应 CLRAWHTF[7:0] 或 CLRAWLTF[7:0] 位写入“1”进行清除
- 短路检测器中断：
  - 稳定的数据数超出 DFSDM_CHyAWSCDR 寄存器中的阈值时会发生此中断
  - 通过（通过 DFSDM_CHyCFGR1 寄存器 SCDEN 位选择的通道的） DFSDM_FLTxCR2 寄存器的 SCDIE 位使能
  - 由 DFSDM_FLTxISR 寄存器的 SCDF[7:0] 位指示（还会报告发生短路检测器事件的通道）
  - 通过向 DFSDM_FLTxICR 寄存器中的相应 CLRSCDF[7:0] 位写入“1”进行清除
- 通道时钟缺失中断：
  - CKINy 引脚上出现时钟缺失时会发生此中断（请参见*第 30.4.4 节：串行通道收发器的时钟缺失检测*）
  - 通过 （通过 DFSDM_CHyCFGR1 寄存器 CKABEN 位选择的通道的） DFSDM_FLTxCR2 寄存器的 CKABIE 位使能
  - 由 DFSDM_FLTxISR 寄存器中的 CKABF[y] 位指示
  - 通过向 DFSDM_FLTxICR 寄存器中的 CLRCKABF[y] 位写入“1”进行清除

表 235. DFSDM 中断请求

中断事件	事件标志	事件/中断清除方法	中断使能控制位
注入转换结束	JEOCF	读取 DFSDM_FLTxJDATAR	JEOCIE
常规转换结束	REOCF	读取 DFSDM_FLTxRDATAR	REOCIE
注入数据溢出	JOVRF	写入 CLRJOVRF = 1	JOVRIE
常规数据溢出	ROVRF	写入 CLRROVRF = 1	ROVRIE
模拟看门狗	AWDF、AWHTF[7:0]、AWLTF[7:0]	写入 CLRAWHTF[7:0] = 1 写入 CLRAWLTF[7:0] = 1	AWDIE、(AWDCH[7:0])
短路检测器	SCDF[7:0]	写入 CLRSCDF[7:0] = 1	SCDIE、(SCDEN)
通道时钟缺失	CKABF[7:0]	写入 CLRCKABF[7:0] = 1	CKABIE、(CKABEN)

30.6 DFSDM DMA 传输

为降低 CPU 干预，可使用 DMA 传输将转换传输至存储器。注入转换的 DMA 传输通过将 DFSDM_FLTxCR1 寄存器的 JDMAEN 位置 1 使能。常规转换的 DMA 传输通过将 DFSDM_FLTxCR1 寄存器的 RDMAEN 位置 1 使能。

注： *通过 DMA 传输，中断标志在注入或常规转换结束时自动清零（DFSDM_FLTxISR 寄存器中的 JEOCF 或 REOCF 位），因为 DMA 正在读取 DFSDM_FLTxJDATAR 或 DFSDM_FLTxRDATAR 寄存器。*

30.7 DFSDM 通道 y 寄存器 (y=0..7)

30.7.1 DFSDM 通道配置 y 寄存器 (DFSDM_CHyCFGR1) (y=0..7)

DFSDM channel configuration y register

该寄存器指定通道 y (y = 0..7) 所使用的参数。

偏移地址：0x00 + 0x20 * y

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DFSDMEN	CKOUTSRC	Res.	Res.	Res.	Res.	Res.	Res.	CKOUTDIV[7:0]							
rw	rw							rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATPACK[1:0]		DATMPX[1:0]		Res.	Res.	Res.	CHINSEL	CHEN	CKABEN	SCDEN	Res.	SPICKSEL[1:0]		SITP[1:0]	
rw	rw	rw	rw				rw	rw	rw	rw		rw	rw	rw	rw

位 31 **DFSDMEN**：DFSDM 接口全局使能 (Global enable for DFSDM interface)

0：禁止 DFSDM 接口

1：使能 DFSDM 接口

如果已使能 DFSDM 接口，则该接口将根据使能的 y 通道和使能的 x 滤波器设置（DFSDM_CHyCFGR1 的 CHEN 位和 DFSDM_FLTxCR1 的 DFEN 位）进行工作。通过设置 DFSDMEN = 0 清除数据：

– 所有寄存器 DFSDM_FLTxISR 均被设为复位状态 (x = 0..3)

– 所有寄存器 DFSDM_FLTxAWSR 均被设为复位状态 (x = 0..3)

注： DFSDMEN 仅存在于 DFSDM_CH0CFGR1 寄存器（通道 y = 0）中。

位 30 **CKOUTSRC**：输出串行时钟源选择 (Output serial clock source selection)

0：输出时钟的源来自系统时钟

1：输出时钟的源来自音频时钟

只有在（DFSDM_CH0CFGR1 寄存器中）DFSDMEN = 0 时，才能修改该值。

注： CKOUTSRC 仅存在于 DFSDM_CH0CFGR1 寄存器（通道 y = 0）中。

位 29:24 保留，必须保持复位值。

位 23:16 **CKOUTDIV[7:0]**：输出串行时钟分频器 (Output serial clock divider)

0：禁止输出时钟生成（CKOUT 信号被设为低电平状态）

1 - 255：为 CKOUT 信号定义串行时钟输出的系统时钟分频，范围为 2 - 256（分频器值 = CKOUTDIV + 1）。

CKOUTDIV 还定义了时钟缺失检测的阈值。

只有在（DFSDM_CH0CFGR1 寄存器中）DFSDMEN = 0 时，才能修改该值。

如果 DFSDMEN = 0（DFSDM_CH0CFGR1 寄存器中），则 CKOUT 信号被置为低电平状态（在 DFSDMEN = 0 后的一个 DFSDM 时钟周期内执行设置）。

注： CKOUTDIV 仅存在于 DFSDM_CH0CFGR1 寄存器（通道 y = 0）中。

位 15:14 **DATPACK[1:0]**：DFSDM_CHyDATINR 寄存器数据封装模式 (Data packing mode in DFSDM_CHyDATINR register)

0：标准：DFSDM_CHyDATINR 寄存器中的输入数据仅存储在 INDAT0[15:0] 中。要清空 DFSDM_CHyDATINR 寄存器，必须由 DFSDM 滤波器从通道 y 中读取一个采样。

1：交错：DFSDM_CHyDATINR 寄存器中的输入数据存储为两个采样：

– 第一个采样位于 INDAT0[15:0] 中（已分配至通道 y）

– 第二个采样位于 INDAT1[15:0] 中（已分配至通道 y）

要清空 DFSDM_CHyDATINR 寄存器，必须由数字滤波器从通道 y 中读取两个采样（INDAT0[15:0] 部分被读为第一个采样，而 INDAT1[15:0] 部分则被读为下一个采样）。

2：双通道：DFSDM_CHyDATINR 寄存器中的输入数据存储为两个采样：

– 第一个采样位于 INDAT0[15:0] 中（已分配至通道 y）

– 第二个样采样于 INDAT1[15:0] 中（已分配至通道 y+1）

要清空 DFSDM_CHyDATINR 寄存器，必须由数字滤波器从通道 y 中读取第一个样本，且必须由另一数字滤波器从通道 y+1 中读取第二个样本。只有在通道数为偶数（y = 0、2、4、6）时才支持双通道模式，对于通道数为奇数（y = 1、3、5、7）的情况，DFSDM_CHyDATINR 被写保护。如果偶数通道被设为双通道模式，则下一奇数通道必须进入标准模式 (DATPACK[1:0] = 0)，以便与偶数通道正确配合。

3：保留

只有在（DFSDM_CHyCFGR1 寄存器中）CHEN = 0时，才能修改该值。

位 13:12 **DATMPX[1:0]**：通道 y 输入数据复用器 (Input data multiplexer for channel y)

0：通道 y 输入数据取自外部串行输入，为 1 位值。DFSDM_CHyDATINR 寄存器被写保护。

1：通道 y 输入数据取自内部模数转换器 ADC_{y+1} 输出寄存器更新，为 16 位值（若 ADC_{y+1} 存在）。ADC 输出数据被写入 DFSDM_CHyDATINR 寄存器的 INDAT0[15:0] 部分。

2：通道 y 输入数据取自内部 DFSDM_CHyDATINR 寄存器（通过 CPU/DMA 直接写操作实现）。可写入一个或两个 16 位数据采样，具体取决于 DATPACK[1:0] 位域设置。

3：保留。

只有在（DFSDM_CHyCFGR1 寄存器中）CHEN = 0时，才能修改该值。

注： DATMPX[1:0] = 1 仅适用于 ADC1 和 ADC2。

位 11:9 保留，必须保持复位值。

位 8 **CHINSEL**：通道输入选择 (Channel inputs selection)

0：通道输入取自同一通道 y 的引脚。

1：通道输入取自下一通道（通道 (y+1) 取 8 的模）的引脚。

只有在（DFSDM_CHyCFGR1 寄存器中）CHEN = 0时，才能修改该值。

位 7 **CHEN**：通道 y 使能 (Channel y enable)

0：禁止通道 y

1：使能通道 y

如果使能了通道 y，则会根据给定通道设置开始接收串行数据。

位 6 **CKABEN**：通道 y 时钟缺失检测器使能 (Clock absence detector enable on channel y)

0：禁止通道 y 时钟缺失检测器

1：使能通道 y 时钟缺失检测器

位 5 **SCDEN**：通道 y 短路检测器使能 (Short-circuit detector enable on channel y)

0：输入通道 y 将不受短路检测器保护

1：输入通道 y 将持续受短路检测器保护

位 4 保留，必须保持复位值。

位 3:2 **SPICKSEL[1:0]**：通道 y SPI 时钟选择 (SPI clock select for channel y)

0：时钟来自外部 CKINy 输入——基于 SITP[1:0] 的采样点

1：时钟来自内部 CKOUT 输出——基于 SITP[1:0] 的采样点

2：时钟来自内部 CKOUT——在 CKOUT 每个第二个下降沿的采样点。
用于连接外部 ΣΔ 调制器，调制器将时钟输入（来自 CKOUT）除以 2，来生成输出串行通信时钟（该输出时钟变化
在各个时钟输入上升沿有效）。

3：时钟来自内部 CKOUT 输出——在 CKOUT 每个第二个上升沿的采样点。
用于连接外部 ΣΔ 调制器，调制器将时钟输入（来自 CKOUT）除以 2，来生成输出串行通信时钟（该输出时钟变化在各个时钟输入下降沿有效）。
只有在（DFSDM_CHyCFGR1 寄存器中）CHEN = 0时，才能修改该值。

位 1:0 **SITP[1:0]**：通道 y 串行接口类型 (Serial interface type for channel y)

00：上升沿选通数据 SPI

01：下降沿选通数据 SPI

10：DATINy 引脚曼彻斯特编码输入：上升沿 = 逻辑 0，下降沿 = 逻辑 1

11：DATINy 引脚曼彻斯特编码输入：上升沿 = 逻辑 1，下降沿 = 逻辑 0

只有在（DFSDM_CHyCFGR1 寄存器中的）CHEN = 0 时，才能修改该值。

30.7.2 DFSDM 通道配置 y 寄存器 (DFSDM_CHyCFGR2) (y=0..7)

DFSDM channel configuration y register

该寄存器指定通道 y (y = 0..7) 所使用的参数。

偏移地址：0x04 + 0x20 * y

复位值：0x0000 0000

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

位 31:8 **OFFSET[23:0]**：通道 y 24 位校准偏移 (24-bit calibration offset for channel y)

对于通道 y 的每一次转换结果，都执行 OFFSET。
该值由软件设置。

位 7:3 **DTRBS[4:0]**：通道 y 数据右移位 (Data right bit-shift for channel y)

0-31：定义积分器输出数据结果的移位——将向右移多少位以获得最终结果。在进行偏移校正之前执行移位。数据移位将结果四舍五入为最接近的整数值。移位结果的符号保留（以便得到有效 24 位有符号格式的结果数据）。
只有在（DFSDM_CHyCFGR1 寄存器中）CHEN = 0时，才能修改该值。

位 2:0 保留，必须保持复位值。

30.7.3 DFSDM 通道模拟看门狗和短路检测器寄存器 (DFSDM_CHyAWSCDR) (y = 0..7)

DFSDM channel analog watchdog and short-circuit detector register

通道 y (y = 0..7) 短路检测器和模拟看门狗设置

偏移地址：0x08 + 0x20 * y

复位值：0x0000 0000

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

位 31:24 保留，必须保持复位值。

位 23:22 **AWFORD[1:0]**：通道 y 模拟看门狗 Sinc 滤波器阶数 (Analog watchdog Sinc filter order on channel y)

0：FastSinc 滤波器类型
1：$Sinc^1$ 滤波器类型
2：$Sinc^2$ 滤波器类型
3：$Sinc^3$ 滤波器类型

$Sinc^x$ 滤波器类型传递函数：

$$H(z) = \left(\frac{1 - z^{-FOSR}}{1 - z^{-1}}\right)^x$$

FastSinc 滤波器类型传递函数：

$$H(z) = \left(\frac{1 - z^{-FOSR}}{1 - z^{-1}}\right)^2 \cdot (1 + z^{-(2 \cdot FOSR)})$$

只有在（DFSDM_CHyCFGR1 寄存器中）CHEN = 0时，才能修改该位。

位 21 保留，必须保持复位值。

位 20:16 **AWFOSR[4:0]**：通道 y 模拟看门狗滤波器过采样率（抽取率）(Analog watchdog filter oversampling ratio (decimation rate) on channel y)

0 - 31：定义 Sinc 类型滤波器的长度，长度范围为 1 - 32 (AWFOSR + 1)。该数字还表示模拟数据速率的抽取率。

只有在（DFSDM_CHyCFGR1 寄存器中）CHEN = 0时，才能修改该位。

注： 如果 AWFOSR = 0，则滤波器无影响（滤波器旁路）。

位 15:12 **BKSCD[3:0]**：通道 y 短路检测器断路信号分配 (Break signal assignment for short-circuit detector on channel y)

BKSCD[i] = 0：断路 i 信号未分配至通道 y 短路检测器
BKSCD[i] = 1：断路 i 信号已分配至通道 y 短路检测器

位 11:8 保留，必须保持复位值。

位 7:0 **SCDT[7:0]**：通道 y 的短路检测器阈值 (short-circuit detector threshold for channel y)

这些位可由软件写入，用来定义短路检测器的计数器阈值。如果达到该值，则在给定通道上发生短路检测器事件。

30.7.4 DFSDM 通道看门狗滤波器数据寄存器 (DFSDM_CHyWDATR) (y=0..7)

DFSDM channel watchdog filter data register

该寄存器包含的数据来自于与输入通道 y (y = 0..7) 关联的模拟看门狗滤波器。

偏移地址：0x0C + 0x20 * y

复位值：0x0000 0000

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

位 31:16 保留，必须保持复位值。

位 15:0 **WDATA[15:0]**：输入通道 y 看门狗数据 (Input channel y watchdog data)

由输入通道 y 模拟看门狗滤波器转换的数据。该通道进行连续的数据转换（无触发），转换分辨率受一定限制（OSR=1...32/sinc 阶数 = 1...3）。

30.7.5 DFSDM 通道数据输入寄存器 (DFSDM_CHyDATINR) (y=0..7)

DFSDM channel data input register

该寄存器包含的 16 位输入数据将通过 DFSDM 滤波器模块进行处理。

偏移地址：0x10 + 0x20 * y

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
INDAT1[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INDAT0[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:16 **INDAT1[15:0]**：通道 y 或通道 y+1 的输入数据 (Input data for channel y or channel y+1)

当 DATMPX[1:0] = 1 或 DATMPX[1:0] = 2 时，输入并行通道数据将通过数字滤波器进行处理。
数据可由 CPU/DMA 写入（当 DATMPX[1:0] = 2 时），也可以由内部 ADC 直接写入（当 DATMPX[1:0] = 1 时）。
当 DATPACK[1:0] = 0（标准模式）时
INDAT0[15:0] 被写保护（不用于输入采样）。
当 DATPACK[1:0] = 1（交错模式）时
第二个通道 y 数据采样存储到 INDAT1[15:0] 中。第一个通道 y 数据采样存储到 INDAT0[15:0] 中。
DFSDM_FLTx 滤波器依次读取该两个采样，作为两个通道 y 数据采样。
当 DATPACK[1:0] = 2（双模式）时
对于偶数 y 通道：INDAT1[15:0] 中的采样自动复制到通道 (y+1) 的 INDAT0[15:0] 中。
对于奇数 y 通道：INDAT1[15:0] 被写保护。
请参见*第 30.4.6 节：并行数据输入*以获得更多信息。
INDAT0[15:1] 采用 16 位带符号格式。

位 15:0 **INDAT0[15:0]**：通道 y 输入数据 (Input data for channel y)

当 DATMPX[1:0] = 1 或 DATMPX[1:0] = 2 时，输入并行通道数据将通过数字滤波器进行处理。
数据可由 CPU/DMA 写入（当 DATMPX[1:0] = 2 时），也可以由内部 ADC 直接写入（当 DATMPX[1:0] = 1 时）。
当 DATPACK[1:0] = 0（标准模式）时
通道 y 数据采样存储到 INDAT0[15:0] 中。
当 DATPACK[1:0] = 1（交错模式）时
第一个通道 y 数据采样存储到 INDAT0[15:0] 中。第二个通道 y 数据采样存储到 INDAT1[15:0] 中。
DFSDM_FLTx 滤波器依次读取该两个采样，作为两个通道 y 数据采样。
当 DATPACK[1:0] = 2（双模式）时
对于偶数 y 通道：通道 y 数据采样存储到 INDAT0[15:0] 中。
对于奇数 y 通道：INDAT0[15:0] 被写保护。
请参见*第 30.4.6 节：并行数据输入*以获得更多信息。
INDAT0[15:0] 采用 16 位带符号格式。

30.8 DFSDM 滤波器 x 模块寄存器 (x=0..3)

30.8.1 DFSDM 控制寄存器 1 (DFSDM_FLTxCR1)

DFSDM control register 1

偏移地址：0x100 + 0x80 * x，x = 0...3

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	AWFSEL	FAST	Res.	Res.	RCH[2:0]			Res.	Res.	RDMAEN	Res.	RSYNC	RCONT	RSWSTART	Res.
	rw	rw			rw	rw	rw			rw		rw	rw	r0w	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	JEXTEN[1:0]		JEXTSEL[4:0]					Res.	Res.	JDMAEN	JSCAN	JSYNC	Res.	JSWSTART	DFEN
	rw	rw	rw	rw	rw	rw	rw			rw	rw	rw		r0w	rw

位 31 保留，必须保持复位值。

位 30 **AWFSEL**：模拟看门狗快速模式选择 (Analog watchdog fast mode select)

0：数据输出值的模拟看门狗（在数字滤波器之后）。在偏移校正和移位后进行比较

1：通道收发器值的模拟看门狗（在看门狗滤波器之后）

位 29 **FAST**：常规转换的快速转换模式选择 (Fast conversion mode selection for regular conversions)

0：禁止快速转换模式

1：使能快速转换模式

在连续模式下进行常规转换时，如果使能了快速模式，则每次转换（第一次转换除外）都比在标准模式下的转换执行得快。该位对于不连续的转换没有影响。

只有在 DFEN = 0 (DFSDM_FLTxCR1) 时才能修改该位。

如果 FAST=0（或者 FAST=1 连续模式下的第一次转换），则：

$t = [F_{OSR} * (I_{OSR}-1 + F_{ORD}) + F_{ORD}] / f_{CKIN}$.....（适用于 $Sinc^x$ 滤波器）

$t = [F_{OSR} * (I_{OSR}-1 + 4) + 2] / f_{CKIN}$.....（适用于 FastSinc 滤波器）

如果在连续模式下 FAST=1（第一次转换除外），则：

$t = [F_{OSR} * I_{OSR}] / f_{CKIN}$

如果 $F_{OSR} = F_{OSR}[9:0] + 1 = 1$ 时（滤波器旁路，仅积分器有效），则：

$t = I_{OSR} / f_{CKIN}$（... 但是 CNVCNT = 0）

其中，f_{CKIN} 为（给定通道 CKINy 引脚上的）通道输入时钟频率，在并行数据输入的情况下，其表示输入数据速率。

位 28:27 保留，必须保持复位值。

位 26:24 **RCH[2:0]**：常规通道选择 (Regular channel selection)

0：将通道 0 选作常规通道

1：将通道 1 选作常规通道

...

7：将通道 7 选作常规通道

当 RCIP = 1 时写入这些位，并在下一个常规转换开始时生效。这在连续模式下（当 RCONT = 1 时）特别有用。它还会影响待处理的常规转换（因正在进行注入转换而暂停）。

位 23:22 保留，必须保持复位值。

位 21 **RDMAEN**：使能 DMA 通道以读取常规转换数据 (DMA channel enabled to read data for the regular conversion)

0：未使能 DMA 通道来读取常规数据
1：已使能 DMA 通道来读取常规数据
只有在 DFEN = 0 (DFSDM_FLTxCR1) 时才能修改该位。

位 20 保留，必须保持复位值。

位 19 **RSYNC**：使用 DFSDM_FLT0 同步启动常规转换 (Launch regular conversion synchronously with DFSDM_FLT0)

0：不使用 DFSDM_FLT0 同步启动常规转换
1：在 DFSDM_FLT0 中启动常规转换的同时，在该 DFSDM_FLTx 中启动常规转换
只有在 DFEN = 0 (DFSDM_FLTxCR1) 时才能修改该位。

位 18 **RCONT**：常规转换的连续模式选择 (Continuous mode selection for regular conversions)

0：对于每个转换请求，只转换一次常规通道
1：在发出每个转换请求后，重复转换常规通道
在连续常规转换进行过程中，向该位写入“0”将立即停止连续模式。

位 17 **RSWSTART**：软件启动常规通道转换 (Software start of a conversion on the regular channel)

0：写入“0”无影响
1：写入“1”会发出启动常规通道转换的请求，并会使 RCIP 变为“1”。如果已经设置 RCIP=1，则对 RSWSTART 进行的写操作无效。如果 RSYNC = 1，则写入“1”无效。
此位始终读为“0”。

位 16:15 保留，必须保持复位值。

位 14:13 **JEXTEN[1:0]**：注入转换触发使能和触发边沿选择 (Trigger enable and trigger edge selection for injected conversions)

00：禁止触发检测
01：所选触发的每个上升沿都会发出启动注入转换的请求
10：所选触发的每个下降沿都会发出启动注入转换的请求
11：所选触发的上升沿和下降沿都会发出启动注入转换的请求
只有在 DFEN = 0 (DFSDM_FLTxCR1) 时才能修改该位。

位 12:8 **JEXTSEL[4:0]**：启动注入转换的触发信号选择 (Trigger signal selection for launching injected conversions)

0x0-0x1F：根据下表选择的触发输入（内部或外部触发）。
只有在 DFEN = 0 (DFSDM_FLTxCR1) 时才能修改该位。

注： 同步触发的延迟最大为一个 $f_{DFSDMCLK}$ 时钟周期（存在确定的抖动），异步触发的延迟为 2-3 个 $f_{DFSDMCLK}$ 时钟周期（抖动长达 1 个周期）。

	DFSDM_FLT0	DFSDM_FLT1	DFSDM_FLT2	DFSDM_FLT3
0x00	dfsdm_jtrg0	dfsdm_jtrg0	dfsdm_jtrg0	dfsdm_jtrg0
0x01	dfsdm_jtrg1	dfsdm_jtrg1	dfsdm_jtrg1	dfsdm_jtrg1
...				
0x1E	dfsdm_jtrg30	dfsdm_jtrg30	dfsdm_jtrg30	dfsdm_jtrg30
0x1F	dfsdm_jtrg31	dfsdm_jtrg31	dfsdm_jtrg31	dfsdm_jtrg31

请参见*表 231：DFSDM 触发器连接*。

位 7:6 保留，必须保持复位值。

位 5 **JDMAEN**：使能 DMA 通道以读取注入通道组的数据 (DMA channel enabled to read data for the injected channel group)

0：未使能 DMA 通道来读取注入数据
1：已使能 DMA 通道来读取注入数据
只有在 DFEN = 0 (DFSDM_FLTxCR1) 时才能修改该位。

位 4 **JSCAN**：注入转换的扫描转换模式 (Scanning conversion mode for injected conversions)

0：对注入通道组执行一次通道转换，然后选择该通道组下一个通道。

1：从选择的最低通道开始，对注入通道组执行连续转换。

只有在 DFEN = 0 (DFSDM_FLTxCR1) 时才能修改该位。

如果 JSCAN=0，则对 JCHG 进行写入操作会将通道选择复位为最小所选通道。

位 3 **JSYNC**：使用 DFSDM_FLT0 JSWSTART 触发同步启动注入转换 (Launch an injected conversion synchronously with the DFSDM_FLT0 JSWSTART trigger)

0：不使用 DFSDM_FLT0 同步启动注入转换

1：在 DFSDM_FLT0 中通过 JSWSTART 触发启动注入转换的同时，在该 DFSDM_FLTx 中启动注入转换

只有在 DFEN = 0 (DFSDM_FLTxCR1) 时才能修改该位。

位 2 保留，必须保持复位值。

位 1 **JSWSTART**：启动注入通道组转换 (Start a conversion of the injected group of channels)

0：写入“0”无影响。

1：写入“1”会发出转换注入通道组请求，同时会使 JCIP 变为“1”。如果已经设置 JCIP = 1，则对 JSWSTART 进行的写操作无效。如果 JSYNC = 1，则写入“1”无效。

此位始终读为“0”。

位 0 **DFEN**：DFSDM_FLTx 使能 (DFSDM_FLTx enable)

0：禁止 DFSDM_FLTx。给定 DFSDM_FLTx 的所有转换均立即停止，DFSDM_FLTx 的所有功能均停止。

1：使能 DFSDM_FLTx。如果使能了 DFSDM_FLTx，则 DFSDM_FLTx 会根据其设置开始工作。

通过设置 DFEN = 0 清零的数据：

– 寄存器 DFSDM_FLTxISR 被设为复位状态

– 寄存器 DFSDM_FLTxAWSR 被设为复位状态

30.8.2 DFSDM 控制寄存器 2 (DFSDM_FLTxCR2)

DFSDM control register 2

偏移地址：0x104 + 0x80 * x，x = 0...3

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	AWDCH[7:0]							
								rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EXCH[7:0]								Res.	CKABIE	SCDIE	AWDIE	ROVRIE	JOVRIE	REOCIE	JEOCIE
rw	rw	rw	rw	rw	rw	rw	rw		rw	rw	rw	rw	rw	rw	rw

位 31:24 保留，必须保持复位值。

位 23:16 **AWDCH[7:0]**：模拟看门狗通道选择 (Analog watchdog channel selection)
这些位用于选择由模拟看门狗持续监控的输入通道
AWDCH[y] = 0：在通道 y 上禁止模拟看门狗
AWDCH[y] = 1：在通道 y 上使能模拟看门狗

位 15:8 **EXCH[7:0]**：极值检测器通道选择 (Extremes detector channel selection)
这些位选择极值检测器要采用的输入通道
EXCH[y] = 0：极值检测器不接受通道 y 的数据
EXCH[y] = 1：极值检测器接受通道 y 的数据

位 7 保留，必须保持复位值。

位 6 **CKABIE**：时钟缺失中断使能 (Clock absence interrupt enable)
0：禁止检测通道输入时钟缺失中断
1：使能检测通道输入时钟缺失中断
请参见 DFSDM_FLTxISR 中的 CKABF[7:0] 说明。
注： CKABIE 仅存在于 DFSDM_FLT0CR2 寄存器中（滤波器 x = 0）

位 5 **SCDIE**：短路检测器中断使能 (Short-circuit detector interrupt enable)
0：禁止短路检测器中断
1：使能短路检测器中断
请参见 DFSDM_FLTxISR 中的 SCDF[7:0] 说明。
注： SCDIE 仅存在于 DFSDM_FLT0CR2 寄存器中（滤波器 x = 0）

位 4 **AWDIE**：模拟看门狗中断使能 (Analog watchdog interrupt enable)
0：禁止模拟看门狗中断
1：使能模拟看门狗中断
请参见 DFSDM_FLTxISR 中的 AWDF 说明。

位 3 **ROVRIE**：常规数据溢出中断使能 (Regular data overrun interrupt enable)
0：禁止常规数据溢出中断
1：使能常规数据溢出中断
请参见 DFSDM_FLTxISR 中的 ROVRF 说明。

位 2 **JOVRIE**：注入数据溢出中断使能 (Injected data overrun interrupt enable)
0：禁止注入数据溢出中断
1：使能注入数据溢出中断
请参见 DFSDM_FLTxISR 中的 JOVRF 说明。

位 1 **REOCIE**：常规转换结束中断使能 (Regular end of conversion interrupt enable)
0：禁止常规转换结束中断
1：使能常规转换结束中断
请参见 DFSDM_FLTxISR 中的 REOCF 说明。

位 0 **JEOCIE**：注入转换结束中断使能 (Injected end of conversion interrupt enable)
0：禁止注入转换结束中断
1：使能注入转换结束中断
请参见 DFSDM_FLTxISR 中的 JEOCF 说明。

30.8.3 DFSDM 中断和状态寄存器 (DFSDM_FLTxISR)

DFSDM interrupt and status register

偏移地址：0x108 + 0x80 * x，x = 0...3

复位值：0x00FF 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SCDF[7:0]								CKABF[7:0]							
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	RCIP	JCIP	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	AWDF	ROVRF	JOVRF	REOCF	JEOCF
	r	r									r	r	r	r	r

位 31:24 **SCDF[7:0]**：短路检测器标志 (short-circuit detector flag)

SDCF[y] = 0：通道 y 上未发生任何短路检测器事件

SDCF[y] = 1：通道 y 上的短路检测器计数器达到在 DFSDM_CHyAWSCDR 寄存器中编程的值

此位将由硬件置 1，该位可由软件使用 DFSDM_FLTxICR 寄存器中的相应 CLRSCDF[y] 位清零。当 CHEN[y] = 0（给定通道被禁止）时，SCDF[y] 还可以由硬件清零。

注： SCDF[7:0] 仅存在于 DFSDM_FLT0ISR 寄存器中（滤波器 x = 0）

位 23:16 **CKABF[7:0]**：时钟缺失标志 (Clock absence flag)

CKABF[y] = 0：通道 y 上存在时钟信号。

CKABF[y] = 1：通道 y 上不存在时钟信号。

当检测到通道 y 上不存在时钟信号时，给定 y 位由硬件置 1。当 CHEN = 0 时（请参见 DFSDM_CHyCFGR1 寄存器），它由硬件保持为 CKABF[y] = 1 状态。当收发器尚未被同步时，它由硬件保持为 CKABF[y] = 1 状态。它可由软件使用 DFSDM_FLTxICR 寄存器中的相应 CLRCKABF[y] 位清零。

注： CKABF[7:0] 仅存在于 DFSDM_FLT0ISR 寄存器中（滤波器 x = 0）

位 15 保留，必须保持复位值。

位 14 **RCIP**：常规转换正在进行状态 (Regular conversion in progress status)

0：未发出转换常规通道的请求

1：常规通道转换正在进行，或一个常规转换请求正在等待处理

当 RCIP = 1 时，忽略启动常规转换的请求。

位 13 **JCIP**：注入转换正在进行状态 (Injected conversion in progress status)

0：未发出转换注入通道组的请求（软件和触发器均未发出请求）

1：注入通道组转换处于正在进行状态（向 JSWSTART 写入了“1”）或者一个注入转换请求正在等待处理（由于触发检测）

当 JCIP = 1 时，忽略启动注入转换的请求。

位 12:5 保留，必须保持复位值。

位 4 **AWDF**：模拟看门狗 (Analog watchdog)

0：未发生模拟看门狗事件

1：模拟看门狗模块检测到电压超出在 DFSDM_FLTxAWLTR 或 DFSDM_FLTxAWHTR 寄存器中编程的值。

此位将由硬件置 1，它由软件通过以下方式清零：将 DFSDM_FLTxAWSR 寄存器中的所有源标志位 AWHTF[7:0] 和 AWLTF[7:0] 清零（通过向 DFSDM_FLTxAWCFR 寄存器中的清零位写入“1”实现）。

位 3 **ROVRF**：常规转换溢出标志 (Regular conversion overrun flag)

0：未发生常规转换溢出

1：发生了常规转换溢出，也就是说，常规转换已经完成，但 REOCF 仍然为“1”。RDATAR 不受溢出影响

此位将由硬件置 1，该位可由软件使用 DFSDM_FLTxICR 寄存器中的 CLRROVRF 位清零。

位 2 **JOVRF**：注入转换溢出标志 (Injected conversion overrun flag)

0：未发生注入转换溢出

1：发生了注入转换溢出，也就是说，注入转换已经完成，但 JEOCF 仍然为“1”。JDATAR 不受溢出影响

此位将由硬件置 1，该位可由软件使用 DFSDM_FLTxICR 寄存器中的 CLRJOVRF 位清零。

位 1 **REOCF**：常规转换结束标志 (End of regular conversion flag)

0：未完成常规转换

1：已完成常规转换并可以读取其数据

此位将由硬件置 1，当软件或 DMA 读取 DFSDM_FLTxRDATAR 时，该位清零。

位 0 **JEOCF**：注入转换结束标志 (End of injected conversion flag)

0：未完成注入转换

1：已完成注入转换并可以读取其数据

此位将由硬件置 1，当软件或 DMA 读取 DFSDM_FLTxJDATAR 时，该位清零。

注： *对于每个标志位，可通过将 DFSDM_FLTxCR2 中的相应位置 1 来使能中断。如果调用了中断，则必须在退出中断服务程序之前将该标志清零。*

当 DFEN = 0 时，DFSDM_FLTxISR 的所有位都会自动复位。

30.8.4 DFSDM 中断标志清零寄存器 (DFSDM_FLTxICR)

DFSDM interrupt flag clear register

偏移地址：0x10C + 0x80 * x，x = 0...3

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CLRSCDF[7:0]								CLRCKABF[7:0]							
rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CLRR OVRF	CLR J OVRF	Res.	Res.
												rc_w1	rc_w1		

位 31:24 **CLRSCDF[7:0]**：清零短路检测器标志 (Clear the short-circuit detector flag)

CLRSCDF[y] = 0：写入“0”无影响

CLRSCDF[y] = 1：向位置 y 写入“1”会将 DFSDM_FLTxISR 寄存器中的相应 SCDF[y] 位清零

注： CLRSCDF[7:0] 仅存在于 DFSDM_FLT0ICR 寄存器中（滤波器 x = 0）

位 23:16 **CLRCKABF[7:0]**：清零时钟缺失标志 (Clear the clock absence flag)

CLRCKABF[y] = 0：写入“0”无影响

CLRCKABF[y] = 1：向位置 y 写入“1”会将 DFSDM_FLTxISR 寄存器中的相应 CKABF[y] 位清零。当收发器尚未被同步时，时钟缺失标志会置 1，且无法由 CLRCKABF[y] 清零。

注： CLRCKABF[7:0] 仅存在于 DFSDM_FLT0ICR 寄存器中（滤波器 x = 0）

位 15:4 保留，必须保持复位值。

位 3 **CLRROVRF**：将常规转换溢出标志清零 (Clear the regular conversion overrun flag)

0：写入“0”无影响

1：写入“1”会将 DFSDM_FLTxISR 寄存器中的 ROVRF 位清零

位 2 **CLRJOVRF**：将注入转换溢出标志清零 (Clear the injected conversion overrun flag)

0：写入“0”无影响

1：写入“1”会将 DFSDM_FLTxISR 寄存器中的 JOVRF 位清零

位 1:0 保留，必须保持复位值。

注： DFSDM_FLTxICR 的位始终读为“0”。

30.8.5 DFSDM 注入通道组选择寄存器 (DFSDM_FLTxJCHGR)

DFSDM injected channel group selection register

偏移地址：0x110 + 0x80 * x，x = 0...3

复位值：0x0000 0001

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

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

位 31:8 保留，必须保持复位值。

位 7:0 **JCHG[7:0]**：注入通道组选择 (Injected channel group selection)

JCHG[y] = 0：通道 y 不属于注入组

JCHG[y] = 1：通道 y 属于注入组

如果 JSCAN = 1，则依次转换每一个所选通道。首先转换最小通道（通道 0，如果已选择），最后转换最大所选通道。

如果 JSCAN = 0，则只转换其中一个所选通道，然后通道选择移至下一个通道。如果 JSCAN=0，则对 JCHG 进行写入操作会将通道选择复位为最小所选通道。

必须始终至少为注入组选择一个通道。如果写操作导致所有 JCHG 位均变为零，则忽略写操作。

30.8.6 DFSDM 滤波器控制寄存器 (DFSDM_FLTxFCR)

DFSDM filter control register

偏移地址：0x114 + 0x80 * x，x = 0...3

复位值：0x0000 0000

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

位 31:29 **FORD[2:0]**：Sinc 滤波器阶数 (Sinc filter order)

0：FastSinc 滤波器类型

1：Sinc^1 滤波器类型

2：Sinc^2 滤波器类型

3：Sinc^3 滤波器类型

4：Sinc^4 滤波器类型

5：Sinc^5 滤波器类型

6 7：保留

Sinc^x 滤波器类型传递函数：

$$H(z) = \left(\frac{1 - z^{-FOSR}}{1 - z^{-1}}\right)^x$$

FastSinc 滤波器类型传递函数：

$$H(z) = \left(\frac{1 - z^{-FOSR}}{1 - z^{-1}}\right)^2 \cdot (1 + z^{-(2 \cdot FOSR)})$$

只有在 DFEN = 0 (DFSDM_FLTxCR1) 时才能修改该位。

位 28:26 保留，必须保持复位值。

位 25:16 **FOSR[9:0]**：Sinc 滤波器过采样率（抽取率）(Sinc filter oversampling ratio (decimation rate))

0 - 1023：定义 Sinc 类型滤波器的长度，长度范围为 1 - 1024 (F_{OSR} = FOSR[9:0] + 1)。该数字还表示滤波器输出数据速率的抽取率。

只有在 DFEN = 0 (DFSDM_FLTxCR1) 时才能修改该位

注： 如果 FOSR = 0，则滤波器无影响（滤波器旁路）。

位 15:8 保留，必须保持复位值。

位 7:0 **IOSR[7:0]**：积分器过采样率（平均长度）(Integrator oversampling ratio (averaging length))

0 - 255：积分器的长度，长度范围为 1 - 256 (IOSR + 1)。定义一个积分器输出数据采样是由多少个 Sinc 滤波器采样而合成的。积分器的输出数据速率将被减去该数值（额外数据抽取率）。

只有在 DFEN = 0 (DFSDM_FLTxCR1) 时才能修改该位

注： 如果 IOSR = 0，则积分器无影响（积分器旁路）。

30.8.7 注入组的 DFSDM 数据寄存器 (DFSDM_FLTxJDATAR)

DFSDM data register for injected group

偏移地址：0x118 + 0x80 * x，x = 0...3

复位值：0x0000 0000

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

位 31:8 **JDATA[23:0]**：注入组转换数据 (Injected group conversion data)
每次转换完注入组中的一个通道时，得到的数据都会存储在该字段中。JEOCF =1 时数据有效。读取该寄存器会将相应的 JEOCF 清零。

位 7:3 保留，必须保持复位值。

位 2:0 **JDATACH[2:0]**：最近转换的注入通道 (Injected channel most recently converted)
每次转换完注入组中的一个通道时，都会更新 JDATACH[2:0] 以指示转换了哪一个通道。因此，JDATA[23:0] 保持的数据对应于 JDATACH[2:0] 所指示的通道。

注： *可以使用 DMA 从该寄存器中读取数据。可以使用半字访问只读取转换数据的最高有效位 (MSB)。*

读取该寄存器还会将 DFSDM_FLTxISR 的 JEOCF 清零。因此，如果为从该寄存器中读取数据而激活了 DMA，则固件不能读取该寄存器。

30.8.8 常规通道的 DFSDM 数据寄存器 (DFSDM_FLTxRDATAR)

DFSDM data register for the regular channel

偏移地址：0x11C + 0x80 * x，x = 0...3

复位值：0x0000 0000

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

位 31:8 **RDATA[23:0]**：常规通道转换数据 (Regular channel conversion data)
每次完成常规转换时，相应的数据都会存储在该寄存器中。当 REOCF =1 时，数据有效。读取该寄存器会将相应的 REOCF 清零。

位 7:5 保留，必须保持复位值。

位 4 **RPEND**：常规通道待处理数据 (Regular channel pending data)
因在转换期间发生注入通道触发而导致 RDATA[23:0] 中的常规数据被延迟处理

位 3 保留，必须保持复位值。

位 2:0 **RDATACH[2:0]**：最近转换的常规通道 (Regular channel most recently converted)
每次完成常规转换时，都会更新 RDATACH[2:0] 以指示转换了哪一个通道（因为在常规转换期间，可更新 DFSDM_FLTxCR1 寄存器中的常规通道选择 RCH[2:0]）。因此，RDATA[23:0] 保持的数据对应于 RDATACH[2:0] 所指示的通道。

注： *可以使用半字访问只读取转换数据的最高有效位 (MSB)。*
读取该寄存器还会将 DFSDM_FLTxISR 的 REOCF 清零。

30.8.9 DFSDM 模拟看门狗阈值上限寄存器 (DFSDM_FLTxAWHTR)

DFSDM analog watchdog high threshold register

偏移地址：0x120 + 0x80 * x，x = 0...3

复位值：0x0000 0000

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

位 31:8 **AWHT[23:0]**：模拟看门狗阈值上限 (Analog watchdog high threshold)

这些位由软件写入，用于定义模拟看门狗的阈值上限。

注： 对于通道收发器监视器 (AWFSEL = 1)，高 16 位 (AWHT[23:8]) 会定义 16 位阈值，该阈值与模拟看门狗滤波器输出进行比较（因为来自模拟看门狗滤波器的数据的分辨率最高为 16 位）。在这种情况下，不对位 AWHT[7:0] 进行比较。

位 7:4 保留，必须保持复位值。

位 3:0 **BKAWH[3:0]**：模拟看门狗阈值上限事件的断路信号分配 (Break signal assignment to analog watchdog high threshold event)

BKAWH[i] = 0：断路 i 信号不分配到模拟看门狗阈值上限事件

BKAWH[i] = 1：断路 i 信号分配到模拟看门狗阈值上限事件

30.8.10 DFSDM 模拟看门狗阈值下限寄存器 (DFSDM_FLTxAWLTR)

DFSDM analog watchdog low threshold register

偏移地址：0x124 + 0x80 * x，x = 0...3

复位值：0x0000 0000

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

位 31:8 **AWLT[23:0]**：模拟看门狗阈值下限 (Analog watchdog low threshold)

这些位由软件写入，用于定义模拟看门狗的阈值下限。

注： 对于通道收发器监视 (AWFSEL = 1)，仅高 16 位 (AWLT[23:8]) 定义 16 位阈值，该阈值与模拟看门狗滤波器输出进行比较（因为来自模拟看门狗滤波器的数据的分辨率最高为 16 位）。在这种情况下，不对位 AWLT[7:0] 进行比较。

位 7:4 保留，必须保持复位值。

位 3:0 **BKAWL[3:0]**：模拟看门狗阈值下限事件的断路信号分配 (Break signal assignment to analog watchdog low threshold event)

BKAWL[i] = 0：断路 i 信号不分配到模拟看门狗阈值下限事件

BKAWL[i] = 1：断路 i 信号分配到模拟看门狗阈值下限事件

30.8.11 DFSDM 模拟看门狗状态寄存器 (DFSDM_FLTxAWSR)

DFSDM analog watchdog status register

偏移地址：0x128 + 0x80 * x，x = 0...3

复位值：0x0000 0000

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

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

位 31:16 保留，必须保持复位值。

位 15:8 **AWHTF[7:0]**：模拟看门狗阈值上限标志 (Analog watchdog high threshold flag)
AWHTF[y] = 1 指示通道 y 上存在阈值上限错误。该位由硬件置 1。该位可由软件使用 DFSDM_FLTxAWCFR 寄存器中的相应 CLRAWHTF[y] 位清零。

位 7:0 **AWLTF[7:0]**：模拟看门狗阈值下限标志 (Analog watchdog low threshold flag)
AWLTF[y] = 1 指示通道 y 上存在阈值下限错误。该位由硬件置 1。该位可由软件使用 DFSDM_FLTxAWCFR 寄存器中的相应 CLRAWLTF[y] 位清零。

注： *DFEN = 0 时，DFSDM_FLTxAWSR 的所有位都会自动复位。*

30.8.12 DFSDM 模拟看门狗清零标志寄存器 (DFSDM_FLTxAWCFR)

DFSDM analog watchdog clear flag register

偏移地址：0x12C + 0x80 * x，x = 0...3

复位值：0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CLRAWHTF[7:0]								CLRAWLTF[7:0]							
rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1

位 31:16 保留，必须保持复位值。

位 15:8 **CLRAWHTF[7:0]**：清零模拟看门狗阈值上限标志 (Clear the analog watchdog high threshold flag)
CLRAWHTF[y] = 0：写入“0”无影响
CLRAWHTF[y] = 1：向位置 y 写入“1”会将 DFSDM_FLTxAWSR 寄存器中的相应 AWHTF[y] 位清零。

位 7:0 **CLRAWLTF[7:0]**：清零模拟看门狗阈值下限标志 (Clear the analog watchdog low threshold flag)
CLRAWLTF[y] = 0：写入“0”无影响
CLRAWLTF[y] = 1：向位置 y 写入“1”会将 DFSDM_FLTxAWSR 寄存器中的相应 AWLTF[y] 位清零。

30.8.13 DFSDM 极值检测器最大值寄存器 (DFSDM_FLTxEXMAX)

DFSDM Extremes detector maximum register

偏移地址：0x130 + 0x80 * x，x = 0...3

复位值：0x8000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
EXMAX[23:8]															
r1	r0	r0	r0	r0	r0	r0	r0	r0	r0	r0	r0	r0	r0	r0	r0
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EXMAX[7:0]								Res.	Res.	Res.	Res.	Res.	EXMAXCH[2:0]		
r0	r0	r0	r0	r0	r0	r0	r0						r	r	r

位 31:8 **EXMAX[23:0]**：极值检测器最大值 (Extremes detector maximum value)
这些位由硬件置 1，用于指示 DFSDM_FLTx 所转换的最大值。通过读取该寄存器将 EXMAX[23:0] 位复位为值 (0x800000)。

位 7:3 保留，必须保持复位值。

位 2:0 **EXMAXCH[2:0]**：极值检测器最大值数据通道 (Extremes detector maximum data channel)
这些位包含的信息涉及哪一个通道的数据被存储到 EXMAX[23:0]。可通过读取该寄存器将这些位清零。

30.8.14 DFSDM 极值检测器最小值寄存器 (DFSDM_FLTxEXMIN)

DFSDM Extremes detector minimum register

偏移地址：0x134 + 0x80 * x，x = 0...3

复位值：0x7FFF FF00

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
EXMIN[23:8]															
r 0	r1	r1	r1	r1	r1	r1	r1	r1	r1	r1	r1	r1	r1	r1	r1
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EXMIN[7:0]								Res.	Res.	Res.	Res.	Res.	EXMINCH[2:0]		
r1	r1	r1	r1	r1	r1	r1	r1						r	r	r

位 31:8 **EXMIN[23:0]**：极值检测器最小值 (Extremes detector minimum value)
这些位由硬件置 1，用于指示 DFSDM_FLTx 所转换的最小值。通过读取该寄存器将 EXMIN[23:0] 位复位为值 (0x7FFFFF)。

位 7:3 保留，必须保持复位值。

位 2:0 **EXMINCH[2:0]**：极值检测器最小值数据通道 (Extremes detector minimum data channel)
这些位包含的信息涉及哪一个通道的数据被存储到 EXMIN[23:0]。可通过读取该寄存器将这些位清零。

30.8.15 DFSDM 转换定时器寄存器 (DFSDM_FLTxCNVTIMR)

DFSDM conversion timer register

偏移地址：0x138 + 0x80 * x，x = 0...3

复位值：0x0000 0000

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

位 31:4 **CNVCNT[27:0]**：28 位定时器计数转换时间 (28-bit timer counting conversion time)
t = CNVCNT[27:0] / $f_{DFSDMCLK}$

定时器的输入时钟来自 DFSDM 时钟（系统时钟 $f_{DFSDMCLK}$）。转换时间测量始于每次转换开始，并止于每 36 次转换结束（即第一次和最后一次串行采样之间的间隔）。只有在滤波器旁路 (FOSR[9:0] = 0) 的情况下，转换时间测量才会停止，且 CNVCNT[27:0] = 0。时间计时如下：

如果 FAST=0（或者 FAST=1 连续模式下的第一次转换），则：

$t = [F_{OSR} * (I_{OSR}-1 + F_{ORD}) + F_{ORD}] / f_{CKIN}$.....（适用于 $Sinc^x$ 滤波器）

$t = [F_{OSR} * (I_{OSR}-1 + 4) + 2] / f_{CKIN}$.....（适用于 FastSinc 滤波器）

如果在连续模式下 FAST=1（第一次转换除外），则：

$t = [F_{OSR} * I_{OSR}] / f_{CKIN}$

如果 F_{OSR} = FOSR[9:0] + 1 = 1（滤波器旁路，仅积分器有效），则：

CNVCNT = 0（时间计时停止，转换时间：$t = I_{OSR} / f_{CKIN}$）

其中，f_{CKIN} 为（给定通道 CKINy 引脚上的）通道输入时钟频率；在并行数据输入（来自内部 ADC 或来自 CPU/DMA 写操作）的情况下，其表示输入数据速率。

注： 当转换被中断时（例如通过禁止/使能所选通道），定时器也将该中断时间计算在内。

位 3:0 保留，必须保持复位值。

30.8.16 DFSDM寄存器映射

下表对 DFSDM 寄存器进行了汇总。

表 236. DFSDM 寄存器映射和复位值

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	**DFSDM_CH0CFGR1**	DFSDMEN	CKOUTSRC	Res.	Res.	Res.	Res.	Res.	Res.	CKOUTDIV[7:0]								DATPACK[1:0]		DATMPX[1:0]		Res.	Res.	Res.	CHINSEL	CHEN	CKABEN	SCDEN	Res.	SPICKSEL[1:0]		SITP[1:0]	
	reset value	0	0							0	0	0	0	0	0	0	0	0	0	0	0				0	0	0	0		0	0	0	0
0x04	**DFSDM_CH0CFGR2**	OFFSET[23:0]																								DTRBS[4:0]					Res.	Res.	Res.
	reset value	0																								0							
0x08	**DFSDM_CH0AWSCDR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	AWFORD[1:0]		Res.	AWFOSR[4:0]					BKSCD[3:0]				Res.	Res.	Res.	Res.	SCDT[7:0]							
	reset value									0	0		0	0	0	0	0	0	0	0	0					0	0	0	0	0	0	0	0
0x0C	**DFSDM_CH0WDATR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WDATA[15:0]															
	reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x10	**DFSDM_CH0DATINR**	INDAT1[15:0]																INDAT0[15:0]															
	reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x14 - 0x1C	Reserved	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
0x20	**DFSDM_CH1CFGR1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DATPACK[1:0]		DATMPX[1:0]		Res.	Res.	Res.	CHINSEL	CHEN	CKABEN	SCDEN	Res.	SPICKSEL[1:0]		SITP[1:0]	
	reset value																	0	0	0	0				0	0	0	0		0	0	0	0
0x24	**DFSDM_CH1CFGR2**	OFFSET[23:0]																								DTRBS[4:0]					Res.	Res.	Res.
	reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
0x28	**DFSDM_CH1AWSCDR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	AWFORD[1:0]		Res.	AWFOSR[4:0]					BKSCD[3:0]				Res.	Res.	Res.	Res.	SCDT[7:0]							
	reset value									0	0		0	0	0	0	0	0	0	0	0					0	0	0	0	0	0	0	0
0x2C	**DFSDM_CH1WDATR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WDATA[15:0]															
	reset value																0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x30	**DFSDM_CH1DATINR**	INDAT1[15:0]																INDAT0[15:0]															
	reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x34 - 0x3C	Reserved	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

表 236. DFSDM 寄存器映射和复位值（续）

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x40	**DFSDM_CH2CFGR1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DATPACK[1:0]		DATMPX[1:0]		Res.	Res.	Res.	CHINSEL	CHEN	CKABEN	SCDEN	Res.	SPICKSEL[1:0]		SITP[1:0]	
	reset value																	0	0	0	0				0	0	0	0		0	0	0	0
0x44	**DFSDM_CH2CFGR2**	OFFSET[23:0]																								DTRBS[4:0]					Res.	Res.	Res.
	reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
0x48	**DFSDM_CH2AWSCDR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	AWFORD[1:0]		Res.	AWFOSR[4:0]					BKSCD[3:0]				Res.	Res.	Res.	Res.	SCDT[7:0]							
	reset value									0	0		0	0	0	0	0	0	0	0	0					0	0	0	0	0	0	0	0	0
0x4C	**DFSDM_CH2WDATR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WDATA[15:0]															
	reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x50	**DFSDM_CH2DATINR**	INDAT1[15:0]																INDAT0[15:0]															
	reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x54 - 0x5C	Reserved	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
0x60	**DFSDM_CH3CFGR1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DATPACK[1:0]		DATMPX[1:0]		Res.	Res.	Res.	CHINSEL	CHEN	CKABEN	SCDEN	Res.	SPICKSEL[1:0]		SITP[1:0]	
	reset value																	0	0	0	0				0	0	0	0		0	0	0	0
0x64	**DFSDM_CH3CFGR2**	OFFSET[23:0]																								DTRBS[4:0]					Res.	Res.	Res.
	reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
0x68	**DFSDM_CH3AWSCDR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	AWFORD[1:0]		Res.	AWFOSR[4:0]					BKSCD[3:0]				Res.	Res.	Res.	Res.	SCDT[7:0]							
	reset value									0	0		0	0	0	0	0	0	0	0	0					0	0	0	0	0	0	0	0	0
0x6C	**DFSDM_CH3WDATR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WDATA[15:0]															
	reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x70	**DFSDM_CH3DATINR**	INDAT1[15:0]																INDAT0[15:0]															
	reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x74 - 0x7C	Reserved	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
0x80	**DFSDM_CH4CFGR1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DATPACK[1:0]		DATMPX[1:0]		Res.	Res.	Res.	CHINSEL	CHEN	CKABEN	SCDEN	Res.	SPICKSEL[1:0]		SITP[1:0]	
	reset value																	0	0	0	0				0	0	0	0		0	0	0	0

表 236. DFSDM 寄存器映射和复位值（续）

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x84	**DFSDM_CH4CFGR2**	OFFSET[23:0]																								DTRBS[4:0]					Res.	Res.	Res.
	reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
0x88	**DFSDM_CH4AWSCDR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	AWFORD[1:0]		Res.	AWFOSR[4:0]					BKSCD[3:0]				Res.	Res.	Res.	Res.	SCDT[7:0]							
	reset value									0	0		0	0	0	0	0	0	0	0	0					0	0	0	0	0	0	0	0
0x8C	**DFSDM_CH4WDATR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WDATA[15:0]															
	reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x90	**DFSDM_CH4DATINR**	INDAT1[15:0]																INDAT0[15:0]															
	reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x94 - 0x9C	Reserved	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
0xA0	**DFSDM_CH5CFGR1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DATPACK[1:0]		DATMPX[1:0]		Res.	Res.	Res.	CHINSEL	CHEN	CKABEN	SCDEN	Res.	SPICKSEL[1:0]		SITP[1:0]	
	reset value																	0	0	0	0				0	0	0	0		0	0	0	0
0xA4	**DFSDM_CH5CFGR2**	OFFSET[23:0]																								DTRBS[4:0]					Res.	Res.	Res.
	reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
0xA8	**DFSDM_CH5AWSCDR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	AWFORD[1:0]		Res.	AWFOSR[4:0]					BKSCD[3:0]				Res.	Res.	Res.	Res.	SCDT[7:0]							
	reset value									0	0		0	0	0	0	0	0	0	0	0					0	0	0	0	0	0	0	0
0xAC	**DFSDM_CH5WDATR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.			Res.	Res.	Res.	Res.	Res.	Res.	WDATA[15:0]															
	reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0xB0	**DFSDM_CH5DATINR**	INDAT1[15:0]																INDAT0[15:0]															
	reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0xB4 - 0xBC	Reserved	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
0xC0	**DFSDM_CH6CFGR1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DATPACK[1:0]		DATMPX[1:0]		Res.	Res.	Res.	CHINSEL	CHEN	CKABEN	SCDEN	Res.	SPICKSEL[1:0]		SITP[1:0]	
	reset value																	0	0	0	0				0	0	0	0		0	0	0	0
0xC4	**DFSDM_CH6CFGR2**	OFFSET[23:0]																								DTRBS[4:0]					Res.	Res.	Res.
	reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			

表 236. DFSDM 寄存器映射和复位值（续）

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0xC8	**DFSDM_CH6AWSCDR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	AWFORD[1:0]		Res.	AWFOSR[4:0]					BKSCD[3:0]				Res.	Res.	Res.	Res.	SCDT[7:0]							
	reset value									0	0		0	0	0	0	0	0	0	0	0					0	0	0	0	0	0	0	0
0xCC	**DFSDM_CH6WDATR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WDATA[15:0]															
	reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0xD0	**DFSDM_CH6DATINR**	INDAT1[15:0]																INDAT0[15:0]															
	reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0xD4 - 0xDC	Reserved	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
0xE0	**DFSDM_CH7CFGR1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DATPACK[1:0]		DATMPX[1:0]		Res.	Res.	Res.	CHINSEL	CHEN	CKABEN	SCDEN	Res.	SPICKSEL[1:0]		SITP[1:0]	
	reset value																	0	0	0	0				0	0	0	0		0	0	0	0
0xE4	**DFSDM_CH7CFGR2**	OFFSET[23:0]																								DTRBS[4:0]					Res.	Res.	Res.
	reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
0xE8	**DFSDM_CH7AWSCDR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	AWFORD[1:0]		Res.	AWFOSR[4:0]					BKSCD[3:0]				Res.	Res.	Res.	Res.	SCDT[7:0]							
	reset value									0	0		0	0	0	0	0	0	0	0	0					0	0	0	0	0	0	0	0
0xEC	**DFSDM_CH7WDATR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WDATA[15:0]															
	reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0xF0	**DFSDM_CH7DATINR**	INDAT1[15:0]																INDAT0[15:0]															
	reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0xF4 - 0xFC	Reserved	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
0x100	**DFSDM_FLT0CR1**	Res.	AWFSEL	FAST	Res.	Res.	RCH[2:0]			Res.	Res.	RDMAEN	Res.	RSYNC	RCONT	RSW START	Res.	Res.	JEXTEN[1:0]		JEXTSEL[4:0]				Res.	Res.	JDMAEN	JSCAN	JSYNC	Res.	JSW START	DFEN
	reset value		0	0			0	0	0			0		0	0	0			0	0	0	0	0	0	0			0	0	0		0	0
0x104	**DFSDM_FLT0CR2**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	AWDCH[7:0]								EXCH[7:0]								Res.	CKABIE	SCDIE	AWDIE	ROVRIE	JOVRIE	REOCIE	JEOCIE
	reset value									0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		0	0	0	0	0	0	0
0x108	**DFSDM_FLT0ISR**	SCDF[7:0]								CKABF[7:0]								Res.	RCIP	JCIP	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	AWDF	ROVRF	JOVRF	REOCF	JEOCF
	reset value	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1		0	0									0	0	0	0	0

表 236. DFSDM 寄存器映射和复位值（续）

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x10C	**DFSDM_FLT0ICR**	CLRSCDF[7:0]								CLRCKABF[7:0]								Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CLR ROVRF	CLR JOVRF	Res.	Res.
	reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0													0	0		
0x110	**DFSDM_FLT0JCHGR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	JCHG[7:0]							
	reset value																									0	0	0	0	0	0	0	1
0x114	**DFSDM_FLT0FCR**	FORD[2:0]			Res.	Res.	Res.	FOSR[9:0]										Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	IOSR[7:0]							
	reset value	0	0	0				0	0	0	0	0	0	0	0	0	0									0	0	0	0	0	0	0	0
0x118	**DFSDM_FLT0JDATAR**	JDATA[23:0]																								Res.	Res.	Res.	Res.	Res.	JDATACH [2:0]		
	reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0						0	0	0
0x11C	**DFSDM_FLT0RDATAR**	RDATA[23:0]																								Res.	Res.	Res.	RPEND	Res.	RDATA CH[2:0]		
	reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				0		0	0	0
0x120	**DFSDM_FLT0AWHTR**	AWHT[23:0]																								Res.	Res.	Res.	Res.	BKAWH[3:0]			
	reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0					0	0	0	0
0x124	**DFSDM_FLT0AWLTR**	AWLT[23:0]																								Res.	Res.	Res.	Res.	BKAWL[3:0]			
	reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0					0	0	0	0
0x128	**DFSDM_FLT0AWSR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	AWHTF[7:0]								AWLTF[7:0]							
	reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x12C	**DFSDM_FLT0AWCFR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CLRAWHTF[7:0]								CLRAWLTF[7:0]							
	reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x130	**DFSDM_FLT0EXMAX**	EXMAX[23:0]																								Res.	Res.	Res.	Res.	Res.	EXMAXCH[2:0]		
	reset value	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0						0	0	0
0x134	**DFSDM_FLT0EXMIN**	EXMIN[23:0]																								Res.	Res.	Res.	Res.	Res.	EXMINCH[2:0]		
	reset value	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1						0	0	0
0x138	**DFSDM_FLT0CNVTIMR**	CNVCNT[27:0]																												Res.	Res.	Res.	Res.
	reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				
0x13C - 0x17C	Reserved	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

表 236. DFSDM 寄存器映射和复位值（续）

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x180	**DFSDM_FLT1CR1**	Res.	AWFSEL	FAST	Res.	Res.	RCH[2:0]			Res.	Res.	RDMAEN	Res.	RSYNC	RCONT	RSW START	Res.	Res.	JEXTEN[1:0]		JEXTSEL[4:0]					Res.	Res.	JDMAEN	JSCAN	JSYNC	Res.	JSW START	DFEN
	reset value		0	0			0	0	0			0		0	0	0			0	0		0	0	0	0			0	0	0		0	0
0x184	**DFSDM_FLT1CR2**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	AWDCH[7:0]								EXCH[7:0]								Res.	Res.	Res.	AWDIE	ROVRIE	JOVRIE	REOCIE	JEOCIE
	reset value									0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				0	0	0	0	0
0x188	**DFSDM_FLT1ISR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RCIP	JCIP	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	AWDF	ROVRF	JOVRF	REOCF	JEOCF
	reset value																		0	0									0	0	0	0	0
0x18C	**DFSDM_FLT1ICR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CLR ROVRF	CLR JOVRF	Res.	Res.
	reset value																													0	0		
0x190	**DFSDM_FLT1JCHGR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	JCHG[7:0]							
	reset value																									0	0	0	0	0	0	0	1
0x194	**DFSDM_FLT1FCR**	FORD[2:0]			Res.	Res.	Res.	FOSR[9:0]										Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	IOSR[7:0]							
	reset value	0	0	0				0	0	0	0	0	0	0	0	0	0									0	0	0	0	0	0	0	0
0x198	**DFSDM_FLT1JDATAR**	JDATA[23:0]																								Res.	Res.	Res.	Res.	Res.	JDATACH[2:0]		
	reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0						0	0	0
0x19C	**DFSDM_FLT1RDATAR**	RDATA[23:0]																								Res.	Res.	Res.	RPEND	Res.	RDATACH[2:0]		
	reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				0		0	0	0
0x1A0	**DFSDM_FLT1AWHTR**	AWHT[23:0]																								Res.	Res.	Res.	Res.	BKAWH[3:0]			
	reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0					0	0	0	0
0x1A4	**DFSDM_FLT1AWLTR**	AWLT[23:0]																								Res.	Res.	Res.	Res.	BKAWL[3:0]			
	reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0					0	0	0	0
0x1A8	**DFSDM_FLT1AWSR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	AWHTF[7:0]								AWLTF[7:0]							
	reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x1AC	**DFSDM_FLT1AWCFR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CLRAWHTF[7:0]								CLRAWLTF[7:0]							
	reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

表 236. DFSDM 寄存器映射和复位值（续）

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x1B0	**DFSDM_FLT1EXMAX**	EXMAX[23:0]																								Res.	Res.	Res.	Res.	Res.	EXMAXCH[2:0]		
	reset value	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0						0	0	0
0x1B4	**DFSDM_FLT1EXMIN**	EXMIN[23:0]																								Res.	Res.	Res.	Res.	Res.	EXMINCH[2:0]		
	reset value	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1						0	0	0
0x1B8	**DFSDM_FLT1CNVTIMR**	CNVCNT[27:0]																												Res.	Res.	Res.	Res.
	reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				
0x1BC - 0x1FC	Reserved	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
0x200	**DFSDM_FLT2CR1**	Res.	AWFSEL	FAST	Res.	Res.	RCH[2:0]			Res.	Res.	RDMAEN	Res.	RSYNC	RCONT	RSW START	Res.	Res.	JEXTEN[1:0]		JEXTSEL[4:0]					Res.	Res.	JDMAEN	JSCAN	JSYNC	Res.	JSW START	DFEN
	reset value		0	0			0	0	0			0		0	0	0			0	0	0	0	0	0	0			0	0	0		0	0
0x204	**DFSDM_FLT2CR2**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	AWDCH[7:0]								EXCH[7:0]								Res.	-	-	AWDIE	ROVRIE	JOVRIE	REOCIE	JEOCIE
	reset value									0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		-	-	0	0	0	0	0
0x208	**DFSDM_FLT2ISR**	-								-								Res.	RCIP	JCIP	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	AWDF	ROVRF	JOVRF	REOCF	JEOCF
	reset value	-								-									0	0									0	0	0	0	0
0x20C	**DFSDM_FLT2ICR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CLR ROVRF	CLR JOVRF	Res.	Res.
	reset value																													0	0		
0x210	**DFSDM_FLT2JCHGR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	JCHG[7:0]							
	reset value																									0	0	0	0	0	0	0	1
0x214	**DFSDM_FLT2FCR**	FORD[2:0]			Res.	Res.	Res.	FOSR[9:0]										Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	IOSR[7:0]							
	reset value	0	0	0				0	0	0	0	0	0	0	0	0	0									0	0	0	0	0	0	0	0
0x218	**DFSDM_FLT2JDATAR**	JDATA[23:0]																								Res.	Res.	Res.	Res.	Res.	JDATACH[2:0]		
	reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0						0	0	0

表 236. DFSDM 寄存器映射和复位值（续）

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x21C	**DFSDM_ FLT2RDATAR**	RDATA[23:0]																								Res.	Res.	Res.	RPEND	Res.	RDATA CH[2:0]		
	reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				0		0	0	0
0x220	**DFSDM_ FLT2AWHTR**	AWHT[23:0]																								Res.	Res.	Res.	Res.	BKAWH[3:0]			
	reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0					0	0	0	0
0x224	**DFSDM_ FLT2AWLTR**	AWLT[23:0]																								Res.	Res.	Res.	Res.	BKAWL[3:0]			
	reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0					0	0	0	0
0x228	**DFSDM_ FLT2AWSR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	AWHTF[7:0]								AWLTF[7:0]							
	reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x22C	**DFSDM_ FLT2AWCFR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CLRAWHTF[7:0]								CLRAWLTF[7:0]							
	reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x230	**DFSDM_ FLT2EXMAX**	EXMAX[23:0]																								Res.	Res.	Res.	Res.	Res.	EXMAX CH[2:0]		
	reset value	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0						0	0	0
0x234	**DFSDM_ FLT2EXMIN**	EXMIN[23:0]																								Res.	Res.	Res.	Res.	Res.	EXMINCH[2:0]		
	reset value	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1						0	0	0
0x238	**DFSDM_ FLT2CNVTIMR**	CNVCNT[27:0]																												Res.	Res.	Res.	Res.
	reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				
0x23C - 0x27C	Reserved	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
0x280	**DFSDM_ FLT3CR1**	Res.	AWFSEL	FAST	Res.	Res.	RCH[2:0]			Res.	Res.	RDMAEN	Res.	RSYNC	RCONT	RSW START	Res.	Res.	JEXTEN[1:0]		JEXTSEL[4:0]					Res.	Res.	JDMAEN	JSCAN	JSYNC	Res.	JSW START	DFEN
	reset value		0	0			0	0	0			0		0	0	0			0	0	0	0	0	0	0			0	0	0		0	0
0x284	**DFSDM_ FLT3CR2**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	AWDCH[7:0]								EXCH[7:0]								Res.	Res.	Res.	AWDIE	ROVRIE	JOVRIE	REOCIE	JEOCIE
	reset value									0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				0	0	0	0	0
0x288	**DFSDM_ FLT3ISR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RCIP	JCIP	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	AWDF	ROVRF	JOVRF	REOCF	JEOCF
	reset value																		0	0									0	0	0	0	0
0x28C	**DFSDM_ FLT3ICR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CLR ROVRF	CLR JOVRF	Res.	Res.
	reset value																													0	0		

表 236. DFSDM 寄存器映射和复位值（续）

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x290	**DFSDM_FLT3JCHGR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	JCHG[7:0]							
	reset value																									0	0	0	0	0	0	0	1
0x294	**DFSDM_FLT3FCR**	FORD[2:0]			Res.	Res.	Res.	FOSR[9:0]										Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	IOSR[7:0]							
	reset value	0	0	0				0	0	0	0	0	0	0	0	0	0									0	0	0	0	0	0	0	0
0x298	**DFSDM_FLT3JDATAR**	JDATA[23:0]																								Res.	Res.	Res.	Res.	Res.	JDATACH[2:0]		
	reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0						0	0	0
0x29C	**DFSDM_FLT3RDATAR**	RDATA[23:0]																								Res.	Res.	Res.	RPEND	Res.	RDATACH[2:0]		
	reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				0		0	0	0
0x2A0	**DFSDM_FLT3AWHTR**	AWHT[23:0]																								Res.	Res.	Res.	Res.	BKAWH[3:0]			
	reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0					0	0	0	0
0x2A4	**DFSDM_FLT3AWLTR**	AWLT[23:0]																								Res.	Res.	Res.	Res.	BKAWL[3:0]			
	reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0					0	0	0	0
0x2A8	**DFSDM_FLT3AWSR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	AWHTF[7:0]								AWLTF[7:0]							
	reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x2AC	**DFSDM_FLT3AWCFR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CLRAWHTF[7:0]								CLRAWLTF[7:0]							
	reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x2B0	**DFSDM_FLT3EXMAX**	EXMAX[23:0]																								Res.	Res.	Res.	Res.	Res.	EXMAXCH[2:0]		
	reset value	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0						0	0	0
0x2B4	**DFSDM_FLT3EXMIN**	EXMIN[23:0]																								Res.	Res.	Res.	Res.	Res.	EXMINCH[2:0]		
	reset value	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1						0	0	0
0x2B8	**DFSDM_FLT3CNVTIMR**	CNVCNT[27:0]																												Res.	Res.	Res.	Res.
	reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				
0x2BC - 0x3FC	Reserved	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

有关寄存器边界地址的信息，请参见*第 2.2.2 节：存储器映射和寄存器边界地址*。

31 数字摄像头接口 (DCMI)

31.1 DCMI 简介

数字摄像头接口是一个同步并行接口，能够接收外部 8 位、10 位、12 位或 14 位 CMOS 摄像头模块发出的高速数据流。可支持不同的数据格式：YCbCr4:2:2/RGB565 逐行视频和压缩数据 (JPEG)。

此接口适用于黑白摄像头、X24 和 X5 摄像头，并假定所有预处理（如调整大小）都在摄像头模块中执行。

31.2 DCMI 主要特性

- 8 位、10 位、12 位或 14 位并行接口
- 内嵌码/外部行同步和帧同步
- 连续模式或快照模式
- 裁剪功能
- 支持以下数据格式：
  - 8/10/12/14 位逐行视频：单色或原始拜尔格式
  - YCbCr 4:2:2 逐行视频
  - RGB 565 逐行视频
  - 压缩数据：JPEG

31.3 DCMI 时钟

数字摄像头接口使用 DCMI_PIXCLK 和 HCLK 这两个时钟域。伴随 DCMI_PIXCLK 产生的信号稳定之后，将在 HCLK 上升沿时对这些信号进行采样。HCLK 域中的一个使能信号用于指示摄像头发出的数据已稳定，可进行采样。DCMI_PIXCLK 的最大周期必须大于 2.5 倍 HCLK 周期。

31.4 DCMI 功能概述

数字摄像头接口是一个同步并行接口，可接收高速数据流。该接口包含多达 14 条数据线 (D13-D0) 和一条像素时钟线 (DCMI_PIXCLK)。像素时钟的极性可以编程，因此可以在像素时钟的上升沿或下降沿捕获数据。

这些数据以 32 位（4 字节）对齐，存储到 32 位数据寄存器 (DCMI_DR) 中，然后通过通用 DMA 进行传输。图像缓冲区由 DMA 管理，而不是由摄像头接口管理。

从摄像头接收的数据可以按行/帧来组织（原始 YUB/RGB/拜尔模式），也可以是一系列 JPEG 图像。要使能 JPEG 图像接收，必须将 JPEG 位（DCMI_CR 寄存器的位 3）置 1。

数据流可由可选的 DCMI_HSYNC（水平同步）信号和 DCMI_VSYNC（垂直同步）信号硬件同步，或者通过数据流中嵌入的同步码同步。

31.4.1 DCMI 框图

图 225 显示了 DCMI 框图。

图 225. DCMI 框图

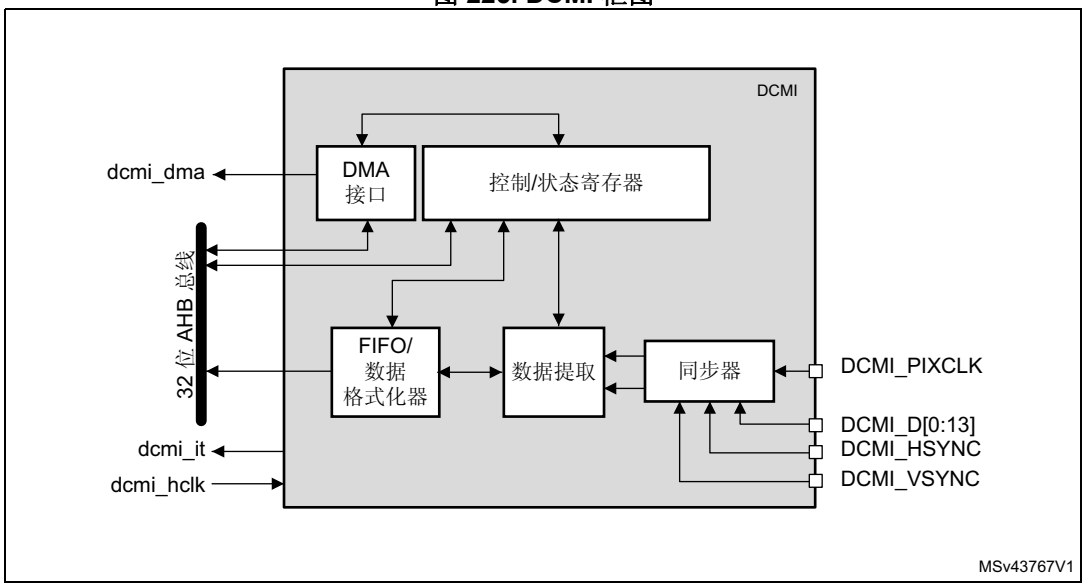
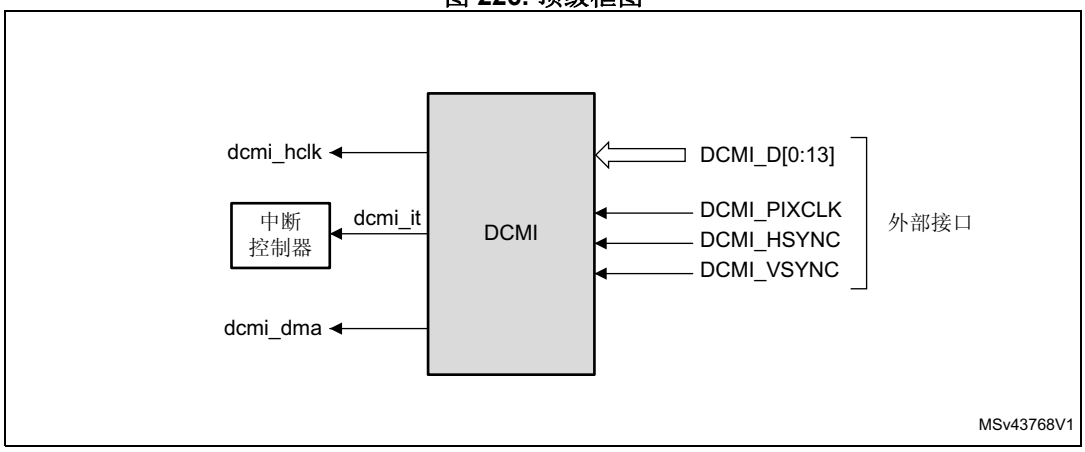


图 226. 顶级框图



31.4.2 DCMI 内部信号

表 237 显示了 DCMI 内部信号。

表 237. DCMI 内部信号

名称	信号类型	说明
dcmi_dma	数字输出	DCMI DMA 请求
dcmi_it	数字输出	DCMI 中断请求
dcmi_hclk	数字输入	DCMI 接口时钟

31.4.3 DMA 接口

当 DCMI_CR 寄存器中的 CAPTURE 位置 1 时，激活 DMA 接口。摄像头接口每次在其寄存器中收到一个完整的 32 位数据块时，都将触发一个 DMA 请求。

31.4.4 DCMI 物理接口

该接口由 11/13/15/17 个输入信号组成。仅支持从模式。

根据 DCMI_CR 寄存器中 EDM[1:0] 位的设置，摄像头接口可以捕获 8 位、10 位、12 位或 14 位数据。如果使用的位数少于 14，则必须将未使用的输入引脚接地。

表 238 显示了 DCMI 引脚。

表 238. DCMI 外部信号

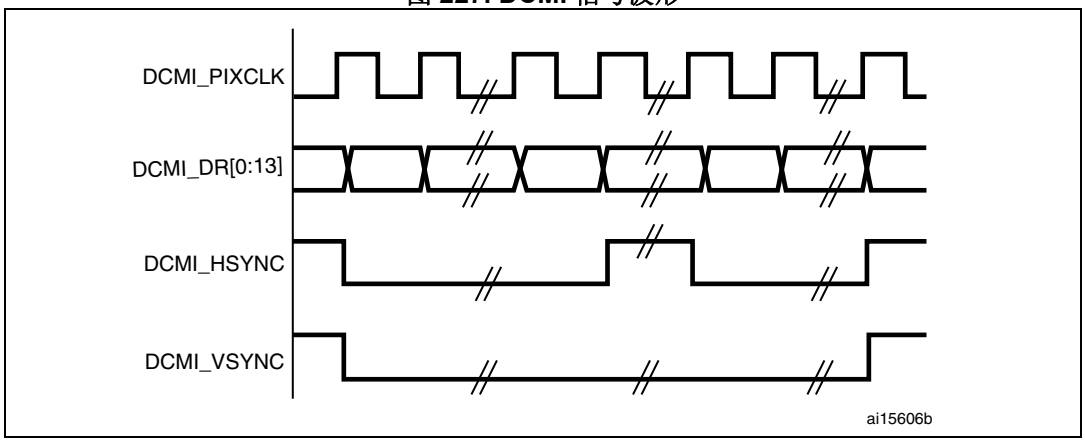
信号名称		信号类型	信号说明
8 位 10 位 12 位 14 位	DCMI_D[0..7] DCMI_D[0..9] DCMI_D[0..11] DCMI_D[0..13]	数字输入	DCMI 数据
DCMI_PIXCLK		数字输入	像素时钟
DCMI_HSYNC		数字输入	水平同步/数据有效
DCMI_VSYNC		数字输入	垂直同步

数据与 DCMI_PIXCLK 保持同步，并根据像素时钟的极性在像素时钟上升沿/下降沿发生变化。

DCMI_HSYNC 信号指示行的开始/结束。

DCMI_VSYNC 信号指示帧的开始/结束。

图 227. DCMI 信号波形



1. DCMI_PIXCLK 的捕获沿为下降沿，DCMI_HSYNC 和 DCMI_VSYNC 的有效状态为 1。
2. DCMI_HSYNC 和 DCMI_VSYNC 的状态可同时发生更改。

8 位数据

当 DCMI_CR 中的 EDM[1:0] 编程为“00”时，接口将捕获其输入 (DCMI_D[0:7]) 的 8 个 LSB，并将其存储为 8 位数据。DCMI_D[13:8] 输入则忽略。在此情况下，要捕获 32 位字，摄像头接口需要花费四个像素时钟周期。

捕获的第一个数据字节放置在 32 位字的 LSB 位置，捕获的第四个数据字节放置在 32 位字的 MSB 位置。*表 239* 列举了捕获到的数据字节在两个 32 位字中的位置排布。

表 239. 捕获的数据字节在 32 位字（宽 8 位）中的位置排布

字节地址	31:24	23:16	15:8	7:0
0	$D_{n+3}[7:0]$	$D_{n+2}[7:0]$	$D_{n+1}[7:0]$	$D_n[7:0]$
4	$D_{n+7}[7:0]$	$D_{n+6}[7:0]$	$D_{n+5}[7:0]$	$D_{n+4}[7:0]$

10 位数据

当 DCMI_CR 中的 EDM[1:0] 编程为“01”时，摄像头接口将捕获其输入 DCMI_D[0..9] 的 10 位数据，并将其存储为 16 位字的 10 个最低有效位。DCMI_DR 寄存器中的其余最高有效位（位 11 到 15）将清零。因此，在此情况下，每两个像素时钟周期会生成一个 32 位数据字。

捕获的第一个数据放置在 32 位字的 LSB 位置，捕获的第二个数据放置在 32 位字的 MSB 位置，如*表 240* 中所示。

表 240. 捕获的数据字节在 32 位字（宽 10 位）中的位置排布

字节地址	31:26	25:16	15:10	9:0
0	0	$D_{n+1[9:0]}$	0	$D_n[9:0]$
4	0	$D_{n+3}[9:0]$	0	$D_{n+2}[9:0]$

12 位数据

当 DCMI_CR 中的 EDM[1:0] 编程为“10”时，摄像头接口将捕获其输入 DCMI_D[0..11] 的 12 位数据，并将其存储为 16 位字的 12 个最低有效位。其余最高有效位将清零。因此，在此情况下，每两个像素时钟周期会生成一个 32 位数据字。

捕获的第一个数据放置在 32 位字的 LSB 位置，捕获的第二个数据放置在 32 位字的 MSB 位置，如*表 241* 中所示。

表 241. 捕获的数据字节在 32 位字（宽 12 位）中的位置排布

字节地址	31:28	27:16	15:12	11:0
0	0	D_{n+1}[11:0]	0	D_n[11:0]
4	0	D_{n+3}[11:0]	0	D_{n+2}[11:0]

14 位数据

当 DCMI_CR 中的 EDM[1:0] 编程为“11”时，摄像头接口将捕获其输入 DCMI_D[0..13] 的 14 位数据，并将其存储为 16 位字的 14 个最低有效位。其余最高有效位将清零。因此，在此情况下，每两个像素时钟周期会生成一个 32 位数据字。

捕获的第一个数据放置在 32 位字的 LSB 位置，捕获的第二个数据放置在 32 位字的 MSB 位置，如*表 242* 中所示。

表 242. 捕获的数据字节在 32 位字（宽 14 位）中的位置排布

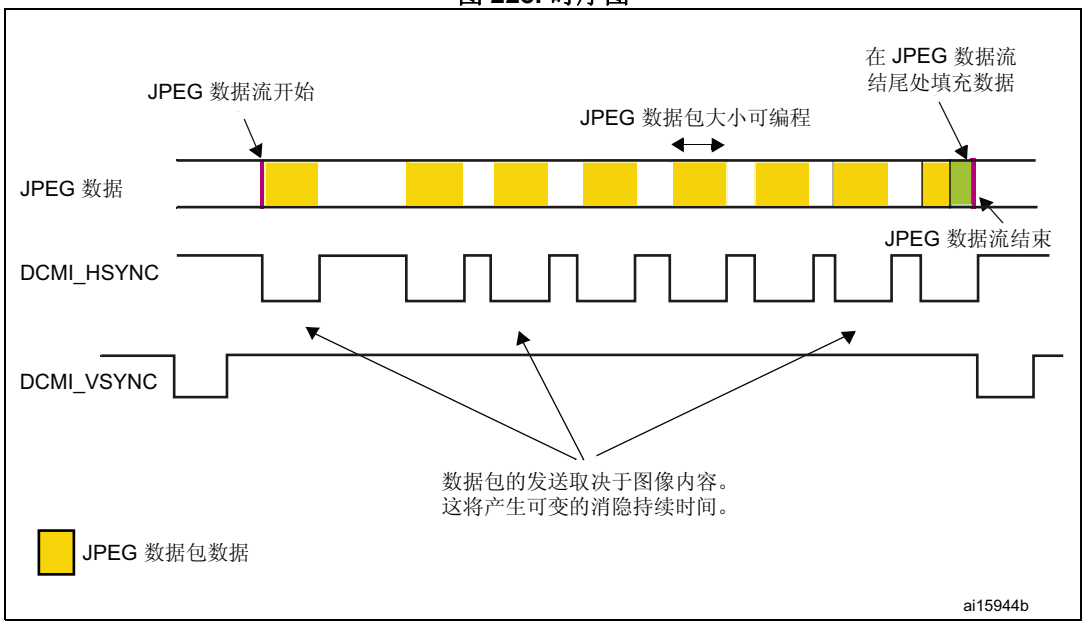
字节地址	31:30	29:16	15:14	13:0
0	0	D_{n+1}[13:0]	0	D_n[13:0]
4	0	D_{n+3}[13:0]	0	D_{n+2}[13:0]

31.4.5 同步

数字摄像头接口支持内嵌码同步或硬件（DCMI_HSYNC 和 DCMI_VSYNC）同步。使用内嵌码同步时，由数字摄像头模块确保 0x00 和 0xFF 值仅用于同步（不用于数据中）。只有 8 位并行数据接口宽度支持内嵌码同步码（即，DCMI_CR 寄存器中的 EDM[1:0] 位应为“00”）。

对于压缩数据，DCMI 仅支持硬件同步模式。这种情况下，DCMI_VSYNC 指示图像的开始/结束，DCMI_HSYNC 则用作“数据有效”信号。*图 228* 显示了相应的时序图。

图 228. 时序图



硬件同步模式

硬件同步模式下将使用两个同步信号 (DCMI_HSYNC/DCMI_VSYNC)。

根据摄像头模块/模式的不同，可能在水平/垂直同步期间内发送数据。
由于系统会忽略 DCMI_HSYNC/DCMI_VSYNC 信号有效电平期间内接收的所有数据，DCMI_HSYNC/DCMI_VSYNC 信号相当于消隐信号。

为了正确地将图像传输到 DMA/RAM 缓冲区，数据传输将与 DCMI_VSYNC 信号同步。选择硬件同步模式并启用捕获（DCMI_CR 中的 CAPTURE 位置 1）时，数据传输将与 DCMI_VSYNC 信号的无效电平同步（开始下一帧时）。

之后传输便可以连续执行，由 DMA 将连续帧传输到多个连续的缓冲区或一个具有循环特性的缓冲区。为了允许 DMA 管理连续帧，每一帧结束时都将激活 VSIF（垂直同步中断标志）。

内嵌码同步模式

在此同步模式下，将使用数据流中嵌入的 32 位码来同步数据流。这些码使用数据中*不*再使用的值 0x00/0xFF。共有 4 种同步码类型，均采用 0xFF0000XY 格式。只有 8 位并行数据接口支持内嵌码同步（DCMI_CR 寄存器中的 EDM[1:0] 位应编程为“00”）。对于其它数据宽度，此模式将造成无法预知的结果，因此不得使用。

注： *（在隔行扫描模式下）摄像头模块具有 8 个此类编码。因此，摄像头接口不支持隔行扫描模式（否则每帧数据的一半都会被丢弃）。*

- 模式 2

  四个内嵌码通知以下事件

  – 帧开始 (FS)
  – 帧结束 (FE)
  – 行开始 (LS)
  – 行结束 (LE)

  4 个同步码采用的格式 0xFF0000XY 中的 XY 值可编程（参见*第 31.7.7 节：DCMI 内嵌同步码寄存器 (DCMI_ESCR)*）。

  将 0xFF 编程为“帧结束”意味着所有未使用的编码均被解释为有效帧结束编码。

  在此模式下，一旦使能摄像头接口，将在首次出现帧结束 (FE) 同步码并且后接帧开始 (FS) 同步码之后开始捕获帧。

- 模式 1

  摄像头模式 1 是另一种编码。此模式与 ITU656 兼容。

  这些同步码通知另一组事件：

  – SAV（有效行）——行开始
  – EAV（有效行）——行结束
  – SAV（消隐）——帧间消隐期内的行开始
  – EAV（消隐）——帧间消隐期内的行结束

可通过对同步码进行如下编程来支持此模式：

- FS ≤ 0xFF
- FE ≤ 0xFF
- LS ≤ SAV（有效）
- LE ≤ EAV（有效）

此外还针对帧/行开始和帧/行结束同步码实现了非屏蔽位功能。这样可以仅使用同步码中未被屏蔽的位进行比较。因此，可以选择一个位用于同步码的比较，来检测帧/行起始和帧/行结束。这意味着可以多个同步码表示帧/行的起始和结束，它们仅在未被屏蔽的位相同即可。

示例

FS = 0xA5

FS 的非屏蔽码 = 0x10

这种情况下，只需要比较数据码的第 4 位来检测是否是 FS 信号。

31.4.6 捕获模式

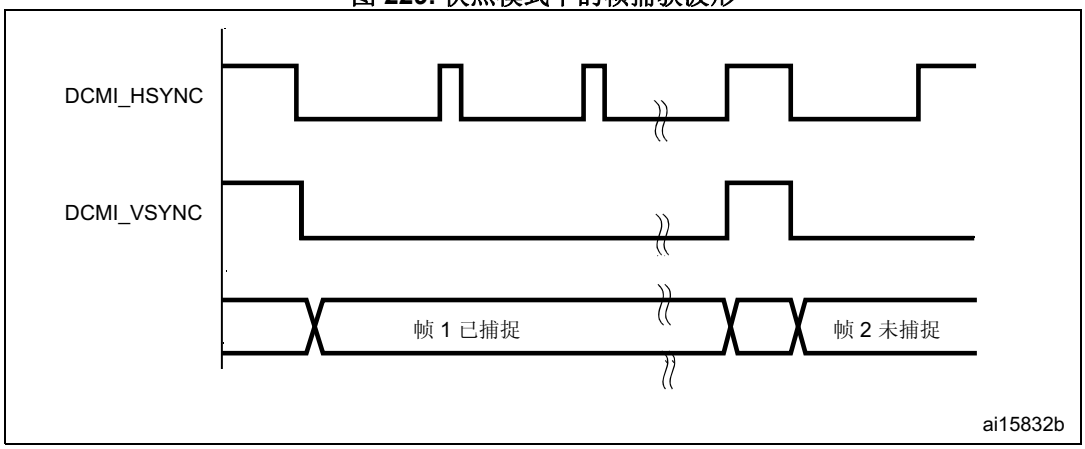
此接口支持两种类型的捕获：快照（单帧）和连续采集。

快照模式（单帧）

此模式下只捕获单帧（DCMI_CR 寄存器中的 CM = “1”）。在 DCMI_CR 中的 CAPTURE 位置 1 后，该接口将等待系统检测帧开始，然后再对数据进行采样。收到完整的第一帧后，将自动禁止摄像头接口（DCMI_CR 中的 CAPTURE 位清零）。如果使能相应中断，将生成中断 (IT_FRAME)。

如果发生溢出，这帧将丢失并且 CAPTURE 位清零。

图 229. 快照模式下的帧捕获波形

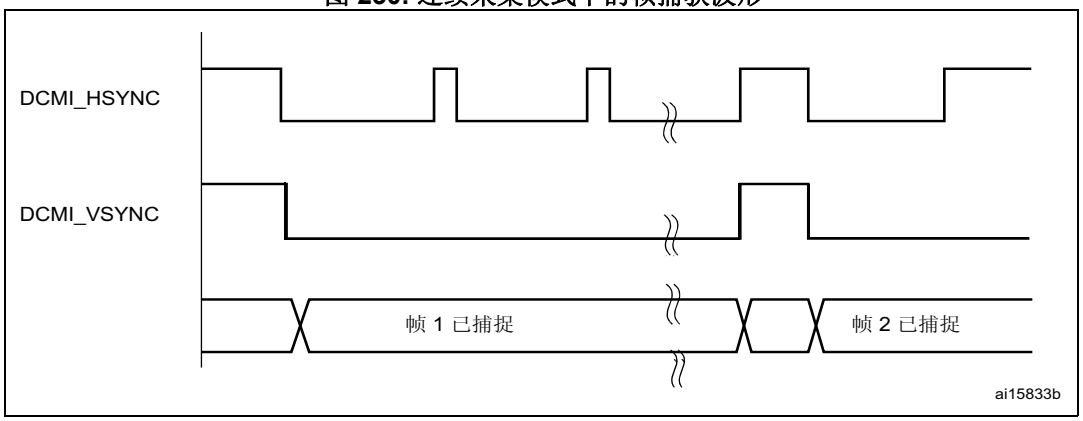


1. 此例中，DCMI_HSYNC 和 DCMI_VSYNC 的有效状态为 1。
2. DCMI_HSYNC 和 DCMI_VSYNC 的状态可同时发生更改。

连续采集模式

在此模式下（DCMI_CR 中的 CM 位 = “0”），一旦 DCMI_CR 中的 CAPTURE 位置 1，将在下一个 DCMI_VSYNC 或内嵌同步码帧起始时启动采集过程，具体取决于同步模式。该过程一直持续到 DCMI_CR 中的 CAPTURE 位清零。CAPTURE 位清零后，采集过程将持续到当前帧结束。

图 230. 连续采集模式下的帧捕获波形



1. 此例中，DCMI_HSYNC 和 DCMI_VSYNC 的有效状态为 1。
2. DCMI_HSYNC 和 DCMI_VSYNC 的状态可同时发生更改。

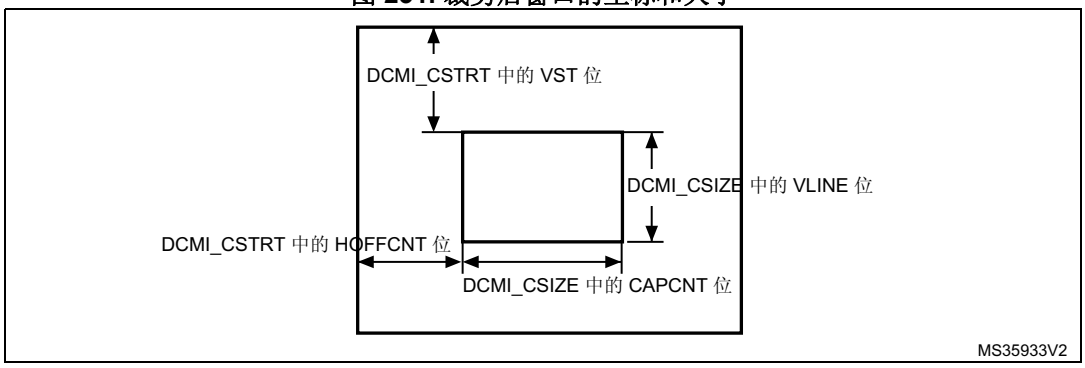
在连续采集模式下，可以通过配置 DCMI_CR 中的 FCRC 位来选择采集所有图片，或每两帧采集一次图片，或每四帧采集一张图片，以此降低帧捕获率。

注： *在硬件同步模式下（DCMI_CR 中的 ESS = “0”），即使 DCMI_CR 中的 CAPTURE = “0”，也将生成 IT_VSYNC 中断（如果使能）。因此为进一步降低帧捕获率，IT_VSYNC 中断可与快照模式结合使用，用来统计 2 次捕获之间的帧数。这并不适用于内嵌码同步模式。*

31.4.7 裁剪功能

摄像头接口可以使用裁剪功能从收到的图像中选择一个矩形窗口。起始（左上角）坐标和窗口大小（用像素时钟数表示的水平尺寸以及用行数表示的垂直尺寸）由两个 32 位寄存器（DCMI_CWSTRT 和 DCMI_CWSIZE）指定。窗口大小用像素时钟数（水平尺寸）和行数（垂直尺寸）表示。

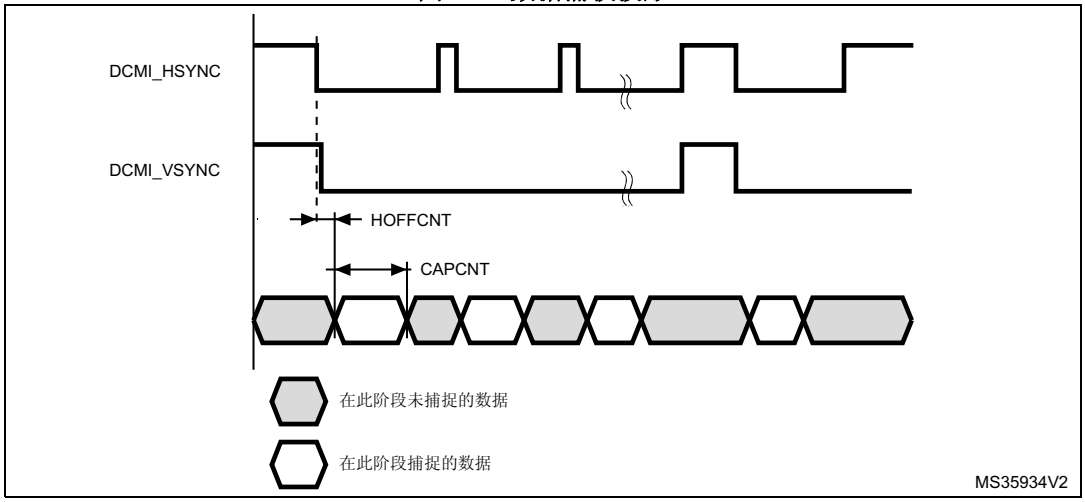
图 231. 裁剪后窗口的坐标和大小



这些寄存器将捕获窗口的起点坐标指定为某一行号（从 0 开始）和某一个像素时钟数（从 0 开始），窗口大小则指定为行数和像素时钟数。CAPCNT 值只能是 4 的倍数（两个最低有效位强制为 0），才能通过 DMA 正确传输数据。

如果在捕获 DCMI_CWSIZE 寄存器中指定行数完成之前，VSYNC 信号已有效，那么捕获将停止，并且在中断使能时生成 IT_FRAME 中断。

图 232. 数据捕获波形



1. 此例中，DCMI_HSYNC 和 DCMI_VSYNC 的有效状态为 1。
2. DCMI_HSYNC 和 DCMI_VSYNC 的状态可同时发生更改。

31.4.8 JPEG 格式

要允许接收 JPEG 图像，必须将 DCMI_CR 寄存器中的 JPEG 位置 1。JPEG 图像不按行和帧存储，因此 DCMI_VSYNC 信号用于启动捕获过程，而 DCMI_HSYNC 则用作数据使能信号。行中包含的字节数可能不是 4 的倍数，因此处理此类情况时应十分谨慎，因为需要每次从捕获的数据形成一个完整的 32 位字时，才生成一个 DMA 请求。检测到帧结束并且尚未凑成 32 位字时，将使用“0”进行填充，并触发一个 DMA 请求。

裁剪功能和内嵌码同步不适用于 JPEG 格式。

31.4.9 FIFO

输入模式

为了对 AHB 上的数据传输率加以管理，在摄像头接口上实现了 4 个字深度的 FIFO。DCMI 配有一个简单的 FIFO 控制器，每次摄像头接口从 AHB 读取数据时读指针递增，每次摄像头接口向 FIFO 写入数据时写指针递增。因为没有溢出保护，如果数据传输率超过了 AHB 接口能够承受的速率，FIFO 中的数据就会被覆盖。

如果同步信号出错，或者 FIFO 发生溢出，FIFO 将复位，DCMI 接口将等待新的数据帧开始。

31.5 数据格式说明

31.5.1 数据格式

支持三种类型的数据：

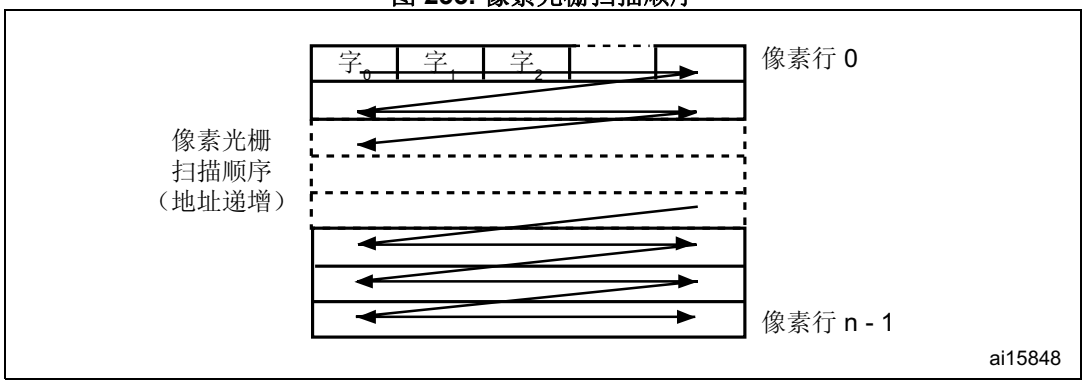
- 8 位逐行视频：单色或原始拜尔格式
- YCbCr 4:2:2 逐行视频
- RGB565 逐行视频。采用 16 位（5 位表示蓝色，5 位表示红色，6 位表示绿色）编码的像素需要两个时钟周期传输。

压缩数据：JPEG

对于 B&W、YCbCr 或 RGB 数据，最大输入大小为 2048 × 2048 像素。JPEG 压缩模式无限制。

对于单色、RGB 和 YCbCr，帧缓冲区以光栅模式存储。使用 32 位字。仅支持小端对齐格式。

图 233. 像素光栅扫描顺序



31.5.2 单色格式

特性：

- 光栅格式
- 每个像素 8 位

表 243 显示了数据的存储方式。

表 243. 单色逐行视频格式的数据存储

字节地址	31:24	23:16	15:8	7:0
0	n + 3	n + 2	n + 1	n
4	n + 7	n + 6	n + 5	n + 4

31.5.3 RGB 格式

特性：

- 光栅格式
- RGB
- 交替：一个缓冲区：R,G&B 交替：BRGBRGBRG：等
- 对显示输出的优化

RGB 平面格式与标准的 OS 帧缓冲显示格式兼容。
仅支持 16 BPP（每个像素 16 位）：RGB565（每 32 位字表示 2 个像素）。
不支持 24 BPP（调色板格式）和灰度格式。像素按照光栅扫描顺序进行存储，即从顶部像素行到底部像素行，从像素行的左侧到右侧。像素分量包括 R（红色）、G（绿色）和 B（蓝色）。所有分量的空间分辨率都相同（格式 4:4:4）。一帧数据中各个分量交替间隔存储。

表 244 显示了数据的存储方式。

表 244. 以 RGB 逐行视频格式存储数据

字节地址	31:27	26:21	20:16	15:11	10:5	4:0
0	红色 n + 1	绿色 n + 1	蓝色 n + 1	红色 n	绿色 n	蓝色 n
4	红色 n + 4	绿色 n + 3	蓝色 n + 3	红色 n + 2	绿色 n + 2	蓝色 n + 2

31.5.4 YCbCr 格式

特性：

- 光栅格式
- YCbCr 4:2:2
- 交替：一个缓冲区：Y、Cb&Cr 交替：CbYCrYCbYCr 等

像素分量包括 Y（亮度）、Cb 和 Cr（蓝色色度和红色色度）。每个分量都采用 8 位进行编码。亮度和色度（交替）存储在一起，如*表 245* 中所示。

表 245. YCbCr 逐行视频格式下的数据存储

字节地址	31:24	23:16	15:8	7:0
0	Y n + 1	Cr n	Y n	Cb n
4	Y n + 3	Cr n + 2	Y n + 2	Cb n + 2

31.5.5 YCbCr 格式——仅含 Y 分量

特性：

- 光栅格式
- YCbCr 4:2:2
- 缓冲区仅包含 Y 分量信息——单色图像

像素分量包括 Y（亮度）、Cb 和 Cr（蓝色色度和红色色度）。在此模式下，将丢弃色度信息。仅存储每个像素采用 8 位进行编码的亮度分量，如*表 246* 所示。

结果为单色图像，其分辨率与原始 YCbCr 数据的分辨率相同。

表 246. YCbCr 逐行视频格式下的数据存储——Y 分量提取模式

字节地址	31:24	23:16	15:8	7:0
0	Y n + 3	Y n + 2	Y n + 1	Y n
4	Y n + 7	Y n + 6	Y n + 5	Y n + 4

31.5.6 半分辨率图像提取

这是对先前接收模式的修改，适用于单色、RGB 或 Y 分量提取模式。

此模式仅允许存储半分辨率图像。可通过 OELS 和 LSM 控制位进行选择。

31.6 DCMI 中断

有五种中断源。所有中断都可通过软件屏蔽。全局中断 (dcmi_it) 是所有单个中断的逻辑或运算所得结果。*表 247* 列出了所有中断。

表 247. DCMI 中断

中断名称	中断事件
IT_LINE	表示行结束
IT_FRAME	表示帧捕获结束
IT_OVR	表示接收的数据发生溢出错误
IT_VSYNC	表示同步帧
IT_ERR	表示内嵌码同步帧检测期间检测到错误
DCMI_IT	以上中断的逻辑或结果

31.7 DCMI 寄存器说明

所有 DCMI 寄存器都必须作为 32 位字访问，否则将发生总线错误。

31.7.1 DCMI 控制寄存器 (DCMI_CR)

DCMI control register

偏移地址：0x00

复位值：0x0000 0x0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OELS	LSM	OEBS	BSM	
											RW	RW	RW	RW	RW

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	ENABLE	Res.	Res.	EDM		FCRC		VSPOL	HSPOL	PCKPOL	ESS	JPEG	CROP	CM	CAPTURE
	rw			rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:21 保留，必须保持复位值。

位 20 **OELS**：奇数/偶数行选择（行选择开始）(Odd/Even Line Select (Line Select Start))

此位与 LSM 字段 (LSM = 1) 一起使用

0：接口在帧开始后捕获第一行，同时丢弃第二行

1：接口在帧开始后捕获第二行，同时丢弃第一行

位 19 **LSM**：行选择模式 (Line Select mode)

0：接口捕获所有接收到的行

1：接口每两行捕获一行

位 18 **OEBS**：奇数/偶数字节选择（字节选择开始）(Odd/Even Byte Select (Byte Select Start))

此位与 BSM 字段 (BSM <> 00) 一起使用

0：接口从帧/行开始捕获第一个数据（字节或双字节），同时丢弃第二个字节

1：接口从帧/行开始捕获第二个数据（字节或双字节），同时丢弃第一个字节

位 17:16 **BSM[1:0]**：字节选择模式 (Byte Select mode)

00：接口捕获所有接收到的数据
01：接口对接收到的数据每隔一个字节进行捕获
10：接口每四个字节捕获一个字节
11：接口每四个字节捕获两个字节

注： 此模式仅在 EDM[1:0]=00 时有效。对于所有其它 EDM 值，必须将此位域编程为复位值。

位 15 保留，必须保持复位值。

位 14 **ENABLE**：DCMI 使能 (DCMI enable)

0：禁止 DCMI
1：使能 DCMI

注： 使能此位之前，应对 DCMI 配置寄存器进行适当的设置

位 13:12 保留，必须保持复位值。

位 11:10 **EDM[1:0]**：扩展数据模式 (Extended data mode)

00：接口每个像素时钟捕获 8 位数据
01：接口每个像素时钟捕获 10 位数据
10：接口每个像素时钟捕获 12 位数据
11：接口每个像素时钟捕获 14 位数据

位 9:8 **FCRC[1:0]**：帧捕获率控制 (Frame capture rate control)

这些位定义了帧捕获频率。仅在连续采集模式下有效。快照模式下将被忽略。
00：捕获所有帧
01：每隔一帧捕获一次（带宽降低 50%）
10：每隔三帧捕获一次（带宽降低 75%）
11：保留

位 7 **VSPOL**：垂直同步极性 (Vertical synchronization polarity)

此位指示数据在并行接口上无效时 DCMI_VSYNC 引脚的电平。
0：DCMI_VSYNC 低电平有效
1：DCMI_VSYNC 高电平有效

位 6 **HSPOL**：水平同步极性 (Horizontal synchronization polarity)

此位指示数据在并行接口上无效时 DCMI_HSYNC 引脚的电平。
0：DCMI_HSYNC 低电平有效
1：DCMI_HSYNC 高电平有效

位 5 **PCKPOL**：像素时钟极性 (Pixel clock polarity)

此位用来配置像素时钟的捕获沿
0：下降沿有效。
1：上升沿有效。

位 4 **ESS**：内嵌码同步选择 (Embedded synchronization select)

0：硬件同步，数据捕获（帧/行开始/停止）由 DCMI_HSYNC/DCMI_VSYNC 信号同步。
1：内嵌码同步，数据捕获由数据流中嵌入的同步码同步。

注： 仅对 8 位并行数据有效。ESS 位置 1 时，将忽略 HSPOL/VSPOL。

JPEG 模式下会禁止此位。

位 3 **JPEG**：JPEG 格式 (JPEG format)

0：未经压缩的视频格式

1：此位用于 JPEG 数据传输。DCMI_HSYNC 信号用作数据使能信号。此模式下无法使用裁剪和内嵌码同步功能（ESS 位）。

位 2 **CROP**：裁剪功能 (Crop feature)

0：捕获完整图像。这种情况下，图像帧包含的字节总数应该为 4 的倍数。

1：仅捕获剪裁寄存器所指定的窗口中的数据。如果窗口大小超出图片大小，则仅捕获图片大小。

位 1 **CM**：捕获模式 (Capture mode)

0：连续采集模式——收到的数据将通过 DMA 传输到目标存储区。缓冲区位置和模式（线性或循环缓冲区）由系统 DMA 控制。

1：快照模式（单帧）——一旦激活，接口将等待帧开始，然后通过 DMA 传输单帧。帧结束时将自动复位 CAPTURE 位。

位 0 **CAPTURE**：使能捕获 (Capture enable)

0：禁止捕获。

1：使能捕获。

摄像头接口等待第一帧开始，然后生成一个 DMA 请求以将收到的数据传输到目标存储器中。

在快照模式下，收到的第一帧结束时将自动使 CAPTURE 位清零。

在连续采集模式下，如果在执行捕获操作时通过软件将此位清零，则帧结束后此位的清零才生效。

注：　使能此位之前，应对 DMA 控制器和所有 DCMI 配置寄存器进行适当的编程。

31.7.2 DCMI 状态寄存器 (DCMI_SR)

DCMI status register

偏移地址：0x04

复位值：0x0000 0x0000

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

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

位 31:3 保留，必须保持复位值。

位 2 **FNE**：FIFO 非空 (FIFO not empty)

此位指示 FIFO 的状态

1：FIFO 包含有效数据

0：FIFO 为空

位 1 **VSYNC**

此位指示配置了适当极性的 DCMI_VSYNC 引脚的状态。

使用内嵌码同步时，此位的含义如下：

0：有效帧

1：在帧之间同步

如果使用内嵌码同步，则仅当 DCMI_CR 中的 CAPTURE 位置 1 时，此位才有意义。

位 0 **HSYNC**

此位指示配置了适当极性的 DCMI_HSYNC 引脚的状态。

使用内嵌码同步时，此位的含义如下：

0：有效行

1：在行之间同步

如果使用内嵌码同步，则仅当 DCMI_CR 中的 CAPTURE 位置 1 时，此位才有意义。

31.7.3 DCMI 原始中断状态寄存器 (DCMI_SR)

DCMI raw interrupt status register

偏移地址：0x08

复位值：0x0000 0x0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LINE_RIS	VSYNC_RIS	ERR_RIS	OVR_RIS	FRAME_RIS
											r	r	r	r	r

DCMI_RIS 用来指示原始中断状态，以只读方式访问。执行读操作时，此寄存器返回的是对应中断未被 DCMI_IER 寄存器屏蔽的原始状态。

位 31:5 保留，必须保持复位值。

位 4 **LINE_RIS**：行原始中断状态 (Line raw interrupt status)

当 DCMI_HSYNC 信号从无效状态更改为有效状态时，此位置 1。即使行无效也会变成高电平。

如果使用内嵌码同步，则仅当 DCMI_CR 中的 CAPTURE 位置 1 时，此位才置 1。

向 DCMI_ICR 中的 LINE_ISC 位写入“1”即可将此位清零。

位 3 **VSYNC_RIS**：DCMI_VSYNC 原始中断状态 (DCMI_VSYNC raw interrupt status)

当 DCMI_VSYNC 信号从无效状态更改为有效状态时，此位置 1。

如果使用内嵌码同步，则仅当 DCMI_CR 中的 CAPTURE 位置 1 时，此位才置 1。

向 DCMI_ICR 中的 VSYNC_ISC 位写入“1”即可将此位清零。

位 2 **ERR_RIS**：同步错误原始中断状态 (Synchronization error raw interrupt status)

0：未检测到同步错误。

1：未按正确顺序接收内嵌码同步字符。

此位仅在内嵌码同步模式下有效。向 DCMI_ICR 中的 ERR_ISC 位写入“1”即可将此位清零。

注： 此位仅适用于内嵌码同步模式。

位 1 **OVR_RIS**：溢出原始中断状态 (Overrun raw interrupt status)

0：未发生数据缓冲区溢出。

1：发生数据缓冲区溢出，数据 FIFO 损坏。

向 DCMI_ICR 中的 OVR_ISC 位写入“1”即可将此位清零。

位 0 **FRAME_RIS**：捕获完成原始中断状态 (Capture complete raw interrupt status)

0：没有新的捕获数据。

1：已捕获一帧。

对一帧数据或裁剪窗口内的数据捕获完毕后，此位置 1。

如果捕获裁剪窗口，则将在裁剪窗口最后一行的行结束时将此位置 1。即使捕获的帧为空（例如，窗口超出帧范围），此位也将置 1。

向 DCMI_ICR 中的 FRAME_ISC 位写入“1”即可将此位清零。

31.7.4 DCMI 中断使能寄存器 (DCMI_IER)

DCMI interrupt enable register

偏移地址：0x0C

复位值：0x0000 0x0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LINE_IE	VSYNC_IE	ERR_IE	OVR_IE	FRAME_IE
											rw	rw	rw	rw	rw

DCMI_IER 寄存器用于使能中断。当其中任一位置 1 时，将使能相应中断。此寄存器可读写。

位 31:5 保留，必须保持复位值。

位 4 **LINE_IE**：使能行中断 (Line interrupt enable)
0：接收到行时不生成中断
1：完全接收到行时生成一个中断

位 3 **VSYNC_IE**：DCMI_VSYNC 中断使能 (DCMI_VSYNC interrupt enable)
0：不生成中断
1：DCMI_VSYNC 每次从无效电平变为有效电平时都生成一个中断
DCMI_VSYNC 信号的有效电平由 VSPOL 位定义。

位 2 **ERR_IE**：同步错误中断使能 (Synchronization error interrupt enable)
0：不生成中断
1：如果未按正确顺序接收内嵌码同步代码，则生成一个中断
注： 此位仅适用于内嵌码同步模式。

位 1 **OVR_IE**：溢出中断使能 (Overrun interrupt enable)
0：不生成中断
1：如果 DMA 无法在收到新数据（32 位）之前传输上一个数据，则生成一个中断

位 0 **FRAME_IE**：捕获完成中断使能 (Capture complete interrupt enable)
0：不生成中断
1：接收完成一帧数据或裁剪窗口内的数据（在裁剪模式下），生成一个中断

31.7.5 DCMI 屏蔽中断状态寄存器 (DCMI_MIS)

DCMI masked interrupt status register

此 DCMI_MIS 寄存器是一个只读寄存器。执行读操作时，此寄存器将返回相应中断的当前屏蔽状态（取决于 DCMI_IER 中相应中断的值）。如果 DCMI_IER 中的相应使能位和 DCMI_RIS 中的相应位都置 1，则此寄存器中的对应位将置 1。

偏移地址：0x10

复位值：0x0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LINE_MIS	VSYNC_MIS	ERR_MIS	OVR_MIS	FRAME_MIS
											r	r	r	r	r

位 31:5 保留，必须保持复位值。

位 4 **LINE_MIS**：行屏蔽中断状态 (Line masked interrupt status)
此位指示屏蔽行中断的状态
0：接收到行时不生成中断。
1：收到完整一行数据且 DCMI_IER 中的 LINE_IE 位置 1 时生成一个中断。

位 3 **VSYNC_MIS**：VSYNC 屏蔽中断状态 (VSYNC masked interrupt status)
此位指示屏蔽 VSYNC 中断的状态
0：DCMI_VSYNC 电平跳变时不生成中断。
1：DCMI_VSYNC 每次从无效电平变为有效电平且 DCMI_IER 中的 VSYNC_IE 位置 1 时都生成一个中断。
DCMI_VSYNC 信号的有效电平由 VSPOL 位定义。

位 2 **ERR_MIS**：同步错误屏蔽中断状态 (Synchronization error masked interrupt status)
此位指示屏蔽同步错误中断
0：发生同步错误时不生成中断。
1：如果未按正确顺序接收内嵌同步码且 DCMI_IER 中的 ERR_IE 位置 1，则生成一个中断。
注： 此位仅适用于内嵌码同步模式。

位 1 **OVR_MIS**：溢出屏蔽中断状态 (Overrun masked interrupt status)
此位指示屏蔽溢出中断的状态
0：发生溢出时不生成中断。
1：如果 DMA 无法在收到新数据（32 位）之前传输上一个数据且 DCMI_IER 中的 OVR_IE 位置 1，则生成一个中断。

位 0 **FRAME_MIS**：捕获完成屏蔽中断状态 (Capture complete masked interrupt status)
此位指示屏蔽捕获完成中断的状态
0：完成捕获后不生成中断。
1：接收完成一帧数据或裁剪窗口内的数据（在裁剪模式下），且 DCMI_IER 中的 FRAME_IE 位置 1 时生成一个中断。

31.7.6 DCMI 中断清零寄存器 (DCMI_ICR)

DCMI interrupt clear register

偏移地址：0x14

复位值：0x0000 0x0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LINE_ISC	VSYNC_ISC	ERR_ISC	OVR_ISC	FRAME_ISC
											w	w	w	w	w

DCMI_ICR 寄存器为只写寄存器。向此寄存器的某一位写入“1”可将 DCMI_RIS 和 DCMI_MIS 寄存器中的相应位清零。写入 “0”则不会带来任何影响。

位 15:5 保留，必须保持复位值。

位 4 **LINE_ISC**：线中断状态清零 (line interrupt status clear)
在此位中写入“1”可将 DCMI_RIS 寄存器中的 LINE_RIS 清零

位 3 **VSYNC_ISC**：垂直同步中断状态清零 (Vertical Synchronization interrupt status clear)
在此位中写入“1”可将 DCMI_RIS 中的 VSYNC_RIS 位清零

位 2 **ERR_ISC**：同步错误中断状态清零 (Synchronization error interrupt status clear)
在此位中写入“1”可将 DCMI_RIS 中的 ERR_RIS 位清零
注： 此位仅适用于内嵌码同步模式。

位 1 **OVR_ISC**：溢出中断状态清零 (Overrun interrupt status clear)
在此位中写入“1”可将 DCMI_RIS 中的 OVR_RIS 位清零

位 0 **FRAME_ISC**：捕获完成中断状态清零 (Capture complete interrupt status clear)
在此位中写入“1”可将 DCMI_RIS 中的 FRAME_RIS 位清零

31.7.7 DCMI 内嵌同步码寄存器 (DCMI_ESCR)

DCMI embedded synchronization code register

偏移地址：0x18

复位值：0x0000 0x0000

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

位 31:24 **FEC**：帧结束分隔码 (Frame end delimiter code)

此字节指定帧结束分隔码。由 4 个字节组成，格式为 0xFF 0x00 0x00 FEC。
如果将 FEC 编程为 0xFF，所有未使用的其它代码 (0xFF0000XY) 都将视为帧结束分隔符。

位 23:16 **LEC**：行结束分隔码 (Line end delimiter code)

此字节指定行结束分隔码。由 4 个字节组成，格式为 0xFF 0x00 0x00 LEC。

位 15:8 **LSC**：行开始分隔码 (Line start delimiter code)

此字节指定行开始分隔码。由 4 个字节组成，格式为 0xFF 0x00 0x00 LSC。

位 7:0 **FSC**：帧开始分隔码 (Frame start delimiter code)

此字节指定帧开始分隔码。由 4 个字节组成，格式为 0xFF 0x00 0x00 FSC。
如果将 FSC 编程为 0xFF，将检测不到任何帧开始分隔符。但在 FEC 代码后第一次出现 LSC 时将视为帧分隔符的开始。

31.7.8 DCMI 内嵌码同步取消屏蔽寄存器 (DCMI_ESUR)

DCMI embedded synchronization unmask register

偏移地址：0x1C

复位值：0x0000 0x0000

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

位 31:24 **FEU**：帧结束分隔符取消屏蔽 (Frame end delimiter unmask)

此字节指定对帧结束分隔码的屏蔽。

0：将帧结束分隔符与收到的数据进行比较时，将屏蔽 DCMI_ESCR 中 FEC 字节中的相应位。

1：将帧结束分隔符与收到的数据进行比较时，将比较 DCMI_ESCR 中 FEC 字节中的相应位

位 23:16 **LEU**：行结束分隔符取消屏蔽 (Line end delimiter unmask)

此字节指定对行结束分隔码的屏蔽。

0：将行结束分隔符与收到的数据进行比较时，将屏蔽 DCMI_ESCR 中 LEC 字节中的相应位

1：将行结束分隔符与收到的数据进行比较时，将比较 DCMI_ESCR 中 LEC 字节中的相应位

位 15:8 **LSU**：行开始分隔符取消屏蔽 (Line start delimiter unmask)

此字节指定对行开始分隔码的屏蔽。

0：将行开始分隔符与收到的数据进行比较时，将屏蔽 DCMI_ESCR 中 LSC 字节中的相应位

1：将行开始分隔符与收到的数据进行比较时，将比较 DCMI_ESCR 中 LSC 字节中的相应位

位 7:0 **FSU**：帧开始分隔符取消屏蔽 (Frame start delimiter unmask)

此字节指定对帧开始分隔码的屏蔽。

0：将帧开始分隔符与收到的数据进行比较时，将屏蔽 DCMI_ESCR 中 FSC 字节中的相应位

1：将帧开始分隔符与收到的数据进行比较时，将比较 DCMI_ESCR 中 FSC 字节中的相应位

31.7.9 DCMI 裁剪窗口起点 (DCMI_CWSTRT)

DCMI crop window start

偏移地址：0x20

复位值：0x0000 0x0000

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

位 31:29 保留，必须保持复位值。

位 28:16 **VST[12:0]**：窗口开始的行数 (Vertical start line count)

图像捕获从此行开始。对之前行的数据不予捕获。

0x0000 => 行 1

0x0001 => 行 2

0x0002 => 行 3

....

位 15:14 保留，必须保持复位值。

位 13:0 **HOFFCNT[13:0]**：窗口开始的水平方向偏移值 (Horizontal offset count)

窗口行内，每行在捕获数据前需空出的像素时钟个数。

31.7.10 DCMI 裁剪窗口大小 (DCMI_CWSIZE)

DCMI crop window size

偏移地址：0x24

复位值：0x0000 0x0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	VLINE13:0]													
		rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	CAPCNT[13:0]													
		rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:30 保留，必须保持复位值。

位 29:16 **VLINE[13:0]**：垂直行计数 (Vertical line count)

窗口内包含的行数。

0x0000 => 1 行

0x0001 => 2 行

0x0002 => 3 行

....

位 15:14 保留，必须保持复位值。

位 13:0 **CAPCNT[13:0]**：捕获计数 (Capture count)

窗口内要捕获的像素时钟数。其值应与表示不同并行接口宽度的字对齐数据相对应。

0x0000 => 1 个像素

0x0001 => 2 个像素

0x0002 => 3 个像素

....

31.7.11 DCMI 数据寄存器 (DCMI_DR)

DCMI data register

偏移地址：0x28

复位值：0x0000 0x0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Byte3								Byte2							
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Byte1								Byte0							
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:24 数据字节 3

位 23:16 数据字节 2

位 15:8 数据字节 1

位 7:0 数据字节 0

数字摄像头接口每收到 32 位数据，才触发一次 DMA 请求。4 字深度的 FIFO 可为 DMA 传输留出足够时间并避免出现 DMA 溢出情况。

31.7.12 DCMI 寄存器映射

表 248 汇总了 DCMI 寄存器。

表 248. DCMI 寄存器地址映射和复位值

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	DCMI_CR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OELS	LSM	OEBS	BSM		Res.	ENABLE	Res.	Res.	EDM		FCRC		VSPOL	HSPOL	PCKPOL	ESS	JPEG	CROP	CM	CAPTURE
	Reset value												0	0	0	0	0		0			0	0	0	0	0	0	0	0	0	0	0	0
0x04	DCMI_SR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FNE	VSYNC	HSYNC
	Reset value																														0	0	0
0x08	DCMI_RIS	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LINE_RIS	VSYNC_RIS	ERR_RIS	OVR_RIS	FRAME_RIS
	Reset value																												0	0	0	0	0
0x0C	DCMI_IER	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LINE_IE	VSYNC_IE	ERR_IE	OVR_IE	FRAME_IE
	Reset value																												0	0	0	0	0
0x10	DCMI_MIS	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LINE_MIS	VSYNC_MIS	ERR_MIS	OVR_MIS	FRAME_MIS
	Reset value																												0	0	0	0	0
0x14	DCMI_ICR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LINE_ISC	VSYNC_ISC	ERR_ISC	OVR_ISC	FRAME_ISC
	Reset value																												0	0	0	0	0
0x18	DCMI_ESCR	FEC								LEC								LSC								FSC							
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x1C	DCMI_ESUR	FEU								LEU								LSU								FSU							
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x20	DCMI_CWSTRT	Res.	Res.	Res.	VST[12:0]													Res.	Res.	HOFFCNT[13:0]													
	Reset value				0	0	0	0	0	0	0	0	0	0	0	0	0			0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x24	DCMI_CWSIZE	Res.	Res.	VLINE13:0]														Res.	Res.	CAPCNT[13:0]													
	Reset value			0	0	0	0	0	0	0	0	0	0	0	0	0	0			0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x28	DCMI_DR	Byte3								Byte2								Byte1								Byte0							
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

有关寄存器边界地址的信息，请参见*第 100 页的第 2.2.2 节*。

32 LCD-TFT 显示控制器 (LTDC)

32.1 简介

LCD-TFT（液晶显示器——薄膜晶体管）显示器控制器提供并行数字 RGB（红色、绿色、蓝色）以及水平同步、垂直同步、像素时钟和数据使能信号，这些信号直接输出到不同 LCD 和 TFT 面板的接口。

32.2 LTDC 主要特性

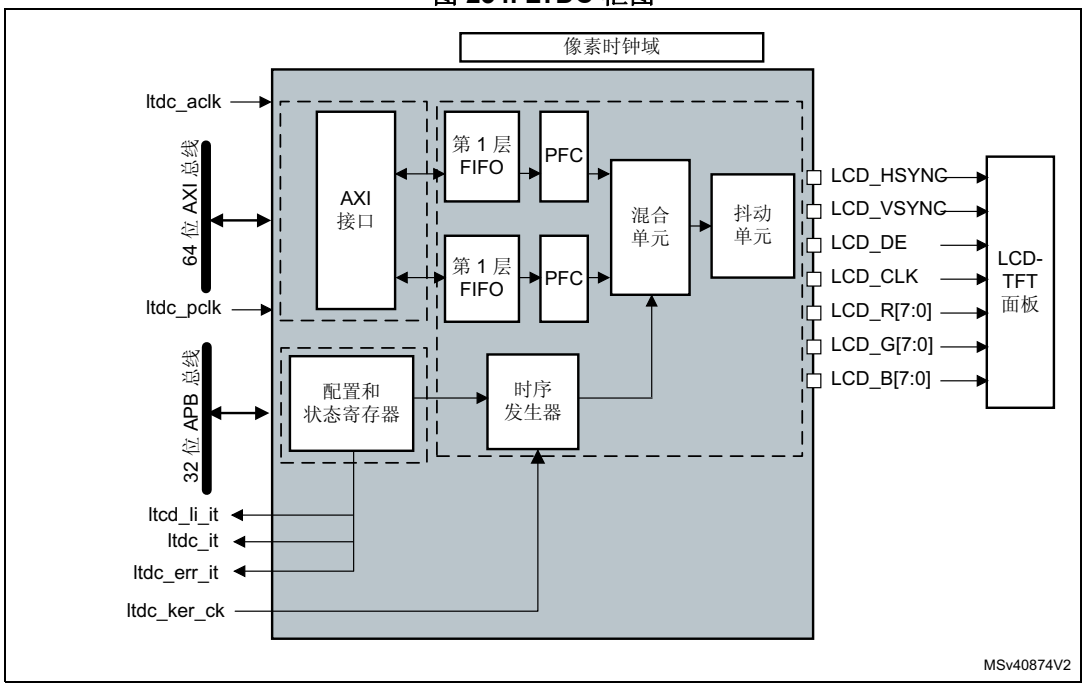
- 24 位 RGB 并行像素输出；每像素 8 位 (RGB888)
- 2 个带有专用 FIFO 的显示层（FIFO 深度 64x64 位）
- 查色表 (CLUT)，每个图层最高 256 种颜色（256x24 位）
- 可针对不同显示面板编程时序
- 可编程背景色
- 可编程 HSync、VSync 和数据使能信号的极性
- 每层有多达 8 个输入颜色格式可供选择
  - ARGB8888
  - RGB888
  - RGB565
  - ARGB1555
  - ARGB4444
  - L8（8 位 Luminance 或 CLUT）
  - AL44（4 位 alpha + 4 位 luminance）
  - AL88（8 位 alpha + 8 位 luminance）
- 每通道的低位采用伪随机抖动输出
  - 红色、绿色、蓝色的抖动宽度为 2 位
- 使用 alpha 值（每像素或常数）在两层之间灵活混合
- 色键（透明颜色）
- 可编程窗口位置和大小
- 支持薄膜晶体管 (TFT) 彩色显示器
- AXI 主接口支持 16 个双字的突发
- 最高 4 个可编程中断事件

32.3 LTDC 功能说明

32.3.1 LTDC 框图

LTDC 的框图如*图 234：LTDC 框图*所示。

图 234. LTDC 框图



层 FIFO：每层一个 64x64 位 FIFO。

PFC：执行像素格式转换的像素格式转换器，从层的所选输入像素格式转换为字。

AXI 接口：用于将数据从存储器传输到 FIFO。

有关混合单元、抖动单元和时序发生器的信息：请参见*第 32.4.1 节*和*第 32.4.2 节*。

32.3.2 LCD-TFT 内部信号

表 249 列出了 LCD-TFT 内部信号。

表 249. LCD-TFT 内部信号

名称	信号类型	说明
ltdc_aclk	数字输入	AXI 时钟域的 LCD-TFT 寄存器时钟
ltdc_pclk	数字输入	LCD-TFT 寄存器接口时钟
ltdc_ker_ck	数字输入	用于生成 LCD_CLK （像素时钟）的 LCD-TFT 内核时钟
ltdc_li_it	数字输出	用于 MDMA 的 LCD-TFT 线中断触发
ltdc_it	数字输出	LCD-TFT 全局中断请求
ltdc_err_it	数字输出	LCD-TFT 全局错误中断请求

32.3.3 LCD-TFT 引脚和外部信号接口

表 250 总结了 LTDC 信号接口。

表 250. LCD-TFT 引脚和信号接口

LCD-TFT 信号	I/O	说明
LCD_CLK	O	时钟输出
LCD_HSYNC	O	水平同步
LCD_VSYNC	O	垂直同步
LCD_DE	O	非数据使能
LCD_R[7:0]	O	数据：8 位红色数据
LCD_G[7:0]	O	数据：8 位绿色数据
LCD_B[7:0]	O	数据：8 位蓝色数据

必须通过用户程序配置 LTDC-TFT 控制器引脚。未使用的引脚可用于其他功能。

对于高达 24 位 (RGB888) 的 LTDC 输出，如果使用低于 8bpp 的像素深度将 RGB565 或 RGB666 输出到 16 位或 18 位显示器，则 RGB 显示数据线必须连接到 LCD-TFT 控制器 RGB 数据线的 MSB。例如，当 LCD-TFT 控制器与 RGB565 16 位显示器相连时，LCD 显示器的 R[4:0]、G[5:0] 和 B[4:0] 数据线引脚必须连接至 LCD-TFT 控制器的 LCD_R[7:3]、LCD_G[7:2] 和 LCD_B[7:3]。

32.3.4 LTDC 复位和时钟

LCD-TFT 控制器外设使用 3 个时钟域：

- AXI 时钟域 (ltdc_aclk)

  该时钟域包含 LCD-TFT AXI 主接口，该接口用于将数据从存储器传输到 FIFO 层和帧缓冲区配置寄存器中。
- APB 时钟域 (ltdc_pclk)

  该时钟域包含全局配置寄存器和中断寄存器。
- 像素时钟域 (LCD_CLK)

  该时钟域包含像素数据生成、层配置寄存器以及 LCD-TFT 接口信号发生器。LCD_CLK 输出应按照面板要求配置。LCD_CLK 由特定的 PLL 输出生成（请参见“复位和时钟控制”部分）。

表 251 汇总了各个寄存器的时钟域。

表 251. 各个寄存器的时钟域

LTDC 寄存器	时钟域
LTDC_LxCR	ltdc_aclk
LTDC_LxCFBAR	
LTDC_LxCFBLR	
LTDC_LxCFBLNR	

表 251. 各个寄存器的时钟域 （续）

LTDC 寄存器	时钟域
LTDC_SRCR	ltdc_pclk
LTDC_IER	
LTDC_ISR	
LTDC_ICR	
LTDC_SSCR	像素时钟 (LCD_CLK)
LTDC_BPCR	
LTDC_AWCR	
LTDC_TWCR	
LTDC_GCR	
LTDC_BCCR	
LTDC_LIPCR	
LTDC_CPSR	
LTDC_CDSR	
LTDC_LxWHPCR	
LTDC_LxWVPCR	
LTDC_LxCKCR	
LTDC_LxPFCR	
LTDC_LxCACR	
LTDC_LxDCCR	
LTDC_LxBFCR	
LTDC_LxCLUTWR	

访问 LTDC 寄存器时必须小心，APB 总线在访问期间会停止并持续一段给定的时间（请参见 *表 252*）。

表 252. LTDC 寄存器访问和更新持续时间

	寄存器时钟域		
	AXI 域	APB 域	像素时钟域
寄存器读访问持续时间	7 x ltdc_pclk + 5 x ltdc_aclk	7 x ltdc_pclk	7 x ltdc_pclk + 5 x ltdc_ker_clk
寄存器写访问持续时间	6 x ltdc_pclk + 5 x ltdc_aclk	6 x ltdc_pclk	6 x ltdc_pclk + 5 x ltdc_ker_clk

通过将 RCC_APBRSTR 寄存器中的相应位置 1 可将 LCD 控制器复位。这将复位三个时钟域。

32.4 LTDC 可编程参数

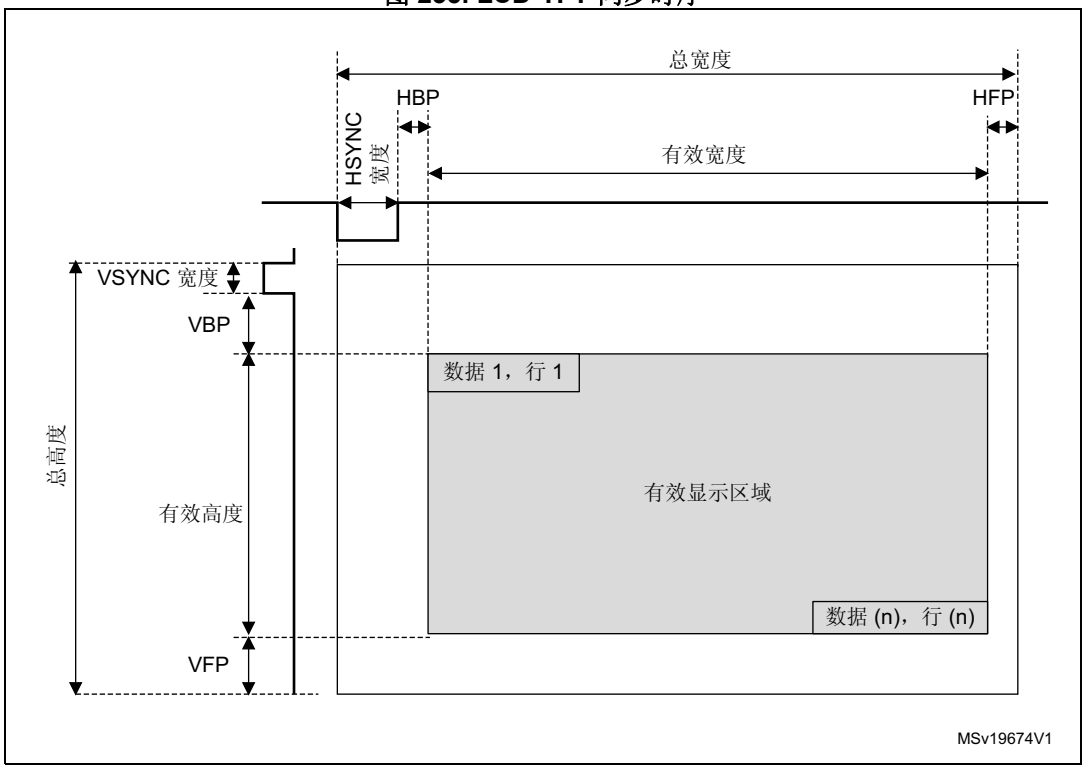
LCD-TFT 控制器提供灵活的可配置参数。其可通过 **LTDC_GCR** 寄存器使能或禁止。

32.4.1 LTDC 全局配置参数

同步时序：

图 235 显示了框图*图 234* 中所示的同步时序发生器模块生成的可配置时序参数。该模块生成水平和垂直同步时序面板信号、像素时钟和数据使能信号。

图 235. LCD-TFT 同步时序



注： *HBP 和 HFP 分别为水平后沿周期和水平前沿周期。*

VBP 和 VFP 分别为垂直后沿周期和垂直前沿周期。

LCD-TFT 可编程同步时序包括：

- HSYNC 和 VSYNC 宽度：水平和垂直同步宽度，通过编程 **LTDC_SSCR** 寄存器中的 ***HSYNC Width - 1*** 和 ***VSYNC Width - 1*** 的值进行配置。
- HBP 和 VBP：水平和垂直同步后沿宽度，通过编程 **LTDC_BPCR** 寄存器中的累加值 ***HSYNC Width + HBP - 1*** 和累加值 ***VSYNC Width + VBP - 1*** 进行配置。
- 有效宽度和有效高度：有效宽度和有效高度通过编程 **LTDC_AWCR** 寄存器中的累加值 ***HSYNC Width + HBP + Active Width - 1*** 和累加值 ***VSYNC Width + VBP + Active Height - 1*** 进行配置（仅支持最高 1024x768）。
- 总宽度：总宽度通过编程 **LTDC_TWCR** 寄存器中的累加值 ***HSYNC Width + HBP + Active Width + HFP - 1*** 进行配置。HFP 为水平前沿周期。
- 总高度：总高度通过编程 **LTDC_TWCR** 寄存器中的累加值 ***VSYNC Height + VBP + Active Height + VFP - 1*** 进行配置。VFP 为垂直前沿周期。

注： *使能 LTDC 时，产生的时序以 X/Y=0/0 位置作为垂直同步区域中的第一个水平同步像素，随后是后沿、有效数据显示区域和前沿。*

禁止 LTDC 时，时序发生器模块复位为 X= *总宽度 - 1*、Y= *总高度- 1*，并在垂直同步阶段和 FIFO 刷新前保持上一个像素。因此，仅连续输出消隐数据。

同步时序配置示例：

TFT-LCD 时序（应从面板数据手册中提取）：

- 水平和垂直同步宽度：0xA 像素，0x2 行
- 水平和垂直后沿：0x14 像素，0x2 行
- 有效宽度和有效高度：0x140 像素，0xF0 行 (320x240)
- 水平前沿：0xA 像素
- 垂直前沿：0x4 行

LTDC 时序寄存器中编程的值将为：

- LTDC_SSCR 寄存器：将编程为 0x00090001。（HSW[11:0] 为 0x9 且 VSH[10:0] 为 0x1）
- LTDC_BPCR 寄存器：将编程为 0x001D0003。（AHBP[11:0] 为 0x1D(0xA+ 0x13)，AVBP[10:0] 为 0x3(0x2 + 0x1)）
- LTDC_AWCR 寄存器：将编程为 0x015D00F3。（AAW[11:0] 为 0x15D(0xA +0x14 +0x13F)，AAH[10:0] 为 0xF3(0x2 +0x2 + 0xEF)）
- LTDC_TWCR 寄存器：将编程为 0x00000167。（TOTALW[11:0] 为 0x167(0xA +0x14 + 0x140 + 0x9)）
- LTDC_THCR 寄存器：将编程为 0x000000F7。（TOTALH[10:0] 为 0xF7(0x2 +0x2 + 0xF0 + 3)）

可编程极性

水平和垂直同步、数据使能和像素时钟输出信号的极性可通过 **LTDC_GCR** 寄存器编程为高电平有效或低电平有效。

背景色

恒定的背景色 (RGB888) 可通过 **LTDC_BCCR** 寄存器编程。它用于与底层混合。

抖动

使用 LFSR 的伪随机抖动技术用于向各个像素颜色通道值（R、G或B）添加小的随机值（阈值），从而当 18 位显示器上显示 24 位数据时，可在某些情况下对 MSB 进行舍入操作。因此，抖动技术用于对各帧中不同的数据进行舍入操作。

伪随机抖动技术的过程是将 LSB 与阈值比较，并在 LSB 部分 >= 阈值时，仅向 MSB 部分加 1。一旦应用抖动技术，通常会减少 LSB。

添加的伪随机值的宽度为每个颜色通道 2 位；红色 2 位、绿色 2 位及蓝色 2 位。

使能 LCD-TFT 控制器后，LFSR 以第一个有效像素开始运行，并且即使在消隐周期内和抖动关闭时也保持运行状态。如果禁止 LTDC，LFSR 将复位。

可通过 **LTDC_GCR** 寄存器实时开启和关闭抖动。

重载影子寄存器

一些配置寄存器执行影子操作。对活动寄存器执行写操作时，或在 **LTDC_SRCR** 寄存器配置阶段之后的垂直消隐周期开始时，可将影子寄存器值立即重载到活动寄存器中。如果选择了立即重载配置，则只应在所有新寄存器完成写操作后激活重载。

不应在重载完成前再次修改影子寄存器。读取影子寄存器将返回实际有效值。新写入的值只能在重载发生后读取。

如果在 **LTDC_IER** 寄存器中相应使能，则可产生寄存器重载中断。

影子寄存器均为第 1 层和第 2 层寄存器，但 **LTDC_LxCLUTWR** 寄存器除外。

中断产生事件

有关中断配置，请参见*第 32.5 节：LTDC 中断*。

32.4.2 层可编程参数

最多可单独使能、禁止和配置两个层。层显示顺序固定，即自下而上。如果使能两个层，则层 2 为顶部显示窗口。

窗口

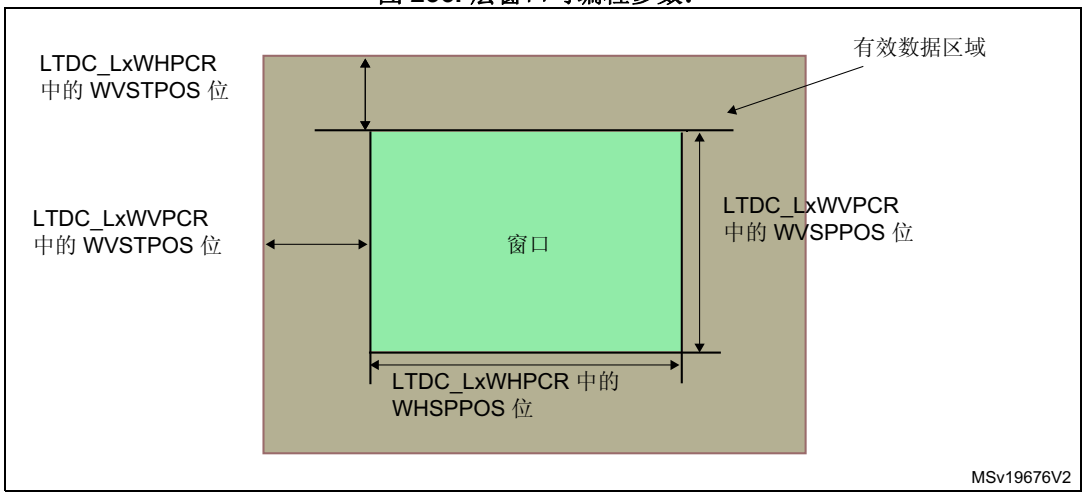
可为每个层定位和调整大小，各个层必须位于有效显示区域内。

窗口位置和大小通过左上和右下的 X/Y 位置以及包含同步、后沿大小和有效数据区域的内部时序发生器配置。请参见 **LTDC_LxWHPCR** 和 **LTDC_WVPCR** 寄存器。

可编程层位置和大小定义了一行中的第一个/最后一个可见像素和窗口中的第一个/最后一个可见行。它允许显示完整的图像帧，也允许只显示图像帧的一部分。请参见*图 236*。

- 层中的第一个和最后一个可见像素通过配置 **LTDC_LxWHPCR** 寄存器中的 WHSTPOS[11:0] 和 WHSPPOS[11:0] 进行设置。
- 层中的第一个和最后一个可见行通过配置 **LTDC_LxWVPCR** 寄存器中的 WVSTPOS[10:0] 和 WVSPPOS[10:0] 进行设置。

图 236. 层窗口可编程参数：



像素输入格式

可编程像素格式用于层的帧缓冲区中存储的数据。

可通过 **LTDC_LxPFCR** 寄存器为每个层配置多达 8 个输入像素格式。

像素数据从帧缓冲区中读取，随后按照以下方式转换为内部 8888 (ARGB) 格式：

- 宽度低于 8 位的分量通过位重复扩展到 8 位。所选位范围多次拼接，直至其超过 8 位。在得到的向量中，选择高 8 位。例如：5 位 RGB565 红色通道将变为（位位置）：43210432（低 3 位由 5 位中的高 3 位填充）。

下图说明了像素数据映射与所选格式的关系。

表 253. 像素数据映射与颜色格式的关系

ARGB8888			
@+3 A_x[7:0]	@+2 R_x[7:0]	@+1 G_x[7:0]	@ B_x[7:0]
@+7 A_{x+1}[7:0]	@+6 R_{x+1}[7:0]	@+5 G_{x+1}[7:0]	@+4 B_{x+1}[7:0]
RGB888			
@+3 B_{x+1}[7:0]	@+2 R_x[7:0]	@+1 G_x[7:0]	@ B_x[7:0]
@+7 G_{x+2}[7:0]	@+6 B_{x+2}[7:0]	@+5 R_{x+1}[7:0]	@+4 G_{x+1}[7:0]
RGB565			
@+3 R_{x+1}[4:0] G_{x+1}[5:3]	@+2 G_{x+1}[2:0] B_{x+1}[4:0]	@+1 R_x[4:0] G_x[5:3]	@ G_x[2:0] B_x[4:0]
@+7 R_{x+3}[4:0] G_{x+3}[5:3]	@+6 G_{x+3}[2:0] B_{x+3}[4:0]	@+5 R_{x+2}[4:0] G_{x+2}[5:3]	@+4 G_{x+2}[2:0] B_{x+2}[4:0]
ARGB1555			
@+3 A_{x+1}[0]R_{x+1}[4:0] G_{x+1}[4:3]	@+2 G_{x+1}[2:0] B_{x+1}[4:0]	@+1 A_x[0] R_x[4:0] G_x[4:3]	@ G_x[2:0] B_x[4:0]
@+7 A_{x+3}[0]R_{x+3}[4:0] G_{x+3}[4:3]	@+6 G_{x+3}[2:0] B_{x+3}[4:0]	@+5 A_{x+2}[0]R_{x+2}[4:0]G_{x+2}[4:3]	@+4 G_{x+2}[2:0] B_{x+2}[4:0]
ARGB4444			
@+3 A_{x+1}[3:0]R_{x+1}[3:0]	@+2 G_{x+1}[3:0] B_{x+1}[3:0]	@+1 A_x[3:0] R_x[3:0]	@ G_x[3:0] B_x[3:0]
@+7 A_{x+3}[3:0]R_{x+3}[3:0]	@+6 G_{x+3}[3:0] B_{x+3}[3:0]	@+5 A_{x+2}[3:0]R_{x+2}[3:0]	@+4 G_{x+2}[3:0] B_{x+2}[3:0]

表 253. 像素数据映射与颜色格式的关系（续）

ARGB8888			
L8			
@+3 L_{x+3}[7:0]	@+2 L_{x+2}[7:0]	@+1 L_{x+1}[7:0]	@ L_x[7:0]
@+7 L_{x+7}[7:0]	@+6 L_{x+6}[7:0]	@+5 L_{x+5}[7:0]	@+4 L_{x+4}[7:0]
AL44			
@+3 A_{x+3}[3:0] L_{x+3}[3:0]	@+2 A_{x+2}[3:0] L_{x+2}[3:0]	@+1 A_{x+1}[3:0] L_{x+1}[3:0]	@ A_x[3:0] L_x[3:0]
@+7 A_{x+7}[3:0] L_{x+7}[3:0]	@+6 A_{x+6}[3:0] L_{x+6}[3:0]	@+5 A_{x+5}[3:0] L_{x+5}[3:0]	@+4 A_{x+4}[3:0] L_{x+4}[3:0]
AL88			
@+3 A_{x+1}[7:0]	@+2 L_{x+1}[7:0]	@+1 A_x[7:0]	@ L_x[7:0]
@+7 A_{x+3}[7:0]	@+6 L_{x+3}[7:0]	@+5 A_{x+2}[7:0]	@+4 L_{x+2}[7:0]

查色表 (CLUT)

可在运行时通过 **LTDC_LxCR** 寄存器为每个层使能 CLUT，CLUT 仅在使用 L8、AL44 和 AL88 输入像素格式时适用于索引色。

首先，CLUT 必须加载用于替换相应像素（索引色）的原始 R、G 和 B 值的 R、G 和 B 值。每个颜色（RGB 值）在 CLUT 内都有自己对应的地址。

R、G 和 B 值及其各自的地址均通过 **LTDC_LxCLUTWR** 寄存器编程。

- 在使用 L8 和 AL88 输入像素格式时，CLUT 必须加载 256 个颜色。各颜色的地址在 **LTDC_LxCLUTWR** 寄存器的 CLUTADD 位中配置。
- 在使用 AL44 输入像素格式时，CLUT 必须仅加载 16 个颜色。各颜色的地址必须通过将 4 位 L 通道重复为 8 位进行填充，具体如下：
  - L0（索引色 0），地址 0x00 处
  - L1，地址 0x11 处
  - L2，地址 0x22 处
  - .....
  - L15，地址 0xFF 处

颜色帧缓冲区地址

每个层的颜色帧缓冲区均有一个起始地址，该地址通过 **LTDC_LxCFBAR** 寄存器进行配置。

当使能某个层时，将从颜色帧缓冲区中获取该数据。

颜色帧缓冲区长度

每层均设置颜色帧缓冲区的总行长（单位为字节）和行数，二者可分别通过 **LTDC_LxCFBLR** 和 **LTDC_LxCFBLNR** 寄存器进行配置。

行长和行数的设置用于阻止超出帧缓存结尾的数据被预取到层对应的 FIFO 中。

- 如果设置为低于所需字节，则会产生 FIFO 下溢中断（如果之前已使能）。
- 如果设置为高于实际所需字节，则将丢弃从 FIFO 中读取的无用数据。无用数据不会显示。

颜色帧缓冲区间距

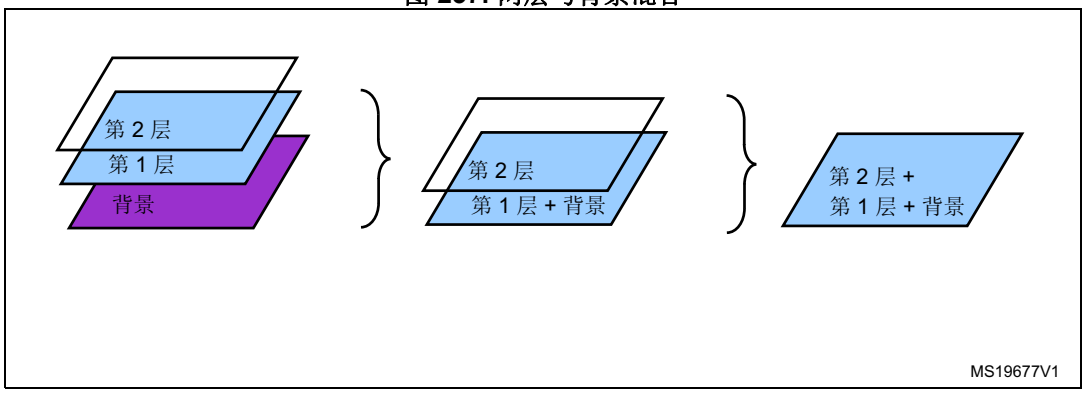
每层的颜色帧缓冲区均具有可配置间距，此间距是一行的开始与下一行开始的距离（以字节为单位）。它通过 **LTDC_LxCFBLR** 寄存器配置。

层混合

混合操作始终有效，两层可按照 **LTDC_LxBFCR** 寄存器中配置的混合系数进行混合。

混合顺序固定，即由下至上。如果使能了两层，首先第 1 层将与背景色混合，随后第 2 层与第 1 层和背景的混合颜色结果再次混合。请参见*图 237*。

图 237. 两层与背景混合



默认颜色

每层可具有 ARGB 格式的默认颜色，该颜色在定义的层窗口外使用或在层禁止时使用。

默认颜色通过 **LTDC_LxDCCR** 寄存器配置。

始终在两层间执行混合操作，即便其中一层禁止也是如此。要避免层禁止时显示默认颜色，需将 LTDC_LxBFCR 寄存器中此层的混合系数设置为其复位值。

色键

色键 (RGB) 可配置为代表透明像素。

使能色键后，当前像素（格式转换后、CLUT 分别混合前的像素）将与色键进行比较。如果当前像素与编程的 RGB 值相匹配，则该像素的所有通道 (ARGB) 均设置为 0。

运行时，可配置色键值并用其替换像素 RGB 值。

色键通过 **LTDC_LxCKCR** 寄存器配置。

色键通过 LTDC_LxCKCR 寄存器配置。编程值取决于像素格式，因为它会在像素格式转换为 ARGB888 后与当前像素进行比较。

示例：如果将中黄色（50% 红 + 50% 绿）用作透明色键，则：

- 在 RGB565 中，中黄色为 0x8400。将 LTDC_LxCKCR 设为 0x848200。
- 在 ARGB8888 中，中黄色为 0x808000，将 LTDC_LxCKCR 设为 0x808000。
- 在所有基于 CLUT 的颜色模式（L8、AL88、AL44）下，将其中一个调色板条目设为中黄色 0x808000，将 LTDC_LxCKCR 设为 0x808000。

32.5 LTDC 中断

LTDC 提供四个可屏蔽中断，这些中断经逻辑或运算到两个中断向量。

中断源可通过 **LTDC_IER** 寄存器单独使能或禁止。将相应的屏蔽位置 1 可使能相应中断。

发生如下事件时会产生两个中断：

- 行中断：达到被编程的行时产生。行中断的位置在 LTDC_LIPCR 寄存器中编程
- 寄存器重载中断：在垂直消隐周期内执行影子寄存器重载时产生
- FIFO 下溢中断：从空的层 FIFO 中请求像素时产生
- 传输错误中断：数据传输期间出现 AXI 总线错误时产生

这些中断事件与 NVIC 控制器相连，如下图所示。

图 238. 中断事件

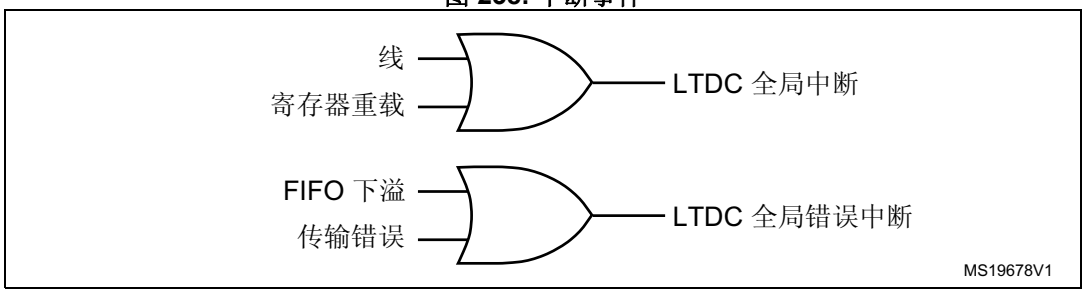


表 254. LTDC 中断请求

中断事件	事件标志	使能控制位
行	LIF	LIE
寄存器重载	RRIF	RRIEN
FIFO 下溢	FUDERRIF	FUDERRIE
传输错误	TERRIF	TERRIE

32.6 LTDC 编程步骤

- 在 RCC 寄存器中使能 LTDC 时钟。
- 按照面板数据表配置所需像素时钟。
- 按照中所述的面板数据表配置同步时序：VSYNC、HSYNC、垂直和水平后沿、有效数据区域以及前沿时序*第 32.4.1 节：LTDC 全局配置参数*。
- 配置 **LTDC_GCR** 寄存器中的同步信号和时钟极性。
- 必要时，配置 **LTDC_BCCR** 寄存器中的背景色。
- 配置 **LTDC_IER** 和 **LTDC_LIPCR** 寄存器中的所需中断。
- 通过执行以下编程操作配置第 1/2 层的参数：
  - 编程 **LTDC_LxWHPCR** 和 **LTDC_WVPCR** 寄存器中的层窗口的水平和垂直位置。层窗口必须位于有效数据区域。
  - 编程 **LTDC_LxPFCR** 寄存器中的像素输入格式。
  - 编程 **LTDC_LxCFBAR** 寄存器中的颜色帧起始地址。
  - 编程 **LTDC_LxCFBLR** 寄存器中的颜色帧缓冲区的行长和间距。
  - 编程 **LTDC_LxCFBLNR** 寄存器中的颜色帧缓冲区的行数。
  - 必要时，在 **LTDC_LxCLUTWR** 寄存器中为 CLUT 加载 RGB 值及其地址。
  - 必要时，分别在 **LTDC_LxDCCR** 和 **LTDC_LxBFCR** 寄存器中配置默认颜色和混合系数。
- 使能 **LTDC_LxCR** 寄存器中的第 1/2 层，必要时使能 CLUT。
- 必要时，可分别在 **LTDC_GCR 和 LTDC_LxCKCR** 寄存器中使能抖动和色键。也可以实时使能这两个功能。
- 通过 **LTDC_SRCR** 寄存器将影子寄存器重载到活动寄存器中。
- 使能 **LTDC_GCR** 寄存器中的 LCD-TFT 控制器。
- 除 CLUT 外，所有层参数均可实时修改。新配置必须通过配置 **LTDC_SRCR** 寄存器立即重载或在垂直消隐周期内重载。

注： *所有层的寄存器均执行影子操作。一旦对某个寄存器执行写操作，便不应在重载完成前再次进行修改。因此，如果在尚未重载时对同一寄存器执行新的写操作，则将覆盖之前的配置。*

32.7 LTDC 寄存器

32.7.1 LTDC 同步大小配置寄存器 (LTDC_SSCR)

LTDC Synchronization Size Configuration Register

此寄存器定义水平同步像素数减 1 以及垂直同步行数减 1。有关配置的示例，请参见*图 235* 和*第 32.4 节：LTDC 可编程参数*。

偏移地址：0x08

复位值：0x0000 0000

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

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

位 31:28 保留，必须保持复位值

位 27:16 **HSW[11:0]**：水平同步宽度 (Horizontal Synchronization Width)（以像素时钟周期为单位）
这些位定义水平同步像素数减 1。

位 15:11 保留，必须保持复位值

位 10:0 **VSH[10:0]**：垂直同步高度 (Vertical Synchronization Height)（以水平扫描行为单位）
这些位定义垂直同步高度减 1。它表示水平同步行数。

32.7.2 LTDC 后沿配置寄存器 (LTDC_BPCR)

LTDC Back Porch Configuration Register

此寄存器定义水平同步像素加水平后沿像素的累加数减 1（**HSYNC 宽度 + HBP- 1**）以及垂直同步行加垂直后沿行的累加数减 1（**VSYNC 高度 + VBP - 1**）。有关配置的示例，请参见*图 235* 和*第 32.4 节：LTDC 可编程参数*。

偏移地址：0x0C

复位值：0x0000 0000

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

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

位 31:28 保留，必须保持复位值

位 27:16 **AHBP[11:0]**：累加水平后沿 (Accumulated Horizontal back porch)（以像素时钟周期为单位）
这些位定义累加水平后沿宽度（水平同步像素加水平后沿像素减 1）。
水平后沿是水平同步信号变为无效到下一扫描行的有效显示开始之间的间隔。

位 15:11 保留，必须保持复位值

位 10:0 **AVBP[10:0]**：累加垂直后沿 (Accumulated Vertical back porch)（以水平扫描行为单位）
这些位定义累加垂直后沿宽度（垂直同步行加垂直后沿行减 1）。
垂直后沿是帧开始到下一帧的首个有效扫描行开始所包含的水平扫描行的数量。

32.7.3 LTDC 有效宽度配置寄存器 (LTDC_AWCR)

LTDC Active Width Configuration Register

此寄存器定义水平同步加水平后沿加有效像素的累加数减 1（**HSYNC 宽度 + HBP + 有效宽度 - 1**）以及垂直同步行加垂直后沿行加有效行的累加数减 1（**VSYNC 高度+ BVBP + 有效高度 - 1**）。有关配置的示例，请参见*图 235* 和*第 32.4 节：LTDC 可编程参数*。

偏移地址：0x10

复位值：0x0000 0000

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

位 31:28 保留，必须保持复位值

位 27:16 **AAW[11:0]**：累加有效宽度 (Accumulated Active Width)（以像素时钟周期为单位）
这些位定义累加有效宽度（水平同步像素加水平后沿像素加有效像素减 1）。
有效宽度是面板扫描行的有效显示区中的像素数。
有关最大像素时钟支持的最大有效宽度，请参见器件数据手册。

位 15:11 保留，必须保持复位值

位 10:0 **AAH[10:0]**：累加有效高度 (Accumulated Active Height)（以水平扫描行为单位）
这些位定义累加高度（垂直同步行加垂直后沿行加有效高度行减 1）。有效高度是面板中的有效行数。
有关最大像素时钟支持的最大有效高度，请参见器件数据手册。

32.7.4 LTDC 总宽度配置寄存器 (LTDC_TWCR)

LTDC Total Width Configuration Register

此寄存器定义水平同步加水平后沿加有效像素加水平前沿的像素累加数减 1（**HSYNC 宽度 + HBP + 有效宽度 + HFP - 1**）以及垂直同步行加垂直后沿行加有效行加垂直前沿行的累加数减 1（**VSYNC 高度 + BVBP + 有效高度 + VFP - 1**）。有关配置的示例，请参见*图 235* 和*第 32.4 节：LTDC 可编程参数*。

偏移地址：0x14

复位值：0x0000 0000

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

位 31:28 保留，必须保持复位值

位 27:16 **TOTALW[11:0]**：总宽度 (Total Width)（以像素时钟周期为单位）
这些位定义累加总宽度（水平同步像素加水平后沿像素加有效宽度加水平前沿像素减 1）。

位 15:11 保留，必须保持复位值

位 10:0 **TOTALH[10:0]**：总高度 (Total Height)（以水平扫描行为单位）
这些位定义累加高度（垂直同步行加垂直后沿行加有效高度加垂直前沿行减 1）。

32.7.5 LTDC 全局控制寄存器 (LTDC_GCR)

LTDC Global Control Register

此寄存器定义 LCD-TFT 控制器的全局配置。

偏移地址：0x18

复位值：0x0000 2220

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
HSPOL	VSPOL	DEPOL	PCPOL	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DEN
rw	rw	rw	rw												rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	DRW[2:0]			Res.	DGW[2:0]			Res.	DBW[2:0]			Res.	Res.	Res.	LTDCEN
	r	r	r		r	r	r		r	r	r				rw

位 31 **HSPOL**：水平同步极性 (Horizontal synchronization polarity)
此位由软件置 1 和清零。
0：水平同步极性低电平有效
1：水平同步极性高电平有效

位 30 **VSPOL**：垂直同步极性 (Vertical synchronization polarity)
此位由软件置 1 和清零。
0：垂直同步低电平有效
1：垂直同步高电平有效

位 29 **DEPOL**：非数据使能极性 (Not Data Enable Polarity)
此位由软件置 1 和清零。
0：非数据使能极性低电平有效
1：非数据使能极性高电平有效

位 28 **PCPOL**：像素时钟极性 (Pixel Clock Polarity)
此位由软件置 1 和清零。
0：像素时钟极性低电平有效
1：像素时钟高电平有效

位 27:17 保留，必须保持复位值

位 16 **DEN**：抖动使能 (Dither Enable)
此位由软件置 1 和清零。
0：禁止抖动
1：使能抖动

位 15 保留，必须保持复位值

位 14:12 **DRW[2:0]**：抖动红色宽度 (Dither Red Width)
这些位返回抖动红色位

位 11 保留，必须保持复位值

位 10:8 **DGW[2:0]**：抖动绿色宽度 (Dither Green Width)
这些位返回抖动绿色位

位 7 保留，必须保持复位值

位 6:4 **DBW[2:0]**：抖动蓝色宽度 (Dither Blue Width)
这些位返回抖动蓝色位

位 3:1 保留，必须保持复位值

位 0 **LTDCEN**：LCD-TFT 控制器使能位 (LCD-TFT controller enable bit)
此位由软件置 1 和清零。
0：禁止 LTDC
1：使能 LTDC

32.7.6 LTDC 影子重载配置寄存器 (LTDC_SRCR)

LTDC Shadow Reload Configuration Register

此寄存器允许立即或在垂直消隐周期内将影子寄存器的值重载到活动寄存器中。影子寄存器均为第 1 层和第 2 层寄存器，但 LTDC_L1CLUTWR 和 LTDC_L2CLUTWR 除外。

偏移地址：0x24

复位值：0x0000 0000

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

位 31:2 保留，必须保持复位值

位 1 **VBR**：垂直消隐重载 (Vertical Blanking Reload)

此位由软件置 1，只有重载后才由硬件清零。（一旦置 1，便无法通过寄存器写操作清零）。

0：无影响

1：影子寄存器在垂直消隐周期（有效显示区后的第一行开始时）内重载

位 0 **IMR**：立即重载 (Immediate Reload)

此位由软件置 1，只有重载后才由硬件清零。

0：无影响

1：影子寄存器立即重载

注： *影子寄存器回读有效值。直至重载完成，才会读取“旧”值。*

32.7.7 LTDC 背景色配置寄存器 (LTDC_BCCR)

LTDC Background Color Configuration Register

此寄存器定义背景色 (RGB888)。

偏移地址：0x2C

复位值：0x0000 0000

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

位 31:24 保留，必须保持复位值

位 23:16 **BCRED[7:0]**：背景红色值 (Background Color Red value)
这些位配置背景红色值

位 15:8 **BCGREEN[7:0]**：背景绿色值 (Background Color Green value)
这些位配置背景绿色值

位 7:0 **BCBLUE[7:0]**：背景蓝色值 (Background Color Blue value)
这些位配置背景蓝色值

32.7.8 LTDC 中断使能寄存器 (LTDC_IER)

LTDC Interrupt Enable Register

此寄存器通过将对应的位置 1 来确定哪一个状态标志位产生中断请求。

偏移地址：0x34

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RRIE	TERRIE	FUIE	LIE
												rw	rw	rw	rw

位 31:4 保留，必须保持复位值

位 3 **RRIE**：寄存器重载中断使能 (Register Reload interrupt enable)
此位由软件置 1 和清零
0：禁止寄存器重载中断
1：使能寄存器重载中断

位 2 **TERRIE**：传输错误中断使能 (Transfer Error Interrupt Enable)
此位由软件置 1 和清零
0：禁止传输错误中断
1：使能传输错误中断

位 1 **FUIE**：FIFO 下溢中断使能 (FIFO Underrun Interrupt Enable)
此位由软件置 1 和清零
0：禁止 FIFO 下溢中断
1：使能 FIFO 下溢中断

位 0 **LIE**：行中断使能 (Line Interrupt Enable)
此位由软件置 1 和清零
0：禁止行中断
1：使能行中断

32.7.9 LTDC 中断状态寄存器 (LTDC_ISR)

LTDC Interrupt Status Register

此寄存器返回中断状态标志

偏移地址：0x38

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RRIF	TERRIF	FUIF	LIF
												r	r	r	r

位 31:4 保留，必须保持复位值

位 3 **RRIF**：寄存器重载中断标志 (Register Reload Interrupt Flag)
0：未产生寄存器重载中断
1：发生垂直消隐重载时（以及到达有效区域后的第一行时）产生寄存器重载中断

位 2 **TERRIF**：传输错误中断标志 (Transfer Error interrupt flag)
0：未产生传输错误中断
1：出现总线错误时产生传输错误中断

位 1 **FUIF**：FIFO 下溢中断标志 (FIFO Underrun Interrupt flag)
0：未产生 FIFO 下溢中断
1：当其中一个层 FIFO 为空并从 FIFO 读取像素数据时，将产生 FIFO 下溢中断

位 0 **LIF**：行中断标志 (Line Interrupt flag)
0：未产生行中断
1：到达编程的行时产生行中断

32.7.10 LTDC 中断清零寄存器 (LTDC_ICR)

LTDC Interrupt Clear Register

偏移地址：0x3C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CRRIF	CTERRIF	CFUIF	CLIF
												w	w	w	w

位 31:4 保留，必须保持复位值

位 3 **CRRIF**：寄存器重载中断清零标志 (Clears Register Reload Interrupt Flag)
0：无影响
1：将 LTDC_ISR 寄存器中的 RRIF 标志清零

位 2 **CTERRIF**：传输错误中断清零标志 (Clears the Transfer Error Interrupt Flag)
0：无影响
1：将 LTDC_ISR 寄存器中的 TERRIF 标志清零

位 1 **CFUIF**：FIFO 下溢中断清零标志 (Clears the FIFO Underrun Interrupt flag)
0：无影响
1：将 LTDC_ISR 寄存器中的 FUDERRIF 标志清零

位 0 **CLIF**：行中断清零标志 (Clears the Line Interrupt Flag)
0：无影响
1：将 LTDC_ISR 寄存器中的 LIF 标志清零

32.7.11 LTDC 行中断位置配置寄存器 (LTDC_LIPCR)

LTDC Line Interrupt Position Configuration Register

此寄存器定义行中断的位置。要编程的行值取决于时序参数。请参见*图 235*。

偏移地址：0x40

复位值：0x0000 0000

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

位 31:11 保留，必须保持复位值

位 10:0 **LIPOS[10:0]**：行中断位置 (Line Interrupt Position)
这些位配置行中断位置

32.7.12 LTDC 当前位置状态寄存器 (LTDC_CPSR)

LTDC Current Position Status Register

偏移地址：0x44

复位值：0x0000 0000

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

位 31:16 **CXPOS[15:0]**：当前 X 位置 (Current X Position)
这些位返回当前 X 位置

位 15:0 **CYPOS[15:0]**：当前 Y 位置 (Current Y Position)
这些位返回当前 Y 位置

32.7.13 LTDC 当前显示状态寄存器 (LTDC_CDSR)

LTDC Current Display Status Register

此寄存器返回由 HSYNC、VSYNC 和水平/垂直 DE 信号控制的当前显示阶段的状态。

示例：如果当前显示阶段为垂直同步阶段，则 VSYNCS 位置 1（高电平有效）。示例：如果当前显示阶段为水平同步阶段，则 HSYNCS 位高电平有效。

偏移地址：0x48

复位值：0x0000 000F

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	HSYNCS	VSYNCS	HDES	VDES
												r	r	r	r

位 31:4 保留，必须保持复位值

位 3 **HSYNCS**：水平同步显示状态 (Horizontal Synchronization display Status)
0：低电平有效
1：高电平有效

位 2 **VSYNCS**：垂直同步显示状态 (Vertical Synchronization display Status)
0：低电平有效
1：高电平有效

位 1 **HDES**：水平数据使能显示状态 (Horizontal Data Enable display Status)
0：低电平有效
1：高电平有效

位 0 **VDES**：垂直数据使能显示状态 (Vertical Data Enable display Status)
0：低电平有效
1：高电平有效

注： *返回的状态不取决于* ***LTDC_GCR*** *寄存器中配置的极性，而是返回当前的有效显示阶段。*

32.7.14 LTDC 第 x 层控制寄存器 (LTDC_LxCR)（其中，x = 1..2）

LTDC Layerx Control Register

偏移地址：0x84 + 0x80 x (*第 x 层* - 1)，*x* = 1 或 2
复位值：0x0000 0000

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

位 31:5 保留，必须保持复位值

位 4 **CLUTEN**：查色表使能 (Color Look-Up Table Enable)
此位由软件置 1 和清零。
0：禁止查色表
1：使能查色表
CLUT 仅对 L8、AL44 和 AL88 像素格式有意义。请参见*第 1081 页的查色表 (CLUT)*。

位 3 保留，必须保持复位值

位 2 保留，必须保持复位值

位 1 **COLKEN**：色键使能 (Color Keying Enable)
此位由软件置 1 和清零。
0：禁止色键
1：使能色键

位 0 **LEN**：层使能 (Layer Enable)
此位由软件置 1 和清零。
0：禁止层
1：使能层

32.7.15 LTDC 第 x 层窗口水平位置配置寄存器 (LTDC_LxWHPCR)（其中 x=1..2）

LTDC Layerx Window Horizontal Position Configuration Register

此寄存器定义第 1 层或第 2 层窗口的水平位置（第一个和最后一个像素）。

一行的第一个可见像素是在 **LTDC_BPCR** 寄存器中编程的 ***AHBP[10:0] bits + 1*** 的值。

一行的最后一个可见像素是在 **LTDC_AWCR** 寄存器中编程的 ***AAW[10:0] bits*** 的值。

偏移地址：0x88 + 0x80 x (*第 x 层* - 1)，*x* = 1 或 2

复位值：0x0000 0000

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

位 31:28 保留，必须保持复位值

位 27:16 **WHSPPOS[11:0]**：窗口水平结束位置 (Window Horizontal Stop Position)
这些位配置层窗口的一行的最后一个可见像素。
WHSPPOS[11:0] 必须 >= ***AHBP[10:0] 位 + 1***（在 LTDC_BPCR 寄存器中编程）。

位 15:12 保留，必须保持复位值

位 11:0 **WHSTPOS[11:0]**：窗口水平起始位置 (Window Horizontal Start Position)
这些位配置层窗口的一行的第一个可见像素。
WHSTPOS[11:0] 必须 <= ***AAW[10:0] 位***（在 LTDC_AWCR 寄存器中编程）。

示例：

LTDC_BPCR 寄存器配置为 0x000E0005（AHBP[11:0] 为 0xE），LTDC_AWCR 寄存器配置为 0x028E01E5（AAW[11:0] 为 0x28E）。要配置大小为 630x460 的窗口的水平位置（有效数据区域中的水平起始偏移为 5 个像素）。

1. 层窗口的第一个像素：WHSTPOS[11:0] 应编程为 0x14 (0xE+1+0x5)
2. 层窗口的最后一个像素：WHSPPOS[11:0] 应编程为 0x28A

32.7.16 LTDC 第 x 层窗口垂直位置配置寄存器 (LTDC_LxWVPCR)（其中 x=1..2）

LTDC Layerx Window Vertical Position Configuration Register

此寄存器定义第 1 层或第 2 层窗口的垂直位置（第一行或最后一行）。

一个帧的第一个可见行是在 **LTDC_BPCR** 寄存器中编程的 ***AVBP[10:0] bits + 1*** 的值。

一个帧的最后一个可见行是在 **LTDC_AWCR** 寄存器中编程的 ***AAH[10:0] bits*** 的值。

偏移地址：0x8C + 0x80 x (*第 x 层*-1)，*x* = 1 或 2

复位值：0x0000 0000

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

位 31:27 保留，必须保持复位值

位 26:16 **WVSPPOS[10:0]**：窗口垂直结束位置 (Window Vertical Stop Position)
这些位配置层窗口的最后一个可见行。
WVSPPOS[10:0] 必须 >= ***AVBP[10:0] 位 + 1***（在 LTDC_BPCR 寄存器中编程）。

位 15:11 保留，必须保持复位值

位 10:0 **WVSTPOS[10:0]**：窗口垂直起始位置 (Window Vertical Start Position)
这些位配置层窗口的第一个可见行。
WVSTPOS[10:0] 必须 <= ***AAH[10:0] 位***（在 LTDC_AWCR 寄存器中编程）。

示例：

LTDC_BPCR 寄存器配置为 0x000E0005（AVBP[10:0] 为 0x5），LTDC_AWCR 寄存器配置为 0x028E01E5（AAH[10:0] 为 0x1E5）。要配置大小为 630x460 的窗口的垂直位置（有效数据区域中的垂直起始偏移为 8 行）：

1. 层窗口的第一行：WVSTPOS[10:0] 应编程为 0xE (0x5 + 1 + 0x8)
2. 层窗口的最后一行：WVSPPOS[10:0] 应编程为 0x1DA

32.7.17 LTDC 第 x 层色键配置寄存器 (LTDC_LxCKCR)（其中 x=1..2）

LTDC Layerx Color Keying Configuration Register

此寄存器定义色键使用的色键值 (RGB)。

偏移地址：0x90 + 0x80 x (*第 x 层* - 1)，*x* = 1 或 2

复位值：0x0000 0000

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

位 31:24 保留，必须保持复位值

位 23:16 **CKRED[7:0]**：色键红色值 (Color Key Red value)

位 15:8 **CKGREEN[7:0]**：色键绿色值 (Color Key Green value)

位 7:0 **CKBLUE[7:0]**：色键蓝色值 (Color Key Blue value)

32.7.18 LTDC 第 x 层像素格式配置寄存器 (LTDC_LxPFCR)（其中 x=1..2）

LTDC Layerx Pixel Format Configuration Register

此寄存器定义层的帧缓冲区中存储的数据所使用的像素格式。像素数据从帧缓冲区读取，随后转换为内部格式 8888 (ARGB)。

偏移地址：0x94 + 0x80 x (*第 x 层* - 1)，*x* = 1 或 2

复位值：0x0000 0000

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

位 31:3 保留，必须保持复位值

位 2:0 **PF[2:0]**：像素格式 (Pixel Format)
这些位配置像素格式
000：ARGB8888
001：RGB888
010：RGB565
011：ARGB1555
100：ARGB4444
101：L8（8 位 Luminance）
110：AL44（4 位 Alpha，4 位 Luminance）
111：AL88（8 位 Alpha，8 位 Luminance）

32.7.19 LTDC 第 x 层常数 Alpha 配置寄存器 (LTDC_LxCACR)（其中 x=1..2）

LTDC Layerx Constant Alpha Configuration Register

此寄存器定义在 alpha 混合中使用的常数 alpha 值（由硬件实现 255 等分）。
请参见 LTDC_LxBFCR 寄存器。

偏移地址：0x98 + 0x80 x (*第 x 层* - 1)，*x* = 1 或 2

复位值：(第 x 层 - 1) 0x0000 00FF

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

位 31:8 保留，必须保持复位值

位 7:0 **CONSTA[7:0]**：常数 Alpha (Constant Alpha)
这些位配置混合时使用的常数 Alpha。常数 Alpha 由硬件实现 255 等分。
示例：如果编程的常数 Alpha 为 0xFF，则常数 Alpha 值为 255/255=1。

32.7.20 LTDC 第 x 层默认颜色配置寄存器 (LTDC_LxDCCR)（其中 x=1..2）

LTDC Layerx Default Color Configuration Register

此寄存器定义采用 ARGB 格式的层的默认颜色。默认颜色在定义的层窗口外使用或在层禁止时使用。复位值 0x00000000 定义了透明黑色。

偏移地址：0x9C + 0x80 x (*第 x 层* - 1)，*x* = 1 或 2

复位值：0x0000 0000

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

位 31:24 **DCALPHA[7:0]**：默认 Alpha 值 (Default Color Alpha)
这些位配置默认 alpha 值

位 23:16 **DCRED[7:0]**：默认颜色红色 (Default Color Red)
这些位配置默认红色值

位 15:8 **DCGREEN[7:0]**：默认颜色绿色 (Default Color Green)
这些位配置默认绿色值

位 7:0 **DCBLUE[7:0]**：默认颜色蓝色 (Default Color Blue)
这些位配置默认蓝色值

32.7.21 LTDC 第 x 层混合系数配置寄存器 (LTDC_LxBFCR)（其中 x=1..2）

LTDC Layerx Blending Factors Configuration Register

此寄存器定义混合系数 F1 和 F2。

通用混合公式为：BC = BF1 x C + BF2 x Cs

- BC = 混合后的颜色
- BF1 = 混合系数 1
- C = 当前层颜色
- BF2 = 混合系数 2
- Cs = 底层混合后的颜色

偏移地址：0xA0 + 0x80 x (*第 x 层* - 1)，*x* = 1 或 2

复位值：0x0000 0607

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	BF1[2:0]			Res.	Res.	Res.	Res.	Res.	BF2[2:0]		
					rw	rw	rw						rw	rw	rw

位 31:11 保留，必须保持复位值

位 10:8 **BF1[2:0]**：混合系数 1 (Blending Factor 1)
这些位选择混合系数 F1
000：保留
001：保留
010：保留
011：保留
100：常数 Alpha
101：保留
110：像素 Alpha x 常数 Alpha
111：保留

位 7:3 保留，必须保持复位值

位 2:0 **BF2[2:0]**：混合系数 2 (Blending Factor 2)
这些位选择混合系数 F2
000：保留
001：保留
010：保留
011：保留
100：保留
101：1——常数 Alpha
110：保留
111：1——（像素 Alpha x 常数 Alpha）

注： *常数 Alpha 值是在寄存器中编程的值，由硬件实现 255 等分*

示例：仅使能第 1 层，BF1 配置为常数 Alpha

BF2 配置为 1—— *常数 Alpha*

常数 Alpha：在 LxCACR 寄存器中编程的常数 Alpha 为 240 (0xF0)。因此，常数 Alpha 值为 240/255 = 0.94

C：当前层颜色为 128

Cs：背景色为 48

第 1 层与背景色混合

BC = 常数 Alpha x C + (1 - 常数 Alpha) x Cs = *0.94 x 128 + (1- 0.94) x 48 = 123*

32.7.22 LTDC 第 x 层颜色帧缓冲区地址寄存器 (LTDC_LxCFBAR)（其中 x=1..2）

LTDC Layerx Color Frame Buffer Address Register

此寄存器定义颜色帧缓冲区的起始地址，该地址必须指向帧缓冲区中存储的层的左上角像素的像素数据地址。

偏移地址：0xAC + 0x80 x (*第 x 层* - 1)，*x* = 1 或 2

复位值：0x0000 0000

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

位 31:0 **CFBADD[31:0]**：颜色帧缓冲区起始地址 (Color Frame Buffer Start Address)
这些位定义颜色帧缓冲区的起始地址。

32.7.23 LTDC 第 x 层颜色帧缓冲区长度寄存器 (LTDC_LxCFBLR)（其中 x=1..2）

LTDC Layerx Color Frame Buffer Length Register

此寄存器定义颜色帧缓冲区的行长和行间距。

偏移地址：0xB0 + 0x80 x (*第 x 层* - 1)，*x* = 1 或 2

复位值：0x0000 0000

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

位 31:29 保留，必须保持复位值。

位 28:16 **CFBP[12:0]**：颜色帧缓冲区间距（以字节为单位）(Color Frame Buffer Pitch in bytes)
这些位定义从像素某行的起始处到下一行的起始处的增量（以字节为单位）。

位 15:13 保留，必须保持复位值

位 12:0 **CFBLL[12:0]**：颜色帧缓冲区行长 (Color Frame Buffer Line Length)
这些位定义一行像素的长度（以字节为单位）+ 7。
行长的计算方法为：有效宽度 x 每像素的字节数 + 7。

示例：

- 采用 RGB565（每像素 2 个字节）格式且宽度为 256 像素的帧缓冲区（每行总字节数为 256x2=512）需要向此寄存器写入值 0x02000207（其中，间距 = 行长）。
- 采用 RGB888（每像素 3 个字节）格式且宽度为 320 像素的帧缓冲区（每行总字节数为 320x3=960）需要向此寄存器写入值 0x03C003C7（其中，间距 = 行长）。

32.7.24 LTDC 第 x 层颜色帧缓冲区行数寄存器 (LTDC_LxCFBLNR)（其中 x=1..2）

LTDC Layerx ColorFrame Buffer Line Number Register

此寄存器定义颜色帧缓冲区中的行数。

偏移地址：0xB4 + 0x80 x (*第 x 层* - 1)，*x* = 1 或 2

复位值：0x0000 0000

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

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

位 31:11 保留，必须保持复位值

位 10:0 **CFBLNBR[10:0]**：帧缓冲区行数 (Frame Buffer Line Number)
这些位用来定义缓冲区中的行数，行数跟有效高度对应。

注： *行数和行长设置定义针对每层从每个帧中获取的数据量。如果配置为低于所需字节，则会产生 FIFO 下溢中断（如果使能）。*

另一方面，起始地址和间距设置定义存储器中每行的正确起始位置。

32.7.25 LTDC 第 x 层 CLUT 写寄存器 (LTDC_LxCLUTWR)（其中 x=1..2）

LTDC Layerx CLUT Write Register

此寄存器定义 CLUT 地址和 RGB 值。

偏移地址：0xC4 + 0x80 x (*第 x 层* - 1)，*x* = 1 或 2

复位值：0x0000 0000

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

位 31:24 **CLUTADD[7:0]**：CLUT 地址 (CLUT Address)
这些位配置每个 RGB 值的 CLUT 地址（CLUT 内的颜色位置）

位 23:16 **RED[7:0]**：红色值 (Red value)
这些位配置红色值

位 15:8 **GREEN[7:0]**：绿色值 (Green value)
这些位配置绿色值

位 7:0 **BLUE[7:0]**：蓝色值 (Blue value)
这些位配置蓝色值

注： *CLUT 写寄存器只应在消隐周期内或在层禁止时配置。CLUT 可通过* ***LTDC_LxCR*** *寄存器使能和禁止。*

CLUT 仅对 L8、AL44 和 AL88 像素格式有意义。

32.7.26 LTDC 寄存器映射

下表对 LTDC 寄存器进行了汇总。有关 LTDC 寄存器的基本地址，请参见寄存器边界地址表。

表 255. LTDC 寄存器映射和复位值

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x0008	LTDC_SSCR	Res.	Res.	Res.	Res.	HSW[11:0]												Res.	Res.	Res.	Res.	Res.	VSH[10:0]										
	Reset value					0	0	0	0	0	0	0	0	0	0	0	0						0	0	0	0	0	0	0	0	0	0	0
0x000C	LTDC_BPCR	Res.	Res.	Res.	Res.	AHBP[11:0]												Res.	Res.	Res.	Res.	Res.	AVBP[10:0]										
	Reset value					0	0	0	0	0	0	0	0	0	0	0	0						0	0	0	0	0	0	0	0	0	0	0
0x0010	LTDC_AWCR	Res.	Res.	Res.	Res.	AAV[11:0]												Res.	Res.	Res.	Res.	Res.	AAH[10:0]										
	Reset value					0	0	0	0	0	0	0	0	0	0	0	0						0	0	0	0	0	0	0	0	0	0	0
0x0014	LTDC_TWCR	Res.	Res.	Res.	Res.	TOTALW[11:0]												Res.	Res.	Res.	Res.	Res.	TOTALH[10:0]										
	Reset value					0	0	0	0	0	0	0	0	0	0	0	0						0	0	0	0	0	0	0	0	0	0	0
0x0018	LTDC_GCR	HSPOL	VSPOL	DEPOL	PCPOL	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DEN	Res.	Res.	DRW[2:0]		Res.	DGW[2:0]			Res.	DBW[2:0]			Res.	Res.	Res.	LTDCEN
	Reset value	0	0	0	0												0		0	1	0		0	1	0		0	1	0				0
0x0024	LTDC_SRCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	VBR	IMR
	Reset value																															0	0
0x002C	LTDC_BCCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BCRED[7:0]								BCGREEN[7:0]					BCBLUE[7:0]										
	Reset value									0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0034	LTDC_IER	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RRIE	TERRIE	FUIE	LIE
	Reset value																													0	0	0	0
0x0038	LTDC_ISR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RRIF	TERRIF	FUIF	LIF
	Reset value																													0	0	0	0

表 255. LTDC 寄存器映射和复位值（续）

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x003C	LTDC_ICR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CRRIF	CTERRIF	CFUIF	CLIF
	Reset value																													0	0	0	0
0x0040	LTDC_LIPCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BCRED[10:0]										
	Reset value																						0	0	0	0	0	0	0	0	0	0	0
0x0044	LTDC_CPSR	CXPOS[15:0]																CYPOS[15:0]															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0048	LTDC_CDSR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	HSYNCS	VSYNCS	HDES	VDES
	Reset value																													1	1	1	1
0x0084	LTDC_L1CR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CLUTEN	Res.	Res.	COLKEN	LEN
	Reset value																												0			0	0
0x0088	LTDC_L1WHPCR	Res.	Res.	Res.	Res.	WHSPPOS[11:0]												Res.	Res.	Res.	Res.	WHSTPOS[11:0]											
	Reset value					0	0	0	0	0	0	0	0	0	0	0	0					0	0	0	0	0	0	0	0	0	0	0	0
0x008C	LTDC_L1WVPCR	Res.	Res.	Res.	Res.	Res.	WVSPPOS[10:0]											Res.	Res.	Res.	Res.	Res.	WVSTPOS[10:0]										
	Reset value						0	0	0	0	0	0	0	0	0	0	0						0	0	0	0	0	0	0	0	0	0	0
0x0090	LTDC_L1CKCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CKRED[7:0]								CKGREEN[7:0]								CKBLUE[7:0]							
	Reset value									0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0094	LTDC_L1PFCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PF[2:0]		
	Reset value																														0	0	0
0x0098	LTDC_L1CACR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CONSTA[7:0]							
	Reset value																									1	1	1	1	1	1	1	1
0x009C	LTDC_L1DCCR	DCALPHA[7:0]								DCRED[7:0]								DCGREEN[7:0]								DCBLUE[7:0]							
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x00A0	LTDC_L1BFCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BF1[2:0]			Res.	Res.	Res.	Res.	Res.	BF2[2:0]		
	Reset value																						1	1	0						1	1	1
0x00AC	LTDC_L1CFBAR	CFBADD[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x00B0	LTDC_L1CFBLR	Res.	Res.	Res.	CFBP[17:0]													Res.	Res.	Res.	CFBLL[12:0]												
	Reset value				0	0	0	0	0	0	0	0	0	0	0	0	0				0	0	0	0	0	0	0	0	0	0	0	0	0

表 255. LTDC 寄存器映射和复位值（续）

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00B4	LTDC_L1CFBLNR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CFBLNBR[10:0]										
	Reset value																						0	0	0	0	0	0	0	0	0	0	0
0x00C4	LTDC_L1CLUTWR	CLUTADD[7:0]								RED[7:0]								GREEN[7:0]								BLUE[7:0]							
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0104	LTDC_L2CR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CLUTEN	Res.	Res.	COLKEN	LEN
	Reset value																												0			0	0
0x0108	LTDC_L2WHPCR	Res.	Res.	Res.	Res.	WHSPPOS[11:0]												Res.	Res.	Res.	Res.	WHSTPOS[11:0]											
	Reset value					0	0	0	0	0	0	0	0	0	0	0	0					0	0	0	0	0	0	0	0	0	0	0	0
0x010C	LTDC_L2WVPCR	Res.	Res.	Res.	Res.	Res.	WVSPPOS[10:0]											Res.	Res.	Res.	Res.	Res.	WVSTPOS[10:0]										
	Reset value						0	0	0	0	0	0	0	0	0	0	0						0	0	0	0	0	0	0	0	0	0	0
0x0110	LTDC_L2CKCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CKRED[7:0]								CKGREEN[7:0]								CKBLUE[7:0]							
	Reset value									0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0114	LTDC_L2PFCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PF[2:0]		
	Reset value																														0	0	0
0x0118	LTDC_L2CACR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CONSTA[7:0]							
	Reset value																									1	1	1	1	1	1	1	1
0x011C	LTDC_L2DCCR	DCALPHA[7:0]								DCRED[7:0]								DCGREEN[7:0]								DCBLUE[7:0]							
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0120	LTDC_L2BFCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BF1[2:0]			Res.	Res.	Res.	Res.	Res.	BF2[2:0]		
	Reset value																						1	1	0						1	1	1
0x012C	LTDC_L2CFBAR	CFBADD[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0130	LTDC_L2CFBLR	Res.	Res.	Res.	CFBP[12:0]													Res.	Res.	Res.	CFBLL[12:0]												
	Reset value				0	0	0	0	0	0	0	0	0	0	0	0	0				0	0	0	0	0	0	0	0	0	0	0	0	0
0x0134	LTDC_L2CFBLNR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CFBLNBR[10:0]										
	Reset value																						0	0	0	0	0	0	0	0	0	0	0
0x0144	LTDC_L2CLUTWR	CLUTADD[7:0]								RED[7:0]								GREEN[7:0]								BLUE[7:0]							
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

有关寄存器边界地址的信息，请参见*第 100 页的第 2.2.2 节*。

33 JPEG 编解码器 (JPEG)

33.1 简介

硬件 8 位 JPEG 编解码器对未压缩的图像数据流进行编码或对 JPEG 压缩模式的图像数据流进行解码， 并可全面管理 JPEG 文件头。

33.2 JPEG 编解码器主要特性

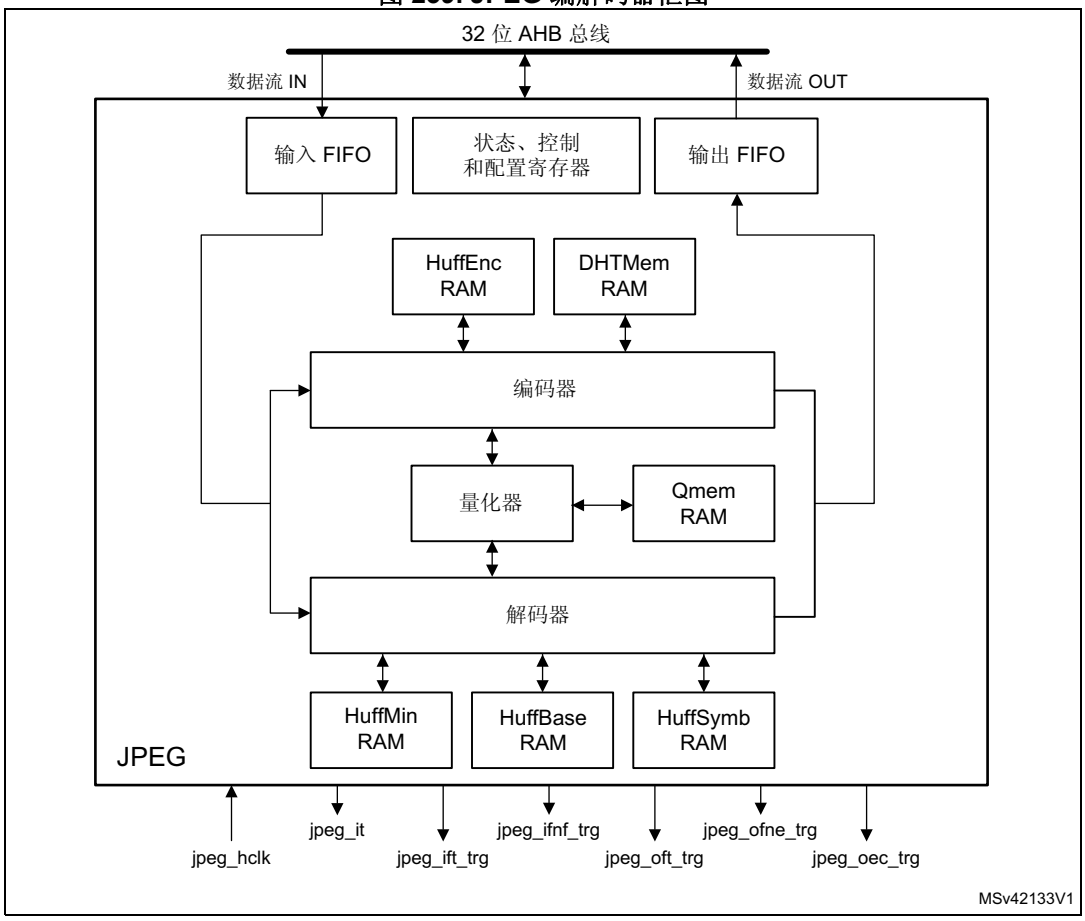
- 高速全同步操作
- 可配置为编码器或解码器
- 对每个像素数据进行编解码只需一个时钟周期
- 支持 RGB、YCbCr、YCMK 和 BW（灰度）图像色彩空间
- 编解码时每图像分量 8 位深度
- 具有使能/禁止功能的 JPEG 文件头生成器/解析器
- 四个可编程量化表
- 在一个时钟周期内完成霍夫曼编解码
- 完全可编程霍夫曼表（两个 AC 表和两个 DC 表）
- 完全可编程最小编码单元 (MCU)
- 并发输入和输出数据流接口

33.3 JPEG 编解码器功能说明

33.3.1 概述

JPEG 编解码器的框图如*图 239：JPEG 编解码器框图*所示。

图 239. JPEG 编解码器框图



33.3.2 JPEG 内部信号

表 256 列出了 JPEG 内部信号。

表 256. JPEG 内部信号

信号名称	信号类型	说明
jpeg_hclk	数字输入	JPEG 内核和寄存器接口时钟
jpeg_it	数字输出	JPEG 全局中断
jpeg_ift_trg	数字输出	用于 MDMA 的 JPEG 输入 FIFO 阈值
jpeg_ifnf_trg	数字输出	用于 MDMA 的 JPEG 输入 FIFO 未满
jpeg_oft_trg	数字输出	用于 MDMA 的 JPEG 输出 FIFO 阈值
jpeg_ofne_trg	数字输出	用于 MDMA 的 JPEG 输出 FIFO 非空
jpeg_oec_trg	数字输出	用于 MDMA 的 JPEG 转换结束

33.3.3 JPEG 解码程序

JPEG 编解码器可以解码在 *ISO/IEC 10918-1* 规范中定义的 JPEG 码流。

它可以有选择性地解析 JPEG 文件头，并相应地更新 JPEG 编解码器寄存器、量化表和霍夫曼表。

通过将 JPEG_CONFR1 寄存器的 DE 位（解码使能）置 1，将 JPEG 编解码器配置为解码模式。

通过将 JPEG_CONFR0 寄存器的 START 位置 1，启动 JPEG 解码。

JPEG 编解码器将为其输入 FIFO 请求数据，可以生成：

- MDMA 触发
- 中断

输入 FIFO 的中断或 MDMA 触发生成

输入 FIFO 使用中断或 MDMA 触发通过两个基于 FIFO 状态的标志来进行管理：

- 输入 FIFO 未满标志：可写入 32 位的值。
- 输入 FIFO 阈值标志：可写入 8 字（32 字节）。

中断或 MDMA 触发生成与 JPEG_CONFR0 寄存器的 START 位无关。输入 FIFO 标志的生成与 JPEG 编解码器内核的状态无关。

如果 FIFO 已满，则会忽略写操作。

解码过程结束后，额外的字节可保留在输入 FIFO 中，且/或者中断请求/MDMA 触发可处于挂起状态。可通过将 JPEG_CR 寄存器的 IFF 位（输入 FIFO 清空）置 1 来清空 FIFO。

清空 FIFO 之前：

- 必须禁止输入 FIFO 的中断，以免在清空 FIFO 时产生不必要的中断请求。
- MDMA 通道必须停止，以免产生不必要的 MDMA 触发。

如果解码过程结束时 FIFO 未清空，剩余的数据将在下一次 JPEG 解码时进行处理。

头文件解析

通过将 JPEG_CONFR1 寄存器的 HDR 位置 1，可以激活文件头解析。

JPEG 文件头解析器支持与 *ISO/IEC 10918-1 附件 B* 中指示的 JPEG 基线算法相关的所有标记。

解析支持的标记时，JPEG 头文件解析器会提取所需参数并将其存储在影子寄存器中。解析结束时，会更新 JPEG 编解码器寄存器。

如果找到 DQT 标记段，与该标记段相关联的量化数据会写入到量化表存储器中。

如果找到 DHT 标记段，与该标记段相关联的霍夫曼表数据会转换为三种不同的表格式（HuffMin、HuffBase 和 HuffSymb）并存储在其各自的存储器中。

解析操作完成后，JPEG_SR 寄存器的 HPDF（文件头解析完成标志）位会置 1。如果 JPEG_CR 寄存器的 EHPIE（文件头解析结束中断使能）位置 1，则会生成中断。

JPEG 解码

只要完成解析 JPEG 文件头，或者对 JPEG 编解码器寄存器和存储器进行了正确的编程，就会对传入数据流进行解码，并将生成的 MCU 发送到输出 FIFO。

如果连续对两张图像进行解码，必须在第二张图像的文件头处理完成后再次将 JPEG_CONFR0 寄存器的 START 位置 1（即使已经置 1）。

输出 FIFO 的中断或 MDMA 触发生成

输出 FIFO 使用中断或 MDMA 触发通过两个基于 FIFO 状态的标志来进行管理：

- 输出 FIFO 非空标志：可读取 32 位值。
- 输出 FIFO 阈值标志：可读取 8 字（32 字节）。

如果输出 FIFO 为空，读操作会返回 0。

如果通过将 JPEG_CONFR0 寄存器的 START 位复位的方式中止 JPEG 编解码器操作，则输出 FIFO 可以被清空。如果需要清空 FIFO，则应通过软件将 JPEG_CR 寄存器的 FF 位（FIFO 清空）置 1 来实现。

清空 FIFO 之前：

- 必须禁止输出 FIFO 的中断，以免在清空 FIFO 时产生不必要的中断请求。
- MDMA 通道必须停止，以免产生不必要的 MDMA 触发。

必须在处理结束时清空输出 FIFO，然后才能更改 JPEG 配置。

33.3.4 JPEG 编码程序

JPEG 编解码器可以对 *ISO/IEC 10918-1* 规范定义的 JPEG 码流进行编码。

JPEG 编解码器可以有选择地生成 JPEG 文件头。

通过将 JPEG_CONFR1 寄存器的 DE 位（解码使能）复位，将 JPEG 编解码器配置为编码模式。

必须将用于对 JPEG 进行编码的配置加载到 JPEG 编码器中：

- JPEG 编解码器配置寄存器
- 量化表
- 霍夫曼表

通过将 JPEG_CONFR0 寄存器的 START 位置 1，启动 JPEG 编解码器。

一旦启动，JPEG 编解码器将为其输入 FIFO 请求数据，可以生成：

- MDMA 触发
- 中断

输入 FIFO 的中断或 MDMA 触发生成

输入 FIFO 使用中断或 MDMA 触发通过两个基于 FIFO 状态的标志来进行管理：

- 输入 FIFO 未满标志：可写入 32 位的值。
- 输入 FIFO 阈值标志：可写入 8 字（32 字节）。

中断或 MDMA 触发生成与 JPEG_CONFR0 寄存器的 START 位无关。输入 FIFO 标志的生成与 JPEG 编解码器内核的状态无关。

如果 FIFO 已满，则会忽略写操作。

编码过程结束后，额外的字节可保留在输入 FIFO 中，且/或者中断请求/MDMA 触发可处于挂起状态。可通过将 JPEG_CR 寄存器的 IFF 位（输入 FIFO 清空）置 1 来清空 FIFO。

清空 FIFO 之前：

- 必须禁止输入 FIFO 的中断，以免在清空 FIFO 时产生不必要的中断请求。
- MDMA 通道必须停止，以免产生不必要的 MDMA 触发。

如果编码过程结束时 FIFO 未清空，剩余的数据将在下一次 JPEG 编码时进行处理。

JPEG 编码

只要生成了 JPEG 文件头，就会对传入的 MCU 进行编码，并将生成的数据流发送到输出 FIFO。

输出 FIFO 的中断或 MDMA 触发生成

输出 FIFO 使用中断或 MDMA 触发通过两个基于 FIFO 状态的标志来进行管理：

- 输出 FIFO 非空标志：可读取 32 位值。
- 输出 FIFO 阈值标志：可读取 8 字（32 字节）。

如果输出 FIFO 为空，读操作会返回 0。

如果通过将 JPEG_CONFR0 寄存器的 START 位复位的方式中止 JPEG 编解码器操作，则输出 FIFO 可以被清空。可通过将 JPEG_CR 寄存器的 FF 位（FIFO 清空）置 1 来清空 FIFO。

清空 FIFO 之前：

- 必须禁止输出 FIFO 的中断，以免在清空 FIFO 时产生不必要的中断请求。
- MDMA 通道必须停止，以免产生不必要的 MDMA 触发。

必须在处理结束时清空输出 FIFO，然后才能更改 JPEG 配置。

只有将输出 FIFO 清空之后，才可将 JPEG_SR 寄存器的 EOCF 位（转换结束标志）清零。

只有将 JPEG_SR 寄存器的 EOCF 位清零后，才允许将 JPEG_CONFR1 寄存器的 HDR 位（文件头处理）或 JPEG_CR 寄存器的 JCEN 位（JPEG 编解码器使能）清零。

33.4 JPEG 编解码器中断

发生如下事件时可生成中断：

- 已达到输入 FIFO 阈值
- 输入 FIFO 未满
- 已达到输出 FIFO 阈值
- 输出 FIFO 非空
- 转换结束
- 文件头解析完成

可以使用单独的中断使能位以提高灵活性。

表 257. JPEG 编解码器中断请求

中断事件	事件标志	使能控制位
已达到输入 FIFO 阈值	IFTF	IFTIE
输入 FIFO 未满	IFNFF	IFNFIE
已达到输出 FIFO 阈值	OFTF	OFTIE
输出 FIFO 非空	OFNEF	OFNEIE
转换结束	EOCF	EOCIE
头文件解析完成	HPDF	HPDIE

33.5 JPEG 编解码器寄存器

33.5.1 JPEG 编解码器控制寄存器 0 (JPEG_CONFR0)

JPEG codec control register 0

偏移地址：0x0000

复位值：0x0000 0000

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

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

位 31:1 保留

位 0 **START**：启动 (Start)
此位可启动或停止编码或解码过程。
0：停止/中止
1：启动
读操作将始终返回 0。

33.5.2 JPEG 编解码器配置寄存器 1 (JPEG_CONFR1)

JPEG codec configuration register 1

偏移地址：0x0004

复位值：0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	HDR	NS[1:0]		COLSPACE[1:0]		DE	Res.	NF[1:0]	

位 31:16 **YSIZE[15:0]**： Y 大小 (Y Size)
该字段定义源图像中的行数。

位 15:9 保留

位 8 **HDR**：文件头处理 (Header Processing)
该位使能文件头处理（生成/解析）。
0：禁止
1：使能

位 7:6 **NS[1:0]**：扫描分量数 (Number of components for scan)
该字段定义用于扫描头文件标记分段的分量数减去 1。

位 5:4 **COLORSPACE[1:0]**：色彩空间 (Color space)
该字段定义用于插入到输出数据流中的量化表数减去 1。
00：灰度（1 个量化表）
01：YUV（2 个量化表）
10：RGB（3 个量化表）
11：CMYK（4 个量化表）

位 3 **DE**：编解码器用作编码器或解码器 (Codec operation as coder or decoder)
此位选择编码或解码过程
0：编码
1：解码

位 2 保留

位 1:0 **NF[1:0]**：色彩分量数 (Number of color components)
该字段定义色彩分量数减去 1。
00：灰度（1 个色彩分量）
01：-（2 个色彩分量）
10：YUV 或 RGB（3 个色彩分量）
11：CMYK（4 个色彩分量）

33.5.3 JPEG 编解码器配置寄存器 2 (JPEG_CONFR2)

JPEG codec configuration register 2

偏移地址：0x0008

复位值：0x0000 0000

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

位 31:26 保留

位 25:0 **NMCU[25:0]**：MCU 数 (Number of MCUs)
对于编码过程：该字段定义要编码的 MCU 数减去 1。
对于解码过程：该字段指示要解码的完整 MCU 单元数减去 1（该字段会在 JPEG 头文件解析后更新）。如果解码图像的 X 或 Y 大小不是 8 或 16 的倍数（取决于子采样过程），必须将生成的不完整 MCU 或空 MCU 与该值相加，以得到生成的 MCU 总数。

33.5.4 JPEG 编解码器配置寄存器 3 (JPEG_CONFR3)

JPEG codec configuration register 3

偏移地址：0x000C

复位值：0x0000 0000

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

位 31:16 **XSIZE[15:0]**：X 大小 (X size)
该字段定义每行的像素数。

位 15:0 保留

33.5.5 JPEG 编解码器配置寄存器 4-7 (JPEG_CONFR4-7)

JPEG codec configuration register 4-7

偏移地址：0x0010 + 0x4 * i，其中“i”为 0 到 3 的图像分量

复位值：0x0000 0000

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

位 31:16 保留

位 15:12 **HSF[3:0]**：水平采样因数 (Horizontal sampling factor)
分量 i 的水平采样因数。

位 11:8 **VSF[3:0]**：垂直采样因数 (Vertical sampling factor)
分量 i 的垂直采样因数。

位 7:4 **NB[3:0]**：块数 (Number of blocks)
属于 MCU 中特定颜色的数据单元数减去 1。

位 3:2 **QT[1:0]**：量化表 (Quantization table)
选择分量 i 所使用的量化表。
00：量化表 0
01：量化表 1
10：量化表 2
11：量化表 3

位 1 **HA**：霍夫曼 AC (Huffman AC)
选择用于对 AC 系数进行编码的霍夫曼表。
0：未选
1：已选

位 0 **HD**：霍夫曼 DC (Huffman DC)
选择用于对 DC 系数进行编码的霍夫曼表。
0：未选
1：已选

33.5.6 JPEG 控制寄存器 (JPEG_CR)

JPEG control register

偏移地址：0x0030

复位值：0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	OFF	IFF	Res.	Res.	Res.	Res.	Res.	Res.	HPDIE	EOCIE	OFNEIE	OFTIE	IFNFIE	IFTIE	JCEN
	r0	r0							rw	rw	rw	rw	rw	rw	rw

位 31:15 保留

位 14 **OFF**：输出 FIFO 清空 (Output FIFO flush)
此位可清空输出 FIFO。
0：无影响
1：清空输出 FIFO
读操作将始终返回 0。

位 13 **IFF**：输入 FIFO 清空 (Input FIFO flush)
此位可清空输入 FIFO。
0：无影响
1：清空输入 FIFO
读操作将始终返回 0。

位 12: 7 保留

位 6 **HPDIE**：文件头解析完成中断使能 (Header parsing done interrupt enable)
该位可使能文件头解析操作完成的中断生成。
0：禁止
1：使能

位 5 **EOCIE**：转换结束中断使能 (End of conversion interrupt enable)
该位可使能转换结束的中断生成。
0：禁止
1：使能

位 4 **OFNEIE**：输出 FIFO 非空中断使能 (Output FIFO not empty interrupt enable)
该位可在输出 FIFO 不为空时使能中断生成。
0：禁止
1：使能

位 3 **OFTIE**：输出 FIFO 阈值中断使能 (Output FIFO threshold interrupt enable)
该位可在输出 FIFO 达到阈值时使能中断生成。
0：禁止
1：使能

位 2 **IFNFIE**：输入 FIFO 未满中断使能 (Input FIFO not full interrupt enable)
该位可在输入 FIFO 不为空时使能中断生成。
0：禁止
1：使能

位 1 **IFTIE**：输入 FIFO 阈值中断使能 (Input FIFO threshold interrupt enable)
该位可在输入 FIFO 达到阈值时使能中断生成。
0：禁止
1：使能

位 0 **JCEN**：JPEG 内核使能 (JPEG core enable)
该位可使能 JPEG 编解码器内核。
0：禁止（内部寄存器被复位）。
1：使能（内部寄存器可访问）。

33.5.7 JPEG 状态寄存器 (JPEG_SR)

JPEG status register

偏移地址：0x0034

复位值：0x0000 0006

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	COF	HPDF	EOCF	OFNEF	OFTF	IFNFF	IFTF	Res.
								ro	ro	ro	ro	ro	ro	ro	

位 31:8 保留

位 7 **COF**：编解码器操作标志 (Codec operation flag)
该位可指示编码/解码操作是否正在进行中。
0：未在进行中
1：正在进行中

位 6 **HPDF**：文件头解析完成标志 (Header parsing done flag)
在解码模式下，该位可指示文件头解析和内部寄存器更新是否完成。
0：未完成
1：已完成

位 5 **EOCF**：转换结束标志 (End of conversion flag)
该位可指示编码/解码过程以及向输出 FIFO 传输数据的过程是否完成。
0：未完成
1：已完成

位 4 **OFNEF**：输出 FIFO 非空标志 (Output FIFO not empty flag)
该位可指示输出 FIFO 中是否有数据可用。
0：为空（无数据可用）
1：非空（有数据可用）

位 3 **OFTF**：输出 FIFO 阈值标志 (Output FIFO threshold flag)
该位可指示输出 FIFO 中的数据量是否达到或超出阈值。
0：低于阈值
1：达到或高于阈值

位 2 **IFNFF**：输入 FIFO 未满标志 (Input FIFO not full flag)
该位可指示输入 FIFO 是否处于未满状态（可写入数据）。
0：已满
1：未满

位 1 **IFTF**：输入 FIFO 阈值标志 (Input FIFO threshold flag)
该位可指示输入 FIFO 中的数据量是否低于阈值。
0：达到或高于阈值
1：低于阈值

位 0 保留

33.5.8 JPEG 清零标志寄存器 (JPEG_CFR)

JPEG clear flag register

偏移地址：0x0038

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CHPDF	CEOCF	Res.	Res.	Res.	Res.	Res.
									w1c	w1c	Res.	Res.	Res.	Res.	Res.

位 31:7 保留

位 6 **CHPDF**：文件头解析完成标志清零 (Clear header parsing done flag)
写入 1 会将 JPEG_SR 寄存器的 HPDF 位清零。
0：无影响
1：清零

位 5 **CEOCF**：转换结束标志清零 (Clear end of conversion flag)
写入 1 会将 JPEG_SR 寄存器的 ECF 位清零。
0：无影响
1：清零

位 4:0 保留

33.5.9 JPEG 数据输入寄存器 (JPEG_DIR)

JPEG data input register

偏移地址：0x0040

复位值：0x0000 0000

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

位 31:0 **DATAIN[31:0]**：数据输入 FIFO (Data input FIFO)
输入 FIFO 数据寄存器。

33.5.10 JPEG 数据输出寄存器 (JPEG_DOR)

JPEG data output register

偏移地址：0x0044

复位值：0x0000 0000

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

位 31:0 **DATAOUT[31:0]**：数据输出 FIFO (Data output FIFO)
输出 FIFO 数据寄存器。

33.5.11 JPEG 编解码器寄存器映射

下表对 JPEG 编解码器寄存器进行了汇总。有关 JPEG 编解码器寄存器的基址，请参见寄存器边界地址表。

表 258. JPEG 编解码器寄存器映射和复位值

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x0000	**JPEG_CONFR0**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	START
	Reset value																																0
0x0004	**JPEG_CONFR1**	YSIZE[15:0]																Res.	Res.	Res.	Res.	Res.	Res.	Res.	HDR	NS[1:0]		COLSPACE[1:0]		DE	Res.	NF[1:0]	
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0								0	0	0	0	0	0		0	0
0x0008	**JPEG_CONFR2**	Res.	Res.	Res.	Res.	Res.	Res.	NMCU[25:0]																									
	Reset value							0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x000C	**JPEG_CONFR3**	XSIZE[15:0]																Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0																
0x0010	**JPEG_CONFR4**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	HSF[3:0]				VSF[3:0]				NB[3:0]				QT[1:0]		HA	HD
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0014	**JPEG_CONFR5**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	HSF[3:0]				VSF[3:0]				NB[3:0]				QT[1:0]		HA	HD
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0018	**JPEG_CONFR6**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	HSF[3:0]				VSF[3:0]				NB[3:0]				QT[1:0]		HA	HD
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x001C	**JPEG_CONFR7**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	HSF[3:0]				VSF[3:0]				NB[3:0]				QT[1:0]		HA	HD
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0020-0x002C	保留	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
0x0030	**JPEG_CR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OFF	IFF	Res.	Res.	Res.	Res.	Res.	Res.	Res.	EOCIE	OFNEIE	OFTIE	IFNFIE	IFTIE	JCEN
	Reset value																		0	0								0	0	0	0	0	0
0x0034	**JPEG_SR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	COF	HPDF	EOCF	OFNEF	OFTF	IFNFF	IFTF	Res.
	Reset value																									0	0	0	0	0	1	1	
0x0038	**JPEG_CFR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CHPDF	CEOCF	Res.	Res.	Res.	Res.	Res.
	Reset value																										0	0					
0x0040	**JPEG_DIR**	DATAIN[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

表 258. JPEG 编解码器寄存器映射和复位值（续）

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x0044	**JPEG_DOR**	DATAOUT[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0050-0x014C	**QMEM**	QMem RAM																															
	Reset value	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
0x0150-0x018C	**HUFFMIN**	HuffMin RAM																															
	Reset value	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
0x0190-0x020C	**HUFFBASE**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	HuffBase RAM									Res.	Res.	Res.	Res.	Res.	Res.	Res.	HuffBase RAM								
	Reset value								X	X	X	X	X	X	X	X	X								X	X	X	X	X	X	X	X	X
0x0210-0x035C	**HUFFSYMB**	HuffSymb RAM																															
	Reset value	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
0x0360-0x04FC	**DHTMEM**	DHTMem RAM																															
	Reset value	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
0x0500-0x07FC	**HUFFENC**	Res.	Res.	Res.	Res.	HuffEnc RAM												Res.	Res.	Res.	Res.	HuffEnc RAM											
	Reset value					X	X	X	X	X	X	X	X	X	X	X	X					X	X	X	X	X	X	X	X	X	X	X	X

有关寄存器边界地址的信息，请参见*第 100 页的第 2.2.2 节*。

34 真随机数发生器 (RNG)

34.1 前言

RNG 是一个真随机数发生器，可向应用程序提供作为 32 位采样的全熵输出，它由一个实时熵源（模拟）和一个内部调节组件构成。

RNG 可作为一个实时熵源用来构建符合 NIST 要求的确定性随机位发生器 (DRBG)。

RNG 真随机数发生器已按照德国 AIS-31 标准进行了验证。

34.2 RNG主要特性

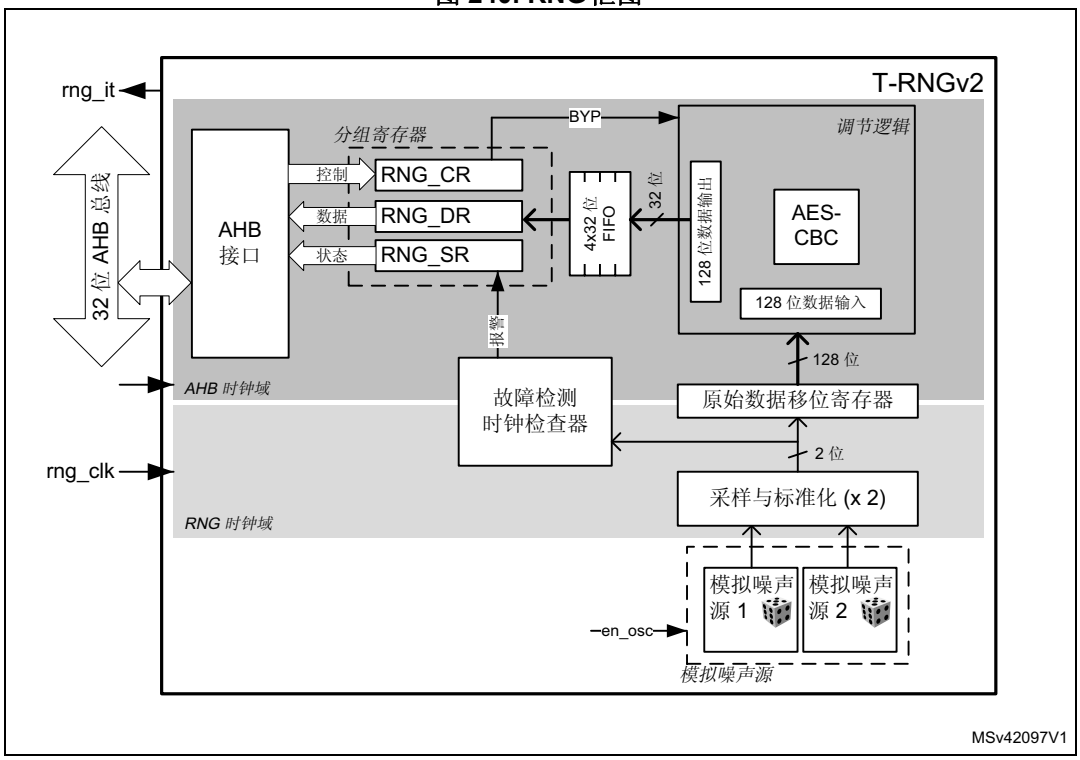
- RNG 提供的 32 位真随机数由模拟熵源生成并使用块密码 AES-CBC 进行调节。
- RNG 按照 AIS-31 预定义类 PTG.2 评估方法进行验证，该评估方法属于德国通用标准 (CC) 方案的组成部分。
- RNG 每 4x54 个 AHB 时钟周期会生成四个 32 位随机采样。
- 支持内置连续基本运行状态测试及相关错误管理。
  - 包括过低采样时钟检测和重复计数测试。
- 可被禁止以降低功耗。
- 具有 AMBA AHB 从外设，只能通过 32 位字进行单次访问（否则会针对写访问产生 AHB 总线错误），请务必注意！任何不等于 32 位的写操作都可能破坏寄存器内容。

34.3 RNG 功能说明

34.3.1 RNG框图

图 240 显示了 RNG 框图。

图 240. RNG框图



34.3.2 RNG 内部信号

表 259 中列出了 RNG 级而非 STM32 产品级（焊盘上）所提供的内部信号，对这些信号有所了解会很有用。

表 259. RNG 内部输入/输出信号

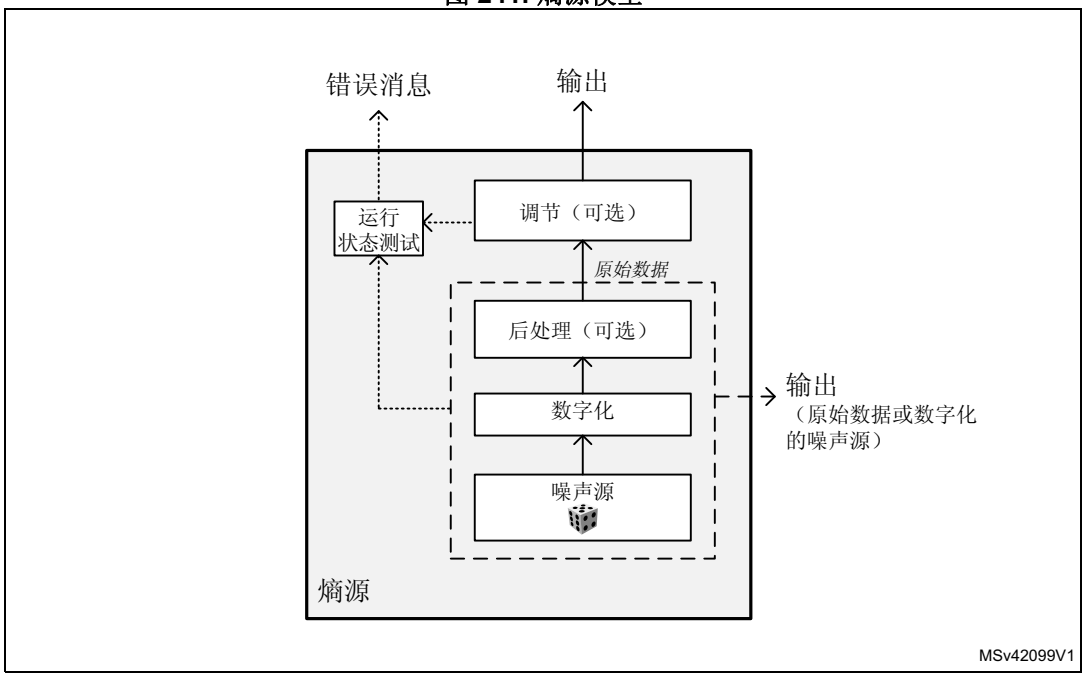
信号名称	信号类型	说明
rng_it	数字输出	RNG 全局中断请求
rng_hclk2	数字输入	AHB2 时钟
RNG_CLK	数字输入	RNG 专用时钟，与 rng_hclk2 异步

34.3.3 随机数生成

真随机数发生器 (RNG) 按确定的间隔通过其 AHB 接口提供真随机数。RNG 可实现*图 241* 所示的熵源模型，并为应用程序提供三个主要功能：

- 采集熵源的位串输出
- 获取噪声源采样，用于验证目的
- 采集连续运行状态测试得到的错误消息

图 241. 熵源模型



RNG 的主要组件包括：

- 物理随机源（模拟噪声源）
- 模拟噪声源的数字化处理阶段
- 模拟噪声源的加密处理调节阶段
- 输出缓冲区，用于输出熵源（已缓冲）和噪声源采样（已缓冲）
- 运行状态监视模块，用于对整个熵源执行测试

下文对所有组件进行了详细介绍。

噪声源：

噪声源组件包含不确定性的、能够提供熵值的活动，确保了由熵源生成的输出位串的不确定性。它包括：

- 两个模拟噪声源，每个噪声源由三个自由运行的环形振荡器输出异或 (XOR) 而成。可以禁止这些模拟振荡器以实现节能，如*第 34.4 节：RNG 低功耗使用*所述。
- 这些输出的采样级由专用时钟输入 (**rng_clk**) 提供时钟信号，输出 2 位原始数据。

由于噪声源输出信号与专用 **rng_clk** 输入不同步，因此可能发生亚稳态问题，噪声源采样级可解决这一问题。另外，该噪声源采样与 AHB 接口时钟频率 (**rng_hclk**) 无关。

注： *第 34.7 节：熵源验证中提供了建议使用的 RNG 时钟频率。*

后期处理

从真随机噪声源获取的采样值由 2 位的位串组成。由于该噪声源输出有偏差，因此 RNG 会利用后期处理组件将偏差降至可接受的水平。

更具体来讲，对于每个噪声源产生的 2 位数值，一位是由 RNG 取自采样噪声源，另一位是对取自采样噪声源的值取反。因此，如果噪声源生成的“1”多于“0”（或者相反），则噪声源会被过滤掉。

调节

RNG 中的调节组件是确定性函数，可以提高生成固定长度位串输出（128 位）的熵率。

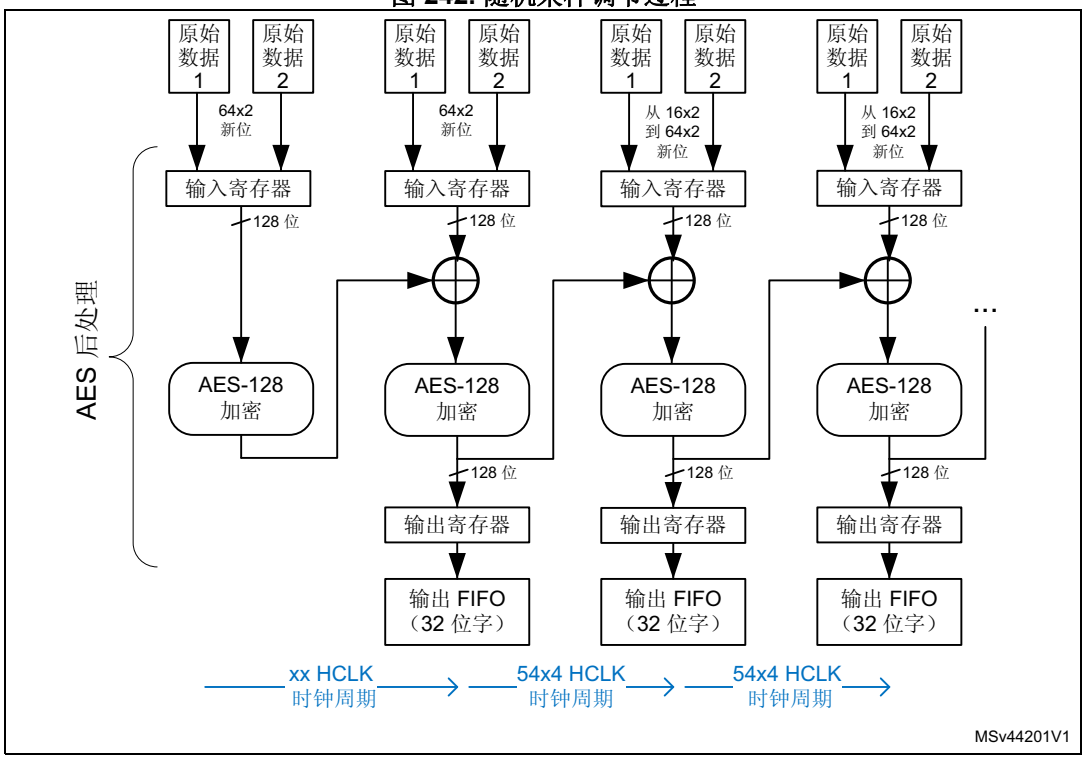
使用的调节算法是基于 AES CBC 的块加密，支持 128 位的密钥长度，因此每 54x4 个 AHB 时钟周期可生成 128 位随机采样，如*图 242* 所示。

注： *第 34.6 节：RNG 处理时间中介绍了 RNG 初始化过程中的延时。*

两个噪声源经过数字化处理后，其输出的原始数据交错在一起，作为 AES 后期处理级的输入（原始位 1 + 原始位 2 + 原始位 1 + 原始位 2 ...）。

另外请注意，如果已经接收了 32 或更多位原始数据，并且输出 FIFO 需要重新填充，则会触发 AES 计算。因此，当 RNG 128 位 FIFO 在 64 个 RNG 时钟周期后被应用程序清空时，RNG 输出熵为最大值。

图 242. 随机采样调节过程



调节组件由更快的 AHB 时钟提供时钟信号。

注： *每个 RNG 时钟周期可向 AES 输入缓冲区载入两个噪声源位。*

输出缓冲区

数据输出缓冲区最多可存储四个从调节组件 (AES-CBC) 输出的 32 位字。如果已通过 RNG_DR 寄存器从输出 FIFO 读取了四个字，则 128 位调节输出寄存器的内容会被推送到输出 FIFO 中，同时自动启动新的 AES 计算。213 个 AHB 时钟周期后（执行 AES 计算的时间），会将四个新字添加到调节输出寄存器。

当通过 RNG_DR 寄存器可获取随机数时，DRDY 标志就从“0”变为“1”。该标志会保持在置 1 状态，直至从 RNG_DR 寄存器读取四个字后输出缓冲区变空。

注： *如果使能了中断，则当该数据就绪标志从“0”变为“1”时，会产生中断。随后，中断会自动由 RNG 清零，如上所述。*

运行状态检查

该组件可确保整个熵源（包括其噪声源）正常启动并按预期运行，能够快速准确地定位故障，从而提供了高可靠性保证。

RNG 可实现以下运行状态检查功能：

1. 行为测试，可在熵源*运行时*进行
   - 重复计数测试，在以下情况时会指示发生错误：
   a) 其中一个噪声源持续不变地输出了 64 位或以上的“0”或“1”，或者 32 个或以上的“01”或“10”。
   b) 两个噪声源都持续不变地输出了 32 位或以上的“0”或“1”，或者 16 个或以上的“01”或“10”。
2. 厂商专用连续测试
   - 实时“过慢”采样时钟检测器，如果一个 RNG 时钟周期（在分频之后）小于 AHB 时钟周期的 32 分之一，则该检测器会指示错误。

RNG_SR 寄存器中的 CECS 和 SECS 状态位会在检测到错误条件时进行指示，*第 34.3.7 节：错误管理*对此进行了详细说明。

注： *检测到错误时生成中断。*

34.3.4 RNG初始化

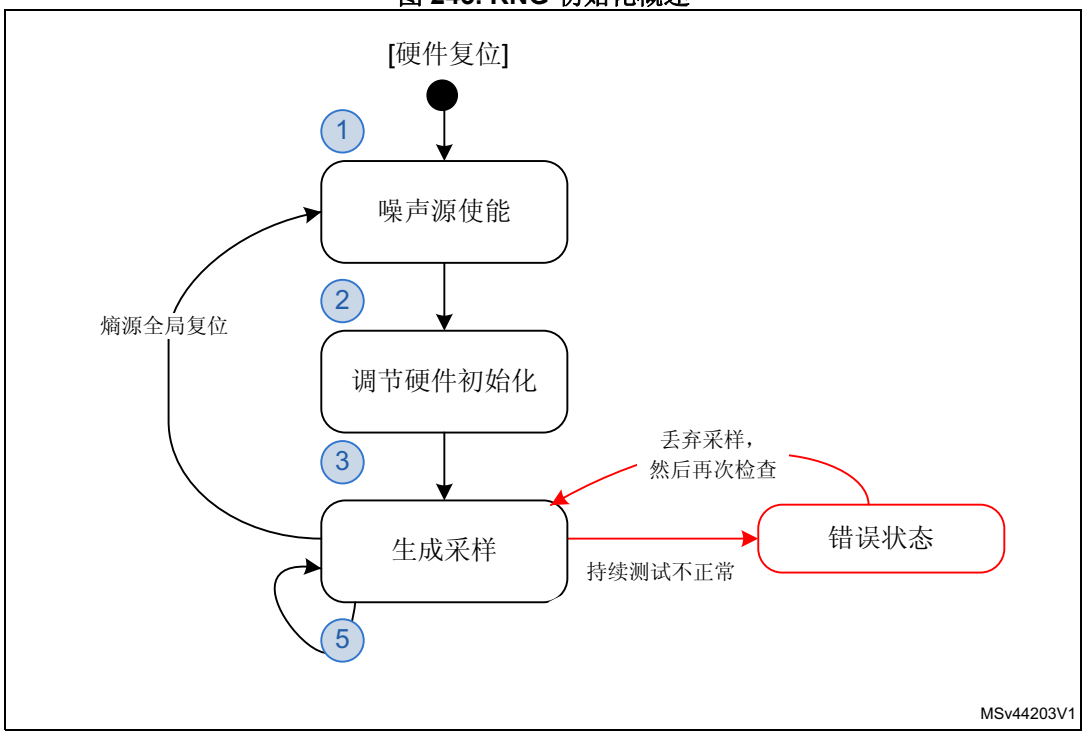
RNG 简化状态机如*图 243* 所示。

如果发生硬件复位，则会发生以下一系列事件：

1. 使能模拟噪声源，逻辑系统立即开始采样模拟输出并填充 128 位调节移位寄存器。
2. 使能调节逻辑系统，使用两个 128 噪声源位（128 位用于密钥）初始化 AES-CBC 后期处理上下文。
3. 再次用 128 位数据填充 AES 内部输入数据缓冲区，并执行一次 AES 后期处理循环。然后使用 AES 后期处理结果填充输出缓冲区。
4. 输出缓冲区会按照 RNG 使用情况自动重填。

相关初始化时间请参见*第 34.6 节：RNG 处理时间*。

图 243. RNG 初始化概述



34.3.5 RNG 操作

正常工作

要利用中断运行 RNG，建议执行以下步骤：

1. 通过将 RNG_CR 寄存器中的 IE 位置 1 来使能中断。同时通过将 RNGEN 位置 1 来使能 RNG。
2. 这以后，准备好随机数时或出现错误时就产生中断。因此，每次发生中断时，应检查：
   - 未发生错误。RNG_SR 寄存器中的 SEIS 和 CEIS 位应设为“0”。
   - 随机数准备就绪。RNG_SR 寄存器中的 DRDY 位必须置“1”。
   - 如果满足以上两个条件，最多可连续四次读取 RNG_DR 寄存器的内容。如果 AES 输出缓冲区中存在有效数据，则应用程序可以额外读取四个字（此时 DRDY 位仍保持为 1）。如果上述条件中有一个或两个都无法满足，则不得读取 RNG_DR 寄存器。如果发生错误，则应使用*第 34.3.7 节*中介绍的错误恢复序列。

要在轮询模式下运行 RNG，建议执行以下步骤：

1. 通过将 RNG_CR 寄存器中的 RNGEN 位置“1”使能随机数生成。
2. 读取 RNG_SR 寄存器并检查：
   - 未发生错误（SEIS 和 CEIS 位应设为“0”）
   - 随机数准备就绪（DRDY 位应置“1”）
3. 如果满足以上两个条件，最多可连续四次读取 RNG_DR 寄存器的内容。如果 AES 输出缓冲区中存在有效数据，则应用程序可以额外读取四个字（此时 DRDY 位仍保持为 1）。如果上述条件中有一个或两个都无法满足，则不得读取 RNG_DR 寄存器。如果发生错误，则应使用*第 34.3.7 节*中介绍的错误恢复序列。

注： *如果数据未就绪（DRDY = “0”），则 RNG_DR 会返回 0。*

低功耗运行

如果应用程序比较关注功耗，则可以使用低功耗策略，如*第 1129 页的第 34.4 节：RNG 低功耗使用*所述。

软件后期处理

无需进行特定的软件后期处理/调节即可满足 AIS-31 认证。如果需要安全强度为 128 位、符合 NIST 要求的 DRBG，则必须基于 RNG 真随机数发生器构建符合要求的随机数发生器软件。

*第 34.3.3 节：随机数生成*介绍了内置的运行状态检查功能。

34.3.6 RNG 时钟

RNG 在两个不同的时钟上运行：AHB 总线时钟和专用 RNG 时钟。

AHB 时钟用于为 AHB 存储寄存器和调节组件提供时钟信号。RNG 时钟用于噪声源采样。*第 34.7 节：熵源验证*中详细介绍了建议使用的时钟配置。

注意： 如果 RNG_CR 寄存器中的 CED 位置“0”，那么 RNG 时钟频率**必须大于** AHB 时钟频率的 32 分之一，否则时钟检查器会指示时钟错误（RNG_SR 寄存器中的 CECS 或 CEIS），RNG 将停止生成随机数。

详细信息参见*第 34.3.1 节：RNG 框图*（AHB 和 RNG 时钟域）。

34.3.7 错误管理

运行状态检查模块在生成随机数的同时还验证噪声源行为以及 RNG 源时钟频率是正常的，具体说明见本部分。另外，还描述了相关的错误状态。

时钟错误检测

如果时钟错误检测已使能 (CED = 0) 且 RNG 时钟频率过低，RNG 会停止生成随机数，并会将 **CEIS** 和 **CECS** 位都置“1”，指示发生了时钟错误。在这种情况下，应用程序应检查 RNG 时钟是否配置正确（参见*第 34.3.6 节：RNG 时钟*），随后必须将中断标志 CEIS 位清零。RNG 时钟正常运行后，CECS 将立即自动清零。

仅当 CECS 位置“0”时，RNG 才能正常运行。但需要注意的是，时钟错误对之前生成的随机数没有影响，因此 RNG_DR 寄存器内容仍可使用。

噪声源错误检测

如果发生噪声源（或种子）错误，RNG 会停止生成随机数，并会将 **SEIS** 和 **SECS** 位都置“1”，指示发生了种子错误。如果 RNG_DR 寄存器中有可用值，不能使用该值，因为它可能没有足够的熵。如果在初始化阶段检测到错误，则 RNG 将自动重启整个初始化序列。

要从 RNG 初始化后的种子错误中完全恢复，应使用以下序列：

1. 向 SEIS 位写入“0”，将该位清零。
2. 从 RNG_DR 寄存器读取 12 个字，并将每个字丢弃，将管道清空。
3. 确认 SEIS 仍处于清零状态。随机数生成恢复正常。

34.4 RNG 低功耗使用

考虑到功耗问题，可在 DRDY 位置“1”后立即禁止 RNG，具体方法为将 RNG_CR 寄存器中的 RNGEN 位设为“0”。由于 RNGEN =“0”时 AES 后期处理逻辑系统和输出缓冲区仍处于工作状态，因此软件可使用以下功能：

- 如果输出缓冲区中存在有效字，仍可从 RNG_DR 寄存器中读取四个随机数。
- 如果 AES 输出内部寄存器中存在有效位，仍可从 RNG_DR 寄存器中额外读取四个随机数。如果不是这样的话，则必须由应用程序重新使能 RNG，直至从噪声源采集了至少 32 个新的位并且完成了完整的 AES 计算。这对应于 16 个 RNG 时钟周期来采样新的位，216 个 AHB 时钟周期运行一轮 AES。

禁止 RNG 时，用户会禁用所有模拟种子发生器，各发生器的功耗列于数据手册中的电气特性部分。用户还会对所有由 RNG 时钟提供时钟信号的逻辑系统进行门控。请注意，由于 RNG 需要一定的时间进行初始化，因此该策略会在一段延时后才可在 RNG_DR 寄存器中提供随机采样。

如果 RNG 模块在初始化期间（也就是刚好在 DRDY 位首次置 1 之前）被禁止，则初始化序列将从 RNGEN 位置“1”时停止的位置恢复运行。

34.5 RNG 中断

在 RNG 中，发生以下事件时会产生中断：

- 数据就绪标志
- 种子错误，请参见*第 34.3.7 节：错误管理*
- 时钟错误，请参见*第 34.3.7 节：错误管理*

可以使用专用的中断使能控制位，如*表 260* 所示

表 260. RNG 中断请求

中断事件	事件标志	使能控制位
数据就绪标志	DRDY	IE
种子错误标志	SEIS	IE
时钟错误标志	CEIS	IE

用户可以通过更改 RNG_CR 寄存器中的屏蔽位或通用中断控制位 IE 的方式分别使能或禁止上述中断源。可以从 RNG_SR 寄存器中读取各个中断源的状态。

注： *仅当 RNG 已使能时，才会产生中断。*

34.6 RNG 处理时间

AES 每 4x54 个 AHB 时钟周期可生成四个 32 位随机数，但在设备退出复位状态后需要更长的时间才能生成第一组随机数（请参见*第 34.3.4 节：RNG 初始化*）。

首次使能 RNG 后，会在以下时间后首次提供随机数据：

- 128 RNG 个时钟周期 + 426 个 AHB 周期（当 f_{AHB} < 160 MHz 且 f_{RNG} = 48 MHz 时）
- 192 RNG 个时钟周期 + 213 个 AHB 周期（当 f_{AHB} ≥ 160 MHz 且 f_{RNG} = 48 MHz 时）

34.7 熵源验证

34.7.1 前言

为了评估 RNG 可提供的熵大小，意法半导体按照 AIS-31 PTG.2 测试集对 RNG 进行了测试。测试结果可根据需要提供，客户也可以使用 AIS 参考软件重现测量结果。客户还可以按照较早的 NIST SP800-22 测试集对 RNG 进行测试。

34.7.2 验证条件

意法半导体在以下条件下对 RNG 真随机数发生器进行了验证：

- RNG 时钟 rng_clk= 48 MHz（RNG_CR 寄存器中的 CED 位 =“0”）且 rng_clk = 400 kHz（RNG_CR 寄存器中的 CED 位 =“1”）
- AHB 时钟 rng_hclk = 216 MHz

34.7.3 数据采集

为了运行统计测试，需要对熵源的原始数据和输出进行采样。

RNG 策略是使用 RNG_CR 寄存器中的 BYP 位设置是否绕过调节阶段，从而可以使用同一个 32 位缓冲区输出两种采样。换句话说，如果应用程序需要获取原始数据，则建议采用以下序列：

1. 向 RNG_CR 寄存器中的 RNGEN 位写入“0”，向 BYP 位写入“1”。
2. 向 RNG_CR 寄存器中的 RNGEN 位写入“1”，向 BYP 位写入“1”。原始采样与正常输出采样的获取方式相同。

34.8 RNG 寄存器

RNG 与控制寄存器、数据寄存器和状态寄存器相关联。

34.8.1 RNG 控制寄存器 (RNG_CR)

RNG control register

偏移地址：0x000

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BYP	CED	Res.	IE	RNGEN	Res.	Res.
									rw	rw		rw	rw		

位 31:7 保留，必须保持复位值

位 6 **BYP**：旁路模式使能 (Bypass mode enable)
该位可使能或禁止绕过后期处理/调节逻辑系统。此功能用于 RNG 验证。
0：禁止旁路模式。噪声源采样进行后期处理和调节。
1：使能旁路模式 噪声源采样既未进行后期处理，也未进行调节，可直接通过应用程序读取。
仅当 RNGEN=0 时，才考虑向该位写入数据。

位 5 **CED**：时钟错误检测 (Clock error detection)
0：使能时钟错误检测
1：禁止时钟错误检测
当使能 RNG 时，不能实时使能或禁止时钟错误检测，即必须禁止 RNG 才能使能或禁止 CED。

位 4 保留，必须保持复位值。

位 3 **IE**：中断使能 (Interrupt enable)
0：禁止 RNG 中断
1：使能 RNG 中断。如果 RNG_SR 寄存器中 DRDY = “1”、SEIS = “1”或 CEIS = “1”，则中断处于挂起状态。

位 2 **RNGEN**：真随机数发生器使能 (True random number generator enable)
0：禁止真随机数发生器。模拟噪声源关闭，并会对由 RNG 时钟提供时钟信号的逻辑系统进行门控。
1：使能真随机数发生器。

位 1:0 保留，必须保持复位值

34.8.2 RNG 状态寄存器 (RNG_SR)

RNG status register

偏移地址：0x004

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SEIS	CEIS	Res.	Res.	SECS	CECS	DRDY
									rc_w0	rc_w0			r	r	r

位 31:7 保留，必须保持复位值

位 6 **SEIS**：种子错误中断状态 (Seed error interrupt status)

此位与 SECS 同时置 1，通过向其写入“0”来清零。

0：未检测到错误序列

1：至少检测到一个错误序列。有关详细信息，请参见 **SECS** 位说明。

如果 RNG_CR 寄存器中 IE =“1”，则存在待处理的中断。

位 5 **CEIS**：时钟错误中断状态 (Clock error interrupt status)

此位与 CECS 同时置 1，通过向其写入“0”来清零。

0：RNG 时钟正确 (fRNGCLK > fHCLK/32)

1：已检测到 RNG 时钟过慢 (fRNGCLK < fHCLK/32)

如果 RNG_CR 寄存器中 IE =“1”，则存在待处理的中断。

位 4:3 保留，必须保持复位值

位 2 **SECS**：种子错误当前状态 (Seed error current status)

0：目前未检测到错误序列。如果 SEIS 位置 1，则意味着已检测到错误序列并已恢复正常。

1：至少检测到一个以下错误序列：

– 其中一个噪声源持续不变地输出了 64 位或以上的“0”或“1”，或者 32 个或以上的“01”或“10”。

– 两个噪声源都持续不变地输出了 32 位或以上的“0”或“1”，或者 16 个或以上的“01”或“10”。

位 1 **CECS**：时钟错误当前状态 (Clock error current status)

0：RNG 时钟正确 (fRNGCLK > fHCLK/32)。如果 CEIS 位置 1，则意味着已检测到时钟过慢并已恢复正常。

1：RNG 时钟过慢 (fRNGCLK < fHCLK/32)。

注： 只有 RNG_CR 寄存器中的 CED 位置“0”时，CECS 位才有效。

位 0 **DRDY**：数据就绪 (Data ready)

0：RNG_DR 寄存器尚未有效，无可用随机数据

1：RNG_DR 寄存器包含有效随机数据

在输出缓冲区变空之后（读取 RNG_DR 寄存器之后），在生成新的随机值之前，该位返回“0”。

注： 当禁用外设时（RNG_CR 寄存器中的 RNGEN =“0”），可以将 DRDY 位置 1。

如果 RNG_CR 寄存器中的 IE =“1”，则在 DRDY =“1”时生成中断。

34.8.3 RNG 数据寄存器 (RNG_DR)

RNG data register

偏移地址：0x008

复位值：0x0000 0000

RNG_DR 寄存器是只读寄存器，在读取时提供 32 位随机数值。读取该寄存器后，如果输出 FIFO 为空，则该寄存器将在 216 个 AHB 时钟周期之后提供一个新的随机值。

当 DRDY = “1” 时，即使 RNGEN = “0”，该寄存器的内容也是有效的。

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

位 31:0 **RNDATA[31:0]**：随机数据 (Random data)

DRDY = “1” 时有效的 32 位随机数据。DRDY = “0” 时，RNDATA 值为 0。

34.8.4 RNG 寄存器映射

表 261 给出了 RNG 寄存器映射和复位值。

表 261. RNG 寄存器映射和复位映射

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x000	RNG_CR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BYP	CED	Res.	IE	RNGEN	Res.	Res.
	Reset value																										0	0		0	0		
0x004	RNG_SR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SEIS	CEIS	Res.	Res.	SECS	CECS	DRDY
	Reset value																										0	0			0	0	0
0x008	RNG_DR	RNDATA[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

35 加密处理器 (CRYP)

35.1 简介

借助加密处理器 (CRYP)，可使用 DES、Triple-DES (3DES) 或 AES 算法对数据进行加密和解密。加密处理器完全兼容下列标准：

- 联邦信息处理标准出版物（FIPS PUB 46-3，1999 年 10 月）以及美国国家标准协会 (ANSI X9.52) 规定的数据加密标准 (DES) 和3DES (TDES)
- 联邦信息处理标准出版物（FIPS PUB 197，2001 年 11 月）规定的高级加密标准 (AES)

支持多种密钥大小和链接模式：

- DES/TDES 链接模式 ECB 和 CBC，支持标准 56 位密钥（每个密钥带 8 位奇偶校验）
- AES 链接模式 ECB、CBC、CTR、GCM、GMAC、CCM，支持 128、192 或 256 位密钥大小

CRYP 为 32 位 AHB 外设。它支持对传入和传出数据进行 DMA 传输（需要两个 DMA 通道）。该外设还包含输入和输出 FIFO（分别为 8 个字的深度），有助于实现更佳性能。

CRYP 外设可为 STM32 加密库所实现的 AES 和 DES 加密算法提供硬件加速。

35.2 CRYP 主要特性

- 兼容下列标准：
  - NIST *FIPS 出版物 46-3，数据加密标准 (DES)*
  - ANSI *X9.52，三重数据加密算法工作模式*
  - NIST *FIPS 出版物 197，高级加密标准 (AES)*
- AES 对称块密码实现
  - 128 位数据块处理
  - 支持 128 位、192 位和 256 位密钥长度
  - 基于多种链接模式的加密和解密：电子密码本 (ECB)、密码块链接 (CBC)、计数器模式 (CTR)、Galois 计数器模式 (GCM)、Galois 消息认证码模式 (GMAC) 和 CBC-MAC 计数器模式 (CCM)。
  - 14（或 18）个时钟周期，用于在 AES-ECB 模式下处理一个包含 128 位（或 256 位）密钥的 128 位数据块
  - 集成有密钥调度器，包括密钥分散阶段（仅限 ECB 或 CBC 解密）
- DES/TDES 加密/解密实现
  - 64 位数据块处理
  - 支持 64 位、128 位和 192 位密钥长度（包括奇偶校验）
  - 基于 ECB 和 CBC 链接模式的加密和解密
  - 直接执行简单 DES 算法（使用单一密钥 K1）
  - 16（或 64）个时钟周期，用于在 DES（或 TDES）ECB 模式下处理一个 64 位数据块
  - 密文窃取的软件实现

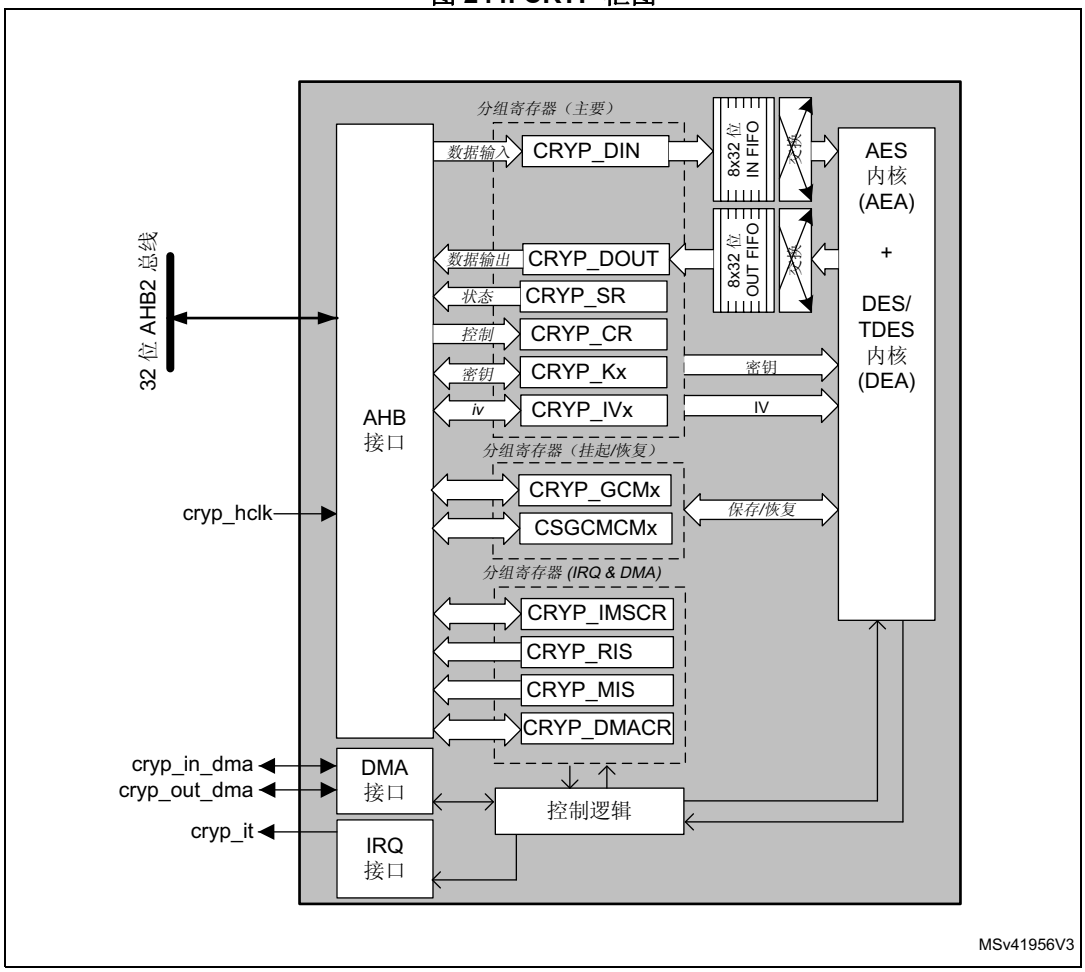
- DES/TDES 和 AES 的共同特性
  - AMBA AHB 从外设，仅支持 32 位字单次访问（否则，会生成 AHB 总线错误，并会忽略写访问）
  - 256 位寄存器，用于存储加密密钥（8 个 32 位寄存器）
  - 128 位寄存器，用于存储初始化向量（4 个 32 位寄存器）
  - 1 个 32 位输入缓冲器，它与八个 32 位字的内部 IN FIFO 相关联，对应于四个传入 DES 块或两个 AES 块
  - 1 个 32 位输出缓冲器，它与八个 32 位字的内部 OUT FIFO 相关联，对应于四个已处理的 DES 块或两个 AES 块
  - 采用自动数据流控制，支持直接存储器访问 (DMA)（使用两个通道，分别用于传入数据和已处理数据）。支持单次传输和突发传输。
  - 采用数据交换逻辑，支持 1 位、8 位、16 位或 32 位数据
  - 在加密处理器需要处理优先级更高的消息时，可通过软件将当前消息挂起（挂起/恢复操作）

35.3 CRYP 功能说明

35.3.1 CRYP 框图

图 244 显示了加密处理器的框图。

图 244. CRYP 框图



35.3.2 CRYP 内部信号

表 262 列出了加密处理器级而非 STM32 产品级（焊盘上）所提供的内部信号，对这些信号有所了解会很有用。

表 262. CRYP 内部输入/输出信号

信号名称	信号类型	说明
cryp_hclk	数字输入	AHB 总线时钟
cryp_it	数字输出	加密处理器全局中断请求
cryp_in_dma	数字输入/输出	IN FIFO DMA 突发请求/确认
cryp_out_dma	数字输入/输出	OUT FIFO DMA 突发请求/确认（包括对 DES 的单次请求）

35.3.3 CRYP DES/TDES 加密内核

概述

DES/Triple-DES (3DES) 加密内核由三部分组成：

- DES 算法（DEA 内核）
- 多个密钥（DES 算法使用一个密钥，TDES 算法使用一到三个密钥）
- 初始化向量（仅在 CBC 模式中使用）

DES/三重 DES 加密内核支持两种工作模式：

- **ALGODIR=0**：使用存储在 CRYP_Kx 寄存器中的密钥实现明文加密。
- **ALGODIR=1**：使用存储在 CRYP_Kx 寄存器中的密钥实现密文解密。

通过对 CRYP_CR 寄存器中的 ALGODIR 位进行编程来选择工作模式。

典型数据处理

*第 35.3.10 节：CRYP DES/TDES 基本链接模式（ECB 和 CBC）*对 DES 模式下加密处理器的典型用法进行了介绍。

注： *中间 DEA 阶段的输出始终不会在加密边界外暴露，CBC 模式下的 IV 寄存器除外。*

DES 密钥和链接模式

TDES 允许使用三种不同的密钥选项：

- *三个独立的密钥*

  第一个选项指定所有的密钥均是独立的，即 K1、K2 和 K3 均是独立的。FIPS PUB 46-3 – 1999（以及 ANSI X9.52 – 1998）将该选项称为“密钥选项 1”，并将这种 TDES 称为“3 密钥 TDES”。
- *两个独立的密钥*

  第二个选项指定 K1 和 K2 是独立的，而 K3 等于 K1，即 K1 和 K2 独立，而 K3 = K1。FIPS PUB 46-3 – 1999（以及 ANSI X9.52 – 1998）将该选项称为“密钥选项 2”，并将这种 TDES 称为“2 密钥 TDES”。
- *三个相同的密钥*

  第三个选项指定 K1、K2 和 K3 相等，即：

  K1 = K2 = K3

  FIPS PUB 46-3 – 1999（以及 ANSI X9.52 – 1998）将第三个选项称为“密钥选项 3”。这种“1 密钥”TDES 与单独 DES 相同。

以下链接算法受 DES 硬件支持，可通过 CRYP_CR 寄存器中的 ALGOMODE 位进行选择：

- 电子密码本（Electronic Code Book，ECB）
- 密码块连接（Cipher Block Chaining，CBC）

*第 35.3.10 节：CRYP DES/TDES 基本链接模式（ECB 和 CBC）*详细介绍了这些模式。

35.3.4 CRYP AES 加密内核

概述

AES 加密内核由以下部分组成：

- AES 算法（AEA 内核）
- 基于二元 Galois 域的乘法器 (GF2mul)
- 密钥信息
- 初始化向量 (IV) 或随机值信息
- 链接算法逻辑（XOR、反馈/计数器、屏蔽）

AES 内核适用于密钥长度为 128 位、192 位或 256 位的 128 位数据块（四字）。根据链接模式的不同，外设可能需要一个 128 位初始化向量 (IV)，也可能不需要。

加密外设支持以下两种工作模式：

- **ALGODIR=0**：使用存储在 CRYP_Kx 寄存器中的密钥实现明文加密。
- **ALGODIR=1**：使用存储在 CRYP_Kx 寄存器中的密钥实现密文解密。选择 ECB 和 CBC 链接模式时，加密外设会自动执行初始密钥生成过程。

通过对 CRYP_CR 寄存器中的 ALGODIR 位进行编程来选择工作模式。

典型数据处理

*第 35.3.11 节：CRYP AES 基本链接模式（ECB 和 CBC）*介绍了 AES 模式下加密处理器的典型用法。

注： *中间 AEA 阶段的输出始终不会在加密边界外显示，IV 寄存器除外。*

AES 链接模式

以下链接算法受加密处理器支持，可通过 CRYP_CR 寄存器中的 ALGOMODE 位进行选择：

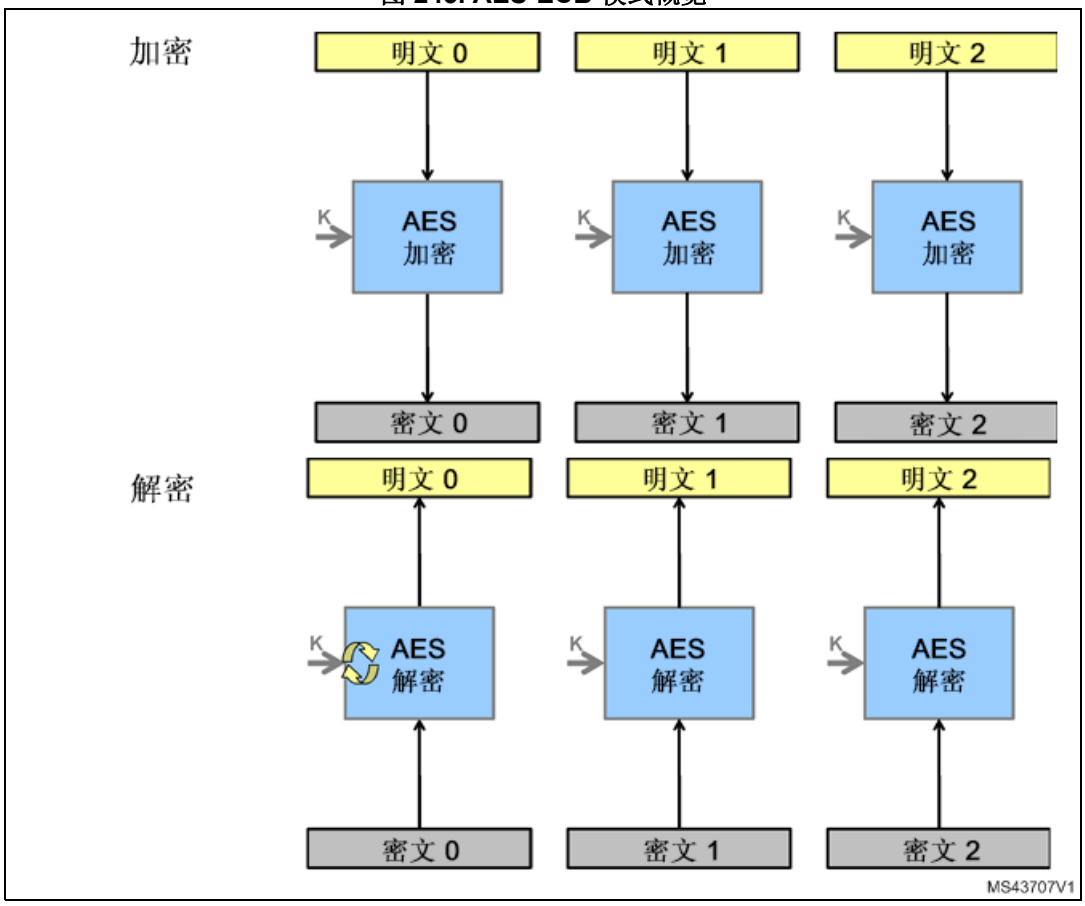
- 电子密码本（Electronic Code Book，ECB）
- 密码块连接（Cipher Block Chaining，CBC）
- 计数器模式 (CTR)
- Galois 计数器模式 (GCM)
- Galois 消息认证码模式 (GMAC)
- CBC-MAC 计数器模式 (CCM)

后续子章节对这些链接模式进行了简单介绍。

有关详细说明，请参见*第 35.3.11 节：CRYP AES 基本链接模式（ECB 和 CBC）*以及之前的章节。

AES 电子密码本 (ECB)

图 245. AES-ECB 模式概览

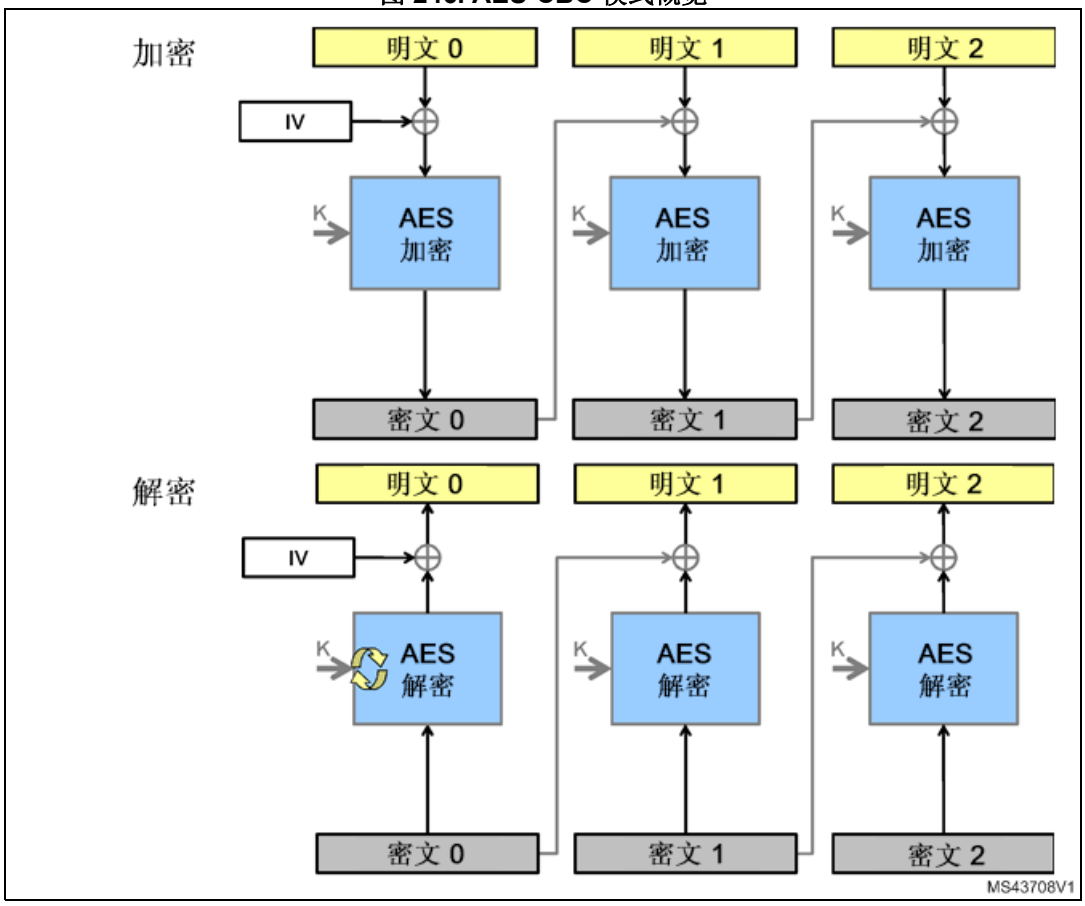


ECB 是最简单的工作模式。无链接操作，且无特殊初始化阶段。消息会划分到各个块中，并对各个块分别进行加密或解密。

注： *对于解密，在处理第一个块之前需要特殊的密钥调度。*

AES 密码块链接 (CBC)

图 246. AES-CBC 模式概览

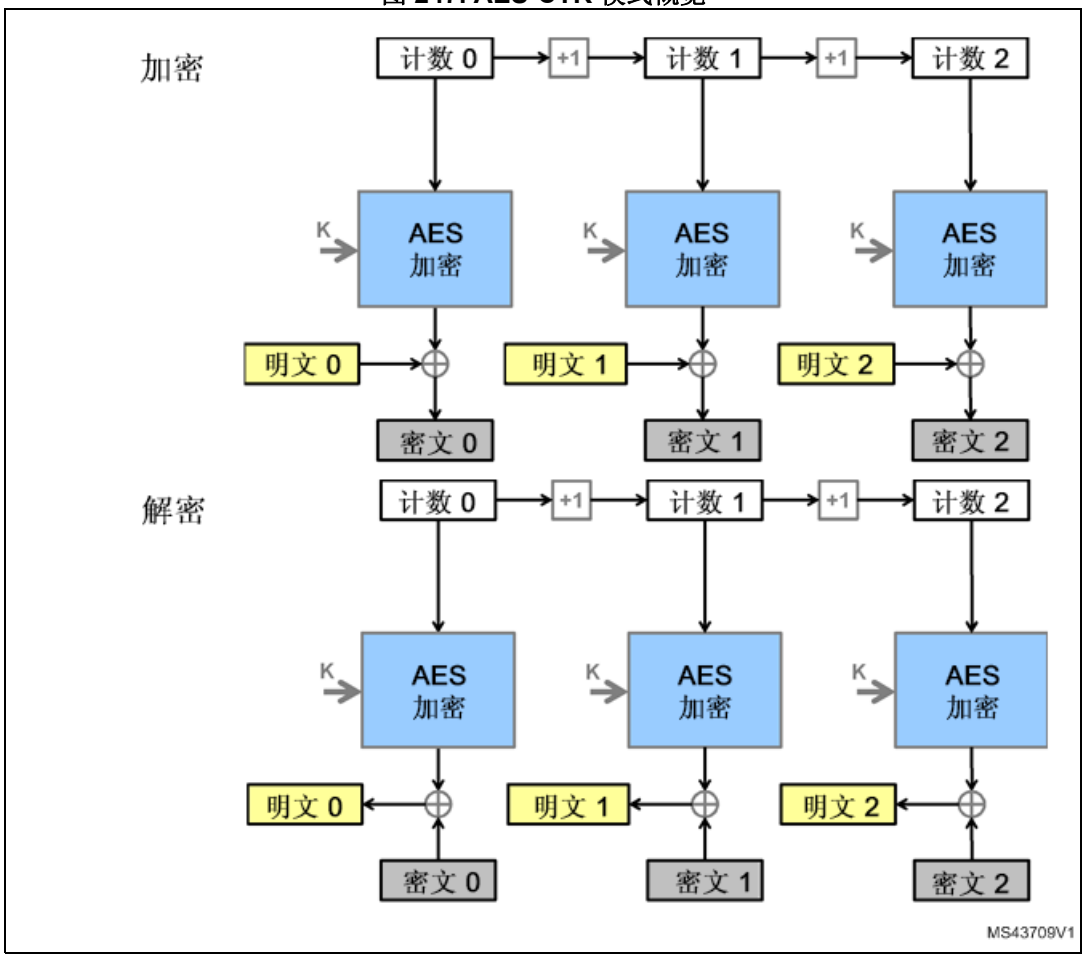


在 CBC 工作模式下，每个块的输出均与下一个块的输入相连。为了确保每条消息都是唯一的，会在第一个块处理过程中使用初始化向量。

注： *对于解密，在处理第一个块之前需要特殊的密钥调度。*

AES 计数器模式 (CTR)

图 247. AES-CTR 模式概览

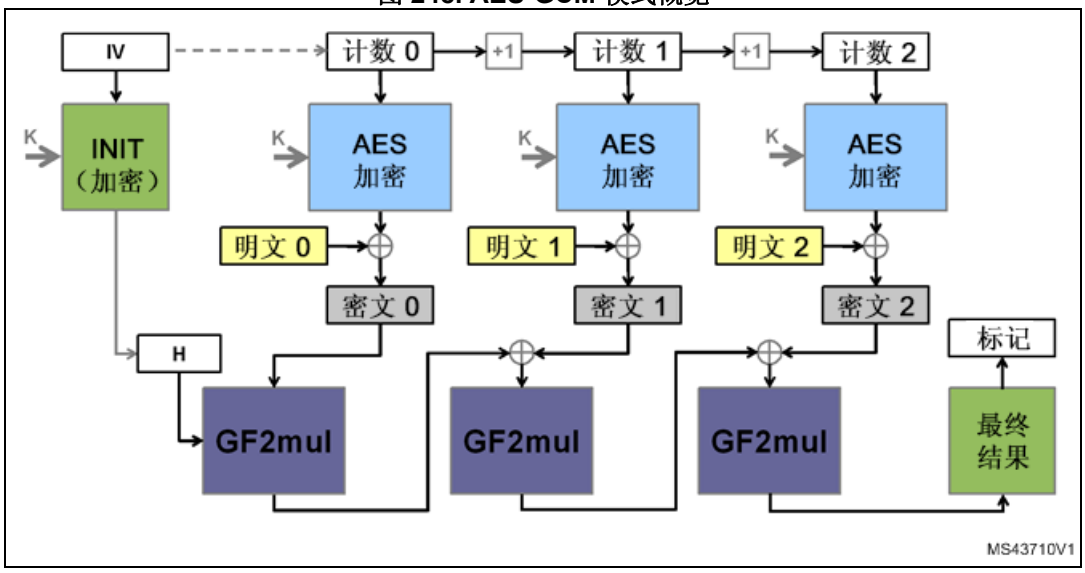


CTR 模式使用 AES 内核生成密钥流；这些密钥将与明文进行异或运算，以获得密文（如 NIST *特别出版物 800-38A，块密码工作模式的建议*中所述）。

注： *与 ECB 和 CBC 模式不同，CTR 解密无需密钥调度，因为在该链接方案中，AES 内核始终在加密模式下用于生成计数器块。*

AES Galois/计数器模式 (GCM)

图 248. AES-GCM 模式概览

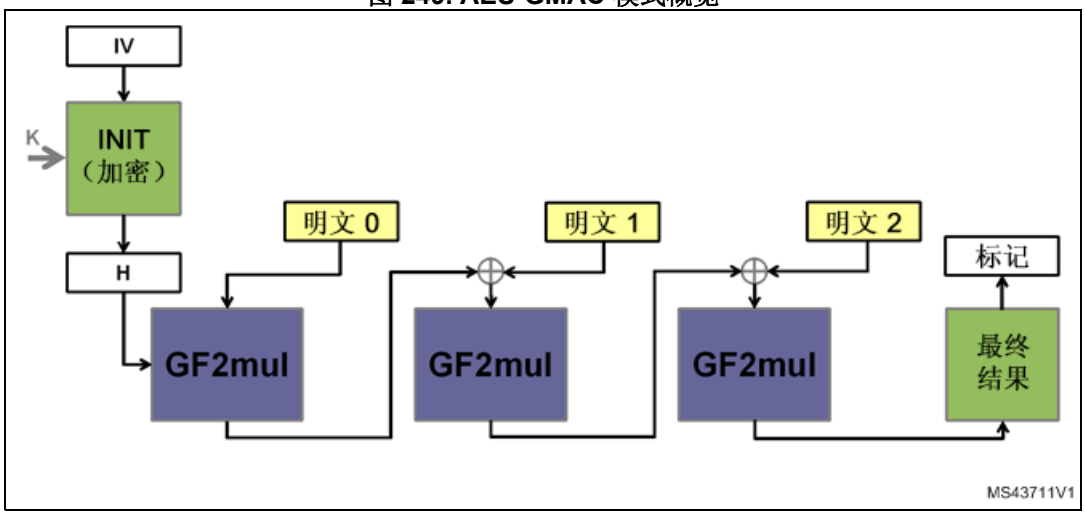


在 Galois/计数器模式 (GCM) 下，将对明文消息进行加密，同时会对消息认证码 (MAC) 进行计算，从而生成相应的密文及其 MAC（也称为认证标记）。NIST *特别出版物 800-38D，块密码工作模式的建议 - Galois/计数器模式 (GCM) 和 GMAC* 中对此进行了相关定义。

GCM 模式基于 AES 计数器模式，可实现保密性。它使用固定有限域乘法器来计算消息认证码。消息末尾处需要一个初始值和一个特定的 128 位块。

AES Galois 消息认证码 (GMAC)

图 249. AES-GMAC 模式概览

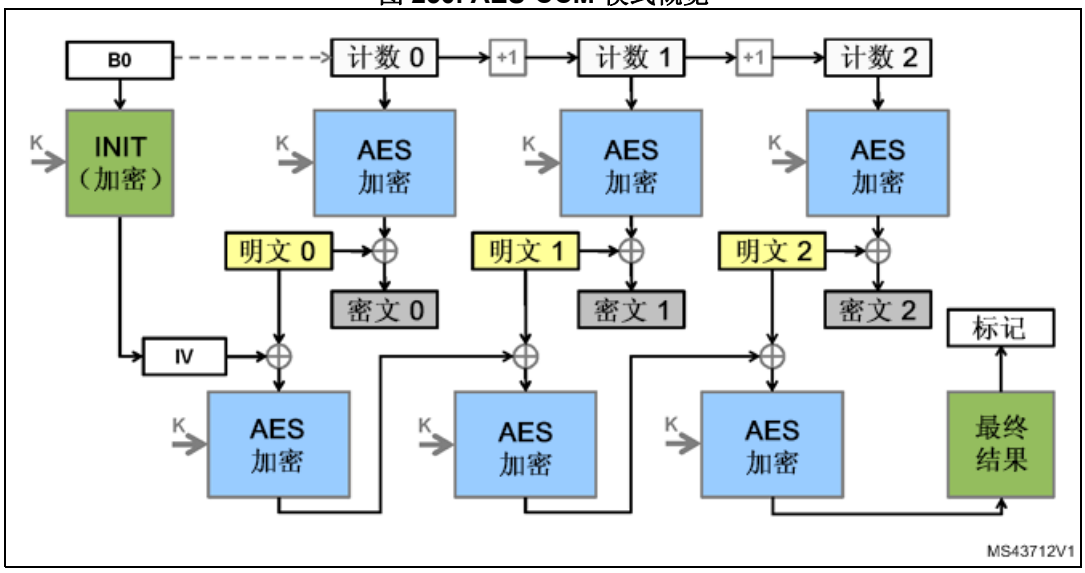


Galois 消息认证码 (GMAC) 允许认证消息并生成相应的消息认证码 (MAC)。NIST *特别出版物 800-38D，块密码工作模式的建议 - Galois/计数器模式 (GCM) 和 GMAC* 中对此进行了相关定义。

GMAC 与 Galois/计数器模式 (GCM) 类似，只不过它应用于仅由明文认证数据组成的消息（即只有标头，无有效负载）。

AES CBC-MAC 计数器模式 (CCM)

图 250. AES-CCM 模式概览



在密码块链接-消息认证码计数器模式 (CCM) 下，将对明文消息进行加密，同时会对消息认证码 (MAC) 进行计算，从而生成相应的密文以及相应的 MAC（也称为标记）。NIST *特别出版物 800-38C，块密码工作模式的建议 - CCM 模式实现认证和保密性*中对此进行了介绍。

CCM 模式基于 AES 计数器模式，可实现保密性，并且使用 CBC 来计算消息认证码。它需要一个初始值。

与 GCM CCM 链接模式一样，AES-CCM 模式可应用于仅由明文认证数据组成的消息（即只有标头，无有效负载）。请注意，这种 CCM 使用方式不被称为 CMAC（它并非类似于 GCM/GMAC），NIST 不推荐这种用途。

35.3.5 用于执行密码操作的 CRYP 过程

简介

为了理解加密外设的工作原理，下面对典型密码操作进行了介绍。有关不同密码模式下的外设使用的详细信息，请参见具体章节，例如*第 35.3.11 节：CRYP AES 基本链接模式（ECB 和 CBC）*。

图 251 和*图 252* 中的流程图介绍了 STM32 加密库如何实现 DES（或 AES）算法。加密处理器为以下加密算法的执行实现了加速：

- 以下模式下的 AES-128、AES-192 和 AES-256 位：ECB、CBC、CTR、CCM、GCM
- 以下模式下的 DES 和 TripleDES (3DES)：ECB、CBC

注： *有关加密库的更多详细信息，请参见使用手册 UM1924“STM32 加密库”（可从 www.st.com 获取）*

图 251. STM32 加密库 DES/TDES 流程图

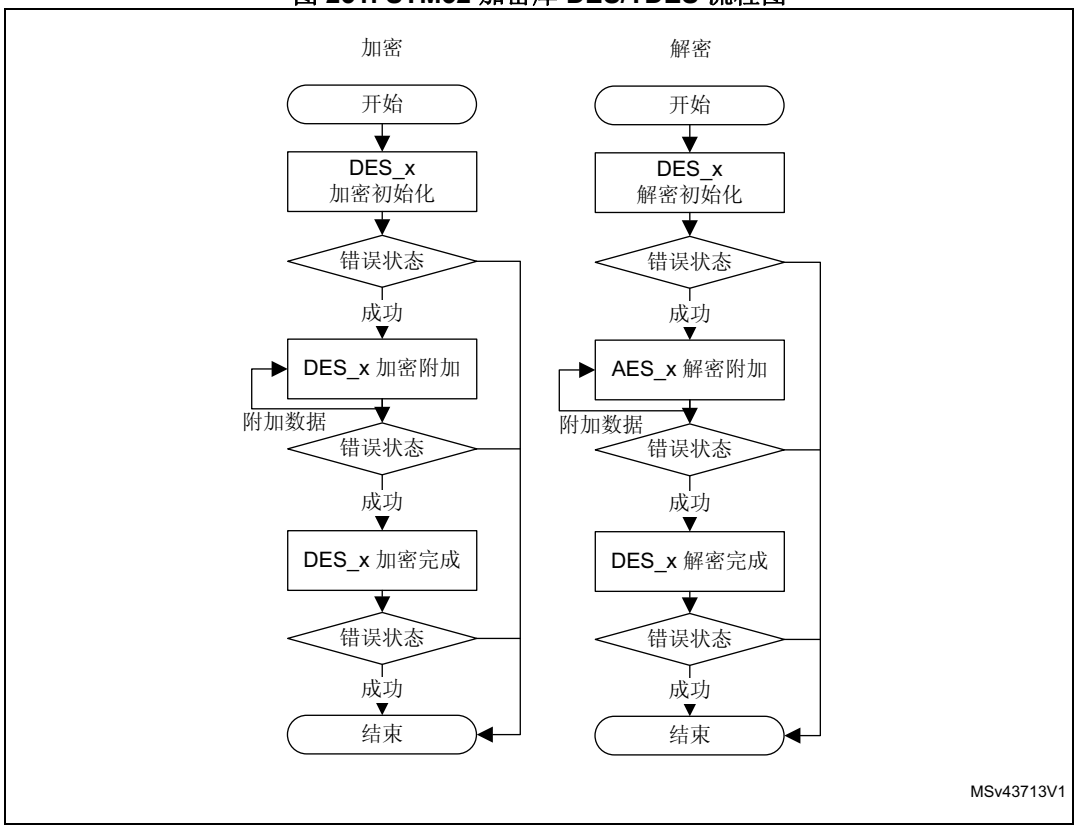
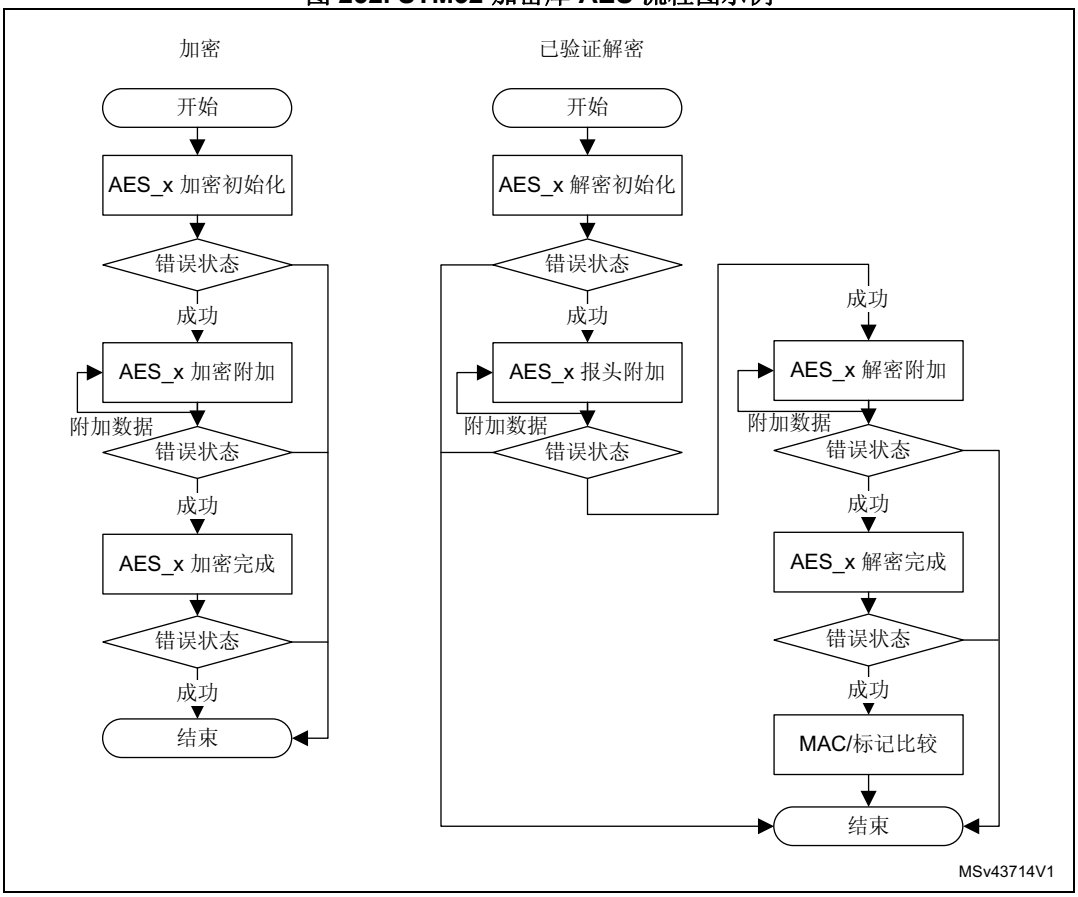


图 252. STM32 加密库 AES 流程图示例



CRYP初始化

1. 初始化加密处理器。除了 AES-ECB 和 AES-CBC 解密中的密钥准备过程需要遵循特定顺序以外，操作顺序并不重要。
   a) 将 CRYP_CR 寄存器中的 CRYPEN 位置 0，以禁止加密处理器。
   b) 使用 CRYP_CR 寄存器中的 KEYSIZE 位配置密钥大小（129 位、192 位或 256 位，仅限 AES）
   c) 将对称密钥写入 CRYP_KxL/R 寄存器（需写入 2 到 8 个寄存器，具体取决于算法）
   d) 使用 CRYP_CR 寄存器中的 DATATYPE 位配置数据类型（1 位、8 位、16 位或 32 位）
   e) 在 AES-ECB 或 AES-CBC 模式下解密时，准备已写入的密钥。先将 CRYP_CR 寄存器中的 ALGOMODE 位置为 0b111 来配置密钥准备模式。随后向 CRYPEN 位写入 1：BUSY 位随即自动置 1。等待 BUSY 位返回 0（CRYPEN 位也会自动清零）：已准备好密钥供解密使用
   f) 使用 CRYP_CR 寄存器中的 ALGOMODE 位配置算法和链接。
   g) 使用 CRYP_CR 寄存器中的 ALGODIR 位配置方向（加密/解密）。
   h) 必要时（例如，CBC 或 CTR 链接模式），向 CRYP_IVxL/R 寄存器中写入初始化向量。
2. 向 CRYP_CR 寄存器中的 FFLUSH 位写入 1，刷新 IN 和 OUT FIFO。

针对所有情况的初始警告

如果选择 ECB 或 CBC 模式，且数据不是 64 位（对于 DES）或 128 位（对于 AES）的倍数，则倒数第二个块管理要将比下述序列更为复杂。更多详细信息，请参见*第 35.3.8 节：CRYP 窃取和数据填充*。

在轮询模式下使用 CPU 附加数据

1. 将 CRYP_CR 寄存器中的 CRYPEN 位置 1，以使能加密处理器。
2. 向 IN FIFO 中写入数据（一个块或直到 FIFO 已满）。
3. 重复以下步骤，直到处理完数据的倒数第二个块：
   a) 等待至非空标志 OFNE 置 1，然后读取 OUT FIFO（一个块或直到 FIFO 为空）。
   b) 等待至非满标志 IFNF 置 1，然后写入 IN FIFO（一个块或直到 FIFO 已满），但是最后一个块的情况除外。
4. 加密处理器自动将 BUSY 位置 1。处理结束时，BUSY 位会返回至 0，两个 FIFO 均为空（IN FIFO 空标志 IFEM=1 且 OUT FIFO 非空标志 OFNE=0）。
5. 如果下一处理块为最后一个块，则 CPU 必须用零填充（如适用）数据，以获得完整块。
6. 操作完成后，可通过将 CRYP_CR 寄存器中的 CRYPEN 位清零来禁止加密处理器。

在中断模式下使用 CPU 附加数据

1. 将 CRYP_IMSCR 寄存器中的 INIM 和 OUTIM 位置 1，以使能中断。
2. 将 CRYP_CR 寄存器中的 CRYPEN 位置 1，以使能加密处理器。
3. 在管理输入数据的中断服务程序中：
   a) 如果正在加载的是最后一个块，则 CPU 必须用零填充（如适用）数据，以获得完整块。然后将该块加载到 IN FIFO 中。
   b) 如果不是最后一个块，则将数据加载到 IN FIFO 中。只能加载一个块（对于 DES 为 2 个字，对于 AES 为 4 个字），或加载数据直至 FIFO 已满。
   c) 在所有情况下，写入数据的最后一个字后，通过清除 INIM 中断屏蔽禁止中断。
4. 在管理输入数据的中断服务程序中：
   a) 从 OUT FIFO 中读取输出数据。只能读取一个块（对于 DES 为 2 个字，对于 AES 为 4 个字），或读取数据直至 FIFO 为空。
   b) 读取最后一个字后，INIM 和 BUSY 位均设为 0，两个 FIFO 均为空（IFEM=1 且 OFNE=0）。将 OUTIM 位清零可禁止中断，而将 CRYPEN 位清零可禁止外设。
   c) 如果读取明文数据的最后一个块（即解密），则可选择丢弃不属于消息/有效负载一部分的数据。

使用 DMA 附加数据

1. 可选择用零进行填充的方式来准备最后一个数据块，以获得完整块。
2. 将 DMA 控制器配置为传输来自存储器的输入数据以及将来自外设的输出数据传输到存储器，如*第 35.3.19 节：CRYP DMA 接口*所述。应将 DMA 配置为在完成传输时设置一个中断，以指示处理过程已完成。
3. 通过将 CRYP_CR 寄存器中的 CRYPEN 位置 1 使能加密处理器，然后通过将 CRYP_DMACR 寄存器中的 DIEN 和 DOEN 位置 1 来使能 DMA IN 和 OUT 请求。
4. 所有传输和处理过程均由 DMA 和加密处理器管理。DMA 中断表示处理过程已完成。两个 FIFO 通常均为空，且 BUSY 标志设为 0。

注意： 在填满加密处理器输入 FIFO 之前，由 DMA 控制器清空加密处理器输出 FIFO 至关重要。为此，应对 DMA 控制器进行相应配置，以使从加密外设到存储器的传输优先级高于从存储器到加密外设的传输。

35.3.6 CRYP 忙碌状态

满足以下所有条件时，即表示加密处理器繁忙，正在处理数据（CRYP_SR 寄存器中的 BUSY 设为 1）：

- CRYP_CR 寄存器中的 CRYPEN = 1。
- 输入 FIFO 中有足够的数据（对于 DES 或 TDES 算法模式至少有两个字，对于 AES 算法模式至少有四个字）。
- 输出 FIFO 中有足够的可用空间（对于 DES 至少有两个字单元，对于 AES 至少有四个字单元）。

当加密处理器繁忙时，会忽略针对 CRYP_Kx(L/R)R 密钥寄存器、CRYP_IVx(L/R)R 初始化寄存器或 CRYP_CR 寄存器的位 [9:2] 的写操作（即，不会修改这些寄存器）。因此，不能在加密处理器处理数据块时修改其配置。

在 BUSY 位置 1 时，可将 CRYPEN 位清零。在这种情况下，在硬件清零 BUSY 位之前，会先完成正在进行的 DES/TDES 或 AES 处理过程（即，字结果被写入输出 FIFO）。

注： *如果应用程序需要挂起一条消息，以处理另一条优先级更高的消息，请参见第 35.3.9 节：CRYP 挂起/恢复操作*

在 DES 或 TDES 模式下处理某个块时，如果输出 FIFO 已满并且输入 FIFO 至少含一个新块，则输入 FIFO 会弹出新块且 BUSY 位保持置 1，直到有足够的空间可将此新块存储到输出 FIFO。

35.3.7 为解密准备 CRYP AES 密钥

执行 AES **ECB 或 CBC 解密时**，必须准备 AES 密钥，即，在执行解密之前需要完整的加密密钥调度。换言之，必须将加密最后一轮中的密钥用作解密的第一轮密钥。

在除 AES ECB 或 CBC 解密外的所有 AES 模式下，均无需此准备过程。

如果应用软件以某种方式存储了为解密准备的初始密钥，则对于要用给定密钥解密的所有数据，密钥调度操作只能执行一次。

注： *密钥准备操作的延迟为 14、16 或 18 个时钟周期，具体取决于密钥大小（128、192 或 256 位）。*

按以下步骤执行 CRYP 密钥准备过程：

1. 将加密密钥写入 K0...K3 密钥寄存器。
2. 在 CRYP_CR 中将 ALGOMODE 位编程为 0b111。在 CRYPEN 设为 1 时写入该值会立即启动 AES 轮来准备密钥。CRYP_SR 寄存器中的 BUSY 位置 1。
3. 完成密钥处理之后，得到的密钥会复制回 K0...K3 密钥寄存器，并且 BUSY 位会清零。

注： *由于 CRYPEN 位域在密钥准备结束后由硬件复位，因此应用软件必须再次将其置 1 以进行下一次操作。*

35.3.8 CRYP 窃取和数据填充

在 **ECB** 或 **CBC** 模式下使用 DES 或 AES 算法来管理不是块大小（对于 DES 为 64 位，对于 AES 为 128 位）倍数的消息时，请使用 NIST *特别出版物 800-38A，块密码工作模式的建议：CBC 模式密文窃取的三种变型*中介绍的窃取技术。由于加密处理器未实现这类技术，因此**最后两个块**必须由应用程序以特殊方式处理。

注： *本参考手册中未对密文窃取技术进行介绍。*

类似地，在除 ECB 或 CBC 之外的模式下使用 AES 算法时，应用程序必须先用零填充不完整的输入数据块（即，短于 128 位的块）后再进行加密（即，应在数据串的尾端附加额外的位）。解密后需要丢弃额外填充的位。加密处理器不会对**最后一个块**执行自动数据填充操作，因此应用程序应遵循*第 35.3.5 节：用于执行密码操作的 CRYP 过程*中给出的建议来管理不是 128 位倍数的消息。

注： *填充数据根据 CRYP_CR 寄存器中的 DATATYPE 字段，以类似于正常数据的方式进行交换（有关详细信息，请参见第 35.3.16 节：CRYP 数据寄存器和数据交换）。*

对于该版本的加密处理器，当最后一个块的有效负载大小**低于 128 位**，需要采取特殊的解决方案以便在进行 **GCM 加密**或 **CCM 解密**时正确计算认证标记。下面对这套解决方案进行了介绍：

- 在 GCM 加密有效负载阶段且在插入小于 128 位的最后一个明文块之前，应用程序必须执行以下序列：
  a) 通过将 CRYP_CR 中的 CRYPEN 位置 0 来禁止外设。
  b) 将 CRYP_IV1R 寄存器内容加载到临时变量中。将值减去 1，然后将结果重新插入 CRYP_IV1R 寄存器中。
  c) 通过向 CRYP_CR 寄存器中的 ALGOMODE 位域写入 0b0110 来将 AES 模式切换为 CTR 模式。
  a) 将 CRYP_CR 中的 CRYPEN 位置 1，再次使能外设。
  b) 用零填充最后一个块（小于 128 位），以便获得 128 位的完整块，然后将其写入 CRYP_DIN 寄存器中。
  c) 加密完成后，从 CRYP_DOUT 寄存器中读取生成的 128 位密文，并将其存储为中间数据。
  d) 通过向 CRYP_CR 寄存器中的 ALGOMODE 位域写入 0b1000 来再次将 AES 模式切换为 GCM 模式。
  e) 通过向 CRYP_CR 寄存器中的 GCM_CCMPH 位域写入 0b11 来选择最后阶段。
  f) 在中间数据中，将最后有效负载块的填充位所对应的位置 0，然后将得到的数据插入 CRYP_DIN 寄存器中。
  g) 完成该操作后，从 CRYP_DOUT 中读取数据。必须将这些数据丢弃。
  h) 按*第 35.3.13 节：CRYP AES Galois/ 计数器模式 (GCM)*所述应用正常最后阶段。

- 在 CCM 解密有效负载阶段且在插入小于 128 位的最后一个密文块之前，应用程序必须执行以下序列：
  a) 通过将 CRYP_CR 中的 CRYPEN 位置 0 来禁止外设。
  b) 将 CRYP_IV1R 加载到临时变量（此处名为 `Iv1temp`）中。
  c) 将 CRYP_CSGCMCCM0R、CRYP_CSGCMCCM1R、CRYP_CSGCMCCM2R 和 CRYP_CSGCMCCM3R 寄存器的内容按照从 LSB 到 MSB 的顺序加载到 128 位临时变量（此处名为 `temp1`）中。
  d) 将之前存储在 `IV1temp` 中的内容加载到 CRYP_IV1R 中。
  e) 通过向 CRYP_CR 寄存器中的 ALGOMODE 位域写入 0b0110 来将 AES 模式切换为 CTR 模式。
  a) 将 CRYP_CR 中的 CRYPEN 位置 1，再次使能外设。
  b) 用零填充最后一个块（小于 128 位），以便获得 128 位的完整块，然后将其写入 CRYP_DIN 寄存器中。
  c) 解密完成后，从 DOUT 寄存器中读取生成的 128 位数据，并将其存储为中间数据（此处名为 `intdata_o`）。
  d) 将 CRYP_CSGCMCCM0R、CRYP_CSGCMCCM1R、CRYP_CSGCMCCM2R 和 CRYP_CSGCMCCM3R 寄存器的内容按照从 LSB 到 MSB 的顺序再次加载到新的 128 位临时变量（此处名为 `temp2`）中。
  e) 通过向 CRYP_CR 寄存器中的 ALGOMODE 位域写入 0b1001 来再次将 AES 模式切换为 CCM 模式。
  f) 通过向 CRYP_CR 寄存器中的 GCM_CCMPH 位域写入 0b01 来选择标头阶段。
  g) 在中间数据（`intdata_o`，在 CTR 模式下生成）中，将最后一个有效负载块的填充位所对应的位置 0，与 `temp1` 进行异或运算，与 `temp2` 进行异或运算，并将得到的数据插入 CRYP_DIN 寄存器中。即：

     ```
     CRYP_DIN=(intdata_o AND mask) XOR temp1 XOR temp2。
     ```

  h) 等待操作完成。
  i) 按*第 35.3.15 节：CRYP AES CBC-MAC 计数器模式 (CCM)* 所述应用正常最后阶段。

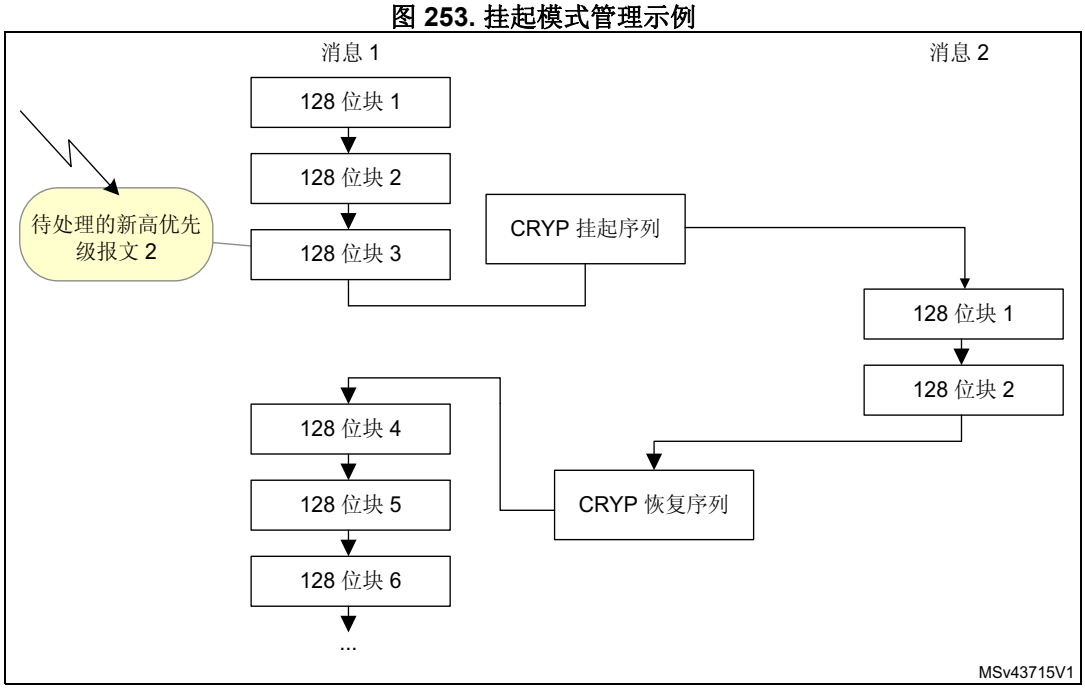
35.3.9 CRYP 挂起/恢复操作

如果需要处理另一条优先级较高的消息，则可将消息挂起。该最高优先级消息完成发送后，可在加密或解密模式下恢复挂起的消息。

挂起/恢复操作不会破坏链接运算，并且再次使能加密处理器以接收下一个数据块时，可立即恢复消息处理。

图 253 所示为挂起/恢复操作示例：为发送优先级更高、长度更短的消息 2 而将消息 1 挂起（AES 算法）。

图 253. 挂起模式管理示例



各 AES 模式部分对挂起/恢复操作进行了详细说明。

35.3.10 CRYP DES/TDES 基本链接模式（ECB 和 CBC）

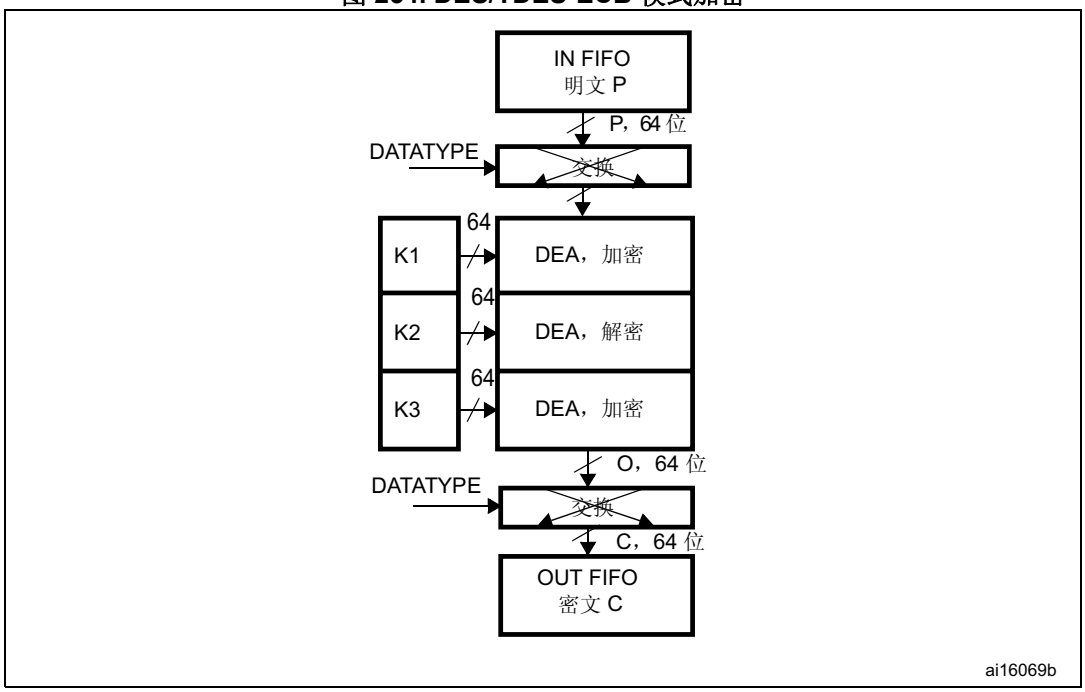
概述

FIPS PUB 46-3 – 1999（以及 ANSI X9.52-1998）对 DES 计算内核所提供的四种工作模式中涉及的处理过程进行了详尽的解释，这四种模式分别为：TDES-ECB 加密、TDES-ECB 解密、TDES-CBC 加密和 TDES-CBC 解密。本部分仅对每种模式进行了简要解释。

DES/TDES-ECB 加密

图 254 介绍了 DES 和 TDES 电子密码本 (DES/TDES-ECB) 模式中的加密。通过向 CRYP_CR 中的 ALGOMODE 和 ALGODIR 分别写入 0b000 和 0 来选择该模式。

图 254. DES/TDES-ECB 模式加密



1. K：密钥；C：密文；I：输入块；O：输出块；P：明文。

64 位明文数据块 (P) 经过位/字节/半字交换后用作输入块 (I)。输入块通过 DEA 在加密状态下使用 K1 进行加密处理。上述处理过程的输出会直接反馈到 DEA 的输入，在解密状态下使用 K2 执行 DES。上述处理过程的输出会直接反馈到 DEA 的输入，在加密状态下使用 K3 执行 DES。生成的 64 位输出块 (O) 在执行位/字节/半字交换之后，以密文 (C) 形式推入 OUT FIFO。

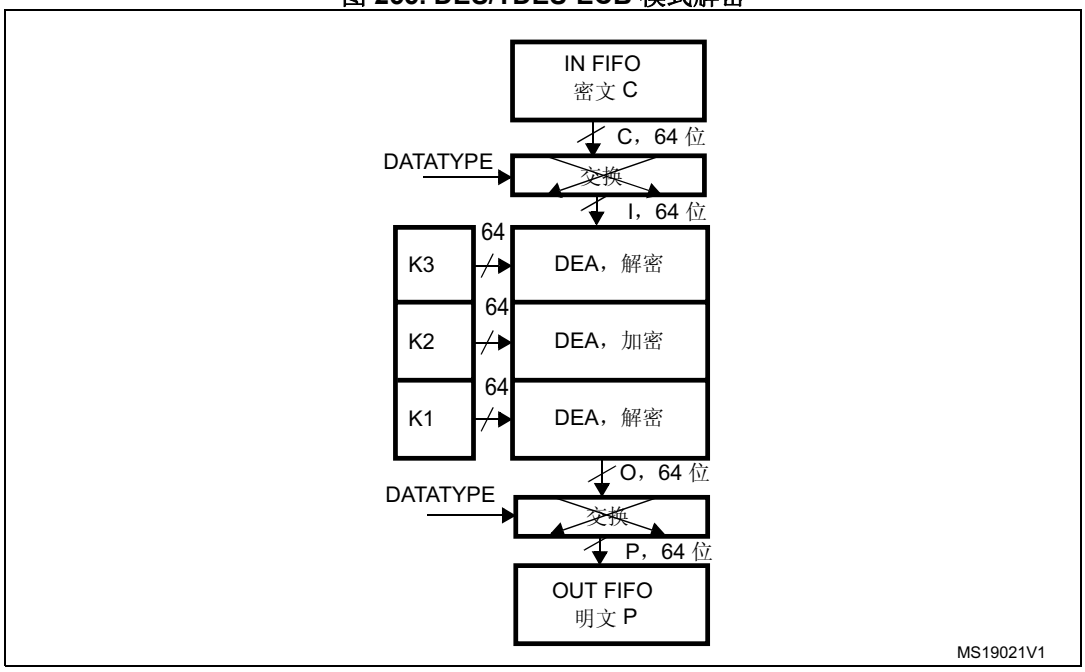
注： *有关数据交换的更多信息，请参见第 35.3.16 节：CRYP 数据寄存器和数据交换。*

有关详细的 DES/TDES 加密序列，请参见第 35.3.5 节：用于执行密码操作的 CRYP 过程。

DES/TDES-ECB 模式解密

图 255 介绍了 DES 和 TDES 电子密码本 (DES/TDES-ECB) 模式中的解密。通过向 CRYP_CR 中的 ALGOMODE 和 ALGODIR 分别写入 0b000 和 1 来选择该模式。

图 255. DES/TDES-ECB 模式解密



1. K：密钥；C：密文；I：输入块；O：输出块；P：明文。

64 位官方块 (C) 经过位/字节/半字交换后，作为输入块 (I)。该密钥序列与加密过程中使用的密钥序列相反。输入块通过 DEA 在解密状态下使用 K3 进行解密处理。上述处理过程的输出会直接反馈到 DEA 的输入，在加密状态下使用 K2 执行 DES。新结果会直接反馈到 DEA 的输入，在解密状态下使用 K1 执行 DES。生成的 64 位输出块 (O) 在进行位/字节/半字交换后，产生明文 (P)。

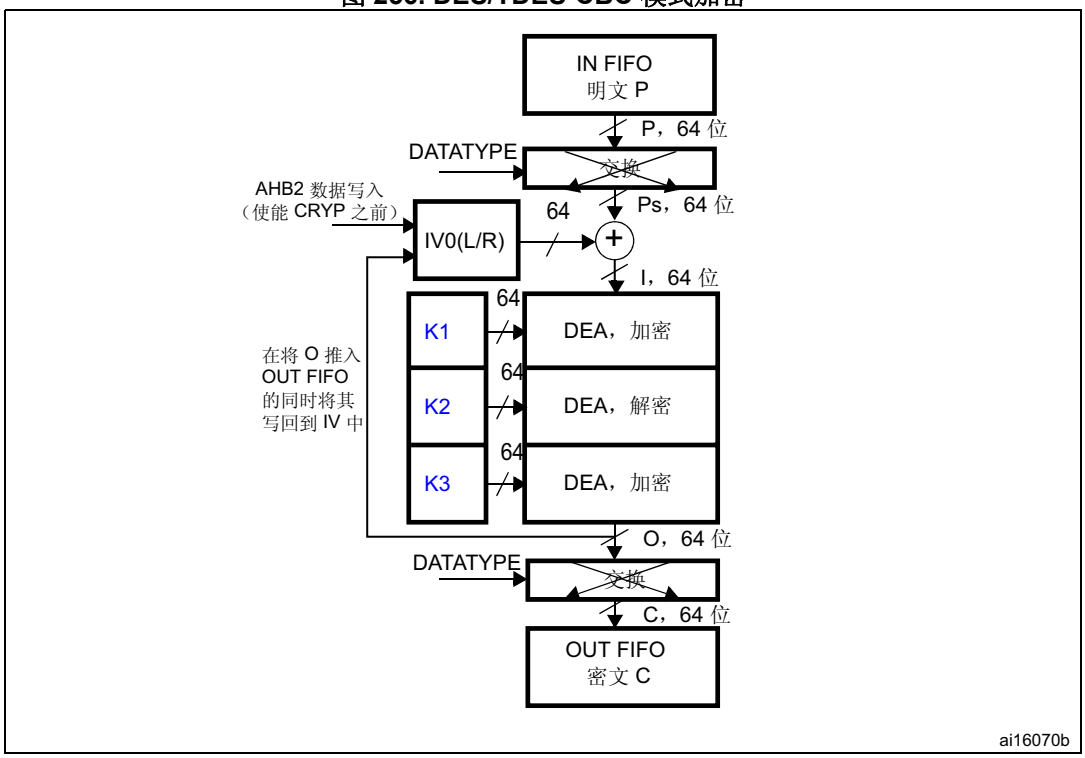
注： *有关数据交换的更多信息，请参见第 35.3.16 节：CRYP 数据寄存器和数据交换。*

有关详细的 DES/TDES 加密序列，请参见第 35.3.5 节：用于执行密码操作的 CRYP 过程。

DES/TDES-CBC 加密

图 256 介绍了 DES 和 TDES 密码块链接 (DES/TDES-ECB) 模式中的加密。通过向 CRYP_CR 中的 ALGOMODE 和 ALGODIR 分别写入 0b001 和 0 来选择该模式。

图 256. DES/TDES-CBC 模式加密



K：密钥；C：密文；I：输入块；O：输出块；Ps：交换前（解码时）或交换后（编码时）的明文；P：明文；IV：初始化向量。

该模式首先将明文消息分成多个 64 位数据块。在 TCBC 加密中，执行位/字节/半字交换后获得的第一个输入块 (I_1) 是通过一个 64 位初始化向量 IV 与第一个明文数据块 (P_1) 进行异或运算形成的 ($I_1 = IV \oplus P_1$)。输入块通过 DEA 在加密状态下使用 K1 进行加密处理。上述处理过程的输出会直接反馈到 DEA 的输入，在解密状态下使用 K2 执行 DES。上述处理过程的输出会直接反馈到 DEA 的输入，在加密状态下使用 K3 执行 DES。生成的 64 位输出块 (O_1) 将直接用作密文 (C_1)，即 $C_1 = O_1$。

然后，第一个密文块与第二个明文数据块进行异或运算，从而生成第二个输入块 (I_2) = ($C_1 \oplus$ P2)。注意，此时的 I_2 和 P_2 指的是第二个块。第二个输入块通过 TDEA 处理而生成第二个密文块。

此加密处理会不断将后续密文块和明文块链接到一起，直到消息中最后一个明文块得到加密为止。

如果消息中包含的数据块数不是整数，则应按照应用程序指定的方式对最后的不完整数据块进行加密。

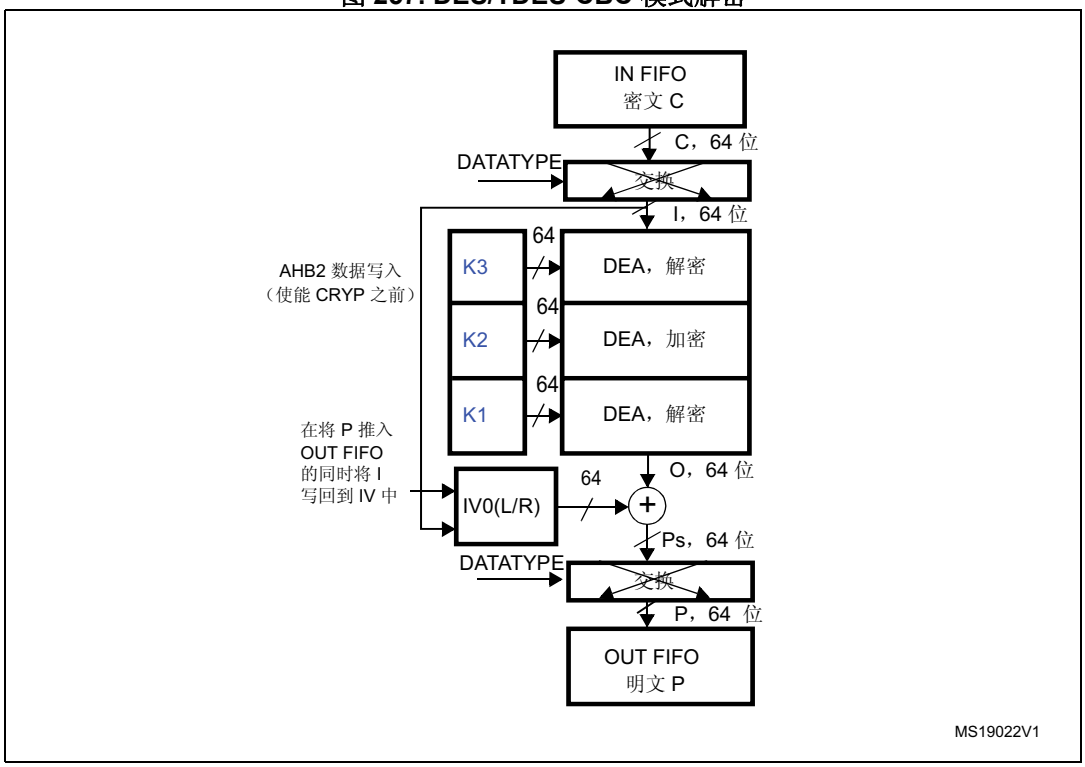
注： *有关数据交换的更多信息，请参见第 35.3.16 节：CRYP 数据寄存器和数据交换。*

有关详细的 DES/TDES 加密序列，请参见第 35.3.5 节：用于执行密码操作的 CRYP 过程。

DES/TDES-CBC 解密

图 256 介绍了 DES 和 TDES 密码块链接 (DES/TDES-ECB) 模式中的解密。通过向 CRYP_CR 中的 ALGOMODE 和 ALGODIR 分别写入 0b001 和 1 来选择该模式。

图 257. DES/TDES-CBC 模式解密



1. K：密钥；C：密文；I：输入块；O：输出块；Ps：交换前（解码时）或交换后（编码时）的明文；P：明文；IV：初始化向量。

在该模式下，第一个密文块 (C_1) 将直接用作输入块 (I_1)。该密钥序列与加密过程中使用的密钥序列相反。输入块通过 DEA 在解密状态下使用 K3 进行解密处理。上述处理过程的输出会直接反馈到 DEA 的输入，在加密状态下使用 K2 执行 DES。生成值会直接反馈到 DEA 的输入，在解密状态下使用 K1 处理 DES。生成的输出块与 IV（必须与加密期间使用的相同）进行异或运算，从而生成第一个明文块 ($P_1 = O_1 \oplus IV$)。

然后，第二个密文块将用作下一个输入块，并由 TDEA 进行处理。生成的输出块与第一个密文块进行异或运算，从而生成第二个明文数据块 ($P_2 = O_2 \oplus C1$)。（注意，P2 和 O2 指的是第二个数据块。）

DES/TDES-CBC 解密过程将以此方式继续进行，直到最后一个完整密文块得到解密为止。

必须按照应用程序指定的方式对不完整数据块密文进行解密。

注： *有关数据交换的更多信息，请参见第 35.3.16 节：CRYP 数据寄存器和数据交换。*

有关详细的 DES/TDES 加密序列，请参见第 35.3.5 节：用于执行密码操作的 CRYP 过程。

ECB/CBC 模式下的 DES/TDES 挂起/恢复操作

在中断当前消息之前，用户应用程序必须遵循以下步骤：

1. 如果使用 DMA，则通过将 CRYP_DMACR 寄存器中的 DIEN 位清零来停止 DMA 对 IN FIFO 的数据传输。
2. 等待 IN 和 OUT FIFO 均为空（CRYP_SR 中的 IFEM=1 且 OFNE=0），并且 BUSY 位清零。或者，由于输入 FIFO 最多可包含四个未处理的 DES 块，因此应用程序可根据实际情况决定在不等待 IN FIFO 变空的情况下中断加密处理过程。此时，还可以：
   a) 等待至 OUT FIFO 为空 (OFNE=0)。
   b) 读回 IN FIFO 中加载的尚未处理的数据，并将其保存到存储器中，直到 IN FIFO 为空。
3. 如果使用 DMA，则通过将 CRYP_DMACR 寄存器中的 DOEN 位清零来停止 DMA 对 OUT FIFO 的数据传输。
4. 通过将 CRYP_CR 中的 CRYPEN 位置 0 来禁止加密处理器，然后保存当前配置（CRYP_CR 寄存器中的位 [9:2]）。如果选择 CBC 模式，则保存初始化向量寄存器，因为 CRYP_IVx 寄存器在数据处理过程中已发生更改，不再是初始值。

注： *无需保存密钥寄存器，因为应用程序已知初始密钥值。*

5. 如果使用 DMA，则保存 DMA 控制器状态（例如，指向 IN 和 OUT 数据传输的指针以及剩余字节数）。

要恢复消息处理，用户应用程序必须遵循以下序列：

1. 如果使用 DMA，则重新配置 DMA 控制器以完成 FIFO IN 和 FIFO OUT 传输的剩余部分。
2. 确保通过读取 CRYP_CR 中的 CRYPEN 位（必须为 0）来禁止加密处理器。
3. 使用 CRYP_CR 中的初始设置再次配置加密处理器，以及使用所保存的配置再次配置密钥寄存器。
4. 如果选择 CBC 模式，则使用保存的配置恢复 CRYP_IVx 寄存器。
5. 可选择将上下文保存期间保存的数据写入 IN FIFO。
6. 通过将 CRYPEN 位置 1 来使能加密处理器。
7. 如果使用 DMA，则通过将 CRYP_DMACR 寄存器中的 DIEN 和 DOEN 位置 1 再次使能对加密处理器的 DMA 请求。

35.3.11 CRYP AES 基本链接模式（ECB 和 CBC）

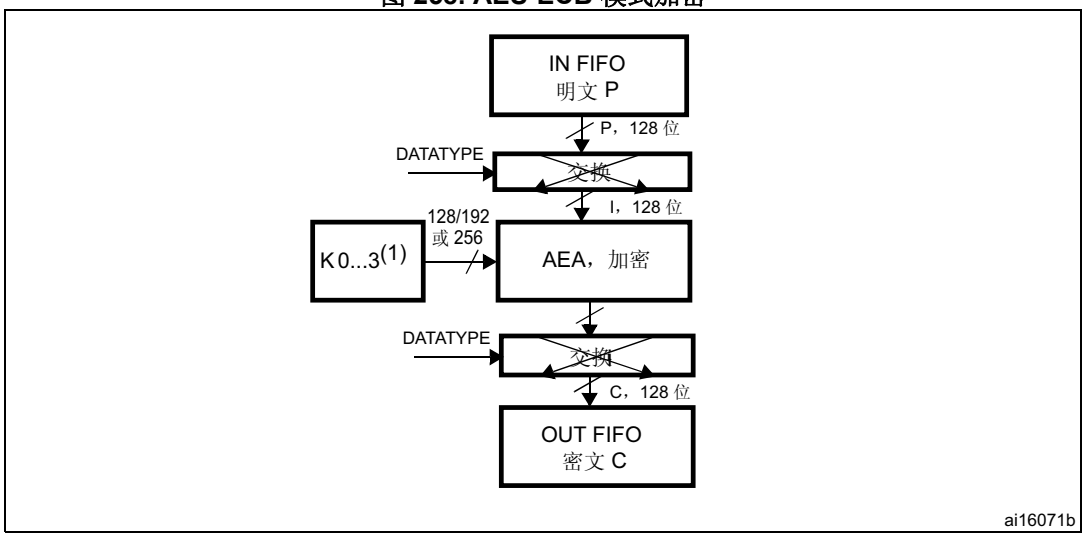
概述

FIPS PUB 197（2001 年 11 月 26 日）对 AES 计算内核所提供的四种基本工作模式中涉及的处理过程进行了详尽的解释，这四种模式分别为：AES-ECB 加密、AES-ECB 解密、AES-CBC 加密和 AES-CBC 解密。本部分仅对每种模式进行了简要解释。

AES ECB 加密

图 258 介绍了 AES 电子密码本 (AES-ECB) 模式加密。通过向 CRYP_CR 中的 ALGOMODE 和 ALGODIR 分别写入 0b100 和 0 来选择该模式。

图 258. AES-ECB 模式加密



1. K：密钥；C：密文；I：输入块；O：输出块；P：明文。
2. 如果密钥大小 = 128：则密钥 = [K3 K2]。
   如果密钥大小 = 192：则密钥 = [K3 K2 K1]。
   如果密钥大小 = 256：则密钥 = [K3 K2 K1 K0]。

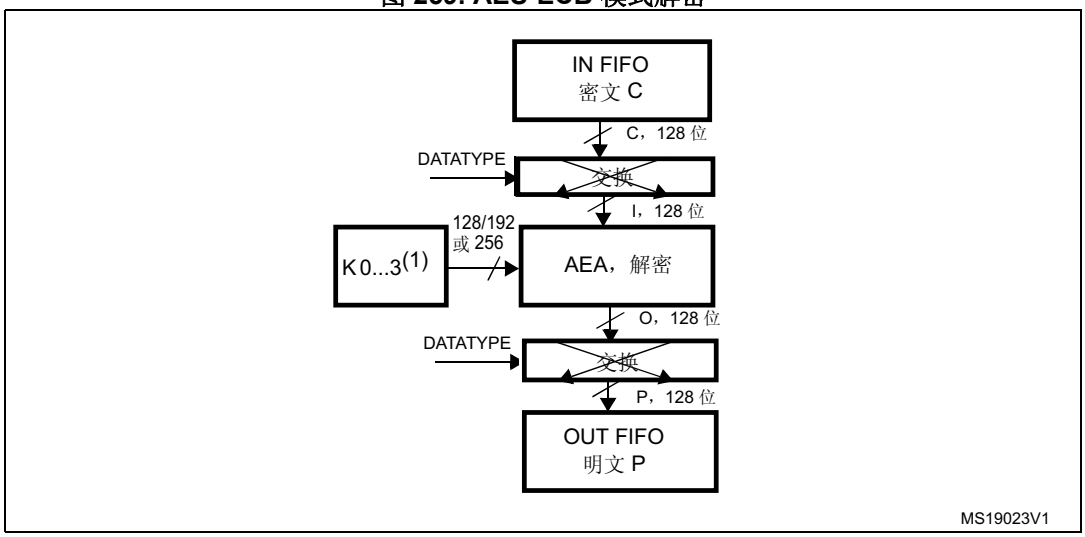
在该模式下，128 位明文数据块 (P) 经过位/字节/半字交换后用作输入块 (I)。输入块通过 AEA 在加密状态下使用 128 位、192 位或 256 位密钥进行处理。执行位/字节/半字交换后，生成的 128 位输出块 (O) 将用作密文 (C)。该密文随后推入 OUT FIFO。

有关数据交换的更多信息，请参见*第 35.3.16 节：CRYP 数据寄存器和数据交换*。

AES ECB 解密

图 259 介绍了 AES 电子密码本 (AES-ECB) 模式解密。通过向 CRYP_CR 中的 ALGOMODE 和 ALGODIR 分别写入 0b100 和 1 来选择该模式。

图 259. AES-ECB 模式解密



1. K：密钥；C：密文；I：输入块；O：输出块；P：明文。
2. 如果密钥大小 = 128 => 密钥 = [K3 K2]。
   如果密钥大小 = 192 => 密钥 = [K3 K2 K1]。
   如果密钥大小 = 256 => 密钥 = [K3 K2 K1 K0]。

要在 ECB 模式下执行 AES 解密，需要准备密钥（需要针对加密执行完整的密钥计划），方法为：收集最后一个轮密钥并将其用作解密密文的第一个轮密钥。该准备阶段通过 AES 内核计算完成。有关如何准备密钥的详细信息，请参见*第 35.3.7 节：为解密准备 CRYP AES 密钥*。

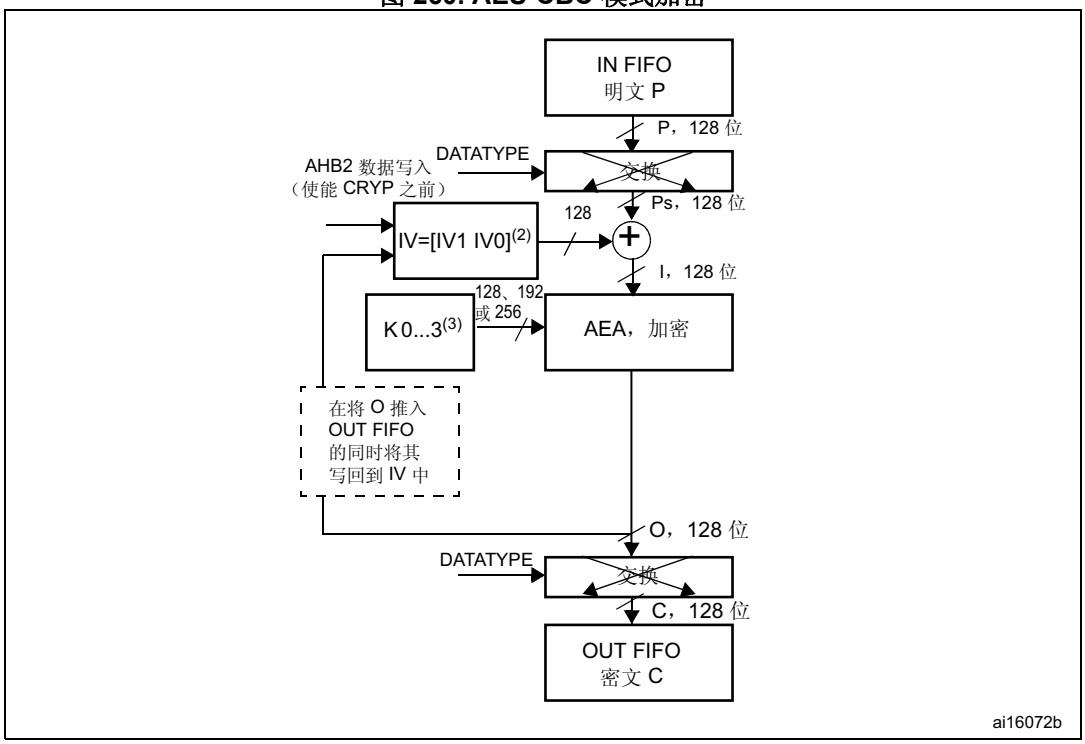
完成密钥准备后，按以下所述执行解密：对 128 位密文块 (C) 执行位/字节/半字交换后将其用作输入块 (I)。该密钥序列与加密处理中的密钥序列相反。执行位/字节/半字交换，生成的 128 位输出块 (O) 将产生明文 (P)。AES-CBC 解密过程将以此方式继续进行，直到最后一个完整密文块得到解密为止。

有关数据交换的更多信息，请参见*第 35.3.16 节：CRYP 数据寄存器和数据交换*。

AES CBC 加密

图 260 所示为 AES 密码分组链接 (AES-CBC) 模式加密。通过向 CRYP_CR 中的 ALGOMODE 和 ALGODIR 分别写入 0b101 和 0 来选择该模式。

图 260. AES-CBC 模式加密



1. K：密钥；C：密文；I：输入块；O：输出块；Ps：交换前（解码时）或交换后（编码时）的明文；P：明文；IV：初始化向量。
2. IVx = [IVxR IVxL]，R = 右，L = 左。
3. 如果密钥大小 = 128 => 密钥 = [K3 K2]。
   如果密钥大小 = 192 => 密钥 = [K3 K2 K1]。
   如果密钥大小 = 256 => 密钥 = [K3 K2 K1 K0]。

在该模式下，执行位/字节/半字交换后获得的第一个输入块 (I_1) 是通过一个 128 位初始化向量 IV 与第一个明文数据块 (P_1) 进行异或运算形成的 (I_1 = IV ⊕ P_1)。输入块通过 AEA 在加密状态下使用 128 位、192 位或 256 位密钥 (K0...K3) 进行处理。生成的 128 位输出块 (O_1) 将直接用作密文 (C_1)，即 C_1 = O_1。然后，第一个密文块与第二个明文数据块进行异或运算，从而生成第二个输入块 (I_2) = (C_1 ⊕ P2)。注意，此时的 I_2 和 P_2 指的是第二个块。第二个输入块通过 AEA 处理而生成第二个密文块。此加密处理会不断将后续密文块和明文块链接到一起，直到消息中最后一个明文块得到加密为止。

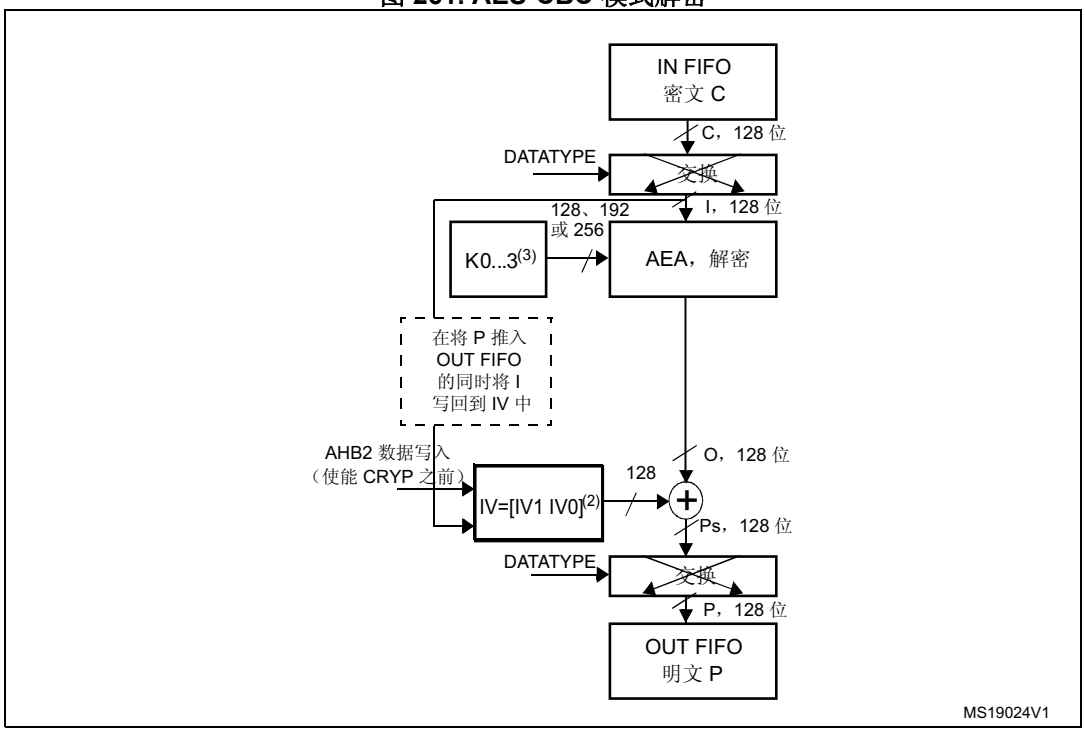
如果消息中包含的数据块数不是整数，则应按照应用程序指定的方式对最后的不完整数据块进行加密，如*第 35.3.8 节：CRYP 窃取和数据填充*所述。

有关数据交换的更多信息，请参见*第 35.3.16 节：CRYP 数据寄存器和数据交换*。

AES CBC 加密

图 261 所示为 AES 密码块链接 (AES-CBC) 模式解密。通过向 CRYP_CR 中的 ALGOMODE 和 ALGODIR 分别写入 0b101 和 1 来选择该模式。

图 261. AES-CBC 模式解密



1. K：密钥；C：密文；I：输入块；O：输出块；Ps：交换前（解码时）或交换后（编码时）的明文；P：明文；IV：初始化向量。
2. IVx = [IVxR IVxL]，R = 右，L = 左。
3. 如果密钥大小 = 128 => 密钥 = [K3 K2]。
   如果密钥大小 = 192 => 密钥 = [K3 K2 K1]。
   如果密钥大小 = 256 => 密钥 = [K3 K2 K1 K0]。

在 CBC 模式下（如 ECB 模式），必须准备密钥才可以执行 AES 解密。有关如何准备密钥的详细信息，请参见*第 35.3.7 节：为解密准备 CRYP AES 密钥*。

完成密钥准备过程后，按以下所述执行解密：直接将第一个 128 位密文块 (C_1) 用作输入块 (I_1)。输入块通过 AEA 在解密状态下使用 128 位、192 位或 256 位密钥进行处理。生成的输出块与 128 位初始化向量 IV（必须与加密期间使用的相同）进行异或运算，从而生成第一个明文块 ($P_1 = O_1 \oplus IV$)。

然后，第二个密文块将用作下一个输入块，并由 AEA 进行处理。生成的输出块与第一个密文块进行异或运算，从而生成第二个明文数据块 ($P_2 = O_2 \oplus C_1$)。（注意，P2 和 O2 指的是第二个数据块。） AES-CBC 解密过程将以此方式继续进行，直到最后一个完整密文块得到解密为止。

必须按照应用程序指定的方式对不完整数据块密文进行解密，如*第 35.3.8 节：CRYP 窃取和数据填充*所述。

有关数据交换的更多信息，请参见*第 35.3.16 节：CRYP 数据寄存器和数据交换*。

ECB/CBC 模式下的 AES 挂起/恢复操作

在中断当前消息之前，用户应用程序必须遵循以下序列：

1. 如果使用 DMA，则通过将 CRYP_DMACR 寄存器中的 DIEN 位清零来停止 DMA 对 IN FIFO 的数据传输。
2. 等待 IN 和 OUT FIFO 均为空（CRYP_SR 中的 IFEM=1 且 OFNE=0），并且 BUSY 位清零。
3. 如果使用 DMA，则通过将 CRYP_DMACR 寄存器中的 DOEN 位清零来停止 DMA 对 OUT FIFO 的数据传输。
4. 通过将 CRYP_CR 中的 CRYPEN 位置 0 来禁止 CRYP，然后保存当前配置（CRYP_CR 寄存器中的位 [9:2]）。如果未选择 ECB 模式，则保存初始化向量寄存器，因为 CRYP_IVx 寄存器在数据处理过程中已发生更改，不再是初始值。

注： *无需保存密钥寄存器，因为应用程序已知初始密钥值。*

5. 如果使用 DMA，则保存 DMA 控制器状态（例如，指向 IN 和 OUT 数据传输的指针以及剩余字节数）。

要恢复消息处理，用户应用程序必须遵循以下序列：

1. 如果使用 DMA，则重新配置 DMA 控制器以完成 FIFO IN 和 FIFO OUT 传输的剩余部分。
2. 确保通过读取 CRYP_CR 中的 CRYPEN 位（必须设为 0）来禁止加密处理器。
3. 使用 CRYP_CR 中的初始设置再次配置加密处理器，以及使用所保存的配置再次配置密钥寄存器。
4. 针对 AES-ECB 或 AES-CBC 解密，必须再次准备密钥，如*第 35.3.7 节：为解密准备 CRYP AES 密钥*所述。
5. 如果未选择 ECB 模式，则使用保存的配置恢复 CRYP_IVx 寄存器。
6. 通过将 CRYPEN 位置 1 来使能加密处理器。
7. 如果使用 DMA，通过将 CRYP_DMACR 寄存器中的 DIEN 和 DOEN 位置 1 再次使能来自加密处理器的 DMA 请求。

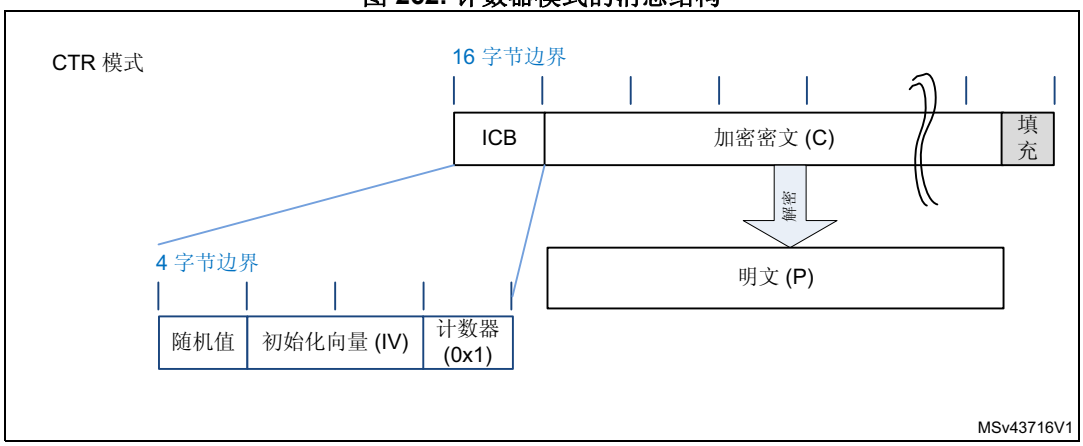
35.3.12 CRYP AES 计数器模式 (AES-CTR)

概述

AES 计数器模式 (CTR) 使用 AES 块作为密钥流生成器。然后，生成的密钥与明文进行异或运算来获得密文。

NIST *特别出版物 800-38A，块密码工作模式的建议*对 CTR 链接进行了相关定义。*图 262* 给出了 CTR 模式下的典型消息结构。

图 262. 计数器模式的消息结构



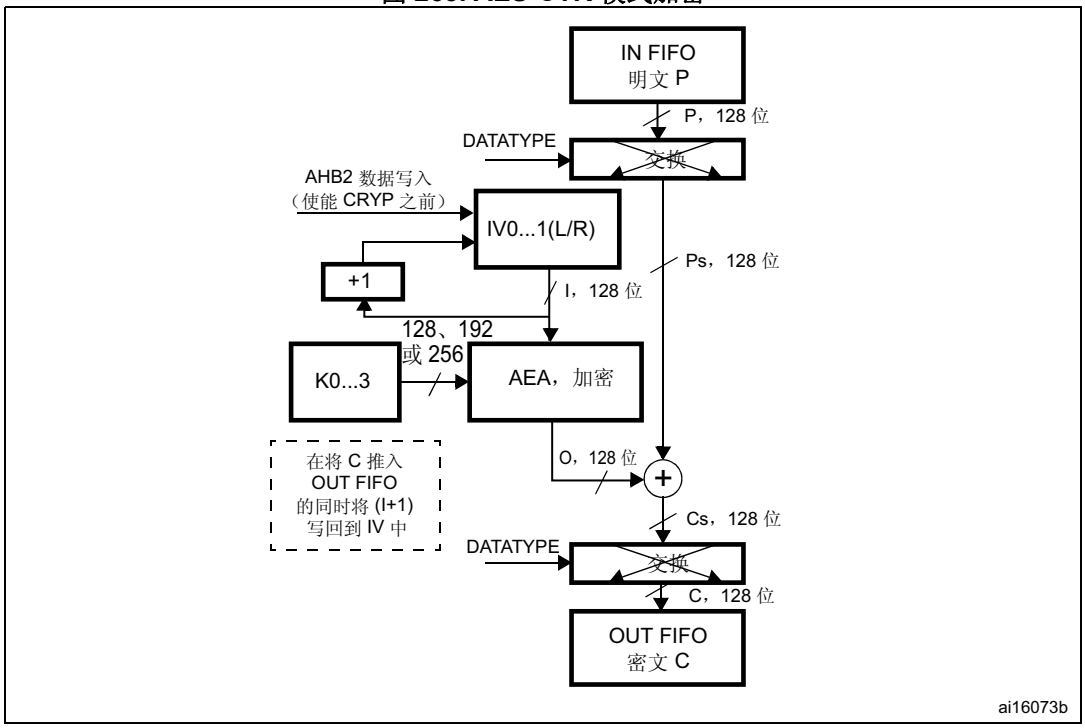
该消息的结构如下：

- 16 字节的初始计数器块 (ICB)，它由三个不同的字段组成：
  - *随机值*：32 位一次性值（即，应为每一次新的通信分配新的随机值）。
  - *初始化向量* (IV)：64 位值，对于给定密钥模式的每次执行，该值都必须唯一。
  - *计数器*：32 位大端模式的整数，每当处理完一个块，该值递增。应将计数器的初始值设为 1。
- 明文 (P) 将经过认证并被加密为密文 C（长度已知）。该长度可以不是 16 字节的倍数，在这种情况下需要明文填充。

AES CTR 处理

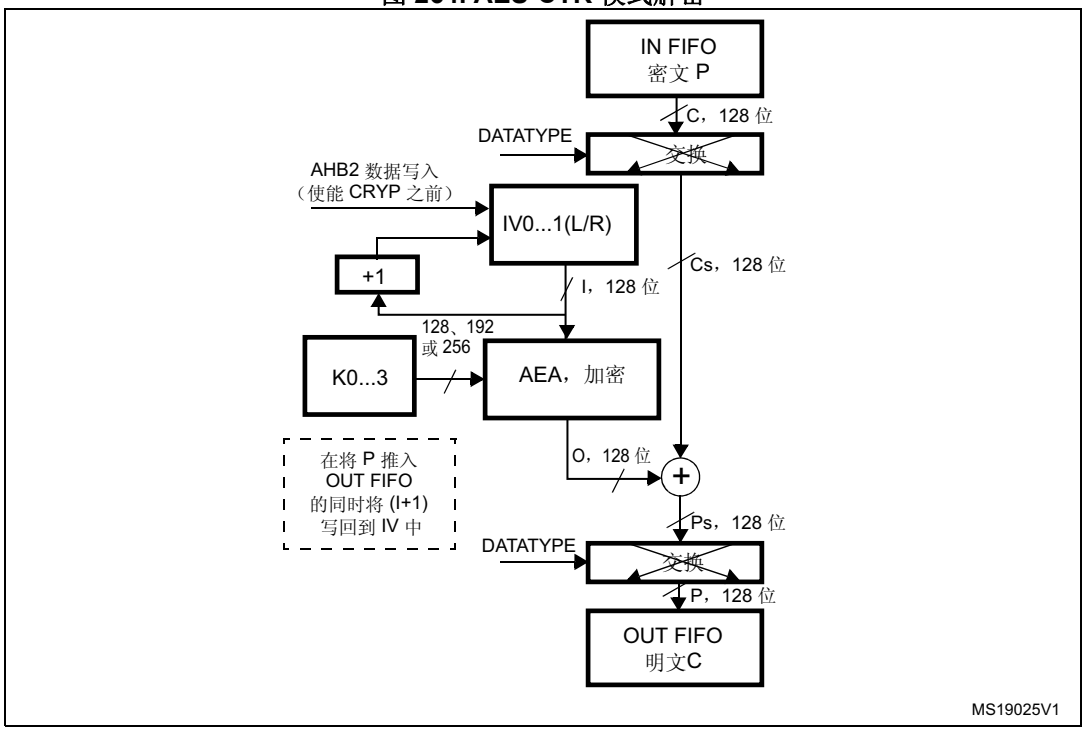
图 263（*图 264*）介绍了该外设中实现的 AES-CTR 加密（解密）过程。通过向 CRYP_CR 中的 ALGOMODE 位域写入 0b110 来选择该模式。

图 263. AES-CTR 模式加密



1. K：密钥；C：密文；I：输入块；O：输出块；Ps：交换前（解码时）或交换后（编码时）的明文；Cs：交换后（解码时）或交换前（编码时）的密文；P：明文；IV：初始化向量。

图 264. AES-CTR 模式解密



1. K：密钥；C：密文；I：输入块；O：输出块；Ps：交换前（解码时）或交换后（编码时）的明文；Cs：交换后（解码时）或交换前（编码时）的密文；P：明文；IV：初始化向量。

在 CTR 模式下，输出块会与后续输入块先进行异或运算，然后再输入到相应算法中。外设中的初始化向量必须按*表 263* 所示进行初始化。

表 263. 计数器模式初始化向量

CRYP_IV1R[31:0]	CRYP_IV1L[31:0]	CRYP_IV0R[31:0]	CRYP_IV0L[31:0]
随机值	IV[63:32]	IV[31:0]	32 位计数器 = 0x1

在 CBC 模式下，处理第一个数据块时仅使用 CRYP_IVx 寄存器一次，而在 CTR 模式下则有所不同，处理每个数据块都使用 IV 寄存器，并且外设会使 32 个最低有效位递增（其他 96 个最高有效位保持不变）。

CTR 解密与 CTR 加密并无不同，因为内核始终会对当前的计数器块加密以产生密钥数据流，该密钥数据流与输入中的明文或密码进行异或运算。因此，将 ALGOMODE 设为 0b110 时，ALGODIR 无关紧要。

注： *在该模式下，不得为解密准备密钥。*

必须使用以下序列在 CTR 链接模式下执行加密或解密：

1. 确保通过将 CRYP_CR 寄存器中的 CRYPEN 位清零来禁止加密处理器。
2. 按如下所述配置 CRYP_CR：
   a) 将 ALGOMODE 位编程为 0b110 来选择 CTR 模式。
   b) 通过 DATATYPE 位配置数据类型（1 位、8 位、16 位或 32 位）。
3. 按*表 263* 所述初始化 CRYP_KEYRx 中的密钥寄存器（128 位、192 位和 256 位）以及初始化向量 (IV)。
4. 向 CRYP_CR 寄存器中的 FFLUSH 位写入 1，刷新 IN 和 OUT FIFO
5. 如果是最后一个块，则可选择用零填充数据以获得完整的块。
6. 在加密处理器中附加数据并读取结果。*第 35.3.5 节：用于执行密码操作的 CRYP 过程*介绍了三种可能的情况。
7. 重复上一步，直到处理完倒数第二个块。对于最后一个块，执行前两步。对于这最后一个块，当其大小小于 16 字节时，驱动程序必须将不属于数据的一部分的位丢弃。

CTR 模式下的挂起/恢复操作

该模式与 CBC 模式相似，可以中断消息、发送优先级更高的消息，并可恢复已中断的消息。有关详细的 CBC 序列，请参见*第 35.3.11 节：CRYP AES 基本链接模式（ECB 和 CBC）*。

注： *与 CBC 模式类似，恢复操作期间必须重新加载 IV 寄存器。*

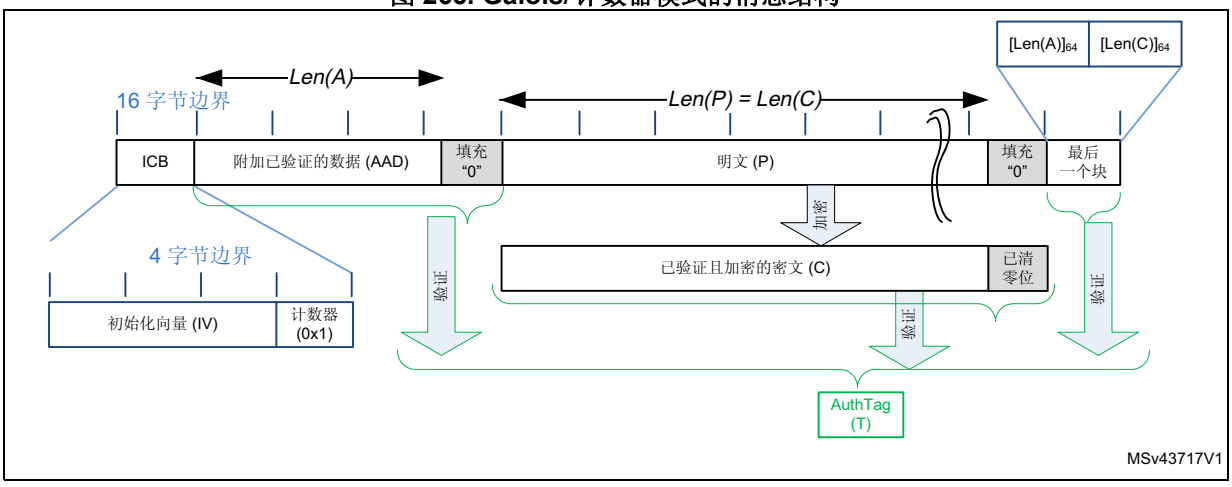
35.3.13 CRYP AES Galois/计数器模式 (GCM)

概述

AES Galois/计数器模式 (GCM) 允许加密和验证明文，以及生成对应的密文和标记（也称为消息认证码）。为确保保密性，GCM 算法基于 AES 计数器模式。它使用固定有限域乘法器来生成标记。

NIST *特别出版物 800-38D，块密码工作模式的建议 - Galois/计数器模式 (GCM) 和 GMAC* 中对 GCM 链接进行了相关定义。*图 265* 所示为 GCM 模式下的典型消息结构。

图 265. Galois/计数器模式的消息结构



该消息结构的定义如下：

- 16 字节的初始计数器块 (ICB)，它由两个不同的字段组成：
  - *初始化向量* (IV)：96 位值，对于给定密钥模式的条件下，每次执行，该值都必须唯一。请注意，GCM 标准支持短于 96 位的 IV，在这种情况下应遵循严格的规则。
  - *计数器*：32 位大端模式的整数，每当一个块被处理，此值递增。根据 NIST 规范，处理有效负载的第一个块时，计数器值为 0x2。
- 认证的标头 A（也称为附加认证数据）具有已知长度 *Len(A)*，此长度值可以不是 16 字节的倍数，但不能超过 2^{64}-1 位。消息的这一部分仅经过认证，未被加密。
- 明文消息 (P) 将经过认证且被加密为密文 C，其具有已知长度 *Len(P)*，该长度值可以不是 16 字节的倍数，但不能超过 2^{32} -2 个 128 位块。

注： *GCM 标准规定密文 C 的位长度与明文 P 的位长度相同。*

- 当消息某一部分（AAD 或 P）的长度不是 16 字节的倍数时，需要采取特殊填充方案。
- 最后一个块由 A 的长度（64 位）和密文 C 的长度（64 位）组成，如*表 264* 所示。

表 264. GCM 最后一个块定义

字节序	位 0	位 32	位 64	位 96
输入数据	0x0	标头长度 [31:0]	0x0	有效负载长度 [31:0]

AES GCM 处理

通过向 CRYP_CR 中的 ALGOMODE 位域写入 0b110 来选择该模式。

GCM 模式下明文的保密性机制是计数器模式的变型，其具有特定 32 位递增函数，可生成必要的计数器块序列。

CRYP_IV 寄存器用于处理每个数据块。加密处理器自动递增计数器块的 32 个最低有效位。应用程序写入的第一个计数器块 (CB1) 等于初始计数器块递增 1（请参见*表 265*）。

表 265. GCM 模式 IV 寄存器初始化

寄存器	CRYP_IV1R[31:0]	CRYP_IV1L[31:0]	CRYP_IV0R[31:0]	CRYP_IV0L[31:0]
输入数据	ICB[127:96]	ICB[95:64]	ICB[63:32]	ICB[31:0] 32 位计数器 = 0x2

注： *在该模式下，不得为解密准备密钥。*

GCM 模式下的认证机制基于散列函数（称为 *Gf2mul*），其在二元 Galois 域内执行与固定参数（称为散列子密钥 (H)）的乘法。

要处理 GCM 消息，驱动程序必须经过四个阶段，后续子章节对此进行了介绍。

- 初始化阶段：外设准备 GCM 散列子密钥 (H) 并执行 IV 处理。
- 标头阶段：外设处理附加认证数据 (AAD)（仅使用散列计算）。
- 有效负载阶段：外设基于散列计算、密钥流加密和数据异或运算处理明文 (P)。其操作方式与密文 (C) 类似。
- 最后阶段：外设使用数据的最后一个块生成认证标记 (T)。

1. **GCM 初始化阶段**

   在此第一步中，将在内部计算和保存 GCM 散列子密钥 (H)，供处理所有块使用。建议遵守下列顺序：

   a) 确保通过将 CRYP_CR 寄存器中的 CRYPEN 位清零来禁止加密处理器。
   b) 通过将 CRYP_CR 中的 ALGOMODE 位编程为 0b01000 来选择 GCM 链接模式。
   c) 将 CRYP_CR 中的 GCM_CCMPH 位配置为 0b00 来指示正在进行初始化阶段。
   d) 按照*表 265* 的定义初始化 CRYP_KEYRx 中的密钥寄存器（128 位、192 位和 256 位）以及初始化向量 (IV)。
   e) 将 CRYPEN 位置 1 以开始计算散列密钥。
   f) 等待 CRYPEN 位由加密处理器清零后进入下一个阶段。

2. **GCM 标头阶段**

   在 GCM 初始化阶段后方能执行以下序列： 在跳转到有效负载阶段之前，必须先执行完该序列。加密与解密的顺序完全相同。

   g) 将 CRYP_CR 中的 GCM_CCMPH 位置为 0b01 来指示正在进行标头阶段。
   h) 将 CRYPEN 位置 1 开始接受数据。
   i) 如果是附加认证数据的最后一个块，则可选择用零填充数据以获得完整的块。
   j) 在加密处理器中添加附加认证数据。*第 35.3.5 节：用于执行密码操作的 CRYP 过程*介绍了三种可能的情况。
   k) 重复上一步，直到处理完倒数第二个附加认证数据块。对于最后一个块，执行前两步。提供所有附加认证数据后，请等待 BUSY 标志清零后进入下一阶段。

注： *如无附加认证数据（即 Len(A)=0），则可以跳过该阶段。*

在标头和有效负载阶段，CRYPEN 位不会自动由加密处理器清零。

3. **GCM 有效负载阶段（加密或解密）**

   如果有效负载大小不为空，则必须在 GCM 标头阶段后执行此序列。在此阶段期间，加密/解密的有效负载将存储在 CRYP_DOUT 寄存器中。

   l) 将 CRYPEN 位置 0。
   m) 将 CRYP_CR 寄存器中的 GCM_CCMPH 位配置为 0b10 来指示正在进行有效负载阶段。
   n) 通过 CRYP_CR 中的 ALGODIR 位来选择算法方向（0 表示加密，1 表示解密）。
   o) 将 CRYPEN 位置 1 开始接受数据。
   p) 如果是最后一个明文块，则可选择用零填充数据以获得完整的块。有关加密的更多详细信息，请参见*第 35.3.8 节：CRYP 窃取和数据填充*。
   q) 在加密处理器中附加有效负载数据并读取结果。*第 35.3.5 节：用于执行密码操作的 CRYP 过程*介绍了三种可能的情况。
   r) 重复上一步，直到倒数第二个明文块被加密或直到最后一个密文块被解密。对于最后一个明文块（仅限加密），执行前两步。对于最后一个块，当其大小小于 16 字节时，驱动程序必须将不属于明文或密文的一部分的位丢弃。提供完所有有效负载数据后，等待 BUSY 标志清零。

注： *如无有效负载数据（即 Len(C)=0），则可以跳过该阶段（请参见 GMAC 模式）。*

4. **GCM 最后阶段**

   在此最后一步中，加密处理器会生成 GCM 认证标记并将其存储在 CRYP_DOUT 寄存器中。

   s) 将 CRYP_CR 中的 GCM_CCMPH[1:0] 位配置为 0b11 来指示正在进行最后阶段。将同一寄存器中的 ALGODIR 位置 0。

   t) 将输入写入 CRYP_DIN 寄存器四次。输入必须包含附加认证数据的长度（以位为单位，64 位编码）和有效负载的长度（以位为单位，64 位编码），如*表 264* 所示。

注： *在此最后阶段，必须根据在 CRYP_CR 寄存器中编程的 DATATYPE 交换数据。*

   u) 等待 CRYP_SR 寄存器中的 OFNE 标志（FIFO 输出非空）置 1。

   v) 读取 CRYP_DOUT 寄存器四次：此输出即为认证标记。

   w) 禁止加密处理器（CRYP_CR 中的 CRYPEN 位 = 0）。

   x) 如果正在执行经验证的解密，请将生成的标记与通过消息传递的预期标记进行比较。

GCM 模式下的挂起/恢复操作

在标头或有效负载阶段中断当前消息之前，用户应用程序必须遵循以下序列：

1. 如果使用 DMA，则通过将 CRYP_DMACR 寄存器中的 DIEN 位清零来停止 DMA 对 IN FIFO 的数据传输。
2. 等待 IN 和 OUT FIFO 均为空（CRYP_SR 寄存器中的 IFEM=1 且 OFNE=0），并且 BUSY 位清零。
3. 如果使用 DMA，则通过将 CRYP_DMACR 寄存器中的 DOEN 位清零来停止 DMA 对 OUT FIFO 的数据传输。
4. 通过将 CRYP_CR 中的 CRYPEN 位置 0 来禁止加密处理器，然后保存当前配置（CRYP_CR 寄存器中的位 [9:2]、位 [17:16] 和位 19）。此外，还应保存初始化向量寄存器，因为 CRYP_IVx 寄存器在数据处理过程中已发生更改，不再是初始值。

注： *无需保存密钥寄存器，因为应用程序已知其初始密钥值。*

5. 保存上下文交换寄存器：CRYP_CSGCMCCM0..7 和 CRYP_CSGCM0..7
6. 如果使用 DMA，则保存 DMA 控制器状态（指向 IN 和 OUT 数据传输的指针以及剩余字节数等）。

要恢复消息处理，用户必须遵循以下序列：

1. 如果使用 DMA，则重新配置 DMA 控制器以完成 FIFO IN 和 FIFO OUT 传输的剩余部分。
2. 确保通过读取 CRYP_CR 中的 CRYPEN 位（必须为 0）来禁止加密处理器。
3. 使用 CRYP_CR 中的初始设置再次配置加密处理器，以及使用所保存的配置配置密钥寄存器。
4. 恢复上下文交换寄存器：CRYP_CSGCMCCM0..7 和 CRYP_CSGCM0..7
5. 使用保存的配置恢复 CRYP_IVx 寄存器。
6. 通过将 CRYPEN 位置 1 来使能加密处理器。
7. 如果使用 DMA，则通过将 CRYP_DMACR 寄存器中的 DIEN 和 DOEN 位置 1 再次使能对加密处理器的 DMA 请求。

注： *在标头阶段，不使用 DMA OUT FIFO 传输。*

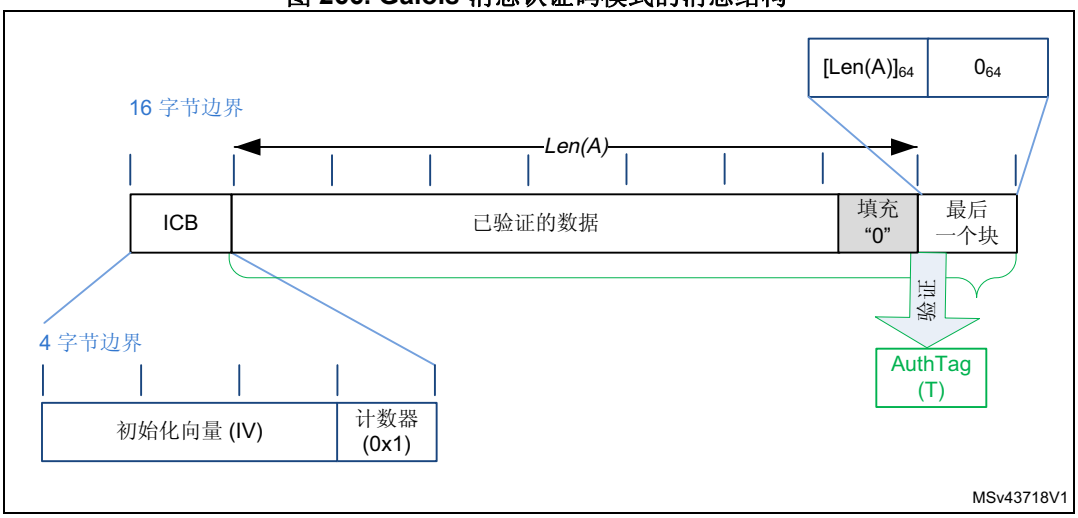
35.3.14 CRYP AES Galois 消息认证码 (GMAC)

概述

Galois 消息认证码 (GMAC) 支持认证明文以及生成相应的标记信息（也称为消息认证码）。它基于 GCM 算法，NIST *特别出版物 800-38D，块密码工作模式的建议 - Galois/计数器模式 (GCM) 和 GMAC* 中对此进行了相关定义。

图 266 给出了 GMAC 模式下的典型消息结构。

图 266. Galois 消息认证码模式的消息结构



AES GMAC 处理

通过向 CRYP_CR 中的 ALGOMODE 位域写入 0b110 来选择该模式。

GMAC 算法相当于应用在仅由标头构成的消息上的 GCM 算法。因此，除了不使用有效负载阶段 (3) 之外，所有步骤和设置均与 GCM 模式相同。

GMAC 模式下的挂起/恢复操作

除了只能中断标头阶段 (2) 外，GMAC 与 GCM 算法完全相同。

35.3.15 CRYP AES CBC-MAC 计数器模式 (CCM)

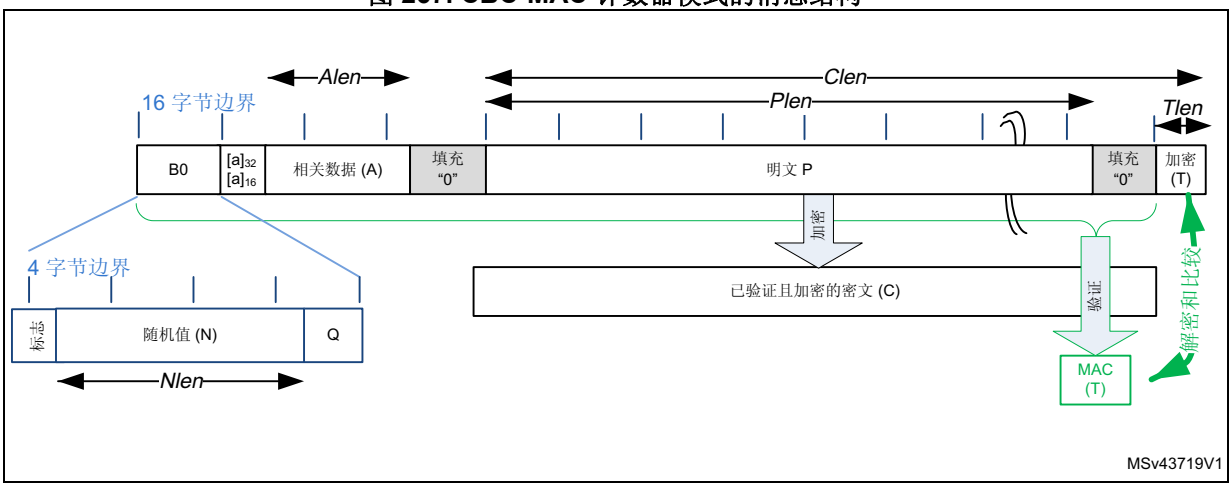
概述

AES 密码块链接-消息认证码计数器模式 (CCM) 算法可用于加密和认证明文，以及生成对应的密文和标记（也称为消息认证码）。为实现保密性，CCM 算法基于 AES 计数器模式。它使用密码块链接技术来生成消息认证码。这通常被称为 CBC-MAC。

注： *NIST 不允许将该 CBC-MAC 用作 CCM 规范上下文之外的认证模式。*

NIST *特别出版物 800-38C，块密码工作模式的建议 - CCM 模式实现认证和保密性*中对 CCM 链接进行了介绍。*图 267* 给出了 CCM 模式下的典型消息结构。

图 267. CBC-MAC 计数器模式的消息结构



该消息的结构如下：

- 一个 16 字节的首个认证块（标准将其定义为 B0），它由三个不同的字段组成：
  - Q：位字符串，表示 P 的字节长度 (Plen)。
  - *随机值 (N)*：一次性值（即，应为每一次新的通信分配新的随机值）。随机值 *Nlen* 的大小 + *Plen* 的大小应等于 15 个字节。
  - *标志*：最高有效字节，包含四个控制信息标志（根据标准规定）。它包含两个 3 位字符串，用于编码值 t（MAC 长度，以字节表示）和 *q*（明文长度，例如 Plen<2^{8q} 个字节）。请注意，与 q 相关的计数器块范围为 2^{8q-4}，即，如果 q 的最大值为 8，则密码中使用的计数器块应为 60 位。

注： *加密外设只能管理长度 Plen < 2^{36} +1 字节的填充明文/密文消息。*

- 与相关数据 (A) 关联的 16 字节块 (B)。

  消息的这一部分仅经过认证，未被加密。这部分具有已知长度 *ALen*，此长度值可以不是 16 字节的倍数（请参见*图 267*）。标准还具有以下规定：对于第一个消息块 (B1) 的 MSB 位，以字节表示的相关数据长度 (*a*) 必须按如下所述进行编码：
  - 如果 0 < a < 2^{16}-2^8，则将其编码为 $[a]_{16}$，即两个字节。
  - 如果 2^{16}-2^8 < a < 2^{32}，则将其编码为 0xff || 0xfe || $[a]_{32}$，即六个字节。
  - 如果 2^{32} < a < 2^{64}，则将其编码为 0xff || 0xff || $[a]_{64}$，即十个字节。

- 与明文消息 (P) 相关的 16 字节块 (B)，此明文消息将经过认证并被加密为密文 C（具有已知长度 *Plen*）。该长度可以不是 16 字节的倍数（请参见*图 267*），但不能超出 2^{32} 个 128 位块。
- 长度为 *Tlen* 的加密 MAC (T) 被附加到总长度为 *Clen* 的密文 C。
- 当消息某一部分（A 或 P）的长度不是 16 字节的倍数时，需要采取特殊填充方案。

注： *CCM 链接模式同样只能与相关数据搭配使用（即，无有效负载）。*

例如，*NIST 特别出版物 800-38C* 的 C.1 部分给出了以下内容：

N：10111213 141516（Nlen= 56 位或 0x7 字节）

A：00010203 04050607（Alen= 64 位或 0x8 字节）

P：20212223（Plen = 32 位，即 Q = 0x4 字节）

T：6084341b（Tlen = 32 位或 t = 4）

```
B0: 4f101112 13141516 00000000 00000004
B1: 00080001 02030405 06070000 00000000
B2: 20212223 00000000 00000000 00000000
CTR0: 0710111213 141516 00000000 00000000
CTR1: 0710111213 141516 00000000 00000001
```

以下章节对控制块 CTRx 的使用进行了介绍。必须通过软件来管理第一个块 (B0) 生成 CTR0 的过程。

AES CCM 处理

通过向 CRYP_CR 中的 ALGOMODE 位域写入 0b1001 来选择该模式。

生成-加密过程的数据输入包括一个有效随机值、一个有效的有效负载字符串和一个有效的相关数据字符串，这些数据均经过正确格式化。对经过格式化的数据应用 CBC 链接机制可生成长度已知的 MAC。计数器模式加密需要足够长的计数器块序列作为输入，其将应用于有效负载字符串并单独应用于 MAC。得到的数据称为密文 C，它是明文 P 的生成-加密过程的输出。

CRYP_IV 寄存器用于处理每个数据块。加密处理器以第一个块 (B0) 定义的位长度自动使 CTR 计数器递增。由应用程序写入的第一个计数器 CTR1 相当于 B0（前 5 位清零，并且包含 P 字节长度的最高有效位也清零），然后按 1 递增（请参见*表 266*）。

表 266. CCM 模式 IV 寄存器初始化

寄存器	CRYP_IV0L[31:0]	CRYP_IV0R[31:0]	CRYP_IV1L[31:0]	CRYP_IV1R[31:0]
字节序	IV[0:31]	IV[32:63]	IV[64:95]	IV[96:127]
输入数据	B0[31:0]，其中的 5 个最高有效位置 0（标志位）	B0[63:32]	B0[95:64]	B0[127:96]，其中除了位 0 置 1 外，Q 长度位均置 0

注： *在该模式下，不得为解密准备密钥。*

要处理 CCM 消息，驱动程序必须经过四个阶段，下文对此进行了说明。

- 初始化阶段：外设处理第一个块并准备第一个计数器块。
- 标头阶段：外设处理相关数据 (A)（仅使用散列计算）。
- 有效负载阶段：外设基于散列计算、计数器块加密和数据异或运算处理明文 (P)。其操作方式与密文 (C) 类似。
- 最后阶段：外设生成消息认证码 (MAC)。

1. CCM 初始化阶段

在此第一步中，将 CCM 消息的第一个块 (B0) 编程到 CRYP_DIN 寄存器中。在此阶段期间，CRYP_DOUT 寄存器不包含任何输出数据。建议遵守下列顺序：

a) 确保通过将 CRYP_CR 寄存器中的 CRYPEN 位清零来禁止加密处理器。

b) 通过将 CRYP_CR 寄存器中的 ALGOMODE 位编程为 0b01001 来选择 CCM 链接模式。

c) 将 CRYP_CR 中的 GCM_CCMPH 位配置为 0b00 来指示正在进行初始化阶段。

d) 按照*表 266* 的定义初始化 CRYP_KEYRx 中的密钥寄存器（128 位、192 位和 256 位）以及包含 CTR1 信息的初始化向量 (IV)。

e) 将 CRYP_CR 中的 CRYPEN 位置 1 开始接受数据。

f) 向 CRYP_DIN 寄存器中写入 B0 数据包，然后等待 CRYPEN 位由加密处理器清零后进入下一个阶段。

注： *在此初始化阶段，必须根据在 CRYP_CR 寄存器中编程的 DATATYPE 交换数据。*

2. CCM 标头阶段

在 CCM 初始化阶段后方能执行以下序列： 在跳转到有效负载阶段之前，必须先执行完该序列。加密与解密的顺序完全相同。在此阶段期间，CRYP_DOUT 寄存器不包含任何输出数据。

g) 将 CRYP_CR 中的 GCM_CCMPH 位置为 0b01 来指示正在进行标头阶段。

h) 将 CRYPEN 位置 1 开始接受数据。

i) 如果是相关数据的最后一个块，则可选择用零填充数据以获得完整的块。

j) 在加密处理器中附加相关数据。*第 35.3.5 节：用于执行密码操作的 CRYP 过程*介绍了三种可能的情况。

k) 重复上一步，直到处理完倒数第二个相关数据块。对于最后一个块，执行前两步。提供完所有附加认证数据后，等待 BUSY 标志清零。

注： *如无相关数据 (AIen=0)，则可以跳过该阶段。*

注： *必须使用相关数据长度格式化相关数据的首个块 B1。此任务必须由驱动程序进行管理。*

3. **CCM 有效负载阶段（加密或解密）**

   如果有效负载大小不为空，则必须在 CCM 标头阶段后执行此序列。在此阶段期间，加密/解密的有效负载将存储在 CRYP_DOUT 寄存器中。

   l) 将 CRYPEN 位置 0。

   m) 将 CRYP_CR 中的 GCM_CCMPH 位配置为 0b10 来指示正在进行有效负载阶段。

   n) 通过 CRYP_CR 中的 ALGODIR 位来选择算法方向（0 表示加密，1 表示解密）。

   o) 将 CRYPEN 位置 1 开始接受数据。

   p) 如果是最后一个明文块，则可选择用零填充数据以获得完整的块（仅限加密）。有关解密的更多详细信息，请参见*第 35.3.8 节：CRYP 窃取和数据填充*。

   q) 在加密处理器中附加有效负载数据并读取结果。*第 35.3.5 节：用于执行密码操作的 CRYP 过程*介绍了三种可能的情况。

   r) 重复上一步，直到倒数第二个明文块被加密或直到最后一个密文块被解密。对于最后一个明文块（仅限加密），执行前两步。对于最后一个密文块（仅限解密），当其大小小于 16 字节时，驱动程序必须将不属于明文的一部分的数据丢弃。提供完所有有效负载数据后，等待 BUSY 标志清零。

注： *如无有效负载数据（即，Plen=0 或 Clen=Tlen），则可以跳过该阶段。*

注： *解密密文 C 时，不要忘了移除 $LSB_{Tlen}(C)$ 加密标记信息。*

4. **CCM 最后阶段**

   在此最后一步中，加密处理器会生成 CCM 认证标记并将其存储在 CRYP_DOUT 寄存器中。

   s) 将 CRYP_CR 中的 GCM_CCMPH[1:0] 位配置为 0b11，以指示正在进行最后阶段，并将同一寄存器中的 ALGODIR 位置 0。

   t) 在 CRYP_DIN 中加载*表 266* 给出的 CTR0 信息（位 [0] 置 0）。

注： *在此最后阶段，必须根据在 CRYP_CR 寄存器中编程的 DATATYPE 交换数据。*

   u) 等待 CRYP_SR 寄存器中的 OFNE 标志（FIFO 输出非空）置 1。

   v) 读取 CRYP_DOUT 寄存器四次：此输出即为加密的 CCM 标记。

   w) 禁止加密处理器（CRYP_CR 中的 CRYPEN 位置 0）。

   x) 如果正在执行认证解密，则将生成的加密标记与在密文中填充的加密标记进行比较，即，$LSB_{Tlen}(C) = MSB_{Tlen}$（CRYP_DOUT 数据）。

CCM 模式下的挂起/恢复操作

在有效负载阶段中断当前消息之前，用户应用程序必须遵循以下序列：

1. 如果使用 DMA，则通过将 CRYP_DMACR 寄存器中的 DIEN 位清零来停止 DMA 对 IN FIFO 的数据传输。
2. 等待 IN 和 OUT FIFO 均为空（CRYP_SR 寄存器中的 IFEM=1 且 OFNE=0），并且 BUSY 位清零。
3. 如果使用 DMA，则通过将 CRYP_DMACR 寄存器中的 DOEN 位清零来停止 DMA 对 OUT FIFO 的数据传输。
4. 通过将 CRYP_CR 中的 CRYPEN 位置 0 来禁止加密处理器，然后保存当前配置（CRYP_CR 寄存器中的位 [9:2]、位 [17:16] 和位 19）。此外，还应保存初始化向量寄存器，因为 CRYP_IVx 寄存器在数据处理过程中已发生更改，不再是初始值。

注： *无需保存密钥寄存器，因为应用程序已知其初始密钥值。*

5. 保存上下文交换寄存器：CRYP_CSGCMCCM0..7。
6. 如果使用 DMA，则保存 DMA 控制器状态（指向 IN 和 OUT 数据传输的指针以及剩余字节数等）。

要恢复消息处理，用户应用程序必须遵循以下序列：

1. 如果使用 DMA，则重新配置 DMA 控制器以完成 FIFO IN 和 FIFO OUT 传输的剩余部分。
2. 确保通过读取 CRYP_CR 中的 CRYPEN 位（必须为 0）来禁止加密处理器。
3. 使用 CRYP_CR 中的初始设置再次配置加密处理器，以及使用所保存的配置再次配置密钥寄存器。
4. 恢复上下文交换寄存器：CRYP_CSGCMCCM0..7。
5. 使用保存的配置恢复 CRYP_IVx 寄存器。
6. 通过将 CRYPEN 位置 1 来使能加密处理器。
7. 如果使用 DMA，则通过将 CRYP_DMACR 寄存器中的 DIEN 和 DOEN 位置 1 再次使能对加密处理器的 DMA 请求。

注： *在标头阶段，不使用 DMA OUT FIFO 传输。*

35.3.16 CRYP 数据寄存器和数据交换

简介

CRYP_DIN 寄存器是外设的 32 位宽数据输入寄存器。最多可将四个 64 位 (TDES) 或两个 128 位 (AES) 明文（加密时）或密文（解密时）块输入到输入 FIFO 中，一次输入一个 32 位字。

写入 FIFO 的首个字为输入块的 LSB。最后写入的是输入块的 MSB。DATATYPE = 00（无数据交换）时，CRYP_DIN 数据字节序如下所述：

- 在 DES/TDES 模式下：

  数据块的位 1（最左位）对应输入到 FIFO 中的首个字的 MSB（位 31），位 64（最右位）对应输入到 FIFO 中的第二个字的 LSB（位 0）。
- 在 AES 模式下：

  数据块的位 0（最左位）对应写入到 FIFO 中的首个字的 MSB（位 31），位 127（最右位）对应写入到 FIFO 中的第四个字的 LSB（位 0）。

类似地，CRYP_DOUT 寄存器是外设的 32 位宽数据输出寄存器。该寄存器为只读寄存器，最多可接收来自输出 FIFO 的四个 64 位 (TDES) 或两个 128 位 (AES) 明文（加密时）或密文（解密时）块，一次接收一个 32 位字。

与输入数据相似，输出块的 LSB 是从输出 FIFO 读取的首个字。最后读取的是输出块的 MSB。DATATYPE = 00（无数据交换）时，CRYP_DOUT 数据字节序如下所述：

- 在 DES/TDES 模式下：

  数据块的位 1（最左位）对应从 FIFO 中读取的首个字的 MSB（位 31），位 64（最右位）对应从 FIFO 中读取的第二个字的 LSB（位 0）。
- 在 AES 模式下：

  数据块的位 0（最左位）对应从 FIFO 中读取的首个字的 MSB（位 31），位 127（最右位）对应从 FIFO 中读取的第四个字的 LSB（位 0）。

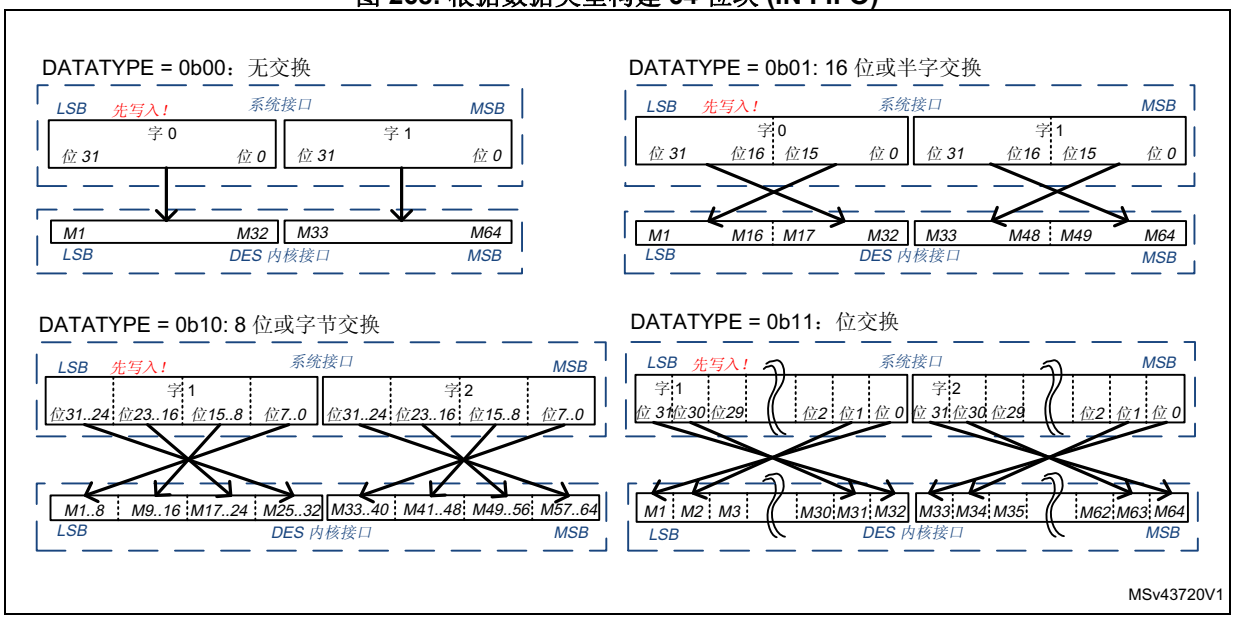
DES/TDES 数据交换功能

根据要处理的数据类型（例如，当数据为 ASCII 文本流时的字节交换），必须先对从输入 FIFO 读取的数据进行位、字节、半字或无交换操作，然后再将其输入小端 DES 处理内核。小端 DES 处理内核产生的数据必须先进行同样的交换操作，然后才能写入输出 FIFO。

图 268 显示了 DES 处理内核如何使用由驱动程序弹出到 IN FIFO 中的两个连续 32 位字构建 64 位数据块 M1...64。该操作将根据 CRYP_CR 寄存器中的 DATATYPE 位域来完成。

注： *IN FIFO 和 CRYP 数据块之间，以及 CRYP 数据块和 OUT FIFO 之间执行相同的交换操作。*

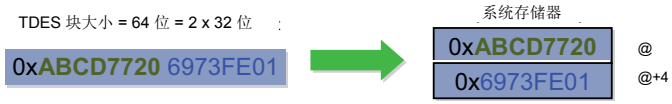
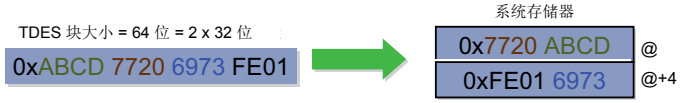
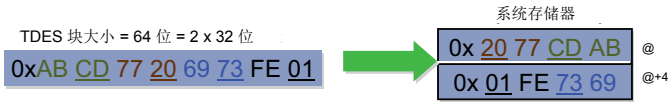
图 268. 根据数据类型构建 64 位块 (IN FIFO)



注： *CRYP 密钥寄存器 (CRYP_Kx(L/R)) 和初始化寄存器 (CRYP_IVx(L/R)) 对所选交换模式不敏感。它们有固定的小端配置（请分别参见第 35.3.17 节和第 35.3.18 节）。*

表 267 给出了数据交换的典型示例。

表 267. DES/TDES 数据交换功能

CRYP_CR 中的 DATATYPE	要执行的交换	数据块表示（64 位） 0xABCD7720 6973FE01 系统存储器数据 （明文或密码）
0b00	无交换	地址 @：0xABCD7720（LSB，先写入） 地址 @+4：0x6973FE01 
0b01	半字（16 位）交换	地址 @：0x7720ABCD（已交换 LSB，先写入） 地址 @+4：0xFE016973 
0b10	字节（8位）交换	地址 @：0x2077CDAB（已交换 LSB，先写入） 地址 @+4：0x01FE7369 
0b11	位交换	LSB 数据字：0xABCD7720 0b1010 1011 1100 1101 0111 0111 0010 0000 MSB 数据字：0x6973FE01 0b0110 1001 0111 0011 1111 1110 0000 0001 地址 @：0x04EEB3D5（已交换 LSB，先写入） 地址 @+4：0x807FCE96

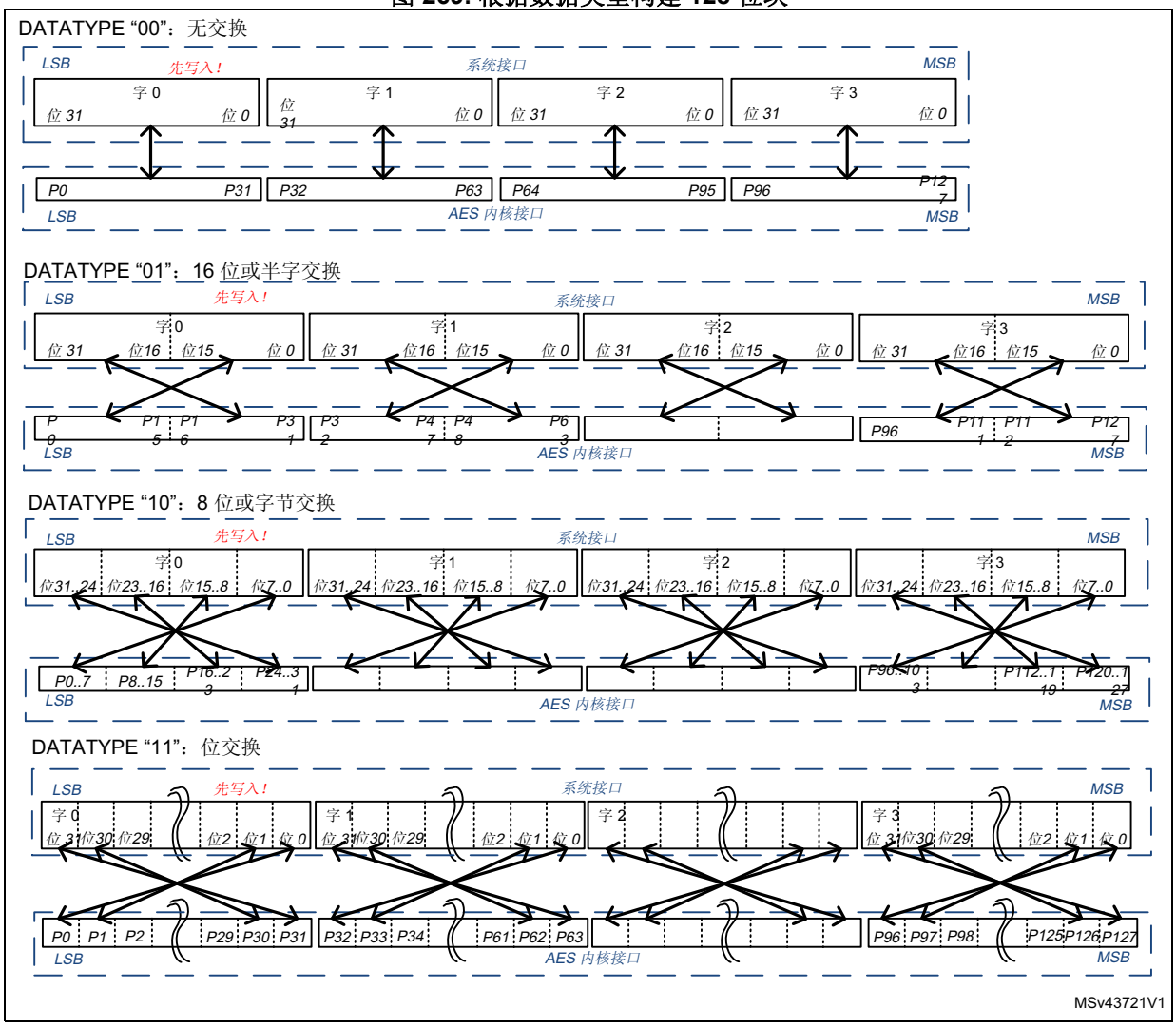
AES 数据交换功能

根据要处理的数据类型（例如，当数据为 ASCII 文本流时的字节交换），必须先对从输入 FIFO 读取的数据进行位、字节、半字或无交换操作，然后再将其输入小端 AES 处理内核。小端 AES 处理内核产生的数据必须先进行同样的交换操作，然后才能写入输出 FIFO。

图 269 显示了 AES 处理内核如何使用由驱动程序写入到 CRYP_DIN 寄存器的四个连续 32 位字构建 128 位数据块 P0..127。该操作将根据 CRYP 寄存器 (CRYP_CR) 中的 DATATYPE 位域来完成。

注： *CRYP_DIN 和 CRYP 数据块之间，以及 CRYP 数据块和 CRYP_DOUT 之间执行相同的交换操作。*

图 269. 根据数据类型构建 128 位块



注： *交换操作仅与 CRYP_DOUT 和 CRYP_DIN 寄存器有关。 CRYP_KxL/KxR 和 CRYP_IVxL/IVxR 寄存器对所选交换模式不敏感。它们有固定的小端配置（请参见第 35.3.17 节和第 35.3.18 节）。*

表 268 给出了数据交换的典型示例。

表 268. AES 数据交换功能

CRYP_CR 中的 DATATYPE	要执行的交换	数据块表示（64 位） 0x4E6F7720 69732074
		系统存储器数据（小端）
0b00	无交换	地址 @：0x4E6F7720（LSB，先写入） 地址 @+4：0x69732074
0b01	半字（16 位）交换	地址 @：0x77204E6F（已交换 LSB，先写入） 地址 @+4：0x20746973
0b10	字节（8位）交换	地址 @：0x20776F4E（已交换 LSB，先写入） 地址 @+4：0x74207369
0b11	位交换	LSB 数据字：0x4E6F7720 0b0100 1110 0110 1111 0111 0111 0010 0000 MSB 数据字：0x69732074 0b0110 1001 0111 0011 0010 0000 0111 0100 地址 @：0x4EEF672（已交换 LSB，先写入） 地址 @+4：0x2E04CE96

35.3.17 CRYP 密钥寄存器

CRYP_Kx 寄存器用于存储加密或解密密钥。

这类寄存器使用小端模式配置进行组织，总共有八个，如*表 269* 所示。

表 269. 密钥寄存器 CRYP_KxR/LR 字节序（TDES K1/2/3 和 AES 128/192/256 位密钥）

K0LR[31:0]	K0RR[31:0]	K1LR[31:0]	K1RR[31:0]	K2LR[31:0]	K2RR[31:0]	K3LR[31:0]	K3RR[31:0]
-	-	K1[1:32]	K1[33:64]	K2[1:32]	K2[33:64]	K2[1:32]	K2[33:64]
K0LR[31:0]	**K0RR[31:0]**	**K1LR[31:0]**	**K1RR[31:0]**	**K2LR[31:0]**	**K2RR[31:0]**	**K3LR[31:0]**	**K3RR[31:0]**
-	-	-	-	k[0:31]	k[32:63]	k[64:95]	k[96:127]
K0LR[31:0]	**K0RR[31:0]**	**K1LR[31:0]**	**K1RR[31:0]**	**K2LR[31:0]**	**K2RR[31:0]**	**K3LR[31:0]**	**K3RR[31:0]**
-	-	k[0:31]	k[32:63]	k[64:95]	k[96:127]	k[128:159]	k[160:191]
K0LR[31:0]	**K0RR[31:0]**	**K1LR[31:0]**	**K1RR[31:0]**	**K2LR[31:0]**	**K2RR[31:0]**	**K3LR[31:0]**	**K3RR[31:0]**
k[0:31]	k[32:63]	k[64:95]	k[96:127]	k[128:159]	k[160:191]	k[192:223]	k[224:255]

注： *DES/TDES 密钥包含加密处理器不使用的 8 位奇偶校验信息。换言之，不使用各个 64 位密钥值 Kx[1:64] 的位 8、16、24、32、40、48、56 和 64。*

可将密钥视为四个 64 位数据项。因此，密钥在系统存储器中的数据格式和表示形式均不同于明文或密文数据。

当 CRYP_SR 寄存器的 BUSY 位置 1 时，针对 CRYP_Kx(L/R) 寄存器的任何写操作都会被忽略（即，寄存器内容不会被修改）。因此，软件在修改密钥寄存器之前，必须检查 BUSY 是否等于 0。

密钥寄存器不受数据交换功能（由 CRYP_CR 寄存器中 DATATYPE 值控制）的影响。

有关 CRYP_Kx(L/R) 寄存器的详细说明，请参见*第 35.6 节：CRYP 寄存器*。

35.3.18 CRYP 初始化向量寄存器

CRYP_IVxL/IVxR 寄存器用于存储初始化向量或随机值，具体取决于所选链接模式。在使用过程中，完成每一轮的 TDES 或 AES 内核计算后，内核都会更新这些寄存器。

这类寄存器使用小端模式配置进行组织，总共有四个，如*表 270* 所示。

表 270. 初始化向量寄存器 CRYP_IVxR 字节序

CRYP_IV1R[31:0]	CRYP_IV1L[31:0]	CRYP_IV0R[31:0]	CRYP_IV0L[31:0]
IV[96:127]	IV[64:95]	IV[32:63]	IV[0:31]

可将初始化向量寄存器视为两个 64 位数据项。因此，初始化向量寄存器在系统存储器中的数据格式和表示形式均不同于明文或密文数据。

当 CRYP_SR 寄存器的 BUSY 位置 1 时，针对 CRYP_IV0...1(L/R) 寄存器的任何写操作都会被忽略（即，寄存器内容不会被修改）。因此，软件在修改初始化向量之前，必须检查 CRYP_SR 寄存器中的 BUSY 是否等于 0。

读取 CRYP_IV0...1(L/R) 寄存器会返回最新计数器值（对于管理挂起模式很有用，CCM/GCM 除外）。

注： *在 DES/TDES 模式下，仅使用 CRYP_IV0x。*

初始化向量寄存器不受数据交换功能（由 CRYP_CR 寄存器中 DATATYPE 值控制）的影响。

有关 CRYP_IVxL/IVxR 寄存器的详细说明，请参见*第 35.6 节：CRYP 寄存器*。

35.3.19 CRYP DMA 接口

加密处理器可使用一个接口连接 DMA（直接存储器访问）控制器。DMA 操作通过 CRYP DMA 控制寄存器 (CRYP_DMACR) 进行控制。

使用 DMA 的数据输入

可通过将 CRYP_DMACR 寄存器中的 DIEN 位置 1 使能 DMA，从而向加密外设中写入数据。如果该位置 1，则加密处理器每次需要向 CRYP_DIN 寄存器中写入字时，都会在 INPUT 阶段发出 DMA 请求。

表 271 给出了通过 DMA 控制器将数据从存储器传输到加密处理器时的推荐配置。

表 271. 针对存储器到外设的 DMA 传输的加密处理器配置

DMA 通道 控制寄存器字段	编程建议
传输数据量大小	消息长度，128 位的倍数。此 128 位粒度就 DES 而言对应于两个块，就 AES 而言对应于一个块。 根据所选算法和模式，可能需要采用特殊填充/密文窃取技术。有关详细信息，请参见*第 35.3.8 节：CRYP 窃取和数据填充*。
源突发大小（存储器）	CRYP FIFO_size /2 /transfer_width = **4**
目标突发大小（外设）	CRYP FIFO_size /2 /transfer_width = 4 （FIFO_size = 8x32 位，transfer_width = 32 位）
DMA FIFO 大小	CRYP FIFO_size /2 = 16 字节
源传输宽度（存储器）	32 位字
目标传输宽度（外设）	32 位字
源地址递增（存储器）	是，每次 32 位传输后。
目标地址递增（外设）	应使用 CRYP_DIN 的固定地址（无递增）

使用 DMA 的数据输出

要使能 DMA 以从 AES 外设读取数据，请将 CRYP_DMACR 寄存器中的 DOEN 位置 1。如果该位置 1，则加密处理器每次需要从 CRPY_DOUT 寄存器中读取字时，都会在 OUTPUT 阶段发出 DMA 请求。

表 272 给出了通过 DMA 控制器将数据从加密处理器传输到存储器时的推荐配置。

表 272. 针对外设到存储器的 DMA 传输的加密处理器配置

DMA 通道 控制寄存器字段	编程建议
传输数据量大小	消息长度，128 位的倍数。此 128 位粒度就 DES 而言对应于两个块，就 AES 而言对应于一个块。 根据所用的算法，必须丢弃额外的位。
源突发大小（外设）	使用 *DES* 时： 单次传输（突发大小 = **1**） 使用 *AES* 时： CRYP FIFO_size /2 /transfer_width = **4** （FIFO_size = 8x32 位，transfer_width = 32 位）
目标突发大小 （存储器）	CRYP FIFO_size /2 /transfer_width = **4**
DMA FIFO 大小	CRYP FIFO_size /2 = 16 字节
源传输宽度（外设）	32 位字
存储器传输宽度 （存储器）	32 位字
源地址递增（外设）	应使用 CRYP_DOUT 的固定地址（无递增）。
目标地址递增 （存储器）	是，每次 32 位传输后。

DMA 模式

使用 AES 时，加密处理器通过 `cryp_in_dma` 和 `cryp_out_dma` 内部输入/输出信号管理两个 DMA 传输请求，请求信号的有效条件如下：

- 对于 IN FIFO：每次 CRYP 从 FIFO 中读取一个块时
- 对于 OUT FIFO：每次加密处理器将一个块写入 FIFO 中时

使用 DES 时，加密处理器通过 `cryp_in_dma` 和 `cryp_out_dma` 内部输入/输出信号管理两个 DMA 传输请求，请求信号的有效条件如下：

- 对于 IN FIFO：每次加密处理器从 FIFO 中读取两个块时
- 对于 OUT FIFO：每次加密处理器将一个字写入 FIFO 中（单次传输）时。请注意，如果已将两个块写入 FIFO，则也会触发突发传输

如果已禁止加密外设或已将 DMA 使能位清零（CRYP_DMACR 寄存器中的 DIEN 位用于 IN FIFO，DOEN 位用于 OUT FIFO），则会禁止所有请求信号。

注意： 在填满 CRYP 输入 FIFO 之前，由 DMA 控制器清空加密外设输出 FIFO 至关重要。为此，应对 DMA 控制器进行相应配置，以使从外设到存储器的传输优先级高于从存储器到外设的传输。

有关 DMA 操作的更多详细信息，请参见*第 35.3.5 节：用于执行密码操作的 CRYP 过程*。

35.3.20 CRYP 差错管理

加密处理器不生成错误标志。

35.4 CRYP 中断

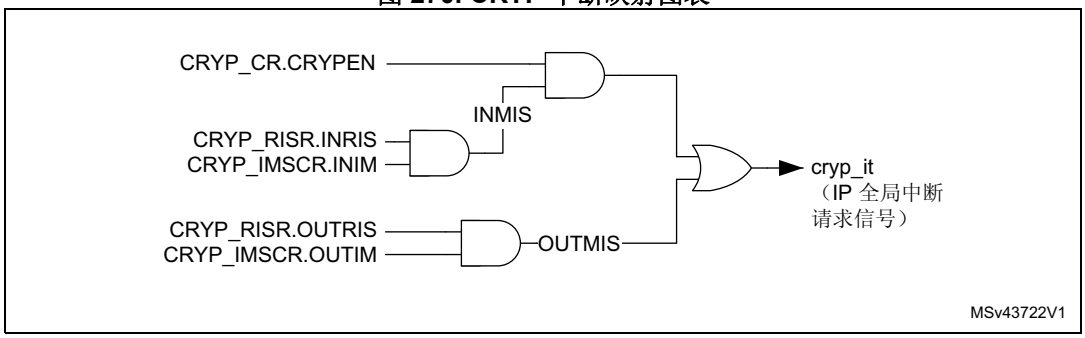
概述

加密处理器可生成两个独立的可屏蔽中断源，用以通知发生以下事件：

- 输入 FIFO 为空或未满
- 输出 FIFO 已满或非空

这两个中断源合为同一个中断信号，而该中断信号是 CRYP 外设发出的唯一中断信号，用于驱动 NVIC（嵌套向量中断控制器）。*图 270* 对中断逻辑进行了汇总。

图 270. CRYP 中断映射图表



通过更改 CRYP_IMSCR 寄存器中的屏蔽位，可单独使能或禁止各个 CRYP 中断源。将相应的屏蔽位置 1 以使能中断。

关于各个可屏蔽中断源的状态，通过 CRYP_RISR 寄存器可以读取原始中断状态，通过 CRYP_MISR 寄存器可以读取屏蔽中断状态。可以从 CRYP_SR 寄存器中读取各个事件标志源的状态。

表 273 对可用功能进行了总结。

表 273. CRYP 中断请求

中断事件	事件标志（中断状态）	使能控制位	事件标志（源）
输出 FIFO 已满 (Output FIFO full)	OUTRIS、OUTMIS	OUTIM 和 CRYPEN	OFFU
输出 FIFO 非空 (Output FIFO not empty)			OFNE
输入 FIFO 未满 (Input FIFO not full)	OUTRIS、OUTMIS	INIM 和 CRYPEN	IFNF
输入 FIFO 为空 (Input FIFO empty)			IFEM

输出 FIFO 服务中断 - OUTMIS

当输出 FIFO 中存在一个或多个（32 位字）数据项时，即会产生输出 FIFO 服务中断。通过读取输出 FIFO 的数据，直到读完所有有效（32 位）字（即，该中断的状态与输出 FIFO 非空标志 OFNE 一致时），即可将此中断清除。

输出 FIFO 服务中断 OUTMIS 不是通过 CRYP 使能位使能。因此，如果输出 FIFO 非空，即使禁止 CRYP 之后也不会强制 OUTMIS 信号为低电平。

输入 FIFO 服务中断 - INMIS

当输入 FIFO 中少于四个字时，会产生输入 FIFO 服务中断。对输入 FIFO 执行写操作直到其中所含字不小于四字，这样即可将此中断清除。

输入 FIFO 服务中断 INMIS 通过 CRYP 使能位使能。因此，禁止 CRYP 之后，即使输入 FIFO 为空，INMIS 信号也为低（无效）。

35.5 CRYP 处理时间

下表汇总了每种工作模式下处理 128 位块需要的时间。

表 274. ECB、CBC 和 CTR 模式下每个 128 位块的处理时间（以时钟周期计）

算法/密钥大小	ECB	CBC	CTR
128b	14	14	14
192b	16	16	16
256b	18	18	18

表 275. GCM 和 CCM 模式下每个 128 位块的处理时间（以时钟周期计）

算法/密钥大小	GCM					CCM				
	初始化	标头	有效负载	变量	总计	初始化	标头	有效负载	变量	总计
128b	24	10	14	14	**62**	12	14	25	14	**65**
192b	28	10	16	16	**70**	14	16	29	16	**75**
256b	32	10	18	18	**78**	16	18	33	18	**85**

35.6 CRYP 寄存器

加密内核连接若干控制和状态寄存器、八个密钥寄存器和四个初始化向量寄存器。

35.6.1 CRYP 控制寄存器 (CRYP_CR)

CRYP control register

偏移地址：0x00

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.				ALGOMODE[3]	Res.	GCM_CCMPH	
												rw		rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CRYPEN	FFLUSH	Res.	Res.	Res.	Res.	KEYSIZE		DATATYPE		ALGOMODE[2:0]			ALGODIR	Res.	Res.
rw	w					rw	rw	rw	rw	rw	rw	rw	rw		

位 31:20 保留，必须保持复位值

位 19 **ALGOMODE[3]**：请参见位 [5:3] 说明

位 18 保留，必须保持复位值

位 17:16 **GCM_CCMPH**：GCM 或 CCM 阶段选择 (GCM or CCM Phase selection)

如果未在 ALGOMODE 字段中选择 GCM、GMAC 或 CCM 算法，则此位域无影响。

00：初始化阶段
01：标头阶段
10：有效负载阶段
11：最后阶段

位 15 **CRYPEN**：CRYP 处理器使能 (CRYP processor Enable)

0：禁止加密处理器外设
1：使能加密处理器外设

当密钥准备过程结束 (ALGOMODE= 0b111) 时或者在 GCM/GMAC 或 CCM 初始化阶段后，硬件会自动将该位清零。

位 14 **FFLUSH**：CRYP FIFO 刷新 (CRYP FIFO Flush)

0：禁止 FIFO 刷新
1：使能 FIFO 刷新

CRYPEN = 0 时，在此位中写入 1 可刷新 IN 和 OUT FIFO（即，将复位 FIFO 的读指针和写指针）。若在此位中写入 0 则不会产生影响。CRYPEN = 1 时，在该位中写入 0 或 1 都不会产生影响。

读取该位时，始终返回 0。

FFLUSH 位必须在 BUSY=0 时才能置 1。否则，虽然 FIFO 会进行刷新，但在刷新操作后可能会将所处理的块压入输出 FIFO，从而产生 FIFO 非空的情况。

位 13:10 保留，必须保持复位值

位 9:8 **KEYSIZE**：密钥大小选择 (Key Size selection)（仅限 AES 模式）

此位域定义了 AES 加密内核使用的密钥位长度。此位域在 DES 或 TDES 模式中可忽略。

00：128 位密钥长度
01：192 位密钥长度
10：256 位密钥长度
11：保留，不使用该值

BUSY=1 时写入 KEYSIZE 位无影响。这些位只能在 BUSY=0 时进行配置。

位 7:6 **DATATYPE**：数据类型选择 (Data type selection)

该位域定义写入 CRYP_DIN 寄存器或从 CRYP_DOUT 寄存器读取的数据的格式。有关详细信息，请参见*第 35.3.16 节：CRYP 数据寄存器和数据交换*。

00：32 位数据。不交换各个字。压入 IN FIFO（或 OUT FIFO 弹出）的首个字形成数据块的位 1...32，第二个字形成数据块的位 33...64。
01：16 位数据或半字。可将压入 IN FIFO（或 OUT FIFO 弹出）的每个字视为 2 个已相互交换的半字。
10：8 位数据或字节。可将压入 IN FIFO（或 OUT FIFO 弹出）的每个字视为 4 个已相互交换的字。
11：位数据或位串。可将压入 IN FIFO（或 OUT FIFO 弹出）的每个字视为 32 个已相互交换的位（字符串的首个位在位置 0）。

BUSY=1 时写入 DATATYPE 位无影响。这些位只能在 BUSY=0 时进行配置。

位 [5:3] **ALGOMODE[2:0]**：算法模式 (Algorithm mode)

以下定义包括位 19：
0000：TDES-ECB（三 DES 电子密码本）。
0001：TDES-CBC（三 DES 密码分组链接）。
0010：DES-ECB（简易 DES 电子密码本）。
0011：DES-CBC（简易 DES 密码分组链接）。
0100：AES-ECB（AES 电子密码本）。
0101：AES-CBC（AES 密码分组链接）。
0110：AES-CTR（AES 计数器模式）。
0111：针对 ECB 或 CBC 解密准备 AES 密钥。
1000：AES-GCM（Galois 计数器模式）和 AES-GMAC（Galois 消息认证码模式）。
1001：AES-CCM（CBC-MAC 计数器模式）。

BUSY=1 时写入 ALGOMODE 位无影响。这些位只能在 BUSY=0 时进行配置。

位 2 **ALGODIR**：算法方向 (Algorithm direction)

0：加密
1：解密

BUSY=1 时写入 ALGODIR 位无影响。该位只能在 BUSY=0 时进行配置。

位 1:0 保留，必须保持复位值

35.6.2 CRYP 状态寄存器 (CRYP_SR)

CRYP status register

偏移地址：0x04

复位值：0x0000 0003

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BUSY	OFFU	OFNE	IFNF	IFEM
											r	r	r	r	r

位 31:5 保留，必须保持复位值

位 4 **BUSY**：忙碌位 (Busy bit)

0：CRYP 内核当前没有处理任何数据。原因是：

– CRYP 内核已禁止（CRYP_CR 寄存器的 CRYPEN=0）且最后的处理过程已完成，

– 或者 CRYP 内核正在等待输入 FIFO 的数据足够多或输出 FIFO 的自由空间足够大（即，在这两种情况下，DES 中至少有 2 个字，AES 中至少有 4 个字）。

1：CRYP 内核目前正在处理一个数据块或正在准备密钥（仅限 AES ECB 或 CBC 解密）。

位 3 **OFFU**：输出 FIFO 已满标志 (Output FIFO full flag)

0：输出 FIFO 未满

1：输出 FIFO 已满

位 2 **OFNE**：输出 FIFO 非空标志 (Output FIFO not empty flag)

0：输出 FIFO 为空

1：输出 FIFO 非空

位 1 **IFNF**：输入 FIFO 未满标志 (Input FIFO not full flag)

0：输入 FIFO 已满

1：输入 FIFO 未满

位 0 **IFEM**：输入 FIFO 为空标志 (Input FIFO empty flag)

0：输入 FIFO 非空

1：输入 FIFO 为空

35.6.3 CRYP 数据输入寄存器 (CRYP_DIN)

CRYP data input register

偏移地址：0x08

复位值：0x0000 0000

CRYP_DIN 寄存器为数据输入寄存器。其宽度为 32 位。最多可将四个 64 位 (TDES) 或两个 128 位 (AES) 明文（加密时）或密文（解密时）块输入到输入 FIFO 中，一次输入一个 32 位字。

为了满足不同的数据大小，通过配置 CRYP_CR 寄存器中的 DATATYPE 位，数据可在处理之后进行交换。更多详细信息，请参见*第 35.3.16 节：CRYP 数据寄存器和数据交换*。

写入 CRYP_DIN 寄存器时，数据会压入输入 FIFO。

- 如果 CRYPEN = 1，则在 DES/TDES 模式下至少有两个 32 位字（或者 AES 模式下至少有四个字）已压入输入 FIFO 中，并且输出 FIFO 中至少有两个字的自由空间时（或者AES 模式下，至少有四个字的自由空间），CRYP 引擎会启动加密或解密过程。

读取 CRYP_DIN 寄存器时：

- 如果 CRYPEN = 0，则 FIFO 会弹出数据，之后，返回输入 FIFO 中的数据，从最早的数据（首先读取）到最新的数据（最后读取）。在执行每次读取操作前必须检查 IFEM 标志，以确保 FIFO 非空。
- 如果 CRYPEN = 1，则会返回一个未定义值。

注： *读取 CRYP_DIN 寄存器一次或多次之后，在处理新数据之前必须通过将 FFLUSH 位置 1 来刷新 FIFO。*

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

位 31:0 **DATAIN**：数据输入 (Data input)

读取时，FIFO 会弹出数据（返回最后写入的值），如果 CRYPEN=0 则会返回相应值。如果 CRYPEN=1，则 DATAIN 寄存器会返回一个未定义值。

写入时，会将当前寄存器内容压入 FIFO 内。

35.6.4 CRYP 数据输出寄存器 (CRYP_DOUT)

CRYP data output register

偏移地址：0x0C

复位值：0x0000 0000

CRYP_DOUT 寄存器为数据输出寄存器。该寄存器为 32 位宽度的只读寄存器。最多可接收来自输出 FIFO 的四个 64 位 (TDES) 或两个 128 位 (AES) 明文（加密时）或密文（解密时）块，一次接收一个 32 位字。

为了满足不同的数据大小，通过配置 CRYP_CR 寄存器中的 DATATYPE 位，数据可在处理之后进行交换。更多详细信息，请参见*第 35.3.16 节：CRYP 数据寄存器和数据交换*。

读取 CRYP_DOUT 寄存器时，会返回最后输入输出 FIFO 的数据（由读指针指定）。

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

位 31:0 **DATAOUT**：数据输出 (Data output)

读取时，返回输出 FIFO 的内容（读指针所指向的内容），否则会返回一个未定义的值。

写入时，无影响。

35.6.5 CRYP DMA 控制寄存器 (CRYP_DMACR)

CRYP DMA control register

偏移地址：0x10

复位值：0x0000 0000

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

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

位 31:2 保留，必须保持复位值

位 1 **DOEN**：DMA 输出使能 (DMA output enable)

该位置 1 时，外设会在输出数据阶段自动生成 DMA 请求。

0：禁止用于传出数据传输的 DMA

1：使能用于传出数据传输的 DMA

位 0 **DIEN**：DMA 输入使能 (DMA input enable)

该位置 1 时，外设会在输入数据阶段自动生成 DMA 请求。

0：禁止用于传入数据传输的 DMA

1：使能用于传入数据传输的 DMA

35.6.6 CRYP 中断屏蔽置位/清零寄存器 (CRYP_IMSCR)

CRYP interrupt mask set/clear register

偏移地址：0x14

复位值：0x0000 0000

CRYP_IMSCR 寄存器为中断屏蔽置位/清零寄存器。该寄存器为读/写寄存器。执行读取操作时，此寄存器会给出应用于相关中断屏蔽的当前值。向该特定位写入 1 时会将屏蔽置 1，从而使能要读取的中断。向该位写入 0 时会清除相应的屏蔽。外设复位后，所有位均清零。

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

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

位 31:2 保留，必须保持复位值

位 1 **OUTIM**：输出 FIFO 服务中断屏蔽 (Output FIFO service interrupt mask)
0：屏蔽输出 FIFO 服务中断
1：不屏蔽输出 FIFO 服务中断

位 0 **INIM**：输入 FIFO 服务中断屏蔽 (Input FIFO service interrupt mask)
0：屏蔽输入 FIFO 服务中断
1：不屏蔽输入 FIFO 服务中断

35.6.7 CRYP 原始中断状态寄存器 (CRYP_RISR)

CRYP raw interrupt status register

偏移地址：0x18

复位值：0x0000 0001

CRYP_RISR 寄存器为原始中断状态寄存器。该寄存器为只读寄存器。执行读操作时，该寄存器会给出相应中断的当前原始状态，即中断信息（不考虑 CRYP_IMSCR 屏蔽）。写操作无影响。

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

位 31:2 保留，必须保持复位值

位 1 **OUTRIS**：输出 FIFO 服务原始中断状态 (Output FIFO service raw interrupt status)
该位提供输出 FIFO 中断信息（不考虑相应 CRYP_IMSCR 屏蔽）。
0：原始中断未挂起
1：原始中断挂起

位 0 **INRIS**：输入 FIFO 服务原始中断状态 (Input FIFO service raw interrupt status)
该位提供输入 FIFO 中断信息（不考虑相应 CRYP_IMSCR 屏蔽）。
0：原始中断未挂起
1：原始中断挂起

35.6.8 CRYP 屏蔽中断状态寄存器 (CRYP_MISR)

CRYP masked interrupt status register

偏移地址：0x1C

复位值：0x0000 0000

CRYP_RISR 寄存器为屏蔽中断状态寄存器。该寄存器为只读寄存器。执行读操作时，该寄存器会给出相应中断的当前屏蔽状态，即中断信息（考虑 CRYP_IMSCR 屏蔽）。写操作无影响。

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

位 31:2 保留，必须保持复位值

位 1 **OUTMIS**：输出 FIFO 服务屏蔽中断状态 (Output FIFO service masked interrupt status)

该位提供输出 FIFO 中断信息（不考虑相应 CRYP_IMSCR 屏蔽）。

0：中断未挂起

1：中断挂起

位 0 **INMIS**：输入 FIFO 服务屏蔽中断状态 (Input FIFO service masked interrupt status)

该位提供输入 FIFO 中断信息（不考虑相应 CRYP_IMSCR 屏蔽）。

0：中断未挂起

1：CRYPEN = 1 时中断挂起

35.6.9 CRYP 密钥寄存器 0L (CRYP_K0LR)

CRYP key register 0L

偏移地址：0x20

复位值：0x0000 0000

CRYP 密钥寄存器包含加密密钥。

- 在 DES/TDES 模式下，密钥为 64 位二进制值（按从左到右编号，最左位为位 1），分别命名为 K1、K2 和 K3（不使用 K0）。每个密钥由 56 个信息位和 8 个奇偶校验位组成。
- 在 AES 模式下，可将密钥视为一个 128 位、192 位或 256 位长的序列，$K_0K_1K_2...K_{127/191/255}$。AES 密钥按如下方式输入寄存器中：
  - 对于 AES-128：$k_0..k_{127}$ 对应 $b_{127}..b_0$（不使用 $b_{255}..b_{128}$）
  - 对于 AES-192：$k_0..k_{191}$ 对应 $b_{191}..b_0$（不使用 $b_{255}..b_{192}$）
  - 对于 AES-256：$k_0..k_{255}$ 对应 $b_{255}..b_0$

在所有情况下，密钥位 K_0 都是 CRYP 内部存储器的最左位，寄存器位 b_0 都是相应 CRYP_KxLR 密钥寄存器的最右位。

更多信息，请参见*第 35.3.17 节：CRYP 密钥寄存器*。

注： *当加密处理器繁忙时（CRYP_SR 寄存器中的位 BUSY = 1），会忽略对这些寄存器的写访问。*

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
K255	K254	K253	K252	K251	K250	K249	K248	K247	K246	K245	K244	K243	K242	K241	K240
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
K239	K238	K237	K236	K235	K234	K233	K232	K231	K230	K229	K228	K227	K226	K225	K224
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

位 x - 224 **Kx**：AES 密钥位 (AES key bit x)（x = 224 到 255）
注： 该寄存器不适用于 DES 模式

35.6.10 CRYP 密钥寄存器 0R (CRYP_K0RR)

CRYP key register 0R

偏移地址：0x24

复位值：0x0000 0000

有关详细信息，请参见*第 35.6.9 节：CRYP 密钥寄存器 0L (CRYP_K0LR)*。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
K223	K222	K221	K220	K219	K218	K217	K216	K215	K214	K213	K212	K211	K210	K209	K208
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
K207	K206	K205	K204	K203	K202	K201	K200	K199	K198	K197	K196	K195	K194	K193	K192
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

位 x - 192 **Kx**：AES 密钥位 (AES key bit x)（x = 192 到 223）
注： 该寄存器不适用于 DES 模式

35.6.11 CRYP 密钥寄存器 1L (CRYP_K1LR)

CRYP key register 1L

偏移地址：0x28

复位值：0x0000 0000

有关详细信息，请参见*第 35.6.9 节：CRYP 密钥寄存器 0L (CRYP_K0LR)*。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
K191	K190	K189	K188	K187	K186	K185	K184	K183	K182	K181	K180	K179	K178	K177	K176
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
K175	K174	K173	K172	K171	K170	K169	K168	K167	K166	K165	K164	K163	K162	K161	K160
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

位 x - 160 **Kx**：AES 密钥位 (AES key bit x)（x = 160 到 191）
在 DES 模式下，K192 对应于密钥 K1 位 1，K160 对应于密钥 K1 位 32。

35.6.12 CRYP 密钥寄存器 1R (CRYP_K1RR)

CRYP key register 1R

偏移地址：0x2C

复位值：0x0000 0000

有关详细信息，请参见*第 35.6.9 节：CRYP 密钥寄存器 0L (CRYP_K0LR)*。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
K159	K158	K157	K156	K155	K154	K153	K152	K151	K150	K149	K148	K147	K146	K145	K144
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
K143	K142	K141	K140	K139	K138	K137	K136	K135	K134	K133	K132	K131	K130	K129	K128
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

位 x - 128 **Kx**：AES 密钥位 (AES key bit x)（x = 128 到 159）
在 DES 模式下，K159 对应于密钥 K1 位 33，K128 对应于密钥 K1 位 64。

35.6.13 CRYP 密钥寄存器 2L (CRYP_K2LR)

CRYP key register 2L

偏移地址：0x30

复位值：0x0000 0000

有关详细信息，请参见*第 35.6.9 节：CRYP 密钥寄存器 0L (CRYP_K0LR)*。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
K127	K126	K125	K124	K123	K122	K121	K120	K119	K118	K117	K116	K115	K114	K113	K112
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
K111	K110	K109	K108	K107	K106	K105	K104	K103	K102	K101	K100	K99	K98	K97	K96
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

位 x - 96 **Kx**：AES 密钥位 (AES key bit x)（x = 96 到 127）
在 DES 模式下，K127 对应于密钥 K2 位 1，K96 对应于密钥 K2 位 32。

35.6.14 CRYP 密钥寄存器 2R (CRYP_K2RR)

CRYP key register 2R

偏移地址：0x34

复位值：0x0000 0000

有关详细信息，请参见*第 35.6.9 节：CRYP 密钥寄存器 0L (CRYP_K0LR)*。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
K95	K95	K93	K92	K91	K90	K89	K88	K87	K86	K85	K84	K83	K82	K81	K80
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
K79	K78	K77	K76	K75	K74	K73	K72	K71	K70	K69	K68	K67	K66	K65	K64
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

位 x - 64 **Kx**：AES 密钥位 (AES key bit x)（x = 64 到 95）
在 DES 模式下，K95 对应于密钥 K2 位 33，K64 对应于密钥 K2 位 64。

35.6.15 CRYP 密钥寄存器 3L (CRYP_K3LR)

CRYP key register 3L

偏移地址：0x38

复位值：0x0000 0000

有关详细信息，请参见*第 35.6.9 节：CRYP 密钥寄存器 0L (CRYP_K0LR)*。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
K63	K62	K61	K60	K59	K58	K57	K56	K55	K54	K53	K52	K51	K50	K49	K48
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
K47	K46	K45	K44	K43	K42	K41	K40	K39	K38	K37	K36	K35	K34	K33	K32
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

位 x - 32 **Kx**：AES 密钥位 (AES key bit x)（x = 32 到 63）

在 DES 模式下，K63 对应于密钥 K3 位 1，K32 对应于密钥 K3 位 32。

35.6.16 CRYP 密钥寄存器 3R (CRYP_K3RR)

CRYP key register 3R

偏移地址：0x3C

复位值：0x0000 0000

有关详细信息，请参见*第 35.6.9 节：CRYP 密钥寄存器 0L (CRYP_K0LR)*。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
K31	K30	K29	K28	K27	K26	K25	K24	K23	K22	K21	K20	K19	K18	K17	K16
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
K15	K14	K13	K12	K11	K10	K9	K8	K7	K6	K5	K4	K3	K2	K1	K0
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

位 x **Kx**：AES 密钥位 (AES key bit x)（x = 0 到 31）

在 DES 模式下，K31 对应于密钥 K3 位 33，K0 对应于密钥 K3 位 64。

35.6.17 CRYP 初始化向量寄存器 0L (CRYP_IV0LR)

CRYP initialization vector register 0L

偏移地址：0x40

复位值：0x0000 0000

CRYP_IV0...1(L/R)R 是初始化向量专用的左字和右字寄存器（DES/TDES 模式下为 64 位，AES 模式下为 128 位）。更多信息，请参见*第 35.3.18 节：CRYP 初始化向量寄存器*。

IV0 是初始化向量的最左位，而 IV63（DES，TDES）或 IV127 (AES) 是初始化向量的最右位。IV1(L/R)R 仅用于 AES 模式。仅 CRYP_IV0(L/R) 用于 DES/TDES 模式。

注： *当加密处理器繁忙时（CRYP_SR 寄存器中的位 BUSY = 1），会忽略对这些寄存器的写访问。*

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IV0	IV1	IV2	IV3	IV4	IV5	IV6	IV7	IV8	IV9	IV10	IV11	IV12	IV13	IV14	IV15
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IV16	IV17	IV18	IV19	IV20	IV21	IV22	IV23	IV24	IV25	IV26	IV27	IV28	IV29	IV30	IV31
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31 - x **IVx**：初始化向量位 x (Initialization vector bit x)（x = 0 到 31）

35.6.18 CRYP 初始化向量寄存器 0R (CRYP_IV0RR)

CRYP initialization vector register 0R

偏移地址：0x44

复位值：0x0000 0000

有关详细信息，请参见*第 35.6.17 节：CRYP 初始化向量寄存器 0L (CRYP_IV0LR)*。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IV32	IV33	IV34	IV35	IV36	IV37	IV38	IV39	IV40	IV41	IV42	IV43	IV44	IV45	IV46	IV47
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IV48	IV49	IV50	IV51	IV52	IV53	IV54	IV55	IV56	IV57	IV58	IV59	IV60	IV61	IV62	IV63
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 63 - x **IVx**：初始化向量位 x (Initialization vector bit x)（x = 32 到 63）

35.6.19 CRYP 初始化向量寄存器 1L (CRYP_IV1LR)

CRYP initialization vector register 1L

偏移地址：0x48

复位值：0x0000 0000

有关详细信息，请参见*第 35.6.17 节：CRYP 初始化向量寄存器 0L (CRYP_IV0LR)*。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IV64	IV65	IV66	IV67	IV68	IV69	IV70	IV71	IV72	IV73	IV74	IV75	IV76	IV77	IV78	IV79
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IV80	IV81	IV82	IV83	IV84	IV85	IV86	IV87	IV88	IV89	IV90	IV91	IV92	IV93	IV94	IV95
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 95 - x **IVx**：初始化向量位 x (Initialization vector bit x)（x = 64 到 95）

注：　该寄存器不适用于 DES 模式

35.6.20 CRYP 初始化向量寄存器 1R (CRYP_IV1RR)

CRYP initialization vector register 1R

偏移地址：0x4C

复位值：0x0000 0000

有关详细信息，请参见*第 35.6.17 节：CRYP 初始化向量寄存器 0L (CRYP_IV0LR)*。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IV96	IV97	IV98	IV99	IV100	IV101	IV102	IV103	IV104	IV105	IV106	IV107	IV108	IV109	IV110	IV111
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IV112	IV113	IV114	IV115	IV116	IV117	IV118	IV119	IV120	IV121	IV122	IV123	IV124	IV125	IV126	IV127
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 127 - x **IVx**：初始化向量位 x (Initialization vector bit x)（x = 96 到 127）

注： 该寄存器不适用于 DES 模式

35.6.21 CRYP 上下文交换 GCM-CCM 寄存器 (CRYP_CSGCMCCMxR)

CRYP context swap GCM-CCM registers

偏移地址：0x050 + x* 0x4（x=0 到 7）

复位值：0x0000 0000

在选择 GCM/GMAC 或 CCM 算法后，这些寄存器含有 CRYP 处理器的全部内部寄存器状态。如果高优先级的任务需要使用加密处理器，而该处理器正在执行其他任务，则需要使用上述寄存器执行上下文交换。

出现此类事件时，需要对 CRYP_CSGCMCCM0..7R 和 CRYP_CSGCM0..7R（GCM/GMAC 模式）或 CRYP_CSGCMCCM0..7R（CCM 模式）寄存器进行读操作，并且需要将读出的值保存到系统存储器空间。之后，可使用加密处理器执行高优先级的任务。完成加密计算之后，可从存储器中读出保存的上下文并将其写回相应的上下文交换寄存器。

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

位 31:0 **CRYP_CSGCMCCMxR**：用于 GCM、GMAC 和 CCM 模式的 CRYP 内部状态寄存器 (CRYP internal state registers for GCM, GMAC and CCM modes)

注： 该寄存器不用于 DES/TDES 或其他未指定的 AES 模式

35.6.22 CRYP 上下文交换 GCM 寄存器 (CRYP_CSGCMxR)

CRYP context swap GCM registers

偏移地址：0x070 + x* 0x4（x=0 到 7）

复位值：0x0000 0000

有关详细信息，请参见*第 35.6.21 节：CRYP 上下文交换 GCM-CCM 寄存器 (CRYP_CSGCMCCMxR)*。

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

位 31:0 **CRYP_CSGCMxR**：用于 GCM 和 GMAC 模式的 CRYP 内部状态寄存器 (CRYP internal state registers for GCM and GMAC modes)

注：　该寄存器不用于 DES/TDES 或其他未指定的 AES 模式

35.6.23 CRYP 寄存器映射

表 276. CRYP 寄存器映射和复位值

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00 0x00	CRYP_CR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.				ALGOMODE[3]	Res.	GCM_CCMPH		CRYPEN	FFLUSH	Res.	Res.	Res.	Res.	KEYSIZE		DATATYPE		ALGOMODE[2:0]			ALGODIR	Res.	Res.
	Reset value									0	0	0	0	0		0	0	0	0					0	0	0	0	0	0	0	0		
0x04	CRYP_SR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BUSY	OFFU	OFNE	IFNF	IFEM
	Reset value																												0	0	0	1	1
0x08	CRYP_DIN	DATAIN																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0C	CRYP_DOUT	DATAOUT																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x10	CRYP_DMACR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DOEN	DIEN
	Reset value																															0	0
0x14	CRYP_IMSCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OUTIM	INIM
	Reset value																															0	0
0x18	CRYP_RISR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OUTRIS	INRIS
	Reset value																															0	1
0x1C	CRYP_MISR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OUTMIS	INMIS
	Reset value																															0	0
0x20	CRYP_K0LR	CRYP_K0LR																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x24	CRYP_K0RR	CRYP_K0RR																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
... ...																																	
0x38	CRYP_K3LR	CRYP_K3LR																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x3C	CRYP_K3RR	CRYP_K3RR																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x40	CRYP_IV0LR	CRYP_IV0LR																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

表 276. CRYP 寄存器映射和复位值（续）

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x44	CRYP_IV0RR	CRYP_IV0RR																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x48	CRYP_IV1LR	CRYP_IV1LR																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x4C	CRYP_IV1RR	CRYP_IV1RR																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x50	CRYP_CSGCMCCM0R	CRYP_CSGCMCCM0R																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x54	CRYP_CSGCMCCM1R	CRYP_CSGCMCCM1R																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x58	CRYP_CSGCMCCM2R	CRYP_CSGCMCCM2R																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x5C	CRYP_CSGCMCCM3R	CRYP_CSGCMCCM3R																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x60	CRYP_CSGCMCCM4R	CRYP_CSGCMCCM4R																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x64	CRYP_CSGCMCCM5R	CRYP_CSGCMCCM5R																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x68	CRYP_CSGCMCCM6R	CRYP_CSGCMCCM6R																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x6C	CRYP_CSGCMCCM7R	CRYP_CSGCMCCM7R																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x70	CRYP_CSGCM0R	CRYP_CSGCM0R																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x74	CRYP_CSGCM1R	CRYP_CSGCM1R																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x78	CRYP_CSGCM2R	CRYP_CSGCM2R																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x7C	CRYP_CSGCM3R	CRYP_CSGCM3R																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x80	CRYP_CSGCM4R	CRYP_CSGCM4R																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x84	CRYP_CSGCM5R	CRYP_CSGCM5R																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

表 276. CRYP 寄存器映射和复位值（续）

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x88	CRYP_CSGCM6R	CRYP_CSGCM6R																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x8C	CRYP_CSGCM7R	CRYP_CSGCM7R																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

36 散列处理器 (HASH)

36.1 前言

散列处理器完全兼容安全散列算法（SHA-1、SHA-224 和 SHA-256）、MD5（消息摘要算法 5）散列算法和适合多种应用的 HMAC（密钥散列消息认证码）散列算法。HMAC 算法通过散列函数来对消息进行验证。HMAC 算法需要两次调用 SHA-1、SHA-224、SHA-256 或 MD5 散列函数。

对长达 ($2^{64} – 1$) 位的消息，散列处理器计算消息摘要（SHA-1 算法为 160 位，SHA-256 算法为 256 位，SHA-224 算法为 224 位，MD5 算法为 128 位）。

36.2 散列主要特性

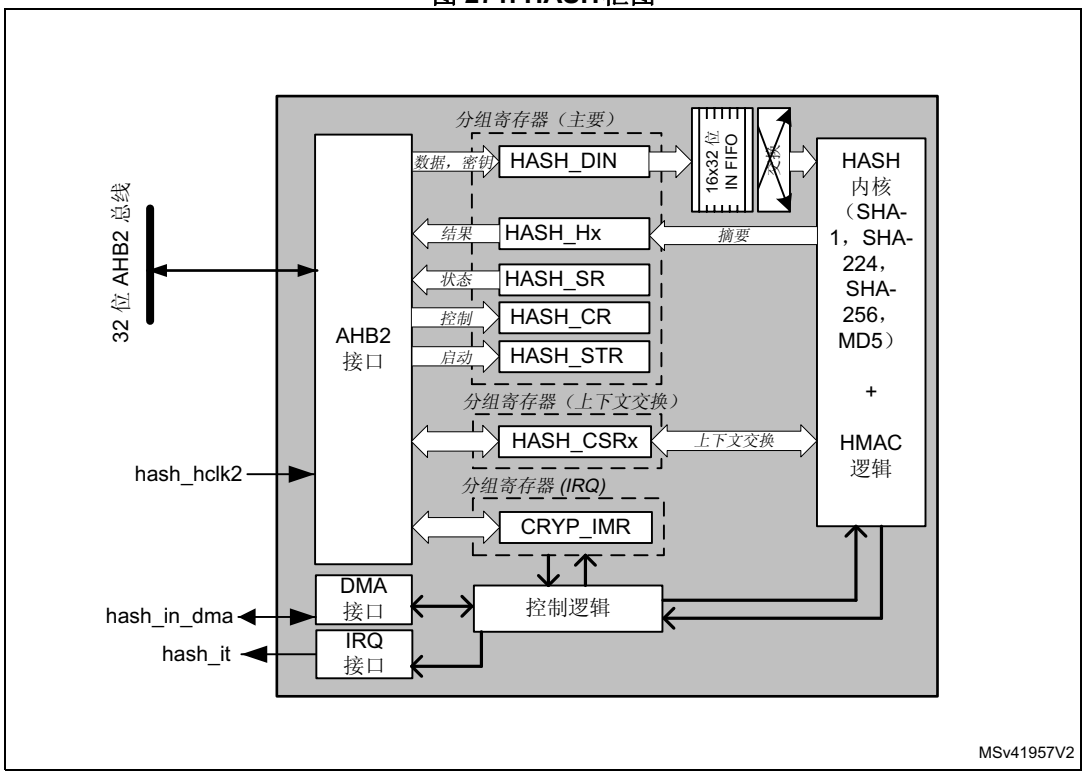
- 适合于数据验证应用，符合以下标准：
  - FIPS PUB 180-1（联邦信息处理标准出版物 180-1）*安全散列标准规范* (SHA-1)
  - FIPS PUB 180-2（联邦信息处理标准出版物 180-2）*安全散列标准规范*（SHA-224 和 SHA-256）
  - 互联网工程任务组 (IETF) 征求意见 RFC 1321 *MD5 消息摘要算法*
  - 互联网工程任务组 (IETF) 征求意见 RFC 2104 *HMAC：密钥散列消息认证*
- 连续消息块中摘要的对应 32 位字添加到彼此之中，以构成整个消息的摘要
  - 可自动进行 32 位字交换，以兼容输入位串的内部小端模式表示法
  - 支持的字交换：位、字节、半字和 32 位字
- 可自动填充来补足输入位串，从而适应 512 位（16×32 位）的摘要最小块大小
- 单个 32 位输入寄存器，它与十六个 32 位字的内部输入 FIFO 相关联，对应于一个块的大小
- 快速计算 SHA-1、SHA-224、SHA-256 以及 MD5
  - 使用 SHA-1（或 SHA-256）算法处理一个 512 位数据块需要 82（或 66）个时钟周期
  - 使用 MD5 算法处理一个 512 位数据块需要 66 个时钟周期
- AHB 从外设，仅支持 32 位字访问（否则会产生 AHB 错误）
- 8×32 位字（H0 到 H7）用于输出消息摘要
- 数据流自动控制，支持使用一个通道进行直接存储器访问 (DMA)。支持固定 4 个突发
- 每 32 位字可中断一次消息摘要计算
  - 可重载摘要寄存器
  - 散列计算挂起/恢复机制，包括使用 DMA

36.3 散列功能说明

36.3.1 HASH 框图

图 271 显示了散列处理器的框图。

图 271. HASH框图



36.3.2 HASH 内部信号

表 277 列出了 HASH 级而非产品级（焊盘上）所提供的内部信号，对这些信号有所了解会很有用。

表 277. HASH 内部输入/输出信号

信号名称	信号类型	说明
hash_hclk2	数字输入	AHB2 总线时钟
hash_it	数字输出	散列处理器全局中断请求
hash_in_dma	数字输入/输出	DMA 突发请求/确认

36.3.3 关于安全散列算法

散列处理器完全兼容由 FIPS PUB 180-1 标准 (SHA1)、FIPS PUB 180-2 标准（SHA-224、SHA-256）和 IETF RFC1321 出版物 (MD5) 定义的安全散列算法。

对于每个算法，HASH 计算消息或数据文件的压缩表示。更具体来讲，输入中提供的任何消息长度低于 2^{64} 位时，SHA-1、SHA-224、SHA-256 和 MD5 处理内核将分别生成一个 160 位、224 位、256 位和 128 位的输出位串，称为消息摘要。然后可以通过数字签名算法来处理此消息摘要，以便生成或验证消息的签名。

对消息摘要而不是对消息签名通常可提高处理的效率，因为消息摘要通常比消息要小得多。数字签名的验证程序和数字签名的创建程序必须使用相同散列算法。

SHA-1、SHA-224 和 SHA-256 以及 MD5 安全可靠，因为要找出某个与给定消息摘要对应的消息，或找出两个生成相同消息摘要的不同消息，在计算层面无法实现。对传输中的消息进行任何更改都极有可能产生不同的消息摘要，从而导致签名验证失败。

36.3.4 消息数据馈送

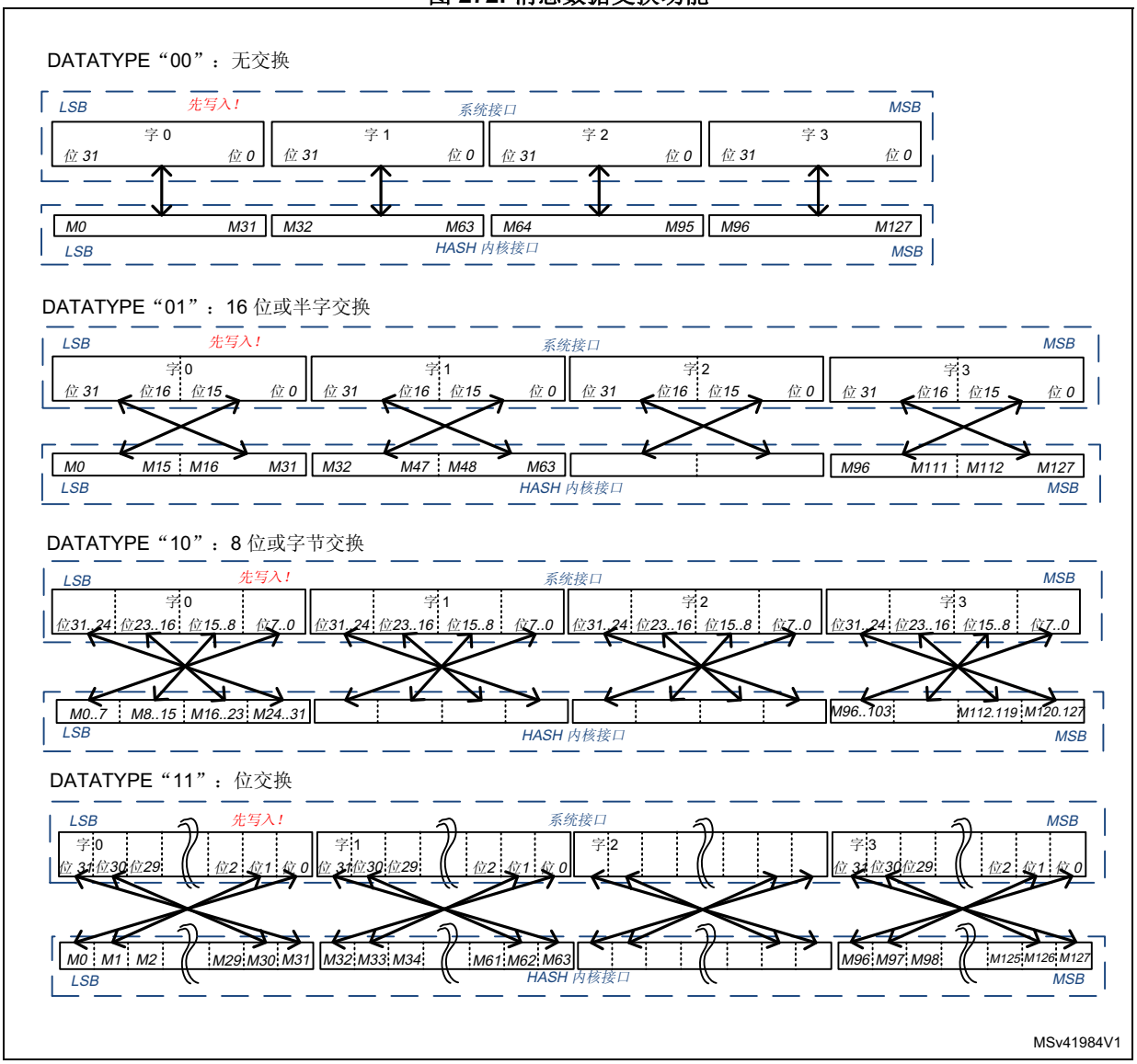
要由 HASH 处理的消息（或数据文件）应视为位串。根据 FIPS PUB 180-1 和 180-2 标准，该消息位串从左到右增长（十六进制字采用“大端”约定来表示），从而使得在每个字内，最高有效位存储在最左侧位的位置上。以位串表示为“01100001 01100010 01100011”的消息串“abc”为例，其 32 位字表示为 0x00636261，8 位字表示为 0x61626300。

通过将数据写入到 HASH_DIN 寄存器，数据将一次性输入到 HASH 的一个 32 位字中。每次在寄存器中写入新数据时，HASH_DIN 寄存器的当前内容将传输到 16 字输入 FIFO (IN FIFO)。因此，HASH_DIN 和输入 FIFO 构成一个长度为十七个 32 位字的 FIFO（名为 IN 缓冲器）。

根据要处理的数据类型（例如，当数据是 ASCII 文本流时的字节交换），必须先对从输入 FIFO 读取的数据进行位、字节、半字或无交换操作，然后再将其输入小端散列处理内核。*图 272* 显示了如何根据 HASH 控制寄存器 (HASH_CR) 中的 DATATYPE 位域，使用由驱动程序弹出到 IN FIFO 中的一个 32 位字构建散列处理内核 32 位数据块 M0...31。

当禁用位交换时（DATATYPE=“00”），HASH_DIN 数据字节序如下所述：消息的最低有效位必须在输入到散列处理器的第一个字中的 MSB 位置，位串的第 32 位必须在输入到散列处理器的第二个字中的 MSB 位置，依此类推。

图 272. 消息数据交换功能



36.3.5 消息摘要计算

当计算消息摘要时，散列处理器会按顺序处理 512 位块。这样，DMA 或 CPU 每次将 16 x 32 位字（= 512 位）写入到散列处理器时，散列处理器将自动开始计算消息摘要。此操作称为“部分摘要计算”。

如*第 36.3.4 节：消息数据馈送*中所述，待处理的消息将一次性输入到 HASH 32 位字中，从而写入 HASH_DIN 寄存器以填充输入 FIFO。应用程序应使用以下序列对该数据执行散列计算。

1. 使用 HASH_CR 寄存器初始化散列处理器：
   - 使用 ALGO 字段选择正确的算法。如果需要，可使用 HASH_CR 中的 DATATYPE 位域为消息输入字编程正确的交换操作。
   - 如果已选择 HMAC 模式，则设置 MODE = 1 并使用 LKEY 选择密钥长度。
   - 如果最后一个字中的有效位数不是 32 位，则更新 NBLW 以定义最后一个字中的有效位数。在这种情况下，HASH 可以应用自动填充。
2. 通过将 HASH_CR 中的 INIT 位置 1 来完成初始化。如果通过 DMA 传输数据，也将 DMAE 位置 1。

注意： 编程步骤 2 时，务必事先或同时设置正确的配置值（ALGO、DATATYPE、HMAC 模式、密钥长度、NBLW）。

3. 开始通过写入 HASH_DIN 寄存器来填充数据，自动通过 DMA 传输数据的情况除外。请注意，只有当块的最后一个值输入 IN FIFO 后，才能开始处理块。部分或最终摘要计算的处理方式取决于数据馈送到处理器的方式：
   a) 通过软件填充数据时：
   - 当软件将额外的一个字（实际上是下一个块的第一个字）写入到 HASH_DIN 寄存器时，将触发部分摘要计算。一旦处理器再次准备就绪（HASH_SR 中的 DINIS = 1），软件即可将新数据写入 HASH_DIN。这种机制可防止 HASH 引入等待状态。
   - 当输入最后一个块并且软件向 DCAL 位写入 1 时，将触发最终摘要计算。如果消息长度不是 512 位的整数倍，则必须在写入 DCAL 位之前写入 HASH_STR 寄存器中的 NBLW 字段（有关详细信息，请参见*第 36.3.6 节*）。

   b) 通过 DMA 以单次 DMA 传输（MDMAT 位 =“0”）填充数据时：
   - 每次 FIFO 已满时，将自动触发部分摘要计算。
   - 当最后一个块传输到 HASH_DIN 寄存器后，将自动触发最终摘要计算（硬件将 DCAL 位置 1）。如果消息长度不是 512 位的整数倍，则必须在使能 DMA 之前写入 HASH_STR 寄存器中的 NBLW 字段（有关详细信息，请参见*第 36.3.6 节*）。

   c) 当使用多次 DMA 传输（MDMAT 位=“1”）填充数据时：
   - 可针对单次 DMA 传输触发部分摘要计算。但是，当最后一个块传输到 HASH_DIN 寄存器后，不会自动触发最终摘要计算（硬件不将 DCAL 位置 1）。这使得散列处理器能够接收新的 DMA 传输（作为此摘要计算的一部分）。为了启动最终摘要计算，软件必须在最后一次 DMA 传输之前将 MDMAT 位置 0，以便如同针对单次 DMA 传输（请参见之前的说明）一样触发最终摘要计算。
4. 计算完成后，可以按照*表 278* 中所述从输出寄存器读取摘要。

表 278. 散列处理器输出

算法	有效输出寄存器	最高有效位	摘要大小（位）
MD5	HASH_H0 到 HASH_H3	HASH_H0[31]	128
SHA-1	HASH_H0 到 HASH_H4	HASH_H0[31]	160
SHA-224	HASH_H0 到 HASH_H6	HASH_H0[31]	224
SHA-256	HASH_H0 到 HASH_H7	HASH_H0[31]	256

关于 HMAC 详细指令的更多信息，请参见*第 36.3.7 节：HMAC 运算*。

36.3.6 消息填充

概述

当计算消息的压缩表示时，循环执行将数据馈送到散列处理器（每 512 位块执行一次自动部分摘要计算）的过程，直到原始消息的最后几位被写入 HASH_DIN 寄存器。

因为消息的长度（位数）可以是任何整数值，因此写入到散列处理器的最后一个字的有效位数在 1 到 32 之间。这个最后一个字的有效位数必须写到 HASH_STR 突寄存器的 NBLW 位段中，以便消息填充能够在最终消息摘要计算之前得到正确处理。

填充处理

*第 36.3.5 节：消息摘要计算*中介绍了使能或禁止 DMA 时的详细填充序列。

填充示例

根据联邦信息处理标准 PUB 180-1 和 PUB 180-2 规定，消息填充包括添加一个“1”，后跟 *k* 个“0”以及一个 64 位整数（其数值等于消息长度 *L*（位数））。经过这三个填充操作后生成的消息长度为 *L* + 1 + *k* + 64（512 位的整数倍）。

对于散列处理器，“1”将添加到 HASH_DIN 寄存器中写入的最后一个字，位位置由 NBLW 位域定义，而其余更高的位将被清零（“0”）。

FIPS PUB180-2 提供的示例

我们假定原始消息是采用 ASCII 二进制编码格式的“abc”，长度 L=24：

```
字节 0    字节 1    字节 2    字节 3
01100001 01100010 01100011 UUUUUUUU
<-- 写入 HASH_DIN 的第一个字 -->
```

NBLW 必须以值 24 加载：“1”在位串中的位 24 处追加（在以上位串中从左到右开始计数），这对应于 HASH_DIN 寄存器中的位 31（小端模式约定）：

```
01100001 01100010 01100011 1UUUUUUU
```

由于 L = 24，因此以上位串中的位数是 25，并将追加 423 个“0”位，现在为 448 位。

这将生成十六进制（大端模式的字节字）：

```
61626380 00000000 00000000 00000000
00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000
00000000 00000000
```

将追加双字格式的消息长度值 L，即 00000000 00000018。因此，最后填充的消息为十六进制（大端模式的字节字）：

```
61626380 00000000 00000000 00000000
00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000018
```

如果散列处理器被编程为在 HASH_DIN 输入寄存器（HASH_CR 中的 DATATYPE = 10）中交换字节，则必须按照以下序列输入上述消息：

1. 将 0xUU636261 写入到 HASH_DIN 寄存器（其中的“U”意味着无关）。
2. 将 0x18 写入到 HASH_STR 寄存器（写入到 HASH_DIN 寄存器的最后一个字中的有效位数是 24，因为原始消息长度为 24 位）。
3. 将 0x10 写入到 HASH_STR 寄存器以启动消息填充（如上文所述）以及随后执行摘要计算。
4. 完成散列计算后，HASH_Hx 寄存器 (x = 0...4) 中的消息摘要可供 SHA-1 算法使用。对于此 FIPS 示例，预期值如下：

```
HASH_H0 = 0xA9993E36
HASH_H1 = 0x4706816A
HASH_H2 = 0xBA3E2571
HASH_H3 = 0x7850C26C
HASH_H4 = 0x9CD0D89D
```

36.3.7 HMAC 运算

概述

根据互联网工程任务组 *RFC2104，HMAC：密钥散列消息认证*规定，HMAC 算法以不可逆方式将正在处理的消息与用户所选的密钥进行绑定，从而用于消息验证。该算法由两个嵌套的散列运算组成：

```
HMAC(message) = Hash((key | pad) XOR [0x5C]n
              | Hash((key | pad) XOR [0x36]n | message))
```

其中：

- $[X]_n$ 表示 X 重复 *n* 次，其中的 *n* 等于底层散列函数数据块的大小（对于 SHA-1、SHA224、SHA-256、MD5 散列算法是 512 位，即 n = 64）。
- pad 是将密钥扩展到上面定义的长度 *n* 所需的零序列。如果密钥长度大于 *n*，应用程序应首先使用 Hash() 函数对密钥进行散列处理，然后将得到的字节串用作 HMAC 的实际密钥。
- | 表示串联运算符

HMAC 处理

计算 HMAC 需要下列四个步骤：

1. 当 MODE 位为 1 且 ALGO 位已根据所需算法设好对应的值时，通过向 INIT 位写入 1 对块进行初始化。如果正在使用的密钥的长度超过 64 字节，则还必须将 LKEY 位置 1。在这种情况下，根据 HMAC 规范的要求，散列处理器将使用密钥的散列代替真实密钥。
2. 必须将用于内部散列函数的密钥提供给散列处理器：

   密钥加载操作与消息位串加载操作遵循相同的机制，即将密钥数据写入 HASH_DIN 并通过写入 HASH_STR 寄存器来完成传输。

注： *有关字节序的详细信息，可参见第 36.3.4 节：消息数据馈送。*

3. 输入了最后一个密钥字并且计算开始后，散列处理器便开始生成内部密钥材料。一旦完成此操作，即可接受消息位字符串，如*第 36.3.4 节：消息数据馈送*中所述。
4. 在最后一轮散列之后，散列处理器将返回“就绪”以表明其可以接收用于外部散列函数的密钥（通常，此密钥与用于内部散列函数的密钥相同）。在输入了密钥的最后一个字并且计算开始后，HMAC 结果存储于 HASH_H0...HASH_H7 寄存器。

注： *HMAC 本原的计算延迟取决于密钥和消息的长度，如第 36.5 节：HASH 处理时间中所述。*

HMAC 示例

下面是一个由 NIST 指定的 HMAC SHA-1 算法示例（HASH_CR 中的 ALGO =“00”且 MODE =“1”）。

我们假定原始消息是采用 ASCII 二进制编码格式的“Sample message for keylen=blocklen”（keylen=blocklen 时的示例消息），长度 L = 34 字节。如果 HASH 编程为无交换模式（HASH_CR 中的 DATATYPE = 00），则必须将以下数据按顺序加载至 HASH_DIN 寄存器：

1. **由 NIST 指定的内部散列密钥**输入（长度 = 64，即不填充）。由于密钥长度 = 64，HASH_CR 寄存器中的 LKEY 位置 0

```
00010203 04050607 08090A0B 0C0D0E0F 10111213 14151617
18191A1B 1C1D1E1F 20212223 24252627 28292A2B 2C2D2E2F
30313233 34353637 38393A3B 3C3D3E3F
```

2. **消息**输入（长度 = 34，即需要填充）。必须将 HASH_STR 设置为 0x20，以开始消息填充和内部散列计算（将“U”视为无关）

```
53616D70 6C65206D 65737361 67652066 6F72206B 65796C65
6E3D626C 6F636B6C 656EUUUU
```

3. **外部散列密钥**输入（长度 = 64，即不填充）。在此输入与内部散列密钥相同的密钥。
4. **最终外部散列计算**由 HASH 执行。HMAC-SHA1 存储于 HASH_Hx 寄存器（x = 0 ... 4）中，如下所示：

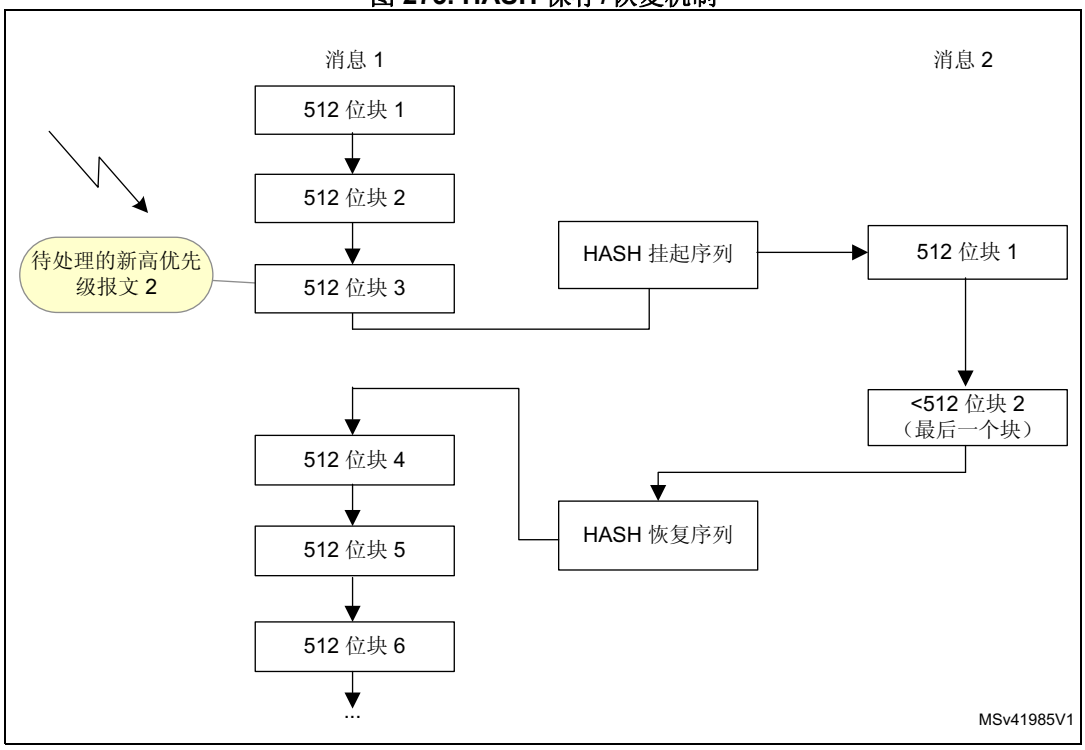
```
HASH_H0 = 0x5FD596EE
HASH_H1 = 0x78D5553C
HASH_H2 = 0x8FF4E72D
HASH_H3 = 0x266DFD19
HASH_H4 = 0x2366DA29
```

36.3.8 上下文交换

概述

可以中断散列/HMAC 操作以执行另一个优先级较高的处理。处理完优先级较高的任务后，中断过程完成，如*图 273* 所示。

图 273. HASH 保存/恢复机制



要这样做，必须将被中断的任务的上下文从散列寄存器保存到存储器，然后从存储器恢复到散列寄存器。

以下说明由软件或 DMA 控制数据流的过程。

通过软件加载数据

如果不使用 DMA 将消息加载到散列处理器，则只有在当前未处理任何块时才能保存上下文。这意味着用户应用程序必须等待至 DINIS = 1（最后一个块已处理且输入 FIFO 为空）或 NBW ≠ 0（FIFO 未满且当前未进行任何处理）。详细过程如下所述。

- **当前上下文保存**

  在中断当前消息摘要计算之前，应用程序必须将以下寄存器的内容存储到存储器中：
  - HASH_IMR
  - HASH_STR
  - HASH_CR
  - HASH_CSR0 到 HASH_CSR53。

- **当前上下文恢复**

  要恢复处理中断的消息，应用程序必须遵循以下步骤：

  a) 将存储器中保存的值写入以下寄存器中：HASH_IMR、HASH_STR 和 HASH_CR

  b) 通过将 HASH_CR 寄存器中的 INIT 位置 1 来初始化散列处理器

  c) 将存储器中保存的值写入 HASH_CSR0 到 HASH_CSR53 寄存器中

  d) 从之前的中断点重新开始处理

通过 DMA 加载数据

如果使用 DMA 将消息加载到散列处理器，则无法预测是否正在进行 DMA 传输。因此，用户应用程序必须停止 DMA 传输，然后等待至散列处理器准备就绪后再中断当前消息摘要计算。详细程序如下所述。

- **当前上下文保存**

  在使用 DMA 中断当前消息摘要计算之前，应用程序必须遵循以下步骤：

  a) 将 DMAE 位清零以禁止 DMA 接口

  b) 等待直至当前 DMA 传输完成（等待 HASH_SR 寄存器中的 DMAS = 0）。请注意，块不一定已完全传输到散列

  c) 在 DMA 控制器中禁止对应的通道

  d) 等到散列处理器就绪（未在处理任何块），也就是等待 DINIS = 1

- **当前上下文恢复**

  要恢复处理使用 DMA 中断的消息，应用程序必须遵循以下步骤：

  a) 重新配置 DMA 控制器，以便其在没有再次被中断的情况下继续传输消息直到结束。

  b) 通过将 DMAE 位置 1，从之前的中断点重新开始处理。

注： *如果上下文交换不涉及 HMAC 操作，则不需要保存和恢复 HASH_CSR38 到 HASH_CSR53 寄存器。*

如果在两个块之间进行上下文交换（最后一个块已完全处理并且下一个块尚未推入到 IN FIFO，HASH_CR 寄存器中 NBW = 000），则不需要保存和恢复 HASH_CSR22 到 HASH_CSR37 寄存器。

36.3.9 HASH DMA 接口

散列处理器可使用一个接口连接 DMA 控制器。通过将 HASH_CR 寄存器中的 DMAE 位置 1，可以使用此 DMA 向 HASH 中写入数据。当该位置 1 时，如果 FIFO 中有足够的空闲字以支持四个字的突发，HASH 会向 DMA 控制器发送突发请求信号。

一旦接收到四个 32 位字，HASH 便会将自动重新启动此过程，检查 FIFO 大小，并在 FIFO 状态允许突发接收时产生新的请求。有关详细信息，请参见*第 36.3.5 节：消息摘要计算*。

在开始 DMA 传输之前，软件必须对将要复制到 HASH_DIN 寄存器的最后一个字中的有效位数进行编程。这可通过向 HASH_STR 寄存器中写入以下值来完成：

```
NBLW = Len(Message)% 32
```

其中，“x%32”表示 x 除以 32 的余数。

HASH_SR 寄存器中的 DMAS 位提供有关 DMA 接口活动的信息。该位与 DMAE 一起设置，并且在 DMAE 清零且未在进行任何 DMA 传输时将被清零。

注： *没有任何中断与 DMAS 位相关联。*

36.3.10 HASH 错误管理

HASH 硬件不生成错误标志。

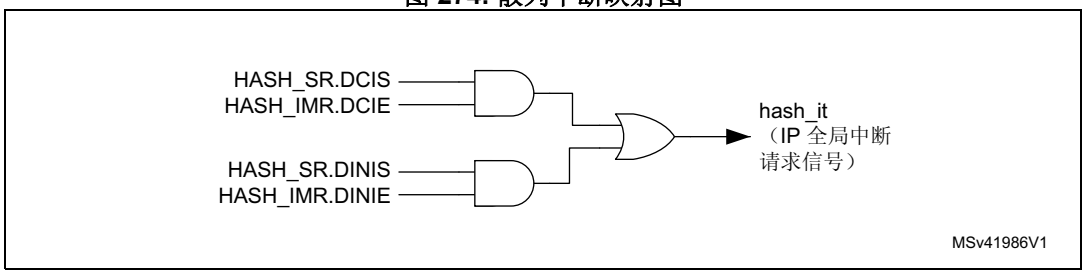
36.4 HASH 中断

散列处理器可生成两个单独的可屏蔽中断源，用以通知发生以下事件：

- 摘要计算完成 (DCIS)
- 数据输入缓冲器就绪 (DINIS)

这两个中断源连接到同一全局中断请求信号，如*图 274* 所示。

图 274. 散列中断映射图



通过更改 HASH_IMR 寄存器中的掩码位，可以单独使能或禁止以上中断源。将相应的屏蔽位置 1 以使能中断。

可以从 HASH_SR 寄存器中读取各个中断事件的状态。*表 279* 对可用功能进行了总结。

表 279. HASH 中断请求

中断事件	事件标志	使能控制位
摘要计算完成标志	DCIS	DCIE
数据输入缓冲器准备好获取新的块标志	DINIS	DINIE

36.5 HASH 处理时间

表 280 汇总了每种工作模式下处理 512 位中间块需要的时间。

表 280. 处理时间（时钟周期数）

工作模式	FIFO 负荷(1)	计算阶段	总计
MD5	16	50	**66**
SHA-1	16	66	**82**
SHA-224	16	50	**66**
SHA-256	16	50	**66**

1. 必须将向处理器中加载 16 字的块所需的时间添加到该值中。

处理消息的最后一个块（或者 HMAC 中某个密钥的最后一个块）所需要的时间可能会更长。此时间取决于最后一个块的长度以及密钥的大小（在 HMAC 模式中）。

与处理中间块相比，此时间可能按以下系数增加：

- 对于散列消息，该系数为 **1 到 2.5**
- 对于 HMAC 输入密钥，该系数在 **2.5 左右**
- 对于 HMAC 消息，该系数为 **1 到 2.5**
- 对于 HMAC 输出密钥，如果是短密钥，则该系数在 **2.5 左右**
- 对于 HMAC 输出密钥，如果是长密钥，则该系数为 **3.5 到 5**

36.6 散列寄存器

散列内核与多个控制和状态寄存器以及五个消息摘要寄存器相关联。所有这些寄存器都只能通过 32 位字来访问，否则将生成 AHB2 错误。

36.6.1 HASH 控制寄存器 (HASH_CR)

HASH control register

偏移地址：0x00

复位值：0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	MDMAT	DINNE	NBW				ALGO[0]	MODE	DATATYPE		DMAE	INIT	Res.	Res.
		rw	r	r	r	r	r	rw	rw	rw	rw	rw	w		

位 31:19 保留，必须保持复位值

位 18 **ALGO [1]**：请参见位 7 说明

位 17 保留，必须保持复位值

位 16 **LKEY**：长密钥选择 (Long key selection)

在 HMAC 模式中，该位在短密钥（≤ 64 字节）或长密钥（> 64 字节）之间进行选择

0：短密钥（≤ 64 字节）

1：长密钥（> 64 字节）

注： 仅当 INIT 位置 1 并且 MODE = 1 时此选择才有效。在计算期间更改该位不起作用。

位 15 保留，必须保持复位值

位 14 保留，必须保持复位值

位 13 **MDMAT**：多个 DMA 传输 (Multiple DMA Transfers)

如果在需要多个 DMA 传输时对大型文件进行散列运算，则将此位置 1。

0：在 DMA 传输结束时自动将 DCAL 置 1。

1：在 DMA 传输结束时不自动将 DCAL 置 1。

位 12 **DINNE**：DIN 非空 (DIN not empty)

当 HASH_DIN 寄存器包含有效数据时（也就是在至少写入一次之后），该位将置 1。当 INIT 位（初始化）或 DCAL 位（上一个消息处理已完成）写入到 1 时，会将此位清零。

0：数据输入缓冲器中不存在任何数据

1：输入缓冲器至少包含一个字的数据

位 11:8 **NBW**：已推入的字数 (Number of words already pushed)

该位域反映了消息中已推入到 IN FIFO 的字数。如果在 DINNE = 1 时对 HASH_DIN 寄存器执行写访问，则 NBW 递增 (+1)。

当 INIT 位写入 1 或者当摘要计算启动时（DCAL 写入 1 或者 DMA 传输结束），NBW 将变为 0000。

如果未使用 DMA：

0000 且 DINNE=0：没有任何字推入到 DIN 缓冲器，即 HASH_DIN 寄存器和 IN FIFO 均为空。

0000 且 DINNE=1：有一个字已推入到 DIN 缓冲器，即 HASH_DIN 寄存器包含一个字而 IN FIFO 为空。

0001：有两个字已推入到 DIN 缓冲器，即 HASH_DIN 寄存器和 IN FIFO 分别包含一个字。

...

1111：16 个字已推入到 DIN 缓冲器

如果使用了 DMA：

则 NBW 刚好是通过 DMA 已推入到 IN FIFO 的字数。

位 18 和位 7 **ALGO[1:0]**：算法选择 (Algorithm selection)

这些位用于选择 SHA-1、SHA-224、SHA256 或 MD5 算法：

00：选择 SHA-1 算法

01：选择 MD5 算法

10：选择 SHA224 算法

11：选择 SHA256 算法

注： 仅当 INIT 位置 1 时此选择才有效。在计算期间更改该位不起作用。

位 6 **MODE**：模式选择 (Mode selection)

该位针对所选算法选择散列或 HMAC 模式：

0：选择散列模式

1：选择 HMAC 模式。如果正在使用的密钥的长度超过 64 字节，则必须 LKEY 置 1。

注： 仅当 INIT 位置 1 时此选择才有效。在计算期间更改该位不起作用。

位 5:4 **DATATYPE**：数据类型选择 (Data type selection)

这些位定义在 HASH_DIN 寄存器中输入的数据格式：

00：32 位数据。写入到 HASH_DIN 的数据将由散列处理直接使用，不会重新排序。

01：16 位数据或半字。写入到 HASH_DIN 的数据被视为两个半字，并且在交换之后由散列处理使用。

10：8 位数据或字节。写入到 HASH_DIN 的数据被视为四个字节，并且在交换之后由散列处理使用。

11：位数据或位串。写入到 HASH_DIN 的数据被视为 32 位（位串的第一位处于位置 0），并且在交换之后由散列处理使用（位串的第一位处于位置 31）。

位 3 **DMAE**：DMA 使能 (DMA enable)

0：禁止 DMA 传输

1：使能 DMA 传输 散列内核可以接收数据时，便发送 DMA 请求。

该位置 1 后，将在消息的最后一个数据被写入散列处理器时由硬件清零。

当正在进行 DMA 传输时，将此位置 0 不会中止当前传输。HASH 的 DMA 接口在内部继续保持使能状态，直到传输完成或 INIT 被写入 1。

将 INIT 位置 1 不会清零 DMAE 位。

位 2 **INIT**：初始化消息摘要计算 (Initialize message digest calculation)
将该位写入到 1 将会复位散列处理器内核，以使散列可以计算新消息的消息摘要。
若在此位中写入 0 则不会产生影响。读取该位将始终返回 0。

位 1:0 保留，必须保持复位值

36.6.2 散列数据输入寄存器 (HASH_DIN)

HASH data input register

偏移地址：0x04

复位值：0x0000 0000

HASH_DIN 是数据输入寄存器。其宽度为 32 位。该寄存器用于以 512 位的块来输入消息。编程 HASH_DIN 寄存器时，在 AHB 数据总线上呈现的值将被“推入”散列内核，并且寄存器获取 AHB 数据总线上呈现的新值。要获得正确的消息格式，必须先对 HASH_CR 寄存器中的 DATATYPE 位进行配置。

16 字的块写入到 HASH_DIN 寄存器后，将启动中间摘要计算。

- 如果未使用 DMA，将新数据写入到 HASH_DIN 寄存器来启动，即写入下一个块的第一个字（中间摘要计算）
- 如果使用了 DMA，则自动启动

最后一个块写入到 HASH_DIN 寄存器之后，将启动最终摘要计算（包括填充）。

- 通过在 HASH_STR 寄存器中将 DCAL 位写入 1 来启动（最终摘要计算）
- 如果使用了 DMA 并且 MDMAT 位设置为 0，则自动启动

当摘要计算（中间或最终）正在进行时，如果对 HASH_DIN 寄存器执行新的写访问，则将在 AHB2 总线上插入等待状态，直至散列计算完成。

读取 HASH_DIN 寄存器时，将访问该位置中写入的最后一个字（复位后为零）。

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

位 31:0 **DATAIN**：数据输入 (Data input)
读取该寄存器会返回当前寄存器内容。
写入该寄存器会将当前寄存器内容推入到 IN FIFO，并且寄存器获取在 AHB 数据总线上呈现的新值。

36.6.3 散列启动寄存器 (HASH_STR)

HASH start register

偏移地址：0x08

复位值：0x0000 0000

HASH_STR 寄存器有两个功能：

- 用于定义散列处理器中输入的消息的最后一个字的有效位数（也就是写入到 HASH_DIN 寄存器的最后一个数据中有效的最低有效位的数量）
- 通过将 DCAL 位写入到 1，启动对消息中最后一个块的处理

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

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

位 31:9 保留，必须保持复位值

位 8 **DCAL**：摘要计算 (Digest calculation)

将该位写 1 将开始使用先前写入的 NBLW 值来填充消息，并且，由于 INIT 位已被写 1，因此将使用写入到 IN FIFO 的所有数据字开始计算最终消息摘要。

读取该位将返回 0。

位 7:5 保留，必须保持复位值

位 4:0 **NBLW**：最后一个字中的有效位数 (Number of valid bits in the last word)

当消息位串的最后一个字被写入 HASH_DIN 寄存器并且经过内部数据交换后，散列处理器会只接收如下指定的有效位：

0x00：写入的最后一个数据的全部 32 位均是有效消息位，即 M[31:0]

0x01：写入的最后一个数据中只有一位（在交换后）是有效消息位，即 M[0]

0x02：写入的最后一个数据中只有两位（在交换后）是有效消息位，即 M[1:0]

0x03：写入的最后一个数据中只有三位（在交换后）是有效消息位，即 M[2:0]

...

0x1F：写入的最后一个数据中只有31位（在交换后）是有效消息位，即 M[30:0]

上述机制仅在 DCAL = 0 时有效。如果在 DCAL 置 1 时写入 NBLW 位，则 NBLW 位域保持不变。换句话说，配置 NBLW 和设置 DCAL 不能同时进行。

读取 NBLW 位将返回写入到 NBLW 的最后一个值。

36.6.4 HASH 摘要寄存器 (HASH_HR0..7)

HASH digest registers

这些寄存器包含消息摘要结果，其命名方式如下：

1. 在 SHA1 算法说明中分别为 H0、H1、H2、H3 和 H4。
   在这种情况下，不使用 HASH_H5 到 HASH_H7 寄存器并且其读取值为零。
2. 在 MD5 算法说明中分别为 A、B、C 和 D。
   在这种情况下，不使用 HASH_H4 到 HASH_H7 寄存器并且其读取值为零。
3. 在 SHA224 算法说明中分别为 H0 到 H6。
   在这种情况下，不使用 HASH_H7 并且其读取值为零。
4. 在 SHA256 算法说明中分别为 H0 到 H7。

在所有情况下，摘要的最高有效位均存储在 HASH_H0[31] 中。

如果当散列内核在计算中间摘要或最终消息摘要时（也就是在 DCAL 位已经写入到 1 时），对这些寄存器之一进行读访问，则读取操作将被拖延直到散列计算完成。

注： *H0、H1、H2、H3 和 H4 映射到两个存储区域。*

HASH_HR0

偏移地址：0x0C 和 0x310

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
H0															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
H0															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

HASH_HR1

偏移地址：0x10 和 0x314

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
H1															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
H1															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

HASH_HR2

偏移地址：0x14 和 0x318

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
H2															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
H2															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

HASH_HR3

偏移地址：0x18 和 0x31C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
H3															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
H3															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

HASH_HR4

偏移地址：0x1C 和 0x320

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
H4															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
H4															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

HASH_HR5

偏移地址：0x324

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
H5															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
H5															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

HASH_HR6

偏移地址：0x328

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
H6															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
H6															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

HASH_HR7

偏移地址：0x32C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
H7															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
H7															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

注：　当启动新位流的摘要计算时（通过向 INIT 位写入 1），这些寄存器将被强制设为其复位值。

36.6.5 散列中断使能寄存器 (HASH_IMR)

HASH interrupt enable register

偏移地址：0x20

复位值：0x0000 0000

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

位 31:2 保留，必须保持复位值

位 1 **DCIE**：摘要计算完成中断使能 (Digest calculation completion interrupt enable)

0：禁止摘要计算完成中断

1：使能摘要计算完成中断

位 0 **DINIE**：数据输入中断使能 (Data input interrupt enable)

0：禁止数据输入中断

1：使能数据输入中断

36.6.6 散列状态寄存器 (HASH_SR)

HASH status register

偏移地址：0x24

复位值：0x0000 0001

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BUSY	DMAS	DCIS	DINIS
												r	r	rc_w0	rc_w0

位 31:4 保留，必须保持复位值

位 3 **BUSY**：忙碌位 (Busy bit)

0：当前未在处理任何块
1：散列内核正在处理某个数据块

位 2 **DMAS**：DMA 状态 (DMA Status)

该位提供有关 DMA 接口活动的信息。该位与 DMAE 一起设置，并且在 DMAE=0 且未在进行任何 DMA 传输时将被清零。没有任何中断与该位相关联。

0：禁止 DMA 接口 (DMAE=0) 并且未在进行任何传输
1：使能 DMA 接口 (DMAE=1) 或者某个传输在进行

位 1 **DCIS**：摘要计算完成中断状态 (Digest calculation completion interrupt status)

该位是在摘要就绪时（也就是整个消息已处理完毕时）由硬件设置。要将该位清零可写入 0，或者向 HASH_CR 寄存器中的 INIT 位写入 1。

0：HASH_Hx 寄存器中无任何摘要
1：摘要计算已完成，摘要存储于 HASH_Hx 寄存器中。如果 HASH_IMR 寄存器中将 DCIE 位置 1，则产生中断

位 0 **DINIS**：数据输入中断状态 (Data input interrupt status)

该位是在输入缓冲器准备好获取新块时（16 个位置空闲）由硬件置 1。要将该位清零写入 0，或者写 HASH_DIN 寄存器。

0：输入缓冲器中的空闲位置少于 16 个
1：可以将一个新块输入到输入缓冲器中。如果 HASH_IMR 寄存器中将 DINIE 位置 1，则产生中断

36.6.7 散列上下文交换寄存器 (HASH_CSRx)

HASH context swap registers

这些寄存器包含散列处理器的完整内部寄存器状态。如果高优先级的任务需要使用散列处理器，而该处理器正在执行其他任务，则必须使用上述寄存器进行上下文交换。

发生此类事件时，必须读取 HASH_CSRx 寄存器，并且读取值必须保存到系统存储器空间中。然后有优先权的任务便可以使用散列处理器，在散列计算完成时，可以从存储器中读取保存的上下文并回写到 HASH_CSRx 寄存器中。

HASH_CSR0

偏移地址：0x0F8

复位值：0x0000 0002

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CS0															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CS0															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

HASH_CSRx（x = 1 至 53）

偏移地址：0x0F8 + x * 0x4

复位值：0x0000 0000

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

36.6.8 散列寄存器映射

表 281 汇总了散列寄存器映射和复位值。

表 281. HASH 寄存器映射和复位值

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	HASH_CR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ALGO[1]	Res.	LKEY	Res.	Res.	MDMAT	DINNE	NBW				ALGO[0]	MODE	DATATYPE		DMAE	INIT	Res.	Res.
	Reset value														0		0			0	0	0	0	0	0	0	0	0	0	0	0		
0x04	HASH_DIN	DATAIN																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x08	HASH_STR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DCAL	Res.	Res.	Res.	NBLW				
	Reset value																								0				0	0	0	0	0
0x0C	HASH_HR0	H0																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x10	HASH_HR1	H1																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x14	HASH_HR2	H2																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x18	HASH_HR3	H3																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x1C	HASH_HR4	H4																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x20	HASH_IMR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DCIE	DINIE
	Reset value																														0	0	
0x24	HASH_SR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BUSY	DMAS	DCIS	DINIS
	Reset value																													0	0	0	1
0xF8	HASH_CSR0	CSR0																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
0xFC	HASH_CSR1	CSR1																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
...																																	
0x1CC	HASH_CSR53	CSR53																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Reserved																																	
0x310	HASH_HR0	H0																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x314	HASH_HR1	H1																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x318	HASH_HR2	H2																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x31C	HASH_HR3	H3																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x320	HASH_HR4	H4																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x324	HASH_HR5	H5																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x328	HASH_HR6	H6																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x32C	HASH_HR7	H7																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

37 高分辨率定时器 (HRTIM)

37.1 简介

高分辨率定时器可生成多达 10 路高度精确定时数字信号，主要用于驱动开关模式电源或照明系统等电源转换系统，但也可用于，一般对时间分辨率有极高要求的应用。

该定时器采用模块化架构，可生成独立波形或耦合波形。波形由独立式定时信号（使用计数器和比较单元）以及多种外部事件（如模拟或数字反馈以及同步信号）确定，因此可生成大量不同的控制信号（PWM、相移、恒定 Ton...），从而满足大部分转换拓扑的需求。

为实现控制和监测用途，该定时器还具有定时测量功能，并连接到内置的 ADC 和 DAC 转换器。此外，该定时器还具有轻载管理模式，能够处理各种故障机制，从而实现安全关断。

37.2 主要特性

- 多个定时单元
  - 所有输出均支持全分辨率，可在触发单脉冲模式下调整占空比、频率和脉宽
  - 6 个 16 位定时单元（每个定时单元包含 1 个独立计数器和 4 个比较单元）
  - 10 路输出可通过任何定时单元控制，每条通道多达 32 个置位/复位源
  - 模块化结构可满足多种配有 1 或 2 个开关的独立转换器的需求，也可满足少数大型多开关拓扑的需求
- 多达 10 个外部事件，可用于任何定时单元
  - 可编程极性和边沿有效性
  - 5 个事件用于快速异步模式
  - 5 个事件用于可编程数字滤波器
  - 利用消隐和窗口模式实现伪事件过滤
- 多条通道可连接到内置模拟外设
  - 4 个用于 ADC 转换器的触发信号
  - 3 个用于 DAC 转换器的触发信号
  - 3 个用于模拟信号调理的比较器
- 丰富的保护机制
  - 5 路故障输入可组合使用并关联到任何定时单元
  - 可编程极性和边沿有效性，数字滤波器
  - 对谐振变换器配有专门的延时保护
- 多个 HRTIM 实例可与外部同步输入/输出同步
- 多功能输出级
  - 全分辨率时间插入
  - 可编程输出极性
  - 斩波模式
- 突发模式控制器，可同时处理多个转换器上的轻载操作
- 7 个中断向量，每个向量最多具有 14 个源
- 6 个 DMA 请求，最多具有 14 个源，可通过突发模式实现多寄存器更新

37.3 功能描述

37.3.1 概述

HRTIM 可分为以下几个子实体：

- 主定时器
- 定时单元（定时器 A 到定时器 E）
- 输出级
- 突发模式控制器
- 外部事件和故障信号调节逻辑，由所有定时器共享
- 系统接口

主定时器基于 16 位递增计数器。它可通过 4 个比较单元置位/复位 10 路输出中的任何一路，并向 5 个定时器单元提供同步信号。其主要用途是使定时器单元受唯一的时钟源控制。交错降压转换器是一个典型的应用示例，主定时器在其中管理着多个单元之间的相移。

定时器单元既可以独立工作，也可以与其他定时器（包括主定时器）配合工作。每个定时器都可控制两路输出。输出置位/复位事件可以由定时单元比较寄存器触发，或者由来自主定时器、其他定时器的事件或外部事件触发。

输出级有多种用途

- 在互补 PWM 模式下配置 2 路输出时添加死区
- 将载波频率添加到调制信号上
- 通过将输出异步置为预定义的安全电平来管理故障事件

在轻载运行的情况下，突发模式控制器可控制一个或多个定时器。突发长度和周期、以及输出的空闲状态可通过编程设定。

外部事件和故障信号调理逻辑包括：

- 输入选择 MUX（例如，为给定外部事件通道选择数字输入或片上时钟源）
- 极性和边沿有效性编程
- 数字滤波（对 10 条通道中的 5 条通道进行滤波）

系统接口允许 HRTIM 与 MCU 的其余部分进行交互：

- 向 CPU 发出中断请求
- 通过 DMA 控制器自动访问存储器，包括 HRTIM 特有的突发模式
- 触发 ADC 和 DAC 转换器

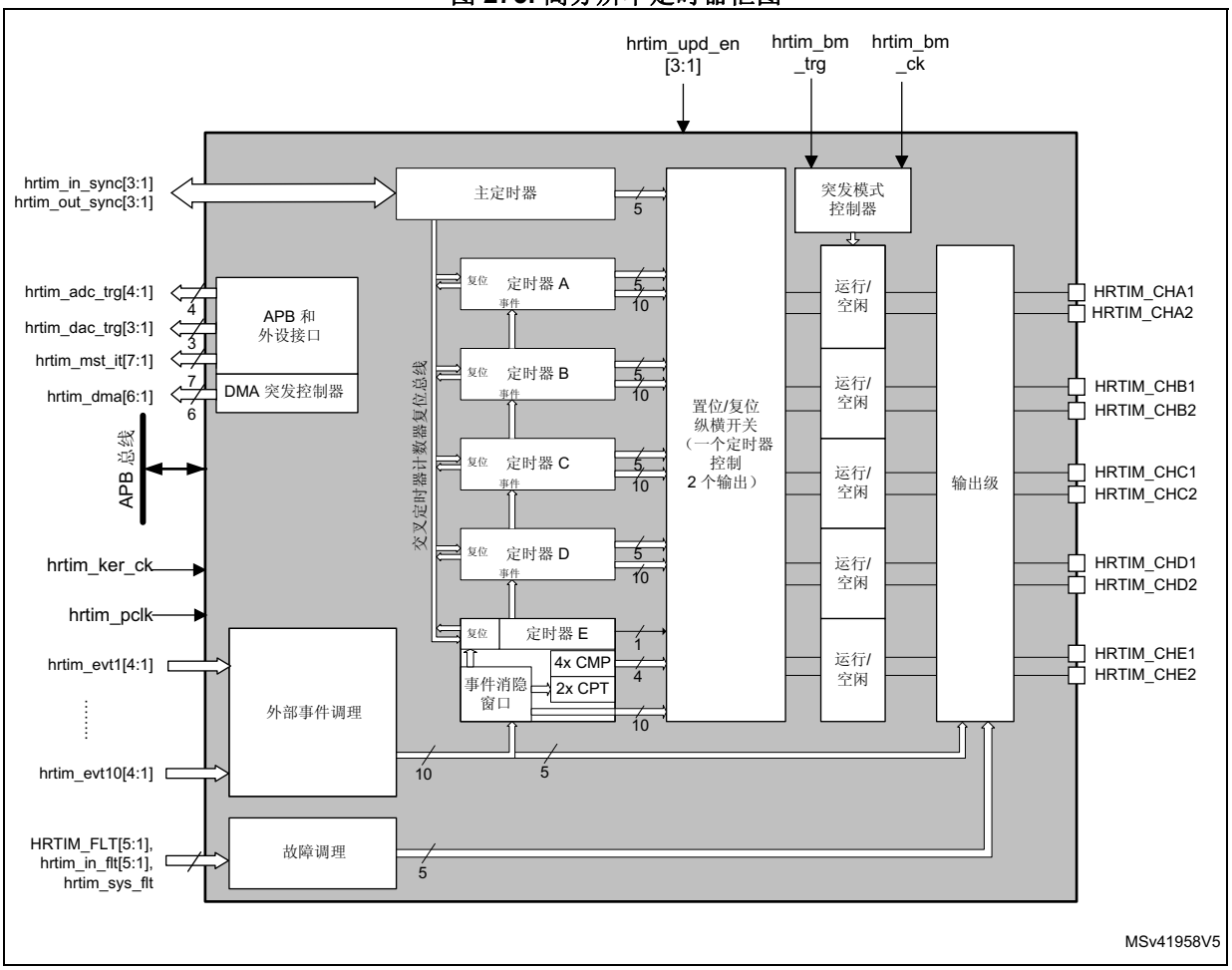
HRTIM 寄存器分为 7 组：

- 主定时器寄存器
- 定时器 A 到定时器 E 寄存器
- 通用寄存器，用于所有定时器单元共用的功能

注： *根据文档编写约定，在文本和寄存器中对于 5 个定时单元的引用统一用“x”字母表示（x 可以是 A 到 E 的任意值）。*

定时器的框图如*图 275* 所示。

图 275. 高分辨率定时器框图



37.3.2 HRTIM 引脚和内部信号

下表汇总了片上和片外的 HRTIM 输入和输出。

表282. HRTIM 输入/输出汇总

信号名称	信号类型	说明
HRTIM_CHA1、HRTIM_CHA2、HRTIM_CHB1、HRTIM_CHB2、HRTIM_CHC1、HRTIM_CHC2、HRTIM_CHD1、HRTIM_CHD2、HRTIM_CHE1、HRTIM_CHE2	输出	主 HRTIM 定时器输出。这些输出成对工作时（HRTIM_CHx1 和 HRTIM_CHx2），可以插入死区或独立工作。
HRTIM_FLT[5:1]，hrtim_in_flt[5:1]	数字输入	故障输入：置为有效时立即禁止 HRTIM 输出（5 路片上输入和 5 路片外 HRTIM_FLTx 输入）。

表282. HRTIM 输入/输出汇总（续）

信号名称	信号类型	说明
hrtim_sys_flt	数字输入	涵盖 MCU 内部故障事件（时钟安全系统、SRAM 奇偶校验错误、Cortex®-M7 LOCKUP (HardFault)、PVD 输出）的系统故障。
hrtim_in_sync[3:1]	数字输入	将整个 HRTIM 与其他内部或外部定时器资源进行同步的同步输入： hrtim_in_sync1：保留 hrtim_in_sync2：时钟源为常规 TIMx 定时器（通过片上互连） hrtim_in_sync3：时钟源为外部 HRTIM（通过 HRTIM_SCIN 输入引脚）
hrtim_out_sync[2:1]	数字输出	此输出用于级联或同步多个片上或片外 HRTIM 实例： hrtim_out_sync1：保留 hrtim_out_sync2：目标为片外 HRTIM 或外设（通过 HRTIM_SCOUT 输出引脚）
hrtim_evt1[4:1]	数字输入	外部事件：10 个事件均可在 4 个时钟源中选择，可选择片上时钟源（来自其他内置外设：比较器、ADC 模拟看门狗、TIMx 定时器、触发输出）或片外时钟源（HRTIM_EEVx 输入引脚）
hrtim_evt2[4:1]		
hrtim_evt3[4:1]		
hrtim_evt4[4:1]		
hrtim_evt5[4:1]		
hrtim_evt6[4:1]		
hrtim_evt7[4:1]		
hrtim_evt8[4:1]		
hrtim_evt9[4:1]		
hrtim_evt10[4:1]		
hrtim_upd_en[3:1]	数字输入	HRTIM 寄存器更新使能输入（片上互连）会触发从影子寄存器到活动寄存器的传输操作
hrtim_bm_trg	数字输入	突发模式触发事件（片上互连）
hrtim_bm_ck[4:1]	数字输入	突发模式时钟（片上互连）
hrtim_adc_trg[4:1]	数字输出	ADC 转换开始触发信号
hrtim_dac_trg[3:1]	数字输出	DAC 转换更新触发信号
hrtim_mst_it[7:1]	数字输出	中断请求
hrtim_dma[6:1]	数字输出	DMA 请求
hrtim_pclk	数字输入	APB 时钟
hrtim_ker_ck	数字输入	HRTIM 内核时钟（以下称为 f_{HRTIM}）

37.3.3 时钟

HRTIM 必须由 t_{HRTIM} 系统时钟提供才能实现全分辨率。HRTIM 中的所有时钟均由该参考时钟生成。

术语定义

f_{HRTIM}：主 HRTIM 时钟 (hrtim_ker_ck)。所有后续时钟均由该时钟源生成，并与该时钟源同步。

f_{DTG}：死区发生器时钟。为了方便起见，本文档中仅使用 t_{DTG} 周期 ($t_{DTG} = 1/f_{DTG}$)。

f_{CHPFRQ}：斩波级时钟源。

f_{1STPW}：定义斩波模式下初始脉冲长度的时钟源。为了方便起见，本文档中仅使用 t_{1STPW} 周期 ($t_{1STPW} = 1/f_{1STPW}$)。

f_{BRST}：突发模式控制器计数器时钟。

$f_{SAMPLING}$：对故障或外部事件输入进行采样时所需的时钟。

f_{FLTS}：由用作 $f_{SAMPLING}$ 时钟源的 f_{HRTIM} 生成的时钟，用于对故障事件进行过滤。

f_{EEVS}：由用作 $f_{SAMPLING}$ 时钟源的 f_{HRTIM} 生成的时钟，用于对外部事件进行过滤。

f_{pclk} (hrtim_pclk)：APB 总线时钟，寄存器读/写访问时需要。

定时器时钟和预分频器

HRTIM 中的每个定时器都有独立的时钟预分频器，以供用户调整定时器分辨率。（请参见 *表 283*）

表 283. 定时器分辨率和最小 PWM 频率（f_{HRTIM} = 400 MHz 时）

CKPSC[2:0](1)	预分频比	$f_{COUNTER}$	分辨率	最小 PWM 频率
101	1	400 MHz	2.5 ns	6.1 kHz
110	2	400/2 MHz = 200 MHz	5 ns	3.05 kHz
111	4	400/4 MHz = 100 MHz	10 ns	1.5 kHz

1. CKPSC[2:0] 保留从 000 到 100 的值。

全分辨率可用于边沿定位、PWM 周期调整以及外部触发的脉冲持续时间。

初始化

启动时，务必先初始化预分频器位域，然后再写入比较和周期寄存器。定时器使能后（HRTIM_MCR 寄存器中的 MCEN 或 TxCEN 位已置 1），不能修改预分频器。

如果有多个定时器已使能，预分频器会与先启动的定时器的预分频器同步。

警告： **仅当计数器与输出行为与其他定时器的信息和信号无关时，主定时器和 TIMA..E 定时器才能使用不同的预分频比。如果以下某个事件从一个定时单元（或主定时器）传输到另一个定时单元，则务必要在这些定时器中配置相同的预分频比：输出置位/复位事件、计数器复位事件、更新事件、外部事件过滤或捕获触发。预分频系数不相等会导致结果不可预测。**

死区发生器时钟

死区预分频器由 f_{HRTIM} / 8 / $2^{(DTPRSC[2:0])}$ 提供，通过 HRTIM_DTxR 寄存器中的 DTPRSC[2:0] 位编程。

当 f_{HRTIM} = 400 MHz 时，t_{DTG} 的范围是 2.5 ns 到 20 ns。

斩波级时钟

斩波级时钟源 f_{CHPFRQ} 由 f_{HRTIM} 生成，使用 16 到 256 的分频系数，因此 1.56 MHz <= f_{CHPFRQ} <= 25 MHz（f_{HRTIM} = 400 MHz 时）。

t_{1STPW} 是斩波模式下初始脉冲的长度，通过 HRTIM_CHPxR 寄存器中的 STRPW[3:0] 位编程，其计算公式如下：

t_{1STPW} = (STRPW[3:0]+1) x 16 x t_{HRTIM}。

计算时使用 f_{HRTIM} / 16 作为时钟源（f_{HRTIM}= 400 MHz 时为 25 MHz）。

突发模式预分频器

突发模式控制器计数器时钟 f_{BRST} 可由多个时钟源提供，其中一个时钟源由 f_{HRTIM} 生成。

在这种情况下，f_{BRST} 的范围为 f_{HRTIM} 到 f_{HRTIM} / 32768（f_{HRTIM} = 400 MHz 时为 12.2 kHz）。

故障输入采样时钟

故障输入噪音抑制滤波器的时间常量是通过 $f_{SAMPLING}$ 定义的，可以是 f_{HRTIM} 或 f_{FLTS}。

f_{FLTS} 是由 f_{HRTIM} 生成的，其范围为 400 MHz 到 50 MHz（f_{HRTIM} = 400 MHz 时）。

外部事件输入采样时钟

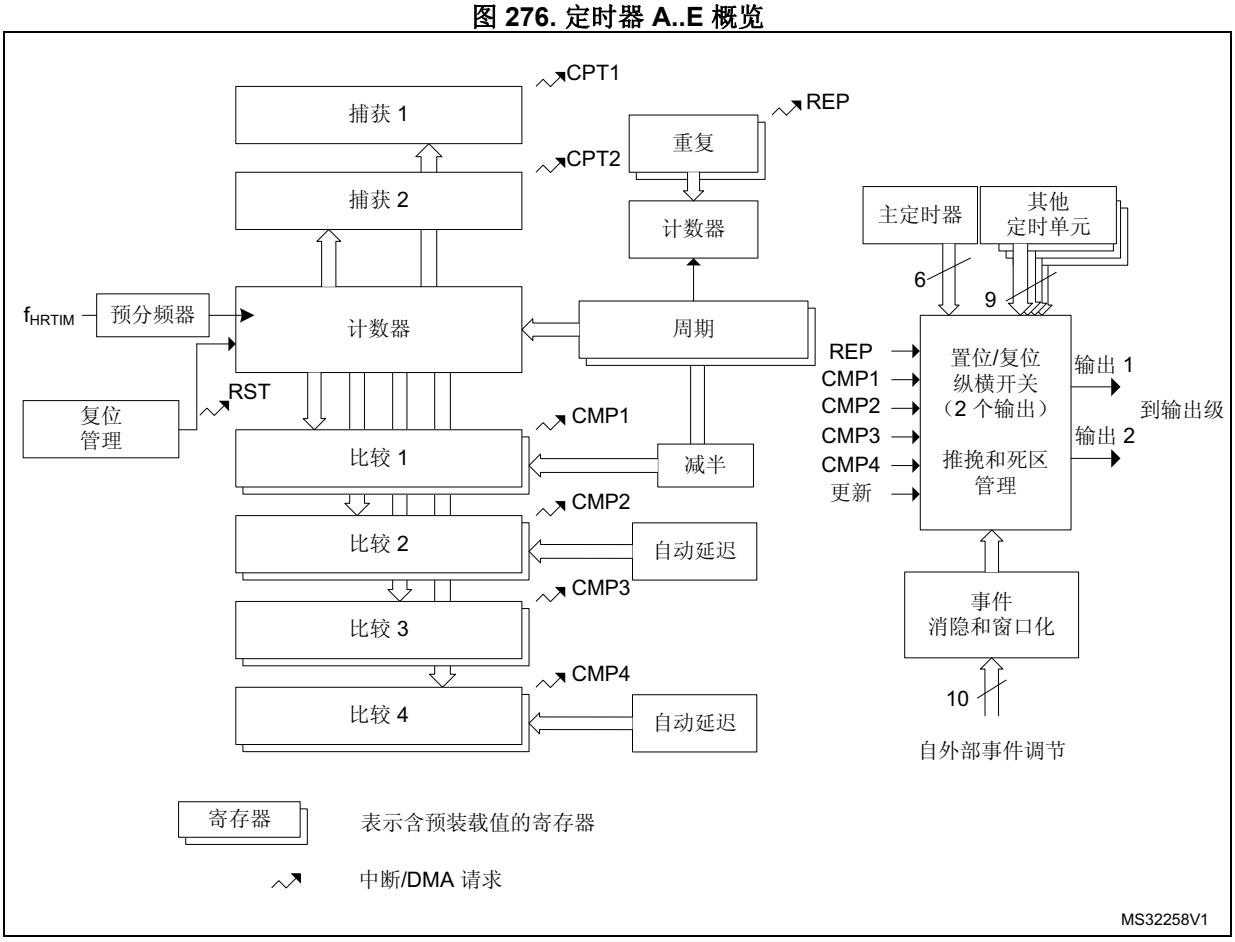
故障输入噪音抑制滤波器的时间常量是通过 $f_{SAMPLING}$ 定义的，可以是 f_{HRTIM} 或 f_{EEVS}。

f_{EEVS} 是由 f_{HRTIM} 生成的，其范围为 400 MHz 到 50 MHz（f_{HRTIM} = 400 MHz 时）。

37.3.4 定时器 A..E 定时单元

HRTIM 嵌入了 5 个完全相同的定时单元（这些定时单元由采用自动重载机制的 16 位递增计数器组成，用于定义计数周期）、4 个比较单元和 2 个捕获单元，如*图 276* 所示。每个单元都包含对 2 路输出的所有控制功能，因此可作为独立定时器工作。

图 276. 定时器 A..E 概览



周期和比较值必须在与高分辨率实现相关的上下限范围内，具体数值列于*表 284* 中：

- 最小值必须大于或等于 3 个 f_{HRTIM} 时钟周期
- 最大值必须小于或等于 0xFFFF - 1 个 f_{HRTIM} 时钟周期

表 284. 周期和比较寄存器最小值和最大值

CKPSC[2:0] 值(1)	最小值	最大值
≥ 5	0x0003	0xFFFD

1. CKPSC[2:0] 小于 5 的值被保留。

注： *如果比较值大于周期寄存器值，则不会生成比较匹配事件。*

计数器工作模式

定时器 A..E 可在连续（自由运行）模式下工作，也可以单发方式工作，此时会由复位事件触发开始计数，工作模式通过 HRTIM_TIMxCR 控制寄存器中的 CONT 位设置。附加的 RETRIG 位可用于选择单发操作是可再触发的或不可再触发的。*表 285* 和*图 277* 以及*图 278* 总结了工作模式的详细信息。

表 285. 定时器工作模式

CONT	RETRIG	工作模式	启动/停止条件 时钟和事件生成
0	0	单发 不可再触发	将 TxEN 位置 1 会使能定时器，但不会启动计数器。 第一个复位事件会触发计数器开始计数，但计数器在达到 PER 值之前会忽略任何后续复位事件。 随后会生成 PER 事件，计数器停止计数。 发生复位事件后，计数器会重新从 0x0000 开始计数。
0	1	单发 可再触发	将 TxEN 位置 1 会使能定时器，但不会启动计数器。 如果计数器停止计数，则复位事件会使计数器开始计数，否则会将计数器清零。计数器达到 PER 值后，会生成 PER 事件，计数器会停止计数。 发生复位事件后，计数器会重新从 0x0000 开始计数。
1	X	连续模式	将 TxEN 位置 1 会使能定时器，同时会启动计数器。 计数器达到 PER 值后，会翻转到 0x0000 并重新开始计数。 可以随时复位计数器。

可以随时清零 TxEN 位，以禁止定时器并停止计数。

图 277. 连续定时器工作模式

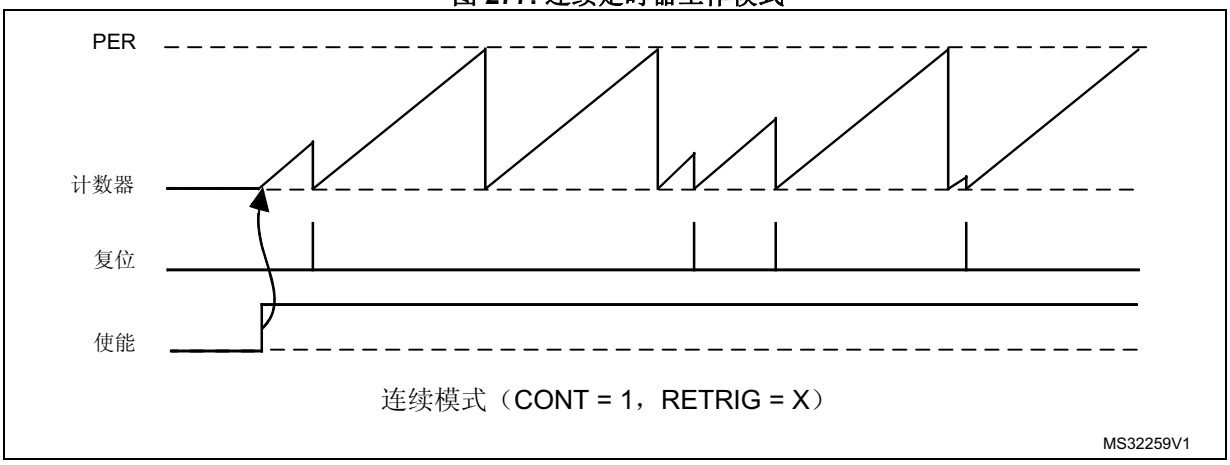
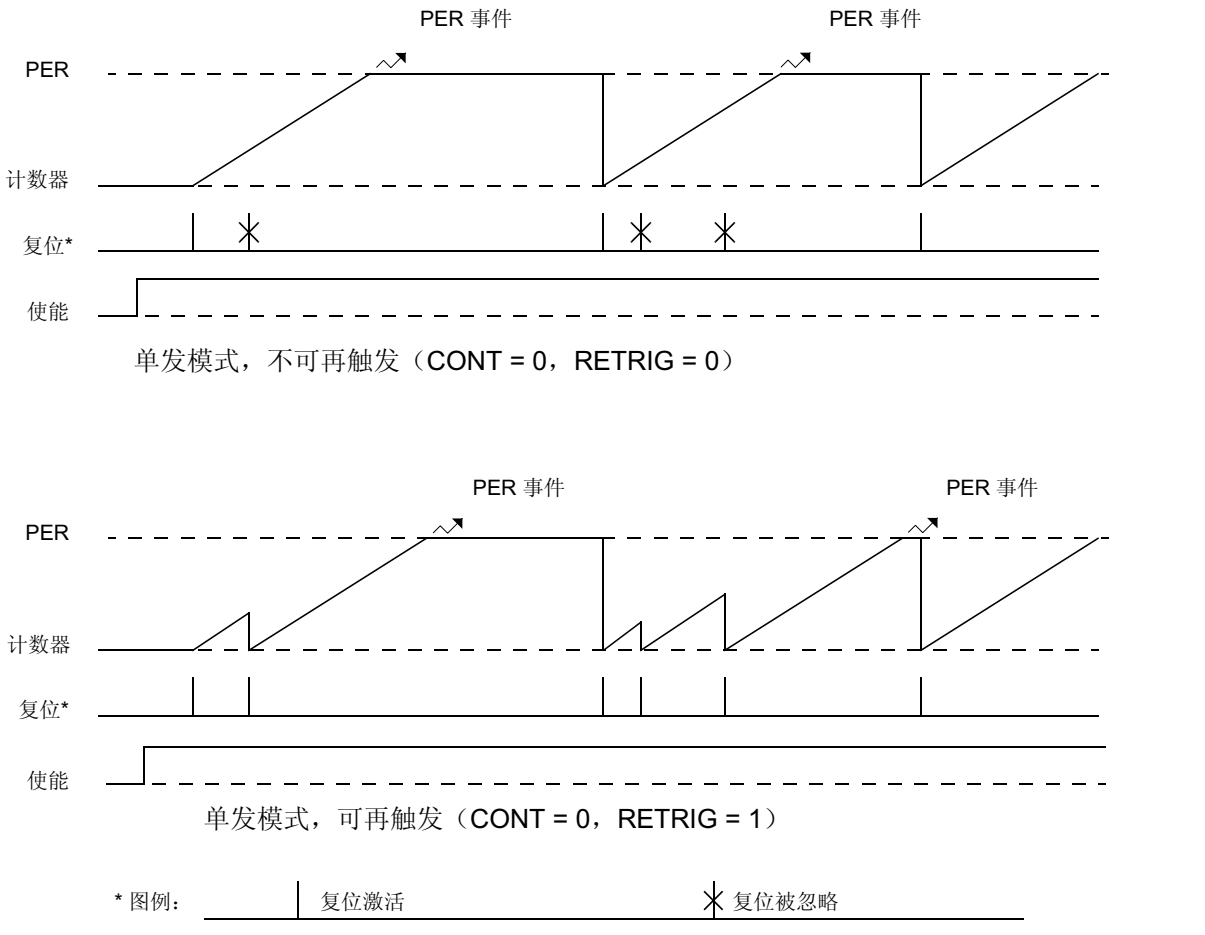


图 278. 单发定时器工作模式



翻转事件

在连续模式下，如果计数器在达到 HRTIM_PERxR 寄存器中设置的周期值后恢复为 0，则会生成计数器翻转事件。

该事件在 HRTIM 中用于多种用途：

- 置位/复位输出
- 触发寄存器内容更新（从预装载寄存器传输到活动寄存器）
- 触发 IRQ 或 DMA 请求
- 作为突发模式时钟源或突发启动触发信号
- 作为 ADC 触发信号
- 使重复计数器递减

如果初始计数器值大于定时器启动时的周期值，或者在计数器已超过该值时设置了新周期，计数器不会复位：计数器将在达到最大周期值时溢出，并且重复计数器不会递减。

定时器复位

定时单元计数器的复位可通过多达 30 种事件触发，这些事件可同时在 HRTIM_RSTxR 寄存器中选择，具体包括以下复位源：

- 定时单元：比较 2、比较 4 和更新（3 个事件）
- 主定时器：复位和比较 1..4（5 个事件）
- 外部事件 EXTEVNT1..10（10 个事件）
- 所有其他定时单元（例如，对于定时器 A，则为定时器 B..E）：比较 1、2 和 4（12 个事件）

可同时选择多个事件处理多个复位源。在这种情况下，会对多个复位请求进行或运算。如果在同一 f_{HRTIM} 时钟周期内生成 2 个计数器复位事件，则会考虑后一个定时器复位事件。

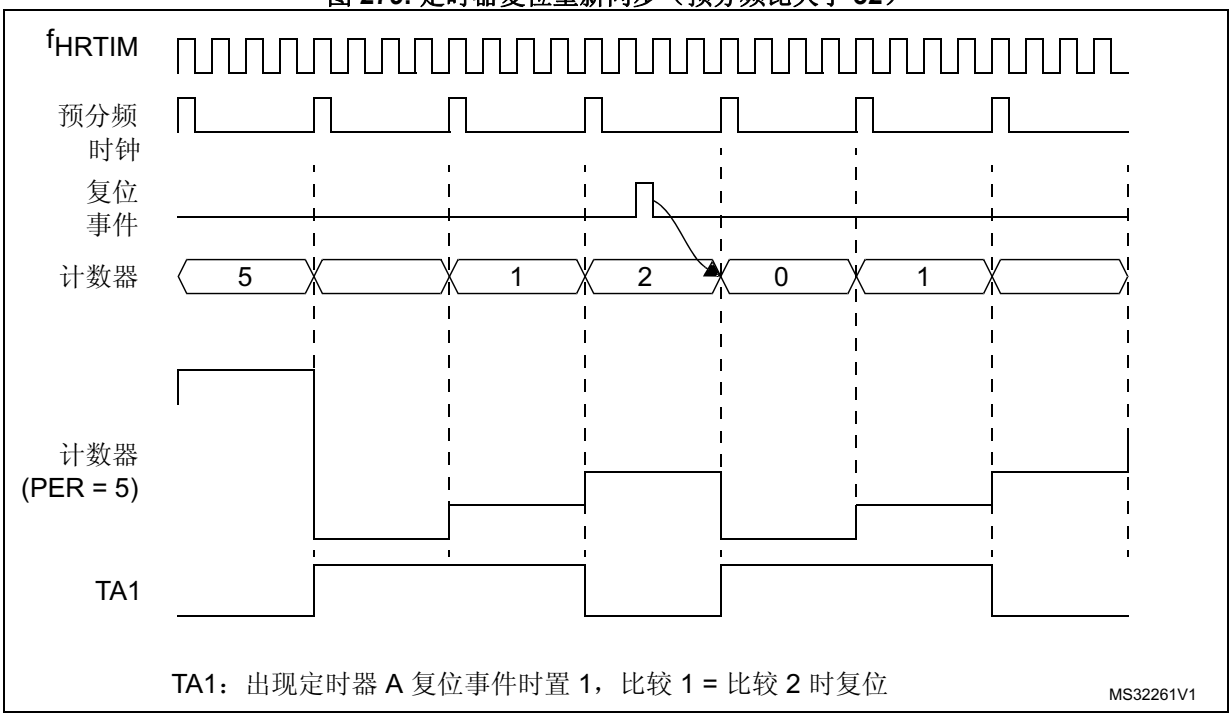
此外，还可以使用 HRTIM_CR2 寄存器中的 TxRST 位对计数器执行软件复位。这些控制位分组到一个寄存器中，从而可同时复位多个计数器。

仅当相关计数器已使能后（TxCEN 位置 1），才会考虑复位请求。

如果 f_{HRTIM} 时钟预分频比大于 1，计数器复位事件会延迟到预分频时钟的下一有效边沿，这样可确保在输出跳变同步到复位事件（通常是恒定 Ton 时间转换器）时生成的波形无抖动。

图 279 显示了时钟预分频比为 4（f_{HRTIM} 除以 4）时的复位处理方式。

图 279. 定时器复位重新同步（预分频比大于 32）



重复计数器

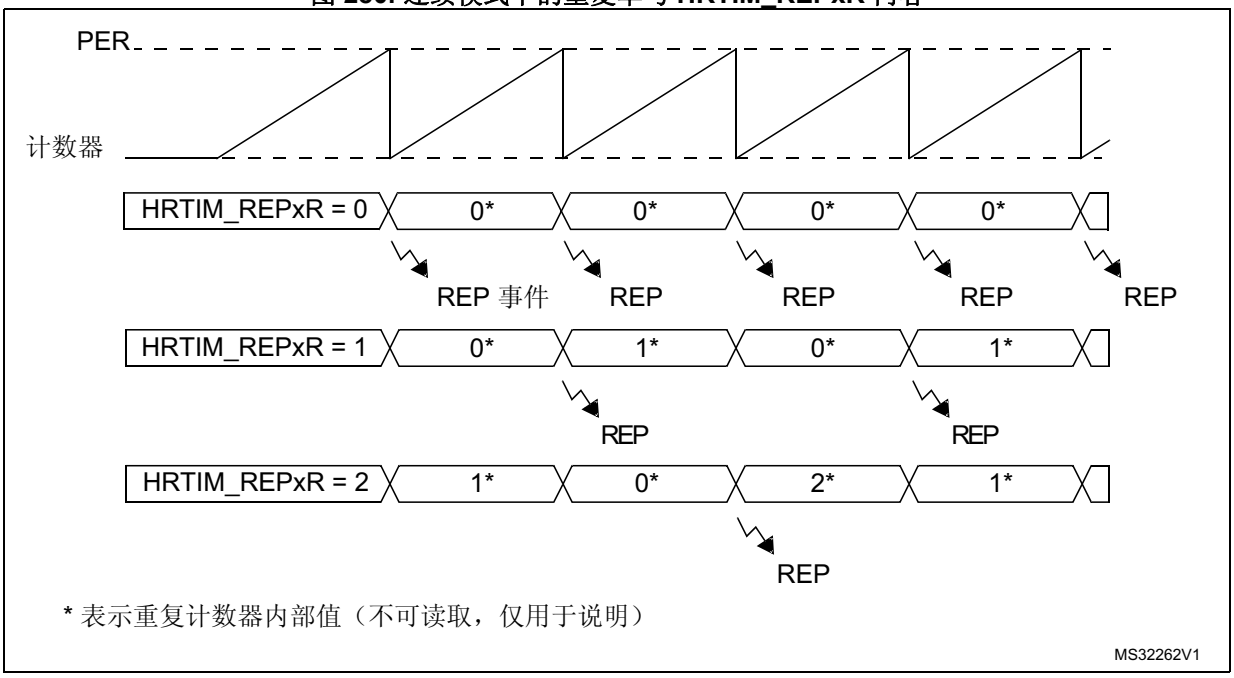
软件通常会在达到周期值时生成中断，从而在下一周期开始之前留出最长的时间用于处理。重复计数器的主要用途是通过分离开关频率和中断频率来调整周期中断率并分担 CPU 的负荷。

定时单元包含重复计数器。该计数器无法读取，但只可使用 HRTIM_REPxR 寄存器中的自动重载值进行编程。

定时器使能后（TXCEN 位置 1），重复计数器会初始化为 HRTIM_REPxR 寄存器的内容。定时器使能后，每次计数器由于复位事件或计数器翻转而清空时，重复计数器都会减 1。当重复计数器达到 0 后，会发出 REP 中断或 DMA 请求（若使能，使用 HRTIM_DIER 寄存器中的 REPIE 和 REPDE 位）。

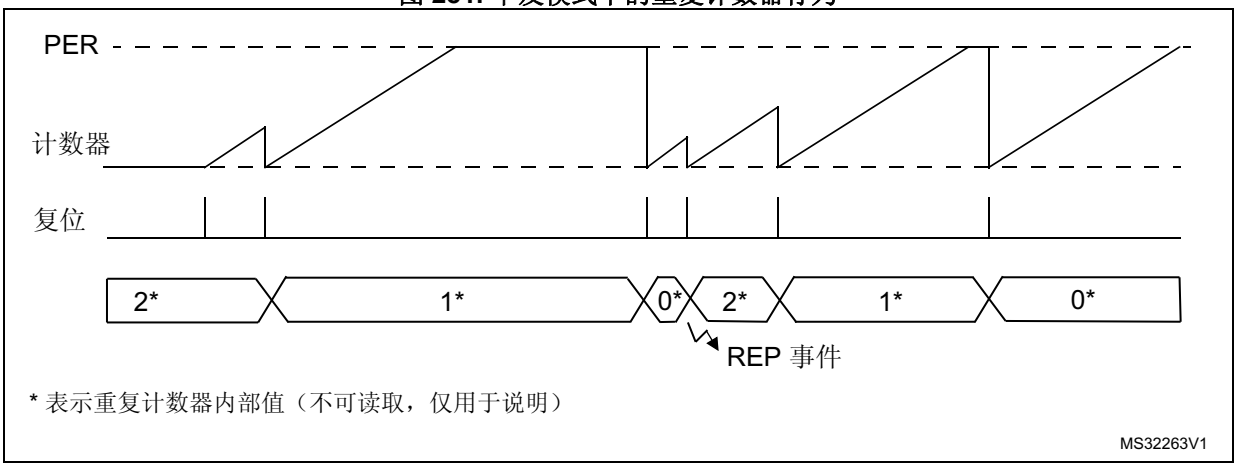
如果 HRTIM_REPxR 寄存器设为 0，会在每个周期产生中断。如果值大于 0，则会在 (HRTIM_REPxR + 1) 个周期后产生 REP 中断。*图 280* 显示了连续模式下重复计数器取不同值时的操作。

图 280. 连续模式下的重复率与 HRTIM_REPxR 内容



无论在连续模式还是单发模式下，如果计数器在达到周期值（可变频率操作）之前复位，则也可使用重复计数器（如下面的*图 281* 所示）。复位会使重复计数器在计数器使能后（TxCEN 位置 1）执行第一次启动时递减。

图 281. 单发模式下的重复计数器行为



来自 hrtim_in_sync[3:1] 源的复位或启动事件会像其他任何复位事件一样使重复计数器递减。但在通过 SYNCIN 启动的单发模式下（HRTIM_TIMxCR 寄存器中的 SYNCSTRTx 位置 1），重复计数器仅会在周期后出现第一个复位事件时递减。任何后续的复位事件都不会更改重复计数器的值，直至计数器通过新的 hrtim_in_sync[3:1] 输入请求重启。

置位/复位纵横开关

“置位”事件相当于跳变为输出有效状态，而“复位”事件则相当于跳变为输出无效状态。波形的极性在输出级中定义，以适应正逻辑或负逻辑外部组件：对于正极性 (POLx = 0)，有效电平对应于逻辑电平 1；对于负极性 (POLx = 1)，有效电平对应于逻辑电平 0。

每个定时单元都会控制两路输出的置位/复位纵横开关。这两路输出可通过多达 32 种事件置位、复位或切换，这些事件可从以下源中选择：

- 定时单元：周期、比较 1..4、寄存器更新（6 个事件）
- 主定时器：周期、比较 1..4、HRTIM 同步（6 个事件）
- 所有其他定时单元（例如，对于定时器 A，则为定时器 B..E）：TIMEVNT1..9（*表 286* 中介绍的 9 个事件）
- 外部事件 EXTEVNT1..10（10 个事件）
- 软件强制（1 个事件）

事件源会进行或运算，可同时选择多个事件。

每路输出均由两个 32 位寄存器控制，一个寄存器包含置位编码 (HRTIM_SETxyR)，另一个寄存器包含复位编码 (HRTIM_RSTxyR)，其中 x 代表定时单元 A..E，y 代表输出 1 或 2（例如 HRTIM_SETA1R、HRTIM_RSTC2R...）。

如果为置位和复位选择了相同事件，则会切换输出状态。每个 t_{HRTIM} 周期输出状态的切换次数不能超过 1 次：如果同一周期中有两个连续的切换事件，则仅会考虑第一个切换事件。

仅当计数器使能后（TxCEN 位置 1），才会考虑置位和复位请求，但软件在定时器启动时强制请求允许预置输出的情况除外。

表 286 汇总了来自其他定时单元的可用于置位和复位输出的事件。编号对应于寄存器中列出的定时器事件（如 TIMEVNTx），空白位置表示不可用事件。

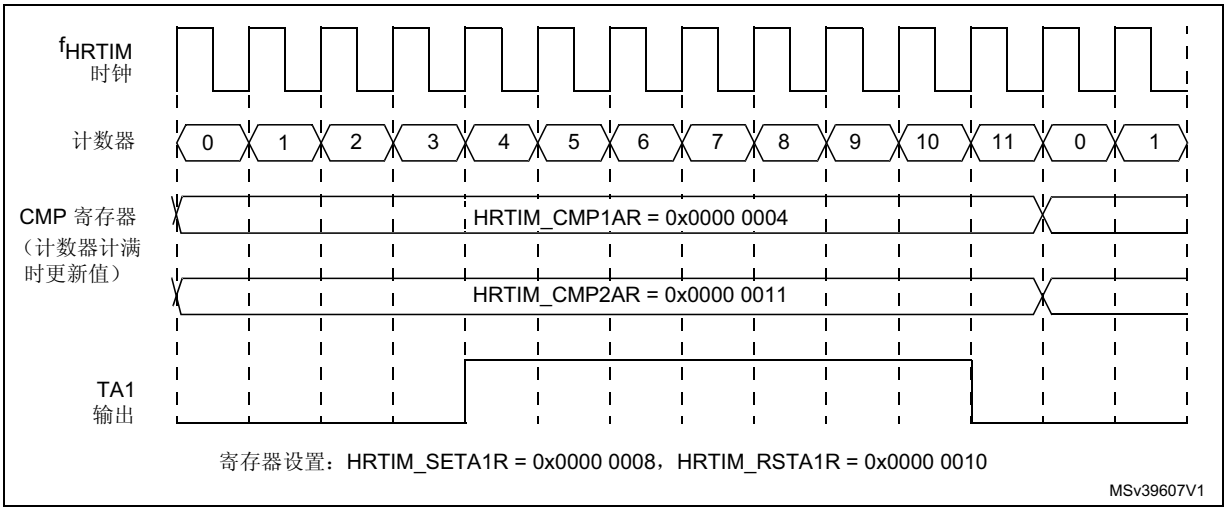
例如，定时器 A 输出可由以下事件置位或复位：定时器 B 比较 1、2 和 4，定时器 C 比较 2 和 3... 定时器 E 比较 3 将列为 HRTIM_SETA1R 中的 TIMEVNT8。

表 286. 定时器 A 到 E 之间的事件映射

源		定时器 A				定时器 B				定时器 C				定时器 D				定时器 E			
		CMP1	CMP2	CMP3	CMP4	CMP1	CMP2	CMP3	CMP4	CMP1	CMP2	CMP3	CMP4	CMP1	CMP2	CMP3	CMP4	CMP1	CMP2	CMP3	CMP4
目标	定时器 A	-	-	-	-	1	2	-	3	-	4	5	-	6	7	-	-	-	-	8	9
	定时器 B	1	2	-	3	-	-	-	-	-	-	4	5	-	-	6	7	8	9	-	-
	定时器 C	-	1	2	-	-	3	4	-	-	-	-	-	-	5	-	6	-	7	8	9
	定时器 D	1	-	-	2	-	3	-	4	5	-	6	7	-	-	-	-	8	-	-	9
	定时器 E	-	-	1	2	-	-	3	4	5	6	-	-	7	8	-	9	-	-	-	-

图 282 显示了如何通过两个比较事件生成 PWM 信号。

图 282. 比较事件对输出的操作：发生比较 1 时置位，发生比较 2 时复位



发生更新事件时置位/复位

半占空比模式

此模式用于生成占空比固定为 50%、频率可变的方波信号（通常用于使用谐振拓扑的转换器），允许在设定新周期时自动将占空比强制设为周期值的一半。

要使能此模式，应向 HRTIM_TIMxCR 寄存器中的 HALF 位写入 1。HRTIM_PERxR 寄存器写入数值后，会自动将比较 1 值更新为 HRTIM_PERxR/2 值。

生成方波的输出必须编程为在发生 CMP1 事件时进行一次跳变，在发生周期事件时进行一次跳变，具体如下：

– HRTIM_SETxyR = 0x0000 0008，HRTIM_RSTxyR = 0x0000 0004 或
– HRTIM_SETxyR = 0x0000 0004，HRTIM_RSTxyR = 0x0000 0008

HALF 模式会覆盖 HRTIM_CMP1xR 寄存器的内容。访问 HRTIM_PERxR 寄存器仅会更新比较 1 内部寄存器。用户可访问的 HRTIM_CMP1xR 寄存器不会更新为 HRTIM_PERxR / 2 的值。

当使能预装载（PREEN = 1、MUDIS、TxUDIS）时，则会在发生更新事件时刷新比较 1 活动寄存器。如果禁止预装载 (PREEN= 0)，则在 HRTIM_PERxR 写入数值后，比较 1 活动寄存器会立即更新。

如果使能了 HALF 模式，周期必须大于或等于 6 个 f_{HRTIM} 时钟周期。

捕获

定时单元能够在内部和外部事件的触发下捕获计数器值。捕获的目的是：

- 测量事件到达时间或发生间隔。
- 在自动延迟模式下更新比较 2 和比较 4 值（请参见 *自动延迟模式*）。

捕获以 f_{HRTIM} 分辨率执行。

定时器包含 2 个捕获寄存器：HRTIM_CPT1xR 和 HRTIM_CPT2xR。捕获触发事件在 HRTIM_CPT1xCR 和 HRTIM_CPT2xCR 寄存器中编程。

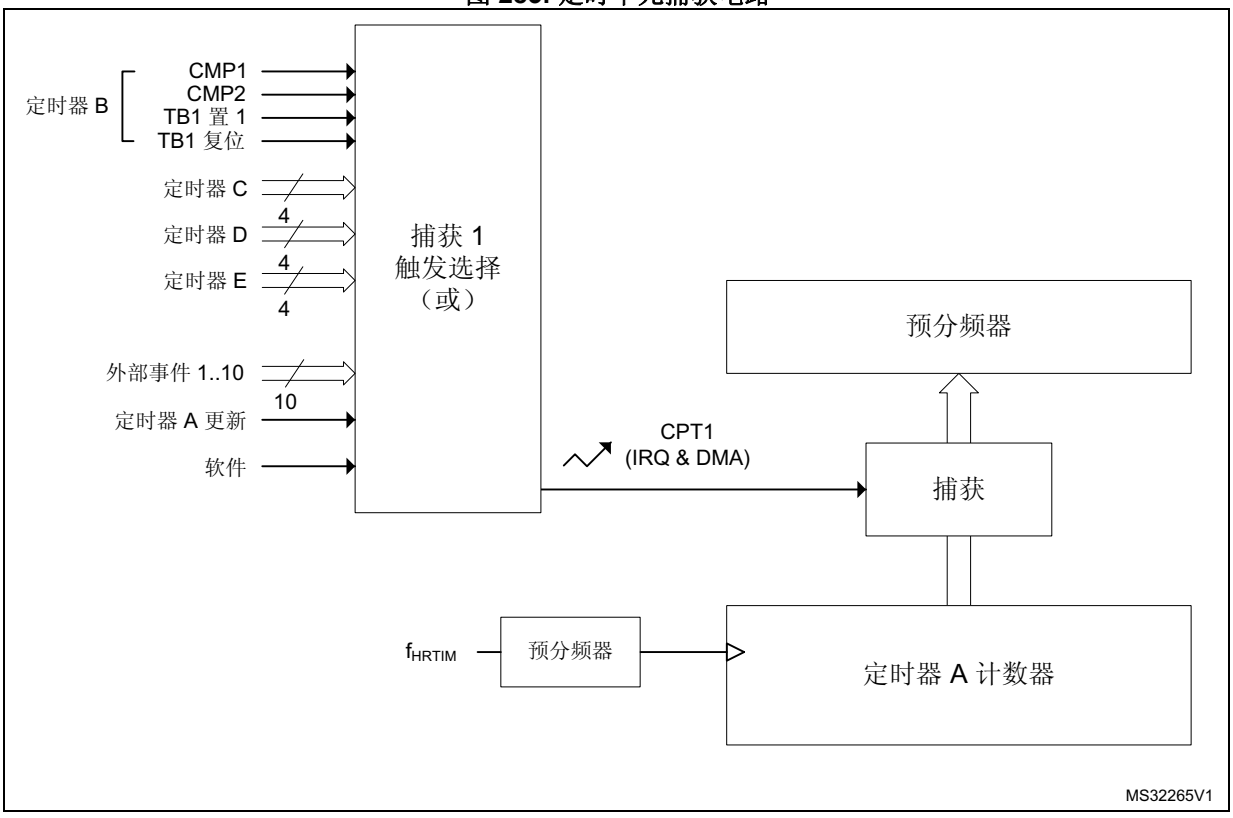
定时单元计数器的捕获可通过多达 28 种事件触发，这些事件可同时在 HRTIM_CPT1xCR 和 HRTIM_CPT2xCR 寄存器中选择，具体包括以下触发源：

- 外部事件，EXTEVNT1..10（10 个事件）
- 所有其他定时单元（例如，对于定时器 A，则为定时器 B..E）：比较 1、2 和输出 1 置位/复位事件（16 个事件）
- 定时单元：更新（1 个事件）
- 软件捕获（1 个事件）

可同时选择多个事件处理多个捕获触发信号。在这种情况下，会对多个并发触发请求进行或运算。如果 HRTIM_TIMxDIER 寄存器中的 CPTxIE 和 CPTxDE 位已置 1，捕获可生成中断或 DMA 请求。

电路未采用避免重复捕获的机制：即使前一个值未读取、或者捕获标志未清零，也会触发新捕获。

图 283. 定时单元捕获电路



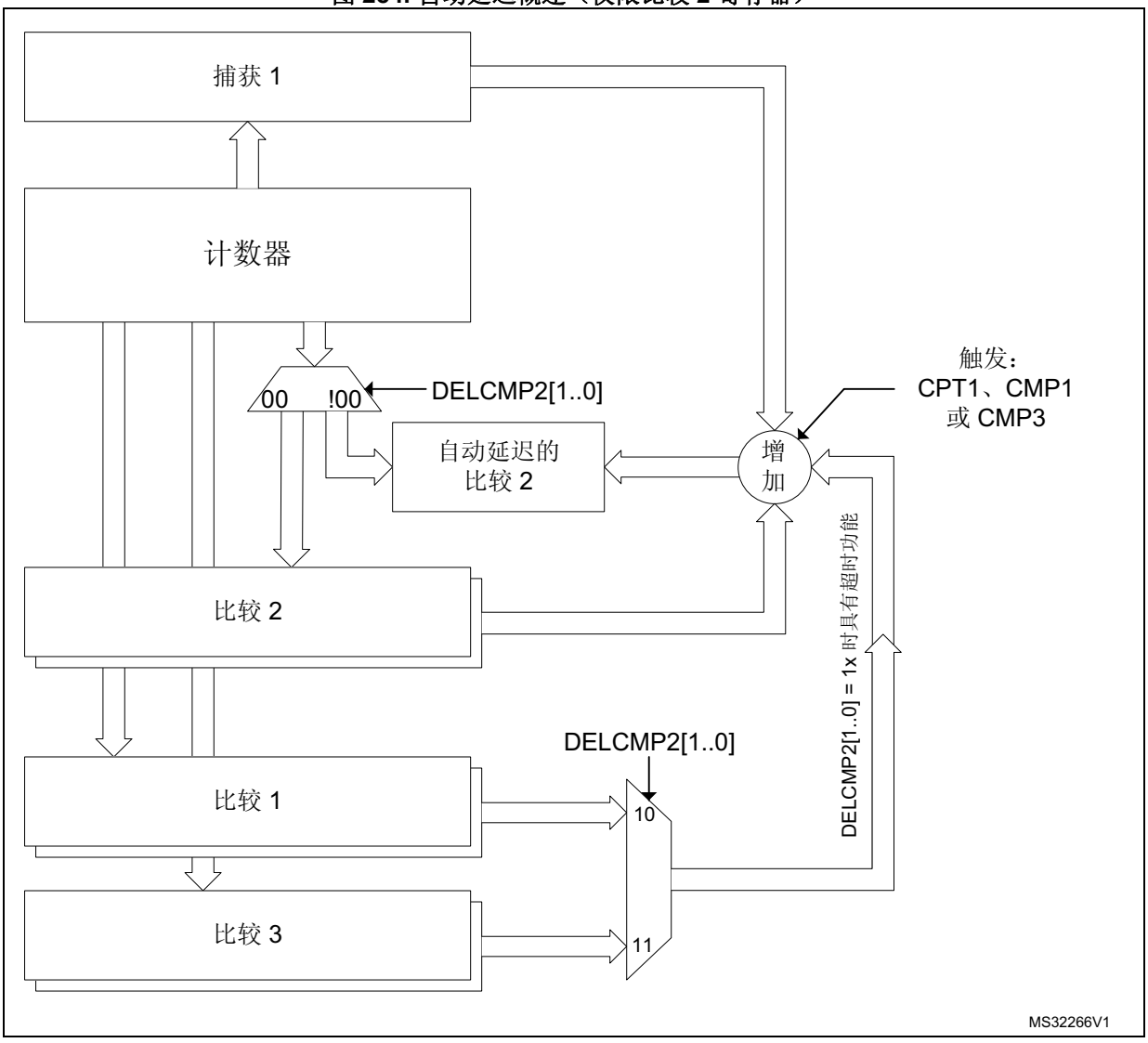
自动延迟模式

此模式可相对于捕获事件生成比较事件，这样一来，便可在捕获后的设定时间进行输出更改等操作。在这种情况下，会独立于定时器计数器值进行比较匹配。这样可生成时序与外部事件同步的波形，而无需进行软件计算和中断维护。

只要未触发捕获事件，便会忽略 HRTIM_CMPxR 寄存器的内容（计数器值与比较值相匹配的情况下，不会生成比较事件）。一旦触发了捕获事件，则会对在 HRTIM_CMPxR 中编程的比较值与 HRTIM_CPTxyR 中捕获的计数器值求和，并会用得出的结果更新内部自动延迟比较计数器，如*图 284* 所示。自动延迟比较寄存器属于定时单元内部的寄存器，不能读取。HRTIM_CMPxR 预装载寄存器的内容在计算后不会修改。

此特性仅适用于比较 2 和比较 4 寄存器。比较 2 寄存器与捕获 1 相关联，而比较 4 寄存器与捕获 2 相关联。与常规模式一样，HRTIM_CMP2xR 和 HRTIM_CMP4xR 比较寄存器不能编程为小于 3 个 f_{HRTIM} 时钟周期的值。

图 284. 自动延迟概述（仅限比较 2 寄存器）



自动延迟比较寄存器的有效期从捕获开始，到周期事件为止：计数器达到周期值后，系统会重新设置，比较寄存器在发生捕获之前会处于禁用状态。

HRTIM_TIMxCR 寄存器中的 DELCMP2[1:0] 和 DELCMP4[1:0] 位可配置自动延迟模式，具体如下：

- 00
  常规比较模式：会直接将 HRTIM_CMP2xR 和 HRTIM_CMP4xR 寄存器的内容与计数器值进行比较。
- 01
  自动延迟模式：会重新计算比较 2 和比较 4 寄存器值，并在捕获 1/2 事件后用计算值与计数器值进行比较。
- 1X
  具有超时的自动延迟模式：会重新计算比较 2 和比较 4 寄存器值，并在捕获 1/2 事件后用计算值与计数器值进行比较；如果缺少捕获 1/2 事件，则在比较 1 匹配 (DELCMPx[1:0]= 10) 或比较 3 匹配 (DELCMPx[1:0]= 11) 后用计算值与计数器值进行比较，以便实现超时功能。

进行捕获时，会与 (HRTIM_CMP2/4xR + HRTIM_CPT1/2xR) 值进行比较。如果周期内未触发捕获，行为将取决于 DELCMPx[1:0] 值：

- DELCMPx[1:0] = 01：未生成比较事件
- DELCMPx[1:0] = 10 或 11：与 2 个比较寄存器值之和进行比较（例如 HRTIM_CMP2xR + HRTIM_CMP1xR）。如果捕获是在 CMPx + CMP1（或 CMPx + CMP3）后触发的，则不会考虑 捕获。

下一 PWM 周期开始时会再次使能捕获。

如果自动延迟求和的结果大于 0xFFFF（溢出），则会忽略该值，并且新周期开始之前不会生成比较事件。

注： *如果从一个值重新编程为另一个值，以便正确地重新初始化自动延迟机制，则必须复位 DELCMPx[1:0] 位域，例如：*

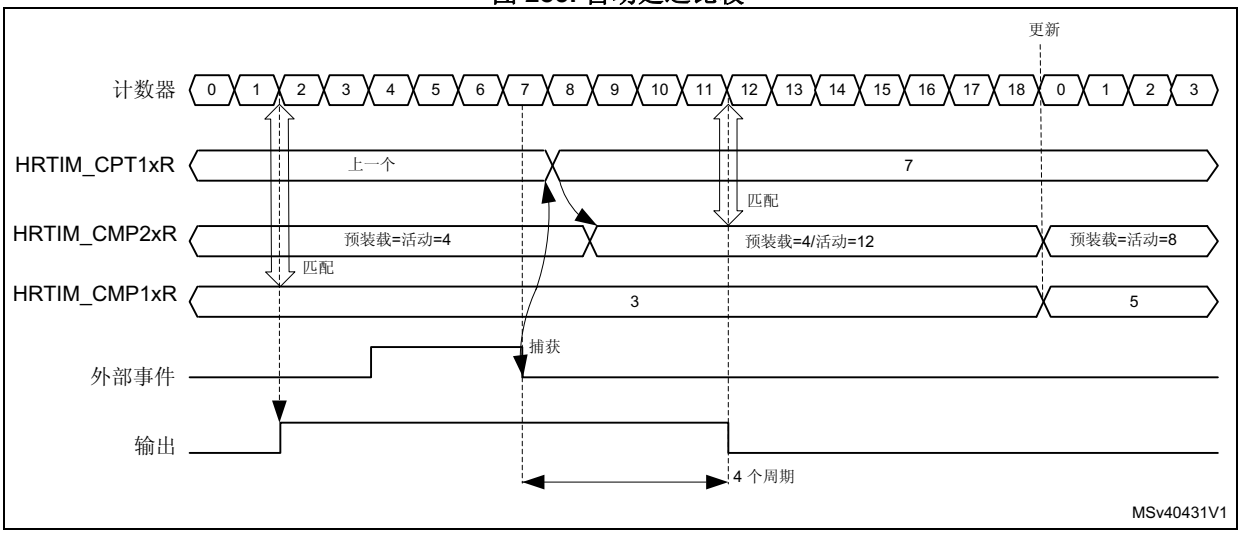
- DELCMPx[1:0] = 10
- DELCMPx[1:0] = 00
- DELCMPx[1:0] = 11

图 285 举例说明了以下信号是怎样生成的：

- 计数器等于比较 1 值时，输出置位
- 给定外部事件下降沿 4 个周期后，输出复位

注： *为了简化示意图，图中所示的外部事件信号不含任何重新同步延迟：实际情况下，由于需要通过内部重新同步阶段来处理外部输入信号，因此下降沿与捕获事件之间会有 1 到 2 个 f_{HRTIM} 时钟周期的延迟。*

图 285. 自动延迟比较



使用常规比较通道（例如比较 1）进行输出置位：计数器与比较寄存器的内容匹配后，输出会立即变为有效状态。

使用延迟比较进行输出复位：仅当发生了捕获事件时，才会生成比较事件。如果计数器与延迟比较值（计数值值 = 4）匹配，则不会生成事件。捕获事件由外部事件触发后，捕获寄存器的内容会立即与延迟比较值相加，得出新的比较值。示例中，自动延迟值 4 与的捕获值 7 相加，得出自动延迟比较寄存器中的值 11。从此时起，可生成比较事件，并会在计数器等于 11 时生成，比较事件会使输出复位。

自动延迟模式下的重复捕获管理

使能自动延迟模式时（DELCMPx[1:0] = 01、10、11），将阻止重复捕获。

如果同一计数周期内出现多个捕获请求，只会考虑使用第一个捕获请求来计算自动延迟比较值。仅可在以下条件下进行新捕获：

- 自动延迟比较具有匹配的计数器值（比较事件）
- 计数器已翻转（周期）
- 定时器已复位

更改自动延迟比较值

如果已预装载自动延迟比较值（PREEN 位置 1），则无论比较寄存器是何时写入数值的、是否发生了捕获事件（请参见*图 285*，其中，延迟是在计数器翻转时更改的），都会在发生下一更新事件（例如周期事件）时考虑新的比较值。

如果已禁止预装载（PREEN 位复位），则即使比较值在捕获事件发生后进行了修改，也会会立即考虑新的比较值，如下例所示：

1. 在 t1，DELCMP2 = 1
2. 在 t2，CMP2_act = 0x40 => 比较禁止
3. 在 t3，发生捕获事件，捕获到值 CPTR1 = 0x20。=> 比较使能，比较值 = 0x60
4. 在 t4，CMP2_act = 0x100（计数器值达到 CPTR1 + 0x40 之前）=> 比较仍处于使能状态，新比较值 = 0x120
5. 在 t5，计数器值达到周期值 => 比较禁止，cmp2_act = 0x100

同样，如果 CMP1(CMP3) 的值在 DELCMPx = 10 或 11 时发生了变化，并且已禁止预装载：

1. 在 t1，DELCMP2 = 2
2. 在 t2，CMP2_act = 0x40 => 比较禁止
3. 在 t3，发生 CMP3 事件 - 发生捕获 1 事件之前，CMP3_act = 0x50 => 比较使能，比较值 = 0x90
4. 在 t4，CMP3_act = 0x100（计数器值达到 0x90 之前）=> 比较仍处于使能状态，将在 CMP3_act= 0x140 时发生比较 2 事件。

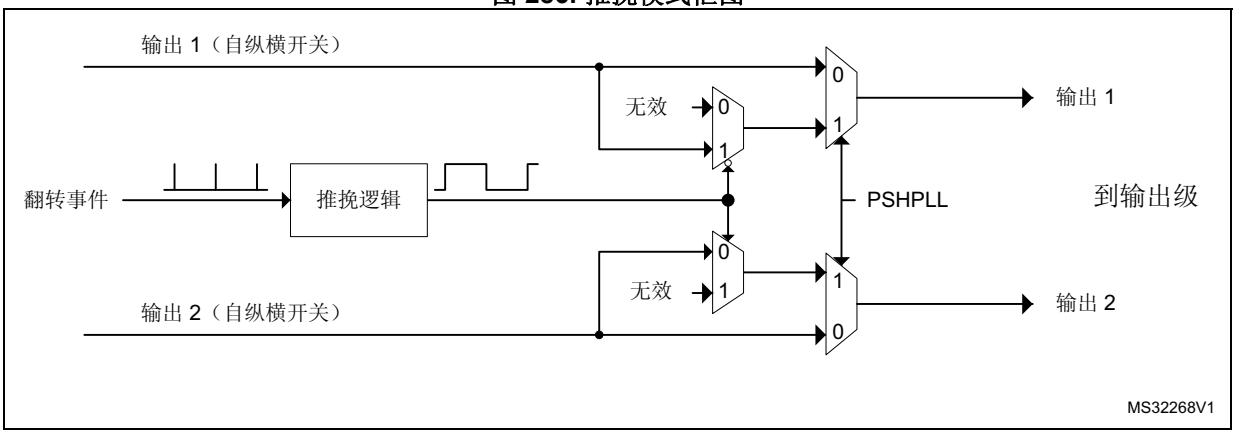
推挽模式

该模式的主要目的是使用推挽拓扑驱动转换器。如果需要使用延迟空闲保护（通常用于谐振转换器），也需要使能该模式（请参见*第 37.3.9 节：延迟保护*）。

通过将 HRTIM_TIMxCR 寄存器中的 PSHPLL 位置 1，即可使能推挽模式。

在该模式下，会按照周期将纵横开关生成的信号交替地施加到输出 1 和输出 2 上（如果信号施加到输出 1 上，则输出 2 保持其无效状态，反之亦然）。重定向速率（推挽频率）由定时器的周期事件定义，如*图 286* 所示。推挽周期是定时器计数周期的二倍。

图 286. 推挽模式框图



仅当定时器在连续模式下工作时，才能使用推挽模式：计数器使能（TxCEN 置 1）后不得复位。需要禁止定时器以停止推挽操作，并且重新使能计数器之前需要将其复位。

两路输出的信号波形由 HRTIM_SETxyR 和 HRTIM_RSTxyR 定义。需要使 HRTIM_SETx1R = HRTIM_SETx2R 且 HRTIM_RSTx1R = HRTIM_RSTx2R，才能使两路输出的波形完全相同，并实现平衡的操作。不过，仍可对两路输出进行不同的编程，以实现其他用途。

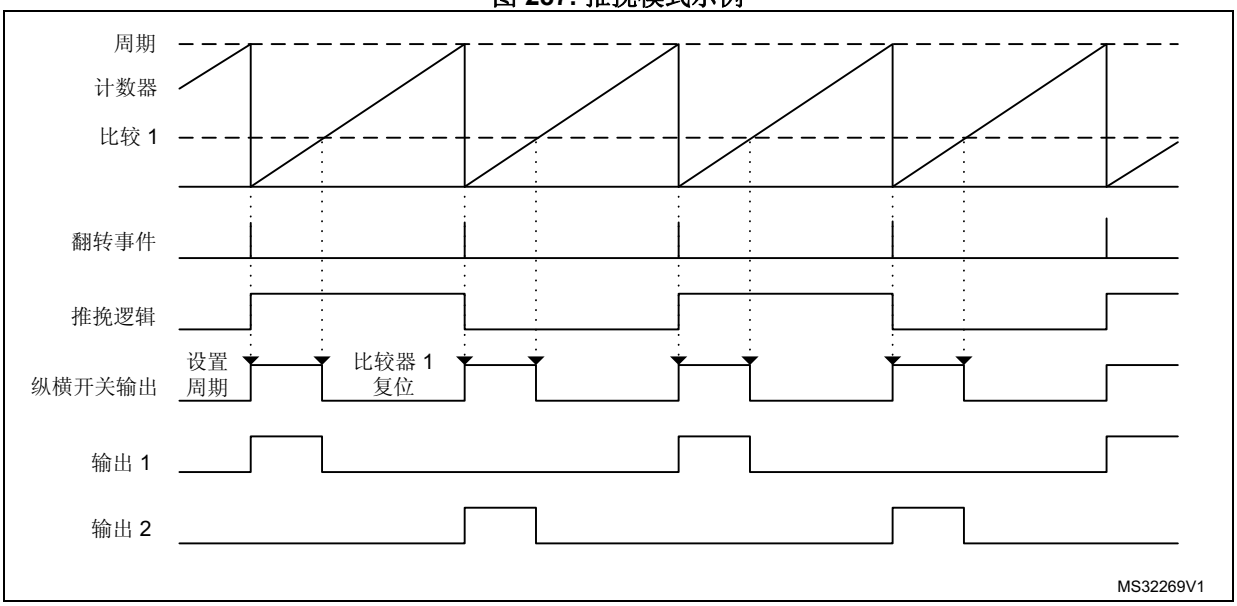
注： *如果使能了死区，则不能使用推挽操作（这两种功能互斥）。*

HRTIM_TIMxISR 中的 CPPSAT 状态位指示目前哪一路输出上的信号有效。CPPSTAT 在推挽模式禁用时复位。

在*图 287* 中提供的示例中，定时器内部波形的定义如下：

- 发生周期事件时，输出置位
- 发生比较 1 匹配事件时，输出复位

图 287. 推挽模式示例



死区

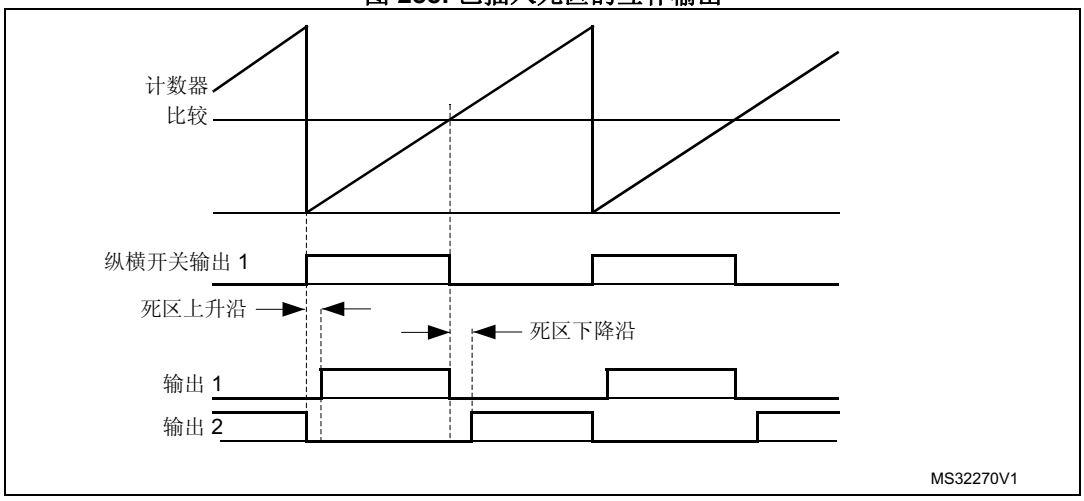
死区插入单元可通过单个参考波形生成一对互补信号，并且有效状态跳变之间的延迟可编程。这通常用于使用半桥或全桥的拓扑，可简化软件操作流程：只需要对 1 个波形进行编程和控制即可驱动两路输出。

死区插入通过将 HRTIM_OUTxR 寄存器的 DTEN 位置 1 来使能。互补信号基于为输出 1 定义的参考波形构建而成，使用 HRTIM_SETx1R 和 HRTIM_RSTx1R 寄存器：如果 DTEN 位置 1，则 HRTIM_SETx2R 和 HRTIM_RSTx2R 寄存器无效。

注： *死区不能与推挽模式同时使用。*

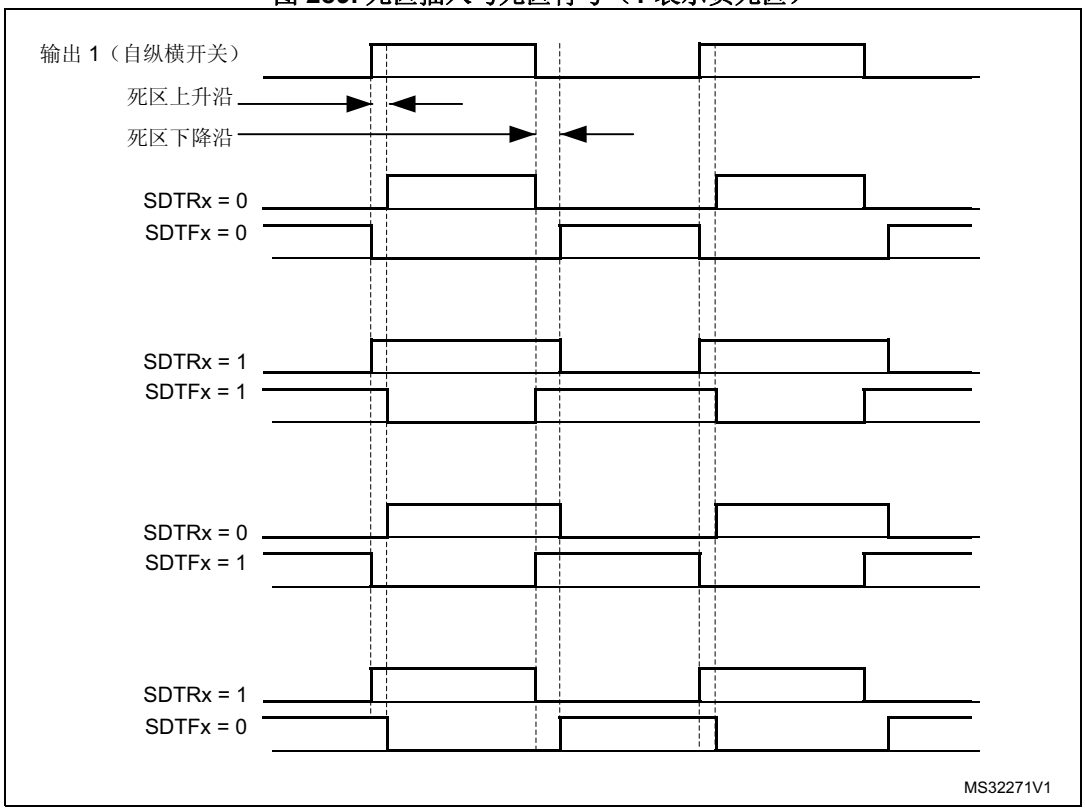
可按照与参考波形上升沿与下降沿的关系定义两个死区，如图 288 所示。

图 288. 已插入死区的互补输出



如果需要使用一些控制叠加，可定义负死区值，此时要使用死区符号位（HRTIM_DTxR 寄存器中的 SDTFx 和 SDTRx 位）。*图 289* 显示了各符号对应的互补信号波形。

图 289. 死区插入与死区符号（1 表示负死区）



死区值使用 DTFx[8:0] 和 DTRx[8:0] 位域定义，基于根据 DTPRSC[2:0] 位预分频的特定时钟，具体如下：

t_{DTx} = +/- DTx[8:0] x t_{DTG}

其中，x 为 R 或 F，t_{DTG} = ($2^{(DTPRSC[2:0])}$) x t_{HRTIM}。

表 287 给出了根据预分频值得出的分辨率和最大绝对值。

表287. 死区分辨率和最大绝对值

DTPRSC[2:0](1)	t_{DTG}	t_{DTx} 最大值	f_{HRTIM} = 400 MHz	
			t_{DTG} (ns)	$\|t_{DTx}\|$ 最大值 (µs)
011	t_{HRTIM}	511 * t_{DTG}	2.5	1.28
100	2 * t_{HRTIM}		5	2.56
101	4 * t_{HRTIM}		10	5.11
110	8 * t_{HRTIM}		20	10.22
111	16 * t_{HRTIM}		40	20.44

1. DTPRSC[2:0] 保留值 000、001、010。

图 290 到*图 293* 展示了在所有死区配置下，死区发生器对脉宽小于死区值的参考波形的处理方式。

图 290. 低脉宽的互补输出 (SDTRx = SDTFx = 0)

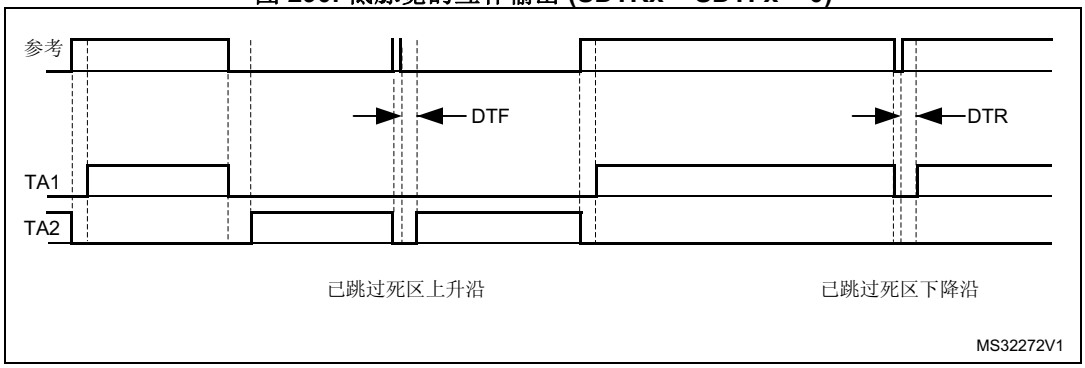


图 291. 低脉宽的互补输出 (SDTRx = SDTFx = 1)

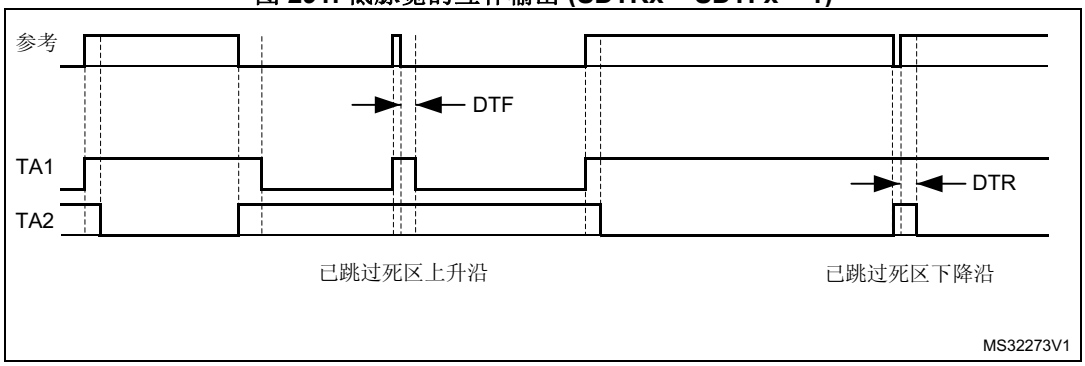


图 292. 低脉宽的互补输出（SDTRx = 0，SDTFx = 1）

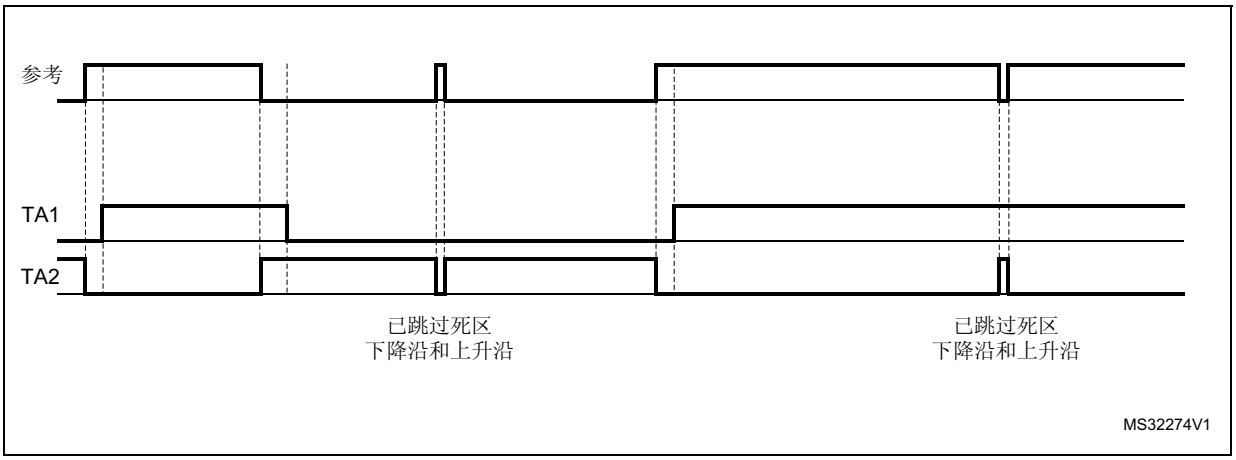
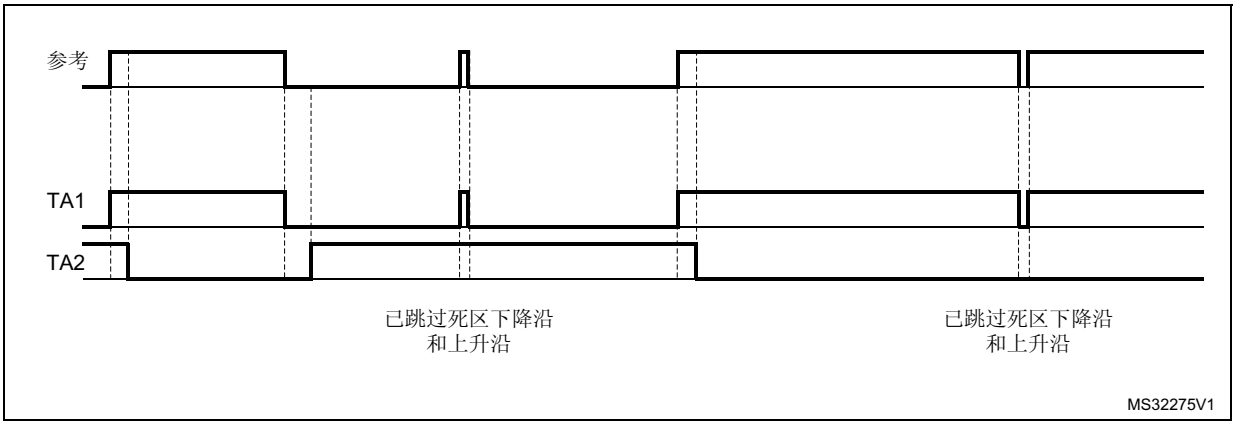


图 293. 低脉宽的互补输出（SDTRx = 1，SDTFx = 0）



出于安全方面的考虑，可使用 DTFLKx、DTRLKx、DTFSLKx 和 DTRSLKx 锁定死区的符号和/或数值，避免对死区寄存器进行误写操作。一旦这些位置 1，在下一次系统复位之前，关联位和位域将变为只读状态。

注意： 以下情况下，不得更改 DTEN 位：
- 定时器使能时（TxEN 位置 1）
- 定时器输出由另一定时器置位/复位时（TxEN 复位时）

否则会引发不可预测的行为。
因此，需要禁止定时器（TxCEN 位复位）并禁止相应的输出。

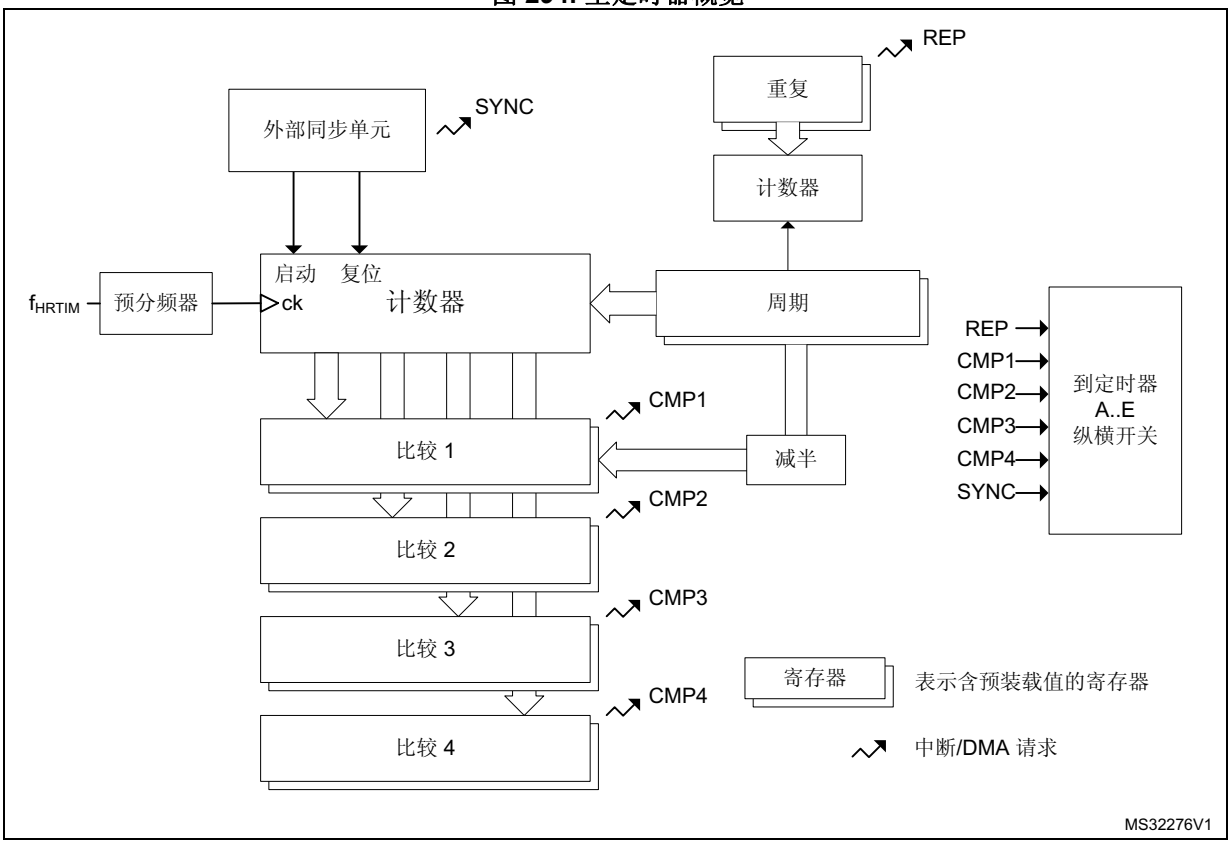
对于 DTEN 必须在突发模式使能的同时置 1 以确保进入突发模式后就有死区的这一特殊情况（BME = 1、DIDL = 1、IDLEM = 1），需要在将 DTEN 位置 1 之前通过软件命令将两路输出强制设为其 IDLES 状态（SST、RST 位）。这样做是为了避免在死区尚未使能之前就进入突发模式时会造成的不良后果。

37.3.5 主定时器

主定时器的主要用途是向 5 个定时单元提供公用信号，以便进行同步或置位/复位输出。主定时器不会直接控制任何输出，但仍可间接地供置位/复位纵横开关使用。

图 294 给出了主定时器的概览。

图 294. 主定时器概览



主定时器采用的架构与定时单元非常相似，二者的不同之处如下：

- 主定时器未关联输出，也没有输出相关的控制
- 主定时器没有自己的纵横开关单元，也没有推挽或死区模式
- 主定时器只能通过外部同步电路复位
- 主定时器没有捕获单元，也没有自动延迟模式
- 主定时器不含外部事件消隐和窗口电路
- 主定时器的中断/DMA 请求数量有限：比较 1..4、重复、寄存器更新和外部同步事件

主定时器控制寄存器包含主定时器和 A..E 定时单元的所有定时器使能位。这样可通过单次写访问同时启动所有定时器。

主定时器还会利用 MCU 内部和外部（输入/输出）资源处理整个 HRTIM 定时器的外部同步（请参见*第 37.3.17 节：将 HRTIM 与其他定时器或 HRTIM 实例同步*）。

主定时器控制寄存器的映射偏移与定时单元寄存器的偏移相同。

37.3.6 置位/复位事件优先级和窄脉冲管理

本节介绍了在 3 个连续 t_{HRTIM} 周期内出现多个置位和/或复位请求时，输出波形是如何产生的。

每个 t_{HRTIM} 周期内都会执行仲裁，具体分 2 步：

1. 对于每个有效事件，会确定所需输出跳变（置位、复位或切换）。
2. 在有效事件之间执行预定义的仲裁（从最高优先级到最低优先级）：CMP4 ⟶ CMP3 ⟶ CMP2 ⟶ CMP1 ⟶ PER，请参见*并发置位请求/并发复位请求*。

如果两个不同源发出的置位和复位请求是同时的，则复位操作的优先级最高。

并发置位请求/并发复位请求

如果为置位事件选择了多个源，则在同一 f_{HRTIM} 时钟周期内出现多个置位请求时，会执行仲裁。

如果相邻定时器 (TIMEVNT1..9) 发出多个请求，则会考虑最先发出的请求。仲裁分 2 步执行，具体视 源而定（优先级从高到低）：CMP4 ⟶ CMP3 ⟶ CMP2 ⟶ CMP1。

如果主定时器在同一 f_{HRTIM} 时钟周期内发出多个请求，则会应用预定义的仲裁，并会考虑单个请求（优先级从高到低）：

MSTCMP4 → MSTCMP3 → MSTCMP2 → MSTCMP1 → MSTCMPER

如果在同一 f_{HRTIM} 时钟周期内出现多个定时器内部请求，则会应用预定义的仲裁，并会按照以下优先级顺序考虑请求，而不考虑有效定时（优先级从高到低）：

CMP4 → CMP3 → CMP2 → CMP1 → PER

注： *实际上，仅当同时使用自动延迟比较 2 和比较 4 时（也就是由于有效置位/复位与外部事件相关联，无法通过优先级确定的情况下），才要首先考虑这一点。在这种情况下，优先级最高的信号必须与 CMP4 事件相关联。*

最后，最高优先级会分配给非定时相关的事件：EXTEVNT1..10、RESYNC（来自 SYNC 事件（SYNCRSTx 或 SYNCSTRTx 置 1 时）或软件复位）、更新和软件置位 (SST)。

总之，对于同步事件，有效置位（复位）事件将在以下事件之间进行仲裁：

- 任意 TIMEVNT1..9 事件
- 来自主定时器的单个源（按照上文所述的固定仲裁）
- 来自定时器的单个源
- “非时间相关的事件”

仲裁原则同样适用于并发复位请求。在这种情况下，复位请求的优先级最高。

在预分频器时钟周期内发生的置位或复位事件会延迟到预分频时钟的下一有效边沿（针对计数器复位），即使仍在每个 t_{HRTIM} 周期进行仲裁的情况也不例外。

如果同一预分频器时钟周期内发生复位事件后又发生置位事件，将考虑最新事件。

37.3.7 外部事件全局调节

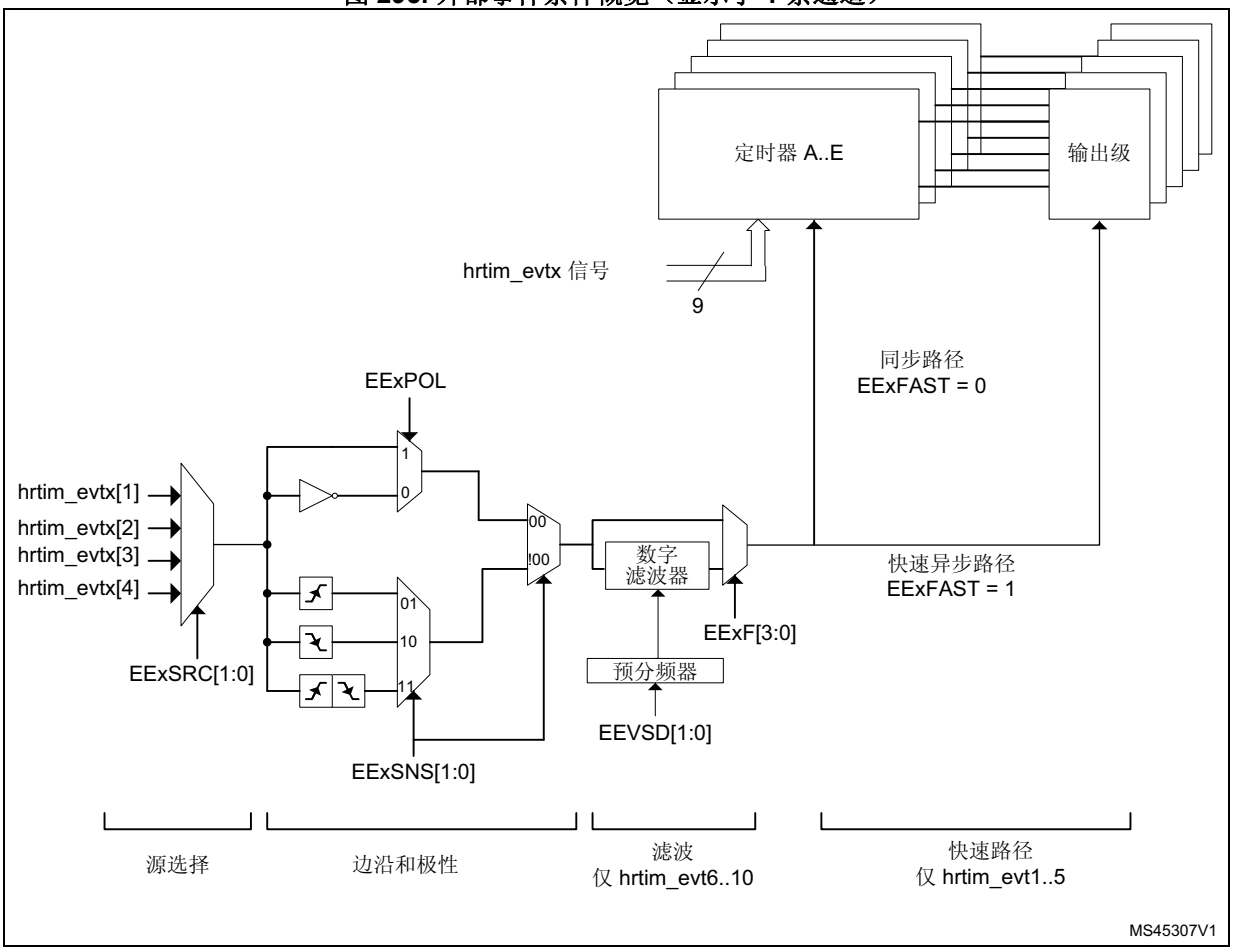
HRTIM 定时器可处理并非定时器中生成的事件，此类事件被称为“外部事件”。外部事件来自多个片上或片下源：

- 内置比较器
- 数字输入引脚（通常连接到片下比较器和过零检测器）
- 其他外设的片上事件（ADC 的模拟看门狗和通用定时器触发输出）

外部事件调节电路可为给定通道选择信号源（使用 4:1 多路复用器），并可将信号源转换为可由纵横开关单元处理的信息（例如，通过在外部事件通道上检测到的下降沿来触发输出复位）。

最多可调节 10 个外部事件通道，这些外部事件通道可同时用于 5 个定时器中的任何一个。由于这种调节通常取决于外部组件（比如过零检测器）和环境条件（滤波器设置通常与应用噪声级和信号有关），因此对于所有定时器是通用的。*图 295* 显示了单条通道的调节逻辑概览。

图 295. 外部事件条件概览（显示了 1 条通道）



10 个外部事件通过 HRTIM_EECR1 和 HRTIM EECR2 寄存器初始化：

- 使用 EExSRC[1:0] 位最多选择 4 个源
- 使用 EExSNS[1:0] 位选择采用电平有效还是边沿有效（上升沿、下降沿或两种边沿）
- 如果选择采用电平有效，则使用 EExPOL 位选择极性
- 使用 EExFAST 位为外部事件 1 到 5 使用低延迟模式（请参见*外部事件延迟*）

注： *即使 EESNS 位已复位（选择电平有效），用作复位、捕获、突发模式、ADC 触发和延迟保护的触发信号的外部事件也是边沿有效：如果 POL = 0，触发在外部事件上升沿有效；如果 POL = 1，触发在外部事件下降沿有效。*

计数器禁止后（TxCEN 位复位），会立即丢弃外部事件，以防止任何输出状态更改和计数器复位，但用作 ADC 触发信号的外部事件除外。

此外，还可以使用 HRTIM_EECR3 寄存器中的 EExF[3:0] 位为外部事件 6 到 10 使能数字噪声滤波器。

数字滤波器由计数器组成，需要使用 N 个有效采样来验证输出跳变。如果输入值在计数器达到 N 值之前发生变化，计数器会复位，跳变会被丢弃（视为伪事件）。如果计数器达到 N，跳变被视为有效，并会作为正确的外部事件进行传输。因此，数字滤波器会向正在进行滤波的外部事件添加延迟，延迟时长视采样时钟和滤波器长度（预期的有效采样数）而定。

采样时钟为 f_{HRTIM} 时钟或由 f_{HRTIM} 预分频而生成的特定时钟 f_{EEVS}，通过 HRTIM_EECR3 寄存器中的 EEVSD[1:0] 位定义。

表 288 总结了与 10 个外部事件通道相关联的可用源和特性。

表 288. 外部事件映射和关联特性

外部事件通道	快速模式	数字滤波器	均衡故障定时器 A、B、C	均衡故障定时器 D、E、F	Src1	Src2	Src3	Src4
1	有	-	-	-	PC10	COMP1	TIM1_TRGO	ADC1_AWD1
2	有	-	-	-	PC12	COMP2	TIM2_TRGO	ADC1_AWD2
3	有	-	-	-	PD5	-	TIM3_TRGO	ADC1_AWD3
4	有	-	-	-	PG11	OPAMP1(1)	TIM7_TRGO	ADC2_AWD1
5	有	-	-	-	PG12	-	LPTIM1 OUT	ADC2_AWD2
6	-	有	有	-	PB4	COMP1	TIM6_TRGO	ADC2_AWD3
7	-	有	有	-	PB5	COMP2	TIM7_TRGO	-
8	-	有	-	有	PB6	-	TIM6_TRGO	TTCAN_TMP
9	-	有	-	有	PB7	OPAMP1(1)	TIM15_TRGO	TTCAN_RTP
10	-	有	-	-	PG13	-	LPTIM2 OUT	TTCAN_SOC

1. OPAMP1_VOUT 可用作高分辨率定时器内部事件源。在这种情况下，OPAMP1_VOUT (PC4) 引脚必须配置为输入模式。来自 GPIO 引脚的数据通过引脚施密特触发器重定向到 HRTIM 外部事件。如果禁止 OPAMP1，则配置为输入模式的 PC4 引脚可用作 HRTIM 外部事件。

外部事件延迟

外部事件调节能够根据性能预期调整外部事件处理时间（以及关联的延迟）：

- 常规工作模式，在该模式下，会在对输出纵横开关进行操作之前，使用时钟对外部事件进行重新采样。此过程会增加一些延迟，但可以访问所有纵横开关功能，从而生成外部触发的高分辨率脉冲。
- 快速工作模式，在该模式下，外部事件与输出操作之间的延迟得到最大限度地缩短。该模式便于实现超快速过流保护等功能。

HRTIM_EECR1 寄存器中的 EExFAST 位可用于定义通道 1 到 5 的工作模式。这会影响输出脉冲上存在的延迟和抖动，具体见下表。

表 289. 输出置位/复位延迟和抖动与外部事件工作模式的关系

EExFAST	响应时间延迟	响应时间抖动	输出脉冲上的抖动 （计数器通过外部事件 复位）
0	5 到 6 个 f_{HRTIM} 时钟周期	1 个 f_{HRTIM} 时钟周期	无抖动，脉宽通过高分辨率保持
1	最短延迟（取决于使用比较器还是数字输入）	最短延迟	1 个 f_{HRTIM} 时钟周期的抖动，脉宽分辨率低至 t_{HRTIM}

当电平检测敏感时，设置 (EExSNS[1:0] = 00)，才能使用 EExFAST 模式；不能用于边沿检测敏感的设置。

可以对外部事件应用事件过滤（消隐和窗口，EExFLTR[3:0] != 0000，请参见*第 37.3.8 节*）。在这种情况下，EExLTCHx 位必须复位：不支持延迟模式，也不支持窗口超时功能。

注： *相关 EExFAST 位置 1 后，不得修改外部事件配置（源和极性）。*

快速外部事件不能用于切换输出：必须在 HRTIM_SETxyR 或 HRTIM_RSTxyR 寄存器中使能，而不能在两个寄存器中同时使能。

如果置位事件和复位事件（来自 2 个独立的快速外部事件）同时发生，则复位事件在纵横开关中的优先级最高，输出变为无效状态。

如果 EExFAST 位置 1，则在外部事件发生后的 11 个 f_{HRTIM} 时钟周期内，输出都不能更改。

图 296 和*图 297* 给出了进行输出置位/复位和计数器复位时对外部事件的实际响应时间示例。

图 296. 外部事件下降沿的延迟（计数器复位和输出置位）

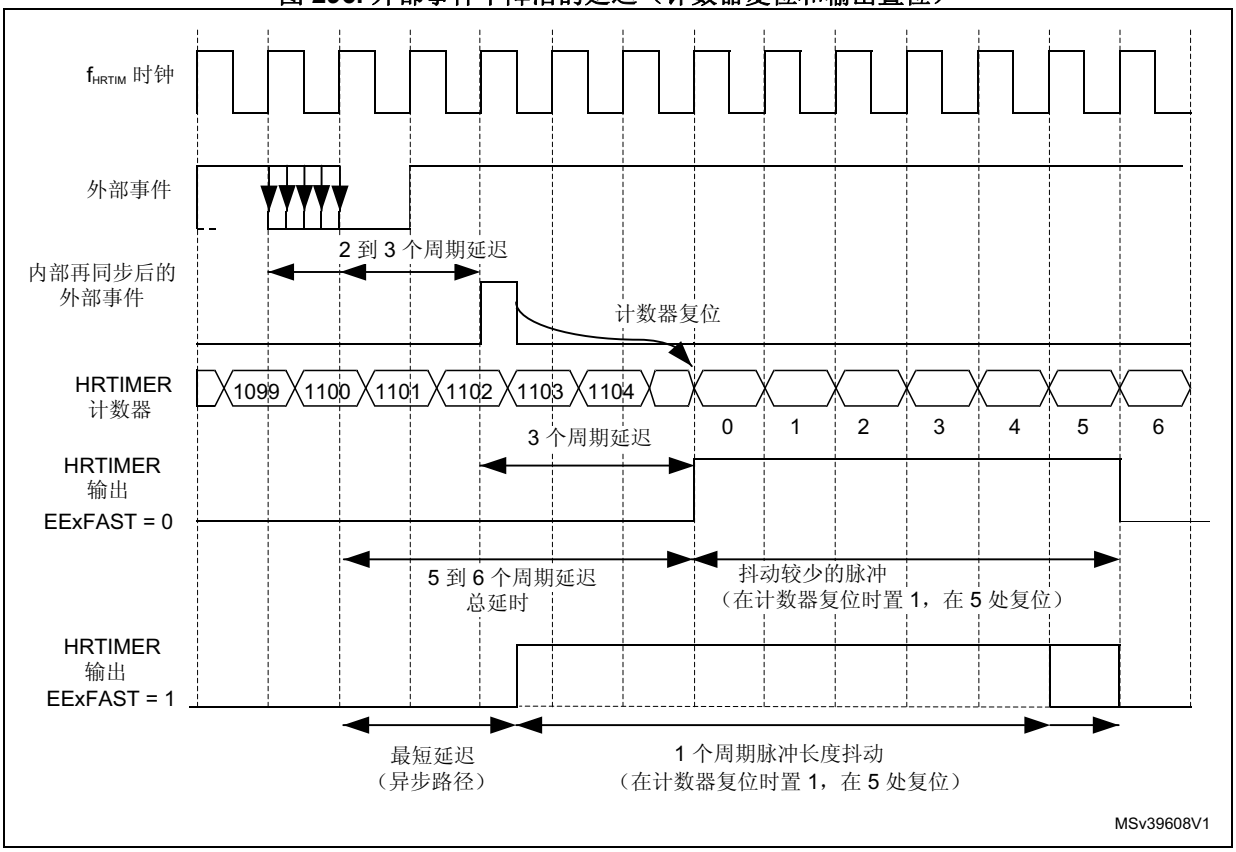
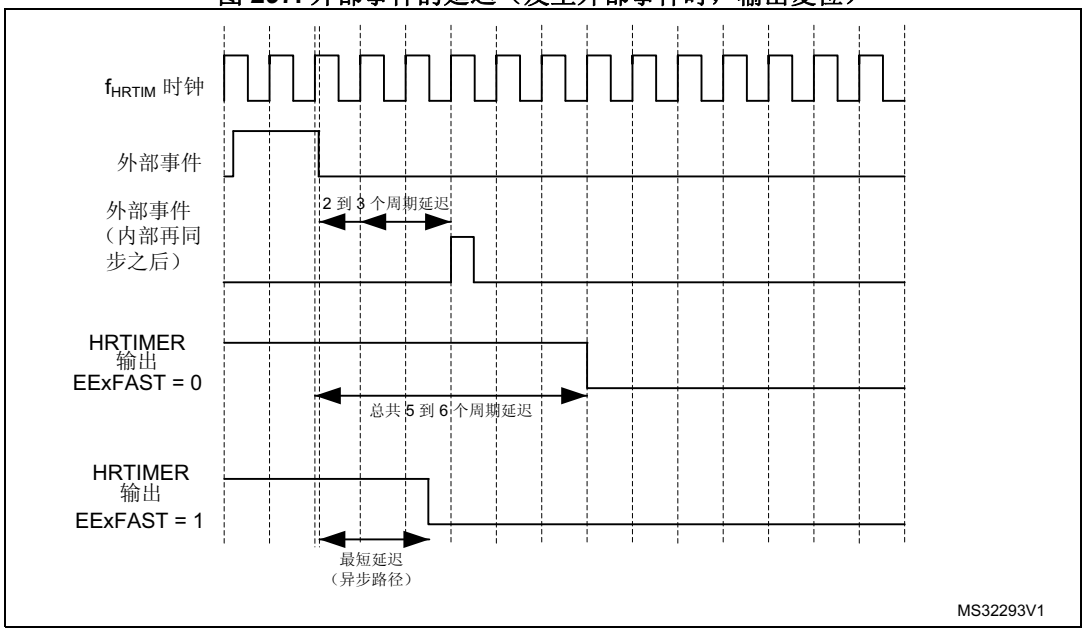


图 297. 外部事件的延迟（发生外部事件时，输出复位）



37.3.8 定时单元中的外部事件过滤

经过调节后，有 10 个外部事件可供所有定时单元使用。

这些事件可直接使用，且在定时单元计数器使能后（TxCEN 位置 1）立即生效。

此外，还可对这些事件进行过滤，以便在限制时间段内执行操作，该时间段通常与计数周期有关。可执行两种操作：

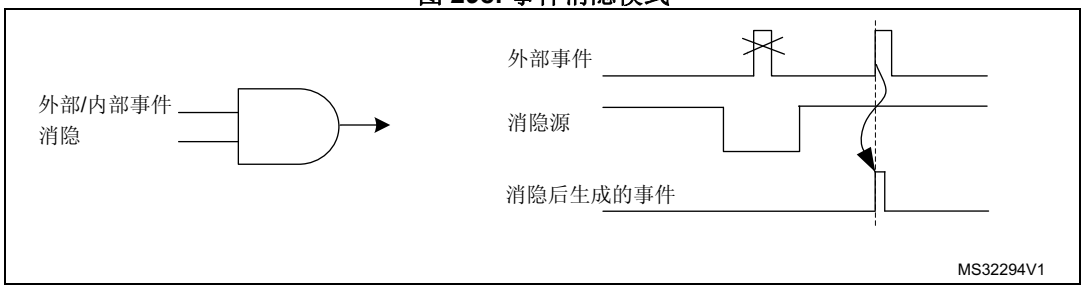
- 消隐模式，在定义的时间段内屏蔽外部事件
- 窗口模式，仅在定义的时间段内使能外部事件

这些模式通过 HRTIM_EEFxR1 和 HRTIM_EEFxR2 寄存器中的 HRTIM_EExFLTR[3:0] 位来使能。TimerA..E 这 5 个定时单元都具有针对这 10 个外部事件的可编程过滤器设置。

消隐模式

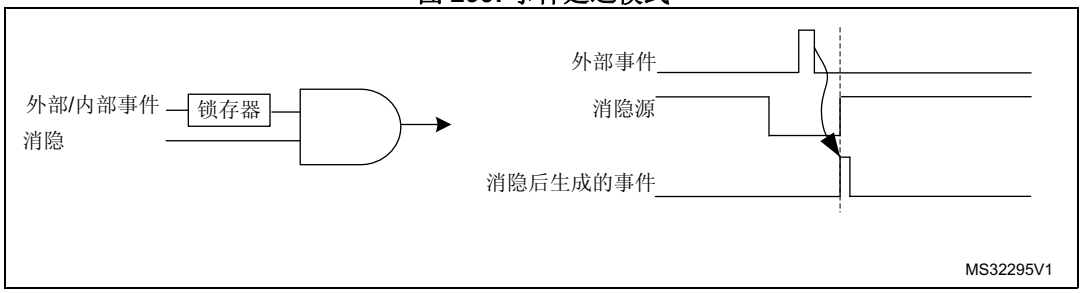
在事件消隐模式下（请参见*图 298*），如果外部事件发生在给定的消隐周期内，则会被忽略。举例来说，为了避免 PWM 周期开始时电流限制因开关噪声而失效，可使用此模式。当 EExFLTR[3:0] 位域的值在 0001 到 1100 范围内时，该模式有效。

图 298. 事件消隐模式



在事件延迟模式下，不会立即考虑外部事件，而是会将其保存（锁存）下来，并在消隐周期结束后立即生成外部事件，如*图 299* 所示。可将 HRTIM_EEFxR1 和 HRTIM_EEFxR2 寄存器中的 EExLTCH 位置 1 来使能此模式。

图 299. 事件延迟模式



消隐信号来自多个源：

- 定时器本身：消隐持续时间从计数器复位开始，到比较匹配为止（对于比较 1 到比较 4，EExFLTR[3:0] = 0001 到 0100）。
- 来自其他定时单元（EExFLTR[3:0] = 0101 到 1100）：消隐持续时间从选定的定时单元计数器复位开始，到其中一个比较匹配为止，或者可完全编程为 Tx2 输出上的波形。在这种情况下，只要 Tx2 信号无效，事件就会被屏蔽（不需要使能输出，而且会在输出级之前获取信号）。

EEXFLTR[3:0] 配置 0101 到 1100 在位说明中被称为 TIMFLTR1 到 TIMFLTR8，各定时单元的配置在位说明中的含义有所不同。*表 290* 给出了每个定时器的 8 个可用选项：CMPx 是指计数器复位到比较匹配期间进行消隐，Tx2 是指通过 HRTIM_SETx2 和 HRTIM_RSTx2 寄存器定义的定时单元 TIMx 输出 2 波形。举例来说，定时器 B (TIMFLTR6) 是定时器 C 输出 2 波形。

表290. 每个定时器的过滤信号映射

	源	定时器 A				定时器 B				定时器 C				定时器 D				定时器 E			
		CMP1	CMP2	CMP4	TA2	CMP1	CMP2	CMP4	TB2	CMP1	CMP2	CMP4	TC2	CMP1	CMP2	CMP4	TD2	CMP1	CMP2	CMP4	TE2
目标	定时器 A	-	-	-	-	1	-	2	3	4	-	5	6	7	-	-	-	-	8	-	-
	定时器 B	1	-	2	3	-	-	-	-	4	5	-	6	-	7	-	-	8	-	-	-
	定时器 C	-	1	-	-	2	-	3	4	-	-	-	-	5	-	6	7	-	-	8	-
	定时器 D	1	-	-	-	-	2	-	-	3	4	-	5	-	-	-	-	6	-	7	8
	定时器 E	-	1	-	-	2	-	-	-	3	-	4	5	6	-	7	8	-	-	-	-

图 300 和*图 301* 举例说明了常规模式和延迟模式下所有边沿有效和电平有效的外部事件消隐。

图 300. 采用边沿有效触发的外部触发消隐

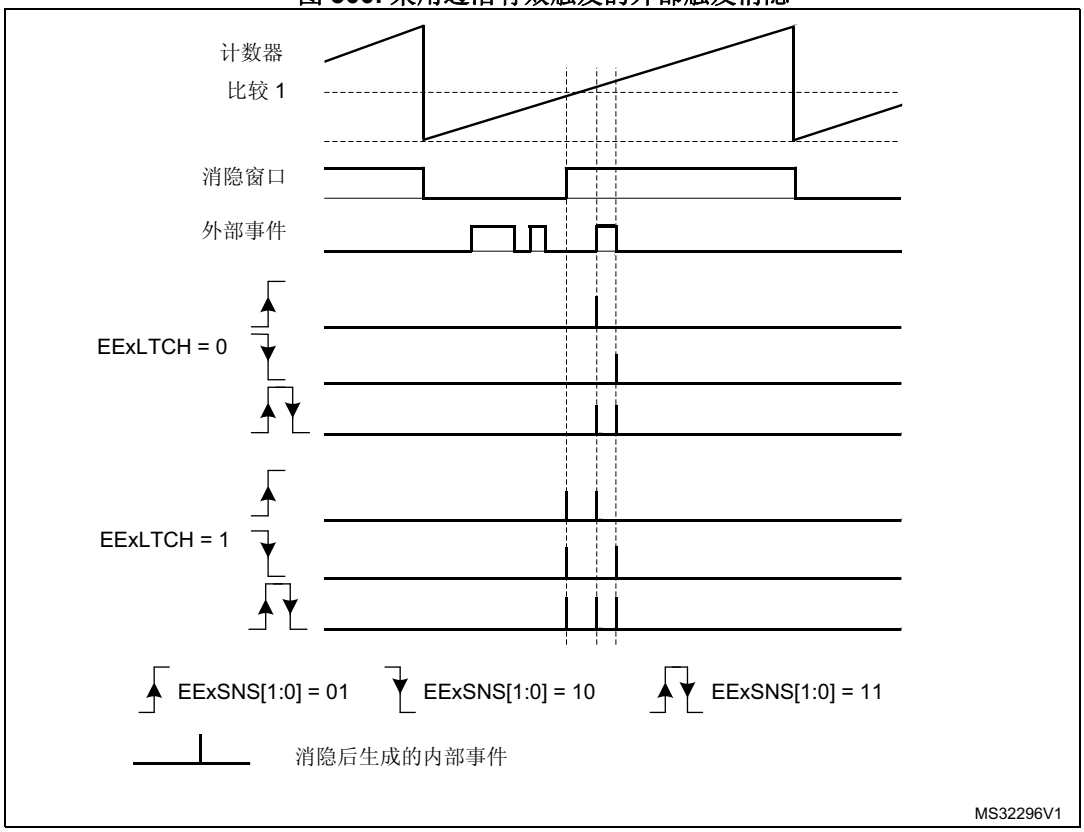
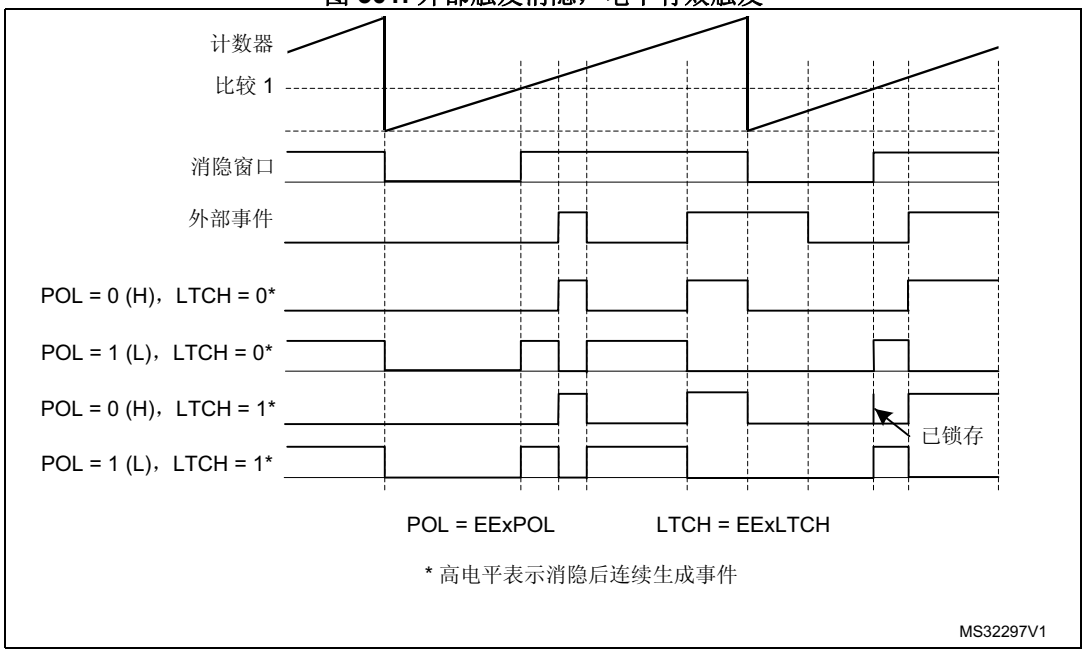


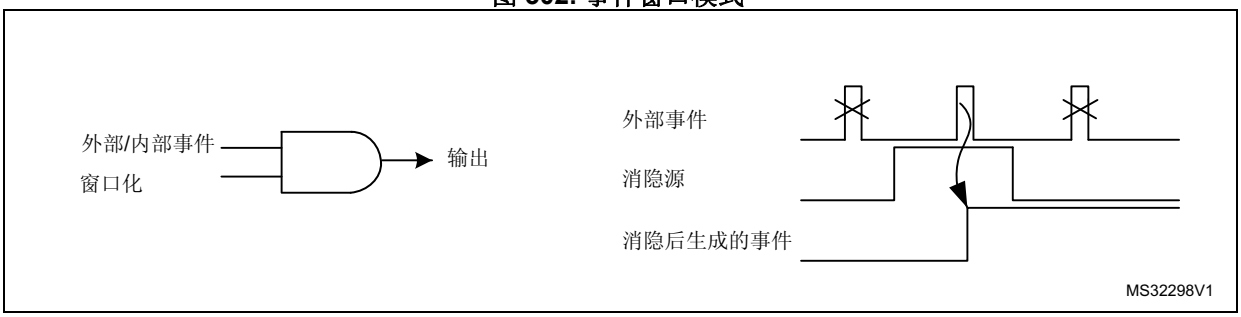
图 301. 外部触发消隐，电平有效触发



窗口模式

在事件窗口模式下，仅当事件在给定事件窗口内发生时才会被考虑，否则会被忽略。当 EExFLTR[3:0] 的值在 1101 到 1111 范围内时，该模式有效。

图 302. 事件窗口模式



EEFxR1 和 EEFxR2 寄存器中的 EExLTCH 可锁存信号，如果该位置 1：在这种情况下，如果事件在窗口期间发生，但在窗口结束时延迟，也会被接受。

- 如果 EExLTCH 位复位，且信号在窗口期间发生，则信号会直接通过。
- 如果 EExLTCH 位复位，并且无信号发生，则会在窗口结束时生成超时事件。

窗口模式可用于过滤同步信号。当缺少预期的同步事件时（例如在转换器启动期间），可通过超时生成功能强制生成默认同步事件。

每个外部事件窗口有 3 个源，其编码如下：

- 1101 和 1110：窗口持续时间从计数器复位开始，到比较匹配为止（分别是比较 2 和比较 3）。
- 1111：窗口与另一定时单元相关，持续时间从计数器复位开始，到其比较 2 匹配为止。源被称为 TIMWIN 的位中说明，请参见*表 291*。举例来说，定时器 B 中的外部事件可通过从定时器 A 计数器复位开始、到定时器 A 比较 2 为止的窗口进行过滤。

表 291. 每个定时器的窗口信号映射 (EEFLTR[3:0] = 1111)

目标	定时器 A	定时器 B	定时器 C	定时器 D	定时器 E
TIMWIN（源）	定时器 B CMP2	定时器 A CMP2	定时器 D CMP2	定时器 C CMP2	定时器 D CMP2

注： *如果外部事件是在快速模式下编程的，则不支持生成超时事件。*

图 303 和*图 304* 介绍了如何根据 EExLTCH 位的设置生成事件，以实现各种边沿有效和电平有效触发。为了便于理解，专门针对超时事件进行了说明。

图 303. 采用边沿有效触发的外部触发窗口

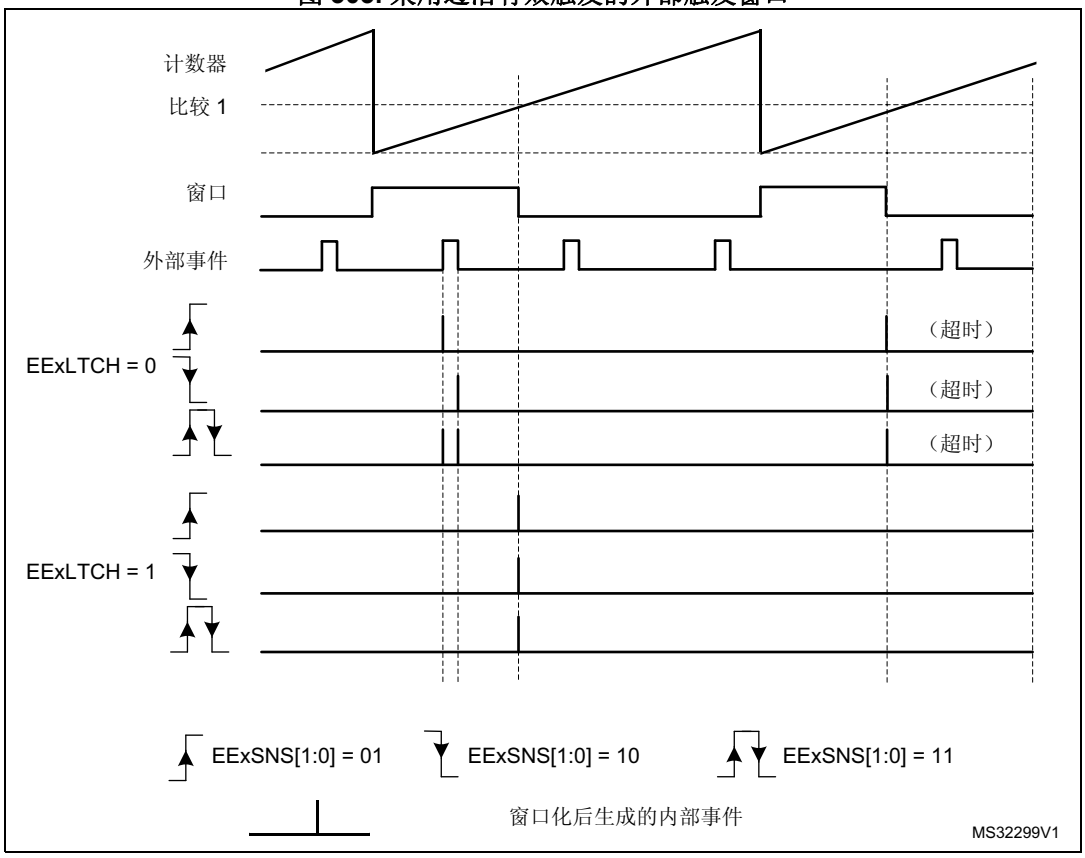
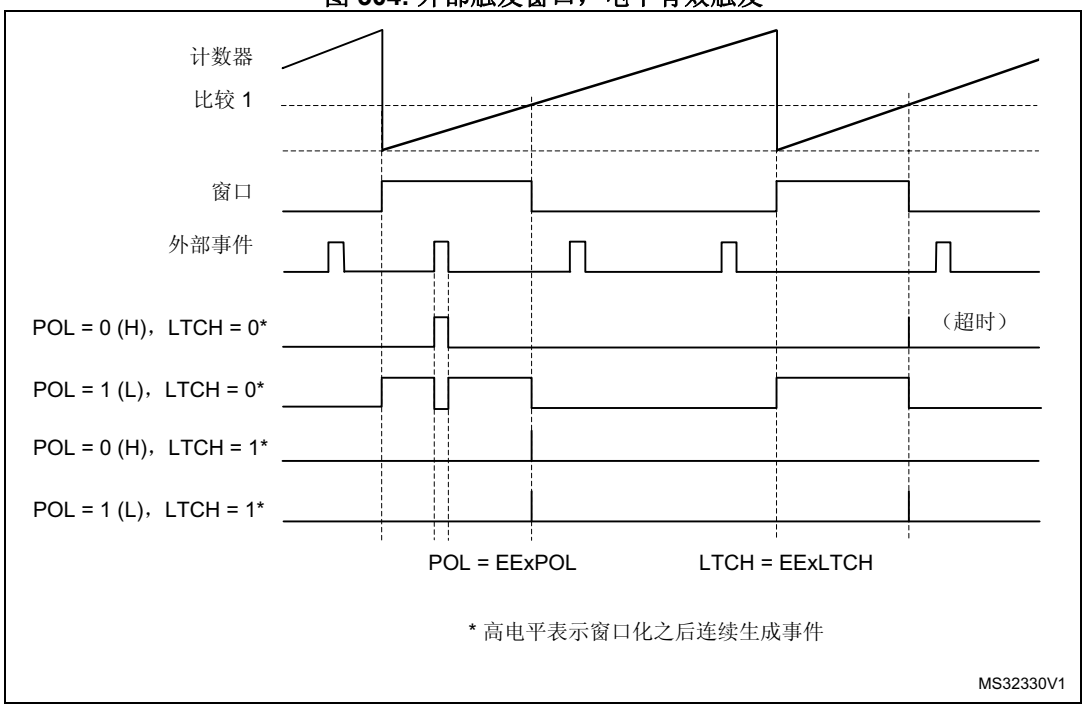


图 304. 外部触发窗口，电平有效触发



37.3.9 延迟保护

如果需要在有效脉冲结束后或推挽周期结束后以延迟的方式关断 PWM 输出，HRTIM 通常会为谐振转换器提供特定的保护机制。这些功能通过 HRTIM_OUTxR 寄存器中的 DLYPRTEN 位使能，并将使用特定的外部事件通道。

延迟空闲模式

在该模式下，有效电平会在保护激活之前结束。所选外部事件会使输出在有效脉冲结束时进入空闲模式（由 HRTIM_RSTx1R 或 HRTIM_RSTx2R 中的输出复位事件定义）。

一旦保护被触发，会永久保持空闲模式，但计数器会继续工作，直至输出重新使能。Tx1OEN 和 Tx2OEN 位不受延迟空闲模式进入的影响。要退出延迟空闲模式并恢复操作，需要将 Tx1OEN 和 Tx2OEN 位重写为 1。输出状态将在发出输出使能命令后第一次跳变为有效状态时更改。

注： *在有效脉冲结束之前，进入了延迟空闲模式便不能立即退出：务必先确保输出处于空闲状态，然后再重新开始运行模式。具体做法可以是等到下一周期，或者轮询 TIMxISR 寄存器中的 O1CPY 和/或 O2CPY 状态位。*

延迟空闲模式可应用于单路输出（DLYPRT[2:0] = x00 或 x01），也可应用于两路输出 (DLYPRT[2:0] = x10)。

进入响应延迟空闲模式后，可通过生成中断或 DMA 请求以作出响应。外部事件到达后，会立即将 HRTIM_TIMxISR 中的 DLYPRT 标志置 1，和输出上的有效脉冲是否结束无关。

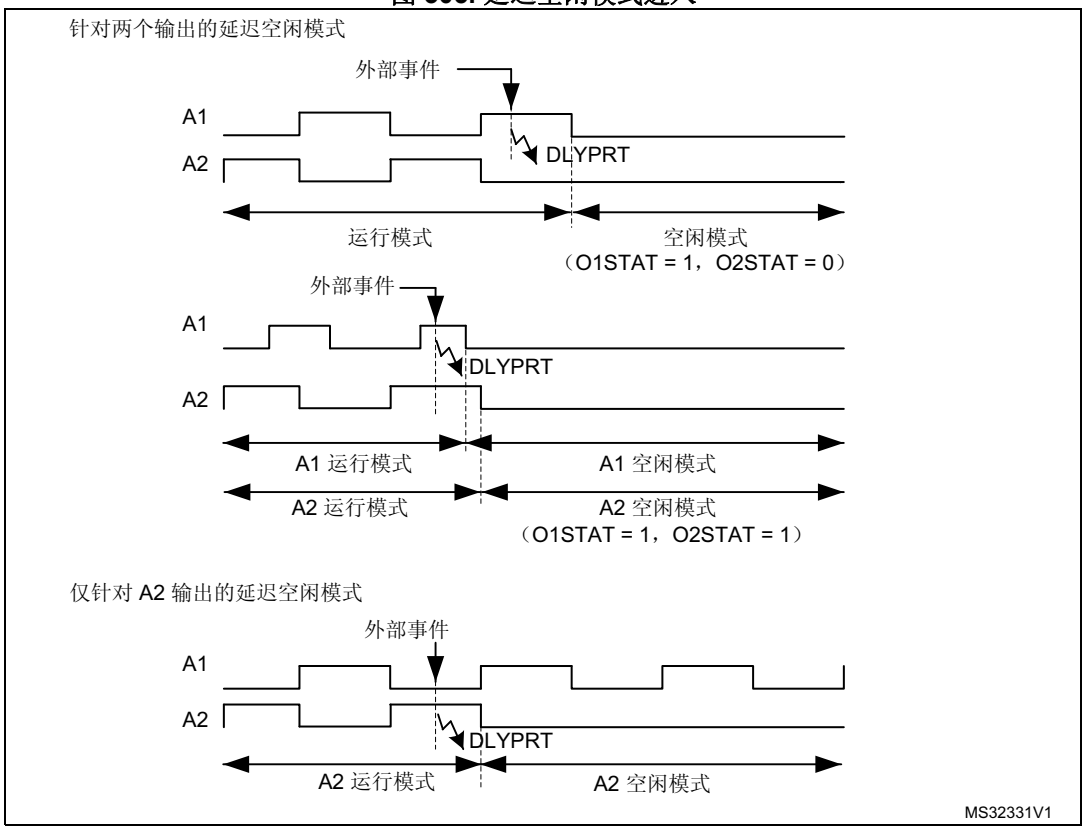
延迟空闲模式触发后，可通过 HRTIM_TIMxISR 中的 O1STAT 和 O2STAT 确定输出状态。即使延迟空闲模式应用于单路输出，也会更新这两个状态位。如果使能了推挽模式，HRTIM_TIMxISR 中的 IPPSTAT 标志会指示延迟保护请求在哪一周期发生。

无论定时器采用何种工作模式（常规、推挽、死区），均可使用此模式。此模式仅支持 2 个外部事件：

- hrtim_evt6 和 hrtim_evt7（对于定时器 A、B 和 C）
- hrtim_evt8 和 hrtim_evt9（对于定时器 D 和 E）

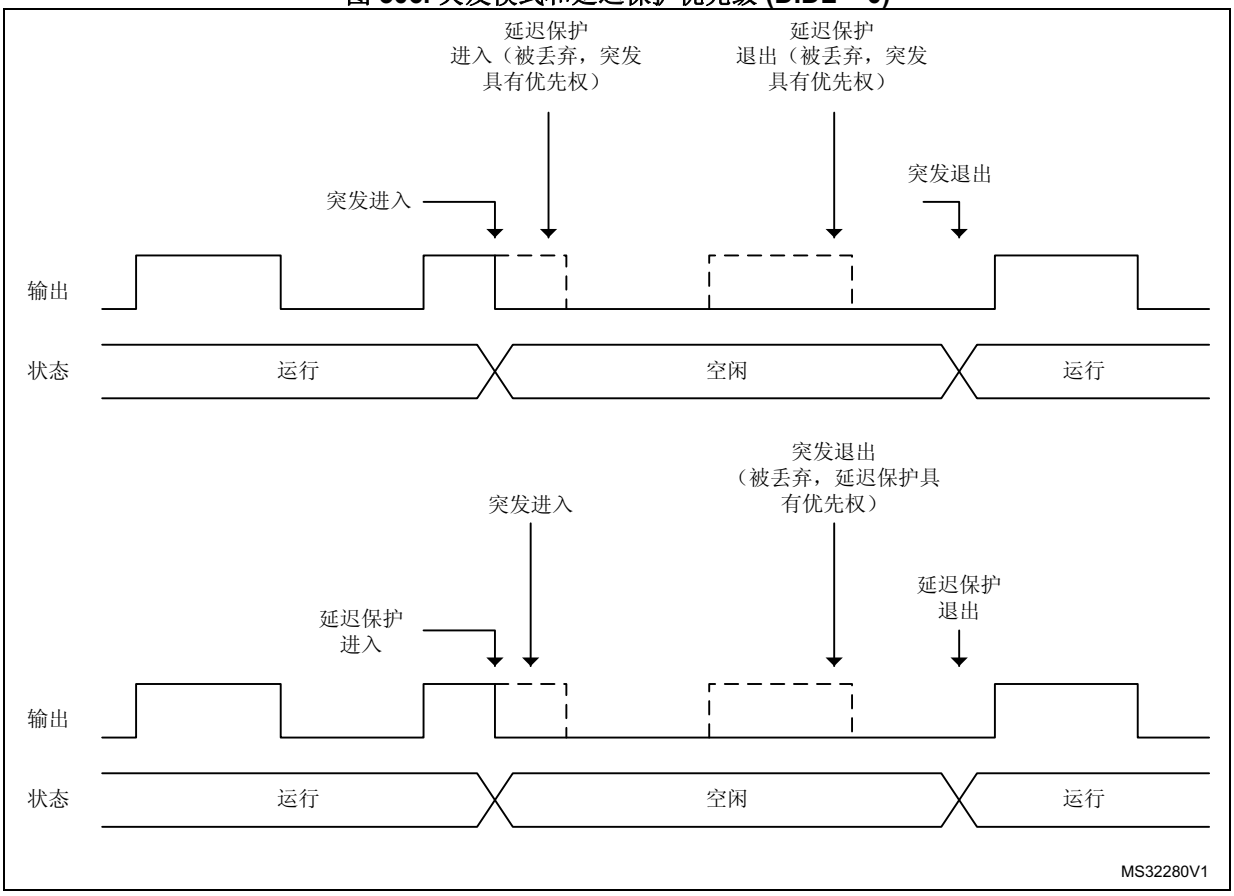
仅当计数器使能后（TxCEN 位置 1），才能触发延迟保护模式。即使 TxEN 位已复位，在 TxyOEN 位置 1 之前，延迟保护模式仍保持有效状态。

图 305. 延迟空闲模式进入



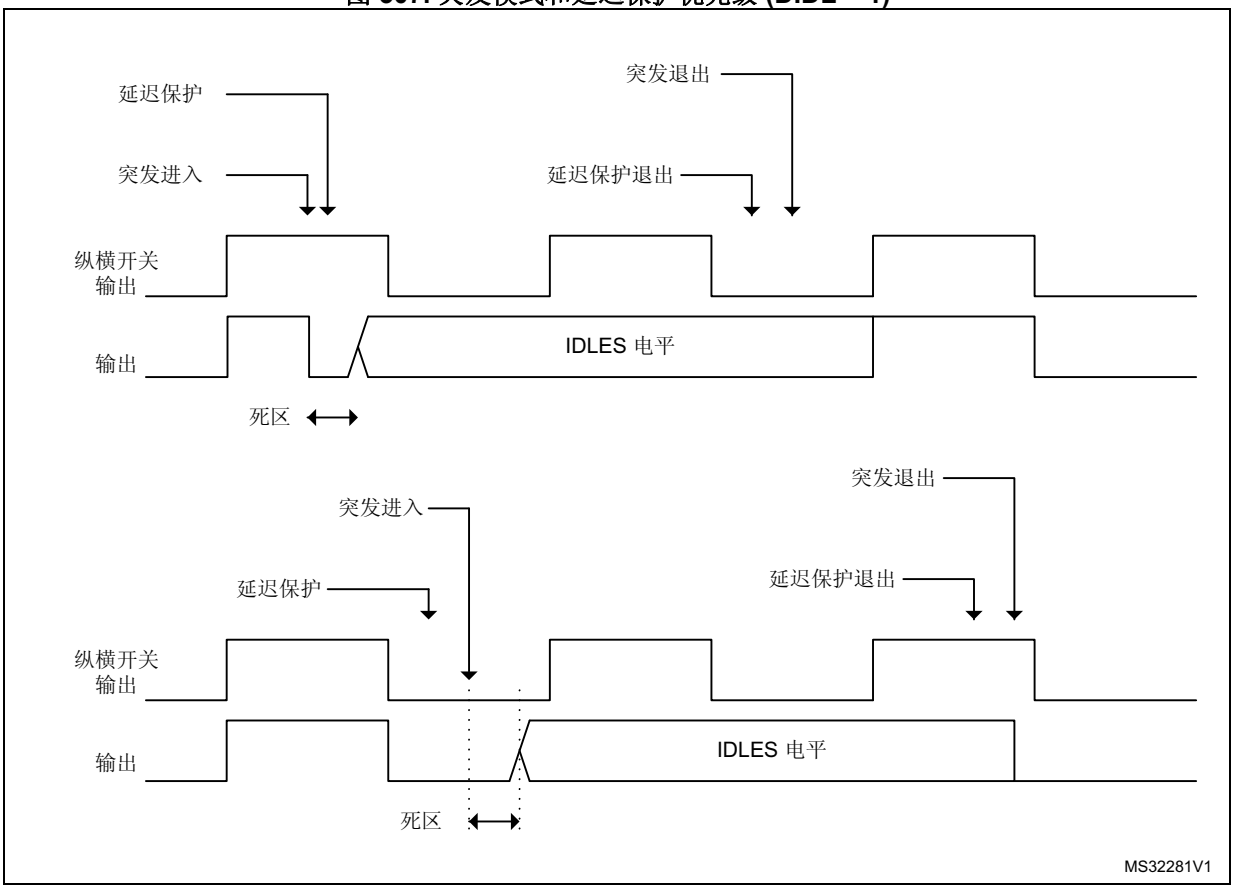
延迟空闲模式的优先级要高于突发模式：一旦触发了延迟空闲保护，就会丢弃任何突发模式退出请求。相反，如果在突发模式有效时退出延迟保护，突发模式将正常恢复，输出将保持为空闲状态，直至退出突发模式。*图 306* 给出了这些不同情况的概览。

图 306. 突发模式和延迟保护优先级 (DIDL = 0)



如果使能了延迟突发模式进入（DIDL 位置 1），则会应用相同的优先级，如下文的*图 307*所示。

图 307. 突发模式和延迟保护优先级 (DIDL = 1)



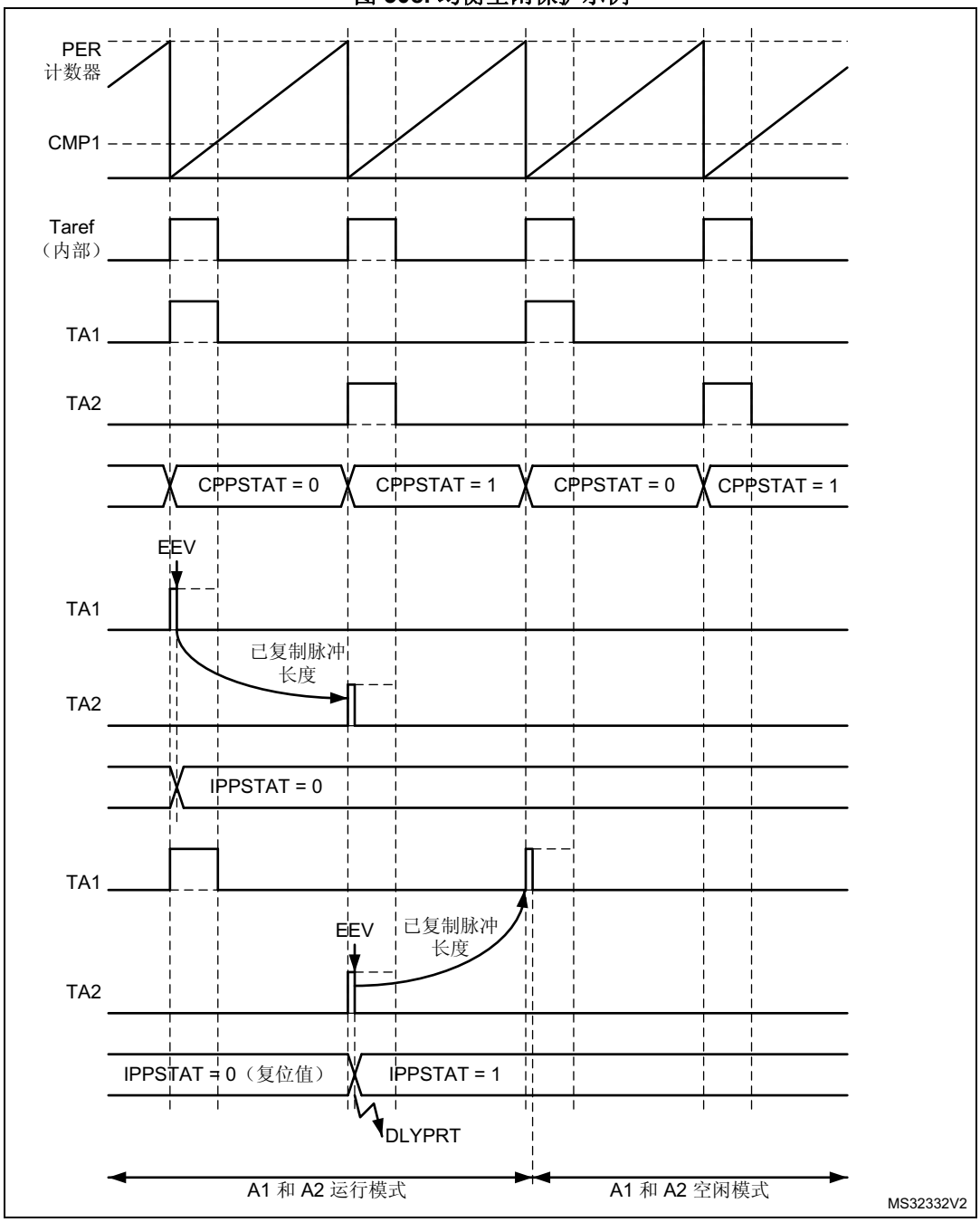
均衡空闲

仅在推挽模式下可用，在有效脉冲因保护方面的原因而缩短的情况下，可在两路输出上实现均衡脉宽。终止时间早于设定时间的脉宽会复制到另一路输出上，两路输出随后会进入空闲状态，直至通过软件恢复正常工作。此模式通过向 HRTIM_OUTxR 中的 DLYPRT[2:0] 位域写入 x11 来使能。

此模式仅支持 2 个外部事件：

- hrtim_evt6 和 hrtim_evt7（对于定时器 A、B 和 C）
- hrtim_evt8 和 hrtim_evt9（对于定时器 D 和 E）

图 308. 均衡空闲保护示例



如果均衡空闲模式已使能，所选外部事件会触发将计数器值捕获到比较 4 活动寄存器（用户不可访问该值）。推挽模式会额外保持一个周期，以重复缩短的脉冲：保持常规输出置位事件的同时，会生成新的输出复位事件。

随后会进入空闲模式，输出会采用由 HRTIM_OUTxR 寄存器中的 IDLESx 位定义的电平。均衡空闲模式进入由 DLYPRT 标志指示，IPPSTAT 标志则指示外部事件在哪一周期发生，可用于确定缩短脉冲的顺序（先是 HRTIM_CHA1，然后是 HRTIM_CHA2，反之亦然）。

定时器操作不会中断（计数器继续运行）。

要使能均衡空闲模式，需要执行下列初始化操作：

- 定时器在连续模式下工作 (CONT = 1)
- 使能推挽模式
- HRTIM_CMP4xR 必须设为 0，并且内容要传输到活动寄存器（例如，强制进行软件更新）
- DELCMP4[1:0] 位域必须设为 00（禁止自动延迟模式）
- DLYPRT[2:0] = x11（使能延迟保护）

注： *均衡空闲工作期间，不得对 HRTIM_CMP4xR 寄存器执行写操作。保留 CMP4 事件，不得用于其他用途。*

在均衡空闲模式下，建议避免使用多个外部事件或基于软件的复位事件，后者会导致输出复位。如果同一周期内此类事件先于均衡空闲请求到达，则会导致输出脉冲不均衡（第 1 个脉冲长度由外部事件或软件复位定义，而第 2 个脉冲长度则由均衡空闲模式进入定义）。

均衡空闲模式下可用的最小脉宽为 4 个 f_{HRTIM} 时钟周期。

如果在计数器达到其最小值之前进行捕获，会先将当前脉冲最多延长为 4 个 f_{HRTIM} 时钟周期，然后再将其复制到次级输出。在任何情况下，脉宽始终都是均衡的。

Tx1OEN 和 Tx2OEN 位不受均衡空闲模式进入的影响。要退出均衡空闲模式并恢复操作，需要同时将 Tx1OEN 和 Tx2OEN 位重写为 1。输出状态将在输出使能后的第一次有效跳变时更改。

可采用与延迟空闲模式进入相似的方法恢复操作。例如，如果外部事件到达时输出 1 激活（延迟空闲模式在输出 2 脉冲后生效），可先为输出 1 启动重启序列。为此，需要轮询 HRTIM_TIMxISR 寄存器中的 CPPSTAT 位。使用上例（IPPSTAT 标志等于 0），操作将在 CPPSTAT 位为 0 零时恢复。

为了获得特定的重启序列，可轮序 CPPSTAT 了解哪一输出将先激活。这样一来，便可使用与空闲模式进入序列相同的序列进行重启：如果 EEV 在输出 1 激活期间到达，则将在输出 1 激活时 (CPPSTAT = 0) 发起重启序列。

注： *正在执行脉冲均衡序列时，不得禁止均衡空闲模式。需要等到 CMP4 标志置 1（指示序列已结束）才能复位 DLYPRTEN 位。*

仅当计数器使能后（TxCEN 位置 1），才能触发均衡空闲保护模式。即使 TxCEN 位已复位，在 TxyOEN 位置 1 之前，延迟保护模式仍保持有效状态。

下列情况下，均衡空闲模式可与突发模式一起使用：

- TxBM 位必须复位（突发期间保持计数器时钟，请参见*第 37.3.13 节*）。
- 输出处于突发空闲状态时，不得触发均衡空闲保护。

均衡空闲模式的优先级要高于突发模式：一旦触发了均衡空闲保护，就会丢弃任何突发模式退出请求。相反，如果在突发模式有效时退出延迟保护，突发模式将正常恢复。

注： *虽然输出状态在空闲模式下会冻结，但在延迟保护后的空闲时间段内，仍会在辅助输出上生成一些事件（请参见第 37.3.16 节）：*
- 输出置位/复位中断或 DMA 请求
- 基于输出信号的外部事件过滤
- 由置位/复位触发的捕获事件

37.3.10 寄存器预装载和更新管理

大部分 HRTIM 寄存器均已缓冲，可根据需要进行预装载。这样做通常可避免波形被与有效事件（置位/复位）不同步的寄存器更新更改。

使能预装载模式后，受访问的寄存器变为影子寄存器。收到软件发出的更新请求或与事件同步的更新请求后，影子寄存器的内容会传输到活动寄存器。

默认情况下，HRTIM_MCR 和 HRTIM_TIMxCR 寄存器中的 PREEN 位会复位，寄存器不会预装载：任何写操作均会直接更新活动寄存器。如果 PREEN 位在定时器运行且预装载已使能时复位，预装载寄存器的内容会直接传输到活动寄存器中。

每个定时单元和主定时器都有自己的 PREEN 位。如果 PRREN 已置 1，仅当发生更新事件时，预装载寄存器才会使能，其内容才会传输到活动寄存器。

如果需要使用预装载功能，可选择两种方法初始化定时器：

- 刚好在定时器初始化结束时使能 PREEN 位，以在定时器使能之前将预装载寄存器内容传输到活动寄存器（通过将 MCEN 和 TxCEN 位置 1）。
- 在初始化过程中的任何时间使能 PREEN 位，并恰好在启动之前强制进行软件更新。

表 292 列出了可预装载的寄存器，并总结了可用更新事件。

表 292. HRTIM 可预装载控制寄存器和关联的更新源

定时器	可预装载寄存器	预装载使能	更新源
主定时器	HRTIM_DIER HRTIM_MPER HRTIM_MREP HRTIM_MCMP1R HRTIM_MCMP2R HRTIM_MCMP3R HRTIM_MCMP4R	HRTIM_MCR 中的 PREEN 位	软件 重复事件 突发 DMA 事件 突发 DMA 事件后的重复事件
定时器 x x = A..E	HRTIM_TIMxDIER HRTIM_TIMxPER HRTIM_TIMxREP HRTIM_TIMxCMP1R HRTIM_TIMxCMP1CR HRTIM_TIMxCMP2R HRTIM_TIMxCMP3R HRTIM_TIMxCMP4R HRTIM_DTxR HRTIM_SETx1R HRTIM_RSTx1R HRTIM_SETx2R HRTIM_RSTx2R HRTIM_RSTxR	HRTIM_TIMxCR 中的 PREEN 位	软件 TIMx 重复事件 TIMx 复位事件 突发 DMA 事件 来自其他定时器（TIMy、主定时器）的更新事件 突发 DMA 事件后的更新事件 更新使能输入 1..3 更新使能输入 1..3 后的更新事件
HRTIM 通用寄存器	HRTIM_ADC1R HRTIM_ADC2R HRTIM_ADC3R HRTIM_ADC4R	TIMx 或主定时器更新，取决于 HRTIM_CR1 中的 ADxUSRC[2:0] 位（如果所选定时器中的 PREEN = 1）	

主定时器有 4 个更新选项：

1. 软件：将 1 写入 HRTIM_CR2 中的 MSWU 位会立即强制更新寄存器。在这种情况下，会取消任何待定的硬件更新请求。
2. 当主计数器翻转且主重复计数器等于 0 时，会进行更新。当 HRTIM_MCR 中的 MREPU 位置 1 时，会使能此操作。
3. 突发 DMA 结束后进行更新（有关详细信息，请参见*第 37.3.21 节*）。当 HRTIM_MCR 中的 BRSTDMA[1:0] = 01 时，会使能此操作。可以同时设置 MREPU=1 和 BRSTDMA=01。*注：如果 SWU 位置 1，可在突发序列结束后立即进行更新（即强制更新模式）。如果 SWU 位复位，将在突发序列结束后发生下一个更新事件时进行更新。*
4. 突发 DMA 结束后主计数器翻转时，会进行更新。当 HRTIM_MCR 中的 BRSTDMA[1:0] = 10 时，会使能此操作。

中断或 DMA 请求可通过主定时器更新事件生成。

各个定时器 (TIMA..E) 还可以通过以下方式进行更新：

- 软件：将 1 写入 HRTIM_CR2 中的 TxSWU 位会立即强制更新寄存器。在这种情况下，会取消任何挂起的硬件更新请求。
- 当计数器翻转且重复计数器等于 0 时，会进行更新。当 HRTIM_TIMxCR 中的 TxREPU 位置 1 时，会使能此操作。
- 当计数器复位或在连续模式下翻转时，会进行更新。当 HRTIM_TIMxCR 中的 TxRSTU 位置 1 时，会使能此操作。此操作用于在单发模式下工作的定时器等。
- 突发 DMA 结束后立即进行更新。当 HRTIM_TIMxCR 中的 UPDGAT[3:0] = 0001 时，会使能此操作。
- 在突发 DMA 结束后发生更新事件时会进行更新（更新事件可通过 TxREPU、MSTU 或 TxU 使能）。当 HRTIM_TIMxCR 中的 UPDGAT[3:0] = 0010 时，会使能此操作。
- 更新使能输入 1..3 上接收到请求时，会进行更新。当 HRTIM_TIMxCR 中的 UPDGAT[3:0] = 0011、0100、0101 时，会使能此操作。
- 在更新使能输入 1..3 上接收到请求后发生更新事件时会进行更新（更新事件可通过 TxREPU、MSTU 或 TxU 使能）。当 HRTIM_TIMxCR 中的 UPDGAT[3:0] = 0110、0111、1000 时，会使能此操作。
- 与其他任何定时器或主定时器更新同步进行更新（例如，TIMA 可与 TIMB 同步更新）。此操作通过将 HRTIM_TIMxCR 寄存器中的 MSTU 和 TxU 位置 1 来使能，用于需要使用多个定时器的转换器。

更新使能输入 1..3 可使更新事件与来自通用定时器的片上事件同步。这些输入为上升沿有效。

表 293 列出了更新使能输入与片上源之间的连接。

表 293. 更新使能输入和源

更新使能输入	更新源
更新使能输入 1	TIM16_OC
更新使能输入 2	TIM17_OC
更新使能输入 3	TIM6_TRGO

这样可将低频更新请求与高频信号同步（例如，在不得不用 100Hz 的速率更新 100KHz 的翻转计数器时）。

注： *CKPSC[2:0] > 5 时，更新事件会同步到预分频器时钟。*

中断或 DMA 请求可通过 Timx 更新事件生成。

无论选择哪一种更新事件，HRTIM_CR1 寄存器中的 MUDIS 和 TxUDIS 位都可以暂时禁止将预装载寄存器的内容传输至活动寄存器。这样便可修改并联的定时器中的多个寄存器。这些位复位后，会立即执行常规更新事件。

MUDIS 和 TxUDIS 位全部分组到同一寄存器中。这样便可同时禁止和恢复更新并联工作的定时器（不需要同步）。

下例为实际用例。第一个电源转换器通过主定时器、TIMB 和 TIMC 控制。TIMB 和 TIMC 必须同时通过主定时器重复事件更新。第二个转换器与 TIMA、TIMD 和 TIME 并行运行，TIMD、TIME 必须通过 TIMA 重复事件更新。

第一个转换器

在 HRTIM_MCR 中，MREPU 位已置 1：将在主定时器计数器重复周期结束时进行更新。在 HRTIM_TIMBCR 和 HRTIM_TIMCCR 中，MSTU 位必须置 1 才能通过主定时器同时更新 TIMB 和 TIMC 定时器。

如果必须通过软件调整电源转换器的设定值，则必须在对寄存器执行写访问更新数值（例如比较值）之前将 HRTIM_CR 寄存器的 MUDIS、TBUDIS 和 TCUDIS 位置 1。从此刻起，会忽略任何硬件更新请求，并可无风险地访问预装载寄存器，将其中的内容传输到活动寄存器中。软件处理结束后，必须将 MUDIS、TBUDIS 和 TCUDIS 位复位。发生主定时器重复事件后，会立即将预装载寄存器的内容传输到活动寄存器。

第二个转换器

在 HRTIM_TIMACR 中，TAREPU 位已置 1：将在定时器 A 计数器重复周期结束时进行更新。在 HRTIM_TIMDCR 和 HRTIM_TIMECR 中，TAU 位必须置 1 才能通过定时器 A 同时更新 TIMD 和 TIME 定时器。

如果必须通过软件调整电源转换器的设定值，则必须在对寄存器执行写访问更新数值（例如比较值）之前将 HRTIM_CR 寄存器的 TAUDIS、TDUDIS 和 TEUDIS 位置 1。从此刻起，会忽略任何硬件更新请求，并可无风险地访问预装载寄存器，将其中的内容传输到活动寄存器中。软件处理结束后，可将 TAUDIS、TDUDIS 和 TEUDIS 位复位：发生定时器 A 重复事件后，会立即将预装载寄存器的内容传输到活动寄存器。

37.3.11 事件在多个定时器之间的传播

HRTIM 提供多种方式在多个定时单元（包括主定时器）之间逐级传输事件或共享事件，以充分利用其模块化架构的优势。这些是需要使用多路同步输出的转换器的主要特性。

本节总结了各种选项，并详细说明了事件是否会在 HRTIM 中传播以及怎样传播。

由主定时器更新触发的 TIMx 更新

表 294 中列出的源将生成主定时器更新。该表指出了是否可使用源事件在任何 TIMx 定时单元中触发同步更新。

工作条件：HRTIM_TIMxCR 寄存器中的 MSTU 位置 1。

表294. 主定时器更新事件传播

源	条件	传播	注释
突发 DMA 结束	BRSTDMA[1:0] = 01	无	必须在 TIMxCR 中进行 (UPDGAT[3:0] = 0001)
突发 DMA 结束后的翻转事件	BRSTDMA[1:0] = 10	有	-
计数器翻转引发的重复事件	MREPU = 1	有	-
计数器复位（通过 HRTIM_SCIN 或软件）引发的重复事件		无	-
软件更新	MSWU = 1	无	所有软件更新位 (TxSWU) 均分组到 HRTIM_CR2 寄存器中，可用于同步更新

由 TIMy 更新触发的 TIMx 更新

表 295 中列出的源将生成 TIMy 更新。该表指出了是否可使用给定事件在另一定时器或多个 TIMx 定时器中触发同步更新。

工作条件：HRTIM_TIMxCR 寄存器中的 TyU 位置 1（源 = TIMy，目标 = TIMx）。

表295. TIMx 更新事件传播

源	条件	传播	注释
突发 DMA 结束	UPDGAT[3:0] = 0001	无	必须直接在 HRTIM_TIMxCR 中进行 (UPDGAT[3:0] = 0001)
更新使能输入引发的更新	UPDGAT[3:0] = 0011、0100、0101	无	必须直接在 HRTIM_TIMxCR 中进行（UPDGAT[3:0] = 0011、0100、0101）
主定时器更新	HRTIM_TIMyCR 中的 MSTU = 1	无	必须在 HRTIM_TIMxCR 中的 MSTU = 1 时进行
另一 TIMx 更新 (TIMz>TIMy>TIMx)	HRTIM_TIMyCR 中的 TzU = 1 TIMxCR 中的 TyU = 1	无	必须在 HRTIM_TIMxCR 中的 TzU = 1 时进行 HRTIM_TIMyCR 中的 TzU = 1
计数器翻转引发的重复事件	TyREPU = 1	有	-
计数器复位引发的重复事件	TyREPU = 1	-	请参见下文中的计数器复位情况
计数器翻转	TyRSTU = 1	有	-
计数器软件复位	HRTIM_CR2 中的 TyRST = 1	无	可与 HRTIM_CR2 寄存器中的更新同时进行
TIMz 比较引发的计数器复位	HRTIM_RSTyR 中的 TIMzCMPn	无	必须通过 HRTIM_RSTxR 中的 TIMzCMPn 进行
外部事件引发的计数器复位	HRTIM_RSTyR 中的 EXTEVNTn	有	-
主定时器比较或主定时器周期引发的计数器复位	HRTIM_RSTyR 中的 MSTCMPn 或 MSTPER	无	-

表295. TIMx 更新事件传播（续）

源	条件	传播	注释
TIMy 比较引发的计数器复位	HRTIM_RSTyR 中的 CMPn	有	-
更新引发的计数器复位	HRTIM_RSTyR 中的 UPDT	无	传播会导致进入锁定状态（更新会引发复位，进而引发更新）
HRTIM_SCIN 引发的计数器复位	HRTIM_TIMyCR 中的 SYNCRSTy	无	-
软件更新	TySWU = 1	无	所有软件更新位 (TxSWU) 均分组到 HRTIM_CR2 寄存器中，可用于同步更新

引发 TIMx 更新的 TIMx 计数器复位

表 296 列出了计数器复位源，并指出了这些源是否可用于生成更新事件。

工作条件：HRTIM_TIMxCR 寄存器中的 TxRSTU 位。

表296. 能够生成更新事件的复位事件

源	条件	传播	注释
计数器翻转		有	
更新事件	HRTIM_RSTxR 中的 UPDT	无	传播会导致进入锁定状态（更新会引发复位，进而引发更新）
外部事件	HRTIM_RSTxR 中的 EXTEVNTn	有	-
TIMy 比较	HRTIM_RSTxR 中的 TIMyCMPn	有	-
主定时器比较	HRTIM_RSTxR 中的 MSTCMPn	有	-
主定时器周期	HRTIM_RSTxR 中的 MSTPER	有	-
比较 2 和 4	HRTIM_RSTxR 中的 CMPn	有	-
软件	HRTIM_CR2 中的 TxRST = 1	有	-
HRTIM_SCIN	HRTIM_TIMxCR 中的 SYNCRSTx	有	-

引发 TIMx 计数器复位的 TIMx 更新

表 297 列出了更新事件源，并指出了这些源是否可用于生成计数器复位事件。

工作条件：HRTIM_RSTxR 寄存器中的 UPDT 位置 1。

表297. 用于定时器复位的更新事件传播

源	条件	传播	注释
突发 DMA 结束	UPDGAT[3:0] = 0001	有	-
更新使能输入引发的更新	UPDGAT[3:0] = 0011、0100、0101	有	-
突发 DMA 后的翻转引发的主定时器更新	HRTIM_TIMxCR 中的 MSTU = 1 HRTIM_MCR 中的 BRSTDMA[1:0] = 10	有	-
翻转后的重复事件引发的主定时器更新	HRTIM_TIMxCR 中的 MSTU = 1 HRTIM_MCR 中的 MREPU = 1	有	-
计数器复位（通过软件或 HRTIM_SCIN 复位）后的重复事件引发的主定时器更新		无	-
软件触发的主定时器更新	HRTIM_TIMxCR 中的 MSTU = 1 HRTIM_CR2 中的 MSWU = 1	无	所有软件更新位 (TxSWU) 均分组到 HRTIM_CR2 寄存器中，可用于同步更新
TIMy 计数器翻转引发的 TIMy 更新	HRTIM_TIMxCR 中的 TyU = 1 HRTIM_TIMyCR 中的 TyRSTU = 1	有	-
TIMy 重复事件引发的 TIMy 更新	HRTIM_TIMxCR 中的 TyU = 1 HRTIM_TIMyCR 中的 TyREPU = 1	有	-
外部事件或 TIMy 比较（通过 TIMy 复位）引发的 TIMy 更新	HRTIM_TIMxCR 中的 TyU = 1 HRTIM_TIMyCR 中的 TyRSTU = 1 HRTIM_RSTyCR 中的 EXTEVNTn 或 CMP4/2	有	-
上面列出的源以外的源引发的 TIMy 更新	HRTIM_TIMxCR 中的 TyU = 1	无	-
翻转后的重复事件	HRTIM_TIMxCR 中的 TxREPU = 1	有	-
计数器复位后的重复事件		无	-
定时器复位	HRTIM_TIMxCR 中的 TxRSTU = 1	无	传播会导致进入锁定状态（复位会引发更新，进而引发复位）
软件	HRTIM_CR2 中的 TxSWU	无	-

37.3.12 输出管理

每个定时单元都控制着一对输出。输出有三种工作状态：

- RUN：此状态为主要工作状态，在该状态下，输出会获取在纵横开关单元中设定的有效电平或无效电平。
- IDLE：当输出通过软件禁用或处于突发模式工作期间（此时，正常工作模式下会暂时禁止输出，更多详细信息，请参见*第 37.3.13 节*），该状态是 HRTIM 复位后的默认工作状态。该状态永久有效或无效。
- FAULT：此状态为安全状态，FAULTx 输入上存在关断请求时会进入此状态。该状态可以是永久有效、无效或高阻态 (Hi-Z)。

输出状态由 HRTIM_OENR 寄存器中的 TxyOEN 位以及 HRTIM_ODSR 寄存器中的 TxyODS 位指示，如*表 298* 所示。

表 298. 输出状态编程，x= A..E，y = 1 或 2

TxyOEN（控制/状态）（通过软件置 1，通过硬件清零）	TxyODS（状态）	输出工作状态
1	x	RUN
0	0	IDLE
0	1	FAULT

TxyOEN 位既是控制位，也是状态位：该位必须通过软件置 1，才能使输出处于 RUN 模式。输出恢复 IDLE 或 FAULT 模式时，该位会通过硬件清零。TxyOEN 位清零后，TxyODS 位会指示输出处于 IDLE 状态还是 FAULT 状态。HRTIM_ODISR 寄存器中的第三位可通过软件禁止输出。

图 309. 输出管理概览

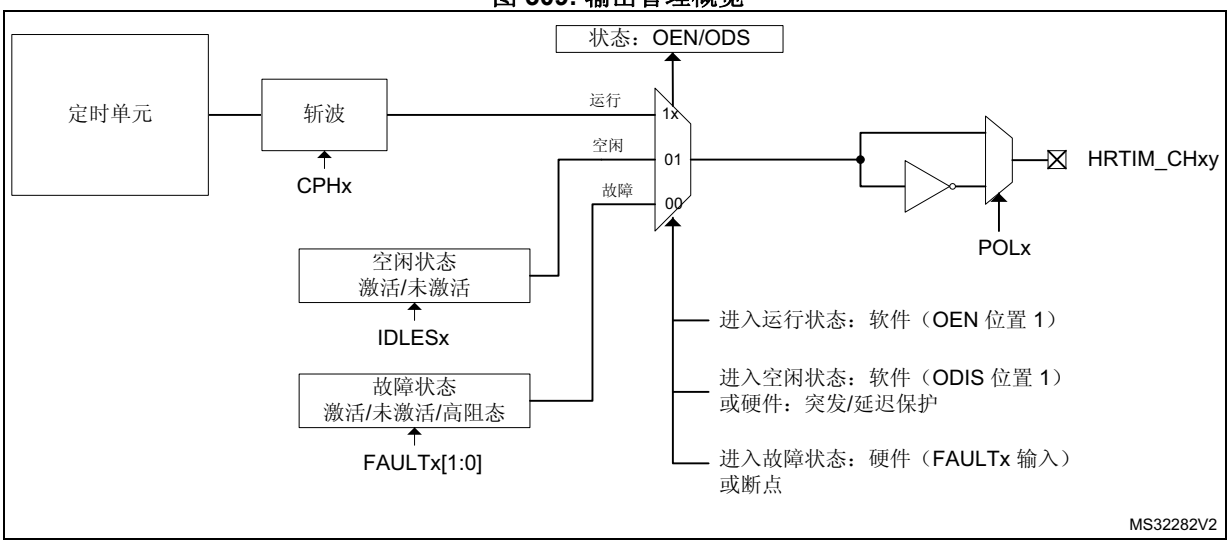
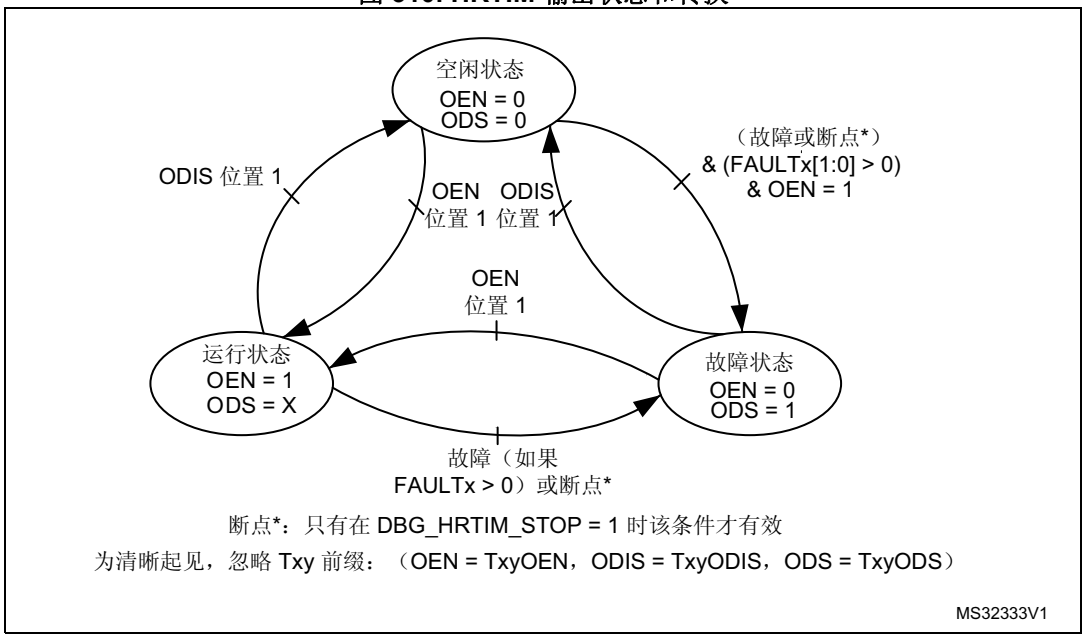


图 310 总结了三种状态对应的位值以及转换的触发方式。任何外部或内部故障源均可触发故障（请参见*第 37.3.15 节*），当突发模式或延迟保护有效时，可进入空闲状态。

图 310. HRTIM 输出状态和转换



FAULT 和 IDLE 电平定义为有效或无效。有效（或无效）是指定时器输出上可引发电源开关闭合（对于无效状态，则为电源开关断开）的电平。

IDLE 状态的优先级最高：即使 FAULT 条件仍有效，也可以实现由 ODIS 位置 1 触发的 FAULT → IDLE 转换。

FAULT 状态的优先级要高于 RUN 状态：如果 TxyOEN 位置 1 的同时发生了故障事件，则将进入 FAULT 状态。IDLE → FAULT 转换中给出了相应的条件（如*图 310* 所示）：需要使能故障保护（FAULTx[1:0] 位 = 01、10、11），且 Txy OEN 位在故障有效时置 1（如果 DBG_HRTIM_STOP = 1，则在断点期间置 1）。

输出极性通过 HRTIM_OUTxR 中的 POLx 位编程。如果 POLx = 0，极性为正（输出高电平有效）；如果 POLx = 1，极性为负（输出低电平有效）。实际上，极性的定义取决于要驱动的电源开关（PMOS 与 NMOS）或栅极驱动器极性。

每路输出 FAULT 状态下的输出电平通过 HRTIM_OUTxR 中的 FAULTx[1:0] 位配置，具体如下：

- 00：输出永不进入故障状态，停留在 RUN 或 IDLE 状态
- 01：处于 FAULT 状态时，输出为有效电平
- 10：处于 FAULT 状态时，输出为无效电平
- 11：处于 FAULT 状态时，输出为三态。必须通过诸如上拉或下拉电阻在外部强制进入安全状态。

注： *只要输出处于 FAULT 状态，便不得更改 FAULTx[1:0] 位。*

IDLE 状态下的输出电平通过 HRTIM_OUTxR 中的 IDLESx 位配置，具体如下：

- 0：处于 IDLE 状态时，输出为无效电平
- 1：处于 IDLE 状态时，输出为有效电平

当 TxyOEN 位置 1 进入 RUN 状态时，输出会立即连接到纵横开关输出。如果定时器时钟停止，电平将为无效（HRTIM 复位后）或相当于 RUN 电平（定时器停止、且输出禁止时）。

在 HRTIM 初始化过程中，在使输出进入 RUN 模式之前，可在 HRTIM_SETx1R 和 HRTIM_RSTx1R 寄存器中使用软件强制输出置位和复位预先设定输出电平。

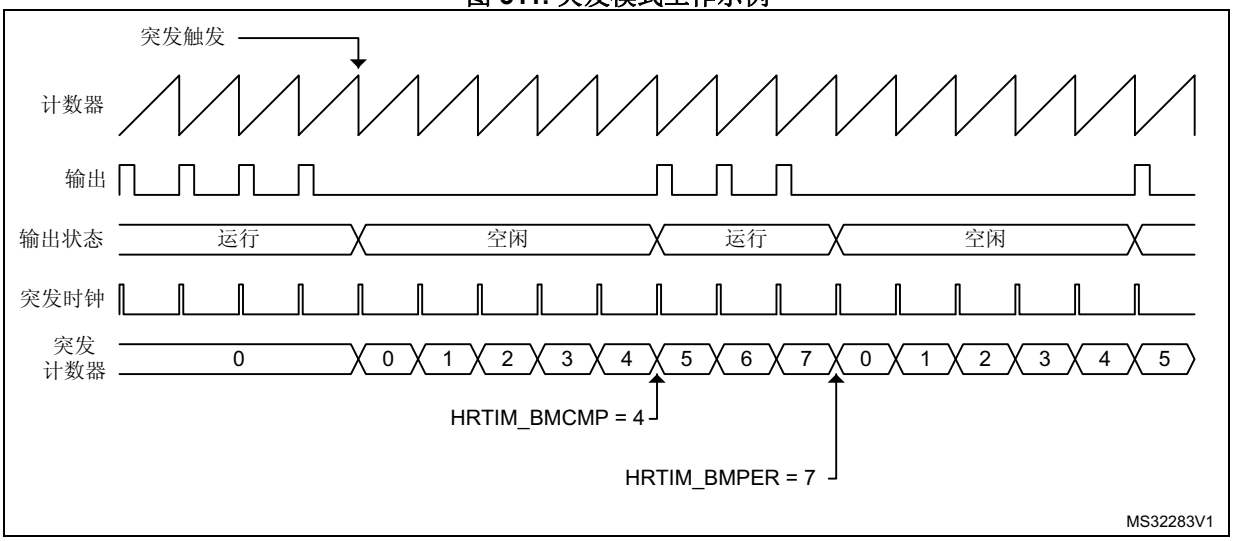
37.3.13 突发模式控制器

突发模式控制器可通过硬件使输出交替地进入 IDLE 状态和 RUN 状态，以便通过可编程周期性和占空比跳过一些开关周期。

轻载运行时，突发模式工作常用于电源转换器中。该模式减少了输出跳变次数以及关联的开关损耗，从而显著提高转换器效率。

在突发模式下运行时，会在空闲周期（等于几个计数周期）后输出一个或几个脉冲，空闲周期内通常不会生成任何输出脉冲，如*图 311* 中的示例所示。

图 311. 突发模式工作示例



突发模式控制器包括：

- 可由各种源提供时钟的计数器（HRTIM 之内或之外，通常是 PWM 周期结束时）。
- 定义空闲周期数的比较寄存器：HRTIM_BMCMP。
- 定义突发重复率的周期寄存器（相当于空闲周期与运行周期之和）：HRTIM_BMPER。

突发模式控制器能够接管对 10 路 PWM 输出中的任何一路的控制。突发模式工作期间，每路输出的状态通过 HRTIM_OUTxR 寄存器中的 IDLESx 和 IDLEMx 位编程，如*表 299* 所述。

表299. 突发模式的定时器输出编程

IDLEMx	IDLESx	突发模式期间的输出状态
0	X	无操作：输出不受突发模式工作的影响
1	0	突发模式期间输出无效
1	1	突发模式期间输出有效

注： *突发模式有效时，不得更改 IDLEMx 位。*

突发模式控制器仅会作用于输出级。空闲周期内仍会生成一些事件：

- 输出置位/复位中断或 DMA 请求
- 基于 Tx2 输出信号的外部事件过滤
- 由输出置位/复位触发的捕获事件

突发模式期间，即使 TxBM 位已置 1，HRTIM_SCOUT 输出上也不会生成启动事件或复位事件。

工作模式

在给定的定时单元上使用突发模式之前，必须使能计数器（TxCEN 位置 1）。突发模式通过 HRTIM_BMCR 寄存器中的 BME 位使能。

通过 HRTIM_BMCR 寄存器中的 BMOM 位，定时单元可在连续模式或单发模式下工作。BMOM = 1 时，使能连续模式。在 HRTIM_BMCR 中的 BMSTAT 位复位终止突发工作之前，都会保持突发工作状态。

在单发模式下 (BMOM = 0)，会执行一次空闲序列，随后会触发突发模式，之后会立即恢复正常定时器操作。

空闲周期和运行周期的持续时间通过突发模式计数器和 2 个寄存器定义。HRTIM_BMCMPR 寄存器以计数定义所选定时器处于空闲状态的持续时间（空闲周期）。HRTIM_BMPER 定义总突发模式周期（空闲周期与运行周期之和）。触发初始突发模式后，空闲周期长度为 HRTIM_BMCMPR+1，总突发周期为 HRTIM_BMPER+1。

注： *突发模式周期不得小于或等于通过 DTRx[8:0] 和 DTFx[8:0] 位域定义的死区持续时间。*

突发模式工作期间，定时单元和主定时器的计数器可停止和复位。HRTIM_BMCR 保留了 6 个控制位来实现此目的：MTBM（主定时器）和 TABM..TEBM（用于定时器 A..E）。

当 MTBM 或 TxBM 位复位时，会保留计数器时钟。举例来说，这样可在多相位系统中保持与其他定时器的相位关系。

如果 MTBM 或 TxBM 位已置 1，则在突发空闲周期期间，相应的计数器会停止，并会保持在复位状态下。这样便可使定时器在退出空闲状态时重新开始一个完整的周期。如果 SYNCSRC[1:0] = 00 或 10（主定时器 启动或定时器 A 启动时的同步输出），退出突发模式时，会在 HRTIM_SCOUT 输出上发送一个脉冲。

注： *当均衡空闲模式激活时 (DLYPRT[1:0] = 0x11)，不得将 TxBM 位置 1。*

突发模式时钟

突发模式控制器计数器可由多个时钟源提供时钟，具体通过 HRTIM_BMCR 寄存器中的 BMCLK[3:0] 位来选择：

- BMCLK[3:0] = 0000 到 0101：主定时器和 TIMA..E 复位/翻转事件。这样可使突发模式空闲周期和运行周期与定时单元计数周期对齐（均处于自由运行和计数器复位模式）。
- BMCLK[3:0] = 0110 到 1001：时钟由通用定时器提供，如*表 300* 所示。在这种情况下，突发模式空闲周期和运行周期不必与定时单元计数周期对齐（输出上的脉冲可能中断，从而导致波形的占空比被修改）。
- BMCLK[3:0] = 1010：f_{HRTIM} 时钟由通过 HRTIM_BMCR 寄存器中的 BMPRSC[3:0] 位定义的系数进行预分频。在这种情况下，突发模式空闲周期和运行周期不必与定时单元计数周期对齐（输出上的脉冲可能中断，从而导致波形的占空比被修改）。

表 300. 来自通用定时器的突发模式时钟源

BMCLK[3:0]	时钟源
0110	hrtim_bm_ck1：TIM16 OC
0111	hrtim_bm_ck2：TIM17 OC
1000	hrtim_bm_ck3：TIM7 TRGO
1001	hrtim_bm_ck4：保留

TIMxx OC 输出上的脉宽长度必须至少为 N 个 f_{HRTIM} 时钟周期才能被 HRTIM 突发模式控制器检测到。

突发模式触发

要触发突发模式工作，可使用 32 个源，这些源通过 HRTIM_BMTRGR 寄存器选择：

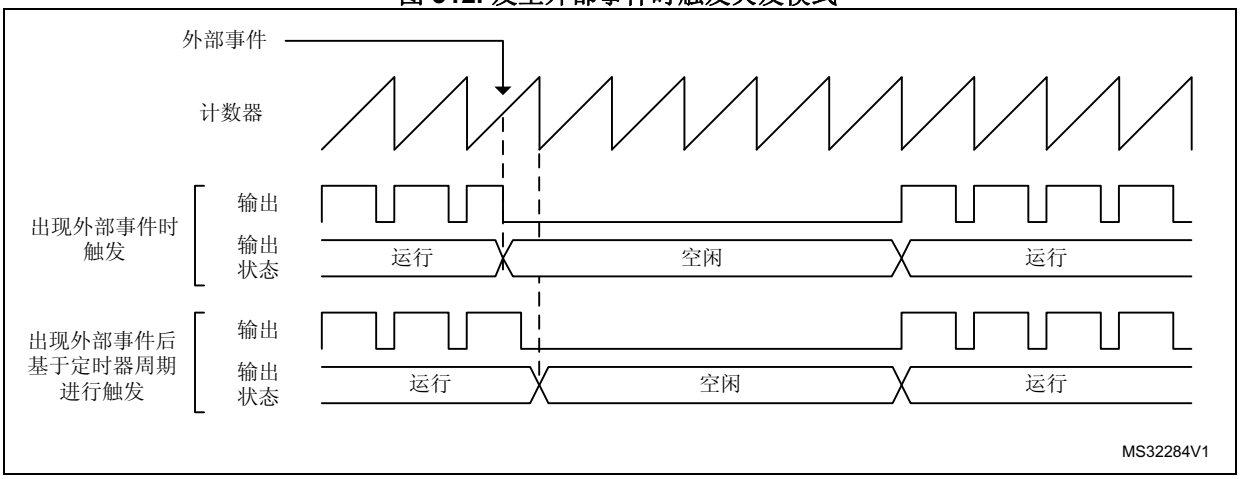
- 软件触发（通过软件置 1，通过硬件复位）
- 6 个主定时器事件：重复、复位/翻转、比较 1 到 4
- 来自定时器 A..E 的 5 x 4 个事件：重复、复位/翻转、比较 1 到 2
- hrtim_evt7（包括 TIMA 事件过滤）和 hrtim_evt8（包括 TIMD 事件过滤）
- hrtim_evt7 之后的定时器 A 周期（包括 TIMA 事件过滤）
- hrtim_evt8 之后的定时器 D 周期（包括 TIMD 事件过滤）
- 来自其他通用定时器的片上事件（hrtim_bm_trg 输出：TIM7_TRGO 输出）

这些源可结合起来使用，实现多源并发触发。

突发模式不可再次触发。在连续模式下，突发模式终止之前，会忽略新的触发；而在单发模式下，包含运行周期的当前突发结束之前（HRTIM_BMPER+1 个周期），会忽略触发。这一点同样适用于软件触发（即使软件位已丢弃，也会通过硬件复位）。

图 312 显示了突发模式是如何响应外部事件而启动的（立即启动，或在事件后的定时器周期启动）。

图 312. 发生外部事件时触发突发模式



对于 TAEEV7 和 TDEEV8 组合触发（在外部事件后的定时器周期触发），无论采用何种突发模式编程以及正在进行何种突发操作，外部事件检测始终有效：

- 如果在外部事件与定时器周期事件之间突发模式已使能 (BME=1) 或触发已使能（BMTRG 寄存器中的 TAEEV7 或 TDEEV8 位置 1），则会触发突发模式。
- 即使外部事件在突发结束之前发生，也会再次触发单发突发模式（只要突发之后发生相应的周期事件）。

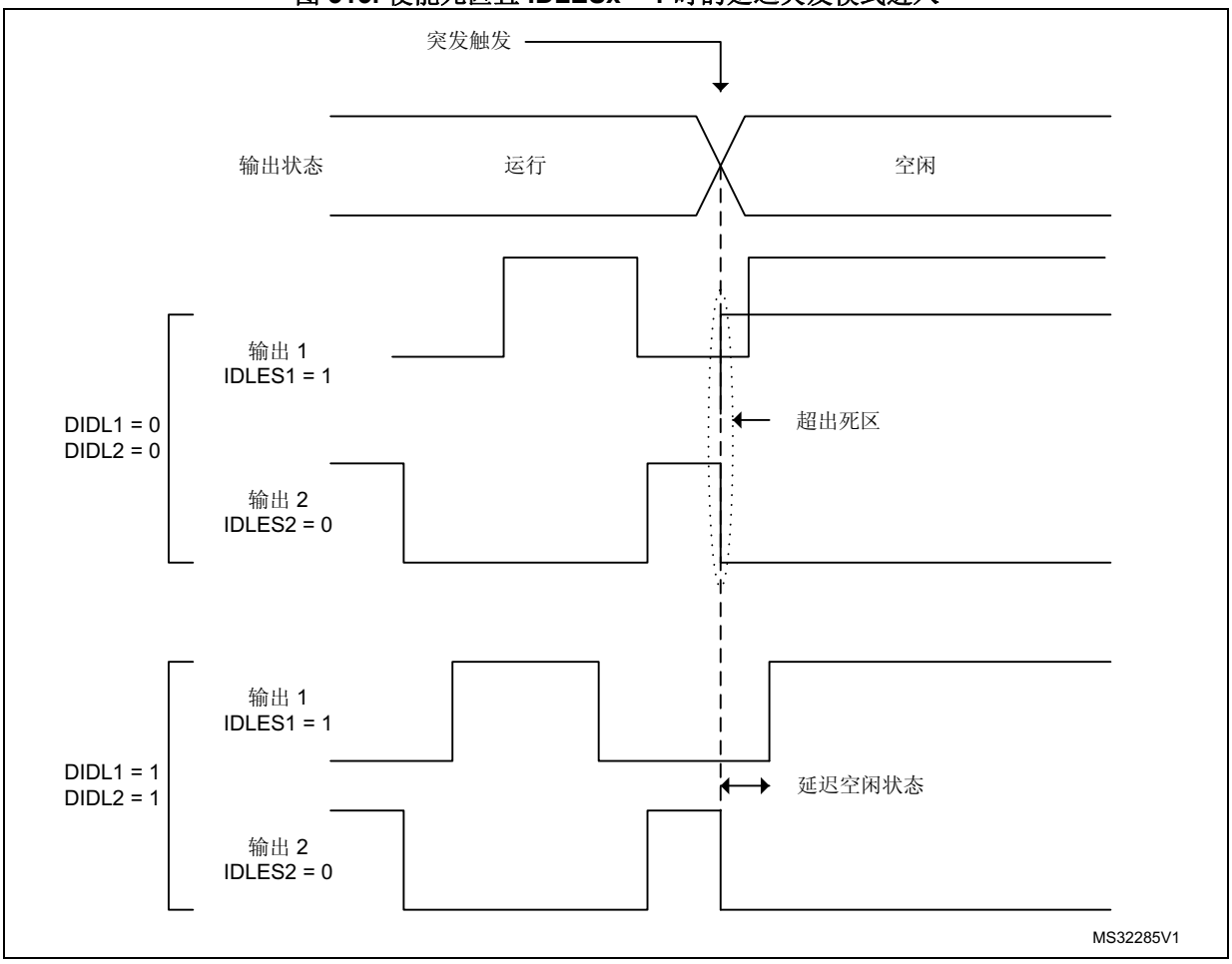
注： *TAEEV7 和 TDEEV8 触发仅在周期事件后有效。如果计数器在周期事件之前复位，则会丢弃挂起的 hrtim_evt7/8 事件。*

突发模式延迟进入

默认情况下，输出将在突发模式触发后立即获取其空闲电平（空闲电平取决于 IDLES1 和 IDLES2 设置）。

还可以延迟进入突发模式，并在输出获取其空闲状态之前的可编程周期内将输出强制为无效状态。驱动两路互补输出（其中一路输出具有有效空闲状态）时，可利用此特性避免超出死区，如*图 313* 所示。这样可避免半桥中出现击穿电流的风险，但会导致对突发模式进入的延迟响应。

图 313. 使能死区且 IDLESx = 1 时的延迟突发模式进入



延迟突发进入模式通过 HRTIM_OUTxR 寄存器中的 DIDLx 位使能（每路输出一个使能位）。该模式会在输出获取其空闲状态之前强制插入死区。每路 TIMx 输出都有自己的死区值：

- DIDL1 = 1 时，为输出 1 上的 DTRx[8:0]
- DIDL2 = 1 时，为输出 2 上的 DTFx[8:0]

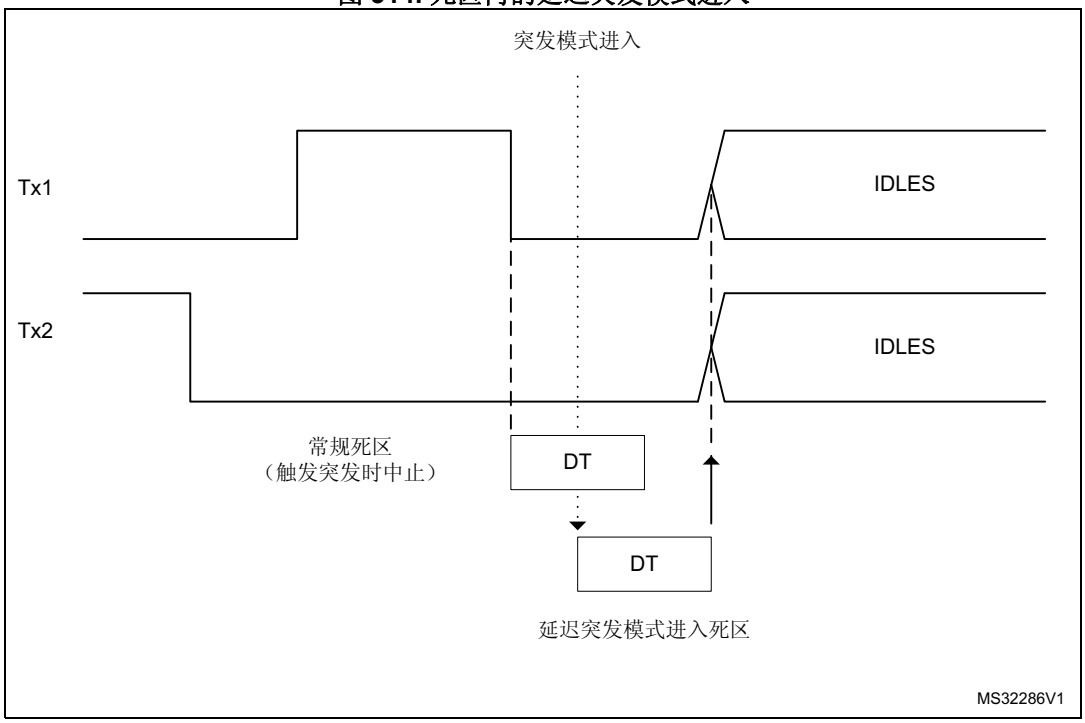
仅当其中一路输出在突发模式期间具有有效空闲电平 (IDLES=1)、且使用的死区为正值（SDTR/SDTF 设为 0）时，才可将 DIDLx 位置 1。

注： *延迟突发进入模式使用死区发生器资源。因此，如果 2 个 DIDLx 位中有任何一位置 1，且相应的定时单元使用死区插入（HRTIM_OUTxR 中的 DTEN 位置 1），则无法将 timerx 输出 2 用作外部事件过滤器（Tx2 过滤信号不可用）。*

如果由 DTRx[8:0] 和 DTFx[8:0] 定义的持续时间小于 3 个 f_{HRTIM} 时钟循环周期，则必须应用 *第 37.3.6 节* 中列出的与窄脉冲管理相关的限制。

如果突发模式进入在常规死区内到达，死区会中止，并会重新开始新的对应于无效周期的死区，如 *图 314* 所示。

图 314. 死区内的延迟突发模式进入



突发模式退出

突发模式退出通过软件强制进行（连续模式下），或在经过空闲周期后进行（单发模式下）。两种情况下，计数器都会立即重启（如果计数器通过 MTBM 或 TxBM 位 = 1 保持在复位状态），但输出状态仅会在编程的置位/复位事件后才会从空闲模式有效跳变为工作模式。

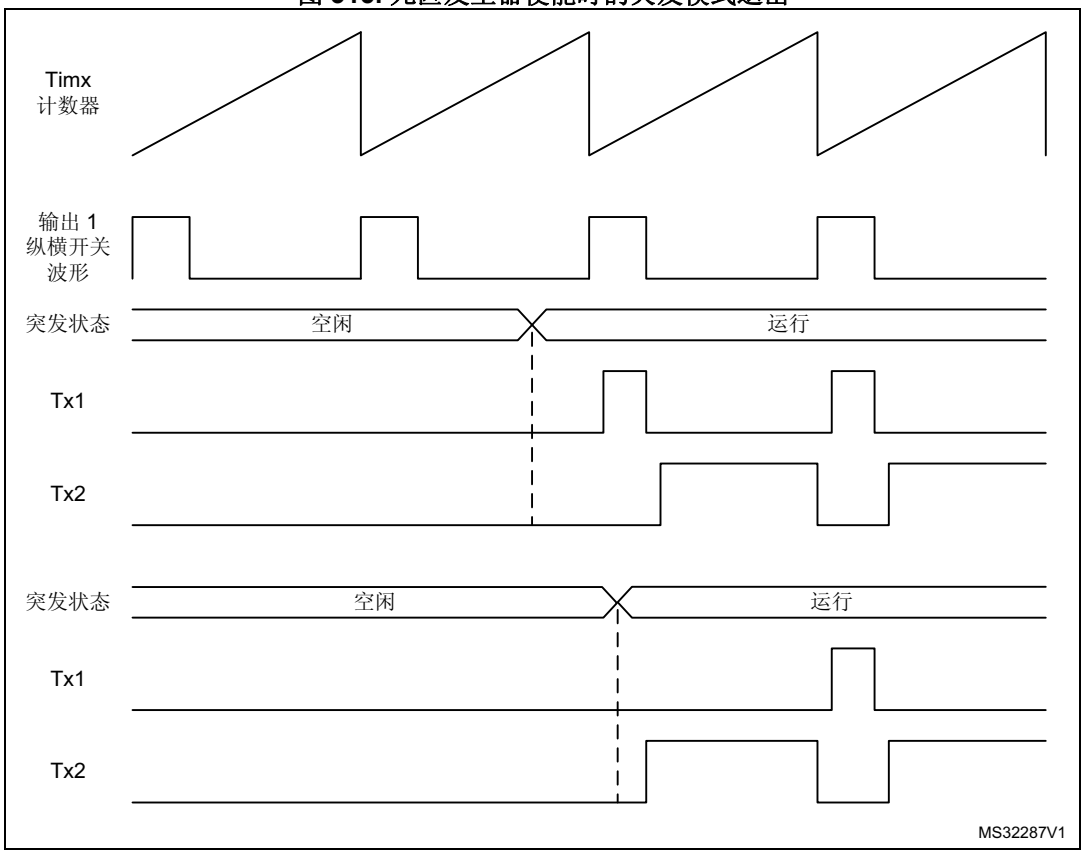
如果 HRTIM_IER 寄存器中的 BMPERIE 使能位置 1，则会在单发模式和连续模式下生成突发周期中断。该中断可用于在连续突发模式下将突发模式退出与突发周期进行同步。

图 315 显示了死区使能后是如何恢复正常工作的。虽然突发模式退出是立即进行的，但仅对任何互补输出上的第一个置位事件有效。

图中显示了两种不同情况：

1. 纵横开关输出波形上的信号无效时，突发模式结束。Tx1 输出上发生置位事件时会恢复 Tx1 和 Tx2 的有效状态，Tx2 输出不会在退出突发模式时获取互补电平。
2. 突发模式在纵横开关输出波形有效时结束：Tx2 输出上发生置位事件时会恢复工作，Tx1 不会在退出突发模式时立即获取有效电平。

图 315. 死区发生器使能时的突发模式退出



如果使能了推挽模式，上述行为会略有不同。如果输出无效，推挽模式会在周期开始使强制进行输出复位，如果前一周期内输出为高电平，则会对称地将输出强设置为有效电平。

因此，即使未明确编程跳变，也可在退出突发模式时复位空闲状态有效的输出。出于对称性的原因，即使未明确编程跳变，只要输出进入空闲状态时为有效电平，即可在退出突发模式时将输出置 1。

突发模式寄存器预装载和更新

BMPREN 位（突发模式预装载使能）可进行突发模式比较并预装载周期寄存器（HRTIM_BMCMP 和 HRTIM_BMPER）。

当 BMPREN 置 1 时，预装载寄存器的内容会传输到活动寄存器：

- 突发模式使能时 (BME = 1)
- 突发模式周期结束时

如果对 HRTIM_BMPER 周期寄存器进行写操作，则会暂时禁止更新，直到向 HRTIM_BMCMP 比较寄存器进行写操作为止，这样可确保两个寄存器在进行修改时是一致的。

如果只需要更改比较寄存器，则只需要进行一次写操作。如果只需要更改周期，还需要重新写入比较寄存器，以便将新值纳入考虑范围。

BMPREN 位复位时，对 BMCMPR 和 BMPER 进行写访问会直接更新活动寄存器。此时，对于以下 2 种情况，需要考虑何时在总突发周期内进行更新：

a) 比较寄存器更新

如果新的比较值大于当前突发模式计数器值，则会在当前周期内考虑新的比较值。

如果新的比较值小于当前突发模式计数器值，在连续模式下，会在下一个周期内考虑新的比较值；在单发模式下，则会忽略新的比较值（不会出现比较匹配的情况，并将持续处于空闲状态，直至空闲周期结束）。

b) 周期寄存器更新

如果新的周期值大于当前突发模式计数器值，则会在当前周期内考虑此更改。

注： *如果新的周期值小于当前突发模式计数器值，则不会考虑新的周期值，突发模式计数器将溢出（达到 0xFFFF 时溢出），更改将在下一周期生效。在单发模式下，计数器将在达到 0xFFFF 时翻转，突发模式将重新启动并再持续一个周期，直到达到新的编程值。*

通过复合寄存器进行突发模式仿真

突发模式控制器仅会控制单个转换器的一个定时器或一组定时器。需要为多个独立定时器使用突发模式时，可以使用 DMA 和 HRTIM_CMP1CxR 复合寄存器仿真简单的突发模式控制器，复合寄存器会保存重复寄存器和比较 1 寄存器的别名。

此方法适用于仅需要使用简单 PWM 来更新占空比的转换器（通常是降压转换器）。在这种情况下，使用 CMP1 寄存器复位输出（并定义占空比），而输出会在发生周期事件时置 1。

此时，对 CMP1CxR 进行一次 32 位写访问便足以定义占空比（使用 CMP1 值）以及保持该占空比的周期数（使用重复值）。要实现突发模式，只需要在连续模式下通过 DMA（在发生重复事件时）传输两个 32 位数据，数据结构如下：

CMPC1xR = {REP_Run; CMP1 = Duty_Cycle}, {REP_Idle; CMP1 = 0}

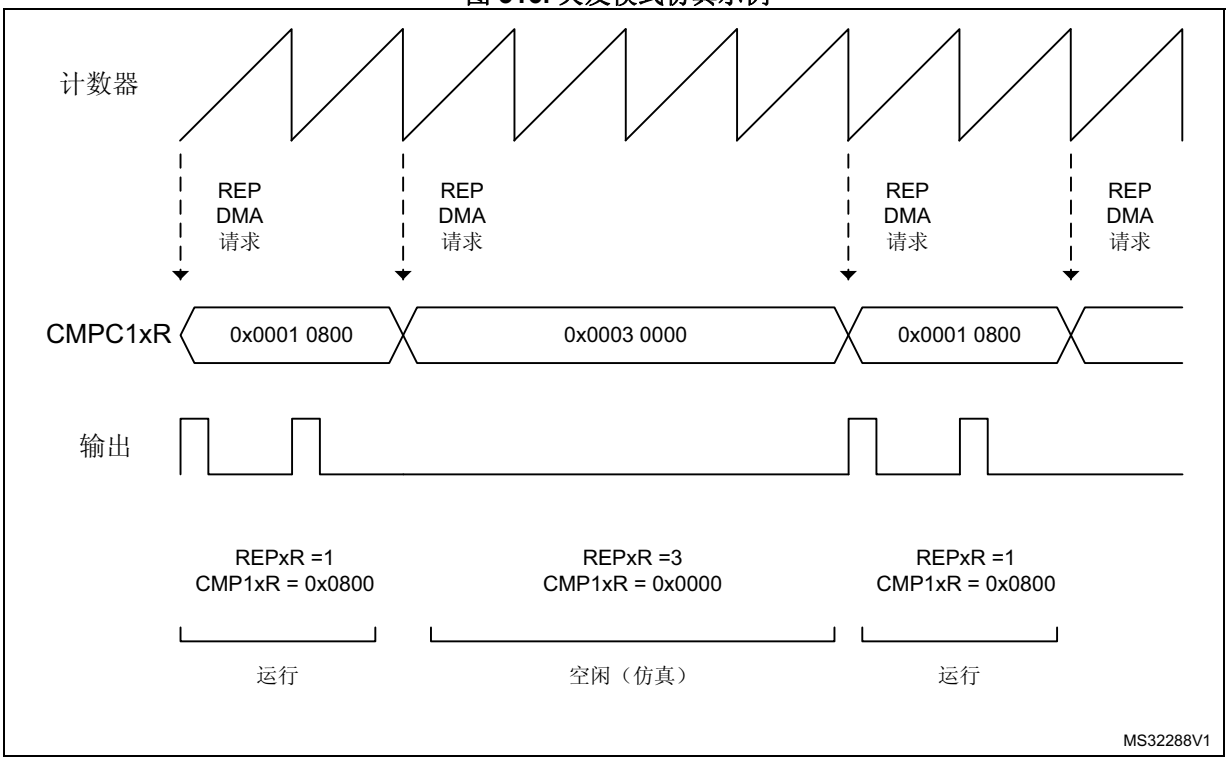
举例来说，数值：

{0x0003 0000}：CMP1 = 0，持续 3 个周期

{0x0001 0800}：CMP1 = 0x0800，持续 1 个周期

提供的突发模式每 6 个 PWM 周期会有 2 个周期有效，如*图 316* 所示。

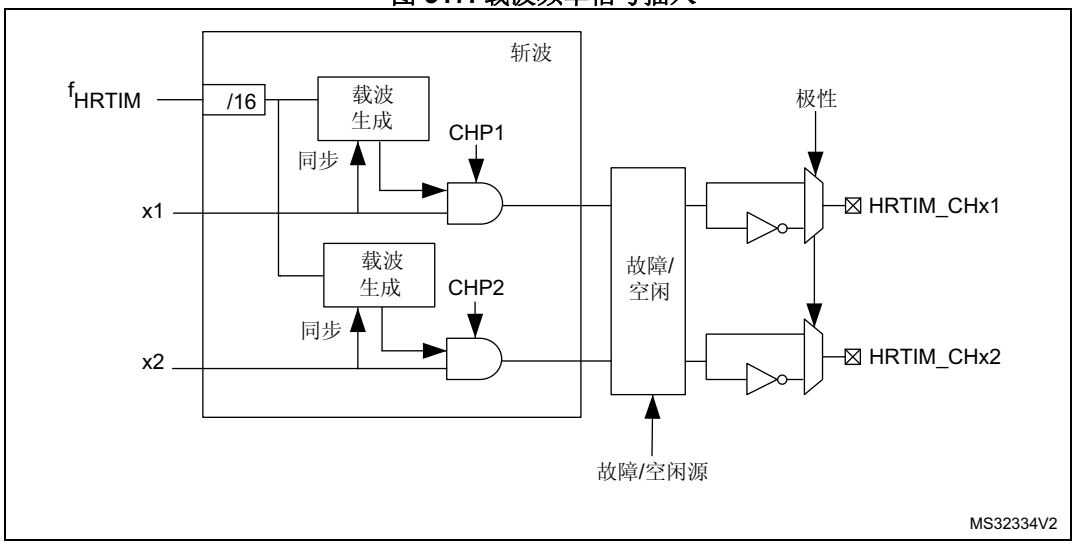
图 316. 突发模式仿真示例



37.3.14 斩波

可在定时单元输出信号上添加高频载波，以驱动隔离变压器。此操作是在插入极性之前对输出级执行的（如*图 317* 所示），会使用 HRTIM_OUTxR 寄存器中的 CHP1 和 CHP2 位分别在输出 1 和输出 2 上使能斩波。

图 317. 载波频率信号插入



可通过 HRIM_CHPxR 寄存器调整斩波参数，以在脉冲开始处定义一个特定脉宽，后跟频率和占空比可编程的载波频率，如*图 318* 所示。

CARFRQ[3:0] 位按照公式 $F_{CHPFRQ} = f_{HRTIM} / (16 \times (CARFRQ[3:0]+1))$ 定义频率，范围从 156 MHz 到 25 MHz（对于 f_{HRTIM} = 400 MHz)。

可通过 CARDTY[2:0] 调整占空比（步长为 1/8），占空比范围为 0/8 到 7/8。
CARDTY[2:0] = 000 时（占空比 = 0/8），输出波形仅包含参考波形上升沿之后的启动脉冲，不会添加任何载波。

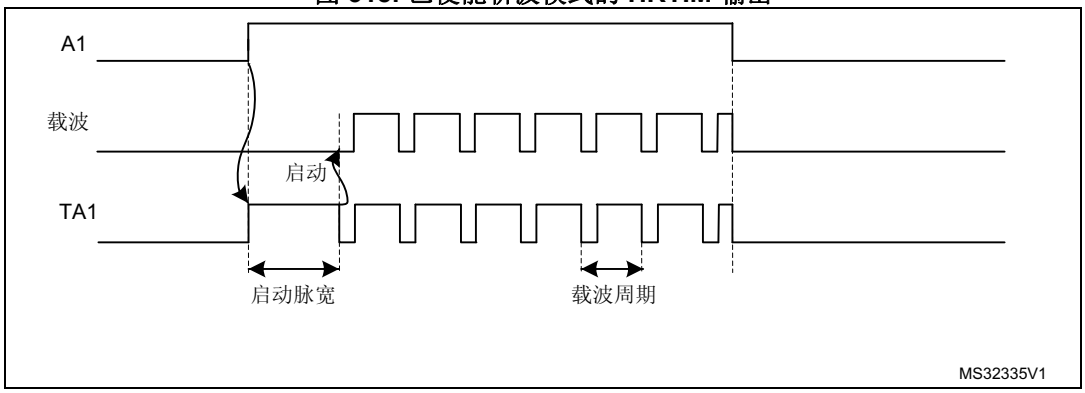
初始脉冲的脉宽是通过 STRPW[3:0] 位域定义的，具体如下：
$t1STPW = (STRPW[3:0]+1) \times 16 \times t_{HRTIM}$。脉宽范围为 40 ns 到 0.63 µs（对于 f_{HRTIM}=400 MHz）。

载波频率参数是根据 f_{HRTIM} 频率定义的，与 CKPSC[2:0] 设置无关。

在斩波模式下，载波频率和初始脉宽会通过逻辑与函数与参考波形相结合。初始脉冲结束时会执行同步，以获得重复的信号波形。

斩波信号会在输出波形有效状态结束时停止，不会等到当前载波周期结束。因此，斩波信号包含的脉冲可能比编程的脉冲短。

图 318. 已使能斩波模式的 HRTIM 输出



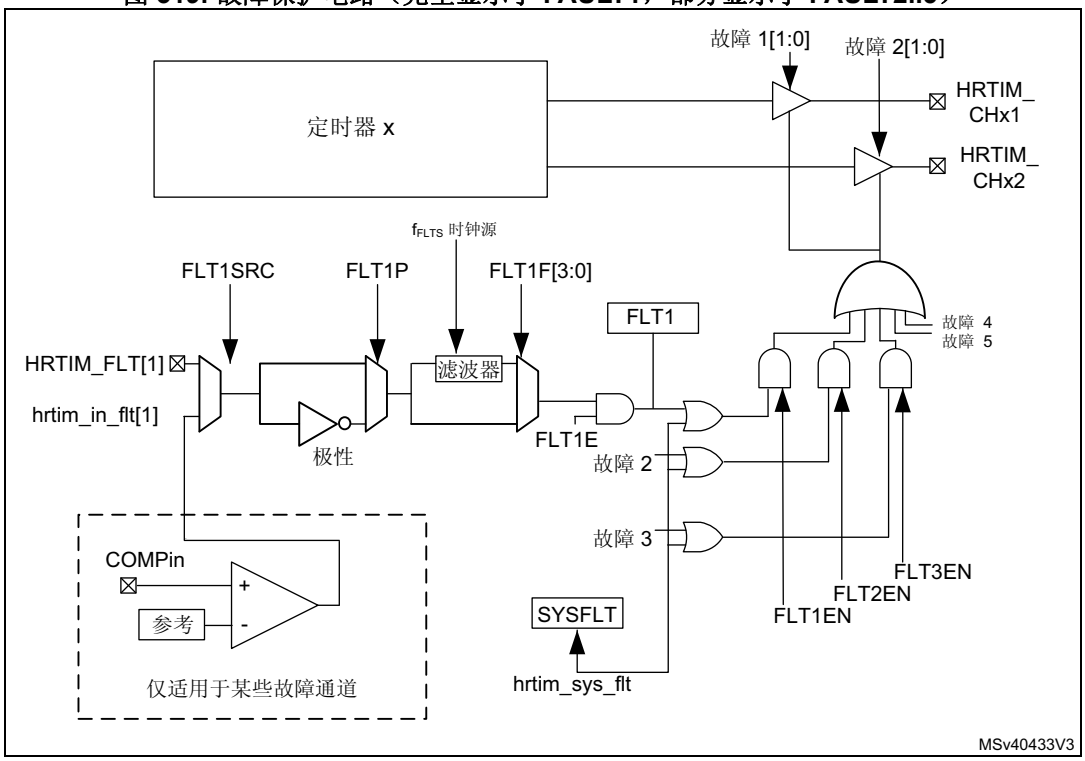
注： *必须在通过 HRTIM_OENR 寄存器中的 TxyOEN 位使能输出之前将 CHP1 和 CHP2 位置 1。*
斩波模式激活时（两个 CHPx 位中至少有一个置 1），不能修改 CARFRQ[2:0]、CARDTY[2:0] 和 STRPW[3:0] 位域。

37.3.15 故障保护

HRTIMER 具有通用的故障保护电路，可在发生异常操作时禁止输出。一旦触发故障，输出便会进入预定义的安全状态。输出通过软件重新使能之前，都会保持此状态。如果是永久故障请求，即使软件尝试重新使能输出，输出也将保持其故障状态，直至故障源消失。

HRTIM 具有 5 个 FAULT 输入通道；所有通道均可用，并可结合起来用于 5 个定时单元中的每一个，如*图 319* 所示。

图 319. 故障保护电路（完全显示了 FAULT1，部分显示了 FAULT2..5）



在连接到定时单元之前，每条故障通道均可完全通过 HRTIM_FLTINR1 和 HRTIM_FLTINR2 寄存器进行配置。FLTxSRC 位用于选择故障信号源，可以是数字输入或内部事件（内置比较器输出）。

表 301 总结了可用于 10 个故障通道的源：

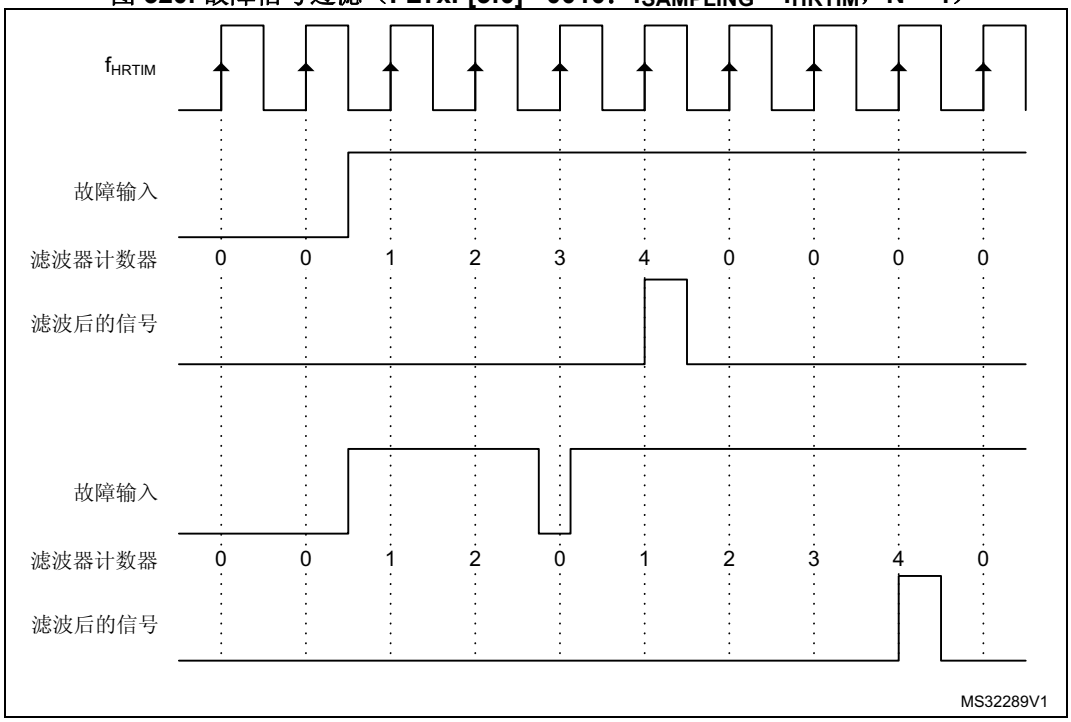
表 301. 故障输入

故障通道	外部输入 (FLTxSRC = 0)	片上源 (FLTxSRC = 1)
FAULT 1	PA15	COMP1
FAULT 2	PC11	COMP2
FAULT 3	PD4	NC
FAULT 4	PB3	NC
FAULT 5	PG10	NC

可通过 HRTIM_FLTINRx 寄存器中的 FLTxP 极性位选择信号的极性，以定义有效电平。如果 FLTxP = 0，信号低电平有效；如果 FLTxP = 1，信号高电平有效。

可在极性设置后过滤故障信息。如果 FLTxF[3:0] 位域设为 0000，则不会对信号进行过滤，信号将独立于 f_{HRTIM} 时钟异步操作。对于所有其他 FLTxF[3:0] 位域值，会对信号进行数字过滤。数字滤波器由计数器组成，需要使用 N 个有效样本来验证输出跳变。如果输入值在计数器达到 N 值之前发生变化，计数器会复位，跳变会被丢弃（视为伪事件）。如果计数器达到 N，跳变被视为有效，并会作为正确的外部事件进行传输。因此，数字滤波器会向正在进行滤波的外部事件添加延迟，延迟时长视采样时钟和滤波器长度（预期的有效样本数）而定。*图 320* 显示了伪故障信号的过滤方式。

图 320. 故障信号过滤（FLTxF[3:0]= 0010：$f_{SAMPLING} = f_{HRTIM}$，N = 4）



过滤周期从 2 个 f_{HRTIM} 时钟周期到 8 个 32 分频的 f_{FLTS} 时钟周期。f_{FLTS} 通过 HRTIM_FLTINR2 寄存器中的 FLTSD[1:0] 位定义。*表 302* 总结了采样速率和滤波器长度。必须将滤波器长度减去 1 个采样时钟周期的抖动，以考虑因采样造成的不确定性，从而实现有效过滤。

表 302. 采样速率和滤波器长度与 FLTFxF[3:0] 和时钟设置的关系

	f_{FLTS} 与 FLTSD[1:0]				f_{HRTIM} = 400 MHz 时的滤波器长度	
FLTFxF[3:0]	**00**	**01**	**10**	**11**	**最小值**	**最大值**
0001,0010,0011	f_{HRTIM}	f_{HRTIM}	f_{HRTIM}	f_{HRTIM}	f_{HRTIM}，N =2 5 ns	f_{HRTIM}，N =8 20 ns
0100, 0101	f_{HRTIM} /2	f_{HRTIM} /4	f_{HRTIM} /8	f_{HRTIM} /16	f_{HRTIM} /2，N = 6 30 ns	f_{HRTIM} /16，N = 8 320 ns
0110, 0111	f_{HRTIM} /4	f_{HRTIM} /8	f_{HRTIM} /16	f_{HRTIM} /32	f_{HRTIM} /4，N = 6 60 ns	f_{HRTIM} /32，N = 8 640 ns
1000, 1001	f_{HRTIM} /8	f_{HRTIM} /16	f_{HRTIM} /32	f_{HRTIM} /64	f_{HRTIM} /8，N = 6 120 ns	f_{HRTIM} /64，N = 8 1.28 µs
1010, 1011, 1100	f_{HRTIM} /16	f_{HRTIM} /32	f_{HRTIM} /64	f_{HRTIM} /128	f_{HRTIM} /16，N = 5 200 ns	f_{HRTIM} /128，N = 8 2.56 µs
1101, 1110, 1111	f_{HRTIM} /32	f_{HRTIM} /64	f_{HRTIM} /128	f_{HRTIM} /256	f_{HRTIM} /32，N = 5 400 ns	f_{HRTIM} /256，N = 8 5.12 µs

系统故障输入 (hrtim_sys_flt)

该故障由 MCU B 类电路提供（有关详细信息，请参见系统配置控制器 (SYSCFG) 部分），对应于来自以下源的系统故障：

- 时钟安全系统
- SRAM 奇偶校验器
- Cortex®-M7-lockup 信号
- PVD 检测器

此输入会覆盖 FAULT 输入，并禁止所有 FAULTy[1:0] = 01、10、11 的输出。

对于每条 FAULT 通道，可通过 HRTIM_FLTxR 寄存器中仅可写入一次的 FLTxLCK 位锁定 FLTxE、FLTxP、FLTxSRC、FLTxF[3:0] 位（使其成为只读位），以实现功能安全。使能后，故障调节设置会冻结，直到下一次 HRTIM 复位或系统复位。

按上述方法对故障信号进行调节后，信号会发送到定时单元。对于任一定时单元，都会使用 HRTIM_FLTxR 寄存器中的 FLT1EN 到 FLT5EN 位使能 5 条故障通道，并且可同时选择 5 条通道（只要输出受故障机制保护，便会自动使能 sysfault）。这样便可实现：

- 一条故障通道同时禁止多个定时单元
- 多条故障通道进行或运算来禁止单个定时单元

HRTIM_FLTxR 寄存器中仅可写入一次的 FLTLCK 位可在下一次复位之前锁定 FLTxEN 位（使其成为只读位），以实现功能安全。使能后，定时单元故障相关设置会冻结，直到下一次 HRTIM 复位或系统复位。

对于每个定时器，故障期间的输出状态是通过 HRTIM_OUTxR 寄存器中的 FAULT1[1:0] 和 FAULT2[1:0] 位定义的（请参见*第 37.3.12 节*）。

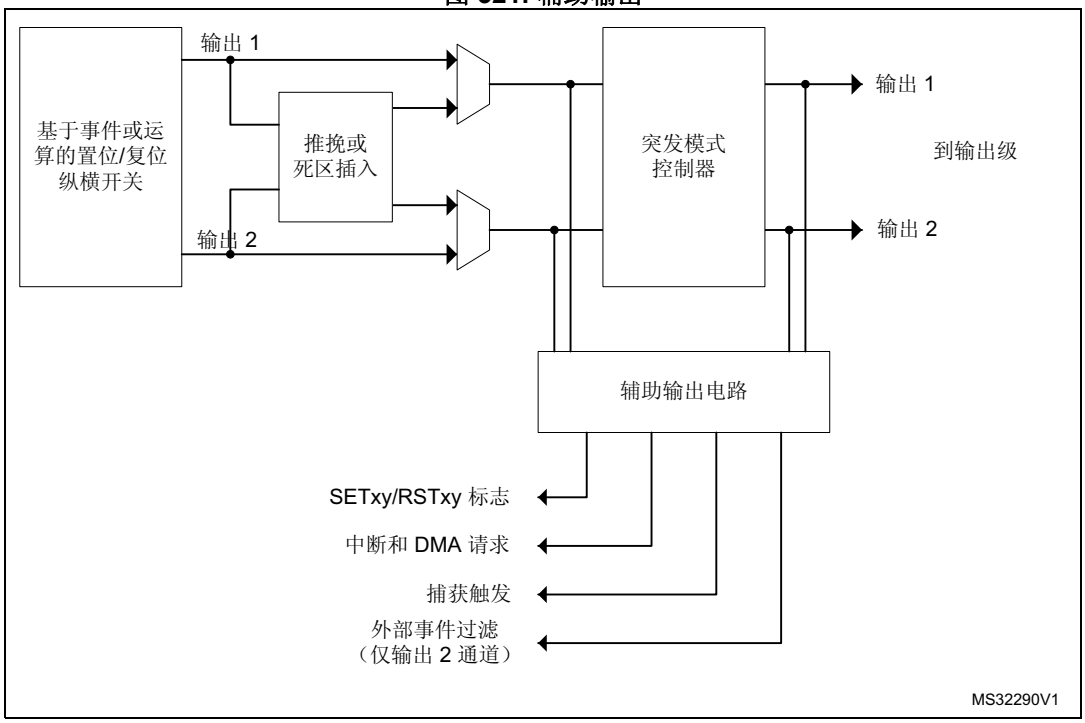
37.3.16 辅助输出

定时器 A 和 E 所具有的辅助输出与进入输出级的常规输出并行工作。辅助输出可提供以下内部状态、事件和信号：

- SETxy 和 RSTxy 状态标志，以及相应的中断和 DMA 请求
- 输出置位/复位后的捕获触发事件
- Tx2 输出复制后的外部事件过滤信号（有关详细信息，请参见*第 37.3.8 节*）

辅助输出会在突发模式控制器之前或之后获取，具体视 HRTIM 工作模式而定。*图 321* 给出了概览。

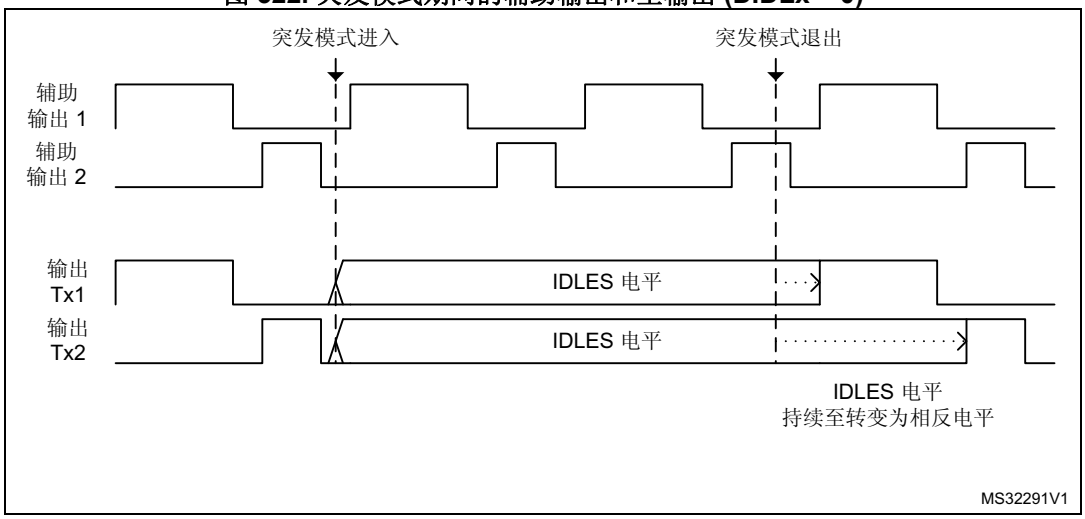
图 321. 辅助输出



默认情况下，辅助输出是输出 Tx1 和 Tx2 的副本。例外情况包括：

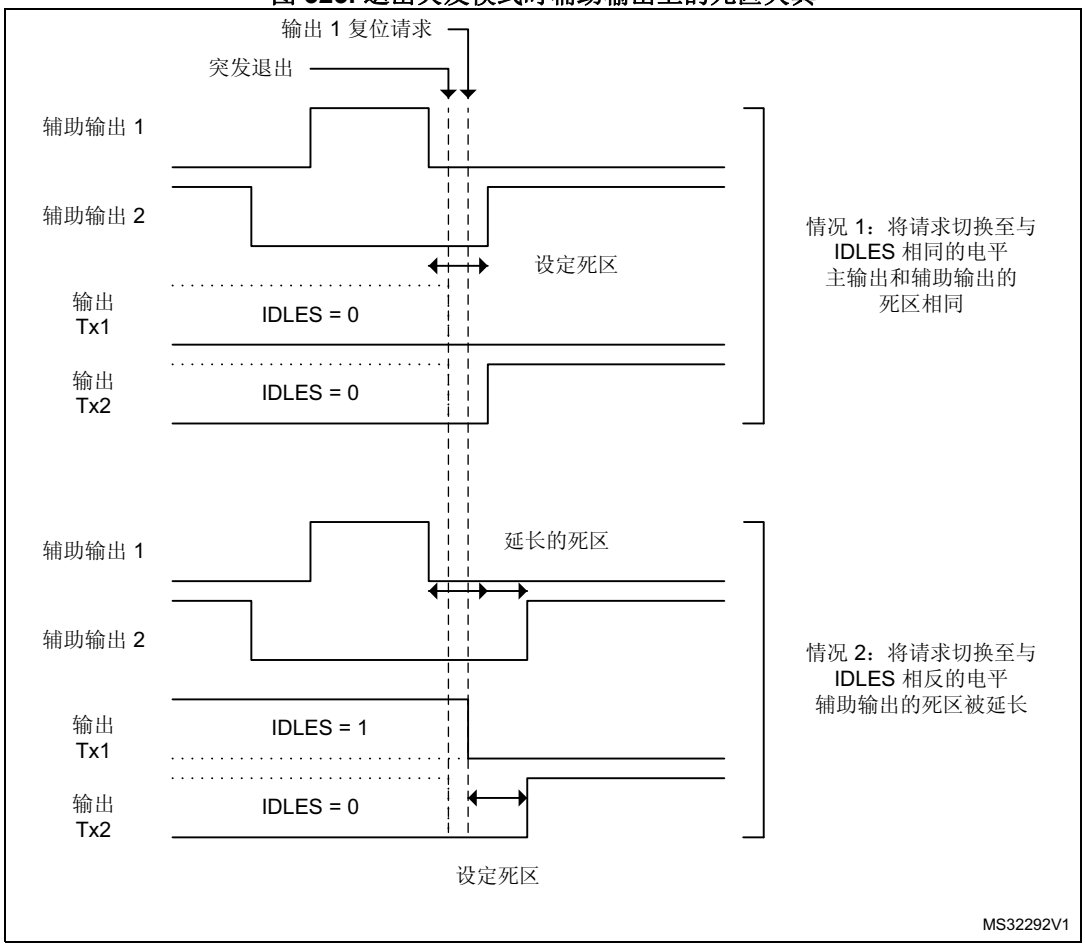
- 禁止死区时 (DTEN = 0) 的延迟空闲和均衡空闲保护。触发保护后，辅助输出会保持不变，并会遵循纵横开关发出的信号。相反，如果使能了死区 (DTEN = 1)，主输出和辅助输出都会强制设为无效电平。
- 突发模式（TCEN=1，IDLEMx=1）；有两种情况：
  a) 如果 DTEN=0 或 DIDLx=0，辅助输出将不受突发模式进入的影响，而会继续遵循纵横开关发出的参考信号（请参见*图 322*）。
  b) 如果同时使能了死区 (DTEN=1) 和延迟突发模式进入 (DIDLx=1)，则辅助输出与主输出的行为相同。它们会在死区持续时间结束后强制设为 IDLES 电平，随后会在所有突发周期内保持这一电平。突发模式终止后，会保持 IDLES 电平，直至跳变到相反电平（与主输出相似）。

图 322. 突发模式期间的辅助输出和主输出 (DIDLx = 0)



如果在死区内发生从突发模式退出或者在延迟保护后重新使能输出，则辅助输出上的信号会略有失真。这种情况下，应用到辅助输出的死区会延长，以便能够符合主输出上的死区要求。*图 323* 给出了一些例子。

图 323. 退出突发模式时辅助输出上的死区失真



37.3.17 将 HRTIM 与其他定时器或 HRTIM 实例同步

HRTIM 可作为主单元（生成同步信号）或从单元（等待触发被同步）来同步多个 HRTIM 实例。此功能也可用于将 HRTIM 与其他外部或片上定时器进行同步。同步电路在主定时器内进行控制。

同步输出

本节介绍了必须对 HRTIM 进行何种配置才能同步外部资源并充当主定时器单元。

可选择四种事件作为要发送到同步输出的源，方法是使用 HRTIM_MCR 寄存器中的 SYNCSRC[1:0] 位，具体如下：

- 00：主定时器启动
  当 MCEN 位置 1，或者定时器在单发模式下达到周期值后重新启动时，会生成该事件。如果计数期间发生复位（CONT 或 RETRIG 位置 1），也会生成该事件。
- 01：主定时器比较 1 事件
- 10：定时器 A 启动
  当 TACEN 位置 1，或者计数器复位并重新开始计数（响应此次复位）时，会生成该事件。以下计数器复位事件不会传播到同步输出：连续模式下的计数器翻转、单发不可再触发模式下被丢弃的复位请求。仅当计数期间发生复位时（CONT 或 RETRIG 位置 1），才会考虑复位。
- 11：定时器比较 1 事件

HRTIM_MCR 寄存器中的 SYNCOUT[1:0] 位指定了同步事件的生成方式。

如果 SYNCOUT[1:0] = 1x，则会在 HRTIM_SCOUT 输出引脚上生成同步脉冲。SYNCOUT[0] 位指定了同步信号的极性。如果 SYNCOUT[0] = 0，HRTIM_SCOUT 引脚具有低空闲电平，并会发出长度为 16 个 f_{HRTIM} 时钟周期的正脉冲进行同步。如果 SYNCOUT[0] = 1，空闲电平为高电平，并会生成负脉冲。

注： *同步脉冲后会有 16 个 f_{HRTIM} 时钟周期的空闲电平，在此期间，会丢弃任何新的同步请求。因此，最大同步频率为 $f_{HRTIM}/32$。*

SYNCOUT[1:0] 位使能后（也就是位域值不是 00 后），会立即在 HRTIM_SCOUT 引脚上施加空闲电平。

必须在配置 MCU 输出和计数器使能之前执行同步输出初始化步骤，具体执行步骤如下：

1. 配置 HRTIM_MCR 中的 SYNCOUT[1:0] 和 SYNCSRC[1:0] 位域
2. 配置 HRTIM_SCOUT 引脚（请参见通用 I/O 部分）
3. 使能主定时器或定时器 A 计数器（MCEN 或 TACEN 位置 1）

使能同步输入模式并同时启动计数器（使用 SYNCSTRTM/SYNCSTRTx 位）和同步输出模式（SYNCSRC[1:0] = 00 或 10）后，仅当计数器将在运行时启动或复位时，才会生成输出脉冲。如果复位请求将计数器清零而不启动计数器，则不会影响同步输出。

同步输入

HRTIM 可通过外部源进行同步，具体通过对 HRTIM_MCR 寄存器中的 SYNCIN[1:0] 位进行编程来实现：

- 00：禁止同步输入
- 01：保留配置
- 10：片上 TIM1 通用定时器（TIM1 TRGO 输出）
- 11：HRTIM_SCIN 输入引脚上的正脉冲

目标定时器（主定时器或定时单元）使能后（MCEN 和/或 TxCEN 位置 1），不能更改此位域。

HRTIM_SCIN 输入为上升沿有效。定时器行为是通过 HRTIM_MCR 和 HRTIM_TIMxCR 寄存器中的下列位定义的（有关详细信息，请参见*表 303*）：

- 同步启动：传入信号会启动定时器的计数器（SYNCSTRTM 和/或 SYNCSTRTx 位置 1）。TxCEN (MCEN) 位必须置 1 才能使能定时器并使计数器准备好启动。在连续模式下，计数器在收到同步信号之前不会启动。
- 同步复位：传入信号会复位计数器（SYNCRSTM 和/或 SYNCRSTx 位置 1）。该事件会像其他任何复位事件一样使重复计数器递减。

仅当相关计数器使能后（MCEN 或 TxCEN 位置 1），才会考虑同步事件。同步请求会触发 SYNC 中断。

注： *如果当前计数器值大于有效周期值，同步启动事件会复位计数器。*

同步事件的作用取决于定时器工作模式，请参见*表 303* 中总结的内容。

表 303. 同步事件的作用与定时器工作模式之间的关系

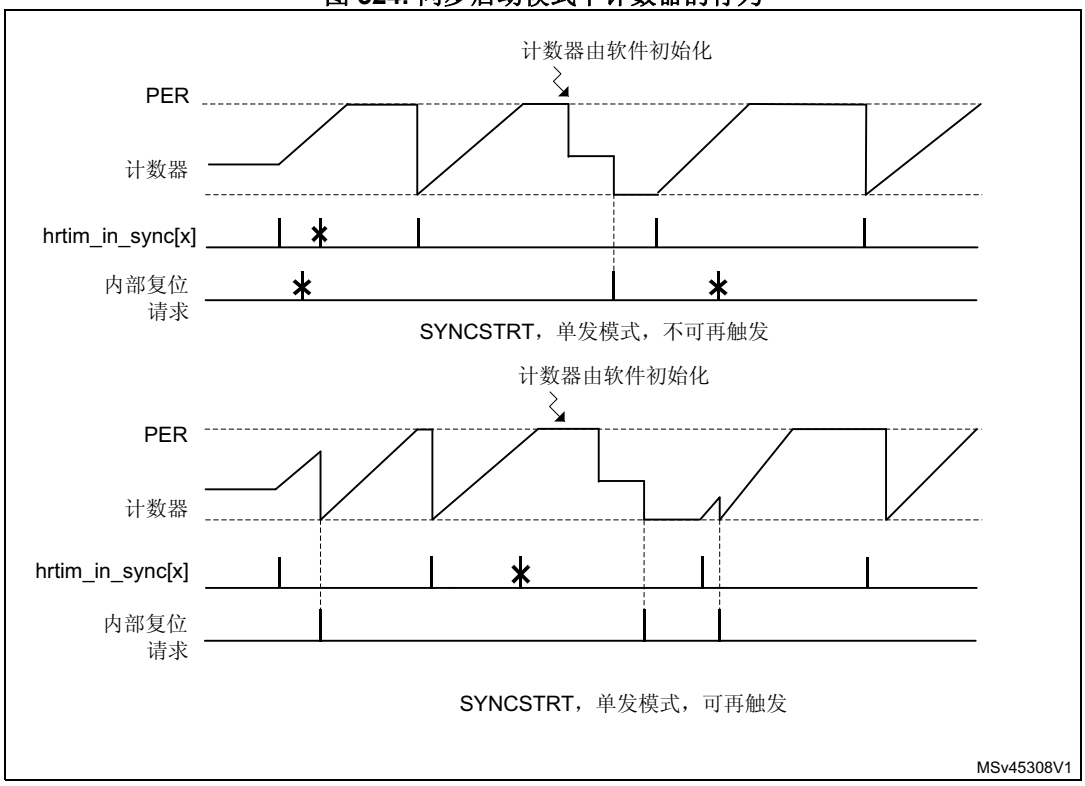
工作模式	SYNC RSTx	SYNC STRTx	SYNC 复位或启动事件后的行为
单发 不可再触发	0	1	当计数器停止并且满足以下条件时，会考虑启动事件： – MCEN 或 TxCEN 位置 1。 – 已达到周期时间。 如果在计数器因达到周期值而停止时发生启动事件，则会将计数器复位。复位请求会将计数器清零，但不会启动计数器（计数器仅可通过同步事件重启）。任何在计数期间发生的复位事件都会被忽略（与在常规不可再触发模式下时一样）。
	1	X	复位事件将启动定时器计数操作。仅当计数器停止并且满足以下条件时，才会考虑复位事件： – MCEN 或 TxCEN 位置 1。 – 已达到周期时间。 如果选择了多个复位请求（来自 HRTIM_SCIN 和内部事件），则仅会考虑第一个到达的请求。

表 303. 同步事件的作用与定时器工作模式之间的关系（续）

工作模式	SYNC RSTx	SYNC STRTx	SYNC 复位或启动事件后的行为
单发 可再触发	0	1	仅当计数器未启动或周期结束时，计数器启动才有效。任何在计数器启动后发生的同步事件都无效。 如果在计数器因达到周期值而停止时发生启动事件，则会将计数器复位。复位请求会将计数器清零，但不会启动计数器（计数器仅可通过同步事件启动）。计数过程中发生的复位事件会被考虑（与在常规可再触发模式下时一样）。
	1	X	HRTIM_SCIN 发出的复位请求会被当作来自内部事件的 HRTIM 计数器复位请求进行考虑，并将启动或重新启动定时器计数操作。 如果选择了多个复位请求，则会考虑第一个到达的请求。
连续模式	0	1	定时器已使能（MCEN 或 TxCEN 位置 1），并正在等待同步事件启动计数器。计数器启动后发生的任何同步事件均不起作用（计数器仅可通过同步事件启动）。复位请求会将计数器清零，但不会启动计数器。
	1	X	HRTIM_SCIN 发出的复位请求会被当作来自内部事件的 HRTIM 计数器复位请求进行考虑，并将启动或重新启动定时器计数操作。如果选择了多个复位请求，则会考虑第一个到达的请求。

图 324 显示了单发模式下如何进行同步启动。

图 324. 同步启动模式下计数器的行为



37.3.18 ADC 触发

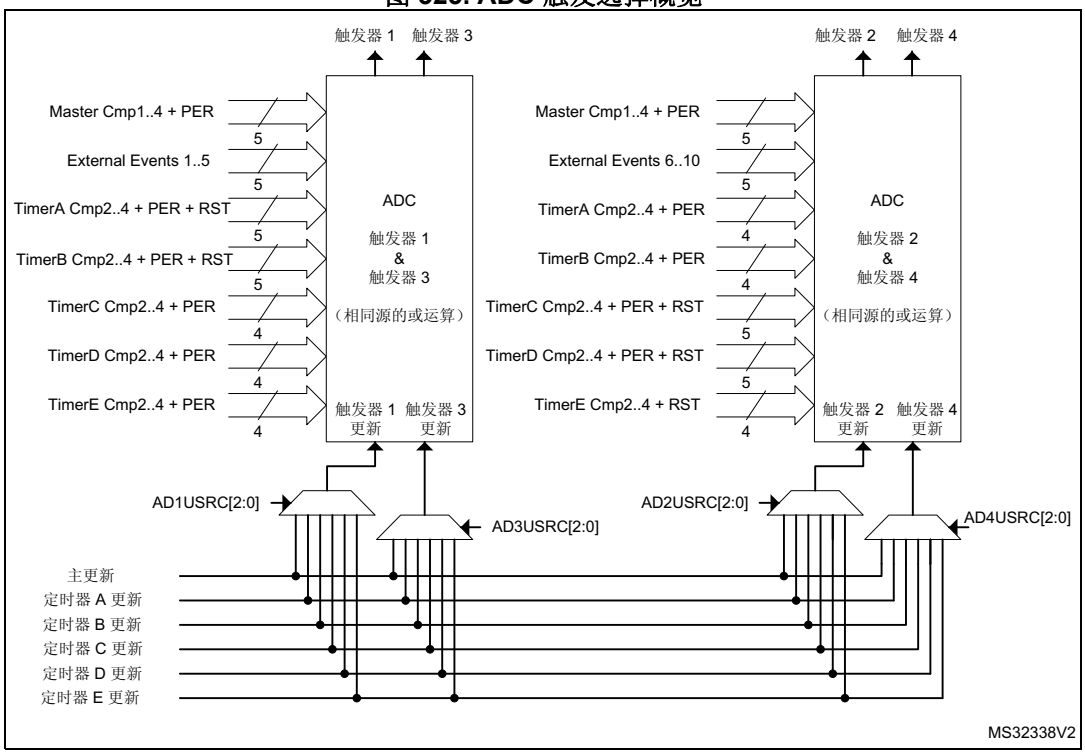
ADC 可由主定时器和 5 个定时单元触发。

可通过 4 个独立的触发信号同时启动 2 个 ADC 的常规定序器和注入定序器。在 HRTIM_ADC1R 到 HRTIM_ADC4R 寄存器中，最多可为每路触发输出合并 32 个事件（逻辑或运算），如*图 325* 所示。触发信号 1 和 3 与 2 和 4 将使用一组相同的源。

外部事件可用作触发信号。这类触发信号会在 HRTIM_EECRx 寄存器中定义的调节结束后立即获取，并且与 EEFxR1 和 EEFxR2 寄存器设置无关。

可通过同时选择多个源的方式在单个开关周期内进行多次触发。典型用例是用于非重叠多相转换器，可通过单个 ADC 触发输出对所有相位进行连续采样。

图 325. ADC 触发选择概览



HRTIM_ADC1R 到 HRTIM_ADC4R 寄存器会预装载，并可与相关定时器同步更新。更新源是通过 HRTIM_CR1 寄存器中的 ADxUSRC[2:0] 位定义的。

举例来说，如果 ADC 触发 1 输出定时器 A CMP2 事件 (HRTIM_ADC1R = 0x0000 0400)，HRTIM_ADC1R 通常将与定时器 A 同步更新 (AD1USRC[2:0] = 001)。

如果源定时器中禁止预装载（PREEN 位复位），HRTIM_ADCxR 寄存器也不会预装载：写访问会立即更新触发源。

37.3.19 DAC 触发

HRTIMER 可使嵌入式 DAC 与定时器同步更新。

主定时器和定时器单元发出的更新事件可在 3 路 hrtim_dac_trgx 输出的任意一路上生成 DAC 更新触发信号。

注： *每个定时器都有自己的 DAC 相关控制寄存器。*

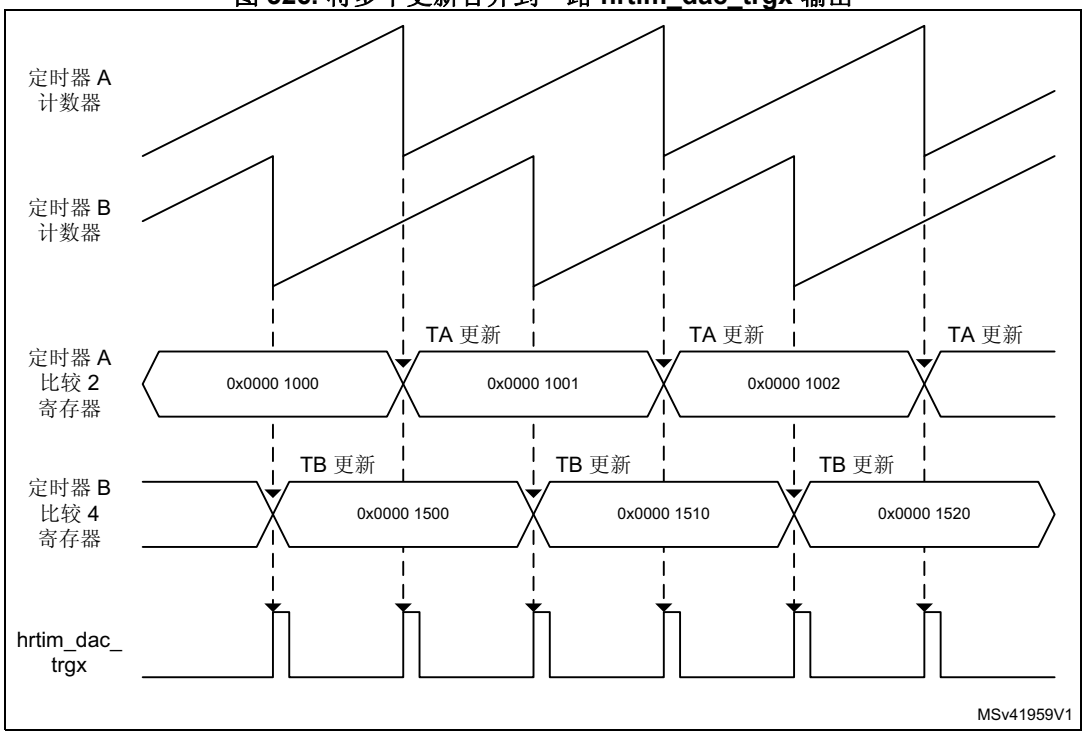
HRTIM_MCR 和 HRTIM_TIMxCR 寄存器 DACSYNC[1:0] 位的编程值含义如下：

- 00：未生成更新
- 01：在 hrtim_dac_trg1 上生成更新
- 10：在 hrtim_dac_trg2 上生成更新
- 11：在 hrtim_dac_trg3 上生成更新

会在 hrtim_dac_trgx 输出上生成 1 个 f_{HRTIM} 时钟周期的输出脉冲。

如果多个定时器中使能了 DACSYNC[1:0] 位，hrtim_dac_trgx 输出将由所有定时器更新事件的或运算结果决定。例如，如果定时器 A 和定时器 B 中 DACSYNC = 1，那么定时器 A 中的更新事件将与定时器 B 中的更新事件进行或运算，在相应的 hrtim_dac_trgx 输出上生成 DAC 更新触发信号，如*图 326* 所示。

图 326. 将多个更新合并到一路 hrtim_dac_trgx 输出



hrtim_dac_trgx 引脚连接到 ADC，具体如下：

- hrtim_dac_trg1：DAC1_CH1 触发输入 9（DAC1 外设的 DAC_CR 中的 TSEL1[2:0] = 1001）
- hrtim_dac_trg2：DAC1_CH2 触发输入 10（DAC1 外设的 DAC_CR 中的 TSEL1[2:0] = 1010）
- hrtim_dac_trg3：未连接

37.3.20 HRTIM 中断

主定时器可生成 7 个中断：

- 主定时器寄存器更新
- 接收到同步事件
- 主定时器重复事件
- 主定时器比较 1 到 4 事件

每个定时单元可生成 14 个中断：

- 延迟保护已触发
- 计数器复位或翻转事件
- 输出 1 和输出 2 复位（从有效电平跳变为无效电平）
- 输出 1 和输出 2 置位（从无效电平跳变为有效电平）
- 捕获 1 和 2 事件
- 定时单元寄存器更新
- 重复事件
- 比较 1 到 4 事件

会为整个 HRTIM 生成 8 个全局中断：

- 系统故障和故障 1 到 5（不考虑定时单元的因素）
- 突发模式周期结束

中断请求会分到 7 个向量组中，具体如下：

- hrtim_mst_it：主定时器中断（主定时器更新、同步输入、重复、MCMP1..4）和全局中断，故障除外（突发模式周期）
- hrtim_tima_it：TIMA 中断
- hrtim_timb_it：TIMB 中断
- hrtim_timc_it：TIMC 中断
- hrtim_timd_it：TIMD 中断
- hrtim_time_it：TIME 中断
- hrtim_fault_it：所有的故障中断，允许高优先级的中断向量控制

表 304 总结了中断请求及其映射以及关联的控制和状态位。

表304. HRTIM 中断汇总

中断向量	中断事件	事件标志	使能控制位	标志清零位
hrtim_mst_it	突发模式周期结束	BMPER	BMPERIE	BMPERC
	主定时器寄存器更新	MUPD	MUPDIE	MUPDC
	接收到同步事件	SYNC	SYNCIE	SYNCC
	主定时器重复事件	MREP	MREPIE	MREPC
	主定时器比较 1 到 4 事件	MCMP1	MCMP1IE	MCP1C
		MCMP2	MCMP2IE	MCP2C
		MCMP3	MCMP3IE	MCP3C
		MCMP4	MCMP4IE	MCP4C
hrtim_tima_it hrtim_timb_it hrtim_timc_it hrtim_timd_it hrtim_time_it	延迟保护已触发	DLYPRT	DLYPRTIE	DLYPRTC
	计数器复位或翻转事件	RST	RSTIE	RSTC
	输出 1 和输出 2 复位（从有效电平跳变为无效电平）	RSTx1	RSTx1IE	RSTx1C
		RSTx2	RSTx2IE	RSTx2C
	输出 1 和输出 2 置位（从无效电平跳变为有效电平）	SETx1	SETx1IE	SETx1C
		SETx2	SETx2IE	SETx2C
	捕获 1 和 2 事件	CPT1	CPT1IE	CPT1C
		CPT2	CPT2IE	CPT2C
	定时单元寄存器更新	UPD	UPDIE	UPDC
	重复事件	REP	REPIE	REPC
	比较 1 到 4 事件	CMP1	CMP1IE	CMP1C
		CMP2	CMP2IE	CMP2C
		CMP3	CMP3IE	CMP3C
		CMP4	CMP4IE	CMP4C
hrtim_fault_it	系统故障	SYSFLT	SYSFLTIE	SYSFLTC
	故障 1 到 5	FLT1	FLT1IE	FLT1C
		FLT2	FLT2IE	FLT2C
		FLT3	FLT3IE	FLT3C
		FLT4	FLT4IE	FLT4C
		FLT5	FLT5IE	FLT5C

37.3.21 DMA

大部分能够生成中断的事件也可以生成 DMA 请求，甚至可以同时生成中断和 DMA 请求。每个定时器（主定时器、TIMA...E）都有自己的 DMA 使能寄存器。

各 DMA 请求会在进行或运算后发送到 6 条通道中，具体如下：

- 主定时器 1 条通道
- 每个定时单元 1 条通道

注： *禁止 DMA 通道之前（TIMxDIER 中的 DMA 使能位复位），需要先禁止 DMA 控制器。*

表 305 总结了事件及其关联的 DMA 使能位。

表 305. HRTIM DMA 请求汇总

DMA 通道	事件	支持 DMA	DMA 使能位
hrtim_dma1（主定时器）	突发模式周期结束	无	N/A
	主定时器寄存器更新	有	MUPDDE
	接收到同步事件	有	SYNCDE
	主定时器重复事件	有	MREPDE
	主定时器比较 1 到 4 事件	有	MCMP1DE
		有	MCMP2DE
		有	MCMP3DE
		有	MCMP4DE
hrtim_dma2（定时器 A） hrtim_dma3（定时器 B） hrtim_dma4（定时器 C） hrtim_dma5（定时器 D） hrtim_dma6（定时器 E）	延迟保护已触发	有	DLYPRTDE
	计数器复位或翻转事件	有	RSTDE
	输出 1 和输出 2 复位（从有效电平跳变为无效电平）	有	RSTx1DE
		有	RSTx2DE
	输出 1 和输出 2 置位（从无效电平跳变为有效电平）	有	SETx1DE
		有	SETx2DE
	捕获 1 和 2 事件	有	CPT1DE
		有	CPT2DE
	定时单元寄存器更新	有	UPDDE
	重复事件	有	REPDE
	比较 1 到 4 事件	有	CMP1DE
		有	CMP2DE
		有	CMP3DE
		有	CMP4DE
N/A	系统故障	无	N/A
	故障 1 到 5	无	N/A
	突发模式周期结束	无	N/A

突发 DMA 传输

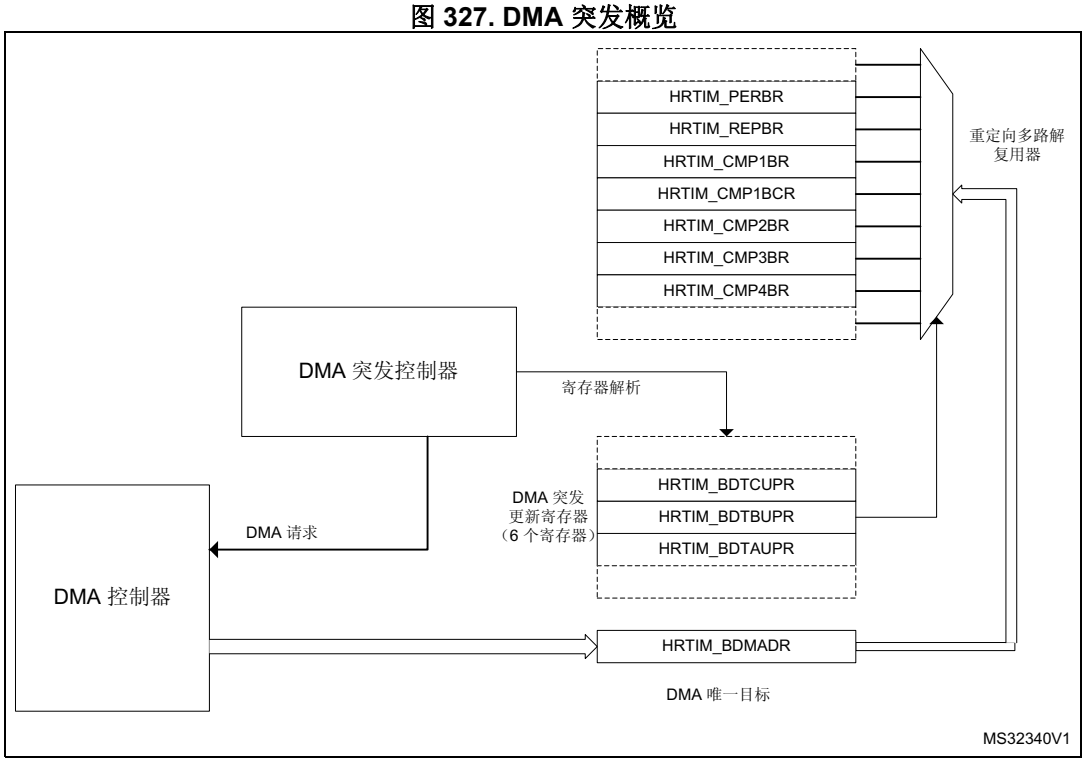
除了标准 DMA 请求之外，HRTIM 还配有 DMA 突发控制器，可通过单个 DMA 请求更新多个寄存器。具体包括：

- 仅通过一条 DMA 通道更新多个数据寄存器，
- 如果转换器使用多路定时器输出，可动态地对一个或多个定时单元重新进行编程。

突发 DMA 功能仅可用于一条 DMA 通道，但可选择 6 条通道中的任意一条进行突发 DMA 传输。

通过 DMA 写入，编程那些寄存器是核心内容。主定时器和 TIMA..E 包含突发 DMA 更新寄存器，其中的大部分控制和数据寄存器都关联到选择位：HRTIM_BDMUPR、HRTIM_BDTAUPR 到 HRTIM_BDTEUPR（这一点仅适用于寄存器的写访问）。重定向机制允许自动将 DMA 写访问转发到 HRTIM 寄存器，如*图 327* 所示。

图 327. DMA 突发概览



发生 DMA 触发时，HRTIM 会生成多个 32 位 DMA 请求，并会解析更新寄存器。如果控制位置 1，写访问会重定向到关联的寄存器。如果此位复位，则会跳过寄存器更新并恢复寄存器解析，直至检测到此位再次置 1 触发新请求。对 6 个更新寄存器（HRTIM_BDMUPR、5x HRTIM_BDTxUPR）进行解析后，突发结束，系统准备好处理另一 DMA 触发（请参见*图 328* 上的流程图）。

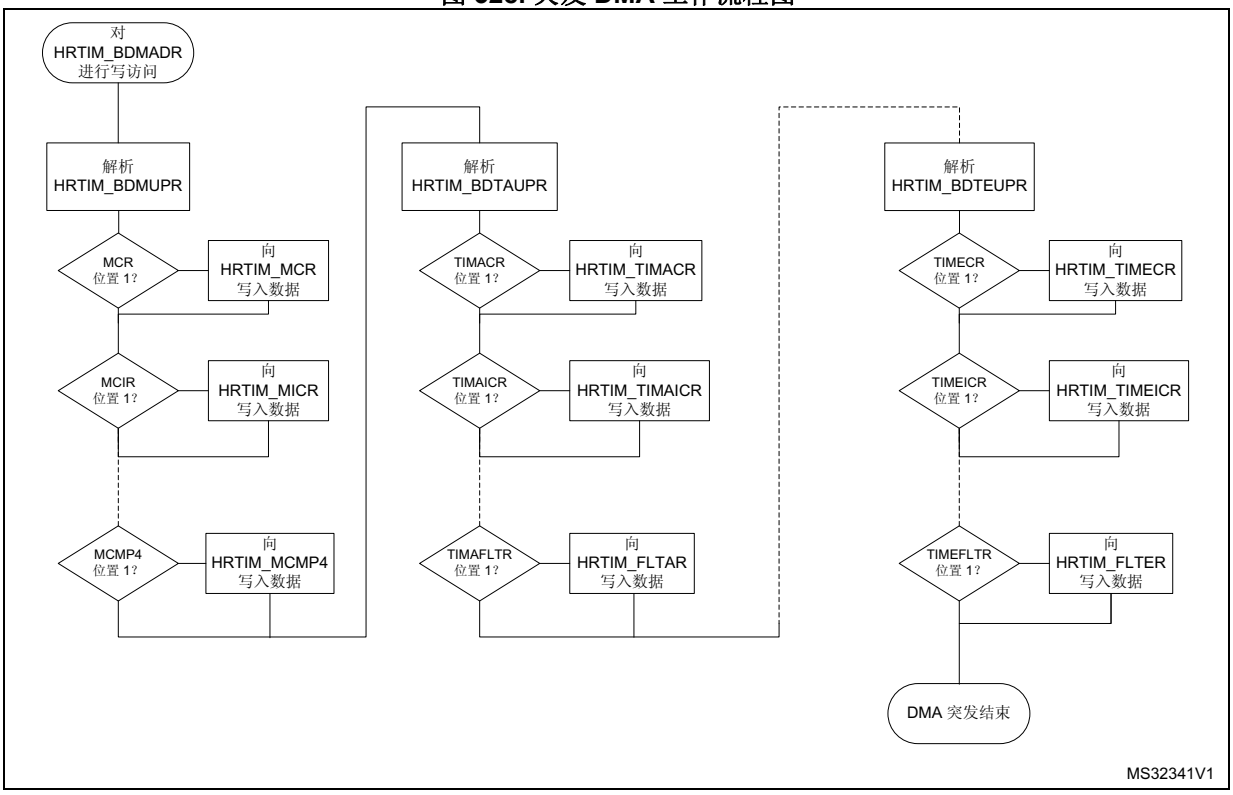
注： *任何在突发正在进行时发生的触发都会被丢弃，但最后一次数据传输期间发生的触发除外。*

突发 DMA 模式永久使能（没有使能位）。突发 DMA 工作是通过对 HRTIM_BDMADR 寄存器进行的第一次写访问启动的。

仅需要使 DMA 控制器指向作为目标的 HRTIM_BDMADR 寄存器，在存储器中指向禁止了外设递增模式的外设配置（HRTIM 会在内部处理到最终目标寄存器的数据重发）。

如果突发 DMA 模式在事务进行过程中中断，要重新初始化突发 DMA 模式，至少需要写入 6 个更新寄存器中的其中一个。

图 328. 突发 DMA 工作流程图



DMA 突发结束后，有多种选项可供使用，具体视寄存器更新策略而定。

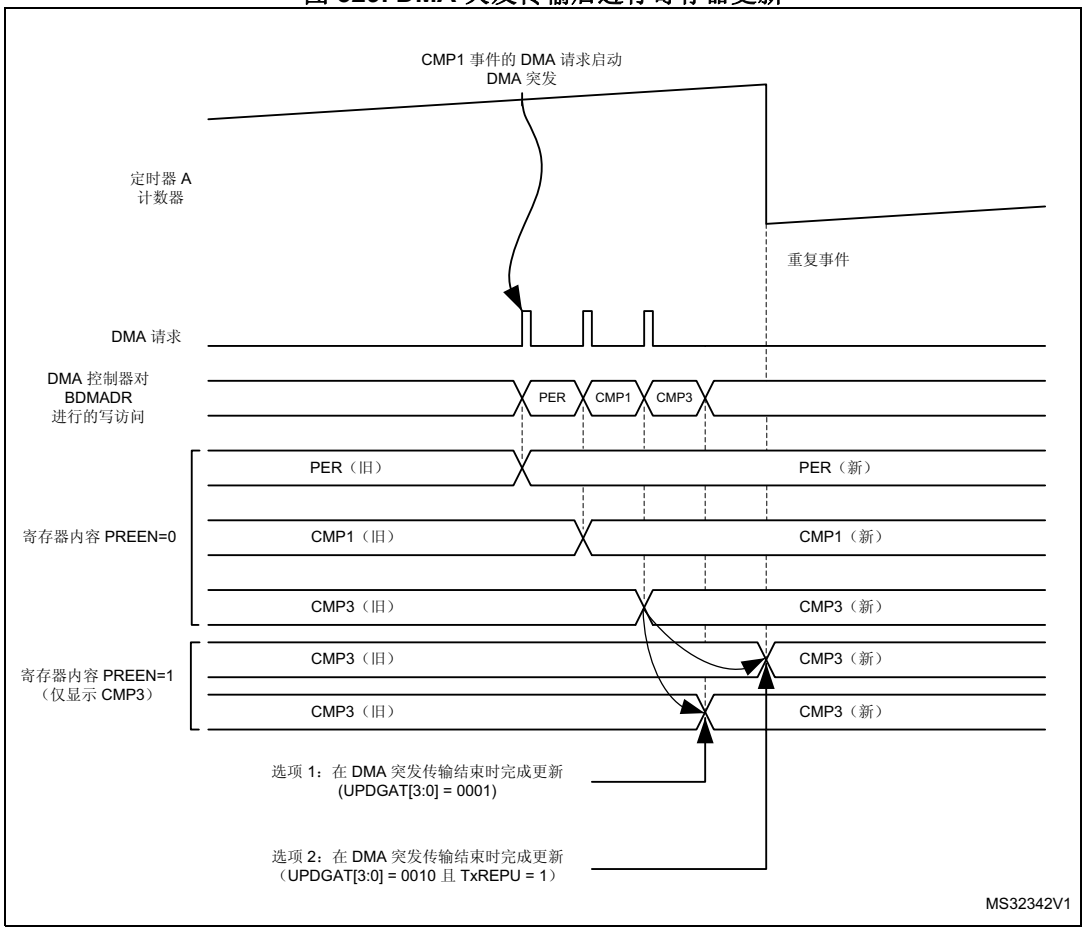
如果 PREEN 位复位（禁止预装载），通过 DMA 写入的值会立即传输到活动寄存器中，并会按照 DMA 事务的速度连续更新寄存器。

预装载使能时（PREEN 位置 1），有 3 个用例：

1. 更新独立于 DMA 突发传输进行（HRTIM_TIMxCR 中的 UPDGAT[3:0] = 0000，HRTIM_MCR 中的 BRSTDMA[1:0] = 00）。在这种情况下，如果需要同时考虑所有传输的数据，用户必须检查 DMA 突发是否在发生更新事件之前结束。相反，如果更新事件在 DMA 传输进行时发生，则仅会装载部分寄存器，完整的寄存器更新将需要 2 个连续的更新事件。
2. 更新在 DMA 突发传输完成后进行（HRTIM_TIMxCR 中的 UPDGAT[3:0] = 0000，HRTIM_MCR 中的 BRSTDMA[1:0] = 01）。此模式可确保同时传输所有新寄存器值。此操作独立于计数器值进行，如有必要，可与常规更新事件相结合（例如，TxRSTU 置 1 的情况下，在发生计数器复位事件时进行更新）。
3. 更新在 DMA 突发传输完成后发生更新事件时进行（HRTIM_TIMxCR 中的 UPDGAT[3:0] = 0010，HRTIM_MCR 中的 BRSTDMA[1:0] = 10）。此模式可确保对所有已传输数据、以及与常规更新事件的同步、与定时器计数器的同步进行一致的更新。在这种情况下，如果正在进行传输时发生常规更新请求，该请求会被丢弃，并会在下一次发出更新请求时进行有效更新。

图 329 上的时序图显示了 3 种情况下的活动寄存器内容：PREEN=0、UPDGAT[3:0] = 0001 以及 UPDGAT[3:0] = 0001（PREEN = 1 时）。

图 329. DMA 突发传输后进行寄存器更新



37.3.22 HRTIM 初始化

本节介绍了建议的 HRTIM 初始化程序，包括其他相关 MCU 外设。

必须在复位和时钟控制单元 (RCC) 中使能 HRTIM 时钟源。

HRTIM 控制寄存器可按照电源转换器拓扑和定时单元用例进行初始化。必须配置所有输入（源、极性、边沿有效性）。

最后必须设置 HRTIM 输出，设置顺序如下：

- 必须使用 HRTIM_OUTxR 中的 POLx 位定义极性
- 必须使用 HRTIM_OUTxR 中的 FAULTx[1:0] 和 IDLESx 位配置 FAULT 和 IDLE 状态

HRTIM 输出已准备好连接到 MCU I/O。在 GPIO 控制器中，必须按照产品数据手册中的复用功能映射表对所选 HRTIM I/O 进行配置。

从此刻起，HRTIM 会控制处于 IDLE 状态的输出。

要将输出配置为 RUN 模式，应将 HRTIM_OENR 寄存器中的 TxyOEN 位置 1。RUN 模式下发生第一个有效置位/复位事件之前，2 路输出都处于无效状态。只要 TxCEN 位复位，任何输出置位/复位事件（使用 SST、SRT 的软件请求除外）都会被忽略，突发模式请求也是如此（会忽略 IDLEM 位值）。同样，任何来自突发模式控制器的计数器复位请求也会被忽略（如果 TxBM 位置 1）。

注： *如果使能了死区插入（DTEN 位置 1），则需要通过软件强制设置输出状态（使用 SST 和 RST 位），使输出在进入 RUN 模式后立即处于互补状态。*

最终可通过将 HRTIM_MCR 中的 TxCEN 或 MCEN 位置 1 的方式启动 HRTIM 工作。

如果 HRTIM 外设通过复位和时钟控制器复位，HRTIM 输出会进入 IDLE 模式且为低电平。建议先断开 HRTIMER 与输出的连接（使用 GPIO 控制器），然后再执行外设复位。

37.3.23 调试

当微控制器进入调试模式（Cortex®-M7 内核停止）时，TIMx 计数器会根据 DBG 模块中的 DBG_HRTIM_STOP 配置位选择继续正常工作或者停止工作。

- DBG_HRTIM_STOP = 0：无行为变化，HRTIM 继续工作。
- DBG_HRTIM_STOP = 1：所有 HRTIM 定时器（包括主定时器）均停止工作。
  如果 FAULTx[1:0] = 01、10、11，RUN 模式下的输出会进入 FAULT 状态，如果 FAULTx[1:0] = 00，输出会保持当前状态。处于空闲状态的输出会保持此状态。即使 MCU 退出停止模式，也会永久保持此状态。这样可在执行步进期间保持安全状态。可通过将 TxyOEN 位置 1 再次使能输出（需要使用调试器）。

MCU 停止工作期间的定时器行为（DBG_HRTIM_STOP = 1 时）

置位/复位纵横开关、死区和推挽单元、空闲/均衡故障检测以及在 RUN 模式下驱动正常输出的所有逻辑均不受调试的影响。输出将在内部维持栓所状态，以便在 TxyOEN 再次置 1 时（MCU 停止工作期间或停止工作之后）重新获取输出的常规信号。如果输出已禁止，关联的触发信号和滤波器也会跟随内部波形。

MCU 停止工作期间，FAULT 输入和事件（任何源）会使能。

如果此时发生故障，则在 MCU 停止工作期间，故障状态位会置 1，TxyOEN 位会复位（TxyOEN 和 TxyODS 不受 DBG_HRTIM_STOP 位状态的影响）。

在调试模式下，会丢弃同步、计数器复位、启动、复位-启动事件以及捕获事件。这样做是为了确保所有相关寄存器在 MCU 停止期间的稳定性。

计数器到达断点时会停止计数。但计数器使能信号不会置位；因此退出调试模式时不会发出启动事件。所有计数器复位和捕获触发都会被禁用，外部事件也会被禁用（只要 MCU 停止工作，就会忽略这类事件）。输出 SET 和 RST 标志会冻结，但强制软件置位/复位的情况除外。调试过程中会屏蔽电平有效事件，但退出调试模式时，电平有效事件会立即再次激活。对于边沿有效事件，如果 MCU 停止工作期间信号保持有效，退出调试模式时不会生成新边沿。

更新事件会被丢弃。这样可避免触发对 hrtim_upd_en[3:1] 输入的更新。DMA 触发会被禁止。突发模式电路冻结：触发会被忽略，突发模式计数器停止工作。

37.4 应用用例

37.4.1 降压转换器

降压转换器通常用于降压。HRTIM 最多能够以 6 个独立的开关频率控制 10 个降压转换器。

转换器通常以固定频率工作，Vin/Vout 比取决于应用到电源开关的占空比 D：

$$V_{out} = D \times V_{in}$$

图 330 中显示的拓扑连接到 ADC 来读取电压。

图 330. 降压转换器拓扑

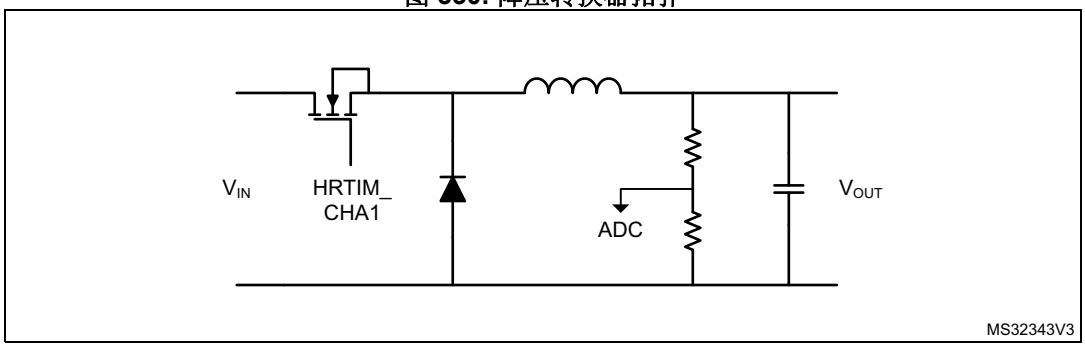
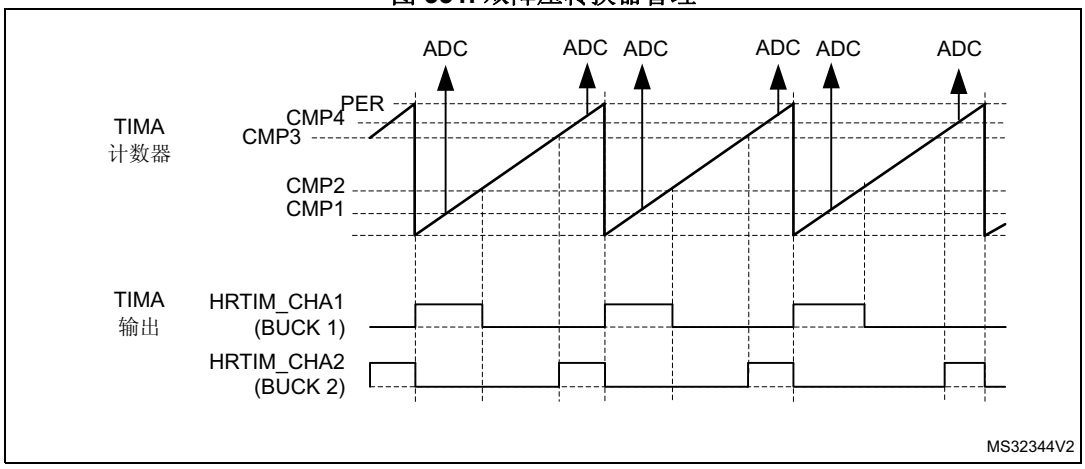


图 331 显示了如何利用相同频率的 PWM 信号来管理两个转换器。输出的定义如下：

- HRTIM_CHA1 在周期开始时置位，在发生 CMP1 时复位
- HRTIM_CHA2 在发生 CMP3 时置位，在发生 PER 时复位

ADC 每个周期会触发两次，分别通过 CMP2 和 CMP4 事件在 ON 时间的中点精确触发。

图 331. 双降压转换器管理

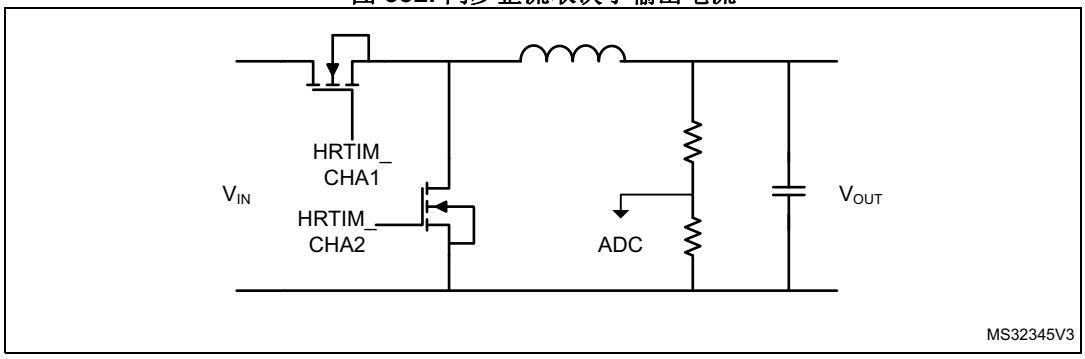


定时器 A..E 提供 10 个成对耦合的降压转换器（每对转换器的开关频率均相同）或 6 个完全独立的转换器（每个转换器都有不同的开关频率），第二种情况使用主定时器作为第 6 个时基。

37.4.2 具有同步整流功能的降压转换器

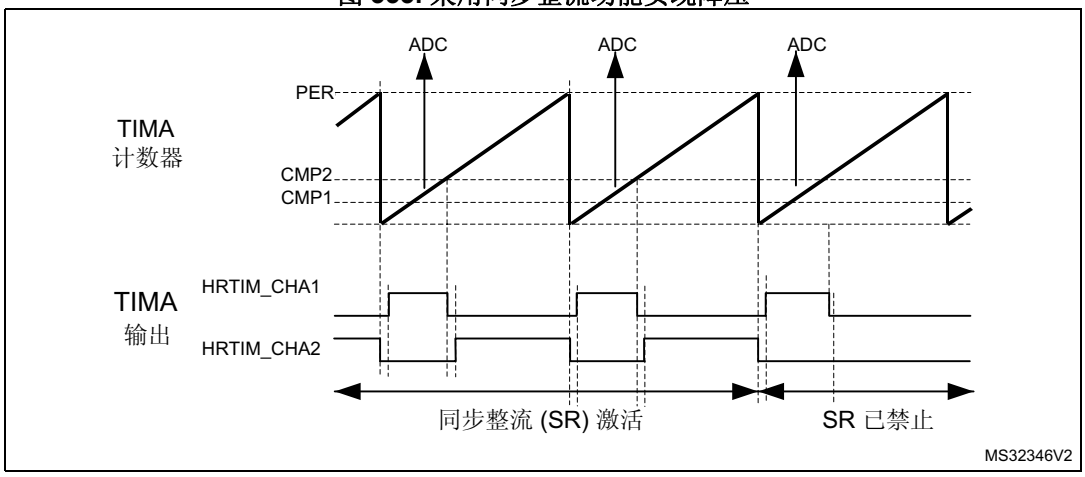
同步整流使用 FET 代替续流二极管，从而最大限度地降低降压转换器中的损耗。同步整流可实时开启或关闭，具体视输出电流水平而定（如*图 332* 所示）。

图 332. 同步整流取决于输出电流



与单开关降压转换器的主要区别是增加了死区，从而可根据 HRTIM_CHA1 上的参考波形在 HRTIM_CHA2 上生成几乎互补的波形（请参见*图 333*）。

图 333. 采用同步整流功能实现降压



37.4.3 多相转换器

多相技术可应用于多种电源转换拓扑（降压、反激）。此项技术的主要优势为：

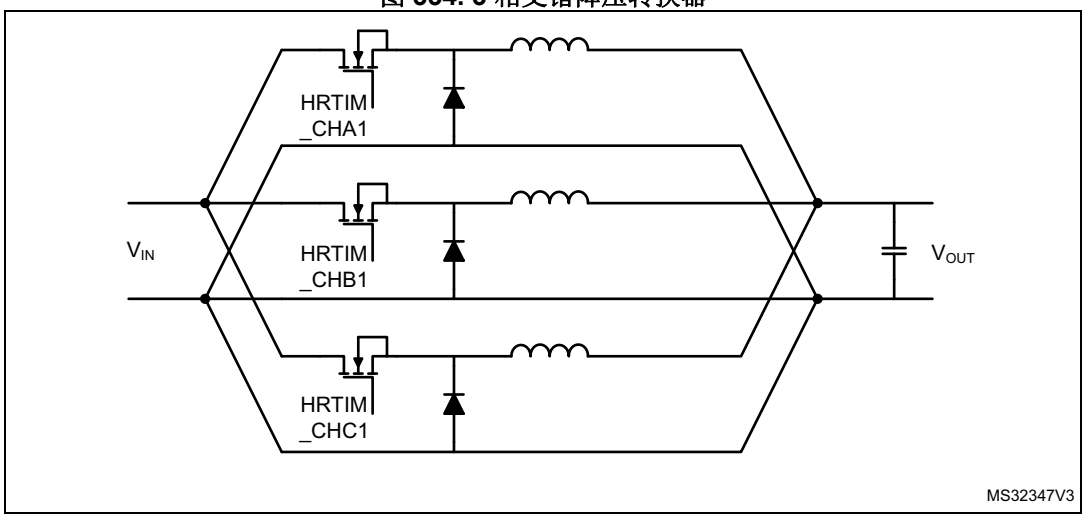
- 减少输入和输出电容的电流波纹
- 减少 EMI
- 通过动态更改相位数（相位分离）提高轻载条件下的效率

HRTIM 能够管理多个转换器。可控制的转换器数量取决于所使用的拓扑和资源（包括 ADC 触发信号）：

- 5 个具有同步整流 (SR) 功能的降压转换器，使用主定时器和 5 个定时器
- 4 个降压转换器（无 SR 功能），使用主定时器和 2 个定时器
- ...

图 335 显示了 3 相交错降压转换器的拓扑。

图 334. 3 相交错降压转换器



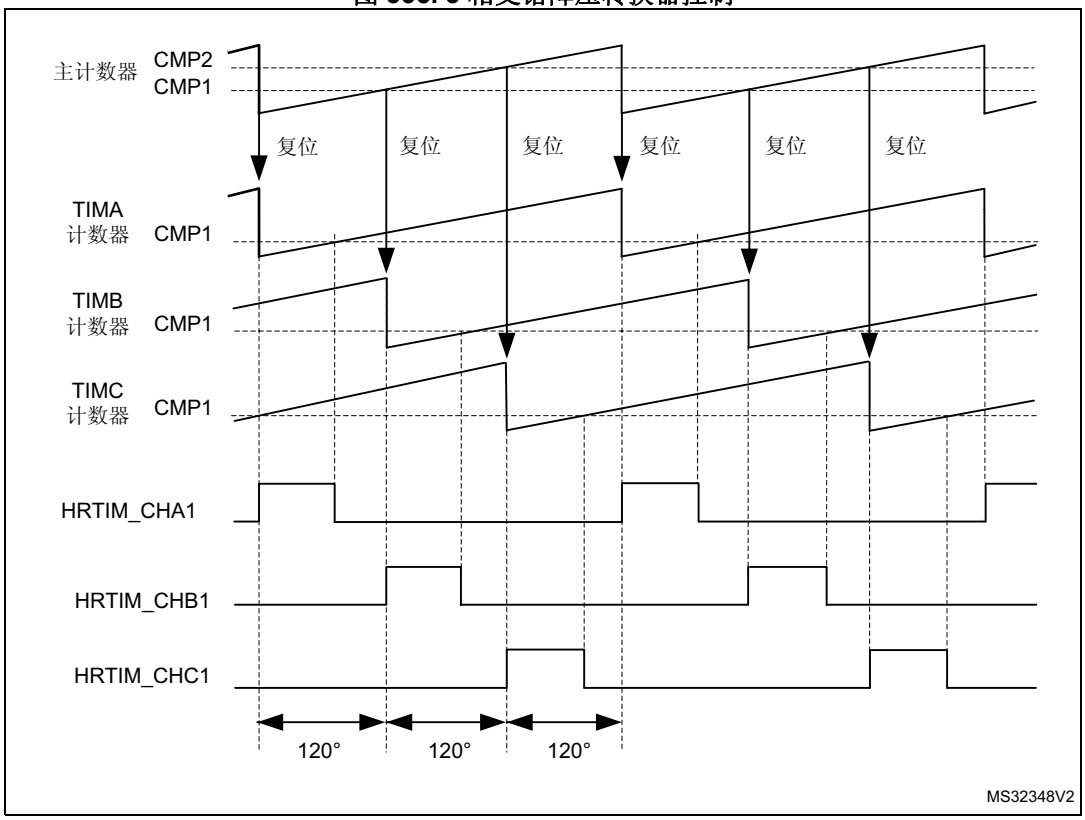
主定时器负责进行相位管理：它通过定期复位定时器来定义转换器之间的相位关系。相移为 360° 除以相位数，所给示例中的相移为 120°。

占空比随后会编程到每个定时器中。输出的定义如下：

- HRTIM_CHA1 在主定时器周期开始时置位，在发生 TACMP1 时复位
- HRTIM_CHB1 在发生主定时器 MCMP1 时置位，在发生 TBCMP1 时复位
- HRTIM_CHC1 在发生主定时器 MCMP2 时置位，在发生 TCCMP1 时复位

发生 TxCMP2 比较事件时会生成 ADC 触发信号。由于所有 ADC 触发源的相位均由于转换器拓扑而发生了偏移，因此可以将所有触发源合并为单个 ADC 触发源，从而节省 ADC 资源（例如将 1 个 ADC 常规通道用于整个多相转换器）。

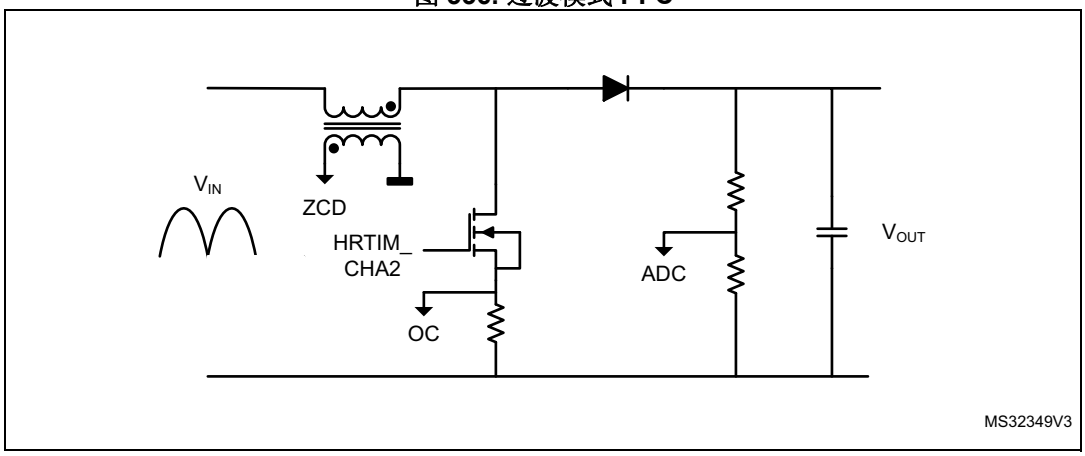
图 335. 3 相交错降压转换器控制



37.4.4 过渡模式功率因数校正

基本工作原理是，在固定的 Ton 时间内建立进入电感的电流。该电流将随后在 Toff 时间内消退，当电流为零时重新开始此循环周期。可通过过零检查电路 (ZCD) 检测此现象，如*图 336* 所示。由于有恒定 Ton 时间，电感中的电流峰值与整流的交流输入电压成比例，这提供了功率因数校正。

图 336. 过渡模式 PFC



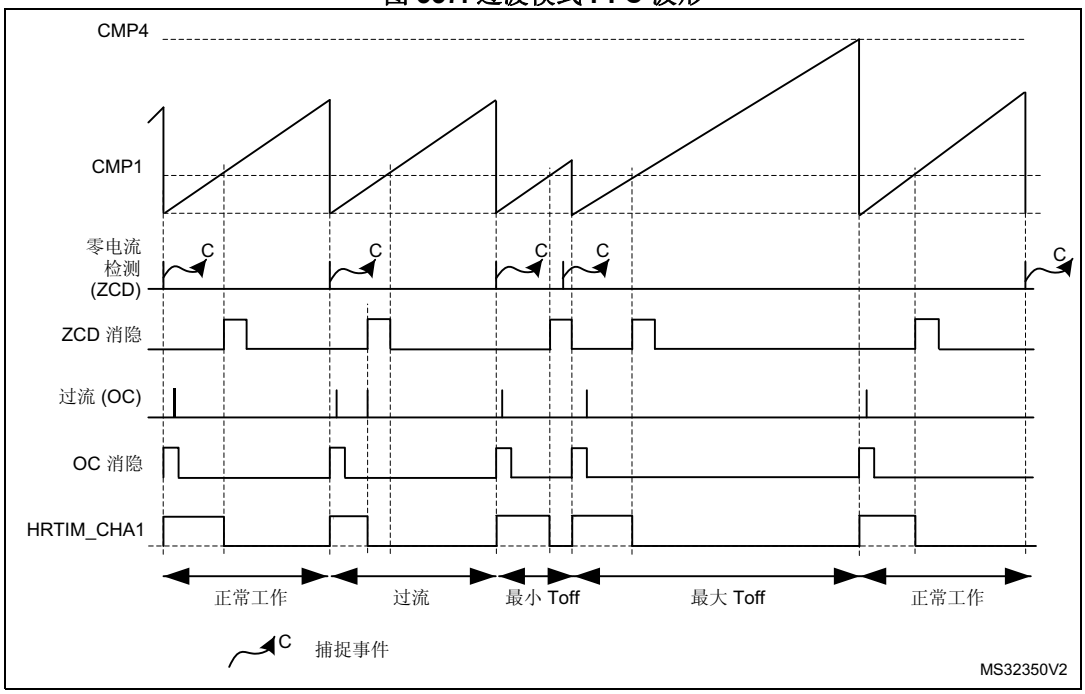
此转换器具有恒定的 Ton 时间，由于 Toff 时间变化其频率也是变化的（取决于输入电压）。它还必须包括一些功能，以便在未检测到过零电压时工作，或在过流 (OC) 时限制 Ton 时间。OC 反馈通常由内置比较器调节，发送到外部事件输入。

图 337 显示了各工作模式中的波形，其参数定义如下：

- Ton Min：屏蔽伪过流（续流二极管恢复电流），用 OC 消隐表示。
- Ton Max：实际上是转换器置位点。它由 CMP1 定义。
- Toff Min：在电流限值接近零时（退磁非常快）限制频率。它由 CMP2 定义。
- Toff Max：避免在无ZCD发生时系统卡死。自动延迟模式下它由 CMP4 定义。

由于值必须与输出下降沿相关，因此两个 Toff 值都会自动延迟。

图 337. 过渡模式 PFC 波形



37.5 HRTIM 寄存器

37.5.1 HRTIM 主定时器控制寄存器 (HRTIM_MCR)

HRTIM Master Timer Control Register

偏移地址：0x0000h

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
BRSTDMA[1:0]		MREPU	Res.	PREEN	DACSYNC[1:0]		Res.	Res.	Res.	TECEN	TDCEN	TCCEN	TBCEN	TACEN	MCEN
rw	rw	rw		rw	rw	rw				rw	rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SYNCSRC[1:0]		SYNCOUT[1:0]		SYNCSTRTM	SYNCRSTM	SYNCIN[1:0]		Res.	Res.	HALF	RETRIG	CONT	CKPSC[2:0]		
rw	rw	rw	rw	rw	rw	rw	rw			rw	rw	rw	rw	rw	rw

位 31:30 **BRSTDMA[1:0]**：*突发 DMA 更新 (Burst DMA Update)*

这些位定义更新与突发 DMA 事务的相对关系。

00：独立于 DMA 突发传输进行更新。

01：DMA 突发传输结束时进行更新。

10：在 DMA 突发传输结束后的主定时器翻转时进行更新。此模式仅适用于连续模式。

11：保留

位 29 **MREPU**：*主定时器重复更新 (Master Timer Repetition update)*

此位定义主定时器重复周期结束后是否进行更新（因翻转或复位事件而更新）。

仅当 BRSTDMA[1:0] = 00 或 01 时，才可将 MREPU 置 1。

0：禁止与重复有关的更新

1：使能与重复有关的更新

位 28 保留，必须保持复位值

位 27 **PREEN**：*预装载使能 (Preload enable)*

此位可使能寄存器预装载机制，并定义对存储器映射寄存器的写访问是在 HRTIM 的活动寄存器中进行还是在预装载寄存器中进行。

0：禁止预装载：直接对活动寄存器进行写访问

1：使能预装载：对预装载寄存器进行写访问

位 26:25 **DACSYNC[1:0]**：*DAC 同步 (DAC Synchronization)*

发生主定时器更新事件时，可使能并生成 DAC 同步事件。这些位会定义在哪路输出上发送 DAC 同步事件（有关连接的详细信息，请参见*第 37.3.19 节：DAC 触发*）。

00：未生成 DAC 触发事件

01：在 hrtim_dac_trg1 上生成触发事件

10：在 hrtim_dac_trg2 上生成触发事件

11：在 hrtim_dac_trg3 上生成触发事件

位 24:22 保留，必须保持复位值

位 21 **TECEN**：*定时器 E 计数器使能 (Timer E counter enable)*

此位用于启动定时器 E 计数器。

0：禁止定时器 E 计数器

1：使能定时器 E 计数器

注：　此位至少在 8 个 f_{HRTIM} 时钟周期内不得更改。

位 20 **TDCEN**：*定时器 D 计数器使能 (Timer D counter enable)*

此位用于启动定时器 D 计数器。

0：禁止定时器 D 计数器

1：使能定时器 D 计数器

注： 此位至少在 8 个 f_{HRTIM} 时钟周期内不得更改。

位 19 **TCCEN**：*定时器 C 计数器使能 (Timer C counter enable)*

此位用于启动定时器 C 计数器。

0：禁止定时器 C 计数器

1：使能定时器 C 计数器

注： 此位至少在 8 个 f_{HRTIM} 时钟周期内不得更改。

位 18 **TBCEN**：*定时器 B 计数器使能 (Timer B counter enable)*

此位用于启动定时器 B 计数器。

0：禁止定时器 B 计数器

1：使能定时器 B 计数器

注： 此位至少在 8 个 f_{HRTIM} 时钟周期内不得更改。

位 17 **TACEN**：*定时器 A 计数器使能 (Timer A counter enable)*

此位用于启动定时器 A 计数器。

0：禁止定时器 A 计数器

1：使能定时器 A 计数器

注： 此位至少在 8 个 f_{HRTIM} 时钟周期内不得更改。

位 16 **MCEN**：*主定时器计数器使能 (Master timer counter enable)*

此位用于启动主定时器计数器。

0：禁止主定时器计数器

1：使能主定时器计数器

注： 此位至少在 8 个 f_{HRTIM} 时钟周期内不得更改。

位 15:14 **SYNCSRC[1:0]**：*同步源 (Synchronization source)*

这些位用于定义将在同步输出 SYNCOUT[2:1] 上发送的源和事件。

00：主定时器启动

01：主定时器比较 1 事件

10：定时器 A 启动/复位

11：定时器比较 1 事件

位 13:12 **SYNCOUT[1:0]**：*同步输出 (Synchronization output)*

这些位用于定义同步输出事件的路由和调节。

00：禁用

01：保留

10：HRTIM_SCOUT 输出上的正脉冲（16 个 f_{HRTIM} 时钟周期）

11：HRTIM_SCOUT 输出上的负脉冲（16 个 f_{HRTIM} 时钟周期）

注： 计数器使能后（TxCEN 位置 1），不得修改该位域。

位 11 **SYNCSTRTM**：*主定时器同步启动 (Synchronization Starts Master)*

该位能够使主定时器在接收到同步输入事件时启动：

0：对主定时器无影响

1：同步输入事件会启动主定时器

位 10 **SYNCRSTM**：*主定时器同步复位 (Synchronization Resets Master)*

该位能够使主定时器在接收到同步输入事件时复位：

0：对主定时器无影响

1：同步输入事件会复位主定时器

位 9:8 **SYNCIN[1:0]**：*同步输入 (Synchronization input)*

这些位定义同步输入源。

00：禁用 HRTIM 未同步，并在独立模式下运行。

01：保留。

10：内部事件，HRTIM 与片上定时器同步（请参见*同步输入*）。

11：外部事件（输入引脚）。HRTIM_SCIN 输入上的正脉冲会触发 HRTIM。

注： 受影响的定时器使能后，不能更改此参数。

位 7:6 保留，必须保持复位值

位 5 **HALF**：*半占空比模式 (Half mode)*

此位用于使能半占空比模式：HRTIM_MPER 寄存器写入内容后，HRTIM_MCMP1xR 活动寄存器会自动更新为 HRTIM_MPER/2 值。

0：禁止半占空比模式

1：使能半占空比模式

位 4 **RETRIG**：*可再触发模式 (Re-triggerable mode)*

此位定义主定时器计数器在单发模式下的行为。

0：定时器不可再触发：仅当计数器已停止时（周期已过），才能进行计数器复位。

1：定时器可再次触发：无论计数器处于何种状态（运行状态或停止状态），均会进行计数器复位。

位 3 **CONT**：*连续模式 (Continuous mode)*

0：定时器在单发模式下工作，达到 MPER 值时会停止工作。

1：定时器在连续（自由运行）模式下工作，达到 MPER 值时会翻转为零。

位 2:0 **CKPSC[2:0]**：*时钟预分频器 (Clock prescaler)*

这些位定义主定时器时钟预分频比。

计数器时钟等效频率 ($f_{COUNTER}$) 等于 $f_{HRCK} / 2^{(CKPSC[2:0]-5)}$。

定时器使能后，不能修改预分频比。

000：保留

001：保留

010：保留

011：保留

100：保留

101：$f_{COUNTER} = f_{HRTIM}$

110：$f_{COUNTER} = f_{HRTIM} / 2$

111：$f_{COUNTER} = f_{HRTIM} / 4$

37.5.2 HRTIM 主定时器中断状态寄存器 (HRTIM_MISR)

HRTIM Master Timer Interrupt Status Register

偏移地址：0x0004h

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MUPD	SYNC	MREP	MCMP4	MCMP3	MCMP2	MCMP1
									r	r	r	r	r	r	r

位 31:7 保留，必须保持复位值

位 6 **MUPD**：主定时器更新中断标志 (Master Update Interrupt Flag)
更新主定时器寄存器时，该位由硬件置 1。
0：未发生主定时器更新中断
1：已发生主定时器更新中断

位 5 **SYNC**：同步输入中断标志 (Sync Input Interrupt Flag)
当接收到同步输入事件时，此位由硬件置 1。
0：未发生同步输入中断
1：已发生同步输入中断

位 4 **MREP**：主定时器重复中断标志 (Master Repetition Interrupt Flag)
主定时器重复周期已过时，该位由硬件置 1。
0：未发生主定时器重复中断
1：已发生主定时器重复中断

位 3 **MCMP4**：主定时器比较 4 中断标志 (Master Compare 4 Interrupt Flag)
请参见 MCMP1 说明

位 2 **MCMP3**：主定时器比较 3 中断标志 (Master Compare 3 Interrupt Flag)
请参见 MCMP1 说明

位 1 **MCMP2**：主定时器比较 2 中断标志 (Master Compare 2 Interrupt Flag)
请参见 MCMP1 说明

位 0 **MCMP1**：主定时器比较 1 中断标志 (Master Compare 1 Interrupt Flag)
当主定时器计数器值与在主定时器比较 1 寄存器中编程的值相匹配时，此位由硬件置 1。
0：未发生主定时器比较 1 中断
1：已发生主定时器比较 1 中断

37.5.3 HRTIM 主定时器中断清零寄存器 (HRTIM_MICR)

HRTIM Master Timer Interrupt Clear Register

偏移地址：0x0008h

复位值：0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MUPDC	SYNCC	MREPC	MCMP4C	MCMP3C	MCMP2C	MCMP1C
									w	w	w	w	w	w	w

位 31:7 保留，必须保持复位值

位 6 **MUPDC**：主定时器更新中断标志清零 (Master update Interrupt flag clear)
将 1 写入此位时，HRTIM_MISR 寄存器中的 MUPDC 标志将清零

位 5 **SYNCC**：同步输入中断标志清零 (Sync Input Interrupt flag clear)
将 1 写入此位时，HRTIM_MISR 寄存器中的 SYNC 标志将清零

位 4 **MREPC**：重复中断标志清零 (Repetition Interrupt flag clear)
将 1 写入此位时，HRTIM_MISR 寄存器中的 MREP 标志将清零

位 3 **MCMP4C**：主定时器比较 4 中断标志清零 (Master Compare 4 Interrupt flag clear)
将 1 写入此位时，HRTIM_MISR 寄存器中的 MCMP4 标志将清零

位 2 **MCMP3C**：主定时器比较 3 中断标志清零 (Master Compare 3 Interrupt flag clear)
将 1 写入此位时，HRTIM_MISR 寄存器中的 MCMP3 标志将清零

位 1 **MCMP2C**：主定时器比较 2 中断标志清零 (Master Compare 2 Interrupt flag clear)
将 1 写入此位时，HRTIM_MISR 寄存器中的 MCMP2 标志将清零

位 0 **MCMP1C**：主定时器比较 1 中断标志清零 (Master Compare 1 Interrupt flag clear)
将 1 写入此位时，HRTIM_MISR 寄存器中的 MCMP1 标志将清零

37.5.4 HRTIM 主定时器 DMA/中断使能寄存器 (HRTIM_MDIER)

HRTIM Master Timer DMA / Interrupt Enable Register

偏移地址：0x000Ch

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MUPDDE	SYNCDE	MREPDE	MCMP4DE	MCMP3DE	MCMP2DE	MCMP1DE
									rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MUPDIE	SYNCIE	MREPIE	MCMP4IE	MCMP3IE	MCMP2IE	MCMP1IE
									rw	rw	rw	rw	rw	rw	rw

位 31:23 保留，必须保持复位值

位 22 **MUPDDE**：主定时器更新 DMA 请求使能 (Master Update DMA request Enable)
此位由软件置 1 和清零，用于使能/禁止主定时器更新 DMA 请求。
0：禁止主定时器更新 DMA 请求
1：使能主定时器更新 DMA 请求

位 21 **SYNCDE**：同步输入 DMA 请求使能 (Sync Input DMA request Enable)
此位由软件置 1 和清零，用于使能/禁止同步输入 DMA 请求。
0：禁止同步输入 DMA 请求
1：使能同步输入 DMA 请求

位 20 **MREPDE**：主定时器重复 DMA 请求使能 (Master Repetition DMA request Enable)
此位由软件置 1 和清零，用于使能/禁止主定时器重复 DMA 请求。
0：禁止重复 DMA 请求
1：使能重复 DMA 请求

位 19 **MCMP4DE**：主定时器比较 4 DMA 请求使能 (Master Compare 4 DMA request Enable)
请参见 MCMP1DE 说明

位 18 **MCMP3DE**：主定时器比较 3 DMA 请求使能 (Master Compare 3 DMA request Enable)
请参见 MCMP1DE 说明

位 17 **MCMP2DE**：主定时器比较 2 DMA 请求使能 (Master Compare 2 DMA request Enable)
请参见 MCMP1DE 说明

位 16 **MCMP1DE**：主定时器比较 1 DMA 请求使能 (Master Compare 1 DMA request Enable)
此位由软件置 1 和清零，用于使能/禁止主定时器比较 1 DMA 请求。
0：禁止比较 1 DMA 请求
1：使能比较 1 DMA 请求

位 15:6 保留，必须保持复位值

位 6 **MUPDIE**：主定时器更新中断使能 (Master Update Interrupt Enable)
此位由软件置 1 和清零，用于使能/禁止主定时器寄存器更新中断。
0：禁止主定时器更新中断
1：使能主定时器更新中断

位 5 **SYNCIE**：同步输入中断使能 (Sync Input Interrupt Enable)
此位由软件置 1 和清零，用于使能/禁止同步输入中断
0：禁止同步输入中断
1：使能同步输入中断

位 4 **MREPIE**：主定时器重复中断使能 (Master Repetition Interrupt Enable)
此位由软件置 1 和清零，用于使能/禁止主定时器寄存器重复中断
0：禁止主定时器重复中断
1：使能主定时器重复中断

位 3 **MCMP4IE**：主定时器比较 4 中断使能 (Master Compare 4 Interrupt Enable)
请参见 MCMP1IE 说明

位 2 **MCMP3IE**：主定时器比较 3 中断使能 (Master Compare 3 Interrupt Enable)
请参见 MCMP1IE 说明

位 1 **MCMP2IE**：主定时器比较 2 中断使能 (Master Compare 2 Interrupt Enable)
请参见 MCMP1IE 说明

位 0 **MCMP1IE**：主定时器比较 1 中断使能 (Master Compare 1 Interrupt Enable)
此位由软件置 1 和清零，用于使能/禁止主定时器比较 1 中断
0：禁止比较 1 中断
1：使能比较 1 中断

37.5.5 HRTIM 主定时器计数器寄存器 (HRTIM_MCNTR)

HRTIM Master Timer Counter Register

偏移地址：0x0010h

复位值：0x0000 0000

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

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

位 31:16 保留，必须保持复位值

位 15:0 **MCNT[15:0]**：*计数器值 (Counter value)*

保存主定时器计数器值。仅当主定时器已停止时（HRTIM_MCR 中的 MCEN = 0）时，才能写入此寄存器。

注： 如果设置的计数器值大于 HRTIM_MPER 寄存器值，则不保证定时器的行为。

37.5.6 HRTIM 主定时器周期寄存器 (HRTIM_MPER)

HRTIM Master Timer Period Register

偏移地址：0x0014h

复位值：0x0000 FFDF

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

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

位 31:16 保留，必须保持复位值

位 15:0 **MPER[15:0]**：*主定时器周期值 (Master Timer Period value)*

此定时器定义计数器溢出值。

周期值必须大于或等于 3 个 f_{HRTIM} 时钟周期。

最大值为 0x0000 FFDF。

37.5.7 HRTIM 主定时器重复寄存器 (HRTIM_MREP)

HRTIM Master Timer Repetition Register

偏移地址：0x0018h

复位值：0x0000 0000

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

位 31:8 保留，必须保持复位值

位 7:0 **MREP[7:0]**：*主定时器重复周期值 (Master Timer Repetition period value)*

此寄存器保存主定时器计数器的重复周期值。该寄存器为预装载寄存器，禁止预装载时则为活动寄存器。

37.5.8 HRTIM 主定时器比较 1 寄存器 (HRTIM_MCMP1R)

HRTIM Master Timer Compare 1 Register

偏移地址：0x001Ch

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MCMP1[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:16 保留，必须保持复位值

位 15:0 **MCMP1[15:0]**：*主定时器比较 1 值 (Master Timer Compare 1 value)*

该寄存器保存主定时器比较 1 值。该寄存器为预装载寄存器，禁止预装载时则为活动寄存器。比较值必须大于或等于 3 个 f_{HRTIM} 时钟周期。

37.5.9 HRTIM 主定时器比较 2 寄存器 (HRTIM_MCMP2R)

HRTIM Master Timer Compare 2 Register

偏移地址：0x0024h

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MCMP2[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:16 保留，必须保持复位值

位 15:0 **MCMP2[15:0]**：*主定时器比较 2 值 (Master Timer Compare 2 value)*

该寄存器保存主定时器比较 2 值。该寄存器为预装载寄存器，禁止预装载时则为活动寄存器。比较值必须大于或等于 3 个 f_{HRTIM} 时钟周期。

37.5.10 HRTIM 主定时器比较 3 寄存器 (HRTIM_MCMP3R)

HRTIM Master Timer Compare 3 Register

偏移地址：0x0028h

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MCMP3[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:16 保留，必须保持复位值

位 15:0 **MCMP3[15:0]**：*主定时器比较 3 值 (Master Timer Compare 3 value)*

该寄存器保存主定时器比较 3 值。该寄存器为预装载寄存器，禁止预装载时则为活动寄存器。比较值必须大于或等于 3 个 f_{HRTIM} 时钟周期。

37.5.11 HRTIM 主定时器比较 4 寄存器 (HRTIM_MCMP4R)

HRTIM Master Timer Compare 4 Register

偏移地址：0x002Ch

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MCMP4[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:16 保留，必须保持复位值

位 15:0 **MCMP4[15:0]**：*主定时器比较 4 值 (Master Timer Compare 4 value)*

该寄存器保存主定时器比较 4 值。该寄存器为预装载寄存器，禁止预装载时则为活动寄存器。比较值必须大于或等于 3 个 f_{HRTIM} 时钟周期。

37.5.12 HRTIM Timerx 控制寄存器 (HRTIM_TIMxCR)

HRTIM Timerx Control Register

偏移地址：0x0000h（该偏移地址与定时器 x 基址相关）

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
UPDGAT[3:0]				PREEN	DACSYNC[1:0]		MSTU	TEU	TDU	TCU	TBU	Res.	TxRSTU	TxREPU	Res.
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw		rw	rw	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DELCMP4[1:0]		DELCMP2[1:0]		SYNCSTRTx	SYNCRSTx	Res.	Res.	Res.	PSHPLL	HALF	RETRIG	CONT	CKPSCx[2:0]		
rw	rw	rw	rw	rw	rw				rw	rw	rw	rw	rw	rw	rw

位 31:28 **UPDGAT[3:0]**：*更新门控 (Update Gating)*

这些位定义更新与突发 DMA 事务以及更新使能输入 1 到 3 上的外部更新请求之间的关系（请参见 *表 293：更新使能输入和源*）

如下文所述，更新事件可以是：MSTU、TEU、TDU、TCU、TBU、TAU、TxRSTU、TxREPU。

0000：独立于 DMA 突发传输进行更新

0001：DMA 突发传输结束时进行更新

0010：DMA 突发传输结束后发生更新事件时进行更新

0011：在 HRTIM 更新使能输入 1 (hrtim_upd_en1) 的上升沿进行更新

0100：在 HRTIM 更新使能输入 2 (hrtim_upd_en2) 的上升沿进行更新

0101：在 HRTIM 更新使能输入 3 (hrtim_upd_en3) 的上升沿进行更新

0110：在 HRTIM 更新使能输入 1 (hrtim_upd_en1) 的上升沿后发生更新事件时进行更新

0111：在 HRTIM 更新使能输入 2 (hrtim_upd_en2) 的上升沿后发生更新事件时进行更新

1000：在 HRTIM 更新使能输入 3 (hrtim_upd_en3) 的上升沿后发生更新事件时进行更新

其他代码：保留

注： *编程新值之前，该位域必须复位。*

如果 UPDGAT[3:0] 值等于 0001、0011、0100、0101，可以有多个并发更新源（例如 RSTU 和 DMA 突发）。

位 27 **PREEN**：*预装载使能 (Preload enable)*

此位可使能寄存器预装载机制，并定义对可预装载寄存器的写访问是在活动寄存器中进行还是在预装载寄存器中进行。

0：禁止预装载：直接对活动寄存器进行写访问

1：使能预装载：对预装载寄存器进行写访问

位 26:25 **DACSYNC[1:0]**：*DAC 同步 (DAC Synchronization)*

发生定时器更新时，会生成 DAC 同步事件。这些位会定义在哪路输出上发送 DAC 同步事件（有关连接的详细信息，请参见 *第 37.3.19 节：DAC 触发*）。

00：未生成 DAC 触发事件

01：在 hrtim_dac_trg1 上生成触发事件

10：在 hrtim_dac_trg2 上生成触发事件

11：在 hrtim_dac_trg3 上生成触发事件

位 24 **MSTU**：*主定时器更新 (Master Timer update)*

寄存器更新由主定时器更新触发

0：禁止通过主定时器触发更新

1：使能通过主定时器触发更新

位 23 在 HRTIM_TIMACR、HRTIM_TIMBCR、HRTIM_TIMCCR、HRTIM_TIMDCR 中：

TEU：*定时器 E 更新 (Timer E update)*

寄存器更新由定时器 E 更新触发

0：禁止通过定时器 E 触发更新

1：使能通过定时器 E 触发更新

在 HRTIM_TIMECR 中：

保留，必须保持复位值

位 22 在 HRTIM_TIMACR、HRTIM_TIMBCR、HRTIM_TIMCCR、HRTIM_TIMECR 中：

TDU：*定时器 D 更新 (Timer D update)*

寄存器更新由定时器 D 更新触发

0：禁止通过定时器 D 触发更新

1：使能通过定时器 D 触发更新

在 HRTIM_TIMDCR 中：

保留，必须保持复位值

位 21 在 HRTIM_TIMACR、HRTIM_TIMBCR、HRTIM_TIMDCR、HRTIM_TIMECR 中：

TCU：*定时器 C 更新 (Timer C update)*

寄存器更新由定时器 C 更新触发

0：禁止通过定时器 C 触发更新

1：使能通过定时器 C 触发更新

在 HRTIM_TIMCCR 中：

保留，必须保持复位值

位 20 在 HRTIM_TIMACR、HRTIM_TIMCCR、HRTIM_TIMDCR、HRTIM_TIMECR 中：

TBU：*定时器 B 更新 (Timer B update)*

寄存器更新由定时器 B 更新触发

0：禁止通过定时器 B 触发更新

1：使能通过定时器 B 触发更新

在 HRTIM_TIMBCR 中：

保留，必须保持复位值

位 19 在 HRTIM_TIMBCR、HRTIM_TIMCCR、HRTIM_TIMDCR、HRTIM_TIMECR 中：

TAU：*定时器 A 更新 (Timer A update)*

寄存器更新由定时器 A 更新触发

0：禁止通过定时器 A 触发更新

1：使能通过定时器 A 触发更新

在 HRTIM_TIMACR 中：

保留，必须保持复位值

位 18 **TxRSTU**：*Timerx 复位更新 (Timerx reset update)*

在连续模式下达到周期值后，由 Timerx 计数器复位或翻转为 0 触发寄存器更新。

0：禁止通过定时器 x 复位/翻转触发更新

1：使能通过定时器 x 复位/翻转触发更新

位 17 **TxREPU**：*定时器 x 重复更新 (Timer x Repetition update)*

计数器翻转且 HRTIM_REPx = 0 时触发寄存器更新

0：禁止与重复有关的更新

1：使能与重复有关的更新

位 16 保留，必须保持复位值

位 15:14 **DELCMP4[1:0]**：*CMP4 自动延迟模式 (CMP4 auto-delayed mode)*

该位域定义比较寄存器在标准模式下运行（计数器比较值相等后立即发出比较匹配事件）还是在自动延迟模式下运行（请参见*自动延迟模式*）。

00：CMP4 寄存器始终有效（标准比较模式）

01：会重新计算 CMP4 值，且 CMP4 值在捕获 2 事件后生效

10：会重新计算 CMP4 值，且 CMP4 值在捕获 2 事件后生效；或者会重新计算 CMP4 值，且 CMP4 值在比较 1 匹配后生效（如果缺少捕获 2 事件，则会执行超时功能）

11：会重新计算 CMP4 值，且 CMP4 值在捕获事件后生效；或者会重新计算 CMP4 值，且 CMP4 值在比较 3 匹配后生效（如果缺少捕获事件，则会执行超时功能）

注： 计数器使能后（TxCEN 位置 1），不得修改该位域。

位 13:12 **DELCMP2[1:0]**：*CMP2 自动延迟模式 (CMP2 auto-delayed mode)*

该位域定义比较寄存器在标准模式下运行（计数器比较值相等后立即发出比较匹配事件）还是在自动延迟模式下运行（请参见*自动延迟模式*）。

00：CMP2 寄存器始终有效（标准比较模式）

01：会重新计算 CMP2 值，且 CMP2 值在捕获 1 事件后生效

10：会重新计算 CMP2 值，且 CMP2 值在捕获 1 事件后生效；或者会重新计算 CMP2 值，且 CMP2 值在比较 1 匹配后生效（如果缺少捕获事件，则会执行超时功能）

11：会重新计算 CMP2 值，且 CMP2 值在捕获 1 事件后生效；或者会重新计算 CMP2 值，且 CMP2 值在比较 3 匹配后生效（如果缺少捕获事件，则会执行超时功能）

注： 计数器使能后（TxCEN 位置 1），不得修改该位域。

位 11 **SYNCSTRTx**：*同步启动定时器 x（Synchronization Starts Timer x）*

此位定义定时器 x 在同步事件之后的行为：

0：对定时器 x 无影响

1：同步输入事件会启动定时器 x

位 10 **SYNCRSTx**：*同步复位定时器 x (Synchronization Resets Timer x)*

此位定义定时器 x 在同步事件之后的行为：

0：对定时器 x 无影响

1：同步输入事件会复位定时器 x

位 9:7 保留，必须保持复位值

位 6 **PSHPLL**：*推挽模式使能 (Push-Pull mode enable)*

此位用于使能推挽模式。

0：禁止推挽模式

1：使能推挽模式

注： 计数器使能后（TxCEN 位置 1），不得修改该位域。

位 5 **HALF**：*半占空比模式使能 (Half mode enable)*

此位用于使能半占空比模式：HRTIM_PERxR 寄存器写入内容后，HRTIM_CMP1xR 活动寄存器会自动更新为 HRTIM_PERxR/2 值。

0：禁止半占空比模式

1：使能半占空比模式

位 4 **RETRIG**：*可再触发模式 (Re-triggerable mode)*

此位定义计数器在单发模式下的行为。

0：定时器不可再次触发：如果计数器停止计数（单发模式下计数周期已过，或者连续模式下计数器停止计数），则会进行计数器复位。

1：定时器可再次触发：无论计数器处于何种状态，均会进行计数器复位。

位 3 **CONT**：*连续模式 (Continuous mode)*

此位定义定时器的工作模式。

0：定时器在单发模式下工作，达到 TIMxPER 值时会停止工作。

1：定时器在连续模式下工作，达到 TIMxPER 值时会翻转为零。

位 2:0 **CKPSCx[2:0]**：*HRTIM 定时器 x 时钟预分频器 (HRTIM Timer x Clock prescaler)*

这些位定义主定时器时钟预分频比。

计数器时钟等效频率 ($f_{COUNTER}$) 等于 $f_{HRCK} / 2^{(CKPSC[2:0]-5)}$。

定时器使能后，不能修改预分频比。

000：保留

001：保留

010：保留

011：保留

100：保留

101：$f_{COUNTER} = f_{HRTIM}$

110：$f_{COUNTER} = f_{HRTIM} / 2$

111：$f_{COUNTER} = f_{HRTIM} / 4$

37.5.13 HRTIM Timerx 中断状态寄存器 (HRTIM_TIMxISR)

HRTIM Timerx Interrupt Status Register

偏移地址：0x0004h（该偏移地址与定时器 x 基址相关）

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	O2CPY	O1CPY	O2STAT	O1STAT	IPPSTAT	CPPSTAT
										r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	DLYPRT	RST	RSTx2	SETx2	RSTx1	SETx1	CPT2	CPT1	UPD	Res.	REP	CMP4	CMP3	CMP2	CMP1
	r	r	r	r	r	r	r	r	r		r	r	r	r	r

位 31:22 保留，必须保持复位值

位 21 **O2CPY**：输出 2 副本 (Output 2 Copy)

此状态位是输出级之前输出 2 状态的原始副本（斩波、极性），允许在延迟保护后重新使能输出之前检查当前输出状态。

0：输出 2 无效

1：输出 2 有效

位 20 **O1CPY**：输出 1 副本 (Output 1 Copy)

此状态位是输出级之前输出 1 状态的原始副本（斩波、极性），允许在延迟保护后重新使能输出之前检查当前输出状态。

0：输出 1 无效

1：输出 1 有效

位 19 **O2STAT**：输出 2 状态 (Output 2 Status)

此状态位指示延迟空闲保护触发时的输出 2 状态。进入任何新的延迟保护时，会更新此位。均衡空闲模式下，不会更新此位。

0：输出 2 无效

1：输出 2 有效

位 18 **O1STAT**：输出 1 状态 (Output 1 Status)

此状态位指示延迟空闲保护触发时的输出 1 状态。进入任何新的延迟保护时，会更新此位。均衡空闲模式下，不会更新此位。

0：输出 1 无效

1：输出 1 有效

位 17 **IPPSTAT**：空闲推挽状态 (Idle Push Pull Status)

此状态位指示触发保护时，在推挽模式、均衡故障模式或延迟空闲模式下将信号应用到哪路输出（不考虑输出状态有效还是无效）。

0：输出 1 有效时进行保护，输出 2 强制无效

1：输出 2 有效时进行保护，输出 1 强制无效

位 16 **CPPSTAT**：当前推挽状态 (Current Push Pull Status)

此状态位指示推挽模式下当前将信号应用到哪路输出上。此位仅在该配置中有意义。

0：信号应用到输出 1 上，输出 2 强制无效

1：信号应用到输出 2 上，输出 1 强制无效

位 15 保留

位 14 **DLYPRT**：延迟保护标志 (Delayed Protection Flag)

此位指示延迟空闲或均衡空闲模式进入。

位 13 **RST**：复位和/或翻转中断标志 (Reset and/or roll-over Interrupt Flag)
定时器 x 在连续模式下复位或翻转时，此位由硬件置 1。
0：未发生 TIMx 计数器复位/翻转中断
1：发生 TIMx 计数器复位/翻转中断

位 12 **RSTx2**：输出 2 复位中断标志 (Output 2 Reset Interrupt Flag)
请参见 RSTx1 说明

位 11 **SETx2**：输出 2 置位中断标志 (Output 2 Set Interrupt Flag)
请参见 SETx1 说明

位 10 **RSTx1**：输出 1 复位中断标志 (Output 1 Reset Interrupt Flag)
当 Tx1 输出复位时（从有效模式变为无效模式），此位由硬件置 1。
0：未发生 Tx1 输出复位中断
1：发生 Tx1 输出复位中断

位 9 **SETx1**：输出 1 置位中断标志 (Output 1 Set Interrupt Flag)
当 Tx1 输出置位时（从无效模式变为有效模式），此位由硬件置 1。
0：未发生 Tx1 输出置位中断
1：发生 Tx1 输出置位中断

位 8 **CPT2**：捕获 2 中断标志 (Capture2 Interrupt Flag)
请参见 CPT1 说明

位 7 **CPT1**：捕获 1 中断标志 (Capture1 Interrupt Flag)
发生定时器 x 捕获 1 事件时，该位由硬件置 1。
0：未发生定时器 x 捕获 1 复位中断
1：发生定时器 x 捕获 1 复位中断

位 6 **UPD**：更新中断标志 (Update Interrupt Flag)
发生定时器 x 更新事件时，该位由硬件置 1。
0：未发生定时器 x 更新中断
1：发生定时器 x 更新中断

位 5 保留，必须保持复位值

位 4 **REP**：重复中断标志 (Repetition Interrupt Flag)
定时器 x 重复周期已过时，该位由硬件置 1。
0：未发生定时器 x 重复中断
1：发生定时器 x 重复中断

位 3 **CMP4**：比较 4 中断标志 (Compare 4 Interrupt Flag)
请参见 CMP1 说明

位 2 **CMP3**：比较 3 中断标志 (Compare 3 Interrupt Flag)
请参见 CMP1 说明

位 1 **CMP2**：比较 2 中断标志 (Compare 2 Interrupt Flag)
请参见 CMP1 说明

位 0 **CMP1**：比较 1 中断标志 (Compare 1 Interrupt Flag)
当定时器 x 计数器值与在比较 1 寄存器中编程的值相匹配时，此位由硬件置 1。
0：未发生比较 1 中断
1：发生比较 1 中断

37.5.14 HRTIM Timerx 中断清零寄存器 (HRTIM_TIMxICR)

HRTIM Timerx Interrupt Clear Register

偏移地址：0x0008h（该偏移地址与定时器 x 基址相关）

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	DLYPRTC	RSTC	RSTx2C	SET2xC	RSTx1C	SET1xC	CPT2C	CPT1C	UPDC	Res.	REPC	CMP4C	CMP3C	CMP2C	CMP1C
	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w

位 31:15 保留，必须保持复位值

位 14 **DLYPRTC**：延迟保护标志清零 (Delayed Protection Flag Clear)
将 1 写入此位时，HRTIM_TIMxISR 寄存器中的 DLYPRT 标志将清零

位 13 **RSTC**：复位中断标志清零 (Reset Interrupt flag Clear)
将 1 写入此位时，HRTIM_TIMxISR 寄存器中的 RST 标志将清零

位 12 **RSTx2C**：输出 2 复位标志清零 (Output 2 Reset flag Clear)
将 1 写入此位时，HRTIM_TIMxISR 寄存器中的 RSTx2 标志将清零

位 11 **SETx2C**：输出 2 置位标志清零 (Output 2 Set flag Clear)
将 1 写入此位时，HRTIM_TIMxISR 寄存器中的 SETx2 标志将清零

位 10 **RSTx1C**：输出 1 复位标志清零 (Output 1 Reset flag Clear)
将 1 写入此位时，HRTIM_TIMxISR 寄存器中的 RSTx1 标志将清零

位 9 **SETx1C**：输出 1 置位标志清零 (Output 1 Set flag Clear)
将 1 写入此位时，HRTIM_TIMxISR 寄存器中的 SETx1 标志将清零

位 8 **CPT2C**：捕获 2 中断标志清零 (Capture 2 Interrupt flag Clear)
将 1 写入此位时，HRTIM_TIMxISR 寄存器中的 CPT2 标志将清零

位 7 **CPT1C**：捕获 1 中断标志清零 (Capture 1 Interrupt flag Clear)
将 1 写入此位时，HRTIM_TIMxISR 寄存器中的 CPT1 标志将清零

位 6 **UPDC**：更新中断标志清零 (Update Interrupt flag Clear)
将 1 写入此位时，HRTIM_TIMxISR 寄存器中的 UPD 标志将清零

位 5 保留，必须保持复位值

位 4 **REPC**：重复中断标志清零 (Repetition Interrupt flag Clear)
将 1 写入此位时，HRTIM_TIMxISR 寄存器中的 REP 标志将清零

位 3 **CMP4C**：比较 4 中断标志清零 (Compare 4 Interrupt flag Clear)
将 1 写入此位时，HRTIM_TIMxISR 寄存器中的 CMP4 标志将清零

位 2 **CMP3C**：比较 3 中断标志清零 (Compare 3 Interrupt flag Clear)
将 1 写入此位时，HRTIM_TIMxISR 寄存器中的 CMP3 标志将清零

位 1 **CMP2C**：比较 2 中断标志清零 (Compare 2 Interrupt flag Clear)
将 1 写入此位时，HRTIM_TIMxISR 寄存器中的 CMP2 标志将清零

位 0 **CMP1C**：比较 1 中断标志清零 (Compare 1 Interrupt flag Clear)
将 1 写入此位时，HRTIM_TIMxISR 寄存器中的 CMP1 标志将清零

37.5.15 HRTIM Timerx DMA/中断使能寄存器 (HRTIM_TIMxDIER)

HRTIM Timerx DMA / Interrupt Enable Register

偏移地址：0x000Ch（该偏移地址与定时器 x 基址相关）

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	DLYPRTDE	RSTDE	RSTx2DE	SETx2DE	RSTx1DE	SETx1DE	CPT2DE	CPT1DE	UPDDE	Res.	REPDE	CMP4DE	CMP3DE	CMP2DE	CMP1DE
	rw	rw	rw	rw	rw	rw	rw	rw	rw		rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	DLYPRTIE	RSTIE	RSTx2IE	SETx2IE	RSTx1IE	SET1xIE	CPT2IE	CPT1IE	UPDIE	Res.	REPIE	CMP4IE	CMP3IE	CMP2IE	CMP1IE
	rw	rw	rw	rw	rw	rw	rw	rw	rw		rw	rw	rw	rw	rw

位 31 保留

位 30 **DLYPRTDE**：延迟保护 DMA 请求使能 (Delayed Protection DMA request Enable)
此位由软件置 1 和清零，用于使能/禁止延迟保护 DMA 请求。
0：禁止延迟保护 DMA 请求
1：使能延迟保护 DMA 请求

位 29 **RSTDE**：复位/翻转 DMA 请求使能 (Reset/roll-over DMA request Enable)
此位由软件置 1 和清零，用于使能/禁止连续模式下定时器 x 计数器复位或翻转 DMA 请求。
0：禁止定时器 x 计数器复位/翻转 DMA 请求
1：使能定时器 x 计数器复位/翻转 DMA 请求

位 28 **RSTx2DE**：输出 2 复位 DMA 请求使能 (Output 2 Reset DMA request Enable)
请参见 RSTx1DE 说明

位 27 **SETx2DE**：输出 2 置位 DMA 请求使能 (Output 2 Set DMA request Enable)
请参见 SETx1DE 说明

位 26 **RSTx1DE**：输出 1 复位 DMA 请求使能 (Output 1 Reset DMA request Enable)
此位由软件置 1 和清零，用于使能/禁止 Tx1 输出复位 DMA 请求。
0：禁止 Tx1 输出复位 DMA 请求
1：使能 Tx1 输出复位 DMA 请求

位 25 **SETx1DE**：输出 1 置位 DMA 请求使能 (Output 1 Set DMA request Enable)
此位由软件置 1 和清零，用于使能/禁止 Tx1 输出置位 DMA 请求。
0：禁止 Tx1 输出置位 DMA 请求
1：使能 Tx1 输出置位 DMA 请求

位 24 **CPT2DE**：捕获 2 DMA 请求使能 (Capture 2 DMA request Enable)
请参见 CPT1DE 说明

位 23 **CPT1DE**：捕获 1 DMA 请求使能 (Capture 1 DMA request Enable)
此位由软件置 1 和清零，用于使能/禁止捕获 1 DMA 请求。
0：禁止捕获 1 DMA 请求
1：使能捕获 1 DMA 请求

位 22 **UPDDE**：更新 DMA 请求使能 (Update DMA request Enable)
此位由软件置 1 和清零，用于使能/禁止更新事件 DMA 请求。
0：禁止更新 DMA 请求
1：使能更新 DMA 请求

位 21 保留，必须保持复位值

位 20 **REPDE**：重复 DMA 请求使能 (Repetition DMA request Enable)
此位由软件置 1 和清零，用于使能/禁止重复事件 DMA 请求。
0：禁止重复 DMA 请求
1：使能重复 DMA 请求

位 19 **CMP4DE**：比较 4 DMA 请求使能 (Compare 4 DMA request Enable)
请参见 CMP1DE 说明

位 18 **CMP3DE**：比较 3 DMA 请求使能 (Compare 3 DMA request Enable)
请参见 CMP1DE 说明

位 17 **CMP2DE**：比较 2 DMA 请求使能 (Compare 2 DMA request Enable)
请参见 CMP1DE 说明

位 16 **CMP1DE**：比较 1 DMA 请求使能 (Compare 1 DMA request Enable)
此位由软件置 1 和清零，用于使能/禁止比较 1 DMA 请求。
0：禁止比较 1 DMA 请求
1：使能比较 1 DMA 请求

位 15 保留

位 14 **DLYPRTIE**：延迟保护中断使能 (Delayed Protection Interrupt Enable)
此位由软件置 1 和清零，用于使能/禁止延迟保护中断。
0：禁止延迟保护中断
1：使能延迟保护中断

位 13 **RSTIE**：复位/翻转中断使能 (Reset/roll-over Interrupt Enable)
此位由软件置 1 和清零，用于使能/禁止连续模式下定时器 x 计数器复位或翻转中断。
0：禁止定时器 x 计数器复位/翻转中断
1：使能定时器 x 计数器复位/翻转中断

位 12 **RSTx2IE**：输出 2 复位中断使能 (Output 2 Reset Interrupt Enable)
请参见 RSTx1IE 说明

位 11 **SETx2IE**：输出 2 置位中断使能 (Output 2 Set Interrupt Enable)
请参见 SETx1IE 说明

位 10 **RSTx1IE**：输出 1 复位中断使能 (Output 1 Reset Interrupt Enable)
此位由软件置 1 和清零，用于使能/禁止 Tx1 输出复位中断。
0：禁止 Tx1 输出复位中断
1：使能 Tx1 输出复位中断

位 9 **SETx1IE**：输出 1 置位中断使能 (Output 1 Set Interrupt Enable)
此位由软件置 1 和清零，用于使能/禁止 Tx1 输出置位中断。
0：禁止 Tx1 输出置位中断
1：使能 Tx1 输出置位中断

位 8 **CPT2IE**：捕获 2 中断使能 (Capture 2 Interrupt Enable)
请参见 CPT1IE 说明

位 7 **CPT1IE**：捕获 1 中断使能 (Capture 1 Interrupt Enable)
此位由软件置 1 和清零，用于使能/禁止捕获 1 中断。
0：禁止捕获 1 中断
1：使能捕获 1 中断

位 6 **UPDIE**：更新中断使能 (Update Interrupt Enable)
此位由软件置 1 和清零，用于使能/禁止更新事件中断。
0：禁止更新中断
1：使能更新中断

位 5 保留，必须保持复位值

位 4 **REPIE**：重复中断使能 (Repetition Interrupt Enable)
此位由软件置 1 和清零，用于使能/禁止重复事件中断。
0：禁止重复中断
1：使能重复中断

位 3 **CMP4IE**：比较 4 中断使能 (Compare 4 Interrupt Enable)
请参见 CMP1IE 说明

位 2 **CMP3IE**：比较 3 中断使能 (Compare 3 Interrupt Enable)
请参见 CMP1IE 说明

位 1 **CMP2IE**：比较 2 中断使能 (Compare 2 Interrupt Enable)
请参见 CMP1IE 说明

位 0 **CMP1IE**：比较 1 中断使能 (Compare 1 Interrupt Enable)
此位由软件置 1 和清零，用于使能/禁止比较 1 中断。
0：禁止比较 1 中断
1：使能比较 1 中断

37.5.16 HRTIM Timerx 计数器寄存器 (HRTIM_CNTxR)

HRTIM Timerx Counter Register

偏移地址：0x0010h（该偏移地址与定时器 x 基址相关）

复位值：0x0000 0000

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

位 31:16 保留，必须保持复位值

位 15:0 **CNTx[15:0]**：*Timerx 计数器值 (Timerx Counter value)*

该寄存器保存 Timerx 计数器值。仅当定时器已停止时（HRTIM_TIMxCR 中的 TxCEN = 0）时，才能写入此寄存器。

注： 如果计数器值大于 HRTIM_PERxR 寄存器值，则不保证定时器的行为。

37.5.17 HRTIM Timerx 周期寄存器 (HRTIM_PERxR)

HRTIM Timerx Period Register

偏移地址：0x14h（该偏移地址与定时器 x 基址相关）

复位值：0x0000 FFDF

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

位 31:16 保留，必须保持复位值

位 15:0 **PERx[15:0]**：*Timerx 周期值 (Timerx Period value)*

该寄存器保存定时器 x 周期值。

如果禁止预装载，该寄存器保存预装载寄存器的内容和活动寄存器的内容。

周期值必须大于或等于 3 个 f_{HRTIM} 时钟周期。

最大值为 0x0000 FFDF。

37.5.18 HRTIM Timerx 重复寄存器 (HRTIM_REPxR)

HRTIM Timerx Repetition Register

偏移地址：0x18h（该偏移地址与定时器 x 基址相关）

复位值：0x0000 0000

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

位 31:8 保留，必须保持复位值

位 7:0 **REPx[7:0]**：*Timerx 重复周期值 (Timerx Repetition period value)*

该寄存器保存重复周期值。

该寄存器保存预装载寄存器的内容，如果禁止预装载，则会保存活动寄存器的内容。

37.5.19 HRTIM Timerx 比较 1 寄存器 (HRTIM_CMP1xR)

HRTIM Timerx Compare 1 Register

偏移地址：0x1Ch（该偏移地址与定时器 x 基址相关）

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CMP1x[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:16 保留，必须保持复位值

位 15:0 **CMP1x[15:0]**：*Timerx 比较 1 值 (Timerx Compare 1 value)*

该寄存器保存比较 1 值。

该寄存器保存预装载寄存器的内容，如果禁止预装载，则会保存活动寄存器的内容。

比较值必须大于或等于 3 个 f_{HRTIM} 时钟周期。

37.5.20 HRTIM Timerx 比较 1 复合寄存器 (HRTIM_CMP1CxR)

HRTIM Timerx Compare 1 Compound Register

偏移地址：0x20h（该偏移地址与定时器 x 基址相关）

复位值：0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CMP1x[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:24 保留，必须保持复位值

位 23:16 **REPx[7:0]**：*Timerx 重复值 (Timerx Repetition value)（HRTIM_REPx 寄存器中的别名）*
此位域是 HRTIMx_REPxR 寄存器中 REPx[7:0] 位域的别名。

位 15:0 **CMP1x[15:0]**：*Timerx 比较 1 值 (Timerx Compare 1 value)*
此位域是 HRTIMx_CMP1xR 寄存器中 CMP1x[15:0] 位域的别名。

37.5.21 HRTIM Timerx 比较 2 寄存器 (HRTIM_CMP2xR)

HRTIM Timerx Compare 2 Register

偏移地址：0x24h（该偏移地址与定时器 x 基址相关）

复位值：0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CMP2x[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:16 保留，必须保持复位值

位 15:0 **CMP2x[15:0]**：*Timerx 比较 2 值 (Timerx Compare 2 value)*
该寄存器保存比较 2 值。
该寄存器保存预装载寄存器的内容，如果禁止预装载，则会保存活动寄存器的内容。
比较值必须大于或等于 3 个 f_{HRTIM} 时钟周期。
通过 HRTIM_TIMxCR 中的 DELCMP2[1:0] 使能后，该寄存器可起到自动延迟比较寄存器的作用。

37.5.22 HRTIM Timerx 比较 3 寄存器 (HRTIM_CMP3xR)

HRTIM Timerx Compare 3 Register

偏移地址：0x28h（该偏移地址与定时器 x 基址相关）

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CMP3x[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:16 保留，必须保持复位值

位 15:0 **CMP3x[15:0]**：*Timerx 比较 3 值 (Timerx Compare 3 value)*

该寄存器保存比较 3 值。

该寄存器保存预装载寄存器的内容，如果禁止预装载，则会保存活动寄存器的内容。

比较值必须大于或等于 3 个 f_{HRTIM} 时钟周期。

37.5.23 HRTIM Timerx 比较 4 寄存器 (HRTIM_CMP4xR)

HRTIM Timerx Compare 4 Register

偏移地址：0x2Ch（该偏移地址与定时器 x 基址相关）

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CMP4x[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:16 保留，必须保持复位值

位 15:0 **CMP4x[15:0]**：*Timerx 比较 4 值 (Timerx Compare 4 value)*

该寄存器保存比较 4 值。

该寄存器保存预装载寄存器的内容，如果禁止预装载，则会保存活动寄存器的内容。

比较值必须大于或等于 3 个 f_{HRTIM} 时钟周期。

通过 HRTIM_TIMxCR 中的 DELCMP4[1:0] 使能后，该寄存器可起到自动延迟比较寄存器的作用。

37.5.24 HRTIM Timerx 捕获 1 寄存器 (HRTIM_CPT1xR)

HRTIM Timerx Capture 1 Register

偏移地址：0x30h（该偏移地址与定时器 x 基址相关）

复位值：0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CPT1x[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:16 保留，必须保持复位值

位 15:0 **CPT1x[15:0]**：*Timerx 捕获 1 值 (Timerx Capture 1 value)*
发生捕获 1 事件时，该寄存器保存计数器值。

37.5.25 HRTIM Timerx 捕获 2 寄存器 (HRTIM_CPT2xR)

HRTIM Timerx Capture 2 Register

偏移地址：0x34h（该偏移地址与定时器 x 基址相关）

复位值：0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CPT2x[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:16 保留，必须保持复位值

位 15:0 **CPT2x[15:0]**：*Timerx 捕获 2 值 (Timerx Capture 2 value)*
发生捕获 2 事件时，该寄存器保存计数器值。

37.5.26 HRTIM Timerx 死区寄存器 (HRTIM_DTxR)

HRTIM Timerx Deadtime Register

偏移地址：0x38h（该偏移地址与定时器 x 基址相关）

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DTFLKx	DTFSLKx	Res.	Res.	Res.	Res.	SDTFx	DTFx[8:0]								
rwo	rwo					rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DTRLKx	DTRSLKx	Res.	DTPRSC[1:0]			SDTRx	DTRx[8:0]								
rwo	rwo		rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31 **DTFLKx**：*死区下降沿锁定 (Deadtime Falling Lock)*

若使能这一仅可写入一次的位，则会禁止修改死区（符号和值）。

0：可写入死区下降沿值和符号

1：只能读取死区下降沿值和符号

注： 此位不能预装载

位 30 **DTFSLKx**：*死区下降沿符号锁定 (Deadtime Falling Sign Lock)*

若使能这一仅可写入一次的位，则会禁止修改死区下降沿的符号。

0：可写入死区下降沿符号

1：只能读取死区下降沿符号

注： 此位不能预装载

位 29:26 保留，必须保持复位值

位 25 **SDTFx**：*死区下降沿符号值 (Sign Deadtime Falling value)*

该寄存器决定了死区为正值（信号不重叠）还是负值（信号重叠）。

0：死区在下降沿为正值

1：死区在下降沿为负值

位 24:16 **DTFx[8:0]**：*死区下降沿值 (Deadtime Falling value)*

该寄存器保存参考 PWM 信号下降沿之后的死区值。

t_{DTF} = DTFx[8:0] x t_{DTG}

位 15 **DTRLKx**：*死区上升沿锁定 (Deadtime Rising Lock)*

若使能这一仅可写入一次的位，则会禁止修改死区（符号和值）。

0：可写入死区上升沿值和符号

1：只能读取死区上升沿值和符号

注： 此位不能预装载

位 14 **DTRSLKx**：*死区上升沿符号锁定 (Deadtime Rising Sign Lock)*

若使能这一仅可写入一次的位，则会禁止修改死区上升沿的符号。

0：可写入死区上升沿符号

1：只能读取死区上升沿符号

注： 此位不能预装载

位 13 保留，必须保持复位值

位 12:10 **DTPRSC[2:0]**：*死区预分频器 (Deadtime Prescaler)*

该寄存器保存死区时钟预分频器的值。

$t_{DTG} = (2^{(DTPRSC[2:0]-3)}) \times t_{HRTIM}$

000：保留

001：保留

010：保留

011：tDTG= t_{HRTIM}

100：tDTG= t_{HRTIM} x 2

101：tDTG= t_{HRTIM} x 4

110：tDTG= t_{HRTIM} x 8

111：tDTG= t_{HRTIM} x 16

如果使能了任何锁定位（DTFLKs、DTFSLKx、DTRLKx、DTRSLKx），此位域会变为只读。

位 9 **SDTRx**：*死区上升沿符号值 (Sign Deadtime Rising value)*

该寄存器决定了死区为正值还是负值（信号重叠）。

0：死区在上升沿为正值

1：死区在上升沿为负值

位 8:0 **DTRx[8:0]**：*死区上升沿值 (Deadtime Rising value)*

该寄存器保存参考 PWM 信号上升沿之后的死区值。

$t_{DTR} = DTRx[8:0] \times t_{DTG}$

37.5.27 HRTIM Timerx 输出 1 置位寄存器 (HRTIM_SETx1R)

HRTIM Timerx Output1 Set Register

偏移地址：0x3Ch（该偏移地址与定时器 x 基址相关）

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
UPDATE	EXTEVNT10	EXTEVNT9	EXTEVNT8	EXTEVNT7	EXTEVNT6	EXTEVNT5	EXTEVNT4	EXTEVNT3	EXTEVNT2	EXTEVNT1	TIMEVNT9	TIMEVNT8	TIMEVNT7	TIMEVNT6	TIMEVNT5
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TIMEVNT4	TIMEVNT3	TIMEVNT2	TIMEVNT1	MSTCMP4	MSTCMP3	MSTCMP2	MSTCMP1	MSTPER	CMP4	CMP3	CMP2	CMP1	PER	RESYNC	SST
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31 **UPDATE**：*寄存器更新 (Registers update)（从预装载寄存器传输到活动寄存器）*
寄存器更新事件会强制将输出设为其有效状态。

位 30 **EXTEVNT10**：*外部事件 10 (External Event 10)*
请参见 EXTEVNT1 说明

位 29 **EXTEVNT9**：*外部事件 9 (External Event 9)*
请参见 EXTEVNT1 说明

位 28 **EXTEVNT8**：*外部事件 8 (External Event 8)*
请参见 EXTEVNT1 说明

位 27 **EXTEVNT7**：*外部事件 7 (External Event 7)*
请参见 EXTEVNT1 说明

位 26 **EXTEVNT6**：*外部事件 6 (External Event 6)*
请参见 EXTEVNT1 说明

位 25 **EXTEVNT5**：*外部事件 5 (External Event 5)*
请参见 EXTEVNT1 说明

位 24 **EXTEVNT4**：*外部事件 4 (External Event 4)*
请参见 EXTEVNT1 说明

位 23 **EXTEVNT3**：*外部事件 3 (External Event 3)*
请参见 EXTEVNT1 说明

位 22 **EXTEVNT2**：*外部事件 2 (External Event 2)*
请参见 EXTEVNT1 说明

位 21 **EXTEVNT1**：*外部事件 1 (External Event 1)*
外部事件 1 会强制将输出设为其有效状态。

位 20 **TIMEVNT9**：*定时器事件 9 (Timer Event 9)*
请参见 TIMEVNT1 说明

位 19 **TIMEVNT8**：*定时器事件 8 (Timer Event 8)*
请参见 TIMEVNT1 说明

位 18 **TIMEVNT7**：*定时器事件 7 (Timer Event 7)*
请参见 TIMEVNT1 说明

位 17 **TIMEVNT6**：*定时器事件 6 (Timer Event 6)*
请参见 TIMEVNT1 说明

位 16 **TIMEVNT5**：*定时器事件 5 (Timer Event 5)*
请参见 TIMEVNT1 说明

位 15 **TIMEVNT4**：*定时器事件 4 (Timer Event 4)*
请参见 TIMEVNT1 说明

位 14 **TIMEVNT3**：*定时器事件 3 (Timer Event 3)*
请参见 TIMEVNT1 说明

位 13 **TIMEVNT2**：*定时器事件 2 (Timer Event 2)*
请参见 TIMEVNT1 说明

位 12 **TIMEVNT1**：*定时器事件 1 (Timer Event 1)*
定时器事件 1 会强制将输出设为其有效状态（有关定时器事件分配，请参见*表 286*）

位 11 **MSTCMP4**：*主定时器比较 4 (Master Compare 4)*
主定时器比较 4 事件会强制将输出设为其有效状态。

位 10 **MSTCMP3**：*主定时器比较 3 (Master Compare 3)*
主定时器比较 3 事件会强制将输出设为其有效状态。

位 9 **MSTCMP2**：*主定时器比较 2 (Master Compare 2)*
主定时器比较 2 事件会强制将输出设为其有效状态。

位 8 **MSTCMP1**：*主定时器比较 1 (Master Compare 1)*
主定时器比较 1 事件会强制将输出设为其有效状态。

位 7 **MSTPER**：*主定时器周期 (Master Period)*
主定时器计数器在连续模式下翻转或主定时器在单发模式下复位都会强制将输出设为其有效状态。

位 6 **CMP4**：*定时器 x 比较 4 (Timer x Compare 4)*
定时器 A 比较 4 事件会强制将输出设为其有效状态。

位 5 **CMP3**：*定时器 x 比较 3 (Timer x Compare 3)*
定时器 A 比较 3 事件会强制将输出设为其有效状态。

位 4 **CMP2**：*定时器 x 比较 2 (Timer x Compare 2)*
定时器 A 比较 2 事件会强制将输出设为其有效状态。

位 3 **CMP1**：*定时器 x 比较 1 (Timer x Compare 1)*
定时器 A 比较 1 事件会强制将输出设为其有效状态。

位 2 **PER**：*定时器 x 周期 (Timer x Period)*
定时器 A 周期事件会强制将输出设为其有效状态。

位 1 **RESYNC**：*定时器 A 重新同步 (Timer A resynchronization)*
只有来自软件或 SYNC 输入的定时器 A 复位事件才会强制将输出设为其有效状态。
注： RESYNC=1 时，其他定时器复位事件不会影响输出。

位 0 **SST**：*软件置位触发 (Software Set trigger)*
该位会强制将输出设为其有效状态。该位仅可通过软件置 1，并由硬件复位。
注： 此位不能预装载。

37.5.28 HRTIM Timerx 输出 1 复位寄存器 (HRTIM_RSTx1R)

HRTIM Timerx Output1 Reset Register

偏移地址：0x40h（该偏移地址与定时器 x 基址相关）

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
UPDATE	EXTEVNT10	EXTEVNT9	EXTEVNT8	EXTEVNT7	EXTEVNT6	EXTEVNT5	EXTEVNT4	EXTEVNT3	EXTEVNT2	EXTEVNT1	TIMEVNT9	TIMEVNT8	TIMEVNT7	TIMEVNT6	TIMEVNT5
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TIMEVNT4	TIMEVNT3	TIMEVNT2	TIMEVNT1	MSTCMP4	MSTCMP3	MSTCMP2	MSTCMP1	MSTPER	CMP4	CMP3	CMP2	CMP1	PER	RESYNC	SRT
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:0 请参见 HRTIM_SETx1R 位说明。

这些位定义了可强制将 Tx1 输出设为其无效状态的源。

37.5.29 HRTIM Timerx 输出 2 置位寄存器 (HRTIM_SETx2R)

HRTIM Timerx Output2 Set Register

偏移地址：0x44h（该偏移地址与定时器 x 基址相关）

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
UPDATE	EXTEVNT10	EXTEVNT9	EXTEVNT8	EXTEVNT7	EXTEVNT6	EXTEVNT5	EXTEVNT4	EXTEVNT3	EXTEVNT2	EXTEVNT1	TIMEVNT9	TIMEVNT8	TIMEVNT7	TIMEVNT6	TIMEVNT5
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TIMEVNT4	TIMEVNT3	TIMEVNT2	TIMEVNT1	MSTCMP4	MSTCMP3	MSTCMP2	MSTCMP1	MSTPER	CMP4	CMP3	CMP2	CMP1	PER	RESYNC	SST
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:0 请参见 HRTIM_SETx1R 位说明。

这些位定义了可强制将 Tx2 输出设为其有效状态的源。

37.5.30 HRTIM Timerx 输出 2 复位寄存器 (HRTIM_RSTx2R)

HRTIM Timerx Output2 Reset Register

偏移地址：0x48h（该偏移地址与定时器 x 基址相关）

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
UPDATE	EXTEVNT10	EXTEVNT9	EXTEVNT8	EXTEVNT7	EXTEVNT6	EXTEVNT5	EXTEVNT4	EXTEVNT3	EXTEVNT2	EXTEVNT1	TIMEVNT9	TIMEVNT8	TIMEVNT7	TIMEVNT6	TIMEVNT5
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TIMEVNT4	TIMEVNT3	TIMEVNT2	TIMEVNT1	MSTCMP4	MSTCMP3	MSTCMP2	MSTCMP1	MSTPER	CMP4	CMP3	CMP2	CMP1	PER	RESYNC	SRT
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:0 请参见 HRTIM_SETx1R 位说明。

这些位定义了可强制将 Tx2 输出设为其无效状态的源。

37.5.31 HRTIM Timerx 外部事件过滤寄存器 1 (HRTIM_EEFxR1)

HRTIM Timerx External Event Filtering Register 1

偏移地址：0x4Ch（该偏移地址与定时器 x 基址相关）

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	EE5FLTR[3:0]				EE5LTCH	Res.	EE4FLTR[3:0]				EE4LTCH	Res.	EE3FLTR[3]
			rw	rw	rw	rw	rw		rw	rw	rw	rw	rw		rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EE3FLTR[2:0]			EE3LTCH	Res.	EE2FLTR[3:0]				EE2LTCH	Res.	EE1FLTR[3:0]				EE1LTCH
rw	rw	rw	rw		rw	rw	rw	rw	rw		rw	rw	rw	rw	rw

位 31:29 保留，必须保持复位值

位 28:25 **EE5FLTR[3:0]**：*外部事件 5 过滤器 (External Event 5 filter)*
请参见 EE1FLTR[3:0] 说明

位 24 **EE5LTCH**：*外部事件 5 锁存 (External Event 5 latch)*
请参见 EE1LTCH 说明

位 23 保留，必须保持复位值

位 22:19 **EE4FLTR[3:0]**：*外部事件 4 过滤器 (External Event 4 filter)*
请参见 EE1FLTR[3:0] 说明

位 18 **EE4LTCH**：*外部事件 4 锁存 (External Event 4 latch)*
请参见 EE1LTCH 说明

位 17 保留，必须保持复位值

位 16:13 **EE3FLTR[3:0]**：*外部事件 3 过滤器 (External Event 3 filter)*
请参见 EE1FLTR[3:0] 说明

位 12 **EE3LTCH**：*外部事件 3 锁存 (External Event 3 latch)*
请参见 EE1LTCH 说明

位 11 保留，必须保持复位值

位 10:7 **EE2FLTR[3:0]**：*外部事件 2 过滤器 (External Event 2 filter)*
请参见 EE1FLTR[3:0] 说明

位 6 **EE2LTCH**：*外部事件 2 锁存 (External Event 2 latch)*
请参见 EE1LTCH 说明

位 5 保留，必须保持复位值

位 4:1 **EE1FLTR[3:0]**：*外部事件 1 滤波器 (External Event 1 filter)*

0000：无滤波
0001：消隐持续时间从计数器复位/翻转到比较 1
0010：消隐持续时间从计数器复位/翻转到比较 2
0011：消隐持续时间从计数器复位/翻转到比较 3
0100：消隐持续时间从计数器复位/翻转到比较 4
0101：消隐与另一定时单元相关：TIMFLTR1 源（有关详细信息，请参见*表 290*）
0110：消隐与另一定时单元相关：TIMFLTR2 源（有关详细信息，请参见*表 290*）
0111：消隐与另一定时单元相关：TIMFLTR3 源（有关详细信息，请参见*表 290*）
1000：消隐与另一定时单元相关：TIMFLTR4 源（有关详细信息，请参见*表 290*）
1001：消隐与另一定时单元相关：TIMFLTR5 源（有关详细信息，请参见*表 290*）
1010：消隐与另一定时单元相关：TIMFLTR6 源（有关详细信息，请参见*表 290*）
1011：消隐与另一定时单元相关：TIMFLTR7 源（有关详细信息，请参见*表 290*）
1100：消隐与另一定时单元相关：TIMFLTR8 源（有关详细信息，请参见*表 290*）
1101：窗口持续时间从计数器复位/翻转到比较 2
1110：窗口持续时间从计数器复位/翻转到比较 3
1111：窗口与另一定时单元相关：TIMWIN 源（有关详细信息，请参见*表 291*）

注： *如果使用比较寄存器进行过滤，数值必须大于 0。*
计数器使能后（TxCEN 位置 1），不得修改该位域。

位 0 **EE1LTCH**：*外部事件 1 锁存 (External Event 1 latch)*

0：如果事件 1 在消隐期间发生，会被忽略，如果在窗口持续时间内发生，则会通过。
1：事件 1 会锁存并延迟，直至消隐或窗口周期结束。

注： *如果 EE1LTCH = 0，则会在窗口模式下生成超时事件（EE1FLTR[3:0]=1101、1110、1111），但外部事件在快速模式下编程的情况除外 (EExFAST = 1)。*
计数器使能后（TxCEN 位置 1），不得修改该位域。

37.5.32 HRTIM Timerx 外部事件过滤寄存器 2 (HRTIM_EEFxR2)

HRTIM Timerx External Event Filtering Register 2

偏移地址：0x50h（该偏移地址与定时器 x 基址相关）

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	EE10FLTR[3:0]				EE10LTCH	Res.	EE9FLTR[3:0]				EE9LTCH	Res.	EE8FLTR[3]
			rw	rw	rw	rw	rw		rw	rw	rw	rw	rw		rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EE8FLTR[2:0]			EE8LTCH	Res.	EE7FLTR[3:0]				EE7LTCH	Res.	EE6FLTR[3:0]				EE6LTCH
rw	rw	rw	rw		rw	rw	rw	rw	rw		rw	rw	rw	rw	rw

位 31:29 保留，必须保持复位值

位 28:25 **EE10FLTR[3:0]**：*外部事件 10 过滤器 (External Event 10 filter)*
请参见 EE1FLTR[3:0] 说明

位 24 **EE10LTCH**：*外部事件 10 锁存 (External Event 10 latch)*
请参见 EE1LTCH 说明

位 23 保留，必须保持复位值

位 22:19 **EE9FLTR[3:0]**：*外部事件 9 过滤器 (External Event 9 filter)*
请参见 EE1FLTR[3:0] 说明

位 18 **EE9LTCH**：*外部事件 9 锁存 (External Event 9 latch)*
请参见 EE1LTCH 说明

位 17 保留，必须保持复位值

位 16:13 **EE8FLTR[3:0]**：*外部事件 8 过滤器 (External Event 8 filter)*
请参见 EE1FLTR[3:0] 说明

位 12 **EE8LTCH**：*外部事件 8 锁存 (External Event 8 latch)*
请参见 EE1LTCH 说明

位 11 保留，必须保持复位值

位 10:7 **EE7FLTR[3:0]**：*外部事件 7 过滤器 (External Event 7 filter)*
请参见 EE1FLTR[3:0] 说明

位 6 **EE7LTCH**：*外部事件 7 锁存 (External Event 7 latch)*
请参见 EE1LTCH 说明

位 5 保留，必须保持复位值

位 4:1 **EE6FLTR[3:0]**：*外部事件 6 过滤器 (External Event 6 filter)*
请参见 EE1FLTR[3:0] 说明

位 0 **EE6LTCH**：*外部事件 6 锁存 (External Event 6 latch)*
请参见 EE1LTCH 说明

37.5.33 HRTIM Timerx 复位寄存器 (HRTIM_RSTxR)

HRTIM TimerA 复位寄存器 (HRTIM_RSTAR)

HRTIM TimerA Reset Register

偏移地址：0xD4h

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	TIMECMP4	TIMECMP2	TIMECMP1	TIMDCMP4	TIMDCMP2	TIMDCMP1	TIMCCMP4	TIMCCMP2	TIMCCMP1	TIMBCMP4	TIMBCMP2	TIMBCMP1	EXTEVNT10	EXTEVNT9	EXTEVNT8
	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EXTEVNT7	EXTEVNT6	EXTEVNT5	EXTEVNT4	EXTEVNT3	EXTEVNT2	EXTEVNT1	MSTCMP4	MSTCMP3	MSTCMP2	MSTCMP1	MSTPER	CMP4	CMP2	UPDT	Res.
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	

位 31 保留，必须保持复位值

位 30 **TECPM4**：*定时器 E 比较 4 (Timer E Compare 4)*
定时器 A 计数器会在发生定时器 E 比较 4 事件时复位。

位 29 **TECMP2**：*定时器 E 比较 2 (Timer E Compare 2)*
定时器 A 计数器会在发生定时器 E 比较 2 事件时复位。

位 28 **TECMP1**：*定时器 E 比较 1 (Timer E Compare 1)*
定时器 A 计数器会在发生定时器 E 比较 1 事件时复位。

位 27 **TDCMP4**：*定时器 D 比较 4 (Timer D Compare 4)*
定时器 A 计数器会在发生定时器 D 比较 4 事件时复位。

位 26 **TDCMP2**：*定时器 D 比较 2 (Timer D Compare 2)*
定时器 A 计数器会在发生定时器 D 比较 2 事件时复位。

位 25 **TDCMP1**：*定时器 D 比较 1 (Timer D Compare 1)*
定时器 A 计数器会在发生定时器 D 比较 1 事件时复位。

位 24 **TCCMP4**：*定时器 C 比较 4 (Timer C Compare 4)*
定时器 A 计数器会在发生定时器 C 比较 4 事件时复位。

位 23 **TCCMP2**：*定时器 C 比较 2 (Timer C Compare 2)*
定时器 A 计数器会在发生定时器 C 比较 2 事件时复位。

位 22 **TCCMP1**：*定时器 C 比较 1 (Timer C Compare 1)*
定时器 A 计数器会在发生定时器 C 比较 1 事件时复位。

位 21 **TBCMP4**：*定时器 B 比较 4 (Timer B Compare 4)*
定时器 A 计数器会在发生定时器 B 比较 4 事件时复位。

位 20 **TBCMP2**：*定时器 B 比较 2 (Timer B Compare 2)*
定时器 A 计数器会在发生定时器 B 比较 2 事件时复位。

位 19 **TBCMP1**：*定时器 B 比较 1 (Timer B Compare 1)*
定时器 A 计数器会在发生定时器 B 比较 1 事件时复位。

位 18 **EXTEVNT10**：*外部事件 (External Event)*
定时器 A 计数器会在发生外部事件 10 时复位。

位 17 **EXTEVNT9**：*外部事件 9 (External Event 9)*
定时器 A 计数器会在发生外部事件 9 时复位。

位 16 **EXTEVNT8**：*外部事件 8 (External Event 8)*
定时器 A 计数器会在发生外部事件 8 时复位。

位 15 **EXTEVNT7**：*外部事件 7 (External Event 7)*
定时器 A 计数器会在发生外部事件 7 时复位。

位 14 **EXTEVNT6**：*外部事件 6 (External Event 6)*
定时器 A 计数器会在发生外部事件 6 时复位。

位 13 **EXTEVNT5**：*外部事件 5 (External Event 5)*
定时器 A 计数器会在发生外部事件 5 时复位。

位 12 **EXTEVNT4**：*外部事件 4 (External Event 4)*
定时器 A 计数器会在发生外部事件 4 时复位。

位 11 **EXTEVNT3**：*外部事件 3 (External Event 3)*
定时器 A 计数器会在发生外部事件 3 时复位。

位 10 **EXTEVNT2**：*外部事件 2 (External Event 2)*
定时器 A 计数器会在发生外部事件 2 时复位。

位 9 **EXTEVNT1**：*外部事件 1 (External Event 1)*
定时器 A 计数器会在发生外部事件 1 时复位。

位 8 **MSTCMP4**：*主定时器比较 4 (Master compare 4)*
定时器 A 计数器会在发生主定时器比较 4 事件时复位。

位 7 **MSTCMP3**：*主定时器比较 3 (Master compare 3)*
定时器 A 计数器会在发生主定时器比较 3 事件时复位。

位 6 **MSTCMP2**：*主定时器比较 2 (Master compare 2)*
定时器 A 计数器会在发生主定时器比较 2 事件时复位。

位 5 **MSTCMP1**：*主定时器比较 1 (Master compare 1)*
定时器 A 计数器会在发生主定时器比较 1 事件时复位。

位 4 **MSTPER**：*主定时器周期 (Master timer Period)*
定时器 A 计数器会在发生主定时器周期事件时复位。

位 3 **CMP4**：*定时器 A 比较 4 复位 (Timer A compare 4 reset)*
定时器 A 计数器会在发生定时器 A 比较 4 事件时复位。

位 2 **CMP2**：*定时器 A 比较 2 复位 (Timer A compare 2 reset)*
定时器 A 计数器会在发生定时器 A 比较 2 事件时复位。

位 1 **UPDT**：*定时器 A 更新复位 (Timer A Update reset)*
定时器 A 计数器会在发生更新事件时复位。

位 0 保留，必须保持复位值

HRTIM TimerB 复位寄存器 (HRTIM_RSTBR)

HRTIM TimerB Reset Register

偏移地址：0x154h

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	TIMECMP4	TIMECMP2	TIMECMP1	TIMDCMP4	TIMDCMP2	TIMDCMP1	TIMCCMP4	TIMCCMP2	TIMCCMP1	TIMACMP4	TIMACMP2	TIMACMP1	EXTEVNT10	EXTEVNT9	EXTEVNT8
	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EXTEVNT7	EXTEVNT6	EXTEVNT5	EXTEVNT4	EXTEVNT3	EXTEVNT2	EXTEVNT1	MSTCMP4	MSTCMP3	MSTCMP2	MSTCMP1	MSTPER	CMP4	CMP2	UPDT	Res.
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	

位 30:1 请参见 HRTIM_RSTAR 位说明。

位 30:19 有所不同（来自 TIMA、TIMC、TIMD 和 TIME 的复位信号）

HRTIM TimerC 复位寄存器 (HRTIM_RSTCR)

HRTIM TimerC Reset Register

偏移地址：0x1D4h

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	TIMECMP4	TIMECMP2	TIMECMP1	TIMDCMP4	TIMDCMP2	TIMDCMP1	TIMBCMP4	TIMBCMP2	TIMBCMP1	TIMACMP4	TIMACMP2	TIMACMP1	EXTEVNT10	EXTEVNT9	EXTEVNT8
	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EXTEVNT7	EXTEVNT6	EXTEVNT5	EXTEVNT4	EXTEVNT3	EXTEVNT2	EXTEVNT1	MSTCMP4	MSTCMP3	MSTCMP2	MSTCMP1	MSTPER	CMP4	CMP2	UPDT	Res.
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	

位 30:1 请参见 HRTIM_RSTAR 位说明。

位 30:19 有所不同（来自 TIMA、TIMB、TIMD 和 TIME 的复位信号）

HRTIM TimerD 复位寄存器 (HRTIM_RSTDR)

HRTIM TimerD Reset Register

偏移地址：0x254h

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	TIMECMP4	TIMECMP2	TIMECMP1	TIMCCMP4	TIMCCMP2	TIMCCMP1	TIMBCMP4	TIMBCMP2	TIMBCMP1	TIMACMP4	TIMACMP2	TIMACMP1	EXTEVNT10	EXTEVNT9	EXTEVNT8
	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EXTEVNT7	EXTEVNT6	EXTEVNT5	EXTEVNT4	EXTEVNT3	EXTEVNT2	EXTEVNT1	MSTCMP4	MSTCMP3	MSTCMP2	MSTCMP1	MSTPER	CMP4	CMP2	UPDT	Res.
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	

位 30:1 请参见 HRTIM_RSTAR 位说明。

位 30:19 有所不同（来自 TIMA、TIMB、TIMC 和 TIME 的复位信号）

HRTIM Timerx 复位寄存器 (HRTIM_RSTER)

HRTIM Timerx Reset Register

偏移地址：0x2D4h

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	TIMDCMP4	TIMDCMP2	TIMDCMP1	TIMCCMP4	TIMCCMP2	TIMCCMP1	TIMBCMP4	TIMBCMP2	TIMBCMP1	TIMACMP4	TIMACMP2	TIMACMP1	EXTEVNT10	EXTEVNT9	EXTEVNT8
	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EXTEVNT7	EXTEVNT6	EXTEVNT5	EXTEVNT4	EXTEVNT3	EXTEVNT2	EXTEVNT1	MSTCMP4	MSTCMP3	MSTCMP2	MSTCMP1	MSTPER	CMP4	CMP2	UPDT	Res.
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	

位 30:1 请参见 HRTIM_RSTAR 位说明。

位 30:19 有所不同（来自 TIMA、TIMB、TIMC 和 TIMD 的复位信号）

37.5.34 HRTIM Timerx 斩波寄存器 (HRTIM_CHPxR)

HRTIM Timerx Chopper Register

偏移地址：0x58h（该偏移地址与定时器 x 基址相关）

复位值：0x0000 0000

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

位 31:11 保留，必须保持复位值

位 10:7 **STRPW[3:0]**：*Timerx 启动脉宽 (Timerx start pulsewidth)*

该寄存器定义输出信号上升沿后的初始脉宽。

如果其中一个 CHPx 位置 1，此位域不能修改。

t_{1STPW} = (STRPW[3:0]+1) x 16 x t_{HRTIM}。

0000：40 ns (1/25 MHz)

...

1111：640 ns (16/25 MHz)

位 6:4 **CARDTY[2:0]**：*Timerx 斩波占空比值 (Timerx chopper duty cycle value)*

该寄存器定义载波信号的占空比。如果其中一个 CHPx 位置 1，此位域不能修改。

000：0/8（即，只会出现第一个脉冲）

...

111：7/8

位 3:0 **CARFRQ[3:0]**：*Timerx 载波频率值 (Timerx carrier frequency value)*

该寄存器定义载波频率 F_{CHPFRQ} = f_{HRTIM} / (16 x (CARFRQ[3:0]+1))。

如果其中一个 CHPx 位置 1，此位域不能修改。

0000：25 MHz (f_{HRTIM}/ 16)

...

1111：1.56 MHz (f_{HRTIM} / 256)

37.5.35 HRTIM Timerx 捕获 1 控制寄存器 (HRTIM_CPT1xCR)

HRTIM Timerx Capture 1 Control Register

偏移地址：0x5Ch（该偏移地址与定时器 x 基址相关）

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留（仅针对 TIME）				保留（仅针对 TIMD）				保留（仅针对 TIMC）				保留（仅针对 TIMB）			
TECMP2	TECMP1	TE1RST	TE1SET	TDCMP2	TDCMP1	TD1RST	TD1SET	TCCMP2	TCCMP1	TC1RST	TC1SET	TBCMP2	TBCMP1	TB1RST	TB1SET
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留（仅针对 TIMA）				EXEV10CPT	EXEV9CPT	EXEV8CPT	EXEV7CPT	EXEV6CPT	EXEV5CPT	EXEV4CPT	EXEV3CPT	EXEV2CPT	EXEV1CPT	UPDCPT	SWCPT
TACMP2	TACMP1	TA1RST	TA1SET												
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:0 请参见 HRTIM_CPT2xCR 位说明

37.5.36 HRTIM Timerx 捕获 2 控制寄存器 (HRTIM_CPT2xCR)

HRTIM Timerx Capture 2 Control Register

偏移地址：0x60h（该偏移地址与定时器 x 基址相关）

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
保留（仅针对 TIME）				保留（仅针对 TIMD）				保留（仅针对 TIMC）				保留（仅针对 TIMB）			
TECMP2	TECMP1	TE1RST	TE1SET	TDCMP2	TDCMP1	TD1RST	TD1SET	TCCMP2	TCCMP1	TC1RST	TC1SET	TBCMP2	TBCMP1	TB1RST	TB1SET
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
保留（仅针对 TIMA）				EXEV10CPT	EXEV9CPT	EXEV8CPT	EXEV7CPT	EXEV6CPT	EXEV5CPT	EXEV4CPT	EXEV3CPT	EXEV2CPT	EXEV1CPT	UPDCPT	SWCPT
TACMP2	TACMP1	TA1RST	TA1SET												
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31 **TECMP2**：定时器 E 比较 2 (Timer E Compare 2)
请参见 TACMP1 说明
注： 该位为定时器 E 保留

位 30 **TECMP1**：定时器 E 比较 1 (Timer E Compare 1)
请参见 TACMP1 说明
注： 该位为定时器 E 保留

位 29 **TE1RST**：定时器 E 输出 1 复位 (Timer E output 1 Reset)
请参见 TA1RST 说明
注： 该位为定时器 E 保留

位 28 **TE1SET**：定时器 E 输出 1 置位 (Timer E output 1 Set)
请参见 TA1SET 说明
注： 该位为定时器 E 保留

位 27 **TDCMP2**：定时器 D 比较 2 (Timer D Compare 2)
请参见 TACMP1 说明
注： 该位为定时器 D 保留

位 26 **TDCMP1**：定时器 D 比较 1 (Timer D Compare 1)
请参见 TACMP1 说明
注： 该位为定时器 D 保留

位 25 **TD1RST**：定时器 D 输出 1 复位 (Timer D output 1 Reset)
请参见 TA1RST 说明
注： 该位为定时器 D 保留

位 24 **TD1SET**：定时器 D 输出 1 置位 (Timer D output 1 Set)
请参见 TA1SET 说明
注： 该位为定时器 D 保留

位 23 **TCCMP2**：定时器 C 比较 2 (Timer C Compare 2)
请参见 TACMP1 说明
注： 该位为定时器 C 保留

位 22 **TCCMP1**：定时器 C 比较 1 (Timer C Compare 1)
请参见 TACMP1 说明
注： 该位为定时器 C 保留

位 21 **TC1RST**：定时器 C 输出 1 复位 (Timer C output 1 Reset)
请参见 TA1RST 说明
注： 该位为定时器 C 保留

位 20 **TC1SET**：定时器 C 输出 1 置位 (Timer C output 1 Set)
请参见 TA1SET 说明
注： 该位为定时器 C 保留

位 19 **TBCMP2**：定时器 B 比较 2 (Timer B Compare 2)
请参见 TACMP1 说明
注： 该位为定时器 B 保留

位 18 **TBCMP1**：定时器 B 比较 1 (Timer B Compare 1)
请参见 TACMP1 说明
注： 该位为定时器 B 保留

位 17 **TB1RST**：定时器 B 输出 1 复位 (Timer B output 1 Reset)
请参见 TA1RST 说明
注： 该位为定时器 B 保留

位 16 **TB1SET**：定时器 B 输出 1 置位 (Timer B output 1 Set)
请参见 TA1SET 说明
注： 该位为定时器 B 保留

位 15 **TACMP2**：定时器 A 比较 2 (Timer A Compare 2)
定时器 A 比较 2 会触发捕获 2。
注： 该位为定时器 A 保留

位 14 **TACMP1**：定时器 A 比较 1 (Timer A Compare 1)
定时器 A 比较 1 会触发捕获 2。
注： 该位为定时器 A 保留

位 13 **TA1RST**：定时器 A 输出 1 复位 (Timer A output 1 Reset)
HRTIM_CHA1 输出从有效电平跳变为无效电平会触发捕获 2。
注： 该位为定时器 A 保留

位 12 **TA1SET**：定时器 A 输出 1 置位 (Timer A output 1 Set)
HRTIM_CHA1 输出从无效电平跳变为有效电平会触发捕获 2。
注： 该位为定时器 A 保留

位 11 **EXEV10CPT**：*外部事件 10 捕获 (External Event 10 Capture)*
请参见 EXEV1CPT 说明

位 10 **EXEV9CPT**：*外部事件 9 捕获 (External Event 9 Capture)*
请参见 EXEV1CPT 说明

位 9 **EXEV8CPT**：*外部事件 8 捕获 (External Event 8 Capture)*
请参见 EXEV1CPT 说明

位 8 **EXEV7CPT**：*外部事件 7 捕获 (External Event 7 Capture)*
请参见 EXEV1CPT 说明

位 7 **EXEV6CPT**：*外部事件 6 捕获 (External Event 6 Capture)*
请参见 EXEV1CPT 说明

位 6 **EXEV5CPT**：*外部事件 5 捕获 (External Event 5 Capture)*
请参见 EXEV1CPT 说明

位 5 **EXEV4CPT**：*外部事件 4 捕获 (External Event 4 Capture)*
请参见 EXEV1CPT 说明

位 4 **EXEV3CPT**：*外部事件 3 捕获 (External Event 3 Capture)*
请参见 EXEV1CPT 说明

位 3 **EXEV2CPT**：*外部事件 2 捕获 (External Event 2 Capture)*
请参见 EXEV1CPT 说明

位 2 **EXEV1CPT**：*外部事件 1 捕获 (External Event 1 Capture)*
外部事件 1 会触发捕获 2。

位 1 **UPDCPT**：*更新捕获 (Update Capture)*
更新事件会触发捕获 2。

位 0 **SWCPT**：*软件捕获 (Software Capture)*
此位会强制通过软件触发捕获 2。此位仅置 1，由硬件复位。

37.5.37 HRTIM Timerx 输出寄存器 (HRTIM_OUTxR)

HRTIM Timerx Output Register

偏移地址：0x64h（该偏移地址与定时器 x 基址相关）

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DIDL2	CHP2	FAULT2[1:0]		IDLES2	IDLEM 2	POL2	Res.
								rw	rw	rw	rw	rw	rw	rw	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	DLYPRT[2:0]			DLYPR TEN	DTEN	DIDL1	CHP1	FAULT1[1:0]		IDLES1	IDLEM 1	POL1	Res.
			rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	

位 31:24 保留，必须保持复位值

位 23 **DIDL2**：*进入突发模式空闲状态的输出 2 死区 (Output 2 Deadtime upon burst mode Idle entry)*

此位会在输出切换为其空闲状态之前强制插入死区，从而延迟进入空闲模式。此设置仅适用于在突发模式工作期间进入空闲状态的情况。

0：编程的空闲状态立即应用到输出 2。

1：死区（无效电平）会在进入空闲模式之前插入到输出 2 上。死区值由 DTFx[8:0] 设置。

注： 定时器 x 使能后，不能更改此参数。

仅当其中一路输出在突发模式期间有效 (IDLES=1)、且死区为正值（SDTR/SDTF 设为 0）时，才可设置 DIDL=1。

位 22 **CHP2**：*输出 2 斩波使能 (Output 2 Chopper enable)*

该位在输出 2 上使能斩波

0：输出信号未改变

1：输出信号被载波信号斩波

注： 定时器 x 使能后，不能更改此参数。

位 21:20 **FAULT2[1:0]**：*输出 2 故障状态 (Output 2 Fault state)*

这些位选择输出 2 在故障事件后的状态

00：无操作：输出不受故障输入影响，停留在运行模式下

01：有效

10：无效

11：高阻态

注： 如果 FLTENx 位已置 1，或者输出处于 FAULT 状态，则定时器 x 使能后（TxCEN 位置 1），不能更改此参数。

位 19 **IDLES2**：*输出 2 空闲状态 (Output 2 Idle State)*

此位选择输出 2 的空闲状态

0：无效

1：有效

注： 必须在交由 HRTIM 控制输出之前设置此参数。

位 18 **IDLEM2**：*输出 2 空闲模式 (Output 2 Idle mode)*

此位选择输出 2 的空闲模式

0：无操作：输出不受突发模式工作的影响

1：通过突发模式控制器请求时，输出处于空闲状态。

注： 此位进行了预装载，可在运行期间更改，但不能在突发模式激活时更改。

位 17 **POL2**：*输出 2 极性 (Output 2 polarity)*

此位选择输出 2 极性

0：正极性（输出高电平有效）

1：负极性（输出低电平有效）

注： 定时器 x 使能后，不能更改此参数。

位 16:13 保留，必须保持复位值

位 12:10 **DLYPRT[2:0]**：*延迟保护 (Delayed Protection)*

这些位定义应用延迟保护机制的源和输出。

在 HRTIM_OUTAR、HRTIM_OUTBR、HRTIM_OUTCR 中：

000：发生外部事件 6 时输出 1 进入延迟空闲状态

001：发生外部事件 6 时输出 2 进入延迟空闲状态

010：发生外部事件 6 时输出 1 和输出 2 均进入延迟空闲状态

011：发生外部事件 6 时进入均衡空闲状态

100：发生外部事件 7 时输出 1 进入延迟空闲状态

101：发生外部事件 7 时输出 2 进入延迟空闲状态

110：发生外部事件 7 时输出 1 和输出 2 均进入延迟空闲状态

111：发生外部事件 7 时进入均衡空闲状态

在 HRTIM_OUTDR、HRTIM_OUTER 中：

000：发生外部事件 8 时输出 1 进入延迟空闲状态

001：发生外部事件 8 时输出 2 进入延迟空闲状态

010：发生外部事件 8 时输出 1 和输出 2 均进入延迟空闲状态

011：发生外部事件 8 时进入均衡空闲状态

100：发生外部事件 9 时输出 1 进入延迟空闲状态

101：发生外部事件 9 时输出 2 进入延迟空闲状态

110：发生外部事件 9 时输出 1 和输出 2 均进入延迟空闲状态

111：发生外部事件 9 时进入均衡空闲状态

注： 延迟保护使能后（DLYPRTEN 位置 1），不得修改此位域。

位 9 **DLYPRTEN**：*延迟保护使能 (Delayed Protection Enable)*

此位用于使能延迟保护机制

0：不执行任何操作

1：根据 DLYPRT[2:0] 位使能延迟保护

注： 定时器 x 使能后（TxEN 位置 1），不能更改此参数。

位 8 **DTEN**：*死区使能 (Deadtime enable)*

此位用于使能在输出 1 和输出 2 上插入死区

0：输出 1 信号和输出 2 信号相互独立。

1：会在输出 1 和输出 2 之间插入死区（参考信号来自输出 1 信号发生器）

注： 定时器工作时（TxEN 位置 1），或者其输出通过另一定时器使能和置位/复位时，不能修改此参数。

位 7 **DIDL1**：*进入突发模式空闲状态的输出 1 死区 (Output 1 Deadtime upon burst mode Idle entry)*

此位会在输出切换为其空闲状态之前强制插入死区，从而延迟进入空闲模式。此设置仅适用于在突发模式工作期间进入空闲状态的情况。

0：编程的空闲状态立即应用到输出 1。

1：死区（无效电平）会在进入空闲模式之前插入到输出 1 上。死区值由 DTRx[8:0] 设置。

注： 定时器 x 使能后，不能更改此参数。

仅当其中一路输出在突发模式期间有效 (IDLES=1)、且死区为正值（SDTR/SDTF 设为 0）时，才可设置 DIDL=1。

位 6 **CHP1**：*输出 1 斩波使能 (Output 1 Chopper enable)*

该位在输出 1 上使能斩波

0：输出信号未改变

1：输出信号被载波信号斩波

注： 定时器 x 使能后，不能更改此参数。

位 5:4 **FAULT1[1:0]**：*输出 1 故障状态 (Output 1 Fault state)*

这些位选择输出 1 在故障事件后的状态

00：无操作：输出不受故障输入影响，停留在运行模式下。

01：有效

10：无效

11：高阻态

注： 如果 FLTENx 位已置 1，或者输出处于 FAULT 状态，则定时器 x 使能后（TxCEN 位置 1），不能更改此参数。

位 3 **IDLES1**：*输出 1 空闲状态 (Output 1 Idle State)*

此位选择输出 1 的空闲状态

0：无效

1：有效

注： 必须在 HRTIM 控制输出之前设置此参数。

位 2 **IDLEM1**：*输出 1 空闲模式 (Output 1 Idle mode)*

此位选择输出 1 的空闲模式

0：无操作：输出不受突发模式工作的影响

1：通过突发模式控制器请求时，输出处于空闲状态

注： 此位进行了预装载，可在运行期间更改，但不能在突发模式激活时更改。

位 1 **POL1**：*输出 1 极性 (Output 1 polarity)*

此位选择输出 1 的极性

0：正极性（输出高电平有效）

1：负极性（输出低电平有效）

注： 定时器 x 使能后，不能更改此参数。

位 0 保留

37.5.38 HRTIM Timerx 故障寄存器 (HRTIM_FLTxR)

HRTIM Timerx Fault Register

偏移地址：0x68h（该偏移地址与定时器 x 基址相关）

复位值：0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FLT5EN	FLT4EN	FLT3EN	FLT2EN	FLT1EN
											rw	rw	rw	rw	rw

位 31 **FLTLCK**：*故障源锁定 (Fault sources Lock)*

0：FLT1EN..FLT5EN 位为读/写位

1：FLT1EN..FLT5EN 位为只读位

FLTLCK 位仅可写入一次。该位置 1 后，在下一次系统复位之前不能进行修改。

位 30:5 保留，必须保持复位值

位 4 **FLT5EN**：*故障 5 使能 (Fault 5 enable)*

0：忽略故障 5 输入

1：故障 5 输入有效，可禁止 HRTIM 输出。

位 3 **FLT4EN**：*故障 4 使能 (Fault 4 enable)*

0：忽略故障 4 输入

1：故障 4 输入有效，可禁止 HRTIM 输出。

位 2 **FLT3EN**：*故障 3 使能 (Fault 3 enable)*

0：忽略故障 3 输入

1：故障 3 输入有效，可禁止 HRTIM 输出。

位 1 **FLT2EN**：*故障 2 使能 (Fault 2 enable)*

0：忽略故障 2 输入

1：故障 2 输入有效，可禁止 HRTIM 输出。

位 0 **FLT1EN**：*故障 1 使能 (Fault 1 enable)*

0：忽略故障 1 输入

1：故障 1 输入有效，可禁止 HRTIM 输出。

37.5.39 HRTIM 控制寄存器 1 (HRTIM_CR1)

HRTIM Control Register 1

偏移地址：0x380h

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	AD4USRC[2:0]			AD3USRC[2:0]			AD2USRC[2:0]			AD1USRC[2:0]		
				rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TEUDIS	TDUDIS	TCUDIS	TBUDIS	TAUDIS	MUDIS
										rw	rw	rw	rw	rw	rw

位 31:28 保留，必须保持复位值

位 27:25 **AD4USRC[2:0]**：*ADC 触发 4 更新源 (ADC Trigger 4* Update Source)
请参见 AD1USRC[2:0] 说明

位 24:22 **AD3USRC[2:0]**：*ADC 触发 3 更新源 (ADC Trigger 3* Update Source)
请参见 AD1USRC[2:0] 说明

位 21:19 **AD2USRC[2:0]**：*ADC 触发 2 更新源 (ADC Trigger 2* Update Source)
请参见 AD1USRC[2:0] 说明

位 18:16 **AD1USRC[2:0]**：*ADC 触发 1 更新源 (ADC Trigger 2* Update Source)
这些位定义将触发 HRTIM_ADC1R 寄存器更新的源（从预装载寄存器传输到活动寄存器）。它只定义源定时器。准确条件在定时器自身的 HRTIM_MCR 或 HRTIM_TIMxCR 中定义。
000：主定时器
001：定时器 A
010：定时器 B
011：定时器 C
100：定时器 D
101：定时器 E
110,111：保留

位 15:6 保留，必须保持复位值

位 5 **TEUDIS**：*定时器 E 更新禁止 (Timer E Update Disable)*
请参见 TAUDIS 说明

位 4 **TDUDIS**：*定时器 D 更新禁止 (Timer D Update Disable)*
请参见 TAUDIS 说明

位 3 **TCUDIS**：*定时器 C 更新禁止 (Timer C Update Disable)*
请参见 TAUDIS 说明

位 2 **TBUDIS**：*定时器 B 更新禁止 (Timer B Update Disable)*
请参见 TAUDIS 说明

位 1 **TAUDIS**：*定时器 A 更新禁止 (Timer A Update Disable)*
此位由软件置 1 和清零，用于暂时使能/禁止定时器 A 上的更新事件生成。
0：使能更新。生成所选源后立即进行更新。
1：禁止更新。暂时禁止更新，使软件能够写入多个需要同时考虑的寄存器。

位 0 **MUDIS**：主定时器更新使能 (Master *Update Disable)*
此位由软件置 1 和清零，用于暂时使能/禁止更新事件生成。
0：使能更新。
1：禁止更新。暂时禁止更新，使软件能够写入多个需要同时考虑的寄存器。

37.5.40 HRTIM 控制寄存器 2 (HRTIM_CR2)

HRTIM Control Register 2

偏移地址：0x384h

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	TERST	TDRST	TCRST	TBRST	TARST	MRST	Res.	Res.	TESWU	TDSWU	TCSWU	TBSWU	TASWU	MSWU
		rw	rw	rw	rw	rw	rw			rw	rw	rw	rw	rw	rw

位 31:14 保留，必须保持复位值

位 13 **TERST**：*定时器 E 计数器软件复位 (Timer E counter software reset)*
请参见 TARST 说明

位 12 **TDRST**：*定时器 D 计数器软件复位 (Timer D counter software reset)*
请参见 TARST 说明

位 11 **TCRST**：*定时器 C 计数器软件复位 (Timer C counter software reset)*
请参见 TARST 说明

位 10 **TBRST**：*定时器 B 计数器软件复位 (Timer B counter software reset)*
请参见 TARST 说明

位 9 **TARST**：*定时器 A 计数器软件复位 (Timer A counter software reset)*
将此位置 1 会复位定时器 A 计数器。
此位自动通过硬件复位。

位 8 **MRST**：*主定时器计数器软件复位 (Master Counter software reset)*
将此位置 1 会复位主定时器计数器。
此位自动通过硬件复位。

位 7:6 保留，必须保持复位值

位 5 **TESWU**：*定时器 E 软件更新 (Timer E Software Update)*
请参见 TASWU 说明

位 4 **TDSWU**：*定时器 D 软件更新 (Timer D Software Update)*
请参见 TASWU 说明

位 3 **TCSWU**：*定时器 C 软件更新 (Timer C Software Update)*
请参见 TASWU 说明

位 2 **TBSWU**：*定时器 B 软件更新 (Timer B Software Update)*
请参见 TASWU 说明

位 1 **TASWU**：*定时器 A 软件更新 (Timer A Software Update)*
此位由软件置 1，并由硬件自动复位。它会强制立即执行从预装载寄存器到活动寄存器的传输操作，并会取消任何挂起的更新请求。

位 0 **MSWU**：*主定时器软件更新 (Master Timer Software update)*
此位由软件置 1，并由硬件自动复位。它会强制立即执行从主定时器中的预装载寄存器到活动寄存器的传输操作，并会取消任何挂起的更新请求。

37.5.41 HRTIM 中断状态寄存器 (HRTIM_ISR)

HRTIM Interrupt Status Register

偏移地址：0x388h

复位值：0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SYSFLT	FLT5	FLT4	FLT3	FLT2	FLT1
											r	r	r	r	r

位 31:18 保留，必须保持复位值

位 17 **BMPER**：突发模式周期中断标志 (Burst mode Period Interrupt Flag)
单发突发模式工作结束、或连续模式下突发模式周期结束时，此位会由硬件置 1。通过软件向该位写入 1 可将其清零。
0：未发生突发模式周期中断
1：发生突发模式周期中断

位 16:6 保留，必须保持复位值

位 5 **SYSFLT**：系统故障中断标志 (System Fault Interrupt Flag)
请参见 FLT1 说明

位 4 **FLT5**：故障 5 中断标志 (Fault 5 Interrupt Flag)
请参见 FLT1 说明

位 3 **FLT4**：故障 4 中断标志 (Fault 4 Interrupt Flag)
请参见 FLT1 说明

位 2 **FLT3**：故障 3 中断标志 (Fault 3 Interrupt Flag)
请参见 FLT1 说明

位 1 **FLT2**：故障 2 中断标志 (Fault 2 Interrupt Flag)
请参见 FLT1 说明

位 0 **FLT1**：故障 1 中断标志 (Fault 1 Interrupt Flag)
发生故障 1 事件时，该位由硬件置 1。通过软件向该位写入 1 可将其清零。
0：未发生故障 1 中断
1：发生故障 1 中断

37.5.42 HRTIM 中断清零寄存器 (HRTIM_ICR)

HRTIM Interrupt Clear Register

偏移地址：0x38Ch

复位值：0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SYSFLTC	FLT5C	FLT4C	FLT3C	FLT2C	FLT1C
											w	w	w	w	w

位 31:18 保留，必须保持复位值

位 17 **BMPERC**：突发模式周期标志清零 (Burst mode period flag Clear)
将 1 写入此位时，HRTIM_ISR 寄存器中的 BMPER 标志将清零。

位 16:6 保留，必须保持复位值

位 5 **SYSFLTC**：系统故障中断标志清零 (System Fault Interrupt Flag Clear)
将 1 写入此位时，HRTIM_ISR 寄存器中的 SYSFLT 标志将清零。

位 4 **FLT5C**：故障 5 中断标志清零 (Fault 5 Interrupt Flag Clear)
将 1 写入此位时，HRTIM_ISR 寄存器中的 FLT5 标志将清零。

位 3 **FLT4C**：故障 4 中断标志清零 (Fault 4 Interrupt Flag Clear)
将 1 写入此位时，HRTIM_ISR 寄存器中的 FLT4 标志将清零。

位 2 **FLT3C**：故障 3 中断标志清零 (Fault 3 Interrupt Flag Clear)
将 1 写入此位时，HRTIM_ISR 寄存器中的 FLT3 标志将清零。

位 1 **FLT2C**：故障 2 中断标志清零 (Fault 2 Interrupt Flag Clear)
将 1 写入此位时，HRTIM_ISR 寄存器中的 FLT2 标志将清零。

位 0 **FLT1C**：故障 1 中断标志清零 (Fault 1 Interrupt Flag Clear)
将 1 写入此位时，HRTIM_ISR 寄存器中的 FLT1 标志将清零。

37.5.43 HRTIM 中断使能寄存器 (HRTIM_IER)

HRTIM Interrupt Enable Register

偏移地址：0x390h

复位值：0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SYSFLTIE	FLT5IE	FLT4IE	FLT3IE	FLT2IE	FLT1IE
											rw	rw	rw	rw	rw

位 31:18 保留，必须保持复位值

位 17 **BMPERIE**：突发模式周期中断使能 (Burst mode period Interrupt Enable)
此位由软件置 1 和清零，用于使能/禁止突发模式周期中断。
0：禁止突发模式周期中断
1：使能突发模式周期中断

位 16:6 保留，必须保持复位值

位 5 **SYSFLTIE**：系统故障中断使能 (System Fault Interrupt Enable)
请参见 FLT1IE 说明

位 4 **FLT5IE**：故障 5 中断使能 (Fault 5 Interrupt Enable)
请参见 FLT1IE 说明

位 3 **FLT4IE**：故障 4 中断使能 (Fault 4 Interrupt Enable)
请参见 FLT1IE 说明

位 2 **FLT3IE**：故障 3 中断使能 (Fault 3 Interrupt Enable)
请参见 FLT1IE 说明

位 1 **FLT2IE**：故障 2 中断使能 (Fault 2 Interrupt Enable)
请参见 FLT1IE 说明

位 0 **FLT1IE**：故障 1 中断使能 (Fault 1 Interrupt Enable)
此位由软件置 1 和清零，用于使能/禁止故障 1 中断。
0：禁止故障 1 中断
1：使能故障 1 中断

37.5.44 HRTIM 输出使能寄存器 (HRTIM_OENR)

HRTIM Output Enable Register

偏移地址：0x394h

复位值：0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	TE2OEN	TE1OEN	TD2OEN	TD1OEN	TC2OEN	TC1OEN	TB2OEN	TB1OEN	TA2OEN	TA1OEN
						rs	rs	rs	rs	rs	rs	rs	rs	rs	rs

位 31:10 保留，必须保持复位值

位 9 **TE2OEN**：定时器 E 输出 2 使能 (Timer E Output 2 Enable)
请参见 TA1OEN 说明

位 8 **TE1OEN**：定时器 E 输出 1 使能 (Timer E Output 1 Enable)
请参见 TA1OEN 说明

位 7 **TD2OEN**：定时器 D 输出 2 使能 (Timer D Output 2 Enable)
请参见 TA1OEN 说明

位 6 **TD1OEN**：定时器 D 输出 1 使能 (Timer D Output 1 Enable)
请参见 TA1OEN 说明

位 5 **TC2OEN**：定时器 C 输出 2 使能 (Timer C Output 2 Enable)
请参见 TA1OEN 说明

位 4 **TC1OEN**：定时器 C 输出 1 使能 (Timer C Output 1 Enable)
请参见 TA1OEN 说明

位 3 **TB2OEN**：定时器 B 输出 2 使能 (Timer B Output 2 Enable)
请参见 TA1OEN 说明

位 2 **TB1OEN**：定时器 B 输出 1 使能 (Timer B Output 1 Enable)
请参见 TA1OEN 说明

位 1 **TA2OEN**：定时器 A 输出 2 使能 (Timer A Output 2 Enable)
请参见 TA1OEN 说明

位 0 **TA1OEN**：定时器 A 输出 1 (HRTIM_CHA1) 使能 (Timer A Output 1 (HRTIM_CHA1) Enable)
将此位置 1 会使能定时器 A 输出 1。写入“0”无影响。
读取此位会返回输出使能/禁止状态。
定时器相关故障输入变为有效状态后，此位会立即由硬件异步清零。
0：禁止输出 HRTIM_CHA1。输出处于故障状态或空闲状态。
1：使能输出 HRTIM_CHA1。

注： 禁止状态对应空闲和故障两种状态。输出禁止状态由 HRTIM_ODSR 寄存器中的 TA1ODS 位提供。

37.5.45 HRTIM 输出禁止寄存器 (HRTIM_ODISR)

HRTIM Output Disable Register

偏移地址：0x398h

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	TE2ODIS	TE1ODIS	TD2ODIS	TD1ODIS	TC2ODIS	TC1ODIS	TB2ODIS	TB1ODIS	TA2ODIS	TA1ODIS
						w	w	w	w	w	w	w	w	w	w

位 31:10 保留，必须保持复位值

位 9 **TE2ODIS**：定时器 E 输出 2 禁止 (Timer E Output 2 disable)
请参见 TA1ODIS 说明

位 8 **TE1ODIS**：定时器 E 输出 1 禁止 (Timer E Output 1 disable)
请参见 TA1ODIS 说明

位 7 **TD2ODIS**：定时器 D 输出 2 禁止 (Timer D Output 2 disable)
请参见 TA1ODIS 说明

位 6 **TD1ODIS**：定时器 D 输出 1 禁止 (Timer D Output 1 disable)
请参见 TA1ODIS 说明

位 5 **TC2ODIS**：定时器 C 输出 2 禁止 (Timer C Output 2 disable)
请参见 TA1ODIS 说明

位 4 **TC1ODIS**：定时器 C 输出 1 禁止 (Timer C Output 1 disable)
请参见 TA1ODIS 说明

位 3 **TB2ODIS**：定时器 B 输出 2 禁止 (Timer B Output 2 disable)
请参见 TA1ODIS 说明

位 2 **TB1ODIS**：定时器 B 输出 1 禁止 (Timer B Output 1 disable)
请参见 TA1ODIS 说明

位 1 **TA2ODIS**：定时器 A 输出 2 禁止 (Timer A Output 2 disable)
请参见 TA1ODIS 说明

位 0 **TA1ODIS**：定时器 A 输出 1 (HRTIM_CHA1) 禁止 (Timer A Output 1 (HRTIM_CHA1) disable)
将此位置 1 会禁止定时器 A 输出 1。输出从运行状态或故障状态进入空闲状态。
写入“0”无影响。

37.5.46 HRTIM 输出禁止状态寄存器 (HRTIM_ODSR)

HRTIM Output Disable Status Register

偏移地址：0x39Ch

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	TE2ODS	TE1ODS	TD2ODS	TD1ODS	TC2ODS	TC1ODS	TB2ODS	TB1ODS	TA2ODS	TA1ODS
						r	r	r	r	r	r	r	r	r	r

位 31:10 保留，必须保持复位值

位 9 **TE2ODS**：定时器 E 输出 2 禁止状态 (Timer E Output 2 disable status)
请参见 TA1ODS 说明

位 8 **TE1ODS**：定时器 E 输出 1 禁止状态 (Timer E Output 1 disable status)
请参见 TA1ODS 说明

位 7 **TD2ODS**：定时器 D 输出 2 禁止状态 (Timer D Output 2 disable status)
请参见 TA1ODS 说明

位 6 **TD1ODS**：定时器 D 输出 1 禁止状态 (Timer D Output 1 disable status)
请参见 TA1ODS 说明

位 5 **TC2ODS**：定时器 C 输出 2 禁止状态 (Timer C Output 2 disable status)
请参见 TA1ODS 说明

位 4 **TC1ODS**：定时器 C 输出 1 禁止状态 (Timer C Output 1 disable status)
请参见 TA1ODS 说明

位 3 **TB2ODS**：定时器 B 输出 2 禁止状态 (Timer B Output 2 disable status)
请参见 TA1ODS 说明

位 2 **TB1ODS**：定时器 B 输出 1 禁止状态 (Timer B Output 1 disable status)
请参见 TA1ODS 说明

位 1 **TA2ODS**：定时器 A 输出 2 禁止状态 (Timer A Output 2 disable status)
请参见 TA1ODS 说明

位 0 **TA1ODS**：定时器 A 输出 1 禁止状态 (Timer A Output 1 disable status)
读取此位会返回输出禁止状态。输出有效时（Tx1OEN 或 Tx2OEN = 1），此位无意义。
0：空闲状态下禁止输出 HRTIM_CHA1。
1：故障状态下禁止输出 HRTIM_CHA1。

37.5.47 HRTIM 突发模式控制寄存器 (HRTIM_BMCR)

HRTIM Burst Mode Control Register

偏移地址：0x3A0h

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
BMSTAT	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TEBM	TDBM	TCBM	TBBM	TABM	MTBM
rc_w0										rw	rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	BMPREN	BMPRSC[3:0]				BMCLK[3:0]				BMOM	BME
					rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31 **BMSTAT**：*突发模式状态 (Burst Mode Status)*

此位提供当前工作状态。

0：正常工作

1：正在进行突发工作。向此位写入 0 会使突发模式提前终止。

位 30:22 保留，必须保持复位值

位 21 **TEBM**：*定时器 E 突发模式 (Timer E Burst Mode)*

请参见 TABM 说明

位 20 **TDBM**：*定时器 D 突发模式 (Timer D Burst Mode)*

请参见 TABM 说明

位 19 **TCBM**：*定时器 C 突发模式 (Timer C Burst Mode)*

请参见 TABM 说明

位 18 **TBBM**：*定时器 B 突发模式 (Timer B Burst Mode)*

请参见 TABM 说明

位 17 **TABM**：*定时器 A 突发模式 (Timer A Burst Mode)*

此位定义定时器在突发模式工作期间的行为。突发模式使能后，不能更改此位域。

0：定时器 A 计数器时钟保持，定时器正常工作

1：定时器 A 计数器时钟停止，计数器复位

注： 当均衡空闲模式激活时 (DLYPRT[2:0] = 0x11)，不得将此位置 1

位 16 **MTBM**：*主定时器突发模式 (Master Timer Burst Mode)*

此位定义定时器在突发模式工作期间的行为。突发模式使能后，不能更改此位域。

0：主定时器计数器时钟保持，定时器正常工作

1：主定时器计数器时钟停止，计数器复位

位 15:11 保留，必须保持复位值

位 10 **BMPREN**：*突发模式预装载使能 (Burst Mode Preload Enable)*

此位可使能寄存器预装载机制，并定义对可预装载寄存器（HRTIM_BMCMPR、HRTIM_BMPER）的写访问是在活动寄存器中进行还是在预装载寄存器中进行。

0：禁止预装载：直接对活动寄存器进行写访问

1：使能预装载：对预装载寄存器进行写访问

位 9:6 **BMPRSC[3:0]**：*突发模式预分频器 (Burst Mode Prescaler)*

定义突发模式控制器的 f_{HRTIM} 时钟预分频比。突发模式使能后，不能更改此位域。

0000：时钟未分频
0001：2 分频
0010：4 分频
0011：8 分频
0100：16 分频
0101：32 分频
0110：64 分频
0111：128 分频
1000：256 分频
1001：512 分频
1010：1024 分频
1011：2048 分频
1100：4096 分频
1101：8192 分频
1110：16384 分频
1111：32768 分频

位 5:2 **BMCLK[3:0]**：*突发模式时钟源 (Burst Mode Clock source)*

此位域定义突发模式计数器的时钟源。突发模式使能后，不能更改此位域（有关片上事件 1..4 的连接详细信息，请参见*表 300*）。

0000：主定时器计数器复位/翻转
0001：定时器 A 计数器复位/翻转
0010：定时器 B 计数器复位/翻转
0011：定时器 C 计数器复位/翻转
0100：定时器 D 计数器复位/翻转
0101：定时器 E 计数器复位/翻转
0110：片上事件 1 (hrtim_bm_ck1)，起着突发模式计数器时钟的作用
0111：片上事件 2 (hrtim_bm_ck2)，起着突发模式计数器时钟的作用
1000：片上事件 3 (hrtim_bm_ck3)，起着突发模式计数器时钟的作用
1001：片上事件 4 (hrtim_bm_ck4)，起着突发模式计数器时钟的作用
1010：预分频 f_{HRTIM} 时钟（根据 BMPRSC[3:0] 设置）
其他代码保留

位 1 **BMOM**：*突发模式工作模式 (Burst Mode operating mode)*

此位定义进入一次突发模式还是连续工作。

0：单发模式
1：连续工作

位 0 **BME**：*突发模式使能 (Burst Mode enable)*

此位用于启动已准备好接收启动触发信号的突发模式控制器。
向此位写入 0 会使突发模式提前终止。

0：禁止突发模式
1：使能突发模式

37.5.48 HRTIM 突发模式触发寄存器 (HRTIM_BMTRGR)

HRTIM Burst Mode Trigger Register

偏移地址：0x3A4h

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
OCHPEV	EEV8	EEV7	TDEEV8	TAEEV7	TECMP2	TECMP1	TEREP	TERST	TDCMP2	TDCMP1	TDREP	TDRST	TCCMP2	TCCMP1	TCREP
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TCRST	TBCMP2	TBCMP1	TBREP	TBRST	TACMP2	TACMP1	TAREP	TARST	MSTCMP4	MSTCMP3	MSTCMP2	MSTCMP1	MSTREP	MSTRST	SW
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31 **OCHPEV**：*片上事件 (On-chip Event)*
片上事件上升沿（请参见*突发模式触发*一节）会触发突发模式进入。

位 30 **EEV8**：外部事件 8 *(External Event 8)（应用 TIMD 过滤器）*
通过 TIMD 过滤器调节的外部事件 8 会启动突发模式工作。

位 29 **EEV7**：外部事件 7 *(External Event 7)（应用 TIMA 过滤器）*
通过 TIMA 过滤器调节的外部事件 7 会启动突发模式工作。

位 28 **TDEEV8**：*外部事件 8 后的定时器 D 周期 (Timer D period following External Event 8)*
外部事件 8（通过 TIMD 过滤器调节）后的定时器 D 周期会启动突发模式工作。

位 27 **TAEEV7**：*外部事件 7 后的定时器 A 周期 (Timer A period following External Event 7)*
外部事件 7（通过 TIMA 过滤器调节）后的定时器 A 周期会启动突发模式工作。

位 26 **TECMP2**：*定时器 E 比较 2 事件 (Timer E Compare 2 event)*
请参见 TACMP1 说明

位 25 **TECMP1**：*定时器 E 比较 1 事件 (Timer E Compare 1 event)*
请参见 TACMP1 说明

位 24 **TEREP**：*定时器 E 重复 (Timer E repetition)*
请参见 TAREP 说明

位 23 **TERST**：*定时器 E 计数器复位或翻转 (Timer E counter reset or roll-over)*
请参见 TARST 说明

位 22 **TDCMP2**：*定时器 D 比较 2 事件 (Timer D Compare 2 event)*
请参见 TACMP1 说明

位 21 **TDCMP1**：*定时器 D 比较 1 事件 (Timer D Compare 1 event)*
请参见 TACMP1 说明

位 20 **TDREP**：*定时器 D 重复 (Timer D repetition)*
请参见 TAREP 说明

位 19 **TDRST**：*定时器 D 计数器复位或翻转 (Timer D counter reset or roll-over)*
请参见 TARST 说明

位 18 **TCCMP2**：*定时器 C 比较 2 事件 (Timer C Compare 2 event)*
请参见 TACMP1 说明

位 17 **TCCMP1**：*定时器 C 比较 1 事件 (Timer C Compare 1 event)*
请参见 TACMP1 说明

位 16 **TCREP**：*定时器 C 重复 (Timer C repetition)*
请参见 TAREP 说明

位 15 **TCRST**：*定时器 C 计数器复位或翻转 (Timer C counter reset or roll-over)*
请参见 TARST 说明

位 14 **TBCMP2**：*定时器 B 比较 2 事件 (Timer B Compare 2 event)*
请参见 TACMP1 说明

位 13 **TBCMP1**：*定时器 B 比较 1 事件 (Timer B Compare 1 event)*
请参见 TACMP1 说明

位 12 **TBREP**：*定时器 B 重复 (Timer B repetition)*
请参见 TAREP 说明

位 11 **TBRST**：*定时器 B 计数器复位或翻转 (Timer B counter reset or roll-over)*
请参见 TARST 说明

位 10 **TACMP2**：*定时器 A 比较 2 事件 (Timer A Compare 2 event)*
请参见 TACMP1 说明

位 9 **TACMP1**：*定时器 A 比较 1 事件 (Timer A Compare 1 event)*
定时器 A 比较 1 事件将启动突发模式工作。

位 8 **TAREP**：*定时器 A 重复 (Timer A repetition)*
定时器 A 重复事件将启动突发模式工作。

位 7 **TARST**：*定时器 A 计数器复位或翻转 (Timer A counter reset or roll-over)*
定时器 A 复位或翻转事件将启动突发模式工作。

位 6 **MSTCMP4**：*主定时器比较 4 (Master Compare 4)*
请参见 MSTCMP1 说明

位 5 **MSTCMP3**：*主定时器比较 3 (Master Compare 3)*
请参见 MSTCMP1 说明

位 4 **MSTCMP2**：*主定时器比较 2 (Master Compare 2)*
请参见 MSTCMP1 说明

位 3 **MSTCMP1**：*主定时器比较 1 (Master Compare 1)*
主定时器比较 1 事件将启动突发模式工作。

位 2 **MSTREP**：*主定时器重复 (Master repetition)*
主定时器重复事件将启动突发模式工作。

位 1 **MSTRST**：*主定时器复位或翻转 (Master reset or roll-over)*
主定时器复位和翻转事件将启动突发模式工作。

位 0 **SW**：*软件启动 (Software start)*
此位由软件置 1，并由硬件自动复位。
如果此位置 1，则会立即启动突发模式工作。
如果突发模式未使能（BME 位复位），此位无效。

37.5.49 HRTIM 突发模式比较寄存器 (HRTIM_BMCMPR)

HRTIM Burst Mode Compare Register

偏移地址：0x3A8h

复位值：0x0000 0000

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

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

位 31:16 保留，必须保持复位值

位 15:0 **BMCMP[15:0]**：*突发模式比较值 (Burst Mode compare value)*

定义所选定时器处于空闲状态的周期数。

该寄存器保留预装载寄存器的内容，如果禁止预装载，则会保留活动寄存器的内容。

注： 如果使用未进行预分频的 f_{HRTIM} 时钟作为突发模式时钟源（BMCLK[3:0] = 1010 且 BMPRESC[3:0] = 0000），BMCMP[15:0] 不能设为 0x0000。

37.5.50 HRTIM 突发模式周期寄存器 (HRTIM_BMPER)

HRTIM Burst Mode Period Register

偏移地址：0x3ACh

复位值：0x0000 0000

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

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

位 31:16 保留，必须保持复位值

位 15:0 **BMPER[15:0]**：*突发模式周期 (Burst Mode Period)*

定义突发模式重复周期。

如果禁止预装载，则该寄存器保存预装载寄存器的内容或活动寄存器的内容。

注： 如果突发模式已使能，BMPER[15:0] 不得为空。

37.5.51 HRTIM 定时器外部事件控制寄存器 1 (HRTIM_EECR1)

HRTIM Timer External Event Control Register 1

偏移地址：0x3B0h

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	EE5FAST	EE5SNS[1:0]		EE5POL	EE5SRC[1:0]		EE4FAST	EE4SNS[1:0]		EE4POL	EE4SRC[1:0]		EE3FAST	EE3SNS[1]
		rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EE3SNS[0]	EE3POL	EE3SRC[1:0]		EE2FAST	EE2SNS[1:0]		EE2POL	EE2SRC[1:0]		EE1FAST	EE1SNS[1:0]		EE1POL	EE1SRC[1:0]	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:30 保留，必须保持复位值

位 29 **EE5FAST**：*外部事件 5 快速模式 (External Event 5 Fast mode)*
请参见 EE1FAST 说明

位 28:27 **EE5SNS[1:0]**：*外部事件 5 有效性 (External Event 5 Sensitivity)*
请参见 EE1SNS[1:0] 说明

位 26 **EE5POL**：*外部事件 5 极性 (External Event 5 Polarity)*
请参见 EE1POL 说明

位 25:24 **EE5SRC[1:0]**：*外部事件 5 源 (External Event 5 Source)*
请参见 EE1SRC[1:0] 说明

位 23 **EE4FAST**：*外部事件 4 快速模式 (External Event 4 Fast mode)*
请参见 EE1FAST 说明

位 22:21 **EE4SNS[1:0]**：*外部事件 4 有效性 (External Event 4 Sensitivity)*
请参见 EE1SNS[1:0] 说明

位 20 **EE4POL**：*外部事件 4 极性 (External Event 4 Polarity)*
请参见 EE1POL 说明

位 19:18 **EE4SRC[1:0]**：*外部事件 4 源 (External Event 4 Source)*
请参见 EE1SRC[1:0] 说明

位 17 **EE3FAST**：*外部事件 3 快速模式 (External Event 3 Fast mode)*
请参见 EE1FAST 说明

位 16:15 **EE3SNS[1:0]**：*外部事件 3 有效性 (External Event 3 Sensitivity)*
请参见 EE1SNS[1:0] 说明

位 14 **EE3POL**：*外部事件 3 极性 (External Event 3 Polarity)*
请参见 EE1POL 说明

位 13:12 **EE3SRC[1:0]**：*外部事件 3 源 (External Event 3 Source)*
请参见 EE1SRC[1:0] 说明

位 11 **EE2FAST**：*外部事件 2 快速模式 (External Event 2 Fast mode)*
请参见 EE1FAST 说明

位 10:9 **EE2SNS[1:0]**：*外部事件 2 有效性 (External Event 2 Sensitivity)*
请参见 EE1SNS[1:0] 说明

位 8 **EE2POL**：*外部事件 2 极性 (External Event 2 Polarity)*
请参见 EE1POL 说明

位 7:6 **EE2SRC[1:0]**：*外部事件 2 源 (External Event 2 Source)*
请参见 EE1SRC[1:0] 说明

位 5 **EE1FAST**：*外部事件 1 快速模式 (External Event 1 Fast mode)*
0：外部事件 1 会在作用于输出之前通过 HRTIM 逻辑进行重新同步，以增加 f_{HRTIM} 时钟相关延迟
1：外部事件异步作用于输出上（低延迟模式）
注： 使用了事件的计数器使能后（TxCEN 位置 1），不得修改此位。

位 4:3 **EE1SNS[1:0]**：*外部事件 1 有效性 (External Event 1 Sensitivity)*
00：由 EE1POL 位定义有效电平
01：上升沿，与 EE1POL 位值无关
10：下降沿，与 EE1POL 位值无关
11：上升沿和下降沿，与 EE1POL 位值无关

位 2 **EE1POL**：*外部事件 1 极性 (External Event 1 Polarity)*
仅当 EE1SNS[1:0] = 00 时，此位才有效。
0：外部事件高电平有效
1：外部事件低电平有效
注： 定时器 x 使能后，不能更改此参数。必须在 EE1FAST 位置 1 之前进行配置。

位 1:0 **EE1SRC[1:0]**：*外部事件 1 源 (External Event 1 Source)*
00：hrtim_evt11
01：hrtim_evt12
10：hrtim_evt13
11：hrtim_evt14
注： 定时器 x 使能后，不能更改此参数。必须在 EE1FAST 位置 1 之前进行配置。

37.5.52 HRTIM 定时器外部事件控制寄存器 2 (HRTIM_EECR2)

HRTIM Timer External Event Control Register 2

偏移地址：0x3B4h

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	EE10SNS[1:0]		EE10POL	EE10SRC[1:0]		Res.	EE9SNS[1:0]		EE9POL	EE9SRC[1:0]		Res.	EE8SNS[1]
			rw	rw	rw	rw	rw		rw	rw	rw	rw	rw		rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EE8SNS[0]	EE8POL	EE8SRC[1:0]		Res.	EE7SNS[1:0]		EE7POL	EE7SRC[1:0]		Res.	EE6SNS[1:0]		EE6POL	EE6SRC[1:0]	
rw	rw	rw	rw		rw	rw	rw	rw	rw		rw	rw	rw	rw	rw

位 31:29 保留，必须保持复位值

位 28:27 **EE10SNS[1:0]**：*外部事件 10 有效性 (External Event 10 Sensitivity)*
请参见 EE1SNS[1:0] 说明

位 26 **EE10POL**：*外部事件 10 极性 (External Event 10 Polarity)*
请参见 EE1POL 说明

位 25:24 **EE10SRC[1:0]**：*外部事件 10 源 (External Event 10 Source)*
请参见 EE1SRC[1:0] 说明

位 23 保留，必须保持复位值

位 22:21 **EE9SNS[1:0]**：*外部事件 9 有效性 (External Event 9 Sensitivity)*
请参见 EE1SNS[1:0] 说明

位 20 **EE9POL**：*外部事件 9 极性 (External Event 9 Polarity)*
请参见 EE1POL 说明

位 19:18 **EE9SRC[1:0]**：*外部事件 9 源 (External Event 9 Source)*
请参见 EE1SRC[1:0] 说明

位 17 保留，必须保持复位值

位 16:15 **EE8SNS[1:0]**：*外部事件 8 有效性 (External Event 8 Sensitivity)*
请参见 EE1SNS[1:0] 说明

位 14 **EE8POL**：*外部事件 8 极性 (External Event 8 Polarity)*
请参见 EE1POL 说明

位 13:12 **EE8SRC[1:0]**：*外部事件 8 源 (External Event 8 Source)*
请参见 EE1SRC[1:0] 说明

位 11 保留，必须保持复位值

位 10:9 **EE7SNS[1:0]**：*外部事件 7 有效性 (External Event 7 Sensitivity)*
请参见 EE1SNS[1:0] 说明

位 8 **EE7POL**：*外部事件 7 极性 (External Event 7 Polarity)*
请参见 EE1POL 说明

位 7:6 **EE7SRC[1:0]**：*外部事件 7 源 (External Event 7 Source)*
请参见 EE1SRC[1:0] 说明

位 5 保留，必须保持复位值

位 4:3 **EE6SNS[1:0]**：*外部事件 6 有效性 (External Event 6 Sensitivity)*
请参见 EE1SNS[1:0] 说明

位 2 **EE6POL**：*外部事件 6 极性 (External Event 6 Polarity)*
请参见 EE1POL 说明

位 1:0 **EE6SRC[1:0]**：*外部事件 6 源 (External Event 6 Source)*
请参见 EE1SRC[1:0] 说明

37.5.53 HRTIM 定时器外部事件控制寄存器 3 (HRTIM_EECR3)

HRTIM Timer External Event Control Register 3

偏移地址：0x3B8h

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
EEVSD[1:0]		Res.	Res.	EE10F[3:0]				Res.	Res.	EE9F[3:0]				Res.	Res.
rw	rw			rw	rw	rw	rw			rw	rw	rw	rw		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EE8F[3:0]				Res.	Res.	EE7F[3:0]				Res.	Res.	EE6F[3:0]			
rw	rw	rw	rw			rw	rw	rw	rw			rw	rw	rw	rw

位 31:30 **EEVSD[1:0]**：外部事件采样时钟分频比 (External Event Sampling clock division)
此位域指示定时器时钟频率 (f_{HRTIM}) 与数字滤波器使用的外部事件信号采样时钟 (f_{EEVS}) 之间的分频比。
00：$f_{EEVS}=f_{HRTIM}$
01：$f_{EEVS}=f_{HRTIM}$ / 2
10：$f_{EEVS}=f_{HRTIM}$ / 4
11：$f_{EEVS}=f_{HRTIM}$ / 8

位 29:28 保留，必须保持复位值

位 27:24 **EE10F[3:0]**：外部事件 10 过滤器 (External Event 10 filter)
请参见 EE6F[3:0] 说明

位 23:22 保留，必须保持复位值

位 21:18 **EE9F[3:0]**：外部事件 9 过滤器 (External Event 9 filter)
请参见 EE6F[3:0] 说明

位 17:16 保留，必须保持复位值

位 15:12 **EE8F[3:0]**：外部事件 8 过滤器 (External Event 8 filter)
请参见 EE6F[3:0] 说明

位 11:10 保留，必须保持复位值

位 9:6 **EE7F[3:0]**：外部事件 7 过滤器 (External Event 7 filter)
请参见 EE6F[3:0] 说明

位 4:5 保留，必须保持复位值

位 3:0 **EE6F[3:0]**：外部事件 6 过滤器 (External Event 6 filter)
此位域可定义外部事件 6 输入的采样频率和应用于 hrtim_evt6 的数字滤波器长度。数字滤波器由计数器组成，需要使用 N 个有效样本来验证输出跳变。
0000：禁止滤波器
0001：$f_{SAMPLING}=f_{HRTIM}$，N=2
0010：$f_{SAMPLING}=f_{HRTIM}$，N=4
0011：$f_{SAMPLING}=f_{HRTIM}$，N=8
0100：$f_{SAMPLING}=f_{EEVS/2}$，N=6
0101：$f_{SAMPLING}=f_{EEVS/2}$，N=8
0110：$f_{SAMPLING}=f_{EEVS/4}$，N=6
0111：$f_{SAMPLING}=f_{EEVS/4}$，N=8
1000：$f_{SAMPLING}=f_{EEVS/8}$，N=6
1001：$f_{SAMPLING}=f_{EEVS/8}$，N=8
1010：$f_{SAMPLING}=f_{EEVS/16}$，N=5
1011：$f_{SAMPLING}=f_{EEVS/16}$，N=6
1100：$f_{SAMPLING}=f_{EEVS/16}$，N=8
1101：$f_{SAMPLING}=f_{EEVS/32}$，N=5
1110：$f_{SAMPLING}=f_{EEVS/32}$，N=6
1111：$f_{SAMPLING}=f_{EEVS/32}$，N=8

37.5.54 HRTIM ADC 触发 1 寄存器 (HRTIM_ADC1R)

HRTIM ADC Trigger 1 Register

偏移地址：0x3BCh

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
AD1TEPER	AD1TEC4	AD1TEC3	AD1TEC2	AD1TDPER	AD1TDC4	AD1TDC3	AD1TDC2	AD1TCPER	AD1TCC4	AD1TCC3	AD1TCC2	AD1TBRST	AD1TBPER	AD1TBC4	AD1TBC3
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AD1TBC2	AD1TARST	AD1TAPER	AD1TAC4	AD1TAC3	AD1TAC2	AD1EEV5	AD1EEV4	AD1EEV3	AD1EEV2	AD1EEV1	AD1MPER	AD1MC4	AD1MC3	AD1MC2	AD1MC1
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:0 这些位用于选择 ADC 触发 1 输出 (hrtim_adc_trg1) 的触发源。有关详细信息，请参见 HRTIM_ADC3R 位说明。

37.5.55 HRTIM ADC 触发 2 寄存器 (HRTIM_ADC2R)

HRTIM ADC Trigger 2 Register

偏移地址：0x3C0h

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
AD2TERST	AD2TEC4	AD2TEC3	AD2TEC2	AD2TDRST	AD2TDPER	AD2TDC4	AD2TDC3	AD2TDC2	AD2TCRST	AD2TCPER	AD2TCC4	AD2TCC3	AD2TCC2	AD2TBPER	AD2TBC4
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AD2TBC3	AD2TBC2	AD2TAPER	AD2TAC4	AD2TAC3	AD2TAC2	AD2EEV10	AD2EEV9	AD2EEV8	AD2EEV7	AD2EEV6	AD2MPER	AD2MC4	AD2MC3	AD2MC2	AD2MC1
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:0 这些位用于选择 ADC 触发 2 输出 (hrtim_adc_trg2) 的触发源。有关详细信息，请参见 HRTIM_ADC4R 位说明

37.5.56 HRTIM ADC 触发 3 寄存器 (HRTIM_ADC3R)

HRTIM ADC Trigger 3 Register

偏移地址：0x3C4h

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ADC3 TEPER	ADC3T EC4	ADC3T EC3	ADC3T EC2	ADC3T DPER	ADC3T DC4	ADC3T DC3	ADC3T DC2	ADC3T CPER	ADC3T CC4	ADC3T CC3	ADC3T CC2	ADC3T BRST	ADC3T BPER	ADC3T BC4	ADC3T BC3
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADC3T BC2	ADC3T ARST	ADC3T APER	ADC3T AC4	ADC3T AC3	ADC3T AC2	ADC3E EV5	ADC3E EV4	ADC3E EV3	ADC3E EV2	ADC3E EV1	ADC3M PER	ADC3M C4	ADC3M C3	ADC3M C2	ADC3M C1
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31 **ADC3TEPER**：*定时器 E 周期时的 ADC 触发 3 (ADC trigger 3 on Timer E Period)*
请参见 ADC3TAPER 说明

位 30 **ADC3TEC4**：*定时器 E 比较 4 时的 ADC 触发 3 (ADC trigger 3 on Timer E Compare 4)*
请参见 ADC3TAC2 说明

位 29 **ADC3TEC3**：*定时器 E 比较 3 时的 ADC 触发 3 (ADC trigger 3 on Timer E Compare 3)*
请参见 ADC3TAC2 说明

位 28 **ADC3TEC2**：*定时器 E 比较 2 时的 ADC 触发 3 (ADC trigger 3 on Timer E Compare 2)*
请参见 ADC3TAC2 说明

位 27 **ADC3TDPER**：*定时器 D 周期时的 ADC 触发 3 (ADC trigger 3 on Timer D Period)*
请参见 ADC3TAPER 说明

位 26 **ADC3TDC4**：*定时器 D 比较 4 时的 ADC 触发 3 (ADC trigger 3 on Timer D Compare 4)*
请参见 ADC3TAC2 说明

位 25 **ADC3TDC3**：*定时器 D 比较 3 时的 ADC 触发 3 (ADC trigger 3 on Timer D Compare 3)*
请参见 ADC3TAC2 说明

位 24 **ADC3TDC2**：*定时器 D 比较 2 时的 ADC 触发 3 (ADC trigger 3 on Timer D Compare 2)*
请参见 ADC3TAC2 说明

位 23 **ADC3TCPER**：*定时器 C 周期时的 ADC 触发 3 (ADC trigger 3 on Timer C Period)*
请参见 ADC3TAPER 说明

位 22 **ADC3TCC4**：*定时器 C 比较 4 时的 ADC 触发 3 (ADC trigger 3 on Timer C Compare 4)*
请参见 ADC3TAC2 说明

位 21 **ADC3TCC3**：*定时器 C 比较 3 时的 ADC 触发 3 (ADC trigger 3 on Timer C Compare 3)*
请参见 ADC3TAC2 说明

位 20 **ADC3TCC2**：*定时器 C 比较 2 时的 ADC 触发 3 (ADC trigger 3 on Timer C Compare 2)*
请参见 ADC3TAC2 说明

位 19 **ADC3TBRST**：*定时器 B 复位和计数器翻转时的 ADC 触发 3 (ADC trigger 3 on Timer B Reset and counter roll-over)*
请参见 ADC3TBRST 说明

位 18 **ADC3TBPER**：*定时器 B 周期时的 ADC 触发 3 (ADC trigger 3 on Timer B Period)*
请参见 ADC3TAPER 说明

位 17 **ADC3TBC4**：*定时器 B 比较 4 时的 ADC 触发 3 (ADC trigger 3 on Timer B Compare 4)*
请参见 ADC3TAC2 说明

位 16 **ADC3TBC3**：*定时器 B 比较 3 时的 ADC 触发 3 (ADC trigger 3 on Timer B Compare 3)*
请参见 ADC3TAC2 说明

位 15 **ADC3TBC2**：*定时器 B 比较 2 时的 ADC 触发 3 (ADC trigger 3 on Timer B Compare 2)*
请参见 ADC3TAC2 说明

位 14 **ADC3TARST**：*定时器 A 复位和计数器翻转时的 ADC 触发 3 (ADC trigger 3 on Timer A Reset and counter roll-over)*
此位可在发生定时器 A 复位和翻转事件时在 ADC 触发 1 输出上生成 ADC 触发信号。

位 13 **ADC3TAPER**：*定时器 A 周期时的 ADC 触发 3 (ADC trigger 3 on Timer A Period)*
此位可在发生定时器 A 周期事件时在 ADC 触发 3 输出 (hrtim_adc_trg3) 上生成 ADC 触发信号。

位 12 **ADC3TAC4**：*定时器 A 比较 4 时的 ADC 触发 3 (ADC trigger 3 on Timer A Compare 4)*
请参见 ADC3TAC2 说明

位 11 **ADC3TAC3**：*定时器 A 比较 3 时的 ADC 触发 3 (ADC trigger 4 on Timer A Compare 3)*
请参见 ADC3TAC2 说明

位 10 **ADC3TAC2**：*定时器 A 比较 2 时的 ADC 触发 3 (ADC trigger 3 on Timer A Compare 2)*
此位可在发生定时器 A 比较 2 事件时在 ADC 触发 3 输出 (hrtim_adc_trg3) 上生成 ADC 触发信号。

位 9 **ADC3EEV5**：*外部事件 5 时的 ADC 触发 3 (ADC trigger 3 on External Event 5)*
请参见 ADC3EEV1 说明

位 8 **ADC3EEV4**：*外部事件 4 时的 ADC 触发 3 (ADC trigger 3 on External Event 4)*
请参见 ADC3EEV1 说明

位 7 **ADC3EEV3**：*外部事件 3 时的 ADC 触发 3 (ADC trigger 3 on External Event 3)*
请参见 ADC3EEV1 说明

位 6 **ADC3EEV2**：*外部事件 2 时的 ADC 触发 3 (ADC trigger 3 on External Event 2)*
请参见 ADC3EEV1 说明

位 5 **ADC3EEV1**：*外部事件 1 时的 ADC 触发 3 (ADC trigger 3 on External Event 1)*
此位可在发生外部事件 1 时在 ADC 触发 3 输出 (hrtim_adc_trg3) 上生成 ADC 触发信号。

位 4 **ADC3MPER**：*主定时器周期时的 ADC 触发 3 (ADC trigger 3 on Master Period)*
此位可在发生主定时器周期事件时在 ADC 触发 3 输出 (hrtim_adc_trg3) 上生成 ADC 触发信号。

位 3 **ADC3MC4**：*主定时器比较 4 时的 ADC 触发 3 (ADC trigger 3 on Master Compare 4)*
请参见 ADC3MC1 说明

位 2 **ADC3MC3**：*主定时器比较 3 时的 ADC 触发 3 (ADC trigger 3 on Master Compare 3)*
请参见 ADC3MC1 说明

位 1 **ADC3MC2**：*主定时器比较 2 时的 ADC 触发 3 (ADC trigger 3 on Master Compare 2)*
请参见 ADC3MC1 说明

位 0 **ADC3MC1**：*主定时器比较 1 时的 ADC 触发 3 (ADC trigger 3 on Master Compare 1)*
此位可在发生主定时器比较 1 事件时在 ADC 触发 3 输出 (hrtim_adc_trg3) 上生成 ADC 触发信号。

37.5.57 HRTIM ADC 触发 4 寄存器 (HRTIM_ADC4R)

HRTIM ADC Trigger 4 Register

偏移地址：0x3C8h

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ADC4TERST	ADC4TEC4	ADC4TEC3	ADC4TEC2	ADC4TDRST	ADC4TDPER	ADC4TDC4	ADC4TDC3	ADC4TDC2	ADC4TCRST	ADC4TCPER	ADC4TCC4	ADC4TCC3	ADC4TCC2	ADC4TBPER	ADC4TBC4
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADC4TBC3	ADC4TBC2	ADC4TAPER	ADC4TAC4	ADC4TAC3	ADC4TAC2	ADC4EEV10	ADC4EEV9	ADC4EEV8	ADC4EEV7	ADC4EEV6	ADC4MPER	ADC4MC4	ADC4MC3	ADC4MC2	ADC4MC1
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31 **ADC4TERST**：*定时器 E 复位和计数器翻转时的 ADC 触发 4 (ADC trigger 4 on Timer E Reset and counter roll-over)* (1)

请参见 ADC4TCRST 说明

位 30 **ADC4TEC4**：*定时器 E 比较 4 时的 ADC 触发 4 (ADC trigger 4 on Timer E Compare 4)*

请参见 ADC4TAC2 说明

位 29 **ADC4TEC3**：*定时器 E 比较 3 时的 ADC 触发 4 (ADC trigger 4 on Timer E Compare 3)*

请参见 ADC4TAC2 说明

位 28 **ADC4TEC2**：*定时器 E 比较 2 时的 ADC 触发 4 (ADC trigger 4 on Timer E Compare 2)*

请参见 ADC4TAC2 说明

位 27 **ADC4TDRST**：*定时器 D 复位和计数器翻转时的 ADC 触发 4 (ADC trigger 4 on Timer D Reset and counter roll-over)*(1)

请参见 ADC4TCRST 说明

位 26 **ADC4TDPER**：*定时器 D 周期时的 ADC 触发 4 (ADC trigger 4 on Timer D Period)*

请参见 ADC4TAPER 说明

位 25 **ADC4TDC4**：*定时器 D 比较 4 时的 ADC 触发 4 (ADC trigger 4 on Timer D Compare 4)*

请参见 ADC4TAC2 说明

位 24 **ADC4TDC3**：*定时器 D 比较 3 时的 ADC 触发 4 (ADC trigger 4 on Timer D Compare 3)*

请参见 ADC4TAC2 说明

位 23 **ADC4TDC2**：*定时器 D 比较 2 时的 ADC 触发 4 (ADC trigger 4 on Timer D Compare 2)*

请参见 ADC4TAC2 说明

位 22 **ADC4TCRST**：*定时器 C 复位和计数器翻转时的 ADC 触发 4 (ADC trigger 4 on Timer C Reset and counter roll-over)*(1)

此位可在发生定时器 C 复位和翻转事件时在 ADC 触发 4 输出 (hrtim_adc_trg4) 上生成 ADC 触发信号。

位 21 **ADC4TCPER**：*定时器 C 周期时的 ADC 触发 4 (ADC trigger 4 on Timer C Period)*

请参见 ADC4TAPER 说明

位 20 **ADC4TCC4**：*定时器 C 比较 4 时的 ADC 触发 4 (ADC trigger 4 on Timer C Compare 4)*

请参见 ADC4TAC2 说明

位 19 **ADC4TCC3**：*定时器 C 比较 3 时的 ADC 触发 4 (ADC trigger 4 on Timer C Compare 3)*

请参见 ADC4TAC2 说明

位 18 **ADC4TCC2**：*定时器 C 比较 2 时的 ADC 触发 4 (ADC trigger 4 on Timer C Compare 2)*
请参见 ADC4TAC2 说明

位 17 **ADC4TBPER**：*定时器 B 周期时的 ADC 触发 4 (ADC trigger 4 on Timer B Period)*
请参见 ADC4TAPER 说明

位 16 **ADC4TBC4**：*定时器 B 比较 4 时的 ADC 触发 4 (ADC trigger 4 on Timer B Compare 4)*
请参见 ADC4TAC2 说明

位 15 **ADC4TBC3**：*定时器 B 比较 3 时的 ADC 触发 4 (ADC trigger 4 on Timer B Compare 3)*
请参见 ADC4TAC2 说明

位 14 **ADC4TBC2**：*定时器 B 比较 2 时的 ADC 触发 4 (ADC trigger 4 on Timer B Compare 2)*
请参见 ADC4TAC2 说明

位 13 **ADC4TAPER**：*定时器 A 周期时的 ADC 触发 4 (ADC trigger 4 on Timer A Period)*
此位可在发生定时器 A 事件时在 ADC 触发 4 输出 (hrtim_adc_trg4) 上生成 ADC 触发信号。

位 12 **ADC4TAC4**：*定时器 A 比较 4 时的 ADC 触发 4 (ADC trigger 4 on Timer A Compare 4)*
请参见 ADC4TAC2 说明

位 11 **ADC4TAC3**：*定时器 A 比较 3 时的 ADC 触发 4 (ADC trigger 4 on Timer A Compare 3)*
请参见 ADC4TAC2 说明

位 10 **ADC4TAC2**：*定时器 A 比较 2 时的 ADC 触发 4 (ADC trigger 4 on Timer A Compare 2)*
此位可在发生定时器 A 比较 2 时在 ADC 触发 4 输出 (hrtim_adc_trg4) 上生成 ADC 触发信号。

位 9 **ADC4EEV10**：*外部事件 10 时的 ADC 触发 4 (ADC trigger 4 on External Event 10)*[1]
请参见 ADC4EEV6 说明

位 8 **ADC4EEV9**：*外部事件 9 时的 ADC 触发 4 (ADC trigger 4 on External Event 9)*[1]
请参见 ADC4EEV6 说明

位 7 **ADC4EEV8**：*外部事件 8 时的 ADC 触发 4 (ADC trigger 4 on External Event 8)*[1]
请参见 ADC4EEV6 说明

位 6 **ADC4EEV7**：*外部事件 7 时的 ADC 触发 4 (ADC trigger 4 on External Event 7)*[1]
请参见 ADC4EEV6 说明

位 5 **ADC4EEV6**：*外部事件 6 时的 ADC 触发 4 (ADC trigger 4 on External Event 6)*[1]
此位可在发生外部事件 6 时在 ADC 触发 4 输出 (hrtim_adc_trg4) 上生成 ADC 触发信号。

位 4 **ADC4MPER**：*主定时器周期时的 ADC 触发 4 (ADC trigger 4 on Master Period)*
此位可在发生主定时器周期事件时在 ADC 触发 4 输出 (hrtim_adc_trg4) 上生成 ADC 触发信号。

位 3 **ADC4MC4**：*主定时器比较 4 时的 ADC 触发 4 (ADC trigger 4 on Master Compare 4)*
请参见 ADC4MC1 说明

位 2 **ADC4MC3**：*主定时器比较 3 时的 ADC 触发 4 (ADC trigger 4 on Master Compare 3)*
请参见 ADC4MC1 说明

位 1 **ADC4MC2**：*主定时器比较 2 时的 ADC 触发 4 (ADC trigger 4 on Master Compare 2)*
请参见 ADC4MC1 说明

位 0 **ADC4MC1**：*主定时器比较 1 时的 ADC 触发 4 (ADC trigger 4 on Master Compare 1)*
此位可在发生主定时器比较 1 事件时在 ADC 触发 4 输出 (hrtim_adc_trg4) 上生成 ADC 触发信号。

1. 这些触发信号与 HRTIM_ADC1R/HRTIM_ADC3R 到 HRTIM_ADC2R/HRTIM_ADC4R 不同。

37.5.58 HRTIM 故障输入寄存器 1 (HRTIM_FLTINR1)

HRTIM Fault Input Register 1

偏移地址：0x3D0h

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FLT4LCK	FLT4F[3:0]				FLT4SRC	FLT4P	FLT4E	FLT3LCK	FLT3F[3:0]				FLT3SRC	FLT3P	FLT3E
rwo	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FLT2LCK	FLT2F[3:0]				FLT2SRC	FLT2P	FLT2E	FLT1LCK	FLT1F[3:0]				FLT1SRC	FLT1P	FLT1E
rwo	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31 **FLT4LCK**：故障 4 锁定 (Fault 4 Lock)
请参见 HRTIM_FLTINR2 寄存器中 FLT5LCK 的说明

位 30:27 **FLT4F[3:0]**：故障 4 过滤器 (Fault 4 filter)
请参见 HRTIM_FLTINR2 寄存器中 FLT5F[3:0] 的说明

位 26 **FLT4SRC**：故障 4 源 (Fault 4 source)
请参见 HRTIM_FLTINR2 寄存器中 FLT5SRC 的说明

位 25 **FLT4P**：故障 4 极性 (Fault 4 polarity)
请参见 HRTIM_FLTINR2 寄存器中 FLT5P 的说明

位 24 **FLT4E**：故障 4 使能 (Fault 4 enable)
请参见 HRTIM_FLTINR2 寄存器中 FLT5E 的说明

位 23 **FLT3LCK**：故障 3 锁定 (Fault 3 Lock)
请参见 HRTIM_FLTINR2 寄存器中 FLT5LCK 的说明

位 22:19 **FLT3F[3:0]**：故障 3 过滤器 (Fault 3 filter)
请参见 HRTIM_FLTINR2 寄存器中 FLT5F[3:0] 的说明

位 18 **FLT3SRC**：故障 3 源 (Fault 3 source)
请参见 HRTIM_FLTINR2 寄存器中 FLT5SRC 的说明

位 17 **FLT3P**：故障 3 极性 (Fault 3 polarity)
请参见 HRTIM_FLTINR2 寄存器中 FLT5P 的说明

位 16 **FLT3E**：故障 3 使能 (Fault 3 enable)
请参见 HRTIM_FLTINR2 寄存器中 FLT5E 的说明

位 15 **FLT2LCK**：故障 2 锁定 (Fault 2 Lock)
请参见 HRTIM_FLTINR2 寄存器中 FLT5LCK 的说明

位 14:11 **FLT2F[3:0]**：故障 2 过滤器 (Fault 2 filter)
请参见 HRTIM_FLTINR2 寄存器中 FLT5F[3:0] 的说明

位 10 **FLT2SRC**：故障 2 源 (Fault 2 source)
请参见 HRTIM_FLTINR2 寄存器中 FLT5SRC 的说明

位 9 **FLT2P**：故障 2 极性 (Fault 2 polarity)
请参见 HRTIM_FLTINR2 寄存器中 FLT2P 的说明

位 8 **FLT2E**：故障 2 使能 (Fault 2 enable)
请参见 HRTIM_FLTINR2 寄存器中 FLT5E 的说明

位 7 **FLT1LCK**：故障 1 锁定 (Fault 1 Lock)
请参见 HRTIM_FLTINR2 寄存器中 FLT5LCK 的说明

位 6:3 **FLT1F[3:0]**：故障 1 过滤器 (Fault 1 filter)
请参见 HRTIM_FLTINR2 寄存器中 FLT5F[3:0] 的说明

位 2 **FLT1SRC**：故障 1 源 (Fault 1 source)
请参见 HRTIM_FLTINR2 寄存器中 FLT5SRC 的说明

位 1 **FLT1P**：故障 1 极性 (Fault 1 polarity)
请参见 HRTIM_FLTINR2 寄存器中 FLT5P 的说明

位 0 **FLT1E**：故障 1 使能 (Fault 1 enable)
请参见 HRTIM_FLTINR2 寄存器中 FLT5E 的说明

37.5.59 HRTIM 故障输入寄存器 2 (HRTIM_FLTINR2)

HRTIM Fault Input Register 2

偏移地址：0x3D4h

复位值：0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FLT5LCK	FLT5F[3:0]				FLT5SRC	FLT5P	FLT5E
								rwo	rw	rw	rw	rw	rw	rw	rw

位 31:26 保留，必须保持复位值

位 25:24 **FLTSD[1:0]**：故障采样时钟分频比 (Fault Sampling clock division)

此位域指示定时器时钟频率 (f_{HRTIM}) 与数字滤波器使用的故障信号采样时钟 (f_{FLTS}) 之间的分频比。

00：$f_{FLTS}=f_{HRTIM}$

01：$f_{FLTS}=f_{HRTIM}$ / 2

10：$f_{FLTS}=f_{HRTIM}$ / 4

11：$f_{FLTS}=f_{HRTIM}$ / 8

注： 此位域必须先于任何 FLTxE 使能位写入。

位 23:8 保留，必须保持复位值

位 7 **FLT5LCK**：故障 5 锁定 (Fault 5 Lock)

FLT5LCK 位会修改故障编程位的写属性，以防止误写访问修改写属性。

该位仅可写入一次。该位置 1 后，在下一次系统复位之前不能进行修改。

0：FLT5E、FLT5P、FLT5SRC、FLT5F[3:0] 位为读/写位。

1：FLT5E、FLT5P、FLT5SRC、FLT5F[3:0] 不再可写入（只读模式）

位 6:3 **FLT5F[3:0]**：故障 5 过滤器 (Fault 5 filter)

此位域可定义 FLT5 输入的采样频率和应用于 FLT5 的数字滤波器长度。数字滤波器由事件计数器组成，每 N 个事件才视为一个有效边沿：

0000：无滤波器，FLT5 异步工作

0001：$f_{SAMPLING} = f_{HRTIM}$，N = 2

0010：$f_{SAMPLING} = f_{HRTIM}$，N = 4

0011：$f_{SAMPLING} = f_{HRTIM}$，N = 8

0100：$f_{SAMPLING} = f_{FLTS}/2$，N = 6

0101：$f_{SAMPLING} = f_{FLTS}$，N = 8

0110：$f_{SAMPLING} = f_{FLTS}/4$，N = 6

0111：$f_{SAMPLING} = f_{FLTS}/4$，N = 8

1000：$f_{SAMPLING} = f_{FLTS}/8$，N = 6

1001：$f_{SAMPLING} = f_{FLTS}/8$，N = 8

1010：$f_{SAMPLING} = f_{FLTS}/16$，N = 5

1011：$f_{SAMPLING} = f_{FLTS}/16$，N = 6

1100：$f_{SAMPLING} = f_{FLTS}/16$，N = 8

1101：$f_{SAMPLING} = f_{FLTS}/32$，N = 5

1110：$f_{SAMPLING} = f_{FLTS}/32$，N = 6

1111：$f_{SAMPLING} = f_{FLTS}/32$，N = 8

注： 仅当 FLT5E 使能位复位时，才可写入该位域。

如果 FLT5LOCK 已编程，则不能修改该位域。

位 2 **FLT5SRC**：故障 5 源 (Fault 5 source)

此位用于选择 FAULT5 输入源（有关连接详细信息，请参见*表 301*）。

0：故障 1 输入为 HRTIM_FLT5 输入引脚

1：故障 1 输入为 hrtim_in_flt5 信号

注： 仅当 FLT5E 使能位复位时，才可写入该位域

位 1 **FLT5P**：故障 5 极性 (Fault 5 polarity)

此位用于选择 FAULT5 输入极性。

0：故障 5 输入低电平有效

1：故障 5 输入高电平有效

注： 仅当 FLT5E 使能位复位时，才可写入该位域

位 0 **FLT5E**：故障 5 使能 (Fault 5 enable)

此位用于使能全局 FAULT5 输入电路。

0：禁止故障 5 输入

1：使能故障 5 输入

37.5.60 HRTIM 突发 DMA 主定时器更新寄存器 (HRTIM_BDMUPR)

HRTIM Burst DMA Master timer update Register

偏移地址：0x3D8h

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	MCMP4	MCMP3	MCMP2	MCMP1	MREP	MPER	MCNT	MDIER	MICR	MCR
						rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:10 保留，必须保持复位值

位 9 **MCMP4**：MCMP4R 寄存器更新使能 (MCMP4R register update enable)
请参见 MCR 说明

位 8 **MCMP3**：*MCMP3R 寄存器更新使能 (MCMP3R register update enable)*
请参见 MCR 说明

位 7 **MCMP2**：*MCMP2R 寄存器更新使能 (MCMP2R register update enable)*
请参见 MCR 说明

位 6 **MCMP1**：*MCMP1R 寄存器更新使能 (MCMP1R register update enable)*
请参见 MCR 说明

位 5 **MREP**：*MREP 寄存器更新使能 (MREP register update enable)*
请参见 MCR 说明

位 4 **MPER**：*MPER 寄存器更新使能 (MPER register update enable)*
请参见 MCR 说明

位 3 **MCNT**：*MCNTR 寄存器更新使能 (MCNTR register update enable)*
请参见 MCR 说明

位 2 **MDIER**：*MDIER 寄存器更新使能 (MDIER register update enable)*
请参见 MCR 说明

位 1 **MICR**：*MICR 寄存器更新使能 (MICR register update enable)*
请参见 MCR 说明

位 0 **MCR**：*MCR 寄存器更新使能 (MCR register update enable)*
此位用于定义主定时器 MCR 寄存器是否包含在要通过突发 DMA 更新的寄存器列表中。
0：MCR 寄存器不会通过突发 DMA 访问更新
1：MCR 寄存器会通过突发 DMA 访问更新

37.5.61 HRTIM 突发 DMA Timerx 更新寄存器 (HRTIM_BDTxUPR)

HRTIM Burst DMA Timerx update Register

偏移地址：0x3DCh-0x3ECh

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TIMxFLTR	TIMxOUTR	TIMxCHPR	TIMxRSTR	TIMxEEFR2
											rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TIMxEEFR1	TIMxRST2R	TIMxSET2R	TIMxRST1R	TIMxSET1R	TIMxDTxR	TIMxCMP4	TIMxCMP3	TIMxCMP2	TIMxCMP1	TIMxREP	TIMxPER	TIMxCNT	TIMxDIER	TIMxICR	TIMxCR
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:21 保留，必须保持复位值

位 20 **TIMxFLTR**：HRTIM_FLTxR 寄存器更新使能 (HRTIM_FLTxR register update enable)
请参见 TIMxCR 说明

位 19 **TIMxOUTR**：HRTIM_OUTxR 寄存器更新使能 (HRTIM_OUTxR register update enable)
请参见 TIMxCR 说明

位 18 **TIMxCHPR**：HRTIM_CHPxR 寄存器更新使能 (HRTIM_CHPxR register update enable)
请参见 TIMxCR 说明

位 17 **TIMxRSTR**：HRTIM_RSTxR 寄存器更新使能 (HRTIM_RSTxR register update enable)
请参见 TIMxCR 说明

位 16 **TIMxEEFR2**：HRTIM_EEFxR2 寄存器更新使能 (HRTIM_EEFxR2 register update enable)
请参见 TIMxCR 说明

位 15 **TIMxEEFR1**：HRTIM_EEFxR1 寄存器更新使能 (HRTIM_EEFxR1 register update enable)
请参见 TIMxCR 说明

位 14 **TIMxRST2R**：HRTIM_RST2xR 寄存器更新使能 (HRTIM_RST2xR register update enable)
请参见 TIMxCR 说明

位 13 **TIMxSET2R**：HRTIM_SET2xR 寄存器更新使能 (HRTIM_SET2xR register update enable)
请参见 TIMxCR 说明

位 12 **TIMxRST1R**：HRTIM_RST1xR 寄存器更新使能 (HRTIM_RST1xR register update enable)
请参见 TIMxCR 说明

位 11 **TIMxSET1R**：HRTIM_SET1xR 寄存器更新使能 (HRTIM_SET1xR register update enable)
请参见 TIMxCR 说明

位 10 **TIMxDTR**：HRTIM_DTxR 寄存器更新使能 (HRTIM_DTxR register update enable)
请参见 TIMxCR 说明

位 9 **TIMxCMP4**：HRTIM_CMP4xR 寄存器更新使能 (HRTIM_CMP4xR register update enable)
请参见 TIMxCR 说明

位 8 **TIMxCMP3**：HRTIM_CMP3xR 寄存器更新使能 (HRTIM_CMP3xR register update enable)
请参见 TIMxCR 说明

位 7 **TIMxCMP2**：HRTIM_CMP2xR 寄存器更新使能 (HRTIM_CMP2xR register update enable)
请参见 TIMxCR 说明

位 6 **TIMxCMP1**：HRTIM_CMP1xR 寄存器更新使能 (HRTIM_CMP1xR register update enable)
请参见 TIMxCR 说明

位 5 **TIMxREP**：HRTIM_REPxR 寄存器更新使能 (HRTIM_REPxR register update enable)
请参见 TIMxCR 说明

位 4 **TIMxPER**：HRTIM_PERxR 寄存器更新使能 (HRTIM_PERxR register update enable)
请参见 TIMxCR 说明

位 3 **TIMxCNT**：HRTIM_CNTxR 寄存器更新使能 (HRTIM_CNTxR register update enable)
请参见 TIMxCR 说明

位 2 **TIMxDIER**：HRTIM_TIMxDIER 寄存器更新使能 (HRTIM_TIMxDIER register update enable)
请参见 TIMxCR 说明

位 1 **TIMxICR**：HRTIM_TIMxICR 寄存器更新使能 (HRTIM_TIMxICR register update enable)
请参见 TIMxCR 说明

位 0 **TIMxCR**：HRTIM_TIMxCR 寄存器更新使能 (HRTIM_TIMxCR register update enable)
此位用于定义主定时器 MCR 寄存器是否包含在要通过突发 DMA 更新的寄存器列表中。
0：HRTIM_TIMxCR 寄存器不会通过突发 DMA 访问更新
1：HRTIM_TIMxCR 寄存器会通过突发 DMA 访问更新

37.5.62 HRTIM 突发 DMA 数据寄存器 (HRTIM_BDMADR)

HRTIM Burst DMA Data Register

偏移地址：0x3F0h

复位值：0x0000 0000

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

位 31:0 **BDMADR[31:0]**：突发 DMA 数据寄存器 (Burst DMA Data register)
对该寄存器进行写访问会触发：
– 将数据值复制到在 BDTxUPR 和 BDMUPR 寄存器位中使能的寄存器
– 寄存器指针递增到下一个要填入的位置

37.5.63 HRTIM 寄存器映射

下表总结了 HRTIM 寄存器映射。*表 307* 和*表 308* 中的偏移地址是指*表 306* 中给出的基址偏移。

表306. RTIM 全局寄存器映射

基址偏移	寄存器
0x000 - 0x07F	主定时器
0x080 - 0x0FF	定时器 A
0x100 - 0x17F	定时器 B
0x180 - 0x1FF	定时器 C
0x200 - 0x27F	定时器 D
0x280 - 0x2FF	定时器 E
0x300 - 0x37F	保留
0x380 - 0x3FF	通用寄存器

表307. HRTIM 寄存器映射和复位值：主定时器

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x0000	HRTIM_MCR	BRSTDMA[1:0]		MREPU	Res.	PREEN	DACSYNC[1:0]		Res.	Res.	Res.	TECEN	TDCEN	TCCEN	TBCEN	TACEN	MCEN	SYNCSRC[1:0]		SYNCOUT[1:0]		SYNCSTRTM	SYNCRSTM	SYNCIN[1:0]		Res.	Res.	HALF	RETRIG	CONT	CKPSC[2:0]		
	Reset value	0	0	0		0	0	0				0	0	0	0	0	0	0	0	0	0	0	0	0	0			0	0	0	0	0	0
0x0004	HRTIM_MISR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MUPD	SYNC	MREP	MCMP4	MCMP3	MCMP2	MCMP1
	Reset value																										0	0	0	0	0	0	0
0x0008	HRTIM_MICR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MUPDC	SYNCC	MREPC	MCMP4C	MCMP3C	MCMP2C	MCMP1C
	Reset value																										0	0	0	0	0	0	0
0x000C	HRTIM_MDIER(1)	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MUPDDE	SYNCDE	MREPDE	MCMP4DE	MCMP3DE	MCMP2DE	MCMP1DE	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MUPDIE	SYNCIE	MREPIE	MCMP4IE	MCMP3IE	MCMP2IE	MCMP1IE
	Reset value										0	0	0	0	0	0	0										0	0	0	0	0	0	0
0x0010	HRTIM_MCNTR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MCNT[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

表307. HRTIM 寄存器映射和复位值：主定时器（续）

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x0014	**HRTIM_MPER**(1)	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MPER[15:0]															
	Reset value																	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1
0x0018	**HRTIM_MREP**(1)	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MREP[7:0]							
	Reset value																									0	0	0	0	0	0	0	0
0x001C	**HRTIM_MCMP1R**(1)	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MCMP1[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0020	Reserved	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	Reset value																																
0x0024	**HRTIM_MCMP2R**(1)	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MCMP2[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0028	**HRTIM_MCMP3R**(1)	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MCMP3[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x002C	**HRTIM_MCMP4R**(1)	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MCMP4[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

1. 该寄存器可预装载（请参见*第 1261 页的表 292*）。

表308. HRTIM 寄存器映射和复位值：TIMx (x= A..E)

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x0000	HRTIM_TIMxCR	UPDGAT[3:0]				PREEN	DACSYNC[1:0]		MSTU	TEU	TDU	TCU	TBU	Res.	TxRSTU	TxREPU	Res.	DELCMP4[1:0]		DELCMP2[1:0]		SYNCSTRTx	SYNCRSTx	Res.	Res.	Res.	PSHPLL	HALF	RETRIG	CONT	CKPSCx[2:0]		
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0		0	0		0	0	0	0	0	0				0	0	0	0	0	0	0
0x0004	HRTIM_TIMxISR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	O2CPY	O1CPY	O2STAT	O1STAT	IPPSTAT	CPPSTAT	Res.	DLYPRT	RST	RSTx2	SETx2	RSTx1	SETx1	CPT2	CPT1	UPD	Res.	REP	CMP4	CMP3	CMP2	CMP1
	Reset value											0	0	0	0	0	0		0	0	0	0	0	0	0	0	0		0	0	0	0	0
0x0008	HRTIM_TIMxICR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DLYPRTC	RSTC	RSTx2C	SET2xC	RSTx1C	SET1xC	CPT2C	CPT1C	UPDC	Res.	REPC	CMP4C	CMP3C	CMP2C	CMP1C
	Reset value																	0	0	0	0	0	0	0	0	0	0		0	0	0	0	0
0x000C	HRTIM_TIMxDIER(1)	Res.	DLYPRTDE	RSTDE	RSTx2DE	SETx2DE	RSTx1DE	SET1xDE	CPT2DE	CPT1DE	UPDDE	Res.	REPDE	CMP4DE	CMP3DE	CMP2DE	CMP1DE	Res.	DLYPRTIE	RSTIE	RSTx2IE	SETx2IE	RSTx1IE	SET1xIE	CPT2IE	CPT1IE	UPDIE	Res.	REPIE	CMP4IE	CMP3IE	CMP2IE	CMP1IE
	Reset value		0	0	0	0	0	0	0	0	0		0	0	0	0	0		0	0	0	0	0	0	0	0	0		0	0	0	0	0
0x0010	HRTIM_CNTxR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CNTx[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0014	HRTIM_PERxR(1)	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PERx[15:0]															
	Reset value																	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1
0x0018	HRTIM_REPxR(1)	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REPx[7:0]						
	Reset value																									0	0	0	0	0	0	0	0
0x001C	HRTIM_CMP1xR(1)	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CMP1x[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0020	HRTIM_CMP1CxR(1)	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REPx[7:0]								CMP1x[15:0]															
	Reset value									0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0024	HRTIM_CMP2xR(1)	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CMP2x[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0028	HRTIM_CMP3xR(1)	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CMP3x[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x002C	HRTIM_CMP4xR(1)	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CMP4x[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0030	HRTIM_CPT1xR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CPT1x[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

表308. HRTIM 寄存器映射和复位值：TIMx (x= A..E)（续）

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x0034	HRTIM_CPT2xR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CPT2x[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0038	HRTIM_DTxR(1)	DTFLKx	DTFSLKx	Res.	Res.	Res.	Res.	SDTFx	DTFx[8:0]									DTRLKx	DTRSLKx	Res.	DTPRSC[2:0]			SDTRx	DTRx[8:0]								
	Reset value	0	0					0	0	0	0	0	0	0	0	0	0	0	0		0	0	0	0	0	0	0	0	0	0	0	0	0
0x003C	HRTIM_SETx1R(1)	UPDATE	EXTEVNT10	EXTEVNT9	EXTEVNT8	EXTEVNT7	EXTEVNT6	EXTEVNT5	EXTEVNT4	EXTEVNT3	EXTEVNT2	EXTEVNT1	TIMEVNT9	TIMEVNT8	TIMEVNT7	TIMEVNT6	TIMEVNT5	TIMEVNT4	TIMEVNT3	TIMEVNT2	TIMEVNT1	MSTCMP4	MSTCMP3	MSTCMP2	MSTCMP1	MSTPER	CMP4	CMP3	CMP2	CMP1	PER	RESYNC	SST
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0040	HRTIM_RSTx1R(1)	UPDATE	EXTEVNT10	EXTEVNT9	EXTEVNT8	EXTEVNT7	EXTEVNT6	EXTEVNT5	EXTEVNT4	EXTEVNT3	EXTEVNT2	EXTEVNT1	TIMEVNT9	TIMEVNT8	TIMEVNT7	TIMEVNT6	TIMEVNT5	TIMEVNT4	TIMEVNT3	TIMEVNT2	TIMEVNT1	MSTCMP4	MSTCMP3	MSTCMP2	MSTCMP1	MSTPER	CMP4	CMP3	CMP2	CMP1	PER	RESYNC	SRT
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0044	HRTIM_SETx2R(1)	UPDATE	EXTEVNT10	EXTEVNT9	EXTEVNT8	EXTEVNT7	EXTEVNT6	EXTEVNT5	EXTEVNT4	EXTEVNT3	EXTEVNT2	EXTEVNT1	TIMEVNT9	TIMEVNT8	TIMEVNT7	TIMEVNT6	TIMEVNT5	TIMEVNT4	TIMEVNT3	TIMEVNT2	TIMEVNT1	MSTCMP4	MSTCMP3	MSTCMP2	MSTCMP1	MSTPER	CMP4	CMP3	CMP2	CMP1	PER	RESYNC	SST
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0048	HRTIM_RSTx2R(1)	UPDATE	EXTEVNT10	EXTEVNT9	EXTEVNT8	EXTEVNT7	EXTEVNT6	EXTEVNT5	EXTEVNT4	EXTEVNT3	EXTEVNT2	EXTEVNT1	TIMEVNT9	TIMEVNT8	TIMEVNT7	TIMEVNT6	TIMEVNT5	TIMEVNT4	TIMEVNT3	TIMEVNT2	TIMEVNT1	MSTCMP4	MSTCMP3	MSTCMP2	MSTCMP1	MSTPER	CMP4	CMP3	CMP2	CMP1	PER	RESYNC	SRT
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x004C	HRTIM_EEFxR1	Res.	Res.	Res.	EE5FLTR[3:0]				EE5LTCH	Res.	EE4FLTR[3:0]				EE4LTCH	Res.	EE3FLTR[3:0]				EE3LTCH	Res.	EE2FLTR[3:0]				EE2LTCH	Res.	EE1FLTR[3:0]				EE1LTCH
	Reset value				0	0	0	0	0		0	0	0	0	0		0	0	0	0	0		0	0	0	0	0		0	0	0	0	0
0x0050	HRTIM_EEFxR2	Res.	Res.	Res.	EE10FLTR[3:0]				EE10LTCH	Res.	EE9FLTR[3:0]				EE9LTCH	Res.	EE8FLTR[3:0]				EE8LTCH	Res.	EE7FLTR[3:0]				EE7LTCH	Res.	EE6FLTR[3:0]				EE6LTCH
	Reset value				0	0	0	0	0		0	0	0	0	0		0	0	0	0	0		0	0	0	0	0		0	0	0	0	0
0x0054	HRTIM_RSTAR(1)	Res.	TIMECMP4	TIMECMP2	TIMECMP1	TIMDCMP4	TIMDCMP2	TIMDCMP1	TIMCCMP4	TIMCCMP2	TIMCCMP1	TIMBCMP4	TIMBCMP2	TIMBCMP1	EXTEVNT10	EXTEVNT9	EXTEVNT8	EXTEVNT7	EXTEVNT6	EXTEVNT5	EXTEVNT4	EXTEVNT3	EXTEVNT2	EXTEVNT1	MSTCMP4	MSTCMP3	MSTCMP2	MSTCMP1	MSTPER	CMP4	CMP2	UPDT	Res.
	Reset value		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

表308. HRTIM 寄存器映射和复位值：TIMx (x= A..E)（续）

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x0054	HRTIM_RSTBR(1)	Res.	TIMECMP4	TIMECMP2	TIMECMP1	TIMDCMP4	TIMDCMP2	TIMDCMP1	TIMCCMP4	TIMCCMP2	TIMCCMP1	TIMACMP4	TIMACMP2	TIMACMP1	EXTEVNT10	EXTEVNT9	EXTEVNT8	EXTEVNT7	EXTEVNT6	EXTEVNT5	EXTEVNT4	EXTEVNT3	EXTEVNT2	EXTEVNT1	MSTCMP4	MSTCMP3	MSTCMP2	MSTCMP1	MSTPER	CMP4	CMP2	UPDT	Res.
	Reset value		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x0054	HRTIM_RSTCR(1)	Res.	TIMECMP4	TIMECMP2	TIMECMP1	TIMDCMP4	TIMDCMP2	TIMDCMP1	TIMBCMP4	TIMBCMP2	TIMBCMP1	TIMACMP4	TIMACMP2	TIMACMP1	EXTEVNT10	EXTEVNT9	EXTEVNT8	EXTEVNT7	EXTEVNT6	EXTEVNT5	EXTEVNT4	EXTEVNT3	EXTEVNT2	EXTEVNT1	MSTCMP4	MSTCMP3	MSTCMP2	MSTCMP1	MSTPER	CMP4	CMP2	UPDT	Res.
	Reset value		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x0054	HRTIM_RSTDR(1)	Res.	TIMECMP4	TIMECMP2	TIMECMP1	TIMCCMP4	TIMCCMP2	TIMCCMP1	TIMBCMP4	TIMBCMP2	TIMBCMP1	TIMACMP4	TIMACMP2	TIMACMP1	EXTEVNT10	EXTEVNT9	EXTEVNT8	EXTEVNT7	EXTEVNT6	EXTEVNT5	EXTEVNT4	EXTEVNT3	EXTEVNT2	EXTEVNT1	MSTCMP4	MSTCMP3	MSTCMP2	MSTCMP1	MSTPER	CMP4	CMP2	UPDT	Res.
	Reset value		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x0054	HRTIM_RSTER(1)	Res.	TIMDCMP4	TIMDCMP2	TIMDCMP1	TIMCCMP4	TIMCCMP2	TIMCCMP1	TIMBCMP4	TIMBCMP2	TIMBCMP1	TIMACMP4	TIMACMP2	TIMACMP1	EXTEVNT10	EXTEVNT9	EXTEVNT8	EXTEVNT7	EXTEVNT6	EXTEVNT5	EXTEVNT4	EXTEVNT3	EXTEVNT2	EXTEVNT1	MSTCMP4	MSTCMP3	MSTCMP2	MSTCMP1	MSTPER	CMP4	CMP2	UPDT	Res.
	Reset value		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
0x0058	HRTIM_CHPxR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	STRTPW [3:0]				CARDTY [2:0]			CARFRQ [3:0]			
	Reset value																						0	0	0	0	0	0	0	0	0	0	0
0x005C	HRTIM_CPT1ACR	TECMP2	TECMP1	TE1RST	TE1SET	TDCMP2	TDCMP1	TD1RST	TD1SET	TCCMP2	TCCMP1	TC1RST	TC1SET	TBCMP2	TBCMP1	TB1RST	TB1SET	Res.	Res.	Res.	Res.	EXEV10CPT	EXEV9CPT	EXEV8CPT	EXEV7CPT	EXEV6CPT	EXEV5CPT	EXEV4CPT	EXEV3CPT	EXEV2CPT	EXEV1CPT	UPDCPT	SWCPT
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0					0	0	0	0	0	0	0	0	0	0	0	0
0x005C	HRTIM_CPT1BCR	TECMP2	TECMP1	TE1RST	TE1SET	TDCMP2	TDCMP1	TD1RST	TD1SET	TCCMP2	TCCMP1	TC1RST	TC1SET	Res.	Res.	Res.	Res.	TACMP2	TACMP1	TA1RST	TA1SET	EXEV10CPT	EXEV9CPT	EXEV8CPT	EXEV7CPT	EXEV6CPT	EXEV5CPT	EXEV4CPT	EXEV3CPT	EXEV2CPT	EXEV1CPT	UPDCPT	SWCPT
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0					0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x005C	HRTIM_CPT1CCR	TECMP2	TECMP1	TE1RST	TE1SET	TDCMP2	TDCMP1	TD1RST	TD1SET	Res.	Res.	Res.	Res.	TBCMP2	TBCMP1	TB1RST	TB1SET	TACMP2	TACMP1	TA1RST	TA1SET	EXEV10CPT	EXEV9CPT	EXEV8CPT	EXEV7CPT	EXEV6CPT	EXEV5CPT	EXEV4CPT	EXEV3CPT	EXEV2CPT	EXEV1CPT	UPDCPT	SWCPT
	Reset value	0	0	0	0	0	0	0	0					0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x005C	HRTIM_CPT1DCR	TECMP2	TECMP1	TE1RST	TE1SET	Res.	Res.	Res.	Res.	TCCMP2	TCCMP1	TC1RST	TC1SET	TBCMP2	TBCMP1	TB1RST	TB1SET	TACMP2	TACMP1	TA1RST	TA1SET	EXEV10CPT	EXEV9CPT	EXEV8CPT	EXEV7CPT	EXEV6CPT	EXEV5CPT	EXEV4CPT	EXEV3CPT	EXEV2CPT	EXEV1CPT	UPDCPT	SWCPT
	Reset value	0	0	0	0					0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

表308. HRTIM 寄存器映射和复位值：TIMx (x= A..E)（续）

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x005C	HRTIM_CPT1ECR	Res.	Res.	Res.	Res.	TDCMP2	TDCMP1	TD1RST	TD1SET	TCCMP2	TCCMP1	TC1RST	TC1SET	TBCMP2	TBCMP1	TB1RST	TB1SET	TACMP2	TACMP1	TA1RST	TA1SET	EXEV10CPT	EXEV9CPT	EXEV8CPT	EXEV7CPT	EXEV6CPT	EXEV5CPT	EXEV4CPT	EXEV3CPT	EXEV2CPT	EXEV1CPT	UPDCPT	SWCPT
	Reset value					0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0060	HRTIM_CPT2ACR	TECMP2	TECMP1	TE1RST	TE1SET	TDCMP2	TDCMP1	TD1RST	TD1SET	TCCMP2	TCCMP1	TC1RST	TC1SET	TBCMP2	TBCMP1	TB1RST	TB1SET	Res.	Res.	Res.	Res.	EXEV10CPT	EXEV9CPT	EXEV8CPT	EXEV7CPT	EXEV6CPT	EXEV5CPT	EXEV4CPT	EXEV3CPT	EXEV2CPT	EXEV1CPT	UPDCPT	SWCPT
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0					0	0	0	0	0	0	0	0	0	0	0	0
0x0060	HRTIM_CPT2BCR	TECMP2	TECMP1	TE1RST	TE1SET	TDCMP2	TDCMP1	TD1RST	TD1SET	TCCMP2	TCCMP1	TC1RST	TC1SET	Res.	Res.	Res.	Res.	TACMP2	TACMP1	TA1RST	TA1SET	EXEV10CPT	EXEV9CPT	EXEV8CPT	EXEV7CPT	EXEV6CPT	EXEV5CPT	EXEV4CPT	EXEV3CPT	EXEV2CPT	EXEV1CPT	UPDCPT	SWCPT
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0					0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0060	HRTIM_CPT2CCR	TECMP2	TECMP1	TE1RST	TE1SET	TDCMP2	TDCMP1	TD1RST	TD1SET	Res.	Res.	Res.	Res.	TBCMP2	TBCMP1	TB1RST	TB1SET	TACMP2	TACMP1	TA1RST	TA1SET	EXEV10CPT	EXEV9CPT	EXEV8CPT	EXEV7CPT	EXEV6CPT	EXEV5CPT	EXEV4CPT	EXEV3CPT	EXEV2CPT	EXEV1CPT	UPDCPT	SWCPT
	Reset value	0	0	0	0	0	0	0	0					0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0060	HRTIM_CPT2DCR	TECMP2	TECMP1	TE1RST	TE1SET	Res.	Res.	Res.	Res.	TCCMP2	TCCMP1	TC1RST	TC1SET	TBCMP2	TBCMP1	TB1RST	TB1SET	TACMP2	TACMP1	TA1RST	TA1SET	EXEV10CPT	EXEV9CPT	EXEV8CPT	EXEV7CPT	EXEV6CPT	EXEV5CPT	EXEV4CPT	EXEV3CPT	EXEV2CPT	EXEV1CPT	UPDCPT	SWCPT
	Reset value	0	0	0	0					0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0060	HRTIM_CPT2ECR	Res.	Res.	Res.	Res.	TDCMP2	TDCMP1	TD1RST	TD1SET	TCCMP2	TCCMP1	TC1RST	TC1SET	TBCMP2	TBCMP1	TB1RST	TB1SET	TACMP2	TACMP1	TA1RST	TA1SET	EXEV10CPT	EXEV9CPT	EXEV8CPT	EXEV7CPT	EXEV6CPT	EXEV5CPT	EXEV4CPT	EXEV3CPT	EXEV2CPT	EXEV1CPT	UPDCPT	SWCPT
	Reset value					0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0064	HRTIM_OUTxR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DIDL2	CHP2	FAULT2[1:0]		IDLES2	IDLEM2	POL2	Res.	Res.	Res.	Res.	Res.	DLYPRT[2:0]		DLYPRTEN	DTEN	DIDL1	CHP1	FAULT1[1:0]		IDLES1	IDLEM1	POL1	Res.
	Reset value									0	0	0	0	0	0	0						0		0	0	0	0	0	0	0	0	0	
0x0068	HRTIM_FLTxR	FLTLCK	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FLT5EN	FLT4EN	FLT3EN	FLT2EN	FLT1EN
	Reset value	0																											0	0	0	0	0

1. 该寄存器可预装载（请参见*第 1261 页的表 292*）。

表309. HRTIM 寄存器映射和复位值：常用功能

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x0000	**HRTIM_CR1**	Res.	Res.	Res.	Res.	AD4USRC[2:0]			AD3USRC[2:0]			AD2USRC[2:0]			AD1USRC[2:0]			Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TEUDIS	TDUDIS	TCUDIS	TBUDIS	TAUDIS	MUDIS
	Reset value					0	0	0	0	0	0	0	0	0	0	0	0											0	0	0	0	0	0
0x0004	**HRTIM_CR2**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TERST	TDRST	TCRST	TBRST	TARST	MRST	Res.	Res.	TESWU	TDSWU	TCSWU	TBSWU	TASWU	MSWU
	Reset value																			0	0	0	0	0	0			0	0	0	0	0	0
0x008	**HRTIM_ISR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BMPER	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SYSFLT	FLT5	FLT4	FLT3	FLT2	FLT1
	Reset value															0												0	0	0	0	0	0
0x000C	**HRTIM_ICR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BMPERC	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SYSFLTC	FLT5C	FLT4C	FLT3C	FLT2C	FLT1C
	Reset value															0												0	0	0	0	0	0
0x0010	**HRTIM_IER**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BMPERIE	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SYSFLTIE	FLT5IE	FLT4IE	FLT3IE	FLT2IE	FLT1IE
	Reset value															0												0	0	0	0	0	0
0x0014	**HRTIM_OENR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TE2OEN	TE1OEN	TD2OEN	TD1OEN	TC2OEN	TC1OEN	TB2OEN	TB1OEN	TA2OEN	TA1OEN
	Reset value																							0	0	0	0	0	0	0	0	0	0
0x0018	**HRTIM_DISR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TE2ODIS	TE1ODIS	TD2ODIS	TD1ODIS	TC2ODIS	TC1ODIS	TB2ODIS	TB1ODIS	TA2ODIS	TA1ODIS
	Reset value																							0	0	0	0	0	0	0	0	0	0
0x001C	**HRTIM_ODSR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TE2ODS	TE1ODS	TD2ODS	TD1ODS	TC2ODS	TC1ODS	TB2ODS	TB1ODS	TA2ODS	TA1ODS
	Reset value																							0	0	0	0	0	0	0	0	0	0
0x0020	**HRTIM_BMCR**	BMSTAT	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TEBM	TDBM	TCBM	TBBM	TABM	MTBM	Res.	Res.	Res.	Res.	Res.	BMPREN	BMPRSC[3:0]				BMCLK[3:0]				BMOM	BME
	Reset value	0										0	0	0	0	0	0						0	0	0	0	0	0	0	0	0	0	0
0x0024	**HRTIM_BMTRG**	OCHPEV	Res.	Res.	Res.	Res.	TECMP2	TECMP1	TEREP	TERST	TDCMP2	TDCMP1	TDREP	TDRST	TCCMP2	TCCMP1	TCREP	TCRST	TBCMP2	TBCMP1	TBREP	TBRST	TACMP2	TACMP1	TAREP	TARST	MSTCMP4	MSTCMP3	MSTCMP2	MSTCMP1	MSTREP	MSTRST	SW
	Reset value	0					0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0028	**HRTIM_BMCMPR**(1)	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BMCMP[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

表309. HRTIM 寄存器映射和复位值：常用功能（续）

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x002C	**HRTIM_BMPER**(1)	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BMPER[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0030	**HRTIM_EECR1**	Res.	Res.	EE5FAST	EE5SNS[1:0]		EE5POL	EE5SRC[1:0]		EE4FAST	EE4SNS[1:0]		EE4POL	EE4SRC[1:0]		EE3FAST	EE3SNS[1:0]		EE3POL	EE3SRC[1:0]		EE2FAST	EE2SNS[1:0]		EE2POL	EE2SRC[1:0]		EE1FAST	EE1SNS[1:0]		EE1POL	EE1SRC[1:0]	
	Reset value			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0034	**HRTIM_EECR2**	Res.	Res.	Res.	EE10SNS[1:0]		EE10POL	EE10SRC[1:0]		Res.	EE9SNS[1:0]		EE9POL	EE9SRC[1:0]		Res.	EE8SNS[1:0]		EE8POL	EE8SRC[1:0]		Res.	EE7SNS[1:0]		EE7POL	EE7SRC[1:0]		Res.	EE6SNS[1:0]		EE6POL	EE6SRC[1:0]	
	Reset value				0	0	0	0	0		0	0	0	0	0		0	0	0	0	0		0	0	0	0	0		0	0	0	0	0	0
0x0038	**HRTIM_EECR3**	Res.	Res.	Res.	EE10SNS[1:0]		EE10POL	EE10SRC[1:0]		Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	EE6SRC[1:0]	
	Reset value				0	0	0	0	0		0	0	0	0	0		0	0	0	0	0		0	0	0	0	0		0	0	0	0	0	0
0x003C	**HRTIM_ADC1R**(1)	AD1TEPER	AD1TEC4	AD1TEC3	AD1TEC2	AD1TDPER	AD1TDC4	AD1TDC3	AD1TDC2	AD1TCPER	AD1TCC4	AD1TCC3	AD1TCC2	AD1TBRST	AD1TBPER	AD1TBC4	AD1TBC3	AD1TBC2	AD1TARST	AD1TAPER	AD1TAC4	AD1TAC3	AD1TAC2	AD1EEV5	AD1EEV4	AD1EEV3	AD1EEV2	AD1EEV1	AD1MPER	AD1MC4	AD1MC3	AD1MC2	AD1MC1
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0040	**HRTIM_ADC2R**(1)	AD2TERST	AD2TEC4	AD2TEC3	AD2TEC2	AD2TDRST	AD2TDPER	AD2TDC4	AD2TDC3	AD2TDC2	AD2TCRST	AD2TCPER	AD2TCC4	AD2TCC3	AD2TCC2	AD2TBPER	AD2TBC4	AD2TBC3	AD2TBC2	AD2TAPER	AD2TAC4	AD2TAC3	AD2TAC2	AD2EEV10	AD2EEV9	AD2EEV8	AD2EEV7	AD2EEV6	AD2MPER	AD2MC4	AD2MC3	AD2MC2	AD2MC1
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0044	**HRTIM_ADC3R**(1)	ADC3TEPER	AD1TEC4	AD1TEC3	AD1TEC2	AD1TDPER	AD1TDC4	AD1TDC3	AD1TDC2	AD1TCPER	AD1TCC4	AD1TCC3	AD1TCC2	AD1TBRST	AD1TBPER	AD1TBC4	AD1TBC3	AD1TBC2	AD1TARST	AD1TAPER	AD1TAC4	AD1TAC3	AD1TAC2	AD1EEV5	AD1EEV4	AD1EEV3	AD1EEV2	AD1EEV1	AD1MPER	AD1MC4	AD1MC3	AD1MC2	AD1MC1
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0048	**HRTIM_ADC4R**(1)	AD2TERST	AD2TEC4	AD2TEC3	AD2TEC2	AD2TDRST	AD2TDPER	AD2TDC4	AD2TDC3	AD2TDC2	AD2TCRST	AD2TCPER	AD2TCC4	AD2TCC3	AD2TCC2	AD2TBPER	AD2TBC4	AD2TBC3	AD2TBC2	AD2TAPER	AD2TAC4	AD2TAC3	AD2TAC2	AD2EEV10	AD2EEV9	AD2EEV8	AD2EEV7	AD2EEV6	AD2MPER	AD2MC4	AD2MC3	AD2MC2	AD2MC1
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x004C	Reserved	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
0x0050	**HRTIM_FLTINxR1**	FLT4LCK	FLT4F[3:0]				FLT4SRC	FLT4P	FLT4E	FLT3LCK	FLT3F[3:0]				FLT3SRC	FLT3P	FLT3E	FLT2LCK	FLT2F[3:0]				FLT2SRC	FLT2P	FLT2E	FLT1LCK	FLT1F[3:0]				FLT1SRC	FLT1P	FLT1E
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

表309. HRTIM 寄存器映射和复位值：常用功能（续）

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x0054	HRTIM_FLTINxR2	Res.	Res.	Res.	Res.	Res.	Res.	FLTSD[1:0]		Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FLT5LCK	FLT5F[3:0]				FLT5SRC	FLT5P	FLT5E
	Reset value							0	0																	0	0	0	0	0	0	0	0
0x0058	HRTIM_ BDMUPDR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MCMP4	MCMP3	MCMP2	MCMP1	MREP	MPER	MCNT	MDIER	MICR	MCR
	Reset value																							0	0	0	0	0	0	0	0	0	0
0x005C	HRTIM_BDTAUPR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TIMAFLTR	TIMAOUTR	TIMACHPR	TIMARSTR	TIMAEEFR2	TIMAEEFR1	TIMARST2R	TIMASET2R	TIMARST1R	TIMASET1R	TIMADTxR	TIMACMP4	TIMACMP3	TIMACMP2	TIMACMP1	TIMAREP	TIMAPER	TIMACNT	TIMADIER	TIMAICR	TIMACR
	Reset value												0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0060	HRTIM_ BDTBUPR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TIMBFLTR	TIMBOUTR	TIMBCHPR	TIMBRSTR	TIMBEEFR2	TIMBEEFR1	TIMBRST2R	TIMBSET2R	TIMBRST1R	TIMBSET1R	TIMBDTxR	TIMBCMP4	TIMBCMP3	TIMBCMP2	TIMBCMP1	TIMBREP	TIMBPER	TIMBCNT	TIMBDIER	TIMBICR	TIMBCR
	Reset value												0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0064	HRTIM_ BDTCUPR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TIMCFLTR	TIMCOUTR	TIMCCHPR	TIMCRSTR	TIMCEEFR2	TIMCEEFR1	TIMCRST2R	TIMCSET2R	TIMCRST1R	TIMCSET1R	TIMCDTxR	TIMCCMP4	TIMCCMP3	TIMCCMP2	TIMCCMP1	TIMCREP	TIMCPER	TIMCCNT	TIMCDIER	TIMCICR	TIMCCR
	Reset value												0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0068	HRTIM_ BDTDUPR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TIMDFLTR	TIMDOUTR	TIMDCHPR	TIMDRSTR	TIMDEEFR2	TIMDEEFR1	TIMDRST2R	TIMDSET2R	TIMDRST1R	TIMDSET1R	TIMDDTxR	TIMDCMP4	TIMDCMP3	TIMDCMP2	TIMDCMP1	TIMDREP	TIMDPER	TIMDCNT	TIMDDIER	TIMDICR	TIMDCR
	Reset value												0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x006C	HRTIM_ BDTEUPR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TIMEFLTR	TIMEOUTR	TIMECHPR	TIMERSTR	TIMEEEFR2	TIMEEEFR1	TIMERST2R	TIMESET2R	TIMERST1R	TIMESET1R	TIMEDTxR	TIMECMP4	TIMECMP3	TIMECMP2	TIMECMP1	TIMEREP	TIMEPER	TIMECNT	TIMEDIER	TIMEICR	TIMECR
	Reset value												0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0070	HRTIM_BDMADR	BDMADR[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

1. 该寄存器可预装载（请参见*第 1261 页的表 292*）。

38 高级控制定时器 (TIM1/TIM8)

38.1 TIM1/TIM8 简介

高级控制定时器 (TIM1/TIM8) 包含一个 16 位自动重载计数器，该计数器由可编程预分频器驱动。

此类定时器可用于多种用途，包括测量输入信号的脉冲宽度（输入捕获），或者生成输出波形（输出比较、PWM 和带死区插入的互补 PWM）。

使用定时器预分频器和 RCC 时钟控制器预分频器，可将脉冲宽度和波形周期从几微秒调制到几毫秒。

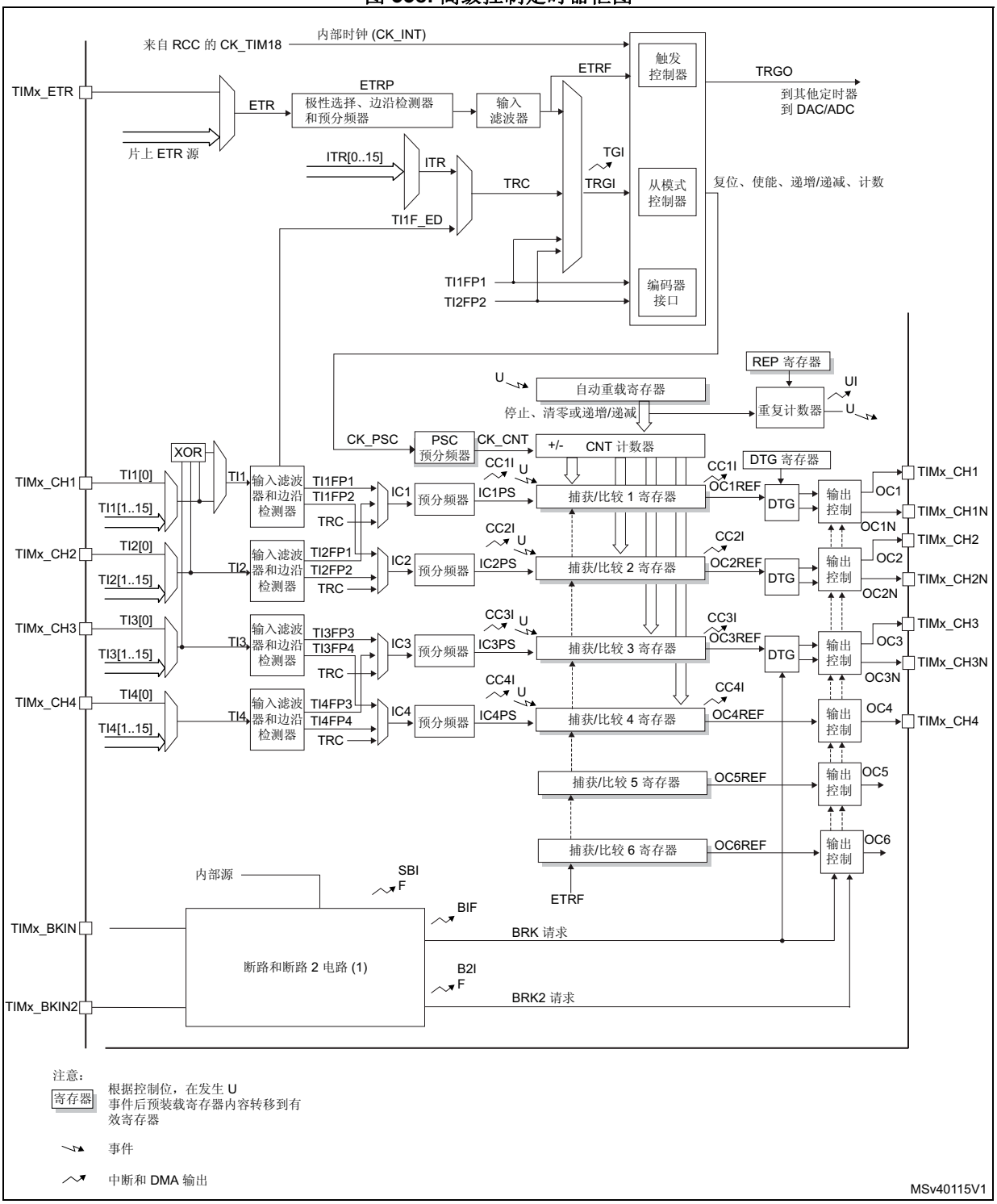
高级控制定时器 (TIM1/TIM8) 和通用 (TIMy) 定时器彼此完全独立，不共享任何资源。如*第 38.3.26 节：定时器同步*中所述，它们可以同步操作。

38.2 TIM1/TIM8 主要特性

TIM1/TIM8 定时器主要特性：

- 16 位递增、递减、递增/递减自动重载计数器。
- 16 位可编程预分频器，用于对计数器时钟频率进行分频（可在运行时修改），分频系数介于 1 到 65536 之间。
- 多达 6 个独立通道，可用于：
  - 输入捕获（但通道 5 和通道 6 除外）
  - 输出比较
  - PWM 生成（边沿和中心对齐模式）
  - 单脉冲模式输出
- 带可编程死区的互补输出。
- 使用外部信号控制定时器且可实现多个定时器互连的同步电路。
- 重复计数器，用于仅在给定数目的计数器周期后更新定时器寄存器。
- 2 个断路输入，用于将定时器的输出信号置于用户可选的安全配置中。
- 发生如下事件时生成中断/DMA 请求：
  - 更新：计数器上溢/下溢、计数器初始化（通过软件或内部/外部触发）
  - 触发事件（计数器启动、停止、初始化或通过内部/外部触发计数）
  - 输入捕获
  - 输出比较
- 支持定位用增量（正交）编码器和霍尔传感器电路。
- 触发输入用作外部时钟或逐周期电流管理。

图 338. 高级控制定时器框图



1. 详细信息，请参见*图 381：断路和断路 2 电路概述*。

38.3 TIM1/TIM8 功能描述

38.3.1 时基单元

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

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

时基单元包括：

- 计数器寄存器 (TIMx_CNT)
- 预分频器寄存器 (TIMx_PSC)
- 自动重载寄存器 (TIMx_ARR)
- 重复计数器寄存器 (TIMx_RCR)

自动重载寄存器是预装载的。对自动重载寄存器执行写入或读取操作时会访问预装载寄存器。预装载寄存器的内容既可以立即传送到影子寄存器，也可以在每次发生更新事件 (UEV) 时传送到影子寄存器，这取决于 TIMx_CR1 寄存器中的自动重载预装载使能位 (ARPE)。当计数器达到上溢值（或者在递减计数时达到下溢值）并且 TIMx_CR1 寄存器中的 UDIS 位为 0 时，将发送更新事件。该更新事件也可由软件产生。下文将针对各配置的更新事件的产生进行详细介绍。

计数器由预分频器输出 CK_CNT 提供时钟，仅当 TIMx_CR1 寄存器中的计数器启动位 (CEN) 置 1 时，才会启动计数器（有关计数器使能的更多详细信息，另请参见从模式控制器的相关说明）。

注意，计数器将在 TIMx_CR1 寄存器的 CEN 位置 1 时刻的一个时钟周期后开始计数。

预分频器说明

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

图 339 和 *图 340* 以一些示例说明在预分频比实时变化时计数器的行为：

图 339. 预分频器分频由 1 变为 2 时的计数器时序图

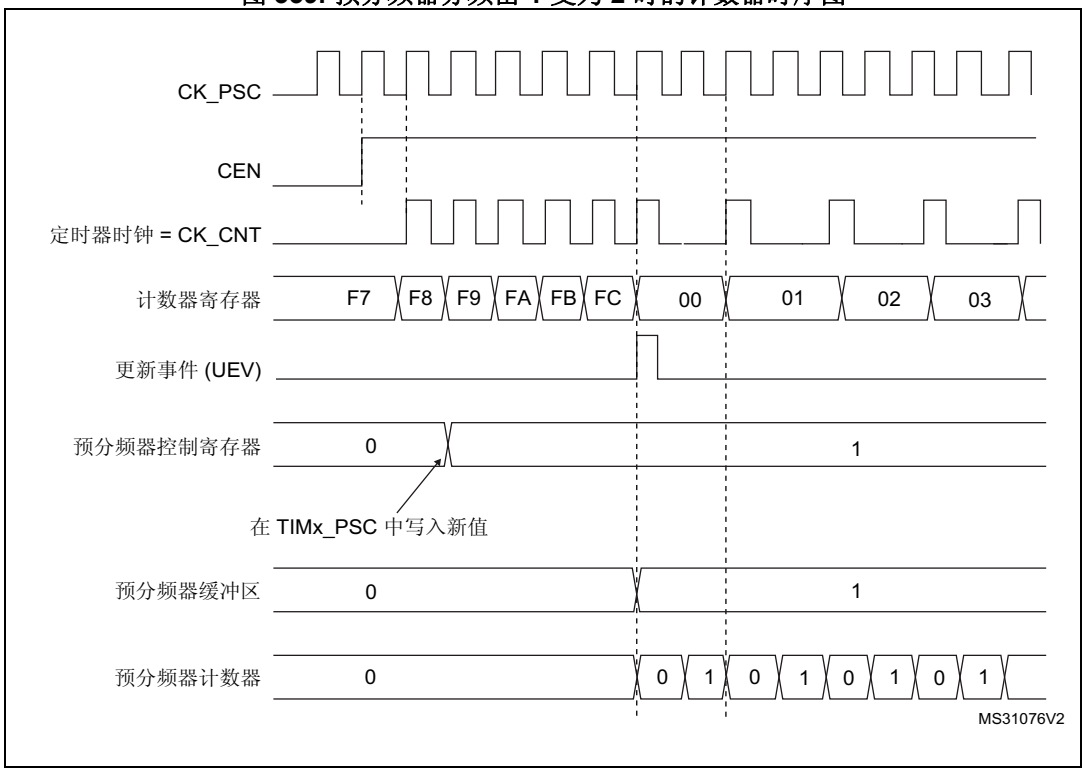
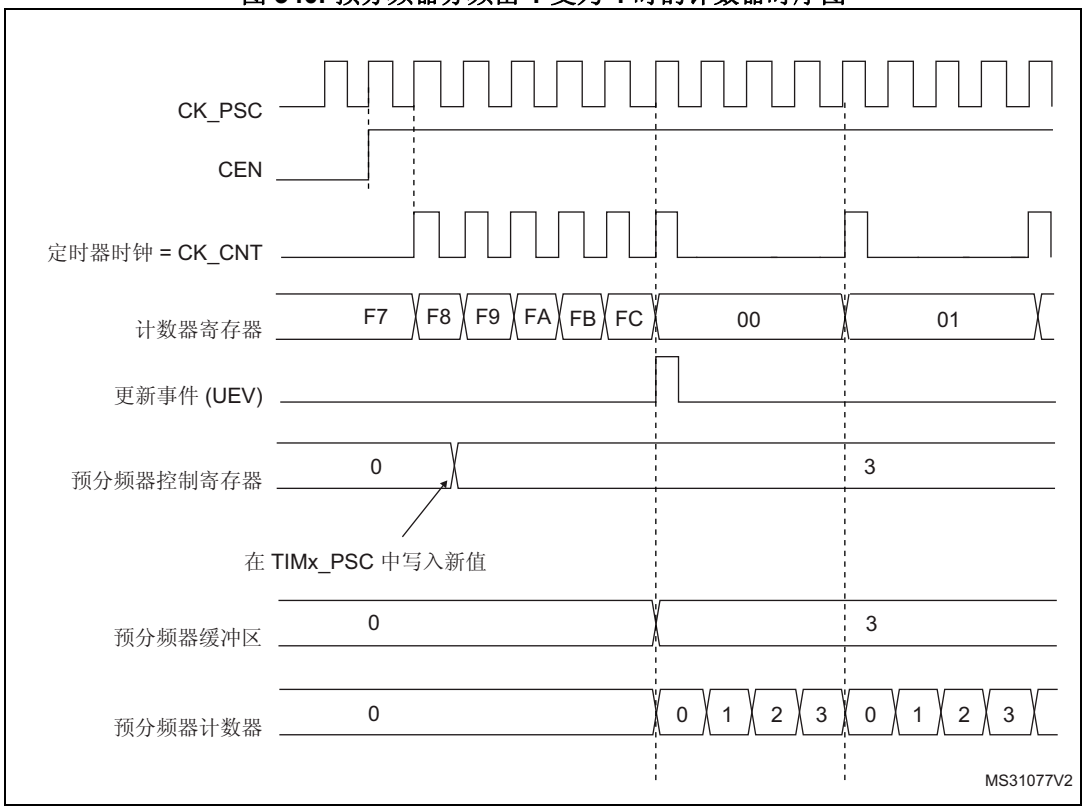


图 340. 预分频器分频由 1 变为 4 时的计数器时序图



38.3.2 计数器模式

递增计数模式

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

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

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

通过软件将 TIMx_CR1 寄存器中的 UDIS 位置 1 可禁止 UEV 事件。这可避免向预装载寄存器写入新值时更新影子寄存器。在 UDIS 位写入 0 之前不会产生任何更新事件。不过，计数器和预分频器计数器都会重新从 0 开始计数（而预分频比保持不变）。此外，如果 TIMx_CR1 寄存器中的 URS 位（更新请求选择）已置 1，则将 UG 位置 1 会生成更新事件 UEV，但不会将 UIF 标志置 1（因此，不会发送任何中断或 DMA 请求）。这样一来，如果在发生捕获事件时将计数器清零，将不会同时产生更新中断和捕获中断。

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

- 重复计数器中将重新装载 TIMx_RCR 寄存器的内容。
- 使用预装载值 (TIMx_ARR) 更新自动重载影子寄存器。
- 预分频器的缓冲区中将重新装载预装载值（TIMx_PSC 寄存器的内容）。

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

图 341. 计数器时序图，1 分频内部时钟

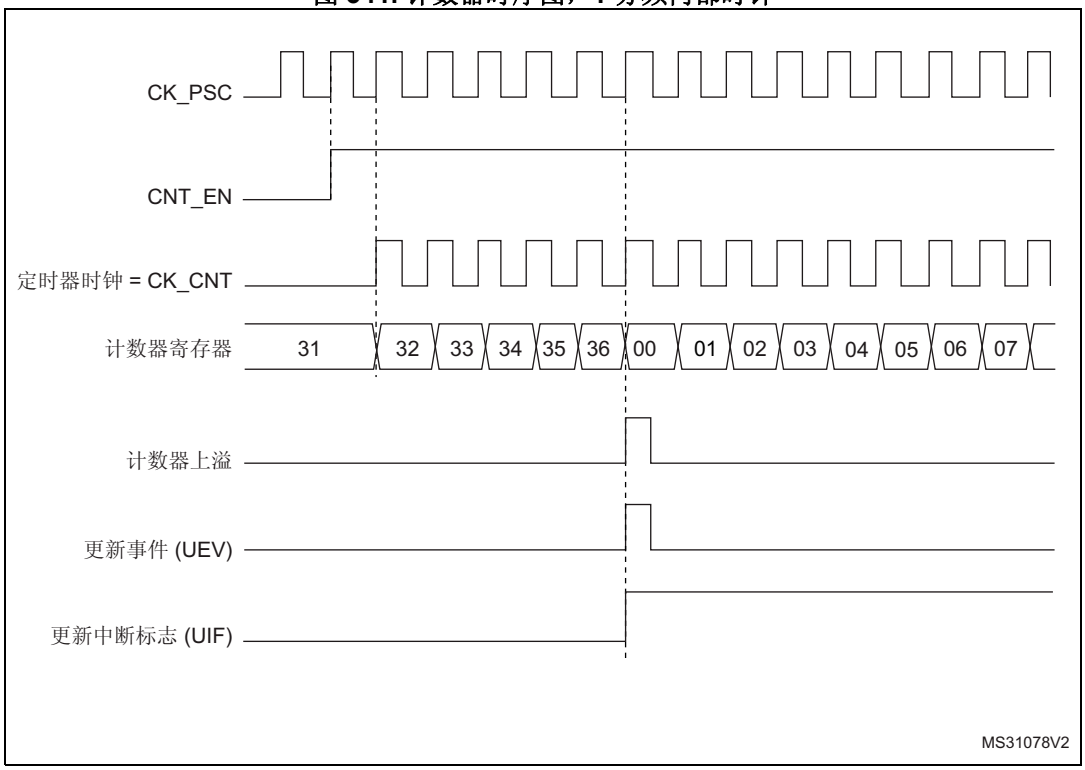


图 342. 计数器时序图，2 分频内部时钟

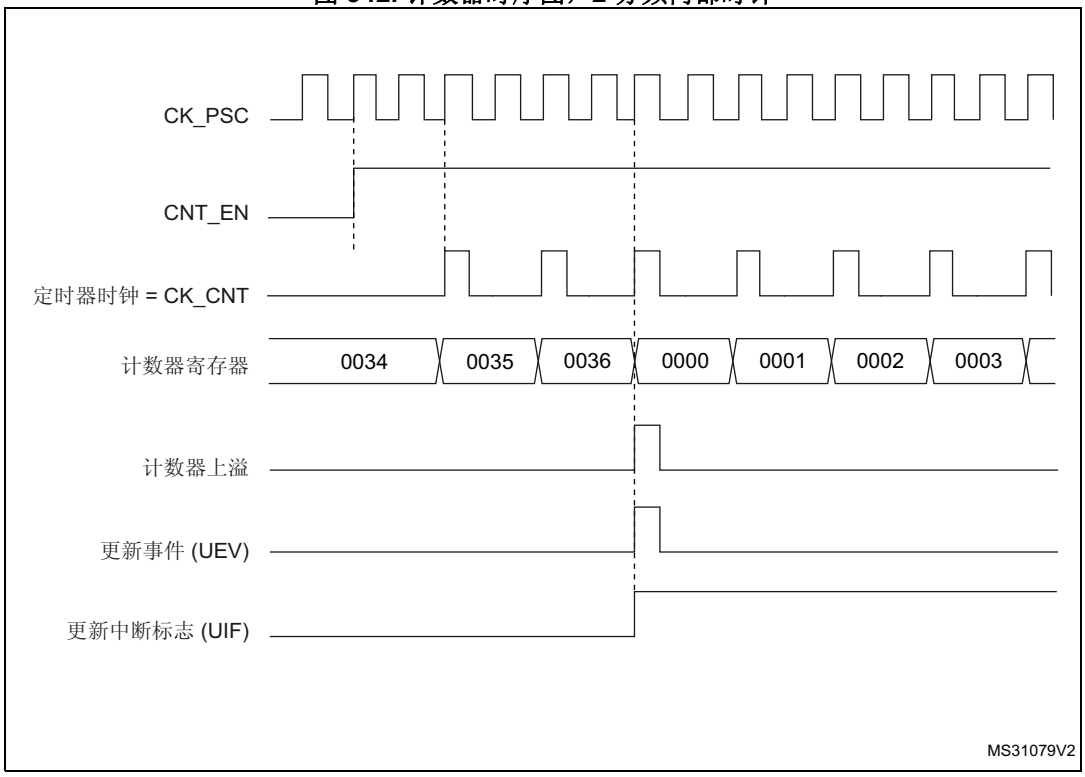


图 343. 计数器时序图，4 分频内部时钟

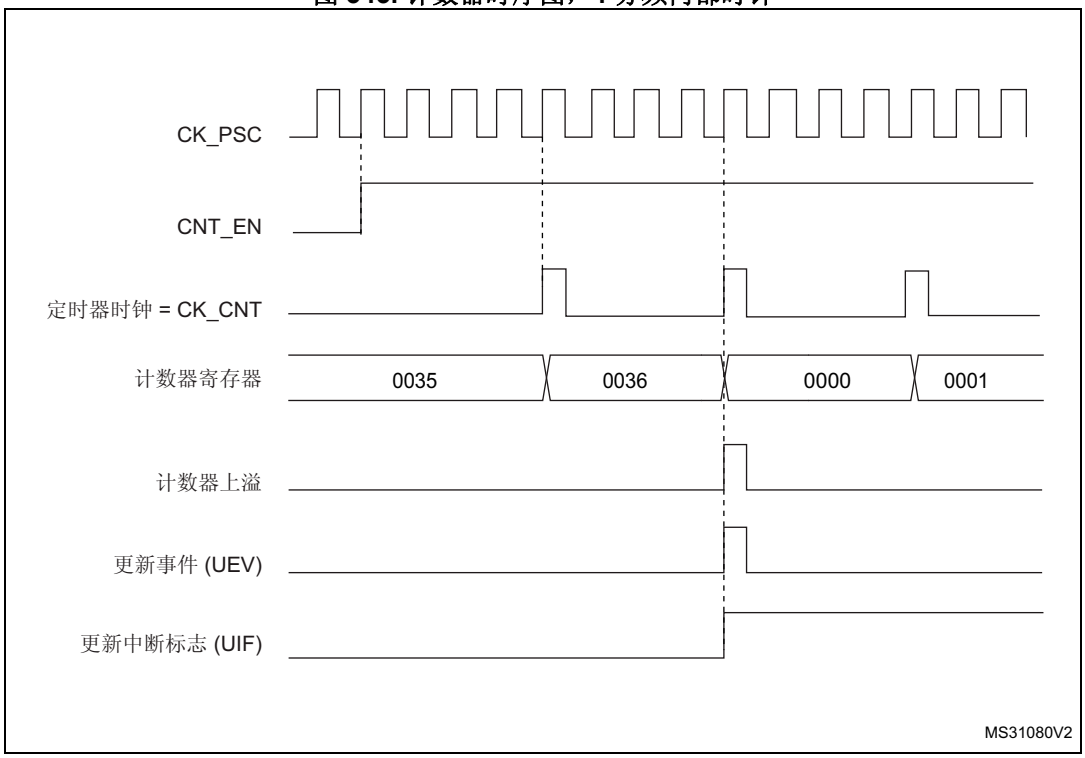


图 344. 计数器时序图，N 分频内部时钟

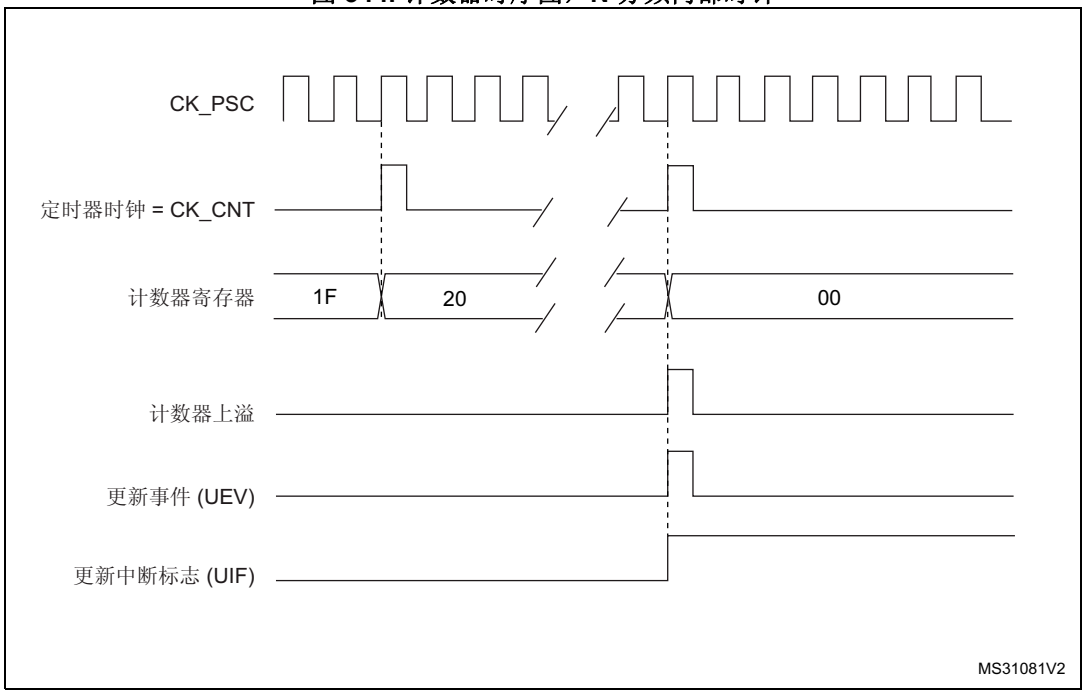


图 345. 计数器时序图，ARPE=0 时更新事件（TIMx_ARR 未预装载）

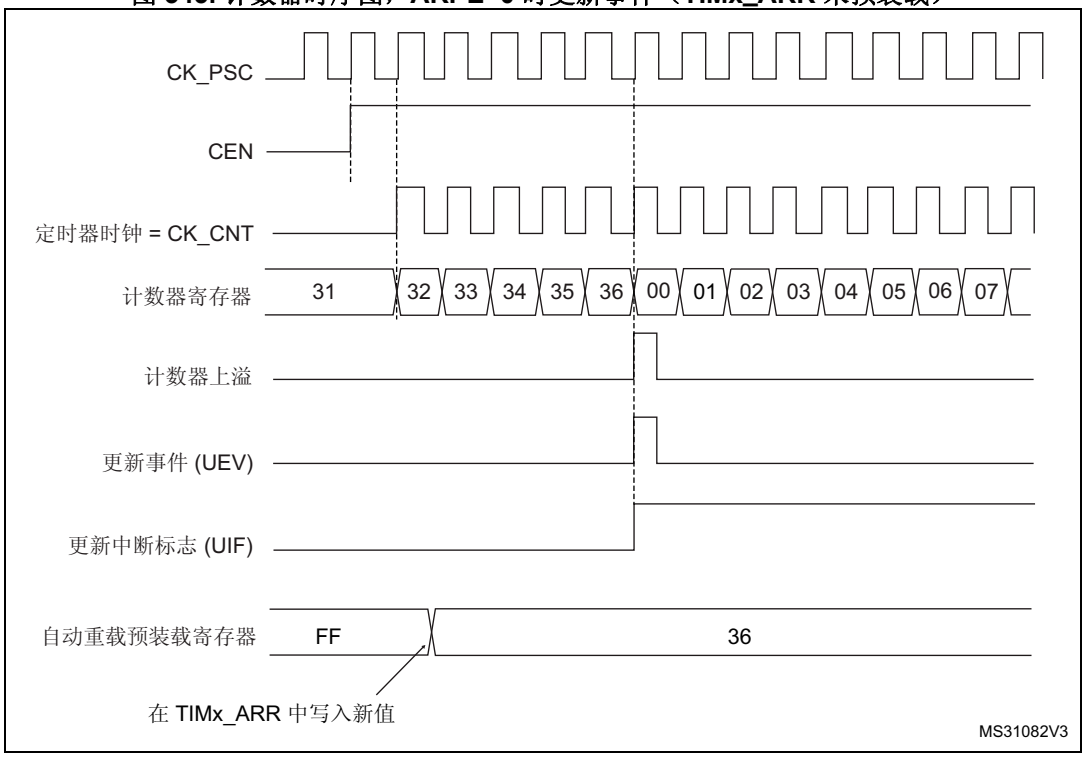
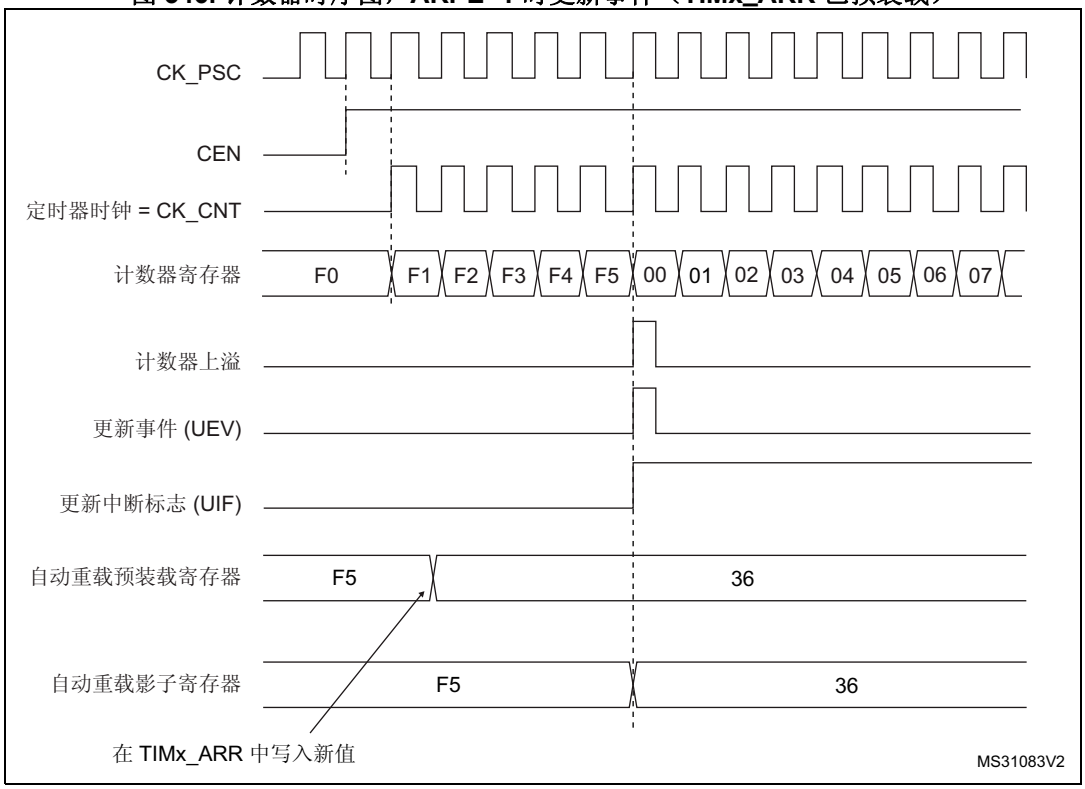


图 346. 计数器时序图，ARPE=1 时更新事件（TIMx_ARR 已预装载）



递减计数模式

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

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

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

通过软件将 TIMx_CR1 寄存器中的 UDIS 位置 1 可禁止 UEV 更新事件。这可避免向预装载寄存器写入新值时更新影子寄存器。在 UDIS 位写入 0 之前不会产生任何更新事件。不过，计数器会重新从当前自动重载值开始计数，而预分频器计数器则重新从 0 开始计数（但预分频比保持不变）。

此外，如果 TIMx_CR1 寄存器中的 URS 位（更新请求选择）已置 1，则将 UG 位置 1 会生成更新事件 UEV，但不会将 UIF 标志置 1（因此，不会发送任何中断或 DMA 请求）。这样一来，如果在发生捕获事件时将计数器清零，将不会同时产生更新中断和捕获中断。

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

- 重复计数器中将重新装载 TIMx_RCR 寄存器的内容。
- 预分频器的缓冲区中将重新装载预装载值（TIMx_PSC 寄存器的内容）。
- 自动重载有效寄存器将以预装载值（TIMx_ARR 寄存器的内容）进行更新。注意，ARR 寄存器更新在计数器重载之前被更新，因此下一个周期就是预期的值。

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

图 347. 计数器时序图，1 分频内部时钟

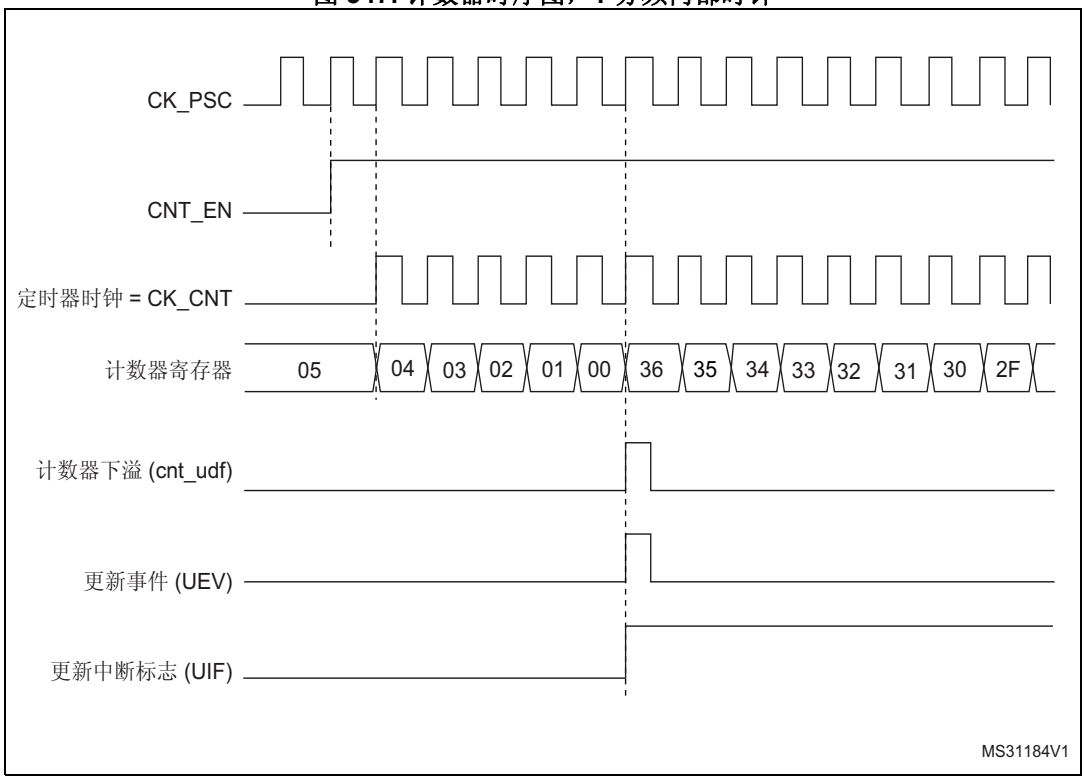


图 348. 计数器时序图，2 分频内部时钟

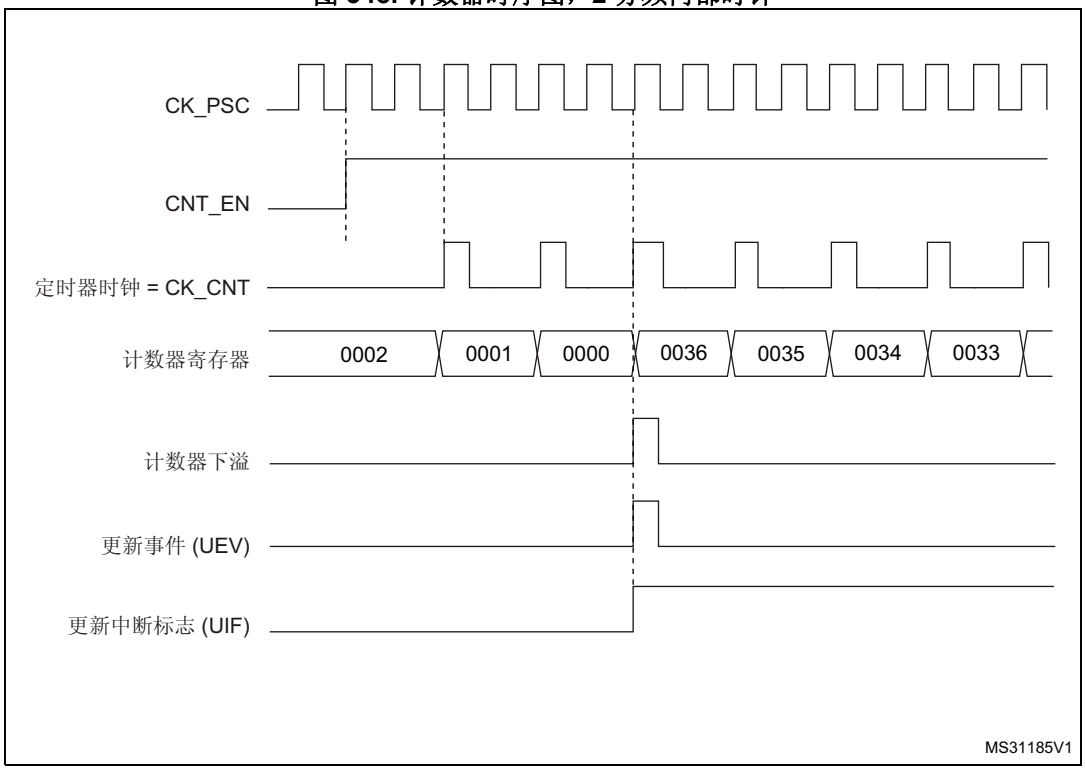


图 349. 计数器时序图，4 分频内部时钟

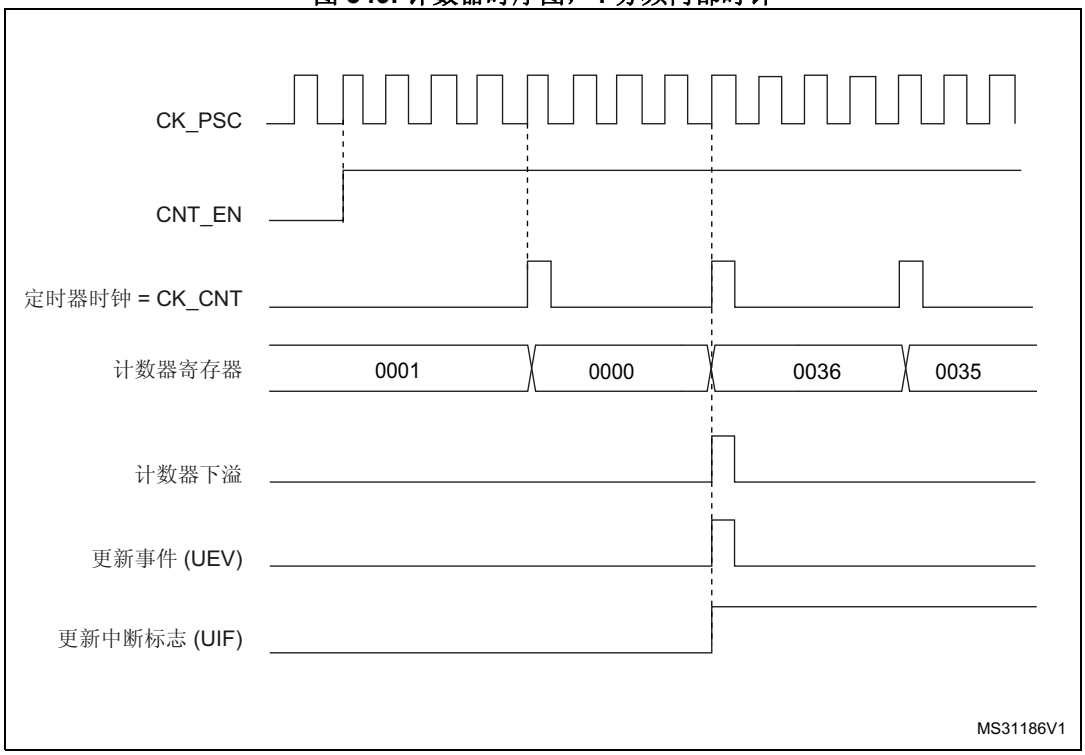


图 350. 计数器时序图，N 分频内部时钟

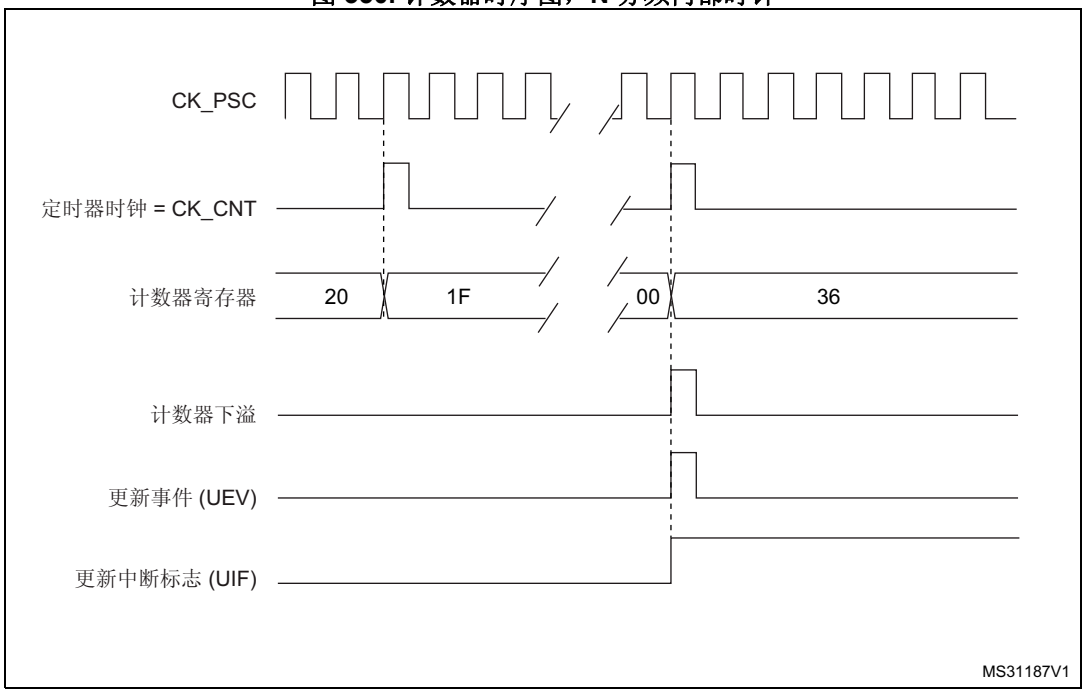
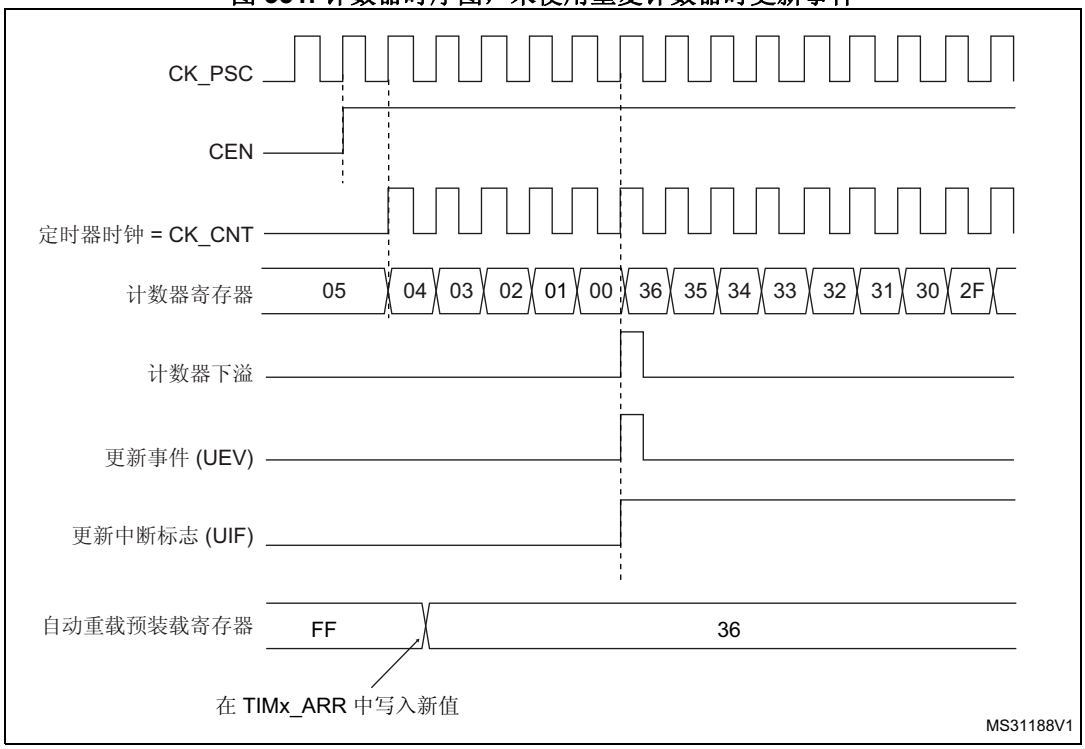


图 351. 计数器时序图，未使用重复计数器时更新事件



中心对齐模式（递增/递减计数）

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

当 TIMx_CR1 寄存器中的 CMS 位不为“00”时，中心对齐模式有效。将通道配置为输出模式时，其输出比较中断标志将在以下模式下置 1，即：计数器递减计数（中心对齐模式 1，CMS = “01”）、计数器递增计数（中心对齐模式 2，CMS = “10”）以及计数器递增/递减计数（中心对齐模式 3，CMS = “11”）。

在此模式下，TIMx_CR1 寄存器的 DIR 方向位不可写入值， 而是由硬件更新并指示当前计数器方向。

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

UEV 更新事件可通过软件禁止，只需将 TIMx_CR1 寄存器中的 UDIS 位置 1。这可避免向预装载寄存器写入新值时更新影子寄存器。在 UDIS 位写入 0 之前不会产生任何更新事件。不过，计数器仍会根据当前自动重载值进行递增和递减计数。

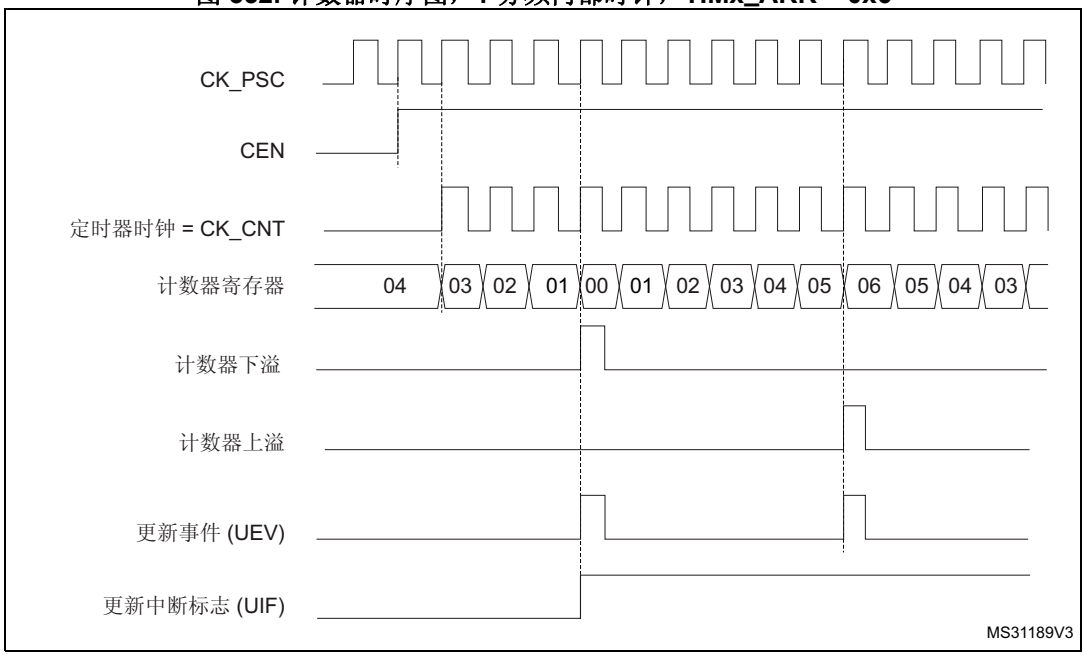
此外，如果 TIMx_CR1 寄存器中的 URS 位（更新请求选择）已置 1，则将 UG 位置 1 会产生 UEV 更新事件，但不会将 UIF 标志置 1（因此，不会发送任何中断或 DMA 请求）。这样一来，如果在发生捕获事件时将计数器清零，将不会同时产生更新中断和捕获中断。

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

- 重复计数器中将重新装载 TIMx_RCR 寄存器的内容。
- 预分频器的缓冲区中将重新装载预装载值（TIMx_PSC 寄存器的内容）。
- 自动重载有效寄存器将以预装载值（TIMx_ARR 寄存器的内容）进行更新。注意，如果更新操作是由计数器上溢触发的，则 ARR 寄存器在计数器重载之前更新，因此，下一个计数周期就是我们所希望的新的周期长度（计数器被重载新的值）。

以下各图以一些示例说明不同时钟频率下计数器的行为。

图 352. 计数器时序图，1 分频内部时钟，TIMx_ARR = 0x6



1. 此处使用中心对齐模式 1（有关详细信息，请参见*第 38.4 节：TIM1/TIM8 寄存器*）。

图 353. 计数器时序图，2 分频内部时钟

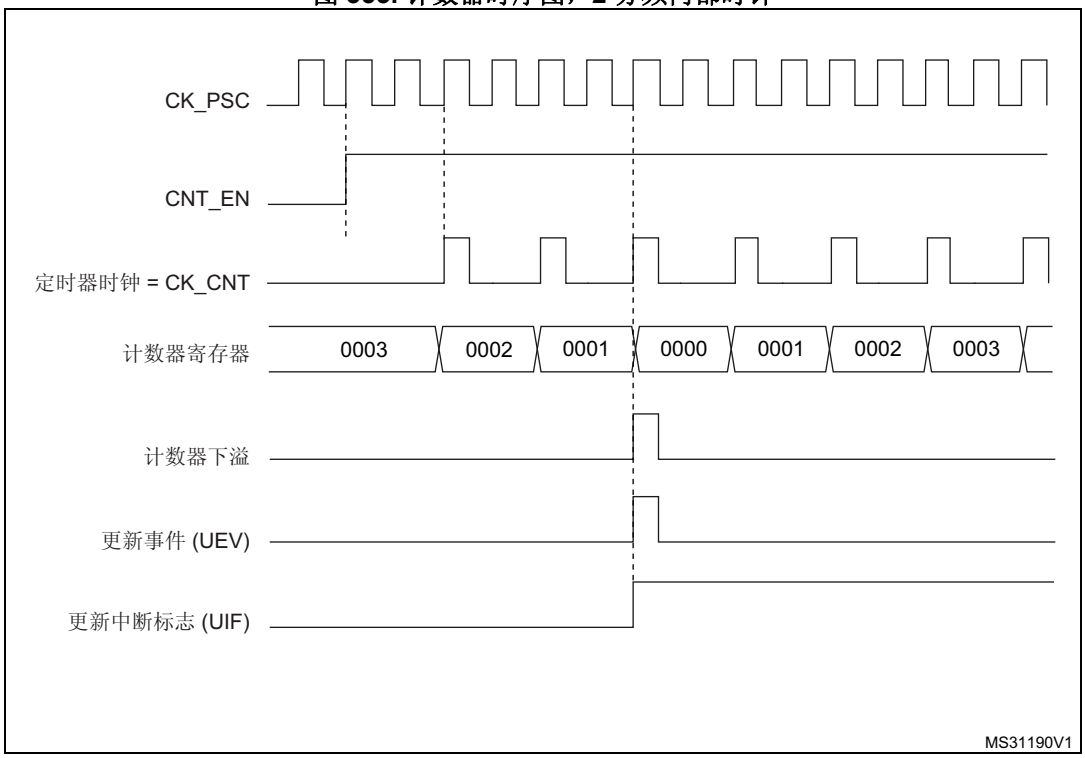


图 354. 计数器时序图，4 分频内部时钟，TIMx_ARR=0x36

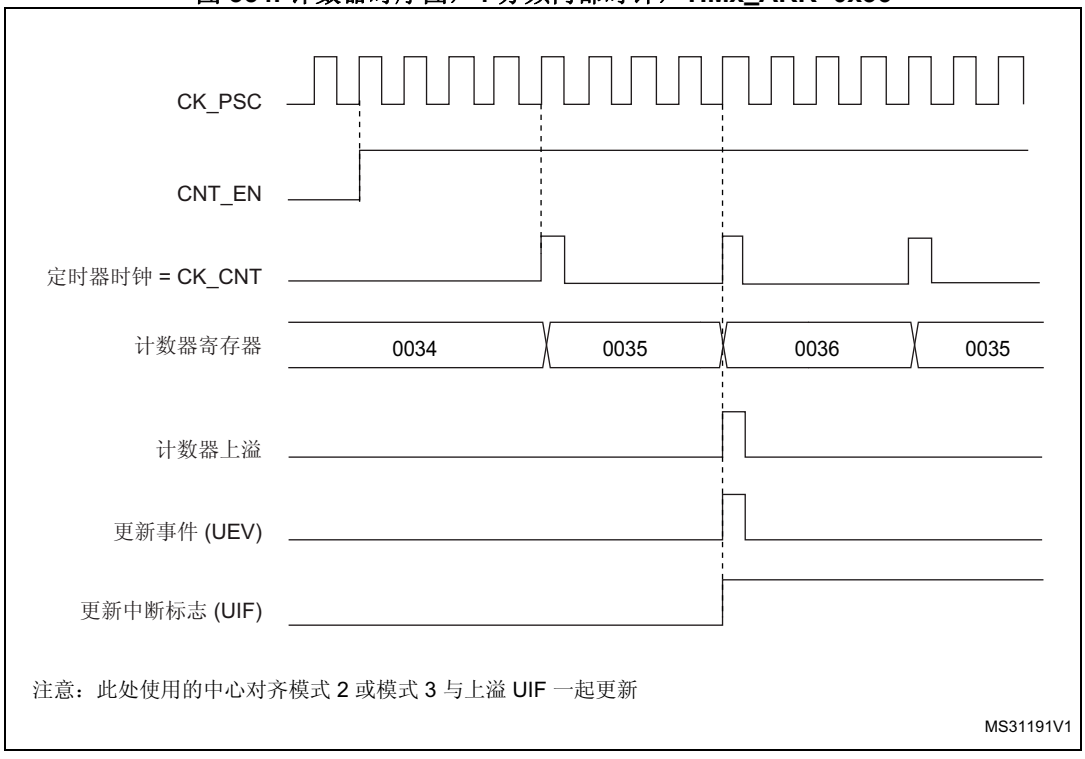


图 355. 计数器时序图，N 分频内部时钟

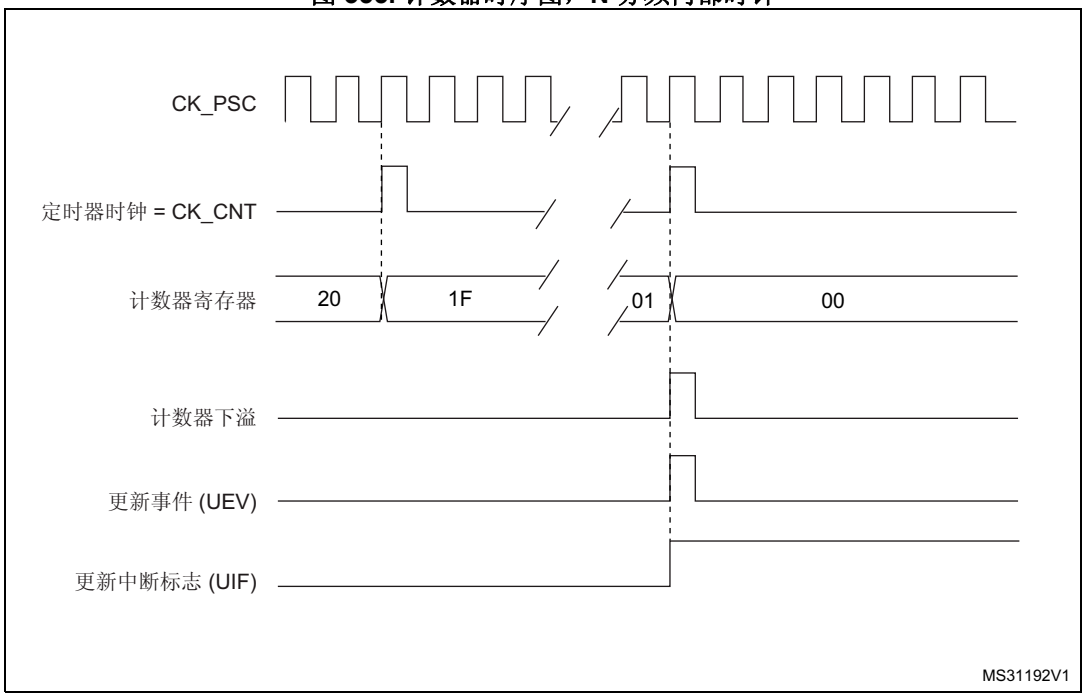


图 356. 计数器时序图，ARPE=1 时的更新事件（计数器下溢）

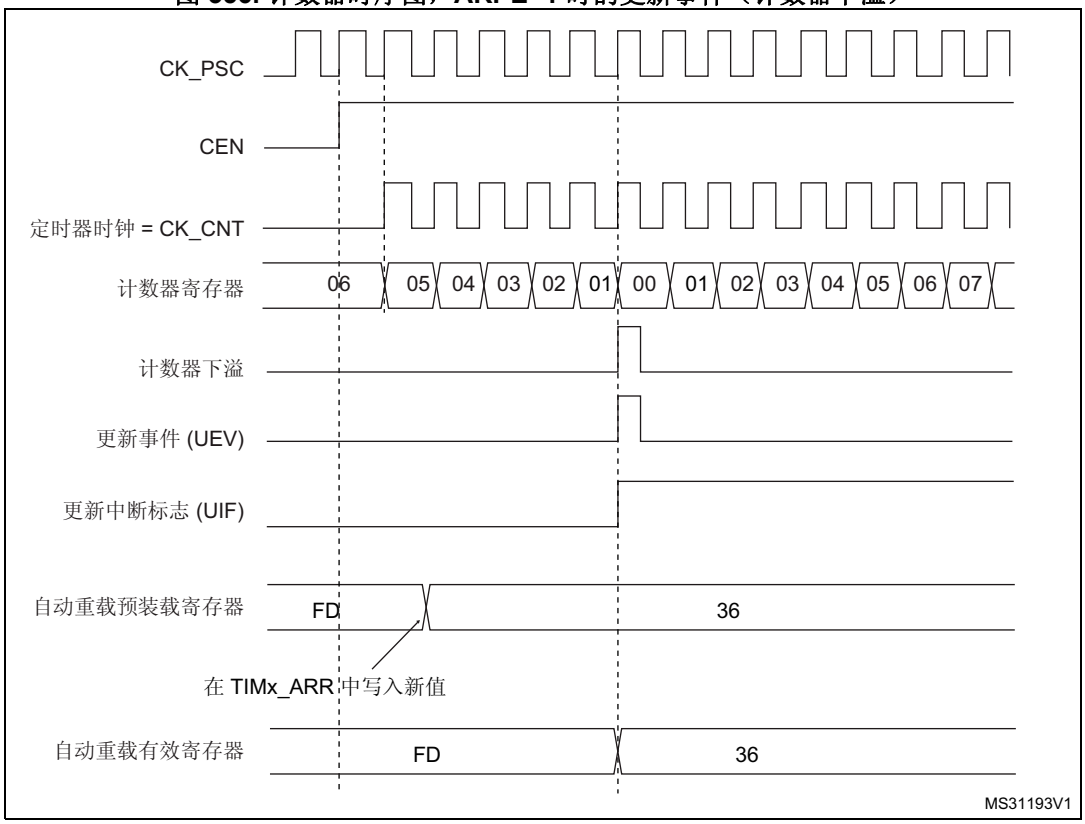
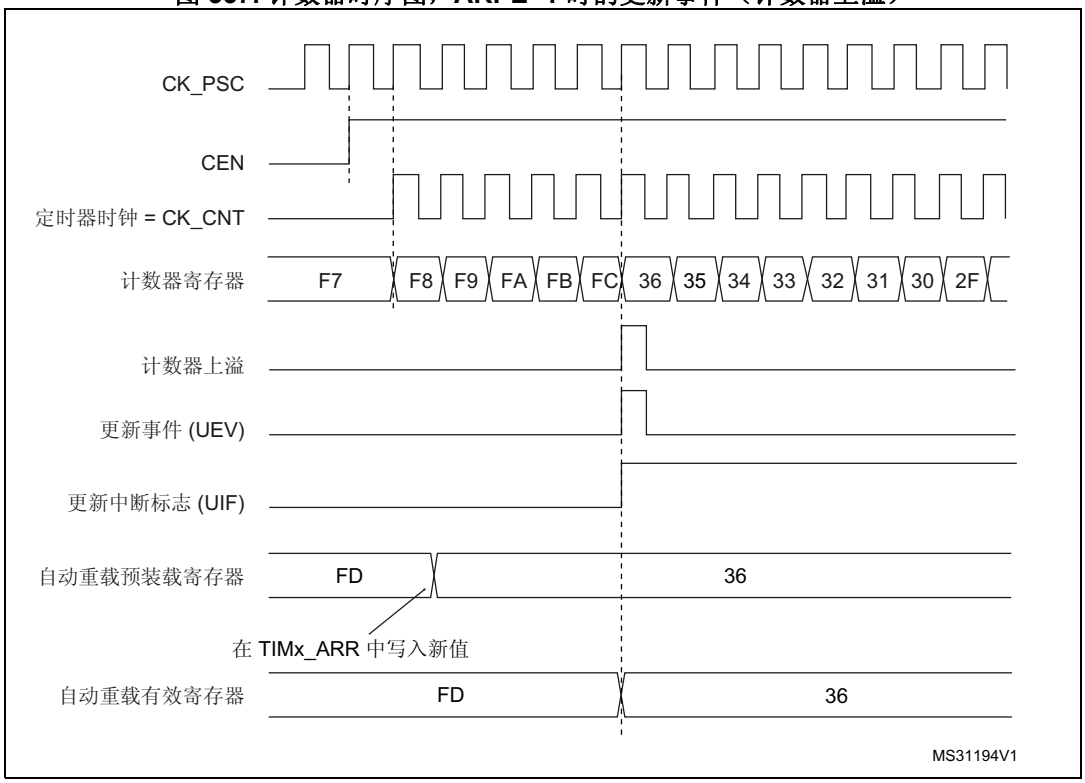


图 357. 计数器时序图，ARPE=1 时的更新事件（计数器上溢）



38.3.3 重复计数器

*第 38.3.1 节：时基单元*介绍如何因计数器上溢/下溢而生成更新事件 (UEV)。实际上，只有当重复计数器达到零时，才会生成更新事件。这在生成 PWM 信号时很有用。

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

重复计数器在下列情况下递减：

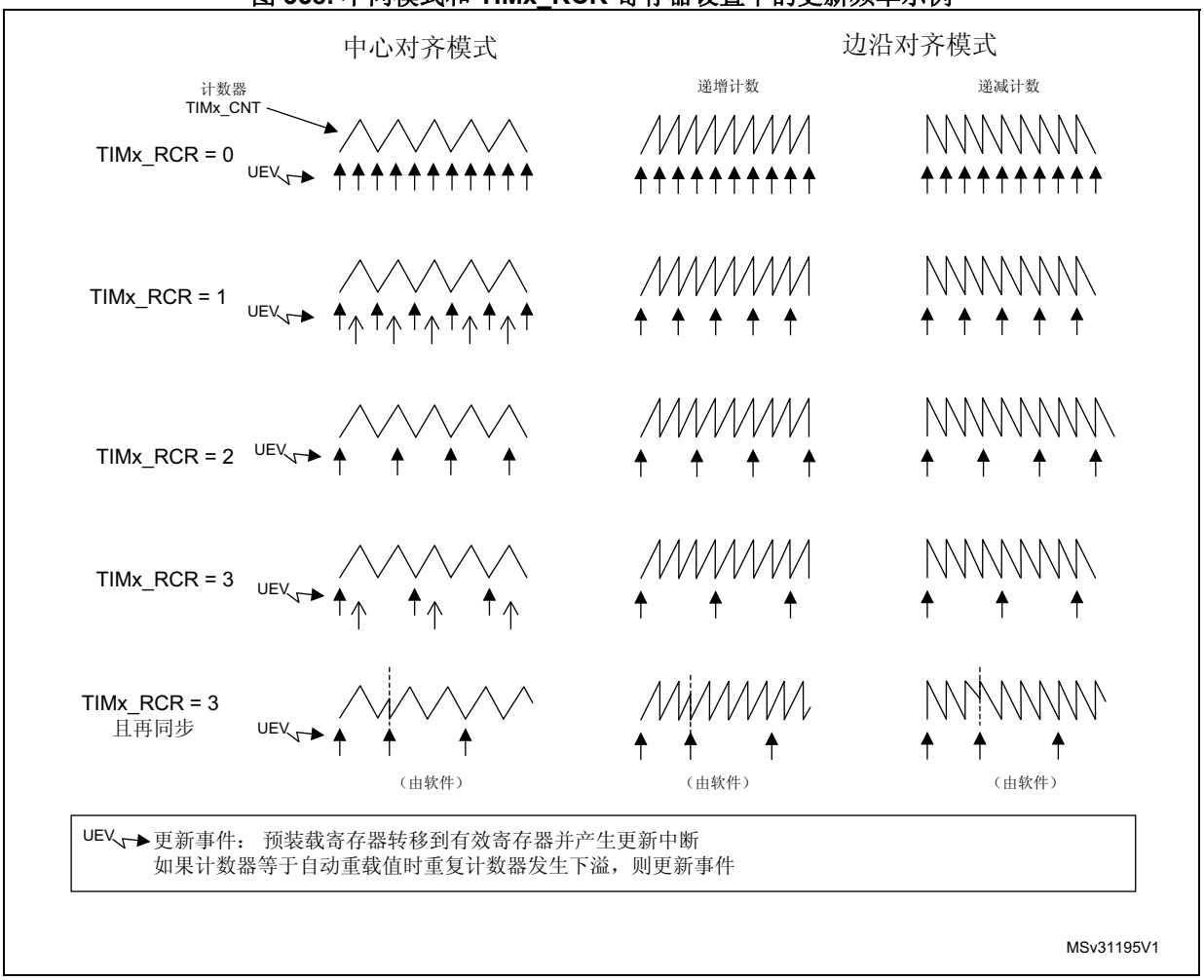
- 递增计数模式下的每个计数器上溢。
- 递减计数模式下的每个计数器下溢。
- 中心对齐模式下每个计数器上溢和计数器下溢。尽管这使得最大重复次数不超过 32768 个 PWM 周期，但在每个 PWM 周期内可更新占空比两次。当在中心对齐模式下，每个 PWM 周期仅刷新一次比较寄存器时，由于模式的对称性，最大分辨率为 $2xT_{ck}$。

重复计数器是自动重载类型；其重复率为 TIMx_RCR 寄存器所定义的值（请参见*图 358*）。当更新事件由软件（通过将 TIMx_EGR 寄存器的 UG 位置 1）或硬件（通过从模式控制器）生成时，无论重复计数器的值为多少，更新事件都将立即发生，并且在重复计数器中重新装载 TIMx_RCR 寄存器的内容。

在中心对齐模式下，如果 RCR 值为奇数，更新事件将在上溢或下溢时发生，这取决于何时写入 RCR 寄存器以及何时启动计数器：如果在启动计数器前写入 RCR，则 UEV 在上溢时发生。如果在启动计数器后写入 RCR，则 UEV 在下溢时发生。

例如，如果 RCR = 3，UEV 将在每个周期的第四个上溢或下溢事件时产生（取决于何时写入 RCR）。

图 358. 不同模式和 TIMx_RCR 寄存器设置下的更新频率示例



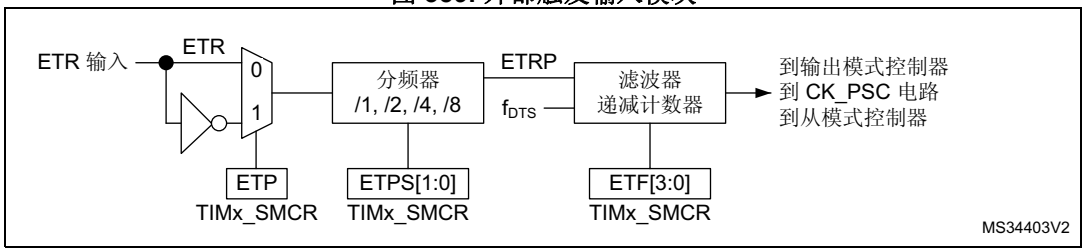
38.3.4 外部触发输入

定时器具有一个外部触发输入 ETR，它可用作：

- 外部时钟（外部时钟模式 2，请参见*第 38.3.5 节*）
- 用于从模式的触发信号（请参见*第 38.3.26 节*）
- 用于逐周期电流调节的 PWM 复位输入（请参见*第 38.3.7 节*）

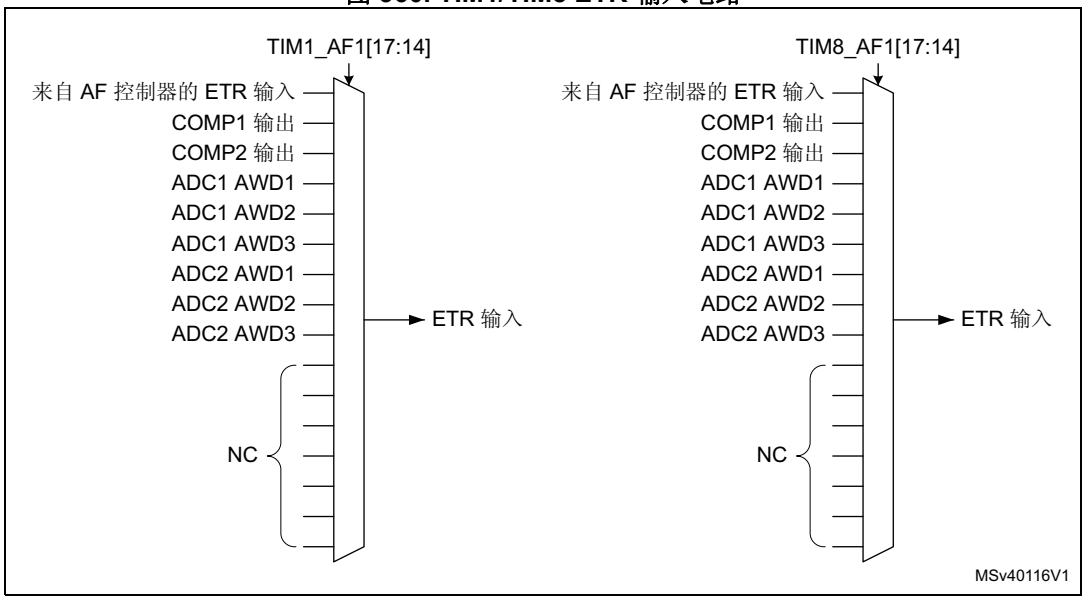
下面的*图 359* 介绍了 ETR 输入的调节过程。输入极性通过 TIMx_SMCR 寄存器中的 ETP 位定义。触发信号可通过 ETPS[1:0] 位域编程的分频比进行预分频，然后通过 ETF[3:0] 位域进行数字滤波。

图 359. 外部触发输入模块



ETR 输入来自多个源：输入引脚（默认配置）、比较器输出和模拟看门狗。使用 ETRSEL[3:0] 位域进行选择。

图 360. TIM1/TIM8 ETR 输入电路



38.3.5 时钟选择

计数器时钟可由下列时钟源提供：

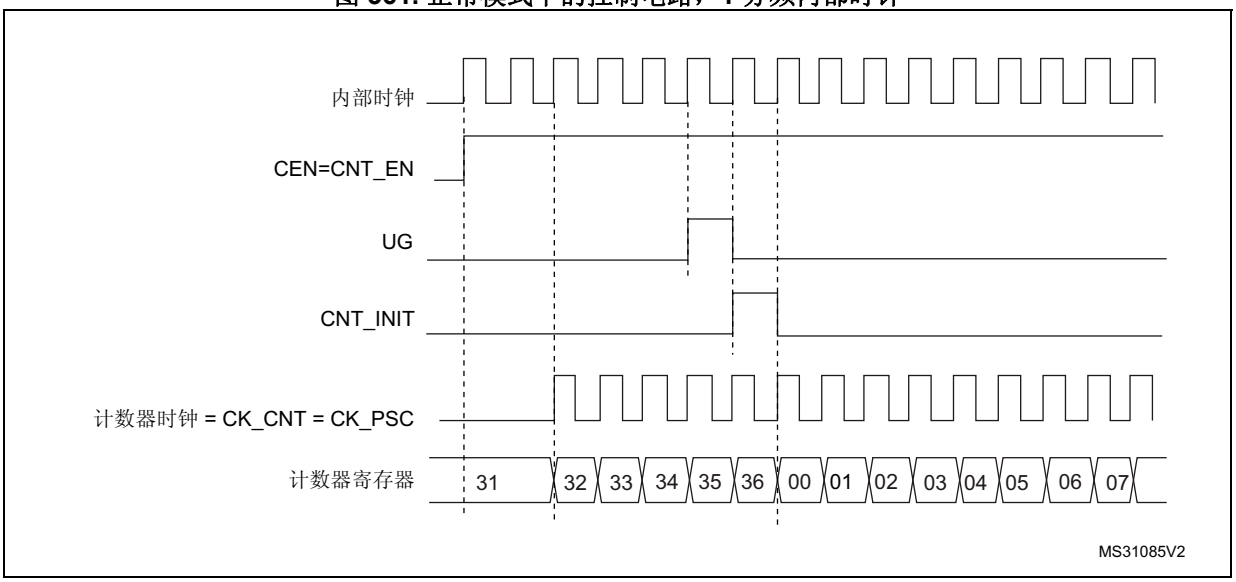
- 内部时钟 (CK_INT)
- 外部时钟模式 1：外部输入引脚
- 外部时钟模式 2：外部触发输入 ETR
- 编码器模式

内部时钟源 (CK_INT)

如果禁止从模式控制器 (SMS=000)，则 CEN 位、DIR 位（TIMx_CR1 寄存器中）和 UG 位（TIMx_EGR 寄存器中）为实际控制位，并且只能通过软件进行更改（UG 除外，仍保持自动清零）。当对 CEN 位写入 1 时，预分频器的时钟就由内部时钟 CK_INT 提供。

图 361 显示了正常模式下控制电路与递增计数器的行为（没有预分频的情况下）。

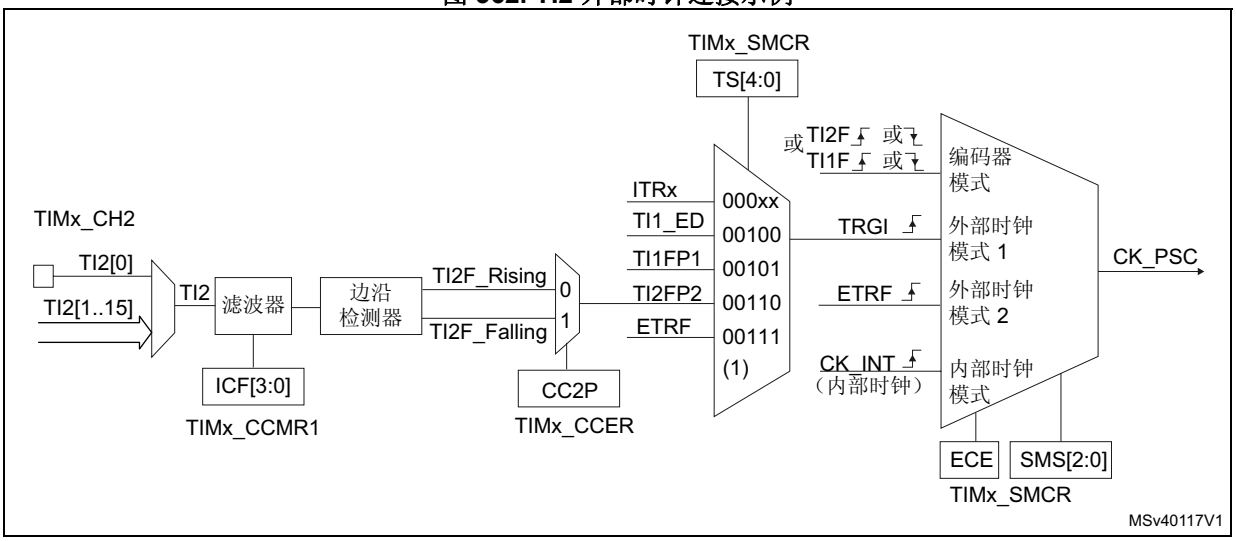
图 361. 正常模式下的控制电路，1 分频内部时钟



外部时钟源模式 1

当 TIMx_SMCR 寄存器中的 SMS=111 时，可选择此模式。 计数器可在选定的输入信号上出现上升沿或下降沿时计数。

图 362. TI2 外部时钟连接示例



1. 保留从 01000 到 11111 的代码

例如，要使递增计数器在 TI2 输入出现上升沿时计数，请执行以下步骤：

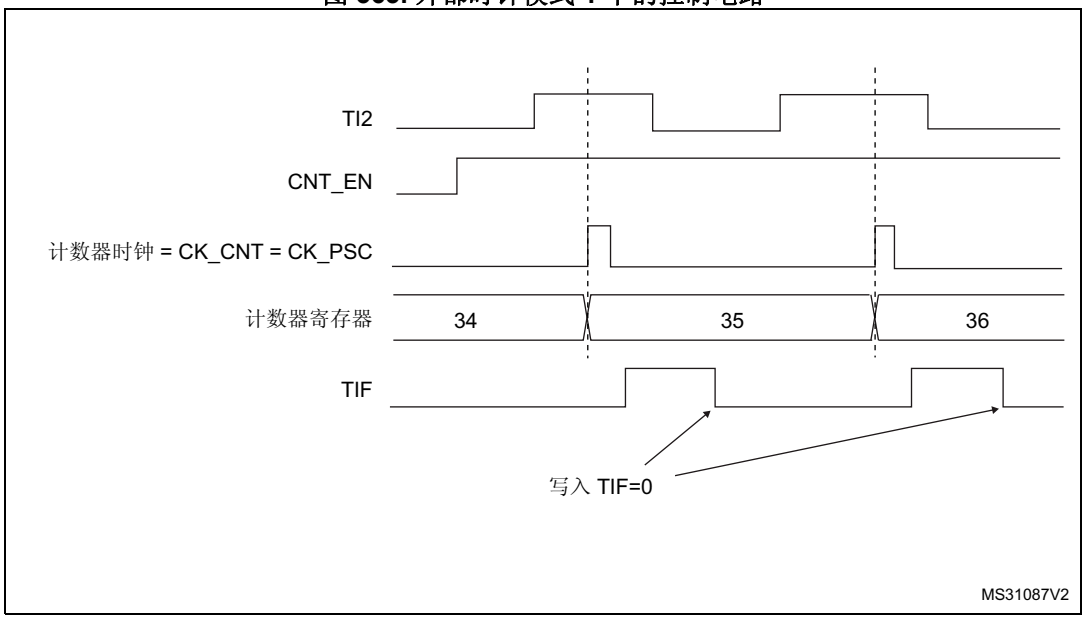
1. 通过在 TIMx_CCMR1 寄存器中写入 CC2S = “01” 来配置通道 2，使其能够检测 TI2 输入的上升沿。
2. 通过在 TIMx_CCMR1 寄存器中写入 IC2F[3:0] 位来配置输入滤波带宽（如果不需要任何滤波器，请保持 IC2F=0000）。
3. 通过在 TIMx_CCER 寄存器中写入 CC2P=0 和 CC2NP=0 来选择上升沿极性。
4. 通过在 TIMx_SMCR 寄存器中写入 SMS=111，使定时器在外部时钟模式 1 下工作。
5. 通过在 TIMx_SMCR 寄存器中写入 TS=00110 来选择 TI2 作为触发输入源。
6. 通过在 TIMx_CR1 寄存器中写入 CEN=1 来使能计数器。

注： *由于捕获预分频器不用于触发操作，因此用户无需对其进行配置。*

当 TI2 出现上升沿时，计数器便会计数一次并且 TIF 标志置 1。

TI2 的上升沿与实际计数器时钟之间的延迟是由于 TI2 输入的重新同步电路引起的。

图 363. 外部时钟模式 1 下的控制电路



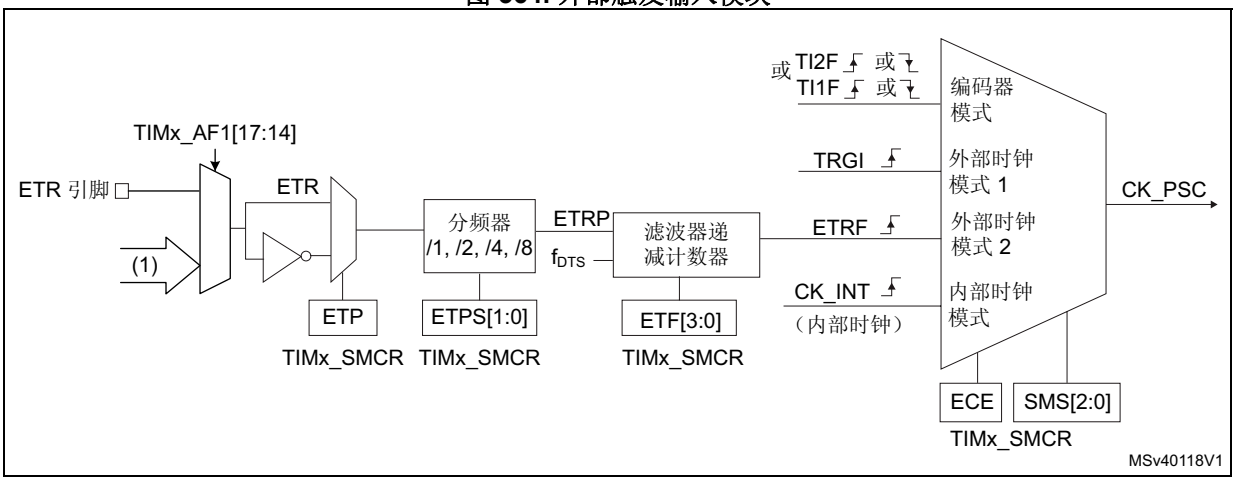
外部时钟源模式 2

通过在 TIMx_SMCR 寄存器中写入 ECE=1 可选择此模式。

计数器可在外部触发输入 ETR 出现上升沿或下降沿时计数。

图 364 简要介绍了外部触发输入模块。

图 364. 外部触发输入模块



1. 请参见*图 360：TIM1/TIM8 ETR 输入电路*。

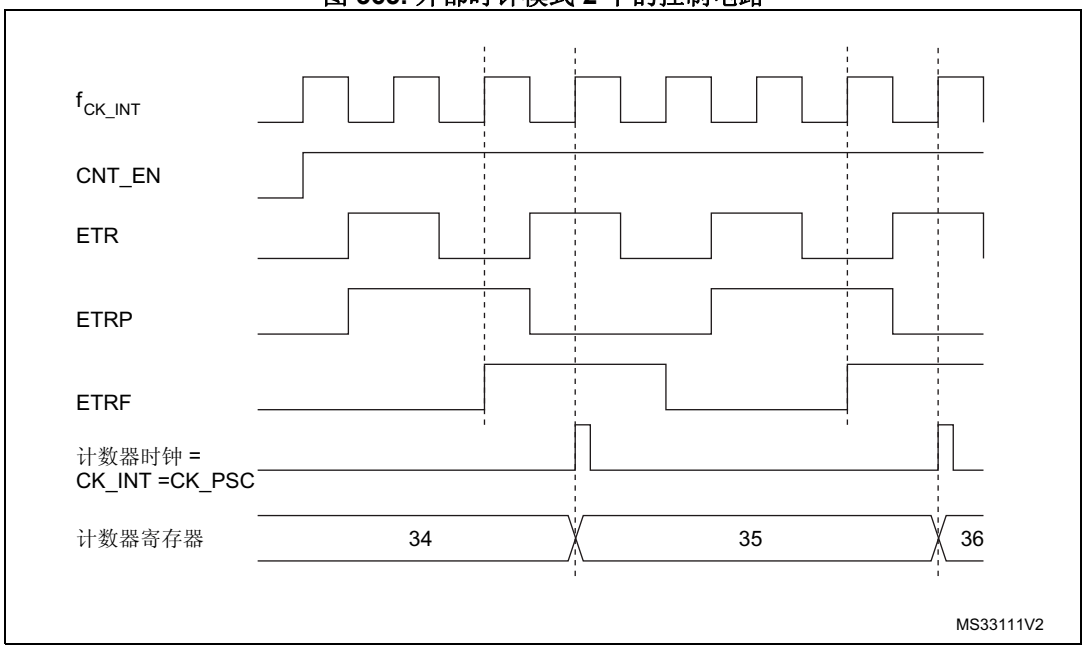
例如，要使递增计数器在 ETR 每出现 2 个上升沿时计数，请执行以下步骤：

1. 由于此例中不需滤波器，因此在 TIMx_SMCR 寄存器中写入 ETF[3:0]=0000。
2. 通过在 TIMx_SMCR 寄存器中写入 ETPS[1:0]=01 来设置预分频器。
3. 通过在 TIMx_SMCR 寄存器中写入 ETP=0 来选择 ETR 引脚的上升沿检测。
4. 通过在 TIMx_SMCR 寄存器中写入 ECE=1 来使能外部时钟模式 2。
5. 通过在 TIMx_CR1 寄存器中写入 CEN=1 来使能计数器。

ETR 每出现 2 个上升沿，计数器计数一次。

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

图 365. 外部时钟模式 2 下的控制电路



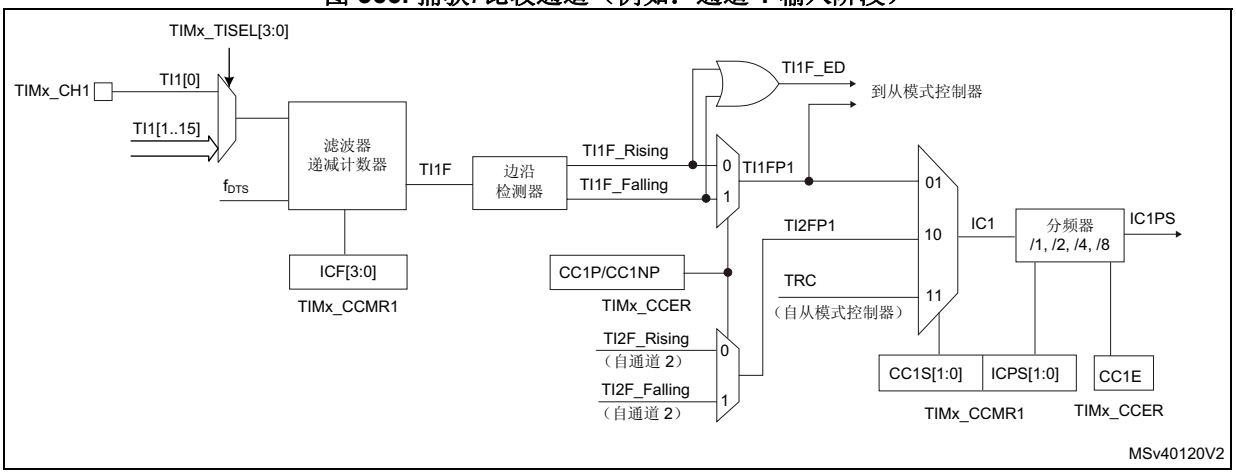
38.3.6 捕获/比较通道

每个捕获/比较通道均围绕一个捕获/比较寄存器（包括一个影子寄存器）、一个捕获输入阶段（数字滤波、多路复用和预分频器，通道 5 和通道 6 除外）和一个输出阶段（比较器和输出控制）构建而成。

图 366 到*图 369* 概括介绍了一个捕获/比较通道。

输入阶段对相应的 TIx 输入进行采样，生成一个滤波后的信号 TIxF。然后，带有极性选择功能的边沿检测器生成一个信号 (TIxFPx)，该信号可用作从模式控制器的触发输入，也可用作捕获命令。该信号先进行预分频 (ICxPS)，而后再进入捕获寄存器。

图 366. 捕获/比较通道（例如：通道 1 输入阶段）



输出阶段生成一个中间波形作为基准：OCxRef（高电平有效）。链的末端决定最终输出信号的极性。

图 367. 捕获/比较通道 1 主电路

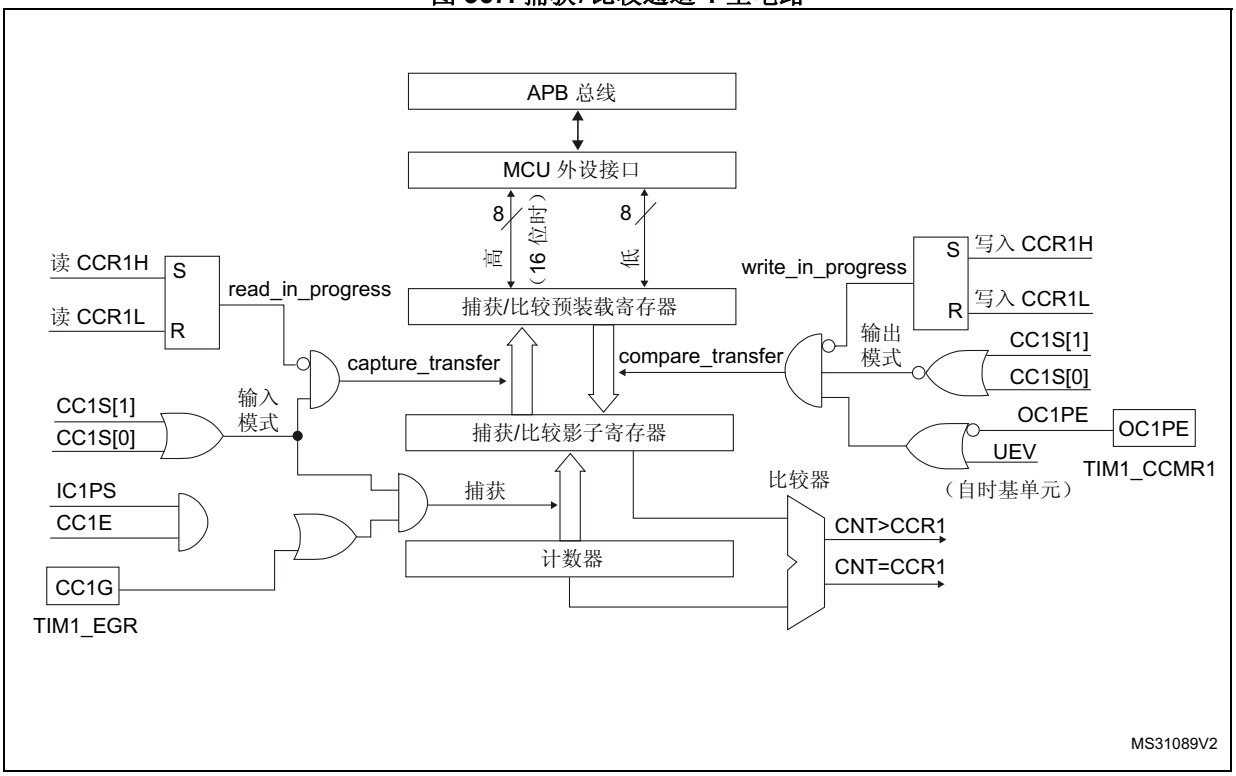
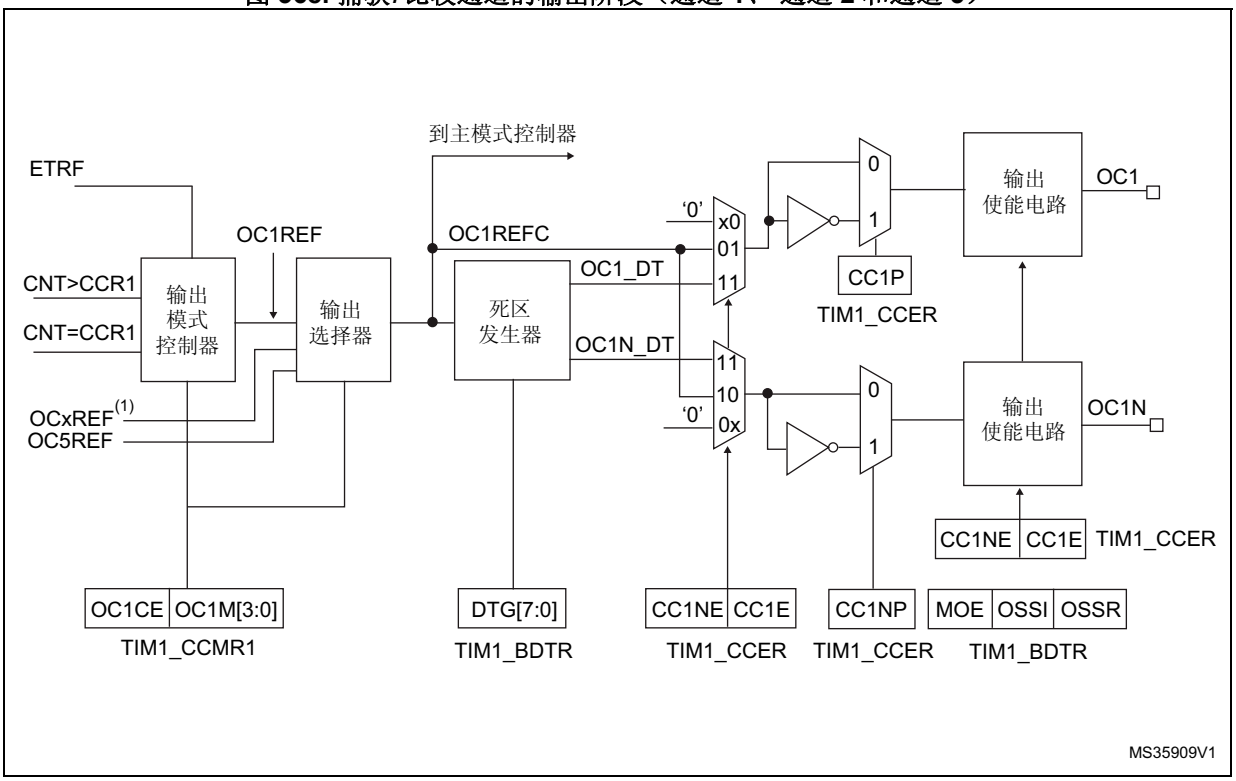


图 368. 捕获/比较通道的输出阶段（通道 1、 通道 2 和通道 3）



1. OCxREF，其中 x 为互补通道的序号。

图 369. 捕获/比较通道的输出阶段（通道 4）

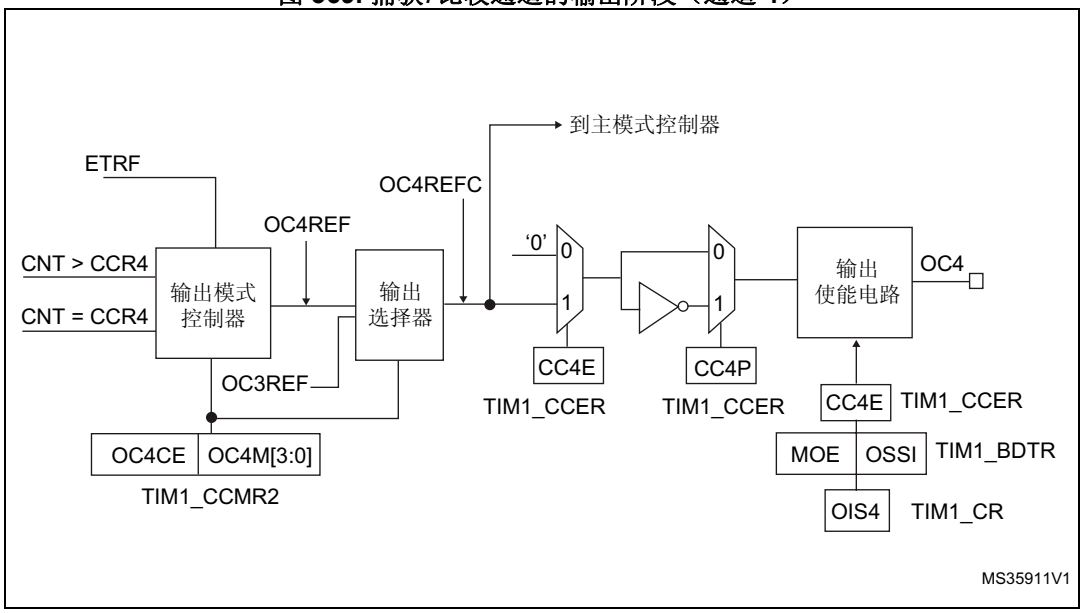
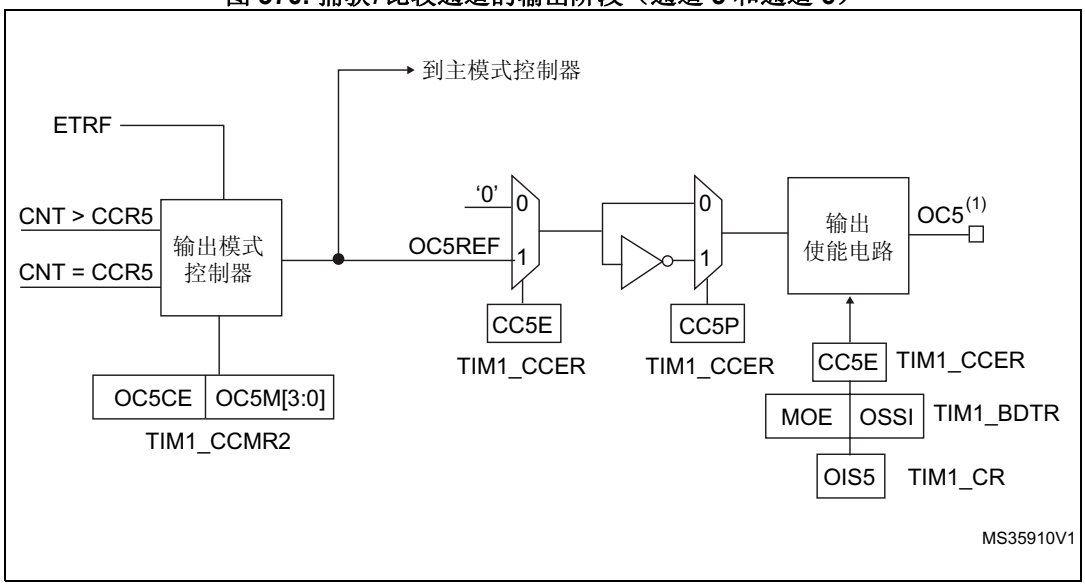


图 370. 捕获/比较通道的输出阶段（通道 5 和通道 6）



1. 不适用于外部。

捕获/比较模块由一个预装载寄存器和一个影子寄存器组成。始终可通过读写操作访问预装载寄存器。

在捕获模式下，捕获实际发生在影子寄存器中，然后将影子寄存器的内容复制到预装载寄存器中。

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

38.3.7 输入捕获模式

在输入捕获模式下，当相应的 ICx 信号检测到跳变沿后，将使用捕获/比较寄存器 (TIMx_CCRx) 来锁存计数器的值。发生捕获事件时，会将相应的 CCXIF 标志（TIMx_SR 寄存器）置 1，并可发送中断或 DMA 请求（如果已使能）。如果发生捕获事件时 CCxIF 标志已处于高位，则会将重复捕获标志 CCxOF（TIMx_SR 寄存器）置 1。可通过软件将 CCxIF 清零，方法是：向 CCxIF 写入“0”，或读取存储在 TIMx_CCRx 寄存器中的已捕获数据。向 CCxOF 写入“0”后会将其清零。

以下示例说明了如何在 TI1 输入出现上升沿时将计数器的值捕获到 TIMx_CCR1 中。具体操作步骤如下：

- 选择有效输入：TIMx_CCR1 必须连接到 TI1 输入，因此向 TIMx_CCMR1 寄存器中的 CC1S 位写入 01。只要 CC1S 不等于 00，就会将通道配置为输入模式，并且 TIMx_CCR1 寄存器将处于只读状态。
- 根据连接到定时器的信号，对所需的输入滤波带宽进行编程（如果输入为 TIx 之一，则对 TIMx_CCMRx 寄存器中的 ICxF 位进行编程）。假设信号边沿变化时，输入信号最多在 5 个内部时钟周期内发生抖动。因此，我们必须将滤波带宽设置为大于 5 个内部时钟周期。在检测到 8 个具有新电平的连续采样（以 f_{DTS} 频率采样）后，可以确认 TI1 上的跳变沿。然后向 TIMx_CCMR1 寄存器中的 IC1F 位写入 0011。
- 通过在 TIMx_CCER 寄存器中将 CC1P 位和 CC1NP 位写入 0，选择 TI1 上的有效转换边沿（本例中为上升沿）。
- 对输入预分频器进行编程。在本例中，我们希望每次有效转换时都执行捕获操作，因此需要禁止预分频器（向 TIMx_CCMR1 寄存器中的 IC1PS 位写入“00”）。
- 通过将 TIMx_CCER 寄存器中的 CC1E 位置 1，允许将计数器的值捕获到捕获寄存器中。
- 如果需要，可通过将 TIMx_DIER 寄存器中的 CC1IE 位置 1 来使能相关中断请求，并且/或者通过将该寄存器中的 CC1DE 位置 1 来使能 DMA 请求。

发生输入捕获时：

- 发生有效跳变沿时，TIMx_CCR1 寄存器会获取计数器的值。
- 将 CC1IF 标志置 1（中断标志）。如果至少发生了两次连续捕获，但 CC1IF 标志未被清零，这样 CC1OF 捕获溢出标志会被置 1。
- 根据 CC1IE 位生成中断。
- 根据 CC1DE 位生成 DMA 请求。

要处理重复捕获，建议在读出捕获溢出标志之前读取数据。这样可避免丢失在读取捕获溢出标志之后与读取数据之前可能出现的重复捕获信息。

注： *通过软件将 TIMx_EGR 寄存器中的相应 CCxG 位置 1 可生成 IC 中断和/或 DMA 请求。*

38.3.8 PWM 输入模式

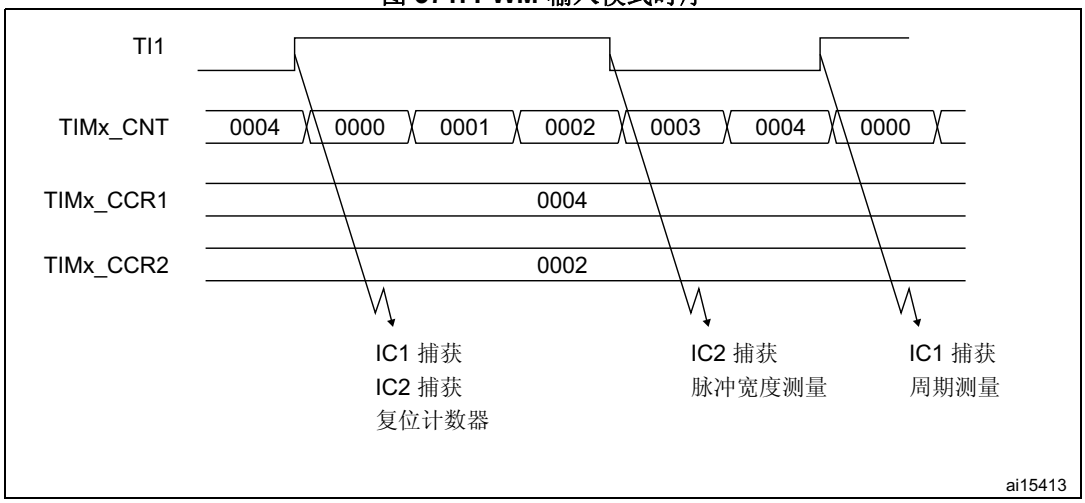
此模式是输入捕获模式的一个特例。其实现步骤与输入捕获模式基本相同，仅存在以下不同之处：

- 两个 ICx 信号被映射至同一个 TIx 输入。
- 这两个 ICx 信号在边沿处有效，但极性相反。
- 选择两个 TIxFP 信号之一作为触发输入，并将从模式控制器配置为复位模式。

例如，用户可通过以下步骤对应用于 TI1 的 PWM 的周期（位于 TIMx_CCR1 寄存器中）和占空比（位于 TIMx_CCR2 寄存器中）进行测量（取决于 CK_INT 频率和预分频器的值）：

- 选择 TIMx_CCR1 的有效输入：向 TIMx_CCMR1 寄存器中的 CC1S 位写入 01（选择 TI1）。
- 选择 TI1FP1 的有效极性（用于在 TIMx_CCR1 中捕获和计数器清零）：向 CC1P 位和 CC1NP 位写入“0”（上升沿有效）。
- 选择 TIMx_CCR2 的有效输入：向 TIMx_CCMR1 寄存器中的 CC2S 写入 10（选择 TI1）。
- 选择 TI1FP2 的有效极性（用于在 TIMx_CCR2 中捕获）：向 CC2P 位和 CC2NP 位写入 CC2P/CC2NP=“10”（下降沿有效）。
- 选择有效触发输入：向 TIMx_SMCR 寄存器中的 TS 位写入 00101（选择 TI1FP1）。
- 将从模式控制器配置为复位模式：向 TIMx_SMCR 寄存器中的 SMS 位写入 0100。
- 使能捕获：向 TIMx_CCER 寄存器中的 CC1E 位和 CC2E 位写入“1”。

图 371. PWM 输入模式时序



38.3.9 强制输出模式

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

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

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

通过向 TIMx_CCMRx 寄存器中的 OCxM 位写入 0100，可将 OCxREF 信号强制设置为低电平。

无论如何，TIMx_CCRx 影子寄存器与计数器之间的比较仍会执行，而且允许将标志置 1。因此可发送相应的中断和 DMA 请求。下面的输出比较模式一节对此进行了介绍。

38.3.10 输出比较模式

此功能用于控制输出波形，或指示已经过某一时间段。通道 1 到通道 4 可用作输出，而通道 5 和通道 6 只能在单片机内部使用（例如，用于产生混合波形或触发 ADC）。

当捕获/比较寄存器与计数器之间相匹配时，输出比较功能：

- 将为相应的输出引脚分配一个可编程值，该值由输出比较模式（TIMx_CCMRx 寄存器中的 OCxM 位）和输出极性（TIMx_CCER 寄存器中的 CCxP 位）定义。匹配时，输出引脚既可保持其电平 (OCxM=0000)，也可设置为有效电平 (OCxM=0001)、无效电平 (OCxM=0010) 或进行翻转 (OCxM=0011)。
- 将中断状态寄存器中的标志置 1（TIMx_SR 寄存器中的 CCxIF 位）。
- 如果相应中断使能位（TIMx_DIER 寄存器中的 CCXIE 位）置 1，将生成中断。
- 如果相应使能位（TIMx_DIER 寄存器的 CCxDE 位，TIMx_CR2 寄存器的 CCDS 位，用来选择 DMA 请求）置 1，将发送 DMA 请求。

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

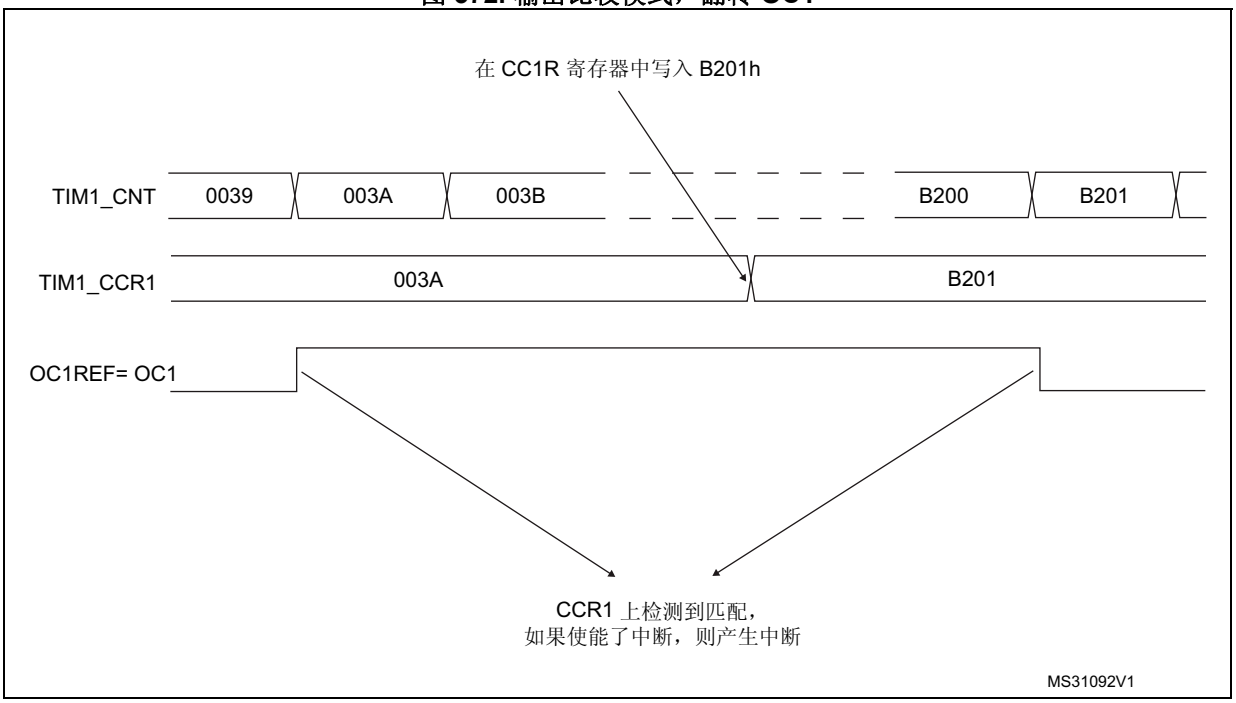
在输出比较模式下，更新事件 UEV 对 OCxREF 和 OCx 输出毫无影响。同步的精度可以达到计数器的一个计数周期。输出比较模式也可用于输出单脉冲（在单脉冲模式下）。

步骤

1. 选择计数器时钟（内部、外部、预分频器）。
2. 在 TIMx_ARR 和 TIMx_CCRx 寄存器中写入所需数据。
3. 如果要生成中断请求，则需将 CCxIE 位置 1。
4. 选择输出模式。例如：
   - 当 CNT 与 CCRx 匹配时，写入 OCxM = 0011 以翻转 OCx 输出引脚
   - 写入 OCxPE = 0 以禁止预装载寄存器
   - 写入 CCxP = 0 以选择高电平有效极性
   - 写入 CCxE = 1 以使能输出
5. 通过将 TIMx_CR1 寄存器中的 CEN 位置 1 来使能计数器。

可通过软件随时更新 TIMx_CCRx 寄存器以控制输出波形，前提是未使能预加载寄存器（OCxPE=“0”，否则 TIMx_CCRx 影子寄存器仅在下一更新事件 UEV 发生时进行更新）。*图 372* 给出了一个示例。

图 372. 输出比较模式，翻转 OC1



38.3.11 PWM 模式

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

各通道可以独立选择 PWM 模式（每个 OCx 输出对应一个 PWM），只需向 TIMx_CCMRx 寄存器的 OCxM 位写入“0110”（PWM 模式 1）或“0111”（PWM 模式 2）。必须通过将 TIMx_CCMRx 寄存器中的 OCxPE 位置 1 使能相应预装载寄存器，最后通过将 TIMx_CR1 寄存器中的 ARPE 位置 1 使能自动重载预装载寄存器（在递增计数或中心对齐模式下）。

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

OCx 极性可通过软件来编程（使用 TIMx_CCER 寄存器的 CCxP 位）。可将其编程为高电平有效或低电平有效。通过 CCxE、CCxNE、MOE、OSSI 和 OSSR 位（TIMx_CCER 和 TIMx_BDTR 寄存器）的组合使能 OCx 输出。有关详细信息，请参见 TIMx_CCER 寄存器说明。

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

根据 TIMx_CR1 寄存器中的 CMS 位状态，定时器能够产生边沿对齐模式或中心对齐模式的 PWM 信号。

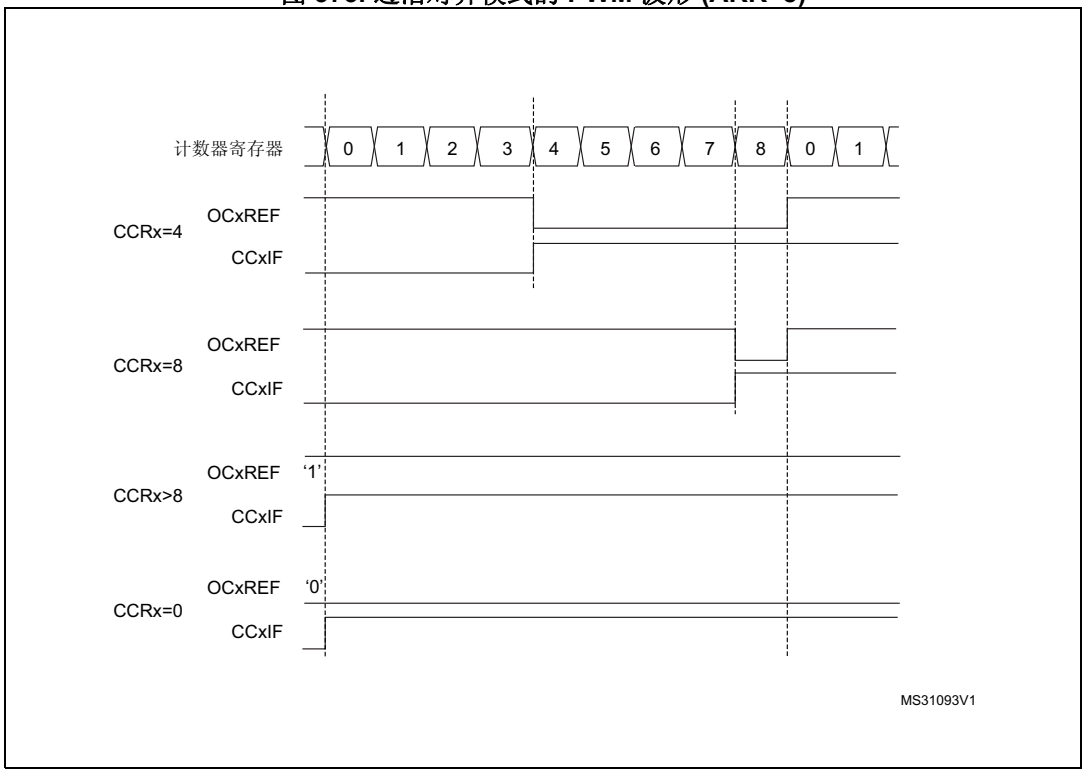
PWM 边沿对齐模式

- 递增计数配置

  当 TIMx_CR1 寄存器中的 DIR 位为低时执行递增计数。请参见*第 1393 页的递增计数模式*。

  以下以 PWM 模式 1 为例。只要 TIMx_CNT < TIMx_CCRx，PWM 参考信号 OCxREF 便为高电平，否则为低电平。如果 TIMx_CCRx 中的比较值大于自动重载值（TIMx_ARR 中），则 OCxREF 保持为“1”。如果比较值为 0，则 OCxRef 保持为“0”。*图 373* 举例介绍边沿对齐模式的一些 PWM 波形 (TIMx_ARR=8)。

图 373. 边沿对齐模式的 PWM 波形 (ARR=8)



- 递减计数配置

  当 TIMx_CR1 寄存器中的 DIR 位为高时执行递减计数。请参见*第 1397 页的递减计数模式*。

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

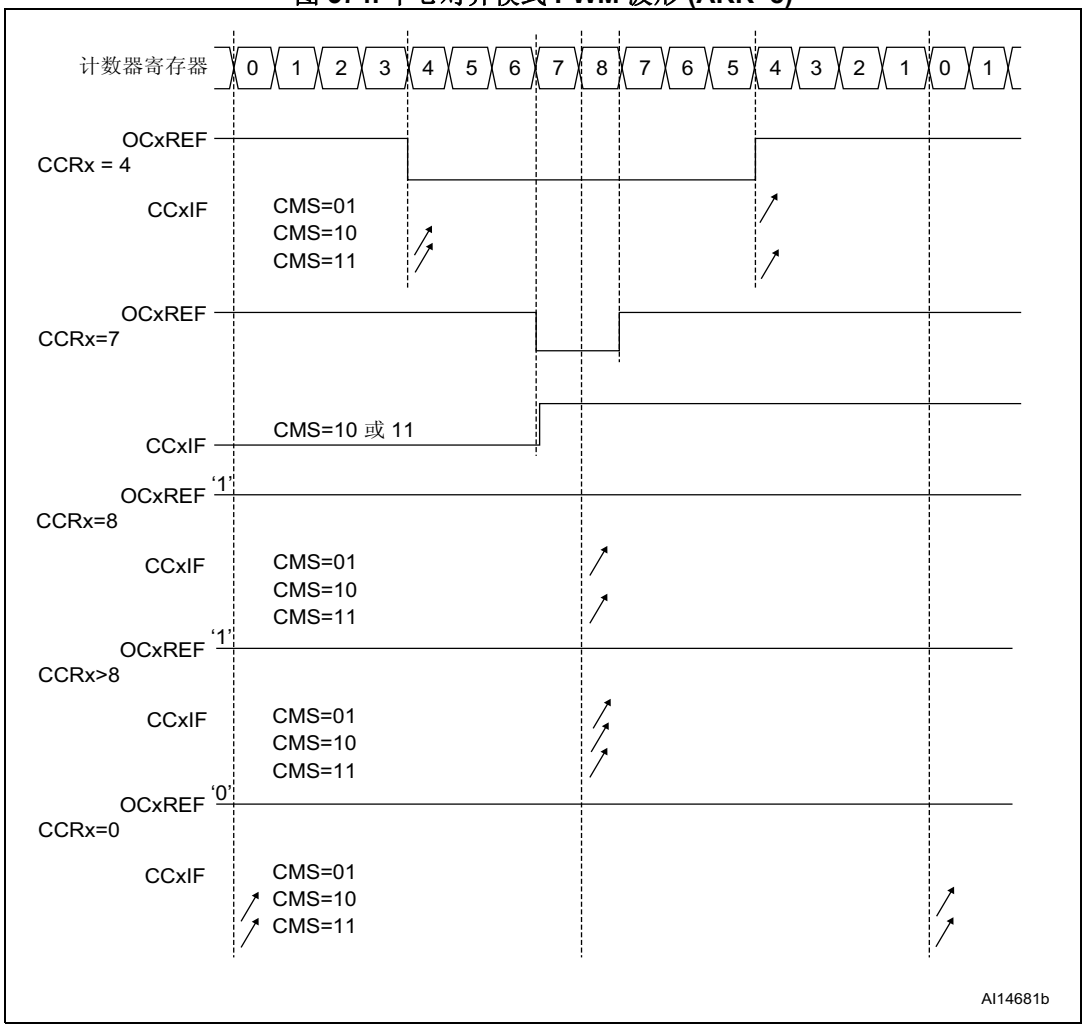
PWM 中心对齐模式

当 TIMx_CR1 寄存器中的 CMS 位不为“00”（其余所有配置对 OCxRef/OCx 信号具有相同的作用），中心对齐模式生效。根据 CMS 位的配置，可以在计数器递增计数、递减计数或同时递增和递减计数时将比较标志置 1。TIMx_CR1 寄存器中的方向位 (DIR) 由硬件更新，不得通过软件更改。请参见*第 1400 页的中心对齐模式（递增/递减计数）*。

图 374 显示了中心对齐模式的 PWM 波形，在此例中：

- TIMx_ARR=8
- PWM 模式为 PWM 模式 1
- 在根据 TIMx_CR1 寄存器中 CMS=01 而选择的中心对齐模式 1 下，当计数器递减计数时，比较标志置 1

图 374. 中心对齐模式 PWM 波形 (ARR=8)



中心对齐模式使用建议

- 启动中心对齐模式时将使用当前的递增/递减计数配置。这意味着计数器将根据写入 TIMx_CR1 寄存器中 DIR 位的值进行递增或递减计数。此外，不得同时通过软件修改 DIR 和 CMS 位。
- 不建议在运行中心对齐模式时对计数器执行写操作，否则将发生意想不到的结果。尤其是：
  - 如果写入计数器中的值大于自动重载值 (TIMx_CNT>TIMx_ARR)，计数方向不会更新。例如，如果计数器之前递增计数，则继续递增计数。
  - 如果向计数器写入 0 或 TIMx_ARR 的值，计数方向会更新，但不生成更新事件 UEV。
- 使用中心对齐模式最为保险的方法是：在启动计数器前通过软件生成更新（将 TIMx_EGR 寄存器中的 UG 位置 1），并且不要在计数器运行过程中对其执行写操作。

38.3.12 不对称 PWM 模式

在不对称模式下，生成的两个中心对齐 PWM 信号间允许存在可编程相移。频率由 TIMx_ARR 寄存器的值确定，而占空比和相移则由一对 TIMx_CCRx 寄存器确定。两个寄存器分别控制递增计数和递减计数期间的 PWM，这样每半个 PWM 周期便会调节一次 PWM：

- OC1REFC（或 OC2REFC）由 TIMx_CCR1 和 TIMx_CCR2 控制
- OC3REFC（或 OC4REFC）由 TIMx_CCR3 和 TIMx_CCR4 控制

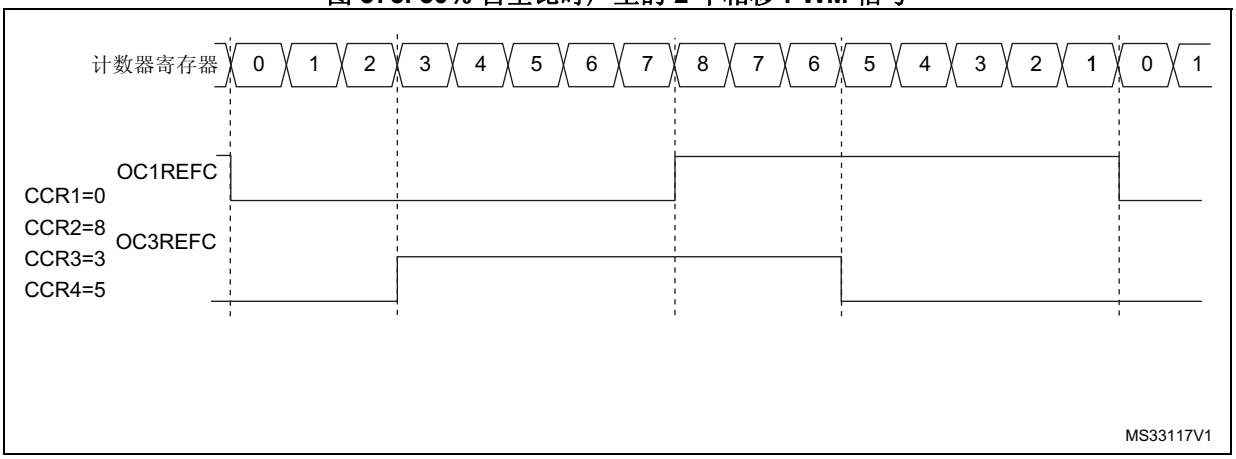
两个通道可以独立选择不对称 PWM 模式（每对 CCR 寄存器一个 OCx 输出），只需向 TIMx_CCMRx 寄存器的 OCxM 位写入“1110”（不对称 PWM 模式 1）或“1111”（不对称 PWM 模式 2）。

注： *出于兼容性原因，OCxM[3:0] 位域分为两部分，最高有效位与最低有效的 3 位不相邻。*

给定通道用作不对称 PWM 通道时，也可使用其互补通道。例如，如果通道 1 上产生 OC1REFC 信号（不对称 PWM 模式 1），则由于不对称 PWM 模式 1 的原因，通道 2 上可输出 OC2REF 信号或 OC2REFC 信号。

图 375 显示了不对称 PWM 模式下可以产生的信号示例（通道 1 到通道 4 在不对称 PWM 模式 1 下配置）。与死区发生器配合使用时，这可控制相移全桥直流到直流转换器。

图 375. 50% 占空比时产生的 2 个相移 PWM 信号



38.3.13 组合 PWM 模式

在组合 PWM 模式下，生成的两个边沿或中心对齐 PWM 信号的各个脉冲间允许存在可编程延时和相移。频率由 TIMx_ARR 寄存器的值确定，而占空比和延时则由两个 TIMx_CCRx 寄存器确定。产生的信号 OCxREFC 由两个参考 PWM 的逻辑或运算或者逻辑与运算组合组成。

- OC1REFC（或 OC2REFC）由 TIMx_CCR1 和 TIMx_CCR2 控制
- OC3REFC（或 OC4REFC）由 TIMx_CCR3 和 TIMx_CCR4 控制

两个通道可以独立选择组合 PWM 模式（每对 CCR 寄存器一个 OCx 输出），只需向 TIMx_CCMRx 寄存器的 OCxM 位写入“1100”（组合 PWM 模式 1）或“1101”（组合 PWM 模式 2）。

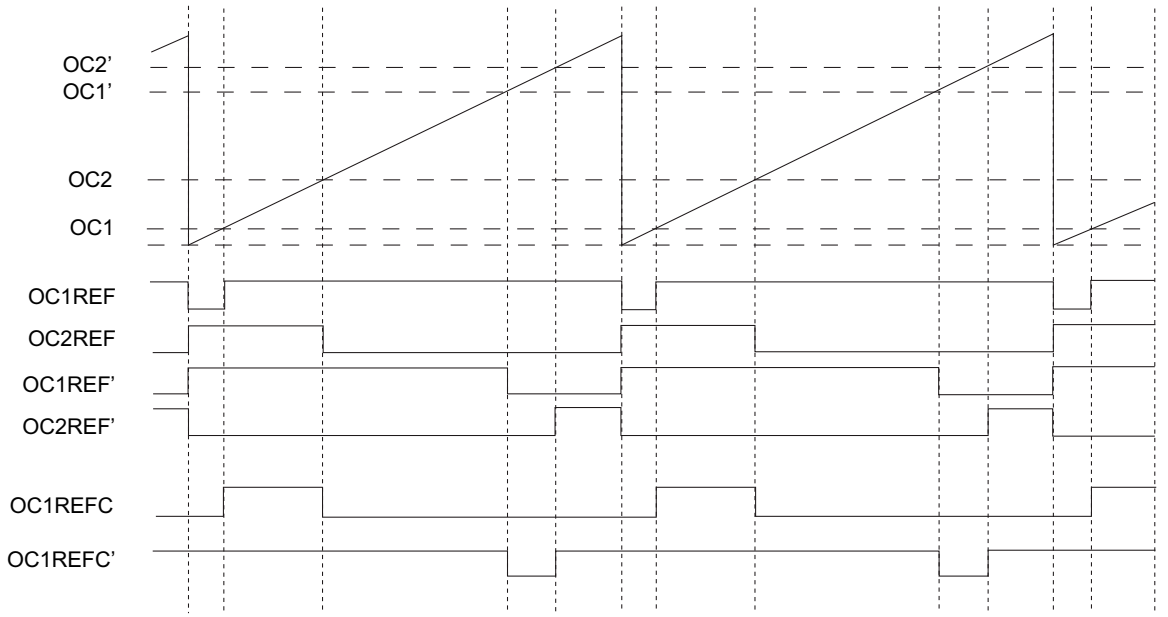
当给定通道用作组合 PWM 通道时，其互补通道必须在相反的 PWM 模式下配置（例如，一个通道在组合 PWM 模式 1 下配置，另一个通道在组合 PWM 模式 2 下配置）。

注： *出于兼容性原因，OCxM[3:0] 位域分为两部分，最高有效位与最低有效的 3 位不相邻。*

图 376 显示了不对称 PWM 模式下可以产生的信号示例，通过以下配置可获得这些信号：

- 通道 1 在组合 PWM 模式 2 下配置。
- 通道 2 在 PWM 模式 1 下配置。
- 通道 3 在组合 PWM 模式 2 下配置。
- 通道 4 在 PWM 模式 1 下配置。

图 376. 通道 1 和通道 3 上的组合 PWM 模式



OC1REFC = OC1REF 与 OC2REF 的与计算

OC1REFC' = OC1REF' 与 OC2REF' 的或计算

MS31094V1

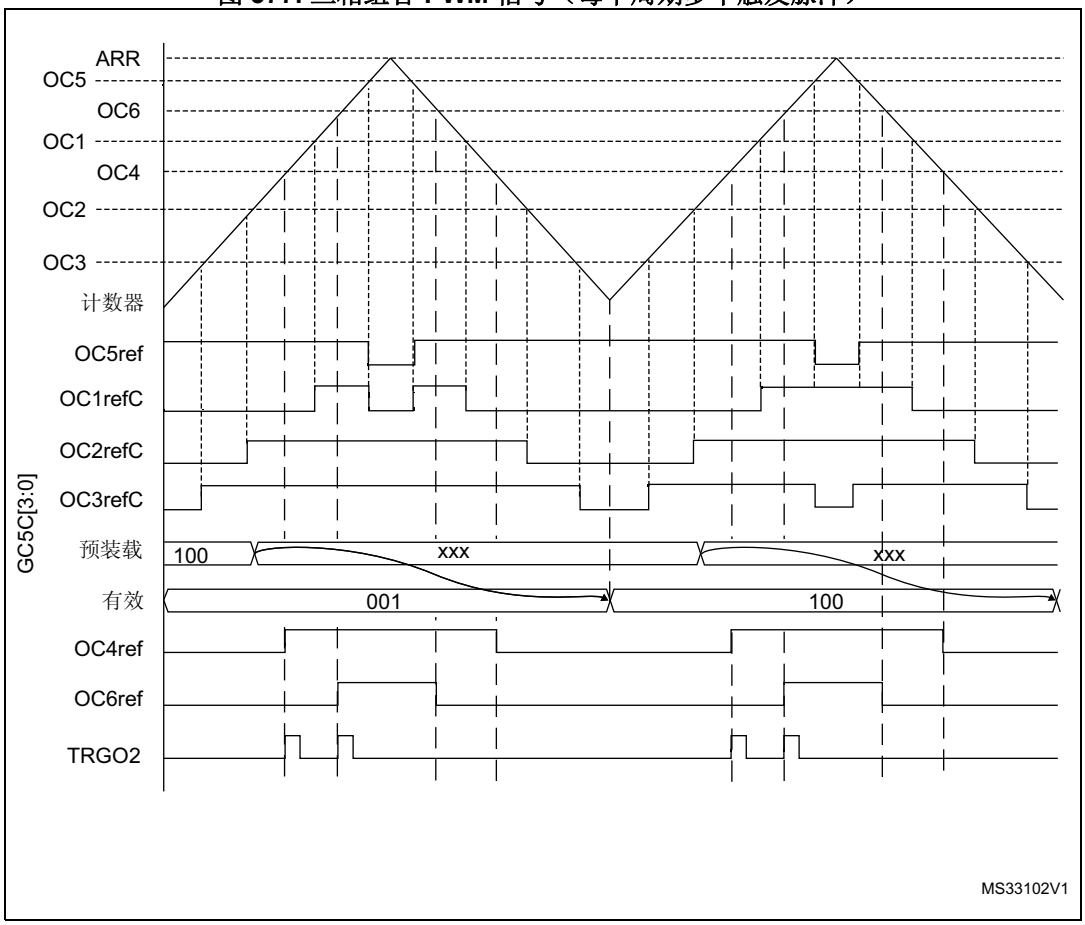
38.3.14 组合三相 PWM 模式

在组合三相 PWM 模式下，产生的一至三个中心对齐 PWM 信号与一个可编程信号间允许在脉冲中间进行逻辑与运算。OC5REF 信号用于定义产生的组合信号。凭借 TIMx_CCR5 中的 3 位 GC5C[3:1]，可以选择 OC5REF 与哪个参考信号组合。产生的信号 OCxREFC 由两个参考 PWM 的逻辑与运算组合组成。

- 如果 GC5C1 置 1，则 OC1REFC 由 TIMx_CCR1 和 TIMx_CCR5 控制
- 如果 GC5C2 置 1，则 OC2REFC 由 TIMx_CCR2 和 TIMx_CCR5 控制
- 如果 GC5C3 置 1，则 OC3REFC 由 TIMx_CCR3 和 TIMx_CCR5 控制

通道 1 到通道 3 可独立选择组合三相 PWM 模式，只需将 3 位 GC5C[3:1] 中的至少一位置 1。

图 377. 三相组合 PWM 信号（每个周期多个触发脉冲）



TRGO2 波形说明了如何根据给定的三相 PWM 信号同步 ADC。更多详细信息，请参见 *第 38.3.27 节：ADC 同步*。

38.3.15 互补输出和死区插入

高级控制定时器 (TIM1/TIM8) 可以输出两路互补信号，并管理输出的关断与接通瞬间。

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

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

互补信号 OCx 和 OCxN 通过以下多个控制位的组合进行激活：TIMx_CCER 寄存器中的 CCxE 和 CCxNE 位以及 TIMx_BDTR 和 TIMx_CR2 寄存器中的 MOE、OISx、OISxN、OSSI 和 OSSR 位。更多详细信息，请参见*第 1465 页的表 313：具有断路功能的互补通道 OCx 和 OCxN 的输出控制位*。应当注意，切换至空闲状态（MOE 下降到 0）的时刻，死区仍然有效。

CCxE 和 CCxNE 位同时置 1 并且 MOE 位置 1（如果存在断路）时，将使能死区插入。每个通道有一个 10 位死区发生器。将基于参考波形 OCxREF 生成 2 个输出 OCx 和 OCxN。如果 OCx 和 OCxN 为高电平有效：

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

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

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

图 378. 带死区插入的互补输出

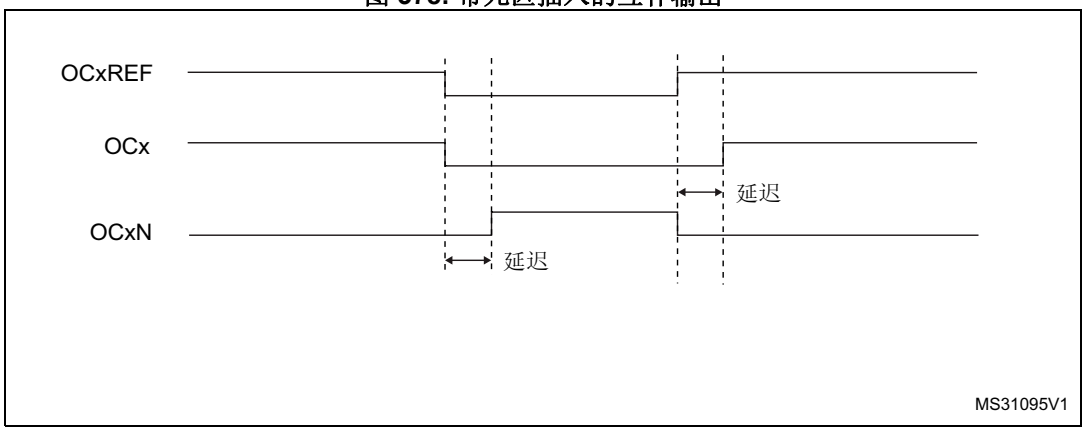


图 379. 延迟时间大于负脉冲宽度的死区波形

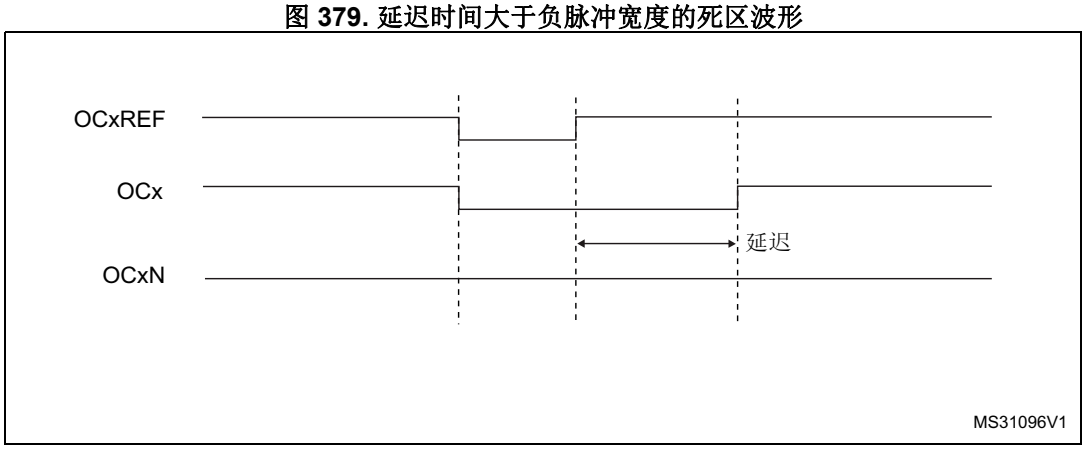
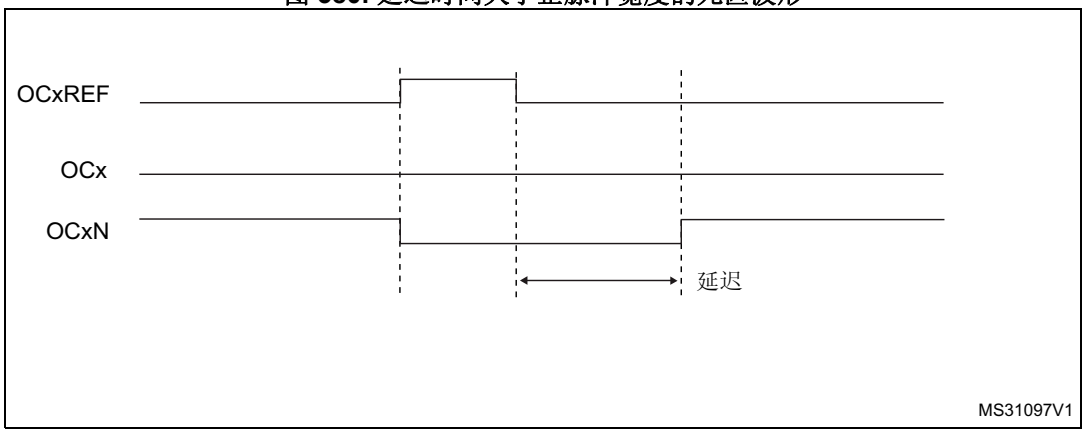


图 380. 延迟时间大于正脉冲宽度的死区波形



死区延迟对于所有通道均相同，可通过 TIMx_BDTR 寄存器中的 DTG 位进行编程。有关延迟时间计算的信息，请参见*第 38.4.18 节：TIM1/TIM8 断路和死区寄存器 (TIMx_BDTR)*。

将 OCxREF 重定向到 OCx 或 OCxN

在输出模式（强制输出模式、输出比较模式或 PWM 模式）下，通过配置 TIMx_CCER 寄存器中的 CCxE 和 CCxNE 位，可将 OCxREF 重定向到 OCx 输出或 OCxN 输出。

通过此功能，可以在一个输出上发送特定波形（如 PWM 或静态有效电平），而同时使互补输出保持其无效电平。或者，使两个输出同时保持无效电平，或者两个输出同时处于有效电平，两者互补并且带死区。

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

38.3.16 使用断路功能

断路功能的目的是保护由 TIM1 和 TIM8 定时器产生的 PWM 信号所驱动的功率开关。两个断路输入通常连接到功率级和三相逆变器的故障输出。激活时，断路电路会关闭 PWM 输出，并将其强制为预定义的安全状态。也可选择一些内部 MCU 事件来触发输出关断。

有两个断路通道。一个断路通道收集系统级故障（时钟失效和奇偶校验错误等）和应用故障（来自输入引脚和内置比较器），可以在死区持续时间后将输出强制为预定义的电平（有效或无效）。断路 2 通道只包括应用故障，能够将输出强制为无效状态。

断路期间的输出使能信号和输出电平取决于多个控制位：

- TIMx_BDTR 寄存器中的 MOE 位，允许通过软件使能/禁止输出，在发生断路和断路 2 事件时复位。
- TIMx_BDTR 寄存器中的 OSSI 位，定义定时器将输出控制在无效状态下，还是释放控制权给 GPIO 控制器（通常使其处于高阻态模式）
- TIMx_CR2 寄存器中的 OISx 和 OISxN 位，将输出设置为关断电平（有效或无效）。无论 OISx 和 OISxN 的值为何，均无法在给定时间将 OCx 和 OCxN 输出同时设置为有效电平。更多详细信息，请参见*第 1465 页的表 313：具有断路功能的互补通道 OCx 和 OCxN 的输出控制位*。

退出复位状态后，断路功能处于禁止状态，MOE 位处于低电平。将 TIMx_BDTR 寄存器中的 BKE 位和 BKE2 位置 1，可使能断路功能。可通过配置同一寄存器中的 BKP 位和 BKP2 位来选择断路输入的极性。BKEx 和 BKPx 位可同时修改。对 BKEx 和 BKPx 位执行写操作时，写操作会在 1 个 APB 时钟周期的延迟后生效。因此，执行写操作后，需要等待 1 个 APB 时钟周期，才能准确回读该位。

由于 MOE 下降沿可能是异步信号，因此在实际信号（作用于输出）与同步控制位（位于 TIMx_BDTR 寄存器中）之间插入了再同步电路， 从而在异步信号与同步信号之间产生延迟。具体而言，如果在 MOE 处于低电平时向其写入 1，则必须首先插入延迟（空指令），才能准确进行读取。这是因为写入的是异步信号，而读取的却是同步信号。

断路 (BRK) 通道的源为：

- 连接到 BKIN 引脚的外部源（由 AFIO 控制器设定），具有极性选择和可选的数字滤波
- 内部源：
  - 比较器的输出，具有极性选择和可选的数字滤波
  - DFSDM1 外设的模拟看门狗输出
  - 系统中断：
    - Cortex® -M7 LOCKUP 输出
    - PVD 输出
    - SRAM 奇偶校验错误信号
    - Flash ECC 错误
    - CSS 检测器产生时钟故障事件

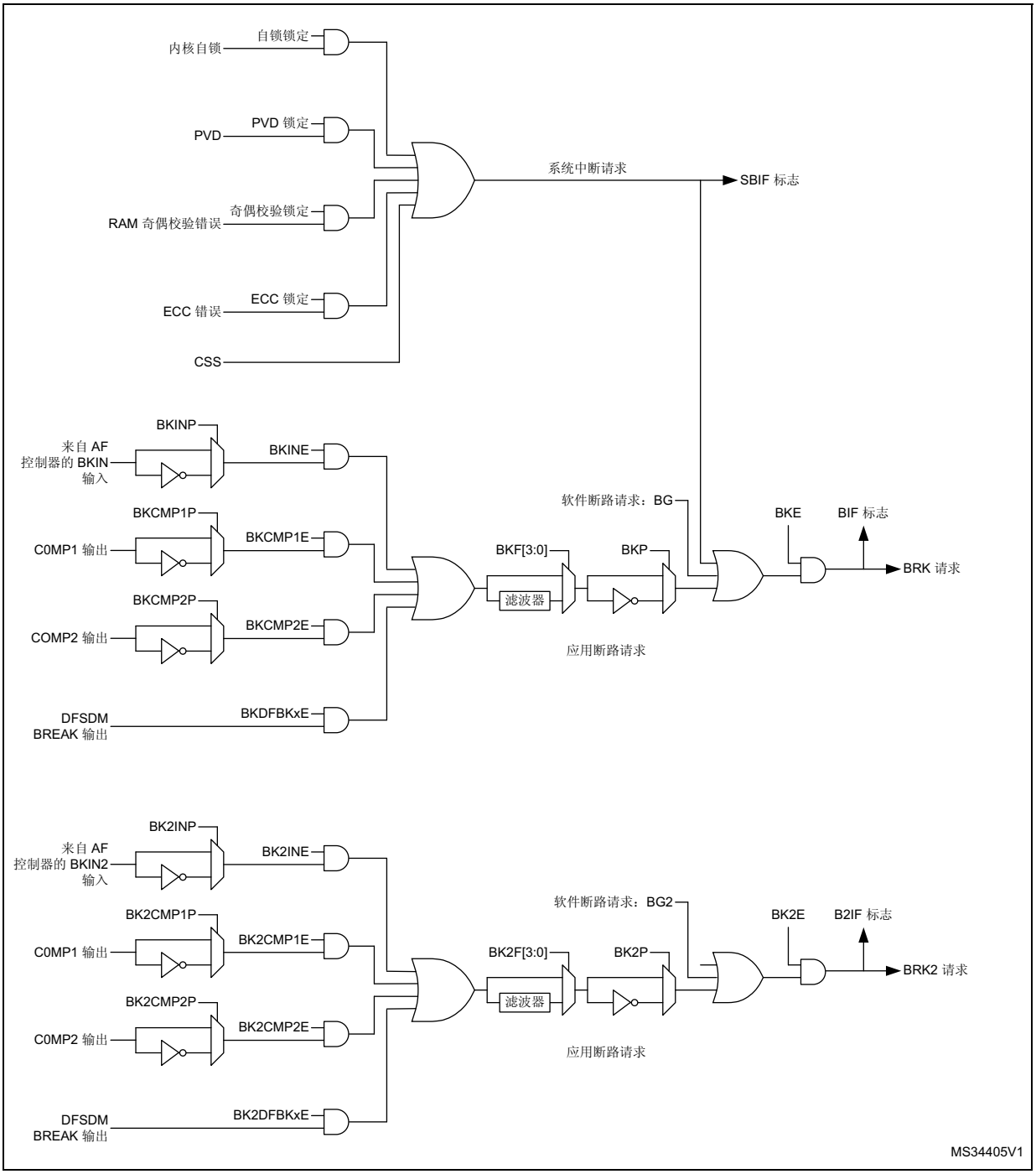
断路 2 (BRK2) 的源：

- 连接到 BKIN 引脚的外部源（由 AFIO 控制器设定），具有极性选择和可选的数字滤波。
- 来自比较器输出的内部源。

也可由软件通过 TIMx_EGR 寄存器中的 BG 和 B2G 位产生断路事件。

在所有源进入定时器 BRK 或 BRK2 输入之前，对其进行 OR 运算，如以下*图 381* 所示。

图 381. 断路和断路 2 电路概述



注： *只有禁止可编程滤波器时才能保证异步（无时钟）操作。如果使能可编程滤波器，必须使用故障安全时钟模式（例如，使用内部 PLL 和/或 CSS）来保证能够处理断路事件。*

发生断路之一（其中一个断路输入上出现所选电平）时：

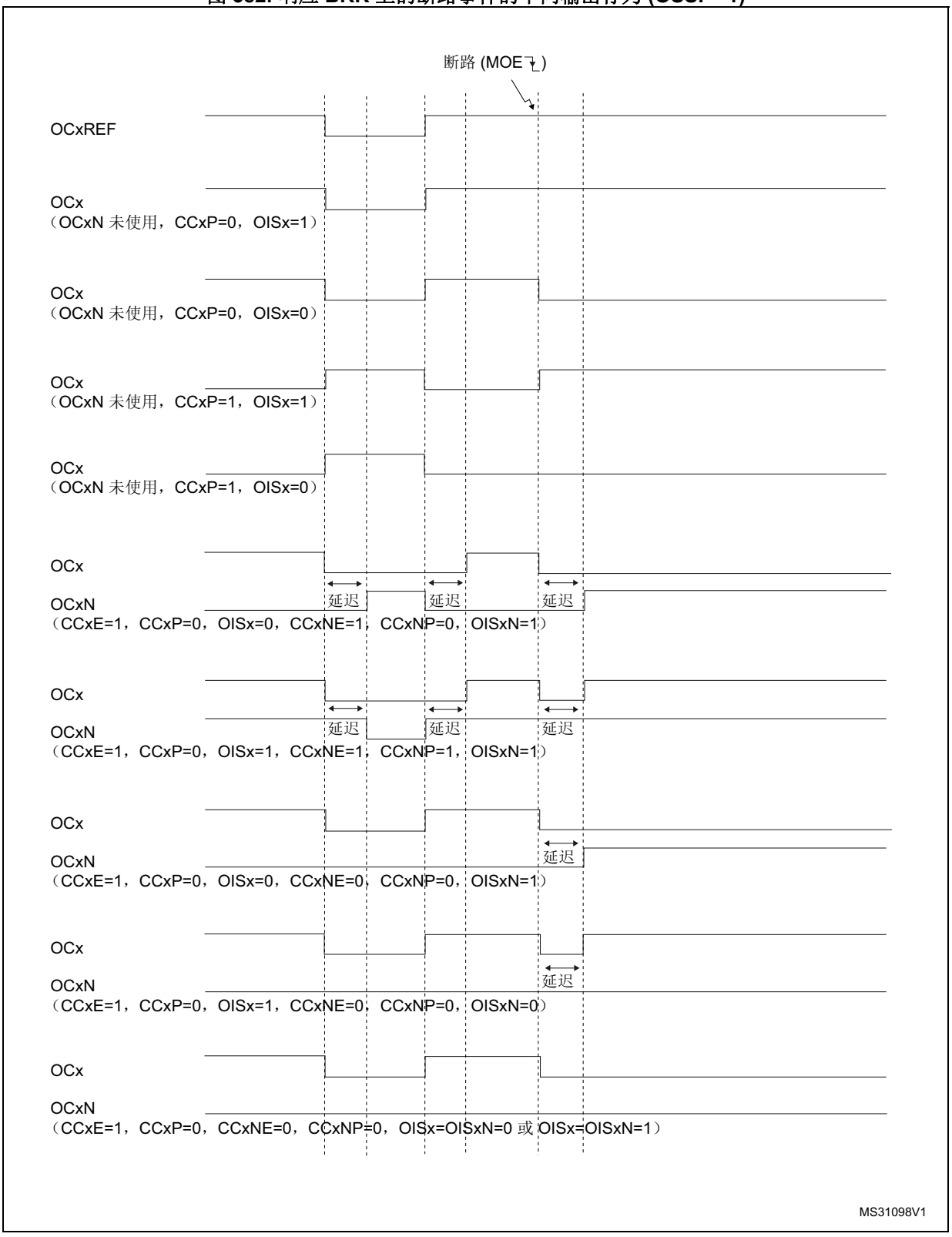
- MOE 位异步清零，使输出处于无效状态、空闲状态甚至释放控制权给 GPIO 控制器（通过 OSSI 位进行选择）。即使 MCU 振荡器关闭，该功能仍然使能。
- MOE=0 时，将以 TIMx_CR2 寄存器 OISx 位中编程的电平驱动每个输出通道。如果 OSSI=0，定时器将释放输出控制（由 GPIO 控制器接管），否则使能输出保持高电平。
- 使用互补输出时：
  - 输出首先置于无效状态（取决于极性）。这是异步操作，因此即使没有为定时器提供时钟，该操作仍有效。
  - 如果定时器时钟仍存在，则将重新激活死区发生器，进而在死区后以 OISx 和 OISxN 位中编程的电平驱动输出。即使在这种情况下，也不能同时将 OCx 和 OCxN 驱动至其有效电平。请注意，MOE 进行再同步，因此死区的持续时间会比通常情况稍长一些（约 2 个 ck_tim 时钟周期）。
  - 如果 OSSI=0，定时器将释放输出控制（由强制高阻态的 GPIO 控制器接管），否则使能输出将保持高电平或在 CCxE 或 CCxNE 位之一为高电平时立即变为高电平。
- 将断路状态标志（TIMx_SR 寄存器中的 SBIF、BIF 和 B2IF 位）置 1。如果 TIMx_DIER 寄存器中的 BIE 位置 1，则会产生中断。如果 TIMx_DIER 寄存器中的 BDE 位置 1，可发送 DMA 请求。
- 如果 TIMx_BDTR 寄存器中的 AOE 位置 1，则 MOE 位会在发生下一更新事件 (UEV) 时自动再次置 1。例如，这可用于执行调节。否则，MOE 将始终保持低电平，直到应用将其再次置 1。这种情况下，这一特性可用于确保安全。可以将断路输入连接到功率驱动器的警报、温度传感器或任何安全元件。

注： *断路输入为电平有效。因此，当断路输入有效电平时，不能将 MOE 位置 1（自动或通过软件）。同时，不能将状态标志 BIF 和 B2IF 清零。*

除断路输入和输出管理外，断路电路内部还实施了写保护，用以保护应用的安全。通过该功能，用户可冻结多个参数配置（死区持续时间、OCx/OCxN 极性和禁止时的状态、OCxM 配置、断路使能和极性）。应用可以通过 TIMx_BDTR 寄存器中的 LOCK 位，从 3 种保护级别中进行选择。请参见*第 38.4.18 节：TIM1/TIM8 断路和死区寄存器 (TIMx_BDTR)*。MCU 复位后只能对 LOCK 位执行一次写操作。

图 382 所示为输出对断路响应行为的示例。

图 382. 响应 BRK 上的断路事件的不同输出行为 (OSSI = 1)



两个断路输入针对定时器输出具有不同的行为：

- BRK 输入可禁止（无效状态）PWM 输出，也可将 PWM 输出强制为预定义的安全状态。
- BRK2 只能禁止（无效状态）PWM 输出。

BRK 输入的优先级高于 BRK2 输入，如*表 310* 所示。

注： *BRK2 必须只在 OSSR = OSSI = 1 时使用。*

表 310. 定时器输出行为与 BRK/BRK2 输入

BRK	BRK2	定时器输出状态	典型用例	
			OCxN输出（下桥臂开关）	OCx输出（上桥臂开关）
有效	X	– 无效，之后强制为输出状态（死区后） – 如果 OSSI = 0，则禁止输出（由 GPIO 逻辑接管控制）	死区插入后开启	关闭
无效	有效	无效	关闭	关闭

图 383 给出了 BRK 和 BRK2 输入上出现有效信号时 OCx 和 OCxN 输出行为示例。在这种情况下，两个输出的极性均为高电平有效（TIMx_CCER 寄存器中的 CCxP = CCxNP = 0）。

图 383. BRK 和 BRK2 引脚使能后的 PWM 输出状态 (OSSI=1)

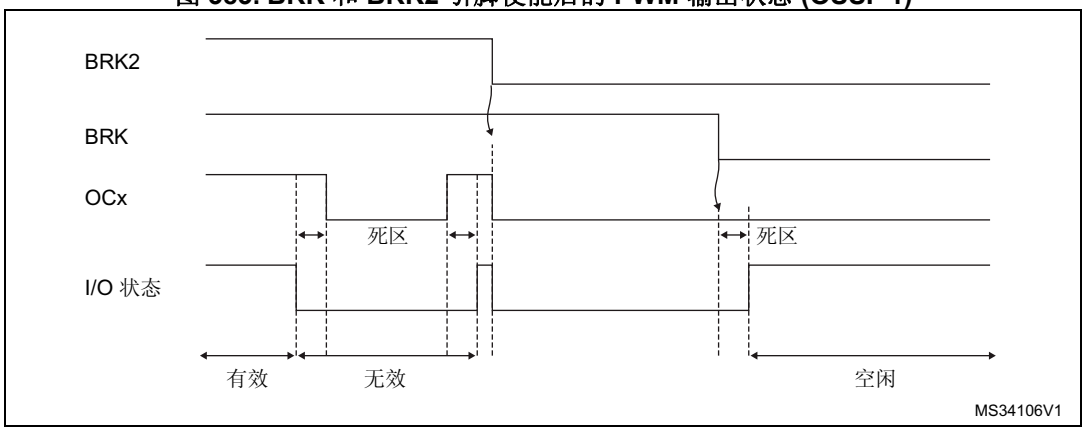
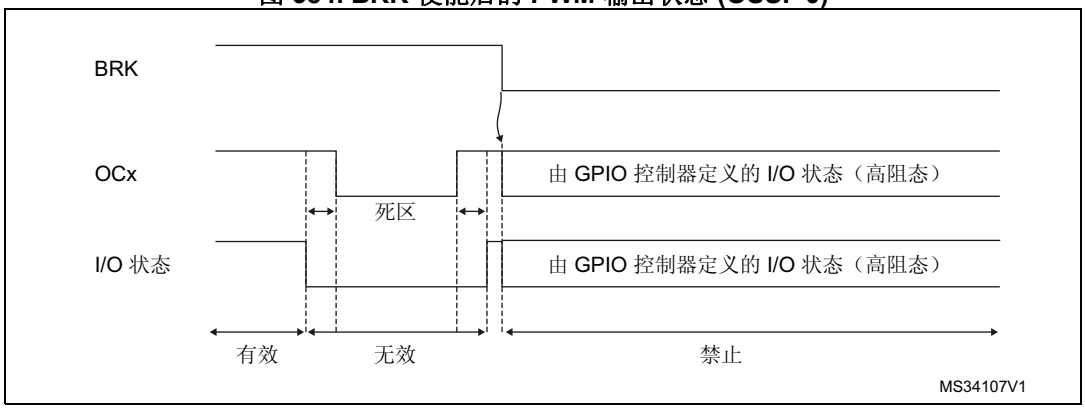


图 384. BRK 使能后的 PWM 输出状态 (OSSI=0)



38.3.17 双向断路输入

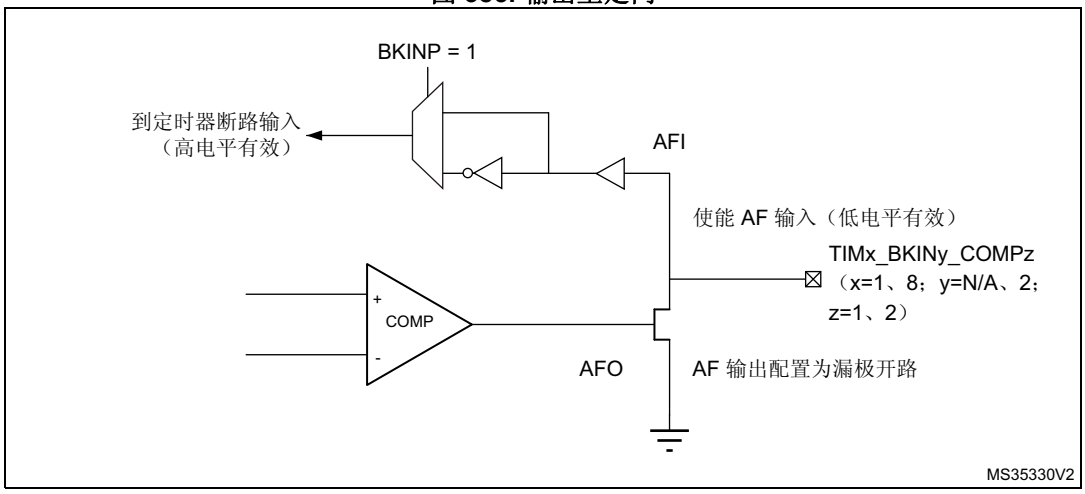
除了来自比较器的常规数字断路输入和内部断路事件，定时器 1 和定时器 8 还具有整合两个源的双向断路输入/输出，如*图 385* 所示。

TIMx_BKINy_COMPz 引脚整合 COMPz 输出（被配置为开漏模式）和 Timerx 的 TIMx_BKINy 输入。他们可以：

——使用单个引脚为外部 MCU 或栅极驱动器关断输入提供全局断路信号。

——在必须将多个内部和外部断路输入合并时，将内部比较器和多个外部开漏比较器输出“或”连接在一起，然后触发断路事件。

图 385. 输出重定向



38.3.18 发生外部事件时清除 OCxREF 信号

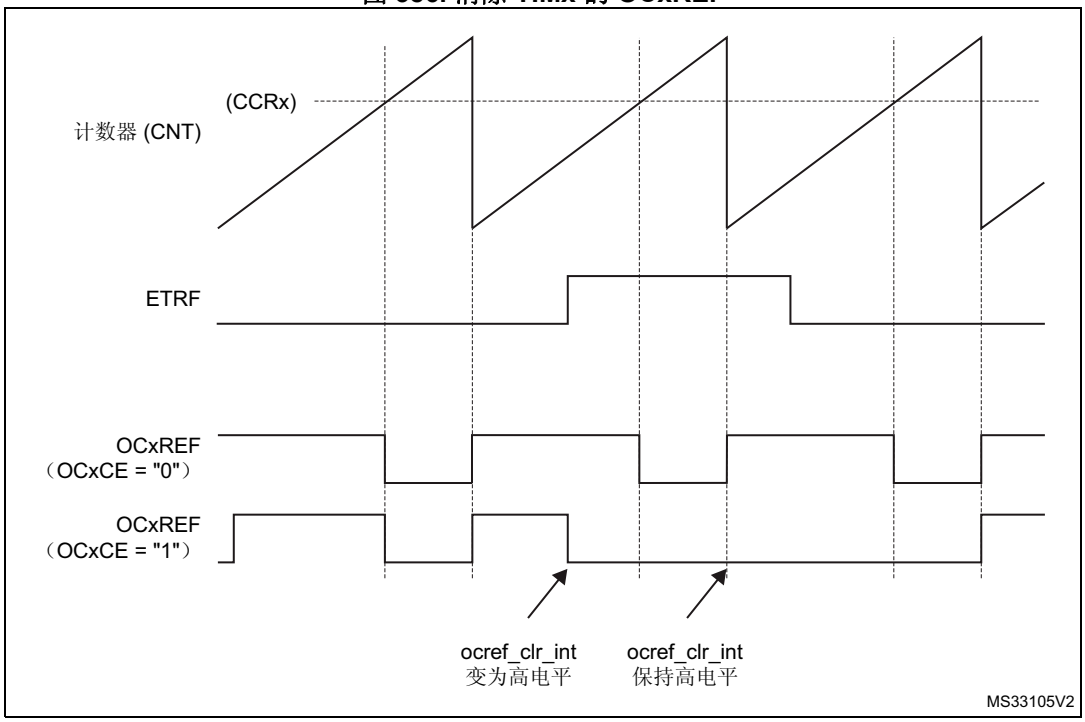
对于给定通道，在 ocref_clr_int 输入上施加高电平（相应 TIMx_CCMRx 寄存器中的 OCxCE 使能位置 1），可将 OCxREF 信号清零。OCxREF 信号将保持低电平，直到发生下一更新事件 (UEV)。该功能只能在输出比较模式和 PWM 模式下使用。在强制模式下不起作用。ocref_clr_int 连接到 ETRF 信号（滤波后的 ETR）。

选择 ETRF 时，ETR 必须配置如下：

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

图 386 对比了使能位 OCxCE 在不同值下的情况，显示了当 ETRF 输入变为高电平时 OCxREF 信号的行为。在本例中，定时器 TIMx 编程为 PWM 模式。

图 386. 清除 TIMx 的 OCxREF



注： *如果 PWM 的占空比为 100% (CCRx>ARR)，则下次计数器溢出时会再次使能 OCxREF。*

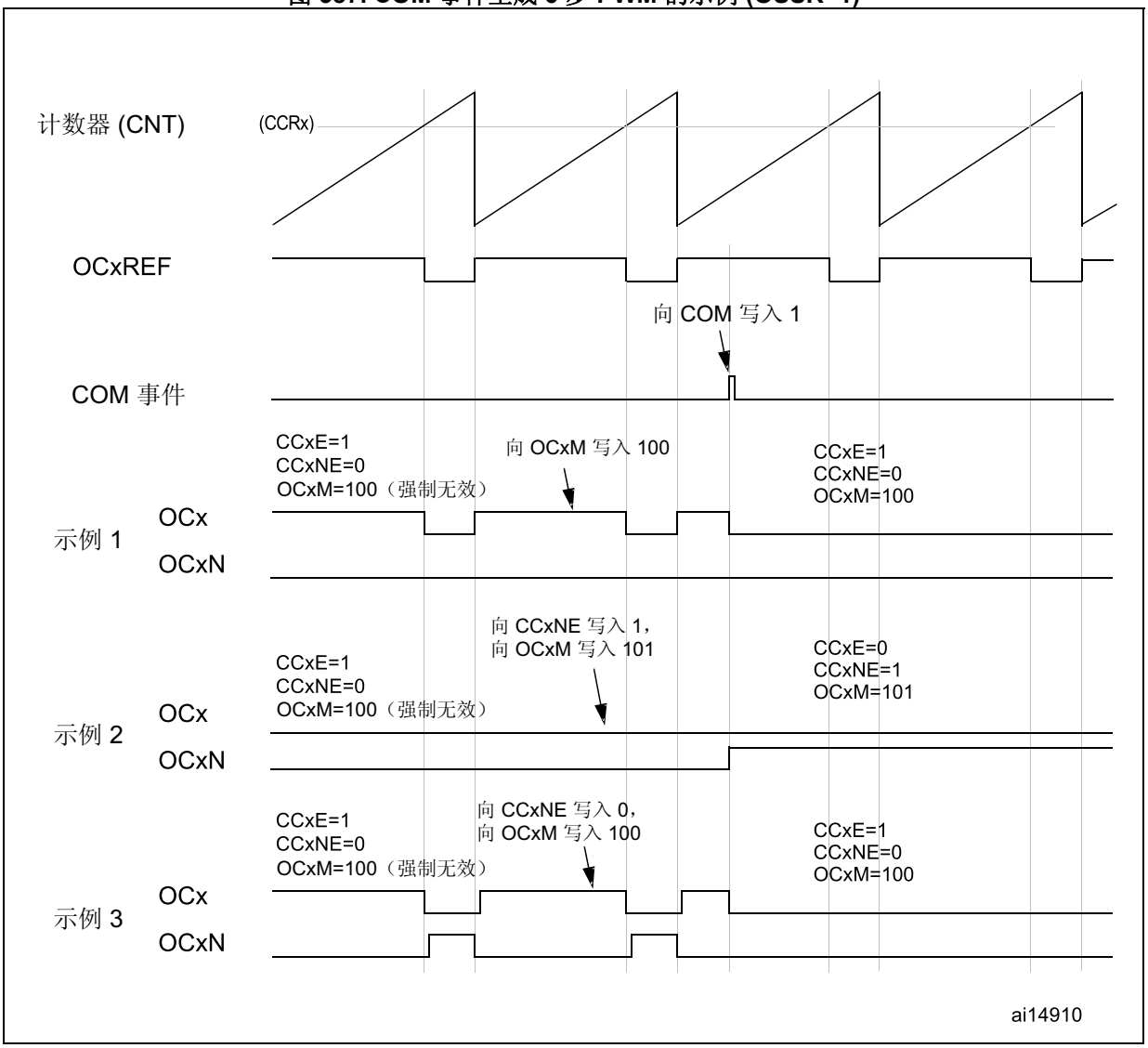
38.3.19 生成 6 步 PWM

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

发生 COM 事件时，某个标志位（TIMx_SR 寄存器中的 COMIF 位）将会置 1。这时，如果 TIMx_DIER 寄存器中的 COMIE 位置 1，将产生中断；如果 TIMx_DIER 寄存器中的 COMDE 位置 1，则将产生 DMA 请求。

图 387 以 3 种不同的编程配置为例，显示了发生 COM 事件时 OCx 和 OCxN 输出的行为。

图 387. COM 事件生成 6 步 PWM 的示例 (OSSR=1)



38.3.20 单脉冲模式

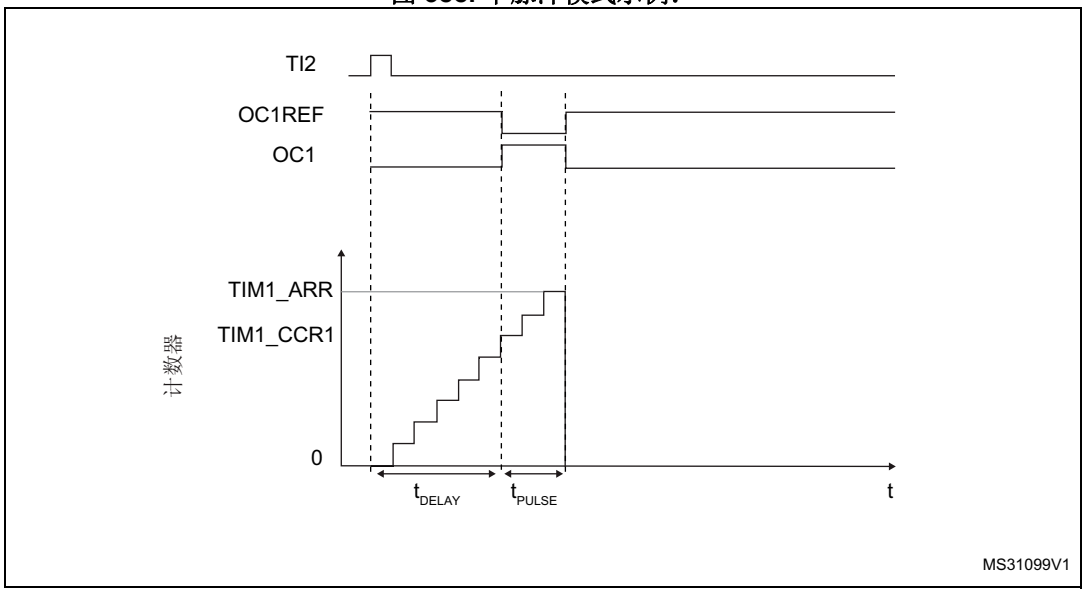
单脉冲模式 (OPM) 是上述模式的一个特例。在这种模式下，计数器可以在一个激励信号的触发下启动，并可在一段可编程的延时后产生一个脉宽可编程的脉冲。

可以通过从模式控制器启动计数器。可以在输出比较模式或 PWM 模式下生成波形。将 TIMx_CR1 寄存器中的 OPM 位置 1，即可选择单脉冲模式。这样，发生下一更新事件 UEV 时，计数器将自动停止。

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

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

图 388. 单脉冲模式示例：



例如，用户希望达到这样的效果：在 TI2 输入引脚检测到上升沿时，经过 t_{DELAY} 的延迟，在 OC1 上产生一个长度为 t_{PULSE} 的正脉冲。

使用 TI2FP2 作为触发 1：

- 在 TIMx_CCMR1 寄存器中写入 CC2S=“01”，以将 TI2FP2 映射到 TI2。
- 在 TIMx_CCER 寄存器中写入 CC2P=“0”和 CC2NP=“0”，使 TI2FP2 能够检测上升沿。
- 在 TIMx_SMCR 寄存器中写入 TS=00110，将 TI2FP2 配置为从模式控制器的触发 (TRGI)。
- 在 TIMx_SMCR 寄存器中写入 SMS=“110”（触发模式），以使用 TI2FP2 启动计数器。

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

- t_{DELAY} 由写入 TIMx_CCR1 寄存器的值定义。
- t_{PULSE} 由自动重载值与比较值之差 (TIMx_ARR - TIMx_CCR1) 来定义。
- 假设希望产生这样的波形：信号在发生比较匹配时从“0”变为“1”，在计数器达到自动重载值时由“1”变为“0”。为此，应在 TIMx_CCMR1 寄存器中写入 OC1M=111，以使能 PWM 模式 2。如果需要，可选择在 TIMx_CCMR1 寄存器的 OC1PE 和 TIMx_CR1 寄存器的 ARPE 中写入“1”，以使能预装载寄存器。这种情况下，必须在 TIMx_CCR1 寄存器中写入比较值并在 TIMx_ARR 寄存器中写入自动重载值，通过将 UG 位置 1 来产生更新，然后等待 TI2 上的外部触发事件。本例中，CC1P 的值为“0”。

在本例中，TIMx_CR1 寄存器中的 DIR 和 CMS 位应为低。

由于仅需要 1 个脉冲（单脉冲模式），因此应向 TIMx_CR1 寄存器的 OPM 位写入“1”，以便在发生下一更新事件（计数器从自动重载值返回到 0）时使计数器停止计数。TIMx_CR1 寄存器中的 OPM 位置“0”时，即选择重复模式。

特殊情况：OCx 快速使能：

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

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

38.3.21 可再触发单脉冲模式 (OPM)

该模式允许计数器可以在一个激励信号的触发下启动，并且能产生长度可编程的脉冲，但与不可再触发单脉冲模式间存在以下差别，如*第 38.3.20 节*所述：

- 发生触发时，脉冲立即产生（无可编程延时）
- 如果在上一个触发完成前发生新的触发，脉冲将延长

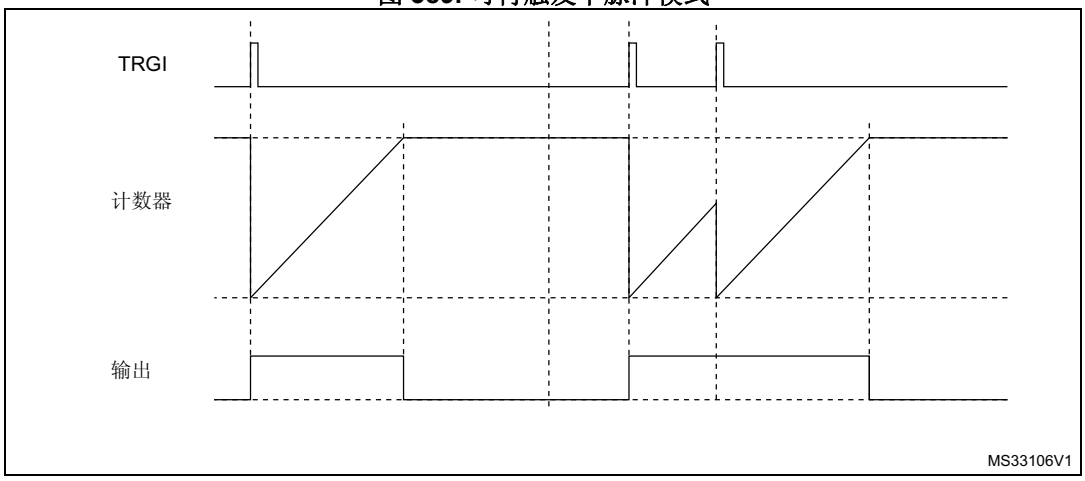
定时器必须处于从模式，TIMx_SMCR 寄存器中的位 SMS[3:0] = “1000”（组合复位 + 触发模式），针对可再触发 OPM 模式 1 或模式 2 将 OCxM[3:0] 位设置为“1000”或“1001”。

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

注： *出于兼容性原因，OCxM[3:0] 和 SMS[3:0] 位域分为两部分，最高有效位与最低有效的 3 位不相邻。*

此模式不能与中心对齐 PWM 模式一起使用。在 TIMx_CR1 中必须设置 CMS[1:0] = 00。

图 389. 可再触发单脉冲模式



38.3.22 编码器接口模式

选择编码器接口模式时，如果计数器仅在 TI2 边沿处计数，在 TIMx_SMCR 寄存器中写入 SMS=“001”；如果计数器仅在 TI1 边沿处计数，写入 SMS=“010”；如果计数器在 TI1 和 TI2 边沿处均计数，则写入 SMS=“011”。

通过编程 TIMx_CCER 寄存器的 CC1P 和 CC2P 位，选择 TI1 和 TI2 极性。如果需要，还可对输入滤波器进行编程。CC1NP 和 CC2NP 必须保持低电平。

TI1 和 TI2 两个输入用于连接正交编码器。请参见 *表 311*。如果使能计数器（在 TIMx_CR1 寄存器的 CEN 位中写入“1”），则计数器的时钟由 TI1FP1 或 TI2FP2 上的每次有效信号转换提供。TI1FP1 和 TI2FP2 是进行输入滤波器和极性选择后 TI1 和 TI2 的信号，如果不进行滤波和反相，则 TI1FP1=TI1，TI2FP2=TI2。将根据两个输入的信号转换序列，产生计数脉冲和方向信号。根据该信号转换序列，计数器相应递增或递减计数，同时硬件对 TIMx_CR1 寄存器的 DIR 位进行相应修改。任何输入（TI1 或 TI2）发生信号转换时，都会计算 DIR 位，无论计数器是仅在 TI1 或 TI2 边沿处计数，还是同时在 TI1 和 TI2 处计数。

编码器接口模式就相当于带有方向选择的外部时钟。这意味着，计数器仅在 0 到 TIMx_ARR 寄存器中的自动重载值之间进行连续计数（根据具体方向，从 0 递增计数到 ARR，或从 ARR 递减计数到 0）。因此，在启动前必须先配置 TIMx_ARR。同样，捕获、比较、重复计数器和触发输出功能继续正常工作。编码器模式和外部时钟模式 2 不兼容，因此不能同时选择。

注： *使能编码器模式时，预分频器必须设置为零。*

在此模式下，计数器会根据正交编码器的速度和方向自动进行修改，因此，其内容始终表示编码器的位置。计数方向对应于所连传感器的旋转方向。下表汇总了可能的组合（假设 TI1 和 TI2 不同时切换）。

表 311. 计数方向与编码器信号的关系

有效边沿	相反信号的电平（TI1FP1 对应 TI2，TI2FP2 对应 TI1）	TI1FP1 信号		TI2FP2 信号	
		上升	下降	上升	下降
仅在 TI1 处计数	高	递减	递增	不计数	不计数
	低	递增	递减	不计数	不计数
仅在 TI2 处计数	高	不计数	不计数	递增	递减
	低	不计数	不计数	递减	递增
在 TI1 和 TI2 处均计数	高	递减	递增	递增	递减
	低	递增	递减	递减	递增

正交编码器可直接与 MCU 相连，无需外部接口逻辑。不过，通常使用比较器将编码器的差分输出转换为数字信号。这样大幅提高了抗噪声性能。用于指示机械零位的第三个编码器输出可与外部中断输入相连，用以触发计数器复位。

图 390 以计数器工作为例，说明了计数信号的产生和方向控制。同时也说明了选择双边沿时如何对输入抖动进行补偿。将传感器靠近其中一个切换点放置时可能出现这种情况。本例中假设配置如下：

- CC1S=“01”（TIMx_CCMR1 寄存器，TI1FP1 映射到 TI1 上）。
- CC2S=“01”（TIMx_CCMR2 寄存器，TI1FP2 映射到 TI2 上）。
- CC1P=“0”，CC1NP=“0”（TIMx_CCER 寄存器，TI1FP1 未反相，TI1FP1=TI1）。
- CC2P=“0”，CC2NP=“0”（TIMx_CCER 寄存器，TI1FP2 未反相，TI1FP2= TI2）。
- SMS=“011”（TIMx_SMCR 寄存器，两个输入在上升沿和下降沿均有效）。
- CEN=“1”（TIMx_CR1 寄存器，使能计数器）。

图 390. 编码器接口模式下的计数器工作示例

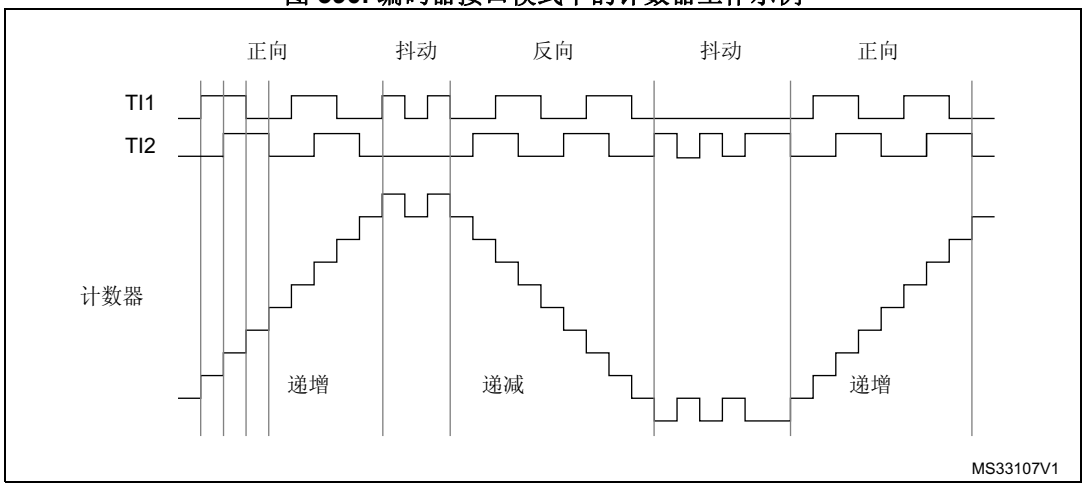
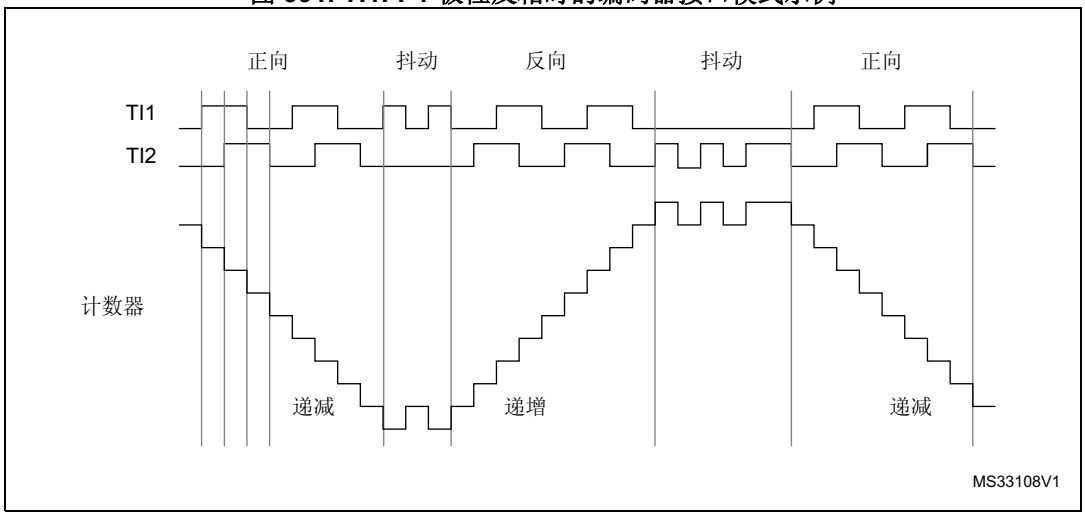


图 391 举例说明 TI1FP1 极性反相时计数器的行为（除 CC1P=“1” 外，其它配置与上例相同）。

图 391. TI1FP1 极性反相时的编码器接口模式示例



定时器配置为编码器接口模式时，会提供传感器当前位置的相关信息。使用另一个配置为捕获模式的定时器测量两个编码器事件之间的周期，可获得动态信息（速度、加速度和减速度）。指示机械零位的编码器输出即可用于此目的。根据两个事件之间的时间间隔，还可定期读取计数器。如果可能，可以将计数器值锁存到第三个输入捕获寄存器来实现此目的（捕获信号必须为周期性信号，可以由另一个定时器产生）； 还可以通过 DMA 请求读取计数器值。

TIMx_CR1 寄存器中的 IUFREMAP 位强制将更新中断标志 (UIF) 连续复制到定时计数器寄存器的位 31 (TIMxCNT[31]) 中。这样便可自动读取计数器值以及由 UIFCPY 标志发出的电位翻转条件。这可避免在后台任务（计数器读）和中断（更新中断）之间共享处理时产生竞争条件，从而简化角速度的计算。

UIF 和 UIFCPY 标志使能之间没有延迟。

在 32 位定时器实现中，当 IUFREMAP 位置 1 时，计数器的位 31 在读访问时由 UIFCPY 标志覆盖（计数器的最高有效位只能在写模式下访问）。

38.3.23 UIF 位重映射

TIMx_CR1 寄存器中的 IUFREMAP 位强制将更新中断标志 UIF 连续复制到定时计数器寄存器的位 31 (TIMxCNT[31]) 中。这样便可自动读取计数器值以及由 UIFCPY 标志发出的电位翻转条件。在特定情况下，这可避免在后台任务（计数器读）和中断（更新中断）之间共享处理时产生竞争条件，从而简化计算。

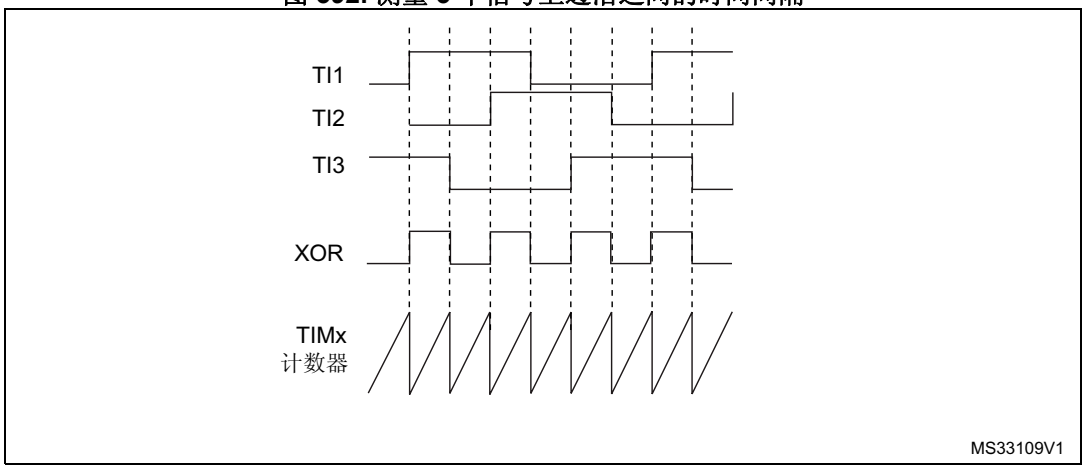
UIF 和 UIFCPY 标志使能之间没有延迟。

38.3.24 定时器输入异或功能

通过 TIMx_CR2 寄存器中的 TI1S 位，可将通道 1 的输入滤波器连接到异或门的输出，从而将 TIMx_CH1 到 TIMx_CH3 这三个输入引脚组合在一起。

异或输出可与触发或输入捕获等所有定时器输入功能配合使用。这样便于测量两个输入信号上边沿之间的间隔（如下面的*图 392* 所示）。

图 392. 测量 3 个信号上边沿之间的时间间隔



38.3.25 连接霍尔传感器

可通过用于产生电机驱动 PWM 信号的高级控制定时器（TIM1 或 TIM8）以及*图 393* 中称为“接口定时器”的另一个定时器 TIMx（TIM2、TIM3 和 TIM4），实现与霍尔传感器的连接。3 个定时器输入引脚（CC1、CC2 和 CC3）通过异或门连接到 TI1 输入通道（通过将 TIMx_CR2 寄存器中的 TI1S 位置 1 来选择），并由“接口定时器”进行捕获。

从模式控制器配置为复位模式；从输入为 TI1F_ED。这样，每当 3 个输入中有一个输入发生切换时，计数器会从 0 开始重新计数。这样将产生由霍尔输入的任何变化而触发的时基。

在“接口定时器”上，捕获/比较通道 1 配置为捕获模式，捕获信号为 TRC（请参见*第 1411 页的图 366：捕获/比较通道（例如：通道 1 输入阶段）*）。捕获值对应于输入上两次变化的间隔时间，可提供与电机转速相关的信息。

“接口定时器”可用于在输出模式下产生脉冲，以通过触发 COM 事件更改高级控制定时器（TIM1 或 TIM8）各个通道的配置。TIM1 定时器用于生成电机驱动 PWM 信号。为此，必须对接口定时器通道进行编程，以便在编程的延迟过后产生正脉冲（在输出比较或 PWM 模式中）。该脉冲通过 TRGO 输出发送到高级控制定时器（TIM1 或 TIM8）。

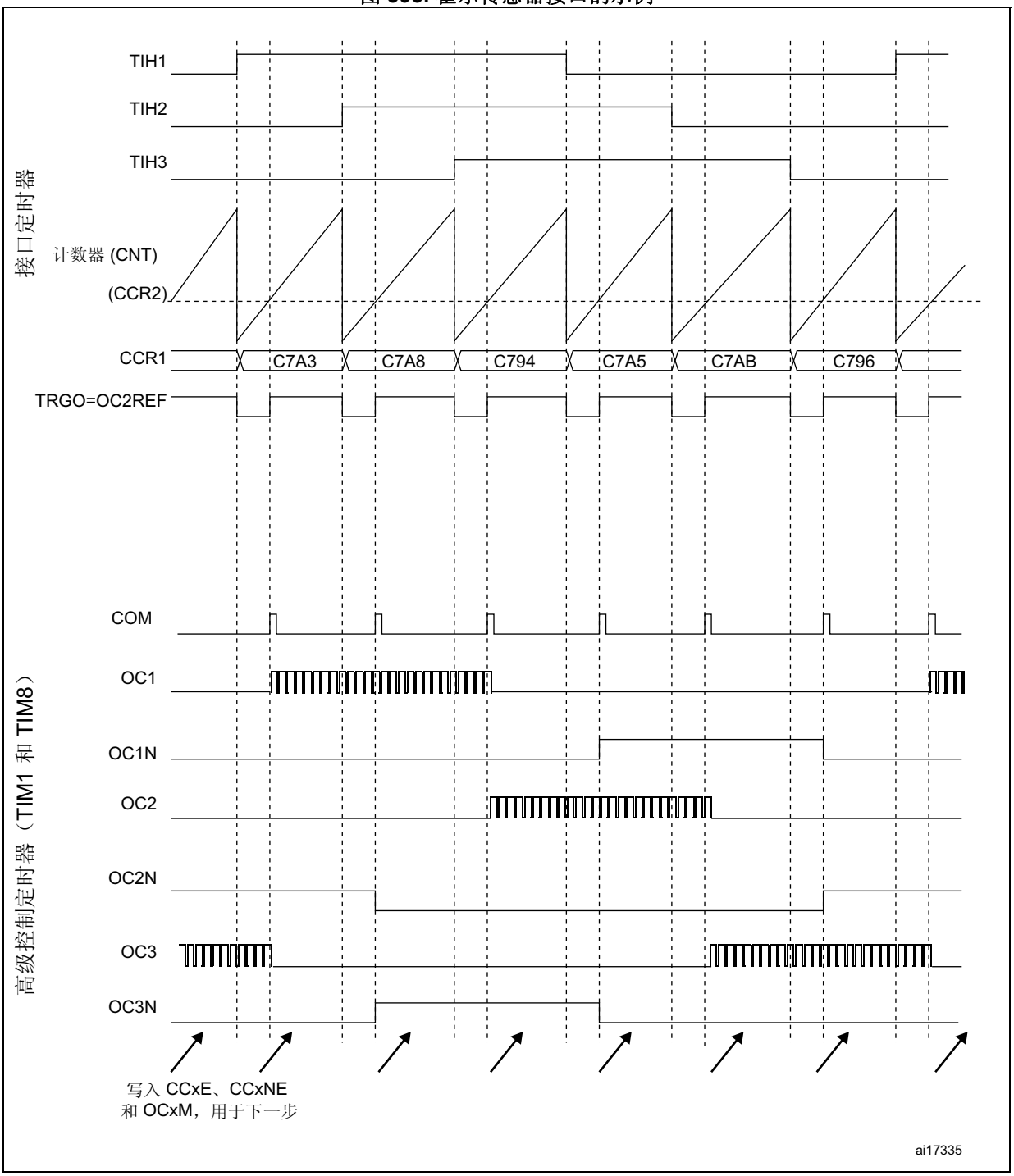
示例：霍尔输入与一个 TIMx 定时器相连接，每当霍尔输入发生更改，需要在所编程的延迟过后更改高级控制定时器 TIM1 的 PWM 配置。

- 向 TIMx_CR2 寄存器的 TI1S 位写入“1”，使 3 个定时器输入经过异或运算后进入 TI1 输入通道。
- 时基编程：向 TIMx_ARR 写入其最大值（计数器必须通过 TI1 的变化清零）。设置预分频器，以得到最大计数器周期，该周期长于传感器上两次变化的间隔时间。
- 将通道 1 编程为捕获模式（选择 TRC）：向 TIMx_CCMR1 寄存器的 CC1S 位写入“01”。如果需要，还可以编程数字滤波器。
- 将通道 2 编程为 PWM 2 模式，并具有所需延迟：向 TIMx_CCMR1 寄存器的 OC2M 位写入“111”，CC2S 位写入“00”。
- 选择 OC2REF 作为 TRGO 上的触发输出：向 TIMx_CR2 寄存器的 MMS 位写入“101”。

在高级控制定时器 TIM1 中，必须选择正确的 ITR 输入作为触发输入，定时器编程为可产生 PWM 信号，捕获/比较控制信号进行预装载（TIMx_CR2 寄存器的 CCPC=1），并且 COM 事件由触发输入控制（TIMx_CR2 寄存器中 CCUS=1）。发生 COM 事件后，在 PWM 控制位（CCxE、OCxM）中写入下一步的配置，此操作可在由 OC2REF 上升沿产生的中断子程序中完成。

图 393 为本示例的示意图。

图 393. 霍尔传感器接口的示例



38.3.26 定时器同步

TIMx 定时器从内部连接在一起，以实现定时器同步或链接。它们可在以下几种模式下实现同步：复位模式、门控模式和触发模式。

从模式：复位模式

当触发输入信号发生变化时，计数器及其预分频器可重新初始化。此外，如果 TIMx_CR1 寄存器中的 URS 位为 0，则会生成更新事件 UEV。然后，所有预装载寄存器（TIMx_ARR 和 TIMx_CCRx）都将更新。

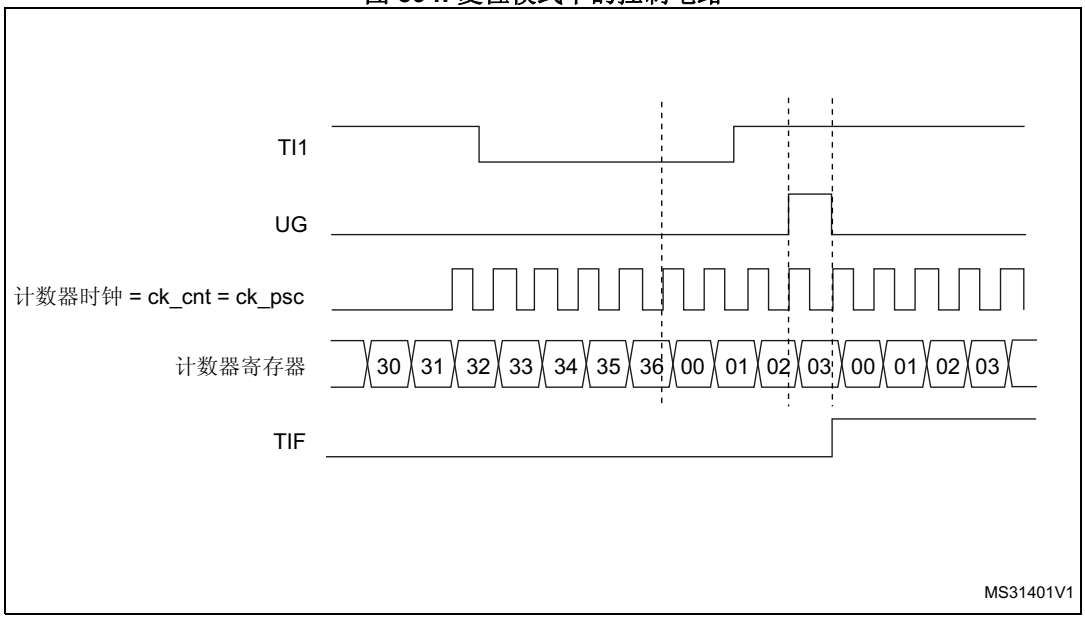
在以下示例中，TI1 输入上出现上升沿时，递增计数器清零：

- 将通道 1 配置为检测 TI1 的上升沿。配置输入滤波带宽（本例中不需要任何滤波器，因此保持 IC1F=0000）。由于捕获预分频器不用于触发操作，因此无需对其进行配置。CC1S 位只选择输入捕获源，即 TIMx_CCMR1 寄存器中的 CC1S = 01。在 TIMx_CCER 寄存器中写入 CC1P=0 和 CC1NP='0'，验证极性（仅检测上升沿）。
- 在 TIMx_SMCR 寄存器中写入 SMS=100，将定时器配置为复位模式。在 TIMx_SMCR 寄存器中写入 TS=00101，选择 TI1 作为输入源。
- 在 TIMx_CR1 寄存器中写入 CEN=1，启动计数器。

计数器开始根据内部时钟计数，然后正常运转，直到出现 TI1 上升沿。当 TI1 出现上升沿时，计数器清零，然后重新从 0 开始计数。同时，触发标志（TIMx_SR 寄存器中的 TIF 位）置 1，使能中断或 DMA 后，还可发送中断或 DMA 请求（取决于 TIMx_DIER 寄存器中的 TIE 和 TDE 位）。

下图显示了自动重载寄存器 TIMx_ARR=0x36 时的相关行为。TI1 的上升沿与实际计数器复位之间的延迟是由于 TI1 输入的重新同步电路引起的。

图 394. 复位模式下的控制电路



从模式：门控模式

输入信号的电平可用来使能计数器。

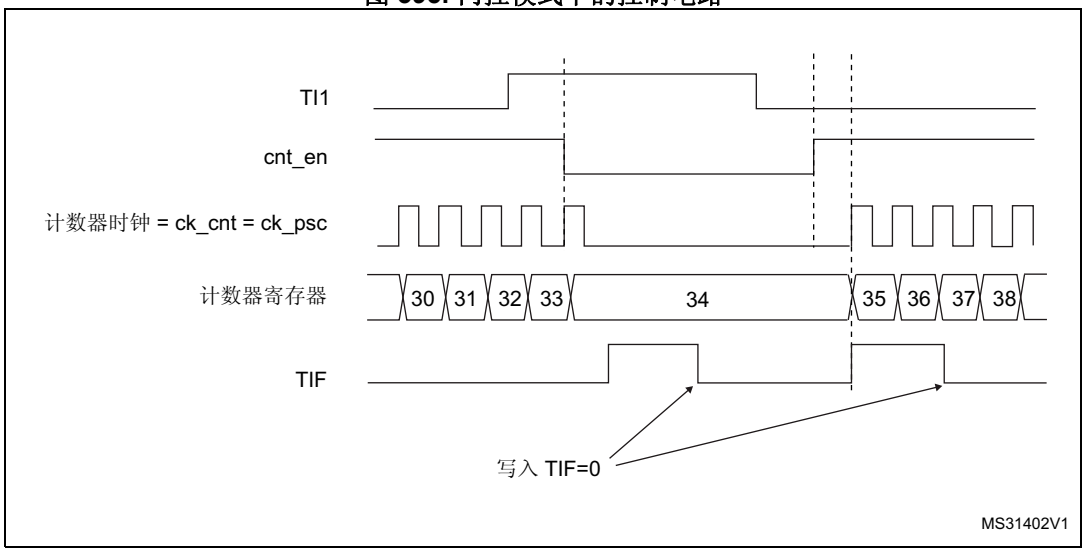
在以下示例中，递增计数器仅在 TI1 输入为低电平时计数：

- 将通道 1 配置为检测 TI1 上的低电平。配置输入滤波带宽（本例中不需要任何滤波器，因此保持 IC1F=0000）。由于捕获预分频器不用于触发操作，因此无需对其进行配置。CC1S 位只选择输入捕获源，即 TIMx_CCMR1 寄存器中的 CC1S=01。在 TIMx_CCER 寄存器中写入 CC1P=1 和 CC1NP=“0”，以确定极性（仅检测低电平）。
- 在 TIMx_SMCR 寄存器中写入 SMS=101，将定时器配置为门控模式。在 TIMx_SMCR 寄存器中写入 TS=00101，选择 TI1 作为输入源。
- 在 TIMx_CR1 寄存器中写入 CEN=1，使能计数器（在门控模式下，如果 CEN=0，则无论触发输入电平如何，计数器都不启动）。

只要 TI1 为低电平，计数器就开始根据内部时钟计数，直到 TI1 变为高电平时停止计数。计数器启动或停止时，TIMx_SR 寄存器中的 TIF 标志都会置 1。

TI1 的上升沿与实际计数器停止之间的延迟是由于 TI1 输入的重新同步电路引起的。

图 395. 门控模式下的控制电路



从模式：触发模式

所选输入上发生某一事件时可以启动计数器。

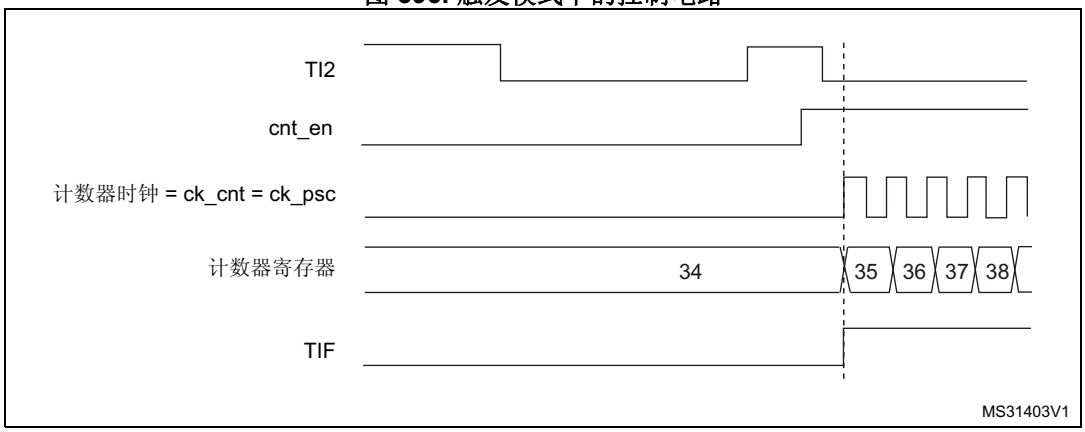
在以下示例中，TI2 输入上出现上升沿时，递增计数器启动：

- 将通道 2 配置为检测 TI2 上的上升沿。配置输入滤波带宽（本例中不需要任何滤波器，因此保持 IC2F=0000）。由于捕获预分频器不用于触发操作，因此无需对其进行配置。CC2S 位只选择输入捕获源，即 TIMx_CCMR1 寄存器中的 CC2S=01。在 TIMx_CCER 寄存器中写入 CC2P=1 和 CC2NP=0，以确定极性（仅检测低电平）。
- 在 TIMx_SMCR 寄存器中写入 SMS=110，将定时器配置为触发模式。在 TIMx_SMCR 寄存器中写入 TS=00110，选择 TI2 作为输入源。

当 TI2 出现上升沿时，计数器开始根据内部时钟计数，并且 TIF 标志置 1。

TI2 的上升沿与实际计数器启动之间的延迟是由于 TI2 输入的重新同步电路引起的。

图 396. 触发模式下的控制电路



从模式：组合复位 + 触发模式

在这种情况下，在出现所选触发输入 (TRGI) 上升沿时，重新初始化计数器，生成一个寄存器更新事件，并启动计数器。

该模式用于单脉冲模式。

从模式：外部时钟模式 2 + 触发模式

外部时钟模式 2 可与另一种从模式（外部时钟模式 1 和编码器模式除外）结合使用。这种情况下，ETR 信号用作外部时钟输入，在复位模式、门控模式或触发模式下工作时，可选择另一个输入作为触发输入。不建议通过 TIMx_SMCR 寄存器中的 TS 位来选择 ETR 作为 TRGI。

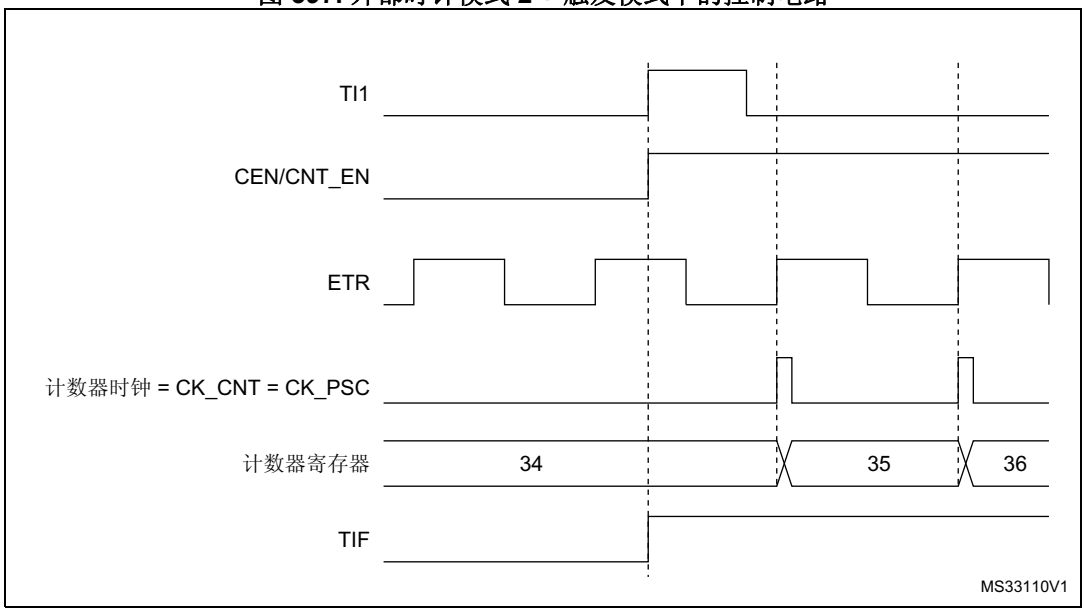
在以下示例中，只要 TI1 出现上升沿，递增计数器即会在 ETR 信号的每个上升沿处递增：

1. 通过对 TIMx_SMCR 寄存器进行如下编程，配置外部触发输入电路：
   - ETF = 0000：无滤波器。
   - ETPS = 00：禁止预分频器。
   - ETP = 0：检测 ETR 的上升沿，并写入 ECE=1，以使能外部时钟模式 2。
2. 如下配置通道 1，以检测 TI 的上升沿：
   - IC1F=0000：无滤波器。
   - 由于捕获预分频器不用于触发操作，因此无需对其进行配置。
   - TIMx_CCMR1 寄存器中 CC1S=01，只选择输入捕获源。
   - TIMx_CCER 寄存器中 CC1P=0 且 CC1NP=“0”，以确定极性（仅检测上升沿）。
3. 在 TIMx_SMCR 寄存器中写入 SMS=110，将定时器配置为触发模式。在 TIMx_SMCR 寄存器中写入 TS=00101，选择 TI1 作为输入源。

TI1 出现上升沿时将使能计数器并且 TIF 标志置 1。然后计数器在 ETR 出现上升沿时计数。

ETR 信号的上升沿与实际计数器复位之间的延迟是由于 ETRP 输入的重新同步电路引起的。

图 397. 外部时钟模式 2 + 触发模式下的控制电路



注： *必须先使能从定时器的时钟，才能从主定时器接收事件；从主定时器接收触发信号时，不得实时更改从定时器的时钟。*

38.3.27 ADC 同步

定时器可通过多种内部信号产生 ADC 触发事件，例如复位、使能或比较事件。也可生成由内部边沿检测器发出的脉冲，例如：

- OC4ref 的上升沿和下降沿
- OC5ref 上的上升沿或 OC6ref 上的下降沿

在重定向到 ADC 的 TRGO2 内部线路上发出触发信号。共有 16 个可能的事件，它们可通过 TIMx_CR2 寄存器中的 MMS2[3:0] 位选择。

第 1423 页的图 377 给出了三相电机驱动的应用示例。

注： *必须先使能从定时器的时钟，才能从主定时器接收事件；从主定时器接收触发信号时，不得实时更改从定时器的时钟。*

注： *必须先使能 ADC 时钟，才能从主定时器接收事件；从定时器接收触发信号时，不得实时更改 ADC 时钟。*

38.3.28 DMA 连续传送模式

TIMx 定时器能够根据一个事件生成多个 DMA 请求。主要目的是能够对定时器的一部分多次重新编程而无需软件开销，但也可用于定期读取一行中的多个寄存器。

DMA 控制器目标唯一，必须指向虚拟寄存器 TIMx_DMAR。发生给定的定时器事件时，定时器会启动 DMA 请求序列（突发）。每次写入 TIMx_DMAR 寄存器都会重定向到其中一个定时器寄存器。

TIMx_DCR 寄存器中的 DBL[4:0] 位设置 DMA 连续传送长度。当对 TIMx_DMAR 地址进行读或写访问时，定时器进行一次连续传送，即传送次数（按半字或字节）。

TIMx_DCR 寄存器中的 DBA[4:0] 位定义 DMA 传送的 DMA 基址（通过 TIMx_DMAR 地址执行读/写访问时）。DBA 定义为从 TIMx_CR1 寄存器地址开始计算的偏移量：

示例：

00000：TIMx_CR1

00001：TIMx_CR2

00010：TIMx_SMCR

例如，定时器 DMA 连续传送功能用于在发生更新事件后将 CCRx 寄存器（x = 2、3、4）的内容更新为通过 DMA 传输到 CCRx 寄存器中的多个半字。

具体操作步骤如下：

1. 将相应的 DMA 通道配置如下：
   - DMA 通道外设地址为 DMAR 寄存器地址。
   - DMA 通道存储器地址为包含要通过 DMA 传输到 CCRx 寄存器的数据的 RAM 缓冲区地址。
   - 要传输的数据量 = 3（参见下文注释）。
   - 禁止循环模式。
2. 通过将 DBA 和 DBL 位域配置如下来配置 DCR 寄存器：
   DBL = 3 次传输，DBA = 0xE。
3. 使能 TIMx 更新 DMA 请求（DIER 寄存器中的 UDE 位置 1）。
4. 使能 TIMx。
5. 使能 DMA 通道。

本例适用于每个 CCRx 寄存器只更新一次的情况。如果每个 CCRx 寄存器要更新两次，则要传输的数据量应为 6。下面以包含 data1、data2、data3、data4、data5 和 data6 的 RAM 缓冲区为例。数据将按照如下方式传输到 CCRx 寄存器：在第一个更新 DMA 请求期间，data1 传输到 CCR2，data2 传输到 CCR3，data3 传输到 CCR4；在第二个更新 DMA 请求期间，data4 传输到 CCR2，data5 传输到 CCR3，data6 传输到 CCR4。

注： *可以将空值写入保留的寄存器中。*

38.3.29 调试模式

当微控制器进入调试模式（带 FPU 的 Cortex®-M7 内核停止）时，TIMx 计数器会根据 DBG 模块中的 DBG_TIMx_STOP 配置位选择继续正常工作或者停止工作。

为了安全起见，当计数器停止 (DBG_TIMx_STOP = 1) 时，输出被禁止（就像 MOE 位被复位一样）。可以将输出强制变为未激活状态（OSSI 位 = 1），或者通过 GPIO 控制器（OSSI 位 = 0）来控制输出，通常将其强制为高阻态。

有关详细信息，请参见*第 60.5.8 节：微控制器调试单元 (DBGMCU)*。

为了安全起见，当计数器停止 (DBGMCU_APB2FZ1 中的 TIMx = 1) 时，输出被禁止（就像 MOE 位被复位一样）。可以将输出强制变为未激活状态（OSSI 位 = 1），或者通过 GPIO 控制器（OSSI 位 = 0）来控制输出，以将其强制为高阻态。

38.4 TIM1/TIM8 寄存器

有关寄存器说明中使用的缩写，请参见相应列表。

38.4.1 TIM1/TIM8控制寄存器 1 (TIMx_CR1)

TIM1/TIM8 control register 1

偏移地址：0x00

复位值：0x0000

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

位 15:12 保留，必须保持复位值。

位 11 **UIFREMAP**：UIF 状态位重映射 (UIF status bit remapping)
0：无重映射。UIF 状态位不复制到 TIMx_CNT 寄存器的位 31。
1：使能重映射。UIF 状态位复制到 TIMx_CNT 寄存器的位 31。

位 10 保留，必须保持复位值。

位 9:8 **CKD[1:0]**：时钟分频 (Clock division)
此位域指示定时器时钟 (CK_INT) 频率与死区发生器以及数字滤波器（ETR、TIx）所使用的死区及采样时钟 (t_{DTS}) 之间的分频比，
00：$t_{DTS}=t_{CK_INT}$
01：$t_{DTS}=2*t_{CK_INT}$
10：$t_{DTS}=4*t_{CK_INT}$
11：保留，不要设置成此值

位 7 **ARPE**：自动重载预装载使能 (Auto-reload preload enable)
0：TIMx_ARR 寄存器不进行缓冲
1：TIMx_ARR 寄存器进行缓冲

位 6:5 **CMS[1:0]**：中心对齐模式选择 (Center-aligned mode selection)
00：边沿对齐模式。计数器根据方向位 (DIR) 递增计数或递减计数。
01：中心对齐模式 1。计数器交替进行递增计数和递减计数。仅当计数器递减计数时，配置为输出的通道（TIMx_CCMRx 寄存器中的 CCxS=00）的输出比较中断标志才置 1。
10：中心对齐模式 2。计数器交替进行递增计数和递减计数。仅当计数器递增计数时，配置为输出的通道（TIMx_CCMRx 寄存器中的 CCxS=00）的输出比较中断标志才置 1。
11：中心对齐模式 3。计数器交替进行递增计数和递减计数。当计数器递增计数或递减计数时，配置为输出的通道（TIMx_CCMRx 寄存器中的 CCxS=00）的输出比较中断标志都会置 1。
注： 只要计数器处于使能状态 (CEN=1)，就不得从边沿对齐模式切换为中心对齐模式。

位 4 **DIR**：方向 (Direction)
0：计数器递增计数
1：计数器递减计数
注： 当定时器配置为中心对齐模式或编码器模式时，该位为只读状态。

位 3 **OPM**：单脉冲模式 (One pulse mode)

0：计数器在发生更新事件时不会停止计数

1：计数器在发生下一更新事件时停止计数（将 CEN 位清零）

位 2 **URS**：更新请求源 (Update request source)

此位由软件置 1 和清零，用以选择 UEV 事件源。

0：使能时，所有以下事件都会产生更新中断或 DMA 请求。此类事件包括：

– 计数器上溢/下溢

– 将 UG 位置 1

– 通过从模式控制器生成的更新事件

1：使能时，只有计数器上溢/下溢会生成更新中断或 DMA 请求。

位 1 **UDIS**：更新禁止 (Update disable)

此位由软件置 1 和清零，用以使能/禁止 UEV 事件生成。

0：使能 UEV。更新 (UEV) 事件可通过以下事件之一生成：

– 计数器上溢/下溢

– 将 UG 位置 1

– 通过从模式控制器生成的更新事件

然后更新影子寄存器的值。

1：禁止 UEV。不会生成更新事件，各影子寄存器的值（ARR、PSC 和 CCRx）保持不变。但如果将 UG 位置 1，或者从从模式控制器接收到硬件复位，则会重新初始化计数器和预分频器。

位 0 **CEN**：计数器使能 (Counter enable)

0：禁止计数器

1：使能计数器

注： *只有事先通过软件将 CEN 位置 1，才可以使用外部时钟、门控模式和编码器模式。而触发模式可通过硬件自动将 CEN 位置 1。*

38.4.2 TIM1/TIM8控制寄存器 2 (TIMx_CR2)

TIM1/TIM8 control register 2

偏移地址：0x04

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MMS2[3:0]				Res.	OIS6	Res.	OIS5
								rw	rw	rw	rw		rw		rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	OIS4	OIS3N	OIS3	OIS2N	OIS2	OIS1N	OIS1	TI1S	MMS[2:0]			CCDS	CCUS	Res.	CCPC
	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw		rw

位 31:24 保留，必须保持复位值。

位 23:20 **MMS2[3:0]**：主模式选择 2 (Master mode selection 2)

这些位可选择将发送到 ADC 以实现同步的信息 (TRGO2)。这些位的组合如下：

0000：**复位**——TIMx_EGR 寄存器中的 UG 位用作触发输出 (TRGO2)。如果复位由触发输入生成（从模式控制器配置为复位模式），则 TRGO2 上的信号相比实际复位会有延迟。

0001：**使能**——计数器使能信号 CNT_EN 用作触发输出 (TRGO2)。该触发输出可用于同时启动多个定时器，或者控制在一段时间内使能从定时器。计数器使能信号由 CEN 控制位与门控模式下的触发输入的逻辑或运算组合而成。当计数器使能信号由触发输入控制时，TRGO2 上会存在延迟，选择主/从模式时除外（请参见 TIMx_SMCR 寄存器中 MSM 位的说明）。

0010：**更新**——选择更新事件作为触发输出 (TRGO2)。例如，主定时器可用作从定时器的预分频器。

0011：**比较脉冲**——CC1IF 标志置 1 时（即使已为高），只要发生捕获或比较匹配，触发输出 (TRGO2) 都会发送一个正脉冲。

0100：**比较**——OC1REF 信号用作触发输出 (TRGO2)
0101：**比较**——OC2REF 信号用作触发输出 (TRGO2)
0110：**比较**——OC3REF 信号用作触发输出 (TRGO2)
0111：**比较**——OC4REF 信号用作触发输出 (TRGO2)
1000：**比较**——OC5REF 信号用作触发输出 (TRGO2)
1001：**比较**——OC6REF 信号用作触发输出 (TRGO2)
1010：**比较脉冲**——OC4REF 上升沿或下降沿时，TRGO2 上生成脉冲
1011：**比较脉冲**——OC6REF 上升沿或下降沿时，TRGO2 上生成脉冲
1100：**比较脉冲**——OC4REF 或 OC6REF 上升沿时，TRGO2 上生成脉冲
1101：**比较脉冲**——OC4REF 上升沿或 OC6REF 下降沿时，TRGO2 上生成脉冲
1110：**比较脉冲**——OC5REF 或 OC6REF 上升沿时，TRGO2 上生成脉冲
1111：**比较脉冲**——OC5REF 上升沿或 OC6REF 下降沿时，TRGO2 上生成脉冲

注： 必须先使能从定时器或 ADC 的时钟，才能从主定时器接收事件；并且从主定时器接收触发信号时，不得实时更改从定时器或 ADC 的时钟。

位 19 保留，必须保持复位值。

位 18 **OIS6**：输出空闲状态 6（OC6 输出）(Output Idle state 6 (OC6 output))
请参见 OIS1 位

位 17 保留，必须保持复位值。

位 16 **OIS5**：输出空闲状态 5（OC5 输出）(Output Idle state 5 (OC5 output))
请参见 OIS1 位

位 15 保留，必须保持复位值。

位 14 **OIS4**：输出空闲状态 4（OC4 输出）(Output Idle state 4 (OC4 output))
请参见 OIS1 位

位 13 **OIS3N**：输出空闲状态 3（OC3N 输出）(Output Idle state 3 (OC3N output))
请参见 OIS1N 位

位 12 **OIS3**：输出空闲状态 3（OC3 输出）(Output Idle state 3 (OC3 output))
请参见 OIS1 位

位 11 **OIS2N**：输出空闲状态 2（OC2N 输出）(Output Idle state 2 (OC2N output))
请参见 OIS1N 位

位 10 **OIS2**：输出空闲状态 2（OC2 输出）(Output Idle state 2 (OC2 output))
请参见 OIS1 位

位 9 **OIS1N**：输出空闲状态 1（OC1N 输出）(Output Idle state 1 (OC1N output))
0：当 MOE=0 时，经过死区时间后 OC1N=0
1：当 MOE=0 时，经过死区时间后 OC1N=1
注：　只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1、2 或 3，此位即无法修改。

位 8 **OIS1**：输出空闲状态 1（OC1 输出）(Output Idle state 1 (OC1 output))
0：当 MOE=0 时，（如果 OC1N 有效，则经过死区时间之后）OC1=0
1：当 MOE=0 时，（如果 OC1N 有效，则经过死区时间之后）OC1=1
注：　只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1、2 或 3，此位即无法修改。

位 7 **TI1S**：TI1 选择 (TI1 selection)
0：TIMx_CH1 引脚连接到 TI1 输入
1：TIMx_CH1、CH2 和 CH3 引脚连接到 TI1 输入（异或组合）

位 6:4 **MMS[1:0]**：主模式选择 (Master mode selection)
这些位可选择主模式下将要发送到从定时器以实现同步的信息 (TRGO)。这些位的组合如下：
000：**复位**——TIMx_EGR 寄存器中的 UG 位用作触发输出 (TRGO)。如果复位由触发输入生成（从模式控制器配置为复位模式），则 TRGO 上的信号相比实际复位会有延迟。
001：**使能**——计数器使能信号 CNT_EN 用作触发输出 (TRGO)。该触发输出可用于同时启动多个定时器，或者控制在一段时间内使能从定时器。计数器使能信号由 CEN 控制位与门控模式下的触发输入的逻辑或运算组合而成。当计数器使能信号由触发输入控制时，TRGO 上会存在延迟，选择主/从模式时除外（请参见 TIMx_SMCR 寄存器中 MSM 位的说明）。
010：**更新**——选择更新事件作为触发输出 (TRGO)。例如，主定时器可用作从定时器的预分频器。
011：**比较脉冲**——一旦发生输入捕获或比较匹配事件，当 CC1IF 标志被置 1 时（即使已为高），触发输出都会发送一个正脉冲。(TRGO)。
100：**比较**——OC1REF 信号用作触发输出 (TRGO)
101：**比较**——OC2REF 信号用作触发输出 (TRGO)
110：**比较**——OC3REF 信号用作触发输出 (TRGO)
111：**比较**——OC4REF 信号用作触发输出 (TRGO)
注：　必须先使能从定时器或 ADC 的时钟，才能从主定时器接收事件；并且从主定时器接收触发信号时，不得实时更改从定时器或 ADC 的时钟。

位 3 **CCDS**：捕获/比较 DMA 选择 (Capture/compare DMA selection)
0：发生 CCx 事件时发送 CCx DMA 请求
1：发生更新事件时发送 CCx DMA 请求

位 2 **CCUS**：捕获/比较控制更新选择 (Capture/compare control update selection)
0：如果捕获/比较控制位进行预装载 (CCPC=1)，仅通过将 COMG 位置 1 来对这些位进行更新
1：如果捕获/比较控制位进行预装载 (CCPC=1)，可通过将 COMG 位置 1 或 TRGI 的上升沿对这些位进行更新。
注：　此位仅对具有互补输出的通道有效。

位 1 保留，必须保持复位值。

位 0 **CCPC**：捕获/比较预装载控制 (Capture/compare preloaded control)
0：CCxE、CCxNE 和 OCxM 位未进行预装载
1：CCxE、CCxNE 和 OCxM 位进行了预装载，写入这些位后，仅当发生换向事件 (COM)（COMG 位置 1 或在 TRGI 上检测到上升沿，取决于 CCUS 位）时才会对这些位进行更新。
注：　此位仅对具有互补输出的通道有效。

38.4.3 TIM1/TIM8 从模式控制寄存器 (TIMx_SMCR)

TIM1/TIM8 slave mode control register

偏移地址：0x08

复位值：0x0000 0000

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

位 31:22 保留，必须保持复位值。

位 21:20 **TS[4:3]**：触发选择——位 4:3 (Trigger selection - bit 4:3)
请参见 TS[2:0] 说明——位 6:4

位 19:17 保留，必须保持复位值。

位 16 **SMS[3]**：从模式选择——位 3 (Slave mode selection - bit 3)
请参见 SMS 说明——位 2:0

位 15 **ETP**：外部触发极性 (External trigger polarity)
此位可选择将 ETR 还是 $\overline{ETR}$ 用于触发操作
0：ETR 未反相，高电平或上升沿有效。
1：ETR 反相，低电平或下降沿有效。

位 14 **ECE**：外部时钟使能 (External clock enable)
此位可使能外部时钟模式 2。
0：禁止外部时钟模式 2
1：使能外部时钟模式 2。计数器时钟由 ETRF 信号的任意有效边沿提供。

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

2：*外部时钟模式 2 可以和以下从模式同时使用：复位模式、门控模式和触发模式。不过此类情况下 TRGI 不得连接 ETRF（TS 位不得为 00111）。*

3：*如果同时使能外部时钟模式 1 和外部时钟模式 2，则外部时钟输入为 ETRF。*

位 13:12 **ETPS[1:0]**：外部触发预分频器 (External trigger prescaler)

外部触发信号 ETRP 频率不得超过 TIMxCLK 频率的 1/4。可通过使能预分频器来降低 ETRP 频率。这种方法在输入快速外部时钟时非常有用。

00：预分频器关闭
01：2 分频 ETRP 频率
10：4 分频 ETRP 频率
11：8 分频 ETRP 频率

位 11:8 **ETF[3:0]**：外部触发滤波器 (External trigger filter)

此位域可定义 ETRP 信号的采样频率和适用于 ETRP 的数字滤波器带宽。数字滤波器由事件计数器组成，每 N 个连续事件才视为一个有效输出边沿：

0000：无滤波器，按 f_{DTS} 频率进行采样
0001：$f_{SAMPLING}=f_{CK_INT}$，N=2
0010：$f_{SAMPLING}=f_{CK_INT}$，N=4
0011：$f_{SAMPLING}=f_{CK_INT}$，N=8
0100：$f_{SAMPLING}=f_{DTS}/2$，N=6
0101：$f_{SAMPLING}=f_{DTS}/2$，N=8
0110：$f_{SAMPLING}=f_{DTS}/4$，N=6
0111：$f_{SAMPLING}=f_{DTS}/4$，N=8
1000：$f_{SAMPLING}=f_{DTS}/8$，N=6
1001：$f_{SAMPLING}=f_{DTS}/8$，N=8
1010：$f_{SAMPLING}=f_{DTS}/16$，N=5
1011：$f_{SAMPLING}=f_{DTS}/16$，N=6
1100：$f_{SAMPLING}=f_{DTS}/16$，N=8
1101：$f_{SAMPLING}=f_{DTS}/32$，N=5
1110：$f_{SAMPLING}=f_{DTS}/32$，N=6
1111：$f_{SAMPLING}=f_{DTS}/32$，N=8

位 7 **MSM**：主/从模式 (Master/slave mode)

0：不执行任何操作
1：当前定时器的触发输入事件 (TRGI) 的动作被推迟，以使当前定时器与其从定时器实现完美同步（通过 TRGO）。此设置适用于由单个外部事件对多个定时器进行同步的情况。

位 6:4 **TS[2:0]**：触发选择 (Trigger selection)

此位域与 TS[4:3] 位组合在一起。
此位域可选择将要用于同步计数器的触发输入。

00000：内部触发 0 (ITR0)
00001：内部触发 1 (ITR1)
00010：内部触发 2 (ITR2)
00011：内部触发 3 (ITR3)
00100：TI1 边沿检测器 (TI1F_ED)
00101：滤波后的定时器输入 1 (TI1FP1)
00110：滤波后的定时器输入 2 (TI2FP2)
00111：外部触发输入 (ETRF)
其它值：保留

有关各定时器 ITRx 含义的详细信息，请参见*第 1453 页的表 312：TIMx 内部触发连接*。

注：　这些位只能在未使用的情况下（例如，SMS=000 时）进行更改，以避免转换时出现错误的边沿检测。

位 3 保留，必须保持复位值。

位 2:0 **SMS**：从模式选择 (Slave mode selection)

选择外部信号时，触发信号 (TRGI) 的有效边沿与外部输入上所选的极性相关（请参见输入控制寄存器和控制寄存器说明）。

0000：禁止从模式——如果 CEN = “1”，预分频器时钟直接由内部时钟提供。
0001：编码器模式 1——计数器根据 TI2FP2 电平在 TI1FP1 边沿递增/递减计数。
0010：编码器模式 2——计数器根据 TI1FP1 电平在 TI2FP2 边沿递增/递减计数。
0011：编码器模式 3——计数器在 TI1FP1 和 TI2FP2 的边沿计数，计数的方向取决于另外一个输入的电平。
0100：复位模式——在出现所选触发输入 (TRGI) 上升沿时，重新初始化计数器并生成一个寄存器更新事件。
0101：门控模式——触发输入 (TRGI) 为高电平时使能计数器时钟。只要触发输入变为低电平，计数器立即停止计数（但不复位）。计数器的启动和停止都被控制。
0110：触发模式——触发信号 TRGI 出现上升沿时启动计数器（但不复位）。只控制计数器的启动。
0111：外部时钟模式 1——由所选触发信号 (TRGI) 的上升沿提供计数器时钟。
1000：组合复位 + 触发模式——在出现所选触发输入 (TRGI) 上升沿时，重新初始化计数器，生成一个寄存器更新事件并启动计数器。
1000 以上的代码：保留。

注：　如果将 TI1F_ED 选作触发输入 (TS=00100)，则不得使用门控模式。实际上，TI1F 每次转换时，TI1F_ED 都输出 1 个脉冲，而门控模式检查的则是触发信号的电平。

注：　必须先使能从定时器的时钟，才能从主定时器接收事件；从主定时器接收触发信号时，不得实时更改从定时器的时钟。

表 312. TIMx 内部触发连接

从 TIM	ITR0 (TS = 00000)	ITR1 (TS = 00001)	ITR2 (TS = 00010)	ITR3 (TS = 00011)
TIM1	TIM15	TIM2	TIM3	TIM4
TIM8	TIM1	TIM2	TIM4	TIM5

38.4.4 TIM1/TIM8 DMA/中断使能寄存器 (TIMx_DIER)

TIM1/TIM8 DMA/interrupt enable register

偏移地址：0x0C

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	TDE	COMDE	CC4DE	CC3DE	CC2DE	CC1DE	UDE	BIE	TIE	COMIE	CC4IE	CC3IE	CC2IE	CC1IE	UIE
	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 15 保留，必须保持复位值。

位 14 **TDE**：触发 DMA 请求使能 (Trigger DMA request enable)
0：禁止触发 DMA 请求
1：使能触发 DMA 请求

位 13 **COMDE**：COM DMA 请求使能 (COM DMA request enable)
0：禁止 COM DMA 请求
1：使能 COM DMA 请求

位 12 **CC4DE**：捕获/比较 4 DMA 请求使能 (Capture/Compare 4 DMA request enable)
0：禁止 CC4 DMA 请求
1：使能 CC4 DMA 请求

位 11 **CC3DE**：捕获/比较 3 DMA 请求使能 (Capture/Compare 3 DMA request enable)
0：禁止 CC3 DMA 请求
1：使能 CC3 DMA 请求

位 10 **CC2DE**：捕获/比较 2 DMA 请求使能 (Capture/Compare 2 DMA request enable)
0：禁止 CC2 DMA 请求
1：使能 CC2 DMA 请求

位 9 **CC1DE**：捕获/比较 1 DMA 请求使能 (Capture/Compare 1 DMA request enable)
0：禁止 CC1 DMA 请求
1：使能 CC1 DMA 请求

位 8 **UDE**：更新 DMA 请求使能 (Update DMA request enable)
0：禁止更新 DMA 请求
1：使能更新 DMA 请求

位 7 **BIE**：断路中断使能 (Break interrupt enable)
0：禁止断路中断
1：使能断路中断

位 6 **TIE**：触发中断使能 (Trigger interrupt enable)
0：禁止触发中断
1：使能触发中断

位 5 **COMIE**：COM 中断使能 (COM interrupt enable)
0：禁止 COM 中断
1：使能 COM 中断

位 4 **CC4IE**：捕获/比较 4 中断使能 (Capture/Compare 4 interrupt enable)
0：禁止 CC4 中断
1：使能 CC4 中断

位 3 **CC3IE**：捕获/比较 3 中断使能 (Capture/Compare 3 interrupt enable)
0：禁止 CC3 中断
1：使能 CC3 中断

位 2 **CC2IE**：捕获/比较 2 中断使能 (Capture/Compare 2 interrupt enable)
0：禁止 CC2 中断
1：使能 CC2 中断

位 1 **CC1IE**：捕获/比较 1 中断使能 (Capture/Compare 1 interrupt enable)
0：禁止 CC1 中断
1：使能 CC1 中断

位 0 **UIE**：更新中断使能 (Update interrupt enable)
0：禁止更新中断
1：使能更新中断

38.4.5 TIM1/TIM8 状态寄存器 (TIMx_SR)

TIM1/TIM8 status register

偏移地址：0x10

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CC6IF	CC5IF
														rc_w0	rc_w0
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	SBIF	CC4OF	CC3OF	CC2OF	CC1OF	B2IF	BIF	TIF	COMIF	CC4IF	CC3IF	CC2IF	CC1IF	UIF
		rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0

位 31:18 保留，必须保持复位值。

位 17 **CC6IF**：比较 6 中断标志 (Compare 6 interrupt flag)
请参见 CC1IF 说明（注意：通道 6 只能配置为输出）

位 16 **CC5IF**：比较 5 中断标志 (Compare 5 interrupt flag)
请参见 CC1IF 说明（注意：通道 5 只能配置为输出）

位 15:14 保留，必须保持复位值。

位 13 **SBIF**：系统断路中断标志 (System Break interrupt flag)
只要系统断路输入变为有效状态，此标志便由硬件置 1。系统断路输入无效后可通过软件对其清零。
此标志必须复位以使 PWM 重新开始工作。
0：未发生断路事件。
1：在系统断路输入上检测到有效电平。如果 TIMx_DIER 寄存器中 BIE=1，则会生成中断。

位 12 **CC4OF**：捕获/比较 4 重复捕获标志 (Capture/Compare 4 overcapture flag)
请参见 CC1OF 说明

位 11 **CC3OF**：捕获/比较 3 重复捕获标志 (Capture/Compare 3 overcapture flag)
请参见 CC1OF 说明

位 10 **CC2OF**：捕获/比较 2 重复捕获标志 (Capture/compare 2 overcapture flag)
请参见 CC1OF 说明

位 9 **CC1OF**：捕获/比较 1 重复捕获标志 (Capture/Compare 1 overcapture flag)
仅当将相应通道配置为输入捕获模式时，此标志位才会由硬件置 1。通过软件写入“0”可将该位清零。
0：未检测到重复捕获。
1：TIMx_CCR1 寄存器中已捕获到计数器值且 CC1IF 标志已置 1。

位 8 **B2IF**：断路 2 中断标志 (Break 2 interrupt flag)
只要断路 2 输入变为有效状态，此标志便由硬件置 1。断路 2 输入无效后可通过软件对其清零。
0：未发生断路事件。
1：在断路 2 输入上检测到有效电平。如果 TIMx_DIER 寄存器中 BIE=1，则会生成中断。

位 7 **BIF**：断路中断标志 (Break interrupt flag)
只要断路输入变为有效状态，此标志便由硬件置 1。断路输入无效后可通过软件对其清零。
0：未发生断路事件。
1：在断路输入上检测到有效电平。如果 TIMx_DIER 寄存器中 BIE=1，则会生成中断。

位 6 **TIF**：触发中断标志 (Trigger interrupt flag)

在除门控模式以外的所有模式下，当使能从模式控制器后在 TRGI 输入上检测到有效边沿时，该标志将由硬件置 1。选择门控模式时，该标志将在计数器启动或停止时置 1。但需要通过软件清零。

0：未发生触发事件。

1：触发中断挂起。

位 5 **COMIF**：COM 中断标志 (COM interrupt flag)

此标志在发生 COM 事件时（捕获/比较控制位 CCxE、CCxNE 和 OCxM 已更新时）由硬件置 1。但需要通过软件清零。

0：未发生 COM 事件。

1：COM 中断挂起。

位 4 **CC4IF**：捕获/比较 4 中断标志 (Capture/Compare 4 interrupt flag)

请参见 CC1IF 说明

位 3 **CC3IF**：捕获/比较 3 中断标志 (Capture/Compare 3 interrupt flag)

请参见 CC1IF 说明

位 2 **CC2IF**：捕获/比较 2 中断标志 (Capture/Compare 2 interrupt flag)

请参见 CC1IF 说明

位 1 **CC1IF**：捕获/比较 1 中断标志 (Capture/Compare 1 interrupt flag)

如果通道 CC1 配置为输出：当计数器与比较值匹配时，此标志由硬件置 1，中心对齐模式下除外（请参见 TIMx_CR1 寄存器中的 CMS 位说明）。但需要通过软件清零。

0：不匹配。

1：TIMx_CNT 计数器的值与 TIMx_CCR1 寄存器的值匹配。当 TIMx_CCR1 的值大于 TIMx_ARR 的值时，CC1IF 位将在计数器发生上溢（递增计数模式和增减计数模式下）或下溢（递减计数模式下）时变为高。

如果通道 CC1 配置为输入：此位将在发生捕获事件时由硬件置 1。通过软件或读取 TIMx_CCR1 寄存器将该位清零。

0：未发生输入捕获事件

1：TIMx_CCR1 寄存器中已捕获到计数器值（IC1 上已检测到与所选极性匹配的边沿）

位 0 **UIF**：更新中断标志 (Update interrupt flag)

该位在发生更新事件时通过硬件置 1。但需要通过软件清零。

0：未发生更新。

1：更新中断挂起。该位在以下情况下更新寄存器时由硬件置 1：

– TIMx_CR1 寄存器中的 UDIS=0，并且重复计数器值上溢或下溢时（重复计数器 = 0 时更新）。

– TIMx_CR1 寄存器中的 URS = 0 且 UDIS = 0，并且由软件使用 TIMx_EGR 寄存器中的 UG 位重新初始化 CNT 时。

– TIMx_CR1 寄存器中的 URS=0 且 UDIS=0，并且 CNT 由触发事件重新初始化时（请参见 *第 38.4.3 节：TIM1/TIM8 从模式控制寄存器 (TIMx_SMCR)*）。

38.4.6 TIM1/TIM8 事件生成寄存器 (TIMx_EGR)

TIM1/TIM8 event generation register

偏移地址：0x14

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	B2G	BG	TG	COMG	CC4G	CC3G	CC2G	CC1G	UG
							w	w	w	w	w	w	w	w	w

位 15:9 保留，必须保持复位值。

位 8 **B2G**：断路 2 生成 (Break 2 generation)
此位由软件置 1 以生成事件，并由硬件自动清零。
0：不执行任何操作
1：生成断路 2 事件。MOE 位清零且 B2IF 标志置 1。使能后可发生相关中断。

位 7 **BG**：断路生成 (Break generation)
此位由软件置 1 以生成事件，并由硬件自动清零。
0：不执行任何操作
1：生成断路事件。MOE 位清零且 BIF 标志置 1。使能后可发生相关中断或 DMA 传输事件。

位 6 **TG**：触发生成 (Trigger generation)
此位由软件置 1 以生成事件，并由硬件自动清零。
0：不执行任何操作
1：TIMx_SR 寄存器中的 TIF 标志置 1。使能后可发生相关中断或 DMA 传输事件。

位 5 **COMG**：捕获/比较控制更新生成 (Capture/Compare control update generation)
该位可通过软件置 1，并由硬件自动清零
0：不执行任何操作
1：CCPC 位置 1 时，可更新 CCxE、CCxNE 和 OCxM 位
注： 此位仅对具有互补输出的通道有效。

位 4 **CC4G**：捕获/比较 4 生成 (Capture/Compare 4 generation)
请参见 CC1G 说明

位 3 **CC3G**：捕获/比较 3 生成 (Capture/Compare 3 generation)
请参见 CC1G 说明

位 2 **CC2G**：捕获/比较 2 生成 (Capture/compare 2 generation)
请参见 CC1G 说明

位 1 **CC1G**：捕获/比较 1 生成 (Capture/Compare 1 generation)
此位由软件置 1 以生成事件，并由硬件自动清零。
0：不执行任何操作
1：通道 1 上生成捕获/比较事件：
如果通道 CC1 配置为输出：
使能时，CC1IF 标志置 1 并发送相应的中断或 DMA 请求。
如果通道 CC1 配置为输入：
TIMx_CCR1 寄存器中将捕获到计数器当前值。使能时，CC1IF 标志置 1 并发送相应的中断或 DMA 请求。如果 CC1IF 标志已为高电平，CC1OF 标志将置 1。

位 0 **UG**：更新生成 (Update generation)
该位可通过软件置 1，并由硬件自动清零。
0：不执行任何操作
1：重新初始化计数器并生成寄存器更新事件。请注意，预分频器计数器也将清零（但预分频比不受影响）。如果选择中心对齐模式或 DIR=0（递增计数），计数器将清零；如果 DIR=1（递减计数），计数器将使用自动重载值 (TIMx_ARR)。

38.4.7 TIM1/TIM8 捕获/比较模式寄存器 1 (TIMx_CCMR1)

TIM1/TIM8 capture/compare mode register 1

偏移地址：0x18

复位值：0x0000 0000

这些通道可用于输入（捕获模式）或输出（比较模式）模式。通道方向通过配置相应的 CCxS 位进行定义。此寄存器的所有其它位在输入模式和输出模式下的功能均不同。对于任一给定位，OCxx 用于说明通道配置为输出时该位对应的功能，ICxx 则用于说明通道配置为输入时该位对应的功能。因此，必须注意同一个位在输入阶段和输出阶段具有不同的含义。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC2M[3]	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC1M[3]
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
							rw								rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OC2 CE	OC2M[2:0]			OC2 PE	OC2 FE	CC2S[1:0]		OC1 CE	OC1M[2:0]			OC1 PE	OC1 FE	CC1S[1:0]	
IC2F[3:0]				IC2PSC[1:0]				IC1F[3:0]				IC1PSC[1:0]			
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

输出比较模式

位 31:25 保留，必须保持复位值。

位 24 **OC2M[3]**：输出比较 2 模式——位 3 (Output Compare 2 mode - bit 3)
请参见 OC2M 说明——位 14:12

位 23:17 保留，必须保持复位值。

位 16 **OC1M[3]**：输出比较 1 模式——位 3 (Output Compare 1 mode - bit 3)
请参见 OC1M 说明——位 6:4

位 15 **OC2CE**：输出比较 2 清零使能 (Output Compare 2 clear enable)

位 14:12 **OC2M[2:0]**：输出比较 2 模式 (Output Compare 2 mode)

位 11 **OC2PE**：输出比较 2 预装载使能 (Output Compare 2 preload enable)

位 10 **OC2FE**：输出比较 2 快速使能 (Output Compare 2 fast enable)

位 9:8 **CC2S[1:0]**：捕获/比较 2 选择 (Capture/Compare 2 selection)
此位域定义通道方向（输入/输出）以及所使用的输入。
00：CC2 通道配置为输出
01：CC2 通道配置为输入，IC2 映射到 TI2 上
10：CC2 通道配置为输入，IC2 映射到 TI1 上
11：CC2 通道配置为输入，IC2 映射到 TRC 上。此模式仅在通过 TS 位（TIMx_SMCR 寄存器）选择内部触发输入时有效
注： 仅当通道关闭时（TIMx_CCER 中的 CC2E = "0"），才可向 CC2S 位写入数据。

位 7 **OC1CE**：输出比较 1 清零使能 (Output Compare 1 clear enable)
0：OC1Ref 不受 ETRF 输入影响
1：ETRF 输入上检测到高电平时，OC1Ref 立即清零

位 6:4 **OC1M**：输出比较 1 模式 (Output Compare 1 mode)

这些位定义提供 OC1 和 OC1N 的输出参考信号 OC1REF 的行为。OC1REF 为高电平有效，而 OC1 和 OC1N 的有效电平则取决于 CC1P 位和 CC1NP 位。

0000：冻结——输出比较寄存器 TIMx_CCR1 与计数器 TIMx_CNT 进行比较不会对输出造成任何影响。（该模式用于生成时基）。

0001：将通道 1 设置为匹配时输出有效电平。当计数器 TIMx_CNT 与捕获/比较寄存器 1 (TIMx_CCR1) 匹配时，OC1REF 信号强制变为高电平。

0010：将通道 1 设置为匹配时输出无效电平。当计数器 TIMx_CNT 与捕获/比较寄存器 1 (TIMx_CCR1) 匹配时，OC1REF 信号强制变为低电平。

0011：翻转——TIMx_CNT=TIMx_CCR1 时，OC1REF 发生翻转。

0100：强制变为无效电平——OC1REF 强制变为低电平。

0101：强制变为有效电平——OC1REF 强制变为高电平。

0110：PWM 模式 1——在递增计数模式下，只要 TIMx_CNT<TIMx_CCR1，通道 1 便为有效状态，否则为无效状态。在递减计数模式下，只要 TIMx_CNT>TIMx_CCR1，通道 1 便为无效状态（OC1REF=“0”），否则为有效状态（OC1REF=“1”）。

0111：PWM 模式 2——在递增计数模式下，只要 TIMx_CNT<TIMx_CCR1，通道 1 便为无效状态，否则为有效状态。在递减计数模式下，只要 TIMx_CNT>TIMx_CCR1，通道 1 便为有效状态，否则为无效状态。

1000：可再触发 OPM 模式 1——在递增计数模式下，通道为有效状态，直至（在 TRGI 信号上）检测到触发事件。然后，在 PWM 模式 1 下进行比较，通道会在下一次更新时再次变为有效状态。在递减计数模式下，通道为无效状态，直至（在 TRGI 信号上）检测到触发事件。然后，在 PWM 模式 1 下进行比较，通道会在下一次更新时再次变为无效状态。

1001：可再触发 OPM 模式 2——在递增计数模式下，通道为无效状态，直至（在 TRGI 信号上）检测到触发事件。然后，在 PWM 模式 2 下进行比较，通道会在下一次更新时再次变为无效状态。在递减计数模式下，通道为有效状态，直至（在 TRGI 信号上）检测到触发事件。然后，在 PWM 模式 2 下进行比较，通道会在下一次更新时再次变为有效状态。

1010：保留。

1011：保留。

1100：组合 PWM 模式 1——OC1REF 与在 PWM 模式 1 下的行为相同。OC1REFC 是 OC1REF 和 OC2REF 的逻辑或运算结果。

1101：组合 PWM 模式 2——OC1REF 与在 PWM 模式 2 下的行为相同。OC1REFC 是 OC1REF 和 OC2REF 的逻辑与运算结果。

1110：不对称 PWM 模式 1——OC1REF 与在 PWM 模式 1 下的行为相同。计数器递增计数时，OC1REFC 输出 OC1REF；计数器递减计数时，OC1REFC 输出 OC2REF。

1111：不对称 PWM 模式 2——OC1REF 与在 PWM 模式 2 下的行为相同。计数器递增计数时，OC1REFC 输出 OC1REF；计数器递减计数时，OC1REFC 输出 OC2REF。

注：　只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 3 且 CC1S=“00”（通道配置为输出），这些位即无法修改。

注：　在 PWM 模式下，仅当比较结果发生改变或输出比较模式由“冻结”模式切换到“PWM”模式时，OCREF 电平才会发生更改。

注：　此位域将在具有互补输出的通道上进行预装载。如果 TIMx_CR2 寄存器中的 CCPC 位置 1，则仅当生成 COM 事件时，OC1M 有效位才会从预装载位获取新值。

位 3 **OC1PE**：输出比较 1 预装载使能 (Output Compare 1 preload enable)

0：禁止与 TIMx_CCR1 相关的预装载寄存器。可随时向 TIMx_CCR1 写入数据，写入后将立即使用新值。

1：使能与 TIMx_CCR1 相关的预装载寄存器。可读/写访问预装载寄存器。TIMx_CCR1 预装载值在每次生成更新事件时都会装载到有效寄存器中。

*注：　**1：**只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 3 且 CC1S=“00”（通道配置为输出），这些位即无法修改。*

*　　**2：**只有单脉冲模式下才可在未验证预装载寄存器的情况下使用 PWM 模式（TIMx_CR1 寄存器中的 OPM 位置 1）。其它情况下则无法保证该行为。*

位 2 **OC1FE**：输出比较 1 快速使能 (Output Compare 1 fast enable)

此位用于加快触发输入事件对 CC 输出的影响。

0：即使触发开启，CC1 也将根据计数器和 CCR1 值正常工作。触发输入出现边沿时，激活 CC1 输出的最短延迟时间为 5 个时钟周期。

1：触发输入上出现有效边沿相当于 CC1 输出上的比较匹配。随后，无论比较结果如何，OC 都设置为比较电平。采样触发输入和激活 CC1 输出的延迟时间缩短为 3 个时钟周期。仅当通道配置为 PWM1 或 PWM2 模式时，OCFE 才会起作用。

位 1:0 **CC1S**：捕获/比较 1 选择 (Capture/Compare 1 Selection)

此位域定义通道方向（输入/输出）以及所使用的输入。

00：CC1 通道配置为输出

01：CC1 通道配置为输入，IC1 映射到 TI1 上

10：CC1 通道配置为输入，IC1 映射到 TI2 上

11：CC1 通道配置为输入，IC1 映射到 TRC 上。此模式仅在通过 TS 位（TIMx_SMCR 寄存器）选择内部触发输入时有效

注： 仅当通道关闭时（TIMx_CCER 中的 CC1E = “0”），才可向 CC1S 位写入数据。

输入捕获模式

位 31:16 保留，必须保持复位值。

位 15:12 **IC2F**：输入捕获 2 滤波器 (Input capture 2 filter)

位 11:10 **IC2PSC[1:0]**：输入捕获 2 预分频器 (Input capture 2 prescaler)

位 9:8 **CC2S**：捕获/比较 2 选择 (Capture/Compare 2 selection)

此位域定义通道方向（输入/输出）以及所使用的输入。

00：CC2 通道配置为输出

01：CC2 通道配置为输入，IC2 映射到 TI2 上

10：CC2 通道配置为输入，IC2 映射到 TI1 上

11：CC2 通道配置为输入，IC2 映射到 TRC 上。此模式仅在通过 TS 位（TIMx_SMCR 寄存器）选择内部触发输入时有效

注： 仅当通道关闭时（TIMx_CCER 中的 CC2E = “0”），才可向 CC2S 位写入数据。

位 7:4 **IC1F[3:0]**：输入捕获 1 滤波器 (Input capture 1 filter)

此位域可定义 TI1 输入的采样频率和适用于 TI1 的数字滤波器带宽。数字滤波器由事件计数器组成，每 N 个连续事件才视为一个有效输出边沿：

0000：无滤波器，按 f_{DTS} 频率进行采样

0001：$f_{SAMPLING}=f_{CK_INT}$，N=2

0010：$f_{SAMPLING}=f_{CK_INT}$，N=4

0011：$f_{SAMPLING}=f_{CK_INT}$，N=8

0100：$f_{SAMPLING}=f_{DTS}/2$，N=6

0101：$f_{SAMPLING}=f_{DTS}/2$，N=8

0110：$f_{SAMPLING}=f_{DTS}/4$，N=6

0111：$f_{SAMPLING}=f_{DTS}/4$，N=8

1000：$f_{SAMPLING}=f_{DTS}/8$，N=6

1001：$f_{SAMPLING}=f_{DTS}/8$，N=8

1010：$f_{SAMPLING}=f_{DTS}/16$，N=5

1011：$f_{SAMPLING}=f_{DTS}/16$，N=6

1100：$f_{SAMPLING}=f_{DTS}/16$，N=8

1101：$f_{SAMPLING}=f_{DTS}/32$，N=5

1110：$f_{SAMPLING}=f_{DTS}/32$，N=6

1111：$f_{SAMPLING}=f_{DTS}/32$，N=8

位 3:2 **IC1PSC**：输入捕获 1 预分频器 (Input capture 1 prescaler)

此位域定义 CC1 输入 (IC1) 的预分频比。只要 CC1E=“0”（TIMx_CCER 寄存器），预分频器便立即复位。

00：无预分频器，捕获输入上每检测到一个边沿便执行捕获
01：每发生 2 个事件便执行一次捕获
10：每发生 4 个事件便执行一次捕获
11：每发生 8 个事件便执行一次捕获

位 1:0 **CC1S**：捕获/比较 1 选择 (Capture/Compare 1 Selection)

此位域定义通道方向（输入/输出）以及所使用的输入。

00：CC1 通道配置为输出
01：CC1 通道配置为输入，IC1 映射到 TI1 上
10：CC1 通道配置为输入，IC1 映射到 TI2 上
11：CC1 通道配置为输入，IC1 映射到 TRC 上。此模式仅在通过 TS 位（TIMx_SMCR 寄存器）选择内部触发输入时有效

注： 仅当通道关闭时（TIMx_CCER 中的 CC1E = “0”），才可向 CC1S 位写入数据。

38.4.8 TIM1/TIM8 捕获/比较模式寄存器 2 (TIMx_CCMR2)

TIM1/TIM8 capture/compare mode register 2

偏移地址：0x1C

复位值：0x0000 0000

请参见上述 CCMR1 寄存器说明。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC4M[3]	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC3M[3]
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
							rw								rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OC4 CE	OC4M[2:0]			OC4 PE	OC4 FE	CC4S[1:0]		OC3 CE.	OC3M[2:0]			OC3 PE	OC3 FE	CC3S[1:0]	
IC4F[3:0]				IC4PSC[1:0]				IC3F[3:0]				IC3PSC[1:0]			
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

输出比较模式

位 31:25 保留，必须保持复位值。

位 24 **OC4M[3]**：输出比较 4 模式——位 3 (Output Compare 4 mode - bit 3)

位 23:17 保留，必须保持复位值。

位 16 **OC3M[3]**：输出比较 3 模式——位 3 (Output Compare 3 mode - bit 3)

位 15 **OC4CE**：输出比较 4 清零使能 (Output compare 4 clear enable)

位 14:12 **OC4M**：输出比较 4 模式 (Output compare 4 mode)

位 11 **OC4PE**：输出比较 4 预装载使能 (Output compare 4 preload enable)

位 10 **OC4FE**：输出比较 4 快速使能 (Output compare 4 fast enable)

位 9:8 **CC4S**：捕获/比较 4 选择 (Capture/Compare 4 selection)
此位域定义通道方向（输入/输出）以及所使用的输入。
00：CC4 通道配置为输出
01：CC4 通道配置为输入，IC4 映射到 TI4 上
10：CC4 通道配置为输入，IC4 映射到 TI3 上
11：CC4 通道配置为输入，IC4 映射到 TRC 上。此模式仅在通过 TS 位（TIMx_SMCR 寄存器）选择内部触发输入时有效
注： 仅当通道关闭时（TIMx_CCER 中的 CC4E = “0”），才可向 CC4S 位写入数据。

位 7 **OC3CE**：输出比较 3 清零使能 (Output compare 3 clear enable)

位 6:4 **OC3M**：输出比较 3 模式 (Output compare 3 mode)

位 3 **OC3PE**：输出比较 3 预装载使能 (Output compare 3 preload enable)

位 2 **OC3FE**：输出比较 3 快速使能 (Output compare 3 fast enable)

位 1:0 **CC3S**：捕获/比较 3 选择 (Capture/compare 3 selection)
此位域定义通道方向（输入/输出）以及所使用的输入。
00：CC3 通道配置为输出
01：CC3 通道配置为输入，IC3 映射到 TI3 上
10：CC3 通道配置为输入，IC3 映射到 TI4 上
11：CC3 通道配置为输入，IC3 映射到 TRC 上。此模式仅在通过 TS 位（TIMx_SMCR 寄存器）选择内部触发输入时有效
注： 仅当通道关闭时（TIMx_CCER 中的 CC3E = “0”），才可向 CC3S 位写入数据。

输入捕获模式

位 31:16 保留，必须保持复位值。

位 15:12 **IC4F**：输入捕获 4 滤波器 (Input capture 4 filter)

位 11:10 **IC4PSC**：输入捕获 4 预分频器 (Input capture 4 prescaler)

位 9:8 **CC4S**：捕获/比较 4 选择 (Capture/Compare 4 selection)
此位域定义通道方向（输入/输出）以及所使用的输入。
00：CC4 通道配置为输出
01：CC4 通道配置为输入，IC4 映射到 TI4 上
10：CC4 通道配置为输入，IC4 映射到 TI3 上
11：CC4 通道配置为输入，IC4 映射到 TRC 上。此模式仅在通过 TS 位（TIMx_SMCR 寄存器）选择内部触发输入时有效
注： 仅当通道关闭时（TIMx_CCER 中的 CC4E = “0”），才可向 CC4S 位写入数据。

位 7:4 **IC3F**：输入捕获 3 滤波器 (Input capture 3 filter)

位 3:2 **IC3PSC**：输入捕获 3 预分频器 (Input capture 3 prescaler)

位 1:0 **CC3S**：捕获/比较 3 选择 (Capture/compare 3 selection)
此位域定义通道方向（输入/输出）以及所使用的输入。
00：CC3 通道配置为输出
01：CC3 通道配置为输入，IC3 映射到 TI3 上
10：CC3 通道配置为输入，IC3 映射到 TI4 上
11：CC3 通道配置为输入，IC3 映射到 TRC 上。此模式仅在通过 TS 位（TIMx_SMCR 寄存器）选择内部触发输入时有效
注： 仅当通道关闭时（TIMx_CCER 中的 CC3E = “0”），才可向 CC3S 位写入数据。

38.4.9 TIM1/TIM8 捕获/比较使能寄存器 (TIMx_CCER)

TIM1/TIM8 capture/compare enable register

偏移地址：0x20

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CC6P	CC6E	Res.	Res.	CC5P	CC5E
										rw	rw			rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CC4NP	Res.	CC4P	CC4E	CC3NP	CC3NE	CC3P	CC3E	CC2NP	CC2NE	CC2P	CC2E	CC1NP	CC1NE	CC1P	CC1E
rw		rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:22 保留，必须保持复位值。

位 21 **CC6P**：捕获/比较 6 输出极性 (Capture/Compare 6 output polarity)
请参见 CC1P 说明

位 20 **CC6E**：捕获/比较 6 输出使能 (Capture/Compare 6 output enable)
请参见 CC1E 说明

位 19:18 保留，必须保持复位值。

位 17 **CC5P**：捕获/比较 5 输出极性 (Capture/Compare 5 output polarity)
请参见 CC1P 说明

位 16 **CC5E**：捕获/比较 5 输出使能 (Capture/Compare 5 output enable)
请参见 CC1E 说明

位 15 **CC4NP**：捕获/比较 4 互补输出极性 (Capture/Compare 4 complementary output polarity)
请参见 CC1NP 说明

位 14 保留，必须保持复位值。

位 13 **CC4P**：捕获/比较 4 输出极性 (Capture/Compare 4 output Polarity)
请参见 CC1P 说明

位 12 **CC4E**：捕获/比较 4 输出使能 (Capture/Compare 4 output enable)
请参见 CC1E 说明

位 11 **CC3NP**：捕获/比较 3 互补输出极性 (Capture/Compare 3 complementary output polarity)
请参见 CC1NP 说明

位 10 **CC3NE**：捕获/比较 3 互补输出使能 (Capture/Compare 3 complementary output enable)
请参见 CC1NE 说明

位 9 **CC3P**：捕获/比较 3 输出极性 (Capture/Compare 3 output polarity)
请参见 CC1P 说明

位 8 **CC3E**：捕获/比较 3 输出使能 (Capture/Compare 3 output enable)
请参见 CC1E 说明

位 7 **CC2NP**：捕获/比较 2 互补输出极性 (Capture/Compare 2 complementary output polarity)
请参见 CC1NP 说明

位 6 **CC2NE**：捕获/比较 2 互补输出使能 (Capture/Compare 2 complementary output enable)
请参见 CC1NE 说明

位 5 **CC2P**：捕获/比较 2 输出极性 (Capture/Compare 2 output polarity)
请参见 CC1P 说明

位 4 **CC2E**：捕获/比较 2 输出使能 (Capture/Compare 2 output enable)
请参见 CC1E 说明

位 3 **CC1NP**：捕获/比较 1 互补输出极性 (Capture/Compare 1 complementary output polarity)
CC1 通道配置为输出：
0：OC1N 高电平有效。
1：OC1N 低电平有效。
CC1 通道配置为输入：
此位与 CC1P 配合使用，用以定义 TI1FP1 和 TI2FP1 的极性。请参见 CC1P 说明。
注：　只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 2 或 3 且 CC1S=“00”（通道配置为输出），此位立即变为不可写状态。
注：　此位将在具有互补输出的通道上进行预装载。如果 TIMx_CR2 寄存器中的 CCPC 位置 1，则仅当生成换向事件时，CC1NP 有效位才会从预装载位获取新值。

位 2 **CC1NE**：捕获/比较 1 互补输出使能 (Capture/Compare 1 complementary output enable)
0：关闭——OC1N 未激活。OC1N 电平是 MOE、OSSI、OSSR、OIS1、OIS1N 和 CC1E 位的函数。
1：开启——在相应输出引脚上输出 OC1N 信号，具体取决于 MOE、OSSI、OSSR、OIS1、OIS1N 和 CC1E 位。
注：　此位将在具有互补输出的通道上进行预装载。如果 TIMx_CR2 寄存器中的 CCPC 位置 1，则仅当生成换向事件时，CC1NE 有效位才会从预装载位获取新值。

位 1 **CC1P**：捕获/比较 1 输出极性 (Capture/Compare 1 output polarity)
CC1 通道配置为输出：
0：OC1 高电平有效
1：OC1 低电平有效
CC1 通道配置为输入：CC1NP/CC1P 位可针对触发或捕获操作选择 TI1FP1 和 TI2FP1 的有效极性。
00：非反相/上升沿触发。电路对 TIxFP1 上升沿敏感（在复位模式、外部时钟模式或触发模式下执行捕获或触发操作），TIxFP1 未反相（在门控模式或编码器模式下执行触发操作）。
01：反相/下降沿触发。电路对 TIxFP1 下升沿敏感（在复位模式、外部时钟模式或触发模式下执行捕获或触发操作），TIxFP1 反相（在门控模式或编码器模式下执行触发操作）。
10：保留，不使用此配置。
11：非反相/上升沿和下降沿均触发。电路对 TIxFP1 上升沿和下降沿都敏感（在复位模式、外部时钟模式或触发模式下执行捕获或触发操作），TIxFP1 未反相（在门控模式下执行触发操作）。编码器模式下不得使用此配置。
注：　只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 2 或 3，此位立即变为不可写状态。
注：　此位将在具有互补输出的通道上进行预装载。如果 TIMx_CR2 寄存器中的 CCPC 位置 1，则仅当生成换向事件时，CC1P 有效位才会从预装载位获取新值。

位 0 **CC1E**：捕获/比较 1 输出使能 (Capture/Compare 1 output enable)

CC1 通道配置为输出：

0：关闭——OC1 未激活。OC1 电平是 MOE、OSSI、OSSR、OIS1、OIS1N 和 CC1NE 位的函数。

1：开启——OC1 信号输出到相应的输出引脚上，具体取决于 MOE、OSSI、OSSR、OIS1、OIS1N 和 CC1NE 位。

CC1 通道配置为输入：此位决定了是否可以实际将计数器值捕获到输入捕获/比较寄存器 1 (TIMx_CCR1) 中。

0：禁止捕获。

1：使能捕获。

注： 此位将在具有互补输出的通道上进行预装载。如果 TIMx_CR2 寄存器中的 CCPC 位置 1，则仅当生成换向事件时，CC1E 有效位才会从预装载位获取新值。

表 313. 具有断路功能的互补通道 OCx 和 OCxN 的输出控制位

控制位					输出状态(1)	
MOE 位	OSSI 位	OSSR 位	CCxE 位	CCxNE 位	OCx 输出状态	OCxN 输出状态
1	X	X	0	0	禁止输出（不由定时器驱动：高阻态） OCx=0、OCxN=0	
		0	0	1	禁止输出（不由定时器驱动：高阻态） OCx=0	OCxREF + 极性 OCxN = OCxREF 异或 CCxNP
		0	1	0	OCxREF + 极性 OCx=OCxREF 异或 CCxP	禁止输出（不由定时器驱动：高阻态） OCxN=0
		X	1	1	OCREF + 极性 + 死区	OCREF 互补项（对 OCREF 进行"非"运算）+ 极性 + 死区
		1	0	1	关闭状态（输出使能为无效状态） OCx=CCxP	OCxREF + 极性 OCxN = OCxREF 异或 CCxNP
		1	1	0	OCxREF + 极性 OCx=OCxREF 异或 CCxP	关闭状态 （输出使能为无效状态） OCxN=CCxNP
0	0	X	X	X	禁止输出（不再由定时器驱动）。输出状态由 GPIO 控制器定义，可以是高电平、低电平或高阻态。	
	1		0	0		
			0	1	关闭状态（输出使能为无效状态） 异步：OCx=CCxP、OCxN=CCxNP （如果触发 BRK 或 BRK2）。 随后（仅当触发 BRK 时才有效），如果存在时钟：在死区后 OCx=OISx 且 OCxN=OISxN，假定 OISx 和 OISxN 并没有都设置成 OCx 及 OCxN 的有效电平（否则在半桥配置下驱动开关时可能导致短路）。 **注意**：BRK2 只能在 OSSI = OSSR = 1 时使用。	
			1	0		
			1	1		

1. 如果一个通道的两个输出均未使用（由 GPIO 接管控制控制），则 OISx、OISxN、CCxP 和 CCxNP 位必须保持清零状态。

注： 与互补通道 OCx 和 OCxN 相连的外部 I/O 引脚的状态取决于通道 OCx 和 OCxN 的状态以及 GPIO 寄存器。

38.4.10 TIM1/TIM8计数器 (TIMx_CNT)

TIM1/TIM8 counter

偏移地址：0x24

复位值：0x0000 0000

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

位 31 **UIFCPY**：UIF 副本 (UIF copy)

该位是 TIMx_ISR 寄存器中 UIF 位的只读副本。如果 TIMxCR1 中的 UIFREMAP 位复位，则位 31 保留，读为 0。

位 30:16 保留，必须保持复位值。

位 15:0 **CNT[15:0]**：计数器值 (Counter value)

38.4.11 TIM1/TIM8 预分频器 (TIMx_PSC)

TIM1/TIM8 prescaler

偏移地址：0x28

复位值：0x0000

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

位 15:0 **PSC[15:0]**：预分频器值 (Prescaler value)

计数器时钟频率 (CK_CNT) 等于 f_{CK_PSC} / (PSC[15:0] + 1)。

PSC 包含每次发生更新事件（包括计数器通过 TIMx_EGR 寄存器中的 UG 位清零时，或在配置为“复位模式”时通过触发控制器清零时）时要装载到有效预分频器寄存器的值。

38.4.12 TIM1/TIM8自动重载寄存器 (TIMx_ARR)

TIM1/TIM8 auto-reload register

偏移地址：0x2C

复位值：0xFFFF

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

位 15:0 **ARR[15:0]**：自动重载值 (Auto-reload value)

ARR 为要装载到实际自动重载寄存器的值。

有关 ARR 更新和行为的更多详细信息，请参见*第 1391 页的第 38.3.1 节：时基单元*。

当自动重载值为空时，计数器不工作。

38.4.13 TIM1/TIM8 重复计数器寄存器 (TIMx_RCR)

TIM1/TIM8 repetition counter register

偏移地址：0x30

复位值：0x0000

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

位 15:0 **REP[15:0]**：重复计数器值 (Repetition Counter value)

使能预装载寄存器时，用户可通过这些位设置比较寄存器的更新频率（即，从预装载寄存器向有效寄存器周期性传输数据）；使能更新中断时，也可设置更新中断的生成速率。

与 REP_CNT 相关的减计数器每次计数到 0 时，都将生成一个更新事件并且计数器从 REP 值重新开始计数。由于只有生成重复更新事件 U_RC 时，REP_CNT 才会重载 REP 值，因此在生成下一重复更新事件之前，无论向 TIMx_RCR 寄存器写入何值都无影响。

这意味着 PWM 模式下 (REP+1) 相当于：

边沿对齐模式下的 PWM 周期数。

中心对齐模式下的 PWM 半周期数。

38.4.14 TIM1/TIM8 捕获/比较寄存器 1 (TIMx_CCR1)

TIM1/TIM8 capture/compare register 1

偏移地址：0x34

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR1[15:0]															
rw/r	rw/r	rw/r	rw/r	rw/r	rw/r	rw/r	rw/r	rw/r	rw/r	rw/r	rw/r	rw/r	rw/r	rw/r	rw/r

位 15:0 **CCR1[15:0]**：捕获/比较 1 值 (Capture/Compare 1 value)

如果通道 CC1 配置为输出：CCR1 为要装载到有效捕获/比较 1 寄存器的值（预装载值）。

如果没有通过 TIMx_CCMR1 寄存器中的 OC1PE 位来使能预装载功能，则该值立刻生效；否则只在发生更新事件时生效（拷贝到有效的捕获/比较寄存器 1）。

有效捕获/比较寄存器中包含要与计数器 TIMx_CNT 进行比较并在 OC1 输出上发出信号的值。

如果通道 CC1 配置为输入：CR1 为上一个输入捕获 1 事件 (IC1) 发生时的计数器值。只能读取 TIMx_CCR1 寄存器，无法对其进行编程。

38.4.15 TIM1/TIM8 捕获/比较寄存器 2 (TIMx_CCR2)

TIM1/TIM8 capture/compare register 2

偏移地址：0x38

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR2[15:0]															
rw/r	rw/r	rw/r	rw/r	rw/r	rw/r	rw/r	rw/r	rw/r	rw/r	rw/r	rw/r	rw/r	rw/r	rw/r	rw/r

位 15:0 **CCR2[15:0]**：捕获/比较 2 值 (Capture/Compare 2 value)

如果通道 CC2 配置为输出：CCR2 为要装载到有效捕获/比较 2 寄存器的值（预装载值）。
如果没有通过 TIMx_CCMR1 寄存器中的 OC2PE 位来使能预装载功能，则该值立刻生效；否则只在发生更新事件时生效（拷贝到有效的捕获/比较寄存器 2）。
有效捕获/比较寄存器中包含要与计数器 TIMx_CNT 进行比较并在 OC2 输出上发出信号的值。

如果通道 CC2 配置为输入：CCR2 为上一个输入捕获 2 事件 (IC2) 发生时的计数器值。只能读取 TIMx_CCR2 寄存器，无法对其进行编程。

38.4.16 TIM1/TIM8 捕获/比较寄存器 3 (TIMx_CCR3)

TIM1/TIM8 capture/compare register 3

偏移地址：0x3C

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR3[15:0]															
rw/r	rw/r	rw/r	rw/r	rw/r	rw/r	rw/r	rw/r	rw/r	rw/r	rw/r	rw/r	rw/r	rw/r	rw/r	rw/r

位 15:0 **CCR3[15:0]**：捕获/比较值 (Capture/Compare value)

如果通道 CC3 配置为输出：CCR3 为要装载到有效捕获/比较 3 寄存器的值（预装载值）。
如果没有通过 TIMx_CCMR2 寄存器中的 OC3PE 位来使能预装载功能，则该值立刻生效；否则只在发生更新事件时生效（拷贝到有效的捕获/比较寄存器 3）。
有效捕获/比较寄存器中包含要与计数器 TIMx_CNT 进行比较并在 OC3 输出上发出信号的值。

如果通道 CC3 配置为输入：CCR3 为上一个输入捕获 3 事件 (IC3) 发生时的计数器值。只能读取 TIMx_CCR3 寄存器，无法对其进行编程。

38.4.17 TIM1/TIM8 捕获/比较寄存器 4 (TIMx_CCR4)

TIM1/TIM8 capture/compare register 4

偏移地址：0x40

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR4[15:0]															
rw/r	rw/r	rw/r	rw/r	rw/r	rw/r	rw/r	rw/r	rw/r	rw/r	rw/r	rw/r	rw/r	rw/r	rw/r	rw/r

位 15:0 **CCR4[15:0]**：捕获/比较值 (Capture/Compare value)

如果通道 CC4 配置为输出：CCR4 为要装载到有效捕获/比较 4 寄存器的值（预装载值）。如果没有通过 TIMx_CCMR2 寄存器中的 OC4PE 位来使能预装载功能，则该值立刻生效；否则只在发生更新事件时生效（拷贝到有效的捕获/比较寄存器 4）。

有效捕获/比较寄存器中包含要与计数器 TIMx_CNT 进行比较并在 OC4 输出上发出信号的值。

如果通道 CC4 配置为输入：CCR4 为上一个输入捕获 4 事件 (IC4) 发生时的计数器值。只能读取 TIMx_CCR4 寄存器，无法对其进行编程。

38.4.18 TIM1/TIM8 断路和死区寄存器 (TIMx_BDTR)

TIM1/TIM8 break and dead-time register

偏移地址：0x44

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	BK2P	BK2E	BK2F[3:0]				BKF[3:0]			
						rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MOE	AOE	BKP	BKE	OSSR	OSSI	LOCK[1:0]		DTG[7:0]							
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

注： *由于可以根据 LOCK 配置锁定位 BK2P、BK2E、BK2F[3:0]、BKF[3:0]、AOE、BKP、BKE、OSSI、OSSR 和 DTG[7:0] 的写操作，因此必须在第一次对 TIMx_BDTR 寄存器执行写访问时对这些位进行配置。*

位 31:26 保留，必须保持复位值。

位 25 **BK2P**：断路 2 极性 (Break 2 polarity)

0：断路输入 BRK2 为低电平有效

1：断路输入 BRK2 为高电平有效

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

注： *对该位执行任何写操作后，都需要经过 1 个 APB 时钟周期的延迟才生效。*

位 24 **BK2E**：断路 2 使能 (Break 2 enable)

该位可使能完整的断路 2 保护（包括连接到 bk_acth 的所有源和相应的 BKIN 源，如*图 381：断路和断路 2 电路概述*所示）。

0：禁止断路 2 功能

1：使能断路 2 功能

注： BRKIN2 必须只在 OSSR = OSSI = 1 时使用。

注： 编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1 后，此位即无法修改。

注： 对该位执行任何写操作后，都需要经过 1 个 APB 时钟周期的延迟才生效。

位 23:20 **BK2F[3:0]**：断路 2 滤波器 (Break 2 filter)

此位域可定义 BRK2 输入的采样频率和适用于 BRK2 的数字滤波器带宽。数字滤波器由事件计数器组成，每 N 个连续事件才视为一个有效输出边沿：

0000：无滤波器，BRK2 异步工作

0001：$f_{SAMPLING}=f_{CK_INT}$，N=2

0010：$f_{SAMPLING}=f_{CK_INT}$，N=4

0011：$f_{SAMPLING}=f_{CK_INT}$，N=8

0100：$f_{SAMPLING}=f_{DTS}/2$，N=6

0101：$f_{SAMPLING}=f_{DTS}/2$，N=8

0110：$f_{SAMPLING}=f_{DTS}/4$，N=6

0111：$f_{SAMPLING}=f_{DTS}/4$，N=8

1000：$f_{SAMPLING}=f_{DTS}/8$，N=6

1001：$f_{SAMPLING}=f_{DTS}/8$，N=8

1010：$f_{SAMPLING}=f_{DTS}/16$，N=5

1011：$f_{SAMPLING}=f_{DTS}/16$，N=6

1100：$f_{SAMPLING}=f_{DTS}/16$，N=8

1101：$f_{SAMPLING}=f_{DTS}/32$，N=5

1110：$f_{SAMPLING}=f_{DTS}/32$，N=6

1111：$f_{SAMPLING}=f_{DTS}/32$，N=8

注： 编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1 后，此位即无法修改。

位 19:16 **BKF[3:0]**：断路滤波器 (Break filter)

此位域可定义 BRK 输入的采样频率和适用于 BRK 的数字滤波器带宽。数字滤波器由事件计数器组成，每 N 个连续事件才视为一个有效输出边沿：

0000：无滤波器，BRK 异步工作

0001：$f_{SAMPLING}=f_{CK_INT}$，N=2

0010：$f_{SAMPLING}=f_{CK_INT}$，N=4

0011：$f_{SAMPLING}=f_{CK_INT}$，N=8

0100：$f_{SAMPLING}=f_{DTS}/2$，N=6

0101：$f_{SAMPLING}=f_{DTS}/2$，N=8

0110：$f_{SAMPLING}=f_{DTS}/4$，N=6

0111：$f_{SAMPLING}=f_{DTS}/4$，N=8

1000：$f_{SAMPLING}=f_{DTS}/8$，N=6

1001：$f_{SAMPLING}=f_{DTS}/8$，N=8

1010：$f_{SAMPLING}=f_{DTS}/16$，N=5

1011：$f_{SAMPLING}=f_{DTS}/16$，N=6

1100：$f_{SAMPLING}=f_{DTS}/16$，N=8

1101：$f_{SAMPLING}=f_{DTS}/32$，N=5

1110：$f_{SAMPLING}=f_{DTS}/32$，N=6

1111：$f_{SAMPLING}=f_{DTS}/32$，N=8

注： 编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1 后，此位即无法修改。

位 15 **MOE**：主输出使能 (Main output enable)

只要断路输入（BRK 或 BRK2）为有效状态，此位便由硬件异步清零。此位由软件置 1，也可根据 AOE 位状态自动置 1。此位仅对配置为输出的通道有效。

0：响应断路事件（2 个）。禁止 OC 和 OCN 输出。

响应断路事件或向 MOE 写入 0 时：OC 和 OCN 输出被禁止或被强制为空闲状态，具体取决于 OSSI 位。

1：如果 OC 和 OCN 输出的相应使能位（TIMx_CCER 寄存器中的 CCxE 和 CCxNE 位）均置 1，则使能 OC 和 OCN 输出。

有关详细信息，请参见 OC/OCN 使能说明（*第 38.4.9 节：TIM1/TIM8 捕获/比较使能寄存器 (TIMx_CCER)*）。

位 14 **AOE**：自动输出使能 (Automatic output enable)

0：MOE 只能由软件置 1

1：MOE 可由软件置 1，也可在发生下一更新事件时自动置 1（如果断路输入 BRK 和 BRK2 均无效）

注：　只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1，此位即无法修改。

位 13 **BKP**：断路极性 (Break polarity)

0：断路输入 BRK 为低电平有效

1：断路输入 BRK 为高电平有效

注：　只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1，此位即无法修改。

注：　对该位执行任何写操作后，都需要经过 1 个 APB 时钟周期的延迟才生效。

位 12 **BKE**：断路使能 (Break enable)

该位可使能完整的断路保护（包括连接到 bk_acth 的所有源和相应的 BKIN 源，如*图 381：断路和断路 2 电路概述*所示）。

0：禁止断路功能

1：使能断路功能

注：　编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1 后，此位即无法修改。

注：　对该位执行任何写操作后，都需要经过 1 个 APB 时钟周期的延迟才生效。

位 11 **OSSR**：运行模式下的关闭状态选择 (Off-state selection for Run mode)

此位在 MOE=1 时作用于配置为输出模式且具有互补输出的通道。如果定时器中没有互补输出，则不存在 OSSR。

有关详细信息，请参见 OC/OCN 使能说明（*第 38.4.9 节：TIM1/TIM8 捕获/比较使能寄存器 (TIMx_CCER)*）。

0：处于无效状态时，禁止 OC/OCN 输出（定时器释放输出控制，由强制高阻态的 GPIO 逻辑接管）。

1：处于无效状态时，一旦 CCxE=1 或 CCxNE=1，便使能 OC/OCN 输出并将其设为无效电平（输出仍由定时器控制）。

注：　编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 2 后，此位即无法修改。

位 10 **OSSI**：空闲模式下的关闭状态选择 (Off-state selection for Idle mode)

当由于断路事件或软件写操作而使 MOE=0 时，此位作用于配置为输出的通道。

有关详细信息，请参见 OC/OCN 使能说明（*第 38.4.9 节：TIM1/TIM8 捕获/比较使能寄存器 (TIMx_CCER)*）。

0：处于无效状态时，禁止 OC/OCN 输出（定时器释放输出控制，由强制高阻态的 GPIO 逻辑接管）。

1：处于无效状态时，首先将 OC/OCN 输出强制为其无效电平，然后在死区后将其强制为空闲电平。定时器始终控制输出。

注：　编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 2 后，此位即无法修改。

位 9:8 **LOCK[1:0]**：锁定配置 (Lock configuration)

这些位用于针对软件错误提供写保护。

00：关闭锁定——不对任何位提供写保护。

01：锁定级别 1，此时无法对 TIMx_BDTR 寄存器中的 DTG 位、TIMx_CR2 寄存器中的 OISx 和 OISxN 位以及 TIMx_BDTR 寄存器中的 BKE/BKP/AOE 位执行写操作。

10：锁定级别 2，此时无法对锁定级别 1 中适用的各位、CC 极性位（TIMx_CCER 寄存器中的 CCxP/CCxNP 位，只要通过 CCxS 位将相关通道配置为输出）以及 OSSR 和 OSSI 位执行写操作。

11：锁定级别 3，此时无法对锁定级别 2 中适用的各位、CC 控制位（TIMx_CCMRx 寄存器中的 OCxM 和 OCxPE 位，只要通过 CCxS 位将相关通道配置为输出）执行写操作。

注： 复位后只能对 LOCK 位执行一次写操作。对 TIMx_BDTR 寄存器执行写操作后其中的内容将冻结，直到下一次复位。

位 7:0 **DTG[7:0]**：配置死区发生器 (Dead-time generator setup)

此位域定义插入到互补输出之间的死区持续时间。DT 与该持续时间相对应。

DTG[7:5]=0xx => DT=DTG[7:0]x t_{dtg}，其中 $t_{dtg}=t_{DTS}$。

DTG[7:5]=10x => DT=(64+DTG[5:0])xt_{dtg}，其中 $T_{dtg}=2xt_{DTS}$。

DTG[7:5]=110 => DT=(32+DTG[4:0])xt_{dtg}，其中 $T_{dtg}=8xt_{DTS}$。

DTG[7:5]=111 => DT=(32+DTG[4:0])xt_{dtg}，其中 $T_{dtg}=16xt_{DTS}$。

示例：如果 T_{DTS}=125ns (8MHz)，则可能的死区值为：

0 到 15875 ns（步长为 125 ns）

16 us 到 31750 ns（步长为 250 ns）

32 us 到 63us（步长为 1 us）

64 us 到 126 us（步长为 2 us）

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

38.4.19 TIM1/TIM8 DMA 控制寄存器 (TIMx_DCR)

TIM1/TIM8 DMA control register

偏移地址：0x48

复位值：0x0000

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

位 15:13 保留，必须保持复位值。

位 12:8 **DBL[4:0]**：DMA 连续传送长度 (DMA burst length)

该 5 位向量定义了 DMA 的传送长度（当对 TIMx_DMAR 地址进行读或写访问时，定时器进行一次连续传送），即传送次数。可按半字或字节进行传送（请参见下面的示例）。

00000：1 次传送
00001：2 次传送
00010：3 次传送
...
10001：18 次传送

示例：以下面的传送为例：DBL = 7 字节且 DBA = TIM2_CR1。

– 如果 DBL = 7 字节且 DBA = TIM2_CR1 表示待传送字节的地址，应通过以下公式给出传送的地址：

(TIMx_CR1 地址) + DBA + (DMA 索引)，其中 DMA 索引 = DBL

在本例中，将为 (TIMx_CR1 地址) + DBA 加上 7 个字节，得到将要复制数据的源/目标地址。在这种情况下，将向自以下地址开始的 7 个寄存器传送数据：(TIMx_CR1 地址) + DBA

根据 DMA 数据大小的配置，可能发生下面几种情况：

– 如果按半字配置 DMA 数据大小，则将向 7 个寄存器中的每一个传送 16 位数据。
– 如果按字节配置 DMA 数据大小，也将向 7 个寄存器传送数据：第一个寄存器包含第一个 MSB 字节，第二个寄存器包含第一个 LSB 字节，依此类推。因此，使用传送定时器时，还必须指定 DMA 传送的数据大小。

位 7:5 保留，必须保持复位值。

位 4:0 **DBA[4:0]**：DMA 基址 (DMA base address)

该 5 位向量定义 DMA 传输的基址（通过 TIMx_DMAR 地址进行读/写访问时）。DBA 定义为从 TIMx_CR1 寄存器地址开始计算的偏移量。

示例：

00000：TIMx_CR1，
00001：TIMx_CR2，
00010：TIMx_SMCR，
...

38.4.20 TIM1/TIM8 全传输 DMA 地址 (TIMx_DMAR)

TIM1/TIM8 DMA address for full transfer

偏移地址：0x4C

复位值：0x0000

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

位 31:0 **DMAB[31:0]**：DMA 连续传送寄存器 (DMA register for burst accesses)

对 DMAR 寄存器执行读或写操作将访问位于如下地址的寄存器：(TIMx_CR1 地址) + (DBA + DMA 索引) x 4

其中 TIMx_CR1 地址为控制寄存器 1 的地址，DBA 为 TIMx_DCR 寄存器中配置的 DMA 基址，DMA 索引由 DMA 传输自动控制，其范围介于 0 到 DBL（TIMx_DCR 寄存器中配置的 DBL）之间。

38.4.21 TIM1/TIM8 捕获/比较模式寄存器 3 (TIMx_CCMR3)

TIM1/TIM8 capture/compare mode register 3

偏移地址：0x54

复位值：0x0000 0000

请参见上述 CCMR1 寄存器说明。通道 5 和通道 6 只能配置为输出。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC6M[3]	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC5M[3]
							rw								rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OC6 CE	OC6M[2:0]			OC6 PE	OC6FE	Res.	Res.	OC5 CE.	OC5M[2:0]			OC5PE	OC5FE	Res.	Res.
rw	rw	rw	rw	rw	rw			rw	rw	rw	rw	rw	rw		

输出比较模式

位 31:25 保留，必须保持复位值。

位 24 **OC6M[3]**：输出比较 6 模式——位 3 (Output Compare 6 mode - bit 3)

位 23:17 保留，必须保持复位值。

位 16 **OC5M[3]**：输出比较 5 模式——位 3 (Output Compare 5 mode - bit 3)

位 15 **OC6CE**：输出比较 6 清零使能 (Output compare 6 clear enable)

位 14:12 **OC6M**：输出比较 6 模式 (Output compare 6 mode)

位 11 **OC6PE**：输出比较 6 预装载使能 (Output compare 6 preload enable)

位 10 **OC6FE**：输出比较 6 快速使能 (Output compare 6 fast enable)

位 9:8 保留，必须保持复位值。

位 7 **OC5CE**：输出比较 5 清零使能 (Output compare 5 clear enable)

位 6:4 **OC5M**：输出比较 5 模式 (Output compare 5 mode)

位 3 **OC5PE**：输出比较 5 预装载使能 (Output Compare 5 preload enable)

位 2 **OC5FE**：输出比较 5 快速使能 (Output compare 5 fast enable)

位 1:0 保留，必须保持复位值。

38.4.22 TIM1/TIM8 捕获/比较寄存器 5 (TIMx_CCR5)

TIM1/TIM8 capture/compare register 5

偏移地址：0x58

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GC5C3	GC5C2	GC5C1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
rw	rw	rw													
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR5[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31 **GC5C3**：通道 5 和通道 3 组 (Group Channel 5 and Channel 3)

通道 3 输出上失真：

0：OC5REF 对 OC3REFC 无影响

1：OC3REFC 是 OC3REFC 和 OC5REF 的逻辑与运算结果

该位可以立即生效，也可预装载并在更新事件后执行（如果在 TIMxCCMR2 中选择了预装载功能）。

注： 也可在组合 PWM 信号上应用此失真。

位 30 **GC5C2**：通道 5 和通道 2 组 (Group Channel 5 and Channel 2)

通道 2 输出上失真：

0：OC5REF 对 OC2REFC 无影响

1：OC2REFC 是 OC2REFC 和 OC5REF 的逻辑与运算结果

该位可以立即生效，也可预装载并在更新事件后执行（如果在 TIMxCCMR1 中选择了预装载功能）。

注： 也可在组合 PWM 信号上应用此失真。

位 29 **GC5C1**：通道 5 和通道 1 组 (Group Channel 5 and Channel 1)

通道 1 输出上失真：

0：OC5REF 对 OC1REFC5 无影响

1：OC1REFC 是 OC1REFC 和 OC5REF 的逻辑与运算结果

该位可以立即生效，也可预装载并在更新事件后执行（如果在 TIMxCCMR1 中选择了预装载功能）。

注： 也可在组合 PWM 信号上应用此失真。

位 28:16 保留，必须保持复位值。

位 15:0 **CCR5[15:0]**：捕获/比较 5 值 (Capture/Compare 5 value)

CCR5 是捕获/比较寄存器 5 的预装载值。

如果没有通过 TIMx_CCMR3 寄存器中的 OC5PE 位来使能预装载功能，则该值立刻生效；否则只在发生更新事件时生效（拷贝到有效的捕获/比较寄存器 5）。

有效捕获/比较寄存器中包含要与计数器 TIMx_CNT 进行比较并在 OC5 输出上发出信号的值。

38.4.23 TIM1/TIM8 捕获/比较寄存器 6 (TIMx_CCR6)

TIM1/TIM8 capture/compare register 6

偏移地址：0x5C

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR6[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 15:0 **CCR6[15:0]**：捕获/比较 6 值 (Capture/Compare 6 value)

CCR6 是捕获/比较寄存器 6 的预装载值。

如果没有通过 TIMx_CCMR3 寄存器中的 OC6PE 位来使能预装载功能，则该值立刻生效；否则只在发生更新事件时生效（拷贝到有效的捕获/比较寄存器 6）。

有效捕获/比较寄存器中包含要与计数器 TIMx_CNT 进行比较并在 OC6 输出上发出信号的值。

38.4.24 TIM1 复用功能选项寄存器 1 (TIM1_AF1)

TIM1 alternate function option register 1

偏移地址：0x60

复位值：0x0000 0001

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ETRSEL[3:2]	
														rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ETRSEL[1:0]		Res.	Res.	BK CMP2P	BK CMP1P	BKINP	BKDF1 BK0E	Res.	Res.	Res.	Res.	Res.	BK CMP2E	BK CMP1E	BKINE
rw	rw			rw	rw	rw	rw						rw	rw	rw

位 31:18 保留，必须保持复位值

位 17:14 **ETRSEL[3:0]**：ETR 源选择 (ETR source selection)

这些位选择 ETR 输入源。

0000：ETR 输入连接到 I/O

0001：COMP1 输出

0010：COMP2 输出

0011：ADC1 AWD1

0100：ADC1 AWD2

0101：ADC1 AWD3

0110：ADC3 AWD1

0111：ADC3 AWD2

1000：ADC3 AWD3

其它值：保留

注： 只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1，这些位即无法修改。

位 13:12 保留，必须保持复位值

位 11 **BKCMP2P**：BRK COMP2 输入极性 (BRK COMP2 input polarity)
此位选择 COMP2 输入灵敏度， 必须与 BKP 极性位一起编程。
0：COMP2 输入为高电平有效
1：COMP2 输入为低电平有效
注： 只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1，此位即无法修改。

位 10 **BKCMP1P**：BRK COMP1 输入极性 (BRK COMP1 input polarity)
此位选择 COMP1 输入灵敏度， 必须与 BKP 极性位一起编程。
0：COMP1 输入为高电平有效
1：COMP1 输入为低电平有效
注： 只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1，此位即无法修改。

位 9 **BKINP**：BRK BKIN 输入极性 (BRK BKIN input polarity)
此位选择 BKIN 复用功能输入灵敏度， 必须与 BKP 极性位一起编程。
0：BKIN 输入为高电平有效
1：BKIN 输入为低电平有效
注： 只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1，此位即无法修改。

位 8 **BKDF1BK0E**：BRK dfsdm1_break[0] 使能 (BRK dfsdm1_break[0] enable)
此位使能定时器 BRK 输入的 dfsdm1_break[0]。dfsdm1_break[0] 输出与其他 BRK 源进行“或”运算。
0：禁止 dfsdm1_break[0] 输入
1：使能 dfsdm1_break[0] 输入
注： 只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1，此位即无法修改。

位 7:3 保留，必须保持复位值

位 2 **BKCMP2E**：BRK COMP2 使能 (BRK COMP2 enable)
此位使能定时器 BRK 输入的 COMP2。COMP2 输出与其他 BRK 源进行“或”运算。
0：禁止 COMP2 输入
1：使能 COMP2 输入
注： 只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1，此位即无法修改。

位 1 **BKCMP1E**：BRK COMP1 使能 (BRK COMP1 enable)
此位使能定时器 BRK 输入的 COMP1。COMP1 输出与其他 BRK 源进行“或”运算。
0：禁止 COMP1 输入
1：使能 COMP1 输入
注： 只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1，此位即无法修改。

位 0 **BKINE**：BRK BKIN 输入使能 (BRK BKIN input enable)
此位使能定时器 BRK 输入的 BKIN 复用功能。BKIN 输入与其他 BRK 源进行“或”运算。
0：禁止 BKIN 输入
1：使能 BKIN 输入
注： 只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1，此位即无法修改。

注： 请参见图 360：TIM1/TIM8 ETR 输入电路和图 381：断路和断路 2 电路概述。

38.4.25 TIM1 复用功能寄存器 2 (TIM1_AF2)

TIM1 Alternate function register 2

偏移地址：0x64

复位值：0x0000 0001

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	BK2 CMP2 P	BK2 CMP1 P	BK2 INP	BK2DF1 BK1E	Res.	Res.	Res.	Res.	Res.	BK2 CMP2E	BK2 CMP1E	BK2INE
				rw	rw	rw	rw						rw	rw	rw

位 31:12 保留，必须保持复位值

位 11 **BK2CMP2P**：BRK2 COMP2 输入极性 (BRK2 COMP2 input polarity)

此位选择 COMP2 输入灵敏度， 必须与 BKP2 极性位一起编程。

0：COMP2 输入为低电平有效

1：COMP2 输入为高电平有效

注： 只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1，此位即无法修改。

位 10 **BK2CMP1P**：BRK2 COMP1 输入极性 (BRK2 COMP1 input polarity)

此位选择 COMP1 输入灵敏度， 必须与 BKP2 极性位一起编程。

0：COMP1 输入为低电平有效

1：COMP1 输入为高电平有效

注： 只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1，此位即无法修改。

位 9 **BK2INP**：BRK2 BKIN2 输入极性 (BRK2 BKIN2 input polarity)

此位选择 BKIN2 复用功能输入灵敏度， 必须与 BKP2 极性位一起编程。

0：BKIN2 输入为低电平有效

1：BKIN2 输入为高电平有效

注： 只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1，此位即无法修改。

位 8 **BK2DF1BK1E**：BRK2 dfsdm1_break[1] 使能 (BRK2 dfsdm1_break[1] enable)

此位使能定时器 BRK2 输入的 dfsdm1_break[1]。dfsdm1_break[1] 输出与其他 BRK2 源进行“或”运算。

0：禁止 dfsdm1_break[1] 输入

1：使能 dfsdm1_break[1] 输入

注： 只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1，此位即无法修改。

位 7:3 保留，必须保持复位值

位 2 **BK2CMP2E**：BRK2 COMP2 使能 (BRK2 COMP2 enable)

此位使能定时器 BRK2 输入的 COMP2。COMP2 输出与其他 BRK2 源进行“或”运算。

0：禁止 COMP2 输入

1：使能 COMP2 输入

注： 只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1，此位即无法修改。

位 1 **BK2CMP1E**：BRK2 COMP1 使能 (BRK2 COMP1 enable)

此位使能定时器 BRK2 输入的 COMP1。COMP1 输出与其他 BRK2 源进行“或”运算。

0：禁止 COMP1 输入

1：使能 COMP1 输入

注： 只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1，此位即无法修改。

位 0 **BK2INE**：BRK2 BKIN 输入使能 (BRK2 BKIN input enable)
此位使能定时器 BRK2 输入的 BKIN2 复用功能。BKIN2 输入与其他 BRK2 源进行“或”运算。
0：禁止 BKIN2 输入
1：使能 BKIN2 输入
注： 只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1，此位即无法修改。

注： 请参见图 381：断路和断路 2 电路概述。

38.4.26 TIM8 复用功能选项寄存器 1 (TIM8_AF1)

TIM8 Alternate function option register 1

偏移地址：0x60

复位值：0x0000 0001

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ETRSEL[3:2]	
														rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ETRSEL[1:0]		Res.	Res.	BK CMP2 P	BK CMP1 P	BKINP	BKDF1 BK2E	Res.	Res.	Res.	Res.	Res.	BK CMP2E	BK CMP1E	BKINE
rw	rw			rw	rw	rw	rw						rw	rw	rw

位 31:18 保留，必须保持复位值

位 17:14 **ETRSEL[3:0]**：ETR 源选择 (ETR source selection)
这些位选择 ETR 输入源。
0000：ETR 输入连接到 I/O
0001：COMP1 输出
0010：COMP2 输出
0011：ADC2 AWD1
0100：ADC2 AWD2
0101：ADC2 AWD3
0110：ADC3 AWD1
0111：ADC3 AWD2
1000：ADC3 AWD3
其它值：保留
注： 只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1，这些位即无法修改。

位 13:12 保留，必须保持复位值

位 11 **BKCMP2P**：BRK COMP2 输入极性 (BRK COMP2 input polarity)
此位选择 COMP2 输入灵敏度， 必须与 BKP 极性位一起编程。
0：COMP2 输入为高电平有效
1：COMP2 输入为低电平有效
注： 只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1，此位即无法修改。

位 10 **BKCMP1P**：BRK COMP1 输入极性 (BRK COMP1 input polarity)
此位选择 COMP1 输入灵敏度， 必须与 BKP 极性位一起编程。
0：COMP1 输入为高电平有效
1：COMP1 输入为低电平有效
注： 只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1，此位即无法修改。

位 9 **BKINP**：BRK BKIN 输入极性 (BRK BKIN input polarity)
此位选择 BKIN 复用功能输入灵敏度， 必须与 BKP 极性位一起编程。
0：BKIN 输入为高电平有效
1：BKIN 输入为低电平有效
注： 只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1，此位即无法修改。

位 8 **BKDF1BK2E**：BRK dfsdm1_break[2] 使能 (BRK dfsdm1_break[2] enable)
此位使能定时器 BRK 输入的 dfsdm1_break[2]。dfsdm1_break[2] 输出与其他 BRK 源进行“或”运算。
0：禁止 dfsdm1_break[2] 输入
1：使能 dfsdm1_break[2] 输入
注： 只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1，此位即无法修改。

位 7:3 保留，必须保持复位值

位 2 **BKCMP2E**：BRK COMP2 使能 (BRK COMP2 enable)
此位使能定时器 BRK 输入的 COMP2。COMP2 输出与其他 BRK 源进行“或”运算。
0：禁止 COMP2 输入
1：使能 COMP2 输入
注： 只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1，此位即无法修改。

位 1 **BKCMP1E**：BRK COMP1 使能 (BRK COMP1 enable)
此位使能定时器 BRK 输入的 COMP1。COMP1 输出与其他 BRK 源进行“或”运算。
0：禁止 COMP1 输入
1：使能 COMP1 输入
注： 只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1，此位即无法修改。

位 0 **BKINE**：BRK BKIN 输入使能 (BRK BKIN input enable)
此位使能定时器 BRK 输入的 BKIN 复用功能。BKIN 输入与其他 BRK 源进行“或”运算。
0：禁止 BKIN 输入
1：使能 BKIN 输入
注： 只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1，此位即无法修改。

注： 请参见图 360：TIM1/TIM8 ETR 输入电路和图 381：断路和断路 2 电路概述。

38.4.27 TIM8 复用功能选项寄存器 2 (TIM8_AF2)

TIM8 Alternate function option register 2

偏移地址：0x64

复位值：0x0000 0001

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	BK2 CMP2 P	BK2 CMP1 P	BK2 INP	BK2DF1 BK3E	Res.	Res.	Res.	Res.	Res.	BK2 CMP2E	BK2 CMP1E	BK2INE
				rw	rw	rw	rw						rw	rw	rw

位 31:12 保留，必须保持复位值

位 11 **BK2CMP2P**：BRK2 COMP2 输入极性 (BRK2 COMP2 input polarity)
此位选择 COMP2 输入灵敏度， 必须与 BKP2 极性位一起编程。
0：COMP2 输入为低电平有效
1：COMP2 输入为高电平有效
注： 只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1，此位即无法修改。

位 10 **BK2CMP1P**：BRK2 COMP1 输入极性 (BRK2 COMP1 input polarity)
此位选择 COMP1 输入灵敏度， 必须与 BKP2 极性位一起编程。
0：COMP1 输入为低电平有效
1：COMP1 输入为高电平有效
注： 只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1，此位即无法修改。

位 9 **BK2INP**：BRK2 BKIN2 输入极性 (BRK2 BKIN2 input polarity)
此位选择 BKIN2 复用功能输入灵敏度， 必须与 BKP2 极性位一起编程。
0：BKIN2 输入为低电平有效
1：BKIN2 输入为高电平有效
注： 只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1，此位即无法修改。

位 8 **BK2DF1BK3E**：BRK2 dfsdm1_break[3] 使能 (BRK2 dfsdm1_break[3] enable)
此位使能定时器 BRK2 输入的 dfsdm1_break[3]。dfsdm1_break[3] 输出与其他 BRK2 源进行“或”运算。
0：禁止 dfsdm1_break[3] 输入
1：使能 dfsdm1_break[3] 输入
注： 只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1，此位即无法修改。

位 7:3 保留，必须保持复位值

位 2 **BK2CMP2E**：BRK2 COMP2 使能 (BRK2 COMP2 enable)
此位使能定时器 BRK2 输入的 COMP2。COMP2 输出与其他 BRK2 源进行“或”运算。
0：禁止 COMP2 输入
1：使能 COMP2 输入
注： 只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1，此位即无法修改。

位 1 **BK2CMP1E**：BRK2 COMP1 使能 (BRK2 COMP1 enable)
此位使能定时器 BRK2 输入的 COMP1。COMP1 输出与其他 BRK2 源进行“或”运算。
0：禁止 COMP1 输入
1：使能 COMP1 输入
注： 只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1，此位即无法修改。

位 0 **BK2INE**：BRK2 BKIN 输入使能 (BRK2 BKIN input enable)
此位使能定时器 BRK2 输入的 BKIN2 复用功能。BKIN2 输入与其他 BRK2 源进行“或”运算。
0：禁止 BKIN2 输入
1：使能 BKIN2 输入
注： 只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1，此位即无法修改。

注： 请参见图 381：断路和断路 2 电路概述。

38.4.28 TIM1 定时器输入选择寄存器 (TIM1_TISEL)

TIM1 timer input selection register

偏移地址：0x68

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	TI4SEL[3:0]				Res.	Res.	Res.	Res.	TI3SEL[3:0]			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	TI2SEL[3:0]				Res.	Res.	Res.	Res.	TI1SEL[3:0]			
				rw	rw	rw	rw						rw	rw	rw

位 31:28 保留，必须保持复位值

位 27:24 **TI4SEL[3:0]**：选择 TI4[0] 到 TI4[15] 输入 (selects TI4[0] to TI4[15] input)
0000：TIM1_CH4 输入
其它值：保留

位 23:20 保留，必须保持复位值

位 19:16 **TI3SEL[3:0]**：选择 TI3[0] 到 TI3[15] 输入 (selects TI3[0] to TI3[15] input)
0000：TIM1_CH3 输入
其它值：保留

位 15:12 保留，必须保持复位值

位 11:8 **TI2SEL[3:0]**：选择 TI2[0] 到 TI2[15] 输入 (selects TI2[0] to TI2[15] input)
0000：TIM1_CH2 输入
其它值：保留

位 7:4 保留，必须保持复位值

位 3:0 **TI1SEL[3:0]**：选择 TI1[0] 到 TI1[15] 输入 (selects TI1[0] to TI1[15] input)
0000：TIM1_CH1 输入
0001：COMP1 输出
其它值：保留

38.4.29 TIM8 定时器输入选择寄存器 (TIM8_TISEL)

TIM8 timer input selection register

偏移地址：0x68

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	TI4SEL[3:0]				Res.	Res.	Res.	Res.	TI3SEL[3:0]			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	TI2SEL[3:0]				Res.	Res.	Res.	Res.	TI1SEL[3:0]			
				rw	rw	rw	rw						rw	rw	rw

位 31:28 保留，必须保持复位值

位 27:24 **TI4SEL[3:0]**：选择 TI4[0] 到 TI4[15] 输入 (selects TI4[0] to TI4[15] input)
0000：TIM8_CH4 输入
其它值：保留

位 23:20 保留，必须保持复位值

位 19:16 **TI3SEL[3:0]**：选择 TI3[0] 到 TI3[15] 输入 (selects TI3[0] to TI3[15] input)
0000：TIM8_CH3 输入
其它值：保留

位 15:12 保留，必须保持复位值

位 11:8 **TI2SEL[3:0]**：选择 TI2[0] 到 TI2[15] 输入 (selects TI2[0] to TI2[15] input)
0000：TIM8_CH2 输入
其它值：保留

位 7:4 保留，必须保持复位值

位 3:0 **TI1SEL[3:0]**：选择 TI1[0] 到 TI1[15] 输入 (selects TI1[0] to TI1[15] input)
0000：TIM8_CH1 输入
0001：COMP2 输出
其它值：保留

38.4.30 TIM1 寄存器映射

TIM1 寄存器可映射为 16 位可寻址寄存器，如下表所述：

表 314. TIM1 寄存器映射和复位值

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	**TIM1_CR1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	UIFREMAP	Res.	CKD [1:0]		ARPE	CMS [1:0]		DIR	OPM	URS	UDIS	CEN
	Reset value																					0		0	0	0	0	0	0	0	0	0	0
0x04	**TIM1_CR2**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MMS2[3:0]				Res.	OIS6	Res.	OIS5	Res.	OIS4	OIS3N	OIS3	OIS2N	OIS2	OIS1N	OIS1	TI1S	MMS [2:0]			CCDS	CCUS	Res.	CCPC
	Reset value									0	0	0	0		0		0		0	0	0	0	0	0	0	0	0	0	0	0	0		0
0x08	**TIM1_SMCR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TS [4:3]		Res.	Res.	Res.	SMS[3]	ETP	ECE	ETPS [1:0]		ETF[3:0]				MSM	TS[2:0]			Res.	SMS[2:0]		
	Reset value											0	0				0	0	0	0	0	0	0	0	0	0	0	0	0		0	0	0
0x0C	**TIM1_DIER**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TDE	COMDE	CC4DE	CC3DE	CC2DE	CC1DE	UDE	BIE	TIE	COMIE	CC4IE	CC3IE	CC2IE	CC1IE	UIE
	Reset value																		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x10	**TIM1_SR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CC6IF	CC5IF	Res.	Res.	SBIF	CC4OF	CC3OF	CC2OF	CC1OF	B2IF	BIF	TIF	COMIF	CC4IF	CC3IF	CC2IF	CC1IF	UIF
	Reset value															0	0			0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x14	**TIM1_EGR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	B2G	BG	TG	COM	CC4G	CC3G	CC2G	CC1G	UG
	Reset value																								0	0	0	0	0	0	0	0	0
0x18	**TIM1_CCMR1** Output Compare mode	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC2M[3]	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC1M[3]	OC2CE	OC2M [2:0]			OC2PE	OC2FE	CC2S [1:0]		OC1CE	OC1M [2:0]			OC1PE	OC1FE	CC1S [1:0]	
	Reset value								0								0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	TIM1_CCMR1 Input Capture mode	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	IC2F[3:0]				IC2PSC [1:0]		CC2S [1:0]		IC1F[3:0]				IC1PSC [1:0]		CC1S [1:0]	
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x1C	**TIM1_CCMR2** Output Compare mode	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC4M[3]	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC3M[3]	OC4CE	OC4M [2:0]			OC4PE	OC4FE	CC4S [1:0]		OC3CE	OC3M [2:0]			OC3PE	OC3FE	CC3S [1:0]	
	Reset value								0								0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	TIM1_CCMR2 Input Capture mode	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	IC4F[3:0]				IC4PSC [1:0]		CC4S [1:0]		IC3F[3:0]				IC3PSC [1:0]		CC3S [1:0]	
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x20	**TIM1_CCER**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CC6P	CC6E	Res.	Res.	CC5P	CC5E	Res.	Res.	CC4P	CC4E	CC3NP	CC3NE	CC3P	CC3E	CC2NP	CC2NE	CC2P	CC2E	CC1NP	CC1NE	CC1P	CC1E
	Reset value											0	0			0	0			0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x24	**TIM1_CNT**	UIFCPY	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CNT[15:0]															
	Reset value	0																0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

表 314. TIM1 寄存器映射和复位值（续）

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x28	**TIM1_PSC**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PSC[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x2C	**TIM1_ARR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ARR[15:0]															
	Reset value																	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0x30	**TIM1_RCR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REP[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x34	**TIM1_CCR1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CCR1[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x38	**TIM1_CCR2**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CCR2[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x3C	**TIM1_CCR3**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CCR3[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x40	**TIM1_CCR4**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CCR4[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x44	**TIM1_BDTR**	Res.	Res.	Res.	Res.	Res.	Res.	BK2P	BK2E	BK2F[3:0]				BKF[3:0]				MOE	AOE	BKP	BKE	OSSR	OSSI	LOCK [1:0]		DT[7:0]							
	Reset value							0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x48	**TIM1_DCR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DBL[4:0]					Res.	Res.	Res.	DBA[4:0]				
	Reset value																				0	0	0	0	0				0	0	0	0	0
0x4C	**TIM1_DMAR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DMAB[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x54	**TIM1_CCMR3** Output Compare mode	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC6M[3]	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC5M[3]	OC6CE	OC6M [2:0]			OC6PE	OC6FE	Res.	Res.	OC5CE	OC5M [2:0]			OC5PE	OC5FE	Res.	Res.
	Reset value								0								0	0	0	0	0	0	0			0	0	0	0	0	0		
0x58	**TIM1_CCR5**	GC5C3	GC5C2	GC5C1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CCR5[15:0]															
	Reset value	0	0	0														0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

表 314. TIM1 寄存器映射和复位值（续）

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x5C	**TIM1_CCR6**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CCR6[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x60	**TIM1_AF1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res ETRSEL [3:0]				Res.	Res.	BKCMP2P	BKCMP1P	BKINP	BKDF1BK0E	Res.	Res.	Res.	Res.	Res.	BKCMP2E	BKCMP1E	BKINE
	Reset value															0	0	0	0			0	0	0	0						0	0	1
0x64	**TIM1_AF2**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BK2CMP2P	BK2CMP1P	BK2INP	BK2DF1BK1E	Res.	Res.	Res.	Res.	Res.	BK2CMP2E	BK2CMP1E	BK2INE
	Reset value																					0	0	0	0						0	0	1
0x68	**TIM1_TISEL**	Res.	Res.	Res.	Res.	TI4SEL[3:0]				Res.	Res.	Res.	Res.	TI3SEL[3:0]				Res.	Res.	Res.	Res.	TI2SEL[3:0]				Res.	Res.	Res.	Res.	TI1SEL[3:0]			
	Reset value					0	0	0	0					0	0	0	0					0	0	0	0						0	0	0

有关寄存器边界地址的信息，请参见*第 2.2.2 节：存储器映射和寄存器边界地址*。

38.4.31 TIM8 寄存器映射

TIM8 寄存器可映射为 16 位可寻址寄存器，如下表所述：

表 315. TIM8 寄存器映射和复位值

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	**TIM8_CR1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	UIFREMAP	Res.	CKD [1:0]		ARPE	CMS [1:0]		DIR	OPM	URS	UDIS	CEN
	Reset value																					0		0	0	0	0	0	0	0	0	0	0
0x04	**TIM8_CR2**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MMS2[3:0]				Res.	OIS6	Res.	OIS5	Res.	OIS4	OIS3N	OIS3	OIS2N	OIS2	OIS1N	OIS1	TI1S	MMS [2:0]			CCDS	CCUS	Res.	CCPC
	Reset value									0	0	0	0		0		0		0	0	0	0	0	0	0	0	0	0	0	0	0		0
0x08	**TIM8_SMCR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TS [4:3]		Res.	Res.	Res.	SMS[3]	ETP	ECE	ETPS [1:0]		ETF[3:0]			MSM	TS[2:0]			Res.	SMS[2:0]			
	Reset value											0	0				0	0	0	0	0	0	0	0	0	0	0	0		0	0	0	
0x0C	**TIM8_DIER**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TDE	COMDE	CC4DE	CC3DE	CC2DE	CC1DE	UDE	BIE	TIE	COMIE	CC4IE	CC3IE	CC2IE	CC1IE	UIE
	Reset value																		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x10	**TIM8_SR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CC6IF	CC5IF	Res.	Res.	SBIF	CC4OF	CC3OF	CC2OF	CC1OF	B2IF	BIF	TIF	COMIF	CC4IF	CC3IF	CC2IF	CC1IF	UIF
	Reset value															0	0			0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x14	**TIM8_EGR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	B2G	BG	TG	COM	CC4G	CC3G	CC2G	CC1G	UG
	Reset value																								0	0	0	0	0	0	0	0	0
0x18	**TIM8_CCMR1** Output Compare mode	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC2M[3]	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC1M[3]	OC2CE	OC2M [2:0]			OC2PE	OC2FE	CC2S [1:0]		OC1CE	OC1M [2:0]			OC1PE	OC1FE	CC1S [1:0]	
	Reset value								0								0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	TIM8_CCMR1 Input Capture mode	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	IC2F[3:0]				IC2PSC [1:0]		CC2S [1:0]		IC1F[3:0]				IC1PSC [1:0]		CC1S [1:0]	
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x1C	**TIM8_CCMR2** Output Compare mode	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC4M[3]	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC3M[3]	OC4CE	OC4M [2:0]			OC4PE	OC4FE	CC4S [1:0]		OC3CE	OC3M [2:0]			OC3PE	OC3FE	CC3S [1:0]	
	Reset value								0								0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	TIM8_CCMR2 Input Capture mode	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	IC4F[3:0]				IC4PSC [1:0]		CC4S [1:0]		IC3F[3:0]				IC3PSC [1:0]		CC3S [1:0]	
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x20	**TIM8_CCER**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CC6P	CC6E	Res.	Res.	CC5P	CC5E	Res.	Res.	CC4P	CC4E	CC3NP	CC3NE	CC3P	CC3E	CC2NP	CC2NE	CC2P	CC2E	CC1NP	CC1NE	CC1P	CC1E
	Reset value											0	0			0	0			0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x24	**TIM8_CNT**	UIFCPY	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CNT[15:0]															
	Reset value	0																0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

表 315. TIM8 寄存器映射和复位值（续）

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x28	**TIM8_PSC**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PSC[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x2C	**TIM8_ARR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ARR[15:0]															
	Reset value																	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0x30	**TIM8_RCR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REP[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x34	**TIMx_CCR1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CCR1[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x38	**TIM8_CCR2**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CCR2[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x3C	**TIM8_CCR3**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CCR3[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x40	**TIM8_CCR4**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CCR4[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x44	**TIM8_BDTR**	Res.	Res.	Res.	Res.	Res.	Res.	BK2P	BK2E	BK2F[3:0]				BKF[3:0]				MOE	AOE	BKP	BKE	OSSR	OSSI	LOCK [1:0]		DT[7:0]							
	Reset value							0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x48	**TIM8_DCR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DBL[4:0]					Res.	Res.	Res.	DBA[4:0]				
	Reset value																				0	0	0	0	0				0	0	0	0	0
0x4C	**TIM8_DMAR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DMAB[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x54	**TIM8_CCMR3** Output Compare mode	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC6M[3]	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC5M[3]	OC6CE	OC6M [2:0]			OC6PE	OC6FE	Res.	Res.	OC5CE	OC5M [2:0]			OC5PE	OC5FE	Res.	Res.
	Reset value								0								0	0	0	0	0	0	0			0	0	0	0	0	0		
0x58	**TIM8_CCR5**	GC5C3	GC5C2	GC5C1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CCR5[15:0]															
	Reset value	0	0	0														0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

表 315. TIM8 寄存器映射和复位值（续）

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x5C	**TIM8_CCR6**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CCR6[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x60	**TIM8_AF1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res ETRSEL [3:0]				Res.	Res.	BKCMP2P	BKCMP1P	BKINP	BKDF1BK2E	Res.	Res.	Res.	Res.	Res.	BKCMP2E	BKCMP1E	BKINE
	Reset value															0	0	0	0			0	0	0	0						0	0	1
0x64	**TIM8_AF2**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BK2CMP2P	BK2CMP1P	BK2INP	BK2DF1BK3E	Res.	Res.	Res.	Res.	Res.	BK2CMP2E	BK2CMP1E	BK2INE
	Reset value																					0	0	0	0						0	0	1
0x68	**TIM8_TISEL**	Res.	Res.	Res.	Res.	TI4SEL[3:0]				Res.	Res.	Res.	Res.	TI3SEL[3:0]				Res.	Res.	Res.	Res.	TI2SEL[3:0]				Res.	Res.	Res.	Res.	TI1SEL[3:0]			
	Reset value					0	0	0	0					0	0	0	0					0	0	0	0						0	0	0	0

有关寄存器边界地址的信息，请参见*第 100 页的第 2.2.2 节*。

39 通用定时器 (TIM2/TIM3/TIM4/TIM5)

39.1 TIM2/TIM3/TIM4/TIM5 简介

通用定时器包含一个 16 位或 32 位自动重载计数器，该计数器由可编程预分频器驱动。

它们可用于多种用途，包括测量输入信号的脉冲宽度（*输入捕获*）或生成输出波形（*输出比较和 PWM*）。

使用定时器预分频器和 RCC 时钟控制器预分频器，可将脉冲宽度和波形周期从几微秒调制到几毫秒。

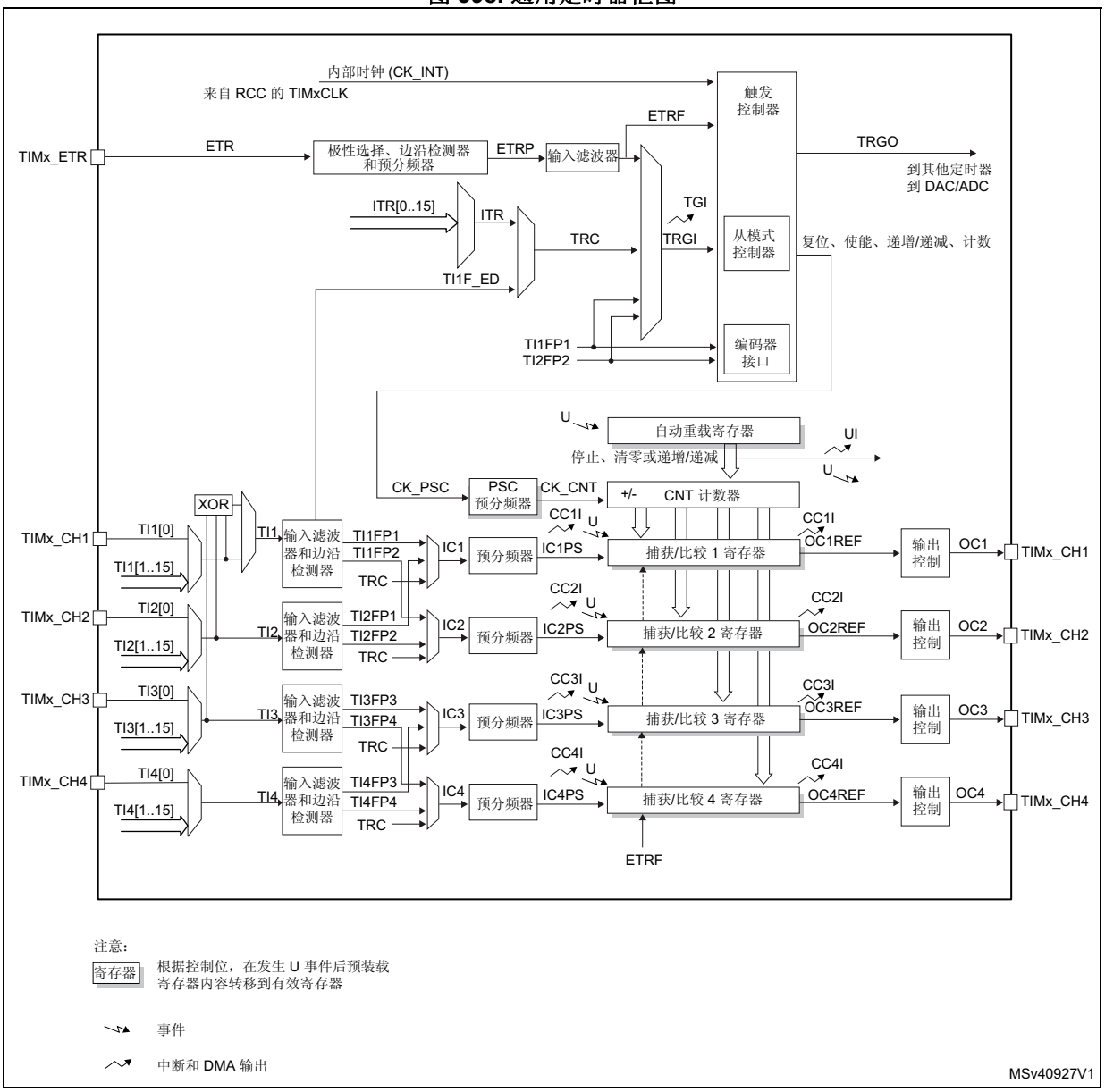
这些定时器彼此完全独立，不共享任何资源。如*第 39.3.19 节：定时器同步*中所述，它们可以同步操作。

39.2 TIM2/TIM3/TIM4/TIM5 主要特性

通用 TIMx 定时器具有以下特性：

- 16 位（TIM3 和 TIM4）或 32 位（TIM2 和 TIM5）递增、递减和递增/递减自动重载计数器。
- 16 位可编程预分频器，用于对计数器时钟频率进行分频（可在运行时修改），分频系数介于 1 到 65535 之间。
- 多达 4 个独立通道，可用于：
  - 输入捕获
  - 输出比较
  - PWM 生成（边沿和中心对齐模式）
  - 单脉冲模式输出
- 使用外部信号控制定时器且可实现多个定时器互连的同步电路。
- 发生如下事件时生成中断/DMA 请求：
  - 更新：计数器上溢/下溢、计数器初始化（通过软件或内部/外部触发）
  - 触发事件（计数器启动、停止、初始化或通过内部/外部触发计数）
  - 输入捕获
  - 输出比较
- 支持定位用增量（正交）编码器和霍尔传感器电路
- 触发输入用作外部时钟或逐周期电流管理

图 398. 通用定时器框图



39.3 TIM2/TIM3/TIM4/TIM5 功能描述

39.3.1 时基单元

可编程定时器的主要模块由一个 16 位/32 位计数器及其相关的自动重装寄存器组成。计数器可递增计数、递减计数或交替进行递增和递减计数。计数器的时钟可通过预分频器进行分频。

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

时基单元包括：

- 计数器寄存器 (TIMx_CNT)
- 预分频器寄存器 (TIMx_PSC)
- 自动重载寄存器 (TIMx_ARR)

自动重载寄存器是预装载的。对自动重载寄存器执行写入或读取操作时会访问预装载寄存器。预装载寄存器的内容既可以立即传送到影子寄存器，也可以在每次发生更新事件 (UEV) 时传送到影子寄存器，这取决于 TIMx_CR1 寄存器中的自动重载预装载使能位 (ARPE)。当计数器达到上溢值（或者在递减计数时达到下溢值）并且 TIMx_CR1 寄存器中的 UDIS 位为 0 时，将发送更新事件。该更新事件也可由软件产生。下文将针对各配置的更新事件的产生进行详细介绍。

计数器由预分频器输出 CK_CNT 提供时钟，仅当 TIMx_CR1 寄存器中的计数器启动位 (CEN) 置 1 时，才会启动计数器（有关计数器使能的更多详细信息，另请参见从模式控制器的相关说明）。

请注意，实际的计数器使能信号 CNT_EN 在 CEN 置 1 的一个时钟周期后被置 1。

预分频器说明

预分频器可对计数器时钟频率进行分频，分频系数介于 1 和 65536 之间。该预分频器基于 16 位/32 位寄存器（TIMx_PSC 寄存器）所控制的 16 位计数器。由于该控制寄存器具有缓冲功能，因此预分频器可实现实时更改。而新的预分频比将在下一更新事件发生时被采用。

图 399 和 *图 400* 以一些示例说明在预分频比实时变化时计数器的行为：

图 399. 预分频器分频由 1 变为 2 时的计数器时序图

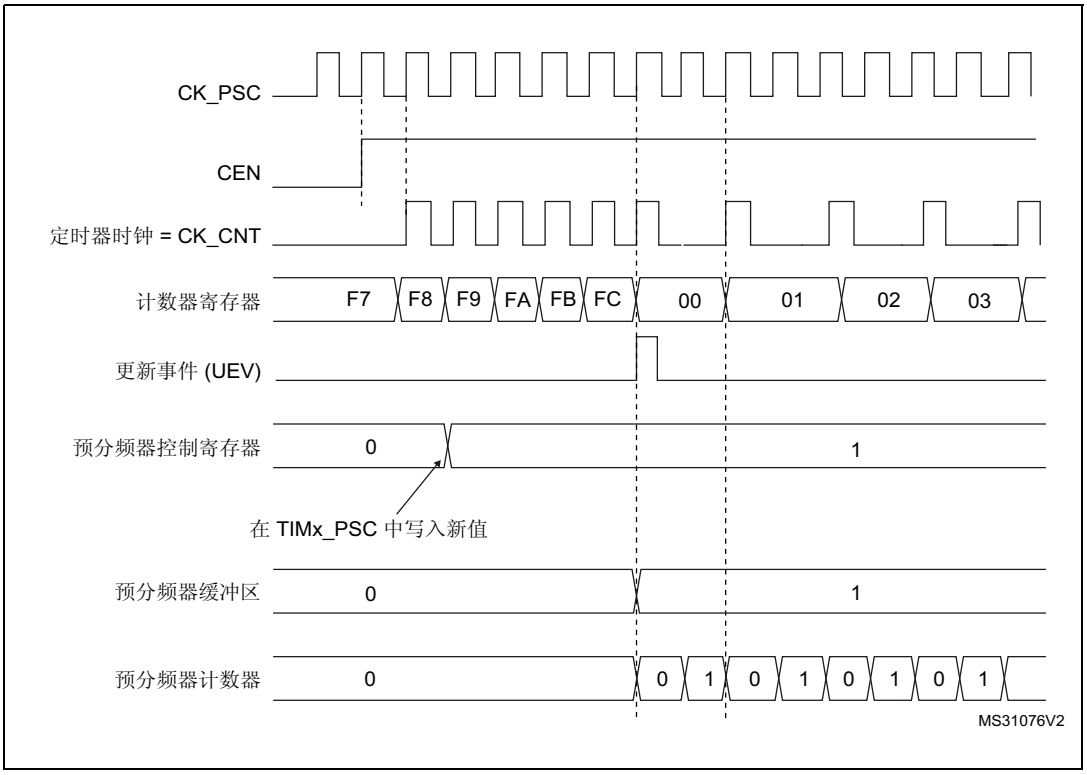
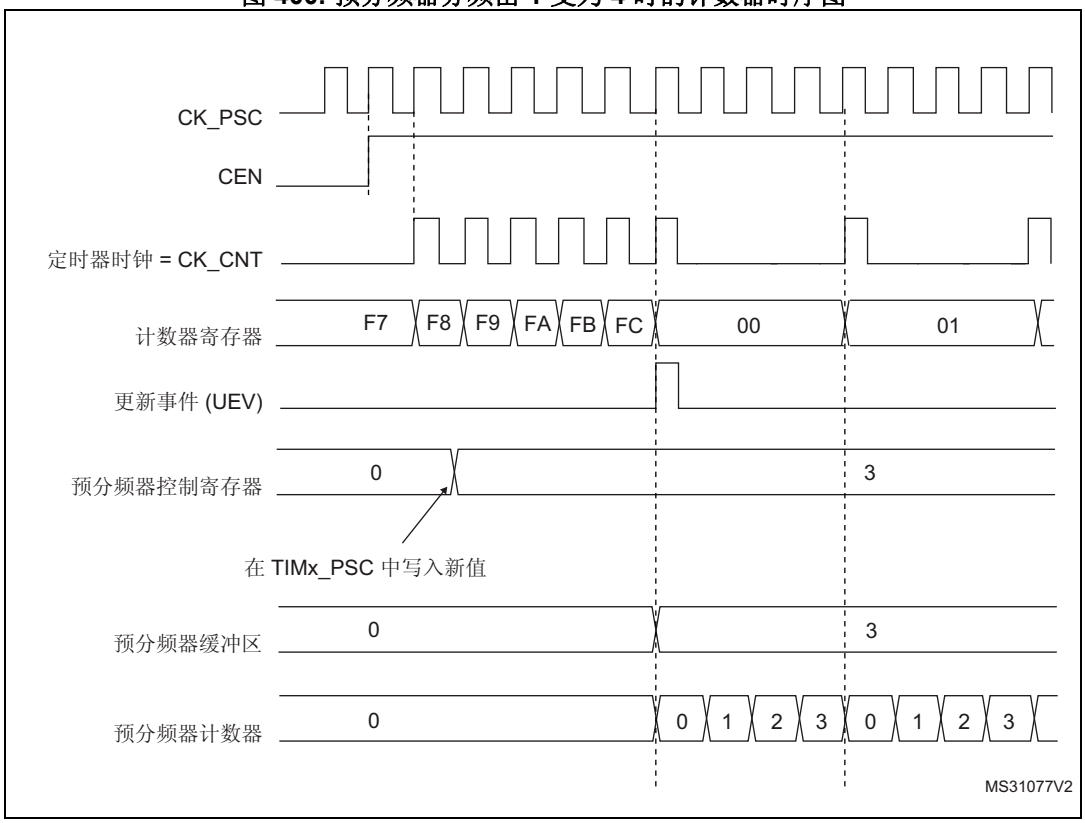


图 400. 预分频器分频由 1 变为 4 时的计数器时序图



39.3.2 计数器模式

递增计数模式

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

每次发生计数器上溢时会生成更新事件，或将 TIMx_EGR 寄存器中的 UG 位置 1（通过软件或使用从模式控制器）也可以生成更新事件。

通过软件将 TIMx_CR1 寄存器中的 UDIS 位置 1 可禁止 UEV 事件。这可避免向预装载寄存器写入新值时更新影子寄存器。在 UDIS 位写入 0 之前不会产生任何更新事件。不过，计数器和预分频器计数器都会重新从 0 开始计数（而预分频比保持不变）。此外，如果 TIMx_CR1 寄存器中的 URS 位（更新请求选择）已置 1，则将 UG 位置 1 会生成更新事件 UEV，但不会将 UIF 标志置 1（因此，不会发送任何中断或 DMA 请求）。这样一来，如果在发生捕获事件时将计数器清零，将不会同时产生更新中断和捕获中断。

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

- 预分频器的缓冲区中将重新装载预装载值（TIMx_PSC 寄存器的内容）
- 使用预装载值 (TIMx_ARR) 更新自动重载影子寄存器

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

图 401. 计数器时序图，1 分频内部时钟

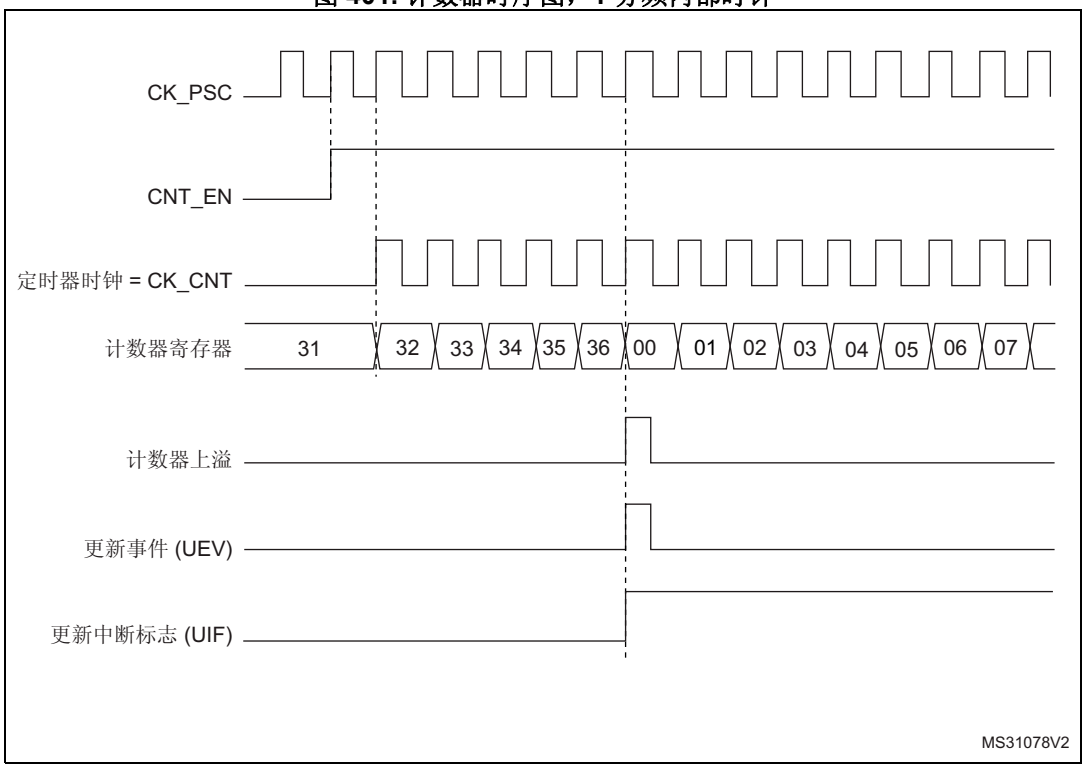


图 402. 计数器时序图，2 分频内部时钟

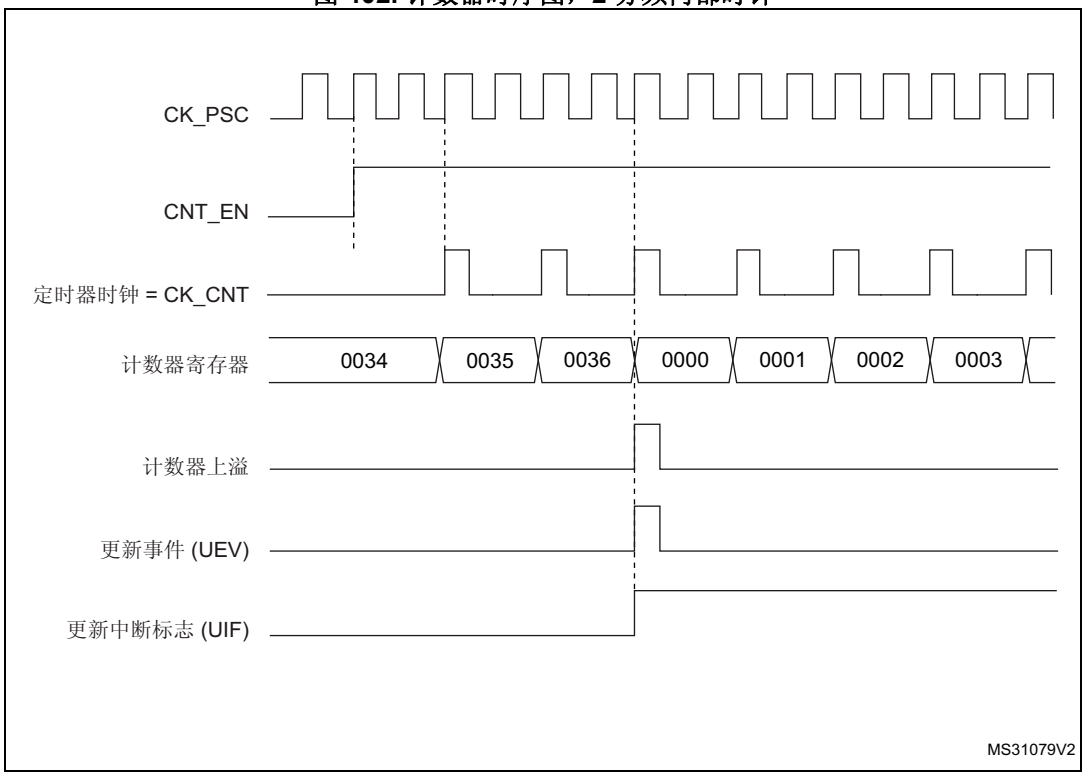


图 403. 计数器时序图，4 分频内部时钟

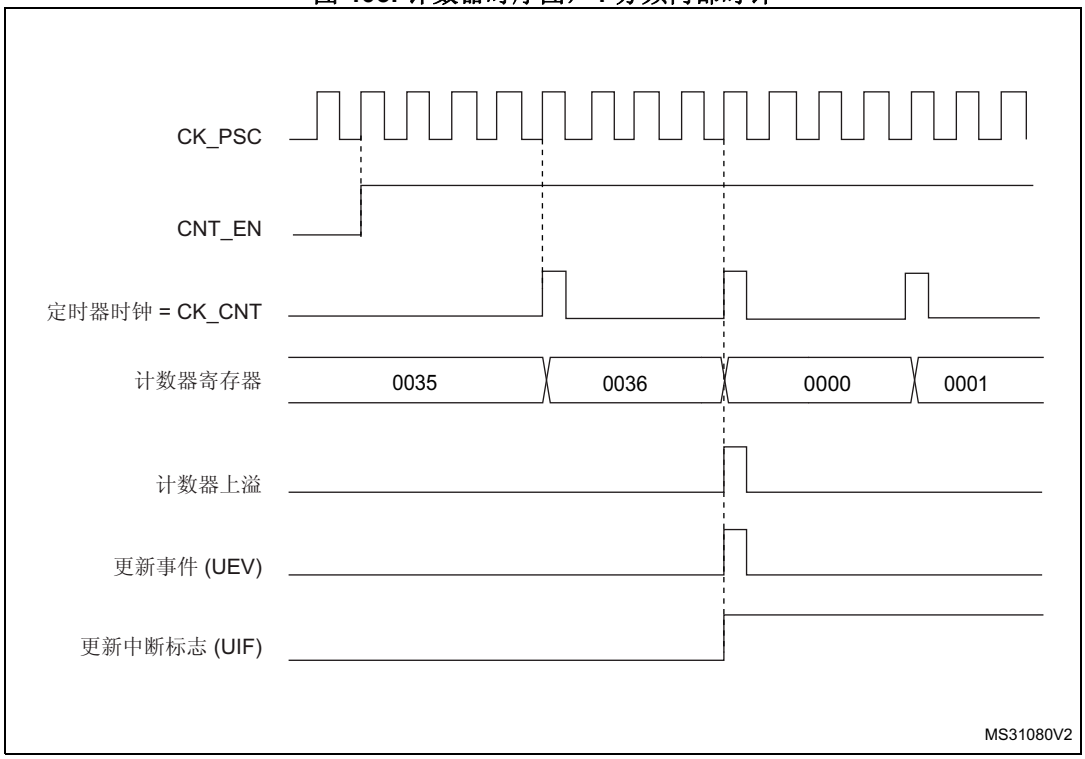


图 404. 计数器时序图，N 分频内部时钟

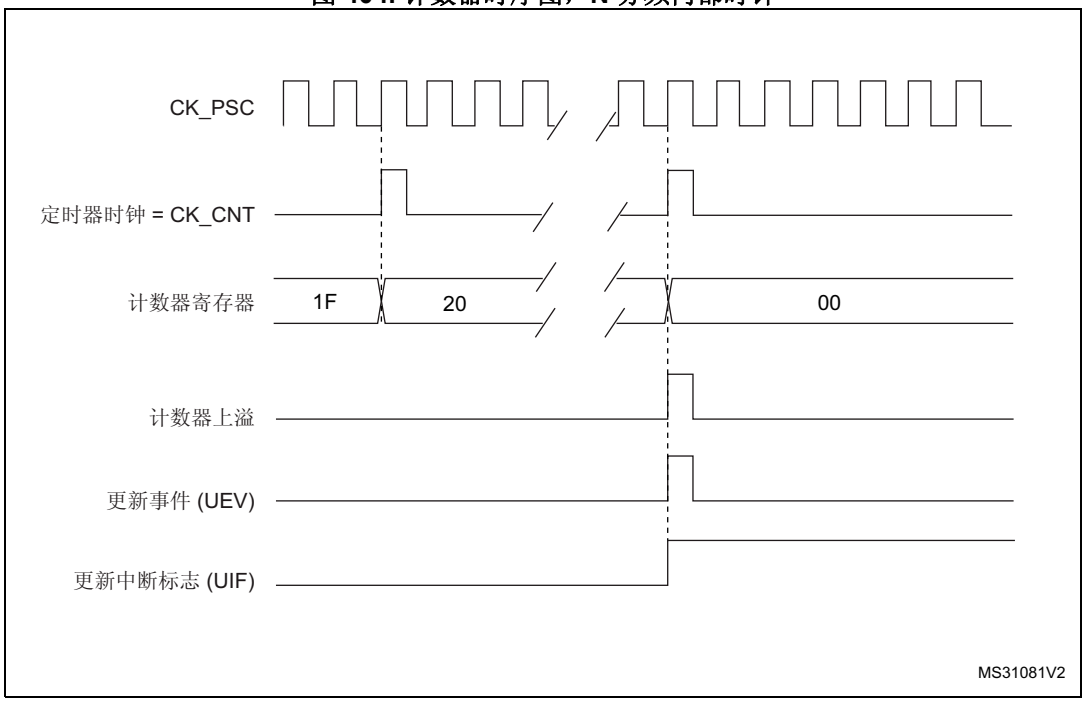


图 405. 计数器时序图，ARPE=0 时更新事件（TIMx_ARR 未预装载）

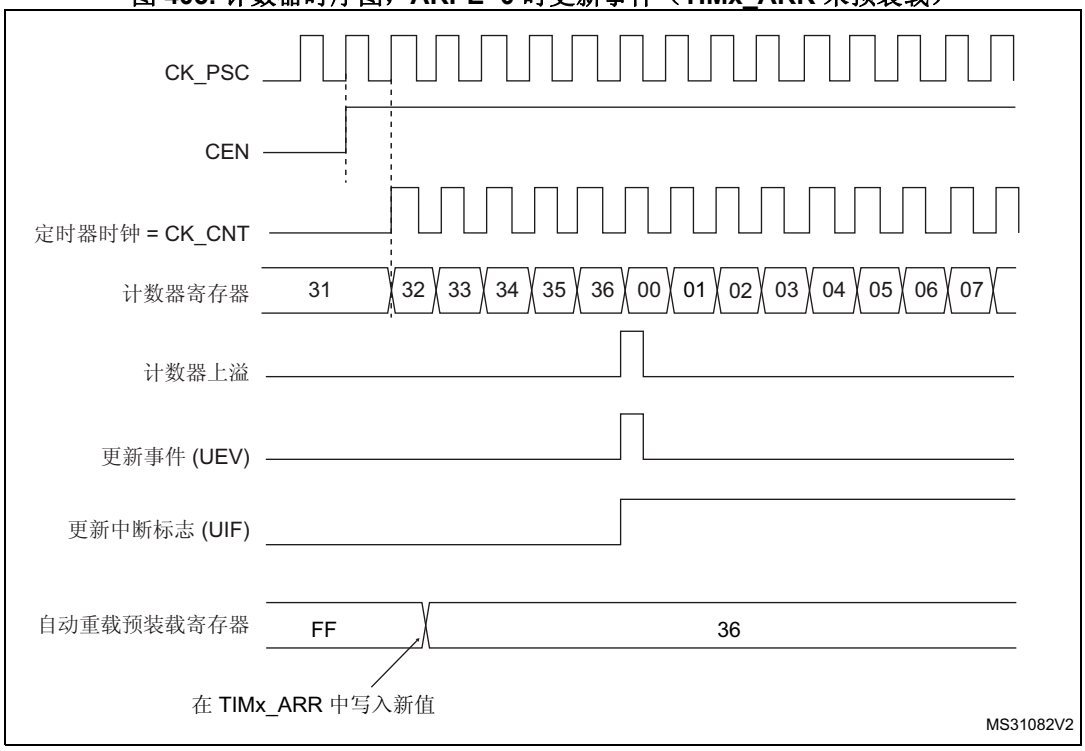
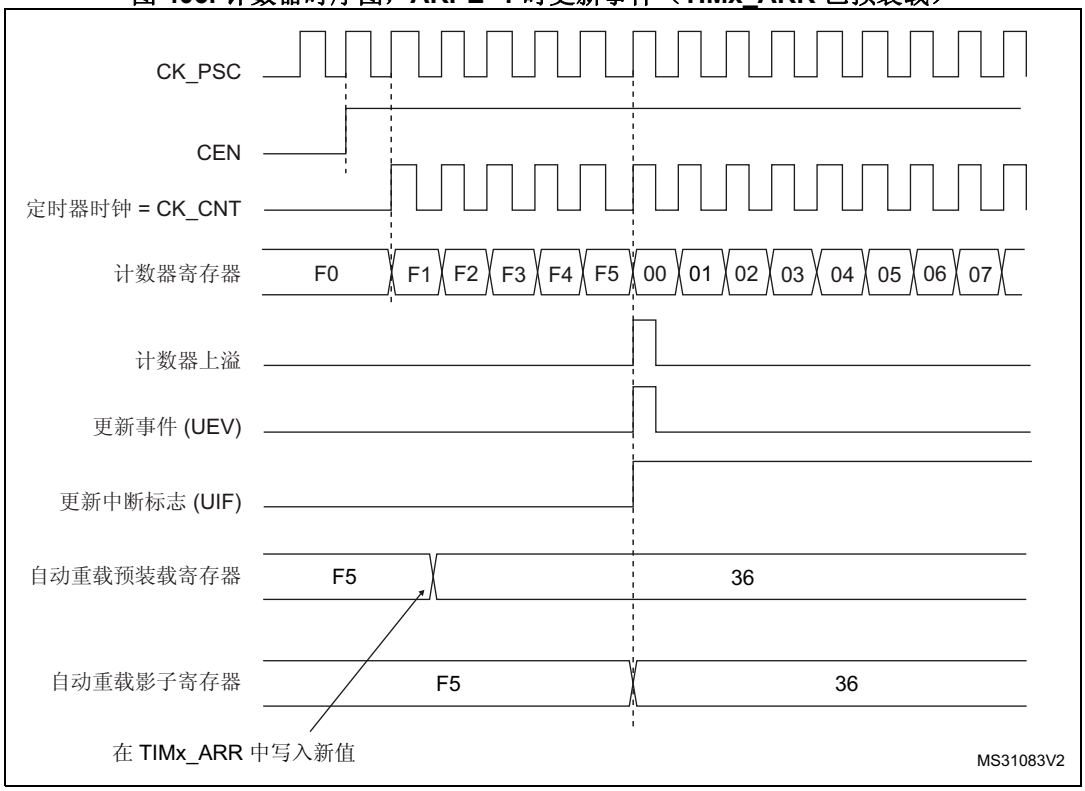


图 406. 计数器时序图，ARPE=1 时更新事件（TIMx_ARR 已预装载）



递减计数模式

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

每次发生计数器下溢时会生成更新事件，或将 TIMx_EGR 寄存器中的 UG 位置 1（通过软件或使用从模式控制器）也可以生成更新事件

通过软件将 TIMx_CR1 寄存器中的 UDIS 位置 1 可禁止 UEV 更新事件。这可避免向预装载寄存器写入新值时更新影子寄存器。在 UDIS 位写入 0 之前不会产生任何更新事件。不过，计数器会重新从当前自动重载值开始计数，而预分频器计数器则重新从 0 开始计数（但预分频比保持不变）。

此外，如果 TIMx_CR1 寄存器中的 URS 位（更新请求选择）已置 1，则将 UG 位置 1 会生成更新事件 UEV，但不会将 UIF 标志置 1（因此，不会发送任何中断或 DMA 请求）。这样一来，如果在发生捕获事件时将计数器清零，将不会同时产生更新中断和捕获中断。

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

- 预分频器的缓冲区中将重新装载预装载值（TIMx_PSC 寄存器的内容）。
- 自动重载有效寄存器将以预装载值（TIMx_ARR 寄存器的内容）进行更新。注意，ARR 寄存器更新在计数器重载之前被更新，因此下一个周期就是预期的值。

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

图 407. 计数器时序图，1 分频内部时钟

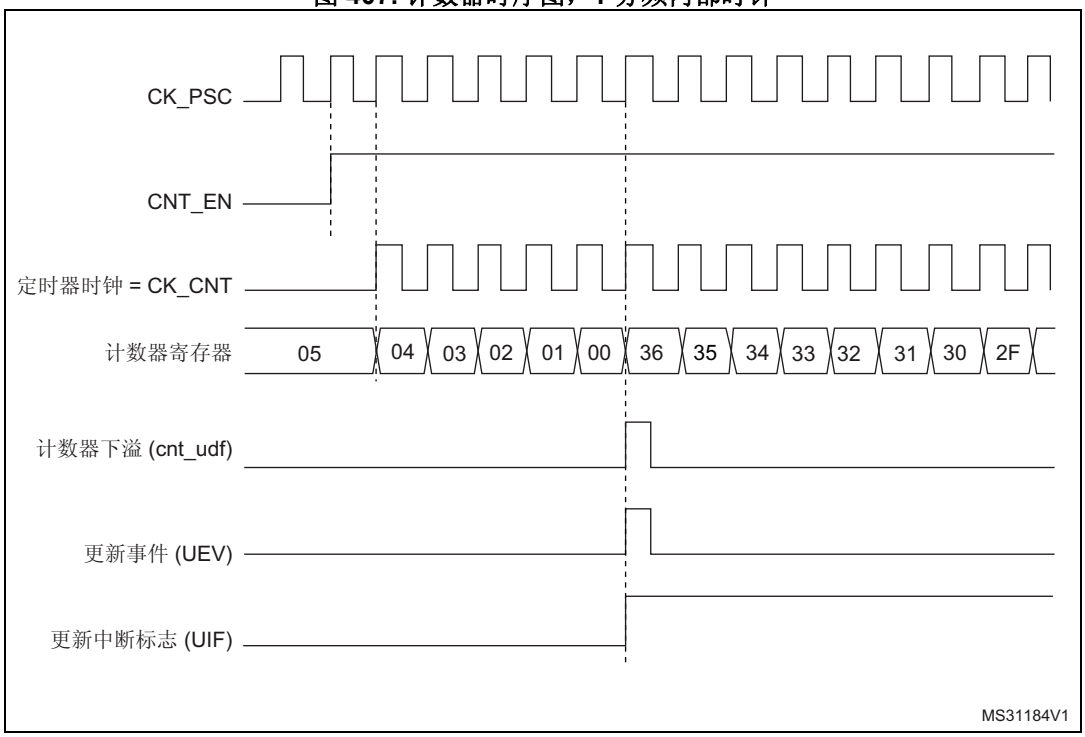


图 408. 计数器时序图，2 分频内部时钟

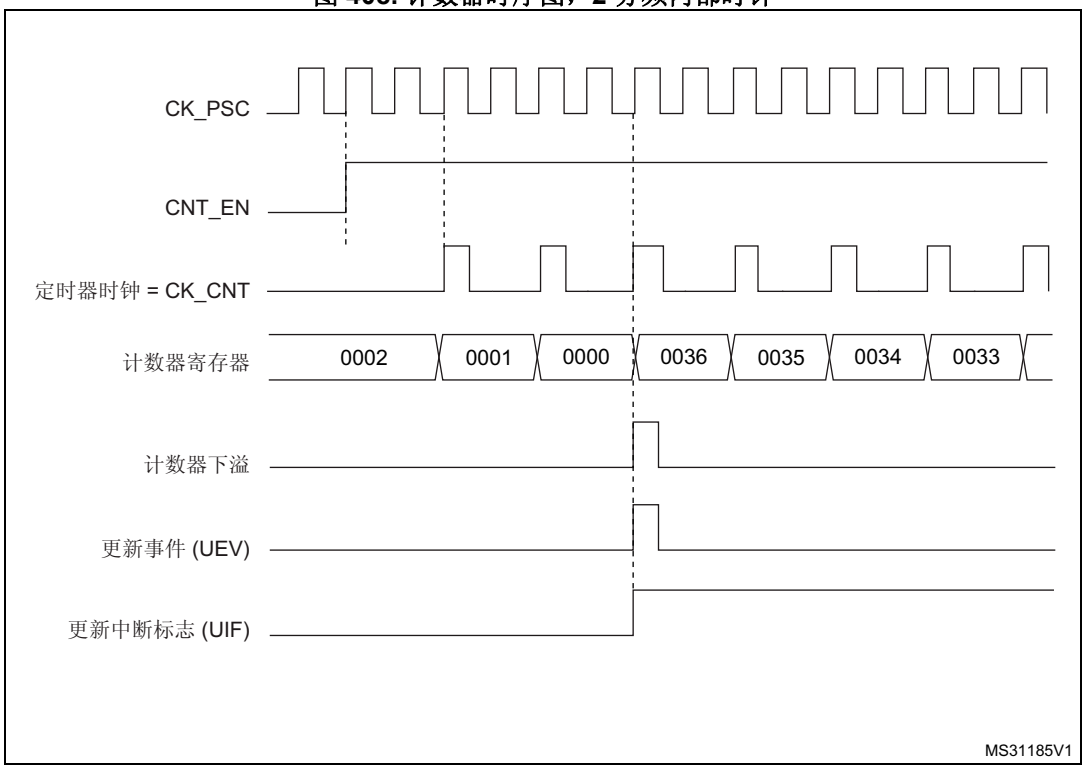


图 409. 计数器时序图，4 分频内部时钟

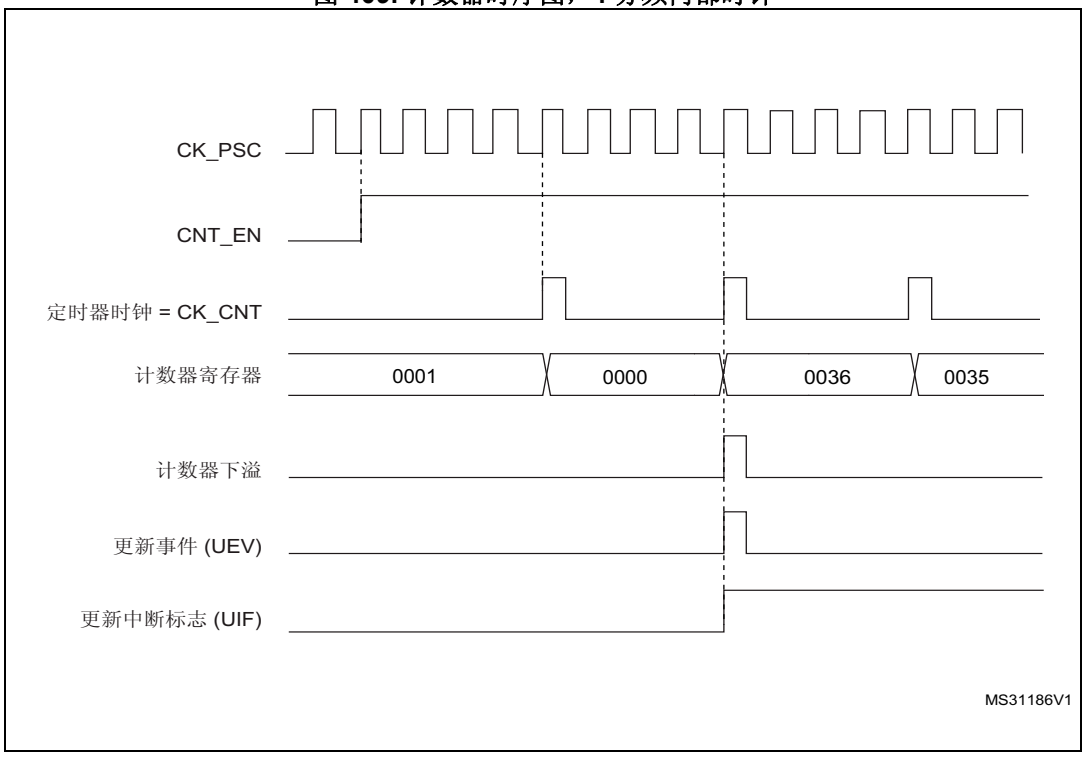


图 410. 计数器时序图，N 分频内部时钟

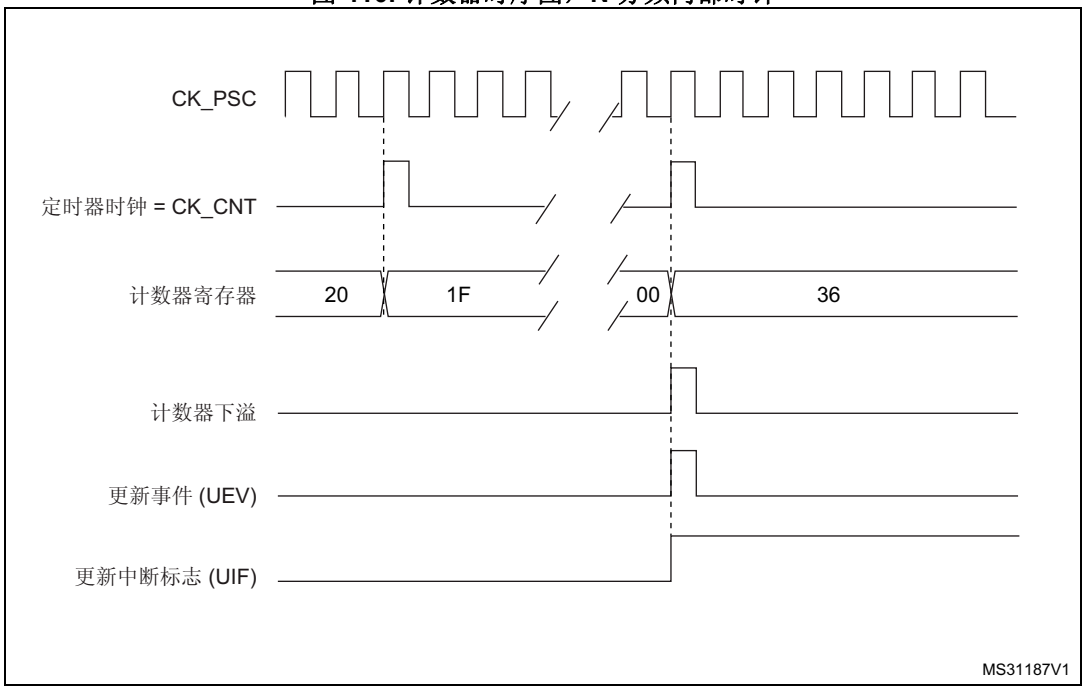
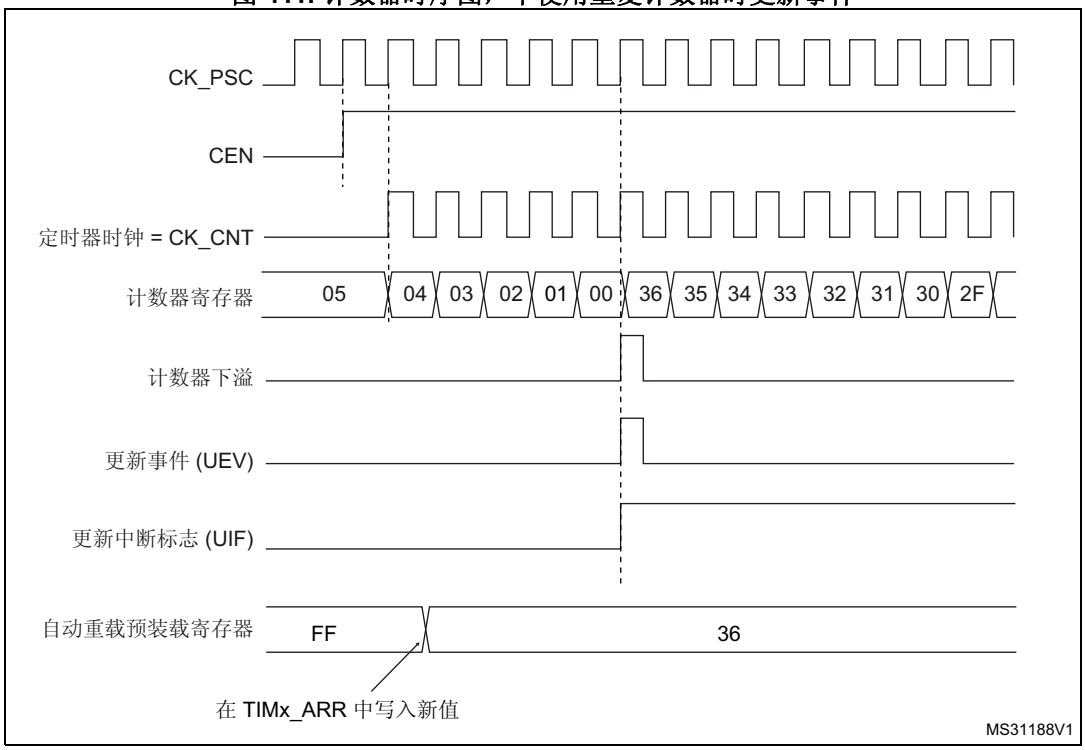


图 411. 计数器时序图，不使用重复计数器时更新事件



中心对齐模式（递增/递减计数）

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

当 TIMx_CR1 寄存器中的 CMS 位不为“00”时，中心对齐模式有效。将通道配置为输出模式时，其输出比较中断标志将在以下模式下置 1，即：计数器递减计数（中心对齐模式 1，CMS = “01”）、计数器递增计数（中心对齐模式 2，CMS = “10”）以及计数器递增/递减计数（中心对齐模式 3，CMS = “11”）。

此模式下无法写入方向位（TIMx_CR1 寄存器中的 DIR 位）。而是由硬件更新并指示当前计数器方向。

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

通过软件将 TIMx_CR1 寄存器中的 UDIS 位置 1 可禁止 UEV 更新事件。这可避免向预装载寄存器写入新值时更新影子寄存器。在 UDIS 位写入 0 之前不会产生任何更新事件。不过，计数器仍会根据当前自动重载值进行递增和递减计数。

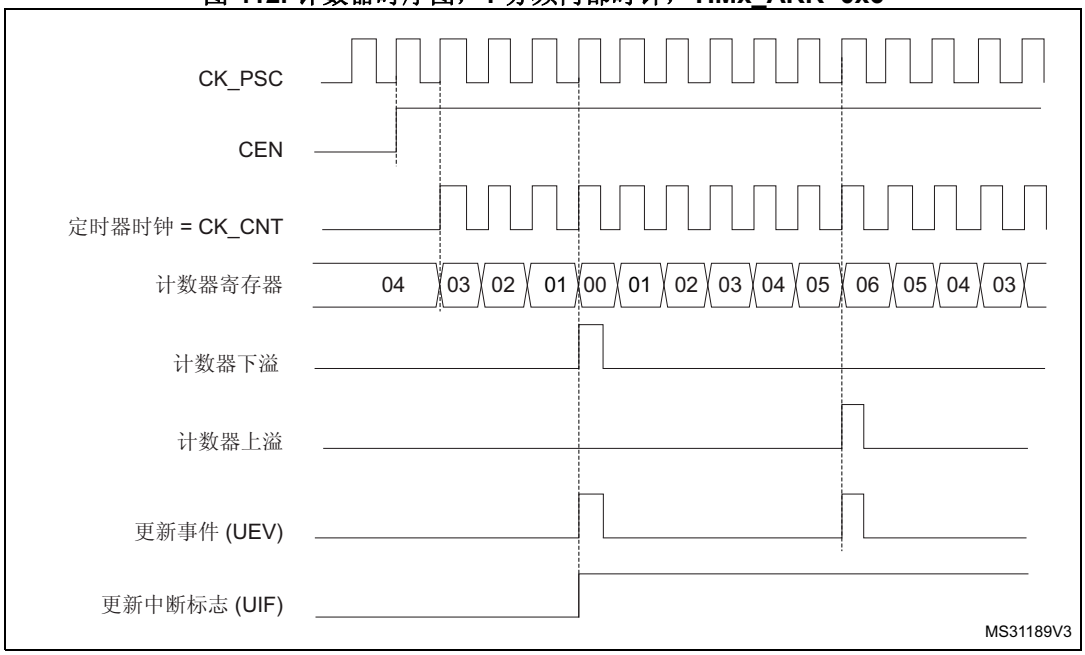
此外，如果 TIMx_CR1 寄存器中的 URS 位（更新请求选择）已置 1，则将 UG 位置 1 会生成更新事件 UEV，但不会将 UIF 标志置 1（因此，不会发送任何中断或 DMA 请求）。这样一来，如果在发生捕获事件时将计数器清零，将不会同时产生更新中断和捕获中断。

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

- 预分频器的缓冲区中将重新装载预装载值（TIMx_PSC 寄存器的内容）。
- 自动重载有效寄存器将以预装载值（TIMx_ARR 寄存器的内容）进行更新。注意，如果更新操作是由计数器上溢触发的，则 ARR 寄存器在计数器重载之前更新，因此，下一个计数周期就是我们所希望的新的周期长度（计数器被重载新的值）。

以下各图以一些示例说明不同时钟频率下计数器的行为。

图 412. 计数器时序图，1 分频内部时钟，TIMx_ARR=0x6



1. 此处使用中心对齐模式 1（有关详细信息，请参见*第 1533 页的第 39.4.1 节：TIMx 控制寄存器 1 (TIMx_CR1)*）。

图 413. 计数器时序图，2 分频内部时钟

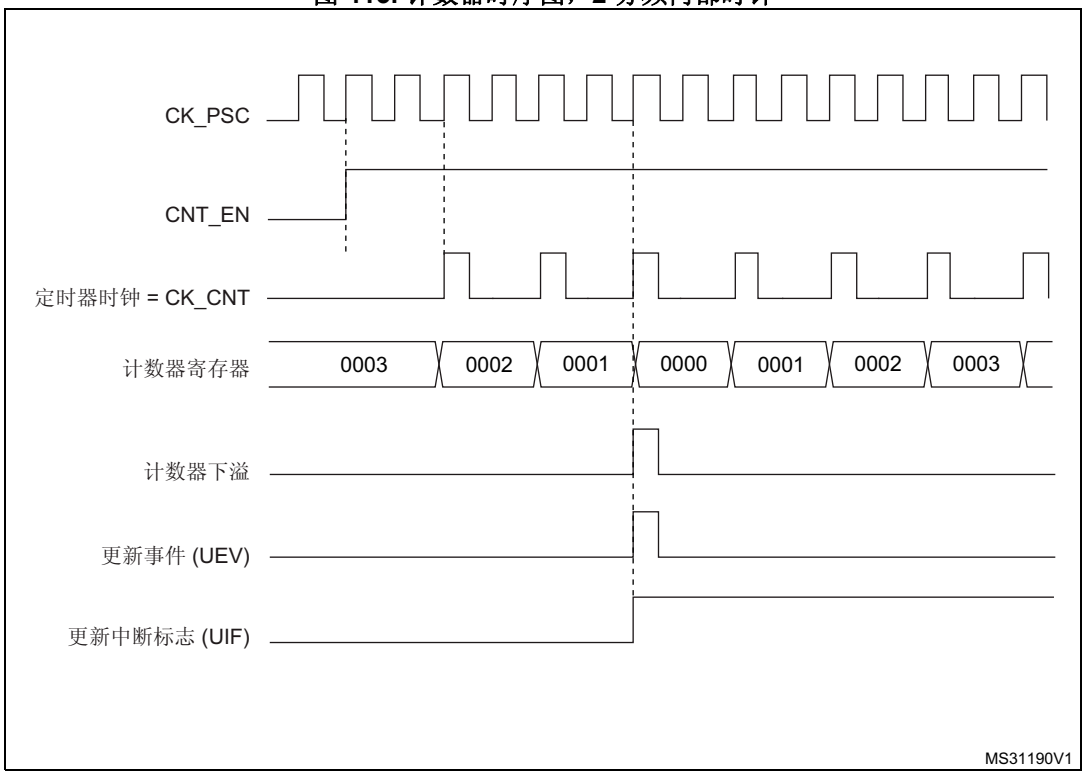
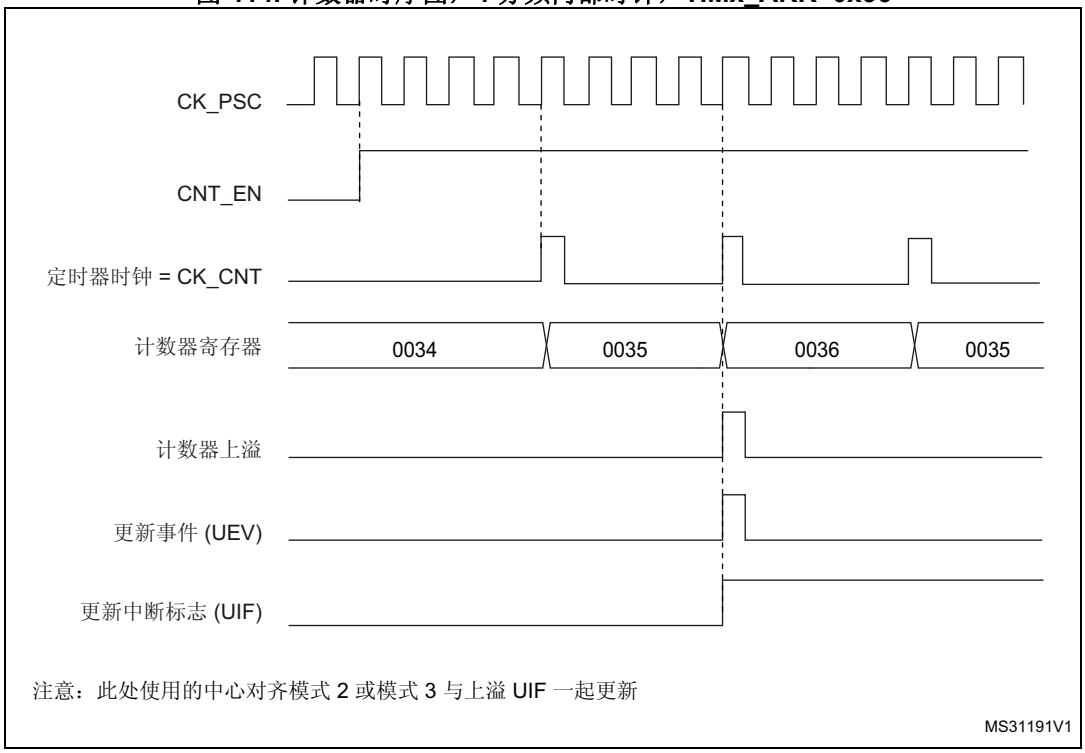


图 414. 计数器时序图，4 分频内部时钟，TIMx_ARR=0x36



1. 中心对齐模式 2 或模式 3 与上溢 UIF 结合使用。

图 415. 计数器时序图，N 分频内部时钟

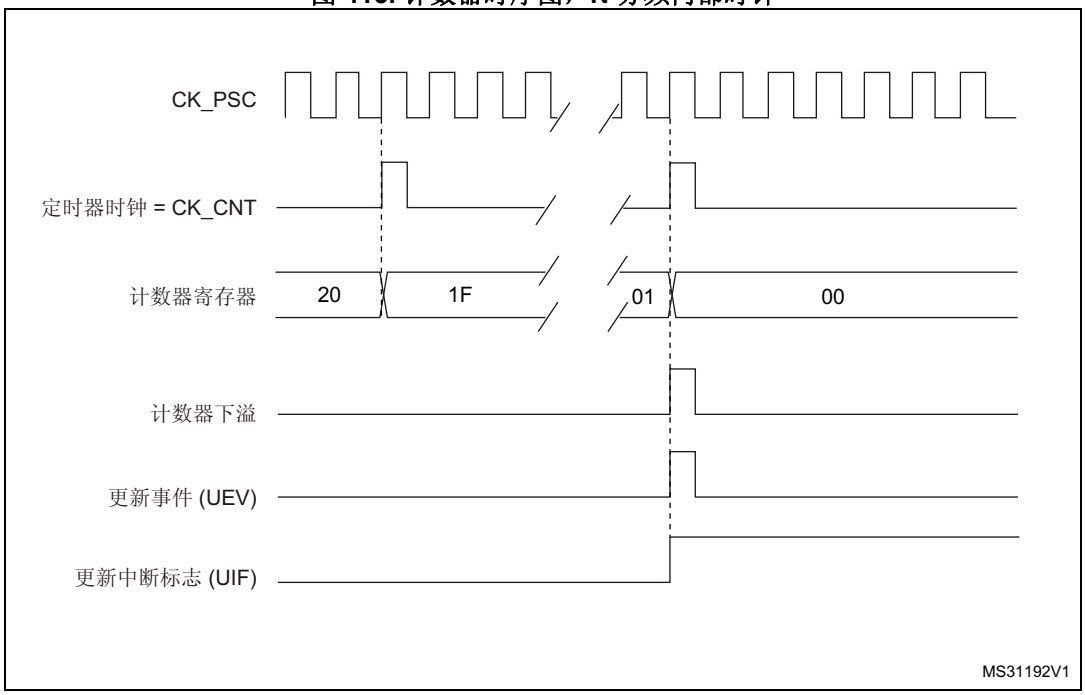


图 416. 计数器时序图，ARPE=1 时的更新事件（计数器下溢）

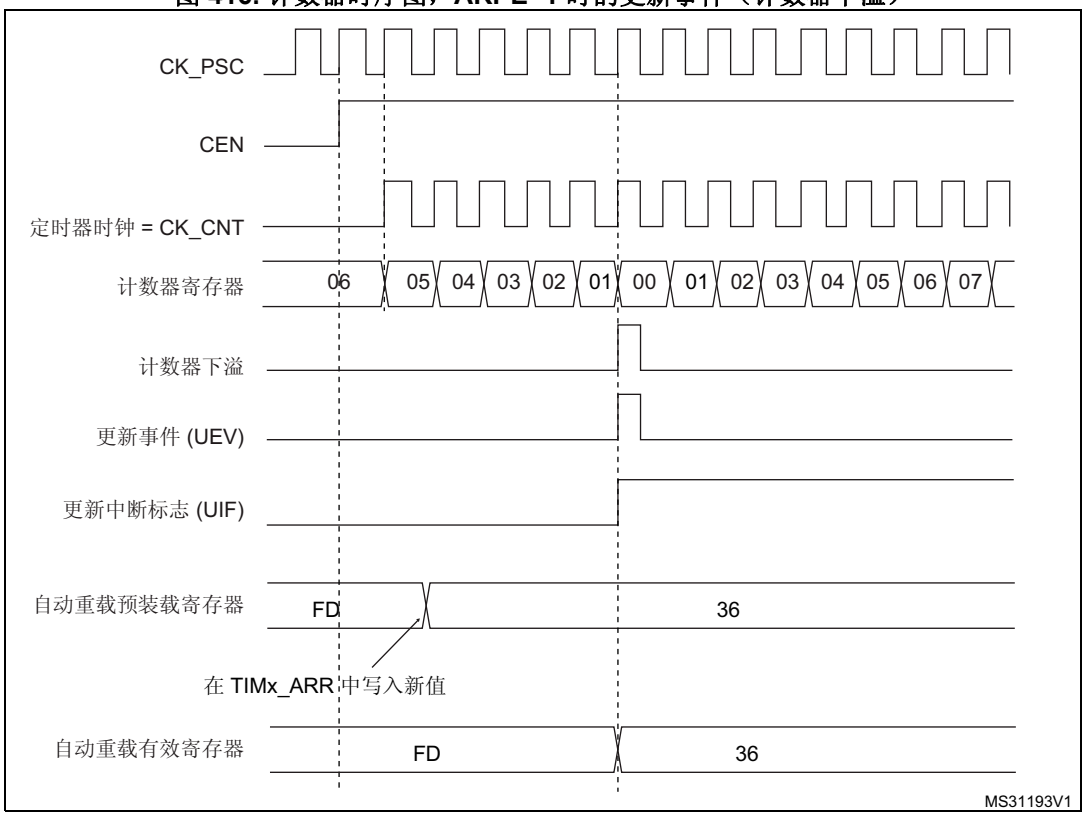
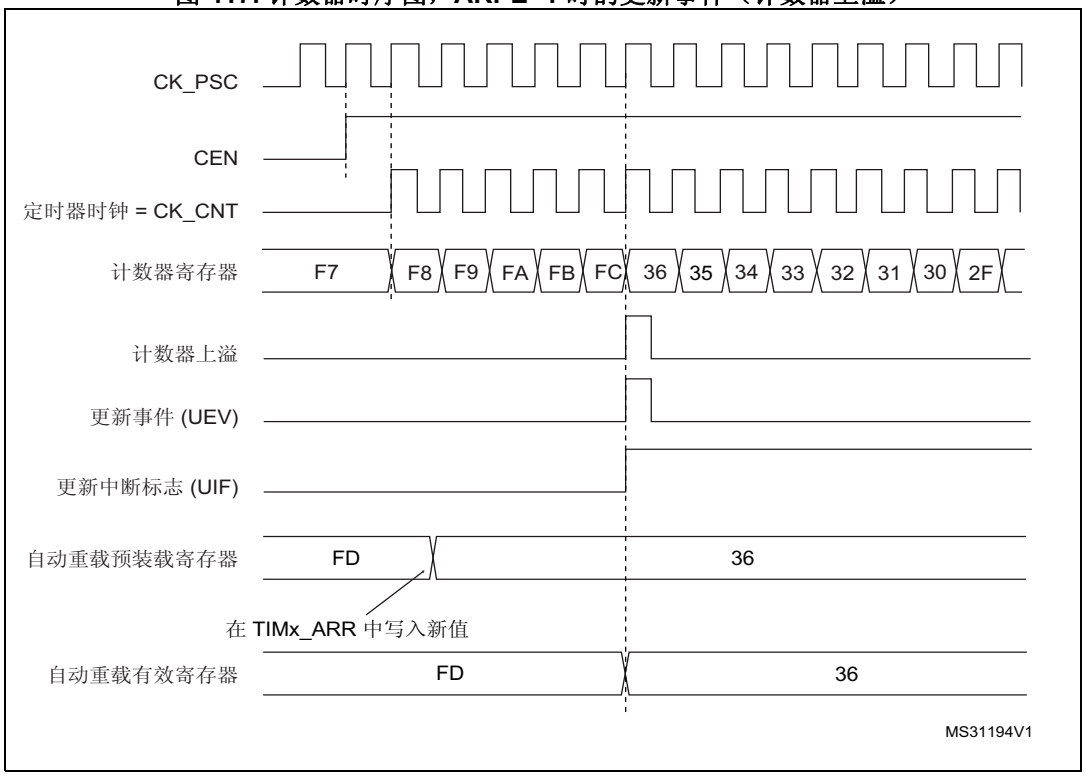


图 417. 计数器时序图，ARPE=1 时的更新事件（计数器上溢）



39.3.3 时钟选择

计数器时钟可由下列时钟源提供：

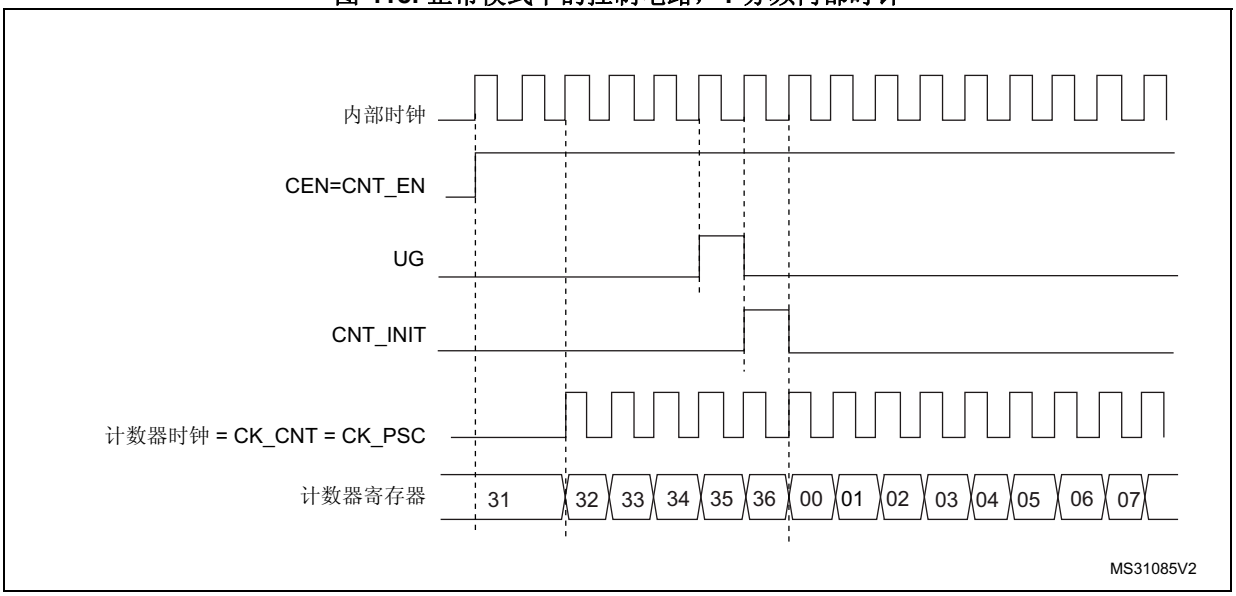
- 内部时钟 (CK_INT)
- 外部时钟模式 1：外部输入引脚 (TIx)
- 外部时钟模式 2：外部触发输入 (ETR)
- 外部触发输入 (ITRx)：使用一个定时器作为另一定时器的预分频器，例如，可将定时器 13 配置为定时器 2 的预分频器。更多详细信息，请参见*第 1527 页的将一个定时器用作另一个定时器的预分频器*。

内部时钟源 (CK_INT)

如果禁止从模式控制器（TIMx_SMCR 寄存器中 SMS=000），则 CEN 位、DIR 位（TIMx_CR1 寄存器中）和 UG 位（TIMx_EGR 寄存器中）为实际控制位，并且只能通过软件进行更改（UG 除外，仍自动清零）。当对 CEN 位写入 1 时，预分频器的时钟就由内部时钟 CK_INT 提供。

图 418 显示了正常模式下控制电路与递增计数器的行为（没有预分频的情况下）。

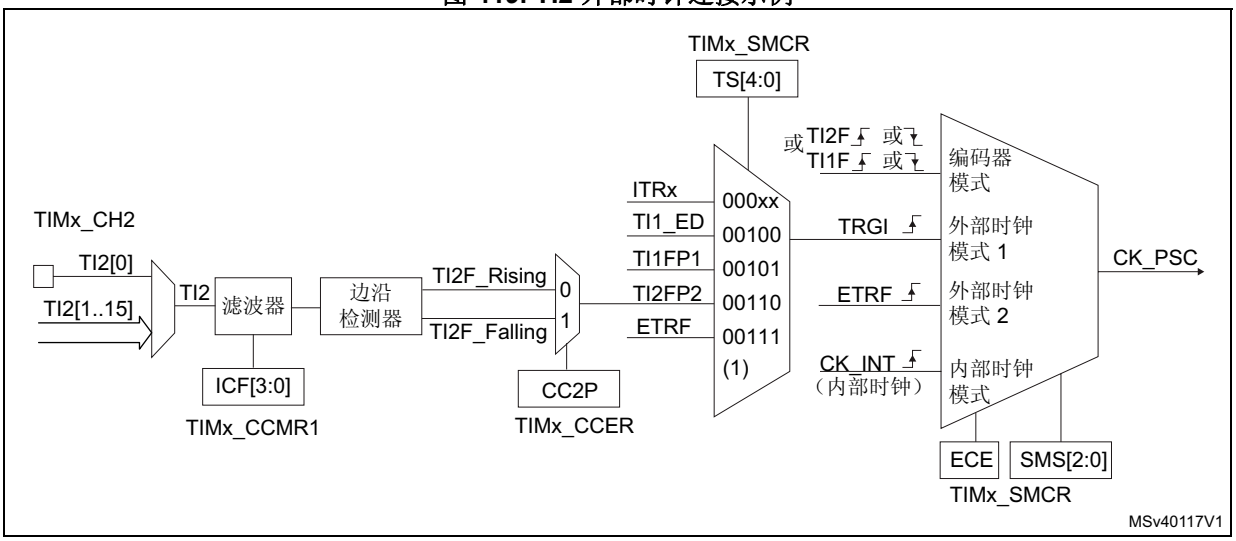
图 418. 正常模式下的控制电路，1 分频内部时钟



外部时钟源模式 1

当 TIMx_SMCR 寄存器中的 SMS=111 时，可选择此模式。计数器可在选定的输入信号上出现上升沿或下降沿时计数。

图 419. TI2 外部时钟连接示例



1. 保留从 01000 到 11111 的代码：ITRy。

例如，要使递增计数器在 TI2 输入出现上升沿时计数，请执行以下步骤：

1. 使用 TIMx_TISEL 寄存器中的 TI2SEL[3:0] 位选择正确的 TI1x 源（内部或外部）。
2. 通过在 TIMx_CCMR1 寄存器中写入 CC2S=“01”来配置通道 2，使其能够检测 TI2 输入的上升沿。
3. 通过在 TIMx_CCMR1 寄存器中写入 IC2F[3:0] 位来配置输入滤波带宽（如果不需要任何滤波器，请保持 IC2F=0000）。

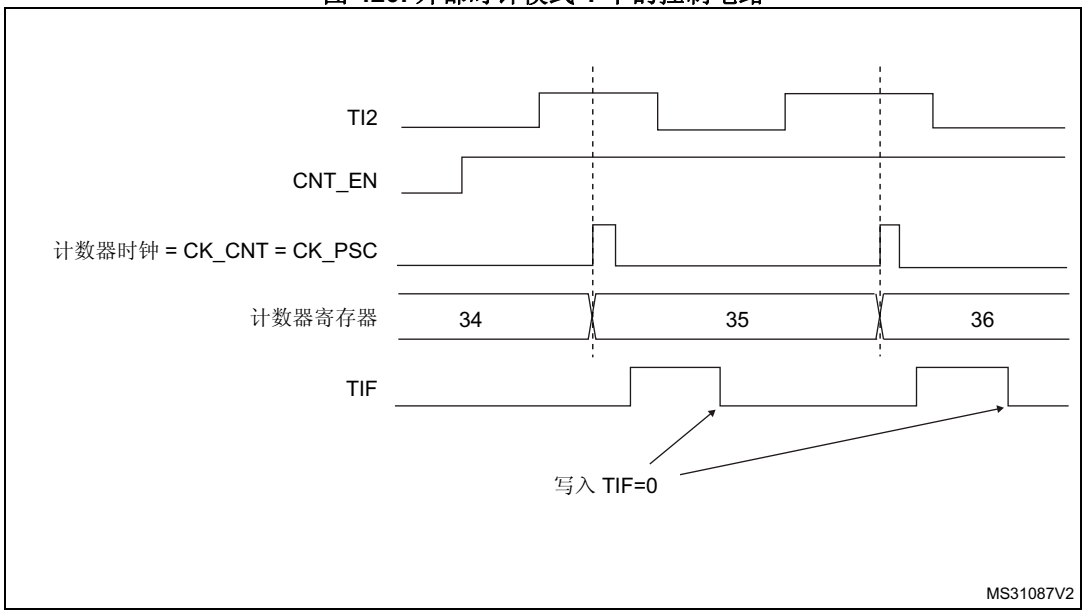
注： *由于捕获预分频器不用于触发操作，因此无需对其进行配置。*

4. 通过在 TIMx_CCER 寄存器中写入 CC2P=0 和 CC2NP=0 来选择上升沿极性。
5. 通过在 TIMx_SMCR 寄存器中写入 SMS=111，使定时器在外部时钟模式 1 下工作。
6. 通过在 TIMx_SMCR 寄存器中写入 TS=00110 来选择 TI2 作为输入源。
7. 通过在 TIMx_CR1 寄存器中写入 CEN=1 来使能计数器。

当 TI2 出现上升沿时，计数器便会计数一次并且 TIF 标志置 1。

TI2 的上升沿与实际计数器时钟之间的延迟是由于 TI2 输入的重新同步电路引起的。

图 420. 外部时钟模式 1 下的控制电路



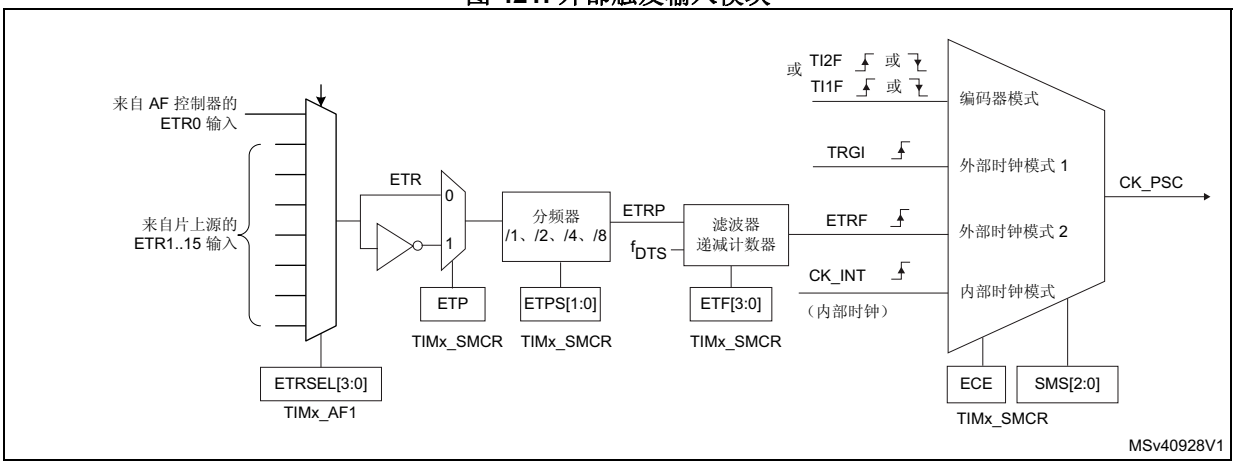
外部时钟源模式 2

通过在 TIMx_SMCR 寄存器中写入 ECE=1 可选择此模式。

计数器可在外部触发输入 ETR 出现上升沿或下降沿时计数。

图 421 简要介绍了外部触发输入模块。

图 421. 外部触发输入模块



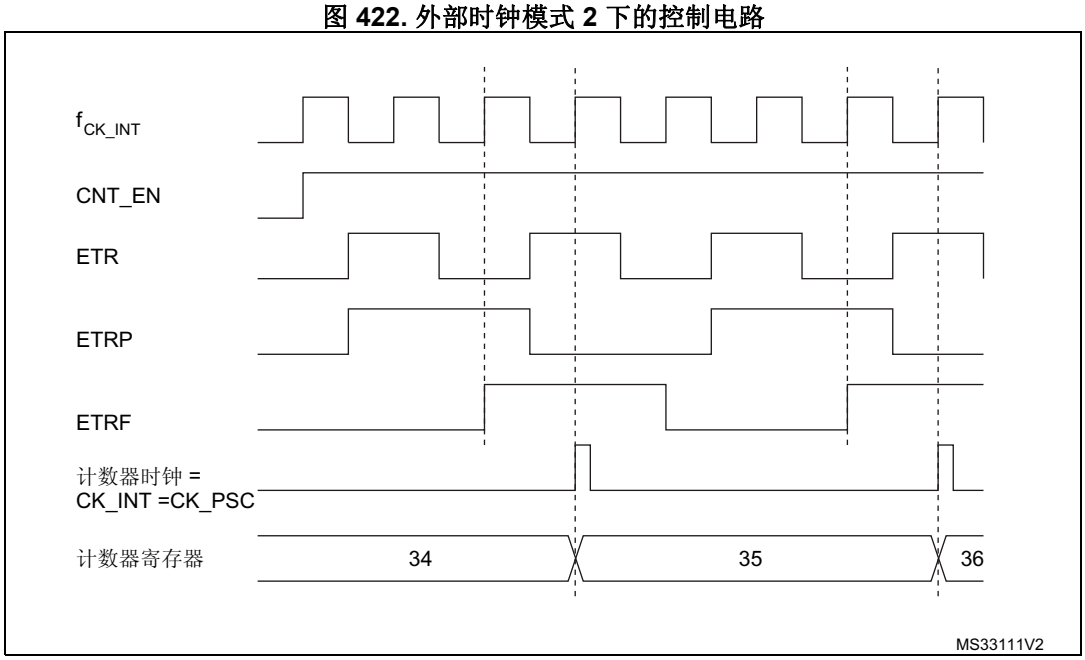
例如，要使递增计数器在 ETR 每出现 2 个上升沿时计数，请执行以下步骤：

1. 使用 TIMx_AF1 寄存器中的 ETRSEL[3:0] 位选择正确的 ETR 源（内部或外部）。
2. 由于此例中不需滤波器，因此在 TIMx_SMCR 寄存器中写入 ETF[3:0]=0000。
3. 通过在 TIMx_SMCR 寄存器中写入 ETPS[1:0]=01 来设置预分频器。
4. 通过在 TIMx_SMCR 寄存器中写入 ETP=0 来选择 ETR 引脚的上升沿检测。
5. 通过在 TIMx_SMCR 寄存器中写入 ECE=1 来使能外部时钟模式 2。
6. 通过在 TIMx_CR1 寄存器中写入 CEN=1 来使能计数器。

ETR 每出现 2 个上升沿，计数器计数一次。

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

图 422. 外部时钟模式 2 下的控制电路



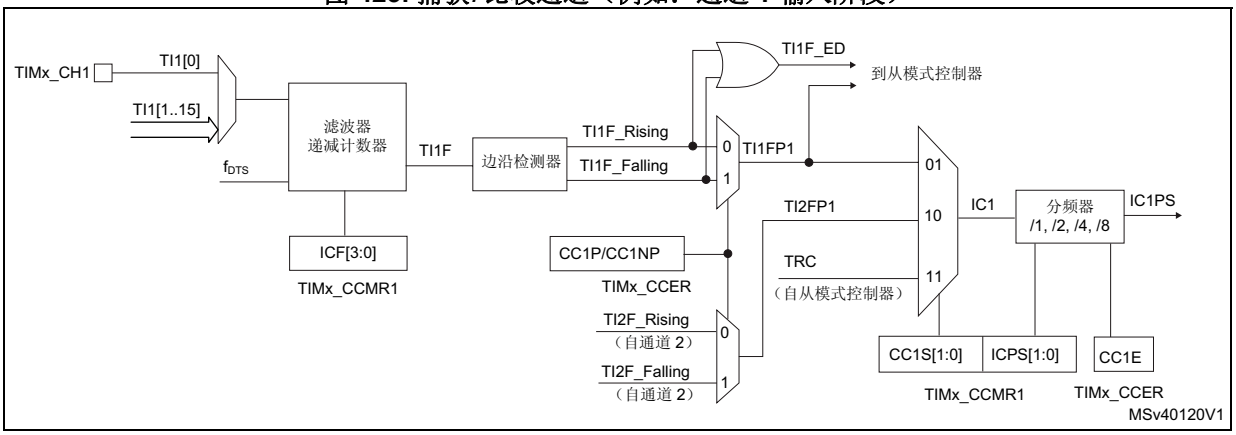
39.3.4 捕获/比较通道

每个捕获/比较通道均围绕一个捕获/比较寄存器（包括一个影子寄存器）、一个捕获输入阶段（数字滤波、多路复用和预分频器）和一个输出阶段（比较器和输出控制）构建而成。

下图简要介绍了一路捕获/比较通道。

输入阶段对相应的 TIx 输入进行采样，生成一个滤波后的信号 TIxF。然后，带有极性选择功能的边沿检测器生成一个信号 (TIxFPx)，该信号可用作从模式控制器的触发输入，也可用作捕获命令。该信号先进行预分频 (ICxPS)，而后再进入捕获寄存器。

图 423. 捕获/比较通道（例如：通道 1 输入阶段）



输出阶段生成一个中间波形作为基准：OCxRef（高电平有效）。链的末端决定最终输出信号的极性。

图 424. 捕获/比较通道 1 主电路

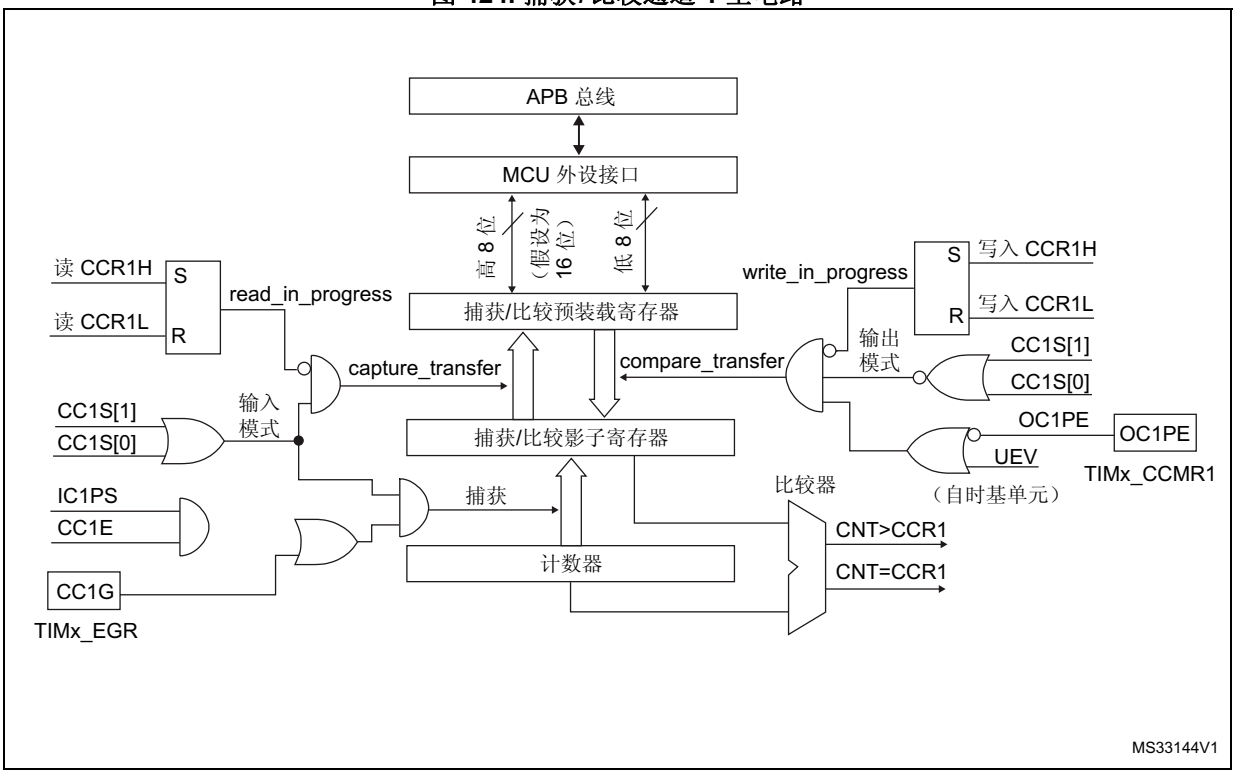
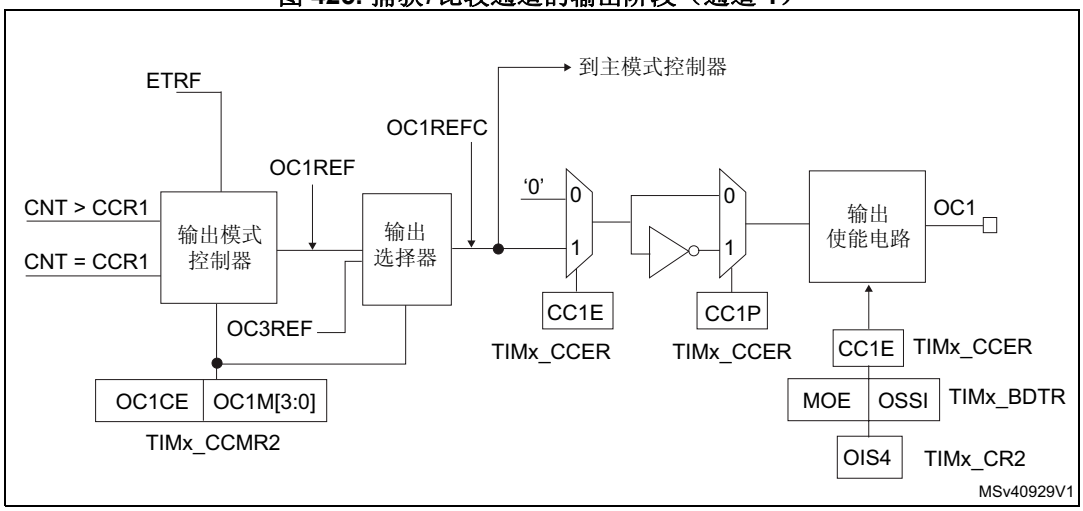


图 425. 捕获/比较通道的输出阶段（通道 1）



捕获/比较模块由一个预装载寄存器和一个影子寄存器组成。始终可通过读写操作访问预装载寄存器。

在捕获模式下，捕获实际发生在影子寄存器中，然后将影子寄存器的内容复制到预装载寄存器中。

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

39.3.5 输入捕获模式

在输入捕获模式下，当相应的 ICx 信号检测到跳变沿后，将使用捕获/比较寄存器 (TIMx_CCRx) 来锁存计数器的值。发生捕获事件时，会将相应的 CCXIF 标志（TIMx_SR 寄存器）置 1，并可发送中断或 DMA 请求（如果已使能）。如果发生捕获事件时 CCxIF 标志已处于高位，则会将重复捕获标志 CCxOF（TIMx_SR 寄存器）置 1。可通过软件方法向 CCxIF 写入 0 来给 CCxIF 清零，或读取存储在 TIMx_CCRx 寄存器中的已捕获数据。向 CCxOF 写入 0 后会将其清零。

以下示例说明了如何在 TI1 输入出现上升沿时将计数器的值捕获到 TIMx_CCR1 中。具体操作步骤如下：

1. 使用 TIMx_TISEL 寄存器中的 TI1SEL[3:0] 位选择正确的 TI1x 源（内部或外部）。
2. 选择有效输入：TIMx_CCR1 必须连接到 TI1 输入，因此向 TIMx_CCMR1 寄存器中的 CC1S 位写入 01。只要 CC1S 不等于 00，就会将通道配置为输入模式，并且 TIMx_CCR1 寄存器将处于只读状态。
3. 根据连接到定时器的信号，对所需的输入滤波带宽进行编程（如果输入为 TIx 之一，则对 TIMx_CCMRx 寄存器中的 ICxF 位进行编程）。假设信号边沿变化时，输入信号最多在 5 个内部时钟周期内发生抖动。因此，我们必须将滤波带宽设置为大于 5 个内部时钟周期。在检测到 8 个具有新电平的连续采样（以 f_{DTS} 频率采样）后，可以确认 TI1 上的跳变沿。然后向 TIMx_CCMR1 寄存器中的 IC1F 位写入 0011。
4. 通过向 TIMx_CCER 寄存器中的 CC1P 位和 CC1NP 位写入 000，选择 TI1 通道的有效跳变沿（本例中为上升沿）。
5. 对输入预分频器进行编程。在本例中，我们希望每次有效转换时都执行捕获操作，因此需要禁止预分频器（向 TIMx_CCMR1 寄存器中的 IC1PS 位写入 00）。
6. 通过将 TIMx_CCER 寄存器中的 CC1E 位置 1，允许将计数器的值捕获到捕获寄存器中。
7. 如果需要，可通过将 TIMx_DIER 寄存器中的 CC1IE 位置 1 来使能相关中断请求，并且/或者通过将该寄存器中的 CC1DE 位置 1 来使能 DMA 请求。

发生输入捕获时：

- 发生有效跳变沿时，TIMx_CCR1 寄存器会获取计数器的值。
- 将 CC1IF 标志置 1（中断标志）。如果至少发生了两次连续捕获，但 CC1IF 标志未被清零，这样 CC1OF 重复捕获标志会被置 1。
- 根据 CC1IE 位生成中断。
- 根据 CC1DE 位生成 DMA 请求。

要处理重复捕获，建议在读出重复捕获标志之前读取数据。这样可避免丢失在读取重复捕获标志之后与读取数据之前可能出现的重复捕获信息。

注： *通过软件将 TIMx_EGR 寄存器中的相应 CCxG 位置 1 可生成 IC 中断和/或 DMA 请求。*

39.3.6 PWM 输入模式

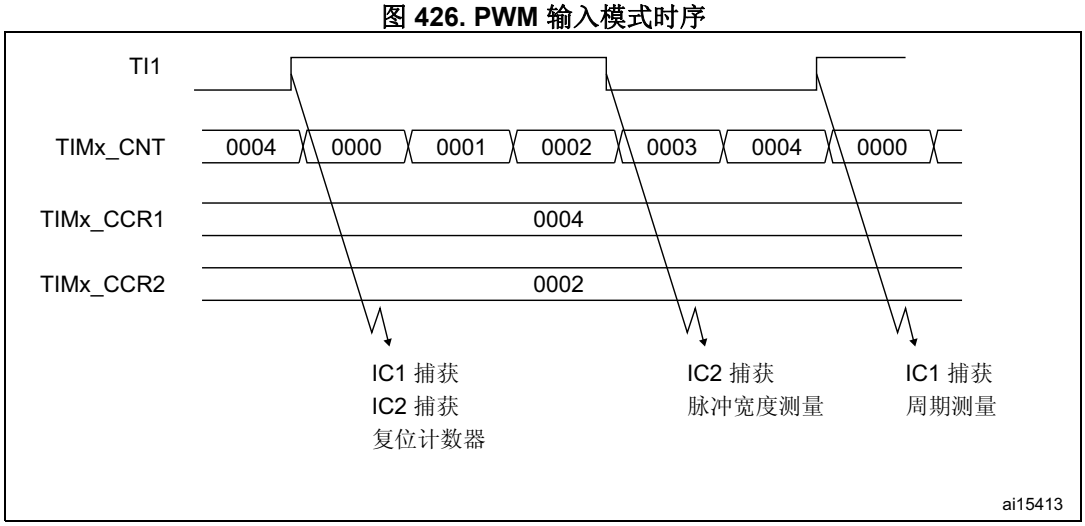
此模式是输入捕获模式的一个特例。其实现步骤与输入捕获模式基本相同，仅存在以下不同之处：

- 两个 ICx 信号被映射至同一个 TIx 输入。
- 这两个 ICx 信号在边沿处有效，但极性相反。
- 选择两个 TIxFP 信号之一作为触发输入，并将从模式控制器配置为复位模式。

例如，可通过以下步骤对应用于 TI1 的 PWM 的周期（位于 TIMx_CCR1 寄存器中）和占空比（位于 TIMx_CCR2 寄存器中）进行测量（取决于 CK_INT 频率和预分频器的值）：

1. 选择 TIMx_CCR1 的有效输入：向 TIMx_CCMR1 寄存器中的 CC1S 位写入 01（选择 TI1）。
2. 选择 TI1FP1 的有效极性（同时用于 TIMx_CCR1 中的捕获和计数器清零）：向 CC1P 位和 CC1NP 位写入“0”（上升沿有效）。
3. 选择 TIMx_CCR2 的有效输入：向 TIMx_CCMR1 寄存器中的 CC2S 写入 10 （选择 TI1）。
4. 选择 TI1FP2 的有效极性（用于 TIMx_CCR2 中的捕获）：向 CC2P 位写入“1”，向 CC2NP 位写入“0”（下降沿有效）。
5. 选择有效触发输入：向 TIMx_SMCR 寄存器中的 TS 位写入 00101（选择 TI1FP1）。
6. 将从模式控制器配置为复位模式：向 TIMx_SMCR 寄存器中的 SMS 位写入 100。
7. 使能捕获：向 TIMx_CCER 寄存器中的 CC1E 位和 CC2E 位写入“1”。

图 426. PWM 输入模式时序



1. PWM 输入模式只能与 TIMx_CH1/TIMx_CH2 信号配合使用，因为只有 TI1FP1 和 TI2FP2 与从模式控制器相连。

39.3.7 强制输出模式

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

要将输出比较信号 (ocxref/OCx) 强制设置为有效电平，只需向相应 TIMx_CCMRx 寄存器中的 OCxM 位写入 101。ocxref 进而强制设置为高电平（OCxREF 始终为高电平有效），同时 OCx 获取 CCxP 极性位的相反值。

例如：CCxP=0（OCx 高电平有效）=> OCx 强制设置为高电平。

通过向 TIMx_CCMRx 寄存器中的 OCxM 位写入 100，可将 ocxref 信号强制设置为低电平。

无论如何，TIMx_CCRx 影子寄存器与计数器之间的比较仍会执行，而且允许将标志置 1。因此可发送相应的中断和 DMA 请求。输出比较模式一节对此进行了介绍。

39.3.8 输出比较模式

此功能用于控制输出波形，或指示已经过某一时间段。

当捕获/比较寄存器与计数器之间相匹配时，输出比较功能：

- 将为相应的输出引脚分配一个可编程值，该值由输出比较模式（TIMx_CCMRx 寄存器中的 OCxM 位）和输出极性（TIMx_CCER 寄存器中的 CCxP 位）定义。匹配时，输出引脚既可保持其电平 (OCXM=000)，也可设置为有效电平 (OCXM=001)、无效电平 (OCXM=010) 或进行翻转 (OCxM=011)。
- 将中断状态寄存器中的标志置 1（TIMx_SR 寄存器中的 CCxIF 位）。
- 如果相应中断使能位（TIMx_DIER 寄存器中的 CCXIE 位）置 1，将生成中断。
- 如果相应使能位（TIMx_DIER 寄存器的 CCxDE 位，TIMx_CR2 寄存器的 CCDS 位，用来选择 DMA 请求）置 1，将发送 DMA 请求。

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

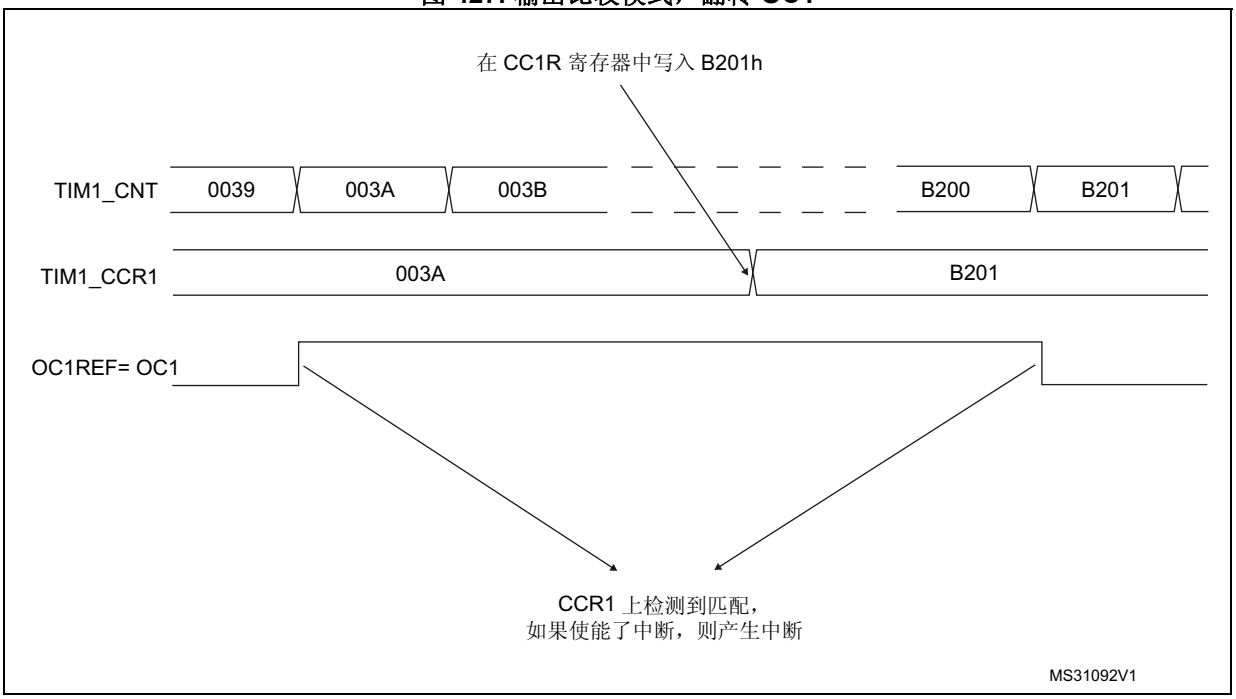
在输出比较模式下，更新事件 UEV 对 ocxref 和 OCx 输出毫无影响。同步的精度可以达到计数器的一个计数周期。输出比较模式也可用于输出单脉冲（在单脉冲模式下）。

步骤

1. 选择计数器时钟（内部、外部、预分频器）。
2. 在 TIMx_ARR 和 TIMx_CCRx 寄存器中写入所需数据。
3. 如果要生成中断和/或 DMA 请求，将 CCxIE 位和/或 CCxDE 位置 1。
4. 选择输出模式。例如，当 CNT 与 CCRx 匹配、未使用预装载 CCRx 并且 OCx 使能且为高电平有效时，必须写入 OCxM=011、OCxPE=0、CCxP=0 和 CCxE=1 来翻转 OCx 输出引脚。
5. 通过将 TIMx_CR1 寄存器中的 CEN 位置 1 来使能计数器。

可随时通过软件更新 TIMx_CCRx 寄存器以控制输出波形，前提是未使能预装载寄存器（OCxPE=0，否则仅当发生下一个更新事件 UEV 时，才会更新 TIMx_CCRx 影子寄存器）。*图 427* 给出了一个示例。

图 427. 输出比较模式，翻转 OC1



39.3.9 PWM 模式

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

通过向 TIMx_CCMRx 寄存器中的 OCxM 位写入 110（PWM 模式 1）或 111（PWM 模式 2），可以独立选择各通道（每个 OCx 输出对应一个 PWM）的 PWM 模式。必须通过将 TIMx_CCMRx 寄存器中的 OCxPE 位置 1 使能相应预装载寄存器，最后通过将 TIMx_CR1 寄存器中的 ARPE 位置 1 使能自动重载预装载寄存器（在递增计数或中心对齐模式下）。

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

OCx 极性可通过软件来编程（使用 TIMx_CCER 寄存器的 CCxP 位）。可将其编程为高电平有效或低电平有效。OCx 输出通过将 TIMx_CCER 寄存器中的 CCxE 位置 1 来使能。有关详细信息，请参见 TIMx_CCERx 寄存器说明。

在 PWM 模式（1 或 2）下，TIMx_CNT 总是与 TIMx_CCRx 进行比较，以确定是 TIMx_CCRx ≤ TIMx_CNT 还是 TIMx_CNT ≤ TIMx_CCRx（取决于计数器计数方向）。不过，为符合 OCREF_CLR 功能（在下一个 PWM 周期之前，ETR 信号上的一个外部事件能够清除 OCREF），OCREF 信号仅在以下情况下变为有效状态：

- 比较结果发生改变，或
- 输出比较模式（TIMx_CCMRx 寄存器中的 OCxM 位）从“冻结”配置（不进行比较，OCxM=“000”）切换为任一 PWM 模式（OCxM=“110”或“111”）。

定时器运行期间，可以通过软件强制 PWM 输出。

根据 TIMx_CR1 寄存器中的 CMS 位状态，定时器能够产生边沿对齐模式或中心对齐模式的 PWM 信号。

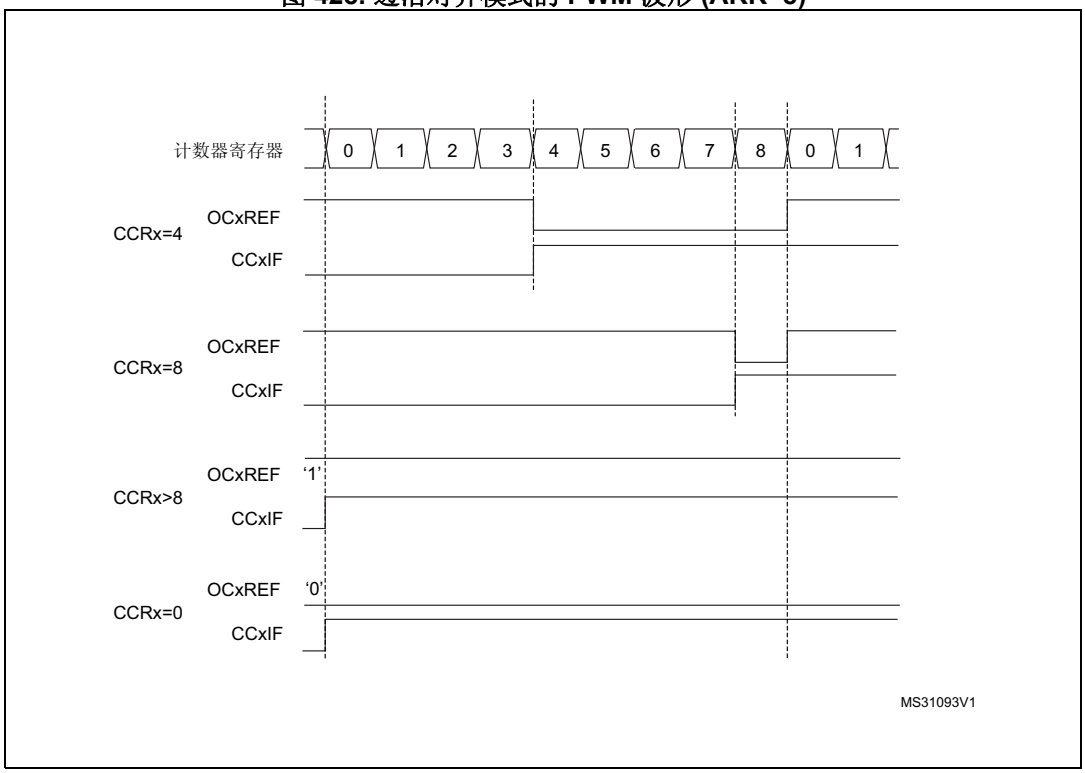
PWM 边沿对齐模式

递增计数配置

当 TIMx_CR1 寄存器中的 DIR 位为低时执行递增计数。请参见*第 1494 页的递增计数模式*。

以下以 PWM 模式 1 为例。只要 TIMx_CNT < TIMx_CCRx，PWM 参考信号 OCxREF 便为高电平，否则为低电平。如果 TIMx_CCRx 中的比较值大于自动重载值（TIMx_ARR 中），则 OCxREF 保持为“1”。如果比较值为 0，则 OCxREF 保持为“0”。*图 428* 举例介绍边沿对齐模式的一些 PWM 波形 (TIMx_ARR=8)。

图 428. 边沿对齐模式的 PWM 波形 (ARR=8)



递减计数配置

当 TIMx_CR1 寄存器中的 DIR 位为高时执行递减计数。请参见*第 1497 页的递减计数模式*。

在 PWM 模式 1 下，只要 TIMx_CNT>TIMx_CCRx，参考信号 ocxref 便为低电平，否则为高电平。如果 TIMx_CCRx 中的比较值大于 TIMx_ARR 中的自动重载值，则 ocxref 保持为 100%。此模式下不可能产生 PWM。

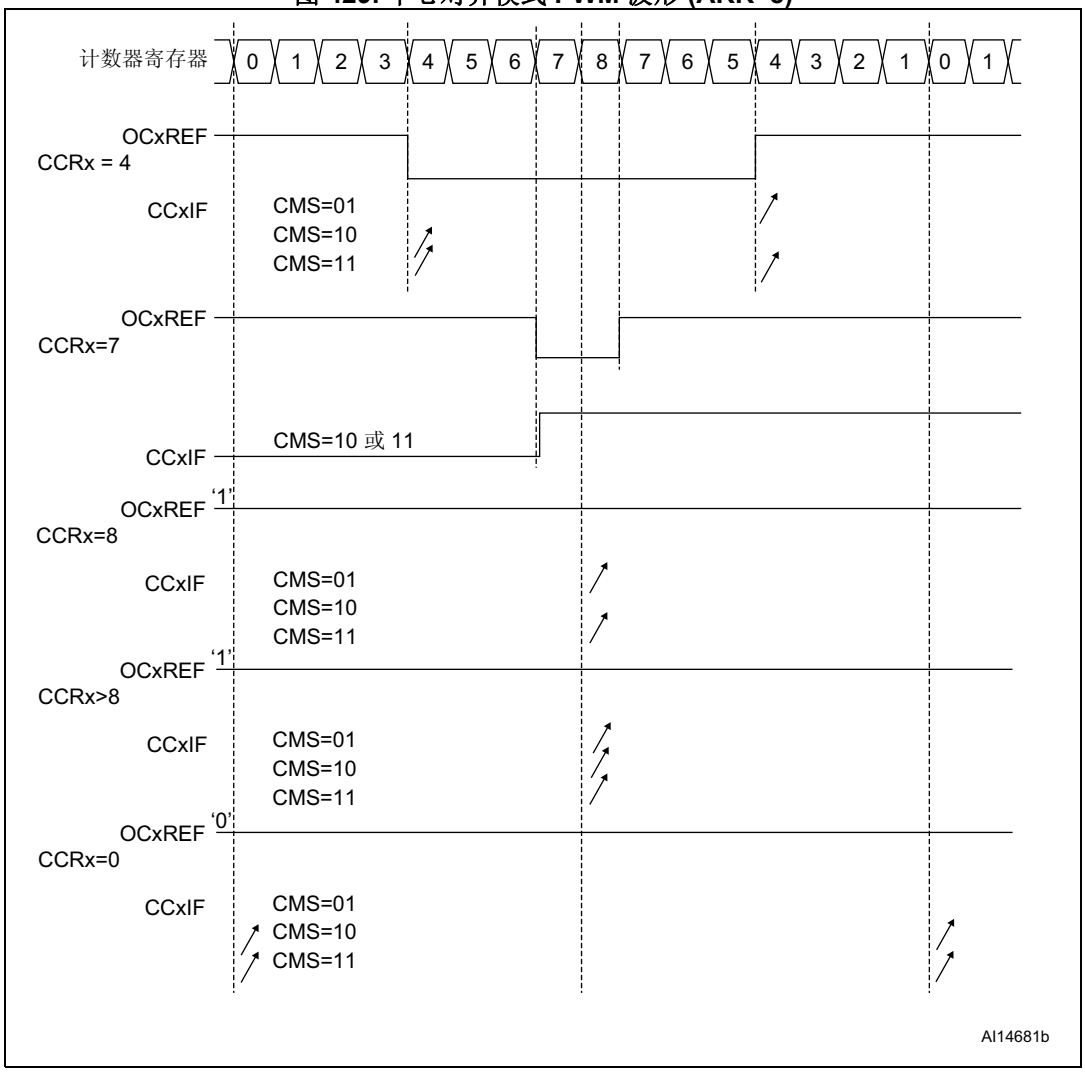
PWM 中心对齐模式

当 TIMx_CR1 寄存器中的 CMS 位不为“00”时（其余所有配置对 ocxref/OCx 信号具有相同的作用），中心对齐模式生效。根据 CMS 位的配置，可以在计数器递增计数、递减计数或同时递增和递减计数时将比较标志置 1。TIMx_CR1 寄存器中的方向位 (DIR) 由硬件更新，不得通过软件更改。请参见*第 1500 页的中心对齐模式（递增/递减计数）*。

图 429 显示了中心对齐模式的 PWM 波形，在此例中：

- TIMx_ARR=8
- PWM 模式为 PWM 模式 1
- 在根据 TIMx_CR1 寄存器中 CMS=01 而选择的中心对齐模式 1 下，当计数器递减计数时，比较标志置 1

图 429. 中心对齐模式 PWM 波形 (ARR=8)



中心对齐模式使用建议：

- 启动中心对齐模式时将使用当前的递增/递减计数配置。这意味着计数器将根据写入 TIMx_CR1 寄存器中 DIR 位的值进行递增或递减计数。此外，不得同时通过软件修改 DIR 和 CMS 位。
- 不建议在运行中心对齐模式时对计数器执行写操作，否则将发生意想不到的结果。尤其是：
  - 如果写入计数器中的值大于自动重载值 (TIMx_CNT>TIMx_ARR)，计数方向不会更新。例如，如果计数器之前递增计数，则继续递增计数。
  - 如果向计数器写入 0 或 TIMx_ARR 的值，计数方向会更新，但不生成更新事件 UEV。
- 使用中心对齐模式最为保险的方法是：在启动计数器前通过软件生成更新（将 TIMx_EGR 寄存器中的 UG 位置 1），并且不要在计数器运行过程中对其执行写操作。

39.3.10 不对称 PWM 模式

在不对称模式下，生成的两个中心对齐 PWM 信号间允许存在可编程相移。频率由 TIMx_ARR 寄存器的值确定，而占空比和相移则由一对 TIMx_CCRx 寄存器确定。两个寄存器分别控制递增计数和递减计数期间的 PWM，这样每半个 PWM 周期便会调节一次 PWM：

- OC1REFC（或 OC2REFC）由 TIMx_CCR1 和 TIMx_CCR2 控制
- OC3REFC（或 OC4REFC）由 TIMx_CCR3 和 TIMx_CCR4 控制

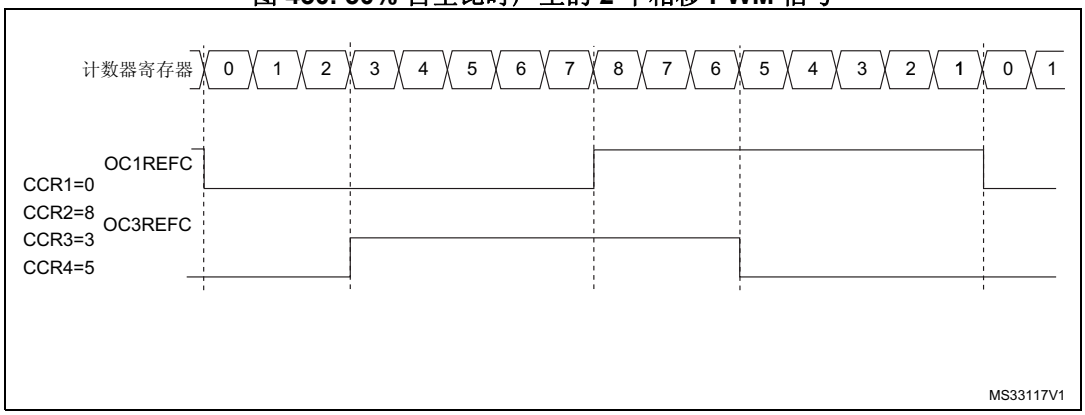
两个通道可以独立选择不对称 PWM 模式（每对 CCR 寄存器一个 OCx 输出），只需向 TIMx_CCMRx 寄存器的 OCxM 位写入“1110”（不对称 PWM 模式 1）或“1111”（不对称 PWM 模式 2）。

注： *出于兼容性原因，OCxM[3:0] 位域分为两部分，最高有效位与最低有效的 3 位不相邻。*

给定通道用作不对称 PWM 通道时，也可使用其辅助通道。例如，如果通道 1 上产生 OC1REFC 信号（不对称 PWM 模式 1），则由于不对称 PWM 模式 2 的原因，通道 2 上可输出 OC2REF 信号或 OC2REFC 信号。

图 430 显示了不对称 PWM 模式下可以产生的信号示例（通道 1 到通道 4 在不对称 PWM 模式 1 下配置）。

图 430. 50% 占空比时产生的 2 个相移 PWM 信号



39.3.11 组合 PWM 模式

在组合 PWM 模式下，生成的两个边沿或中心对齐 PWM 信号的各个脉冲间允许存在可编程延时和相移。频率由 TIMx_ARR 寄存器的值确定，而占空比和延时则由两个 TIMx_CCRx 寄存器确定。产生的信号 OCxREFC 由两个参考 PWM 的逻辑或运算或者逻辑与运算组合组成。

– OC1REFC（或 OC2REFC）由 TIMx_CCR1 和 TIMx_CCR2 控制

– OC3REFC（或 OC4REFC）由 TIMx_CCR3 和 TIMx_CCR4 控制

两个通道可以独立选择组合 PWM 模式（每对 CCR 寄存器一个 OCx 输出），只需向 TIMx_CCMRx 寄存器的 OCxM 位写入“1100”（组合 PWM 模式 1）或“1101”（组合 PWM 模式 2）。

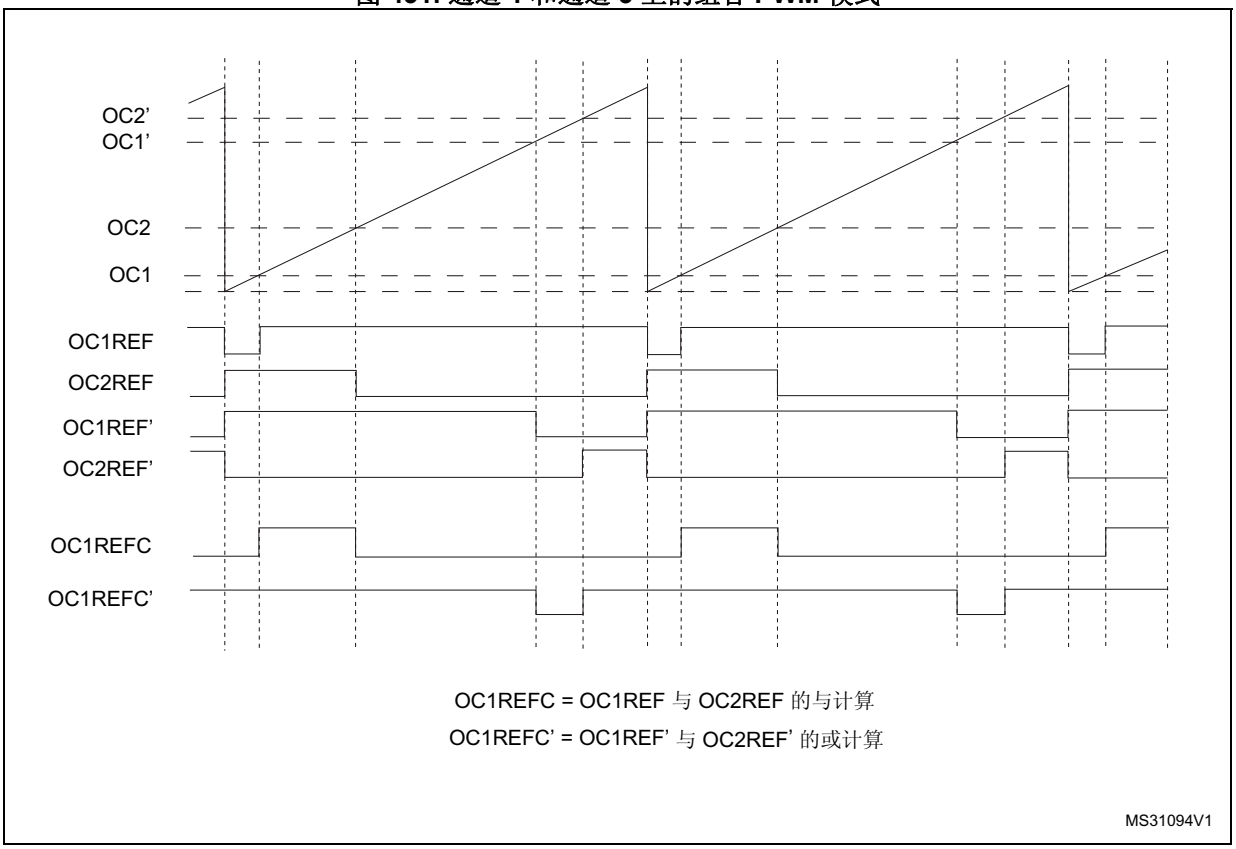
当给定通道用作组合 PWM 通道时，其辅助通道必须在相反的 PWM 模式下配置（例如，一个通道在组合 PWM 模式 1 下配置，另一个通道在组合 PWM 模式 2 下配置）。

注： *出于兼容性原因，OCxM[3:0] 位域分为两部分，最高有效位与最低有效的 3 位不相邻。*

图 431 显示了不对称 PWM 模式下可以产生的信号示例，通过以下配置可获得这些信号：

- 通道 1 在组合 PWM 模式 2 下配置
- 通道 2 在 PWM 模式 1 下配置
- 通道 3 在组合 PWM 模式 2 下配置
- 通道 4 在 PWM 模式 1 下配置

图 431. 通道 1 和通道 3 上的组合 PWM 模式



39.3.12 发生外部事件时清除 OCxREF 信号

对于给定通道，在 ocref_clr_int 输入上施加高电平（相应 TIMx_CCMRx 寄存器中的 OCxCE 使能位置 1），可将 OCxREF 信号清零。OCxREF 信号将保持低电平，直到发生下一更新事件 (UEV)。该功能只能在输出比较模式和 PWM 模式下使用。在强制模式下不起作用。

ocref_clr_int 连接到 ETRF 信号（滤波后的 ETR）。

对于给定通道，在 ETRF 输入施加高电平（相应 TIMx_CCMRx 寄存器中的 OCxCE 使能位置 1），可将 OCxREF 信号复位。OCxREF 信号将保持低电平，直到发生下一更新事件 (UEV)。

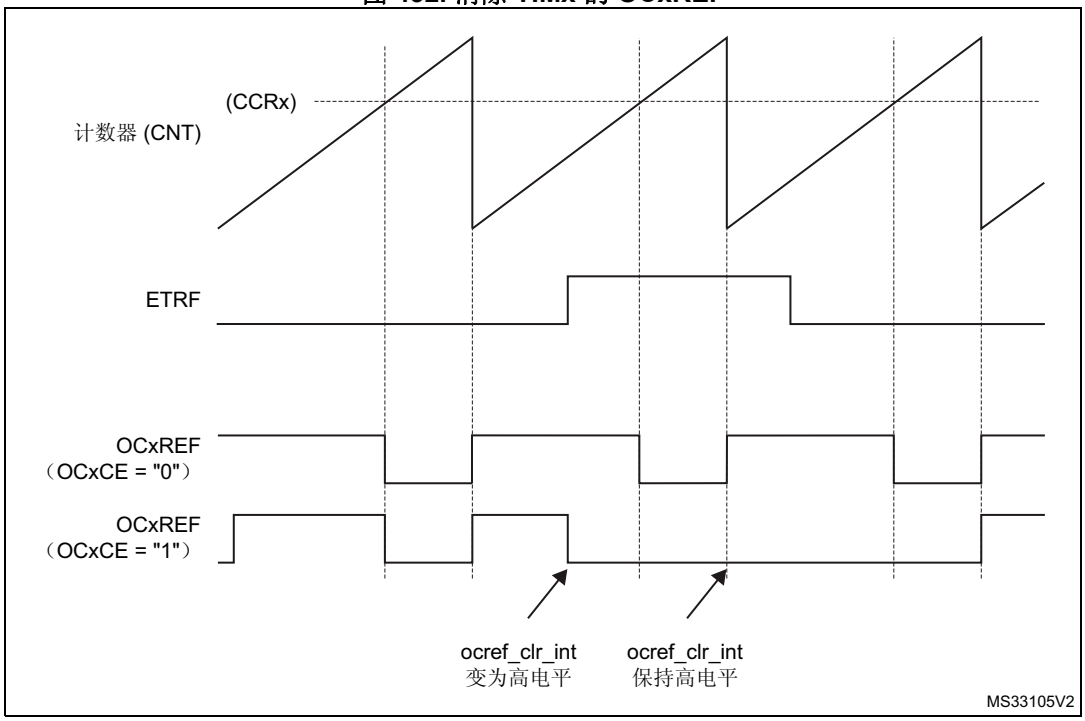
该功能只能在输出比较模式和 PWM 模式下使用。在强制模式下不起作用。

例如，OCxREF 信号可以连接到比较器的输出，用于控制电流。此时，ETR 必须配置如下：

1. 必须关闭外部触发预分频器：TIMx_SMCR 寄存器中的 ETPS[1:0] 位清为 00。
2. 必须禁止外部时钟模式 2：TIM1_SMCR 寄存器中的 ECE 位清为 0。
3. 外部触发极性 (ETP) 和外部触发滤波器 (ETF) 可根据应用需要进行配置。

图 432 对比了不同使能位 OCxCE 值的情况，显示了当 ETRF 输入变为高电平时 OCxREF 信号的行为。在本例中，定时器 TIMx 编程为 PWM 模式。

图 432. 清除 TIMx 的 OCxREF



注： *如果 PWM 的占空比为 100% (CCRx>ARR)，则下次计数器上溢时会再次使能 OCxREF。*

39.3.13 单脉冲模式

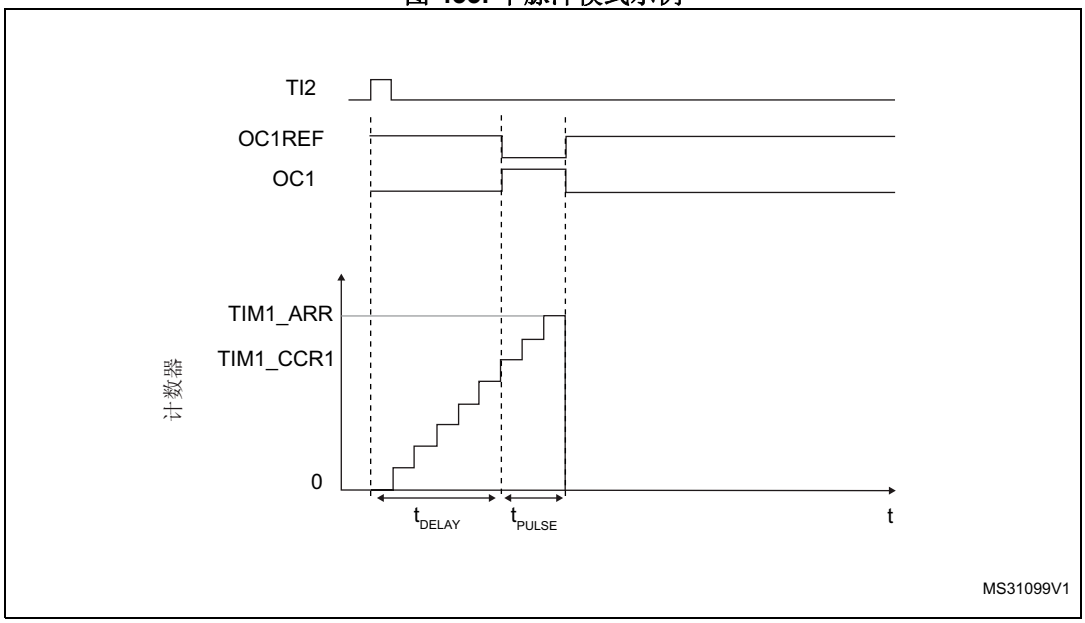
单脉冲模式 (OPM) 是上述模式的一个特例。在这种模式下，计数器可以在一个激励信号的触发下启动，并可在一段可编程的延时后产生一个脉宽可编程的脉冲。

可以通过从模式控制器启动计数器。可以在输出比较模式或 PWM 模式下生成波形。将 TIMx_CR1 寄存器中的 OPM 位置 1，即可选择单脉冲模式。这样，发生下一更新事件 UEV 时，计数器将自动停止。

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

- CNT<CCRx ≤ ARR（特别注意，0 < CCRx）

图 433. 单脉冲模式示例



例如，用户希望达到这样的效果：在 TI2 输入引脚检测到上升沿时，经过 t_{DELAY} 的延迟，在 OC1 上产生一个长度为 t_{PULSE} 的正脉冲。

使用 TI2FP2 作为触发 1：

- 在 TIMx_CCMR1 寄存器中写入 CC2S = 01，将 TI2FP2 映射到 TI2。
- 在 TIMx_CCER 寄存器中写入 CC2P=0 和 CC2NP=“0”，使 TI2FP2 能够检测上升沿。
- 在 TIMx_SMCR 寄存器中写入 TS=00110，将 TI2FP2 配置为从模式控制器的触发 (TRGI)。
- 在 TIMx_SMCR 寄存器中写入 SMS=“110”（触发模式），使用 TI2FP2 启动计数器。

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

- t_{DELAY} 由写入 TIMx_CCR1 寄存器的值定义。
- t_{PULSE} 由自动重载值与比较值之差 (TIMx_ARR - TIMx_CCR1) 来定义。
- 假设希望产生这样的波形：信号在发生比较匹配时从“0”变为“1”，在计数器达到自动重载值时由“1”变为“0”。为此，应在 TIMx_CCMR1 寄存器中写入 OC1M=111，以使能 PWM 模式 2。如果需要，可选择在 TIMx_CCMR1 寄存器的 OC1PE 和 TIMx_CR1 寄存器的 ARPE 中写入 1，使能预装载寄存器。这种情况下，必须在 TIMx_CCR1 寄存器中写入比较值并在 TIMx_ARR 寄存器中写入自动重载值，通过将 UG 位置 1 来产生更新，然后等待 TI2 上的外部触发事件。本例中，CC1P 的值为“0”。

在本例中，TIMx_CR1 寄存器中的 DIR 和 CMS 位应为低。

由于仅需要 1 个脉冲（单脉冲模式），因此应向 TIMx_CR1 寄存器的 OPM 位写入“1”，以便在发生下一更新事件（计数器从自动重载值返回到 0）时使计数器停止计数。TIMx_CR1 寄存器中的 OPM 位置“0”时，即选择重复模式。

特殊情况：OCx 快速使能：

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

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

39.3.14 可再触发单脉冲模式 (OPM)

该模式允许计数器可以在一个激励信号的触发下启动，并且能产生长度可编程的脉冲，但与不可再触发单脉冲模式间存在以下差别，如*第 39.3.13 节*所述：

- 发生触发时，脉冲立即产生（无可编程延时）
- 如果在上一个触发完成前发生新的触发，脉冲将延长

定时器必须处于从模式，TIMx_SMCR 寄存器中的位 SMS[3:0] = “1000”（组合复位 + 触发模式），针对可再触发 OPM 模式 1 或模式 2 将 OCxM[3:0] 位设置为“1000”或“1001”。

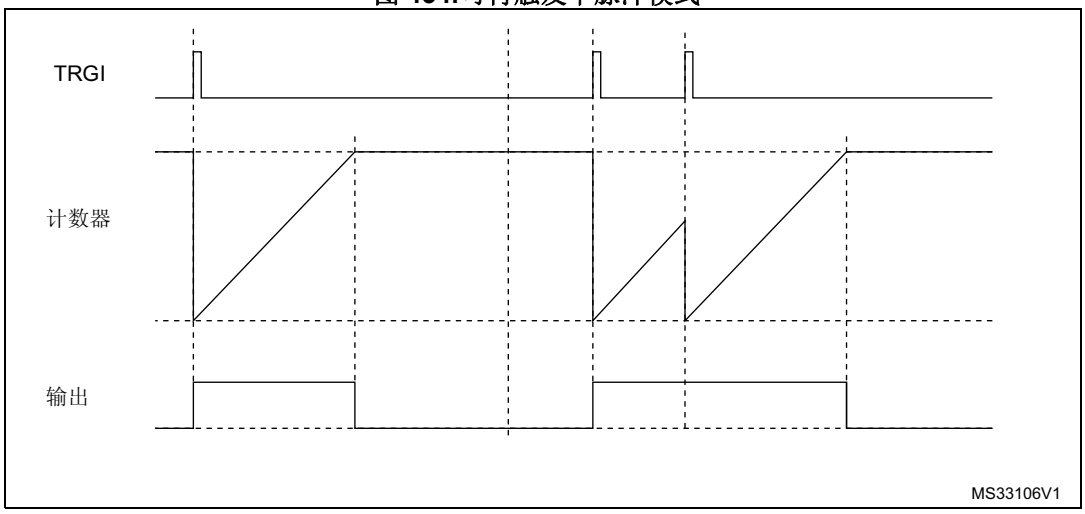
定时器配置为递增计数模式时，相应的 CCRx 必须置 0（ARR 寄存器设置脉冲长度）。如果定时器配置为递减计数模式，CCRx 必须高于或等于 ARR。

注： *在可再触发单脉冲模式下，CCxIF 标志没有意义。*

出于兼容性原因，OCxM[3:0] 和 SMS[3:0] 位域分为两部分，最高有效位与最低有效的 3 位不相邻。

此模式不能与中心对齐 PWM 模式一起使用。在 TIMx_CR1 中必须设置 CMS[1:0] = 00。

图 434.可再触发单脉冲模式



39.3.15 编码器接口模式

选择编码器接口模式时，如果计数器仅在 TI2 边沿处计数，在 TIMx_SMCR 寄存器中写入 SMS=001；如果计数器仅在 TI1 边沿处计数，写入 SMS=010；如果计数器在 TI1 和 TI2 边沿处均计数，则写入 SMS=011。

通过编程 TIMx_CCER 寄存器的 CC1P 和 CC2P 位，选择 TI1 和 TI2 极性。CC1NP 和 CC2NP 必须保持清零。如果需要，还可对输入滤波器进行编程。CC1NP 和 CC2NP 必须保持低电平。

TI1 和 TI2 两个输入用于连接增量编码器。请参见 *表 316*。如果使能计数器（在 TIMx_CR1 寄存器的 CEN 位中写入“1”），则计数器的时钟由 TI1FP1 或 TI2FP2 上的每次有效信号转换提供。TI1FP1 和 TI2FP2 是进行输入滤波器和极性选择后 TI1 和 TI2 的信号，如果不进行滤波和反相，则 TI1FP1=TI1，TI2FP2=TI2。将根据两个输入的信号转换序列，产生计数脉冲和方向信号。根据该信号转换序列，计数器相应递增或递减计数，同时硬件对 TIMx_CR1 寄存器的 DIR 位进行相应修改。任何输入（TI1 或 TI2）发生信号转换时，都会计算 DIR 位，无论计数器是仅在 TI1 或 TI2 边沿处计数，还是同时在 TI1 和 TI2 处计数。

编码器接口模式就相当于带有方向选择的外部时钟。这意味着，计数器仅在 0 到 TIMx_ARR 寄存器中的自动重载值之间进行连续计数（根据具体方向，从 0 递增计数到 ARR，或从 ARR 递减计数到 0）。因此，在启动前必须先配置 TIMx_ARR。同样，捕获、比较、预分频器、触发输出功能继续正常工作。

在此模式下，计数器会根据正交编码器的速度和方向自动进行修改，因此，其内容始终表示编码器的位置。计数方向对应于所连传感器的旋转方向。下表汇总了可能的组合（假设 TI1 和 TI2 不同时切换）。

表 316. 计数方向与编码器信号的关系

有效边沿	相反信号的电平（TI1FP1 对应 TI2，TI2FP2 对应 TI1）	TI1FP1 信号		TI2FP2 信号	
		上升	下降	上升	下降
仅在 TI1 处计数	高	递减	递增	不计数	不计数
	低	递增	递减	不计数	不计数
仅在 TI2 处计数	高	不计数	不计数	递增	递减
	低	不计数	不计数	递减	递增
在 TI1 和 TI2 处均计数	高	递减	递增	递增	递减
	低	递增	递减	递减	递增

外部增量编码器可直接与 MCU 相连，无需外部接口逻辑。不过，通常使用比较器将编码器的差分输出转换为数字信号。这样大幅提高了抗噪声性能。用于指示机械零位的第三个编码器输出可与外部中断输入相连，用以触发计数器复位。

图 435 以计数器工作为例，说明了计数信号的生成和方向控制。同时也说明了选择双边沿时如何对输入抖动进行补偿。将传感器靠近其中一个切换点放置时可能出现这种情况。本例中假设配置如下：

- CC1S=01（TIMx_CCMR1 寄存器，TI1FP1 映射到 TI1 上）
- CC2S=01（TIMx_CCMR2 寄存器，TI2FP2 映射到 TI2 上）
- CC1P 和 CC1NP =“0”（TIMx_CCER 寄存器，TI1FP1 未反相，TI1FP1=TI1）
- CC2P 和 CC2NP =“0”（TIMx_CCER 寄存器，TI2FP2 未反相，TI2FP2= TI2）
- SMS=011（TIMx_SMCR 寄存器，两个输入在上升沿和下降沿均有效）
- CEN = 1（TIMx_CR1 寄存器，使能计数器）

图 435. 编码器接口模式下的计数器工作示例

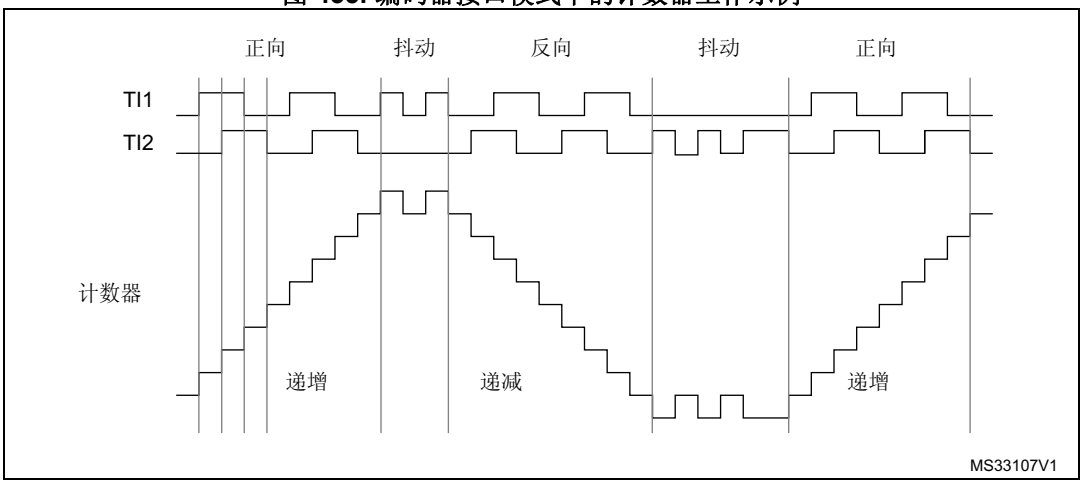
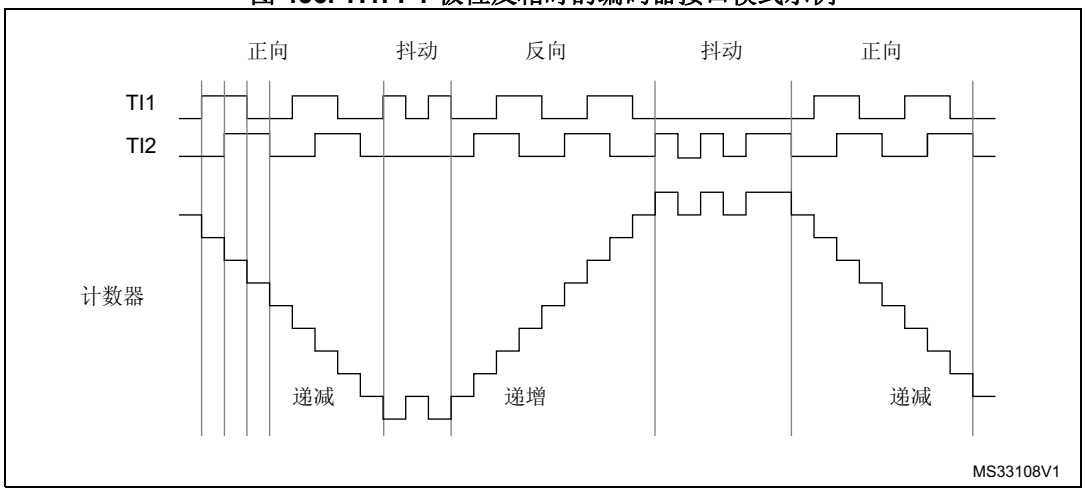


图 436 说明了 TI1FP1 极性反相时的计数器行为示例（除 CC1P=1 外，其它配置与上例相同）。

图 436. TI1FP1 极性反相时的编码器接口模式示例



定时器配置为编码器接口模式时，会提供传感器当前位置的相关信息。使用另一个配置为捕获模式的定时器测量两个编码器事件之间的周期，可获得动态信息（速度、加速度和减速度）。指示机械零位的编码器输出即可用于此目的。根据两个事件之间的时间间隔，还可定期读取计数器。如果可能，可以将计数器值锁存到第三个输入捕获寄存器来实现此目的（捕获信号必须为周期性信号，可以由另一个定时器产生）； 此外，还可以通过实时时钟产生的 DMA 请求读取计数器值。

39.3.16 UIF 位重映射

TIMx_CR1 寄存器中的 IUFREMAP 位强制将更新中断标志 (UIF) 连续复制到定时计数器寄存器的位 31 (TIMxCNT[31]) 中。这样便可自动读取计数器值以及由 UIFCPY 标志发出的电位翻转条件。这可避免在后台任务（计数器读）和中断（更新中断）之间共享处理时产生竞争条件，从而简化角速度的计算。

UIF 和 UIFCPY 标志使能之间没有延迟。

在 32 位定时器实现中，当 IUFREMAP 位置 1 时，计数器的位 31 在读访问时由 UIFCPY 标志覆盖（计数器的最高有效位只能在写模式下访问）。

39.3.17 定时器输入异或功能

借助 TIM1xx_CR2 寄存器中的 TI1S 位，可将通道 1 的输入滤波器连接到异或门的输出，从而将 TIMx_CH1 到 TIMx_CH3 这三个输入引脚组合在一起。

异或输出可与触发或输入捕获等所有定时器输入功能配合使用。

*第 1439 页的第 38.3.25 节：连接霍尔传感器*以连接霍尔传感器为例介绍了此功能。

39.3.18 定时器与外部触发同步

TIMx 定时器可与外部触发以下列模式实现同步：复位模式、门控模式和触发模式。

从模式：复位模式

当触发输入信号发生变化时，计数器及其预分频器可重新初始化。此外，如果 TIMx_CR1 寄存器中的 URS 位为 0，则会生成更新事件 UEV。然后，所有预装载寄存器（TIMx_ARR 和 TIMx_CCRx）都将更新。

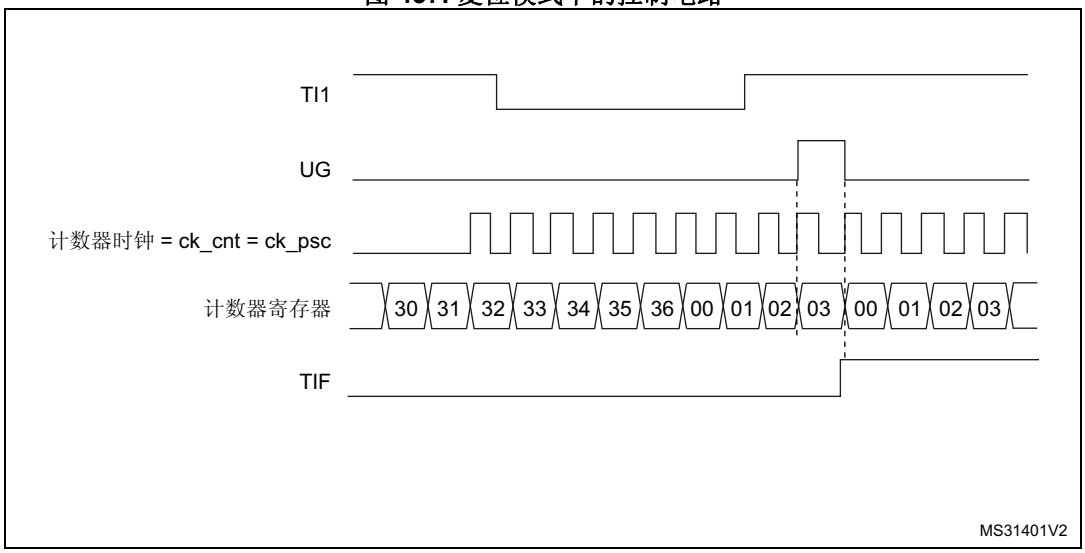
在以下示例中，TI1 输入上出现上升沿时，递增计数器清零：

1. 将通道 1 配置为检测 TI1 的上升沿。配置输入滤波带宽（本例中不需要任何滤波器，因此保持 IC1F=0000）。由于捕获预分频器不用于触发操作，因此无需对其进行配置。CC1S 位只选择输入捕获源，即 TIMx_CCMR1 寄存器中的 CC1S = 01。在 TIMx_CCER 寄存器中写入 CC1P=0 和 CC1NP=0，验证极性（仅检测上升沿）。
2. 在 TIMx_SMCR 寄存器中写入 SMS=100，将定时器配置为复位模式。在 TIMx_SMCR 寄存器中写入 TS=00101，选择 TI1 作为输入源。
3. 在 TIMx_CR1 寄存器中写入 CEN=1，启动计数器。

计数器开始根据内部时钟计数，然后正常运转，直到出现 TI1 上升沿。当 TI1 出现上升沿时，计数器清零，然后重新从 0 开始计数。同时，触发标志（TIMx_SR 寄存器中的 TIF 位）置 1，使能中断或 DMA 后，还可发送中断或 DMA 请求（取决于 TIMx_DIER 寄存器中的 TIE 和 TDE 位）。

下图显示了自动重载寄存器 TIMx_ARR=0x36 时的相关行为。TI1 的上升沿与实际计数器复位之间的延迟是由于 TI1 输入的重新同步电路引起的。

图 437. 复位模式下的控制电路



从模式：门控模式

输入信号的电平可用来使能计数器。

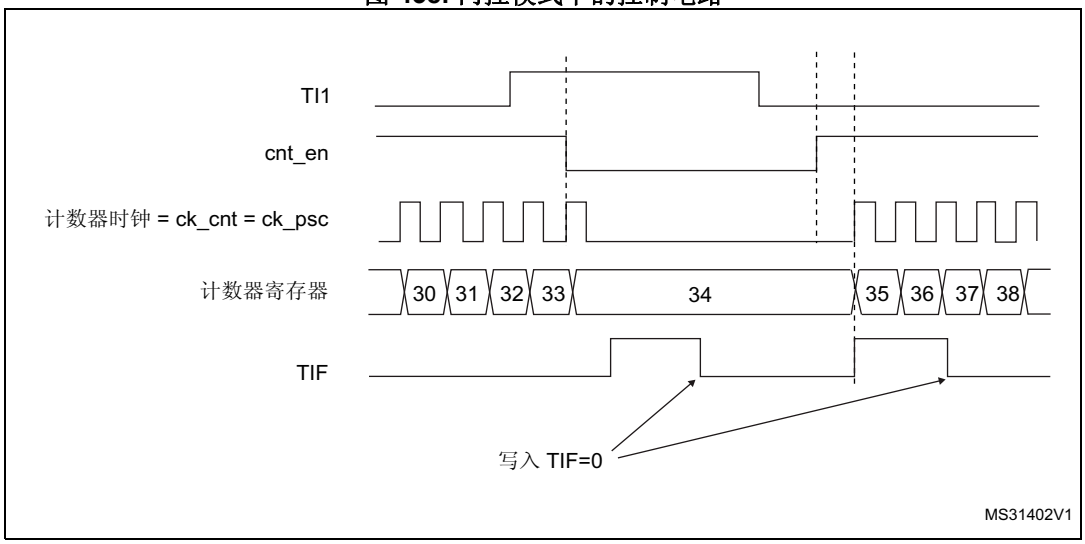
在以下示例中，递增计数器仅在 TI1 输入为低电平时计数：

1. 将通道 1 配置为检测 TI1 上的低电平。配置输入滤波带宽（本例中不需要任何滤波器，因此保持 IC1F=0000）。由于捕获预分频器不用于触发操作，因此无需对其进行配置。CC1S 位只选择输入捕获源，即 TIMx_CCMR1 寄存器中的 CC1S=01。在 TIMx_CCER 寄存器中写入 CC1P=1 和 CC1NP=0，以确定极性（仅检测低电平）。
2. 在 TIMx_SMCR 寄存器中写入 SMS=101，将定时器配置为门控模式。在 TIMx_SMCR 寄存器中写入 TS=00101，选择 TI1 作为输入源。
3. 在 TIMx_CR1 寄存器中写入 CEN=1，使能计数器（在门控模式下，如果 CEN=0，则无论触发输入电平如何，计数器都不启动）。

只要 TI1 为低电平，计数器就开始根据内部时钟计数，直到 TI1 变为高电平时停止计数。计数器启动或停止时，TIMx_SR 寄存器中的 TIF 标志都会置 1。

TI1 的上升沿与实际计数器停止之间的延迟是由于 TI1 输入的重新同步电路引起的。

图 438. 门控模式下的控制电路



1. 由于门控模式作用于电平而非边沿，因此在门控模式下，“CCxP=CCxNP=1”（同时检测上升沿和下降沿）不发挥任何作用。

注： *由于门控模式作用于电平而非边沿，因此在门控模式下，“CCxP=CCxNP=1”（同时检测上升沿和下降沿）不发挥任何作用。*

从模式：触发模式

所选输入上发生某一事件时可以启动计数器。

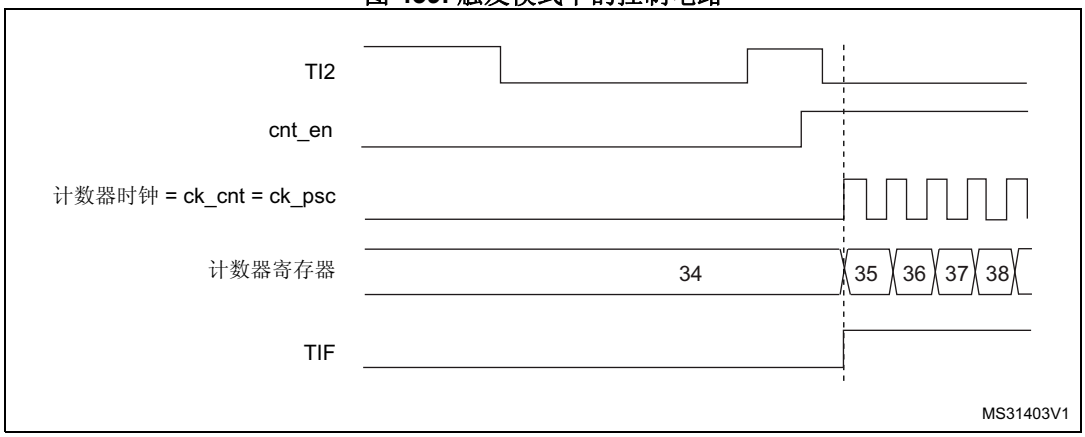
在以下示例中，TI2 输入上出现上升沿时，递增计数器启动：

1. 将通道 2 配置为检测 TI2 上的上升沿。配置输入滤波带宽（本例中不需要任何滤波器，因此保持 IC2F=0000）。由于捕获预分频器不用于触发操作，因此无需对其进行配置。CC2S 位只选择输入捕获源，即 TIMx_CCMR1 寄存器中的 CC2S=01。在 TIMx_CCER 寄存器中写入 CC2P=1 和 CC2NP=0，以确定极性（仅检测低电平）。
2. 在 TIMx_SMCR 寄存器中写入 SMS=110，将定时器配置为触发模式。在 TIMx_SMCR 寄存器中写入 TS=00110，选择 TI2 作为输入源。

当 TI2 出现上升沿时，计数器开始根据内部时钟计数，并且 TIF 标志置 1。

TI2 的上升沿与实际计数器启动之间的延迟是由于 TI2 输入的重新同步电路引起的。

图 439. 触发模式下的控制电路



从模式：外部时钟模式 2 + 触发模式

外部时钟模式 2 可与另一种从模式（外部时钟模式 1 和编码器模式除外）结合使用。这种情况下，ETR 信号用作外部时钟输入，在复位模式、门控模式或触发模式下工作时，可选择另一个输入作为触发输入。不建议通过 TIMx_SMCR 寄存器中的 TS 位来选择 ETR 作为 TRGI。

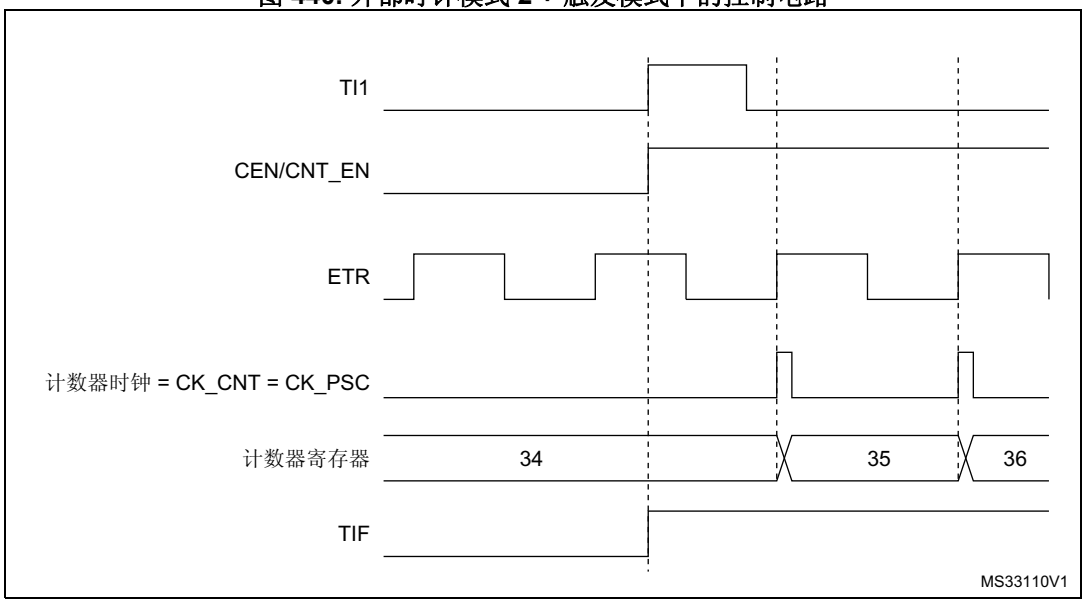
在以下示例中，只要 TI1 出现上升沿，递增计数器即会在 ETR 信号的每个上升沿处递增：

1. 通过对 TIMx_SMCR 寄存器进行如下编程，配置外部触发输入电路：
   - ETF = 0000：无滤波器。
   - ETPS = 00：禁止预分频器。
   - ETP = 0：检测 ETR 的上升沿，并写入 ECE=1，以使能外部时钟模式 2。
2. 如下配置通道 1，以检测 TI 的上升沿：
   - IC1F=0000：无滤波器。
   - 由于捕获预分频器不用于触发操作，因此无需对其进行配置。
   - TIMx_CCMR1 寄存器中 CC1S=01，只选择输入捕获源。
   - TIMx_CCER 寄存器中 CC1P=0 且 CC1NP= 0，以确定极性（仅检测上升沿）。
3. 在 TIMx_SMCR 寄存器中写入 SMS=110，将定时器配置为触发模式。在 TIMx_SMCR 寄存器中写入 TS=00101，选择 TI1 作为输入源。

TI1 出现上升沿时将使能计数器并且 TIF 标志置 1。然后计数器在 ETR 出现上升沿时计数。

ETR 信号的上升沿与实际计数器复位之间的延迟是由于 ETRP 输入的重新同步电路引起的。

图 440. 外部时钟模式 2 + 触发模式下的控制电路

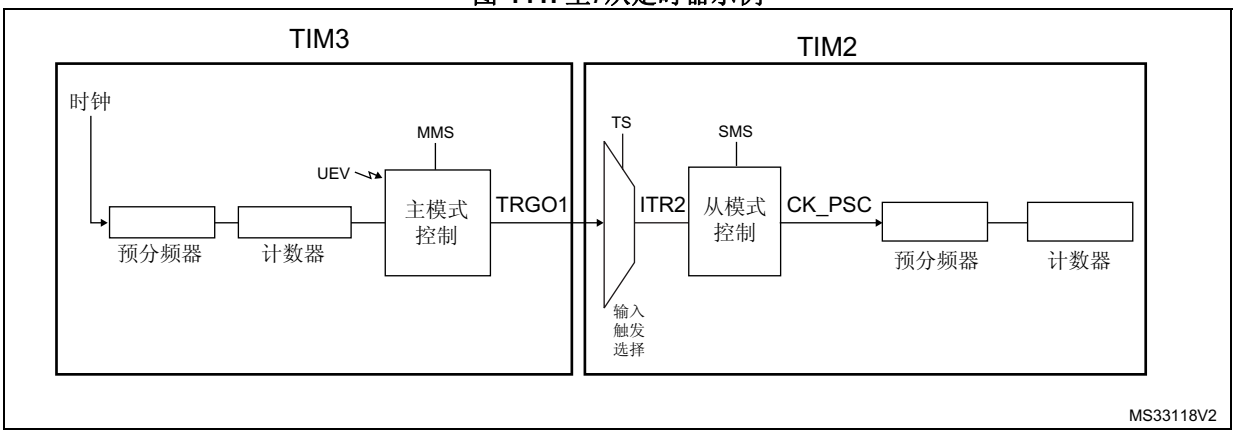


39.3.19 定时器同步

TIMx 定时器从内部连接在一起，以实现定时器同步或链接。当某个定时器配置为主模式时，可对另一个配置为从模式的定时器的计数器执行复位、启动、停止操作或为其提供时钟。

*图 441：主/从定时器示例*简要介绍了触发选择和主模式选择框图。

图 441. 主/从定时器示例



将一个定时器用作另一个定时器的预分频器

例如，可以将 TIM 配置为 TIM2 的预分频器。3 请参见*图 441*。为此：

1. 将 TIM3 配置为主模式，以便每次发生更新事件 UEV 时都输出一个周期性触发信号。如果在 TIM3_CR2 寄存器中写入 MMS=010，则每次生成更新事件时，TRGO 都会输出一个上升沿。
2. 要将 TIM3 的 TRGO 输出连接到 TIM2，必须将 TIM2 配置为从模式，使用 ITR2 作为内部触发。通过 TIM2_SMCR 寄存器中的 TS 位（写入 TS=00010）可对此进行选择。
3. 然后将从模式控制器设为外部时钟模式 1（在 TIM2_SMCR 寄存器中写入 SMS=111）。这样一来，TIM2 的时钟将由 TIM3 周期性触发信号的上升沿（与 TIM3 的计数器上溢对应）提供。
4. 最后必须通过将这两个定时器的相应 CEN 位（TIMx_CR1 寄存器）置 1 同时使能二者。

注： *如果选择 TIM3 的 OCx 信号作为触发输出 (MMS=1xx)，该信号的上升沿将用于驱动 TIM2 的计数器。*

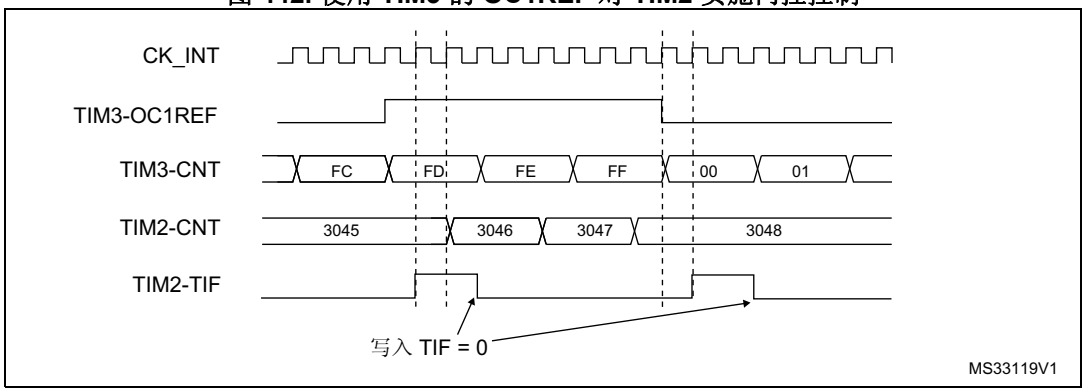
使用一个定时器使能另一个定时器

本例中通过定时器 3 的输出比较 1 来使能 TIM2。相关连接图，请参见*图 441*。仅当 TIM3 的 OC1REF 为高电平时，TIM2 才根据分频后的内部时钟进行计数。两个计数器的时钟频率都基于 CK_INT 通过预分频器执行 3 分频 ($f_{CK_CNT} = f_{CK_INT}/3$)。

1. 将 TIM3 配置为主模式，发送其输出比较 1 参考信号 (OC1REF) 作为触发输出（TIM3_CR2 寄存器中的 MMS=100）。
2. 配置 TIM3 的 OC1REF 波形（TIM3_CCMR1 寄存器）。
3. 配置 TIM2 以接收来自 TIM3 的输入触发（TIM2_SMCR 寄存器中的 TS=00010）。
4. 将 TIM2 配置为门控模式（TIM2_SMCR 寄存器中的 SMS=101）。
5. 通过向 CEN 位（TIM2_CR1 寄存器）写入“1”使能 TIM2。
6. 通过向 CEN 位（TIM3_CR1 寄存器）写入“1”启动 TIM3。

注： *计数器 2 的时钟与计数器 1 不同步，此模式仅影响 TIM2 的计数器使能信号。*

图 442. 使用 TIM3 的 OC1REF 对 TIM2 实施门控控制

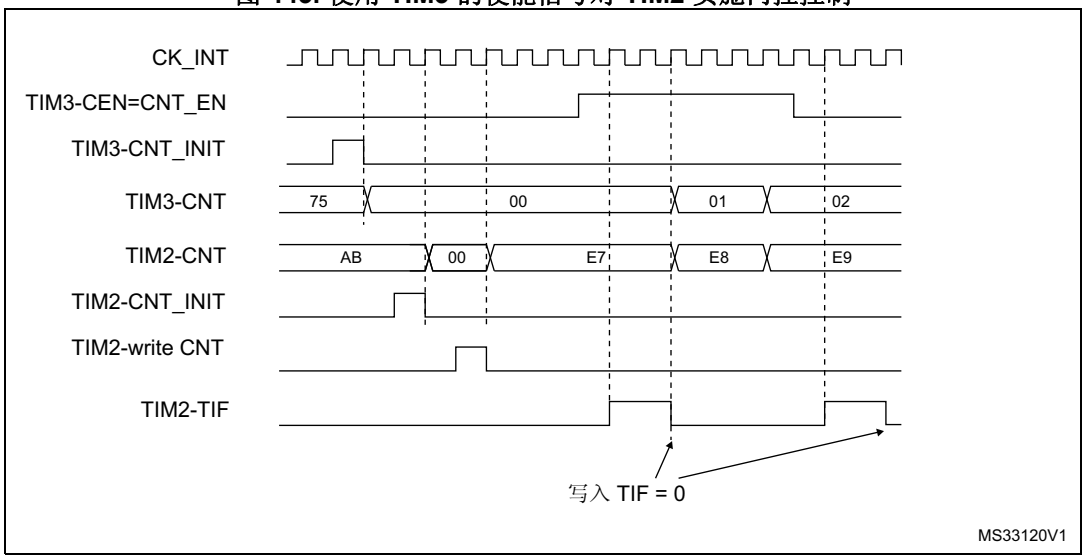


在*图 442* 的示例中，TIM2 的计数器和预分频器在启动前未进行初始化。因此从各自的当前值开始计数。启动定时器 TIM3 之前，通过复位这两个定时器可以从指定值开始计数。这样便可以在定时器计数器中写入所需的任意值。两个定时器都可通过软件使用 TIMx_EGR 寄存器中的 UG 位轻松复位。

在下一示例中（请参见*图 443*），TIM3 与 TIM2 同步。TIM3 为主模式，从 0 开始计数。TIM2 为从模式，从 0xE7 开始计数。两个定时器的预分频比相同。通过向 TIM3_CR1 寄存器中的 CEN 位写入“0”来禁止 TIM3 时，TIM2 将停止：

1. 将 TIM3 配置为主模式，发送其输出比较 1 参考信号 (OC1REF) 作为触发输出（TIM3_CR2 寄存器中的 MMS=100）。
2. 配置 TIM3 的 OC1REF 波形（TIM3_CCMR1 寄存器）。
3. 配置 TIM2 以接收来自 TIM3 的输入触发（TIM2_SMCR 寄存器中的 TS=00010）。
4. 将 TIM2 配置为门控模式（TIM2_SMCR 寄存器中的 SMS=101）。
5. 通过向 UG 位（TIM3_EGR 寄存器）写入“1”复位 TIM3。
6. 通过向 UG 位（TIM2_EGR 寄存器）写入“1”复位 TIM2。
7. 通过在 TIM2 的计数器 (TIM2_CNTL) 中写入“0xE7”使 TIM2 初始化为 0xE7。
8. 通过向 CEN 位（TIM2_CR1 寄存器）写入“1”使能 TIM2。
9. 通过向 CEN 位（TIM3_CR1 寄存器）写入“1”启动 TIM3。
10. 通过向 CEN 位（TIM3_CR1 寄存器）写入“0”停止 TIM3。

图 443. 使用 TIM3 的使能信号对 TIM2 实施门控控制

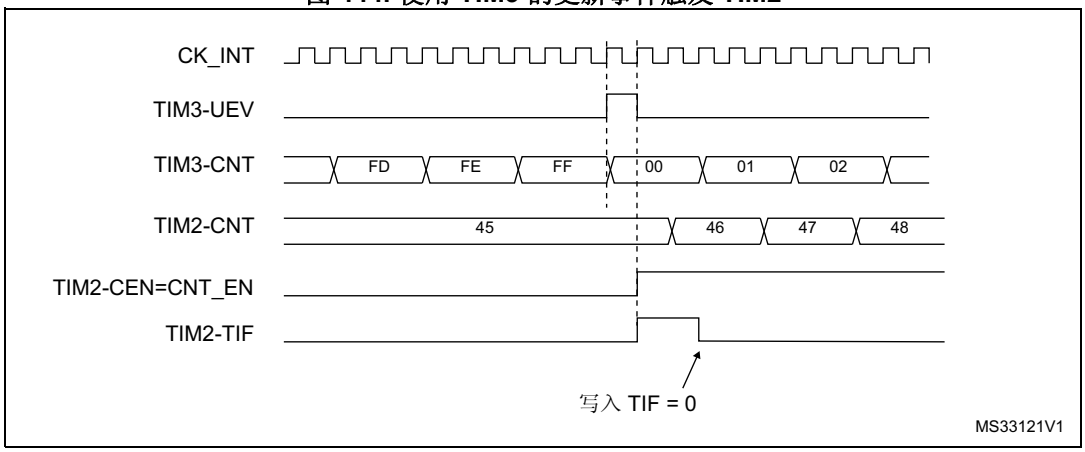


使用一个定时器启动另一个定时器

本例中，使用定时器 3 的更新事件使能定时器 2。相关连接图，请参见*图 441*。只要定时器 3 生成更新事件，定时器 2 便根据分频后的内部时钟从当前值（可以不为 0）开始计数。定时器 2 收到触发信号时，其 CEN 位自动置 1，并且计数器开始计数，直到向 TIM2_CR1 寄存器的 CEN 位写入“0”后停止计数。两个计数器的时钟频率都基于 CK_INT 通过预分频器执行 3 分频 ($f_{CK_CNT} = f_{CK_INT}/3$)。

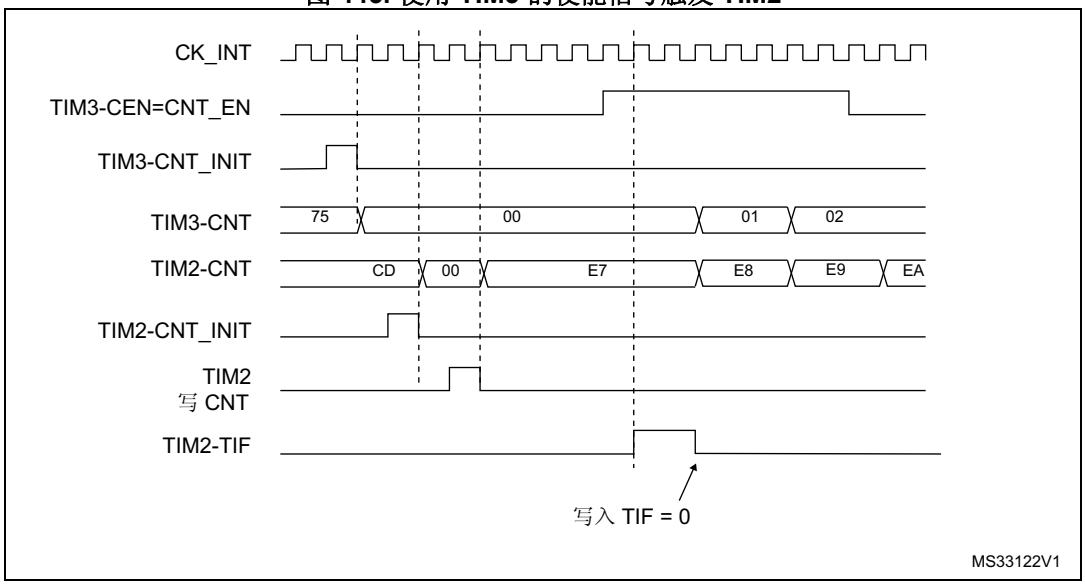
1. 将 TIM3 配置为主模式，发送其更新事件 (UEV) 作为触发输出（TIM3_CR2 寄存器中的 MMS=010）。
2. 配置 TIM3 的周期（TIM3_ARR 寄存器）。
3. 配置 TIM2 以接收来自 TIM3 的输入触发（TIM2_SMCR 寄存器中的 TS=00010）。
4. 将 TIM2 配置为触发模式（TIM2_SMCR 寄存器中的 SMS=110）。
5. 通过向 CEN 位（TIM3_CR1 寄存器）写入“1”启动 TIM3。

图 444. 使用 TIM3 的更新事件触发 TIM2



如上述示例所示，用户可以在开始计数之前初始化两个计数器。*图 445* 显示了与*图 444* 具有相同配置，只不过处于触发模式（TIM2_SMCR 寄存器中的 SMS=110）而非门控模式的计数行为。

图 445. 使用 TIM3 的使能信号触发 TIM2



使用一个外部触发同步的启动 2 个定时器

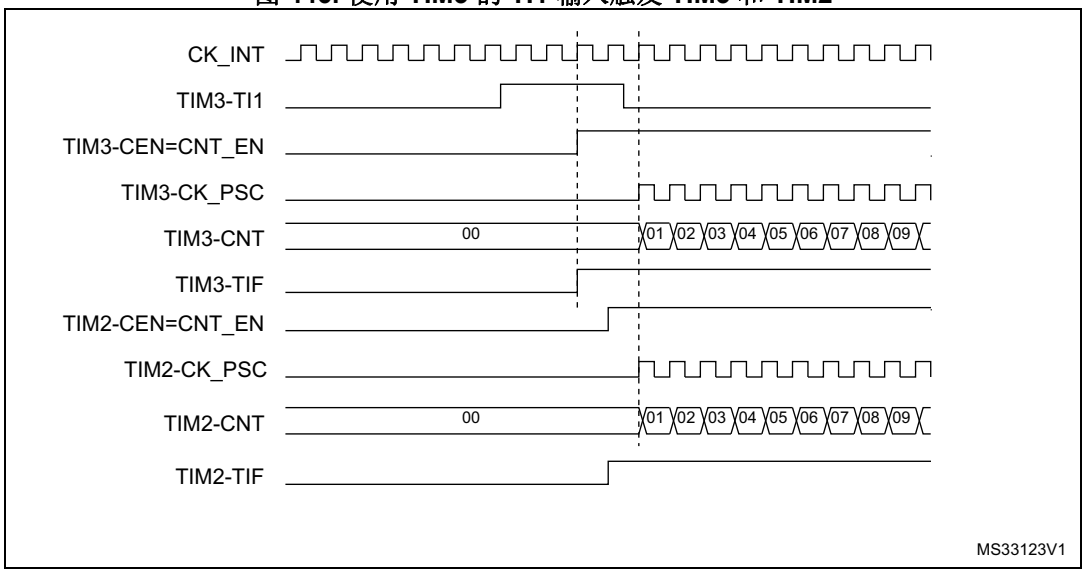
本例中，TIM3 的 TI1 输入出现上升沿时使能 TIM3，使能 TIM3 的同时使能 TIM2。相关连接图，请参见*图 441*。要确保计数器对齐，TIM3 必须配置为主/从模式（对于TI1而言，TIM3为从；对于TIM2 而言，TIM3 为主）：

1. 将 TIM3 配置为主模式，发送其使能信号作为触发输出（TIM3_CR2 寄存器中的 MMS=001）。
2. 将 TIM3 配置为从模式以接收来自 TI1 的输入触发（TIM3_SMCR 寄存器中的 TS=00100）。
3. 将 TIM3 配置为触发模式（TIM3_SMCR 寄存器中的 SMS=110）。
4. 通过写入 MSM=1（TIM3_SMCR 寄存器）将 TIM3 配置为主/从模式。
5. 配置 TIM2 以接收来自 TIM3 的输入触发（TIM2_SMCR 寄存器中的 TS=00000）。
6. 将 TIM2 配置为触发模式（TIM2_SMCR 寄存器中的 SMS=110）。

当 TI1 (TIM3) 出现上升沿时，两个计数器开始根据内部时钟同步计数，并且两个 TIF 标志都置 1。

注： *本例中，两个定时器都在启动之前进行了初始化（通过将各自的 UG 位置 1）。两个计数器都从 0 开始计数，但可以通过对任意一个计数器寄存器 (TIMx_CNT) 进行写操作，在二者之间轻松插入一个偏移量。可注意到主/从模式在 TIM3 的 CNT_EN 与 CK_PSC 之间产生了延迟。*

图 446. 使用 TIM3 的 TI1 输入触发 TIM3 和 TIM2



注： *必须先使能从定时器的时钟，才能从主定时器接收事件；从主定时器接收触发信号时，不得实时更改从定时器的时钟。*

39.3.20 DMA 连续传送模式

TIMx 定时器能够根据一个事件生成多个 DMA 请求。主要目的是能够对定时器的一部分多次重新编程而无需软件开销，但也可用于定期读取一行中的多个寄存器。

DMA 控制器目标唯一，必须指向虚拟寄存器 TIMx_DMAR。发生给定的定时器事件时，定时器会启动 DMA 请求序列（突发）。每次写入 TIMx_DMAR 寄存器都会重定向到其中一个定时器寄存器。

TIMx_DCR 寄存器中的 DBL[4:0] 位设置 DMA 连续传送长度。当对 TIMx_DMAR 地址进行读或写访问时，定时器进行一次连续传送，即传送次数（按半字或字节）。

TIMx_DCR 寄存器中的 DBA[4:0] 位定义 DMA 传送的 DMA 基址（通过 TIMx_DMAR 地址执行读/写访问时）。DBA 定义为从 TIMx_CR1 寄存器地址开始计算的偏移量：

示例：

00000：TIMx_CR1

00001：TIMx_CR2

00010：TIMx_SMCR

例如，定时器 DMA 连续传送功能用于在发生更新事件后将 CCRx 寄存器（x = 2、3、4）的内容更新为通过 DMA 传输到 CCRx 寄存器中的多个半字。

具体操作步骤如下：

1. 将相应的 DMA 通道配置如下：
   – DMA 通道外设地址为 DMAR 寄存器地址。
   – DMA 通道存储器地址为包含要通过 DMA 传输到 CCRx 寄存器的数据的 RAM 缓冲区地址。
   – 要传输的数据量 = 3（参见下文注释）。
   – 禁止循环模式。
2. 通过将 DBA 和 DBL 位域配置如下来配置 DCR 寄存器：
   DBL = 3 次传输，DBA = 0xE。
3. 使能 TIMx 更新 DMA 请求（DIER 寄存器中的 UDE 位置 1）。
4. 使能 TIMx
5. 使能 DMA 通道

本例适用于每个 CCRx 寄存器必须更新一次的情况。如果每个 CCRx 寄存器要更新两次，则要传输的数据量应为 6。下面以包含 data1、data2、data3、data4、data5 和 data6 的 RAM 缓冲区为例。数据将按照如下方式传输到 CCRx 寄存器：在第一个更新 DMA 请求期间，data1 传输到 CCR2，data2 传输到 CCR3，data3 传输到 CCR4；在第二个更新 DMA 请求期间，data4 传输到 CCR2，data5 传输到 CCR3，data6 传输到 CCR4。

注： *可以将空值写入保留的寄存器中。*

39.3.21 调试模式

当微控制器进入调试模式时（带 FPU 的 Cortex®-M7 内核停止），TIMx 计数器会根据 DBGMCU 模块中的 TIMx 配置位选择继续正常工作或者停止工作。有关详细信息，请参见 *第 60.5.8 节：微控制器调试单元 (DBGMCU)*。

为了安全起见，当计数器停止（DBGMCU_APB1FZ2 中的 TIMx = 1）时，输出被禁止（就像 MOE 位被复位一样）。可以将输出强制变为未激活状态（OSSI 位 = 1），或者通过 GPIO 控制器（OSSI 位 = 0）来控制输出，以将其强制为高阻态。

39.4 TIM2/TIM3/TIM4/TIM5 寄存器

有关寄存器说明中使用的缩写，请参见*第 1.1 节*。

外设寄存器可支持半字（16 位）或字（32 位）访问。

39.4.1 TIMx 控制寄存器 1 (TIMx_CR1)

TIMx control register 1

偏移地址：0x00

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	UIF RE-MAP	Res.	CKD[1:0]		ARPE	CMS		DIR	OPM	URS	UDIS	CEN
				rw		rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 15:12 保留，必须保持复位值。

位 11 **UIFREMAP**：UIF 状态位重映射 (UIF status bit remapping)
0：无重映射。UIF 状态位不复制到 TIMx_CNT 寄存器的位 31。
1：使能重映射。UIF 状态位复制到 TIMx_CNT 寄存器的位 31。

位 10 保留，必须保持复位值。

位 9:8 **CKD**：时钟分频 (Clock division)
此位域指示定时器时钟 (CK_INT) 频率与数字滤波器所使用的采样时钟（ETR、TIx）之间的分频比
00：$t_{DTS} = t_{CK_INT}$
01：$t_{DTS} = 2 \times t_{CK_INT}$
10：$t_{DTS} = 4 \times t_{CK_INT}$
11：保留

位 7 **ARPE**：自动重载预装载使能 (Auto-reload preload enable)
0：TIMx_ARR 寄存器不进行缓冲
1：TIMx_ARR 寄存器进行缓冲

位 6:5 **CMS**：中心对齐模式选择 (Center-aligned mode selection)
00：边沿对齐模式。计数器根据方向位 (DIR) 递增计数或递减计数。
01：中心对齐模式 1。计数器交替进行递增计数和递减计数。仅当计数器递减计数时，配置为输出的通道（TIMx_CCMRx 寄存器中的 CxS=00）的输出比较中断标志才置 1。
10：中心对齐模式 2。计数器交替进行递增计数和递减计数。仅当计数器递增计数时，配置为输出的通道（TIMx_CCMRx 寄存器中的 CxS=00）的输出比较中断标志才置 1。
11：中心对齐模式 3。计数器交替进行递增计数和递减计数。当计数器递增计数或递减计数时，配置为输出的通道（TIMx_CCMRx 寄存器中的 CxS=00）的输出比较中断标志都会置 1。
注：　只要计数器处于使能状态 (CEN=1)，就不得从边沿对齐模式切换为中心对齐模式。

位 4 **DIR**：方向 (Direction)
0：计数器递增计数
1：计数器递减计数
注：　当定时器配置为中心对齐模式或编码器模式时，该位为只读状态。

位 3 **OPM**：单脉冲模式 (One-pulse mode)

0：计数器在发生更新事件时不会停止计数

1：计数器在发生下一更新事件时停止计数（将 CEN 位清零）

位 2 **URS**：更新请求源 (Update request source)

此位由软件置 1 和清零，用以选择 UEV 事件源。

0：使能时，所有以下事件都会产生更新中断或 DMA 请求。此类事件包括：

– 计数器上溢/下溢

– 将 UG 位置 1

– 通过从模式控制器生成的更新事件

1：使能时，只有计数器上溢/下溢会生成更新中断或 DMA 请求。

位 1 **UDIS**：更新禁止 (Update disable)

此位由软件置 1 和清零，用以使能/禁止 UEV 事件生成。

0：使能 UEV。更新 (UEV) 事件可通过以下事件之一生成：

– 计数器上溢/下溢

– 将 UG 位置 1

– 通过从模式控制器生成的更新事件

然后更新影子寄存器的值。

1：禁止 UEV。不会生成更新事件，各影子寄存器的值（ARR、PSC 和 CCRx）保持不变。但如果将 UG 位置 1，或者从从模式控制器接收到硬件复位，则会重新初始化计数器和预分频器。

位 0 **CEN**：计数器使能 (Counter enable)

0：禁止计数器

1：使能计数器

注： 只有事先通过软件将 CEN 位置 1，才可以使用外部时钟、门控模式和编码器模式。而触发模式可通过硬件自动将 CEN 位置 1。

在单脉冲模式下，当发生更新事件时会自动将 CEN 位清零。

39.4.2 TIMx 控制寄存器 2 (TIMx_CR2)

TIMx control register 2

偏移地址：0x04

复位值：0x0000

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

位 15:8 保留，必须保持复位值。

位 7 **TI1S**：TI1 选择 (TI1 selection)

0：TIMx_CH1 引脚连接到 TI1 输入

1：TIMx_CH1、CH2 和 CH3 引脚连接到 TI1 输入（异或组合）。另请参见*第 1439 页的第 38.3.25 节：连接霍尔传感器*

位 6:4 **MMS**：主模式选择 (Master mode selection)

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

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

001：**使能**——计数器使能信号 CNT_EN 用作触发输出 (TRGO)。该触发输出可用于同时启动多个定时器，或者控制在一段时间内使能从定时器。计数器使能信号由 CEN 控制位与门控模式下的触发输入的逻辑或运算组合而成。

当计数器使能信号由触发输入控制时，TRGO 上会存在延迟，选择主/从模式时除外（请参见 TIMx_SMCR 寄存器中 MSM 位的说明）。

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

011：**比较脉冲**——一旦发生输入捕获或比较匹配事件，当 CC1IF 标志被置 1 时（即使已为高电平），触发输出都会发送一个正脉冲。(TRGO)

100：**比较**——OC1REF 信号用作触发输出 (TRGO)

101：**比较**——OC2REF 信号用作触发输出 (TRGO)

110：**比较**——OC3REF 信号用作触发输出 (TRGO)

111：**比较**——OC4REF 信号用作触发输出 (TRGO)

注：　必须先使能从定时器或 ADC 的时钟，才能从主定时器接收事件；并且从主定时器接收触发信号时，不得实时更改从定时器或 ADC 的时钟。

位 3 **CCDS**：捕获/比较 DMA 选择 (Capture/compare DMA selection)

0：发生 CCx 事件时发送 CCx DMA 请求

1：发生更新事件时发送 CCx DMA 请求

位 2:0 保留，必须保持复位值。

39.4.3 TIMx 从模式控制寄存器 (TIMx_SMCR)

TIMx slave mode control register

偏移地址：0x08

复位值：0x0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TS[4:3]		Res.	Res.	Res.	SMS[3]
										rw	rw				rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ETP	ECE	ETPS[1:0]		ETF[3:0]				MSM	TS[2:0]			Res.	SMS[2:0]		
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw		rw	rw	rw

位 31:22 保留，必须保持复位值。

位 21:20 **TS[4:3]**：触发选择——位 4:3 (Trigger selection - bit 4:3)
请参见 TS[2:0] 说明——位 6:4。

位 19:17 保留，必须保持复位值。

位 16 **SMS[3]**：从模式选择——位 3 (Slave mode selection - bit 3)
请参见 SMS 说明——位 2:0

位 15 **ETP**：外部触发极性 (External trigger polarity)
此位可选择将 ETR 还是 $\overline{ETR}$ 用于触发操作
0：ETR 未反相，高电平或上升沿有效
1：ETR 反相，低电平或下降沿有效

位 14 **ECE**：外部时钟使能 (External clock enable)
此位可使能外部时钟模式 2。
0：禁止外部时钟模式 2。
1：使能外部时钟模式 2。计数器时钟由 ETRF 信号的任意有效边沿提供。
1：将 ECE 位置 1 与选择外部时钟模式 1 并将 TRGI 连接到 ETRF（SMS=111 且 TS=00111）具有相同效果。
2：外部时钟模式 2 可以和以下从模式同时使用：复位模式、门控模式和触发模式。不过此类情况下 TRGI 不得连接 ETRF（TS 位不得为 00111）。
3：如果同时使能外部时钟模式 1 和外部时钟模式 2，则外部时钟输入为 ETRF。

位 13:12 **ETPS[1:0]**：外部触发预分频器 (External trigger prescaler)
外部触发信号 ETRP 频率不得超过 CK_INT 频率的 1/4。可通过使能预分频器来降低 ETRP 频率。这种方法在输入快速外部时钟时非常有用。
00：预分频器关闭
01：2 分频 ETRP 频率
10：4 分频 ETRP 频率
11：8 分频 ETRP 频率

位 11:8 **ETF[3:0]**：外部触发滤波器 (External trigger filter)

此位域可定义 ETRP 信号的采样频率和适用于 ETRP 的数字滤波器带宽。数字滤波器由事件计数器组成，每 N 个连续事件才视为一个有效输出边沿：

0000：无滤波器，按 f_{DTS} 频率进行采样
0001： $f_{SAMPLING}=f_{CK_INT}$，N=2
0010： $f_{SAMPLING}=f_{CK_INT}$，N=4
0011： $f_{SAMPLING}=f_{CK_INT}$，N=8
0100： $f_{SAMPLING}=f_{DTS}/2$，N=6
0101： $f_{SAMPLING}=f_{DTS}/2$，N=8
0110： $f_{SAMPLING}=f_{DTS}/4$，N=6
0111： $f_{SAMPLING}=f_{DTS}/4$，N=8
1000： $f_{SAMPLING}=f_{DTS}/8$，N=6
1001： $f_{SAMPLING}=f_{DTS}/8$，N=8
1010： $f_{SAMPLING}=f_{DTS}/16$，N=5
1011： $f_{SAMPLING}=f_{DTS}/16$，N=6
1100： $f_{SAMPLING}=f_{DTS}/16$，N=8
1101： $f_{SAMPLING}=f_{DTS}/32$，N=5
1110： $f_{SAMPLING}=f_{DTS}/32$，N=6
1111： $f_{SAMPLING}=f_{DTS}/32$，N=8

位 7 **MSM**：主/从模式 (Master/slave mode)

0：不执行任何操作
1：当前定时器的触发输入事件 (TRGI) 的动作被推迟，以使当前定时器与其从定时器实现完美同步（通过 TRGO）。此设置适用于由单个外部事件对多个定时器进行同步的情况。

位 6:4 **TS[4:0]**：触发选择 (Trigger selection)（请参见 TS[4:3] 的位 21:20）

此位域可选择将要用于同步计数器的触发输入。

00000：内部触发 0 (ITR0)
00001：内部触发 1 (ITR1)
00010：内部触发 2 (ITR2)
00011：内部触发 3 (ITR3)
00100：TI1 边沿检测器 (TI1F_ED)
00101：滤波后的定时器输入 1 (TI1FP1)
00110：滤波后的定时器输入 2 (TI2FP2)
00111：外部触发输入 (ETRF)
01000：内部触发 4 (ITR4)
01001：内部触发 5 (ITR5)
01010：内部触发 6 (ITR6)
01011：内部触发 7 (ITR7)
01100：内部触发 8 (ITR8)
其它：保留

有关各定时器 ITRx 含义的详细信息，请参见*第 1538 页的表 317：TIMx 内部触发连接*。

注： 这些位只能在未使用的情况下（例如，SMS=000 时）进行更改，以避免转换时出现错误的边沿检测。

位 3 保留，必须保持复位值。

位 2:0 **SMS**：从模式选择 (Slave mode selection)

选择外部信号时，触发信号 (TRGI) 的有效边沿与外部输入上所选的极性相关（请参见输入控制寄存器和控制寄存器说明）。

0000：禁止从模式——如果 CEN = “1”，预分频器时钟直接由内部时钟提供。

0001：编码器模式 1——计数器根据 TI2FP2 电平在 TI1FP1 边沿递增/递减计数。

0010：编码器模式 2——计数器根据 TI1FP1 电平在 TI2FP2 边沿递增/递减计数。

0011：编码器模式 3——计数器在 TI1FP1 和 TI2FP2 的边沿计数，计数的方向取决于另外一个输入的电平。

0100：复位模式——在出现所选触发输入 (TRGI) 上升沿时，重新初始化计数器并生成一个寄存器更新事件。

0101：门控模式——触发输入 (TRGI) 为高电平时使能计数器时钟。只要触发输入变为低电平，计数器立即停止计数（但不复位）。计数器的启动和停止都被控制。

0110：触发模式——触发信号 TRGI 出现上升沿时启动计数器（但不复位）。只控制计数器的启动。

0111：外部时钟模式 1——由所选触发信号 (TRGI) 的上升沿提供计数器时钟。

1000：组合复位 + 触发模式——在出现所选触发输入 (TRGI) 上升沿时，重新初始化计数器，生成一个寄存器更新事件并启动计数器。

注： 如果将 TI1F_ED 选作触发输入 (TS=00100)，则不得使用门控模式。实际上，TI1F 每次转换时，TI1F_ED 都输出 1 个脉冲，而门控模式检查的则是触发信号的电平。

注： 必须先使能从定时器的时钟，才能从主定时器接收事件；从主定时器接收触发信号时，不得实时更改从定时器的时钟。

表 317. TIMx 内部触发连接

从 TIM	TIM2	TIM3	TIM4	TIM5
ITR0	TIM1	TIM1	TIM1	TIM1
ITR1	TIM8	TIM2	TIM2	TIM8
ITR2	TIM3	TIM15	TIM3	TIM3
ITR3	TIM4	TIM4	TIM8	TIM4
ITR4	ETH PPS	ETH PPS	-	-
ITR5	USB1 OTG_HS_SOF	-	-	-
ITR6	USB2 OTG_FS_SOF	-	-	fdcan1_soc
ITR7	-	-	-	USB1 OTG_HS_SOF
ITR8	-	-	-	USB2 OTG_FS_SOF

39.4.4 TIMx DMA/中断使能寄存器 (TIMx_DIER)

TIMx DMA/Interrupt enable register

偏移地址：0x0C

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	TDE	Res.	CC4DE	CC3DE	CC2DE	CC1DE	UDE	Res.	TIE	Res.	CC4IE	CC3IE	CC2IE	CC1IE	UIE
	rw		rw	rw	rw	rw	rw		rw		rw	rw	rw	rw	rw

位 15 保留，必须保持复位值。

位 14 **TDE**：触发 DMA 请求使能 (Trigger DMA request enable)
0：禁止触发 DMA 请求。
1：使能触发 DMA 请求。

位 13 保留，必须保持复位值。

位 12 **CC4DE**：捕获/比较 4 DMA 请求使能 (Capture/Compare 4 DMA request enable)
0：禁止 CC4 DMA 请求。
1：使能 CC4 DMA 请求。

位 11 **CC3DE**：捕获/比较 3 DMA 请求使能 (Capture/Compare 3 DMA request enable)
0：禁止 CC3 DMA 请求。
1：使能 CC3 DMA 请求。

位 10 **CC2DE**：捕获/比较 2 DMA 请求使能 (Capture/Compare 2 DMA request enable)
0：禁止 CC2 DMA 请求。
1：使能 CC2 DMA 请求。

位 9 **CC1DE**：捕获/比较 1 DMA 请求使能 (Capture/Compare 1 DMA request enable)
0：禁止 CC1 DMA 请求。
1：使能 CC1 DMA 请求。

位 8 **UDE**：更新 DMA 请求使能 (Update DMA request enable)
0：禁止更新 DMA 请求。
1：使能更新 DMA 请求。

位 7 保留，必须保持复位值。

位 6 **TIE**：触发中断使能 (Trigger interrupt enable)
0：禁止触发中断。
1：使能触发中断。

位 5 保留，必须保持复位值。

位 4 **CC4IE**：捕获/比较 4 中断使能 (Capture/Compare 4 interrupt enable)
0：禁止 CC4 中断。
1：使能 CC4 中断。

位 3 **CC3IE**：捕获/比较 3 中断使能 (Capture/Compare 3 interrupt enable)
0：禁止 CC3 中断。
1：使能 CC3 中断。

位 2 **CC2IE**：捕获/比较 2 中断使能 (Capture/Compare 2 interrupt enable)
0：禁止 CC2 中断。
1：使能 CC2 中断。

位 1 **CC1IE**：捕获/比较 1 中断使能 (Capture/Compare 1 interrupt enable)
0：禁止 CC1 中断。
1：使能 CC1 中断。

位 0 **UIE**：更新中断使能 (Update interrupt enable)
0：禁止更新中断。
1：使能更新中断。

39.4.5 TIMx 状态寄存器 (TIMx_SR)

TIMx status register

偏移地址：0x10

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	CC4OF	CC3OF	CC2OF	CC1OF	Res.	Res.	TIF	Res.	CC4IF	CC3IF	CC2IF	CC1IF	UIF
			rc_w0	rc_w0	rc_w0	rc_w0			rc_w0		rc_w0	rc_w0	rc_w0	rc_w0	rc_w0

位 15:13 保留，必须保持复位值。

位 12 **CC4OF**：捕获/比较 4 重复捕获标志 (Capture/Compare 4 overcapture flag)
请参见 CC1OF 说明

位 11 **CC3OF**：捕获/比较 3 重复捕获标志 (Capture/Compare 3 overcapture flag)
请参见 CC1OF 说明

位 10 **CC2OF**：捕获/比较 2 重复捕获标志 (Capture/compare 2 overcapture flag)
请参见 CC1OF 说明

位 9 **CC1OF**：捕获/比较 1 重复捕获标志 (Capture/Compare 1 overcapture flag)
仅当将相应通道配置为输入捕获模式时，此标志位才会由硬件置 1。通过软件写入“0”可将该位清零。
0：未检测到重复捕获。
1：TIMx_CCR1 寄存器中已捕获到计数器值且 CC1IF 标志已置 1。

位 8:7 保留，必须保持复位值。

位 6 **TIF**：触发中断标志 (Trigger interrupt flag)
在除门控模式以外的所有模式下，当使能从模式控制器后在 TRGI 输入上检测到有效边沿时，该标志将由硬件置 1。选择门控模式时，该标志将在计数器启动或停止时置 1。但需要通过软件清零。
0：未发生触发事件。
1：触发中断挂起。

位 5 保留，必须保持复位值。

位 4 **CC4IF**：捕获/比较 4 中断标志 (Capture/Compare 4 interrupt flag)
请参见 CC1IF 说明

位 3 **CC3IF**：捕获/比较 3 中断标志 (Capture/Compare 3 interrupt flag)
请参见 CC1IF 说明

位 2 **CC2IF**：捕获/比较 2 中断标志 (Capture/Compare 2 interrupt flag)
请参见 CC1IF 说明

位 1 **CC1IF**：捕获/比较 1 中断标志 (Capture/compare 1 interrupt flag)
如果通道 CC1 配置为输出：当计数器与比较值匹配时，此标志由硬件置 1，中心对齐模式（请参见 TIMx_CR1 寄存器中的 CMS 位说明）和可再触发单脉冲模式下除外。但需要通过软件清零。
0：不匹配。
1：TIMx_CNT 计数器的内容与 TIMx_CCR1 寄存器的内容匹配。
如果通道 CC1 配置为输入：此位将在发生捕获事件时由硬件置 1。通过软件或读取 TIMx_CCR1 寄存器将该位清零。
0：未发生输入捕获事件。
1：TIMx_CCR1 寄存器中已捕获到计数器值（IC1 上已检测到与所选极性匹配的边沿）。

位 0 **UIF**：更新中断标志 (Update interrupt flag)
该位在发生更新事件时通过硬件置 1。但需要通过软件清零。
0：未发生更新
1：更新中断挂起。该位在以下情况下更新寄存器时由硬件置 1：
上溢或下溢（对于 TIM2 到 TIM4）以及当 TIMx_CR1 寄存器中的 UDIS = 0 时。
TIMx_CR1 寄存器中的 URS = 0 且 UDIS = 0，并且由软件使用 TIMx_EGR 寄存器中的 UG 位重新初始化 CNT 时。
TIMx_CR1 寄存器中的 URS=0 且 UDIS=0，并且 CNT 由触发事件重新初始化时（参见同步控制寄存器说明）。

39.4.6 TIMx 事件生成寄存器 (TIMx_EGR)

TIMx event generation register

偏移地址：0x14

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TG	Res.	CC4G	CC3G	CC2G	CC1G	UG
									w		w	w	w	w	w

位 15:7 保留，必须保持复位值。

位 6 **TG**：触发生成 (Trigger generation)
此位由软件置 1 以生成事件，并由硬件自动清零。
0：不执行任何操作
1：TIMx_SR 寄存器中的 TIF 标志置 1。使能后可发生相关中断或 DMA 传输事件

位 5 保留，必须保持复位值。

位 4 **CC4G**：捕获/比较 4 生成 (Capture/Compare 4 generation)
请参见 CC1G 说明

位 3 **CC3G**：捕获/比较 3 生成 (Capture/Compare 3 generation)
请参见 CC1G 说明

位 2 **CC2G**：捕获/比较 2 生成 (Capture/compare 2 generation)
请参见 CC1G 说明

位 1 **CC1G**：捕获/比较 1 生成 (Capture/Compare 1 generation)

此位由软件置 1 以生成事件，并由硬件自动清零。

0：不执行任何操作

1：通道 1 上生成捕获/比较事件：

如果通道 CC1 配置为输出：

使能时，CC1IF 标志置 1 并发送相应的中断或 DMA 请求。

如果通道 CC1 配置为输入：

TIMx_CCR1 寄存器中将捕获到计数器当前值。使能时，CC1IF 标志置 1 并发送相应的中断或 DMA 请求。如果 CC1IF 标志已为高，CC1OF 标志将置 1。

位 0 **UG**：更新生成 (Update generation)

该位可通过软件置 1，并由硬件自动清零。

0：不执行任何操作

1：重新初始化计数器并生成寄存器更新事件。请注意，预分频器计数器也将清零（但预分频比不受影响）。如果选择中心对齐模式或 DIR=0（递增计数），计数器将清零；如果 DIR=1（递减计数），计数器将使用自动重载值 (TIMx_ARR)。

39.4.7 TIMx 捕获/比较模式寄存器 1 (TIMx_CCMR1)

TIMx capture/compare mode register 1

偏移地址：0x18

复位值：0x0000

这些通道可用于输入（捕获模式）或输出（比较模式）模式。通道方向通过配置相应的 CCxS 位进行定义。此寄存器的所有其它位在输入模式和输出模式下的功能均不同。对于任一给定位，OCxx 用于说明通道配置为输出时该位对应的功能，ICxx 则用于说明通道配置为输入时该位对应的功能。因此，必须注意同一个位在输入阶段和输出阶段具有不同的含义。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC2M [3]	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC1M [3]
							Res.								Res.
							rw								rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OC2CE	OC2M[2:0]			OC2PE	OC2FE	CC2S[1:0]		OC1CE	OC1M[2:0]			OC1PE	OC1FE	CC1S[1:0]	
IC2F[3:0]				IC2PSC[1:0]				IC1F[3:0]				IC1PSC[1:0]			
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

输出比较模式

位 31:25 保留，始终读为 0。

位 24 **OC2M[3]**：输出比较 2 模式——位 3 (Output Compare 2 mode - bit 3)

位 23:17 保留，始终读为 0。

位 16 **OC1M[3]**：输出比较 1 模式——位 3 (Output Compare 1 mode - bit 3)

位 15 **OC2CE**：输出比较 2 清零使能 (Output Compare 2 clear enable)

位 14:12 **OC2M[2:0]**：输出比较 2 模式 (Output Compare 2 mode)

请参见 OC1M 说明——位 6:4

位 11 **OC2PE**：输出比较 2 预装载使能 (Output Compare 2 preload enable)

位 10 **OC2FE**：输出比较 2 快速使能 (Output Compare 2 fast enable)

位 9:8 **CC2S[1:0]**：捕获/比较 2 选择 (Capture/Compare 2 selection)

此位域定义通道方向（输入/输出）以及所使用的输入。

00：CC2 通道配置为输出

01：CC2 通道配置为输入，IC2 映射到 TI2 上

10：CC2 通道配置为输入，IC2 映射到 TI1 上

11：CC2 通道配置为输入，IC2 映射到 TRC 上。此模式仅在通过 TS 位（TIMx_SMCR 寄存器）选择内部触发输入时有效

注： 仅当通道关闭时（TIMx_CCER 中的 CC2E = 0），才可向 CC2S 位写入数据。

位 7 **OC1CE**：输出比较 1 清零使能 (Output Compare 1 clear enable)

0：OC1Ref 不受 ETRF 输入影响

1：ETRF 输入上检测到高电平时，OC1Ref 立即清零

位 6:4 **OC1M**：输出比较 1 模式 (Output Compare 1 mode)

这些位定义提供 OC1 和 OC1N 的输出参考信号 OC1REF 的行为。OC1REF 为高电平有效，而 OC1 和 OC1N 的有效电平则取决于 CC1P 位和 CC1NP 位。

0000：冻结——输出比较寄存器 TIMx_CCR1 与计数器 TIMx_CNT 进行比较不会对输出造成任何影响。（该模式用于生成时基）。

0001：将通道 1 设置为匹配时输出有效电平。当计数器 TIMx_CNT 与捕获/比较寄存器 1 (TIMx_CCR1) 匹配时，OC1REF 信号强制变为高电平。

0010：将通道 1 设置为匹配时输出无效电平。当计数器 TIMx_CNT 与捕获/比较寄存器 1 (TIMx_CCR1) 匹配时，OC1REF 信号强制变为低电平。

0011：翻转——TIMx_CNT=TIMx_CCR1 时，OC1REF 发生翻转。

0100：强制变为无效电平——OC1REF 强制变为低电平。

0101：强制变为有效电平——OC1REF 强制变为高电平。

0110：PWM 模式 1——在递增计数模式下，只要 TIMx_CNT<TIMx_CCR1，通道 1 便为有效状态，否则为无效状态。在递减计数模式下，只要 TIMx_CNT>TIMx_CCR1，通道 1 便为无效状态 (OC1REF=0)，否则为有效状态 (OC1REF=1)。

0111：PWM 模式 2——在递增计数模式下，只要 TIMx_CNT<TIMx_CCR1，通道 1 便为无效状态，否则为有效状态。在递减计数模式下，只要 TIMx_CNT>TIMx_CCR1，通道 1 便为有效状态，否则为无效状态。

1000：可再触发 OPM 模式 1——在递增计数模式下，通道为有效状态，直至（在 TRGI 信号上）检测到触发事件。然后，在 PWM 模式 1 下进行比较，通道会在下一次更新时再次变为无效状态。在递减计数模式下，通道为无效状态，直至（在 TRGI 信号上）检测到触发事件。然后，在 PWM 模式 1 下进行比较，通道会在下一次更新时再次变为无效状态。

1001：可再触发 OPM 模式 2——在递增计数模式下，通道为无效状态，直至（在 TRGI 信号上）检测到触发事件。然后，在 PWM 模式 2 下进行比较，通道会在下一次更新时再次变为无效状态。在递减计数模式下，通道为有效状态，直至（在 TRGI 信号上）检测到触发事件。然后，在 PWM 模式 2 下进行比较，通道会在下一次更新时再次变为有效状态。

1010：保留。

1011：保留。

1100：组合 PWM 模式 1——OC1REF 与在 PWM 模式 1 下的行为相同。OC1REFC 是 OC1REF 和 OC2REF 的逻辑或运算结果。

1101：组合 PWM 模式 2——OC1REF 与在 PWM 模式 2 下的行为相同。OC1REFC 是 OC1REF 和 OC2REF 的逻辑与运算结果。

1110：不对称 PWM 模式 1——OC1REF 与在 PWM 模式 1 下的行为相同。计数器递增计数时，OC1REFC 输出 OC1REF；计数器递减计数时，OC1REFC 输出 OC2REF。

1111：不对称 PWM 模式 2——OC1REF 与在 PWM 模式 2 下的行为相同。计数器递增计数时，OC1REFC 输出 OC1REF；计数器递减计数时，OC1REFC 输出 OC2REF。

*注： **1**：在 PWM 模式下，仅当比较结果发生改变或输出比较模式由“冻结”模式切换到“PWM”模式时，OCREF 电平才会发生更改。*

位 3 **OC1PE**：输出比较 1 预装载使能 (Output Compare 1 preload enable)

0：禁止与 TIMx_CCR1 相关的预装载寄存器。可随时向 TIMx_CCR1 写入数据，写入后将立即使用新值。

1：使能与 TIMx_CCR1 相关的预装载寄存器。可读/写访问预装载寄存器。TIMx_CCR1 预装载值在每次生成更新事件时都会装载到有效寄存器中。

*注： **1**：只有单脉冲模式下才可在未验证预装载寄存器的情况下使用 PWM 模式（TIMx_CR1 寄存器中的 OPM 位置 1）。其它情况下则无法保证该行为。*

位 2 **OC1FE**：输出比较 1 快速使能 (Output Compare 1 fast enable)

此位用于加快触发输入事件对 CC 输出的影响。

0：即使触发开启，CC1 也将根据计数器和 CCR1 值正常工作。触发输入出现边沿时，激活 CC1 输出的最短延迟时间为 5 个时钟周期。

1：触发输入上出现有效边沿相当于 CC1 输出上的比较匹配。随后，无论比较结果如何，OC 都设置为比较电平。采样触发输入和激活 CC1 输出的延迟时间缩短为 3 个时钟周期。仅当通道配置为 PWM1 或 PWM2 模式时，OCFE 才会起作用。

位 1:0 **CC1S**：捕获/比较 1 选择 (Capture/Compare 1 Selection)

此位域定义通道方向（输入/输出）以及所使用的输入。

00：CC1 通道配置为输出。

01：CC1 通道配置为输入，IC1 映射到 TI1 上。

10：CC1 通道配置为输入，IC1 映射到 TI2 上。

11：CC1 通道配置为输入，IC1 映射到 TRC 上。此模式仅在通过 TS 位（TIMx_SMCR 寄存器）选择内部触发输入时有效

注： 仅当通道关闭时（TIMx_CCER 中的 CC1E = 0），才可向 CC1S 位写入数据。

输入捕获模式

位 31:16 保留，始终读为 0。

位 15:12 **IC2F**：输入捕获 2 滤波器 (Input capture 2 filter)

位 11:10 **IC2PSC[1:0]**：输入捕获 2 预分频器 (Input capture 2 prescaler)

位 9:8 **CC2S**：捕获/比较 2 选择 (Capture/Compare 2 selection)

此位域定义通道方向（输入/输出）以及所使用的输入。

00：CC2 通道配置为输出。

01：CC2 通道配置为输入，IC2 映射到 TI2 上。

10：CC2 通道配置为输入，IC2 映射到 TI1 上。

11：CC2 通道配置为输入，IC2 映射到 TRC 上。此模式仅在通过 TS 位（TIMx_SMCR 寄存器）选择内部触发输入时有效

注： 仅当通道关闭时（TIMx_CCER 中的 CC2E = 0），才可向 CC2S 位写入数据。

位 7:4 **IC1F**：输入捕获 1 滤波器 (Input capture 1 filter)

此位域可定义 TI1 输入的采样频率和适用于 TI1 的数字滤波器带宽。数字滤波器由事件计数器组成，每 N 个连续事件才视为一个有效输出边沿：

0000：无滤波器，按 f_{DTS} 频率进行采样
0001：$f_{SAMPLING}=f_{CK_INT}$，N=2
0010：$f_{SAMPLING}=f_{CK_INT}$，N=4
0011：$f_{SAMPLING}=f_{CK_INT}$，N=8
0100：$f_{SAMPLING}=f_{DTS}/2$，N=6
0101：$f_{SAMPLING}=f_{DTS}/2$，N=8
0110：$f_{SAMPLING}=f_{DTS}/4$，N=6
0111：$f_{SAMPLING}=f_{DTS}/4$，N=8
1000：$f_{SAMPLING}=f_{DTS}/8$，N=6
1001：$f_{SAMPLING}=f_{DTS}/8$，N=8
1010：$f_{SAMPLING}=f_{DTS}/16$，N=5
1011：$f_{SAMPLING}=f_{DTS}/16$，N=6
1100：$f_{SAMPLING}=f_{DTS}/16$，N=8
1101：$f_{SAMPLING}=f_{DTS}/32$，N=5
1110：$f_{SAMPLING}=f_{DTS}/32$，N=6
1111：$f_{SAMPLING}=f_{DTS}/32$，N=8

位 3:2 **IC1PSC**：输入捕获 1 预分频器 (Input capture 1 prescaler)

此位域定义 CC1 输入 (IC1) 的预分频比。只要 CC1E=0（TIMx_CCER 寄存器），预分频器便立即复位。

00：无预分频器，捕获输入上每检测到一个边沿便执行捕获
01：每发生 2 个事件便执行一次捕获
10：每发生 4 个事件便执行一次捕获
11：每发生 8 个事件便执行一次捕获

位 1:0 **CC1S**：捕获/比较 1 选择 (Capture/Compare 1 Selection)

此位域定义通道方向（输入/输出）以及所使用的输入。

00：CC1 通道配置为输出
01：CC1 通道配置为输入，IC1 映射到 TI1 上
10：CC1 通道配置为输入，IC1 映射到 TI2 上
11：CC1 通道配置为输入，IC1 映射到 TRC 上。此模式仅在通过 TS 位（TIMx_SMCR 寄存器）选择内部触发输入时有效

注：　仅当通道关闭时（TIMx_CCER 中的 CC1E = 0），才可向 CC1S 位写入数据。

39.4.8 TIMx 捕获/比较模式寄存器 2 (TIMx_CCMR2)

TIMx capture/compare mode register 2

偏移地址：0x1C

复位值：0x0000

请参见上述 CCMR1 寄存器说明。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC4M [3]	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC3M [3]
							Res.								Res.
							rw								rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OC4CE	OC4M[2:0]			OC4PE	OC4FE	CC4S[1:0]		OC3CE	OC3M[2:0]			OC3PE	OC3FE	CC3S[1:0]	
IC4F[3:0]				IC4PSC[1:0]				IC3F[3:0]				IC3PSC[1:0]			
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

输出比较模式

位 31:25 保留，始终读为 0。

位 24 **OC4M[3]**：输出比较 2 模式——位 3 (Output Compare 2 mode - bit 3)

位 23:17 保留，始终读为 0。

位 16 **OC3M[3]**：输出比较 1 模式——位 3 (Output Compare 1 mode - bit 3)

位 15 **OC4CE**：输出比较 4 清零使能 (Output compare 4 clear enable)

位 14:12 **OC4M**：输出比较 4 模式 (Output compare 4 mode)
请参见 OC1M 说明（TIMx_CCMR1 寄存器中的位 6:4）

位 11 **OC4PE**：输出比较 4 预装载使能 (Output compare 4 preload enable)

位 10 **OC4FE**：输出比较 4 快速使能 (Output compare 4 fast enable)

位 9:8 **CC4S**：捕获/比较 4 选择 (Capture/Compare 4 selection)
此位域定义通道方向（输入/输出）以及所使用的输入。
00：CC4 通道配置为输出
01：CC4 通道配置为输入，IC4 映射到 TI4 上
10：CC4 通道配置为输入，IC4 映射到 TI3 上
11：CC4 通道配置为输入，IC4 映射到 TRC 上。此模式仅在通过 TS 位（TIMx_SMCR 寄存器）选择内部触发输入时有效
注： 仅当通道关闭时（TIMx_CCER 中的 CC4E = 0），才可向 CC4S 位写入数据。

位 7 **OC3CE**：输出比较 3 清零使能 (Output compare 3 clear enable)

位 6:4 **OC3M**：输出比较 3 模式 (Output compare 3 mode)
请参见 OC1M 说明（TIMx_CCMR1 寄存器中的位 6:4）

位 3 **OC3PE**：输出比较 3 预装载使能 (Output compare 3 preload enable)

位 2 **OC3FE**：输出比较 3 快速使能 (Output compare 3 fast enable)

位 1:0 **CC3S**：捕获/比较 3 选择 (Capture/compare 3 selection)
此位域定义通道方向（输入/输出）以及所使用的输入。
00：CC3 通道配置为输出
01：CC3 通道配置为输入，IC3 映射到 TI3 上
10：CC3 通道配置为输入，IC3 映射到 TI4 上
11：CC3 通道配置为输入，IC3 映射到 TRC 上。此模式仅在通过 TS 位（TIMx_SMCR 寄存器）选择内部触发输入时有效
注： 仅当通道关闭时（TIMx_CCER 中的 CC3E = 0），才可向 CC3S 位写入数据。

输入捕获模式

位 31:16 保留，始终读为 0。

位 15:12 **IC4F**：输入捕获 4 滤波器 (Input capture 4 filter)

位 11:10 **IC4PSC**：输入捕获 4 预分频器 (Input capture 4 prescaler)

位 9:8 **CC4S**：捕获/比较 4 选择 (Capture/Compare 4 selection)
此位域定义通道方向（输入/输出）以及所使用的输入。
00：CC4 通道配置为输出
01：CC4 通道配置为输入，IC4 映射到 TI4 上
10：CC4 通道配置为输入，IC4 映射到 TI3 上
11：CC4 通道配置为输入，IC4 映射到 TRC 上。此模式仅在通过 TS 位（TIMx_SMCR 寄存器）选择内部触发输入时有效
注： 仅当通道关闭时（TIMx_CCER 中的 CC4E = 0），才可向 CC4S 位写入数据。

位 7:4 **IC3F**：输入捕获 3 滤波器 (Input capture 3 filter)

位 3:2 **IC3PSC**：输入捕获 3 预分频器 (Input capture 3 prescaler)

位 1:0 **CC3S**：捕获/比较 3 选择 (Capture/compare 3 selection)

此位域定义通道方向（输入/输出）以及所使用的输入。

00：CC3 通道配置为输出

01：CC3 通道配置为输入，IC3 映射到 TI3 上

10：CC3 通道配置为输入，IC3 映射到 TI4 上

11：CC3 通道配置为输入，IC3 映射到 TRC 上。此模式仅在通过 TS 位（TIMx_SMCR 寄存器）选择内部触发输入时有效

注： 仅当通道关闭时（TIMx_CCER 中的 CC3E = 0），才可向 CC3S 位写入数据。

39.4.9 TIMx 捕获/比较使能寄存器 (TIMx_CCER)

TIMx capture/compare enable register

偏移地址：0x20

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CC4NP	Res.	CC4P	CC4E	CC3NP	Res.	CC3P	CC3E	CC2NP	Res.	CC2P	CC2E	CC1NP	Res.	CC1P	CC1E
rw		rw	rw	rw		rw	rw	rw		rw	rw	rw		rw	rw

位 15 **CC4NP**：*捕获/比较 4 互补输出极性 (Capture/Compare 4 complementary output Polarity)*

请参见 CC1NP 说明

位 14 保留，必须保持复位值。

位 13 **CC4P**：*捕获/比较 4 输出极性 (Capture/Compare 4 output Polarity)*

请参见 CC1P 说明

位 12 **CC4E**：*捕获/比较 4 输出使能 (Capture/Compare 4 output enable)*

请参见 CC1E 说明

位 11 **CC3NP**：*捕获/比较 3 互补输出极性 (Capture/Compare 3 complementary output Polarity)*

请参见 CC1NP 说明

位 10 保留，必须保持复位值。

位 9 **CC3P**：*捕获/比较 3 输出极性 (Capture/Compare 3 output polarity)*

请参见 CC1P 说明

位 8 **CC3E**：*捕获/比较 3 输出使能 (Capture/Compare 3 output enable)*

请参见 CC1E 说明

位 7 **CC2NP**：*捕获/比较 2 互补输出极性 (Capture/Compare 2 complementary output Polarity)*

请参见 CC1NP 说明

位 6 保留，必须保持复位值。

位 5 **CC2P**：*捕获/比较 2 输出极性 (Capture/Compare 2 output polarity)*

请参见 CC1P 说明

位 4 **CC2E**：*捕获/比较 2 输出使能 (Capture/Compare 2 output enable)*

请参见 CC1E 说明

位 3 **CC1NP**：*捕获/比较 1 互补输出极性 (Capture/Compare 1 complementary output Polarity)*

CC1 通道配置为输出：在这种情况下，CC1NP 必须保持清零。

CC1 通道配置为输入：该位与 CC1P 配合使用可定义 TI1FP1/TI2FP1 极性。请参见 CC1P 说明。

位 2 保留，必须保持复位值。

位 1 **CC1P**：*捕获/比较 1 输出极性 (Capture/Compare 1 output polarity)*。

CC1 通道配置为输出：

0：OC1 高电平有效

1：OC1 低电平有效

CC1 通道配置为输入：CC1NP/CC1P 位可针对触发或捕获操作选择 TI1FP1 和 TI2FP1 的极性。

00：未反相/上升沿触发

电路对 TIxFP1 上升沿敏感（在复位模式、外部时钟模式或触发模式下执行捕获或触发操作），TIxFP1 未反相（在门控模式或编码器模式下执行触发操作）。

01：反相/下降沿触发

电路对 TIxFP1 下降沿敏感（在复位模式、外部时钟模式或触发模式下执行捕获或触发操作），TIxFP1 反相（在门控模式或编码器模式下执行触发操作）。

10：保留，不使用此配置。

11：未反相/上升沿和下降沿均触发

电路对 TIxFP1 上升沿和下降沿都敏感（在复位模式、外部时钟模式或触发模式下执行捕获或触发操作），TIxFP1 未反相（在门控模式下执行触发操作）。编码器模式下不得使用此配置。

位 0 **CC1E**：*捕获/比较 1 输出使能 (Capture/Compare 1 output enable)*。

CC1 通道配置为输出：

0：关闭——OC1 未激活

1：开启——在相应输出引脚上输出 OC1 信号

CC1 通道配置为输入：此位决定了是否可以实际将计数器值捕获到输入捕获/比较寄存器 1 (TIMx_CCR1) 中。

0：禁止捕获

1：使能捕获

表 318. 标准 OCx 通道的输出控制位

CCxE 位	OCx 输出状态
0	禁止输出（OCx=0、OCx_EN=0）
1	OCx=OCxREF + 极性、OCx_EN=1

注： *与标准 OCx 通道相连的外部 IO 引脚的状态取决于 OCx 通道的状态以及 GPIO 和 AFIO 寄存器。*

39.4.10 TIMx 计数器 (TIMx_CNT)

TIMx counter

偏移地址：0x24

复位值：0x0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CNT[31]或UIFCPY	CNT[30:16]（取决于定时器）														
rw or r	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CNT[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31 值取决于 TIMx_CR1 中的 IUFREMAP。

如果 UIFREMAP = 0

CNT[31]：计数器值的最高有效位（对于 TIM2 和 TIM5）(Most significant bit of counter value (on TIM2 and TIM5))

其它定时器上保留

如果 UIFREMAP = 1

UIFCPY：UIF 副本 (UIF Copy)

该位是 TIMx_ISR 寄存器中 UIF 位的只读副本

位 30:16 **CNT[30:16]**：计数器值的最高有效部分（对于 TIM2 和 TIM5）(Most significant part counter value (on TIM2 and TIM5))

位 15:0 **CNT[15:0]**：计数器值的最低有效部分 (Least significant part of counter value)

39.4.11 TIMx 预分频器 (TIMx_PSC)

TIMx prescaler

偏移地址：0x28

复位值：0x0000

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

位 15:0 **PSC[15:0]**：预分频器值 (Prescaler value)

计数器时钟频率 CK_CNT 等于 f_{CK_PSC} / (PSC[15:0] + 1)。

PSC 包含每次发生更新事件（包括计数器通过 TIMx_EGR 寄存器中的 UG 位清零时，或在配置为“复位模式”时通过触发控制器清零时）时要装载到有效预分频器寄存器的值。

39.4.12 TIMx 自动重载寄存器 (TIMx_ARR)

TIMx auto-reload register

偏移地址：0x2C

复位值：0xFFFF FFFF

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ARR[31:16]（取决于定时器）															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ARR[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:16 **ARR[31:16]**：自动重载值的高 16 位（对于 TIM2 和 TIM5）(High auto-reload value (on TIM2 and TIM5))

位 15:0 **ARR[15:0]**：自动重载值的低 16 位 (Low Auto-reload value)

ARR 为要装载到有效自动重载寄存器的值。

有关 ARR 更新和行为的更多详细信息，请参见*第 1492 页的第 39.3.1 节：时基单元*。

当自动重载值为空时，计数器不工作。

39.4.13 TIMx 捕获/比较寄存器 1 (TIMx_CCR1)

TIMx capture/compare register 1

偏移地址：0x34

复位值：0x0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CCR1[31:16]（取决于定时器）															
rw/r	rw/r	rw/r	rw/r	rw/r	rw/r	rw/r	rw/r	rw/r	rw/r	rw/r	rw/r	rw/r	rw/r	rw/r	rw/r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR1[15:0]															
rw/r	rw/r	rw/r	rw/r	rw/r	rw/r	rw/r	rw/r	rw/r	rw/r	rw/r	rw/r	rw/r	rw/r	rw/r	rw/r

位 31:16 **CCR1[31:16]**：捕获/比较 1 值的高 16 位（对于 TIM2 和 TIM5）(High Capture/Compare 1 value (on TIM2 and TIM5))

位 15:0 **CCR1[15:0]**：捕获/比较 1 值的低 16 位 (Low Capture/Compare 1 value)

如果通道 CC1 配置为输出：

CCR1 是捕获/比较寄存器 1 的预装载值。

如果没有通过 TIMx_CCMR1 寄存器中的 OC1PE 位来使能预装载功能，则该值立刻生效；否则只在发生更新事件时生效（拷贝到有效的捕获/比较寄存器 1）。

有效捕获/比较寄存器中包含要与计数器 TIMx_CNT 进行比较并在 OC1 输出上发出信号的值。

如果通道 CC1 配置为输入：

CCR1 为上一个输入捕获 1 事件 (IC1) 发生时的计数器值。只能读取 TIMx_CCR1 寄存器，无法对其进行编程。

39.4.14 TIMx 捕获/比较寄存器 2 (TIMx_CCR2)

TIMx capture/compare register 2

偏移地址：0x38

复位值：0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CCR2[31:16]（取决于定时器）															
rw/r	rw/r	rw/r	rw/r	rw/r	rw/r	rw/r	rw/r	rw/r	rw/r	rw/r	rw/r	rw/r	rw/r	rw/r	rw/r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR2[15:0]															
rw/r	rw/r	rw/r	rw/r	rw/r	rw/r	rw/r	rw/r	rw/r	rw/r	rw/r	rw/r	rw/r	rw/r	rw/r	rw/r

位 31:16 **CCR2[31:16]**：捕获/比较 2 值的高 16 位（对于 TIM2 和 TIM5）(High Capture/Compare 2 value (on TIM2 and TIM5))。

位 15:0 **CCR2[15:0]**：捕获/比较 2 值的低 16 位 (Low Capture/Compare 2 value)

如果通道 CC2 配置为输出：

CCR2 是捕获/比较寄存器 2 的预装载值。

如果没有通过 TIMx_CCMR1 寄存器中的 OC2PE 位来使能预装载功能，则该值立刻生效；否则只在发生更新事件时生效（拷贝到有效的捕获/比较寄存器 2）。

有效捕获/比较寄存器中包含要与计数器 TIMx_CNT 进行比较并在 OC2 输出上发出信号的值。

如果通道 CC2 配置为输入：

CCR2 为上一个输入捕获 2 事件 (IC2) 发生时的计数器值。只能读取 TIMx_CCR2 寄存器，无法对其进行编程。

39.4.15 TIMx 捕获/比较寄存器 3 (TIMx_CCR3)

TIMx capture/compare register 3

偏移地址：0x3C

复位值：0x0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CCR3[31:16]（取决于定时器）															
rw/r	rw/r	rw/r	rw/r	rw/r	rw/r	rw/r	rw/r	rw/r	rw/r	rw/r	rw/r	rw/r	rw/r	rw/r	rw/r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR3[15:0]															
rw/r	rw/r	rw/r	rw/r	rw/r	rw/r	rw/r	rw/r	rw/r	rw/r	rw/r	rw/r	rw/r	rw/r	rw/r	rw/r

位 31:16 **CCR3[31:16]**：捕获/比较 3 值的高 16 位（对于 TIM2 和 TIM5）(High Capture/Compare 3 value (on TIM2 and TIM5))

位 15:0 **CCR3[15:0]**：捕获/比较 3 值的低 16 位 (Low Capture/Compare 3 value)

如果通道 CC3 配置为输出：

CCR3 是捕获/比较寄存器 3 的预装载值。

如果没有通过 TIMx_CCMR2 寄存器中的 OC3PE 位来使能预装载功能，则该值立刻生效；否则只在发生更新事件时生效（拷贝到有效的捕获/比较寄存器 3）。

有效捕获/比较寄存器中包含要与计数器 TIMx_CNT 进行比较并在 OC3 输出上发出信号的值。

如果通道 CC3 配置为输入：

CCR3 为上一个输入捕获 3 事件 (IC3) 发生时的计数器值。只能读取 TIMx_CCR3 寄存器，无法对其进行编程。

39.4.16 TIMx 捕获/比较寄存器 4 (TIMx_CCR4)

TIMx capture/compare register 4

偏移地址：0x40

复位值：0x0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CCR4[31:16]（取决于定时器）															
rw/r	rw/r	rw/r	rw/r	rw/r	rw/r	rw/r	rw/r	rw/r	rw/r	rw/r	rw/r	rw/r	rw/r	rw/r	rw/r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR4[15:0]															
rw/r	rw/r	rw/r	rw/r	rw/r	rw/r	rw/r	rw/r	rw/r	rw/r	rw/r	rw/r	rw/r	rw/r	rw/r	rw/r

位 31:16 **CCR4[31:16]**：捕获/比较 4 值的高 16 位（对于 TIM2 和 TIM5）(High Capture/Compare 4 value (on TIM2 and TIM5))

位 15:0 **CCR4[15:0]**：捕获/比较 4 值的低 16 位 (Low Capture/Compare 4 value)

1. 如果 CC4 通道配置为输出（CC4S 位）：
   CCR4 是捕获/比较寄存器 4 的预装载值。
   如果没有通过 TIMx_CCMR2 寄存器中的 OC4PE 位来使能预装载功能，则该值立刻生效；否则只在发生更新事件时生效（拷贝到有效的捕获/比较寄存器 4）。
   有效捕获/比较寄存器中包含要与计数器 TIMx_CNT 进行比较并在 OC4 输出上发出信号的值。
2. 如果 CC4 通道配置为输入（TIMx_CCMR4 寄存器中的 CC4S 位）：
   CCR4 为上一个输入捕获 4 事件 (IC4) 发生时的计数器值。只能读取 TIMx_CCR4 寄存器，无法对其进行编程。

39.4.17 TIMx DMA 控制寄存器 (TIMx_DCR)

TIMx DMA control register

偏移地址：0x48

复位值：0x0000

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

位 15:13 保留，必须保持复位值。

位 12:8 **DBL[4:0]**：DMA 连续传送长度 (DMA burst length)

该 5 位向量定义了 DMA 的传送次数（当对 TIMx_DMAR 寄存器进行读或写时，定时器进行一次连续传送）。

00000：1 次传送，
00001：2 次传送，
00010：3 次传送，
...
10001：18 次传送。

位 7:5 保留，必须保持复位值。

位 4:0 **DBA[4:0]**：DMA 基址 (DMA base address)

该 5 位向量定义 DMA 传输的基址（通过 TIMx_DMAR 地址进行读/写访问时）。DBA 定义为从 TIMx_CR1 寄存器地址开始计算的偏移量。

示例：
00000：TIMx_CR1
00001：TIMx_CR2
00010：TIMx_SMCR
...

示例：以下面的传送为例：DBL = 7 次传送且 DBA = TIMx_CR1。这种情况下将向/从自 TIMx_CR1 地址开始的 7 个寄存器传输数据。

39.4.18 TIMx 全传输 DMA 地址 (TIMx_DMAR)

TIMx DMA address for full transfer

偏移地址：0x4C

复位值：0x0000

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

位 15:0 **DMAB[15:0]**：DMA 连续传送寄存器 (DMA register for burst accesses)

对 DMAR 寄存器执行读或写操作将访问位于如下地址的寄存器

(TIMx_CR1 地址) + (DBA + DMA 索引) x 4

其中 TIMx_CR1 地址为控制寄存器 1 的地址，DBA 为 TIMx_DCR 寄存器中配置的 DMA 基址，DMA 索引由 DMA 传输自动控制，其范围介于 0 到 DBL（TIMx_DCR 寄存器中配置的 DBL）之间。

39.4.19 TIM2 复用功能选项寄存器 1 (TIM2_AF1)

TIM2 alternate function option register 1

偏移地址：0x60

复位值：0x0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ETRSEL[3:2]	
														rw	rw

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

位 31:18 保留，必须保持复位值。

位 17:14 **ETRSEL[3:0]**：ETR 源选择 (ETR source selection)

这些位选择 ETR 输入源。

0000：ETR 输入连接到 I/O

0001：COMP1 输出

0010：COMP2 输出

0011：LSE

0100：SAI1 FS_A

0101：SAI1 FS_B

其它：保留

位 13:0 保留，必须保持复位值。

39.4.20 TIM3 复用功能选项寄存器 1 (TIM3_AF1)

TIM3 alternate function option register 1

偏移地址：0x60

复位值：0x0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ETRSEL[3:2]	
														rw	rw

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

位 31:18 保留，必须保持复位值。

位 17:14 **ETRSEL[3:0]**：ETR 源选择 (ETR source selection)

这些位选择 ETR 输入源。

0000：ETR 输入连接到 I/O

0001：COMP1 输出

其它：保留

位 13:0 保留，必须保持复位值。

39.4.21 TIM5 复用功能选项寄存器 1 (TIM5_AF1)

TIM5 alternate function option register 1

偏移地址：0x60

复位值：0x0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ETRSEL[3:2]	
														rw	rw

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

位 31:18 保留，必须保持复位值。

位 17:14 **ETRSEL[3:0]**：ETR 源选择 (ETR source selection)

这些位选择 ETR 输入源。

0000：ETR 输入连接到 I/O

0001：SAI2 FS_A 连接到 ETR 输入

0010：SAI2 FS_B 连接到 ETR 输入

其它：保留

位 13:0 保留，必须保持复位值。

39.4.22 TIM2 定时器输入选择寄存器 (TIM2_TISEL)

TIM2 timer input selection register

偏移地址：0x68

复位值：0x0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	TI4SEL[3:0]				Res.	Res.	Res.	Res.	TI3SEL[3:0]			
				rw	rw	rw	rw					rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	TI2SEL[3:0]				Res.	Res.	Res.	Res.	TI1SEL[3:0]			
				rw	rw	rw	rw					rw	rw	rw	rw

位 31:28 保留，必须保持复位值。

位 27:24 **TI4SEL[3:0]**：TI4[0] 到 TI4[15] 输入选择 (TI4[0] to TI4[15] input selection)

这些位选择 TI4[0] 到 TI4[15] 输入源。

0000：TIM2_CH4 输入

0001：COMP1 输出

0010：COMP2 输出

0011：COMP1 输出或 COMP2 输出

其它：保留

位 23:20 保留，必须保持复位值。

位 19:16 **TI3SEL[3:0]**：TI3[0] 到 TI3[15] 输入选择 (TI3[0] to TI3[15] input selection)

这些位选择 TI3[0] 到 TI3[15] 输入源。

0000：TIM2_CH3 输入

其它：保留

位 15:12 保留，必须保持复位值。

位 11:8 **TI2SEL[3:0]**：TI2[0] 到 TI2[15] 输入选择 (TI2[0] to TI2[15] input selection)
这些位选择 TI2[0] 到 TI2[15] 输入源。
0000：TIM2_CH2 输入
其它：保留

位 7:4 保留，必须保持复位值。

位 3:0 **TI1SEL[3:0]**：TI1[0] 到 TI1[15] 输入选择 (TI1[0] to TI1[15] input selection)
这些位选择 TI1[0] 到 TI1[15] 输入源。
0000：TIM2_CH1 输入
其它：保留

39.4.23 TIM3 定时器输入选择寄存器 (TIM3_TISEL)

TIM3 timer input selection register

偏移地址：0x68

复位值：0x0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	TI4SEL[3:0]				Res.	Res.	Res.	Res.	TI3SEL[3:0]			
				rw	rw	rw	rw					rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	TI2SEL[3:0]				Res.	Res.	Res.	Res.	TI1SEL[3:0]			
				rw	rw	rw	rw					rw	rw	rw	rw

位 31:28 保留，必须保持复位值。

位 27:24 **TI4SEL[3:0]**：TI4[0] 到 TI4[15] 输入选择 (TI4[0] to TI4[15] input selection)
这些位选择 TI4[0] 到 TI4[15] 输入源。
0000：TIM3_CH4 输入
其它：保留

位 23:20 保留，必须保持复位值。

位 19:16 **TI3SEL[3:0]**：TI3[0] 到 TI3[15] 输入选择 (TI3[0] to TI3[15] input selection)
这些位选择 TI3[0] 到 TI3[15] 输入源。
0000：TIM3_CH3 输入
其它：保留

位 15:12 保留，必须保持复位值。

位 11:8 **TI2SEL[3:0]**：TI2[0] 到 TI2[15] 输入选择 (TI2[0] to TI2[15] input selection)
这些位选择 TI2[0] 到 TI2[15] 输入源。
0000：TIM3_CH2 输入
其它：保留

位 7:4 保留，必须保持复位值。

位 3:0 **TI1SEL[3:0]**：TI1[0] 到 TI1[15] 输入选择 (TI1[0] to TI1[15] input selection)
这些位选择 TI1[0] 到 TI1[15] 输入源。
0000：TIM3_CH1 输入
0001：COMP1 输出
0010：COMP2 输出
0011：COMP1 输出或 COMP2 输出
其它：保留

39.4.24 TIM5 定时器输入选择寄存器 (TIM5_TISEL)

TIM5 timer input selection register

偏移地址：0x68

复位值：0x0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	TI4SEL[3:0]				Res.	Res.	Res.	Res.	TI3SEL[3:0]			
				rw	rw	rw	rw					rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	TI2SEL[3:0]				Res.	Res.	Res.	Res.	TI1SEL[3:0]			
				rw	rw	rw	rw					rw	rw	rw	rw

位 31:28 保留，必须保持复位值。

位 27:24 **TI4SEL[3:0]**：TI4[0] 到 TI4[15] 输入选择 (TI4[0] to TI4[15] input selection)
这些位选择 TI4[0] 到 TI4[15] 输入源。
0000：TIM5_CH4 输入
其它：保留

位 23:20 保留，必须保持复位值。

位 19:16 **TI3SEL[3:0]**：TI3[0] 到 TI3[15] 输入选择 (TI3[0] to TI3[15] input selection)
这些位选择 TI3[0] 到 TI3[15] 输入源。
0000：TIM5_CH3 输入
其它：保留

位 15:12 保留，必须保持复位值。

位 11:8 **TI2SEL[3:0]**：TI2[0] 到 TI2[15] 输入选择 (TI2[0] to TI2[15] input selection)
这些位选择 TI2[0] 到 TI2[15] 输入源。
0000：TIM5_CH2 输入
其它：保留

位 7:4 保留，必须保持复位值。

位 3:0 **TI1SEL[3:0]**：TI1[0] 到 TI1[15] 输入选择 (TI1[0] to TI1[15] input selection)
这些位选择 TI1[0] 到 TI1[15] 输入源。
0000：TIM5_CH1 输入
0001：fdcan1_tmp
0010：fdcan1_rtp
其它：保留

39.4.25 TIMx 寄存器映射

TIMx 寄存器按照下表所述方式映射：

表 319. TIM2/TIM3/TIM4/TIM5 寄存器映射和复位值

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	**TIMx_CR1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	UIFREMAP	Res.	CKD [1:0]		ARPE	CMS [1:0]		DIR	OPM	URS	UDIS	CEN
	Reset value																					0		0	0	0	0	0	0	0	0	0	0
0x04	**TIMx_CR2**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TI1S	MMS[2:0]			CCDS	Res.	Res.	Res.
	Reset value																									0	0	0	0	0			
0x08	**TIMx_SMCR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TS [4:3]		Res.	Res.	Res.	SMS[3]	ETP	ECE	ETPS [1:0]		ETF[3:0]				MSM	TS[2:0]			Res.	SMS[2:0]		
	Reset value											0	0				0	0	0	0	0	0	0	0	0	0	0	0	0		0	0	0
0x0C	**TIMx_DIER**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TDE	COMDE	CC4DE	CC3DE	CC2DE	CC1DE	UDE	Res.	TIE	Res.	CC4IE	CC3IE	CC2IE	CC1IE	UIE
	Reset value																		0	0	0	0	0	0	0		0		0	0	0	0	0
0x10	**TIMx_SR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CC4OF	CC3OF	CC2OF	CC1OF	Res.	Res.	TIF	Res.	CC4IF	CC3IF	CC2IF	CC1IF	UIF
	Reset value																				0	0	0	0			0		0	0	0	0	0
0x14	**TIMx_EGR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TG	Res.	CC4G	CC3G	CC2G	CC1G	UG
	Reset value																										0		0	0	0	0	0
0x18	**TIMx_CCMR1** Output Compare mode	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC2M[3]	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC1M[3]	OC2CE	OC2M [2:0]			OC2PE	OC2FE	CC2S [1:0]		OC1CE	OC1M [2:0]			OC1PE	OC1FE	CC1S [1:0]	
	Reset value								0								0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	TIMx_CCMR1 Input Capture mode	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	IC2F[3:0]				IC2 PSC [1:0]		CC2S [1:0]		IC1F[3:0]				IC1 PSC [1:0]		CC1S [1:0]	
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x1C	**TIMx_CCMR2** Output Compare mode	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC4M[3]	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC3M[3]	O24CE	OC4M [2:0]			OC4PE	OC4FE	CC4S [1:0]		OC3CE	OC3M [2:0]			OC3PE	OC3FE	CC3S [1:0]	
	Reset value								0								0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	TIMx_CCMR2 Input Capture mode	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	IC4F[3:0]				IC4 PSC [1:0]		CC4S [1:0]		IC3F[3:0]				IC3 PSC [1:0]		CC3S [1:0]	
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x20	**TIMx_CCER**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CC4NP	Res.	CC4P	CC4E	CC3NP	Res.	CC3P	CC3E	CC2NP	Res.	CC2P	CC2E	CC1NP	Res.	CC1P	CC1E
	Reset value																	0		0	0	0		0	0	0		0	0	0		0	0

表 319. TIM2/TIM3/TIM4/TIM5 寄存器映射和复位值（续）

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x24	**TIMx_CNT**	CNT[31] 或 UIFCPY	CNT[30:16]（仅限 TIM2 和 TIM5，其它定时器上保留）															CNT[15:0]															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x28	**TIMx_PSC**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PSC[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x2C	**TIMx_ARR**	ARR[31:16]（仅限 TIM2 和 TIM5，其它定时器上保留）																ARR[15:0]															
	Reset value	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0x30	Reserved																																
0x34	**TIMx_CCR1**	CCR1[31:16]（仅限 TIM2 和 TIM5，其它定时器上保留）																CCR1[15:0]															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x38	**TIMx_CCR2**	CCR2[31:16]（仅限 TIM2 和 TIM5，其它定时器上保留）																CCR2[15:0]															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x3C	**TIMx_CCR3**	CCR3[31:16]（仅限 TIM2 和 TIM5，其它定时器上保留）																CCR3[15:0]															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x40	**TIMx_CCR4**	CCR4[31:16]（仅限 TIM2 和 TIM5，其它定时器上保留）																CCR4[15:0]															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x44	Reserved																																
0x48	**TIMx_DCR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DBL[4:0]					Res.	Res.	Res.	DBA[4:0]				
	Reset value																				0	0	0	0	0				0	0	0	0	0
0x4C	**TIMx_DMAR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DMAB[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x60	**TIM2_AF1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ETRSEL [3:0]		Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	Reset value															0	0	0	0														
0x60	**TIM3_AF1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ETRSEL [3:0]		Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	Reset value															0	0	0	0														
0x60	**TIM5_AF1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ETRSEL [3:0]		Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	Reset value															0	0	0	0														

表 319. TIM2/TIM3/TIM4/TIM5 寄存器映射和复位值（续）

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x68	**TIM2_TISEL**	Res.	Res.	Res.	Res.	TI4SEL[3:0]				Res.	Res.	Res.	Res.	TI3SEL[3:0]				Res.	Res.	Res.	Res.	TI2SEL[3:0]				Res.	Res.	Res.	Res.	TI1SEL[3:0]			
	Reset value					0	0	0	0					0	0	0	0					0	0	0	0					0	0	0	0
0x68	**TIM3_TISEL**	Res.	Res.	Res.	Res.	TI4SEL[3:0]				Res.	Res.	Res.	Res.	TI3SEL[3:0]				Res.	Res.	Res.	Res.	TI2SEL[3:0]				Res.	Res.	Res.	Res.	TI1SEL[3:0]			
	Reset value					0	0	0	0					0	0	0	0					0	0	0	0					0	0	0	0
0x68	**TIM5_TISEL**	Res.	Res.	Res.	Res.	TI4SEL[3:0]				Res.	Res.	Res.	Res.	TI3SEL[3:0]				Res.	Res.	Res.	Res.	TI2SEL[3:0]				Res.	Res.	Res.	Res.	TI1SEL[3:0]			
	Reset value					0	0	0	0					0	0	0	0					0	0	0	0					0	0	0	0

有关寄存器边界地址的信息，请参见*第 2.2.2 节：存储器映射和寄存器边界地址*。

40 通用定时器 (TIM12/TIM13/TIM14)

40.1 TIM12/TIM13/TIM14 简介

TIM12/TIM13/TIM14 通用定时器包含一个 16 位自动重载计数器，该计数器由可编程预分频器驱动。

它们可用于多种用途，包括测量输入信号的脉冲宽度（输入捕获），或者生成输出波形（输出比较、PWM）。

使用定时器预分频器和 RCC 时钟控制器预分频器，可将脉冲宽度和波形周期从几微秒调制到几毫秒。

这些定时器彼此完全独立，不共享任何资源。如*第 40.3.17 节：定时器同步(TIM12)* 中所述，它们可以同步操作。

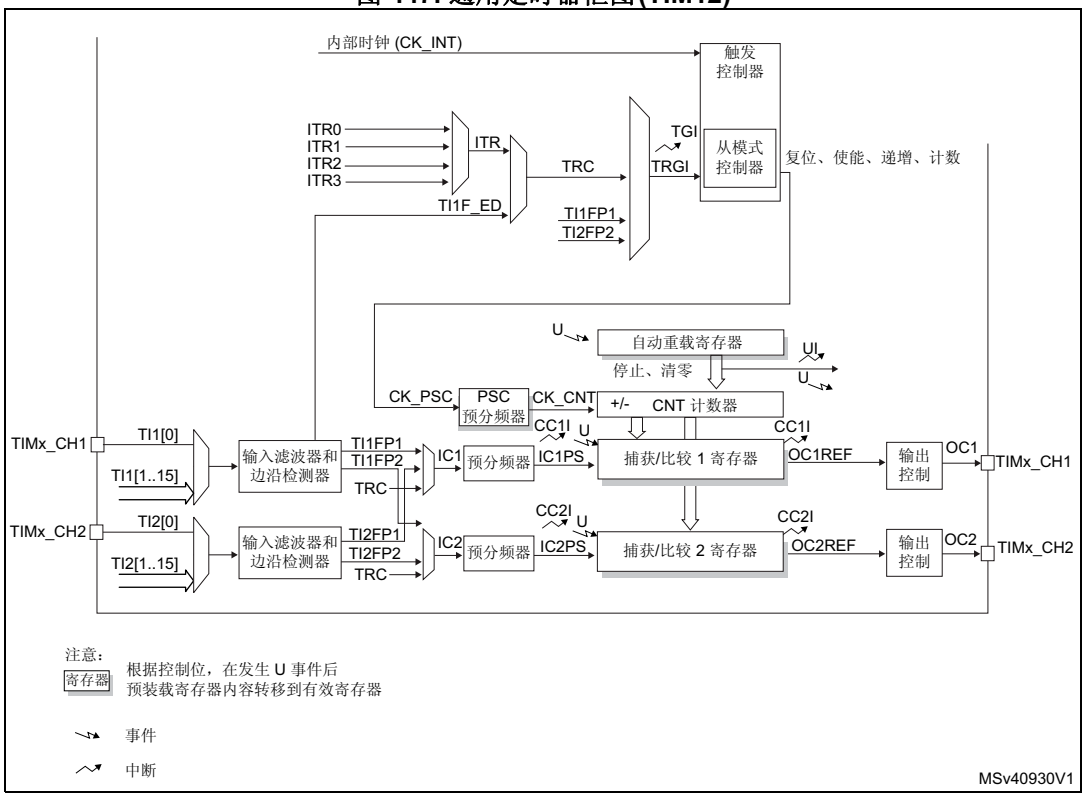
40.2 TIM12/TIM13/TIM14 主要特性

40.2.1 TIM12 主要特性

TIM12 通用定时器具有以下特性：

- 16 位自动重载递增计数器
- 16 位可编程预分频器，用于对计数器时钟频率进行分频（可在运行时修改），分频系数介于 1 和 65536 之间
- 多达 2 个独立通道，可用于：
  - 输入捕获
  - 输出比较
  - PWM 生成（边沿对齐模式）
  - 单脉冲模式输出
- 使用外部信号控制定时器且可实现多个定时器互连的同步电路
- 发生如下事件时生成中断：
  - 更新：计数器上溢、计数器初始化（通过软件或内部触发）
  - 触发事件（计数器启动、停止、初始化或者由内部触发计数）
  - 输入捕获
  - 输出比较

图 447. 通用定时器框图(TIM12)

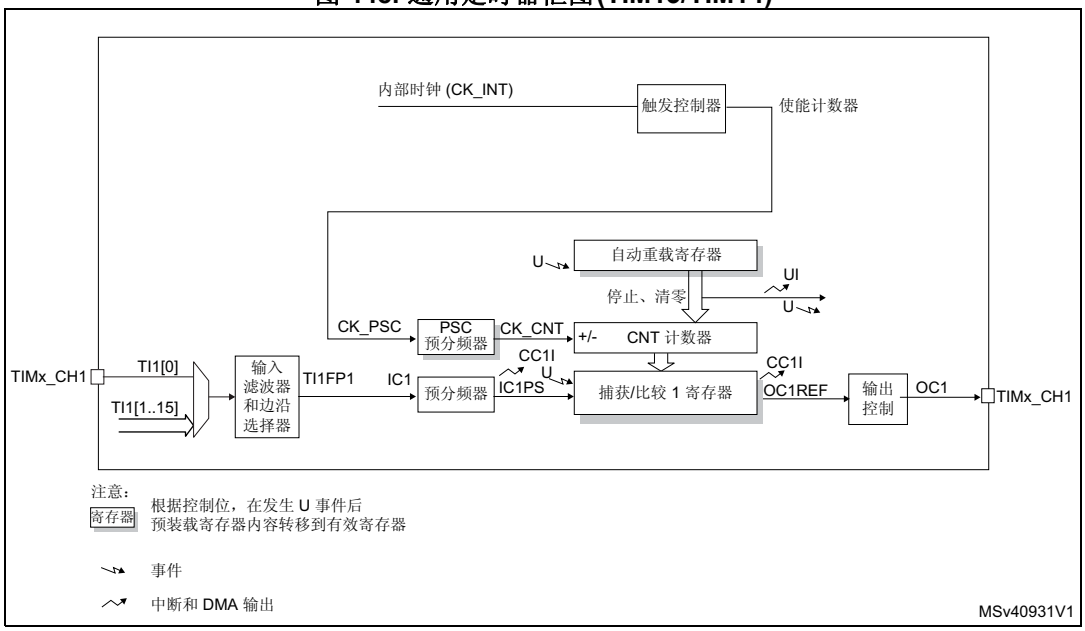


40.2.2 TIM13/TIM14主要特性

通用定时器 TIM13/TIM14 具有以下特性：

- 16 位自动重载递增计数器
- 16 位可编程预分频器，用于对计数器时钟频率进行分频（可在运行时修改），分频系数介于 1 和 65536 之间
- 独立通道，可用于：
  - 输入捕获
  - 输出比较
  - PWM 生成（边沿对齐模式）
  - 单脉冲模式输出
- 发生如下事件时生成中断：
  - 更新：计数器上溢、计数器初始化（通过软件）
  - 输入捕获
  - 输出比较

图 448. 通用定时器框图(TIM13/TIM14)



40.3 TIM12/TIM13/TIM14 功能描述

40.3.1 时基单元

定时器的主要模块由一个 16 位递增计数器及其相关的自动重载寄存器组成。计数器采用递增方式计数。

计数器的时钟可通过预分频器进行分频。

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

时基单元包括：

- 计数器寄存器 (TIMx_CNT)
- 预分频器寄存器 (TIMx_PSC)
- 自动重载寄存器 (TIMx_ARR)

自动重载寄存器是预装载的。对自动重载寄存器执行写入或读取操作时会访问预装载寄存器。预装载寄存器的内容既可以立即传送到影子寄存器，也可以在每次发生更新事件 (UEV) 时传送到影子寄存器，这取决于 TIMx_CR1 寄存器中的自动重载预装载使能位 (ARPE)。当计数器达到上溢值并且 TIMx_CR1 寄存器中的 UDIS 位为 0 时，将发送更新事件。该更新事件也可由软件产生。下文将针对各配置的更新事件的产生进行详细介绍。

计数器由预分频器输出 CK_CNT 提供时钟，仅当 TIMx_CR1 寄存器中的计数器启动位 (CEN) 置 1 时，才会启动计数器（有关计数器使能的更多详细信息，另请参见从模式控制器的相关说明）。

注意，计数器将在 TIMx_CR1 寄存器的 CEN 位置 1 时刻的一个时钟周期后开始计数。

预分频器说明

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

图 449 和*图 450* 以一些示例说明在预分频比实时变化时计数器的行为。

图 449. 预分频器分频由 1 变为 2 时的计数器时序图

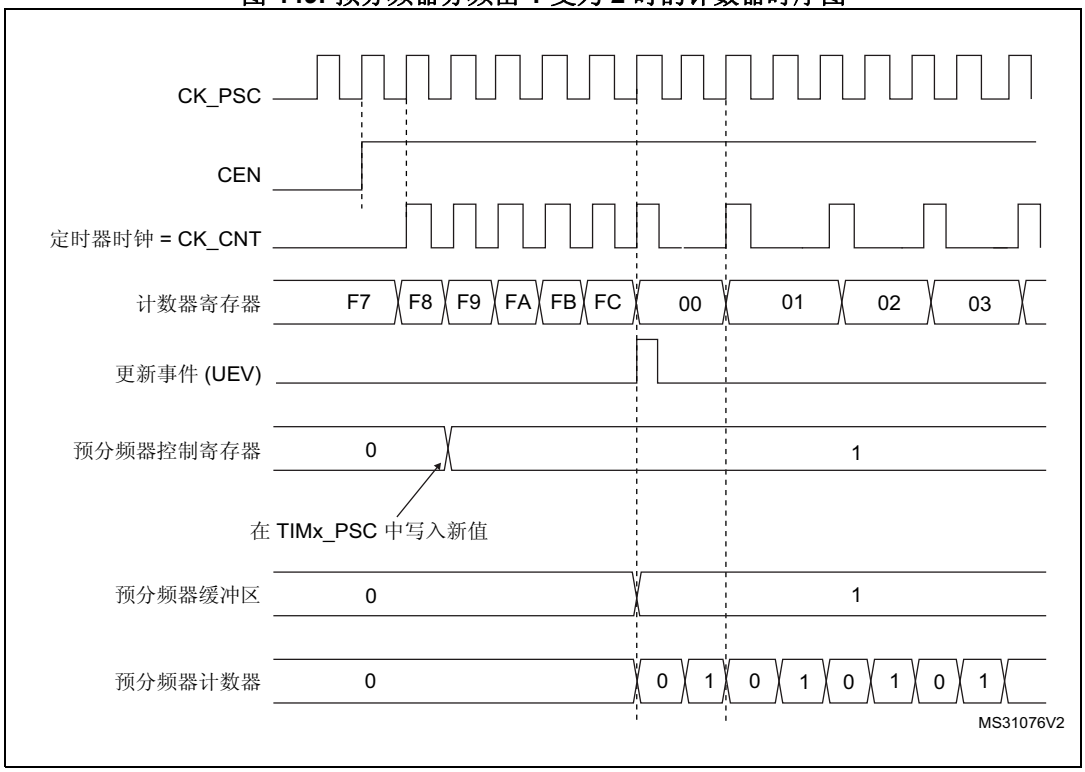
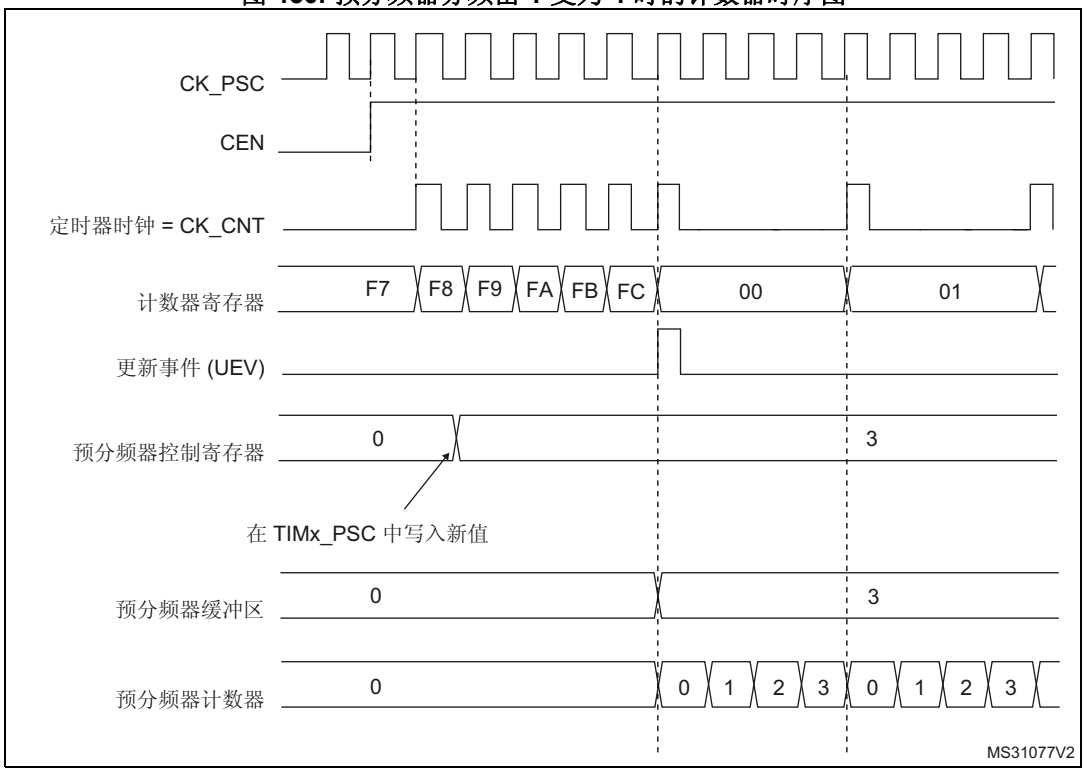


图 450. 预分频器分频由 1 变为 4 时的计数器时序图



40.3.2 计数器模式

递增计数模式

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

设置 TIMx_EGR 寄存器中的 UG 位（通过软件或使用 TIM12 上的从模式控制器）时，也将产生更新事件。

通过软件将 TIMx_CR1 寄存器中的 UDIS 位置 1 可禁止 UEV 事件。这可避免向预装载寄存器写入新值时更新影子寄存器。在 UDIS 位写入 0 之前不会产生任何更新事件。不过，计数器和预分频器计数器都会重新从 0 开始计数（而预分频比保持不变）。此外，如果 TIMx_CR1 寄存器中的 URS 位（更新请求选择）已置 1，则将 UG 位置 1 会生成更新事件 UEV，但不会将 UIF 标志置 1（因此，不会发送任何中断）。这样一来，如果在发生捕获事件时将计数器清零，将不会同时产生更新中断和捕获中断。

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

- 使用预装载值 (TIMx_ARR) 更新自动重载影子寄存器，
- 预分频器的缓冲区中将重新装载预装载值（TIMx_PSC 寄存器的内容）。

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

图 451. 计数器时序图，1 分频内部时钟

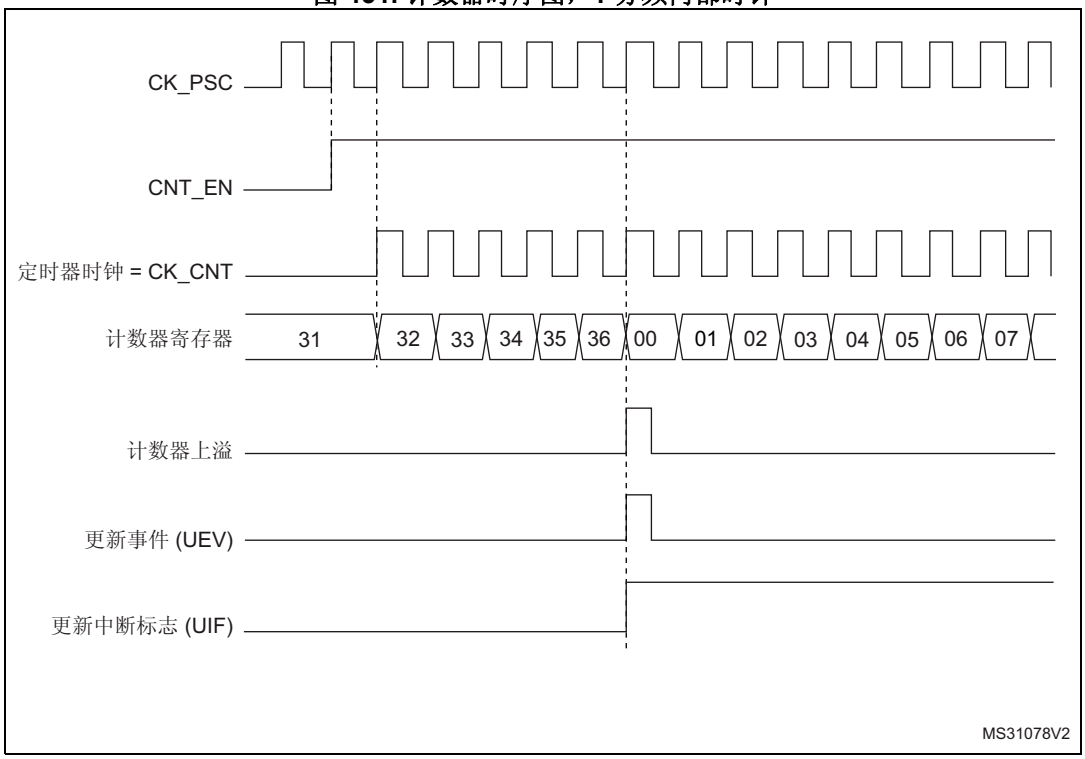


图 452. 计数器时序图，2 分频内部时钟

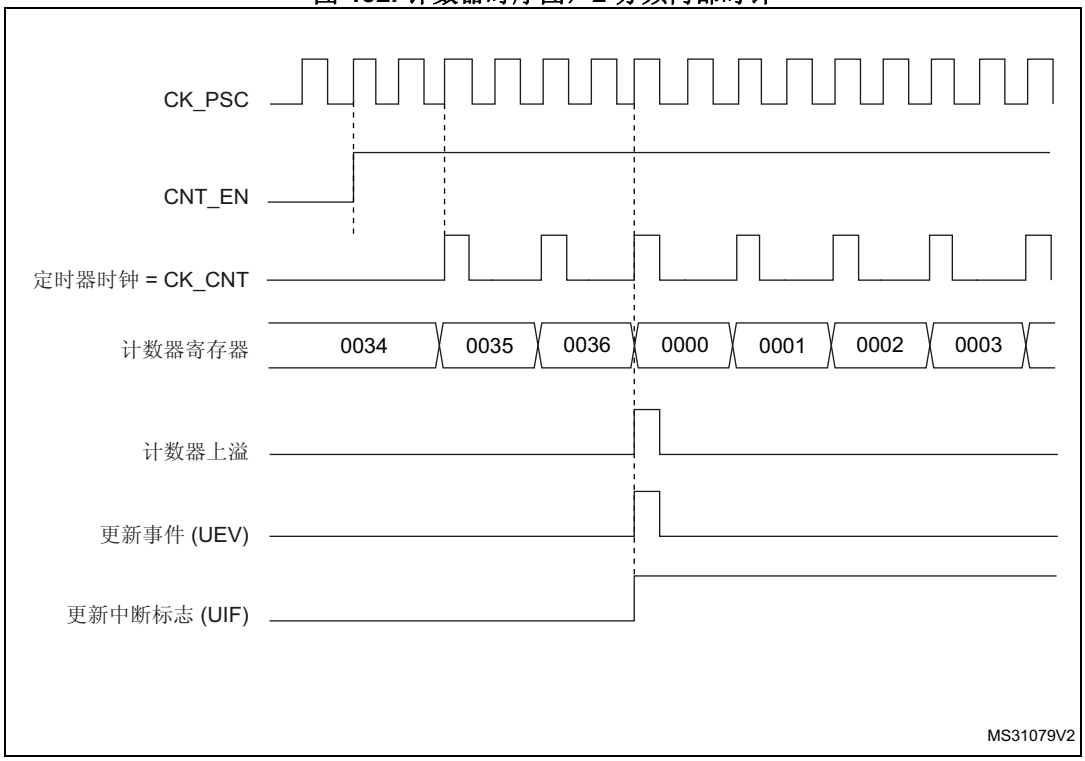


图 453. 计数器时序图，4 分频内部时钟

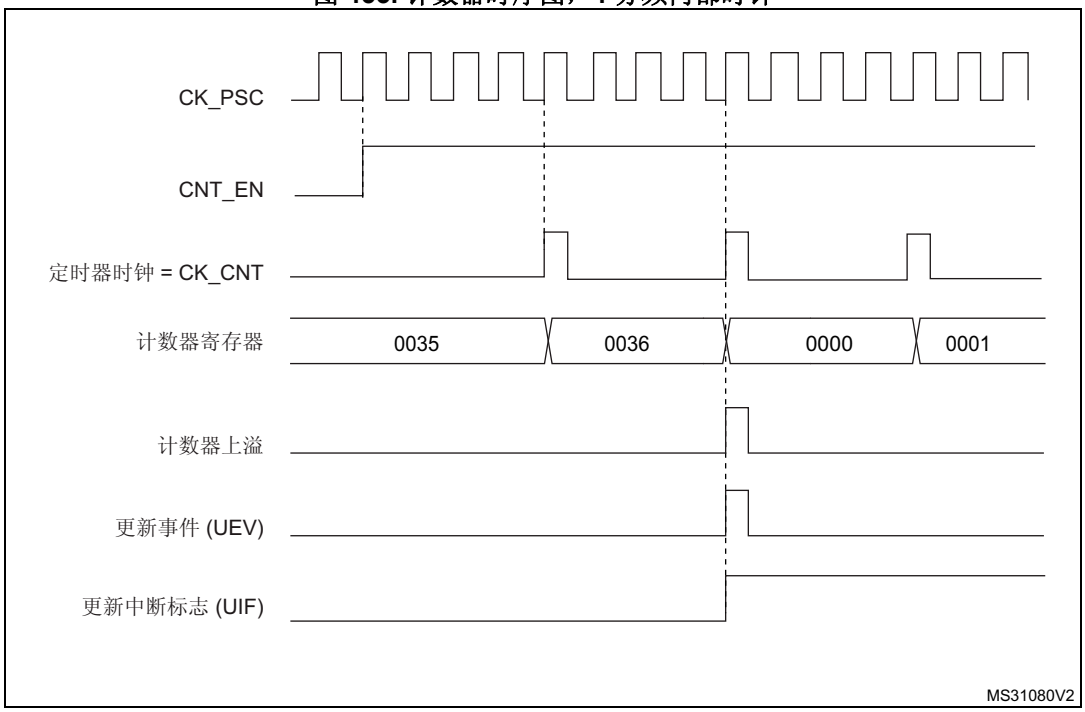


图 454. 计数器时序图，N 分频内部时钟

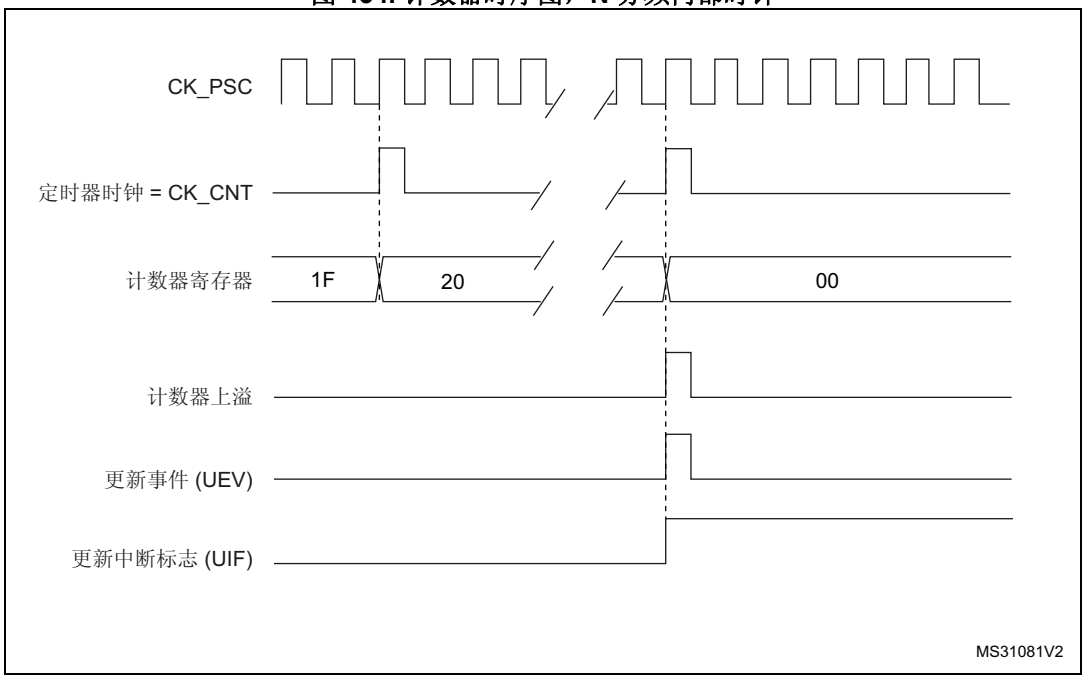


图 455. 计数器时序图，ARPE=0 时更新事件（TIMx_ARR 未预装载）

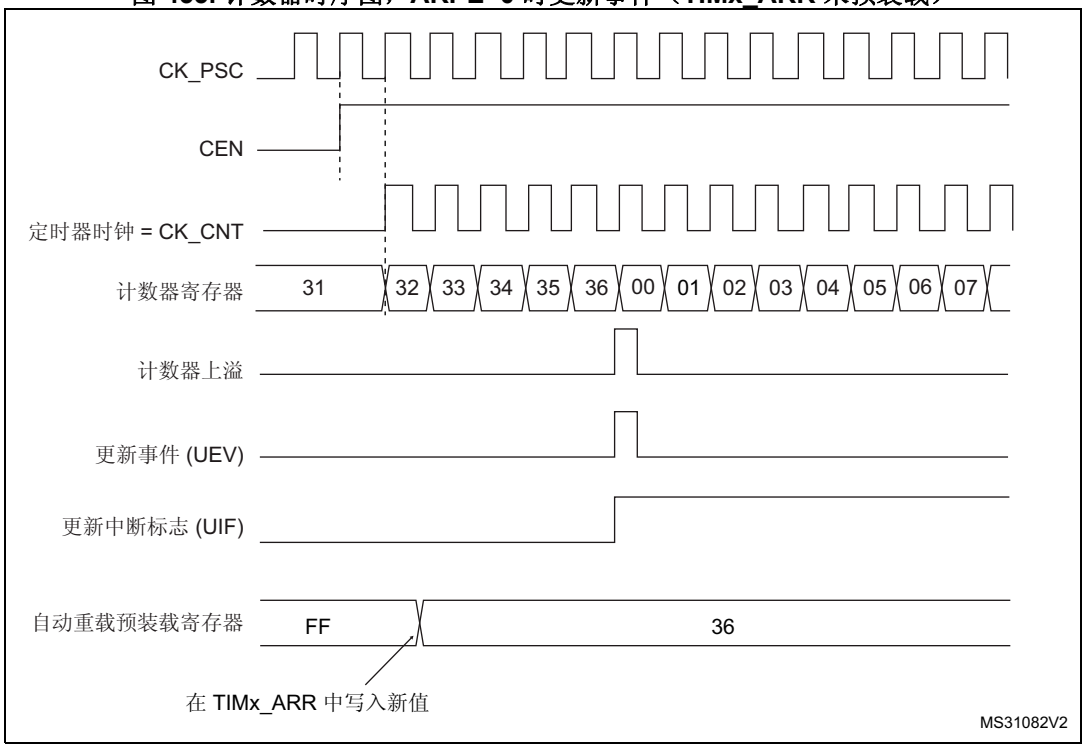
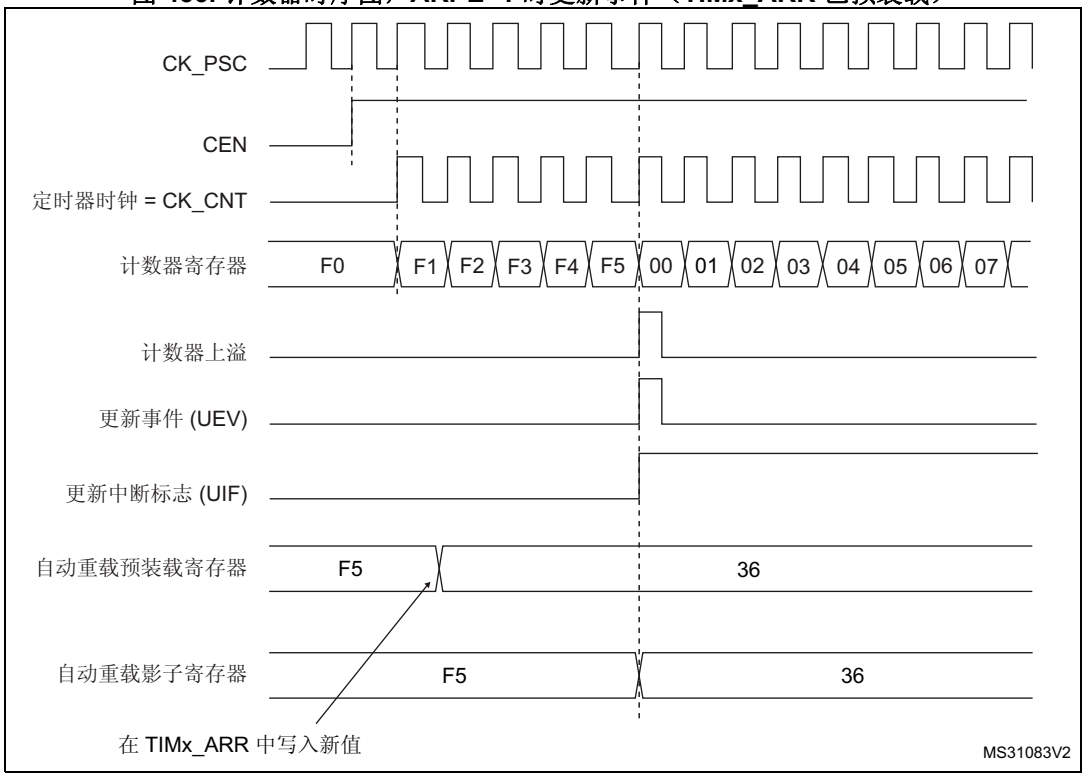


图 456. 计数器时序图，ARPE=1 时更新事件（TIMx_ARR 已预装载）



40.3.3 时钟选择

计数器时钟可由下列时钟源提供：

- 内部时钟 (CK_INT)
- 外部时钟模式 1（针对 TIM12）：外部输入引脚 (TIx)
- 内部触发输入 (ITRx)（针对 TIM12）：连接来自其它定时器的触发输出。例如，其它定时器可配置为 TIM12 的预分频器。更多详细信息，请参见*将一个定时器用作另一个定时器的预分频器*一节。

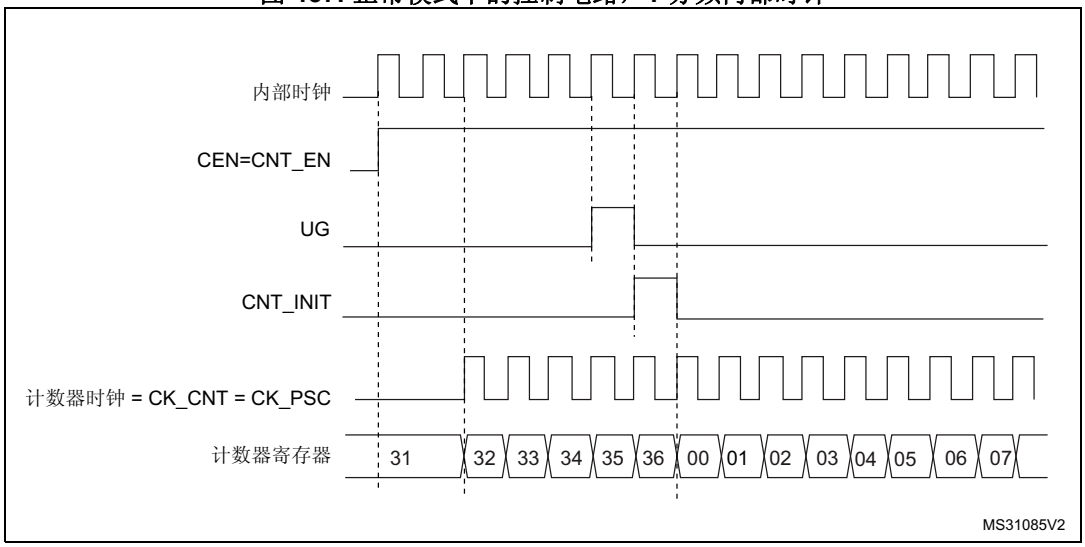
内部时钟源 (CK_INT)

内部时钟源是 TIM13/TIM14 的默认时钟源。

对于 TIM12，在禁止从模式控制器后选择内部时钟源（SMS = “000”）。然后，将 TIMx_CR1 寄存器中的 CEN 位和 TIMx_EGR 寄存器中的 UG 位用作控制位，并且只能通过软件对其进行更改（保持清零状态的 UG 除外）。当对 CEN 位写入 1 时，预分频器的时钟就由内部时钟 CK_INT 提供。

图 457 显示了正常模式下控制电路与递增计数器的行为（没有预分频的情况下）。

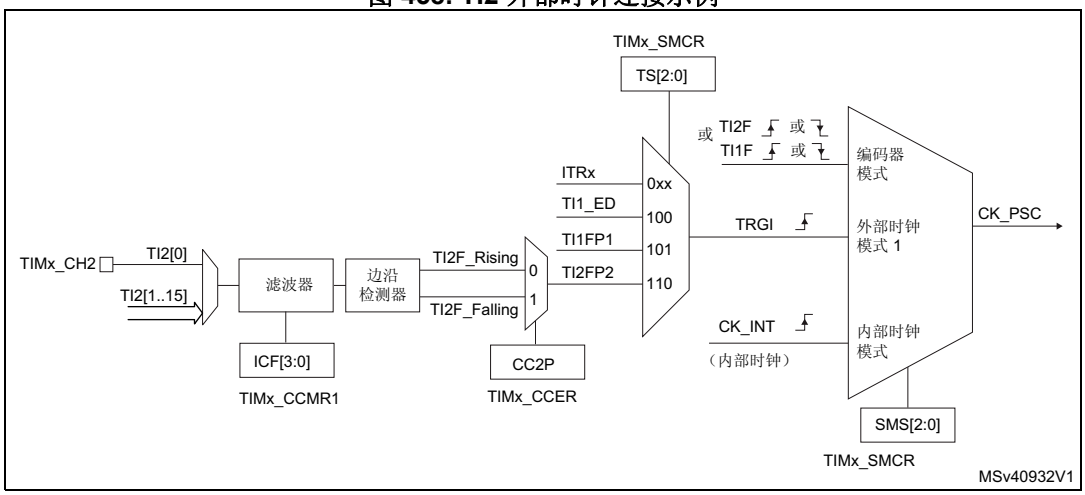
图 457. 正常模式下的控制电路，1 分频内部时钟



外部时钟源模式 1（TIM12）

当 TIMx_SMCR 寄存器中的 SMS=“111”时，可选择此模式。 计数器可在选定的输入信号上出现上升沿或下降沿时计数。

图 458. TI2 外部时钟连接示例



例如，要使递增计数器在 TI2 输入出现上升沿时计数，请执行以下步骤：

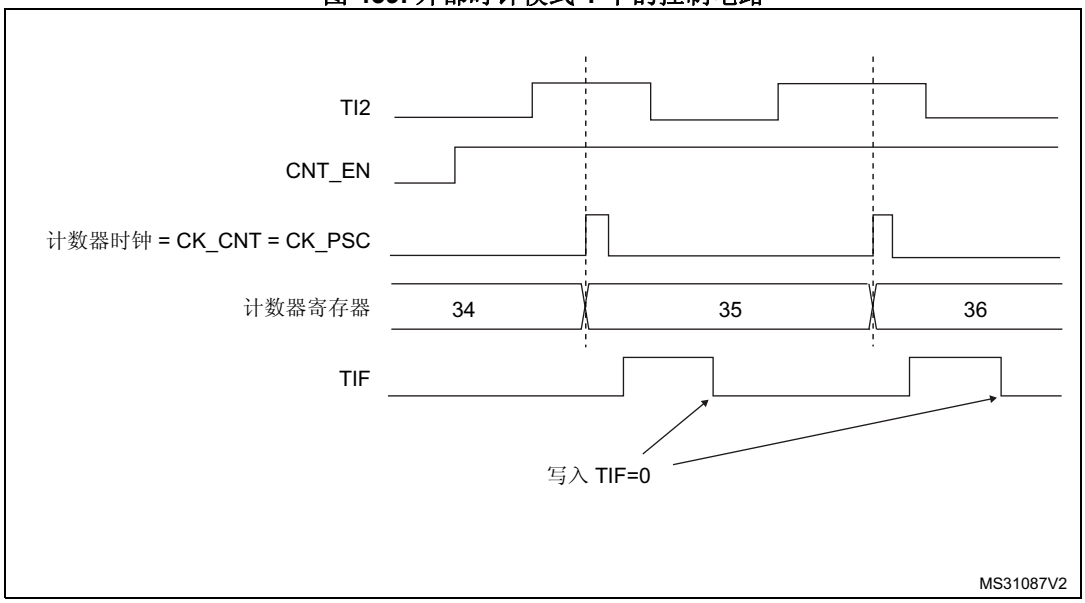
1. 使用 TIMx_TISEL 寄存器中的 TI2SEL[3:0] 位选择正确的 TI2[x] 源（内部或外部）。
2. 通过在 TIMx_CCMR1 寄存器中写入 CC2S =“01”来配置通道 2，使其能够检测 TI2 输入的上升沿。
3. 通过在 TIMx_CCMR1 寄存器中写入 IC2F[3:0] 位来配置输入滤波带宽（如果不需要任何滤波器，请保持 IC2F=“0000”）。
4. 通过在 TIMx_CCER 寄存器中写入 CC2P=“0”和 CC2NP=“0”来选择上升沿极性。
5. 通过在 TIMx_SMCR 寄存器中写入 SMS=“111”，使定时器在外部时钟模式 1 下工作。
6. 通过在 TIMx_SMCR 寄存器中写入 TS=“110”来选择 TI2 作为触发输入源。
7. 通过在 TIMx_CR1 寄存器中写入 CEN=“1”来使能计数器。

注： *由于捕获预分频器不用于触发操作，因此无需对其进行配置。*

当 TI2 出现上升沿时，计数器便会计数一次并且 TIF 标志置 1。

TI2 的上升沿与实际计数器时钟之间的延迟是由于 TI2 输入的重新同步电路引起的。

图 459. 外部时钟模式 1 下的控制电路



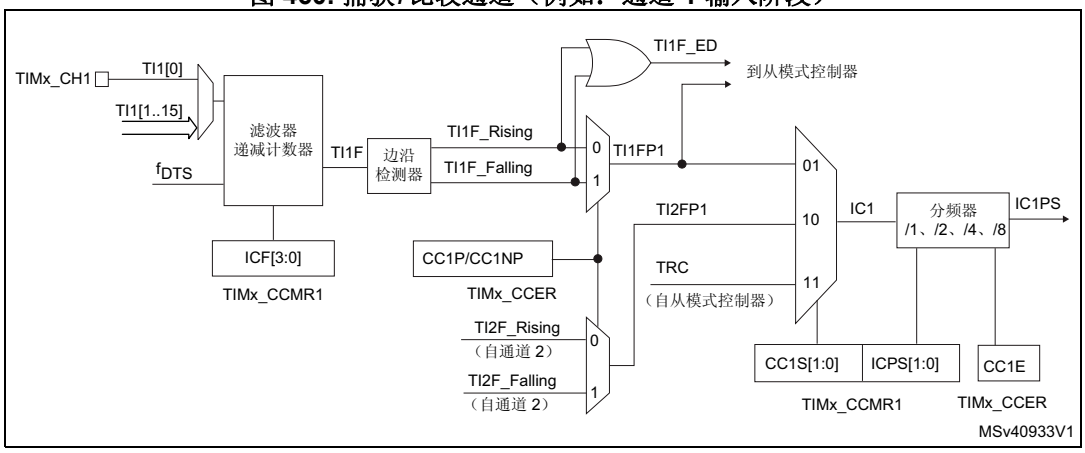
40.3.4 捕获/比较通道

每个捕获/比较通道均围绕一个捕获/比较寄存器（包括一个影子寄存器）、一个捕获输入阶段（数字滤波、多路复用和预分频器）和一个输出阶段（比较器和输出控制）构建而成。

图 460 到*图 462* 概括介绍了一个捕获/比较通道。

输入阶段对相应的 TIx 输入进行采样，生成一个滤波后的信号 TIxF。然后，带有极性选择功能的边沿检测器生成一个信号 (TIxFPx)，该信号可用作从模式控制器的触发输入，也可用作捕获命令。该信号先进行预分频 (ICxPS)，而后再进入捕获寄存器。

图 460. 捕获/比较通道（例如：通道 1 输入阶段）



输出阶段生成一个中间波形作为基准：OCxRef（高电平有效）。链的末端决定最终输出信号的极性。

图 461. 捕获/比较通道 1 主电路

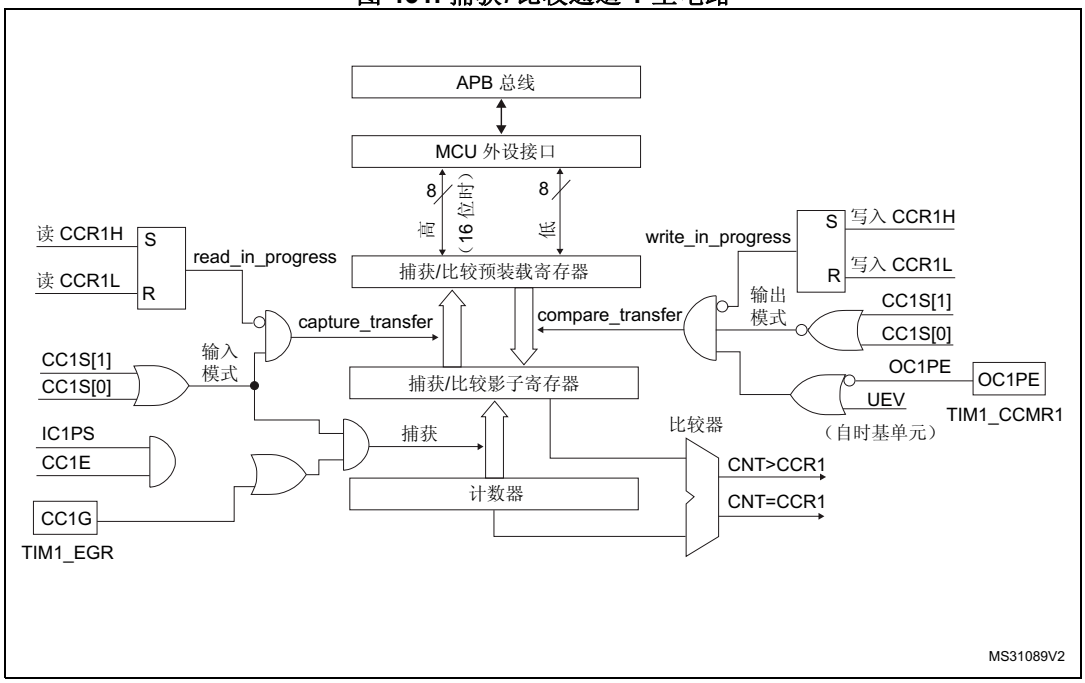
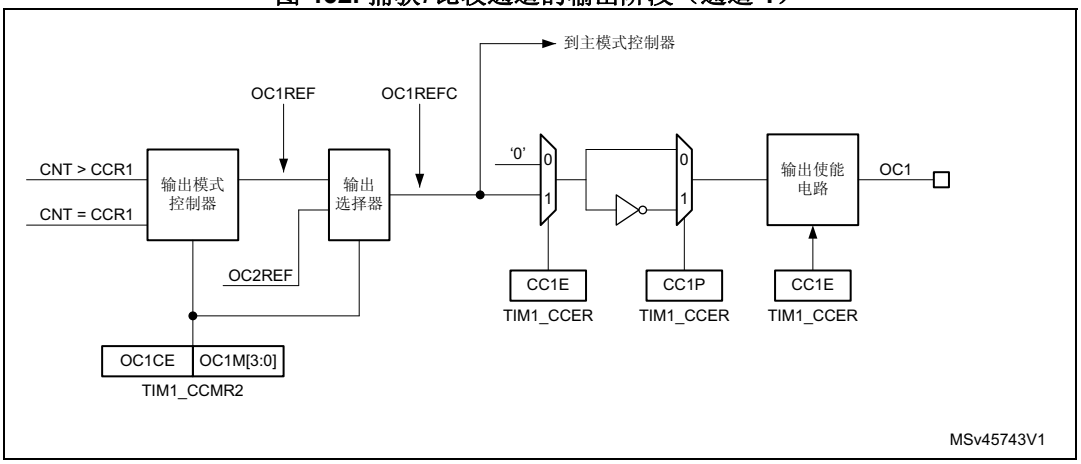


图 462. 捕获/比较通道的输出阶段（通道 1）



捕获/比较模块由一个预装载寄存器和一个影子寄存器组成。始终可通过读写操作访问预装载寄存器。

在捕获模式下，捕获实际发生在影子寄存器中，然后将影子寄存器的内容复制到预装载寄存器中。

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

40.3.5 输入捕获模式

在输入捕获模式下，当相应的 ICx 信号检测到跳变沿后，将使用捕获/比较寄存器 (TIMx_CCRx) 来锁存计数器的值。发生捕获事件时，会将相应的 CCxIF 标志（TIMx_SR 寄存器）置 1，并可发送中断或 DMA 请求（如果已使能）。如果发生捕获事件时 CCxIF 标志已处于高位，则会将重复捕获标志 CCxOF（TIMx_SR 寄存器）置 1。可通过软件将 CCxIF 清零，方法是：向 CCxIF 写入“0”，或读取存储在 TIMx_CCRx 寄存器中的已捕获数据。向 CCxOF 写入“0”后会将其清零。

以下示例说明了如何在 TI1 输入出现上升沿时将计数器的值捕获到 TIMx_CCR1 中。具体操作步骤如下：

1. 使用 TIMx_TISEL 寄存器中的 TI1SEL[3:0] 位选择正确的 TI1[x] 源（内部或外部）。
2. 选择有效输入：TIMx_CCR1 必须连接到 TI1 输入，因此向 TIMx_CCMR1 寄存器中的 CC1S 位写入“01”。只要 CC1S 不等于“00”，就会将通道配置为输入模式，并且 TIMx_CCR1 寄存器将处于只读状态。
3. 根据连接到定时器的信号，对所需的输入滤波带宽进行编程（如果输入为 TIx 输入之一，则对 TIMx_CCMRx 寄存器中的 ICxF 位进行编程）。假设信号边沿变化时，输入信号最多在 5 个内部时钟周期内发生抖动。因此，我们必须将滤波带宽设置为大于 5 个内部时钟周期。在检测到 8 个具有新电平的连续采样（以 f_{DTS} 频率采样）后，可以确认 TI1 上的跳变沿。然后向 TIMx_CCMR1 寄存器中的 IC1F 位写入“0011”。
4. 通过在 TIMx_CCER 寄存器中将 CC1P 位和 CC1NP 位编程为“00”，选择 TI1 上的有效转换边沿（本例中为上升沿）。
5. 对输入预分频器进行编程。在本例中，我们希望每次有效转换时都执行捕获操作，因此需要禁止预分频器（向 TIMx_CCMR1 寄存器中的 IC1PS 位写入“00”）。
6. 通过将 TIMx_CCER 寄存器中的 CC1E 位置 1，允许将计数器的值捕获到捕获寄存器中。
7. 如果需要，可通过将 TIMx_DIER 寄存器中的 CC1IE 位置 1 来使能相关中断请求。

发生输入捕获时：

- 发生有效跳变沿时，TIMx_CCR1 寄存器会获取计数器的值。
- 将 CC1IF 标志置 1（中断标志）。如果至少发生了两次连续捕获，但 CC1IF 标志未被清零，这样 CC1OF 捕获溢出标志会被置 1。
- 根据 CC1IE 位生成中断。

要处理重复捕获，建议在读出捕获溢出标志之前读取数据。这样可避免丢失在读取捕获溢出标志之后与读取数据之前可能出现的重复捕获信息。

注： *通过软件将 TIMx_EGR 寄存器中的相应 CCxG 位置 1 可生成 IC 中断请求。*

40.3.6 PWM 输入模式（仅适用于 TIM12）

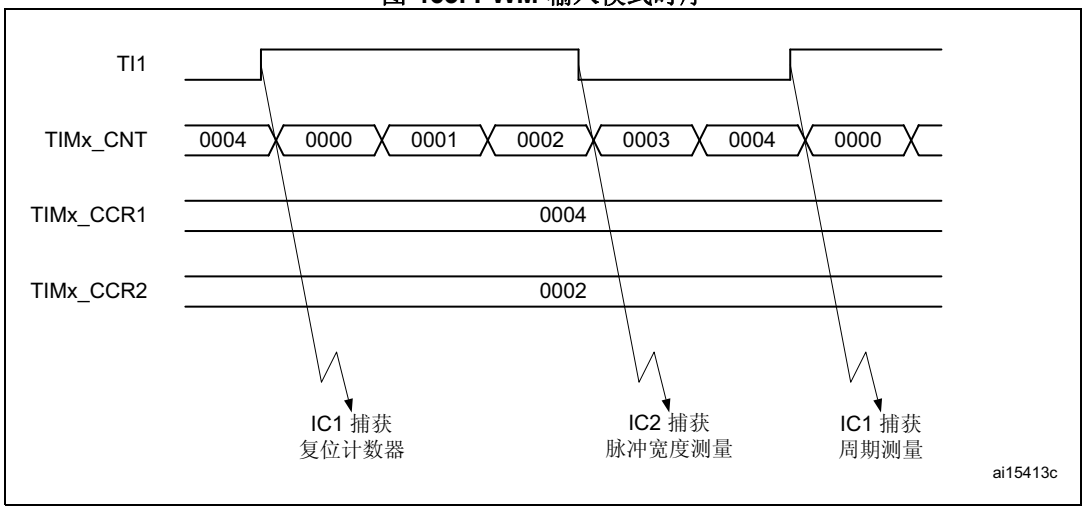
此模式是输入捕获模式的一个特例。其实现步骤与输入捕获模式基本相同，仅存在以下不同之处：

- 两个 ICx 信号被映射至同一个 TIx 输入。
- 这两个 ICx 信号在边沿处有效，但极性相反。
- 选择两个 TIxFP 信号之一作为触发输入，并将从模式控制器配置为复位模式。

例如，可通过以下步骤对应用于 TI1 的 PWM 的周期（位于 TIMx_CCR1 寄存器中）和占空比（位于 TIMx_CCR2 寄存器中）进行测量（取决于 CK_INT 频率和预分频器的值）：

1. 使用 TIMx_TISEL 寄存器中的 TI1SEL[3:0] 位选择正确的 TI1[x] 源（内部或外部）。
2. 选择 TIMx_CCR1 的有效输入：向 TIMx_CCMR1 寄存器中的 CC1S 位写入“01”（选择 TI1）。
3. 选择 TI1FP1 的有效极性（同时用于 TIMx_CCR1 中的捕获和计数器清零）：将 CC1P 位和 CC1NP 位编程为“00”（上升沿有效）。
4. 选择 TIMx_CCR2 的有效输入：向 TIMx_CCMR1 寄存器中的 CC2S 位写入“10”（选择 TI1）。
5. 选择 TI1FP2 的有效极性（用于 TIMx_CCR2 中的捕获）：将 CC2P 位和 CC2NP 位编程为“11”（下降沿有效）。
6. 选择有效触发输入：向 TIMx_SMCR 寄存器中的 TS 位写入“00101”（选择 TI1FP1）。
7. 将从模式控制器配置为复位模式：向 TIMx_SMCR 寄存器中的 SMS 位写入“100”。
8. 使能捕获：向 TIMx_CCER 寄存器中的 CC1E 位和 CC2E 位写入“1”。

图 463. PWM 输入模式时序



1. PWM 输入模式只能与 TIMx_CH1/TIMx_CH2 信号配合使用，因为只有 TI1FP1 和 TI2FP2 与从模式控制器相连。

40.3.7 强制输出模式

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

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

例如：CCxP=“0”（OCx 高电平有效）=> 将 OCx 强制设置为高电平。

通过向 TIMx_CCMRx 寄存器中的 OCxM 位写入“0100”，可将 OCxREF 信号强制设置为低电平。

无论如何，TIMx_CCRx 影子寄存器与计数器之间的比较仍会执行，而且允许将标志置 1。因此可相应发送中断请求。下面的输出比较模式一节对此进行了介绍。

40.3.8 输出比较模式

此功能用于控制输出波形，或指示已经过某一时间段。

当捕获/比较寄存器与计数器之间相匹配时，输出比较功能：

1. 将为相应的输出引脚分配一个可编程值，该值由输出比较模式（TIMx_CCMRx 寄存器中的 OCxM 位）和输出极性（TIMx_CCER 寄存器中的 CCxP 位）定义。匹配时，输出引脚既可保持其电平（OCxM=“0000”），也可设置为有效电平（OCxM=“0001”）、无效电平（OCxM=“0010”）或进行翻转（OCxM=“0011”）。
2. 将中断状态寄存器中的标志置 1（TIMx_SR 寄存器中的 CCxIF 位）。
3. 如果相应中断使能位（TIMx_DIER 寄存器中的 CCxIE 位）置 1，将生成中断。

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

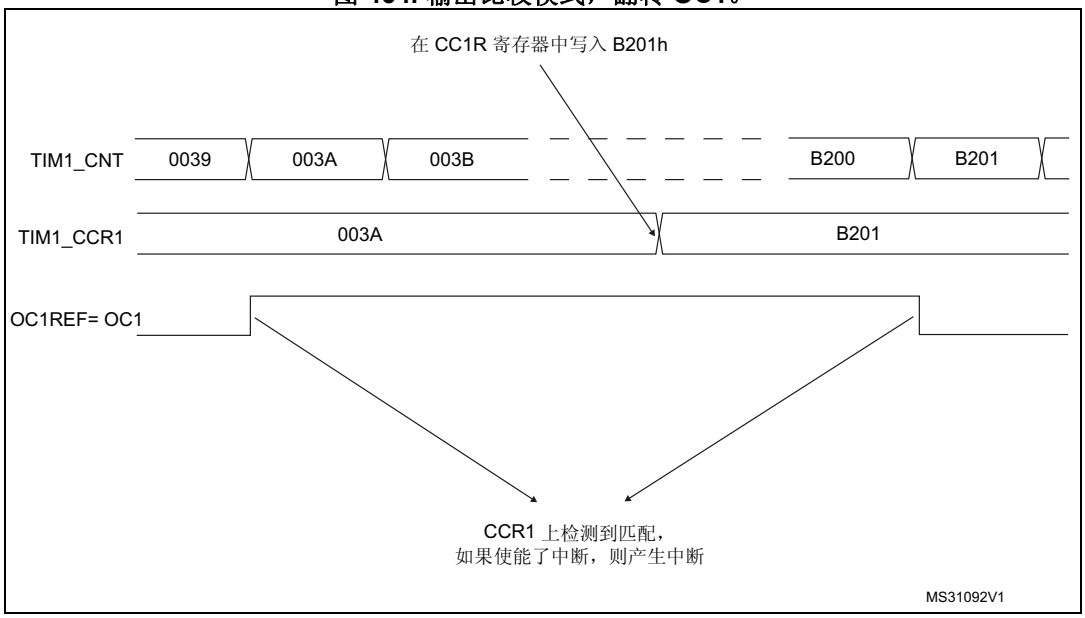
在输出比较模式下，更新事件 UEV 对 OCxREF 和 OCx 输出毫无影响。同步的精度可以达到计数器的一个计数周期。输出比较模式也可用于输出单脉冲（在单脉冲模式下）。

步骤：

1. 选择计数器时钟（内部、外部、预分频器）。
2. 在 TIMx_ARR 和 TIMx_CCRx 寄存器中写入所需数据。
3. 如果要生成中断请求，则需将 CCxIE 位置 1。
4. 选择输出模式。例如：
   - 当 CNT 与 CCRx 匹配时，写入 OCxM =“0011”以翻转 OCx 输出引脚。
   - 写入 OCxPE =“0”以禁止预装载寄存器
   - 写入 CCxP =“0”以选择高电平有效极性
   - 写入 CCxE =“1”以使能输出
5. 通过将 TIMx_CR1 寄存器中的 CEN 位置 1 来使能计数器。

可通过软件随时更新 TIMx_CCRx 寄存器以控制输出波形，前提是未使能预加载寄存器（OCxPE=“0”，否则 TIMx_CCRx 影子寄存器仅在下一更新事件 UEV 发生时进行更新）。*图 464* 给出了一个示例。

图 464. 输出比较模式，翻转 OC1。



40.3.9 PWM 模式

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

各通道可以独立选择 PWM 模式（每个 OCx 输出对应一个 PWM），只需向 TIMx_CCMRx 寄存器的 OCxM 位写入“0110”（PWM 模式 1）或“0111”（PWM 模式 2）。必须通过将 TIMx_CCMRx 寄存器中的 OCxPE 位置 1 使能相应预装载寄存器，最后通过将 TIMx_CR1 寄存器中的 ARPE 位置 1 使能自动重载预装载寄存器（在递增计数或中心对齐模式下）。

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

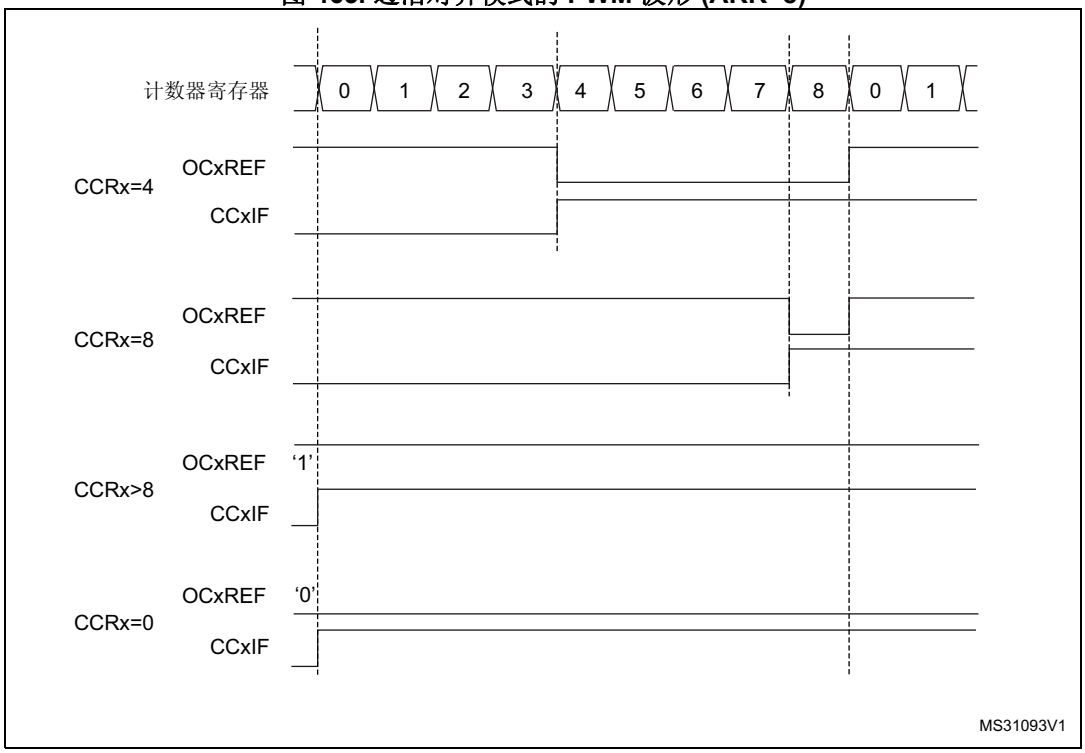
OCx 极性可通过软件来编程（使用 TIMx_CCER 寄存器的 CCxP 位）。可将其编程为高电平有效或低电平有效。OCx 输出通过将 TIMx_CCER 寄存器中的 CCxE 位置 1 来使能。有关详细信息，请参见 TIMx_CCERx 寄存器说明。

在 PWM 模式（1 或 2）下，TIMx_CNT 始终与 TIMx_CCRx 进行比较，以确定 TIMx_CNT ≤ TIMx_CCRx 是否成立。

因为计数器采用递增方式计数，所以定时器能够在边沿对齐模式下生成 PWM。

以下以 PWM 模式 1 为例。只要 TIMx_CNT < TIMx_CCRx，PWM 参考信号 OCxREF 便为高电平，否则为低电平。如果 TIMx_CCRx 中的比较值大于自动重载值（TIMx_ARR 中），则 OCxREF 保持为“1”。如果比较值为 0，则 OCxRef 保持为“0”。*图 465* 举例介绍边沿对齐模式的一些 PWM 波形 (TIMx_ARR=8)。

图 465. 边沿对齐模式的 PWM 波形 (ARR=8)



40.3.10 组合 PWM 模式（仅适用于 TIM12）

在组合 PWM 模式下，生成的两个边沿或中心对齐 PWM 信号的各个脉冲间允许存在可编程延时和相移。频率由 TIMx_ARR 寄存器的值确定，而占空比和延时则由两个 TIMx_CCRx 寄存器确定。产生的信号 OCxREFC 由两个参考 PWM 的逻辑或运算或者逻辑与运算组合组成。

- OC1REFC（或 OC2REFC）由 TIMx_CCR1 和 TIMx_CCR2 寄存器控制

两个通道可以独立选择组合 PWM 模式（每对 CCR 寄存器一个 OCx 输出），只需向 TIMx_CCMRx 寄存器的 OCxM 位写入“1100”（组合 PWM 模式 1）或“1101”（组合 PWM 模式 2）。

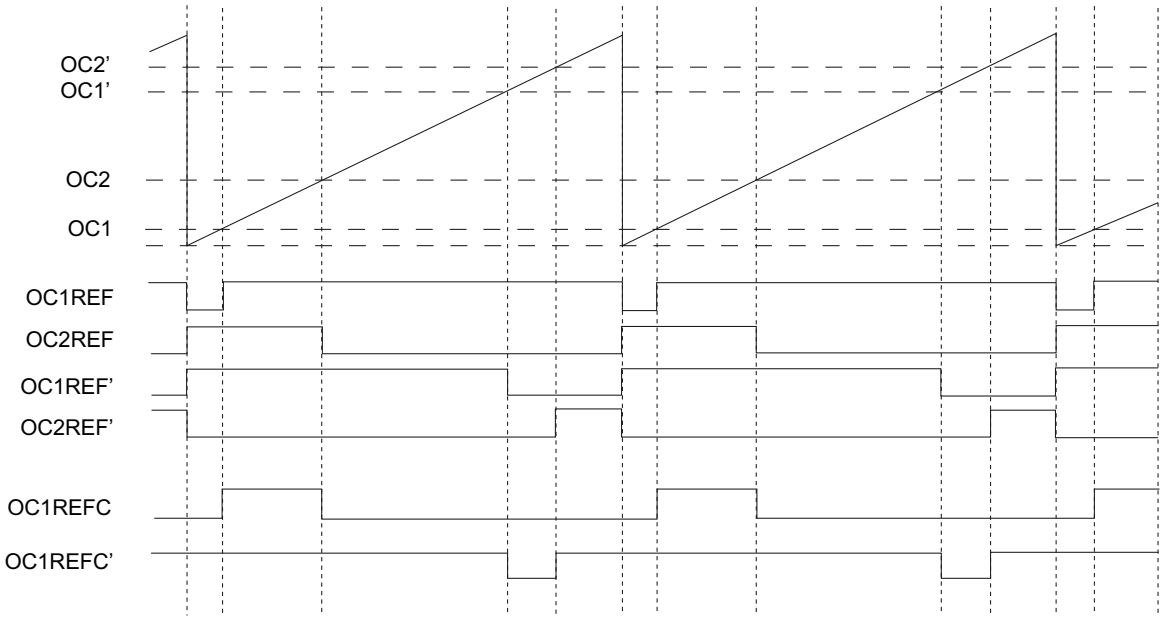
当给定通道用作组合 PWM 通道时，其互补通道必须在相反的 PWM 模式下配置（例如，一个通道在组合 PWM 模式 1 下配置，另一个通道在组合 PWM 模式 2 下配置）。

注： *出于兼容性原因，OCxM[3:0] 位域分为两部分，最高有效位与最低有效的 3 位不相邻。*

图 466 显示了组合 PWM 模式下可以产生的信号示例，通过以下配置可获得这些信号：

- 通道 1 在组合 PWM 模式 2 下配置，
- 通道 2 在 PWM 模式 1 下配置，

图 466. 通道 1 和通道 2 上的组合 PWM 模式



40.3.11 单脉冲模式

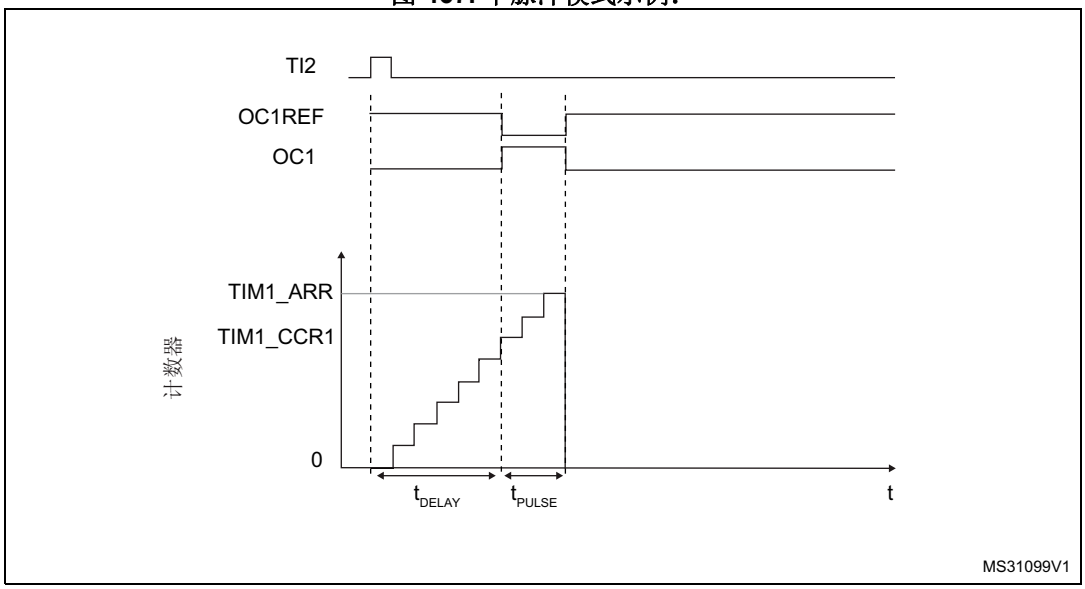
单脉冲模式 (OPM) 是上述模式的一个特例。在这种模式下，计数器可以在一个激励信号的触发下启动，并可在一段可编程的延时后产生一个脉宽可编程的脉冲。

可以通过从模式控制器启动计数器。可以在输出比较模式或 PWM 模式下生成波形。将 TIMx_CR1 寄存器中的 OPM 位置 1，即可选择单脉冲模式。这样，发生下一更新事件 UEV 时，计数器将自动停止。

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

CNT < CCRx ≤ ARR（特别注意，0 < CCRx）

图 467. 单脉冲模式示例：



例如，用户希望达到这样的效果：在 TI2 输入引脚检测到上升沿时，经过 t_{DELAY} 的延迟，在 OC1 上产生一个长度为 t_{PULSE} 的正脉冲。

使用 TI2FP2 作为触发 1：

1. 使用 TIMx_TISEL 寄存器中的 TI2SEL[3:0] 位选择正确的 TI2[x] 源（内部或外部）。
2. 在 TIMx_CCMR1 寄存器中写入 CC2S=“01”，以将 TI2FP2 映射到 TI2。
3. 在 TIMx_CCER 寄存器中写入 CC2P=“0”和 CC2NP=“0”，使 TI2FP2 能够检测上升沿。
4. 在 TIMx_SMCR 寄存器中写入 TS=“00110”，以将 TI2FP2 配置为从模式控制器的触发 (TRGI)。
5. 在 TIMx_SMCR 寄存器中写入 SMS=“110”（触发模式），以使用 TI2FP2 启动计数器。

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

- t_{DELAY} 由写入 TIMx_CCR1 寄存器的值定义。
- t_{PULSE} 由自动重载值与比较值之差 (TIMx_ARR - TIMx_CCR1) 来定义。
- 假设希望产生这样的波形：信号在发生比较匹配时从“0”变为“1”，在计数器达到自动重载值时由“1”变为“0”。为此，应在 TIMx_CCMR1 寄存器中写入 OC1M=“0111”，以使能 PWM 模式 2。如果需要，可选择在 TIMx_CCMR1 寄存器的 OC1PE 和 TIMx_CR1 寄存器的 ARPE 中写入“1”，以使能预装载寄存器。这种情况下，必须在 TIMx_CCR1 寄存器中写入比较值并在 TIMx_ARR 寄存器中写入自动重载值，通过将 UG 位置 1 来产生更新，然后等待 TI2 上的外部触发事件。本例中，CC1P 的值为“0”。

由于仅需要 1 个脉冲（单脉冲模式），因此应向 TIMx_CR1 寄存器的 OPM 位写入“1”，以便在发生下一更新事件（计数器从自动重载值返回到 0）时使计数器停止计数。TIMx_CR1 寄存器中的 OPM 位置“0”时，即选择重复模式。

特殊情况：OCx 快速使能

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

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

40.3.12 可再触发单脉冲模式 (OPM)（仅限 TIM12）

该模式可以启动计数器来响应激励信号，并且能产生长度可编程的脉冲，但与*第 40.3.11 节：单脉冲模式*所描述的不可再触发单脉冲模式间存在以下差别：

- 发生触发时，脉冲立即产生（无可编程延时）
- 如果在上一个触发完成前发生新的触发，脉冲将延长

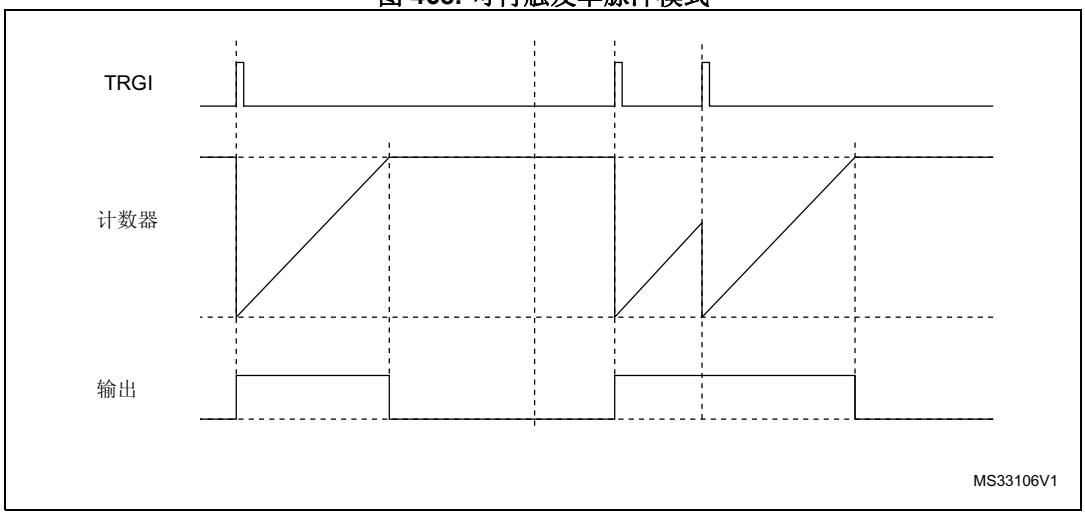
定时器必须处于从模式，TIMx_SMCR 寄存器中的位 SMS[3:0] =“1000”（组合复位 + 触发模式），针对可再触发 OPM 模式 1 或模式 2 将 OCxM[3:0] 位设置为“1000”或“1001”。

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

注： *出于兼容性原因，OCxM[3:0] 和 SMS[3:0] 位域分为两部分，最高有效位与最低有效的 3 位不相邻。*

此模式不能与中心对齐 PWM 模式一起使用。在 TIMx_CR1 中必须设置 CMS[1:0] = 00。

图 468. 可再触发单脉冲模式



40.3.13 UIF 位重映射

TIMx_CR1 寄存器中的 IUFREMAP 位强制将更新中断标志 UIF 连续复制到定时计数器寄存器的位 31 (TIMxCNT[31]) 中。这样便可自动读取计数器值以及由 UIFCPY 标志发出的电位翻转条件。在特定情况下，这可避免在后台任务（计数器读）和中断（更新中断）之间共享处理时产生竞争条件，从而简化计算。

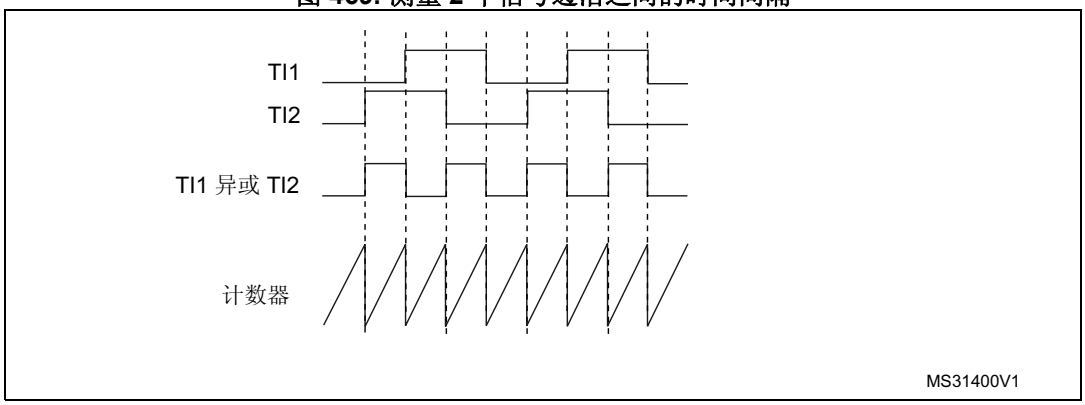
UIF 和 UIFCPY 标志使能之间没有延迟。

40.3.14 定时器输入异或功能

通过 TIMx_CR2 寄存器中的 TI1S 位，可将通道 1 的输入滤波器连接到异或门的输出，从而将 TIMx_CH1 和 TIMx_CH2 这两个输入引脚组合在一起。

异或输出可与触发或输入捕获等所有定时器输入功能配合使用。这样可用于测量两个输入信号边沿之间的间隔，如下面的*图 469* 所示。

图 469. 测量 2 个信号边沿之间的时间间隔



40.3.15 TIM12 外部触发同步

TIM12 定时器可与外部触发以下列模式实现同步：复位模式、门控模式和触发模式。

从模式：复位模式

当触发输入信号发生变化时，计数器及其预分频器可重新初始化。此外，如果 TIMx_CR1 寄存器中的 URS 位为 0，则会生成更新事件 UEV。然后，所有预装载寄存器（TIMx_ARR 和 TIMx_CCRx）都将更新。

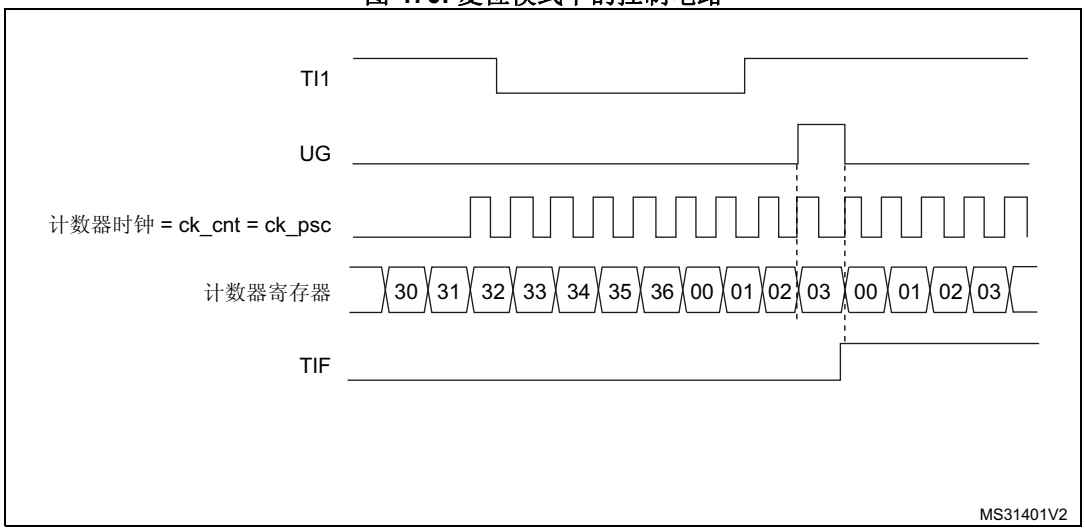
在以下示例中，TI1 输入上出现上升沿时，递增计数器清零：

1. 将通道 1 配置为检测 TI1 的上升沿。配置输入滤波带宽（本例中不需要任何滤波器，因此保持 IC1F=“0000”）。由于捕获预分频器不用于触发操作，因此无需对其进行配置。CC1S 位只选择输入捕获源，即 TIMx_CCMR1 寄存器中的 CC1S = “01”。将 TIMx_CCER 寄存器中的 CC1P 和 CC1NP 编程为“00”，以确定极性（仅检测上升沿）。
2. 在 TIMx_SMCR 寄存器中写入 SMS=“100”，将定时器配置为复位模式。在 TIMx_SMCR 寄存器中写入 TS= “00101”，选择 TI1 作为输入源。
3. 通过在 TIMx_CR1 寄存器中写入 CEN=“1”来启动动计数器。

计数器开始根据内部时钟计数，然后正常运转，直到出现 TI1 上升沿。当 TI1 出现上升沿时，计数器清零，然后重新从 0 开始计数。同时，触发标志（TIMx_SR 寄存器中的 TIF 位）置 1，使能中断，还可发送中断请求（取决于 TIMx_DIER 寄存器中的 TIE 位）。

下图显示了自动重载寄存器 TIMx_ARR=0x36 时的相关行为。TI1 的上升沿与实际计数器复位之间的延迟是由于 TI1 输入的重新同步电路引起的。

图 470. 复位模式下的控制电路



从模式：门控模式

输入信号的电平可用来使能计数器。

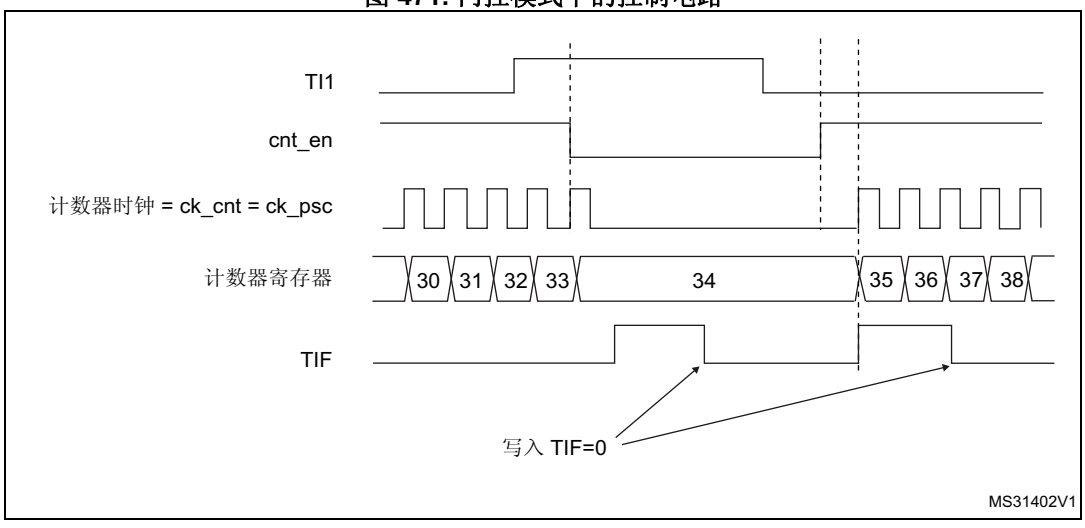
在以下示例中，递增计数器仅在 TI1 输入为低电平时计数：

1. 将通道 1 配置为检测 TI1 上的低电平。配置输入滤波带宽（本例中不需要任何滤波器，因此保持 IC1F=“0000”）。由于捕获预分频器不用于触发操作，因此无需对其进行配置。CC1S 位只选择输入捕获源，即 TIMx_CCMR1 寄存器中的 CC1S =“01”。将 TIMx_CCER 寄存器中的 CC1P 和 CC1NP 分别编程为“1”和“0”，以确定极性（仅检测低电平）。
2. 在 TIMx_SMCR 寄存器中写入 SMS=“101”，将定时器配置为门控模式。在 TIMx_SMCR 寄存器中写入 TS= “00101”，选择 TI1 作为输入源。
3. 在 TIMx_CR1 寄存器中写入 CEN=“1”，使能计数器（在门控模式下，如果 CEN=“0”，则不论触发输入电平如何，计数器都不启动）。

只要 TI1 为低电平，计数器就开始根据内部时钟计数，直到 TI1 变为高电平时停止计数。计数器启动或停止时，TIMx_SR 寄存器中的 TIF 标志都会置 1。

TI1 的上升沿与实际计数器停止之间的延迟是由于 TI1 输入的重新同步电路引起的。

图 471. 门控模式下的控制电路



从模式：触发模式

所选输入上发生某一事件时可以启动计数器。

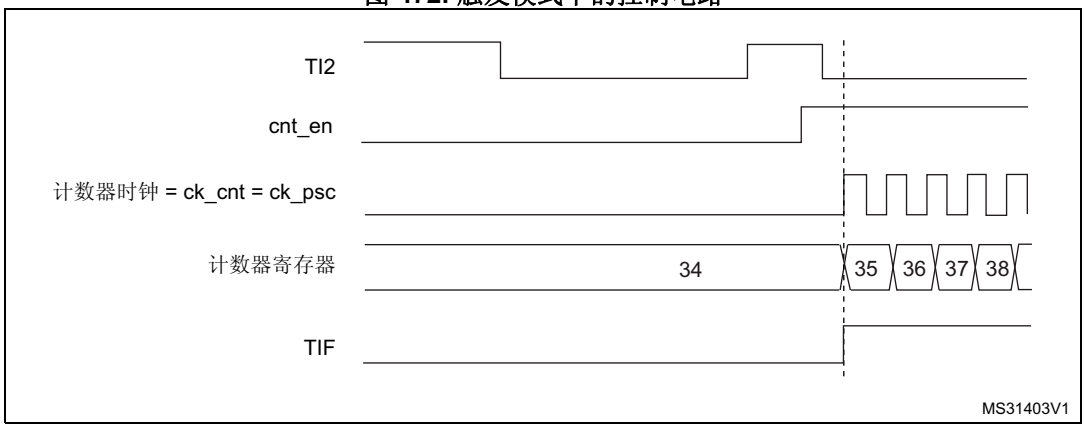
在以下示例中，TI2 输入上出现上升沿时，递增计数器启动：

1. 将通道 2 配置为检测 TI2 上的上升沿。配置输入滤波带宽（本例中不需要任何滤波器，因此保持 IC2F=“0000”）。由于捕获预分频器不用于触发操作，因此无需对其进行配置。CC2S 位只选择输入捕获源，即 TIMx_CCMR1 寄存器中的 CC2S=“01”。将 TIMx_CCER 寄存器中的 CC2P 和 CC2NP 分别编程为“1”和“0”，以确定极性（仅检测低电平）。
2. 在 TIMx_SMCR 寄存器中写入 SMS=“110”，将定时器配置为触发模式。在 TIMx_SMCR 寄存器中写入 TS=“00110”，选择 TI2 作为输入源。

当 TI2 出现上升沿时，计数器开始根据内部时钟计数，并且 TIF 标志置 1。

TI2 的上升沿与实际计数器启动之间的延迟是由于 TI2 输入的重新同步电路引起的。

图 472. 触发模式下的控制电路



40.3.16 从模式——组合复位 + 触发模式

在这种情况下，在出现所选触发输入 (TRGI) 上升沿时，重新初始化计数器，生成一个寄存器更新事件，并启动计数器。

该模式用于单脉冲模式。

40.3.17 定时器同步(TIM12)

TIM 定时器从内部链接在一起，以实现定时器同步或级联。有关详细信息，请参见*第 39.3.19 节：定时器同步*。

注： *必须先使能从定时器的时钟，才能从主定时器接收事件；从主定时器接收触发信号时，不得实时更改从定时器的时钟。*

40.3.18 调试模式

当微控制器进入调试模式（Cortex®-M7 with FPU 内核停止）时，TIMx 计数器会根据 DBGMCU 模块中的 DBG_TIMx_STOP 配置位选择继续正常工作或者停止工作。有关详细信息，请参见*第 60.5.8 节：微控制器调试单元 (DBGMCU)*。

40.4 TIM12 寄存器

有关寄存器说明中使用的缩写，请参见*第 1.1 节*。

外设寄存器的写访问仅支持半字（16 位）或字（32 位）。而读访问可支持字节（8 位）、半字（16 位）或字（32 位）。

40.4.1 TIM12控制寄存器 1 (TIMx_CR1)

TIM12 control register 1

偏移地址：0x00

复位值：0x0000

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

位 15:12 保留，必须保持复位值。

位 11 **UIFREMAP**：UIF 状态位重映射 (UIF status bit remapping)
0：无重映射。UIF 状态位不复制到 TIMx_CNT 寄存器的位 31。
1：使能重映射。UIF 状态位复制到 TIMx_CNT 寄存器的位 31。

位 10 保留，必须保持复位值。

位 9:8 **CKD**：时钟分频 (Clock division)
此位域指示定时器时钟 (CK_INT) 频率与数字滤波器所使用的采样时钟 (TIx) 之间的分频比
00：$t_{DTS} = t_{CK_INT}$
01：$t_{DTS} = 2 \times t_{CK_INT}$
10：$t_{DTS} = 4 \times t_{CK_INT}$
11：保留

位 7 **ARPE**：自动重载预装载使能 (Auto-reload preload enable)
0：TIMx_ARR 寄存器不进行缓冲。
1：TIMx_ARR 寄存器进行缓冲。

位 6:4 保留，必须保持复位值。

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

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

位 1 **UDIS**：更新禁止 (Update disable)

此位由软件置 1 和清零，用以使能/禁止更新事件 (UEV) 生成。

0：使能 UEV。UEV 可通过以下事件之一生成：

– 计数器上溢

– 将 UG 位置 1

然后更新影子寄存器的值。

1：禁止 UEV。不会生成 UEV，各影子寄存器的值（ARR、PSC 和 CCRx）保持不变。如果 UG 位置 1，则会重新初始化计数器和预分频器。

位 0 **CEN**：计数器使能 (Counter enable)

0：禁止计数器

1：使能计数器

在单脉冲模式下，当发生更新事件时会自动将 CEN 位清零。

注： 只有事先通过软件将 CEN 位置 1，才可以使用外部时钟和门控模式。而触发模式可通过硬件自动将 CEN 位置 1。

40.4.2 TIM12 从模式控制寄存器 (TIMx_SMCR)

TIM12 slave mode control register

偏移地址：0x08

复位值：0x0000

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

位 31:22 保留，必须保持复位值。

位 21:20 **TS[4:3]**：触发选择——位 4:3 (Trigger selection - bit 4:3)

请参见 TS[4:0] 说明——位 6:4

位 19:17 保留，必须保持复位值。

位 16 **SMS[3]**：从模式选择——位 3 (Slave mode selection - bit 3)

请参见 SMS 说明——位 2:0

位 15:8 保留，必须保持复位值。

位 7 **MSM**：主/从模式 (Master/slave mode)

0：不执行任何操作

1：当前定时器的触发输入事件 (TRGI) 的动作被推迟，以使当前定时器与其从定时器实现完美同步（通过 TRGO）。此设置适用于要在发生单个外部事件时对多个定时器进行同步的情况。

位 6:4 **TS[4:0]**：触发选择 (Trigger selection)

TS[4:0] 位域可选择将要用于同步计数器的触发输入。

00000：内部触发 0 (ITR0)

00001：内部触发 1 (ITR1)

00010：内部触发 2 (ITR2)

00011：内部触发 3 (ITR3)

00100：TI1 边沿检测器 (TI1F_ED)

00101：滤波后的定时器输入 1 (TI1FP1)

00110：滤波后的定时器输入 2 (TI2FP2)

其它：保留

有关各定时器的 ITRx 含义的更多详细信息，请参见*第 1586 页的表 320：TIMx 内部触发连接*。

注：　这些位只能在未使用的情况下（例如，SMS=“000”时）进行更改，以避免转换时出现错误的边沿检测。

位 3 保留，必须保持复位值。

位 2:0 **SMS**：从模式选择 (Slave mode selection)

选择外部信号时，触发信号 (TRGI) 的有效边沿与外部输入上所选的极性相关（请参见输入控制寄存器和控制寄存器说明）。

0000：禁止从模式——如果 CEN = “1”，预分频器时钟直接由内部时钟提供。

0001：保留

0010：保留

0011：保留

0100：复位模式——在出现所选触发输入 (TRGI) 上升沿时，重新初始化计数器并生成一个寄存器更新事件。

0101：门控模式——触发输入 (TRGI) 为高电平时使能计数器时钟。只要触发输入变为低电平，计数器立即停止计数（但不复位）。计数器的启动和停止都被控制。

0110：触发模式——触发信号 TRGI 出现上升沿时启动计数器（但不复位）。只控制计数器的启动。

0111：外部时钟模式 1——由所选触发信号 (TRGI) 的上升沿提供计数器时钟。

1000：组合复位 + 触发模式——在出现所选触发输入 (TRGI) 上升沿时，重新初始化计数器，生成一个寄存器更新事件并启动计数器。

其他代码：保留。

注：　如果将 TI1F_ED 选作触发输入（TS= “00100”），则不得使用门控模式。实际上，TI1F 每次转换时，TI1F_ED 都输出 1 个脉冲，而门控模式检查的则是触发信号的电平。

注：　必须先使能从定时器的时钟，才能从主定时器接收事件；从主定时器接收触发信号时，不得实时更改从定时器的时钟。

表 320. TIMx 内部触发连接

从 TIM	ITR0 （TS = “00000”）	ITR1 （TS = “00001”）	ITR2 （TS = “00010”）	ITR3 （TS = “00011”）
TIM12	TIM4	TIM5	TIM13 OC1	TIM14 OC1

40.4.3 TIM12 中断使能寄存器 (TIMx_DIER)

TIM12 Interrupt enable register

偏移地址：0x0C

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TIE	Res.	Res.	Res.	CC2IE	CC1IE	UIE
									rw				rw	rw	rw

位 15:7 保留，必须保持复位值。

位 6 **TIE**：触发中断使能 (Trigger interrupt enable)
0：禁止触发中断。
1：使能触发中断。

位 5:3 保留，必须保持复位值。

位 2 **CC2IE**：捕获/比较 2 中断使能 (Capture/Compare 2 interrupt enable)
0：禁止 CC2 中断。
1：使能 CC2 中断。

位 1 **CC1IE**：捕获/比较 1 中断使能 (Capture/Compare 1 interrupt enable)
0：禁止 CC1 中断。
1：使能 CC1 中断。

位 0 **UIE**：更新中断使能 (Update interrupt enable)
0：禁止更新中断。
1：使能更新中断。

40.4.4 TIM12 状态寄存器 (TIMx_SR)

TIM12 status register

偏移地址：0x10

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	CC2OF	CC1OF	Res.	Res.	TIF	Res.	Res.	Res.	CC2IF	CC1IF	UIF
					rc_w0	rc_w0			rc_w0				rc_w0	rc_w0	rc_w0

位 15:11 保留，必须保持复位值。

位 10 **CC2OF**：捕获/比较 2 重复捕获标志 (Capture/compare 2 overcapture flag)
请参见 CC1OF 说明

位 9 **CC1OF**：捕获/比较 1 重复捕获标志 (Capture/Compare 1 overcapture flag)
仅当将相应通道配置为输入捕获模式时，此标志位才会由硬件置 1。通过软件写入“0”可将该位清零。
0：未检测到重复捕获。
1：TIMx_CCR1 寄存器中已捕获到计数器值且 CC1IF 标志已置 1。

位 8:7 保留，必须保持复位值。

位 6 **TIF**：触发中断标志 (Trigger interrupt flag)
在除门控模式以外的所有模式下，当使能从模式控制器后在 TRGI 输入上检测到有效边沿时，该标志将由硬件置 1。选择门控模式时，该标志将在计数器启动或停止时置 1。但需要通过软件清零。
0：未发生触发事件。
1：触发中断挂起。

位 5:3 保留，必须保持复位值。

位 2 **CC2IF**：捕获/比较 2 中断标志 (Capture/Compare 2 interrupt flag)
请参见 CC1IF 说明

位 1 **CC1IF**：捕获/比较 1 中断标志 (Capture/compare 1 interrupt flag)
如果通道 CC1 配置为输出：
当计数器与比较值匹配时，此标志由硬件置 1。但需要通过软件清零。
0：不匹配。
1：TIMx_CNT 计数器的值与 TIMx_CCR1 寄存器的值匹配。当 TIMx_CCR1 的值大于 TIMx_ARR 的值时，CC1IF 位将在计数器发生上溢时变为 1。
如果通道 CC1 配置为输入：
此位将在发生捕获事件时由硬件置 1。通过软件或读取 TIMx_CCR1 寄存器将该位清零。
0：未发生输入捕获事件。
1：TIMx_CCR1 寄存器中已捕获到计数器值（IC1 上已检测到与所选极性匹配的边沿）。

位 0 **UIF**：更新中断标志 (Update interrupt flag)
该位在发生更新事件时通过硬件置 1。但需要通过软件清零。
0：未发生更新。
1：更新中断挂起。该位在以下情况下更新寄存器时由硬件置 1：
– 上溢且当 TIMx_CR1 寄存器中 UDIS =“0”时。
– 当由于 TIMx_CR1 寄存器中 URS =“0”且 UDIS =“0”而通过软件使用 TIMx_EGR 寄存器中的 UG 位重新初始化 CNT 时。
– 当由于 TIMx_CR1 寄存器中 URS =“0”且 UDIS =“0”而通过触发事件（请参见同步控制寄存器说明）重新初始化 CNT 时。

40.4.5 TIM12 事件生成寄存器 (TIMx_EGR)

TIM12 event generation register

偏移地址：0x14

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TG	Res.	Res.	Res.	CC2G	CC1G	UG
									w				w	w	w

位 15:7 保留，必须保持复位值。

位 6 **TG**：触发生成 (Trigger generation)
此位由软件置 1 以生成事件，并由硬件自动清零。
0：不执行任何操作
1：TIMx_SR 寄存器中的 TIF 标志置 1。使能后可发生相关中断

位5:3 保留，必须保持复位值。

位 2 **CC2G**：捕获/比较 2 生成 (Capture/compare 2 generation)
请参见 CC1G 说明

位 1 **CC1G**：捕获/比较 1 生成 (Capture/Compare 1 generation)
此位由软件置 1 以生成事件，并由硬件自动清零。
0：不执行任何操作
1：通道 1 上生成捕获/比较事件：
如果通道 CC1 配置为输出：
使能后，CC1IF 标志置 1 并发送相应中断。
如果通道 CC1 配置为输入：
TIMx_CCR1 寄存器中将捕获到计数器的当前值。使能后，CC1IF 标志置 1 并发送相应中断。如果 CC1IF 标志已为高电平，CC1OF 标志将置 1。

位 0 **UG**：更新生成 (Update generation)
该位可通过软件置 1，并由硬件自动清零。
0：不执行任何操作
1：重新初始化计数器并生成寄存器更新事件。预分频器计数器也将清零，但预分频比不受影响。计数器清零。

40.4.6 TIM12 捕获/比较模式寄存器 1 (TIMx_CCMR1)

TIM12 capture/compare mode register 1

偏移地址：0x18

复位值：0x0000

这些通道可用于输入（捕获模式）或输出（比较模式）模式。通道方向通过配置相应的 CCxS 位进行定义。该寄存器的所有其它位在输入模式和输出模式下的功能不同。对于任一给定位，OCxx 用于说明通道配置为输出模式时该位对应的功能，ICxx 则用于说明通道配置为输入模式时该位对应的功能。因此，必须注意同一个位在输入阶段和输出阶段具有不同的含义。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC2M [3]	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC1M [3]
							Res.								Res.
							rw								rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	OC2M[2:0]			OC2PE	OC2FE	CC2S[1:0]		Res.	OC1M[2:0]			OC1PE	OC1FE	CC1S[1:0]	
IC2F[3:0]				IC2PSC[1:0]				IC1F[3:0]				IC1PSC[1:0]			
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

输出比较模式

位 31:25 保留，始终读为 0

位 24 **OC2M[3]**：输出比较 2 模式——位 3 (Output Compare 2 mode - bit 3)
请参见 OC2M 说明——位 14:12

位 23:17 保留，始终读为 0

位 16 **OC1M[3]**：输出比较 1 模式——位 3 (Output Compare 1 mode - bit 3)
请参见 OC1M 说明——位 6:4

位 15 保留，必须保持复位值。

位 14:12 **OC2M[2:0]**：输出比较 2 模式 (Output Compare 2 mode)
有关位说明，请参见 OC1M[3:0]。

位 11 **OC2PE**：输出比较 2 预装载使能 (Output Compare 2 preload enable)

位 10 **OC2FE**：输出比较 2 快速使能 (Output Compare 2 fast enable)

位 9:8 **CC2S[1:0]**：捕获/比较 2 选择 (Capture/Compare 2 selection)
此位域定义通道方向（输入/输出）以及所使用的输入。
00：CC2 通道配置为输出
01：CC2 通道配置为输入，IC2 映射到 TI2 上
10：CC2 通道配置为输入，IC2 映射到 TI1 上
11：CC2 通道配置为输入，IC2 映射到 TRC 上。此模式仅在通过 TS 位（TIMx_SMCR 寄存器）选择内部触发输入时有效
注： 仅当通道关闭时（TIMx_CCER 中的 CC2E = 0），才可向 CC2S 位写入数据。

位 7 保留，必须保持复位值。

位 6:4 **OC1M[3:0]**：输出比较 1 模式 (Output Compare 1 mode)（请参见 OC1M[3] 的位 16）

这些位定义提供 OC1 的输出参考信号 OC1REF 的行为。OC1REF 为高电平有效，而 OC1 的有效电平则取决于 CC1P。

0000：冻结——输出比较寄存器 TIMx_CCR1 与计数器 TIMx_CNT 进行比较不会对输出造成任何影响。（该模式用于生成时基）。

0001：将通道 1 设置为匹配时输出有效电平。当计数器 TIMx_CNT 与捕获/比较寄存器 1 (TIMx_CCR1) 匹配时，OC1REF 信号强制变为高电平。

0010：将通道 1 设置为匹配时输出无效电平。当计数器 TIMx_CNT 与捕获/比较寄存器 1 (TIMx_CCR1) 匹配时，OC1REF 信号强制变为低电平。

0011：翻转——TIMx_CNT=TIMx_CCR1 时，OC1REF 发生翻转

0100：强制变为无效电平——OC1REF 强制变为低电平

0101：强制变为有效电平——OC1REF 强制变为高电平

0110：PWM 模式 1——只要 TIMx_CNT<TIMx_CCR1，通道 1 便为有效状态，否则为无效状态

0111：PWM 模式 2——只要 TIMx_CNT<TIMx_CCR1，通道 1 便为无效状态，否则为有效状态

1000：可再触发 OPM 模式 1——通道为有效状态，直至 （在 TRGI 信号上）检测到触发事件。然后，与在 PWM 模式 1 下一样，进行比较，通道会在下一次更新时再次变为有效状态。

1001：可再触发 OPM 模式 2——通道为无效状态，直至 （在 TRGI 信号上）检测到触发事件。然后，与在 PWM 模式 2 下一样，进行比较，通道会在下一次更新时再次变为无效状态。

1010：保留，

1011：保留，

1100：组合 PWM 模式 1——OC1REF 与在 PWM 模式 1 下的行为相同。OC1REFC 是 OC1REF 和 OC2REF 的逻辑或运算结果。

1101：组合 PWM 模式 2——OC1REF 与在 PWM 模式 2 下的行为相同。OC1REFC 是 OC1REF 和 OC2REF 的逻辑与运算结果。

1110：保留，

1111：保留

注： 在 PWM 模式 1 或 PWM 模式 2 下，仅当比较结果发生改变或输出比较模式由“冻结”模式切换到“PWM”模式时，OCREF 电平才会发生更改。

位 3 **OC1PE**：输出比较 1 预装载使能 (Output Compare 1 preload enable)

0：禁止与 TIMx_CCR1 相关的预装载寄存器。可随时向 TIMx_CCR1 写入数据，写入后将立即使用新值。

1：使能与 TIMx_CCR1 相关的预装载寄存器。可读/写访问预装载寄存器。TIMx_CCR1 预装载值在每次生成更新事件时都会装载到有效寄存器中

注： 只有单脉冲模式下才可在未验证预装载寄存器的情况下使用 PWM 模式（TIMx_CR1 寄存器中的 OPM 位置 1）。其它情况下则无法保证该行为。

位 2 **OC1FE**：输出比较 1 快速使能 (Output Compare 1 fast enable)

此位用于加快触发输入事件对 CC 输出的影响。

0：即使触发开启，CC1 也将根据计数器和 CCR1 值正常工作。触发输入出现边沿时，激活 CC1 输出的最短延迟时间为 5 个时钟周期

1：触发输入上出现有效边沿相当于 CC1 输出上的比较匹配。随后，无论比较结果如何，OC 都设置为比较电平。采样触发输入和激活 CC1 输出的延迟时间缩短为 3 个时钟周期。仅当通道配置为 PWM1 或 PWM2 模式时，OC1FE 才会起作用。

位 1:0 **CC1S[1:0]**：捕获/比较 1 选择 (Capture/Compare 1 Selection)

此位域定义通道方向（输入/输出）以及所使用的输入。

00：CC1 通道配置为输出

01：CC1 通道配置为输入，IC1 映射到 TI1 上

10：CC1 通道配置为输入，IC1 映射到 TI2 上

11：CC1 通道配置为输入，IC1 映射到 TRC 上。此模式仅在通过 TS 位（TIMx_SMCR 寄存器）选择内部触发输入时有效

注： 仅当通道关闭时（TIMx_CCER 中的 CC1E = 0），才可向 CC1S 位写入数据。

输入捕获模式

位 15:12 **IC2F**：输入捕获 2 滤波器 (Input capture 2 filter)

位 11:10 **IC2PSC[1:0]**：输入捕获 2 预分频器 (Input capture 2 prescaler)

位 9:8 **CC2S**：捕获/比较 2 选择 (Capture/Compare 2 selection)

此位域定义通道方向（输入/输出）以及所使用的输入。
00：CC2 通道配置为输出
01：CC2 通道配置为输入，IC2 映射到 TI2 上
10：CC2 通道配置为输入，IC2 映射到 TI1 上
11：CC2 通道配置为输入，IC2 映射到 TRC 上。此模式仅在通过 TS 位（TIMx_SMCR 寄存器）选择内部触发输入时有效

注： 仅当通道关闭时（TIMx_CCER 中的 CC2E = 0），才可向 CC2S 位写入数据。

位 7:4 **IC1F**：输入捕获 1 滤波器 (Input capture 1 filter)

此位域可定义 TI1 输入的采样频率和适用于 TI1 的数字滤波器带宽。数字滤波器由事件计数器组成，每 N 个连续事件才视为一个有效输出边沿：
0000：无滤波器，按 f_{DTS} 频率进行采样
0001：$f_{SAMPLING}=f_{CK_INT}$，N=2
0010：$f_{SAMPLING}=f_{CK_INT}$，N=4
0011：$f_{SAMPLING}=f_{CK_INT}$，N=8
0100：$f_{SAMPLING}=f_{DTS}/2$，N=6
0101：$f_{SAMPLING}=f_{DTS}/2$，N=8
0110：$f_{SAMPLING}=f_{DTS}/4$，N=6
0111：$f_{SAMPLING}=f_{DTS}/4$，N=8
1000：$f_{SAMPLING}=f_{DTS}/8$，N=6
1001：$f_{SAMPLING}=f_{DTS}/8$，N=8
1010：$f_{SAMPLING}=f_{DTS}/16$，N=5
1011：$f_{SAMPLING}=f_{DTS}/16$，N=6
1100：$f_{SAMPLING}=f_{DTS}/16$，N=8
1101：$f_{SAMPLING}=f_{DTS}/32$，N=5
1110：$f_{SAMPLING}=f_{DTS}/32$，N=6
1111：$f_{SAMPLING}=f_{DTS}/32$，N=8

位 3:2 **IC1PSC**：输入捕获 1 预分频器 (Input capture 1 prescaler)

此位域定义 CC1 输入 (IC1) 的预分频比。
只要 CC1E=“0”（TIMx_CCER 寄存器），预分频器便立即复位。
00：无预分频器，捕获输入上每检测到一个边沿便执行捕获
01：每发生 2 个事件便执行一次捕获
10：每发生 4 个事件便执行一次捕获
11：每发生 8 个事件便执行一次捕获

位 1:0 **CC1S[1:0]**：捕获/比较 1 选择 (Capture/Compare 1 Selection)

此位域定义通道方向（输入/输出）以及所使用的输入。
00：CC1 通道配置为输出
01：CC1 通道配置为输入，IC1 映射到 TI1 上
10：CC1 通道配置为输入，IC1 映射到 TI2 上
11：CC1 通道配置为输入，IC1 映射到 TRC 上。此模式仅在通过 TS 位（TIMx_SMCR 寄存器）选择内部触发输入时有效

注： 仅当通道关闭时（TIMx_CCER 中的 CC1E = 0），才可向 CC1S 位写入数据。

40.4.7 TIM12 捕获/比较使能寄存器 (TIMx_CCER)

TIM12 capture/compare enable register

偏移地址：0x20

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CC2NP	Res.	CC2P	CC2E	CC1NP	Res.	CC1P	CC1E
								rw		rw	rw	rw		rw	rw

位 15:8 保留，必须保持复位值。

位 7 **CC2NP**：捕获/比较 2 互补输出极性 (Capture/Compare 2 complementary output Polarity)
请参见 CC1NP 说明

位 6 保留，必须保持复位值。

位 5 **CC2P**：捕获/比较 2 输出极性 (Capture/Compare 2 output Polarity)
请参见 CC1P 说明

位 4 **CC2E**：捕获/比较 2 输出使能 (Capture/Compare 2 output enable)
请参见 CC1E 说明

位 3 **CC1NP**：捕获/比较 1 互补输出极性 (Capture/Compare 1 complementary output Polarity)
CC1 通道配置为输出：CC1NP 必须保持清零
CC1 通道配置为输入：CC1NP 与 CC1P 配合使用可定义 TI1FP1/TI2FP1 的极性（请参见 CC1P 说明）。

位 2 保留，必须保持复位值。

位 1 **CC1P**：捕获/比较 1 输出极性 (Capture/Compare 1 output Polarity)。
CC1 通道配置为输出：
0：OC1 高电平有效。
1：OC1 低电平有效。
CC1 通道配置为输入：
CC1P 和 CC1NP 位针对捕获操作选择 TI1FP1 极性。
00：未反相/上升沿触发
电路对 TIxFP1 上升沿敏感（在复位模式、外部时钟模式或触发模式下执行捕获或触发操作），TIxFP1 未反相（在门控模式或编码器模式下执行触发操作）。
01：反相/下降沿触发
电路对 TIxFP1 下降沿敏感（在复位模式、外部时钟模式或触发模式下执行捕获或触发操作），TIxFP1 反相（在门控模式或编码器模式下执行触发操作）。
10：保留，不使用此配置。
11：未反相/上升沿和下降沿均触发
电路对 TIxFP1 上升沿和下降沿都敏感（在复位模式、外部时钟模式或触发模式下执行捕获或触发操作），TIxFP1 未反相（在门控模式下执行触发操作）。编码器模式下不得使用此配置。

位 0 **CC1E**：捕获/比较 1 输出使能 (Capture/Compare 1 output enable)。
CC1 通道配置为输出：
0：关闭——OC1 无效。
1：开启——在相应输出引脚上输出 OC1 信号。
CC1 通道配置为输入：
此位决定了是否可以实际将计数器值捕获到输入捕获/比较寄存器 1 (TIMx_CCR1) 中。
0：禁止捕获。
1：使能捕获。

表 321. 标准 OCx 通道的输出控制位

CCxE 位	OCx 输出状态
0	禁止输出（OCx=“0”，OCx_EN=“0”）
1	OCx=OCxREF + 极性，OCx_EN=“1”

注： *与标准 OCx 通道相连的外部 I/O 引脚的状态取决于通道 OCx 的状态以及 GPIO 寄存器。*

40.4.8 TIM12计数器 (TIMx_CNT)

TIM12 counter

偏移地址：0x24

复位值：0x0000 0000

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

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

位 31 **UIFCPY**：UIF 副本 (UIF Copy)
该位是 TIMx_ISR 寄存器中 UIF 位的只读副本。

位 30:16 保留，必须保持复位值。

位 15:0 **CNT[15:0]**：计数器值 (Counter value)

40.4.9 TIM12 预分频器 (TIMx_PSC)

TIM12 prescaler

偏移地址：0x28

复位值：0x0000

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

位 15:0 **PSC[15:0]**：预分频器值 (Prescaler value)
计数器时钟频率 CK_CNT 等于 f_{CK_PSC} / (PSC[15:0] + 1)。
PSC 包含在每次发生更新事件时要装载到有效预分频器寄存器的值。
（包括当计数器通过 TIMx_EGR 寄存器的 UG 位或在“复位模式”下通过触发控制器清零时）。

40.4.10 TIM12自动重载寄存器 (TIMx_ARR)

TIM12 auto-reload register

偏移地址：0x2C

复位值：0xFFFF

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

位 15:0 **ARR[15:0]**：自动重载值 (Auto-reload value)

ARR 为要装载到有效自动重载寄存器的值。

有关 ARR 更新和行为的更多详细信息，请参见*第 1563 页的第 40.3.1 节：时基单元*。

当自动重载值为空时，计数器不工作。

40.4.11 TIM12 捕获/比较寄存器 1 (TIMx_CCR1)

TIM12 capture/compare register 1

偏移地址：0x34

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR1[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 15:0 **CCR1[15:0]**：捕获/比较 1 值 (Capture/Compare 1 value)

如果通道 CC1 配置为输出：

CCR1 是捕获/比较寄存器 1 的预装载值。

如果没有通过 TIMx_CCMR1 寄存器中的 OC1PE 位来使能预装载功能，则该值立刻生效；否则只在发生更新事件时生效（拷贝有效捕获/比较寄存器 1）。

有效捕获/比较寄存器中包含要与计数器 TIMx_CNT 进行比较并在 OC1 输出上发出信号的值。

如果通道 CC1 配置为输入：

CCR1 为上一个输入捕获 1 事件 (IC1) 发生时的计数器值。

40.4.12 TIM12 捕获/比较寄存器 2 (TIMx_CCR2)

TIM12 capture/compare register 2

偏移地址：0x38

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR2[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 15:0 **CCR2[15:0]**：捕获/比较 2 值 (Capture/Compare 2 value)

如果通道 CC2 配置为输出：

CCR2 是捕获/比较寄存器 2 的预装载值。

如果没有通过 TIMx_CCMR2 寄存器中的 OC2PE 位来使能预装载功能，则该值立刻生效；否则只在发生更新事件时生效（拷贝有效捕获/比较寄存器 2）

有效捕获/比较寄存器中包含要与计数器 TIMx_CNT 进行比较并在 OC2 输出上发出信号的值。

如果通道 CC2 配置为输入：

CCR2 为上一个输入捕获 2 事件 (IC2) 发生时的计数器值。

40.4.13 TIM12 定时器输入选择寄存器 (TIM12_TISEL)

TIM12 timer input selection register

偏移地址：0x68

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	TI2SEL[3:0]				Res.	Res.	Res.	Res.	TI1SEL[3:0]			
				rw								rw			

位 15:12 保留，必须保持复位值。

位 11:8 **TI2SEL[3:0]**：选择 TI2[0] 到 TI2[15] 输入 (selects TI2[0] to TI2[15] input)

0000：TIM12_CH2 输入

其它：保留

位 7:4 保留，必须保持复位值。

位 3:0 **TI1SEL[3:0]**：选择 TI1[0] 到 TI1[15] 输入 (selects TI1[0] to TI1[15] input)

0000：TIM12_CH1 输入

其它：保留

40.4.14 TIM12 寄存器映射

TIM12 寄存器可映射为 16 位可寻址寄存器，如下表所述：

表 322. TIM12 寄存器映射和复位值

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	**TIMx_CR1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	UIFREMAP	Res.	CKD [1:0]		ARPE	Res.	Res.	Res.	OPM	URS	UDIS	CEN
	Reset value																					0		0	0	0				0	0	0	0
0x08	**TIMx_SMCR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TS [4:3] Res.		Res.	Res.	Res.	SMS[3]	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MSM	TS[2:0]			Res.	SMS[2:0]		
	Reset value											0	0				0									0	0	0	0		0	0	0
0x0C	**TIMx_DIER**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TIE	Res.	Res.	Res.	CC2IE	CC1IE	UIE
	Reset value																										0				0	0	0
0x10	**TIMx_SR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CC2OF	CC1OF	Res.	Res.	TIF	Res.	Res.	Res.	CC2IF	CC1IF	UIF
	Reset value																						0	0			0				0	0	0
0x14	**TIMx_EGR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TG	Res.	Res.	Res.	CC2G	CC1G	UG
	Reset value																										0				0	0	0
0x18	**TIMx_CCMR1** Output Compare mode	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC2M[3]	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC1M[3]	Res.	OC2M [2:0]			OC2PE	OC2FE	CC2S [1:0]		Res.	OC1M [2:0]			OC1PE	OC1FE	CC1S [1:0]	
	Reset value								0								0		0	0	0	0	0	0	0		0	0	0	0	0	0	0
	TIMx_CCMR1 Input Capture mode	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	IC2F[3:0]				IC2 PSC [1:0]		CC2S [1:0]		IC1F[3:0]				IC1 PSC [1:0]		CC1S [1:0]	
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x1C	Reserved	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
0x20	**TIMx_CCER**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CC2NP	Res.	CC2P	CC2E	CC1NP	Res.	CC1P	CC1E
	Reset value																									0		0	0	0		0	0
0x24	**TIMx_CNT**	UIFCPY	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CNT[15:0]															
	Reset value	0																0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x28	**TIMx_PSC**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PSC[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x2C	**TIMx_ARR**	Reserved																ARR[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

表 322. TIM12 寄存器映射和复位值（续）

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x30	Reserved	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
0x34	**TIMx_CCR1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CCR1[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x38	**TIMx_CCR2**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CCR2[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x3C to 0x64	Reserved	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
0x68	**TIM12_TISEL**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TI2SEL[3:0]				Res.	Res.	Res.	Res.	TI1SEL[3:0]			
	Reset value																					0	0	0	0					0	0	0	0

有关寄存器边界地址的信息，请参见*第 100 页的第 2.2.2 节*。

40.5 TIM13/TIM14 寄存器

外设寄存器的写访问仅支持半字（16 位）或字（32 位）。而读访问可支持字节（8 位）、半字（16 位）或字（32 位）。

40.5.1 TIM13/TIM14控制寄存器 1 (TIMx_CR1)

TIM13/TIM14 control register 1

偏移地址：0x00

复位值：0x0000

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

位 15:12 保留，必须保持复位值。

位 11 **UIFREMAP**：UIF 状态位重映射 (UIF status bit remapping)
0：无重映射。UIF 状态位不复制到 TIMx_CNT 寄存器的位 31。
1：使能重映射。UIF 状态位复制到 TIMx_CNT 寄存器的位 31。

位 10 保留，必须保持复位值。

位 9:8 **CKD**：时钟分频 (Clock division)
此位域指示定时器时钟 (CK_INT) 频率与数字滤波器所使用的采样时钟 (TIx) 之间的分频比
00：$t_{DTS} = t_{CK_INT}$
01：$t_{DTS} = 2 \times t_{CK_INT}$
10：$t_{DTS} = 4 \times t_{CK_INT}$
11：保留

位 7 **ARPE**：自动重载预装载使能 (Auto-reload preload enable)
0：TIMx_ARR 寄存器不进行缓冲
1：TIMx_ARR 寄存器进行缓冲

位 6:4 保留，必须保持复位值。

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

位 2 **URS**：更新请求源 (Update request source)

此位由软件置 1 和清零，用以选择更新中断 (UEV) 源。

0：使能后，所有以下事件都会生成 UEV：

– 计数器上溢

– 将 UG 位置 1

1：使能后，只有计数器上溢会生成 UEV。

位 1 **UDIS**：更新禁止 (Update disable)

此位由软件置 1 和清零，用以使能/禁止更新中断 (UEV) 事件生成。

0：使能 UEV。UEV 可通过以下事件之一生成：

– 计数器上溢

– 将 UG 位置 1

带缓冲的寄存器被加载为预加载数值。

1：禁止 UEV。不会生成 UEV，各影子寄存器的值（ARR、PSC 和 CCRx）保持不变。如果 UG 位置 1，则会重新初始化计数器和预分频器。

位 0 **CEN**：计数器使能 (Counter enable)

0：禁止计数器

1：使能计数器

注：　只有事先通过软件将 CEN 位置 1，才可以使用外部时钟和门控模式。而触发模式可通过硬件自动将 CEN 位置 1。

40.5.2 TIM13/TIM14 中断使能寄存器 (TIMx_DIER)

TIM13/TIM14 Interrupt enable register

偏移地址：0x0C

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CC1IE	UIE
														rw	rw

位 15:2 保留，必须保持复位值。

位 1 **CC1IE**：捕获/比较 1 中断使能 (Capture/Compare 1 interrupt enable)
0：禁止 CC1 中断
1：使能 CC1 中断

位 0 **UIE**：更新中断使能 (Update interrupt enable)
0：禁止更新中断
1：使能更新中断

40.5.3 TIM13/TIM14 状态寄存器 (TIMx_SR)

TIM13/TIM14 status register

偏移地址：0x10

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	CC1OF	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CC1IF	UIF
						rc_w0								rc_w0	rc_w0

位 15:10 保留，必须保持复位值。

位 9 **CC1OF**：捕获/比较 1 重复捕获标志 (Capture/Compare 1 overcapture flag)
仅当将相应通道配置为输入捕获模式时，此标志位才会由硬件置 1。通过软件写入“0”可将该位清零。
0：未检测到重复捕获。
1：TIMx_CCR1 寄存器中已捕获到计数器值且 CC1IF 标志已置 1。

位 8:2 保留，必须保持复位值。

位 1 **CC1IF**：捕获/比较 1 中断标志 (Capture/compare 1 interrupt flag)
如果通道 CC1 配置为输出：
当计数器与比较值匹配时，此标志由硬件置 1。但需要通过软件清零。
0：不匹配。
1：TIMx_CNT 计数器的值与 TIMx_CCR1 寄存器的值匹配。当 TIMx_CCR1 的值大于 TIMx_ARR 的值时， CC1IF 位将在计数器发生上溢时变为 1。
如果通道 CC1 配置为输入：
此位将在发生捕获事件时由硬件置 1。通过软件或读取 TIMx_CCR1 寄存器将该位清零。
0：未发生输入捕获事件。
1：TIMx_CCR1 寄存器中已捕获到计数器值（IC1 上已检测到与所选极性匹配的边沿）。

位 0 **UIF**：更新中断标志 (Update interrupt flag)

该位在发生更新事件时通过硬件置 1。但需要通过软件清零。

0：未发生更新。

1：更新中断挂起。该位在以下情况下更新寄存器时由硬件置 1：

– 上溢且当 TIMx_CR1 寄存器中 UDIS =“0”时。

– 当由于 TIMx_CR1 寄存器中 URS =“0”且 UDIS =“0”而通过软件使用 TIMx_EGR 寄存器中的 UG 位重新初始化 CNT 时。

40.5.4 TIM13/TIM14 事件生成寄存器 (TIMx_EGR)

TIM13/TIM14 event generation register

偏移地址：0x14

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CC1G	UG
														w	w

位 15:2 保留，必须保持复位值。

位 1 **CC1G**：捕获/比较 1 生成 (Capture/Compare 1 generation)

此位由软件置 1 以生成事件，并由硬件自动清零。

0：不执行任何操作

1：通道 1 上生成捕获/比较事件：

如果通道 CC1 配置为输出：

使能后，CC1IF 标志置 1 并发送相应的中断。

如果通道 CC1 配置为输入：

TIMx_CCR1 寄存器中将捕获到计数器当前值。使能后，CC1IF 标志置 1 并发送相应中断。如果 CC1IF 标志已为高电平，CC1OF 标志将置 1。

位 0 **UG**：更新生成 (Update generation)

该位可通过软件置 1，并由硬件自动清零。

0：不执行任何操作

1：重新初始化计数器并生成寄存器更新事件。请注意，预分频器计数器也将清零（但预分频比不受影响）。计数器清零。

40.5.5 TIM13/TIM14 捕获/比较模式寄存器 1 (TIMx_CCMR1)

TIM13/TIM14 capture/compare mode register 1

偏移地址：0x18

复位值：0x0000

这些通道可用于输入（捕获模式）或输出（比较模式）模式。通道方向通过配置相应的 CCxS 位进行定义。此寄存器的所有其它位在输入模式和输出模式下的功能均不同。对于任一给定位，OCxx 用于说明通道配置为输出时该位对应的功能，ICxx 则用于说明通道配置为输入时该位对应的功能。因此，必须注意同一个位在输入阶段和输出阶段具有不同的含义。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC1M[3]
															Res.
															rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC1M[2:0]			OC1PE	OC1FE	CC1S[1:0]	
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	IC1F[3:0]				IC1PSC[1:0]			
								rw	rw	rw	rw	rw	rw	rw	rw

输出比较模式

位 31:17 保留，始终读为 0

位 16 **OC1M[3]**：输出比较 1 模式——位 3 (Output Compare 1 mode - bit 3)
请参见 OC1M 说明——位 6:4

位 15:7 保留，必须保持复位值。

位 6:4 **OC1M[3:0]**：输出比较 1 模式 (Output Compare 1 mode)（请参见 OC1M[3] 的位 16）
这些位定义提供 OC1 的输出参考信号 OC1REF 的行为。OC1REF 为高电平有效，而 OC1 的有效电平则取决于 CC1P 位。
0000：冻结。输出比较寄存器 TIMx_CCR1 与计数器 TIMx_CNT 进行比较不会对输出造成任何影响。
0001：将通道 1 设置为匹配时输出有效电平。当计数器 TIMx_CNT 与捕获/比较寄存器 1 (TIMx_CCR1) 匹配时，OC1REF 信号强制变为高电平。
0010：将通道 1 设置为匹配时输出无效电平。当计数器 TIMx_CNT 与捕获/比较寄存器 1 (TIMx_CCR1) 匹配时，OC1REF 信号强制变为低电平。
0011：翻转——TIMx_CNT = TIMx_CCR1 时，OC1REF 发生翻转。
0100：强制变为无效电平——OC1REF 强制变为低电平。
0101：强制变为有效电平——OC1REF 强制变为高电平。
0110：PWM 模式 1——只要 TIMx_CNT < TIMx_CCR1，通道 1 便为有效状态，否则为无效状态。
0111：PWM 模式 2——只要 TIMx_CNT < TIMx_CCR1，通道 1 便为无效状态，否则为有效状态。
其它值：保留

注： *在 PWM 模式 1 或 PWM 模式 2 下，仅当比较结果发生改变或输出比较模式由“冻结”模式切换到“PWM”模式时，OCREF 电平才会发生更改。*

位 3 **OC1PE**：输出比较 1 预装载使能 (Output Compare 1 preload enable)

0：禁止与 TIMx_CCR1 相关的预装载寄存器。可随时向 TIMx_CCR1 写入数据，写入后将立即使用新值。

1：使能与 TIMx_CCR1 相关的预装载寄存器。可读/写访问预装载寄存器。TIMx_CCR1 预装载值在每次生成更新事件时都会装载到有效寄存器中。

注： 只有单脉冲模式下才可在未验证预装载寄存器的情况下使用 PWM 模式（TIMx_CR1 寄存器中的 OPM 位置 1）。其它情况下则无法保证该行为。

位 2 **OC1FE**：输出比较 1 快速使能 (Output Compare 1 fast enable)

此位用于加快触发输入事件对 CC 输出的影响。

0：即使触发开启，CC1 也将根据计数器和 CCR1 值正常工作。触发输入出现边沿时，激活 CC1 输出的最短延迟时间为 5 个时钟周期。

1：触发输入上出现有效边沿相当于 CC1 输出上的比较匹配。随后，无论比较结果如何，OC 都设置为比较电平。采样触发输入和激活 CC1 输出的延迟时间缩短为 3 个时钟周期。仅当通道配置为 PWM1 或 PWM2 模式时，OC1FE 才会起作用。

位 1:0 **CC1S[1:0]**：捕获/比较 1 选择 (Capture/Compare 1 Selection)

此位域定义通道方向（输入/输出）以及所使用的输入。

00：CC1 通道配置为输出。

01：CC1 通道配置为输入，IC1 映射到 TI1 上。

10：CC1 通道配置为输入，IC1 映射到 TI2 上

11：CC1 通道配置为输入，IC1 映射到 TRC 上。此模式仅在通过 TS 位（TIMx_SMCR 寄存器）选择内部触发输入时有效

注： 仅当通道关闭时（TIMx_CCER 中的 CC1E = 0），才可向 CC1S 位写入数据。

输入捕获模式

位 15:8 保留，必须保持复位值。

位 7:4 **IC1F**：输入捕获 1 滤波器 (Input capture 1 filter)

此位域可定义 TI1 输入的采样频率和适用于 TI1 的数字滤波器带宽。数字滤波器由事件计数器组成，每 N 个连续事件才视为一个有效输出边沿：

0000：无滤波器，按 f_{DTS} 频率进行采样 1000：$f_{SAMPLING}=f_{DTS}/8$，N=6

0001：$f_{SAMPLING}=f_{CK_INT}$，N=2 1001：$f_{SAMPLING}=f_{DTS}/8$，N=8

0010：$f_{SAMPLING}=f_{CK_INT}$，N=4 1010：$f_{SAMPLING}=f_{DTS}/16$，N=5

0011：$f_{SAMPLING}=f_{CK_INT}$，N=8 1011：$f_{SAMPLING}=f_{DTS}/16$，N=6

0100：$f_{SAMPLING}=f_{DTS}/2$，N=6 1100：$f_{SAMPLING}=f_{DTS}/16$，N=8

0101：$f_{SAMPLING}=f_{DTS}/2$，N=8 1101：$f_{SAMPLING}=f_{DTS}/32$，N=5

0110：$f_{SAMPLING}=f_{DTS}/4$，N=6 1110：$f_{SAMPLING}=f_{DTS}/32$，N=6

0111：$f_{SAMPLING}=f_{DTS}/4$，N=8 1111：$f_{SAMPLING}=f_{DTS}/32$，N=8

位 3:2 **IC1PSC**：输入捕获 1 预分频器 (Input capture 1 prescaler)

此位域定义 CC1 输入 (IC1) 的预分频比。

只要 CC1E=“0”（TIMx_CCER 寄存器），预分频器便立即复位。

00：无预分频器，捕获输入上每检测到一个边沿便执行捕获

01：每发生 2 个事件便执行一次捕获

10：每发生 4 个事件便执行一次捕获

11：每发生 8 个事件便执行一次捕获

位 1:0 **CC1S[1:0]**：捕获/比较 1 选择 (Capture/Compare 1 Selection)

此位域定义通道方向（输入/输出）以及所使用的输入。

00：CC1 通道配置为输出

01：CC1 通道配置为输入，IC1 映射到 TI1 上

10：保留

11：保留

注： 仅当通道关闭时（TIMx_CCER 中的 CC1E = 0），才可向 CC1S 位写入数据。

40.5.6 TIM13/TIM14 捕获/比较使能寄存器 (TIMx_CCER)

TIM13/TIM14 capture/compare enable register

偏移地址：0x20

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CC1NP	Res.	CC1P	CC1E
												rw		rw	rw

位 15:4 保留，必须保持复位值。

位 3 **CC1NP**： 捕获/比较 1 互补输出极性 (Capture/Compare 1 complementary output Polarity)。
CC1 通道配置为输出：CC1NP 必须保持清零。
CC1 通道配置为输入：CC1NP 与 CC1P 配合使用可定义 TI1FP1 的极性（请参见 CC1P 说明）。

位 2 保留，必须保持复位值。

位 1 **CC1P**：捕获/比较 1 输出极性 (Capture/Compare 1 output Polarity)。
CC1 通道配置为输出：
0：OC1 高电平有效
1：OC1 低电平有效
CC1 通道配置为输入：
CC1P 和 CC1NP 位针对捕获操作选择 TI1FP1 极性。
00：未反相/上升沿触发
电路对 TI1FP1 上升沿敏感（捕获模式），TI1FP1 未反相。
01：反相/下降沿触发
电路对 TI1FP1 下降沿敏感（捕获模式），TI1FP1 反相。
10：保留，不使用此配置。
11：未反相/上升沿和下降沿均触发
电路对 TI1FP1 上升沿和下降沿均敏感（捕获模式），TI1FP1 未反相。

位 0 **CC1E**：捕获/比较 1 输出使能 (Capture/Compare 1 output enable)。
CC1 通道配置为输出：
0：关闭——OC1 无效
1：开启——在相应输出引脚上输出 OC1 信号
CC1 通道配置为输入：
此位决定了是否可以实际将计数器值捕获到输入捕获/比较寄存器 1 (TIMx_CCR1) 中。
0：禁止捕获
1：使能捕获

表 323. 标准 OCx 通道的输出控制位

CCxE 位	OCx 输出状态
0	禁止输出（OCx=“0”，OCx_EN=“0”）
1	OCx=OCxREF + 极性，OCx_EN=“1”

注： *与标准 OCx 通道相连的外部 I/O 引脚的状态取决于通道 OCx 的状态以及 GPIO 寄存器。*

40.5.7 TIM13/TIM14计数器 (TIMx_CNT)

TIM13/TIM14 counter

偏移地址：0x24

复位值：0x0000 0000

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

位 31 **UIFCPY**：UIF 副本 (UIF Copy)
该位是 TIMx_ISR 寄存器中 UIF 位的只读副本。

位 30:16 保留，必须保持复位值。

位 15:0 **CNT[15:0]**：计数器值 (Counter value)

40.5.8 TIM13/TIM14 预分频器 (TIMx_PSC)

TIM13/TIM14 prescaler

偏移地址：0x28

复位值：0x0000

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

位 15:0 **PSC[15:0]**：预分频器值 (Prescaler value)
计数器时钟频率 CK_CNT 等于 f_{CK_PSC} / (PSC[15:0] + 1)。
PSC 包含在每次发生更新事件时要装载到有效预分频器寄存器的值。
（包括当计数器通过 TIMx_EGR 寄存器的 UG 位或在“复位模式”下通过触发控制器清零时）。

40.5.9 TIM13/TIM14自动重载寄存器 (TIMx_ARR)

TIM13/TIM14 auto-reload register

偏移地址：0x2C

复位值：0xFFFF

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

位 15:0 **ARR[15:0]**：自动重载值 (Auto-reload value)
ARR 为要装载到有效自动重载寄存器的值。
有关 ARR 更新和行为的详细信息，请参见*第 1563 页的第 40.3.1 节：时基单元*。
当自动重载值为空时，计数器不工作。

40.5.10 TIM13/TIM14 捕获/比较寄存器 1 (TIMx_CCR1)

TIM13/TIM14 capture/compare register 1

偏移地址：0x34

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR1[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 15:0 **CCR1[15:0]**：捕获/比较 1 值 (Capture/Compare 1 value)

如果通道 CC1 配置为输出：

CCR1 是捕获/比较寄存器 1 的预装载值。

如果没有通过 TIMx_CCMR1 寄存器中的 OC1PE 位来使能预装载功能，则该值立刻生效；否则只在发生更新事件时生效（拷贝有效捕获/比较寄存器 1）。

有效捕获/比较寄存器中包含要与计数器 TIMx_CNT 进行比较并在 OC1 输出上发出信号的值。

如果通道 CC1 配置为输入：

CCR1 为上一个输入捕获 1 事件 (IC1) 发生时的计数器值。

40.5.11 TIM13 定时器输入选择寄存器 (TIM13_TISEL)

TIM13 timer input selection register

偏移地址：0x68

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TI1SEL[3:0]			
												rw			

位 15:4 保留，必须保持复位值。

位 3:0 **TI1SEL[3:0]**：选择 TI1[0] 到 TI1[15] 输入 (selects TI1[0] to TI1[15] input)

0000：TIM13_CH1 输入

其它：保留

40.5.12 TIM14 定时器输入选择寄存器 (TIM14_TISEL)

TIM14 timer input selection register

偏移地址：0x68

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TI1SEL[3:0]			
												rw			

位 15:4 保留，必须保持复位值。

位 3:0 **TI1SEL[3:0]**：选择 TI1[0] 到 TI1[15] 输入 (selects TI1[0] to TI1[15] input)
0000：TIM14_CH1 输入
其它：保留

40.5.13 TIM13/TIM14 寄存器映射

TIMx 寄存器可映射为 16 位可寻址寄存器，如下表所述：

表 324. TIM13/TIM14 寄存器映射和复位值

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	**TIMx_CR1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	UIFREMAP	Res.	CKD [1:0]		ARPE	Res.	Res.	Res.	OPM	URS	UDIS	CEN
	Reset value																					0		0	0	0				0	0	0	0
0x04	Reserved	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
0x0C	**TIMx_DIER**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CC1IE	UIE
	Reset value																															0	0
0x10	**TIMx_SR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CC1OF	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CC1IF	UIF
	Reset value																							0								0	0
0x14	**TIMx_EGR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CC1G	UG
	Reset value																															0	0
0x18	**TIMx_CCMR1** Output compare mode	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC1M[3]	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC1M [2:0]			OC1PE	OC1FE	CC1S [1:0]	
	Reset value																0										0	0	0	0	0	0	0
	TIMx_CCMR1 Input capture mode	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	IC1F[3:0]				IC1 PSC [1:0]		CC1S [1:0]	
	Reset value																									0	0	0	0	0	0	0	0
0x1C	Reserved	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
0x20	**TIMx_CCER**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CC1NP	Res.	CC1P	CC1E
	Reset value																													0		0	0
0x24	**TIMx_CNT**	UIFCPY	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CNT[15:0]															
	Reset value	0																0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

表 324. TIM13/TIM14 寄存器映射和复位值（续）

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x28	**TIMx_PSC**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PSC[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x2C	**TIMx_ARR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ARR[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x30	Reserved	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
0x34	**TIMx_CCR1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CCR1[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x38 to 0x64	Reserved	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
0x68	**TIM13_TISEL**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TI1SEL[3:0]			
	Reset value																													0	0	0	0
0x68	**TIM14_TISEL**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TI1SEL[3:0]			
	Reset value																													0	0	0	0

有关寄存器边界地址的信息，请参见*第 100 页的第 2.2.2 节*。

41 通用定时器 (TIM15/TIM16/TIM17)

41.1 TIM15/TIM16/TIM17 前言

TIM15/TIM16/TIM17 定时器包含一个 16 位自动重载计数器，该计数器由可编程预分频器驱动。

此类定时器可用于各种用途，包括测量输入信号的脉冲宽度（输入捕获），或者生成输出波形（输出比较、PWM 和带死区插入的互补 PWM）。

使用定时器预分频器和 RCC 时钟控制器预分频器，可将脉冲宽度和波形周期从几微秒调制到几毫秒。

这些定时器彼此完全独立，不共享任何资源。如*第 41.4.21 节：定时器同步 (TIM15)* 中所述，它们可以一起同步操作。

41.2 TIM15 主要特性

TIM15 包括下列特性：

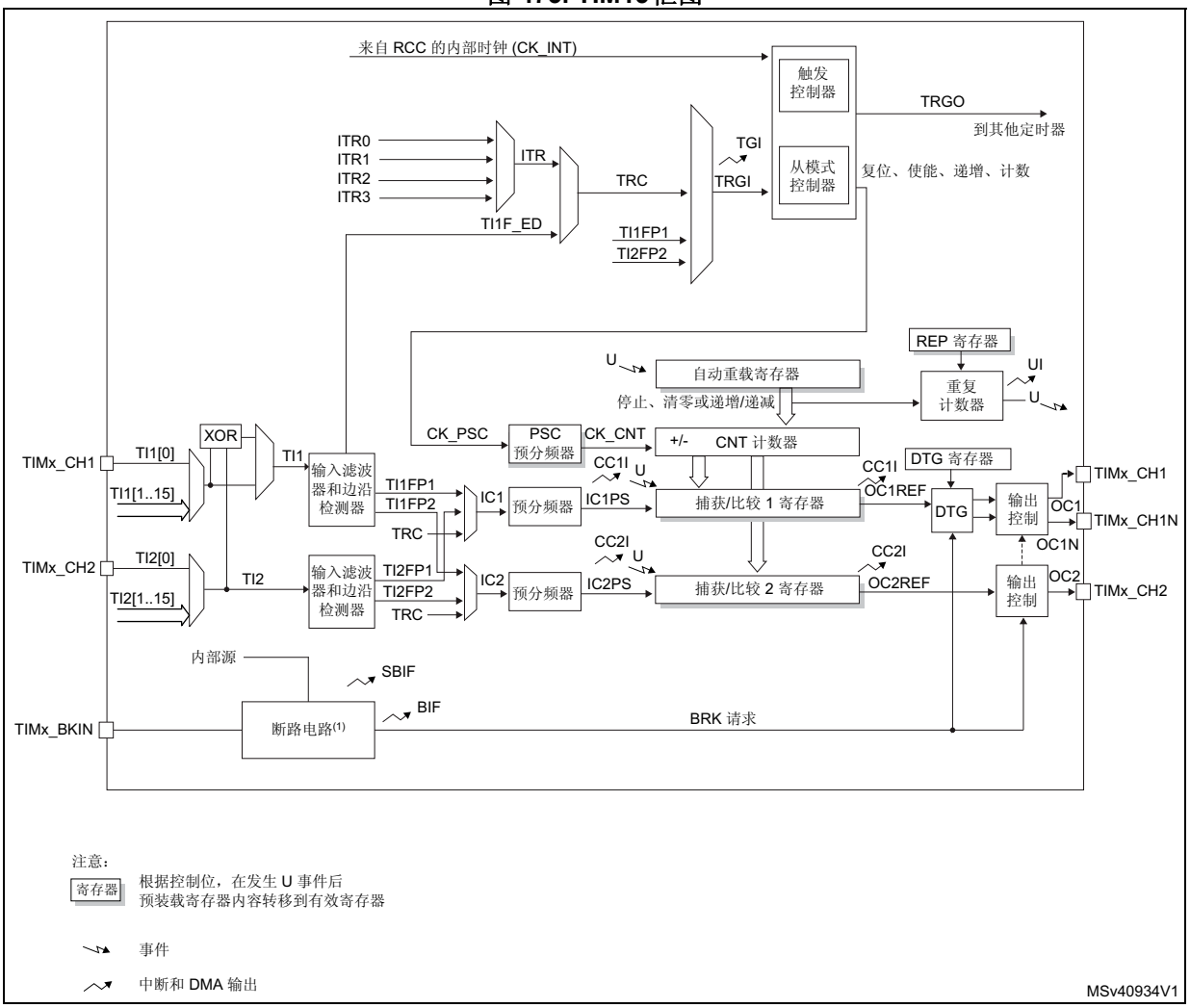
- 16 位自动重载递增计数器。
- 16 位可编程预分频器，用于对计数器时钟频率进行分频（可在运行时修改），分频系数介于 1 和 65535 之间。
- 多达 2 个独立通道，可用于：
  - 输入捕获
  - 输出比较
  - PWM 生成（边沿模式）
  - 单脉冲模式输出
- 带可编程死区时间的互补输出（仅适用于通道 1）。
- 使用外部信号控制定时器且可实现多个定时器互连的同步电路。
- 用于仅在给定计数器周期值后，更新定时器的寄存器的值。
- 可以用来断开输入，将定时器的输出信号，置于复位或已知状态。
- 发生如下事件时生成中断/DMA 请求：
  - 更新：计数器上溢、计数器初始化（通过软件或内部/外部触发）
  - 触发事件（计数器启动、停止、初始化或通过内部/外部触发计数）
  - 输入捕获
  - 输出比较
  - 断路输入（中断请求）

41.3 TIM16/TIM17主要特性

TIM16/TIM17 定时器包括下列特性：

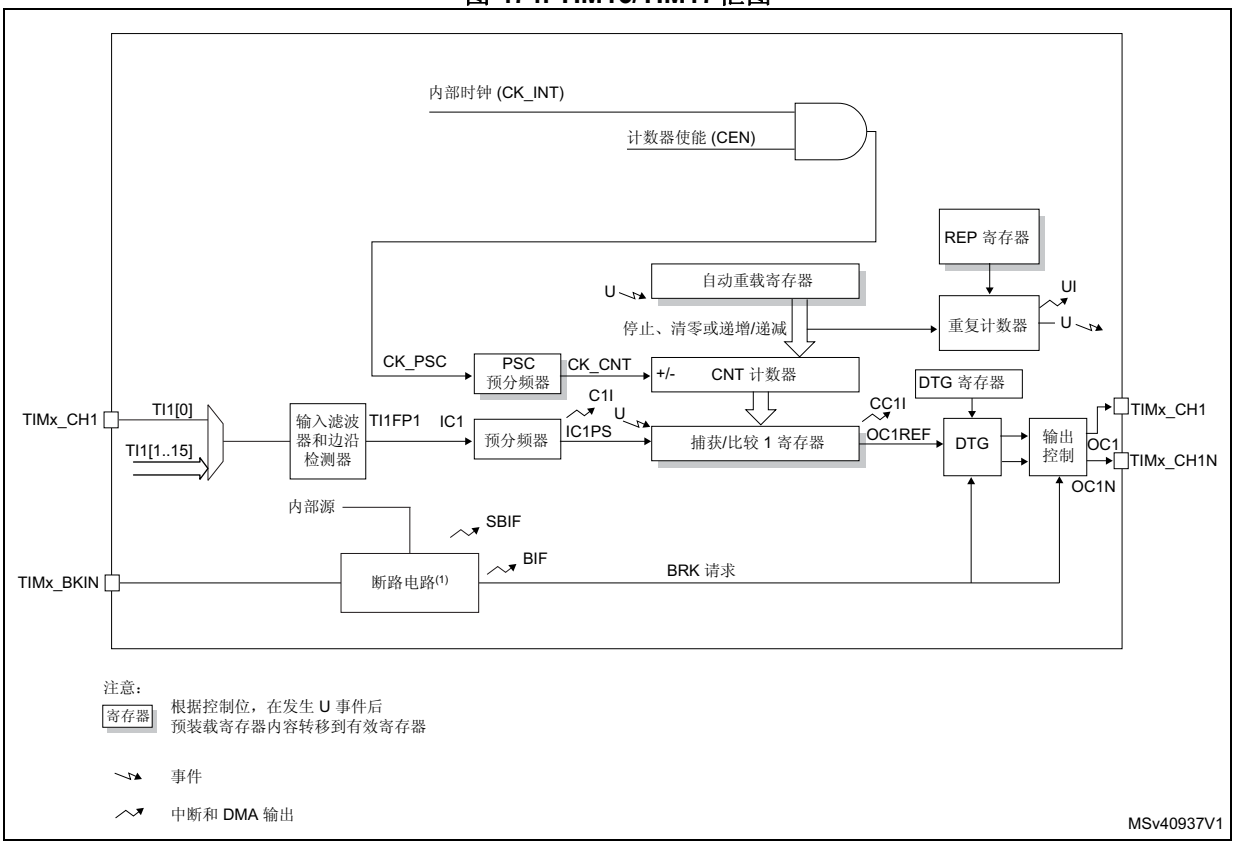
- 16 位自动重载递增计数器。
- 16 位可编程预分频器，用于对计数器时钟频率进行分频（可在运行时修改），分频系数介于 1 和 65535 之间。
- 一个具有以下用途的通道：
  - 输入捕获
  - 输出比较
  - PWM 生成（边沿对齐模式）
  - 单脉冲模式输出
- 带可编程死区的互补输出。
- 重复计数器，用于仅在给定数目的计数器周期后更新定时器寄存器。
- 用于将定时器的输出信号置于复位状态或已知状态的断路输入。
- 发生如下事件时生成中断/DMA 请求：
  - 更新：计数器上溢
  - 输入捕获
  - 输出比较
  - 断路输入

图 473. TIM15框图



1. 内部断路事件源可以是：
   - 由 CSS 生成的时钟故障事件。有关 CSS 的更多信息，请参见*第 8.5.3 节：时钟安全系统 (CSS)*
   - PVD 输出
   - SRAM 奇偶校验错误信号
   - Cortex®-M7 with FPULOCKUP (Hardfault) 输出
   - COMP 输出

图 474. TIM16/TIM17框图



1. 内部断路事件源可以是：
   - 由 CSS 生成的时钟故障事件。有关 CSS 的更多信息，请参见*第 8.5.3 节：时钟安全系统 (CSS)*
   - PVD 输出
   - SRAM 奇偶校验错误信号
   - Cortex®-M7 with FPULOCKUP (Hardfault) 输出
   - COMP 输出

41.4 TIM15/TIM16/TIM17 功能描述

41.4.1 时基单元

可编程高级控制定时器的主要模块是一个 16 位递增计数器及其相关的自动重载寄存器。计数器的时钟可通过预分频器进行分频。

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

时基单元包括：

- 计数器寄存器 (TIMx_CNT)
- 预分频器寄存器 (TIMx_PSC)
- 自动重载寄存器 (TIMx_ARR)
- 重复计数器寄存器 (TIMx_RCR)

自动重载寄存器是预装载的。对自动重载寄存器执行写入或读取操作时会访问预装载寄存器。预装载寄存器的内容既可以直接传送到影子寄存器，也可以在每次发生更新事件 (UEV) 时传送到影子寄存器，这取决于 TIMx_CR1 寄存器中的自动重载预装载使能位 (ARPE)。当计数器达到上溢值并且 TIMx_CR1 寄存器中的 UDIS 位为 0 时，将发送更新事件。该更新事件也可由软件产生。下文将针对各配置的更新事件的产生进行详细介绍。

计数器由预分频器输出 CK_CNT 提供时钟，仅当 TIMx_CR1 寄存器中的计数器启动位 (CEN) 置 1 时，才会启动计数器（有关计数器使能的更多详细信息，另请参见从模式控制器的相关说明）。

注意，计数器将在 TIMx_CR1 寄存器的 CEN 位置 1 时刻的一个时钟周期后开始计数。

预分频器说明

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

图 475 和*图 476* 以一些示例说明在预分频比实时变化时计数器的行为：

图 475. 预分频器分频由 1 变为 2 时的计数器时序图

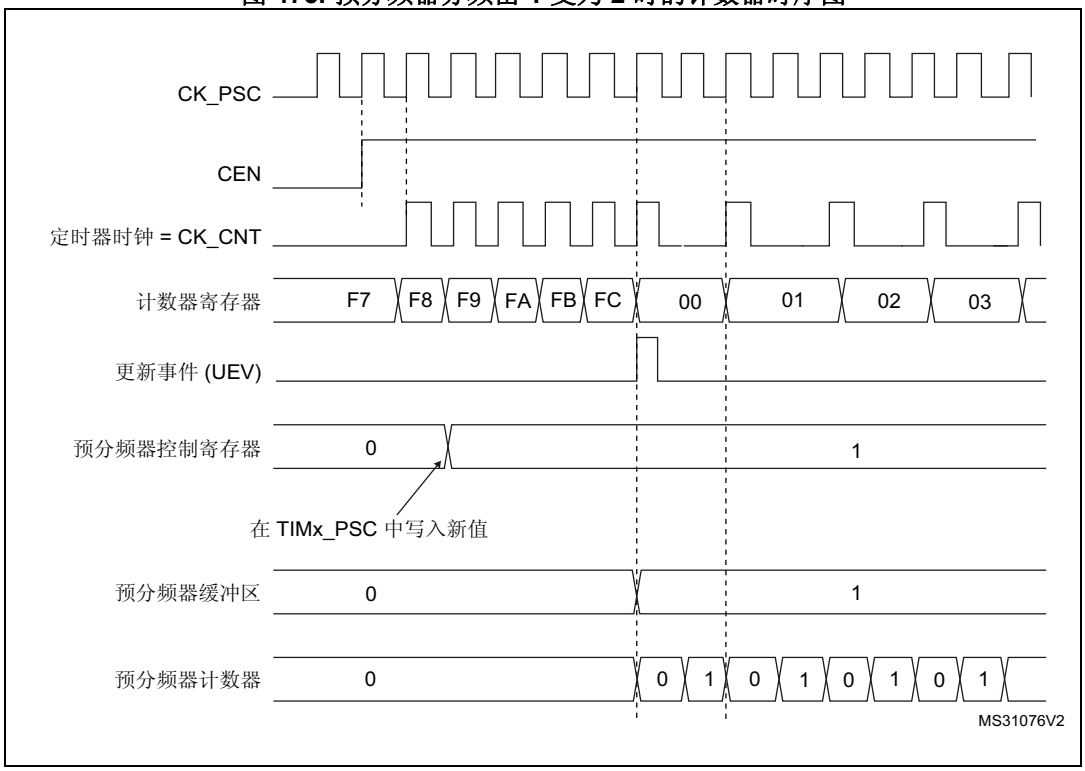
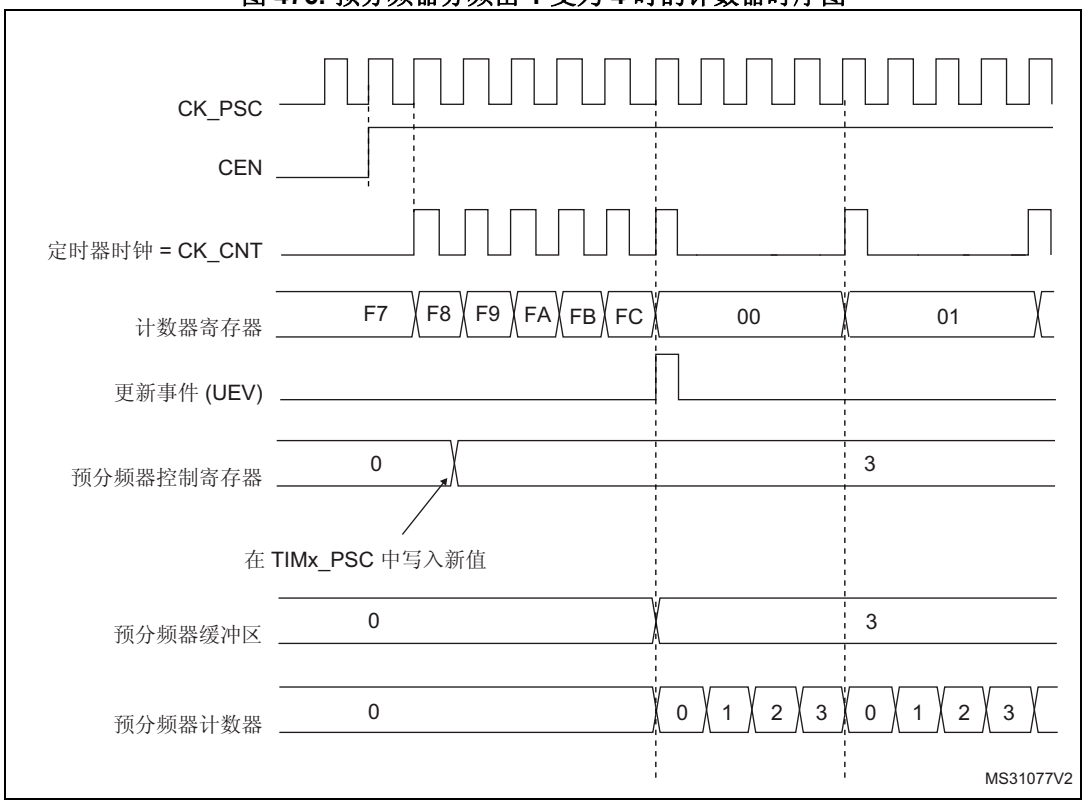


图 476. 预分频器分频由 1 变为 4 时的计数器时序图



41.4.2 计数器模式

递增计数模式

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

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

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

通过软件将 TIMx_CR1 寄存器中的 UDIS 位置 1 可禁止 UEV 事件。这可避免向预装载寄存器写入新值时更新影子寄存器。在 UDIS 位写入 0 之前不会产生任何更新事件。不过，计数器和预分频器计数器都会重新从 0 开始计数（而预分频比保持不变）。此外，如果 TIMx_CR1 寄存器中的 URS 位（更新请求选择）已置 1，则将 UG 位置 1 会生成更新事件 UEV，但不会将 UIF 标志置 1（因此，不会发送任何中断或 DMA 请求）。这样一来，如果在发生捕获事件时将计数器清零，将不会同时产生更新中断和捕获中断。

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

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

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

图 477. 计数器时序图，1 分频内部时钟

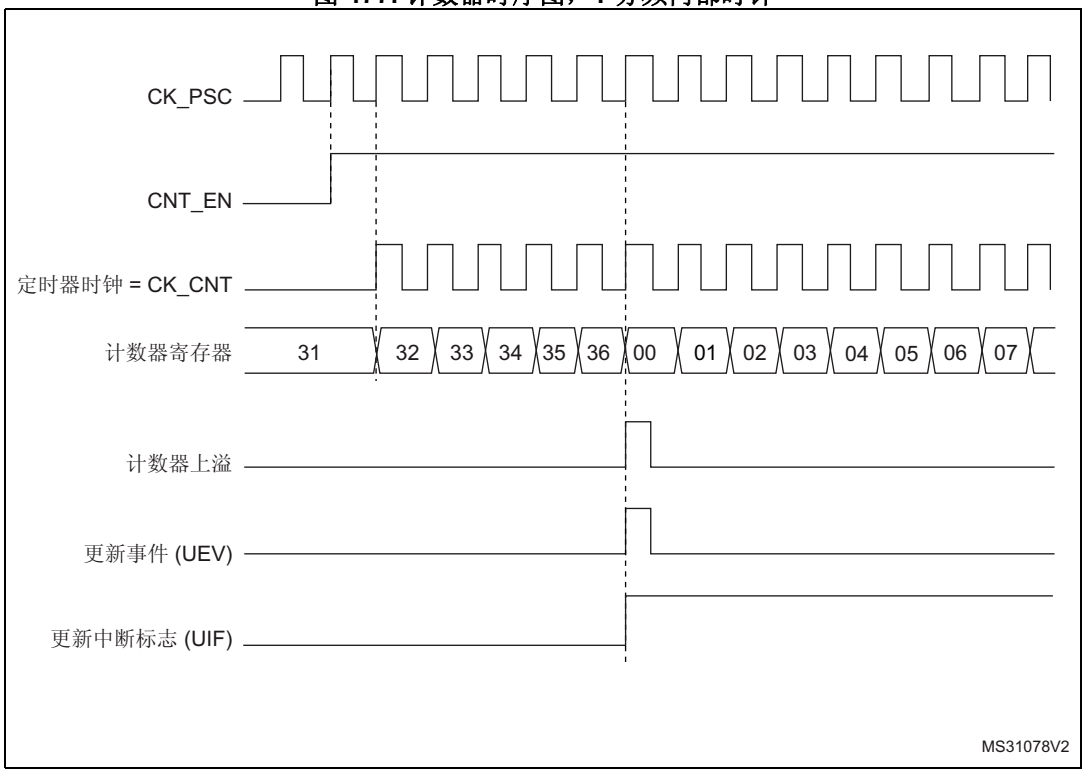


图 478. 计数器时序图，2 分频内部时钟

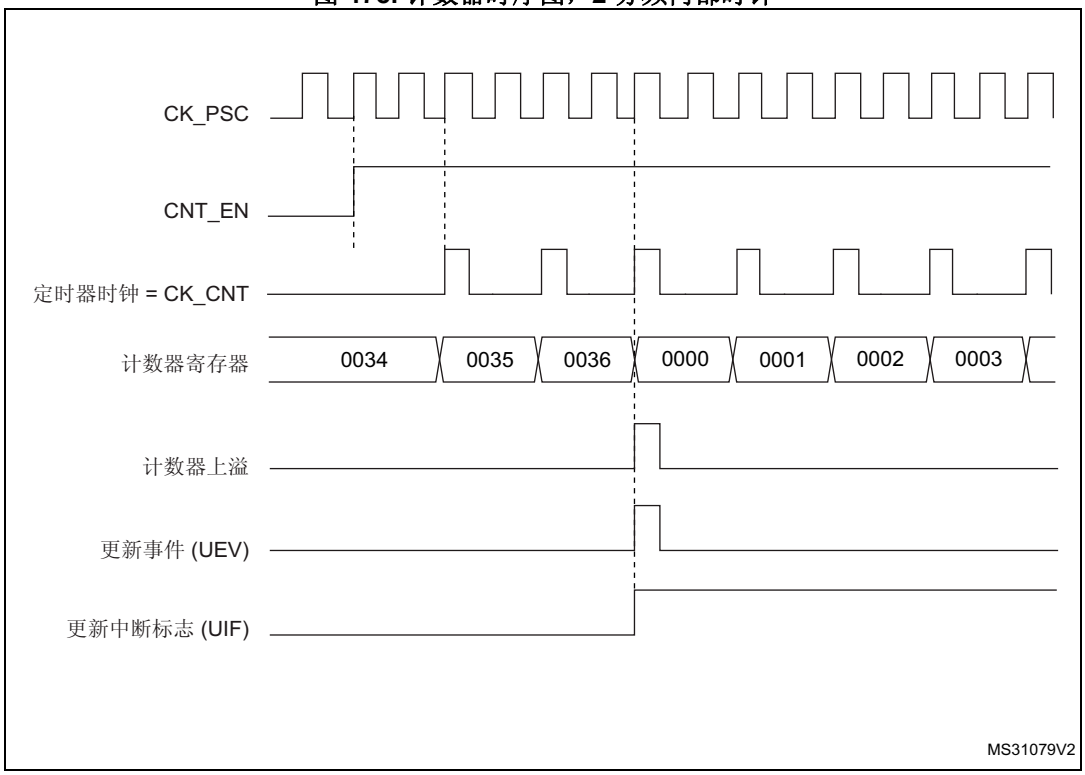


图 479. 计数器时序图，4 分频内部时钟

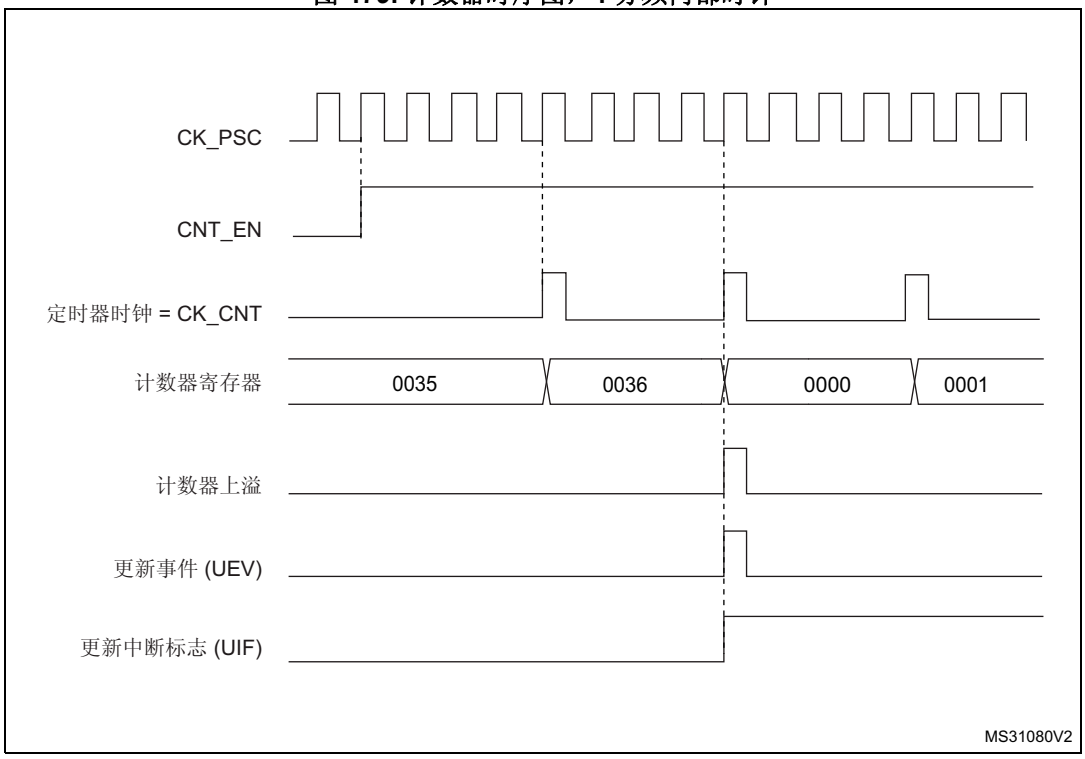


图 480. 计数器时序图，N 分频内部时钟

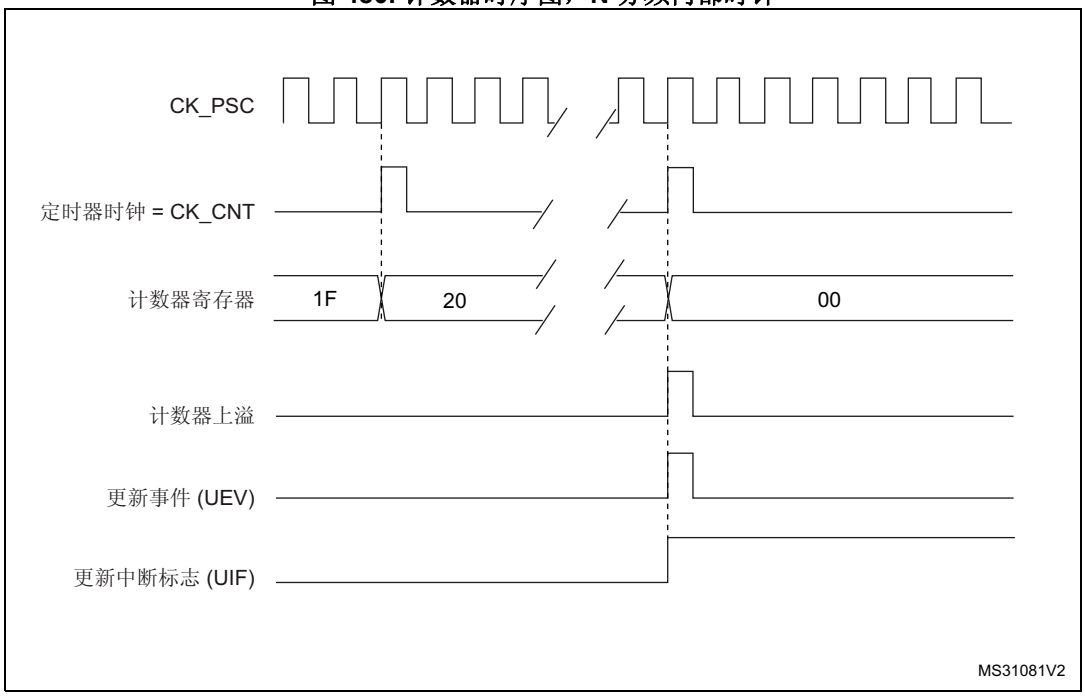


图 481. 计数器时序图，ARPE=0 时更新事件（TIMx_ARR 未预装载）

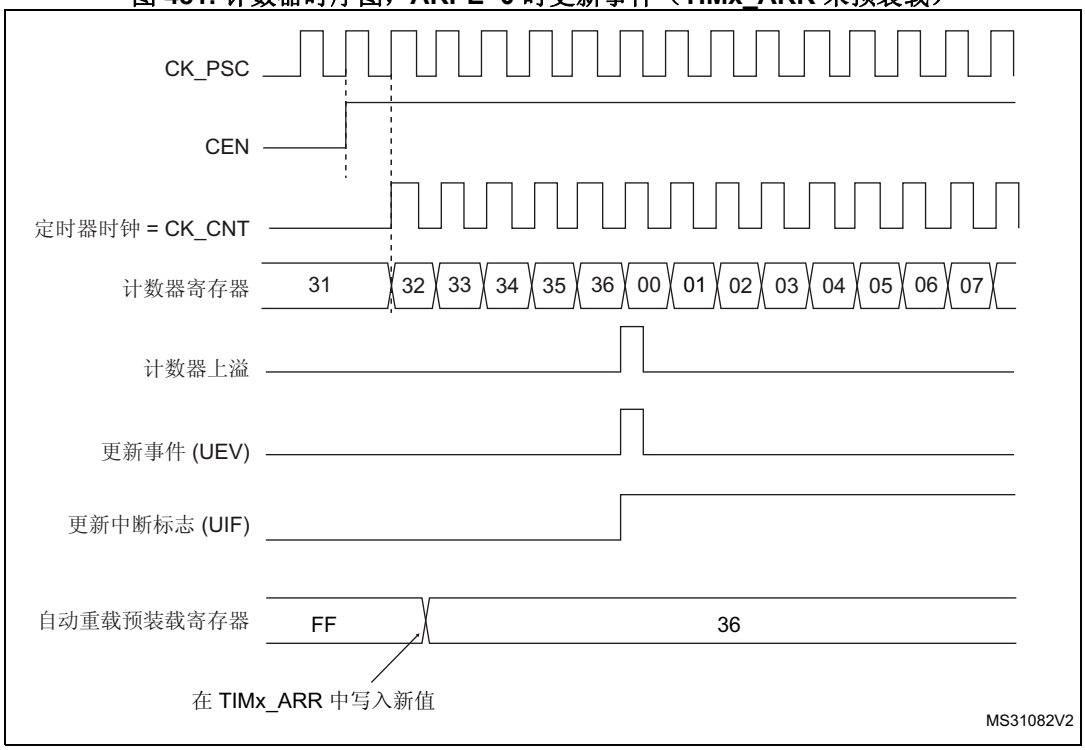
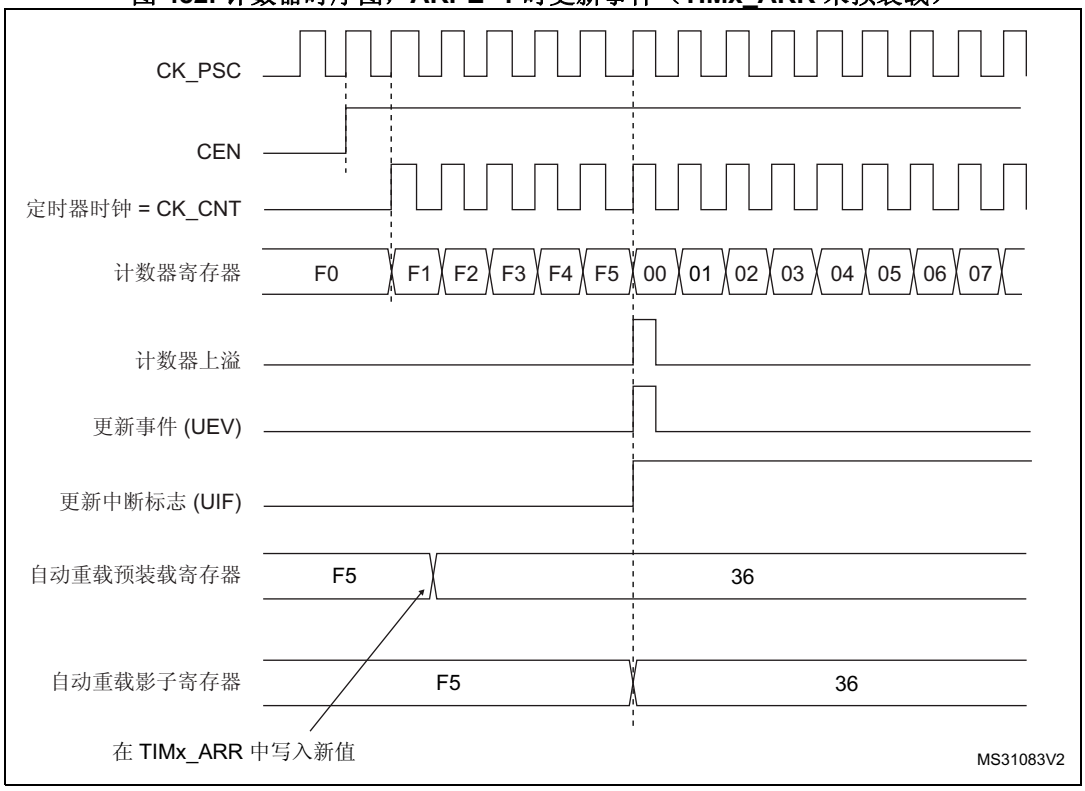


图 482. 计数器时序图，ARPE=1 时更新事件（TIMx_ARR 未预装载）



41.4.3 重复计数器

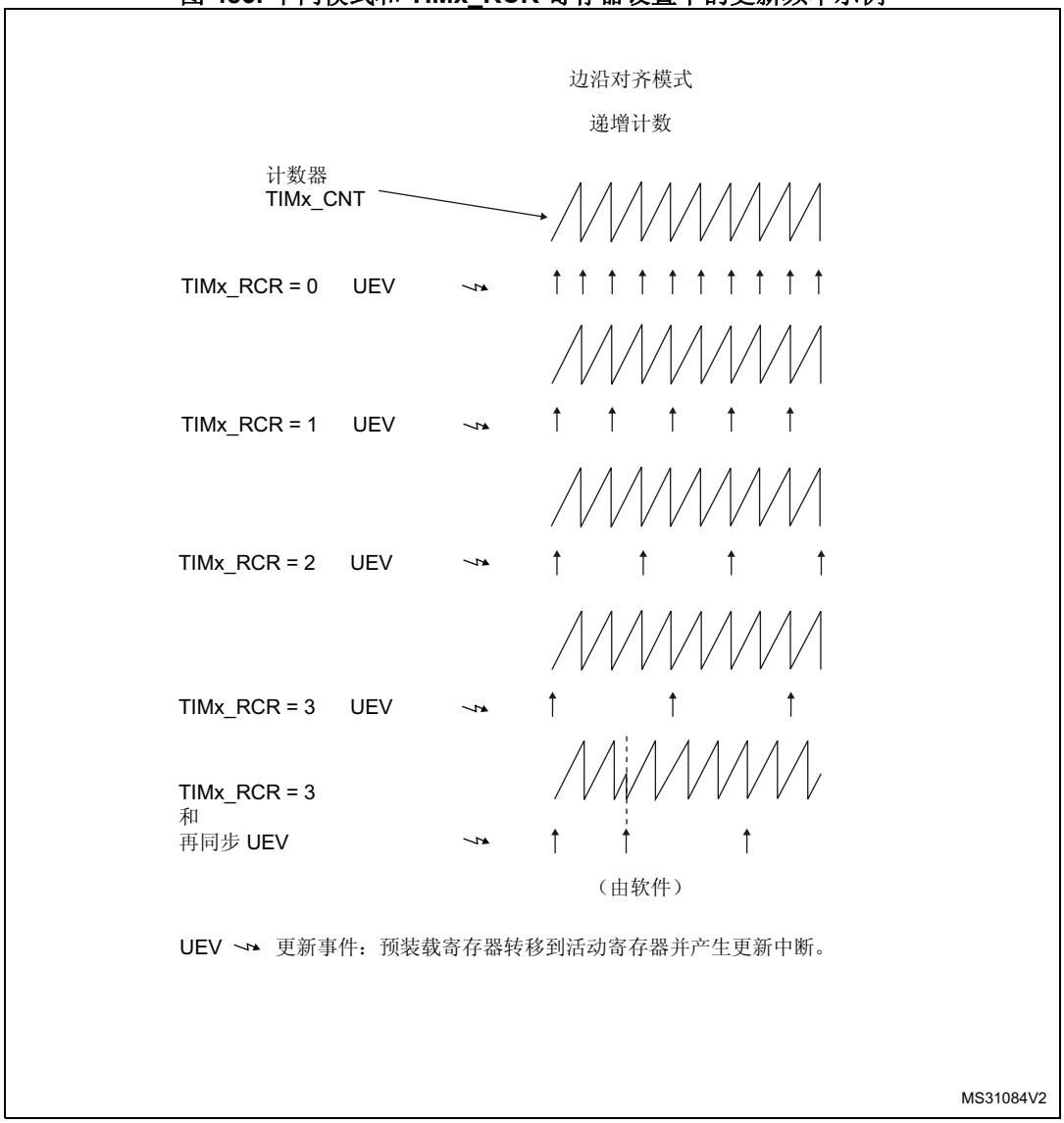
*第 41.4.1 节：时基单元*介绍如何因计数器上溢而生成更新事件 (UEV)。实际上，只有当重复计数器达到零时，才会生成更新事件。这在生成 PWM 信号时很有用。

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

重复计数器在每个计数器上溢时递减。

重复计数器是自动重载类型；其重复率为 TIMx_RCR 寄存器所定义的值（请参见*图 483*）。当更新事件由软件（通过将 TIMx_EGR 寄存器的 UG 位置 1）或硬件（通过从模式控制器）生成时，无论重复计数器的值为多少，更新事件都将立即发生，并且在重复计数器中重新装载 TIMx_RCR 寄存器的内容。

图 483. 不同模式和 TIMx_RCR 寄存器设置下的更新频率示例



41.4.4 时钟选择

计数器时钟可由下列时钟源提供：

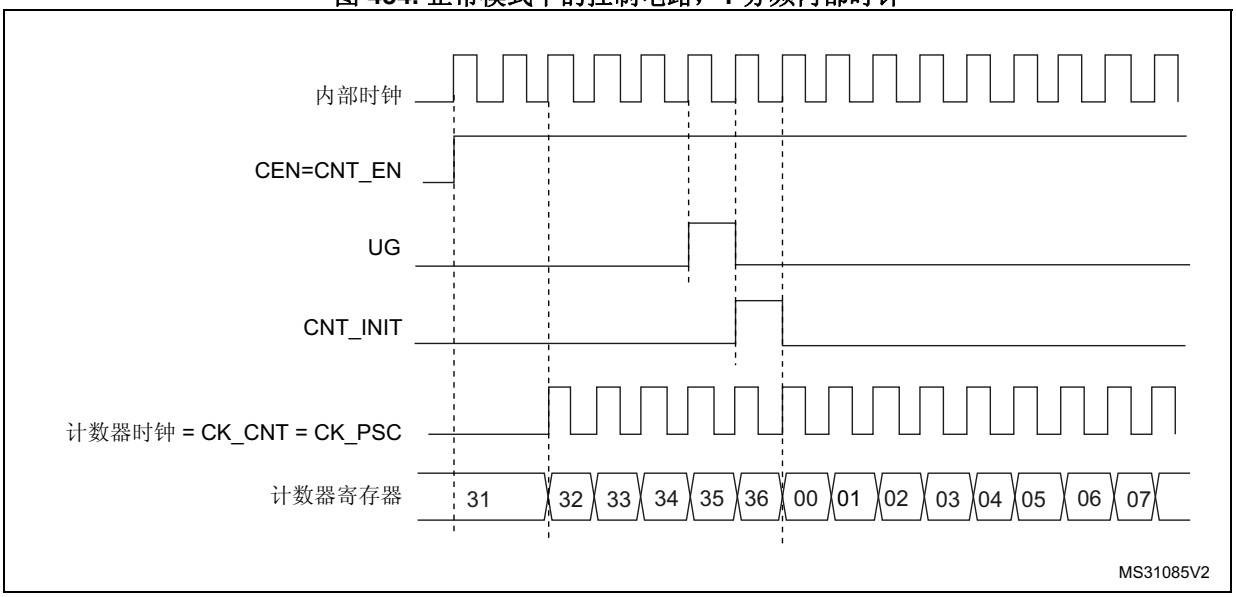
- 内部时钟 (CK_INT)
- 外部时钟模式 1：外部输入引脚
- 内部触发输入 (ITRx)（仅适用于 TIM15）：使用一个定时器作为另一个定时器的预分频器，例如可以将 TIM1 配置为 TIM15 的预分频器。更多详细信息，请参见*第 1527 页的将一个定时器用作另一个定时器的预分频器*。

内部时钟源 (CK_INT)

如果禁止从模式控制器 (SMS=000)，则 CEN 位、DIR 位（TIMx_CR1 寄存器中）和 UG 位（TIMx_EGR 寄存器中）为实际控制位，并且只能通过软件进行更改（UG 除外，仍保持自动清零）。当对 CEN 位写入 1 时，预分频器的时钟就由内部时钟 CK_INT 提供。

图 484 显示了正常模式下控制电路与递增计数器的行为（没有预分频的情况下）。

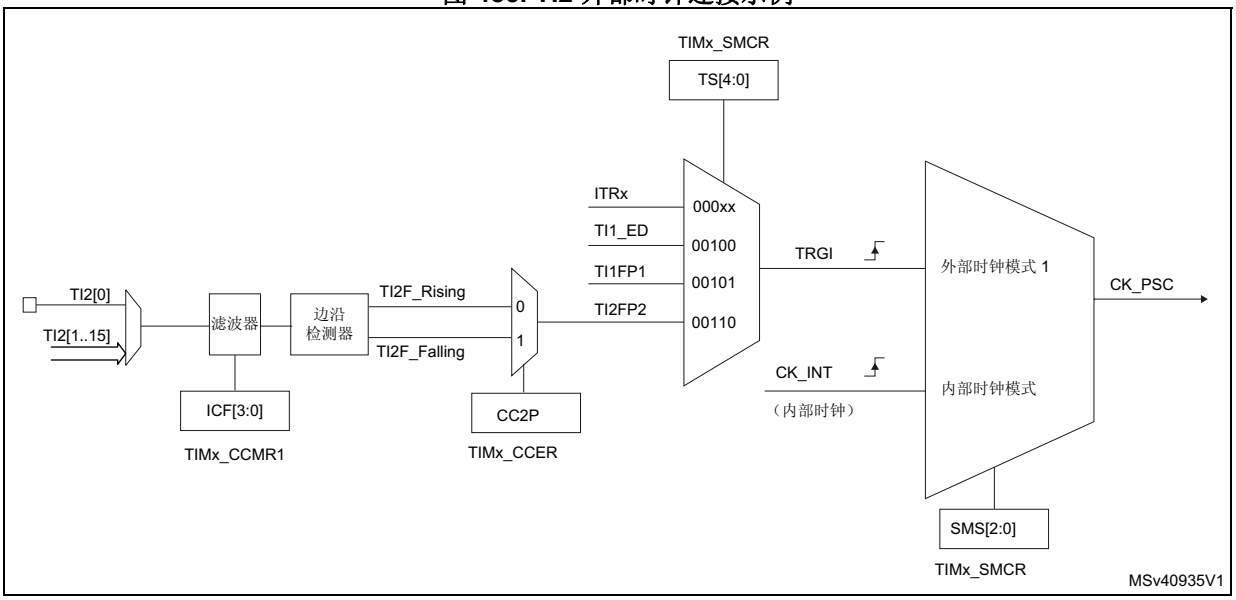
图 484. 正常模式下的控制电路，1 分频内部时钟



外部时钟源模式 1

当 TIMx_SMCR 寄存器中的 SMS=111 时，可选择此模式。计数器可在选定的输入信号上出现上升沿或下降沿时计数。

图 485. TI2 外部时钟连接示例



例如，要使递增计数器在 TI2 输入出现上升沿时计数，请执行以下步骤：

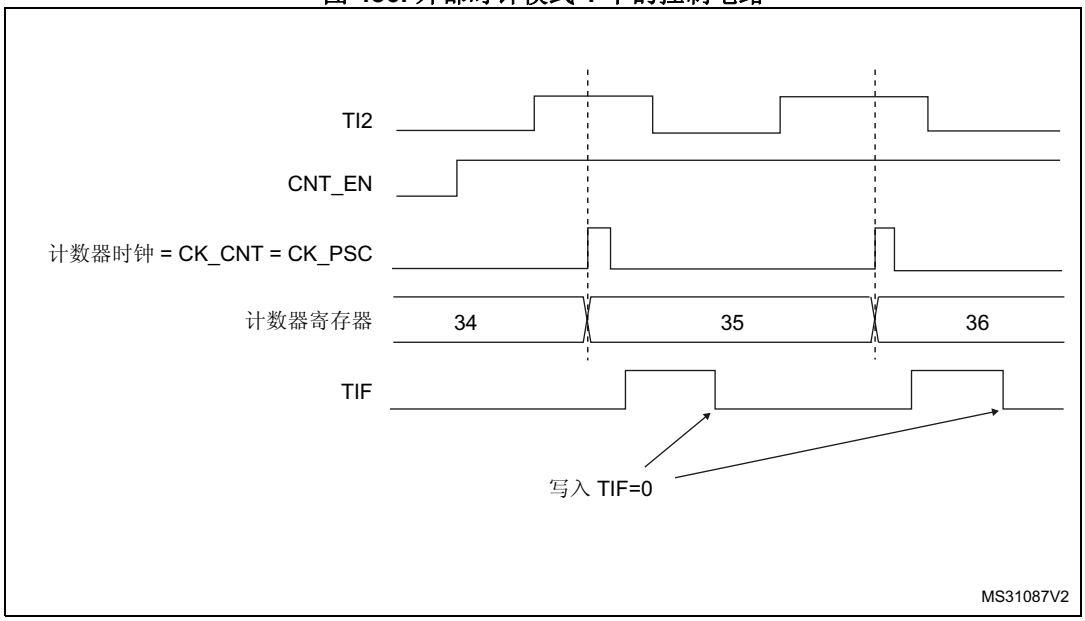
1. 使用 TIMx_TISEL 寄存器中的 TI2SEL[3:0] 位选择正确的 TI2[x] 源（内部或外部）。
2. 通过在 TIMx_CCMR1 寄存器中写入 CC2S = “01” 来配置通道 2，使其能够检测 TI2 输入的上升沿。
3. 通过在 TIMx_CCMR1 寄存器中写入 IC2F[3:0] 位来配置输入滤波带宽（如果不需要任何滤波器，请保持 IC2F=0000）。
4. 通过在 TIMx_CCER 寄存器中写入 CC2P = 0 来选择上升沿极性。
5. 通过在 TIMx_SMCR 寄存器中写入 SMS=111，使定时器在外部时钟模式 1 下工作。
6. 通过在 TIMx_SMCR 寄存器中写入 TS=00110 来选择 TI2 作为触发输入源。
7. 通过在 TIMx_CR1 寄存器中写入 CEN=1 来使能计数器。

注： *由于捕获预分频器不用于触发操作，因此无需对其进行配置。*

当 TI2 出现上升沿时，计数器便会计数一次并且 TIF 标志置 1。

TI2 的上升沿与实际计数器时钟之间的延迟是由于 TI2 输入的重新同步电路引起的。

图 486. 外部时钟模式 1 下的控制电路



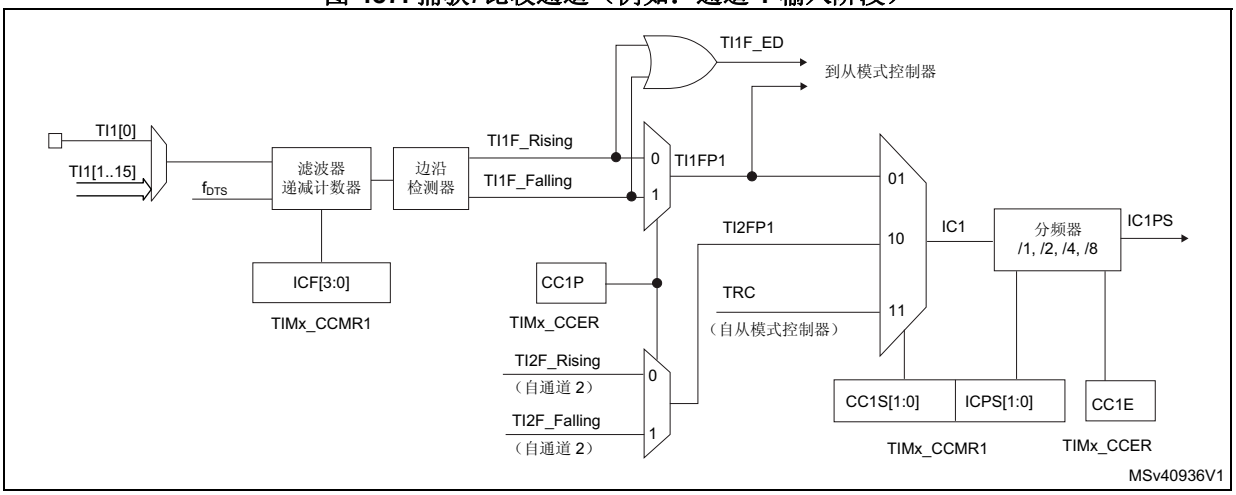
41.4.5 捕获/比较通道

每个捕获/比较通道均围绕一个捕获/比较寄存器（包括一个影子寄存器）、一个捕获输入阶段（数字滤波、多路复用和预分频器）和一个输出阶段（比较器和输出控制）构建而成。

图 487 到*图 490* 概括介绍了一个捕获/比较通道。

输入阶段对相应的 TIx 输入进行采样，生成一个滤波后的信号 TIxF。然后，带有极性选择功能的边沿检测器生成一个信号 (TIxFPx)，该信号可用作从模式控制器的触发输入，也可用作捕获命令。该信号先进行预分频 (ICxPS)，而后再进入捕获寄存器。

图 487. 捕获/比较通道（例如：通道 1 输入阶段）



输出阶段生成一个中间波形作为基准：OCxRef（高电平有效）。链的末端决定最终输出信号的极性。

图 488. 捕获/比较通道 1 主电路

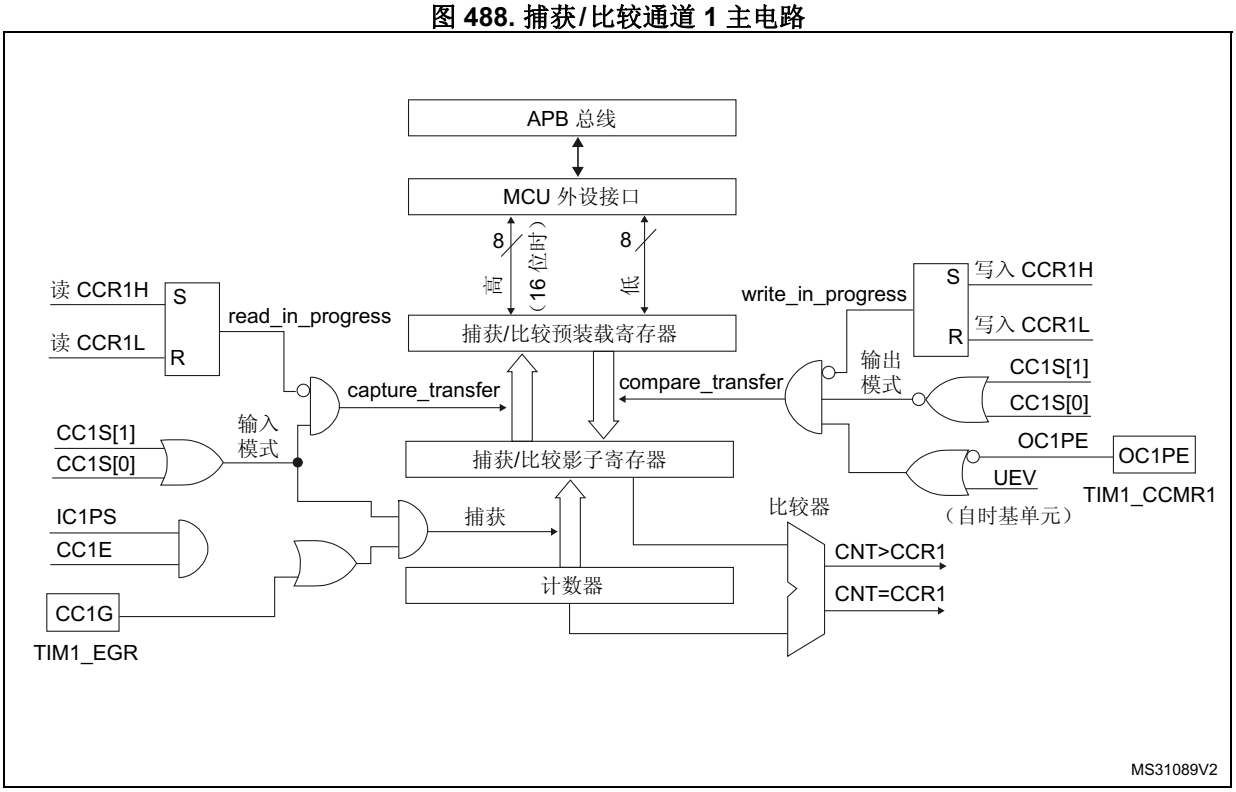


图 489. 捕获/比较通道的输出阶段（通道 1）

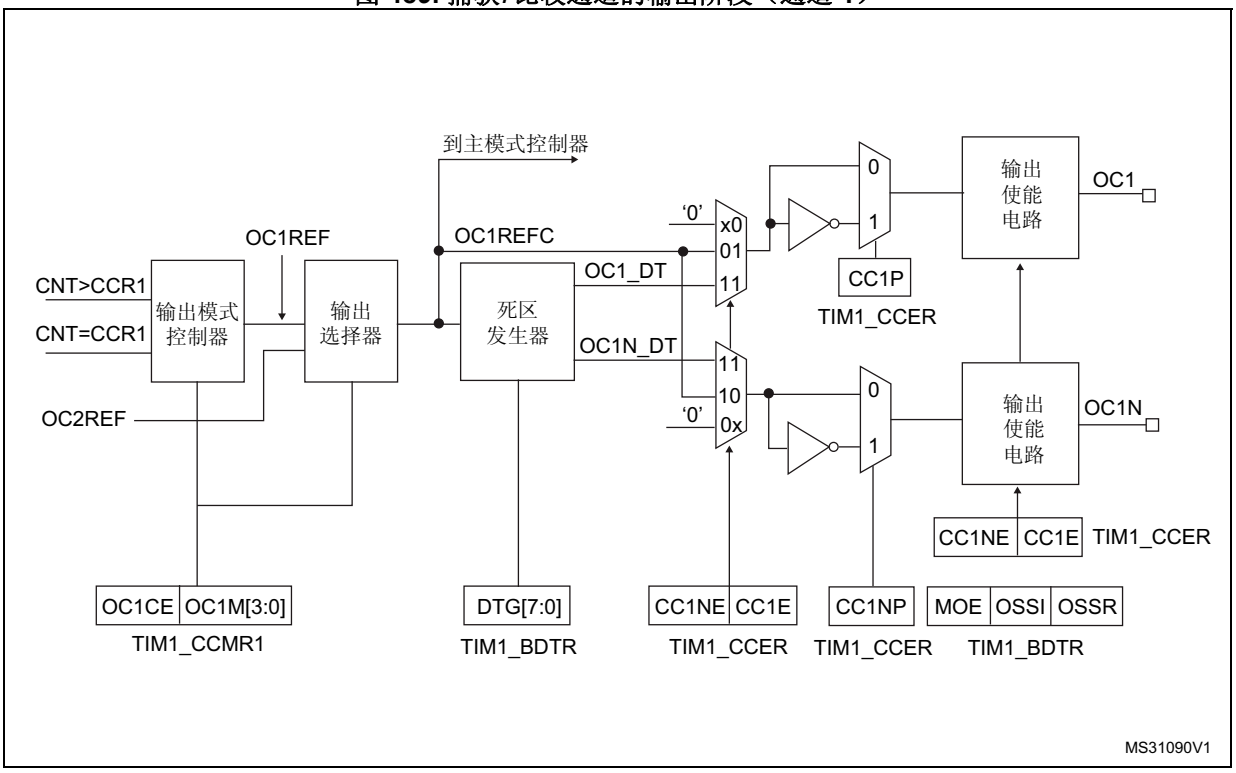
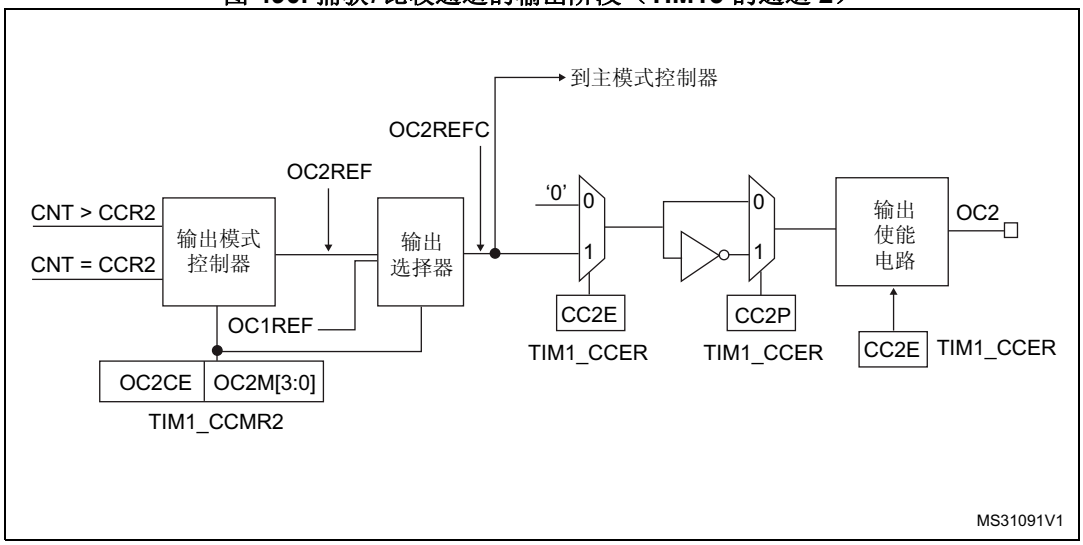


图 490. 捕获/比较通道的输出阶段（TIM15 的通道 2）



捕获/比较模块由一个预装载寄存器和一个影子寄存器组成。始终可通过读写操作访问预装载寄存器。

在捕获模式下，捕获实际发生在影子寄存器中，然后将影子寄存器的内容复制到预装载寄存器中。

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

41.4.6 输入捕获模式

在输入捕获模式下，当相应的 ICx 信号检测到跳变沿后，将使用捕获/比较寄存器 (TIMx_CCRx) 来锁存计数器的值。发生捕获事件时，会将相应的 CCXIF 标志（TIMx_SR 寄存器）置 1，并可发送中断或 DMA 请求（如果已使能）。如果发生捕获事件时 CCxIF 标志已处于高位，则会将重复捕获标志 CCxOF（TIMx_SR 寄存器）置 1。可通过软件将 CCxIF 清零，方法是：向 CCxIF 写入“0”，或读取存储在 TIMx_CCRx 寄存器中的已捕获数据。向 CCxOF 写入“0”后会将其清零。

以下示例说明了如何在 TI1 输入出现上升沿时将计数器的值捕获到 TIMx_CCR1 中。具体操作步骤如下：

1. 使用 TIMx_TISEL 寄存器中的 TI1SEL[3:0] 位选择正确的 TI1x 源（内部或外部）。
2. 选择有效输入：TIMx_CCR1 必须连接到 TI1 输入，因此向 TIMx_CCMR1 寄存器中的 CC1S 位写入 01。只要 CC1S 不等于 00，就会将通道配置为输入模式，并且 TIMx_CCR1 寄存器将处于只读状态。
3. 根据连接到定时器的信号，对所需的输入滤波带宽进行编程（如果输入为 TIx 之一，则对 TIMx_CCMRx 寄存器中的 ICxF 位进行编程）。假设信号边沿变化时，输入信号最多在 5 个内部时钟周期内发生抖动。因此，我们必须将滤波带宽设置为大于 5 个内部时钟周期。在检测到 8 个具有新电平的连续采样（以 f_{DTS} 频率采样）后，可以确认 TI1 上的跳变沿。然后向 TIMx_CCMR1 寄存器中的 IC1F 位写入 0011。
4. 通过在 TIMx_CCER 寄存器中将 CC1P 位写入 0，选择 TI1 上的有效转换边沿（本例中为上升沿）。
5. 对输入预分频器进行编程。在本例中，我们希望每次有效转换时都执行捕获操作，因此需要禁止预分频器（向 TIMx_CCMR1 寄存器中的 IC1PS 位写入“00”）。
6. 通过将 TIMx_CCER 寄存器中的 CC1E 位置 1，允许将计数器的值捕获到捕获寄存器中。
7. 如果需要，可通过将 TIMx_DIER 寄存器中的 CC1IE 位置 1 来使能相关中断请求，并且/或者通过将该寄存器中的 CC1DE 位置 1 来使能 DMA 请求。

发生输入捕获时：

- 发生有效跳变沿时，TIMx_CCR1 寄存器会获取计数器的值。
- 将 CC1IF 标志置 1（中断标志）。如果至少发生了两次连续捕获，但 CC1IF 标志未被清零，这样 CC1OF 捕获溢出标志会被置 1。
- 根据 CC1IE 位生成中断。
- 根据 CC1DE 位生成 DMA 请求。

要处理重复捕获，建议在读出捕获溢出标志之前读取数据。这样可避免丢失在读取捕获溢出标志之后与读取数据之前可能出现的重复捕获信息。

注： *通过软件将 TIMx_EGR 寄存器中的相应 CCxG 位置 1 可生成 IC 中断和/或 DMA 请求。*

41.4.7 PWM 输入模式（仅适用于 TIM15）

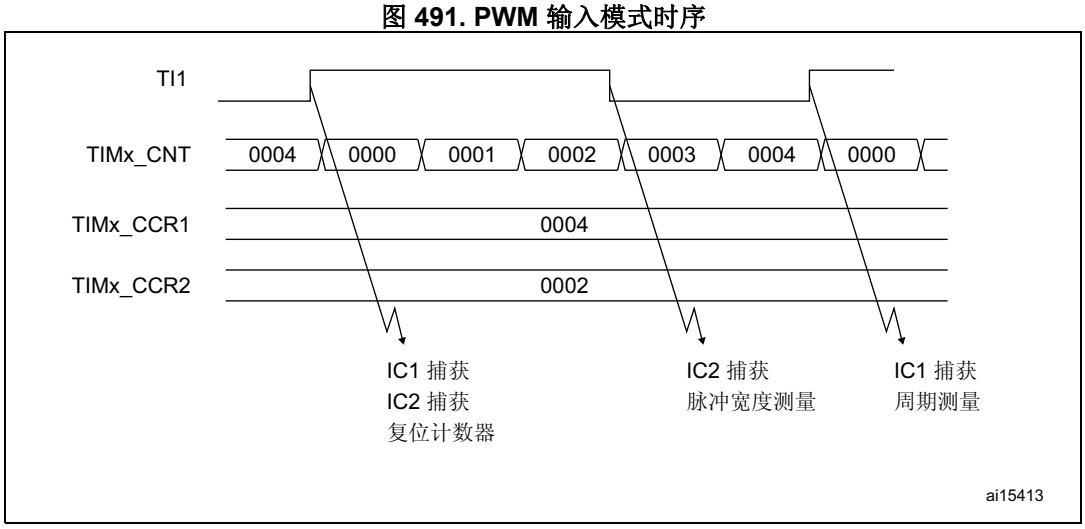
此模式是输入捕获模式的一个特例。其实现步骤与输入捕获模式基本相同，仅存在以下不同之处：

- 两个 ICx 信号被映射至同一个 TIx 输入。
- 这两个 ICx 信号在边沿处有效，但极性相反。
- 选择两个 TIxFP 信号之一作为触发输入，并将从模式控制器配置为复位模式。

例如，可通过以下步骤对应用于 TI1 的 PWM 的周期（位于 TIMx_CCR1 寄存器中）和占空比（位于 TIMx_CCR2 寄存器中）进行测量（取决于 CK_INT 频率和预分频器的值）：

1. 使用 TIMx_TISEL 寄存器中的 TI1SEL[3:0] 位选择正确的 TI1[x] 源（内部或外部）。
2. 选择 TIMx_CCR1 的有效输入：向 TIMx_CCMR1 寄存器中的 CC1S 位写入 01（选择 TI1）。
3. 选择 TI1FP1 的有效极性（用于在 TIMx_CCR1 中捕获和计数器清零）：向 CC1P 位和 CC1NP 位写入“0”（上升沿有效）。
4. 选择 TIMx_CCR2 的有效输入：向 TIMx_CCMR1 寄存器中的 CC2S 写入 10（选择 TI1）。
5. 选择 TI1FP2 的有效极性（用于在 TIMx_CCR2 中捕获）：向 CC2P 位和 CC2NP 位写入“1”（下降沿有效）。
6. 选择有效触发输入：向 TIMx_SMCR 寄存器中的 TS 位写入 00101（选择 TI1FP1）。
7. 将从模式控制器配置为复位模式：向 TIMx_SMCR 寄存器中的 SMS 位写入 100。
8. 使能捕获：向 TIMx_CCER 寄存器中的 CC1E 位和 CC2E 位写入“1”。

图 491. PWM 输入模式时序



1. PWM 输入模式只能与 TIMx_CH1/TIMx_CH2 信号配合使用，因为只有 TI1FP1 和 TI2FP2 与从模式控制器相连。

41.4.8 强制输出模式

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

要将输出比较信号 (OCXREF/OCx) 强制设置为有效电平，只需向相应 TIMx_CCMRx 寄存器中的 OCxM 位写入 101。OCXREF 进而强制设置为高电平（OCxREF 始终为高电平有效），同时 OCx 获取 CCxP 极性位的相反值。

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

通过向 TIMx_CCMRx 寄存器中的 OCxM 位写入 100，可将 OCxREF 信号强制设置为低电平。

无论如何，TIMx_CCRx 影子寄存器与计数器之间的比较仍会执行，而且允许将标志置 1。因此可发送相应的中断和 DMA 请求。下面的输出比较模式一节对此进行了介绍。

41.4.9 输出比较模式

此功能用于控制输出波形，或指示已经过某一时间段。

当捕获/比较寄存器与计数器之间相匹配时，输出比较功能：

- 将为相应的输出引脚分配一个可编程值，该值由输出比较模式（TIMx_CCMRx 寄存器中的 OCxM 位）和输出极性（TIMx_CCER 寄存器中的 CCxP 位）定义。匹配时，输出引脚既可保持其电平 (OCXM=000)，也可设置为有效电平 (OCXM=001)、无效电平 (OCXM=010) 或进行翻转 (OCxM=011)。
- 将中断状态寄存器中的标志置 1（TIMx_SR 寄存器中的 CCxIF 位）。
- 如果相应中断使能位（TIMx_DIER 寄存器中的 CCXIE 位）置 1，将生成中断。
- 如果相应使能位（TIMx_DIER 寄存器的 CCxDE 位，TIMx_CR2 寄存器的 CCDS 位，用来选择 DMA 请求）置 1，将发送 DMA 请求。

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

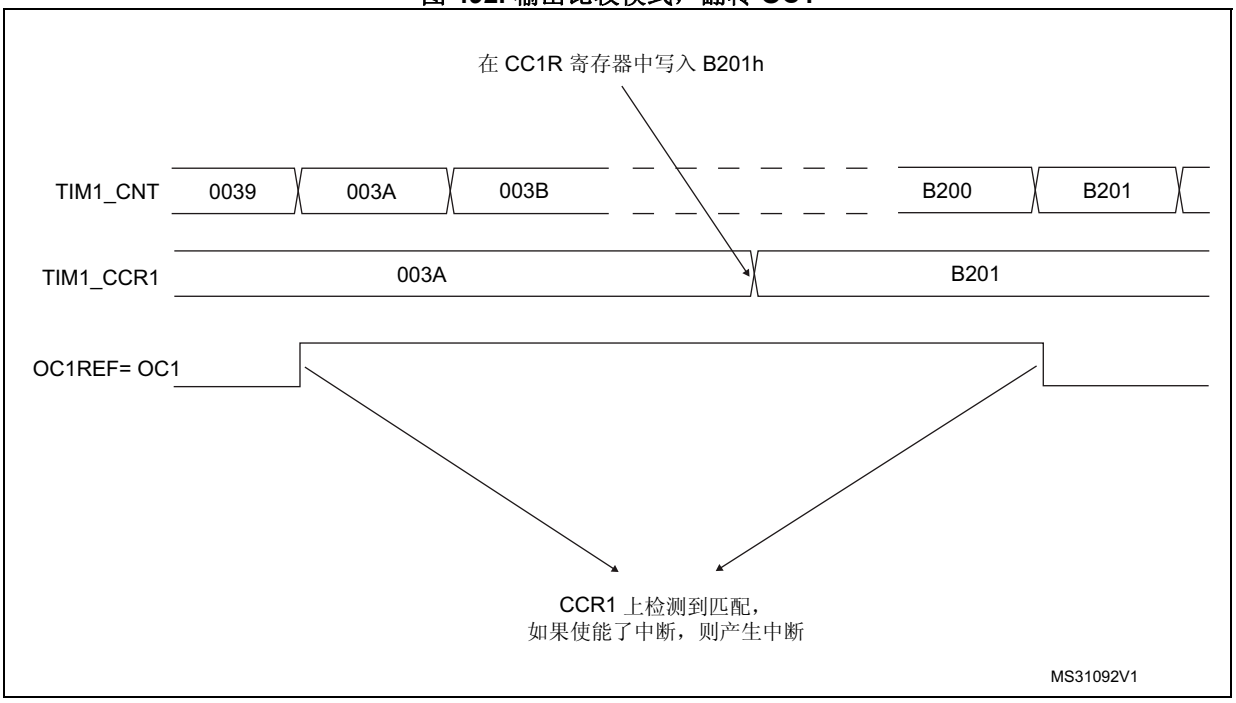
在输出比较模式下，更新事件 UEV 对 OCxREF 和 OCx 输出毫无影响。同步的精度可以达到计数器的一个计数周期。输出比较模式也可用于输出单脉冲（在单脉冲模式下）。

步骤

1. 选择计数器时钟（内部、外部、预分频器）。
2. 在 TIMx_ARR 和 TIMx_CCRx 寄存器中写入所需数据。
3. 如果要生成中断请求，则需将 CCxIE 位置 1。
4. 选择输出模式。例如：
   - 当 CNT 与 CCRx 匹配时，写入 OCxM = 011 以翻转 OCx 输出引脚
   - 写入 OCxPE = 0 以禁止预装载寄存器
   - 写入 CCxP = 0 以选择高电平有效极性
   - 写入 CCxE = 1 以使能输出
5. 通过将 TIMx_CR1 寄存器中的 CEN 位置 1 来使能计数器。

可通过软件随时更新 TIMx_CCRx 寄存器以控制输出波形，前提是未使能预加载寄存器（OCxPE=“0”，否则 TIMx_CCRx 影子寄存器仅在下一更新事件 UEV 发生时进行更新）。*图 492* 给出了一个示例。

图 492. 输出比较模式，翻转 OC1



41.4.10 PWM 模式

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

各通道可以独立选择 PWM 模式（每个 OCx 输出对应一个 PWM），只需向 TIMx_CCMRx 寄存器的 OCxM 位写入“110”（PWM 模式 1）或“111”（PWM 模式 2）。必须通过将 TIMx_CCMRx 寄存器中的 OCxPE 位置 1 使能相应预装载寄存器，最后通过将 TIMx_CR1 寄存器中的 ARPE 位置 1 使能自动重载预装载寄存器（在递增计数或中心对齐模式下）。

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

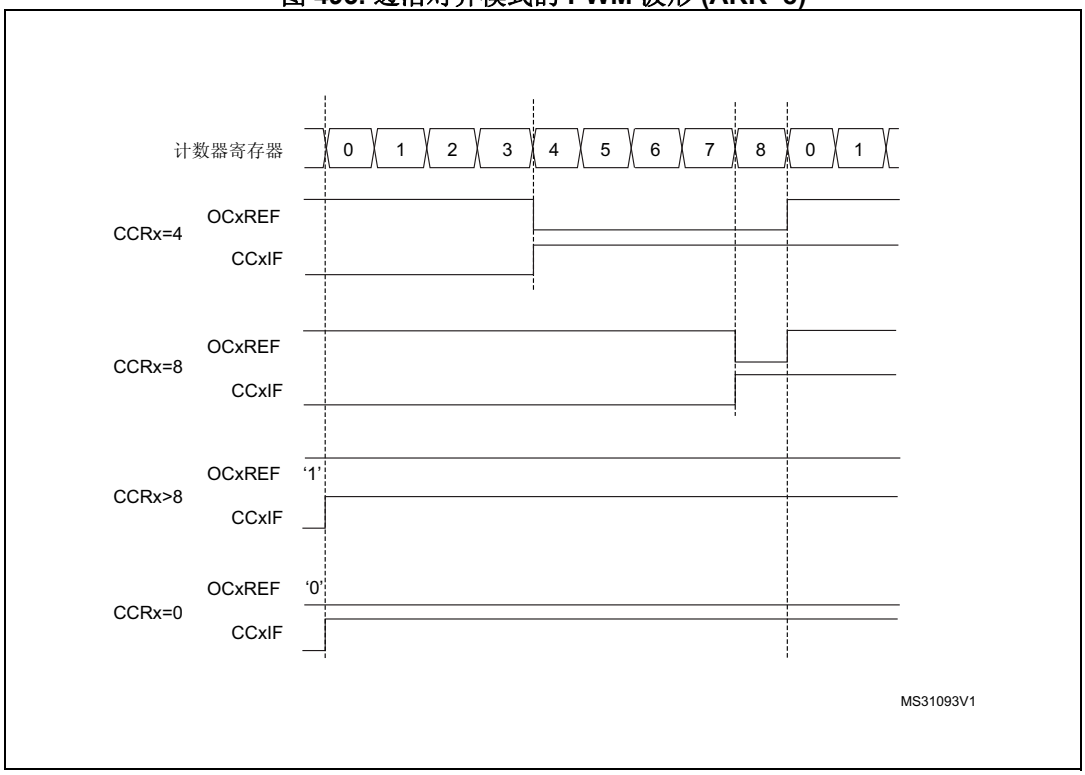
OCx 极性可通过软件来编程（使用 TIMx_CCER 寄存器的 CCxP 位）。可将其编程为高电平有效或低电平有效。通过 CCxE、CCxNE、MOE、OSSI 和 OSSR 位（TIMx_CCER 和 TIMx_BDTR 寄存器）的组合使能 OCx 输出。有关详细信息，请参见 TIMx_CCER 寄存器说明。

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

TIM15/TIM16/TIM17 只能递增计数。请参见*第 1616 页的递增计数模式*。

以下以 PWM 模式 1 为例。只要 TIMx_CNT < TIMx_CCRx，PWM 参考信号 OCxREF 便为高电平，否则为低电平。如果 TIMx_CCRx 中的比较值大于自动重载值（TIMx_ARR 中），则 OCxREF 保持为“1”。如果比较值为 0，则 OCxRef 保持为“0”。*图 493* 举例介绍边沿对齐模式的一些 PWM 波形 (TIMx_ARR=8)。

图 493. 边沿对齐模式的 PWM 波形 (ARR=8)



41.4.11 组合 PWM 模式（仅适用于 TIM15）

在组合 PWM 模式下，生成的两个边沿或中心对齐 PWM 信号的各个脉冲间允许存在可编程延时和相移。频率由 TIMx_ARR 寄存器的值确定，而占空比和延时则由两个 TIMx_CCRx 寄存器确定。产生的信号 OCxREFC 由两个参考 PWM 的逻辑或运算或者逻辑与运算组合组成。

- OC1REFC（或 OC2REFC）由 TIMx_CCR1 和 TIMx_CCR2 寄存器控制

两个通道可以独立选择组合 PWM 模式（每对 CCR 寄存器一个 OCx 输出），只需向 TIMx_CCMRx 寄存器的 OCxM 位写入“1100”（组合 PWM 模式 1）或“1101”（组合 PWM 模式 2）。

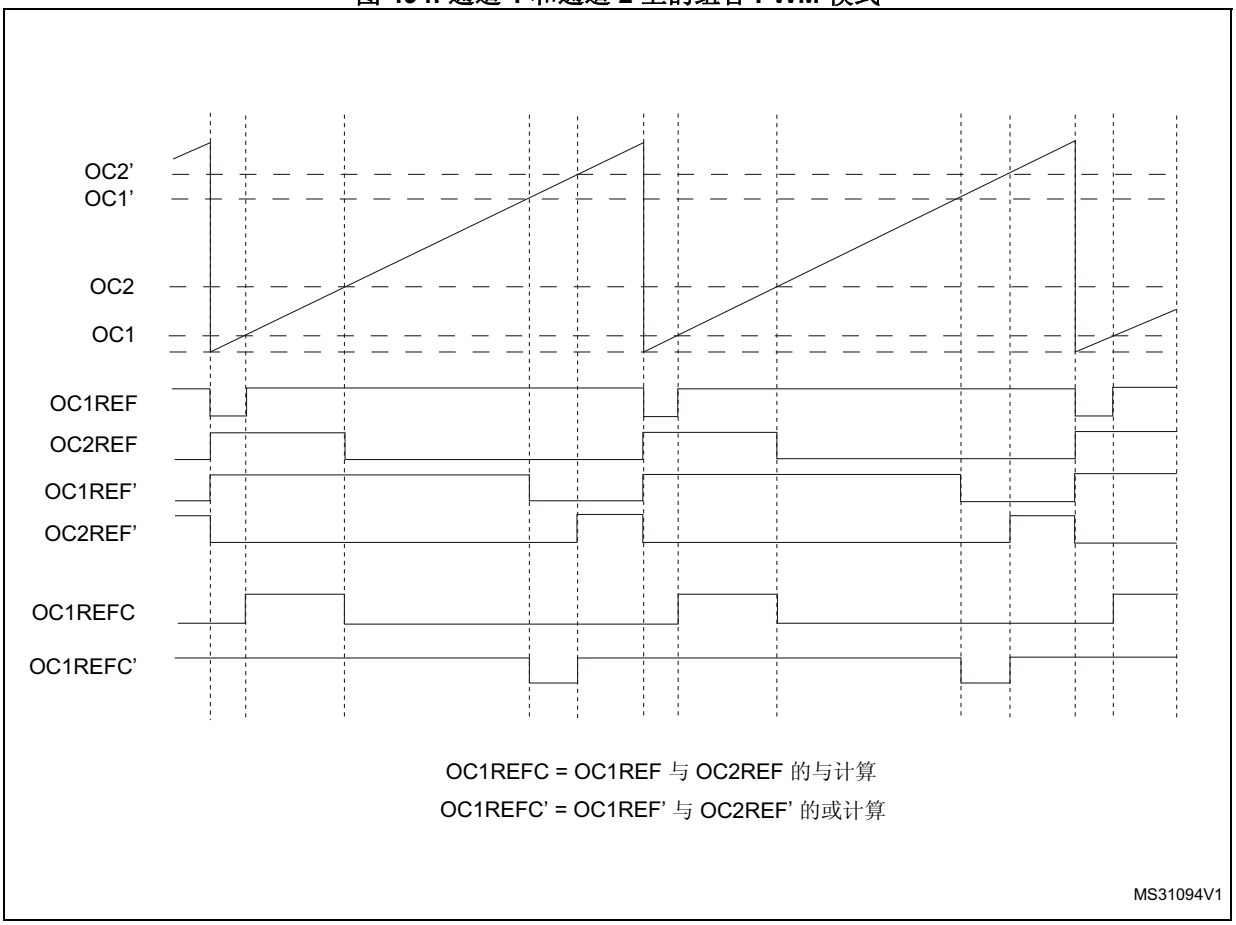
当给定通道用作组合 PWM 通道时，其互补通道必须在相反的 PWM 模式下配置（例如，一个通道在组合 PWM 模式 1 下配置，另一个通道在组合 PWM 模式 2 下配置）。

注： *出于兼容性原因，OCxM[3:0] 位域分为两部分，最高有效位与最低有效的 3 位不相邻。*

图 494 显示了不对称 PWM 模式下可以产生的信号示例，通过以下配置可获得这些信号：

- 通道 1 在组合 PWM 模式 2 下配置
- 通道 2 在 PWM 模式 1 下配置

图 494. 通道 1 和通道 2 上的组合 PWM 模式



41.4.12 互补输出和死区插入

TIM15/TIM16/TIM17 通用定时器可以输出一路互补信号，并管理输出的关断和接通。

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

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

互补信号 OCx 和 OCxN 通过以下多个控制位的组合进行激活：TIMx_CCER 寄存器中的 CCxE 和 CCxNE 位以及 TIMx_BDTR 和 TIMx_CR2 寄存器中的 MOE、OISx、OISxN、OSSI 和 OSSR 位。更多详细信息，请参见*第 1677 页的表 328：具有断路功能的互补通道 OCx 和 OCxN 的输出控制位 (TIM16/17)*。应当注意，切换至空闲状态（MOE 下降到 0）的时刻，死区仍然有效。

CCxE 和 CCxNE 位同时置 1 并且 MOE 位置 1（如果存在断路）时，将使能死区插入。每个通道有一个 10 位死区发生器。将基于参考波形 OCxREF 生成 2 个输出 OCx 和 OCxN。如果 OCx 和 OCxN 为高电平有效：

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

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

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

图 495. 带死区插入的互补输出。

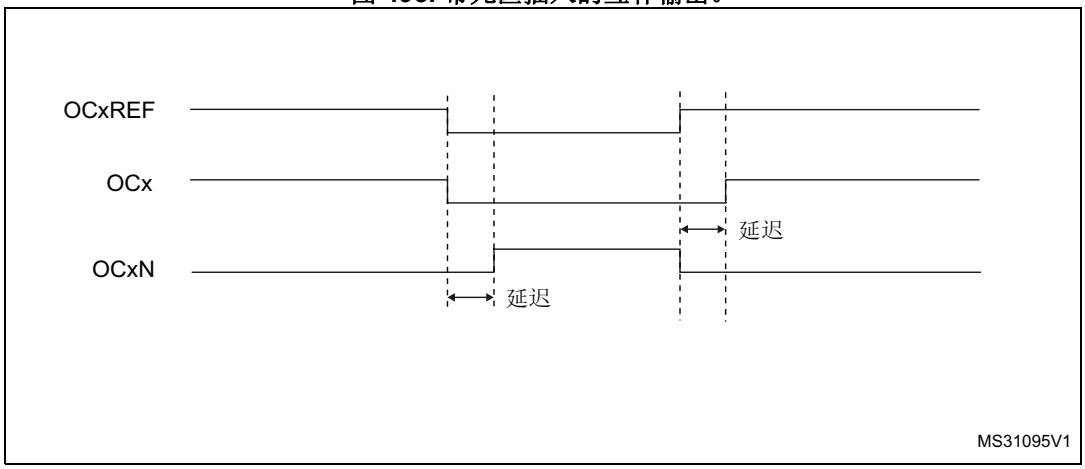


图 496. 延迟时间大于负脉冲宽度的死区波形。

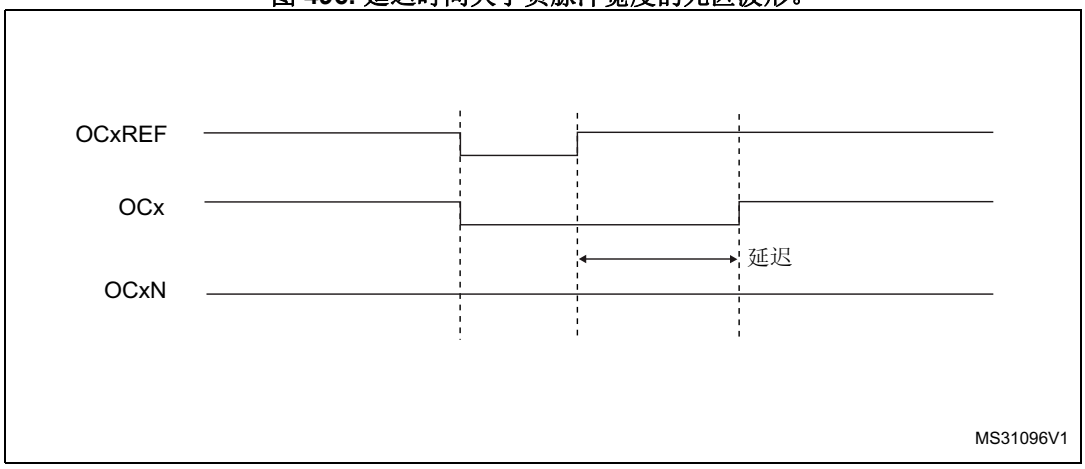
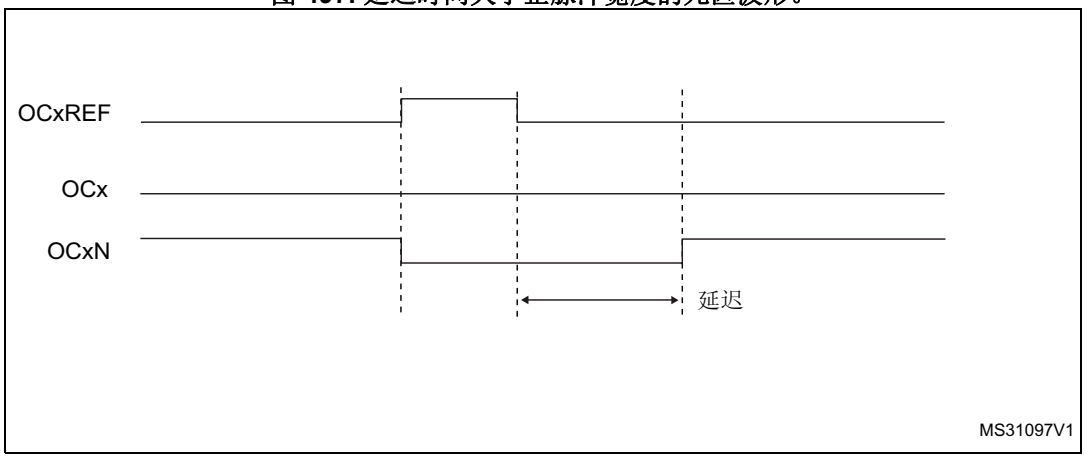


图 497. 延迟时间大于正脉冲宽度的死区波形。



死区延迟对于所有通道均相同，可通过 TIMx_BDTR 寄存器中的 DTG 位进行编程。
有关延迟时间计算的信息，请参见*第 1680 页的第 41.6.13 节：TIM16/TIM17 断路和死区寄存器 (TIMx_BDTR)*。

将 OCxREF 重定向到 OCx 或 OCxN

在输出模式（强制输出模式、输出比较模式或 PWM 模式）下，通过配置 TIMx_CCER 寄存器中的 CCxE 和 CCxNE 位，可将 OCxREF 重定向到 OCx 输出或 OCxN 输出。

通过此功能，可以在一个输出上发送特定波形（如 PWM 或静态有效电平），而同时使互补输出保持其无效电平。或者，使两个输出同时保持无效电平，或者两个输出同时处于有效电平，两者互补并且带死区。

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

41.4.13 使用断路功能

断路功能的目的是保护由 TIM15/TIM16/TIM17 定时器生成的 PWM 信号所驱动的电源开关。断路输入通常被连接到功率级和三相逆变器的故障输出。激活时，断路电路会关闭 PWM 输出，并将其强制为预定义的安全状态。

断路通道收集系统级故障（时钟失效和奇偶校验错误等）和应用故障（来自输入引脚和内置比较器），可以在死区持续时间后将输出强制为预定义的电平（有效或无效）。

断路期间的输出使能信号和输出电平取决于多个控制位：

- TIMx_BDTR 寄存器中的 MOE 位，允许通过软件使能/禁止输出，在发生断路和断路 2 事件时复位。
- TIMx_BDTR 寄存器中的 OSSI 位，定义定时器将输出控制在无效状态下，还是释放对 GPIO 控制器的控制（通常使其处于高阻态模式）。
- TIMx_CR2 寄存器中的 OISx 和 OISxN 位，将输出设置为关断电平（有效或无效）。无论 OISx 和 OISxN 的值为何，均无法在给定时间将 OCx 和 OCxN 输出同时设置为有效电平。更多详细信息，请参见*第 1657 页的表 326：具有断路功能的互补通道 OCx 和 OCxN 的输出控制位 TIM15*。

退出复位状态后，断路功能处于禁止状态，MOE 位处于低电平。通过设置 TIMx_BDTR 寄存器中的 BKE 位来使能断路功能。断路输入的极性可通过该寄存器中的 BKP 位来选择。BKE 和 BKP 位可同时修改。对 BKE 和 BKP 位执行写操作时，写操作会在 1 个 APB 时钟周期的延迟后生效。因此，执行写操作后，需要等待 1 个 APB 时钟周期，才能准确回读该位。

由于 MOE 下降沿可能是异步信号，因此在实际信号（作用于输出）与同步控制位（位于 TIMx_BDTR 寄存器中）之间插入了再同步电路，从而在异步信号与同步信号之间产生延迟。具体而言，如果在 MOE 处于低电平时向其写入 1，则必须首先插入延迟（空指令），才能准确进行读取。这是因为写入的是异步信号，而读取的却是同步信号。

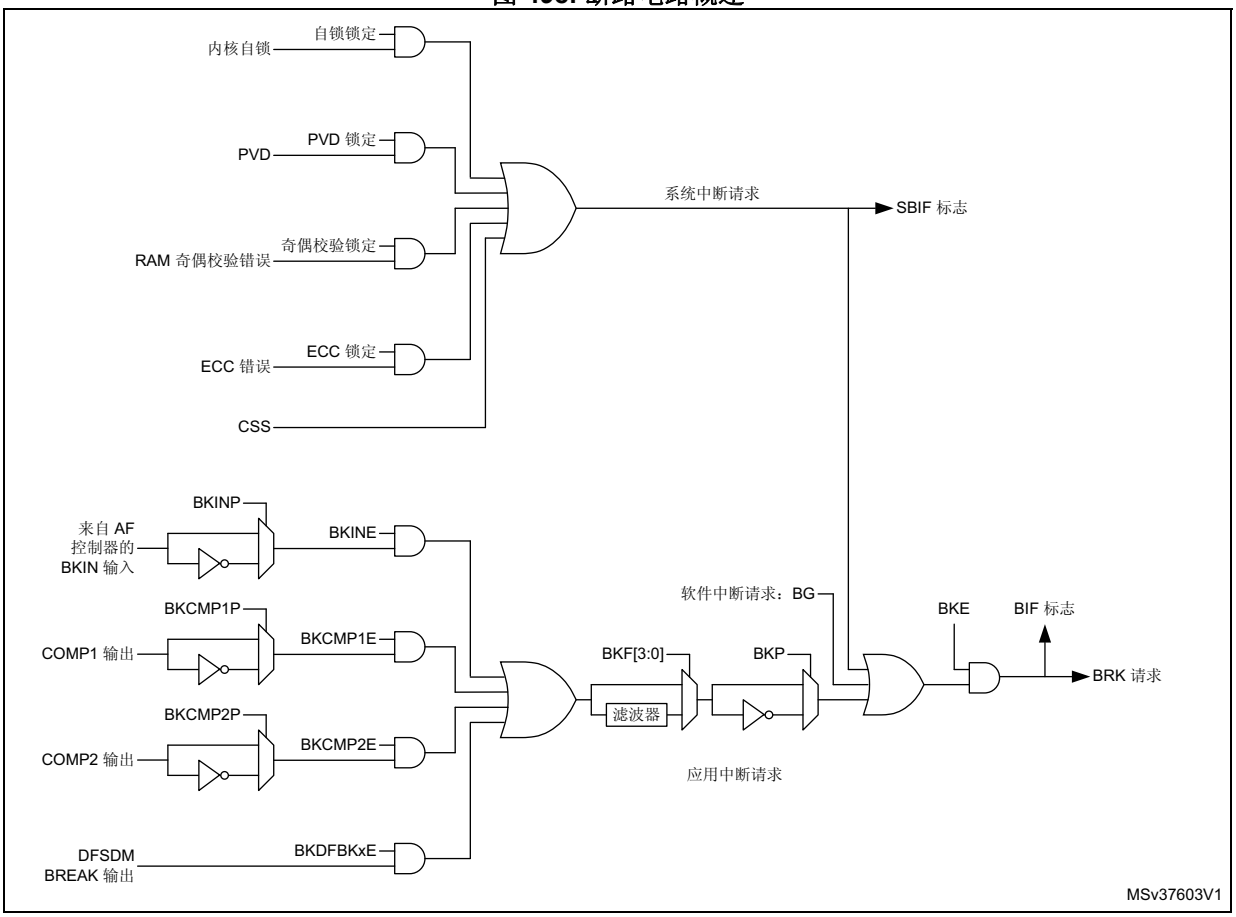
- 滤波器可编程（TIMx_BDTR 寄存器中的 BKF[3:0] 位），以避免伪事件。

可以使用 TIMx_OR2 寄存器从多个源产生断路，这些源可以单独使能并使用可编程边沿有效。

断路 (BRK) 通道的源为：

- 连接到 BKIN 引脚的外部源（由 AFIO 控制器设定），具有极性选择和可选的数字滤波
- 内部源：
  - 比较器的输出，具有极性选择和可选的数字滤波
  - DFSDM1 外设的模拟看门狗输出
  - 系统中断：
    - 带 FPU 的 Cortex®-M7 LOCKUP 输出
    - PVD 输出
    - SRAM 奇偶校验错误信号
    - Flash ECC 错误
    - CSS 检测器产生时钟故障事件

图 498. 断路电路概述



注： *只有禁止可编程滤波器时才能保证异步（无时钟）操作。如果使能可编程滤波器，必须使用故障安全时钟模式（例如，使用内部 PLL 和/或 CSS）来保证能够处理断路事件。*

注意： 只有禁止可编程滤波器时才能保证异步（无时钟）操作。如果使能可编程滤波器，必须使用故障安全时钟模式（例如，使用内部 PLL 和/或 CSS）来保证能够处理断路事件。

发生断路（断路输入上出现所选电平）时：

- MOE 位异步清零，使输出处于无效状态、空闲状态甚至释放对 AFIO 控制器的控制（通过 OSSI 位进行选择）。即使 MCU 振荡器关闭，该功能仍然有效。
- MOE=0 时，将以 TIMx_CR2 寄存器 OISx 位中编程的电平驱动每个输出通道。如果 OSSI = 0，定时器将释放输出控制（由 AFIO 控制器接管），否则使能输出保持高电平。
- 使用互补输出时：
  - 输出首先置于复位状态或无效状态（取决于极性）。这是异步操作，因此即使没有为定时器提供时钟，该操作仍有效。
  - 如果定时器时钟仍存在，则将重新激活死区发生器，进而在死区后以 OISx 和 OISxN 位中编程的电平驱动输出。即使在这种情况下，也不能同时将 OCx 和 OCxN 驱动至其有效电平。请注意，MOE 进行再同步，因此死区的持续时间会比通常情况长一些（约 2 个 ck_tim 时钟周期）。
  - 如果 OSSI = 0，定时器将释放使能输出（由强制高阻态的 AFIO 控制器接管），否则使能输出将保持高电平或在 CCxE 或 CCxNE 位之一为高电平时立即变为高电平。
- 将断路状态标志（TIMx_SR 寄存器中的 BIF 位）置 1。如果 TIMx_DIER 寄存器中的 BIE 位置 1，可产生中断。如果 TIMx_DIER 寄存器中的 BDE 位置 1，可发送 DMA 请求。
- 如果 TIMx_BDTR 寄存器中的 AOE 位置 1，则 MOE 位会在发生下一更新事件 (UEV) 时自动再次置 1。这一特性有许多用处，比如，可用于实现调节器的功能。否则，MOE 将始终保持低电平，直到再次向该位写入“1”。这种情况下，这一特性可用于确保安全。可以将断路输入连接到功率驱动器的警报、温度传感器或任何安全元件。

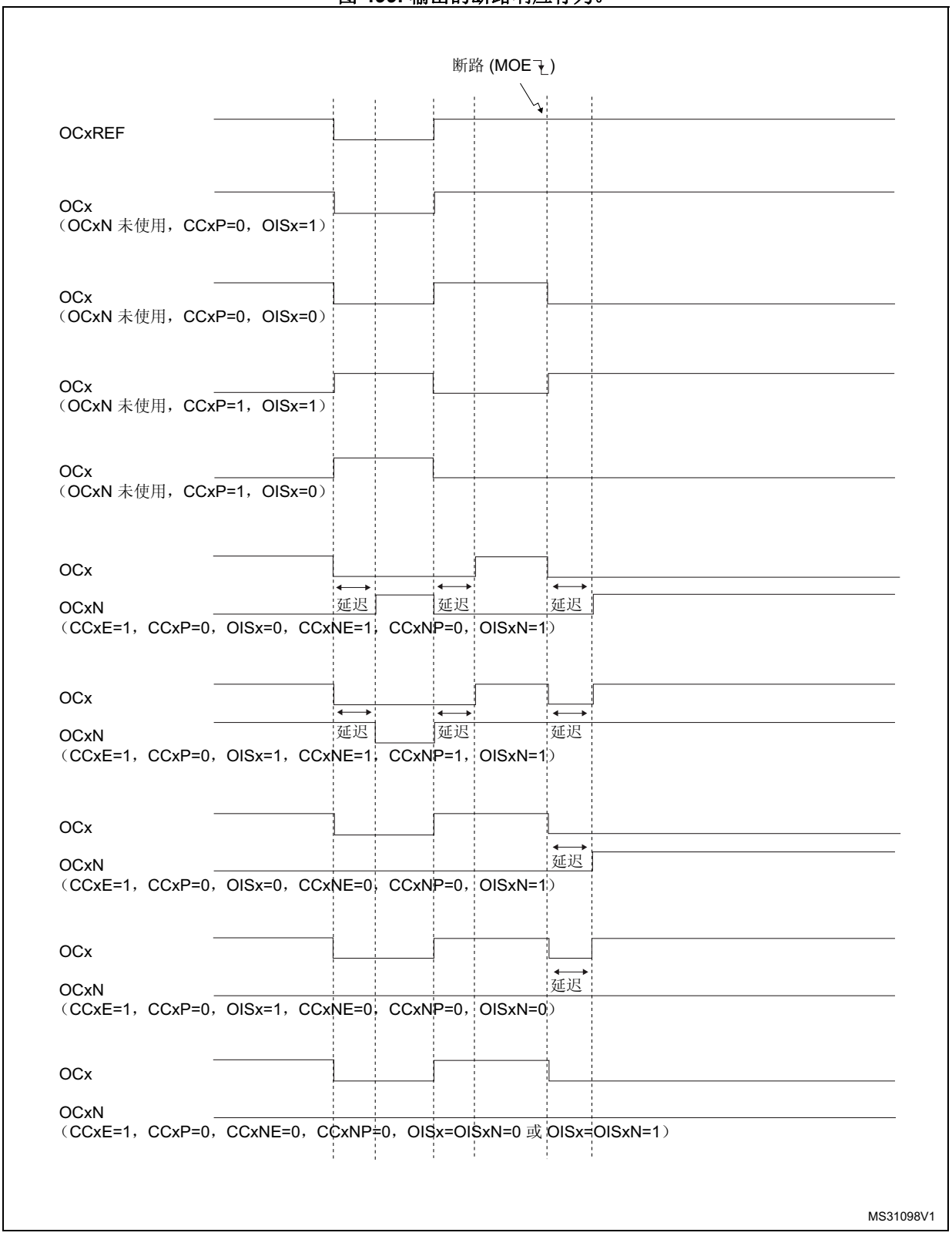
注： *断路输入为电平有效。因此，当断路输入有效电平时，不能将 MOE 位置 1（自动或通过软件）。同时，不能将状态标志 BIF 清零。*

断路可由 BRK 输入生成，该输入具有可编程极性，其使能位 BKE 位于 TIMx_BDTR 寄存器中。

除断路输入和输出管理外，断路电路内部还实施了写保护，用以保护应用的安全。通过该功能，用户可冻结多个参数配置（死区持续时间、OCx/OCxN 极性和禁止时的状态、OCxM 配置、断路使能和极性）。可以通过 TIMx_BDTR 寄存器中的 LOCK 位，从 3 种保护级别中进行选择。请参见*第 1680 页的第 41.6.13 节：TIM16/TIM17 断路和死区寄存器 (TIMx_BDTR)*。MCU 复位后只能对 LOCK 位执行一次写操作。

图 499 所示为输出对断路响应行为的示例。

图 499. 输出的断路响应行为。



41.4.14 单脉冲模式

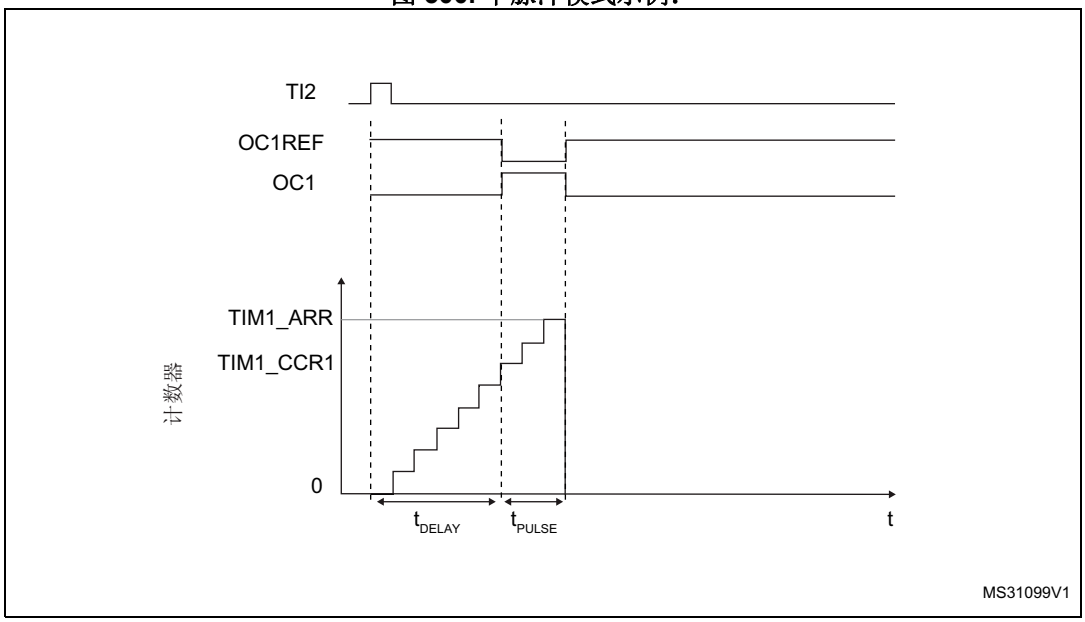
单脉冲模式 (OPM) 是上述模式的一个特例。在这种模式下，计数器可以在一个激励信号的触发下启动，并可在一段可编程的延时后产生一个脉宽可编程的脉冲。

可以通过从模式控制器启动计数器。可以在输出比较模式或 PWM 模式下生成波形。将 TIMx_CR1 寄存器中的 OPM 位置 1，即可选择单脉冲模式。这样，发生下一更新事件 UEV 时，计数器将自动停止。

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

- CNT < CCRx ≤ ARR（特别注意，0 < CCRx）

图 500. 单脉冲模式示例：



例如，用户希望达到这样的效果：在 TI2 输入引脚检测到上升沿时，经过 t_{DELAY} 的延迟，在 OC1 上产生一个长度为 t_{PULSE} 的正脉冲。

使用 TI2FP2 作为触发 1：

1. 使用 TIMx_TISEL 寄存器中的 TI2SEL[3:0] 位选择恰当的 TI2[x] 源（内部或外部）。
2. 在 TIMx_CCMR1 寄存器中写入 CC2S=“01”，以将 TI2FP2 映射到 TI2。
3. 在 TIMx_CCER 寄存器中写入 CC2P=“0”和 CC2NP=“0”，使 TI2FP2 能够检测上升沿。
4. 在 TIMx_SMCR 寄存器中写入 TS=“00110”，以将 TI2FP2 配置为从模式控制器的触发 (TRGI)。
5. 在 TIMx_SMCR 寄存器中写入 SMS =“110”（触发模式），以使用 TI2FP2 启动计数器。

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

- t_{DELAY} 由写入 TIMx_CCR1 寄存器的值定义。
- t_{PULSE} 由自动重载值与比较值之差 (TIMx_ARR - TIMx_CCR1) 来定义。
- 假设希望产生这样的波形：信号在发生比较匹配时从“0”变为“1”，在计数器达到自动重载值时由“1”变为“0”。为此，应在 TIMx_CCMR1 寄存器中写入 OC1M=111，以使能 PWM 模式 2。如果需要，可选择在 TIMx_CCMR1 寄存器的 OC1PE 和 TIMx_CR1 寄存器的 ARPE 中写入“1”，以使能预装载寄存器。这种情况下，必须在 TIMx_CCR1 寄存器中写入比较值并在 TIMx_ARR 寄存器中写入自动重载值，通过将 UG 位置 1 来产生更新，然后等待 TI2 上的外部触发事件。本例中，CC1P 的值为“0”。

由于仅需要 1 个脉冲，因此应向 TIMx_CR1 寄存器的 OPM 位写入“1”，以便在发生下一更新事件（计数器翻转从自动重载值返回到 0）时使计数器停止计数。

特殊情况：OCx 快速使能。

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

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

41.4.15 可再触发单脉冲模式 (OPM)（仅限 TIM15）

该模式可以由激励信号启动计数器，并且能产生长度可编程的脉冲，但与不可再触发单脉冲模式间存在以下差别，如*第 41.4.14 节*所述：

- 发生触发时，脉冲立即产生（无可编程延时）
- 如果在上一个触发完成前发生新的触发，脉冲将延长

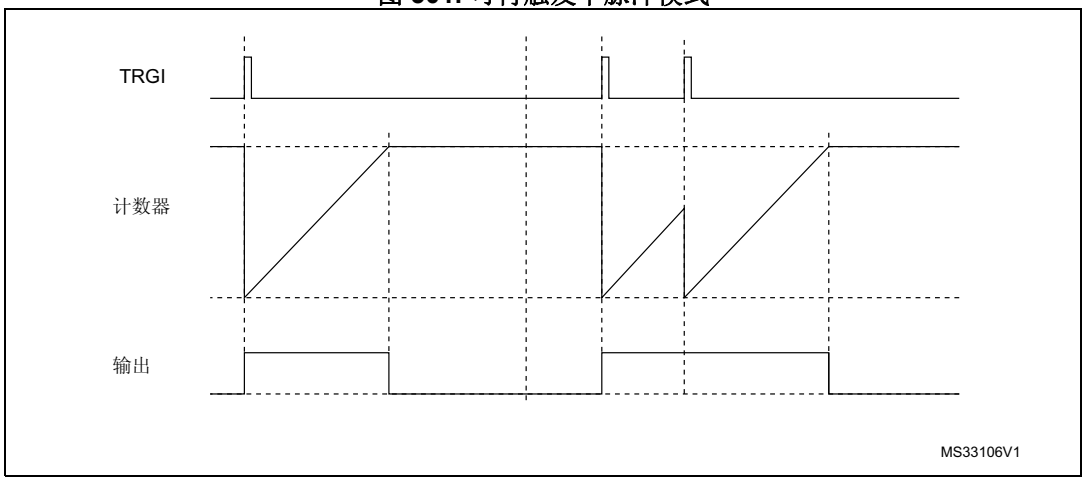
定时器必须处于从模式，TIMx_SMCR 寄存器中的位 SMS[3:0] =“1000”（组合复位 + 触发模式），针对可再触发 OPM 模式 1 或模式 2 将 OCxM[3:0] 位设置为“1000”或“1001”。

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

注： *出于兼容性原因，OCxM[3:0] 和 SMS[3:0] 位域分为两部分，最高有效位与最低有效的 3 位不相邻。*

此模式不能与中心对齐 PWM 模式一起使用。在 TIMx_CR1 中必须设置 CMS[1:0] = 00。

图 501. 可再触发单脉冲模式



41.4.16 UIF 位重映射

TIMx_CR1 寄存器中的 IUFREMAP 位强制将更新中断标志 UIF 连续复制到定时计数器寄存器的位 31 (TIMxCNT[31]) 中。这样便可自动读取计数器值以及由 UIFCPY 标志发出的电位翻转条件。在特定情况下，这可避免在后台任务（计数器读）和中断（更新中断）之间共享处理时产生竞争条件，从而简化计算。

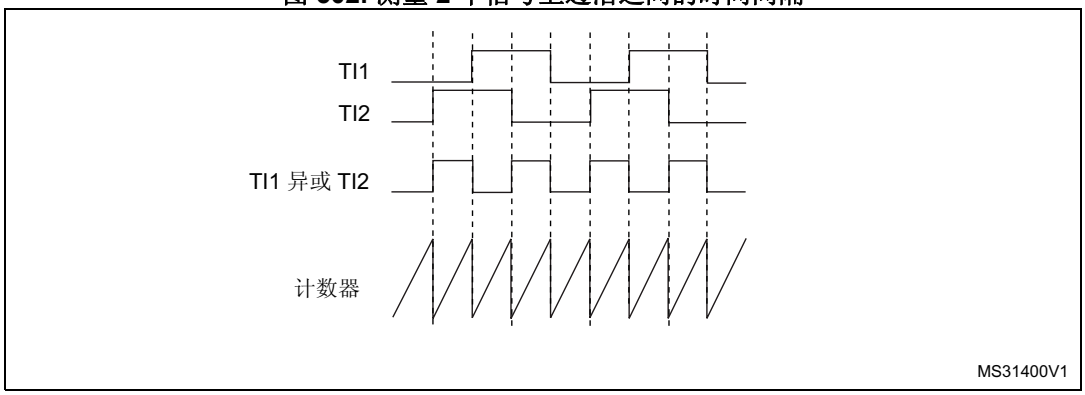
UIF 和 UIFCPY 标志使能之间没有延迟。

41.4.17 定时器输入异或功能（仅适用于 TIM15）

通过 TIMx_CR2 寄存器中的 TI1S 位，可将通道 1 的输入滤波器连接到异或门的输出，从而将 TIMx_CH1 和 TIMx_CH2 这两个输入引脚组合在一起。

异或输出可与触发或输入捕获等所有定时器输入功能配合使用。这样可用于测量两个输入信号上边沿之间的间隔，如下面的*图 502* 所示。

图 502. 测量 2 个信号上边沿之间的时间间隔



41.4.18 外部触发同步（仅适用于 TIM15）

TIM 定时器从内部链接在一起，以实现定时器同步或级联。

TIM15 定时器可与外部触发以下列模式实现同步：复位模式、门控模式和触发模式。

从模式：复位模式

当触发输入信号发生变化时，计数器及其预分频器可重新初始化。此外，如果 TIMx_CR1 寄存器中的 URS 位处于低电平，则会生成更新事件 UEV。然后，所有预装载寄存器（TIMx_ARR 和 TIMx_CCRx）都将更新。

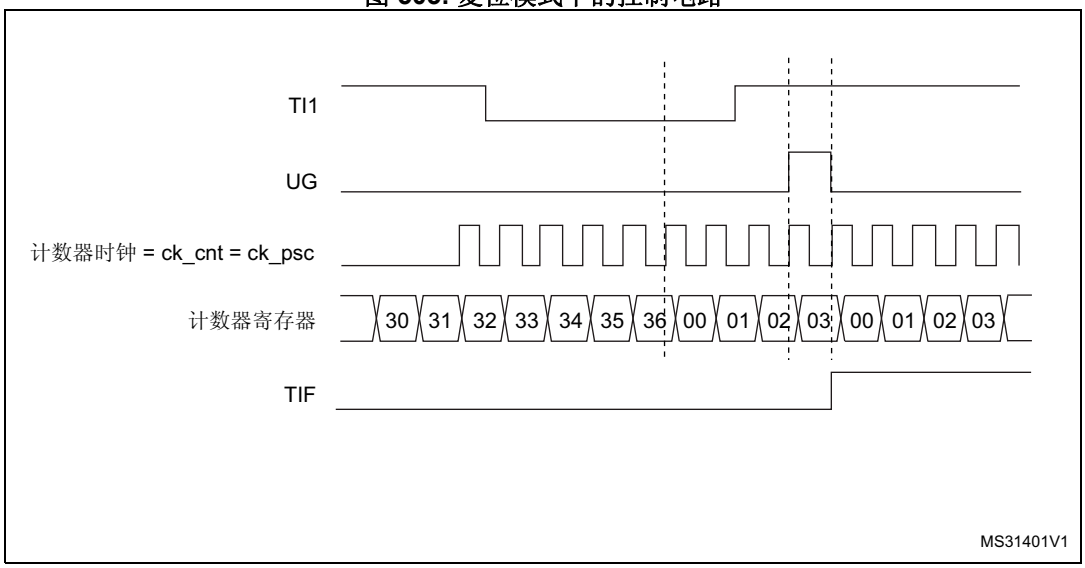
在以下示例中，TI1 输入上出现上升沿时，递增计数器清零：

1. 将通道 1 配置为检测 TI1 的上升沿。配置输入滤波带宽（本例中不需要任何滤波器，因此保持 IC1F=0000）。由于捕获预分频器不用于触发操作，因此无需对其进行配置。CC1S 位只选择输入捕获源，即 TIMx_CCMR1 寄存器中的 CC1S = 01。在 TIMx_CCER 寄存器中写入 CC1P =“0”和 CC1NP =“0”，使极性有效（仅检测上升沿）。
2. 在 TIMx_SMCR 寄存器中写入 SMS=100，将定时器配置为复位模式。在 TIMx_SMCR 寄存器中写入 TS=00101，选择 TI1 作为输入源。
3. 在 TIMx_CR1 寄存器中写入 CEN=1，启动计数器。

计数器开始根据内部时钟计数，然后正常运转，直到出现 TI1 上升沿。当 TI1 出现上升沿时，计数器清零，然后重新从 0 开始计数。同时，触发标志（TIMx_SR 寄存器中的 TIF 位）置 1，使能中断或 DMA 后，还可发送中断或 DMA 请求（取决于 TIMx_DIER 寄存器中的 TIE 和 TDE 位）。

下图显示了自动重载寄存器 TIMx_ARR=0x36 时的相关行为。TI1 的上升沿与实际计数器复位之间的延迟是由于 TI1 输入的重新同步电路引起的。

图 503. 复位模式下的控制电路



从模式：门控模式

输入信号的电平可用来使能计数器。

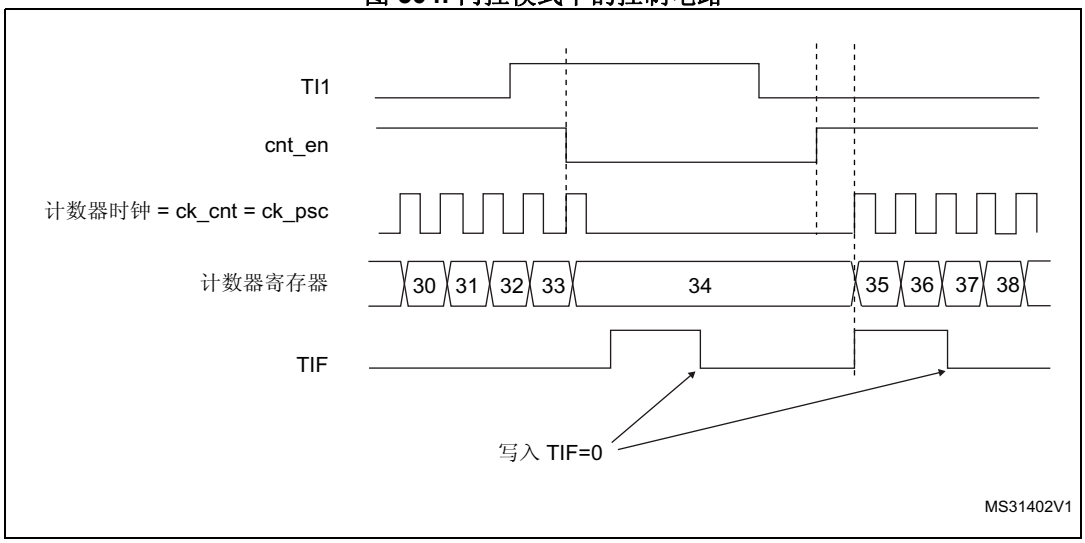
在以下示例中，递增计数器仅在 TI1 输入为低电平时计数：

1. 将通道 1 配置为检测 TI1 上的低电平。配置输入滤波带宽（本例中不需要任何滤波器，因此保持 IC1F=0000）。由于捕获预分频器不用于触发操作，因此无需对其进行配置。CC1S 位只选择输入捕获源，即 TIMx_CCMR1 寄存器中的 CC1S=01。在 TIMx_CCER 寄存器中写入 CC1P = 1 和 CC1NP = “0”，以确定极性（仅检测低电平）。
2. 在 TIMx_SMCR 寄存器中写入 SMS=101，将定时器配置为门控模式。在 TIMx_SMCR 寄存器中写入 TS=00101，选择 TI1 作为输入源。
3. 在 TIMx_CR1 寄存器中写入 CEN=1，使能计数器（在门控模式下，如果 CEN=0，则无论触发输入电平如何，计数器都不启动）。

只要 TI1 为低电平，计数器就开始根据内部时钟计数，直到 TI1 变为高电平时停止计数。计数器启动或停止时，TIMx_SR 寄存器中的 TIF 标志都会置 1。

TI1 的上升沿与实际计数器停止之间的延迟是由于 TI1 输入的重新同步电路引起的。

图 504. 门控模式下的控制电路



从模式：触发模式

所选输入上发生某一事件时可以启动计数器。

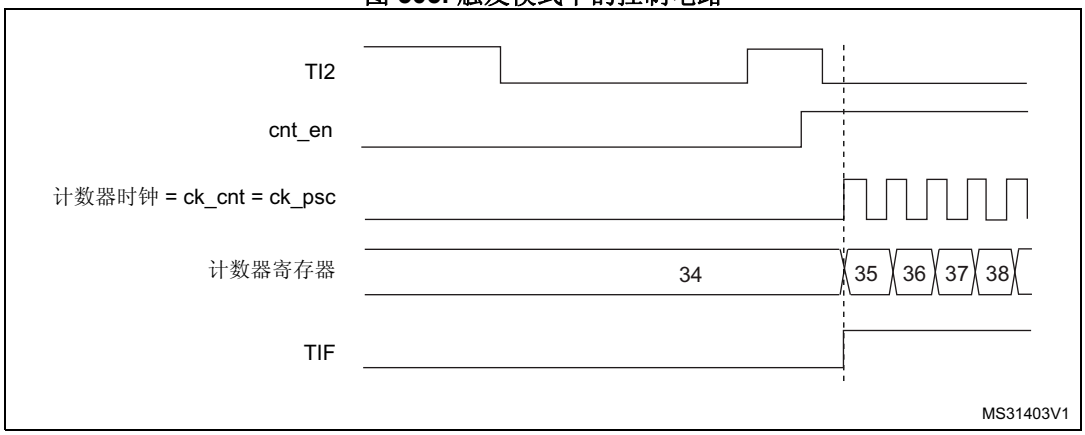
在以下示例中，TI2 输入上出现上升沿时，递增计数器启动：

1. 将通道 2 配置为检测 TI2 上的上升沿。配置输入滤波带宽（本例中不需要任何滤波器，因此保持 IC2F=0000）。由于捕获预分频器不用于触发操作，因此无需对其进行配置。CC2S 位只选择输入捕获源，即 TIMx_CCMR1 寄存器中的 CC2S=01。在 TIMx_CCER 寄存器中写入 CC2P =“1”和 CC2NP =“0”，以确定极性（仅检测低电平）。
2. 在 TIMx_SMCR 寄存器中写入 SMS = 110，将定时器配置为触发模式。通过在 TIMx_SMCR 寄存器中写入 TS=00110 来选择 TI2 作为输入源。

当 TI2 出现上升沿时，计数器开始根据内部时钟计数，并且 TIF 标志置 1。

TI2 的上升沿与实际计数器启动之间的延迟是由于 TI2 输入的重新同步电路引起的。

图 505. 触发模式下的控制电路



41.4.19 从模式——组合复位 + 触发模式

在这种情况下，在出现所选触发输入 (TRGI) 上升沿时，重新初始化计数器，生成一个寄存器更新事件，并启动计数器。

该模式用于单脉冲模式。

41.4.20 DMA 连续传送模式

TIMx 定时器能够根据一个事件生成多个 DMA 请求。主要目的是能够对几个定时器寄存器多次重新编程而无需软件开销，但也可用于定期读取一行中的多个寄存器。

DMA 控制器目标唯一，必须指向虚拟寄存器 TIMx_DMAR。发生给定的定时器事件时，定时器会启动 DMA 请求序列（突发）。每次写入 TIMx_DMAR 寄存器都会重定向到其中一个定时器寄存器。

TIMx_DCR 寄存器中的 DBL[4:0] 位设置 DMA 连续传送长度。当对 TIMx_DMAR 地址进行读或写访问时，定时器进行一次连续传送，即传送次数（按半字或字节）。

TIMx_DCR 寄存器中的 DBA[4:0] 位定义 DMA 传送的 DMA 基址（通过 TIMx_DMAR 地址执行读/写访问时）。DBA 定义为从 TIMx_CR1 寄存器地址开始计算的偏移量。

示例：

00000：TIMx_CR1，

00001：TIMx_CR2，

00010：TIMx_SMCR，

例如，定时器 DMA 连续传送功能用于在发生更新事件后将 CCRx 寄存器（x = 2、3、4）的内容更新为通过 DMA 传输到 CCRx 寄存器中的多个半字。

具体操作步骤如下：

1. 将相应的 DMA 通道配置如下：
   - DMA 通道外设地址为 DMAR 寄存器地址。
   - DMA 通道存储器地址为包含要通过 DMA 传输到 CCRx 寄存器的数据的 RAM 缓冲区地址。
   - 要传输的数据量 = 3（参见下文注释）。
   - 禁止循环模式。
2. 通过将 DBA 和 DBL 位域配置如下来配置 DCR 寄存器：
   DBL = 3 次传输，DBA = 0xE。
3. 使能 TIMx 更新 DMA 请求（DIER 寄存器中的 UDE 位置 1）。
4. 使能 TIMx。
5. 使能 DMA 通道。

本例适用于每个 CCRx 寄存器只更新一次的情况。如果每个 CCRx 寄存器要更新两次，则要传输的数据量应为 6。下面以包含 data1、data2、data3、data4、data5 和 data6 的 RAM 缓冲区为例。数据将按照如下方式传输到 CCRx 寄存器：在第一个更新 DMA 请求期间，data1 传输到 CCR2，data2 传输到 CCR3，data3 传输到 CCR4；在第二个更新 DMA 请求期间，data4 传输到 CCR2，data5 传输到 CCR3，data6 传输到 CCR4。

注： *可以将空值写入保留的寄存器中。*

41.4.21 定时器同步 (TIM15)

TIMx 定时器从内部连接在一起，以实现定时器同步或链接。有关详细信息，请参见 *第 41.4.21 节：定时器同步 (TIM15)*。

注： *必须先使能从定时器的时钟，才能从主定时器接收事件；从主定时器接收触发信号时，不得运行中更改从定时器的时钟。*

41.4.22 调试模式

当微控制器进入调试模式（Cortex®-M7 with FPU 内核停止）时，TIMx 计数器会根据 DBGMCU 模块中的 TIMx 位选择继续正常工作或者停止工作。有关详细信息，请参见 *第 60 节：调试基础结构*。

为了安全起见，当计数器停止 (DBGMCU_APB2FZ1 中的 TIMx = 1) 时，输出被禁止（就像 MOE 位被复位一样）。可以将输出强制变为未激活状态（OSSI 位 = 1），或者通过 GPIO 控制器（OSSI 位 = 0）来控制输出，以将其强制为高阻态。

41.5 TIM15寄存器

有关寄存器说明中使用的缩写，请参见*第 1.1 节*。

41.5.1 TIM15 控制寄存器 1 (TIM15_CR1)

TIM15 control register 1

偏移地址：0x00

复位值：0x0000

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

位 15:12 保留，必须保持复位值。

位 11 **UIFREMAP**：UIF 状态位重映射 (UIF status bit remapping)
0：无重映射。UIF 状态位不复制到 TIMx_CNT 寄存器的位 31。
1：使能重映射。UIF 状态位复制到 TIMx_CNT 寄存器的位 31。

位 10 保留，必须保持复位值。

位 9:8 **CKD[1:0]**：时钟分频 (Clock division)
此位域指示定时器时钟 (CK_INT) 频率与死区发生器以及数字滤波器 (TIx) 所使用的死区及采样时钟 (t_{DTS}) 之间的分频比，
00：$t_{DTS} = t_{CK_INT}$
01：$t_{DTS} = 2*t_{CK_INT}$
10：$t_{DTS} = 4*t_{CK_INT}$
11：保留，不要设置成此值

位 7 **ARPE**：自动重载预装载使能 (Auto-reload preload enable)
0：TIMx_ARR 寄存器不进行缓冲
1：TIMx_ARR 寄存器进行缓冲

位 6:4 保留，必须保持复位值。

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

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

位 1 **UDIS**：更新禁止 (Update disable)

此位由软件置 1 和清零，用以使能/禁止 UEV 事件生成。

0：使能 UEV。更新 (UEV) 事件可通过以下事件之一生成：

– 计数器上溢/下溢

– 将 UG 位置 1

– 通过从模式控制器生成的更新事件

然后更新影子寄存器的值。

1：禁止 UEV。不会生成更新事件，各影子寄存器的值（ARR、PSC 和 CCRx）保持不变。但如果将 UG 位置 1，或者从从模式控制器接收到硬件复位，则会重新初始化计数器和预分频器。

位 0 **CEN**：计数器使能 (Counter enable)

0：禁止计数器

1：使能计数器

注： 只有事先通过软件将 CEN 位置 1，才可以使用外部时钟和门控模式。而触发模式可通过硬件自动将 CEN 位置 1。

41.5.2 TIM15 控制寄存器 2 (TIM15_CR2)

TIM15 control register 2

偏移地址：0x04

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	OIS2	OIS1N	OIS1	TI1S	MMS[2:0]			CCDS	CCUS	Res.	CCPC
					rw	rw	rw	rw	rw	rw	rw	rw	rw		rw

位 15:11 保留，必须保持复位值。

位 10 **OIS2**：输出空闲状态 2（OC2 输出）(Output Idle state 2 (OC2 output))

0：当 MOE = 0 时，OC2 = 0

1：当 MOE = 0 时，OC2 = 1

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

位 9 **OIS1N**：输出空闲状态 1（OC1N 输出）(Output Idle state 1 (OC1N output))

0：当 MOE=0 时，经过死区时间后 OC1N=0

1：当 MOE=0 时，经过死区时间后 OC1N=1

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

位 8 **OIS1**：输出空闲状态 1（OC1 输出）(Output Idle state 1 (OC1 output))

0：当 MOE=0 时，（如果 OC1N 有效，则经过死区时间之后）OC1=0

1：当 MOE=1 时，（如果 OC1N 有效，则经过死区时间之后）OC1=0

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

位 7 **TI1S**：TI1 选择 (TI1 selection)

0：TIMx_CH1 引脚连接到 TI1 输入

1：TIMx_CH1、CH2 引脚连接到 TI1 输入（异或组合）

位 6:4 **MMS[1:0]**：主模式选择 (Master mode selection)

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

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

001：**使能**——计数器使能信号 CNT_EN 用作触发输出 (TRGO)。该触发输出可用于同时启动多个定时器，或者控制在一段时间内使能从定时器。计数器使能信号由 CEN 控制位与门控模式下的触发输入的逻辑或运算组合而成。当计数器使能信号由触发输入控制时，TRGO 上会存在延迟，选择主/从模式时除外（请参见 TIMx_SMCR 寄存器中 MSM 位的说明）。

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

011：**比较脉冲**——一旦发生输入捕获或比较匹配事件，当 CC1IF 标志被置 1 时（即使已为高电平），触发输出都会发送一个正脉冲。(TRGO)。

100：**比较**——OC1REF 信号用作触发输出 (TRGO)。

101：**比较**——OC2REF 信号用作触发输出 (TRGO)。

位 3 **CCDS**：捕获/比较 DMA 选择 (Capture/compare DMA selection)

0：发生 CCx 事件时发送 CCx DMA 请求

1：发生更新事件时发送 CCx DMA 请求

位 2 **CCUS**：捕获/比较控制更新选择 (Capture/compare control update selection)

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

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

注： 此位仅对具有互补输出的通道有效。

位 1 保留，必须保持复位值。

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

0：CCxE、CCxNE 和 OCxM 位未进行预装载

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

注： 此位仅对具有互补输出的通道有效。

41.5.3 TIM15 从模式控制寄存器 (TIM15_SMCR)

TIM15 slave mode control register

偏移地址：0x08

复位值：0x0000 0000

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

位 31:22 保留，必须保持复位值。

位 21:20 **TS[4:3]**：触发选择——位 4:3 (Trigger selection - bit 4:3)
请参见 TS[4:0] 说明——位 6:4。

位 19:17 保留，必须保持复位值。

位 16 **SMS[3]**：从模式选择——位 3 (Slave mode selection - bit 3)
请参见 SMS 说明——位 2:0

位 15:8 保留，必须保持复位值。

位 7 **MSM**：主/从模式 (Master/slave mode)
0：不执行任何操作
1：当前定时器的触发输入事件 (TRGI) 的动作被推迟，以使当前定时器与其从定时器实现完美同步（通过 TRGO）。此设置适用于由单个外部事件对多个定时器进行同步的情况。

位 6:4 **TS[4:0]**：触发选择 (Trigger selection)
此位域可选择将要用于同步计数器的触发输入。
00000：内部触发 0 (ITR0)
00001：内部触发 1 (ITR1)
00010：内部触发 2 (ITR2)
00011：内部触发 3 (ITR3)
00100：TI1 边沿检测器 (TI1F_ED)
00101：滤波后的定时器输入 1 (TI1FP1)
00110：滤波后的定时器输入 2 (TI2FP2)
其它：保留
有关各定时器 ITRx 含义的详细信息，请参见*第 1648 页的表 325：TIMx 内部触发连接*。
注：　这些位只能在未使用的情况下（例如，SMS=000 时）进行更改，以避免转换时出现错误的边沿检测。

位 3 保留，必须保持复位值。

位 2:0 **SMS**：从模式选择 (Slave mode selection)

选择外部信号时，触发信号 (TRGI) 的有效边沿与外部输入上所选的极性相关（请参见输入控制寄存器和控制寄存器说明）。

0000：禁止从模式——如果 CEN = “1”，预分频器时钟直接由内部时钟提供。
0001：保留
0010：保留
0011：保留
0100：复位模式——在出现所选触发输入 (TRGI) 上升沿时，重新初始化计数器并生成一个寄存器更新事件。
0101：门控模式——触发输入 (TRGI) 为高电平时使能计数器时钟。只要触发输入变为低电平，计数器立即停止计数（但不复位）。计数器的启动和停止都被控制。
0110：触发模式——触发信号 TRGI 出现上升沿时启动计数器（但不复位）。只控制计数器的启动。
0111：外部时钟模式 1——由所选触发信号 (TRGI) 的上升沿提供计数器时钟。
1000：组合复位 + 触发模式——在出现所选触发输入 (TRGI) 上升沿时，重新初始化计数器，生成一个寄存器更新事件并启动计数器。
其他代码：保留。

注：　如果将 TI1F_ED 选作触发输入（TS= “00100”），则不得使用门控模式。实际上，TI1F 每次转换时，TI1F_ED 都输出 1 个脉冲，而门控模式检查的则是触发信号的电平。

注：　必须先使能从定时器的时钟，才能从主定时器接收事件；从主定时器接收触发信号时，不得实时更改从定时器的时钟。

表 325. TIMx 内部触发连接

从 TIM	ITR0 (TS = 00000)	ITR1 (TS = 00001)	ITR2 (TS = 00010)	ITR3 (TS = 00011)
TIM15	TIM1	TIM3	TIM16, OC1	TIM17 OC1

41.5.4 TIM15 DMA/中断使能寄存器 (TIM15_DIER)

TIM15 DMA/interrupt enable register

偏移地址：0x0C

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	TDE	COMDE	Res.	Res.	CC2DE	CC1DE	UDE	BIE	TIE	COMIE	Res.	Res.	CC2IE	CC1IE	UIE
	rw	rw			rw	rw	rw	rw	rw	rw			rw	rw	rw

位 15 保留，必须保持复位值。

位 14 **TDE**：触发 DMA 请求使能 (Trigger DMA request enable)
0：禁止触发 DMA 请求
1：使能触发 DMA 请求

位 13 **COMDE**：COM DMA 请求使能 (COM DMA request enable)
0：禁止 COM DMA 请求
1：使能 COM DMA 请求

位 12:11 保留，必须保持复位值。

位 10 **CC2DE**：捕获/比较 2 DMA 请求使能 (Capture/Compare 2 DMA request enable)
0：禁止 CC2 DMA 请求
1：使能 CC2 DMA 请求

位 9 **CC1DE**：捕获/比较 1 DMA 请求使能 (Capture/Compare 1 DMA request enable)
0：禁止 CC1 DMA 请求
1：使能 CC1 DMA 请求

位 8 **UDE**：更新 DMA 请求使能 (Update DMA request enable)
0：禁止更新 DMA 请求
1：使能更新 DMA 请求

位 7 **BIE**：断路中断使能 (Break interrupt enable)
0：禁止断路中断
1：使能断路中断

位 6 **TIE**：触发中断使能 (Trigger interrupt enable)
0：禁止触发中断
1：使能触发中断

位 5 **COMIE**：COM 中断使能 (COM interrupt enable)
0：禁止 COM 中断
1：使能 COM 中断

位 4:3 保留，必须保持复位值。

位 2 **CC2IE**：捕获/比较 2 中断使能 (Capture/Compare 2 interrupt enable)
0：禁止 CC2 中断
1：使能 CC2 中断

位 1 **CC1IE**：捕获/比较 1 中断使能 (Capture/Compare 1 interrupt enable)
0：禁止 CC1 中断
1：使能 CC1 中断

位 0 **UIE**：更新中断使能 (Update interrupt enable)
0：禁止更新中断
1：使能更新中断

41.5.5 TIM15 状态寄存器 (TIM15_SR)

TIM15 status register

偏移地址：0x10

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	CC2OF	CC1OF	Res.	BIF	TIF	COMIF	Res.	Res.	CC2IF	CC1IF	UIF
					rc_w0	rc_w0		rc_w0	rc_w0	rc_w0			rc_w0	rc_w0	rc_w0

位 15:11 保留，必须保持复位值。

位 10 **CC2OF**：捕获/比较 2 重复捕获标志 (Capture/compare 2 overcapture flag)
请参见 CC1OF 说明

位 9 **CC1OF**：捕获/比较 1 重复捕获标志 (Capture/Compare 1 overcapture flag)
仅当将相应通道配置为输入捕获模式时，此标志位才会由硬件置 1。通过软件写入“0”可将该位清零。
0：未检测到重复捕获。
1：TIMx_CCR1 寄存器中已捕获到计数器值且 CC1IF 标志已置 1。

位 8 保留，必须保持复位值。

位 7 **BIF**：断路中断标志 (Break interrupt flag)
只要断路输入变为有效状态，此标志便由硬件置 1。断路输入无效后可通过软件对其清零。
0：未发生断路事件。
1：在断路输入上检测到有效电平。

位 6 **TIF**：触发中断标志 (Trigger interrupt flag)
该标志在发生触发事件时由硬件置 1（在除门控模式以外的所有模式下，当使能从模式控制器后在 TRGI 输入上检测到有效边沿时，在门控模式情况下选择两个边沿）。选择门控模式时，该标志将在计数器启动或停止时置 1。但需要通过软件清零。
0：未发生触发事件。
1：触发中断挂起。

位 5 **COMIF**：COM 中断标志 (COM interrupt flag)
此标志在发生一个 COM 事件时（一旦捕获/比较控制位——CCxE、CCxNE、OCxM——已更新）由硬件置 1。但需要通过软件清零。
0：未发生 COM 事件。
1：COM 中断挂起。

位 4:3 保留，必须保持复位值。

位 2 **CC2IF**：捕获/比较 2 中断标志 (Capture/Compare 2 interrupt flag)
请参见 CC1IF 说明

位 1 **CC1IF**：捕获/比较 1 中断标志 (Capture/Compare 1 interrupt flag)
如果通道 CC1 配置为输出：当计数器与比较值匹配时，此标志由硬件置 1。但需要通过软件清零。
0：不匹配。
1：TIMx_CNT 计数器的值与 TIMx_CCR1 寄存器的值匹配。当 TIMx_CCR1 的值大于 TIMx_ARR 的值时，CC1IF 位将在计数器发生上溢时变为高电平。
如果通道 CC1 配置为输入：此位将在发生捕获事件时由硬件置 1。通过软件或读取 TIMx_CCR1 寄存器将该位清零。
0：未发生输入捕获事件
1：TIMx_CCR1 寄存器中已捕获到计数器值（IC1 上已检测到与所选极性匹配的边沿）

位 0 **UIF**：更新中断标志 (Update interrupt flag)
该位在发生更新事件时通过硬件置 1。但需要通过软件清零。
0：未发生更新。
1：更新中断挂起。该位在以下情况下更新寄存器时由硬件置 1：
– TIMx_CR1 寄存器中的 UDIS = 0，当重复计数器的值上溢时（重复计数器 = 0 时更新）。
– TIMx_CR1 寄存器中的 URS = 0 且 UDIS = 0，并且由软件使用 TIMx_EGR 寄存器中的 UG 位重新初始化 CNT 时。
– TIMx_CR1 寄存器中的 URS=0 且 UDIS=0，并且 CNT 由触发事件重新初始化时（请参见 *第 41.5.3 节：TIM15 从模式控制寄存器 (TIM15_SMCR)*）。

41.5.6 TIM15 事件产生寄存器 (TIM15_EGR)

TIM15 event generation register

偏移地址：0x14

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BG	TG	COMG	Res.	Res.	CC2G	CC1G	UG
								w	w	rw			w	w	w

位 15:8 保留，必须保持复位值。

位 7 **BG**：断路生成 (Break generation)
此位由软件置 1 以生成事件，并由硬件自动清零。
0：不执行任何操作
1：生成断路事件。MOE 位清零且 BIF 标志置 1。使能后可发生相关中断或 DMA 传输事件。

位 6 **TG**：触发生成 (Trigger generation)
此位由软件置 1 以生成事件，并由硬件自动清零。
0：不执行任何操作
1：TIMx_SR 寄存器中的 TIF 标志置 1。使能后可发生相关中断或 DMA 传输事件。

位 5 **COMG**：捕获/比较控制更新生成 (Capture/Compare control update generation)
该位可通过软件置 1，并由硬件自动清零。
0：不执行任何操作
1：CCPC 位置 1 时，可更新 CCxE、CCxNE 和 OCxM 位
注： 此位仅对具有互补输出的通道有效。

位 4:3 保留，必须保持复位值。

位 2 **CC2G**：捕获/比较 2 生成 (Capture/compare 2 generation)
请参见 CC1G 说明

位 1 **CC1G**：捕获/比较 1 生成 (Capture/Compare 1 generation)
此位由软件置 1 以生成事件，并由硬件自动清零。
0：不执行任何操作
1：通道 1 上生成捕获/比较事件：
如果通道 CC1 配置为输出：
使能时，CC1IF 标志置 1 并发送相应的中断或 DMA 请求。
如果通道 CC1 配置为输入：
TIMx_CCR1 寄存器中将捕获到计数器当前值。使能时，CC1IF 标志置 1 并发送相应的中断或 DMA 请求。如果 CC1IF 标志已为高电平，CC1OF 标志将置 1。

位 0 **UG**：更新生成 (Update generation)
该位可通过软件置 1，并由硬件自动清零。
0：不执行任何操作。
1：重新初始化计数器并生成寄存器更新事件。请注意，预分频器计数器也将清零（但预分频比不受影响）。

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

TIM15 capture/compare mode register 1

偏移地址：0x18

复位值：0x0000 0000

这些通道可用于输入（捕获模式）或输出（比较模式）模式。通道方向通过配置相应的 CCxS 位进行定义。此寄存器的所有其它位在输入模式和输出模式下的功能均不同。对于任一给定位，OCxx 用于说明通道配置为输出时该位对应的功能，ICxx 则用于说明通道配置为输入时该位对应的功能。因此，必须注意同一个位在输入阶段和输出阶段具有不同的含义。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC2M [3]	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC1M [3]
							Res.								Res.
							rw								rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	OC2M[2:0]			OC2 PE	OC2 FE	CC2S[1:0]		Res.	OC1M[2:0]			OC1 PE	OC1 FE	CC1S[1:0]	
IC2F[3:0]				IC2PSC[1:0]				IC1F[3:0]				IC1PSC[1:0]			
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

输出比较模式：

位 31:25 保留，始终读为 0

位 24 **OC2M[3]**：输出比较 2 模式——位 3 (Output Compare 2 mode - bit 3)

位 23:17 保留，始终读为 0

位 16 **OC1M[3]**：输出比较 1 模式——位 3 (Output Compare 1 mode - bit 3)
请参见 OC1M 说明——位 6:4

位 15 保留，始终读为 0

位 14:12 **OC2M[2:0]**：输出比较 2 模式 (Output Compare 2 mode)

位 11 **OC2PE**：输出比较 2 预装载使能 (Output Compare 2 preload enable)

位 10 **OC2FE**：输出比较 2 快速使能 (Output Compare 2 fast enable)

位 9:8 **CC2S[1:0]**：捕获/比较 2 选择 (Capture/Compare 2 selection)
此位域定义通道方向（输入/输出）以及所使用的输入。
00：CC2 通道配置为输出。
01：CC2 通道配置为输入，IC2 映射到 TI2 上。
10：CC2 通道配置为输入，IC2 映射到 TI1 上。
11：CC2 通道配置为输入，IC2 映射到 TRC 上。此模式仅在通过 TS 位（TIMx_SMCR 寄存器）选择内部触发输入时有效
注： 仅当通道关闭时（TIMx_CCER 中的 CC2E = “0”），才可向 CC2S 位写入数据。

位 7 保留，始终读为 0

位 6:4 **OC1M**：输出比较 1 模式 (Output Compare 1 mode)

这些位定义提供 OC1 和 OC1N 的输出参考信号 OC1REF 的行为。OC1REF 为高电平有效，而 OC1 和 OC1N 的有效电平则取决于 CC1P 位和 CC1NP 位。

0000：冻结——输出比较寄存器 TIMx_CCR1 与计数器 TIMx_CNT 进行的比较不会对输出造成任何影响。

0001：将通道 1 设置为匹配时输出有效电平。当计数器 TIMx_CNT 与捕获/比较寄存器 1 (TIMx_CCR1) 匹配时，OC1REF 信号强制变为高电平。

0010：将通道 1 设置为匹配时输出无效电平。当计数器 TIMx_CNT 与捕获/比较寄存器 1 (TIMx_CCR1) 匹配时，OC1REF 信号强制变为低电平。

0011：翻转——TIMx_CNT=TIMx_CCR1 时，OC1REF 发生翻转。

0100：强制变为无效电平——OC1REF 强制变为低电平。

0101：强制变为有效电平——OC1REF 强制变为高电平。

0110：PWM 模式 1——只要 TIMx_CNT<TIMx_CCR1，通道 1 便为有效状态，否则为无效状态。

0111：PWM 模式 2——只要 TIMx_CNT<TIMx_CCR1，通道 1 便为无效状态，否则为有效状态。

1000：可再触发 OPM 模式 1——在递增计数模式下，通道为有效状态，直至（在 TRGI 信号上）检测到触发事件。然后，在 PWM 模式 1 下进行比较，通道会在下一次更新时再次变为有效状态。在递减计数模式下，通道为无效状态，直至（在 TRGI 信号上）检测到触发事件。然后，在 PWM 模式 1 下进行比较，通道会在下一次更新时再次变为无效状态。

1001：可再触发 OPM 模式 2——在递增计数模式下，通道为无效状态，直至（在 TRGI 信号上）检测到触发事件。然后，在 PWM 模式 2 下进行比较，通道会在下一次更新时再次变为无效状态。在递减计数模式下，通道为有效状态，直至（在 TRGI 信号上）检测到触发事件。然后，在 PWM 模式 1 下进行比较，通道会在下一次更新时再次变为有效状态。

1010：保留

1011：保留

1100：组合 PWM 模式 1——OC1REF 与在 PWM 模式 1 下的行为相同。OC1REFC 是 OC1REF 和 OC2REF 的逻辑或运算结果。

1101：组合 PWM 模式 2——OC1REF 与在 PWM 模式 2 下的行为相同。OC1REFC 是 OC1REF 和 OC2REF 的逻辑与运算结果。

1110：保留

1111：保留

注： ***1****：只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 3 且 CC1S=“00”（通道配置为输出），这些位即无法修改。*

2*：在 PWM 模式下，仅当比较结果发生改变或输出比较模式由“冻结”模式切换到“PWM”模式时，OCREF 电平才会发生更改。*

3*：此位域将在具有互补输出的通道上进行预装载。TIMx_CR2 寄存器中的 CCPC 位置 1，则仅当生成 COM 事件时，OC1M 有效位才会从预装载位获取新值。*

位 3 **OC1PE**：输出比较 1 预装载使能 (Output Compare 1 preload enable)

0：禁止与 TIMx_CCR1 相关的预装载寄存器。可随时向 TIMx_CCR1 写入数据，写入后将立即使用新值。

1：使能与 TIMx_CCR1 相关的预装载寄存器。可读/写访问预装载寄存器。TIMx_CCR1 预装载值在每次生成更新事件时都会装载到活动寄存器中。

*注： **1**：只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 3 且 CC1S=“00”（通道配置为输出），这些位即无法修改。*

***2**：只有单脉冲模式下才可在未验证预装载寄存器的情况下使用 PWM 模式（TIMx_CR1 寄存器中的 OPM 位置 1）。其它情况下则无法保证该行为。*

位 2 **OC1FE**：输出比较 1 快速使能 (Output Compare 1 fast enable)

此位用于加快触发输入事件对 CC 输出的影响。

0：即使触发开启，CC1 也将根据计数器和 CCR1 值正常工作。触发输入出现边沿时，激活 CC1 输出的最短延迟时间为 5 个时钟周期。

1：触发输入上出现有效边沿相当于 CC1 输出上的比较匹配。随后，无论比较结果如何，OC 都设置为比较电平。采样触发输入和激活 CC1 输出的延迟时间缩短为 3 个时钟周期。仅当通道配置为 PWM1 或 PWM2 模式时，OCFE 才会起作用。

位 1:0 **CC1S**：捕获/比较 1 选择 (Capture/Compare 1 Selection)

此位域定义通道方向（输入/输出）以及所使用的输入。

00：CC1 通道配置为输出。

01：CC1 通道配置为输入，IC1 映射到 TI1 上。

10：CC1 通道配置为输入，IC1 映射到 TI2 上。

11：CC1 通道配置为输入，IC1 映射到 TRC 上。此模式仅在通过 TS 位（TIMx_SMCR 寄存器）选择内部触发输入时有效

注： 仅当通道关闭时（TIMx_CCER 中的 CC1E = “0”），才可向 CC1S 位写入数据。

输入捕获模式

位 31:16 保留，始终读为 0

位 15:12 **IC2F**：输入捕获 2 滤波器 (Input capture 2 filter)

位 11:10 **IC2PSC[1:0]**：输入捕获 2 预分频器 (Input capture 2 prescaler)

位 9:8 **CC2S**：捕获/比较 2 选择 (Capture/Compare 2 selection)

此位域定义通道方向（输入/输出）以及所使用的输入。

00：CC2 通道配置为输出

01：CC2 通道配置为输入，IC2 映射到 TI2 上

10：CC2 通道配置为输入，IC2 映射到 TI1 上

11：CC2 通道配置为输入，IC2 映射到 TRC 上。此模式仅在通过 TS 位（TIMx_SMCR 寄存器）选择内部触发输入时有效

注： 仅当通道关闭时（TIMx_CCER 中的 CC2E = “0”），才可向 CC2S 位写入数据。

位 7:4 **IC1F[3:0]**：输入捕获 1 滤波器 (Input capture 1 filter)

此位域可定义 TI1 输入的采样频率和适用于 TI1 的数字滤波器带宽。数字滤波器由事件计数器组成，每 N 个连续事件才视为一个有效输出变化：

0000：无滤波器，按 f_{DTS} 频率进行采样
0001：$f_{SAMPLING}=f_{CK_INT}$，N=2
0010：$f_{SAMPLING}=f_{CK_INT}$，N=4
0011：$f_{SAMPLING}=f_{CK_INT}$，N=8
0100：$f_{SAMPLING}=f_{DTS}/2$，N=6
0101：$f_{SAMPLING}=f_{DTS}/2$，N=8
0110：$f_{SAMPLING}=f_{DTS}/4$，N=6
0111：$f_{SAMPLING}=f_{DTS}/4$，N=8
1000：$f_{SAMPLING}=f_{DTS}/8$，N=6
1001：$f_{SAMPLING}=f_{DTS}/8$，N=8
1010：$f_{SAMPLING}=f_{DTS}/16$，N=5
1011：$f_{SAMPLING}=f_{DTS}/16$，N=6
1100：$f_{SAMPLING}=f_{DTS}/16$，N=8
1101：$f_{SAMPLING}=f_{DTS}/32$，N=5
1110：$f_{SAMPLING}=f_{DTS}/32$，N=6
1111：$f_{SAMPLING}=f_{DTS}/32$，N=8

位 3:2 **IC1PSC**：输入捕获 1 预分频器 (Input capture 1 prescaler)

此位域定义 CC1 输入 (IC1) 的预分频比。只要 CC1E=“0”（TIMx_CCER 寄存器），预分频器便立即复位。

00：无预分频器，捕获输入上每检测到一个边沿便执行捕获
01：每发生 2 个事件便执行一次捕获
10：每发生 4 个事件便执行一次捕获
11：每发生 8 个事件便执行一次捕获

位 1:0 **CC1S**：捕获/比较 1 选择 (Capture/Compare 1 Selection)

此位域定义通道方向（输入/输出）以及所使用的输入。

00：CC1 通道配置为输出
01：CC1 通道配置为输入，IC1 映射到 TI1 上
10：CC1 通道配置为输入，IC1 映射到 TI2 上
11：CC1 通道配置为输入，IC1 映射到 TRC 上。此模式仅在通过 TS 位（TIMx_SMCR 寄存器）选择内部触发输入时有效

注： 仅当通道关闭时（TIMx_CCER 中的 CC1E = “0”），才可向 CC1S 位写入数据。

41.5.8 TIM15 捕获/比较使能寄存器 (TIM15_CCER)

TIM15 capture/compare enable register

偏移地址：0x20

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CC2NP	Res.	CC2P	CC2E	CC1NP	CC1NE	CC1P	CC1E
								rw		rw	rw	rw	rw	rw	rw

位 15:8 保留，必须保持复位值。

位 7 **CC2NP**：捕获/比较 2 互补输出极性 (Capture/Compare 2 complementary output polarity)
请参见 CC1NP 说明

位 6 保留，必须保持复位值。

位 5 **CC2P**：捕获/比较 2 输出极性 (Capture/Compare 2 output polarity)
请参见 CC1P 说明

位 4 **CC2E**：捕获/比较 2 输出使能 (Capture/Compare 2 output enable)
请参见 CC1E 说明

位 3 **CC1NP**：捕获/比较 1 互补输出极性 (Capture/Compare 1 complementary output polarity)
CC1 通道配置为输出：
0：OC1N 高电平有效。
1：OC1N 低电平有效。
CC1 通道配置为输入：
此位与 CC1P 配合使用，用以定义 TI1FP1 和 TI2FP1 的极性。请参见 CC1P 说明。

注： *1. 只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 2 或 3 且 CC1S = “00”（通道配置为输出），此位立即变为不可写状态。*
2. 在那些互补输出的通道上，此位已经被预装载。如果 TIMx_CR2 寄存器中的 CCPC 位置 1，则仅当生成换向事件时，CC1NP 有效位才会从预装载位获取新值。

位 2 **CC1NE**：捕获/比较 1 互补输出使能 (Capture/Compare 1 complementary output enable)
0：关闭——OC1N 未激活。OC1N 电平是 MOE、OSSI、OSSR、OIS1、OIS1N 和 CC1E 位的函数。
1：开启——在相应输出引脚上输出 OC1N 信号，具体取决于 MOE、OSSI、OSSR、OIS1、OIS1N 和 CC1E 位。

位 1 **CC1P**：捕获/比较 1 输出极性 (Capture/Compare 1 output polarity)
CC1 通道配置为输出：
0：OC1 高电平有效
1：OC1 低电平有效
CC1 通道配置为输入：CC1NP/CC1P 位可针对触发或捕获操作选择 TI1FP1 和 TI2FP1 的极性。
00：非反相/上升沿触发。电路对 TIxFP1 上升沿敏感（在复位模式、外部时钟模式或触发模式下执行捕获或触发操作），TIxFP1 未反相（在门控模式下执行触发操作）。
01：反相/下降沿触发。电路对 TIxFP1 下降沿敏感（在复位模式、外部时钟模式或触发模式下执行捕获或触发操作），TIxFP1 反相（在门控模式下执行触发操作）。
10：保留，不使用此配置。
11：未反相/边沿触发。电路对 TIxFP1 上升沿和下降沿都敏感（在复位模式、外部时钟模式或触发模式下执行捕获或触发操作），TIxFP1 未反相（在门控模式下执行触发操作）。

注： *1. 只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 2 或 3，此位立即变为不可写状态。*
2. 此位将在具有互补输出的通道上进行预装载。如果 TIMx_CR2 寄存器中的 CCPC 位置 1，则仅当生成换向事件时，CC1P 有效位才会从预装载位获取新值。

位 0 **CC1E**：捕获/比较 1 输出使能 (Capture/Compare 1 output enable)
CC1 通道配置为输出：
0：关闭——OC1 未激活。OC1 电平是 MOE、OSSI、OSSR、OIS1、OIS1N 和 CC1NE 位的函数。
1：开启——OC1 信号输出到相应的输出引脚上，具体取决于 MOE、OSSI、OSSR、OIS1、OIS1N 和 CC1NE 位。
CC1 通道配置为输入：此位决定了是否可以实际将计数器值捕获到输入捕获/比较寄存器 1 (TIMx_CCR1) 中。
0：禁止捕获
1：使能捕获

表 326. 具有断路功能的互补通道 OCx 和 OCxN 的输出控制位 TIM15

控制位					输出状态(1)	
MOE 位	OSSI 位	OSSR 位	CCxE 位	CCxNE 位	OCx 输出状态	OCxN 输出状态
1	X	X	0	0	禁止输出（不由定时器驱动：高阻态） OCx=0 OCxN=0、OCxN_EN=0	
		0	0	1	禁止输出（不由定时器驱动：高阻态） OCx=0	OCxREF + 极性 OCxN = OCxREF 异或 CCxNP
		0	1	0	OCxREF + 极性 OCx=OCxREF 异或 CCxP	禁止输出（不由定时器驱动：高阻态） OCxN=0
		X	1	1	OCREF + 极性 + 死区	OCREF 互补项（对 OCREF 进行"非"运算）+ 极性 + 死区
		1	0	1	关闭状态（输出使能为无效状态） OCx=CCxP	OCxREF + 极性 OCxN = OCxREF 异或 CCxNP
		1	1	0	OCxREF + 极性 OCx=OCxREF 异或 CCxP、 OCx_EN=1	关闭状态 （输出使能为无效状态） OCxN=CCxNP、OCxN_EN=1
0	0	X	X	X	禁止输出（不再由定时器驱动）。输出状态由 GPIO 控制器定义，可以是高电平、低电平或高阻态。	
	1		0	0		
			0	1	关闭状态（输出使能为无效状态） 异步：OCx = CCxP、OCxN = CCxNP 那么如果时钟存在：在死区后 Ocx = OISx 且 OCxN = OISxN，从而假定 OISx 和 OISxN 在有效状态下与 OCX 和 OCxN 不对应	
			1	0		
			1	1		

1. 如果一个通道的两个输出均未使用（由 GPIO 控制器接管控制），则 OISx、OISxN、CCxP 和 CCxNP 位必须保持清零状态。

注： *与互补通道 OCx 和 OCxN 相连的外部 I/O 引脚的状态取决于通道 OCx 和 OCxN 的状态以及 AFIO 寄存器。*

41.5.9 TIM15 计数器 (TIM15_CNT)

TIM15 counter

偏移地址：0x24

复位值：0x0000 0000

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

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

位 31 **UIFCPY**：UIF 副本 (UIF Copy)
该位是 TIMx_ISR 寄存器中 UIF 位的只读副本。

位 30:16 保留，必须保持复位值。

位 15:0 **CNT[15:0]**：计数器值 (Counter value)

41.5.10 TIM15 预分频器 (TIM15_PSC)

TIM15 prescaler

偏移地址：0x28

复位值：0x0000

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

位 15:0 **PSC[15:0]**：预分频器值 (Prescaler value)
计数器时钟频率 (CK_CNT) 等于 f_{CK_PSC} / (PSC[15:0] + 1)。
PSC 包含每次发生更新事件（包括计数器通过 TIMx_EGR 寄存器中的 UG 位清零时，或在配置为“复位模式”时通过触发控制器清零时）时要装载到活动预分频器寄存器的值。

41.5.11 TIM15 自动重载寄存器 (TIM15_ARR)

TIM15 auto-reload register

偏移地址：0x2C

复位值：0xFFFF

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

位 15:0 **ARR[15:0]**：自动重载值 (Auto-reload value)
ARR 为要装载到实际自动重载寄存器的值。
有关 ARR 更新和行为的更多详细信息，请参见*第 1614 页的第 41.4.1 节：时基单元*。
当自动重载值为空时，计数器不工作。

41.5.12 TIM15 重复计数器寄存器 (TIM15_RCR)

TIM15 repetition counter register

偏移地址：0x30

复位值：0x0000

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

位 15:8 保留，必须保持复位值。

位 7:0 **REP[7:0]**：重复计数器值 (Repetition Counter value)

使能预装载寄存器时，用户可通过这些位设置比较寄存器的更新频率（即，从预装载寄存器向活动寄存器周期性传输数据）；使能更新中断时，也可设置更新中断的生成速率。

与 REP_CNT 相关的减计数器每次计数到 0 时，都将生成一个更新事件并且计数器从 REP 值重新开始计数。由于只有生成重复更新事件 U_RC 时，REP_CNT 才会重载 REP 值，因此在生成下一重复更新事件之前，无论向 TIMx_RCR 寄存器写入何值都无影响。

这意味着在 PWM 模式 (REP+1)下 对应于边沿对齐模式的 PWM 周期数。

41.5.13 TIM15 捕获/比较寄存器 1 (TIM15_CCR1)

TIM15 capture/compare register 1

偏移地址：0x34

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR1[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 15:0 **CCR1[15:0]**：捕获/比较 1 值 (Capture/Compare 1 value)

如果通道 CC1 配置为输出：

CCR1 是捕获/比较寄存器 1 的预装载值。

如果没有通过 TIMx_CCMR1 寄存器中的 OC1PE 位来使能预装载功能，则该值立刻生效； 否则只在发生更新事件时生效（拷贝到实际起作用的捕获/比较寄存器 1）。

实际捕获/比较寄存器中包含要与计数器 TIMx_CNT 进行比较并在 OC1 输出上发出信号的值。

如果通道 CC1 配置为输入：

CCR1 为上一个输入捕获 1 事件 (IC1) 发生时的计数器值。

41.5.14 TIM15 捕获/比较寄存器 2 (TIM15_CCR2)

TIM15 capture/compare register 2

偏移地址：0x38

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR2[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 15:0 **CCR2[15:0]**：捕获/比较 2 值 (Capture/Compare 2 value)

如果通道 CC2 配置为输出：

CCR2 是捕获/比较寄存器 2 的预装载值。

如果没有通过 TIMx_CCMR2 寄存器中的 OC2PE 位来使能预装载功能，则该值立刻生效； 否则只在发生更新事件时生效（拷贝到实际起作用的捕获/比较寄存器 2）。

实际捕获/比较寄存器中包含要与计数器 TIMx_CNT 进行比较并在 OC2 输出上发出信号的值。

如果通道 CC2 配置为输入：

CCR2 为上一个输入捕获 2 事件 (IC2) 发生时的计数器值。

41.5.15 TIM15 断路和死区寄存器 (TIM15_BDTR)

TIM15 break and dead-time register

偏移地址：0x44

复位值：0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MOE	AOE	BKP	BKE	OSSR	OSSI	LOCK[1:0]		DTG[7:0]							
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

注： *由于可以根据 LOCK 配置锁定位 BKF[3:0]、AOE、BKP、BKE、OSSI、OSSR 和 DTG[7:0] 位的写操作，因此必须在第一次对 TIMx_BDTR 寄存器执行写访问时对这些位进行配置。*

位 31:20 保留，必须保持复位值。

位 19:16 **BKF[3:0]**：断路滤波器 (Break filter)

此位域可定义 BRK 输入信号的采样频率和适用于 BRK 的数字滤波器采样长度。数字滤波器由事件计数器组成，每 N 个事件才视为一个有效输出边沿：

0000：无滤波器，BRK 异步工作
0001：$f_{SAMPLING}=f_{CK_INT}$，N=2
0010：$f_{SAMPLING}=f_{CK_INT}$，N=4
0011：$f_{SAMPLING}=f_{CK_INT}$，N=8
0100：$f_{SAMPLING}=f_{DTS}/2$，N=6
0101：$f_{SAMPLING}=f_{DTS}/2$，N=8
0110：$f_{SAMPLING}=f_{DTS}/4$，N=6
0111：$f_{SAMPLING}=f_{DTS}/4$，N=8
1000：$f_{SAMPLING}=f_{DTS}/8$，N=6
1001：$f_{SAMPLING}=f_{DTS}/8$，N=8
1010：$f_{SAMPLING}=f_{DTS}/16$，N=5
1011：$f_{SAMPLING}=f_{DTS}/16$，N=6
1100：$f_{SAMPLING}=f_{DTS}/16$，N=8
1101：$f_{SAMPLING}=f_{DTS}/32$，N=5
1110：$f_{SAMPLING}=f_{DTS}/32$，N=6
1111：$f_{SAMPLING}=f_{DTS}/32$，N=8

注： 编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1 后，此位即无法修改。

位 15 **MOE**：主输出使能 (Main output enable)

只要断路输入变为有效状态，此位便由硬件异步清零。此位由软件置 1，也可根据 AOE 位状态自动置 1。此位仅对配置为输出的通道有效。

0：OC 和 OCN 输出被禁止或被强制为空闲状态，具体取决于 OSSI 位。
1：如果 OC 和 OCN 输出的相应使能位（TIMx_CCER 寄存器中的 CCxE 和 CCxNE 位）均置 1，则使能 OC 和 OCN 输出。

有关详细信息，请参见 OC/OCN 使能说明（*第 1655 页的第 41.5.8 节：TIM15 捕获/比较使能寄存器 (TIM15_CCER)*）。

位 14 **AOE**：自动输出使能 (Automatic output enable)

0：MOE 只能由软件置 1
1：MOE 可由软件置 1，也可在发生下一更新事件时自动置 1(如果断路输入无效）

注： 只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1，此位即无法修改。

位 13 **BKP**：断路极性 (Break polarity)

0：断路输入 BRK 为低电平有效
1：断路输入 BRK 为高电平有效

*注： **1**：只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1，此位即无法修改。*
__2__：对该位执行任何写操作后，都需要经过 1 个 APB 时钟周期的延迟才生效。

位 12 **BKE**：断路使能 (Break enable)

0：禁止断路输入（BRK 和 CCS 时钟故障事件）
1：使能断路输入（BRK 和 CCS 时钟故障事件）
编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1 后，此位即无法修改。

注： 对该位执行任何写操作后，都需要经过 1 个 APB 时钟周期的延迟才生效。

位 11 **OSSR**：运行模式下的关闭状态选择 (Off-state selection for Run mode)

此位在 MOE = 1 时作用于配置为输出模式且具有互补输出的通道。如果定时器中没有互补输出，则不存在 OSSR。

有关详细信息，请参见 OC/OCN 使能说明（*第 1655 页的第 41.5.8 节：TIM15 捕获/比较使能寄存器 (TIM15_CCER)*）。

0：处于无效状态时，禁止 OC/OCN 输出（定时器释放输出控制，由强制高阻态的 AFIO 逻辑接管）。

1：处于无效状态时，一旦 CCxE=1 或 CCxNE=1，便使能 OC/OCN 输出并将其设为无效电平（输出仍由定时器控制）。

注：　编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 2 后，此位即无法修改。

位 10 **OSSI**：空闲模式下的关闭状态选择 (Off-state selection for Idle mode)

此位在 MOE=0 时作用于配置为输出的通道。

有关详细信息，请参见 OC/OCN 使能说明（*第 1655 页的第 41.5.8 节：TIM15 捕获/比较使能寄存器 (TIM15_CCER)*）。

0：处于无效状态时，禁止 OC/OCN 输出（OC/OCN 使能输出信号 = 0）

1：处于无效状态时，一旦 CCxE = 1 或 CCxNE = 1，便将 OC/OCN 输出首先强制为其空闲电平。然后设置 OC/OCN 使能输出信号=1

注：　编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 2 后，此位即无法修改。

位 9:8 **LOCK[1:0]**：锁定配置 (Lock configuration)

这些位用于针对软件错误提供写保护。

00：关闭锁定——不对任何位提供写保护。

01：锁定级别 1，此时无法对 TIMx_BDTR 寄存器中的 DTG 位、TIMx_CR2 寄存器中的 OISx 和 OISxN 位以及 TIMx_BDTR 寄存器中的 BKE/BKP/AOE 位执行写操作。

10：锁定级别 2，此时无法对锁定级别 1 中适用的各位、CC 极性位（TIMx_CCER 寄存器中的 CCxP/CCxNP 位，只要通过 CCxS 位将相关通道配置为输出）以及 OSSR 和 OSSI 位执行写操作。

11：锁定级别 3，此时无法对锁定级别 2 中适用的各位、CC 控制位（TIMx_CCMRx 寄存器中的 OCxM 和 OCxPE 位，只要通过 CCxS 位将相关通道配置为输出）执行写操作。

注：　复位后只能对 LOCK 位执行一次写操作。对 TIMx_BDTR 寄存器执行写操作后其中的内容将冻结，直到下一次复位。

位 7:0 **DTG[7:0]**：配置死区发生器 (Dead-time generator setup)

此位域定义插入到互补输出之间的死区持续时间。DT 与该持续时间相对应。

DTG[7:5]=0xx => DT=DTG[7:0]x t_{dtg}，其中 $t_{dtg}=t_{DTS}$。

DTG[7:5]=10x => DT=(64+DTG[5:0])xt_{dtg}，其中 $T_{dtg}=2xt_{DTS}$。

DTG[7:5]=110 => DT=(32+DTG[4:0])xt_{dtg}，其中 $T_{dtg}=8xt_{DTS}$。

DTG[7:5]=111 => DT=(32+DTG[4:0])xt_{dtg}，其中 $T_{dtg}=16xt_{DTS}$。

示例：如果 T_{DTS}=125ns (8MHz)，则可能的死区值为：

0 到 15875 ns（步长为 125 ns），

16 µs 到 31750 ns（步长为 250 ns），

32 µs 到 63 µs（步长为 1 µs），

64 µs 到 126 µs（步长为 2 µs）。

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

41.5.16 TIM15 DMA 控制寄存器 (TIM15_DCR)

TIM15 DMA control register

偏移地址：0x48

复位值：0x0000

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

位 15:13 保留，必须保持复位值。

位 12:8 **DBL[4:0]**：DMA 连续传送长度 (DMA burst length)

该 5 位域定义了 DMA 的传送长度（当对 TIMx_DMAR 寄存器进行读或写时，定时器进行一次连续传送）。

00000：1 次传送，
00001：2 次传送，
00010：3 次传送，
...
10001：18 次传送。

位 7:5 保留，必须保持复位值。

位 4:0 **DBA[4:0]**：DMA 基址 (DMA base address)

该 5 位域定义 DMA 传输的基址（通过 TIMx_DMAR 地址进行读/写访问时）。DBA 定义为从 TIMx_CR1 寄存器地址开始计算的偏移量。

示例：

00000：TIMx_CR1，
00001：TIMx_CR2，
00010：TIMx_SMCR，
...

41.5.17 TIM15 全传输 DMA 地址 (TIM15_DMAR)

TIM15 DMA address for full transfer

偏移地址：0x4C

复位值：0x0000

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

位 15:0 **DMAB[15:0]**：DMA 连续传送寄存器 (DMA register for burst accesses)

对 DMAR 寄存器执行读或写操作将访问位于如下地址的寄存器

(TIMx_CR1 地址) + (DBA + DMA 索引) x 4

其中 TIMx_CR1 地址为控制寄存器 1 的地址，DBA 为 TIMx_DCR 寄存器中配置的 DMA 基址，DMA 索引由 DMA 传输自动控制，其范围介于 0 到 DBL（TIMx_DCR 寄存器中配置的 DBL）之间。

41.5.18 TIM15 复用寄存器 1 (TIM15_AF1)

TIM15 alternate register 1

偏移地址：0x60

复位值：0x0000 0001

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	BKCMP2P	BKCMP1P	BKINP	BKDF1BK0E	Res.	Res.	Res.	Res.	Res.	BKCMP2E	BKCMP1E	BKINE
				rw	rw	rw	rw						rw	rw	rw

位 31:12 保留，必须保持复位值。

位 11 **BKCMP2P**：BRK COMP2 输入极性 (BRK COMP2 input polarity)
此位选择 COMP2 输入灵敏度，必须与 BKP 极性位一起编程。
0：COMP2 输入为低电平有效
1：COMP2 输入为高电平有效
注： 只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1，此位即无法修改。

位 10 **BKCMP1P**：BRK COMP1 输入极性 (BRK COMP1 input polarity)
此位选择 COMP1 输入灵敏度，必须与 BKP 极性位一起编程。
0：COMP1 输入为低电平有效
1：COMP1 输入为高电平有效
注： 只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1，此位即无法修改。

位 9 **BKINP**：BRK BKIN 输入极性 (BRK BKIN input polarity)
此位选择 BKIN 复用功能输入灵敏度，必须与 BKP 极性位一起编程。
0：BKIN 输入为低电平有效
1：BKIN 输入为高电平有效
注： 只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1，此位即无法修改。

位 8 **BKDF1BK0E**：BRK dfsdm1_break[0] 使能 (BRK dfsdm1_break[0] enable)
此位使能定时器 BRK 输入的 dfsdm1_break[0]。dfsdm1_break[0] 输出与其他 BRK 源进行“或”运算。
0：禁止 dfsdm1_break[0] 输入
1：使能 dfsdm1_break[0] 输入
注： 只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1，此位即无法修改。

位 7:3 保留，必须保持复位值

位 2 **BKCMP2E**：BRK COMP2 使能 (BRK COMP2 enable)
此位使能定时器 BRK 输入的 COMP2。COMP2 输出与其他 BRK 源进行“或”运算。
0：禁止 COMP2 输入
1：使能 COMP2 输入
注： 只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1，此位即无法修改。

位 1 **BKCMP1E**：BRK COMP1 使能 (BRK COMP1 enable)
此位使能定时器 BRK 输入的 COMP1。COMP1 输出与其他 BRK 源进行“或”运算。
0：禁止 COMP1 输入
1：使能 COMP1 输入
注： 只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1，此位即无法修改。

位 0 **BKINE**：BRK BKIN 输入使能 (BRK BKIN input enable)
此位使能定时器 BRK 输入的 BKIN 复用功能。BKIN 输入与其他 BRK 源进行“或”运算。
0：禁止 BKIN 输入
1：使能 BKIN 输入
注： 只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1，此位即无法修改。

41.5.19 TIM15 输入选择寄存器 (TIM15_TISEL)

TIM15 input selection register

偏移地址：0x68

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	TI2SEL[3:0]				Res.	Res.	Res.	Res.	TI1SEL[3:0]			
				rw	rw	rw	rw					rw	rw	rw	rw

位 31:12 保留，必须保持复位值。

位 11:8 **TI2SEL[3:0]**：选择 TI2[0] 到 TI2[15] 输入 (selects TI2[0] to TI2[15] input)
0000：TIM15_CH2 输入
0001：TIM2_CH2 输入
0010：TIM3_CH2 输入
0011：TIM4_CH2 输入
其它：保留

位 7:4 保留，必须保持复位值。

位 3:0 **TI1SEL[3:0]**：选择 TI1[0] 到 TI1[15] 输入 (selects TI1[0] to TI1[15] input)
0000：TIM15_CH1 输入
0001：TIM2_CH1 输入
0010：TIM3_CH1 输入
0011：TIM4_CH1 输入
0100：LSE
0101：CSI
0110：MCO2
其它：保留

41.5.20 TIM15寄存器映射

TIM15 寄存器可映射为 16 位可寻址寄存器，如下表所述：

表 327. TIM15 寄存器映射和复位值

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	**TIM15_CR1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	UIFREMAP	Res.	CKD [1:0]		ARPE	Res.	Res.	Res.	OPM	URS	UDIS	CEN
	Reset value																					0		0	0	0				0	0	0	0
0x04	**TIM15_CR2**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OIS2	OIS1N	OIS1	TI1S	MMS[2:0]			CCDS	CCUS	Res.	CCPC
	Reset value																						0	0	0	0	0	0	0	0	0		0
0x08	**TIM15_SMCR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TS [4:3]		Res.	Res.	Res.	SMS[3]	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MSM	TS[2:0]			Res.	SMS[2:0]		
	Reset value											0	0				0									0	0	0	0		0	0	0
0x0C	**TIM15_DIER**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TDE	COMDE	Res.	Res.	CC2DE	CC1DE	UDE	BIE	TIE	COMIE	Res.	Res.	CC2IE	CC1IE	UIE
	Reset value																		0	0			0	0	0	0	0	0			0	0	0
0x10	**TIM15_SR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CC2OF	CC1OF	Res.	BIF	TIF	COMIF	Res.	Res.	CC2IF	CC1IF	UIF
	Reset value																						0	0		0	0	0			0	0	0
0x14	**TIM15_EGR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BG	TG	COMG	Res.	Res.	CC2G	CC1G	UG
	Reset value																									0	0	0			0	0	0
0x18	**TIM15_CCMR1** Output Compare mode	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC2M[3]	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC1M[3]	Res.	OC2M [2:0]			OC2PE	OC2FE	CC2S [1:0]		Res.	OC1M [2:0]			OC1PE	OC1FE	CC1S [1:0]	
	Reset value								0								0		0	0	0	0	0	0	0		0	0	0	0	0	0	0
	TIM15_CCMR1 Input Capture mode	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	IC2F[3:0]				IC2 PSC [1:0]		CC2S [1:0]		IC1F[3:0]				IC1 PSC [1:0]		CC1S [1:0]	
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x20	**TIM15_CCER**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CC2NP	Res.	CC2P	CC2E	CC1NP	CC1NE	CC1P	CC1E
	Reset value																									0		0	0	0	0	0	0
0x24	**TIM15_CNT**	UIFCPY or Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CNT[15:0]															
	Reset value	0																0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

表 327. TIM15 寄存器映射和复位值（续）

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x28	**TIM15_PSC**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PSC[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x2C	**TIM15_ARR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ARR[15:0]															
	Reset value																	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0x30	**TIM15_RCR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REP[7:0]							
	Reset value																									0	0	0	0	0	0	0	0
0x34	**TIM15_CCR1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CCR1[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x38	**TIM15_CCR2**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CCR2[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x44	**TIM15_BDTR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BKF[3:0]				MOE	AOE	BKP	BKE	OSSR	OSSI	LOCK [1:0]		DT[7:0]							
	Reset value													0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x48	**TIM15_DCR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DBL[4:0]					Res.	Res.	Res.	DBA[4:0]				
	Reset value																				0	0	0	0	0				0	0	0	0	0
0x4C	**TIM15_DMAR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DMAB[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x60	**TIM15_AF1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BKCMP2P	BKCMP1P	BKINP	BKDF1BK0E	Res.	Res.	Res.	Res.	Res.	BKCMP2E	BKCMP1E	BKINE
	Reset value																					0	0	0	0						0	0	1
0x68	**TIM15_TISEL**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TI2SEL[3:0]				Res.	Res.	Res.	Res.	TI1SEL[3:0]			
	Reset value																					0	0	0	0					0	0	0	0

有关寄存器边界地址的信息，请参见*第 100 页的第 2.2.2 节*。

41.6 TIM16/TIM17 寄存器

有关寄存器说明中使用的缩写，请参见*第 94 页的第 1.1 节*。

41.6.1 TIM16/TIM17控制寄存器 1 (TIMx_CR1)

TIM16/TIM17 control register 1

偏移地址：0x00

复位值：0x0000

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

位 15:12 保留，必须保持复位值。

位 11 **UIFREMAP**：UIF 状态位重映射 (UIF status bit remapping)
0：无重映射 UIF 状态位不复制到 TIMx_CNT 寄存器的位 31。
1：使能重映射。UIF 状态位复制到 TIMx_CNT 寄存器的位 31。

位 10 保留，必须保持复位值。

位 9:8 **CKD[1:0]**：时钟分频 (Clock division)
此位域指示定时器时钟 (CK_INT) 频率与死区发生器以及数字滤波器 (TIx) 所使用的死区及采样时钟 (t_{DTS}) 之间的分频比，
00：$t_{DTS}=t_{CK_INT}$
01：$t_{DTS}=2*t_{CK_INT}$
10：$t_{DTS}=4*t_{CK_INT}$
11：保留，不要设置成此值

位 7 **ARPE**：自动重载预装载使能 (Auto-reload preload enable)
0：TIMx_ARR 寄存器不进行缓冲
1：TIMx_ARR 寄存器进行缓冲

位 6:4 保留，必须保持复位值。

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

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

位 1 **UDIS**：更新禁止 (Update disable)

此位由软件置 1 和清零，用以使能/禁止 UEV 事件生成。

0：使能 UEV。更新 (UEV) 事件可通过以下事件之一生成：

– 计数器上溢/下溢

– 将 UG 位置 1

– 通过从模式控制器生成的更新事件

然后更新影子寄存器的值。

1：禁止 UEV。不会生成更新事件，各影子寄存器的值（ARR、PSC 和 CCRx）保持不变。但如果将 UG 位置 1，或者从从模式控制器接收到硬件复位，则会重新初始化计数器和预分频器。

位 0 **CEN**：计数器使能 (Counter enable)

0：禁止计数器

1：使能计数器

注： 只有事先通过软件将 CEN 位置 1，才可以使用外部时钟和门控模式。而触发模式可通过硬件自动将 CEN 位置 1。

41.6.2 TIM16/TIM17 控制寄存器 2 (TIMx_CR2)

TIM16/TIM17 control register 2

偏移地址：0x04

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	OIS1N	OIS1	Res.	Res.	Res.	Res.	CCDS	CCUS	Res.	CCPC
						rw	rw					rw	rw		rw

位 15:10 保留，必须保持复位值。

位 9 **OIS1N**：输出空闲状态 1（OC1N 输出）(Output Idle state 1 (OC1N output))

0：当 MOE=0 时，经过死区时间后 OC1N=0

1：当 MOE=0 时，经过死区时间后 OC1N=1

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

位 8 **OIS1**：输出空闲状态 1（OC1 输出）(Output Idle state 1 (OC1 output))

0：当 MOE=0 时，（如果 OC1N 有效，则经过死区时间之后）OC1=0

1：当 MOE=1 时，（如果 OC1N 有效，则经过死区时间之后）OC1=0

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

位 7:4 保留，必须保持复位值。

位 3 **CCDS**：捕获/比较 DMA 选择 (Capture/compare DMA selection)

0：发生 CCx 事件时发送 CCx DMA 请求

1：发生更新事件时发送 CCx DMA 请求

位 2 **CCUS**：捕获/比较控制更新选择 (Capture/compare control update selection)

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

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

注： 此位仅对具有互补输出的通道有效。

位 1 保留，必须保持复位值。

位 0 **CCPC**：捕获/比较预装载控制 (Capture/compare preloaded control)
0：CCxE、CCxNE 和 OCxM 位未进行预装载
1：CCxE、CCxNE 和 OCxM 位在写入后被预加载，只有当 COM 位置 1 时才进行更新。
注： 此位仅对具有互补输出的通道有效。

41.6.3 TIM16/TIM17 DMA/中断使能寄存器 (TIMx_DIER)

TIM16/TIM17 DMA/interrupt enable register

偏移地址：0x0C

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	COMDE	Res.	Res.	Res.	CC1DE	UDE	BIE	Res.	COMIE	Res.	Res.	Res.	CC1IE	UIE
		rw				rw	rw	rw		rw				rw	rw

位 15:14 保留，必须保持复位值。

位 13 **COMDE**：COM DMA 请求使能 (COM DMA request enable)
0：禁止 COM DMA 请求
1：使能 COM DMA 请求

位 12:10 保留，必须保持复位值。

位 9 **CC1DE**：捕获/比较 1 DMA 请求使能 (Capture/Compare 1 DMA request enable)
0：禁止 CC1 DMA 请求
1：使能 CC1 DMA 请求

位 8 **UDE**：更新 DMA 请求使能 (Update DMA request enable)
0：禁止更新 DMA 请求
1：使能更新 DMA 请求

位 7 **BIE**：断路中断使能 (Break interrupt enable)
0：禁止断路中断
1：使能断路中断

位 6 保留，必须保持复位值。

位 5 **COMIE**：COM 中断使能 (COM interrupt enable)
0：禁止 COM 中断
1：使能 COM 中断

位 4:2 保留，必须保持复位值。

位 1 **CC1IE**：捕获/比较 1 中断使能 (Capture/Compare 1 interrupt enable)
0：禁止 CC1 中断
1：使能 CC1 中断

位 0 **UIE**：更新中断使能 (Update interrupt enable)
0：禁止更新中断
1：使能更新中断

41.6.4 TIM16/TIM17 状态寄存器 (TIMx_SR)

TIM16/TIM17 status register

偏移地址：0x10

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	CC1OF	Res.	BIF	Res.	COMIF	Res.	Res.	Res.	CC1IF	UIF
						rc_w0		rc_w0		rc_w0				rc_w0	rc_w0

位 15:10 保留，必须保持复位值。

位 9 **CC1OF**：捕获/比较 1 重复捕获标志 (Capture/Compare 1 overcapture flag)

仅当将相应通道配置为输入捕获模式时，此标志位才会由硬件置 1。通过软件写入“0”可将该位清零。

0：未检测到重复捕获。

1：TIMx_CCR1 寄存器中已捕获到计数器值且 CC1IF 标志已置 1。

位 8 保留，必须保持复位值。

位 7 **BIF**：断路中断标志 (Break interrupt flag)

只要断路输入变为有效状态，此标志便由硬件置 1。断路输入无效后可通过软件对其清零。

0：未发生断路事件。

1：在断路输入上检测到有效电平。

位 6 保留，必须保持复位值。

位 5 **COMIF**：COM 中断标志 (COM interrupt flag)

发生一个 COM 事件时，会由硬件设置这个标志（一旦捕获/比较控制位——CCxE、CCxNE、OCxM——已更新）。但需要通过软件清零。

0：未发生 COM 事件。

1：COM 中断挂起。

位 4:2 保留，必须保持复位值。

位 1 **CC1IF**：捕获/比较 1 中断标志 (Capture/Compare 1 interrupt flag)

如果通道 CC1 配置为输出：

当计数器与比较值匹配时，此标志由硬件置 1。但需要通过软件清零。

0：不匹配。

1：TIMx_CNT 计数器的值与 TIMx_CCR1 寄存器的值匹配。当 TIMx_CCR1 的值大于 TIMx_ARR 的值时，CC1IF 位将在计数器发生上溢时变为高电平。

如果通道 CC1 配置为输入：

此位将在发生捕获事件时由硬件置 1。通过软件或读取 TIMx_CCR1 寄存器将该位清零。

0：未发生输入捕获事件

1：TIMx_CCR1 寄存器中已捕获到计数器值（IC1 上已检测到与所选极性匹配的边沿）

位 0 **UIF**：更新中断标志 (Update interrupt flag)

该位在发生更新事件时通过硬件置 1。但需要通过软件清零。

0：未发生更新。

1：更新中断挂起。该位在以下情况下更新寄存器时由硬件置 1：

– TIMx_CR1 寄存器中的 UDIS = 0，并且重复计数器值上溢时（重复计数器 = 0 时更新）。

– TIMx_CR1 寄存器中的 URS = 0 且 UDIS = 0，并且由软件使用 TIMx_EGR 寄存器中的 UG 位重新初始化 CNT 时。

41.6.5 TIM16/TIM17 事件生成寄存器 (TIMx_EGR)

TIM16/TIM17 event generation register

偏移地址：0x14

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BG	Res.	COMG	Res.	Res.	Res.	CC1G	UG
								w		w				w	w

位 15:8 保留，必须保持复位值。

位 7 **BG**：断路生成 (Break generation)

此位由软件置 1 以生成事件，并由硬件自动清零。

0：不执行任何操作。

1：生成断路事件。MOE 位清零且 BIF 标志置 1。使能后可发生相关中断或 DMA 传输事件。

位 6 保留，必须保持复位值。

位 5 **COMG**：捕获/比较控制更新生成 (Capture/Compare control update generation)

该位可通过软件置 1，并由硬件自动清零。

0：不执行任何操作

1：CCPC 位置 1 时，可更新 CCxE、CCxNE 和 OCxM 位

注： 此位仅对具有互补输出的通道有效。

位 4:2 保留，必须保持复位值。

位 1 **CC1G**：捕获/比较 1 生成 (Capture/Compare 1 generation)

此位由软件置 1 以生成事件，并由硬件自动清零。

0：不执行任何操作。

1：通道 1 上生成捕获/比较事件：

如果通道 CC1 配置为输出：

使能时，CC1IF 标志置 1 并发送相应的中断或 DMA 请求。

如果通道 CC1 配置为输入：

TIMx_CCR1 寄存器中将捕获到计数器当前值。使能时，CC1IF 标志置 1 并发送相应的中断或 DMA 请求。如果 CC1IF 标志已为高电平，CC1OF 标志将置 1。

位 0 **UG**：更新生成 (Update generation)

该位可通过软件置 1，并由硬件自动清零。

0：不执行任何操作。

1：重新初始化计数器并生成寄存器更新事件。请注意，预分频器计数器也将清零（但预分频比不受影响）。

41.6.6 TIM16/TIM17 捕获/比较模式寄存器 1 (TIMx_CCMR1)

TIM16/TIM17 capture/compare mode register 1

偏移地址：0x18

复位值：0x0000 0000

这些通道可用于输入（捕获模式）或输出（比较模式）模式。通道方向通过配置相应的 CCxS 位进行定义。此寄存器的所有其它位在输入模式和输出模式下的功能均不同。对于任一给定位，OCxx 用于说明通道配置为输出时该位对应的功能，ICxx 则用于说明通道配置为输入时该位对应的功能。因此，必须注意同一个位在输入阶段和输出阶段具有不同的含义。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC1M[3]
															Res.
															rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC1M[2:0]			OC1PE	OC1FE	CC1S[1:0]	
								IC1F[3:0]				IC1PSC[1:0]			
								rw	rw	rw	rw	rw	rw	rw	rw

输出比较模式：

位 31:17 保留，始终读为 0

位 16 **OC1M[3]**：输出比较 1 模式（位 3）(Output Compare 1 mode (bit 3))

位 15:7 保留

位 6:4 **OC1M[2:0]**：输出比较 1 模式（位 2 到 0）(Output Compare 1 mode (bits 2 to 0))

这些位定义提供 OC1 和 OC1N 的输出参考信号 OC1REF 的行为。OC1REF 为高电平有效，而 OC1 和 OC1N 的有效电平则取决于 CC1P 位和 CC1NP 位。

0000：冻结——输出比较寄存器 TIMx_CCR1 与计数器 TIMx_CNT 进行比较不会对输出造成任何影响。

0001：将通道 1 设置为匹配时输出有效电平。当计数器 TIMx_CNT 与捕获/比较寄存器 1 (TIMx_CCR1) 匹配时，OC1REF 信号强制变为高电平。

0010：将通道 1 设置为匹配时输出无效电平。当计数器 TIMx_CNT 与捕获/比较寄存器 1 (TIMx_CCR1) 匹配时，OC1REF 信号强制变为低电平。

0011：翻转——TIMx_CNT=TIMx_CCR1 时，OC1REF 发生翻转。

0100：强制变为无效电平——OC1REF 强制变为低电平。

0101：强制变为有效电平——OC1REF 强制变为高电平。

0110：PWM 模式 1——只要 TIMx_CNT<TIMx_CCR1，通道 1 便为有效状态，否则为无效状态。

0111：PWM 模式 2——只要 TIMx_CNT<TIMx_CCR1，通道 1 便为无效状态，否则为有效状态。

所有其它值：保留

注： ***1：****只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 3 且 CC1S=“00”（通道配置为输出），这些位即无法修改。*

2：*在 PWM 模式 1 或 PWM 模式 2 下，仅当比较结果发生改变或输出比较模式由“冻结”模式切换到“PWM”模式时，OCREF 电平才会发生更改。*

位 3 **OC1PE**：输出比较 1 预装载使能 (Output Compare 1 preload enable)

0：禁止与 TIMx_CCR1 相关的预装载寄存器。可随时向 TIMx_CCR1 写入数据，写入后将立即使用新值。

1：使能与 TIMx_CCR1 相关的预装载寄存器。可读/写访问预装载寄存器。TIMx_CCR1 预装载值在每次生成更新事件时都会装载到活动寄存器中。

*注： **1**：只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 3 且 CC1S=“00”（通道配置为输出），这些位即无法修改。*

***2**：只有单脉冲模式下才可在未验证预装载寄存器的情况下使用 PWM 模式（TIMx_CR1 寄存器中的 OPM 位置 1）。其它情况下则无法保证该行为。*

位 2 **OC1FE**：输出比较 1 快速使能 (Output Compare 1 fast enable)

此位用于加快触发输入事件对 CC 输出的影响。

0：即使触发开启，CC1 也将根据计数器和 CCR1 值正常工作。触发输入出现边沿时，激活 CC1 输出的最短延迟时间为 5 个时钟周期。

1：触发输入上出现有效边沿相当于 CC1 输出上的比较匹配。随后，无论比较结果如何，OC 都设置为比较电平。采样触发输入和激活 CC1 输出的延迟时间缩短为 3 个时钟周期。仅当通道配置为 PWM1 或 PWM2 模式时，OC1FE 才会起作用。

位 1:0 **CC1S**：捕获/比较 1 选择 (Capture/Compare 1 Selection)

此位域定义通道方向（输入/输出）以及所使用的输入。

00：CC1 通道配置为输出

01：CC1 通道配置为输入，IC1 映射到 TI1 上

10：CC1 通道配置为输入，IC1 映射到 TI2 上

11：CC1 通道配置为输入，IC1 映射到 TRC 上。此模式仅在通过 TS 位（TIMx_SMCR 寄存器）选择内部触发输入时有效

注： 仅当通道关闭时（TIMx_CCER 中的 CC1E = “0”），才可向 CC1S 位写入数据。

输入捕获模式

位 31:8 保留，必须保持复位值。

位 7:4 **IC1F[3:0]**：输入捕获 1 滤波器 (Input capture 1 filter)

此位域可定义 TI1 输入的采样频率和适用于 TI1 的数字滤波器的采样长度。数字滤波器由事件计数器组成，每 N 个连续事件才视为一个有效输出边沿：

0000：无滤波器，按 f_{DTS} 频率进行采样

0001：$f_{SAMPLING}=f_{CK_INT}$，N=2

0010：$f_{SAMPLING}=f_{CK_INT}$，N=4

0011：$f_{SAMPLING}=f_{CK_INT}$，N=8

0100：$f_{SAMPLING}=f_{DTS}/2$，N=

0101：$f_{SAMPLING}=f_{DTS}/2$，N=8

0110：$f_{SAMPLING}=f_{DTS}/4$，N=6

0111：$f_{SAMPLING}=f_{DTS}/4$，N=8

1000：$f_{SAMPLING}=f_{DTS}/8$，N=6

1001：$f_{SAMPLING}=f_{DTS}/8$，N=8

1010：$f_{SAMPLING}=f_{DTS}/16$，N=5

1011：$f_{SAMPLING}=f_{DTS}/16$，N=6

1100：$f_{SAMPLING}=f_{DTS}/16$，N=8

1101：$f_{SAMPLING}=f_{DTS}/32$，N=5

1110：$f_{SAMPLING}=f_{DTS}/32$，N=6

1111：$f_{SAMPLING}=f_{DTS}/32$，N=8

位 3:2 **IC1PSC**：输入捕获 1 预分频器 (Input capture 1 prescaler)

此位域定义 CC1 输入 (IC1) 的预分频比。

只要 CC1E=“0”（TIMx_CCER 寄存器），预分频器便立即复位。

00：无预分频器，捕获输入上每检测到一个边沿便执行捕获

01：每发生 2 个事件便执行一次捕获

10：每发生 4 个事件便执行一次捕获

11：每发生 8 个事件便执行一次捕获

位 1:0 **CC1S**：捕获/比较 1 选择 (Capture/Compare 1 Selection)

此位域定义通道方向（输入/输出）以及所使用的输入。

00：CC1 通道配置为输出

01：CC1 通道配置为输入，IC1 映射到 TI1 上

10：CC1 通道配置为输入，IC1 映射到 TI2 上

11：CC1 通道配置为输入，IC1 映射到 TRC 上。此模式仅在通过 TS 位（TIMx_SMCR 寄存器）选择内部触发输入时有效

注： 仅当通道关闭时（TIMx_CCER 中的 CC1E = “0”），才可向 CC1S 位写入数据。

41.6.7 TIM16/TIM17 捕获/比较使能寄存器 (TIMx_CCER)

TIM16/TIM17 capture/compare enable register

偏移地址：0x20

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CC1NP	CC1NE	CC1P	CC1E
												rw	rw	rw	rw

位 15:4 保留，必须保持复位值。

位 3 **CC1NP**：捕获/比较 1 互补输出极性 (Capture/Compare 1 complementary output polarity)

CC1 通道配置为输出：

0：OC1N 高电平有效。

1：OC1N 低电平有效。

CC1 通道配置为输入：

此位与 CC1P 配合使用，用以定义 TI1FP1 和 TI2FP1 的极性。请参见 CC1P 说明。

*注： **1.** 只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 2 或 3 且 CC1S=“00”（通道配置为输出），此位立即变为不可写状态。*

***2.** 此位将在具有互补输出的通道上进行预装载。如果 TIMx_CR2 寄存器中的 CCPC 位置 1，则仅当交换事件发生时，CC1NP 有效位才会从预装载位获取新值。*

位 2 **CC1NE**：捕获/比较 1 互补输出使能 (Capture/Compare 1 complementary output enable)

0：关闭——OC1N 未激活。OC1N 电平是 MOE、OSSI、OSSR、OIS1、OIS1N 和 CC1E 位的函数。

1：开启——在相应输出引脚上输出 OC1N 信号，具体取决于 MOE、OSSI、OSSR、OIS1、OIS1N 和 CC1E 位。

位 1 **CC1P**：捕获/比较 1 输出极性 (Capture/Compare 1 output polarity)

CC1 通道配置为输出：

0：OC1 高电平有效

1：OC1 低电平有效

CC1 通道配置为输入：

CC1NP/CC1P 位可针对触发或捕获操作选择 TI1FP1 和 TI2FP1 的极性。

00：非反相/上升沿触发。电路对 TIxFP1 上升沿敏感（在复位模式、外部时钟模式或触发模式下执行捕获或触发操作），TIxFP1 未反相（在门控模式下执行触发操作）。

01：反相/下降沿触发。电路对 TIxFP1 下降沿敏感（在复位模式、外部时钟模式或触发模式下执行捕获或触发操作），TIxFP1 反相（在门控模式下执行触发操作）。

10：保留，不使用此配置。

11：未反相/边沿触发。电路对 TIxFP1 上升沿和下降沿都敏感（在复位模式、外部时钟模式或触发模式下执行捕获或触发操作），TIxFP1 未反相（在门控模式下执行触发操作）。

注： ***1.*** *只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 2 或 3，此位立即变为不可写状态。*

2. *此位将在具有互补输出的通道上进行预装载。如果 TIMx_CR2 寄存器中的 CCPC 位置 1，则仅当生成换向事件时，CC1P 有效位才会从预装载位获取新值。*

位 0 **CC1E**：捕获/比较 1 输出使能 (Capture/Compare 1 output enable)

CC1 通道配置为输出：

0：关闭——OC1 未激活。OC1 电平是 MOE、OSSI、OSSR、OIS1、OIS1N 和 CC1NE 位的函数。

1：开启——OC1 信号输出到相应的输出引脚上，具体取决于 MOE、OSSI、OSSR、OIS1、OIS1N 和 CC1NE 位。

CC1 通道配置为输入：

此位决定了是否可以实际将计数器值捕获到输入捕获/比较寄存器 1 (TIMx_CCR1) 中。

0：禁止捕获

1：使能捕获

表 328. 具有断路功能的互补通道 OCx 和 OCxN 的输出控制位 (TIM16/17)

控制位					输出状态(1)	
MOE 位	OSSI 位	OSSR 位	CCxE 位	CCxNE 位	OCx 输出状态	OCxN 输出状态
1	X	X	0	0	禁止输出（不由定时器驱动：高阻态） OCx=0 OCxN=0、OCxN_EN=0	
		0	0	1	禁止输出（不由定时器驱动：高阻态） OCx=0	OCxREF + 极性 OCxN = OCxREF 异或 CCxNP
		0	1	0	OCxREF + 极性 OCx=OCxREF 异或 CCxP	禁止输出（不由定时器驱动：高阻态） OCxN=0
		X	1	1	OCREF + 极性 + 死区	OCREF 互补项（对 OCREF 进行"非"运算）+ 极性 + 死区
		1	0	1	关闭状态（输出使能为无效状态） OCx=CCxP	OCxREF + 极性 OCxN = OCxREF 异或 CCxNP
		1	1	0	OCxREF + 极性 OCx=OCxREF 异或 CCxP、OCx_EN=1	关闭状态 （输出使能为无效状态） OCxN=CCxNP、OCxN_EN=1
0	0	X	X	X	禁止输出（不再由定时器驱动）。输出状态由 GPIO 控制器定义，可以是高电平、低电平或高阻态。	
	1		0	0		
			0	1	关闭状态（输出使能为无效状态） 异步：OCx = CCxP、OCxN = CCxNP 那么如果时钟存在：在死区后 Ocx = OISx 且 OCxN = OISxN，从而假定 OISx 和 OISxN 在有效状态下与 OCX 和 OCxN 不对应	
			1	0		
			1	1		

1. 如果一个通道的两个输出均未使用（由 GPIO 控制器接管控制），则 OISx、OISxN、CCxP 和 CCxNP 位必须保持清零状态。

注： *与互补通道 OCx 和 OCxN 相连的外部 I/O 引脚的状态取决于通道 OCx 和 OCxN 的状态以及 AFIO 寄存器。*

41.6.8 TIM16/TIM17计数器 (TIMx_CNT)

TIM16/TIM17 counter

偏移地址：0x24

复位值：0x0000 0000

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

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

位 31 **UIFCPY**：UIF 副本 (UIF Copy)
该位是 TIMx_ISR 寄存器中 UIF 位的只读副本。如果 TIMx_CR1 中的 UIFREMAP 位复位，则位 31 保留，读为 0。

位 30:16 保留，必须保持复位值。

位 15:0 **CNT[15:0]**：计数器值 (Counter value)

41.6.9 TIM16/TIM17 预分频器 (TIMx_PSC)

TIM16/TIM17 prescaler

偏移地址：0x28

复位值：0x0000

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

位 15:0 **PSC[15:0]**：预分频器值 (Prescaler value)
计数器时钟频率 (CK_CNT) 等于 f_{CK_PSC} / (PSC[15:0] + 1)。
PSC 包含每次发生更新事件（包括计数器通过 TIMx_EGR 寄存器中的 UG 位清零时，或在配置为“复位模式”时通过触发控制器清零时）时要装载到活动预分频器寄存器的值。

41.6.10 TIM16/TIM17 自动重载寄存器 (TIMx_ARR)

TIM16/TIM17 auto-reload register

偏移地址：0x2C

复位值：0xFFFF

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

位 15:0 **ARR[15:0]**：自动重载值 (Auto-reload value)
ARR 为要装载到实际自动重载寄存器的值。
有关 ARR 更新和行为的更多详细信息，请参见*第 1614 页的第 41.4.1 节：时基单元*。
当自动重载值为空时，计数器不工作。

41.6.11 TIM16/TIM17 重复计数器寄存器 (TIMx_RCR)

TIM16/TIM17 repetition counter register

偏移地址：0x30

复位值：0x0000

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

位 15:8 保留，必须保持复位值。

位 7:0 **REP[7:0]**：重复计数器值 (Repetition Counter value)

使能预装载寄存器时，用户可通过这些位设置比较寄存器的更新频率（即，从预装载寄存器向活动寄存器周期性传输数据）；使能更新中断时，也可设置更新中断的生成速率。

与 REP_CNT 相关的减计数器每次计数到 0 时，都将生成一个更新事件并且计数器从 REP 值重新开始计数。由于只有生成重复更新事件 U_RC 时，REP_CNT 才会重载 REP 值，因此在生成下一重复更新事件之前，无论向 TIMx_RCR 寄存器写入何值都无影响。

这意味着在 PWM 模式 (REP+1)下 对应于边沿对齐模式的 PWM 周期数。

41.6.12 TIM16/TIM17 捕获/比较寄存器 1 (TIMx_CCR1)

TIM16/TIM17 capture/compare register 1

偏移地址：0x34

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CCR1[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 15:0 **CCR1[15:0]**：捕获/比较 1 值 (Capture/Compare 1 value)

如果通道 CC1 配置为输出：

CCR1 是捕获/比较寄存器 1 的预装载值。

如果没有通过 TIMx_CCMR1 寄存器中的 OC1PE 位来使能预装载功能，则该值立刻生效； 否则只在发生更新事件时生效（拷贝到实际起作用的捕获/比较寄存器 1）。

实际捕获/比较寄存器中包含要与计数器 TIMx_CNT 进行比较并在 OC1 输出上发出信号的值。

如果通道 CC1 配置为输入：

CCR1 为上一个输入捕获 1 事件 (IC1) 发生时的计数器值。

41.6.13 TIM16/TIM17 断路和死区寄存器 (TIMx_BDTR)

TIM16/TIM17 break and dead-time register

偏移地址：0x44

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BKF[3:0]			
												rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MOE	AOE	BKP	BKE	OSSR	OSSI	LOCK[1:0]		DTG[7:0]							
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

注： *由于可以根据 LOCK 配置锁定位 BKF[3:0]、AOE、BKP、BKE、OSSI、OSSR 和 DTG[7:0] 位的写操作，因此必须在第一次对 TIMx_BDTR 寄存器执行写访问时对这些位进行配置。*

位 31:20 保留，必须保持复位值。

位 19:16 **BKF[3:0]**：断路滤波器 (Break filter)

此位域可定义 BRK 输入的采样频率和适用于 BRK 的数字滤波器带宽。数字滤波器由事件计数器组成，每 N 个事件才视为一个有效输出边沿：

0000：无滤波器，BRK 异步工作
0001：$f_{SAMPLING}=f_{CK_INT}$，N=2
0010：$f_{SAMPLING}=f_{CK_INT}$，N=4
0011：$f_{SAMPLING}=f_{CK_INT}$，N=8
0100：$f_{SAMPLING}=f_{DTS}/2$，N=6
0101：$f_{SAMPLING}=f_{DTS}/2$，N=8
0110：$f_{SAMPLING}=f_{DTS}/4$，N=6
0111：$f_{SAMPLING}=f_{DTS}/4$，N=8
1000：$f_{SAMPLING}=f_{DTS}/8$，N=6
1001：$f_{SAMPLING}=f_{DTS}/8$，N=8
1010：$f_{SAMPLING}=f_{DTS}/16$，N=5
1011：$f_{SAMPLING}=f_{DTS}/16$，N=6
1100：$f_{SAMPLING}=f_{DTS}/16$，N=8
1101：$f_{SAMPLING}=f_{DTS}/32$，N=5
1110：$f_{SAMPLING}=f_{DTS}/32$，N=6
1111：$f_{SAMPLING}=f_{DTS}/32$，N=8

编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1 后，此位即无法修改。

位 15 **MOE**：主输出使能 (Main output enable)

只要断路输入变为有效状态，此位便由硬件异步清零。此位由软件置 1，也可根据 AOE 位状态自动置 1。此位仅对配置为输出的通道有效。

0：OC 和 OCN 输出被禁止或被强制为空闲状态，具体取决于 OSSI 位。
1：如果 OC 和 OCN 输出的相应使能位（TIMx_CCER 寄存器中的 CCxE 和 CCxNE 位）均置 1，则使能 OC 和 OCN 输出。

有关详细信息，请参见 OC/OCN 使能说明（*第 1675 页的第 41.6.7 节：TIM16/TIM17 捕获/比较使能寄存器 (TIMx_CCER)*）。

位 14 **AOE**：自动输出使能 (Automatic output enable)

0：MOE 只能由软件置 1
1：MOE 可由软件置 1，也可在发生下一更新事件时自动置 1（如果断路输入无效）

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

位 13 **BKP**：断路极性 (Break polarity)

0：断路输入 BRK 为低电平有效

1：断路输入 BRK 为高电平有效

注： ***1.*** *只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1，此位即无法修改。*

2. *对该位执行任何写操作后，都需要经过 1 个 APB 时钟周期的延迟才生效。*

位 12 **BKE**：断路使能 (Break enable)

0：禁止断路输入（BRK 和 CCS 时钟故障事件）

1：使能断路输入（BRK 和 CCS 时钟故障事件）

注： ***1.*** *编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1 后，此位即无法修改。*

2. *对该位执行任何写操作后，都需要经过 1 个 APB 时钟周期的延迟才生效。*

位 11 **OSSR**：运行模式下的关闭状态选择 (Off-state selection for Run mode)

此位在 MOE = 1 时作用于配置为输出模式且具有互补输出的通道。如果定时器中没有互补输出，则不存在 OSSR。

有关详细信息，请参见 OC/OCN 使能说明（*第 1675 页的第 41.6.7 节：TIM16/TIM17 捕获/比较使能寄存器 (TIMx_CCER)*）。

0：处于无效状态时，禁止 OC/OCN 输出（定时器释放输出控制，由强制高阻态的 AFIO 逻辑接管）。

1：处于无效状态时，一旦 CCxE=1 或 CCxNE=1，便使能 OC/OCN 输出并将其设为无效电平（输出仍由定时器控制）。

注： *编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 2 后，此位即无法修改。*

位 10 **OSSI**：空闲模式下的关闭状态选择 (Off-state selection for Idle mode)

此位在 MOE=0 时作用于配置为输出的通道。

有关详细信息，请参见 OC/OCN 使能说明（*第 1675 页的第 41.6.7 节：TIM16/TIM17 捕获/比较使能寄存器 (TIMx_CCER)*）。

0：处于无效状态时，禁止 OC/OCN 输出（OC/OCN 使能输出信号 = 0）

1：处于无效状态时，一旦 CCxE = 1 或 CCxNE = 1，便将 OC/OCN 输出首先强制为其空闲电平。然后设置 OC/OCN 使能输出信号=1

注： *编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 2 后，此位即无法修改。*

位 9:8 **LOCK[1:0]**：锁定配置 (Lock configuration)

这些位用于针对软件错误提供写保护。

00：关闭锁定——不对任何位提供写保护。

01：锁定级别 1，此时无法对 TIMx_BDTR 寄存器中的 DTG 位、TIMx_CR2 寄存器中的 OISx 和 OISxN 位以及 TIMx_BDTR 寄存器中的 BKE/BKP/AOE 位执行写操作。

10：锁定级别 2，此时无法对锁定级别 1 中适用的各位、CC 极性位（TIMx_CCER 寄存器中的 CCxP/CCxNP 位，只要通过 CCxS 位将相关通道配置为输出）以及 OSSR 和 OSSI 位执行写操作。

11：锁定级别 3，此时无法对锁定级别 2 中适用的各位、CC 控制位（TIMx_CCMRx 寄存器中的 OCxM 和 OCxPE 位，只要通过 CCxS 位将相关通道配置为输出）执行写操作。

注： *复位后只能对 LOCK 位执行一次写操作。对 TIMx_BDTR 寄存器执行写操作后其中的内容将冻结，直到下一次复位。*

位 7:0 **DTG[7:0]**配置死区发生器 (Dead-time generator setup)

此位域定义插入到互补输出之间的死区持续时间。DT 与该持续时间相对应。

DTG[7:5]=0xx => DT=DTG[7:0]x t_{dtg}，其中 $t_{dtg}=t_{DTS}$。

DTG[7:5]=10x => DT=(64+DTG[5:0])xt_{dtg}，其中 $T_{dtg}=2xt_{DTS}$。

DTG[7:5]=110 => DT=(32+DTG[4:0])xt_{dtg}，其中 $T_{dtg}=8xt_{DTS}$。

DTG[7:5]=111 => DT=(32+DTG[4:0])xt_{dtg}，其中 $T_{dtg}=16xt_{DTS}$。

示例：如果 T_{DTS}=125ns (8MHz)，则可能的死区值为：

0 到 15875 ns（步长为 125 ns），

16 µs 到 31750 ns（步长为 250 ns），

32 µs 到 63 µs（步长为 1 µs），

64 µs 到 126 µs（步长为 2 µs）

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

41.6.14 TIM16/TIM17 DMA 控制寄存器 (TIMx_DCR)

TIM16/TIM17 DMA control register

偏移地址：0x48

复位值：0x0000

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

位 15:13 保留，必须保持复位值。

位 12:8 **DBL[4:0]**：DMA 连续传送长度 (DMA burst length)

该 5 位向量定义了 DMA 的传送长度（当对 TIMx_DMAR 地址进行读或写访问时，定时器进行一次连续传送），即一个 DMA 要连续传送的次数。可按半字或字节进行传送（请参见下面的示例）。

00000：1 次传送，

00001：2 次传送，

00010：3 次传送，

...

10001：18 次传送。

位 7:5 保留，必须保持复位值。

位 4:0 **DBA[4:0]**：DMA 基址 (DMA base address)

该 5 位域定义 DMA 传输的基址（通过 TIMx_DMAR 地址进行读/写访问时）。DBA 定义为从 TIMx_CR1 寄存器地址开始计算的偏移量。

示例：

00000：TIMx_CR1，

00001：TIMx_CR2，

00010：TIMx_SMCR，

...

示例：以下面的传送为例：DBL = 7 次传送且 DBA = TIMx_CR1。这种情况下将向/从自 TIMx_CR1 地址开始的 7 个寄存器传输数据。

41.6.15 TIM16/TIM17 全传输 DMA 地址 (TIMx_DMAR)

TIM16/TIM17 DMA address for full transfer

偏移地址：0x4C

复位值：0x0000

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

位 15:0 **DMAB[15:0]**：DMA 连续传送寄存器 (DMA register for burst accesses)

对 DMAR 寄存器执行读或写操作将访问位于如下地址的寄存器

(TIMx_CR1 地址) + (DBA + DMA 索引) x 4

其中 TIMx_CR1 地址为控制寄存器 1 的地址，DBA 为 TIMx_DCR 寄存器中配置的 DMA 基址，DMA 索引由 DMA 传输自动控制，其范围介于 0 到 DBL（TIMx_DCR 寄存器中配置的 DBL）之间。

41.6.16 TIM16 复用功能寄存器 1 (TIM16_AF1)

TIM16 alternate function register 1

偏移地址：0x60

复位值：0x0000 0001

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	BKCMP2P	BKCMP1P	BKINP	BKDF1BK1E	Res.	Res.	Res.	Res.	Res.	BKCMP2E	BKCMP1E	BKINE
				rw	rw	rw	rw						rw	rw	rw

位 31:12 保留，必须保持复位值。

位 11 **BKCMP2P**：BRK COMP2 输入极性 (BRK COMP2 input polarity)

此位选择 COMP2 输入灵敏度，必须与 BKP 极性位一起编程。

0：COMP2 输入为低电平有效

1：COMP2 输入为高电平有效

注： 只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1，此位即无法修改。

位 10 **BKCMP1P**：BRK COMP1 输入极性 (BRK COMP1 input polarity)

此位选择 COMP1 输入灵敏度，必须与 BKP 极性位一起编程。

0：COMP1 输入为低电平有效

1：COMP1 输入为高电平有效

注： 只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1，此位即无法修改。

位 9 **BKINP**：BRK BKIN 输入极性 (BRK BKIN input polarity)

此位选择 BKIN 复用功能输入灵敏度，必须与 BKP 极性位一起编程。

0：BKIN 输入为低电平有效

1：BKIN 输入为高电平有效

注： 只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1，此位即无法修改。

位 8 **BKDF1BK0E**：BRK dfsdm1_break[1] 使能 (BRK dfsdm1_break[1] enable)
此位使能定时器 BRK 输入的 dfsdm1_break[1]。dfsdm1_break[1] 输出与其他 BRK 源进行“或”运算。
0：禁止 dfsdm1_break[1] 输入
1：使能 dfsdm1_break[1] 输入
注： 只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1，此位即无法修改。

位 7:3 保留，必须保持复位值

位 2 **BKCMP2E**：BRK COMP2 使能 (BRK COMP2 enable)
此位使能定时器 BRK 输入的 COMP2。COMP2 输出与其他 BRK 源进行“或”运算。
0：禁止 COMP2 输入
1：使能 COMP2 输入
注： 只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1，此位即无法修改。

位 1 **BKCMP1E**：BRK COMP1 使能 (BRK COMP1 enable)
此位使能定时器 BRK 输入的 COMP1。COMP1 输出与其他 BRK 源进行“或”运算。
0：禁止 COMP1 输入
1：使能 COMP1 输入
注： 只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1，此位即无法修改。

位 0 **BKINE**：BRK BKIN 输入使能 (BRK BKIN input enable)
此位使能定时器 BRK 输入的 BKIN 复用功能。BKIN 输入与其他 BRK 源进行“或”运算。
0：禁止 BKIN 输入
1：使能 BKIN 输入
注： 只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1，此位即无法修改。

41.6.17 TIM16 输入选择寄存器 (TIM16_TISEL)

TIM16 input selection register

偏移地址：0x68

复位值：0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TI1SEL[3:0]			
												rw	rw	rw	rw

位 31:4 保留，必须保持复位值。

位 3:0 **TI1SEL[3:0]**：选择 TI1[0] 到 TI1[15] 输入 (selects TI1[0] to TI1[15] input)
0000：TIM16_CH1 输入
0001：LSI
0010：LSE
0011：WKUP_IT
其它：保留

41.6.18 TIM17 复用功能寄存器 1 (TIM17_AF1)

TIM17 alternate function register 1

偏移地址：0x60

复位值：0x0000 0001

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	BKCMP2P	BKCMP1P	BKINP	BKDF1BK2E	Res.	Res.	Res.	Res.	Res.	BKCMP2E	BKCMP1E	BKINE
				rw	rw	rw	rw						rw	rw	rw

位 31:12 保留，必须保持复位值。

位 11 **BKCMP2P**：BRK COMP2 输入极性 (BRK COMP2 input polarity)
此位选择 COMP2 输入灵敏度，必须与 BKP 极性位一起编程。
0：COMP2 输入为低电平有效
1：COMP2 输入为高电平有效
注： 只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1，此位即无法修改。

位 10 **BKCMP1P**：BRK COMP1 输入极性 (BRK COMP1 input polarity)
此位选择 COMP1 输入灵敏度，必须与 BKP 极性位一起编程。
0：COMP1 输入为低电平有效
1：COMP1 输入为高电平有效
注： 只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1，此位即无法修改。

位 9 **BKINP**：BRK BKIN 输入极性 (BRK BKIN input polarity)
此位选择 BKIN 复用功能输入灵敏度，必须与 BKP 极性位一起编程。
0：BKIN 输入为低电平有效
1：BKIN 输入为高电平有效
注： 只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1，此位即无法修改。

位 8 **BKDF1BK2E**：BRK dfsdm1_break[2] 使能 (BRK dfsdm1_break[2] enable)
此位使能定时器 BRK 输入的 dfsdm1_break[2]。dfsdm1_break[2] 输出与其他 BRK 源进行“或”运算。
0：禁止 dfsdm1_break[2] 输入
1：使能 dfsdm1_break[2] 输入
注： 只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1，此位即无法修改。

位 7:3 保留，必须保持复位值

位 2 **BKCMP2E**：BRK COMP2 使能 (BRK COMP2 enable)
此位使能定时器 BRK 输入的 COMP2。COMP2 输出与其他 BRK 源进行“或”运算。
0：禁止 COMP2 输入
1：使能 COMP2 输入
注： 只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1，此位即无法修改。

位 1 **BKCMP1E**：BRK COMP1 使能 (BRK COMP1 enable)
此位使能定时器 BRK 输入的 COMP1。COMP1 输出与其他 BRK 源进行“或”运算。
0：禁止 COMP1 输入
1：使能 COMP1 输入
注： 只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1，此位即无法修改。

位 0 **BKINE**：BRK BKIN 输入使能 (BRK BKIN input enable)
此位使能定时器 BRK 输入的 BKIN 复用功能。BKIN 输入与其他 BRK 源进行“或”运算。
0：禁止 BKIN 输入
1：使能 BKIN 输入
注： 只要编程了 LOCK（TIMx_BDTR 寄存器中的 LOCK 位）级别 1，此位即无法修改。

41.6.19 TIM17 输入选择寄存器 (TIM17_TISEL)

TIM17 input selection register

偏移地址：0x68

复位值：0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TI1SEL[3:0]			
												rw	rw	rw	rw

位 31:4 保留，必须保持复位值。

位 3:0 **TI1SEL[3:0]**：选择 TI1[0] 到 TI1[15] 输入 (selects TI1[0] to TI1[15] input)
0000：TIM17_CH1 输入
0001：SPDIF FS
0010：HSE_1MHz
0011：MCO1
其它：保留

41.6.20 TIM16/TIM17 寄存器映射

TIM16/TIM17寄存器可映射为 16 位可寻址寄存器，如下表所述：

表 329. TIM16/TIM17 寄存器映射和复位值

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	**TIMx_CR1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	UIFREMAP	Res.	CKD [1:0]		ARPE	Res.	Res.	Res.	OPM	URS	UDIS	CEN
	Reset value																					0		0	0	0				0	0	0	0
0x04	**TIMx_CR2**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OIS1N	OIS1	Res.	Res.	Res.	Res.	CCDS	CCUS	Res.	CCPC
	Reset value																							0	0					0	0		0
0x0C	**TIMx_DIER**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	COMDE	Res.	Res.	Res.	CC1DE	UDE	BIE	Res.	COMIE	Res.	Res.	Res.	CC1IE	UIE
	Reset value																			0				0	0	0		0				0	0
0x10	**TIMx_SR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CC1OF	Res.	BIF	Res.	COMIF	Res.	Res.	Res.	CC1IF	UIF
	Reset value																							0		0		0				0	0
0x14	**TIMx_EGR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BG	Res.	COMG	Res.	Res.	Res.	CC1G	UG
	Reset value																									0		0				0	0
0x18	**TIMx_CCMR1** Output Compare mode	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC1M[3]	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OC1M [2:0]			OC1PE	OC1FE	CC1S [1:0]	
	Reset value																0										0	0	0	0	0	0	0
	TIMx_CCMR1 Input Capture mode	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	IC1F[3:0]				IC1PSC [1:0]		CC1S [1:0]	
	Reset value																									0	0	0	0	0	0	0	0
0x20	**TIMx_CCER**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CC1NP	CC1NE	CC1P	CC1E
	Reset value																													0	0	0	0
0x24	**TIMx_CNT**	UIFCPY or Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CNT[15:0]															
	Reset value	0																0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x28	**TIMx_PSC**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PSC[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x2C	**TIMx_ARR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ARR[15:0]															
	Reset value																	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

表 329. TIM16/TIM17 寄存器映射和复位值（续）

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x30	**TIMx_RCR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REP[7:0]							
	Reset value																									0	0	0	0	0	0	0	0
0x34	**TIMx_CCR1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CCR1[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x44	**TIMx_BDTR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BKF[3:0]				MOE	AOE	BKP	BKE	OSSR	OSSI	LOCK [1:0]		DT[7:0]							
	Reset value													0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x48	**TIMx_DCR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DBL[4:0]					Res.	Res.	Res.	DBA[4:0]				
	Reset value																				0	0	0	0	0				0	0	0	0	0
0x4C	**TIMx_DMAR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DMAB[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x60	**TIM16_AF1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BKCMP2P	BKCMP1P	BKINP	BKDF1BK1E	Res.	Res.	Res.	Res.	Res.	BKCMP2E	BKCMP1E	BKINE
	Reset value																					0	0	0	0						0	0	1
0x60	**TIM17_AF1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BKCMP2P	BKCMP1P	BKINP	BKDF1BK2E	Res.	Res.	Res.	Res.	Res.	BKCMP2E	BKCMP1E	BKINE
	Reset value																					0	0	0	0						0	0	1
0x68	**TIM16_TISEL**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TISEL[3:0]			
	Reset value																													0	0	0	0
0x68	**TIM17_TISEL**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TISEL[3:0]			
	Reset value																													0	0	0	0

有关寄存器边界地址的信息，请参见*第 100 页的第 2.2.2 节*。

42 基本定时器 (TIM6/TIM7)

42.1 TIM6/TIM7 简介

基本定时器 TIM6 和 TIM7 包含一个 16 位自动重载计数器，该计数器由可编程预分频器驱动。

此类定时器不仅可用作通用定时器以生成时基，还可以专门用于驱动数模转换器 (DAC)。实际上，此类定时器内部连接到 DAC 并能够通过其触发输出驱动 DAC。

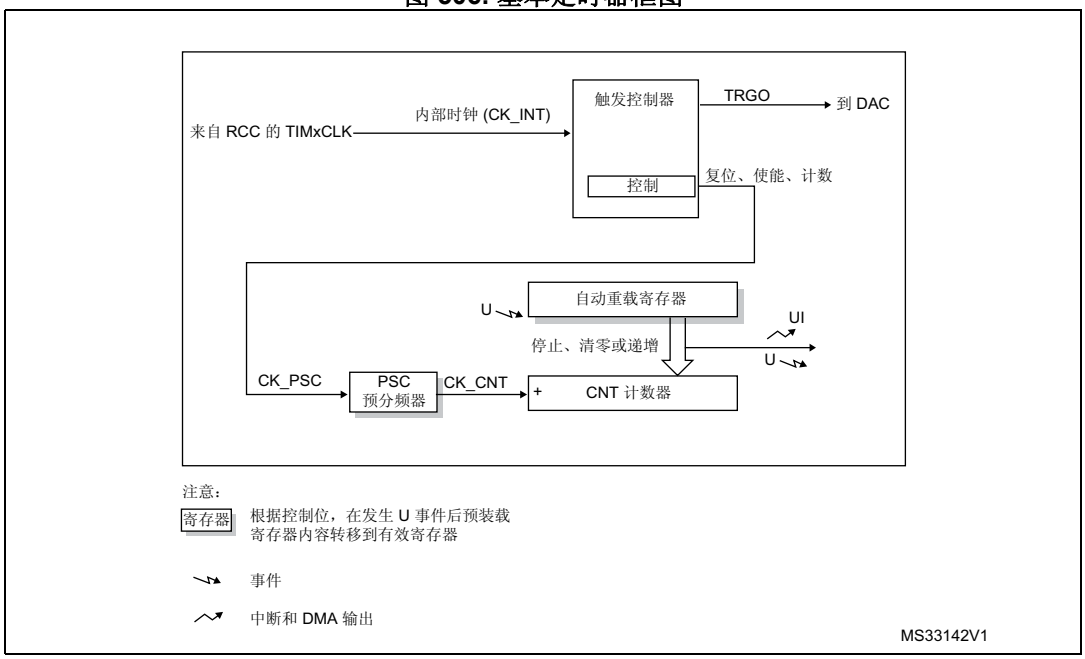
这些定时器彼此完全独立，不共享任何资源。

42.2 TIM6/TIM7 主要特性

基本定时器 (TIM6/TIM7) 的特性包括：

- 16 位自动重载递增计数器
- 16 位可编程预分频器，用于对计数器时钟频率进行分频（可在运行时修改），分频系数介于 1 和 65535 之间
- 用于触发 DAC 的同步电路
- 发生如下更新事件时会生成中断/DMA 请求：计数器上溢

图 506. 基本定时器框图



42.3 TIM6/TIM7 功能说明

42.3.1 时基单元

可编程定时器的主要模块由一个 16 位递增计数器及其相关的自动重载寄存器组成。计数器的时钟可通过预分频器进行分频。

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

时基单元包括：

- 计数器寄存器 (TIMx_CNT)
- 预分频器寄存器 (TIMx_PSC)
- 自动重载寄存器 (TIMx_ARR)

自动重载寄存器是预装载的。每次尝试对自动重载寄存器执行读写操作时，都会访问预装载寄存器。预装载寄存器的内容既可以立即传送到影子寄存器，也可以在每次发生更新事件 UEV 时传送到影子寄存器，这取决于 TIMx_CR1 寄存器中的自动重载预装载使能位 (ARPE)。当计数器达到上溢值并且 TIMx_CR1 寄存器中的 UDIS 位为 0 时，将发送更新事件。该更新事件也可由软件产生。下文将针对各配置的更新事件的产生进行详细介绍。

计数器由预分频器输出 CK_CNT 提供时钟，仅当 TIMx_CR1 寄存器中的计数器启动位 (CEN) 置 1 时，才会启动计数器。

请注意，实际的计数器使能信号 CNT_EN 在 CEN 置 1 的一个时钟周期后被置 1。

预分频器说明

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

图 507 和 *图 508* 以一些示例说明在预分频比实时变化时计数器的行为。

图 507. 预分频器分频由 1 变为 2 时的计数器时序图

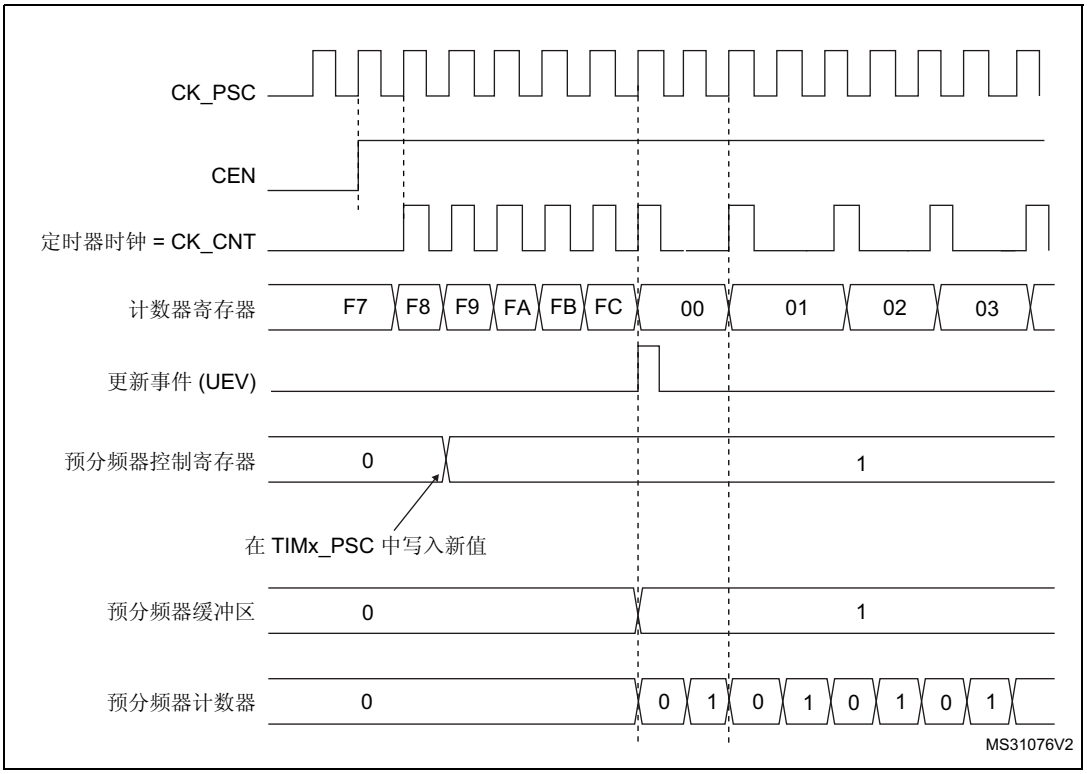
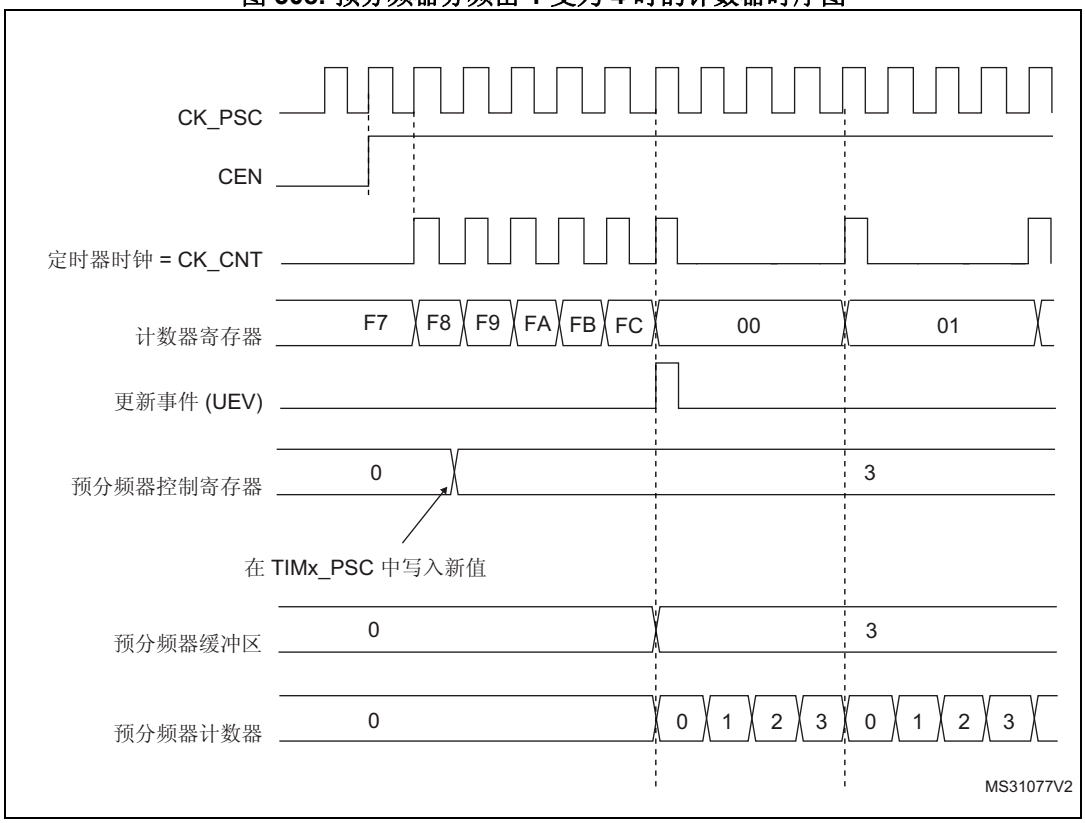


图 508. 预分频器分频由 1 变为 4 时的计数器时序图



42.3.2 计数模式

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

每次发生计数器上溢时会生成更新事件，或将 TIMx_EGR 寄存器中的 UG 位置 1（通过软件或使用从模式控制器）也可以产生更新事件。

通过软件将 TIMx_CR1 寄存器中的 UDIS 位置 1 可禁止 UEV 事件。这可避免向预装载寄存器写入新值时更新影子寄存器。这样，直到 UDIS 位中写入 0 前便不会生成任何更新事件，但计数器和预分频器计数器都会重新从 0 开始计数（而预分频比保持不变）。此外，如果 TIMx_CR1 寄存器中的 URS 位（更新请求选择）已置 1，则将 UG 位置 1 会生成更新事件 UEV，但不会将 UIF 标志置 1（因此，不会发送任何中断或 DMA 请求）。

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

- 预分频器的缓冲区中将重新装载预装载值（TIMx_PSC 寄存器的内容）
- 使用预装载值 (TIMx_ARR) 更新自动重载影子寄存器

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

图 509. 计数器时序图，1 分频内部时钟

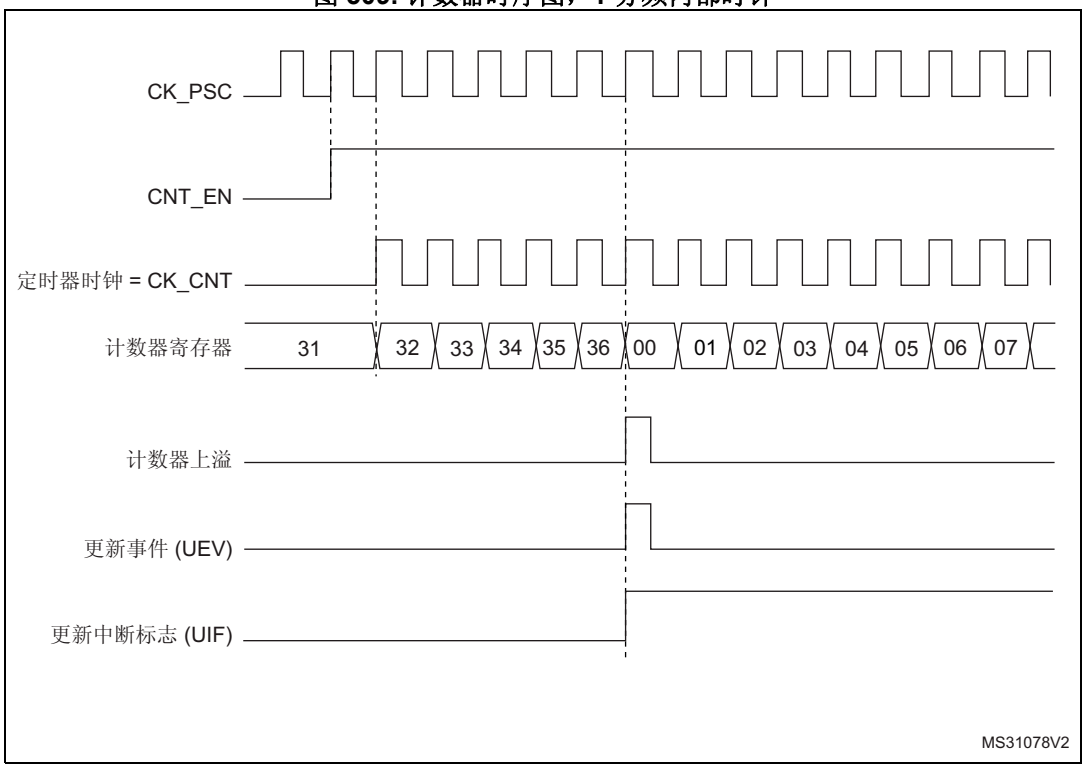


图 510. 计数器时序图，2 分频内部时钟

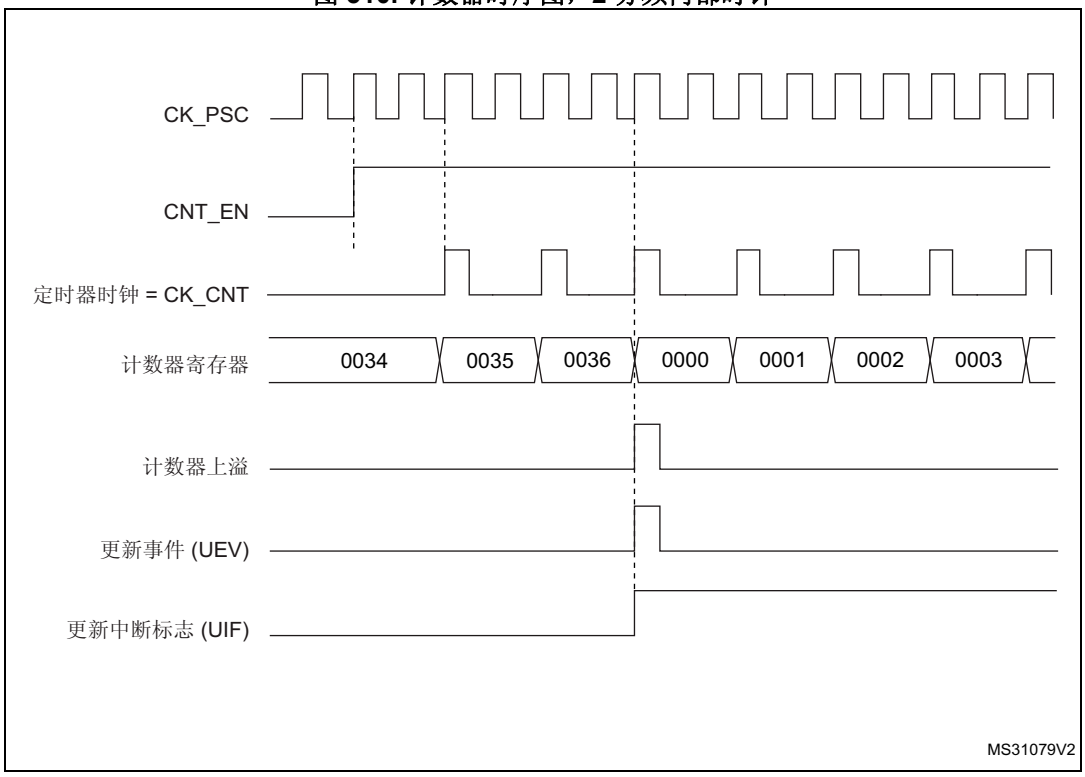


图 511. 计数器时序图，4 分频内部时钟

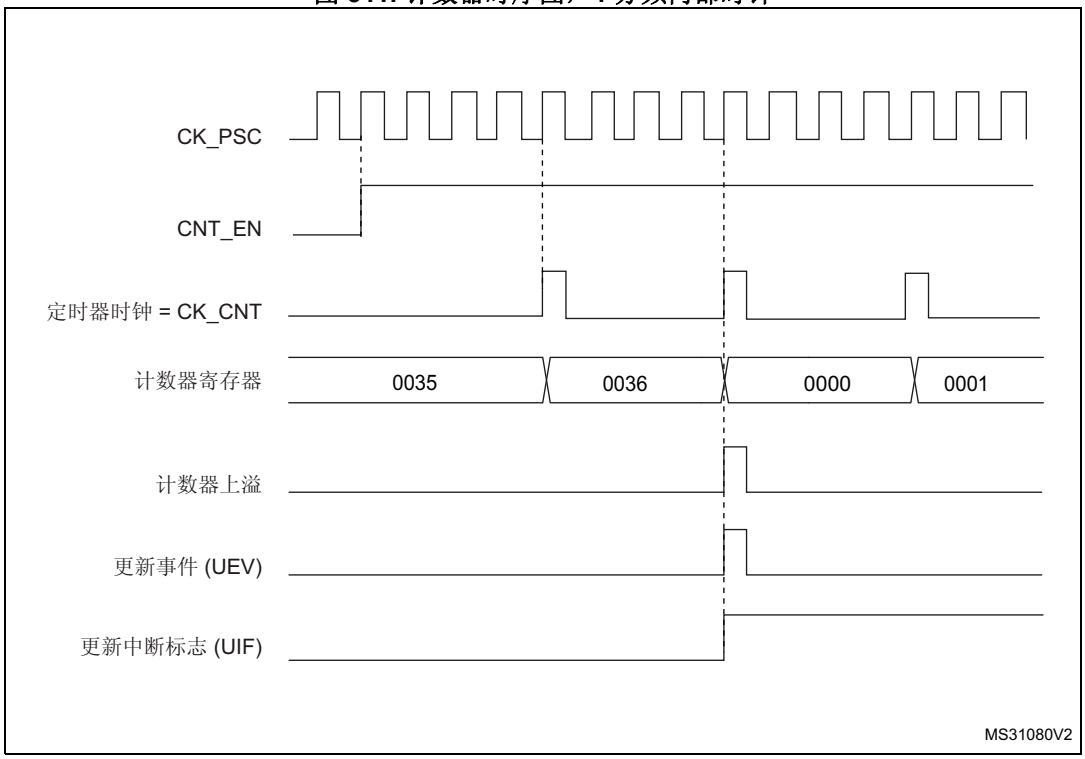


图 512. 计数器时序图，N 分频内部时钟

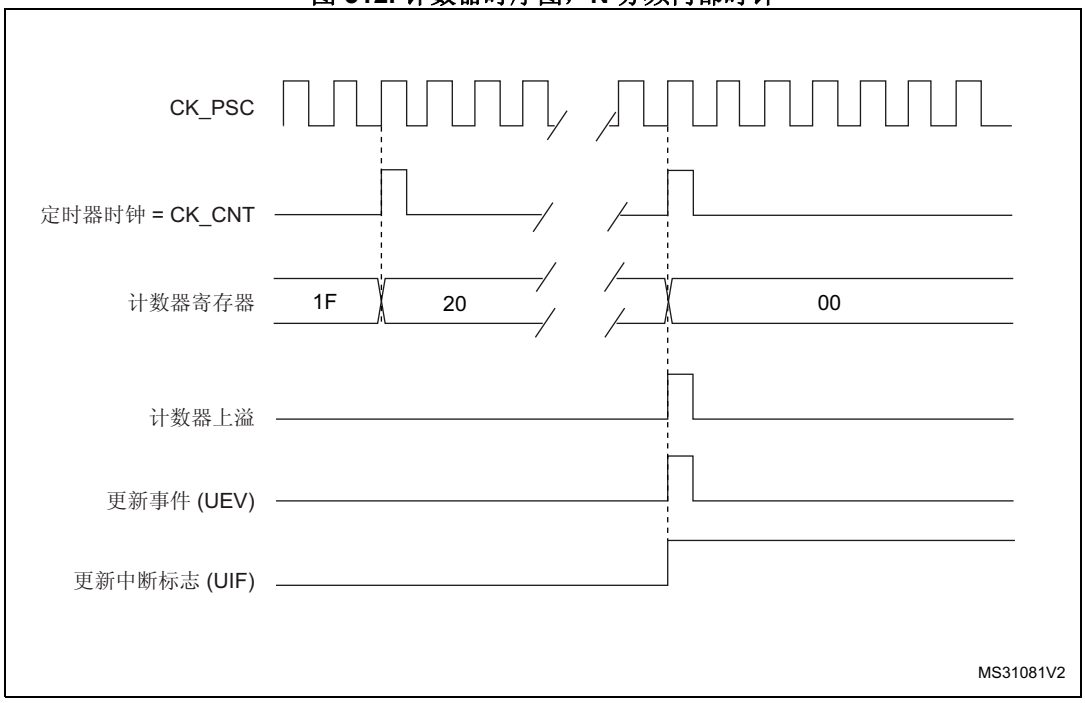


图 513. 计数器时序图，ARPE = 0 时更新事件（TIMx_ARR 未预装载）

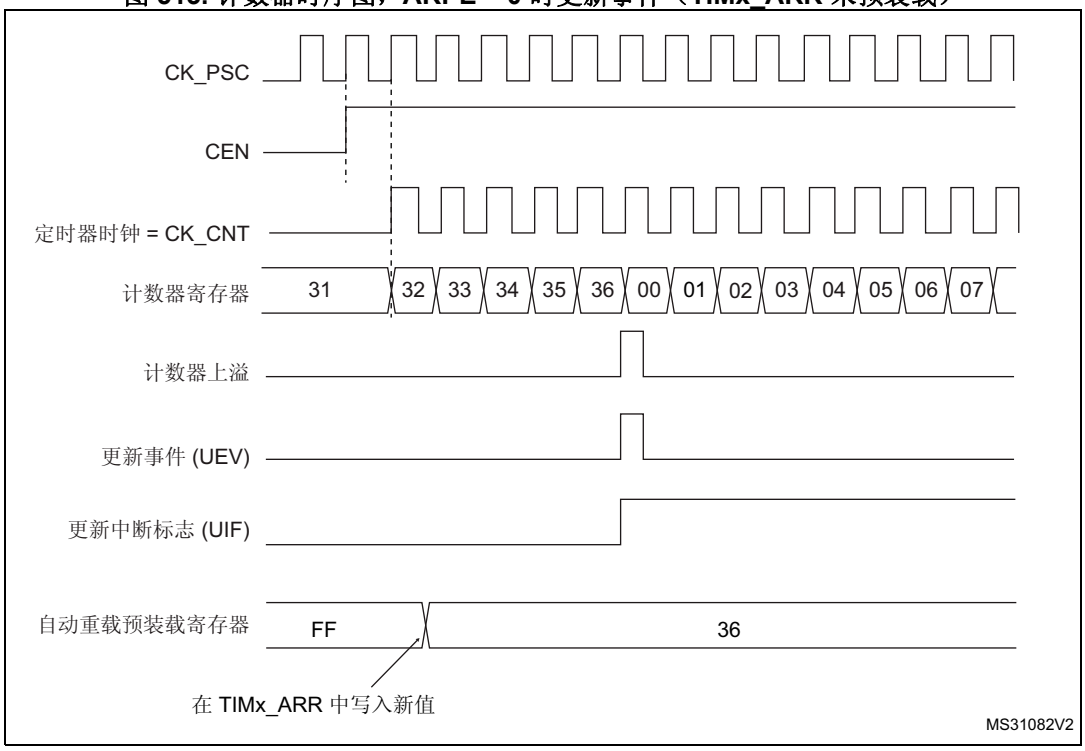
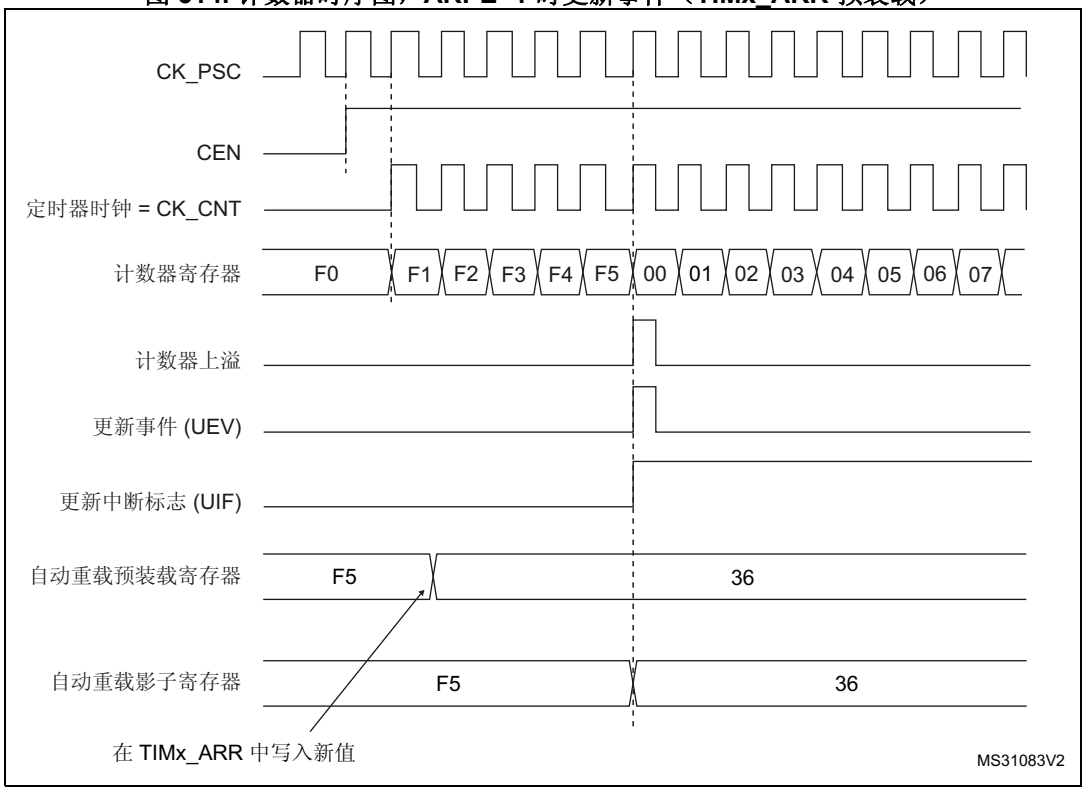


图 514. 计数器时序图，ARPE=1 时更新事件（TIMx_ARR 预装载）



42.3.3 UIF 位重映射

TIMx_CR1 寄存器中的 IUFREMAP 位强制将更新中断标志 UIF 连续复制到定时器计数器寄存器的位 31 (TIMxCNT[31]) 中。这样便可自动读取计数器值以及由 UIFCPY 标志发出的电位翻转条件。在特定情况下，这可避免在后台任务（计数器读）和中断（更新中断）之间共享处理时产生竞争条件，从而简化计算。

UIF 和 UIFCPY 标志使能之间没有延迟。

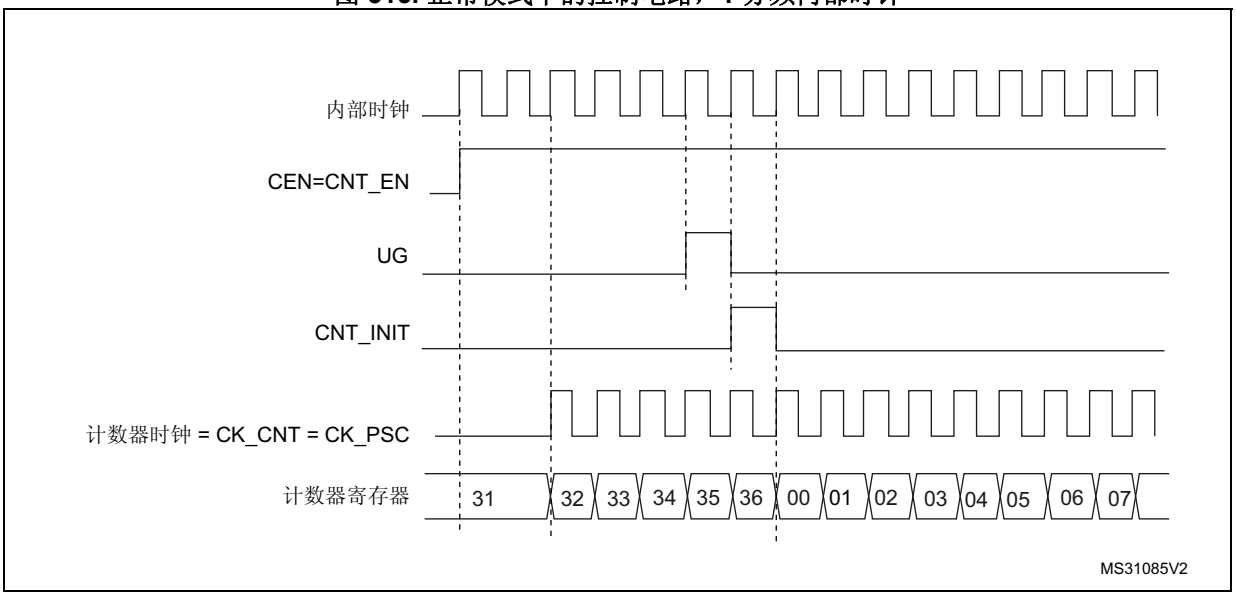
42.3.4 时钟源

计数器时钟由内部时钟 (CK_INT) 源提供。

CEN（TIMx_CR1 寄存器中）和 UG 位（TIMx_EGR 寄存器中）为实际控制位，并且只能通过软件进行更改（除了 UG 位被自动清零外）。当对 CEN 位写入 1 时，预分频器的时钟就由内部时钟 CK_INT 提供。

图 515 显示了正常模式下控制电路与递增计数器的行为（没有预分频的情况下）。

图 515. 正常模式下的控制电路，1 分频内部时钟



42.3.5 调试模式

当微控制器进入调试模式时（Cortex®-M7 with FPU 内核停止），TIMx 计数器会根据 DBGMCU 模块中的 DBG_TIMx_STOP 配置位选择继续正常工作或者停止工作。有关详细信息，请参见*第 60.5.8 节：微控制器调试单元 (DBGMCU)*。

42.4 TIM6/TIM7 寄存器

有关寄存器说明中使用的缩写，请参见*第 94 页的第 1.1 节*。

外设寄存器可支持半字（16 位）或字（32 位）访问。

42.4.1 TIM6/TIM7 控制寄存器 1 (TIMx_CR1)

TIM6/TIM7 control register 1

偏移地址：0x00

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res	Res	Res	Res	UIF RE-MAP	Res	Res	Res	ARPE	Res	Res	Res	OPM	URS	UDIS	CEN
				rw				rw				rw	rw	rw	rw

位 15:12 保留，必须保持复位值。

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

0：无重映射。UIF 状态位不复制到 TIMx_CNT 寄存器的位 31。

1：使能重映射。UIF 状态位复制到 TIMx_CNT 寄存器的位 31。

位 10:8 保留，必须保持复位值。

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

0：TIMx_ARR 寄存器不进行缓冲。

1：TIMx_ARR 寄存器进行缓冲。

位 6:4 保留，必须保持复位值。

位 3 **OPM**：单脉冲模式 (One-pulse mode)

0：计数器在发生更新事件时不会停止计数。

1：计数器在发生下一更新事件时停止计数（将 CEN 位清零）。

位 2 **URS**：更新请求源 (Update request source)

此位由软件置 1 和清零，用以选择 UEV 事件源。

0：使能时，所有以下事件都会产生更新中断或 DMA 请求。此类事件包括：

– 计数器上溢/下溢

– 将 UG 位置 1

– 通过从模式控制器生成的更新事件

1：使能时，只有计数器上溢/下溢会生成更新中断或 DMA 请求。

位 1 **UDIS**：更新禁止 (Update disable)

此位由软件置 1 和清零，用以使能/禁止 UEV 事件生成。

0：使能 UEV。更新 (UEV) 事件可通过以下事件之一生成：

– 计数器上溢/下溢

– 将 UG 位置 1

– 通过从模式控制器生成的更新事件

然后带缓冲的寄存器被加载为预加载数值。

1：禁止 UEV。不会生成更新事件，各影子寄存器的值（ARR 和 PSC）保持不变。但如果将 UG 位置 1，或者从从模式控制器接收到硬件复位，则会重新初始化计数器和预分频器。

位 0 **CEN**：计数器使能 (Counter enable)

0：禁止计数器

1：使能计数器

注： 只有事先通过软件将 CEN 位置 1，才可以使用门控模式。而触发模式可通过硬件自动将 CEN 位置 1。

在单脉冲模式下，当发生更新事件时会自动将 CEN 位清零。

42.4.2 TIM6/TIM7 控制寄存器 2 (TIMx_CR2)

TIM6/TIM7 control register 2

偏移地址：0x04

复位值：0x0000

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

位 15:7 保留，必须保持复位值。

位 6:4 **MMS**：主模式选择 (Master mode selection)

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

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

001：**使能**——计数器使能信号 CNT_EN 用作触发输出 (TRGO)。该触发输出可用于同时启动多个定时器，或者控制在一段时间内使能从定时器。计数器使能信号由 CEN 控制位与门控模式下的触发输入的逻辑或运算组合而成。

当计数器使能信号由触发输入控制时，TRGO 上会存在延迟，选择主/从模式时除外（请参见 TIMx_SMCR 寄存器中对 MSM 位的说明）。

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

注： 必须先使能从定时器或 ADC 的时钟，才能从主定时器接收事件；并且从主定时器接收触发信号时，不得实时更改从定时器或 ADC 的时钟。

位 3:0 保留，必须保持复位值。

42.4.3 TIM6/TIM7 DMA/中断使能寄存器 (TIMx_DIER)

TIM6/TIM7 DMA/Interrupt enable register

偏移地址：0x0C

复位值：0x0000

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

位 15:9 保留，必须保持复位值。

位 8 **UDE**：更新 DMA 请求使能 (Update DMA request enable)

0：禁止更新 DMA 请求。
1：使能更新 DMA 请求。

位 7:1 保留，必须保持复位值。

位 0 **UIE**：更新中断使能 (Update interrupt enable)

0：禁止更新中断。
1：使能更新中断。

42.4.4 TIM6/TIM7 状态寄存器 (TIMx_SR)

TIM6/TIM7 status register

偏移地址：0x10

复位值：0x0000

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	UIF
															rc_w0

位 15:1 保留，必须保持复位值。

位 0 **UIF**：更新中断标志 (Update interrupt flag)

该位在发生更新事件时通过硬件置 1。但需要通过软件清零。

0：未发生更新。

1：更新中断挂起。该位在以下情况下更新寄存器时由硬件置 1：

– 重复计数器值上溢或下溢并且 TIMx_CR1 寄存器中的 UDIS=0 时。

– 如果 TIMx_CR1 寄存器中 URS = 0 且 UDIS = 0，通过软件使用 TIMx_EGR 寄存器中的 UG 位重新初始化 CNT 时。

42.4.5 TIM6/TIM7 事件产生寄存器 (TIMx_EGR)

TIM6/TIM7 event generation register

偏移地址：0x14

复位值：0x0000

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

位 15:1 保留，必须保持复位值。

位 0 **UG**：更新生成 (Update generation)

该位可通过软件置 1，并由硬件自动清零。

0：不执行任何操作。

1：重新初始化定时器计数器并生成寄存器更新事件。请注意，预分频器计数器也将清零（但预分频比不受影响）。

42.4.6 TIM6/TIM7 计数器 (TIMx_CNT)

TIM6/TIM7 counter

偏移地址：0x24

复位值：0x0000

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

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

位 31 **UIFCPY**：UIF 副本 (UIF Copy)
该位是 TIMx_ISR 寄存器中 UIF 位的只读副本。如果 TIMx_CR1 中的 UIFREMAP 位复位，则位 31 保留，读为 0。

位 30:16 保留，必须保持复位值。

位 15:0 **CNT[15:0]**：计数器值 (Counter value)

42.4.7 TIM6/TIM7 预分频器 (TIMx_PSC)

TIM6/TIM7 prescaler

偏移地址：0x28

复位值：0x0000

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

位 15:0 **PSC[15:0]**：预分频器值 (Prescaler value)
计数器时钟频率 CK_CNT 等于 f_{CK_PSC} / (PSC[15:0] + 1)。
PSC 包含在每次发生更新事件时要装载到实际预分频器寄存器的值。
（包括当计数器通过 TIMx_EGR 寄存器的 UG 位或在“复位模式”下通过触发控制器清零时）。

42.4.8 TIM6/TIM7 自动重载寄存器 (TIMx_ARR)

TIM6/TIM7 auto-reload register

偏移地址：0x2C

复位值：0xFFFF

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

位 15:0 **ARR[15:0]**：预分频器值 (Prescaler value)
ARR 为要装载到实际自动重载寄存器的值。
有关 ARR 更新和行为的详细信息，请参见*第 1690 页的第 42.3.1 节：时基单元*。
当自动重载值为空时，计数器不工作。

42.4.9 TIM6/TIM7 寄存器映射

TIMx 寄存器可映射为 16 位可寻址寄存器，如下表所述：

表 330. TIM6/TIM7 寄存器映射和复位值

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	**TIMx_CR1**	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	UIFREMAP	Res	Res	Res	ARPE	Res	Res	Res	OPM	URS	UDIS	CEN
	Reset value																					0				0				0	0	0	0
0x04	**TIMx_CR2**	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	MMS [2:0]			Res	Res	Res	Res
	Reset value																										0	0	0				
0x08	Reserved																																
0x0C	**TIMx_DIER**	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	UDE	Res	Res	Res	Res	Res	Res	Res	UIE
	Reset value																								0								0
0x10	**TIMx_SR**	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	UIF
	Reset value																																0
0x14	**TIMx_EGR**	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	UG
	Reset value																																0
0x18-0x20	Reserved																																
0x24	**TIMx_CNT**	UIFCPY or Res.	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	CNT[15:0]															
	Reset value	0																0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x28	**TIMx_PSC**	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	PSC[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x2C	**TIMx_ARR**	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	ARR[15:0]															
	Reset value																	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

有关寄存器边界地址的信息，请参见*第 2.2.2 节：存储器映射和寄存器边界地址*。

43 低功耗定时器 (LPTIM)

43.1 前言

LPTIM 是一个 16 位定时器，可从降低功耗的最终发展中受益。由于 LPTIM 的时钟源具有多样性，因此 LPTIM 能够在所有电源模式（待机模式除外）下保持运行状态。即使没有内部时钟源，LPTIM 也能运行，鉴于这一点，可将其用作“脉冲计数器”，这种脉冲计数器在某些应用中十分有用。此外，LPTIM 还能将系统从低功耗模式唤醒，因此非常适合实现“超时功能”，而且功耗极低。

LPTIM 引入了一个灵活的时钟方案，该方案能够提供所需的功能和性能，同时还能最大程度地降低功耗。

43.2 LPTIM 主要特性

- 16 位递增计数器
- 3 位预分频器，可采用 8 种分频系数（1、2、4、8、16、32、64 和 128）
- 可选时钟
  - 内部时钟源：LSE、LSI、HSI 或 APB 时钟
  - LPTIM 输入的外部时钟源（在没有 LP 振荡器运行的情况下工作，可在使用脉冲计数器应用场景中使用）
- 16 位 ARR 自动重载寄存器
- 16 位比较寄存器
- 连续/单触发模式
- 可选软件/硬件输入触发
- 可编程数字防抖动干扰滤波器
- 可配置输出：脉冲和 PWM
- 可配置 I/O 极性
- 编码器模式

43.3 LPTIM 实现

表 331 介绍了 STM32H7x3 器件上的 LPTIM 实现。

表 331. STM32H7x3 LPTIM 特性

LPTIM 模式/特性(1)	LPTIM1	LPTIM2	LPTIM3	LPTIM4	LPTIM5
编码器模式	X	X	-	-	-

1. X = 支持。

43.4 LPTIM 功能说明

43.4.1 LPTIM 框图

图 516. 低功耗定时器框图（LPTIM1 和 LPTIM2）

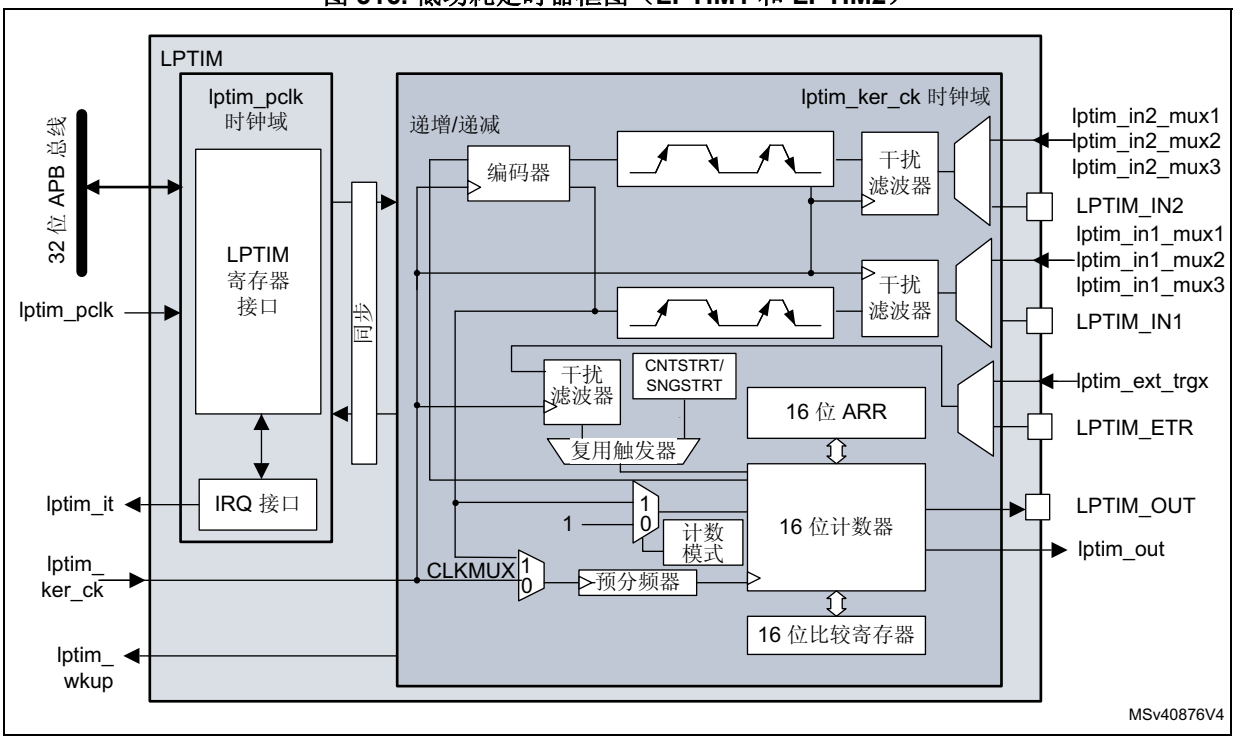


图 517. 低功耗定时器框图 (LPTIM3)

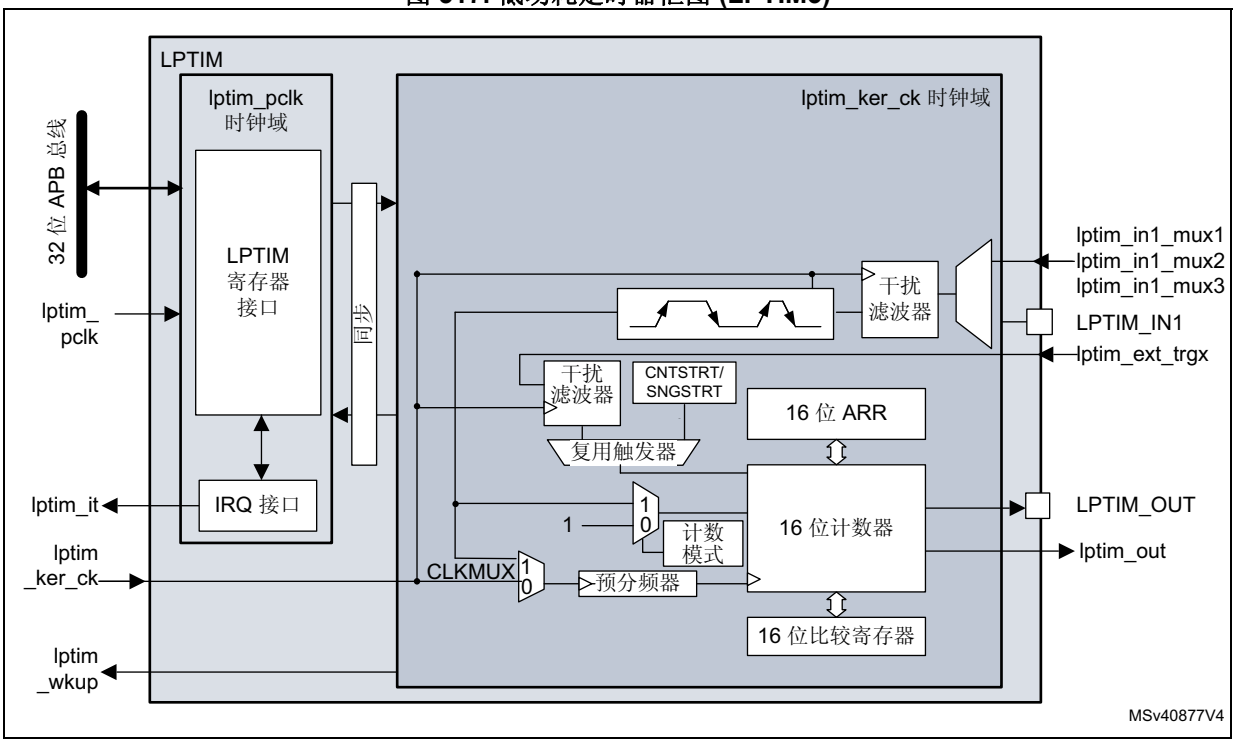
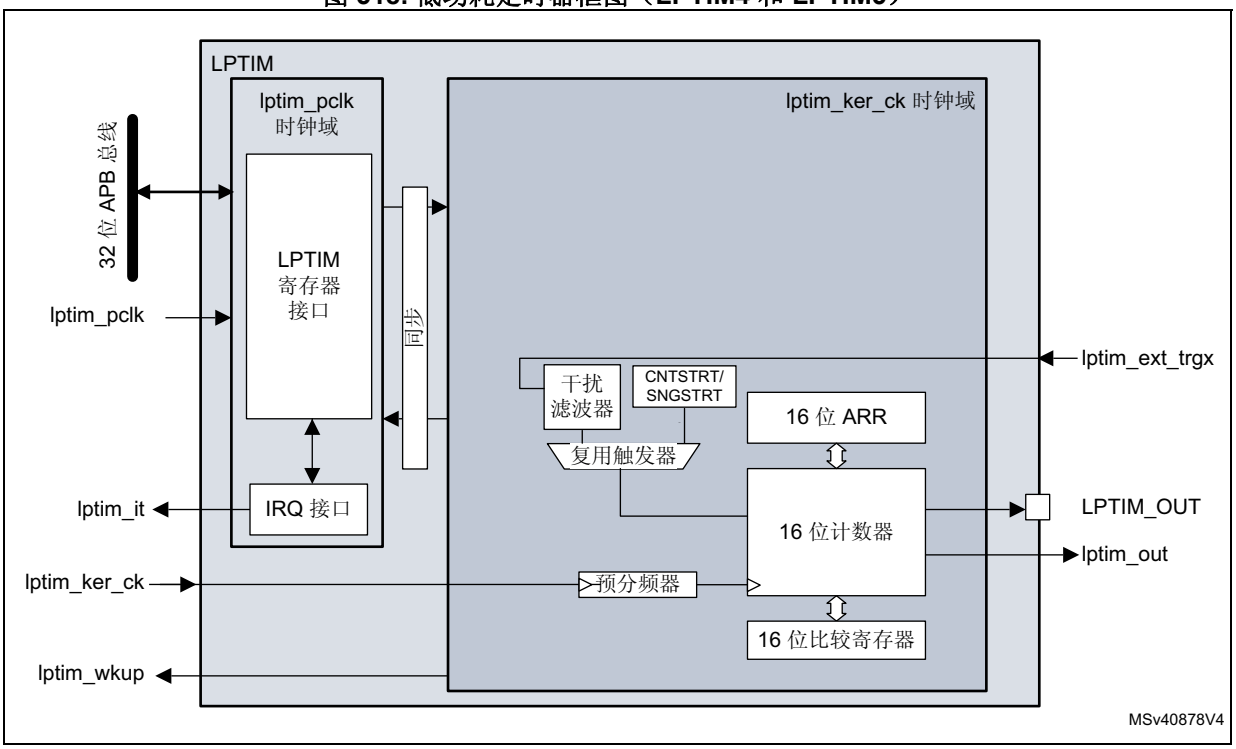


图 518. 低功耗定时器框图（LPTIM4 和 LPTIM5）



43.4.2 LPTIM 引脚和内部信号

表 333 和*表 332* 分别列出了 LPTIM 内部信号和引脚。

表 332. LPTIM 输入/输出引脚

名称	信号类型	说明
LPTIM_IN1	数字输入	复用器输入 0 上 GPIO 引脚的 LPTIM 输入 1
LPTIM_IN2	数字输入	复用器输入 0 上 GPIO 引脚的 LPTIM 输入 2
LPTIM_ETR	数字输入	LPTIM 外部触发 GPIO 引脚
LPTIM_OUT	数字输出	LPTIM 输出 GPIO 引脚

表 333. LPTIM 内部信号

名称	信号类型	说明
lptim_pclk	数字输入	LPTIM APB 时钟域
lptim_ker_ck	数字输入	LPTIM 内核时钟
lptim_in1_mux1	数字输入	连接至复用器输入 1 的内部 LPTIM 输入 1
lptim_in1_mux2	数字输入	连接至复用器输入 2 的内部 LPTIM 输入 1
lptim_in1_mux3	数字输入	连接至复用器输入 3 的内部 LPTIM 输入 1
lptim_in2_mux1	数字输入	连接至复用器输入 1 的内部 LPTIM 输入 2
lptim_in2_mux2	数字输入	连接至复用器输入 2 的内部 LPTIM 输入 2
lptim_in2_mux3	数字输入	连接至复用器输入 3 的内部 LPTIM 输入 2
lptim_ext_trgx	数字输入	LPTIM 外部触发输入 x
lptim_out	数字输出	LPTIM 计数器输出
lptim_it	数字输出	LPTIM 全局中断
lptim_wakeup	数字输出	LPTIM 唤醒事件

43.4.3 LPTIM 输入和触发映射

表 334 至*表 338* 显示了 LPTIM 外部触发连接，而*表 339* 至*表 343* 则显示了 LPTIM 输入 1 和输入 2 连接。

表 334. LPTIM1 外部触发连接

TRIGSEL	外部触发信号
lptim_ext_trig0	用作 LPTIM1_ETR 复用功能的 GPIO 引脚
lptim_ext_trig1	RTC_ALARMA
lptim_ext_trig2	RTC_ALARMB
lptim_ext_trig3	RTC_TAMP1_OUT
lptim_ext_trig4	RTC_TAMP2_OUT
lptim_ext_trig5	RTC_TAMP3_OUT

表 334. LPTIM1 外部触发连接

TRIGSEL	外部触发信号
lptim_ext_trig6	COMP1_OUT
lptim_ext_trig7	COMP2_OUT

表 335. LPTIM2 外部触发连接

TRIGSEL	外部触发信号
lptim_ext_trig0	用作 LPTIM2_ETR 复用功能的 GPIO 引脚
lptim_ext_trig1	RTC_ALARMA
lptim_ext_trig2	RTC_ALARMB
lptim_ext_trig3	RTC_TAMP1_OUT
lptim_ext_trig4	RTC_TAMP2_OUT
lptim_ext_trig5	RTC_TAMP3_OUT
lptim_ext_trig6	COMP1_OUT
lptim_ext_trig7	COMP2_OUT

表 336. LPTIM3 外部触发连接

TRIGSEL	外部触发信号
lptim_ext_trig0	LPTIM2_OUT
lptim_ext_trig1	未连接
lptim_ext_trig2	LPTIM4_OUT
lptim_ext_trig3	LPTIM5_OUT
lptim_ext_trig4	SAI1_FS_A
lptim_ext_trig5	SAI1_FS_B
lptim_ext_trig6	未连接
lptim_ext_trig7	未连接

表 337. LPTIM4 外部触发连接

TRIGSEL	外部触发信号
lptim_ext_trig0	LPTIM2_OUT
lptim_ext_trig1	LPTIM3_OUT
lptim_ext_trig2	未连接
lptim_ext_trig3	LPTIM5_OUT
lptim_ext_trig4	SAI2_FS_A
lptim_ext_trig5	SAI2_FS_B

表 337. LPTIM4 外部触发连接

TRIGSEL	外部触发信号
lptim_ext_trig6	未连接
lptim_ext_trig7	未连接

表 338. LPTIM5 外部触发连接

TRIGSEL	外部触发信号
lptim_ext_trig0	LPTIM2_OUT
lptim_ext_trig1	LPTIM3_OUT
lptim_ext_trig2	LPTIM4_OUT
lptim_ext_trig3	SAI4_FS_A
lptim_ext_trig4	SAI4_FS_B
lptim_ext_trig5	未连接
lptim_ext_trig6	未连接
lptim_ext_trig7	未连接

表 339. LPTIM1 输入 1 连接

lptim_in1_mux	LPTIM1 输入 1 连接位置
lptim_in1_mux0	用作 LPTIM1_IN1 复用功能的 GPIO 引脚
lptim_in1_mux1	COMP1_OUT
lptim_in1_mux2	未连接
lptim_in1_mux3	未连接

表 340. LPTIM1 输入 2 连接

lptim_in2_mux	LPTIM1 输入 2 连接位置
lptim_int2_mux0	用作 LPTIM1_IN2 复用功能的 GPIO 引脚
lptim_in2_mux1	COMP2_OUT
lptim_in2_mux2	未连接
lptim_in2_mux3	未连接

表 341. LPTIM2 输入 1 连接

lptim_in1_mux	LPTIM2 输入 1 连接位置
lptim_in1_mux0	用作 LPTIM2_IN1 复用功能的 GPIO 引脚
lptim_in1_mux1	COMP1_OUT
lptim_in1_mux2	COMP2_OUT
lptim_in1_mux3	COMP1_OUT 或 COMP2_OUT

表 342. LPTIM2 输入 2 连接

lptim_in2_mux	LPTIM2 输入 2 连接位置
lptim_int2_mux0	用作 LPTIM2_IN2 复用功能的 GPIO 引脚
lptim_in2_mux1	COMP2_OUT
lptim_in2_mux2	未连接
lptim_in2_mux3	未连接

表 343. LPTIM3 输入 1 连接

lptim_in1_mux	LPTIM3 输入 1 连接位置
lptim_in1_mux0	未连接
lptim_in1_mux1	SAI4_FS_A
lptim_in1_mux2	SAI4_FS_B
lptim_in1_mux3	未连接

43.4.4 LPTIM 复位和时钟

LPTIM 可通过多个时钟源提供时钟。它可以由内部时钟信号提供时钟，内部时钟信号可通过复位和时钟控制器 (RCC) 在 APB、LSI、LSE 或 HSI 时钟源中进行选择。此外，LPTIM 还可通过注入到其外部 Input1 上的外部时钟信号提供时钟。当通过外部时钟源提供时钟时，LPTIM 可以在下述两种可能配置中的其中一种配置下运行：

- 第一种配置是，LPTIM 通过外部信号提供时钟，但同时通过 APB 或 LSE、LSI 和 HSI 等任何其他内置振荡器为 LPTIM 提供内部时钟信号。
- 第二种配置是，LPTIM 仅由外部时钟源通过外部 Input1 提供时钟。此配置可在进入低功耗模式后所有内置振荡器关闭时，用于实现超时功能或脉冲计数器功能。

对 CKSEL 和 COUNTMODE 位进行编程，可控制 LPTIM 使用外部时钟源还是内部时钟源。

当使用外部时钟源时，可使用 CKPOL 位选择外部时钟信号的有效边沿。如果上升沿和下降沿均为有效边沿，则还应提供内部时钟信号（第一种配置）。在这种情况下，内部时钟信号频率应至少为外部时钟信号频率的五倍。

43.4.5 干扰滤波器

外部（映射到微控制器 GPIO）或内部（映射到芯片级或其它嵌入式外设，例如嵌入式比较器）LPTIM 输入由数字滤波器保护，避免任何毛刺和噪声干扰在 LPTIM 内部传播，从而防止产生意外计数或触发。

在激活数字滤波器之前，首先应向 LPTIM 提供内部时钟源，这是保证滤波器正常工作的必要条件。

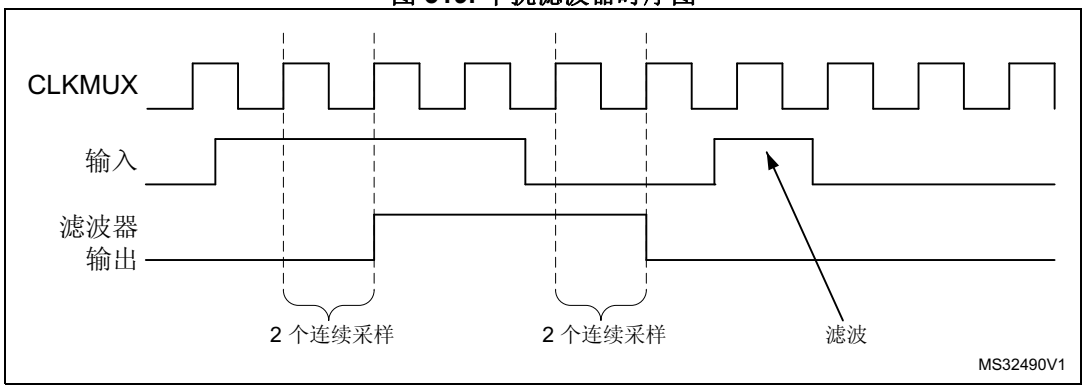
数字滤波器分为两组：

- 第一组数字滤波器保护 LPTIM 外部输入。数字滤波器的敏感性由 CKFLT 位控制。
- 第二组数字滤波器保护 LPTIM 内部触发输入。数字滤波器的敏感性由 TRGFLT 位控制。

注： *数字滤波器的敏感性以组为单位进行控制。无法单独配置同一组内各个数字滤波器的敏感性。*

滤波器的敏感性会影响相同的连续采样的数量，在其中一个 LPTIM 输入上检测到此类连续采样时，才能将某信号电平变化视为有效跳变。*图 519* 给出了编程 2 个连续采样时，干扰滤波器行为的示例。

图 519. 干扰滤波器时序图



注： *不提供内部时钟信号时，必须通过将 CKFLT 和 TRGFLT 位设为 0 来停用数字滤波器。在这种情况下，可使用外部模拟滤波器来防止 LPTIM 外部输入产生干扰。*

43.4.6 预分频器

LPTIM 16 位计数器前面要有一个可配置的 2 次幂预分频器。预分频器的分频比由 PRESC[2:0] 3 位域进行控制。下表列出了所有可能的分频比：

表 344. 预分频器的分频比

编程	分频系数
000	/1
001	/2
010	/4
011	/8
100	/16
101	/32
110	/64
111	/128

43.4.7 触发多路复用器

LPTIM 计数器可通过软件启动，也可以在 8 个触发输入之一上检测到有效边沿后启动。

TRIGEN[1:0] 用于确定 LPTIM 触发源：

- TRIGEN[1:0] 等于“00”时，LPTIM 计数器会在通过软件将 CNTSTRT 位或 SNGSTRT 位其中之一置 1 后立即启动。
- TRIGEN[1:0] 的其余三个可能的值用于配置触发输入使用的有效边沿。LPTIM 计数器会在检测到有效边沿后立即启动。

TRIGEN[1:0] 不等于“00”时，TRIGSEL[2:0] 用于选择使用 8 个触发输入中的哪一个来启动计数器。

外部触发信号视为 LPTIM 的异步信号。因此，检测到触发信号后，由于同步问题，需要延迟两个计数器时钟周期，定时器才能开始运行。

如果在定时器已启动时发生新的触发事件，则此事件将被忽略（除非已使能超时功能）。

注： *必须使能定时器，才能将 SNGSTRT/CNTSTRT 位置 1。当定时器禁止时，对这些位执行的任何写操作都将被硬件丢弃。*

43.4.8 工作模式

LPTIM 支持以下两种工作模式：

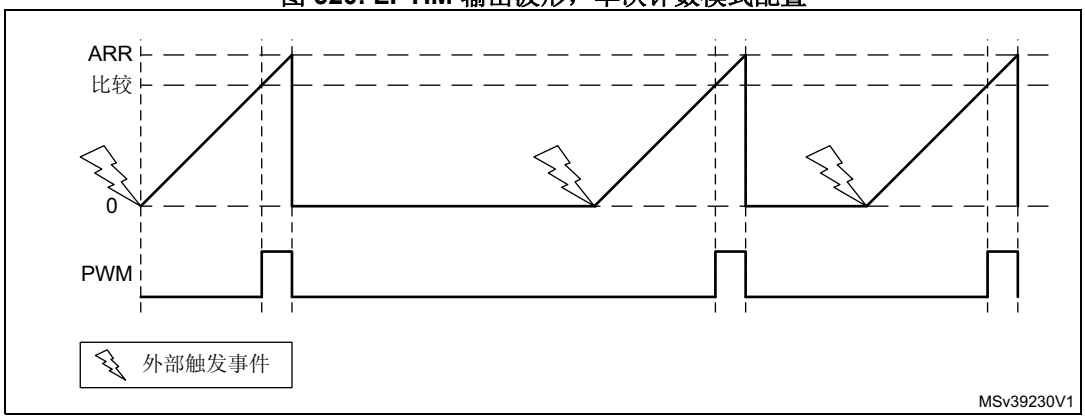
- 连续模式：定时器自由运行，由触发事件启动并且直到被禁止才会停止。
- 单触发模式：定时器由触发事件启动，当达到 ARR 值时停止。

新的触发事件将重新启动定时器。从计数器启动到计数器达到 ARR，这段时间内发生的任何触发事件均将被丢弃。

要使能单触发计数，必须将 SNGSTRT 位置 1。

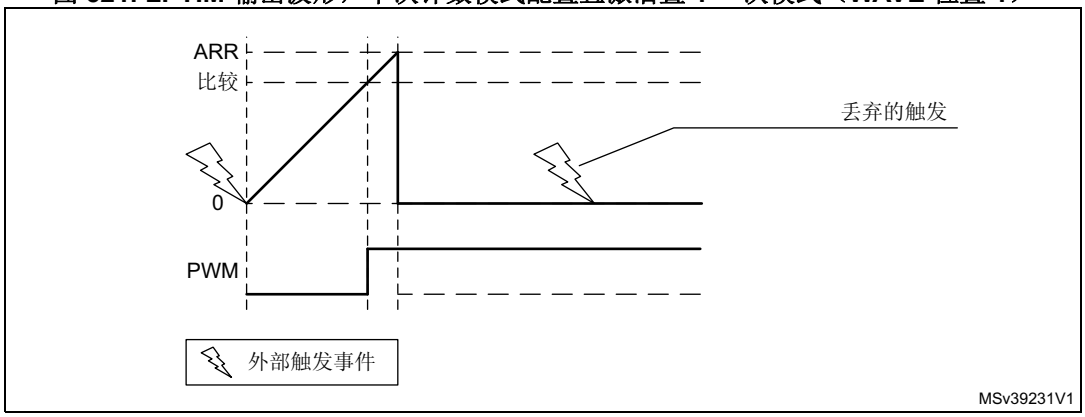
选择外部触发时，在 SNGSTRT 位置 1 后以及计数器寄存器停止后（包含零值）到达的每个外部触发事件都将为计数器启动新的单触发计数周期，如*图 520* 所示。

图 520. LPTIM 输出波形，单次计数模式配置



应注意，当 LPTIM_CFGR 寄存器中的 WAVE 位域置 1 时，将激活置 1 一次模式。在这种情况下，计数器仅会在第一个触发事件后启动一次，任何后续触发事件都将被丢弃，如*图 520* 所示。

图 521. LPTIM 输出波形，单次计数模式配置且激活置 1 一次模式（WAVE 位置 1）



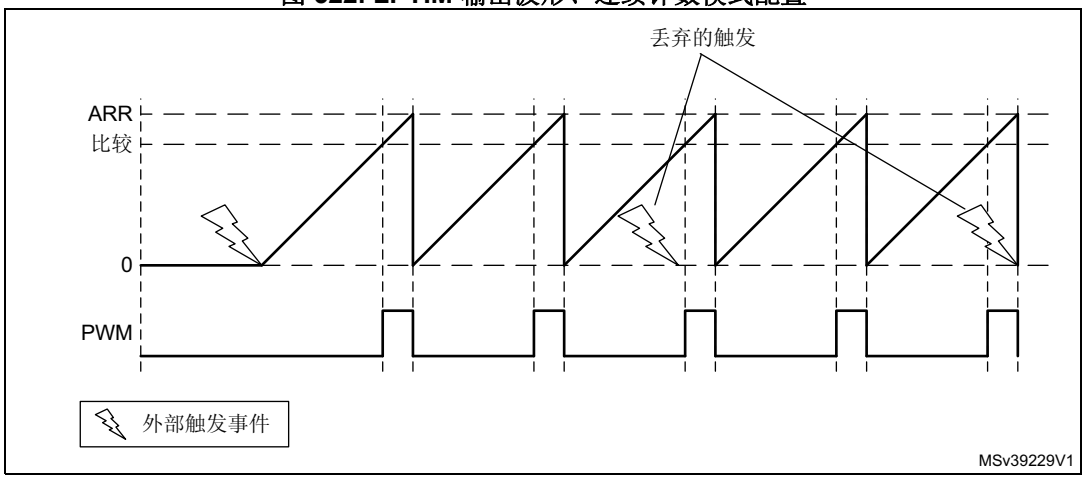
若通过软件启动（TRIGEN[1:0] = “00”），将 SNGSTRT 置 1 会使计数器进行单触发计数。

要使能连续计数，必须将 CNTSTRT 位置 1。

若选择外部触发，则在 CNTSTRT 置 1 后到达的外部触发事件将启动计数器进行连续计数。任何后续的外部触发事件都将被丢弃，如*图 522* 所示。

若通过软件启动（TRIGEN[1:0] = “00”），将 CNTSTRT 置 1 会使计数器开始连续计数。

图 522. LPTIM 输出波形、连续计数模式配置



SNGSTRT 和 CNTSTRT 位只能在定时器使能时（ENABLE 位置 1）置 1。可以将“实时”从单触发模式切换为连续模式。

如果之前选择的是连续模式，则将 SNGSTRT 置 1 会使 LPTIM 切换为单触发模式。计数器（激活时）将在达到 ARR 后立即停止。

如果之前选择的是单触发模式，则将 CNTSTRT 置 1 会使 LPTIM 切换为连续模式。计数器（激活时）将在达到 ARR 后立即重新启动。

43.4.9 超时功能

若在一个选定的触发输入上检测到有效边沿，则可用于复位 LPTIM 计数器。该功能通过 TIMOUT 位进行控制。

第一个触发事件将启动定时器，任何后续的触发事件将复位计数器，且定时器将重新启动。

可实现低功耗超时功能。超时值对应于比较值；如果在预期的时间帧内未发生触发事件，MCU 将由比较匹配事件唤醒。

43.4.10 生成波形

两个 16 位寄存器，LPTIM_ARR（自动重载寄存器）和 LPTIM_CMP（比较寄存器）用于在 LPTIM 输出上生成多个不同的波形。

定时器可生成以下波形：

- PWM 模式：若 LPTIM_CMP 寄存器与 LPTIM_CNT 寄存器匹配，则会立即将 LPTIM 输出置 1。若 LPTIM_ARR 寄存器与 LPTIM_CNT 寄存器匹配，则会立即将 LPTIM 输出复位。
- 单脉冲模式：对于第一个脉冲，输出波形与 PWM 模式输出波形类似，随后输出将永久复位。
- 置 1 一次模式：除输出保持最后一个信号电平外（取决于配置的输出极性），输出波形与单脉冲模式输出波形类似。

上述模式要求 LPTIM_ARR 寄存器的值严格大于 LPTIM_CMP 寄存器的值。

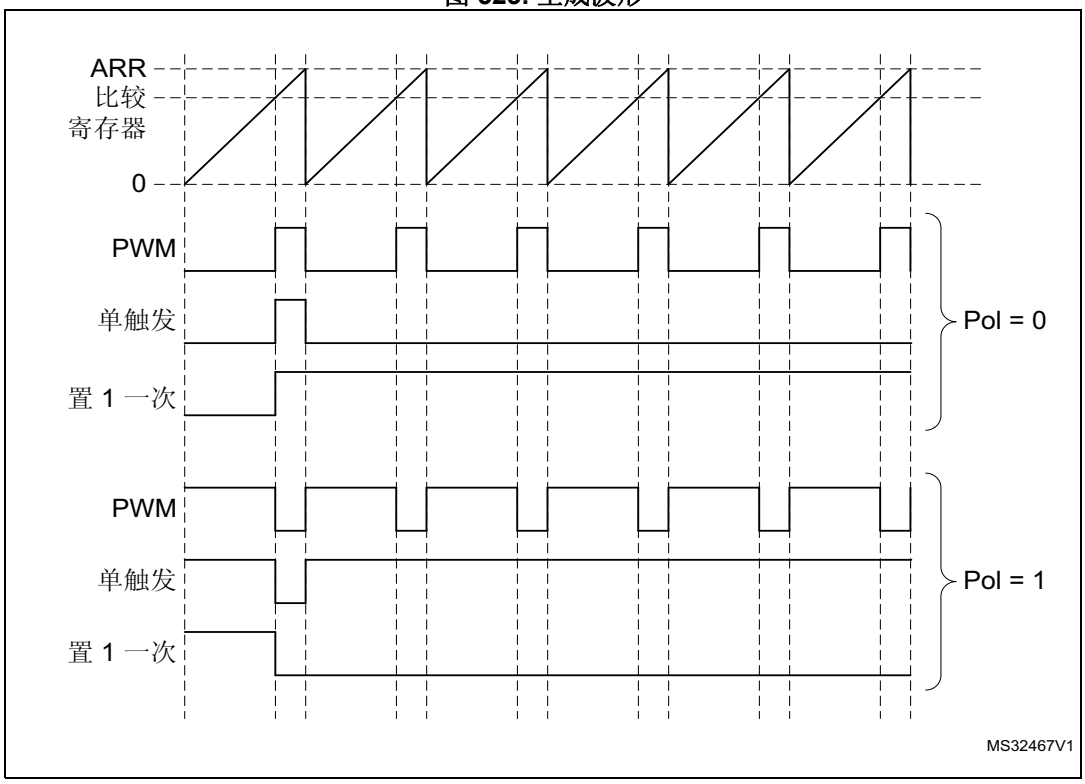
LPTIM 输出波形可通过 WAVE 位配置，具体如下：

- 若将 WAVE 位复位为 0，则会强制 LPTIM 生成 PWM 波形或单脉冲波形，具体取决于将哪个位（CNTSTRT 或 SNGSTRT）置 1。
- 若将 WAVE 位置 1，则会强制 LPTIM 生成置 1 一次模式波形。

WAVPOL 位控制 LPTIM 输出极性。更改立即生效，因此输出默认值将在极性重新配置后立即更改，甚至会在定时器使能前进行更改。

生成的信号的频率高达 LPTIM 时钟频率 2 分频。*图 523* 给出了可能在 LPTIM 输出上生成的三种波形。此外，此图还显示了通过 WAVPOL 位更改极性所产生的效果。

图 523. 生成波形



43.4.11 寄存器更新

LPTIM_ARR 寄存器和 LPTIM_CMP 寄存器在 APB 总线写操作后会立即更新，若定时器已启动，也可在当前周期结束时更新。

PRELOAD 位控制 LPTIM_ARR 寄存器和 LPTIM_CMP 寄存器的更新方式：

- 当 PRELOAD 位复位为 0 时，LPTIM_ARR 寄存器和 LPTIM_CMP 寄存器会在写访问后立即更新。
- 当 PRELOAD 位置 1 时，若定时器已启动，LPTIM_ARR 寄存器和 LPTIM_CMP 寄存器会在当前周期结束时更新。

LPTIM APB 接口和 LPTIM 内核逻辑使用的时钟不同，因此在 APB 写操作后，需要经过一定的延迟，写入值才能用于计数器比较器。在此延迟期间，必须避免向这些寄存器执行其他写操作。

LPTIM_ISR 寄存器中的 ARROK 标志和 CMPOK 标志分别指示 LPTIM_ARR 寄存器和 LPTIM_CMP 寄存器的写操作已完成。

向 LPTIM_ARR 寄存器和 LPTIM_CMP 寄存器执行写操作后，只有在前一次写操作完成后，才能对同一寄存器执行新的写操作。在 ARROK 标志或 CMPOK 标志置 1 前执行连续的写操作将造成无法预知的结果。

43.4.12 计数器模式

LPTIM 计数器可用于对 LPTIM Input1 上的外部事件进行计数，也可用于对内部时钟周期进行计数。CKSEL 位和 COUNTMODE 位用于控制将使用哪些源更新计数器。

若使用 LPTIM 对 Input1 上的外部事件进行计数，计数器可在上升沿、下降沿或两种边沿进行更新，具体取决于写入 CKPOL[1:0] 位的值。

根据 CKSEL 和 COUNTMODE 值，可选择以下计数模式：

- CKSEL = 0：LPTIM 由内部时钟源提供时钟
  - COUNTMODE = 0

    当 LPTIM 由内部时钟源提供时钟，且 LPTIM 计数器根据在 LPTIM 外部 Input1 上检测到的有效边沿进行更新时，不得对提供给 LPTIM 的内部时钟进行预分频（PRESC[2:0] = “000”）。
  - COUNTMODE = 1

    LPTIM 外部 Input1 通过提供给 LPTIM 的内部时钟采样。因此，为了不丢失任何事件，外部 Input1 信号变化的频率决不应超过提供给 LPTIM 的内部时钟的频率。
- CKSEL = 1：LPTIM 由外部时钟源提供时钟

  COUNTMODE 值不相关。

  在这种配置下，LPTIM 无需内部时钟源（已使能干扰滤波器时除外）。注入到 LPTIM 外部 Input1 的信号用作 LPTIM 的系统时钟。此配置适合未使能任何内置振荡器的工作模式。

  对于这种配置，LPTIM 计数器可以在 input1 时钟信号的上升沿或下降沿进行更新，但不可在上升沿和下降沿均更新。

  由于注入到 LPTIM 外部 Input1 的信号也可用于为 LPTIM 内核逻辑提供时钟，计数器递增计数前存在一些初始延时（使能 PTIM 后）。更确切地说，LPTIM 外部 Input1 的前五个有效边沿将丢失（使能 PTIM 后）。

43.4.13 定时器使能

LPTIM_CR 寄存器的 ENABLE 位用于使能/禁止 LPTIM 内核逻辑。将 ENABLE 位置 1 后，需要延迟两个计数器时钟周期，才能真正使能 LPTIM。

LPTIM_CFGR 和 LPTIM_IER 寄存器必须在禁止 LPTIM 后才能修改。

43.4.14 定时器计数器复位；

为了将 LPTIM_CNT 寄存器的内容复位为零，实现了两种复位机制：

- 同步复位机制：同步复位由 LPTIM_CR 寄存器中的 COUNTRST 位控制。在将 COUNTRST 位域置 1 后，复位信号在 LPTIM 内核时钟域中传播。因此，重要的是要注意，要在经历几个 LPTIM 内核逻辑时钟脉冲之后再考虑复位。这将使 LPTIM 计数器在复位触发和生效之间额外计数几个脉冲。由于 COUNTRST 位位于 APB 时钟域中，并且 LPTIM 计数器位于 LPTIM 内核时钟域中，因此当将 1 写入到 COUNTRST 位时，内核时钟需要 3 个时钟周期的延迟用以同步由 APB 时钟域发出的复位信号。
- 异步复位机制：异步复位由位于 LPTIM_CR 寄存器中的 RSTARE 位控制。当该位置 1 时，对 LPTIM_CNT 寄存器的任何读访问都会将其内容复位为零。应在不提供 LPTIM 内核时钟的时间范围内触发异步复位。例如，当 LPTIM Input1 用作外部时钟源时，只有当足够保证 LPIDM Input1 不会发生反转时，才应用异步复位。
  应注意的是，为了实现可靠的 LPTIM_CNT 寄存器内容读取，必须执行两次连续的读访问并进行比较。当两次读访问的值相等时，可认为读访问可靠。然而，当使能了异步复位时，不可能两次读取 LPTIM_CNT 寄存器。

警告： LPTIM 内没有防止两个复位机制同时使用的机制。所以开发人员应该确保这两个机制是排斥地使用。

43.4.15 编码器模式

此模式用于处理来自正交编码器的信号，此正交编码器用于检测旋转元件的角度位置。编码器接口模式就相当于带有方向选择的外部时钟。这意味着，计数器仅在 0 到 LPTIM_ARR 寄存器中编程的自动重载值之间进行连续计数（根据具体方向，从 0 递增计数到 ARR，或从 ARR 递减计数到 0）。因此，在启动前必须先配置 LPTIM_ARR。通过两个外部输入信号 Input1 和 Input2 生成时钟信号作为 LPTIM 计数器时钟。这两个信号间的相位确定计数方向。

仅当 LPTIM 由内部时钟源提供时钟时才可使用编码器模式。Input1 和 Input2 输入上的信号频率不得超过 LPTIM 内部时钟频率 4 分频。必须满足此条件才能确保 LPTIM 正常工作。

方向变化由 LPTIM_ISR 寄存器中的两个递减和递增标志指示。此外，如果通过 LPTIM_IER 寄存器使能，还可为两种方向变化事件产生中断。

要激活编码器模式，必须将 ENC 位置 1。LPTIM 必须首先配置为连续模式。

当编码器模式激活时，LPTIM 计数器按照增量编码器的速度和方向自动修改。因此，其内容始终代表编码器的位置。计数方向由递增和递减标志指示，对应于编码器转子的旋转方向。

根据使用 CKPOL[1:0] 位配置的边沿敏感性，可得几种不同的计数方案。下表汇总了可能的组合（假设 Input1 和 Input2 不同时切换）。

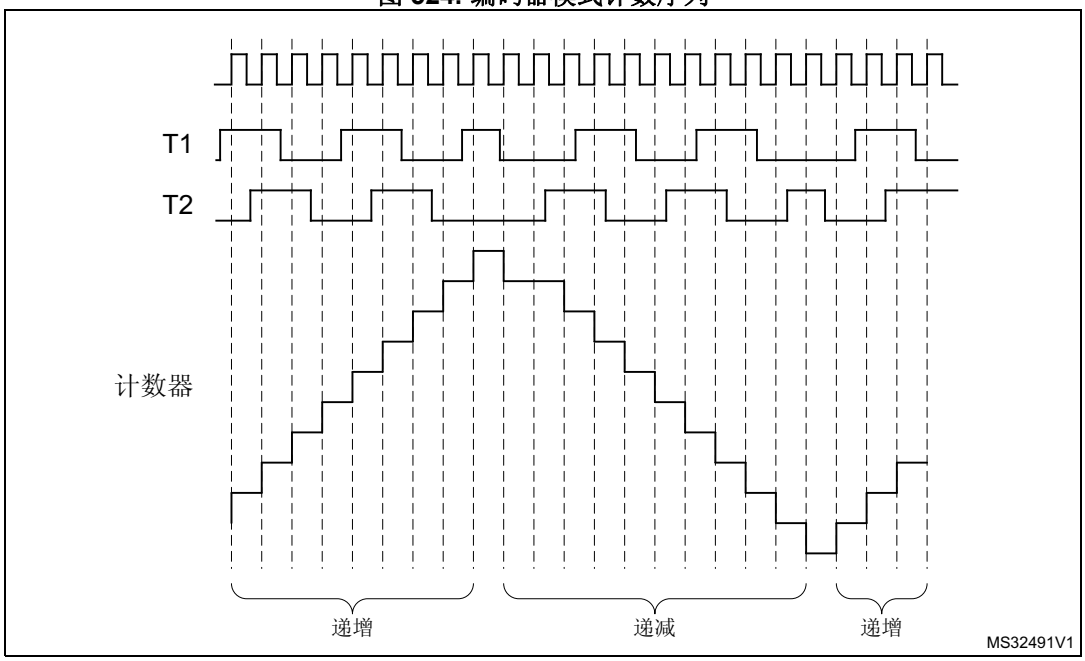
表 345. 编码器计数方案

有效边沿	相反信号的电平（Input1 对应 Input2，Input2 对应 Input1）	Input1 信号		Input2 信号	
		上升	下降	上升	下降
上升沿	高	递减	不计数	递增	不计数
	低	递增	不计数	递减	不计数
下降沿	高	不计数	递增	不计数	递减
	低	不计数	递减	不计数	递增
两种边沿	高	递减	递增	递增	递减
	低	递增	递减	递减	递增

下图所示为编码器模式下配置了两种边沿敏感性的计数序列。

注意： 在此模式下，LPTIM 必须由内部时钟源提供时钟，因此 CKSEL 位必须保持其复位值 0。另外，预分频器分频比必须等于其复位值 1（PRESC[2:0] 位必须为“000”）。

图 524. 编码器模式计数序列



43.5 LPTIM 中断

若以下事件通过 LPTIM_IER 寄存器使能，则这些事件会生成中断/唤醒事件：

- 比较匹配
- 自动重载匹配（编码器模式下无论哪种方向）
- 外部触发事件
- 自动重载寄存器写操作完成
- 比较寄存器写操作完成
- 方向变化（编码器模式），可编程（递增/递减/同时递增和递减）

注： *只要 LPTIM_IER 寄存器（中断使能寄存器）中的位在 LPTIM_ISR 寄存器（状态寄存器）中相应标志置 1 后置 1，就不会触发中断*

表 346. 中断事件

中断事件	说明
比较匹配	当计数器寄存器 (LPTIM_CNT) 的内容与比较寄存器 (LPTIM_CMP) 的内容匹配时，生成中断标志。
自动重载匹配	当计数器寄存器 (LPTIM_CNT) 的内容与自动重新加载寄存器 (LPTIM_ARR) 的内容匹配时，生成中断标志。
外部触发事件	当检测到外部触发事件时，会生成中断标志。
自动重载寄存器写操作完成	当对 LPTIM_ARR 寄存器的写操作完成时，生成中断标志。
比较寄存器写操作完成	当对 LPTIM_CMP 寄存器的写操作完成时，生成中断标志。
方向更改	用于编码器模式。嵌入两个中断标志以发出方向更改的信号： – UP 标志发出递增计数方向更改的信号 – DOWN 标志发出递减计数方向更改的信号

43.6 LPTIM 寄存器

43.6.1 LPTIM 中断和状态寄存器 (LPTIM_ISR)

LPTIM interrupt and status register

偏移地址：0x00

复位值：0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DOWN	UP	ARROK	CMPOK	EXTTRIG	ARRM	CMPM
									r	r	r	r	r	r	r

位 31:7 保留，必须保持复位值。

位 6 **DOWN**：计数方向从递增变为递减 (Counter direction change up to down)
在编码器模式下，由硬件将 DOWN 位置 1 时，会通知应用计数方向由递增变为递减。

位 5 **UP**：计数方向从递减变为递增 (Counter direction change down to up)
在编码器模式下，由硬件将 UP 位置 1 时，会通知应用计数方向由递减变为递增。

位 4 **ARROK**：自动重载寄存器更新成功 (Autoreload register update OK)
由硬件将 ARROK 置 1 时，会通知应用 LPTIM_ARR 寄存器的 APB 总线写操作已成功完成。如果这样，便可启动新的写操作。

位 3 **CMPOK**：比较寄存器更新成功 (Compare register update OK)
由硬件将 CMPOK 置 1 时，会通知应用 LPTIM_CMP 寄存器的 APB 总线写操作已成功完成。如果这样，便可启动新的写操作。

位 2 **EXTTRIG**：外部触发边沿事件 (External trigger edge event)
由硬件将 EXTTRIG 置 1 时，会通知应用所选的外部触发输入上产生有效边沿。如果由于定时器已启动而忽略触发事件，则不会将此标志置 1。

位 1 **ARRM**：自动重载匹配 (Autoreload match)
由硬件将 ARRM 置 1 时，会通知应用 LPTIM_CNT 寄存器的值已达到 LPTIM_ARR 寄存器的值。

位 0 **CMPM**：比较匹配 (Compare match)
由硬件将 CMPM 位置 1 时，会通知应用 LPTIM_CNT 寄存器的值已达到 LPTIM_CMP 寄存器的值。

43.6.2 LPTIM 中断清零寄存器 (LPTIM_ICR)

LPTIM interrupt clear register

偏移地址：0x04

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DOWNCF	UPCF	ARROKCF	CMPOKCF	EXTTRIGCF	ARRMCF	CMPMCF
									w	w	w	w	w	w	w

位 31:7 保留，必须保持复位值。

位 6 **DOWNCF**：方向变为递减清零标志 (Direction change to down Clear Flag)
将 1 写入此位时，LPT_ISR 寄存器中的 DOWN 标志将清零

位 5 **UPCF**：方向变为递增清零标志 (Direction change to UP Clear Flag)
将 1 写入此位时，LPT_ISR 寄存器中的 UP 标志将清零

位 4 **ARROKCF**：自动重载寄存器更新成功清零标志 (Autoreload register update OK Clear Flag)
将 1 写入此位时，LPT_ISR 寄存器中的 ARROK 标志将清零

位 3 **CMPOKCF**：比较寄存器更新成功清零标志 (Compare register update OK Clear Flag)
将 1 写入此位时，LPT_ISR 寄存器中的 CMPOK 标志将清零

位 2 **EXTTRIGCF**：外部触发有效边沿清零标志 (External trigger valid edge Clear Flag)
将 1 写入此位时，LPT_ISR 寄存器中的 EXTTRIG 标志将清零

位 1 **ARRMCF**：自动重载匹配清零标志 (Autoreload match Clear Flag)
将 1 写入此位时，LPT_ISR 寄存器中的 ARRM 标志将清零

位 0 **CMPMCF**：比较匹配清零标志 (compare match Clear Flag)
将 1 写入此位时，LPT_ISR 寄存器中的 CMP 标志将清零

43.6.3 LPTIM 中断使能寄存器 (LPTIM_IER)

LPTIM interrupt enable register

偏移地址：0x08

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DOWNIE	UPIE	ARROKIE	CMPOKIE	EXTTRIGIE	ARRMIE	CMPMIE
									rw	rw	rw	rw	rw	rw	rw

位 31:7 保留，必须保持复位值。

位 6 **DOWNIE**：方向变为递减中断使能 (Direction change to down Interrupt Enable)
0：禁止 DOWN 中断
1：使能 DOWN 中断

位 5 **UPIE**：方向变为递增中断使能 (Direction change to UP Interrupt Enable)
0：禁止 UP 中断
1：使能 UP 中断

位 4 **ARROKIE**：自动重载寄存器更新成功中断使能 (Autoreload register update OK Interrupt Enable)
0：禁止 ARROK 中断
1：使能 ARROK 中断

位 3 **CMPOKIE**：比较寄存器更新成功中断使能 (Compare register update OK Interrupt Enable)
0：禁止 CMPOK 中断
1：使能 CMPOK 中断

位 2 **EXTTRIGIE**：外部触发有效边沿中断使能 (External trigger valid edge Interrupt Enable)
0：禁止 EXTTRIG 中断
1：使能 EXTTRIG 中断

位 1 **ARRMIE**：自动重载匹配中断使能 (Autoreload match Interrupt Enable)
0：禁止 ARRM 中断
1：使能 ARRM 中断

位 0 **CMPMIE**：比较匹配中断使能 (Compare match Interrupt Enable)
0：禁止 CMPM 中断
1：使能 CMPM 中断

注意： 必须在 LPTIM 禁止已时（ENABLE 位复位为 0）才能修改 LPTIM_IER 寄存器

43.6.4 LPTIM 配置寄存器 (LPTIM_CFGR)

LPTIM configuration register

偏移地址：0x0C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	ENC	COUNT MODE	PRELOAD	WAVPOL	WAVE	TIMOUT	TRIGEN		Res.
							rw	rw	rw	rw	rw	rw	rw	rw	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TRIGSEL			Res.	PRESC			Res.	TRGFLT		Res.	CKFLT		CKPOL		CKSEL
rw	rw	rw		rw	rw	rw		rw	rw		rw	rw	rw	rw	rw

位 31:25 保留，必须保持复位值。

位 24 **ENC**：编码器模式使能 (Encoder mode enable)
ENC 位控制编码器模式
0：禁止编码器模式
1：使能编码器模式

位 23 **COUNTMODE**：计数器模式使能 (counter mode enabled)
COUNTMODE 位用于选择 LPTIM 使用哪个时钟源来为计数器提供时钟：
0：计数器在每个内部时钟脉冲后递增
1：计数器在 LPTIM 外部 Input1 上的每个有效时钟脉冲后递增

位 22 **PRELOAD**：寄存器更新模式 (Registers update mode)
PRELOAD 位控制 LPTIM_ARR 寄存器和 LPTIM_CMP 寄存器的更新方式
0：寄存器在每次 APB 总线写访问后更新
1：寄存器在当前 LPTIM 周期结束时更新

位 21 **WAVPOL**：波形极性 (Waveform shape polarity)
WAVEPOL 位控制输出极性
0：LPTIM 输出反映 LPTIM_ARR 寄存器和 LPTIM_CMP 寄存器之间的比较结果
1：LPTIM 输出反映与 LPTIM_ARR 寄存器和 LPTIM_CMP 寄存器之间的比较结果相反的值

位 20 **WAVE**：波形 (Waveform shape)
WAVE 位控制输出波形
0： 停用置 1 一次模式和 PWM/单脉冲波形（具体取决于 OPMODE 位）
1： 激活置 1 一次模式

位 19 **TIMOUT**：超时使能 (Timeout enable)
TIMOUT 位控制超时功能
0：定时器已启动时到达的触发事件将被忽略
1：定时器已启动时到达的触发事件将复位并重新启动计数器

位 18:17 **TRIGEN[1:0]**：触发使能和极性 (Trigger enable and polarity)

TRIGEN 位控制 LPTIM 计数器是否由外部触发信号启动。如果已选择由外部触发信号启动，触发有效边沿的配置有以下三种：

00：软件触发（由软件启动计数）
01：上升沿为有效边沿
10：下降沿为有效边沿
11：上升沿和下降沿均为有效边沿

位 16 保留，必须保持复位值。

位 15:13 **TRIGSEL[2:0]**：触发源选择器 (Trigger selector)

TRIGSEL 位用于选择作为 LPTIM 触发事件的触发源，可用触发源包括以下 8 种：

000：lptim_ext_trig0
001：lptim_ext_trig1
010：lptim_ext_trig2
011：lptim_ext_trig3
100：lptim_ext_trig4
101：lptim_ext_trig5
110：lptim_ext_trig6
111：lptim_ext_trig7

位 12 保留，必须保持复位值。

位 11:9 **PRESC[2:0]**：时钟预分频器 (Clock prescaler)

PRESC 位配置预分频器的分频系数。分频系数可从以下分频系数中选择：

000： /1
001： /2
010: /4
011： /8
100： /16
101： /32
110： /64
111： /128

位 8 保留，必须保持复位值。

位 7:6 **TRGFLT[1:0]**：触发信号的可配置数字滤波器 (Configurable digital filter for trigger)

TRGFLT 值用于设置连续相同采样的数量，若在内部触发信号电平发生变化时检测到此类连续采样，才会将此电平变化视为有效电平切换。必须存在内部时钟源才能使用此功能

00：任何触发信号有效电平变化均视为有效触发。
01：触发信号有效电平变化必须至少稳定 2 个时钟周期，才能将其视为有效触发。
10：触发信号有效电平变化必须至少稳定 4 个时钟周期，才能将其视为有效触发。
11：触发信号有效电平变化必须至少稳定 8 个时钟周期，才能将其视为有效触发。

位 5 保留，必须保持复位值。

位 4:3 **CKFLT[1:0]**：外部时钟的可配置数字滤波器 (Configurable digital filter for external clock)

CKFLT 值用于设置连续相同采样的数量，若在外部时钟信号电平发生变化时检测到此类连续采样，才会将此电平变化视为有效电平切换。必须存在内部时钟源才能使用此功能

00：任何外部时钟信号电平变化均视为有效切换。

01：外部时钟信号电平变化必须至少稳定 2 个时钟周期，才能将其视为有效切换。

10：外部时钟信号电平变化必须至少稳定 4 个时钟周期，才能将其视为有效切换。

11：外部时钟信号电平变化必须至少稳定 8 个时钟周期，才能将其视为有效切换。

位 2:1 **CKPOL[1:0]**：时钟极性 (Clock Polarity)

如果 LPTIM 由外部时钟源提供时钟：

当 LPTIM 由外部时钟源提供时钟时，CKPOL 位用于配置计数所使用的有效边沿：

00：上升沿为用于计数的有效边沿

01：下降沿为用于计数的有效边沿

10：上升沿和下降沿均为有效边沿 当外部时钟信号的上升沿和下降沿均视为有效边沿时，LPTIM 还必须由内部时钟源提供时钟，且内部时钟源频率至少等于外部时钟频率的四倍。

11：不允许

如果将 LPTIM 配置为编码器模式（ENC 位置 1）：

00：编码器子模式 1 激活

01：编码器子模式 2 激活

10：编码器子模式 3 激活

有关编码器模式子模式的更多详细信息，请参见*第 43.4.15 节：编码器模式*。

位 0 **CKSEL**：时钟选择器 (Clock selector)

CKSEL 位选择 LPTIM 将使用的时钟源：

0：LPTIM 由内部时钟源（APB 时钟或任意内置振荡器）提供时钟

1：LPTIM 由外部时钟源通过 LPTIM 外部 Input1 提供时钟

注意： 必须在 LPTIM 已禁止时（ENABLE 位复位为 0）才能修改 LPTIM_CFGR 寄存器。

43.6.5 LPTIM 控制寄存器 (LPTIM_CR)

LPTIM control register

偏移地址：0x10

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RSTARE	COUNTRST	CNTSTRT	SNGSTRT	ENABLE
											w	rs	rw	rw	rw

位 31:5 保留，必须保持复位值。

位 4 **RSTARE**：读操作后复位使能 (Reset after read enable)

此位由软件置 1 和清零。当 RSTARE 置 1 时，对 LPTIM_CNT 寄存器的任何读取访问都将异步复位 LPTIM_CNT 寄存器的内容。

注意： 只能对该位域执行写操作。这意味着该位不能被读回以验证已写入的值。例如，如果该位设置为 1，即使使能了“读操作后复位”功能（由于该位域以前已写入 1），尝试读取它也将返回 0。要关闭“读操作后复位”或确保已经关闭，应（通过将其编程为 0）将该位复位，即使其已经为 0。

位 3 **COUNTRST**：计数器复位 (Counter reset)

此位通过软件置 1，通过硬件清零。当设置为 1 时，该位将触发 LPTIM_CNT 计数器寄存器的同步复位。由于该复位的同步性质，它仅在 3 个 LPTimer 内核时钟周期（LPTimer 内核时钟可能不同于 APB 时钟）的同步延迟之后发生。

注意： COUNTRST 在已由硬件清零之前，不得由软件置“1”。因此，软件应在尝试将 COUNTRST 位置“1”之前检查其是否已清零。

位 2 **CNTSTRT**：定时器以连续模式启动 (Timer start in continuous mode)

此位通过软件置 1，通过硬件清零。

若通过软件启动（TRIGEN[1:0] =“00”），将此位置 1 会使 LPTIM 以连续模式启动。

如果禁止软件启动（TRIGEN[1:0] 不等于“00”），将此位置 1 会使定时器在检测到外部触发信号后立即以连续模式启动。

如果在进行单脉冲模式计数时将此位置 1，则在 LPTIM_ARR 寄存器和 LPTIM_CNT 寄存器下一次匹配时定时器不会停止，LPTIM 计数器将继续以连续模式计数。

只有在使能 LPTIM 时才能将此位置 1。此位由硬件自动复位。

位 1 **SNGSTRT**：LPTIM 以单脉冲模式启动 (LPTIM start in single mode)

此位通过软件置 1，通过硬件清零。

若通过软件启动（TRIGEN[1:0] =“00”），将此位置 1 会使 LPTIM 以单脉冲模式启动。

如果禁止软件启动（TRIGEN[1:0] 不等于“00”），将此位置 1 会使 LPTIM 在检测到外部触发信号后立即以单脉冲模式启动。

如果在 LPTIM 处于连续计数模式时将此位置 1，LPTIM 将在 LPTIM_ARR 寄存器和 LPTIM_CNT 寄存器下一次匹配时停止。

只有在使能 LPTIM 时才能将此位置 1。此位由硬件自动复位。

位 0 **ENABLE**：LPTIM 使能 (LPTIM Enable)

ENABLE 位由软件置 1 和清零。

0：禁止 LPTIM

1：使能 LPTIM

43.6.6 LPTIM 比较寄存器 (LPTIM_CMP)

LPTIM compare register

偏移地址：0x14

复位值：0x0000 0000

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

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

位 31:16 保留，必须保持复位值。

位 15:0 **CMP[15:0]**：比较值 (Compare value)。
CMP 为 LPTIM 所使用的比较值。
必须在 LPTIM 已使能时（ENABLE 位置 1）才能修改 LPTIM_CMP 寄存器的内容。

43.6.7 LPTIM 自动重载寄存器 (LPTIM_ARR)

LPTIM autoreload register

偏移地址：0x18

复位值：0x0000 0001

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

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

位 31:16 保留，必须保持复位值。

位 15:0 **ARR[15:0]**：自动重载值 (Auto reload value)。
ARR 为 LPTIM 的自动重载值。
此值必须严格大于 CMP[15:0] 的值。
必须在 LPTIM 已使能时（ENABLE 位置 1）才能修改 LPTIM_ARR 寄存器的内容。

43.6.8 LPTIM 计数器寄存器 (LPTIM_CNT)

LPTIM counter register

偏移地址：0x1C

复位值：0x0000 0000

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

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

位 31:16 保留，必须保持复位值。

位 15:0 **CNT[15:0]**：计数器值 (Counter value)。

当 LPTIM 通过异步时钟运行时，读取 LPTIM_CNT 寄存器会返回不可靠的值。因此在这种情况下，必须连续执行读访问两次，并验证两次返回的值是否相同。

应注意的是，为了实现可靠的 LPTIM_CNT 寄存器读访问，必须执行两次连续的读访问并进行比较。当两次连续读访问的值相等时，可认为读访问可靠。

43.6.9 LPTIM 配置寄存器 2 (LPTIM_CFGR2)

LPTIM configuration register 2

偏移地址：0x24

复位值：0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	IN2SEL		Res.	Res.	IN1SEL	
										rw	rw			rw	rw

位 31:6 保留，必须保持复位值。

位 5:4 **IN2SEL[1:0]**：LPTIMx 输入 2 选择 (LPTIMx Input 2 selection)

IN2SEL 位控制 LPTIMx 输入 2 多路复用器，它将 LPTIMx 输入 2 连接至其中一个可用输入。

00：lptim_in2_mux0

01：lptim_in2_mux1

10：lptim_in2_mux2

11：lptim_in2_mux3

有关连接详细信息，请参见*表 340：LPTIM1 输入 2 连接*了解 LPTIM1 输入 2 连接，以及参见*表 342：LPTIM2 输入 2 连接*了解 LPTIM2 输入 2 连接。

位 3:2 保留，必须保持复位值。

位 1:0 **IN1SEL[1:0]**：LPTIMx 输入 1 选择 (LPTIMx Input 1 selection)
IN1SEL 位控制 LPTIMx 输入 1 多路复用器，它将 LPTIMx 输入 1 连接至其中一个可用输入。
00：lptim_in1_mux0
01：lptim_in1_mux1
10：lptim_in1_mux2
11：lptim_in1_mux3
有关连接详细信息，请参见*表 339：LPTIM1 输入 1 连接*了解 LPTIM1 输入 1 连接，以及参见*表 341：LPTIM2 输入 1 连接*了解 LPTIM2 输入 1 连接。

注意： 必须在 LPTIM 已禁止时（ENABLE 位清零）才能修改 LPTIM_CFGR2 寄存器。

43.6.10 LPTIM3 配置寄存器 2 (LPTIM3_CFGR2)

LPTIM3 configuration register 2

偏移地址：0x24

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	IN1SEL	
														rw	rw

位 31:2 保留，必须保持复位值。

位 1:0 **IN1SEL**：LPTIM3 输入 1 选择 (LPTIM3 Input1 selection)
IN1SEL 位控制 LPTIM3 输入 1 多路复用器，它将 LPTIM3 输入 1 连接至其中一个可用输入。
00：lptim_in1_mux0
01：lptim_in1_mux1
10：lptim_in1_mux2
11：lptim_in1_mux3
有关连接详细信息，请参见*表 343：LPTIM3 输入 1 连接*。

注意： 必须在 LPTIM 已禁止时（ENABLE 位复位为 0）才能修改 LPTIM3_CFGR2 寄存器。

43.6.11 LPTIM 寄存器映射

下表对 LPTIM 寄存器进行了汇总。

表 347. LPTIM 寄存器映射和复位值

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	LPTIM_ISR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DOWN	UP	ARROK	CMPOK	EXTTRIG	ARRM	CMPM
	Reset value																										0	0	0	0	0	0	0
0x04	LPTIM_ICR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DOWNCF	UPCF	ARROKCF	CMPOKCF	EXTTRIGCF	ARRMCF	CMPMCF
	Reset value																										0	0	0	0	0	0	0
0x08	LPTIM_IER	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DOWNIE	UPIE	ARROKIE	CMPOKIE	EXTTRIGIE	ARRMIE	CMPMIE
	Reset value																										0	0	0	0	0	0	0
0x0C	LPTIM_CFGR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ENC	COUNTMODE	PRELOAD	WAVPOL	WAVE	TIMOUT	TRIGEN[1:0]		Res.	TRIGSEL[2:0]			Res.	PRESC[2:0]			Res.	TRGFLT[1:0]		Res.	CKFLT[1:0]		CKPOL[1:0]		CKSEL
	Reset value								0	0	0	0	0	0	0	0		0	0	0		0	0	0		0	0	0	0	0	0	0	0
0x10	LPTIM_CR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RSTARE	COUNTRST	CNTSTRT	SNGSTRT	ENABLE
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x14	LPTIM_CMP	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CMP[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x18	LPTIM_ARR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ARR[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
0x1C	LPTIM_CNT	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CNT[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x24	LPTIM_CFGR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	IN2SEL[1:0]		Res.	Res.	IN1SEL[1:0]	
	Reset Value																											0	0			0	0

表 347. LPTIM 寄存器映射和复位值（续）

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x24	LPTIM3_ CFGR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	IN1SEL	
	Reset Value																															0	0

有关寄存器边界地址的信息，请参见*第 100 页的第 2.2.2 节*。

44 系统窗口看门狗 (WWDG)

44.1 前言

系统窗口看门狗 (WWDG) 通常被用来监测，由外部干扰或不可预见的逻辑条件造成的应用程序背离正常的运行序列而产生的软件故障。除非程序在 T6 位变成 0 前刷新递减计数器的值，否则看门狗电路在达到预置的时间周期时，会产生一个复位。如果在递减计数器达到窗口寄存器值之前刷新控制寄存器中的 7 位递减计数器值，也会产生复位。这意味着必须在限定的时间窗口内刷新计数器。

WWDG 时钟由 APB 时钟经预分频后提供，通过可配置的时间窗口来检测应用程序非正常的过迟或过早的操作。

WWDG 最适合那些要求看门狗在精确计时窗口起作用的应用程序。

44.2 WWDG 主要特性

- 可编程的自由运行递减计数器
- 复位条件
  - 当递减计数器值小于 0x40 时复位（如果看门狗已激活）
  - 在窗口之外重载递减计数器时复位（如果看门狗已激活）（请参见*图 526*）
- 提前唤醒中断 (EWI)：当递减计数器等于 0x40 时触发（如果已使能且看门狗已激活）

44.3 WWDG 功能说明

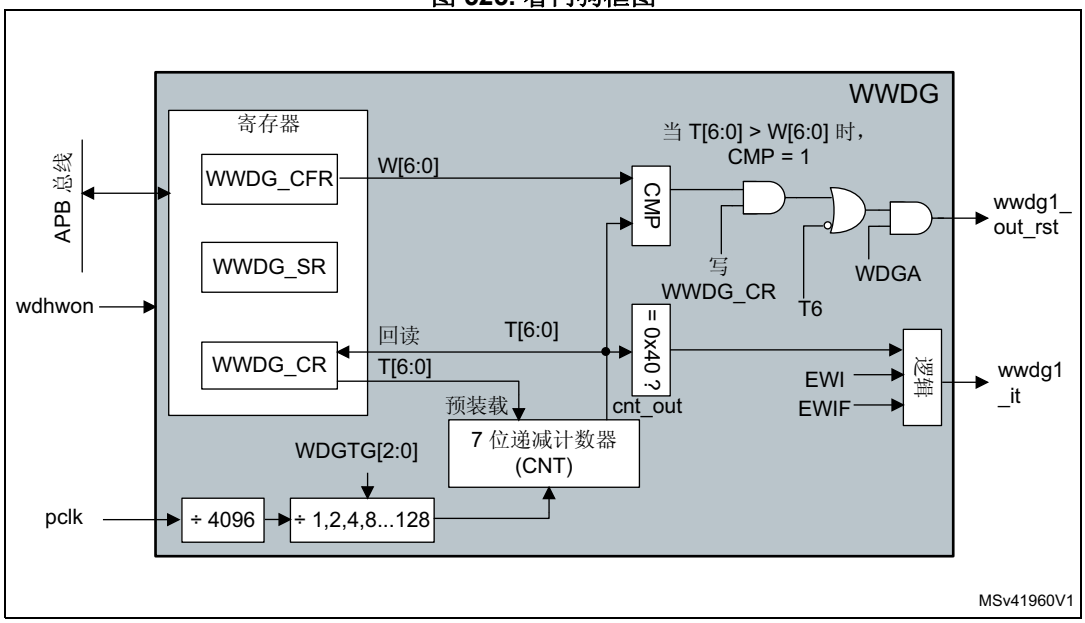
如果激活看门狗（WWDG_CR 寄存器中的 WDGA 位置 1），则当 7 位递减计数器（T[6:0] 位）从 0x40 递减到 0x3F（T6 已清零）时会引发复位。如果软件在重载计数器时，计数器的值大于窗口寄存器，则会产生复位。

应用程序在正常运行过程中必须定期写入 WWDG_CR 寄存器以防止复位。只有当计数器值低于窗口寄存器值且高于 0x3F 时，才能执行此操作。要存储到 WWDG_CR 寄存器中的值必须介于 0xFF 和 0xC0 之间。

请参见*图 525* 了解 WWDG 框图，以及参见*第 44.3.2 节：WWDG 内部信号*。

44.3.1 WWDG框图

图 525. 看门狗框图



44.3.2 WWDG 内部信号

表 348 给出了 WWDG 内部信号列表。

表 348. WWDG 内部输入/输出信号

信号名称	信号类型	说明
pclk	数字输入	APB 总线时钟
wwdg1_out_rst	数字输出	WWDG1 复位信号输出
wwdg1_it	数字输出	WWDG1 中断输出

44.3.3 使能看门狗

在系统复位后，看门狗总是处于关闭状态。可通过设置 WWDG_CR 寄存器中的 WDGA 位来使能看门狗，之后除非执行复位操作，否则不能再次关闭。

44.3.4 控制递减计数器

递减计数器处于自由运行状态，即使禁止看门狗，递减计数器仍继续递减计数。当使能看门狗时，必须将 T6 位置 1，以防止立即复位。

T[5:0] 位包含了看门狗产生复位之前的计时数目； 复位前的延时时间在一个最小值和一个最大值之间变化，这是因为写入 WWDG_CR 寄存器时，预分频器的状态是未知的（请参见 *图 526*）。配置寄存器 (WWDG_CFR) 包含窗口的上限：为防止发生复位，当递减计数器的值低于窗口寄存器值且大于 0x3F 时必须重载。*图 526* 介绍了窗口看门狗的工作过程。

注： *可使用 T6 位产生软件复位（将 WDGA 位置 1 并将 T6 位清零）。*

44.3.5 看门狗中断高级特性

如果在产生实际复位之前必须执行特定的安全操作或数据记录，则可使用提前唤醒中断 (EWI)。通过设置 WWDG_CFR 寄存器中的 EWI 位使能 EWI 中断。当递减计数器的值为 0x40 时，将生成 EWI 中断。在复位器件之前，可以使用相应的中断服务程序 (ISR) 来触发特定操作（例如通信或数据记录）。

在某些应用中，可以使用 EWI 中断来管理软件系统检查和/或系统恢复/功能退化，而不会生成 WWDG 复位。在这种情况下，相应的中断服务程序 (ISR) 可用来重载 WWDG 计数器以避免 WWDG 复位，然后再触发所需操作。

通过将 0 写入 WWDG_SR 寄存器中的 EWIF 位来清除 EWI 中断。

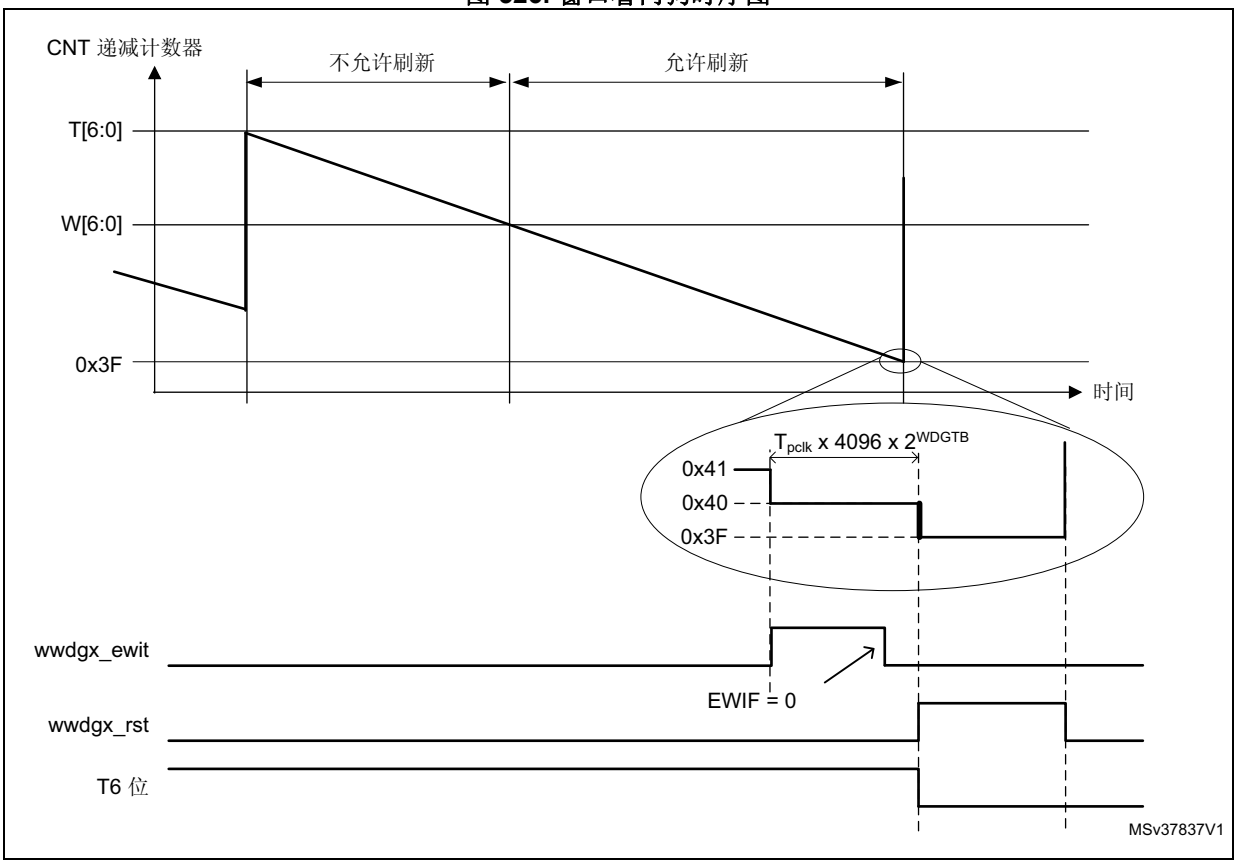
注： *当由于在更高优先级任务中有系统锁定而无法使用 EWI 中断时，最终会产生 WWDG 复位。*

44.3.6 如何设置看门狗超时

可以使用*图 526* 中的公式来计算 WWDG 超时。

警告： **写入 WWDG_CR 寄存器时，始终将 1 写入 T6 位，以避免生成立即复位。**

图 526. 窗口看门狗时序图



超时值的计算公式如下：

$$t_{WWDG} = t_{PCLK} \times 4096 \times 2^{WDGTB[2:0]} \times (T[5:0] + 1) \quad (ms)$$

其中：

t_{WWDG}：WWDG 超时

t_{PCLK}：APB 时钟周期，以 ms 为测量单位

4096：对应于内部分频器的值

例如，假设 APB 频率等于 48 MHz，将 WDGTB[2:0] 设置为 3 并将 T[5:0] 设置为 63：

$$t_{WWDG} = (1/48000) \times 4096 \times 2^3 \times (63 + 1) = 43.69ms$$

有关 t_{WWDG} 的最小值和最大值，请参见数据手册。

44.3.7 调试模式

CPU 进入调试模式时，WWDG1 计数器会继续正常工作，或者会停止工作，具体取决于 DBGMCU_APB3LFZ1。有关详细信息，请参见*第 60 节：调试基础结构*。

44.4 WWDG 寄存器

有关寄存器说明中使用的缩写，请参见*第 94 页的第 1.1 节*。

外设寄存器可支持半字（16 位）或字（32 位）访问。

44.4.1 控制寄存器 (WWDG_CR)

Control register

偏移地址：0x00

复位值：0x0000 007F

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

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

位 31:8 保留，必须保持复位值。

位 7 **WDGA**：激活位 (Activation bit)
此位由软件置 1，只有复位后才由硬件清零。当 WDGA = 1 时，看门狗可产生复位。
0：禁止看门狗
1：使能看门狗

位 6:0 **T[6:0]**：7 位计数器 (7-bit counter)（MSB 到 LSB）
这些位用来存储看门狗计数器的值，每隔 $(4096 \times 2^{WDGTB[2:0]})$ PCLK 个周期递减一次。当它从 0x40 递减到 0x3F（T6 清零）时会产生复位。

44.4.2 配置寄存器 (WWDG_CFR)

Configuration register

偏移地址：0x04

复位值：0x0000 007F

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

位 31:14 保留，必须保持复位值。

位 13:11 **WDGTB[2:0]**：定时器时基 (Timer base)

可按如下方式修改预分频器的时基：

000：CK 计数器时钟 (PCLK div 4096) 分频器 1

001：CK 计数器时钟 (PCLK div 4096) 分频器 2

010：CK 计数器时钟 (PCLK div 4096) 分频器 4

011：CK 计数器时钟 (PCLK div 4096) 分频器 8

100：CK 计数器时钟 (PCLK div 4096) 分频器 16

101：CK 计数器时钟 (PCLK div 4096) 分频器 32

110：CK 计数器时钟 (PCLK div 4096) 分频器 64

111：CK 计数器时钟 (PCLK div 4096) 分频器 128

位 10 保留，必须保持复位值。

位 9 **EWI**：提前唤醒中断 (Early wakeup interrupt)

置 1 后，只要计数器值达到 0x40 就会产生中断。此中断只有在复位后才由硬件清零。

位 8:7 保留，必须保持复位值。

位 6:0 **W[6:0]**：7 位窗口值 (7-bit window value)

这些位包含用于与递减计数器进行比较的窗口值。

44.4.3 状态寄存器 (WWDG_SR)

Status register

偏移地址：0x08

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	EWIF
															rc_w0

位 31:1 保留，必须保持复位值。

位 0 **EWIF**：提前唤醒中断标志 (Early wakeup interrupt flag)
当计数器值达到 0x40 时此位由硬件置 1。它必须由软件通过写入 0 来清零。写入 1 不起作用。
如果不使能中断，此位也会被置 1。

44.4.4 WWDG 寄存器映射

下表提供了 WWDG 寄存器映射和复位值。

表 349. WWDG 寄存器映射和复位值

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	WWDG_CR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WDGA	T[6:0]						
	Reset value																									0	1	1	1	1	1	1	1
0x04	WWDG_CFR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WDGTB[2:0]			Res.	EWI	Res.	Res.	W[6:0]						
	Reset value																			0	0	0		0			1	1	1	1	1	1	1
0x08	WWDG_SR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	EWIF
	Reset value																																0

有关寄存器边界地址的信息，请参见*第 2.2.2 节：存储器映射和寄存器边界地址*。

45 独立看门狗 (IWDG)

45.1 前言

此器件具有一个嵌入式看门狗外设，具有安全性高、定时准确及使用灵活的优点。此独立看门狗外设可检测并解决由软件错误导致的故障，并在计数器达到给定的超时值时触发系统复位。

独立看门狗 (IWDG) 由其专用低速时钟 (LSI) 驱动，因此即便在主时钟发生故障时仍然保持工作状态。

IWDG 最适合应用于需要看门狗作为一个在主程序之外，能够完全独立工作，并且对时间精度要求较低的应用。有关窗口看门狗的详细信息，请参见*第 1729 页的第 44 节*。

45.2 IWDG 主要特性

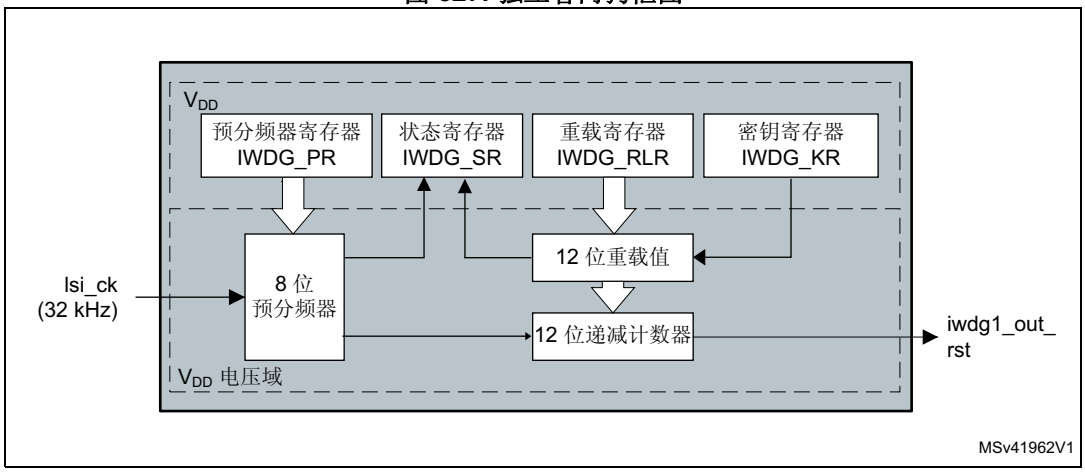
- 自由运行递减计数器
- 时钟由独立 RC 振荡器提供（可在待机和停止模式下运行）
- 复位条件
  - 当递减计数器值小于 0x000 时复位（如果看门狗已激活）
  - 在窗口之外重载递减计数器时复位（如果看门狗已激活）

45.3 IWDG 功能说明

45.3.1 IWDG 框图

图 527 给出了独立看门狗模块的功能框图。

图 527. 独立看门狗框图



1. 看门狗功能由 V_{DD} 电压域供电，在停止模式和待机模式下仍能工作。

通过向*键寄存器 (IWDG_KR)* 中写入值 0x0000 CCCC 来启动独立看门狗时，计数器开始从复位值 0xFFF 递减计数。当计数器计数到终值 (0x000) 时会产生一个复位信号（IWDG 复位）。

任何时候将键值 0x0000 AAAA 写到*键寄存器 (IWDG_KR)* 中，IWDG_RLR 的值就会被重载到计数器，从而避免产生看门狗复位。

45.3.2 IWDG 内部信号

表 350 给出了 IWDG 内部信号列表。

表 350. IWDG 内部输入/输出信号

信号名称	信号类型	说明
lsi_ck	数字输入	LSI 时钟
iwdg1_out_rst	数字输出	IWDG1 复位信号输出

45.3.3 窗口选项

通过在*窗口寄存器 (IWDG_WINR)* 中设置合适的窗口，IWDG 也可以用作窗口看门狗。

当计数器值大于*窗口寄存器 (IWDG_WINR)* 中存储的值时，如果执行重载操作，则会产生复位。

窗口寄存器 (IWDG_WINR) 的默认值为 0x0000 0FFF，因此，如果不更新此默认值，将禁止窗口选项。

窗口值一经更改，便执行重载操作，以便将递减计数器复位为*重载寄存器 (IWDG_RLR)* 值，并方便计算周期数以生成下一次重载。

使能窗口选项时配置 IWDG

1. 通过在*键寄存器 (IWDG_KR)* 中写入 0x0000 CCCC 来使能 IWDG。
2. 通过在*键寄存器 (IWDG_KR)* 中写入 0x0000 5555 来使能寄存器访问。
3. 通过将*预分频器寄存器 (IWDG_PR)* 编程为 0~7 中的数值来配置 IWDG 预分频器。
4. 对*重载寄存器 (IWDG_RLR)* 进行写操作。
5. 等待寄存器更新 (IWDG_SR = 0x0000 0000)。
6. 对*窗口寄存器 (IWDG_WINR)* 进行写操作。这会自动刷新*重载寄存器 (IWDG_RLR)* 中的计数器值。

注： *当状态寄存器 (IWDG_SR) 设置为 0x0000 0000 时，写入窗口值允许刷新计数器值为 RLR 的值。*

禁止窗口选项时配置 IWDG

不使用窗口选项时，可按以下步骤配置 IWDG：

1. 通过在*键寄存器 (IWDG_KR)* 中写入 0x0000 CCCC 来使能 IWDG。
2. 通过在*键寄存器 (IWDG_KR)* 中写入 0x0000 5555 来使能寄存器访问。
3. 通过将*预分频器寄存器 (IWDG_PR)* 编程为 0~7 中的数值来配置预分频器。
4. 对*重载寄存器 (IWDG_RLR)* 进行写操作。
5. 等待寄存器更新 (IWDG_SR = 0x0000 0000)。
6. 刷新计数器值为 IWDG_RLR 的值 (IWDG_KR = 0x0000 AAAA)。

45.3.4 硬件看门狗

如果通过器件选项位使能“硬件看门狗”功能，上电时将自动使能看门狗；如果在计数器计数结束前，若软件没有向*键寄存器 (IWDG_KR)* 写入相应的值，或者在窗口内部重载了递减计数器，则系统会产生复位。

45.3.5 低功耗冻结

根据 IWDG_FZ_STOP 和 IWDG_FZ_STBY 选项配置，IWDG 可分别在停止模式和待机模式期间继续计数或停止计数。如果停止模式或待机模式期间 IWDG 保持运行，它可从此模式唤醒器件。更多详细信息，请参见*第 3.3.12 节：保护机制*。

45.3.6 停止和待机模式下的行为

一旦运行，IWDG 便无法停止。

45.3.7 寄存器访问保护

预分频器寄存器 (IWDG_PR)、*重载寄存器 (IWDG_RLR)* 和*窗口寄存器 (IWDG_WINR)* 寄存器具有写访问保护。若要对其进行修改，用户必须首先对*键寄存器 (IWDG_KR)* 写入代码 0x0000 5555。而写入其他值则会破坏该序列，从而使寄存器访问保护再次生效。这表示重载操作（即写入 0x0000 AAAA）也会启动写保护功能。

状态寄存器指示预分频值、递减计数器重载值或窗口值是否正在被更新。

45.3.8 调试模式

当微控制器进入调试模式时（内核停止），IWDG 计数器会根据 DBG 模块中的 DBG_IWDG_STOP 配置位选择继续正常工作或者停止工作。

45.4 IWDG 寄存器

有关寄存器说明中使用的缩写，请参见*第 94 页的第 1.1 节*。

外设寄存器可支持半字（16 位）或字（32 位）访问。

45.4.1 键寄存器 (IWDG_KR)

Key register

偏移地址：0x00

复位值：0x0000 0000（待机模式时复位）

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

位 31:16 保留，必须保持复位值。

位 15:0 **KEY[15:0]**：键值 (Key value)（只能写，读为 0x0000）

必须每隔一段时间便通过软件对这些位写入键值 0xAAAA，否则当计数器计数到 0 时，看门狗会产生复位。

写入键值 0x5555 可使能对 IWDG_PR、IWDG_RLR 和 IWDG_WINR 寄存器的访问（请参见*第 45.3.7 节：寄存器访问保护*）

写入键值 0xCCCC 可启动看门狗（选中硬件看门狗选项的情况除外）

45.4.2 预分频器寄存器 (IWDG_PR)

Prescaler register

偏移地址：0x04

复位值：0x0000 0000

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

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

位 31:3 保留，必须保持复位值。

位 2:0 **PR[2:0]**：预分频系数 (Prescaler divider)

这些位受写访问保护，请参见*第 45.3.7 节：寄存器访问保护*。通过软件设置这些位来选择计数器时钟的预分频因子。若要更改预分频器的分频系数，*状态寄存器 (IWDG_SR)* 的 PVU 位必须为 0。

000：4 分频
001：8 分频
010：16 分频
011：32 分频
100：64 分频
101：128 分频
110：256 分频
111：256 分频

注： *读取该寄存器会返回 V_{DD} 电压域的预分频器值。如果正在对该寄存器执行写操作，则读取的值可能不是最新的/有效的。因此，只有在状态寄存器 (IWDG_SR) 中的 PVU 位为 0 时，从寄存器读取的值才有效。*

45.4.3 重载寄存器 (IWDG_RLR)

Reload register

偏移地址：0x08

复位值：0x0000 0FFF（待机模式时复位）

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

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

位 31:12 保留，必须保持复位值。

位 11:0 **RL[11:0]**：看门狗计数器重载值 (Watchdog counter reload value)

这些位受写访问保护，请参见*寄存器访问保护*。这个值由软件设置，每次对*键寄存器 (IWDG_KR)*写入值 0xAAAA 时，这个值就会重装载到看门狗计数器中。之后，看门狗计数器便从该装载的值开始递减计数。超时周期由该值和时钟预分频器共同决定。有关超时信息，请参见数据手册。

若要更改重载值，*状态寄存器 (IWDG_SR)* 中的 RVU 位必须为 0。

注： *读取该寄存器会返回 V_{DD} 电压域的重载值。如果正在对该寄存器执行写操作，则读取的值可能不是最新的/ 有效的。因此，只有在状态寄存器 (IWDG_SR) 中的 RVU 位为 0 时，从寄存器读取的值才有效。*

45.4.4 状态寄存器 (IWDG_SR)

Status register

偏移地址：0x0C

复位值：0x0000 0000（待机模式时不复位）

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WVU	RVU	PVU
													r	r	r

位 31:3 保留，必须保持复位值。

位 2 **WVU**：看门狗计数器窗口值更新 (Watchdog counter window value update)
可通过硬件将该位置 1 以指示窗口值正在更新。当在 V_{DD} 电压域下完成重载值更新操作后（需要多达 5 个 RC 40 kHz 周期），会通过硬件将该位复位。
窗口值只有在 WVU 位为 0 时才可更新。
此位只有在通用“窗口”= 1 时才生成

位 1 **RVU**：看门狗计数器重载值更新 (Watchdog counter reload value update)
可通过硬件将该位置 1 以指示重载值正在更新。当在 V_{DD} 电压域下完成重载值更新操作后（需要多达 5 个 RC 40 kHz 周期），会通过硬件将该位复位。
重载值只有在 RVU 位为 0 时才可更新。

位 0 **PVU**：看门狗预分频器值更新 (Watchdog prescaler value update)
可通过硬件将该位置 1 以指示预分频器值正在更新。当在 V_{DD} 电压域下完成预分频器值更新操作后（需要多达 5 个 RC 40 kHz 周期），会通过硬件将该位复位。
预分频器值只有在 PVU 位为 0 时才可更新。

注： *如果应用使用多个重载值、预分频器值或窗口值，则必须等到 RVU 位被复位后才能更改重载值，等到 PVU 位被复位后才能更改预分频器值，而且必须等到 WVU 位被复位后才能更改窗口值。但是，在更新预分频器和/或重载/窗口值之后，则无需等到 RVU、PVU 或 WVU 复位后再继续执行代码（进入低功耗模式时除外）。*

45.4.5 窗口寄存器 (IWDG_WINR)

Window register

偏移地址：0x10

复位值：0x0000 0FFF（待机模式时复位）

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

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

位 31:12 保留，必须保持复位值。

位 11:0 **WIN[11:0]**：看门狗计数器窗口值 (Watchdog counter window value)
这些位受写访问保护，请参见*第 45.3.7 节*，它们包含用于与递减计数器进行比较的窗口值上限。
为防止发生复位，当递减计数器的值低于窗口寄存器值且大于 0x0 时必须重载。
若要更改重载值，*状态寄存器 (IWDG_SR)* 中的 WVU 位必须为 0。

注： *读取该寄存器会返回 V_{DD} 电压域的重载值。如果正在对该寄存器执行写操作，则读取的值可能无效。因此，只有在状态寄存器 (IWDG_SR) 中的 WVU 位为 0 时，从寄存器读取的值才有效。*

45.4.6 IWDG 寄存器映射

下表提供了 IWDG 寄存器映射和复位值。

表 351. IWDG 寄存器映射和复位值

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	**IWDG_KR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	KEY[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x04	**IWDG_PR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PR[2:0]		
	Reset value																														0	0	0
0x08	**IWDG_RLR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RL[11:0]											
	Reset value																					1	1	1	1	1	1	1	1	1	1	1	1
0x0C	**IWDG_SR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WVU	RVU	PVU
	Reset value																														0	0	0
0x10	**IWDG_WINR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WIN[11:0]											
	Reset value																					1	1	1	1	1	1	1	1	1	1	1	1

有关寄存器边界地址的信息，请参见*第 2.2.2 节：存储器映射和寄存器边界地址*。

46 实时时钟 (RTC)

46.1 前言

实时时钟 (RTC) 提供用于管理所有低功耗模式的自动唤醒单元。

实时时钟 (RTC) 是一个独立的 BCD 定时器/计数器。RTC 提供具有可编程闹钟中断功能的日历时钟/日历。

RTC 还包含具有中断功能的周期性可编程唤醒标志。

两个 32 位寄存器包含 BCD 格式的秒、分钟、小时（12 或 24 小时制）、星期几、日期、月份和年份。此外，还可提供二进制格式的亚秒值。

系统可以自动将月份的天数补偿为 28、29（闰年）、30 和 31 天。并且还可以进行夏令时补偿。

其它 32 位寄存器还包含可编程的闹钟亚秒、秒、分钟、小时、星期几和日期。

此外，还可以使用数字校准功能对晶振精度的偏差进行补偿。

备份域复位后，所有 RTC 寄存器都会受到保护，以防止可能的非正常写访问。

无论器件状态如何（运行模式、低功耗模式或处于复位状态），只要电源电压保持在工作范围内，RTC 便不会停止工作。

46.2 RTC 主要特性

RTC 单元的主要特性如下（参见 *STM32F030x4/6STM32F03x、STM32F070x6STM32F04x 和STM32F030x8STM32F05x 器件的详细的 RTC 框图*）：

- 包含亚秒、秒、分钟、小时（12/24 小时制）、星期几、日期、月份和年份的日历。
- 软件可编程的夏令时补偿。
- 具有中断功能的可编程闹钟。可通过任意日历字段的组合触发闹钟。
- 自动唤醒单元，可周期性地生成标志以触发自动唤醒中断。
- 参考时钟检测：可使用更加精确的第二时钟源（50 Hz 或 60 Hz）来提高日历的精确度。
- 利用亚秒级移位特性与外部时钟实现精确同步。
- 数字校准电路（周期性计数器调整）：精度为 0.95 ppm，在数秒钟的校准窗口中获得
- 用于事件保存的时间戳功能。
- 带可配置过滤器和内部上拉的入侵检测事件。
- 可屏蔽中断/事件：
  - 闹钟 A
  - 闹钟 B
  - 唤醒中断
  - 时间戳
  - 入侵检测
- 32 个备份寄存器。

46.3 RTC 功能说明

46.3.1 RTC 框图

图 528. RTC 块概览

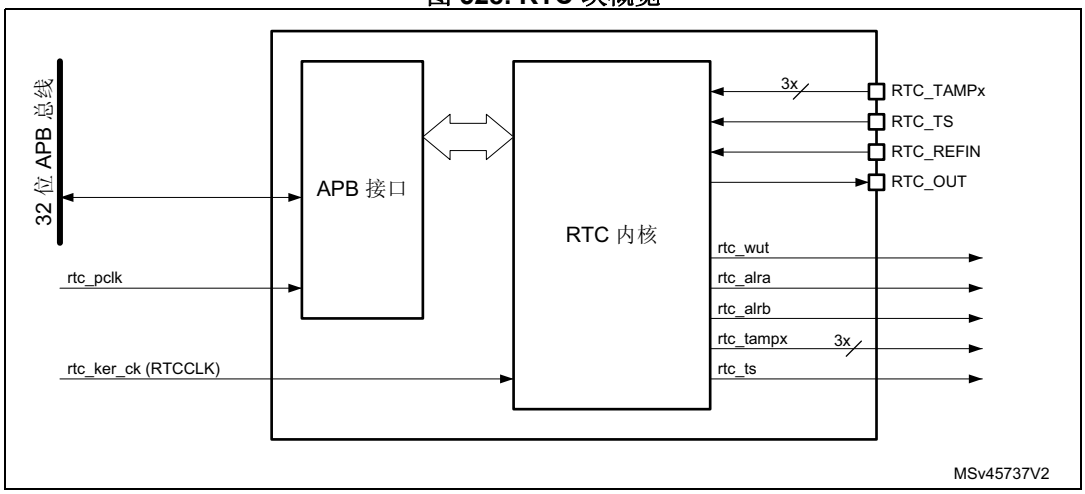


图 529. 详细的 RTC 框图

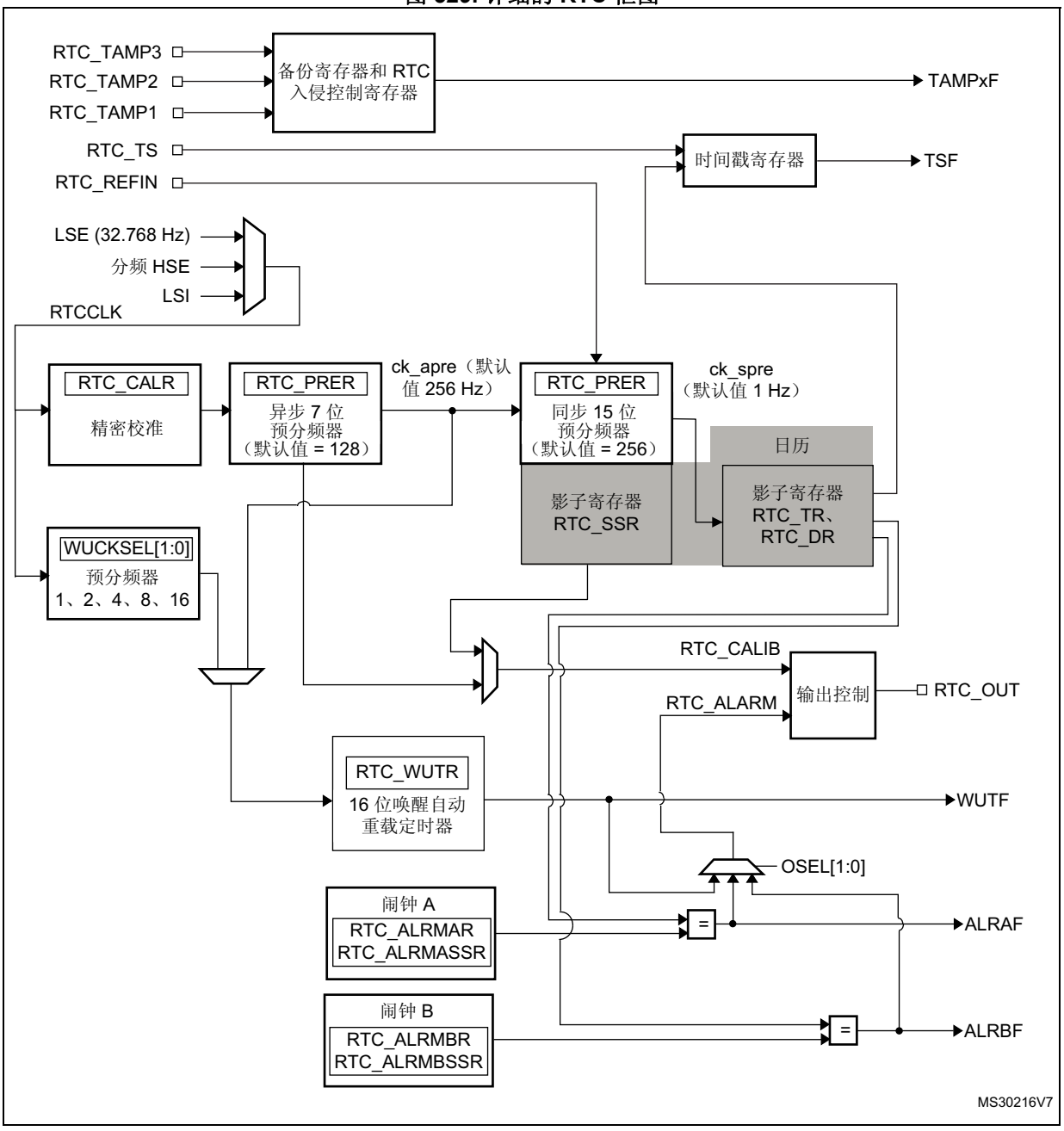
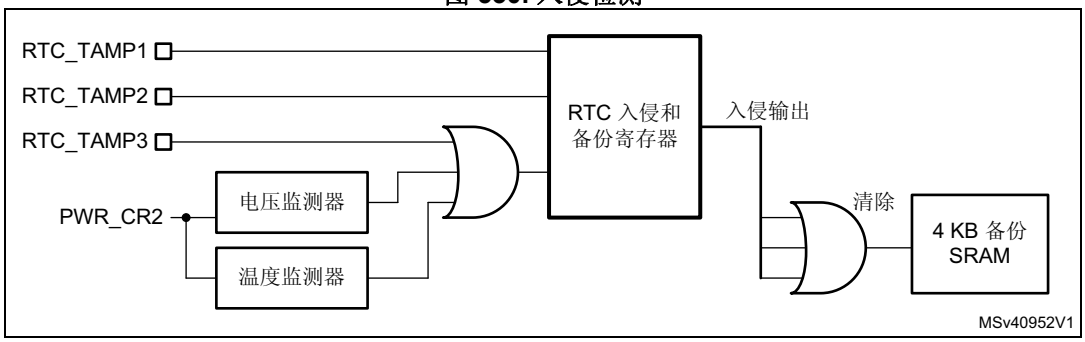


图 530. 入侵检测



RTC 包含：

- 两个闹钟
- 三个来自 I/O 的入侵事件
  - 入侵检测会擦除备份寄存器和备份 RAM。
  - 而且，在擦除操作完成之前，入侵检测禁止软件访问备份 SRAM。 请参见*第 46.3.15 节：入侵检测*。
  - I/O 上的事件、RTC 电源域的过压或欠压或者由过温或欠温检测生成 tamper3 事件检测。这些电压和温度监视检测在 *PWR 控制寄存器 2 (PWR_CR2)* 中使能。
- 来自 I/O 的一个时间戳事件
- 入侵事件检测可生成一个时间戳事件
- 切换到 V_{BAT} 时会生成时间戳
- 32 个 32 位备份寄存器
  - 备份寄存器 (RTC_BKPxR) 在 RTC 域中实现，可在 VDD 电源关闭时通过 VBAT 保持上电状态。
- 输出功能：RTC_OUT，可选择以下两个输出之一：
  - RTC_CALIB：512 Hz 或 1 Hz 时钟输出（LSE 频率为 32.768 kHz）。可通过将 RTC_CR 寄存器中的 COE 位置 1 来使能此输出。
  - RTC_ALARM：可通过配置 RTC_CR 寄存器中的 OSEL[1:0] 位使能此输出，可选择闹钟 A、闹钟 B 或唤醒输出。
- 输入功能：
  - RTC_TS：时间戳事件
  - RTC_TAMP1：tamper1 事件检测
  - RTC_TAMP2：tamper2 事件检测
  - RTC_TAMP3：tamper3 事件检测
  - RTC_REFIN：50 Hz 或 60 Hz 参考时钟输入

46.3.2 RTC 引脚和内部信号

表 352. RTC 引脚和内部信号

信号名称	信号类型	说明
RTC_TS	输入	时间戳输入
RTC_TAMPx（x = 1、2、3）	输入	入侵输入
RTC_REFIN	输入	参考时钟输入
RTC_OUT	输出	RTC 输出
rtc_ker_ck *(RTCCLK)*	内部输入	RTC 时钟源（LSE 时钟、LSI 时钟和 HSE 时钟）
rtc_pclk	内部输入	RTC APB 接口时钟
rtc_wut	内部输出	片上外设的 RTC 唤醒事件输出
rtc_alra	内部输出	片上外设的 RTC 报警 A 事件输出
rtc_alrb	内部输出	片上外设的 RTC 报警 B 事件输出
rtc_tampx	内部输出	片上外设的 RTC Tamper[1..3] 事件输出
rtc_ts	内部输出	片上外设的 RTC 时间戳事件输出

46.3.3 RTC 控制的 GPIO

RTC_OUT、RTC_TS 和 RTC_TAMP1 映射到同一引脚 (PC13)。无论 PC13 的 GPIO 配置如何，PC13 引脚的配置都由 RTC 控制，但 RTC_ALARM 输出开漏模式除外。映射在 PC13 上的 RTC 功能可用于所有低功耗模式和 VBAT 模式。

输出机制遵循*表 353* 中所示的优先级顺序。

表 353. RTC 引脚 PC13 配置(1)

PC13 引脚配置和功能	OSEL[1:0] 位（RTC_ALARM 输出使能）	COE 位（RTC_CALIB 输出使能）	RTC_OUT_RMP 位	RTC_ALARM_TYPE 位	TAMP1E 位（RTC_TAMP1 输入使能）	TSE位（RTC_TS 输入使能）
RTC_ALARM 输出 OD	01 或 10 或 11	无关	0	0	无关	无关
			1			
RTC_ALARM 输出 PP	01 或 10 或 11	无关	0	1	无关	无关
			1			
RTC_CALIB 输出 PP	00	1	0	无关	无关	无关
RTC_TAMP1 输入浮空	00	0	无关	无关	1	0
	00	1	1			
	01 或 10 或 11	0				
RTC_TS 和 RTC_TAMP1 输入浮空	00	0	无关	无关	1	1
	00	1	1			
	01 或 10 或 11	0				

表 353. RTC 引脚 PC13 配置[1]（续）

PC13 引脚配置和功能	OSEL[1:0] 位（RTC_ALARM 输出使能）	COE 位（RTC_CALIB 输出使能）	RTC_OUT_RMP 位	RTC_ALARM_TYPE 位	TAMP1E 位（RTC_TAMP1 输入使能）	TSE位（RTC_TS 输入使能）
RTC_TS 输入浮空	00	0	无关	无关	0	1
	00	1	1			
	01 或 10 或 11	0				
唤醒引脚或标准 GPIO	00	0	无关	无关	0	0
	00	1	1			
	01 或 10 或 11	0				

1. OD：开漏；PP：推挽。

此外，可借助 RTC_OUT_RMP 位将 RTC_OUT 重映射到 PB2 引脚上。在这种情况下，必须将 PB2 GPIO 寄存器配置为类型正确的复用功能。重映射功能如*表 354* 所示。

表 354. RTC_OUT 映射

OSEL[1:0] 位（RTC_ALARM 输出使能）	COE 位（RTC_CALIB 输出使能）	RTC_OUT_RMP 位	RTC_OUT 映射到 PC13	RTC_OUT 映射到 PB2
00	0	0	-	-
00	1		RTC_CALIB	-
01 或 10 或 11	无关		RTC_ALARM	-
00	0	1	-	-
00	1		-	RTC_CALIB
01 或 10 或 11	0		-	RTC_ALARM
01 或 10 或 11	1		RTC_ALARM	RTC_CALIB

下表汇总了所有模式下的 RTC 引脚和功能。

表 355. 各个模式下的 RTC 功能

引脚	RTC 功能	待机模式以外的所有低功耗模式下的功能	待机模式下的功能性	VBAT 模式下的功能性
PC13	RTC_TAMP1 RTC_TS RTC_OUT	是	是	是
PI8	RTC_TAMP2	是	是	是
PC1	RTC_TAMP3	是	是	是
PB2	RTC_OUT	是	否	否
PB15	RTC_REFIN	是	否	否

46.3.4 时钟和预分频器

RTC 时钟源 (RTCCLK) 通过时钟控制器从 LSE 时钟、LSI 振荡器时钟以及 HSE 时钟三者中选择。有关 RTC 时钟源配置的更多信息，请参见*第 8 节：复位和时钟控制 (RCC)*。

可编程的预分频器阶段可生成 1 Hz 的时钟，用于更新日历。为最大程度地降低功耗，预分频器分为 2 个可编程的预分频器（参见*图 529：详细的 RTC 框图*）：

- 一个通过 RTC_PRER 寄存器的 PREDIV_A 位配置的 7 位异步预分频器。
- 一个通过 RTC_PRER 寄存器的 PREDIV_S 位配置的 15 位同步预分频器。

注： *使用两个预分频器时，推荐将异步预分频器配置为较高的值，以最大程度降低功耗。*

要使用频率为 32.768 kHz 的 LSE 获得频率为 1 Hz 的内部时钟 (ck_spre)，需要将异步预分频系数设置为 128，并将同步预分频系数设置为 256。

分频系数的最小值为 1，最大值为 2^{22}。

这对应于约为 4 MHz 的最大输入频率。

f_{ck_apre} 可根据以下公式得出：

$$f_{CK_APRE} = \frac{f_{RTCCLK}}{PREDIV_A + 1}$$

ck_apre 时钟用于为二进制 RTC_SSR 亚秒递减计数器提供时钟。当该计数器计数到 0 时，会使用 PREDIV_S 的内容重载 RTC_SSR。

f_{ck_spre} 可根据以下公式得出：

$$f_{CK_SPRE} = \frac{f_{RTCCLK}}{(PREDIV_S + 1) \times (PREDIV_A + 1)}$$

ck_spre 时钟既可以用于更新日历，也可以用作 16 位唤醒自动重载定时器的时基。为获得较短的超时周期，还可以将 16 位唤醒自动重载定时器与经可编程的 4 位异步预分频器分频的 RTCCLK 一同运行（有关详细信息，请参见*第 46.3.7 节：周期性自动唤醒*）。

46.3.5 实时时钟和日历

RTC 日历时间和日期寄存器可通过与 PCLK（APB 时钟）同步的影子寄存器来访问。这些时间和日期寄存器也可以直接访问，这样可避免等待同步的持续时间。

- RTC_SSR 对应于亚秒
- RTC_TR 对应于时间
- RTC_DR 对应于日期

每隔两个 RTCCLK 周期，便将当前日历值复制到影子寄存器，并将 RTC_ISR 寄存器的 RSF 位置 1（请参见*第 46.6.4 节：RTC 初始化和状态寄存器 (RTC_ISR)*）。在停机和待机模式下不会执行复制操作。退出这两种模式时，影子寄存器会在最长 2 个 RTCCLK 周期后进行更新。

当应用读取日历寄存器时，它会访问影子寄存器的内容。也可以通过将 RTC_CR 寄存器的 BYPSHAD 控制位置 1 来直接访问日历寄存器。默认情况下，该位被清零，用户访问影子寄存器。

在 BYPSHAD=0 模式下读取 RTC_SSR、RTC_TR 或 RTC_DR 寄存器时，APB 时钟频率 (f_{APB}) 必须至少为 RTC 时钟频率 (f_{RTCCLK}) 的 7 倍。

影子寄存器通过系统复位来复位。

46.3.6 可编程闹钟

RTC 单元提供两个可编程闹钟，即闹钟 A 和闹钟 B。以下说明针对闹钟 A，但同样适用于闹钟 B。

可通过 RTC_CR 寄存器中的 ALRAE 位来使能可编程闹钟功能。如果日历亚秒、秒、分钟、小时、日期或日与闹钟寄存器 RTC_ALRMASSR 和 RTC_ALRMAR 中编程的值相匹配，则 ALRAF 标志会被置为 1。可通过 RTC_ALRMAR 寄存器的 MSKx 位以及 RTC_ALRMASSR 寄存器的 MASKSSx 位单独选择各日历字段。可通过 RTC_CR 寄存器中的 ALRAIE 位来使能闹钟中断。

注意： 如果选择秒字段（RTC_ALRMAR 中的 MSK1 位复位），则 RTC_PRER 寄存器中设置的同步预分频器分频系数必须至少为 3，才能确保闹钟正确地运行。

闹钟 A 和闹钟 B（如果已通过 RTC_CR 寄存器中的位 OSEL[1:0] 使能）可连接到 RTC_ALARM 输出。可通过 RTC_CR 寄存器的 POL 位配置 RTC_ALARM 输出极性。

46.3.7 周期性自动唤醒

周期性唤醒标志由 16 位可编程自动重载递减计数器生成。唤醒定时器范围可扩展至 17 位。

可通过 RTC_CR 寄存器中的 WUTE 位来使能此唤醒功能。

唤醒定时器的时钟输入可以是：

- 2、4、8 或 16 分频的 RTC 时钟 (RTCCLK)。

  当 RTCCLK 为 LSE (32.768 kHz) 时，可配置的唤醒中断周期介于 122 µs 和 32 s 之间，且分辨率低至 61 µs。
- ck_spre（通常为 1 Hz 内部时钟）

  当 ck_spre 频率为 1 Hz 时，可得到的唤醒时间为 1s 到 36h 左右，分辨率为 1 秒。这一较大的可编程时间范围分为两部分：
  - WUCKSEL [2:1] = 10 时为 1s 到 18h
  - WUCKSEL [2:1] = 11 时约为 18h 到 36h。在后一种情况下，会将 2^{16} 添加到 16 位计数器当前值。完成初始化序列后（请参见*第 1753 页的编程唤醒定时器*），定时器开始递减计数。在低功耗模式下使能唤醒功能时，递减计数保持有效。此外，当计数器计数到 0 时，RTC_ISR 寄存器的 WUTF 标志会置 1，并且唤醒寄存器会使用其重载值（RTC_WUTR 寄存器值）动重载。

之后必须用软件清零 WUTF 标志。

通过将 RTC_CR2 寄存器中的 WUTIE 位置 1 来使能周期性唤醒中断时，它会使器件退出低功耗模式。

如果已通过 RTC_CR 寄存器的位 OSEL[1:0] 使能周期性唤醒标志，则该标志可连接到 RTC_ALARM 输出。可通过 RTC_CR 寄存器的 POL 位配置 RTC_ALARM 输出极性。

系统复位以及低功耗模式（睡眠、停机和待机）对唤醒定时器没有任何影响。

46.3.8 RTC 初始化和配置

RTC 寄存器访问

RTC 寄存器为 32 位寄存器。除了当 BYPSHAD=0 时对日历影子寄存器执行的读访问之外，APB 接口会在访问 RTC 寄存器时引入 2 个等待周期。

RTC 寄存器写保护

系统复位后，可通过对 PWR_CR1 寄存器中的 DBP 位清零来保护 RTC 寄存器以防止非正常的写访问。必须将 DBP 位置 1 才能使能 RTC 寄存器的写访问。

备份域复位后，所有 RTC 寄存器均受到写保护。通过向写保护寄存器 (RTC_WPR) 写入一个密钥来使能对 RTC 寄存器的写操作。

要解锁所有 RTC 寄存器（RTC_TAMPCR、RTC_BKPxR、RTC_OR 和 RTC_ISR[13:8] 除外）的写保护，需要执行以下步骤：

1. 将“0xCA”写入 RTC_WPR 寄存器。
2. 将“0x53”写入 RTC_WPR 寄存器。

写入一个错误的关键字会再次激活写保护。

保护机制不受系统复位影响。

日历初始化和配置

要编程包括时间格式和预分频器配置在内的初始时间和日期日历值，需按照以下顺序操作：

1. 将 RTC_ISR 寄存器中的 INIT 位置 1 以进入初始化模式。在此模式下，日历计数器将停止工作并且其值可更新。
2. 轮询 RTC_ISR 寄存器中的 INITF 位。当 INITF 置 1 时进入初始化阶段模式。大约需要 2 个 RTCCLK 时钟周期（由于时钟同步）。
3. 要为日历计数器生成 1 Hz 时钟，应编程 RTC_PRER 寄存器中的两个预分频系数。
4. 在影子寄存器（RTC_TR 和 RTC_DR）中加载初始时间和日期值，然后通过 RTC_CR 寄存器中的 FMT 位配置时间格式（12 或 24 小时制）。
5. 通过清零 INIT 位退出初始化模式。随后，自动加载实际日历计数器值，在 4 个 RTCCLK 时钟周期后重新开始计数。

当初始化序列完成之后，日历开始计数。

注： *系统复位后，应用可读取 RTC_ISR 寄存器中的 INITS 标志，以检查日历是否已初始化。如果该标志为 0，表明自系统复位以来，日历还尚未初始化过，其年份字段一直还保持着备份域复位默认值 (0x00)。*

要在初始化之后读取日历，必须首先用软件检查 RTC_ISR 寄存器的 RSF 标志是否置 1。

夏令时

可通过 RTC_CR 寄存器的 SUB1H、ADD1H 和 BKP 位管理夏令时。

利用 SUB1H 或 ADD1H，软件只需单次操作便可在日历中减去或增加一个小时，无需执行整个初始化步骤。

此外，软件还可以使用 BKP 位来记录是否曾经执行过此操作。

编程闹钟

要对可编程的闹钟进行编程或更新，必须执行类似的步骤。以下步骤针对闹钟 A，但同样适用于闹钟 B。

1. 将 RTC_CR 中的 ALRAE 位清零以禁止闹钟 A。
2. 编程闹钟 A 寄存器 (RTC_ALRMASSR/RTC_ALRMAR)。
3. 将 RTC_CR 寄存器中的 ALRAE 位置 1 以再次使能闹钟 A。

注： *由于时钟同步的缘故，RTC_CR 寄存器的每次更改都将需要大约 2 个 RTCCLK 时钟周期来完成。*

编程唤醒定时器

要配置或更改唤醒定时器的自动重载值（RTC_WUTR 中的 WUT[15:0]），需要按照以下顺序操作：

1. 清零 RTC_CR 中的 WUTE 以禁止唤醒定时器。
2. 轮询 RTC_ISR 中的 WUTWF，直到该位置 1，以确保可以访问唤醒自动重载定时器和 WUCKSEL[2:0] 位。大约需要 2 个 RTCCLK 时钟周期（由于时钟同步）。
3. 编程唤醒自动重载值 WUT[15:0]，并选择唤醒时钟（RTC_CR 中的 WUCKSEL[2:0] 位）。将 RTC_CR 寄存器中的 WUTE 位置 1 以再次使能定时器。唤醒定时器重新开始递减计数。由于时钟同步的缘故，在 WUTE 清零后，WUTWF 位也会清零，但需要花费多达 2 个 RTCCLK 时钟周期。

46.3.9 读取日历

当 RTC_CR 寄存器中的 BYPSHAD 控制位清零时

要正确读取 RTC 日历寄存器（RTC_SSR、RTC_TR 和 RTC_DR），APB 时钟频率 (f_{PCLK}) 必须等于或大于 RTC 时钟频率 (f_{RTCCLK}) 的七倍。这可以确保同步机制的安全性。

如果 APB 时钟频率低于 RTC 时钟频率的七倍，则软件必须分两次读取日历时间寄存器和日期寄存器。这样，当两次读取的 RTC_TR 结果相同时，才能确保数据正确。否则必须执行第三次读访问。任何情况下，APB 的时钟频率都不能低于 RTC 的时钟频率。

每次将日历寄存器中的值复制到 RTC_SSR、RTC_TR 和 RTC_DR 影子寄存器时，RTC_ISR 寄存器中的 RSF 位都会置 1。每两个 RTCCLK 周期执行一次复制。为确保这 3 个值来自同一时刻点，读取 RTC_SSR 或 RTC_TR 时会锁定高阶日历影子寄存器中的值，直到读取 RTC_DR。为避免软件对日历执行读访问的时间间隔小于 2 个 RTCCLK 周期：第一次读取日历之后必须通过软件将 RSF 清零，并且软件必须等待到 RSF 置 1 之后才可再次读取 RTC_SSR、RTC_TR 和 RTC_DR 寄存器。

从低功耗模式（停机模式或待机模式）唤醒之后，必须通过软件将 RSF 清零。之后，软件必须等待至 RSF 再次置 1 之后才可以读取 RTC_SSR、RTC_TR 和 RTC_DR 寄存器。

RSF 位必须在唤醒之后而不是进入低功耗模式之前进行清零。

系统复位之后，软件必须等待至 RSF 置 1 之后才可以读取 RTC_SSR、RTC_TR 和 RTC_DR 寄存器。实际上，系统复位会将影子寄存器复位为其默认值。

初始化之后（请参见*第 1752 页的日历初始化和配置*），软件必须等待至 RSF 置 1 之后才可以读取 RTC_SSR、RTC_TR 和 RTC_DR 寄存器。

同步之后（请参见*第 46.3.11 节：RTC 同步*），软件必须等待至 RSF 置 1 之后才可以读取 RTC_SSR、RTC_TR 和 RTC_DR 寄存器。

当 RTC_CR 寄存器中的 BYPSHAD 控制位置 1 时（旁路影子寄存器）

读取日历寄存器时会直接从日历计数器获取值，这样便无需等待至 RSF 位置 1。这对于从低功耗模式（停机模式或待机模式）退出后的情况特别有用，因为影子寄存器在这些模式下不更新。

当 BYPSHAD 位置 1 时，如果在对寄存器的两次读访问之间出现 RTCCLK 沿，则不同寄存器的结果彼此可能不一致。此外，如果在读操作期间出现 RTCCLK 沿，则可能导致其中一个寄存器的值不正确。软件必须分两次读取所有寄存器，然后将两次结果加以比较来确认数据是否一致和正确。此外，软件也可以只比较两次读取日历寄存器得到的结果的最低位。

注： *当 BYPSHAD=1 时，读取日历寄存器的指令需要一个额外的 APB 周期才能完成。*

46.3.10 复位 RTC

日历影子寄存器（RTC_SSR、RTC_TR 和 RTC_DR）以及 RTC 状态寄存器 (RTC_ISR) 的某些位通过所有可用的系统复位源复位为各自的默认值。

相反，以下寄存器则通过备份域复位来复位为各自的默认值并且不受系统复位的影响：RTC 当前日历寄存器、RTC 控制寄存器 (RTC_CR)、预分频器寄存器 (RTC_PRER)、RTC 校准寄存器 (RTC_CALR)、RTC 移位寄存器 (RTC_SHIFTR)、RTC 时间戳寄存器（RTC_TSSSR、RTC_TSTR 和 RTC_TSDR）、RTC 入侵配置寄存器 RTC_TAMPCR)、RTC 备份寄存器 (RTC_BKPxR)、唤醒定时器寄存器 (RTC_WUTR)、闹钟 A 和闹钟 B 寄存器（RTC_ALRMASSR/RTC_ALRMAR 和 RTC_ALRMBSSR/RTC_ALRMBR）以及选项寄存器 (RTC_OR)。

此外，当由 LSE 提供时钟时，如果复位源并非备份域复位源（有关不受系统复位影响的 RTC 时钟源列表的详细信息，请参见“复位和时钟控制器”的“RTC 时钟”部分），则 RTC 将在系统复位时保持运行状态。发生备份域复位时，RTC 会停止工作，并且所有 RTC 寄存器都会设置为各自的复位值。

46.3.11 RTC 同步

RTC 可与高精度的远程时钟同步。在读取亚秒字段后（RTC_SSR 或 RTC_TSSSR），即可计算远程时钟的时间与 RTC 之间的精准偏差。之后，可使用 RTC_SHIFTR 对 RTC 的时钟进行零点几秒的“平移”，经过调整后可消除此偏差。

RTC_SSR 包含同步预分频器计数器的值。这样，便可计算分辨率低至 1/(PREDIV_S + 1) 秒的 RTC 的准确时间。因此，可通过增大同步预分频器的值 (PREDIV_S[14:0]) 来提高分辨率。将 PREDIV_S 设置为 0x7FFF 时，可得到允许的最大分辨率（30.52 µs，时钟频率为 32768 Hz）。

但是，提高 PREDIV_S 意味着必须降低 PREDIV_A 才能将同步预分频器的输出维持在 1 Hz。这样，异步预分频器的输出频率会增大，RTC 的动态功耗也会相应增加。

可以使用 RTC 平移控制寄存器 (RTC_SHIFTR) 对 RTC 进行微调。可以用大小为 1/(PREDIV_S + 1) 秒的分辨率对 RTC_SHIFTR 进行写操作，将时钟平移（延迟或提前）最长 1 秒。在这种平移操作中，会将 SUBFS[14:0] 值加到同步预分频器计数器 SS[15:0] 中：这将使时钟产生延迟。如果同时将 ADD1S 位置 1，则会增加一秒，与此同时减去的时间为零点几秒，因此将使时钟提前。

注意： 初始化平移操作前，用户必须检查确认 SS[15] = 0，以确保不会发生上溢。

对 RTC_SHIFTR 寄存器执行写操作以启动平移操作时，硬件会将 SHPF 标志置 1 以指示平移操作挂起。完成平移操作时，硬件会将将该位清零。

注意： 该同步功能与参考时钟检测功能不兼容：当 REFCKON=1 时，固件不能对 RTC_SHIFTR 执行写操作。

46.3.12 RTC 参考时钟检测

RTC 日历更新可与参考时钟 RTC_REFIN（通常为市电频率，50 Hz 或 60 Hz）同步。RTC_REFIN 参考时钟的精度应高于 32.768 kHz LSE 时钟。使能 RTC_REFIN 检测时（将 RTC_CR 的 REFCKON 位置 1），日历仍由 LSE 提供时钟，而 RTC_REFIN 用于补偿不准确的日历更新频率 (1 Hz)。

每个 1 Hz 时钟边沿都与最近的 RTC_REFIN 时钟边沿进行比较（如果在给定的时间窗口内发现一个边沿）。在大多数情况下，两个时钟边沿恰好对齐。当 1 Hz 时钟由于 LSE 时钟不精确而发生偏离时，RTC 会稍微偏移 1 Hz 时钟，以便后续的 1 Hz 时钟边沿能够对齐。利用这种机制，可使日历像参考时钟一样精确。

RTC 使用 32.768 kHz 石英产生的 256 Hz 时钟 (ck_apre) 检测是否存在参考时钟源。大约在日历每次更新时（每 1 秒钟），便会在时间窗口期间执行一次检测。检测到第一个参考时钟边沿时，该窗口等于 7 个 ck_apre 周期。随后的日历更新使用长度为 3 个 ck_apre 周期的较小窗口。

每次在窗口中检测到参考时钟时，都会行强制输出 ck_apre 时钟的异步预分频器进行重载。当参考时钟与 1 Hz 时钟对齐时，此操作不起作用，因为预分频器会在同一时刻重载。当时钟不对齐时，重载操作会微调后续的 1 Hz 时钟边沿，使其与参考时钟对齐。

如果参考时钟停止（在 3 个 ck_apre 窗口内未出现参考时钟边沿），日历将仅根据 LSE 时钟进行连续更新。RTC 随后使用 ck_spre 边沿上居中的大检测窗口（7 个 ck_apre 周期）等待参考时钟。

使能 RTC_REFIN 检测后，必须将 PREDIV_A 和 PREDIV_S 设置为各自的默认值：

- PREDIV_A = 0x007F
- PREVID_S = 0x00FF

注： *RTC_REFIN 时钟检测在待机模式下不可用。*

46.3.13 RTC 精密数字校准

RTC 频率可采用约 0.954 ppm 的分辨率进行数字校准，校准范围为 -487.1 ppm 到 +488.5 ppm。使用一系列微调（增加和/或减少单独的 RTCCLK 脉冲）进行频率校正。这些微调的分布非常均匀，因此 RTC 的校准效果相当好，即使在短时间内持续观察也是如此。

当输入频率为 32768 Hz 时，精密数字校准的周期约为 2^{20} 个 RTCCLK 脉冲或 32 秒。此周期由一个通过 RTCCLK 提供时钟信号的 20 位计数器 cal_cnt[19:0] 维持。

精密数字校准寄存器 (RTC_CALR) 可指定 32 秒周期内要减少的 RTCCLK 时钟周期数：

- 将位 CALM[0] 置 1 时，32 秒周期内将只减少 1 个脉冲。
- 将 CALM[1] 置 1 时，将减少两个周期。
- 将 CALM[2] 置 1 时，将减少 4 个周期
- 依此类推，将 CALM[8] 置 1 时，将减少 256 个时钟。

注： *CALM[8:0] (RTC_CALR) 可指定 32 秒周期内要减少的 RTCCLK 脉冲数。将位 CALM[0] 置 1 时，32 秒周期内将只减少 1 个脉冲（当 cal_cnt[19:0] = 0x80000 时）；将 CALM[1] 置 1 时，将减少 2 个周期（当 cal_cnt = 0x40000 和 0xC0000 时）；将 CALM[2] 置 1 时，将减少 4 个周期（当 cal_cnt = 0x20000/0x60000/0xA0000/0xE0000 时）；依此类推，将 CALM[8] 置 1 时，将减少 256 个时钟（当 cal_cnt = 0xXX800 时）。*

使用适当分辨率时，CALM 可使 RTC 频率减少最多 487.1 ppm，而 CALP 可用于使频率增加 488.5 ppm。将 CALP 置“1”，可每隔 2^{11} 个 RTCCLK 周期有效插入一个额外的 RTCCLK 脉冲，这意味着每 32 秒周期可增加 512 个时钟。

与 CALM 和 CALP 配合使用时，可在 32 秒周期内增加一个范围为 -511 到 +512 RTCCLK 周期的偏差，对应的校准范围为 -487.1 ppm 到 +488.5 ppm，分辨率约为 0.954 ppm。

若输入频率 (F_{RTCCLK}) 已知，可通过以下公式计算有效校准频率 (F_{CAL})：

$$F_{CAL} = F_{RTCCLK} \times [1 + (CALP \times 512 - CALM) / (2^{20} + CALM - CALP \times 512)]$$

PREDIV_A<3 条件下的校准

当异步预分频器值（RTC_PRER 寄存器中的 PREDIV_A 位）小于 3 时，不能将 CALP 位置 1。如果 CALP 已置 1 并且 PREDIV_A 位的值小于 3，则会忽略 CALP，即假定 CALP 等于 0 而执行校准。

要在 PREDIV_A 小于 3 的条件下执行校准，应降低同步预分频器值 (PREDIV_S) 以便每秒内可加速 8 个 RTCCLK 时钟周期，这意味着每 32 秒可增加 256 个时钟周期。因此，仅使用 CALM 位，可在每 32 秒内有效增加 255 到 256 个时钟脉冲（对应的校准范围为 243.3 ppm 到 244.1 ppm）。

在标称 RTCCLK 频率 32768 Hz 下，当 PREDIV_A 等于 1 时（分频系数为 2），应将 PREDIV_S 设置为 16379 而不是 16383（少 4）。唯一相关的其它情况是，当 PREDIV_A 等于 0 时，应将 PREDIV_S 设置为 32759 而不是 32767（少 8）。

如果以这种方式减少 PREDIV_S，则采用以下公式计算校准输入时钟的有效频率：

$$F_{CAL} = F_{RTCCLK} \times [1 + (256 - CALM) / (2^{20} + CALM - 256)]$$

在这种情况下，如果 RTCCLK 恰好为 32768.00 Hz，则当 CALM[7:0] 等于 0x100 时（CALM 范围的中值），说明设置正确。

验证 RTC 校准

通过测量 RTCCLK 的精确频率，计算正确的 CALM 和 CALP 值以确保 RTC 精度。此外，还为应用提供了一个可选的 1 Hz 输出，用来测量和验证 RTC 精度。

如果在有限的间隔内测量 RTC 的精确频率，则会导致测量期间产生最多 2 个 RTCCLK 时钟周期的测量误差，具体取决于数字校准周期与测量周期的对齐方式。

但是，如果测量周期与校准周期的长度相同，则可以消除此测量误差。在这种情况下，观测到的唯一误差是由数字校准的分辨率导致的误差。

- 默认情况下，校准周期为 32 秒。

在此模式下，测量整个 32 秒内 1 Hz 输出的精度，可确保测量误差在 0.477 ppm 内（32 秒内为 0.5 个 RTCCLK 周期，受校准分辨率限制）。

- 可将 RTC_CALR 寄存器的 CALW16 位置 1，以强制 16 秒的校准周期。

此时，可在 16 秒内测量 RTC 精度，产生的最大误差为 0.954 ppm（16 秒内为 0.5 个 RTCCLK 周期）。但是，由于校准分辨率降低，长期的 RTC 精度也会降到 0.954 ppm：将 CALW16 置 1 时，CALM[0] 位将始终保持为 0。

- 可将 RTC_CALR 寄存器的 CALW8 位置 1，以强制 8 秒的校准周期。

此时，可在 8 秒内测量 RTC 精度，产生的最大误差为 1.907 ppm（8 秒内为 0.5 个 RTCCLK 周期）。长期的 RTC 精度也会降到 1.907 ppm：将 CALW8 置 1 时，CALM[1:0] 位将始终保持为 00。

动态重校准

当 RTC_ISR/INITF=0 时，可动态更新校准寄存器 (RTC_CALR)，具体步骤如下：

1. 轮询 RTC_ISR/RECALPF（重新校准挂起标志）。
2. 如果该标志为 0，则可以根据需要向 RTC_CALR 写入新值。随后 RECALPF 位会被自动置为 1。
3. 新校准设置将在对 RTC_CALR 执行写操作之后的三个 ck_apre 周期内生效。

46.3.14 时间戳功能

将 RTC_CR 寄存器的 TSE 位或 ITSE 位置 1 可使能时间戳。

将 TSE 置 1 时：

当在 RTC_TS 引脚上检测到时间戳事件时，日历会保存到时间戳寄存器（RTC_TSSSR、RTC_TSTR 和 RTC_TSDR）中。

将 ITSE 置 1 时：

当检测到内部时间戳事件时，日历会保存到时间戳寄存器（RTC_TSSSR、RTC_TSTR 和 RTC_TSDR）中。切换至 VBAT 电源可生成内部时间戳事件。

由于内部事件或外部事件而发生时间戳事件时，RTC_ISR 寄存器中的时间戳标志位 (TSF) 将置 1。如果是内部事件，RTC_ISR 寄存器中的 ITSF 标志也将置 1。

通过将 RTC_CR 寄存器中的 TSIE 位置 1，可在发生入侵检测事件时生成中断。

如果在时间戳标志 (TSF) 已置 1 的条件下检测到新的时间戳事件，则时间戳上溢标志 (TSOVF) 将置 1，而时间戳寄存器（RTC_TSTR 和 RTC_TSDR）将保持上一事件的结果。

注： *在发生由同步过程引发的时间戳事件后，TSF 在 2 个 ck_apre 周期置为 1。*

TSOVF 的产生中不存在延迟。也就是说如果两个时间戳事件接连发生，TSOVF 可能为“1”而TSF为“0”。因此，建议只在检测到 TSF 为“1”后再轮询 TSOVF。

注意： 如果在 TSF 位清零后紧接着发生时间戳事件，则 TSF 和 TSOVF 位都将置 1。为防止在时间戳事件发生的同时屏蔽该事件，除非已将 TSF 位读取为“1”，否则应用程序不得将“0”写入 TSF 位。

此外，入侵事件可能导致时间戳被记录。有关 TAMPTS 控制位的说明，请参见*第 46.6.16 节：RTC 入侵配置寄存器 (RTC_TAMPCR)*。

46.3.15 入侵检测

RTC_TAMPx 输入事件既可配置为边沿检测，也可配置为带过滤的电平检测。

入侵检测可配置用于以下目的：

- 擦除 RTC 备份寄存器和备份 SRAM（默认配置）
- 生成中断，能够从停止模式和待机模式唤醒
- 为低功耗定时器生成硬件触发

RTC 备份寄存器

备份寄存器 (RTC_BKPxR) 不会通过系统复位来复位，也不会在器件从待机模式唤醒时复位。

备份寄存器在发生入侵检测事件（请参见*第 46.6.20 节：RTC 备份寄存器 (RTC_BKPxR)* 和*第 1758 页的入侵检测初始化*）时复位，TAMPxNOERASE 位置 1 时或者 RTC_TAMPCR 寄存器中的 TAMPxMF 置 1 时除外。

入侵检测初始化

可通过将 RTC_TAMPCR 寄存器中相应的 TAMPxE 位置 1 来使能各输入。

每个 RTC_TAMPx 入侵检测输入分别与 RTC_ISR 寄存器中的标志 TAMPxF 相关联。

将 TAMPxMF 清零时：

TAMPxF 标志将在引脚上出现入侵事件后使能，并存在下述延迟：

- 当 TAMPFLT 不为 0x0 时（带过滤的电平检测），延迟为 3 个 ck_apre 周期
- 当 TAMPTS=1 时（入侵事件的时间戳），延迟为 3 个 ck_apre 周期
- 当 TAMPFLT=0x0（边沿检测）且 TAMPTS=0 时，无延迟

在此期间，只要 TAMPxF 置 1，就无法检测到同一引脚上出现的新入侵。

将 TAMPxMF 置 1 时：

在上述延迟期间以及在 2.5 个 ck_rtc 附加周期内，无法检测到同一引脚上出现的新入侵。

通过将 RTC_TAMPCR 寄存器中的 TAMPIE 位置 1，可在发生入侵检测事件时（当 TAMPxF 置 1 时）生成中断。当一个或多个 TAMPxMF 置 1 时，不允许将 TAMPIE 置 1。

将 TAMPIE 清零时，可通过将 RTC_TAMPCR 寄存器中相应的 TAMPxIE 位置 1 来分别使能各个入侵引脚事件中断。当相应的 TAMPxMF 置 1 时，不允许将 TAMPxIE 置 1。

出现入侵事件时生成触发输出

低功耗定时器可将入侵事件检测用作触发输入。

将 RTC_TAMPCR 寄存器中的 TAMPxMF 位清零时，必须通过软件将 TAMPxF 标志清零以便在同一引脚上检测新入侵。

将 TAMPxMF 位置 1 时，TAMPxF 标志会被屏蔽，并在 RTC_ISR 寄存器中保持清零。此配置允许在停止模式下自动触发低功耗定时器，无需通过系统唤醒来执行 TAMPxF 清零。在这种情况下，备份寄存器不清零。

入侵事件的时间戳

当 TAMPTS 置“1”时，任何入侵事件都会导致时间戳事件发生。在这种情况下，如同发生正常时间戳事件一样，RTC_ISR 中的 TSF 位或 TSOVF 位会置 1。在 TSF 或 TSOVF 置 1 的同时，受影响的入侵标志寄存器 TAMPxF 也会随之置 1。

对入侵输入的边沿检测

如果 TAMPFLT 位设置为“00”，当相应的 TAMPxTRG 位上出现上升沿或下降沿时，RTC_TAMPx 引脚将生成入侵检测事件。选择边沿检测时，会禁止 RTC_TAMPx 输入上的内部上拉电阻。

注意： 使用边沿检测时，建议在使能入侵检测后（通过读取 GPIO 寄存器）以及向备份寄存器中写入敏感值前立即通过软件检查入侵引脚电平，以确保使能入侵事件检测后才出现有效边沿。当 TAMPFLT = “00”且 TAMPxTRG = 0（上升沿检测）时，如果在使能入侵检测前入侵输入已处于高电平，则可通过硬件来检测入侵事件。

检测到入侵事件并清零后，应当在重新编程备份寄存器 (RTC_BKPxR) 之前禁止 RTC_TAMPx，然后再重新使能（TAMPxE 置 1）。这可防止应用在 RTC_TAMPx 输入值仍指示入侵检测时，对备份寄存器执行写操作。这相当于对 RTC_TAMPx 输入的电平检测。

注： *当 V_{DD} 电源关闭时，入侵检测仍有效。要避免意外复位备份寄存器，应将 RTC_TAMPx 映射到的引脚从外部连接到正确的电平。*

对 RTC_TAMPx 输入的带过滤电平检测

通过将 TAMPFLT 设置为非零值可执行带过滤的电平检测。在 TAMPxTRG 位指定的电平连续出现 2、4 或 8 个（取决于 TAMPFLT 值）采样时生成入侵检测事件。

除非通过将 TAMPPUDIS 置 1 禁止入侵输入，否则在入侵输入的状态被采样前，将通过 I/O 内部上拉电阻对 RTC_TAMPx 输入预充电。预充电的持续时间由 TAMPPRCH 位确定，允许增大 RTC_TAMPx 输入上的电容。

可使用 TAMPFREQ 确定用于电平检测的采样频率，以便使入侵检测延迟与上拉电阻功耗之间达到最佳平衡。

注： *有关上拉电阻的电气特性，请参见数据手册。*

46.3.16 校准时钟输出

将 RTC_CR 寄存器中的 COE 位置 1 时，会在 RTC_CALIB 器件输出上提供一个参考时钟。

如果 RTC_CR 寄存器中的 COSEL 位复位且 PREDIV_A = 0x7F，则 RTC_CALIB 频率为 $f_{RTCCLK/64}$。这相当于 RTCCLK 频率为 32.768 kHz 时，512 Hz 的校准输出。RTC_CALIB 占空比是不规则的：下降沿上存在轻微抖动。因此推荐使用上升沿。

如果 COSEL 置 1 且“PREDIV_S+1”为 256 的非零整数倍（比如：PREDIV_S[7:0] = 0xFF），则 RTC_CALIB 频率为 $f_{RTCCLK}/(256 * (PREDIV_A+1))$。这相当于 RTCCLK 频率为 32.768 kHz 时， 1 Hz 的校准输出，其中预分频器为默认值 （PREDIV_A = Ox7F、 PREDIV_S = 0xFF）。1 Hz 输出会在进行平移操作时受到影响，并且可能在平移操作期间发生翻转 (SHPF = 1)。

注： *选择 RTC_CALIB 或 RTC_ALARM 输出时，RTC_OUT 引脚会自动配置为输出。*

COSEL 位清零时，RTC_CALIB 输出为异步预分频器的第 6 级输出。

COSEL 位置 1 时，RTC_CALIB 输出为同步预分频器的第 8 级输出。

46.3.17 闹钟输出

RTC_CR 寄存器中的 OSEL[1:0] 控制位用于激活闹钟输出 RTC_ALARM，以及选择输出的功能。这些功能可反映 RTC_ISR 寄存器中相应标志的内容。

输出的极性由 RTC_CR 中的 POL 控制位确定，这样当 POL 置 1 时会输出选定标志位的相反值。

闹钟输出

使用 RTC_OR 寄存器中的控制位 RTC_ALARM_TYPE 可将 RTC_ALARM 引脚配置为输出开漏或输出上拉。

注： *使能 RTC_ALARM 输出之后，其优先级高于 RTC_CALIB（与 COE 位无关，该为必须保持清零）。*

选择 RTC_CALIB 或 RTC_ALARM 输出时，RTC_OUT 引脚会自动配置为输出。

46.4 RTC 低功耗模式

表 356. 低功耗模式对 RTC 的作用

模式	说明
停止	外设寄存器内容仍被保持。
待机	当 RTC 时钟源为 LSE 或 LSI 时，RTC 保持工作状态。RTC 闹钟、RTC 入侵事件、RTC 时间戳事件和 RTC 唤醒会使器件退出待机模式。

46.5 RTC 中断

所有 RTC 中断均与 EXTI 控制器相连。请参见*第 20 节：扩展中断和事件控制器 (EXTI)*。

要使能 RTC 闹钟中断，需按照以下顺序操作：

1. 将对应于 RTC 闹钟事件的 EXTI 线配置为中断模式并将其使能，然后选择上升沿有效。
2. 配置 NVIC 中的 RTC_ALARM IRQ 通道并将其使能。
3. 配置 RTC 以生成 RTC 闹钟。

要使能 RTC 入侵中断，需按照以下顺序操作：

1. 将对应于 RTC 入侵事件的 EXTI 线配置为中断模式并将其使能，然后选择上升沿有效。
2. 配置 NVIC 中的 RTC_TAMP_STAMP IRQ 通道并将其使能。
3. 配置 RTC 以检测 RTC 入侵事件。

要使能 RTC 时间戳中断，需按照以下顺序操作：

1. 将对应于 RTC 时间戳事件的 EXTI 线配置为中断模式并将其使能，然后选择上升沿有效。
2. 配置 NVIC 中的 RTC_TAMP_STAMP IRQ 通道并将其使能。
3. 配置 RTC 以检测 RTC 时间戳事件。

要使能唤醒定时器中断，需按照以下顺序操作：

1. 将对应于唤醒定时器事件的 EXTI 线配置为中断模式并将其使能，然后选择上升沿有效。
2. 配置 NVIC 中的 RTC_WKUP IRQ 通道并将其使能。
3. 配置 RTC 以检测 RTC 唤醒定时器事件。

表 357. 中断控制位

中断事件	事件标志	启用控制位	退出睡眠模式	退出停止模式	退出待机模式
闹钟 A	ALRAF	ALRAIE	是	是(1)	是(1)
闹钟 B	ALRBF	ALRBE	是	是(1)	是(1)
RTC_TS 输入（时间戳）	TSF	TSIE	是	是(1)	是(1)
RTC_TAMP1 输入检测	TAMP1F	TAMPIE	是	是(1)	是(1)
RTC_TAMP2 输入检测	TAMP2F	TAMPIE	是	是(1)	是(1)
RTC_TAMP3 输入检测	TAMP3F	TAMPIE	是	是(1)	是(1)
唤醒定时器中断	WUTF	WUTIE	是	是(1)	是(1)

1. 仅当 RTC 时钟源为 LSE 或 LSI 时，才能从停机和待机模式唤醒。

46.6 RTC 寄存器

有关寄存器说明中使用的缩写，请参见参考手册中的*第 94 页的第 1.1 节*。

外设寄存器可按字（32 位）进行访问。

46.6.1 RTC 时间寄存器 (RTC_TR)

RTC time register

RTC_TR 是日历时间影子寄存器。只能在初始化模式下对该寄存器执行写操作。请参见*第 1752 页的日历初始化和配置*和*第 1753 页的读取日历*。

此寄存器受写保护。*第 1752 页的RTC 寄存器写保护*中介绍了写访问的过程。

偏移地址：0x00

备份域复位值：0x0000 0000

系统复位：当 BYPSHAD = 0 时为 0x0000 0000。当 BYPSHAD = 1 时不受影响。

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

位 31-23 保留，必须保持复位值

位 22 **PM**：AM/PM 符号 (AM/PM notation)
0：AM 或 24 小时制
1：PM

位 21:20 **HT[1:0]**：小时的十位（BCD 格式）(Hour tens in BCD format)

位 19:16 **HU[3:0]**：小时的个位（BCD 格式）(Hour units in BCD format)

位 15 保留，必须保持复位值。

位 14:12 **MNT[2:0]**：分钟的十位（BCD 格式）(Minute tens in BCD format)

位 11:8 **MNU[3:0]**：分钟的个位（BCD 格式）(Minute units in BCD format)

位 7 保留，必须保持复位值。

位 6:4 **ST[2:0]**：秒的十位（BCD 格式）(Second tens in BCD format)

位 3:0 **SU[3:0]**：秒的个位（BCD 格式）(Second units in BCD format)

46.6.2 RTC 日期寄存器 (RTC_DR)

RTC date register

RTC_DR 是日历日期影子寄存器。只能在初始化模式下对该寄存器执行写操作。请参见*第 1752 页的日历初始化和配置*和*第 1753 页的读取日历*。

此寄存器受写保护。*第 1752 页的RTC 寄存器写保护*中介绍了写访问的过程。

偏移地址：0x04

备份域复位值：0x0000 2101

系统复位：当 BYPSHAD = 0 时为 0x0000 2101。当 BYPSHAD = 1 时不受影响。

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

位 31:24 保留，必须保持复位值

位 23:20 **YT[3:0]**：年份的十位（BCD 格式）(Year tens in BCD format)

位 19:16 **YU[3:0]**：年份的个位（BCD 格式）(Year units in BCD format)

位 15:13 **WDU[2:0]**：星期几的个位 (Week day units)

000：禁止
001：星期一
...
111：星期日

位 12 **MT**：月份的十位（BCD 格式）(Month tens in BCD format)

位 11:8 **MU**：月份的个位（BCD 格式）(Month units in BCD format)

位 7:6 保留，必须保持复位值。

位 5:4 **DT[1:0]**：日期的十位（BCD 格式）(Date tens in BCD format)

位 3:0 **DU[3:0]**：日期的个位（BCD 格式）(Date units in BCD format)

46.6.3 RTC 控制寄存器 (RTC_CR)

RTC control register

偏移地址：0x08

备份域复位值：0x0000 0000

系统复位：不受影响

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	ITSE	COE	OSEL[1:0]		POL	COSEL	BKP	SUB1H	ADD1H
							rw	rw	rw	rw	rw	rw	rw	w	w

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TSIE	WUTIE	ALRBIE	ALRAIE	TSE	WUTE	ALRBE	ALRAE	Res.	FMT	BYPSHAD	REFCKON	TSEDGE	WUCKSEL[2:0]		
rw	rw	rw	rw	rw	rw	rw	rw		rw	rw	rw	rw	rw	rw	rw

位 31:25 保留，必须保持复位值。

位 24 **ITSE**：内部事件时间戳使能 (timestamp on internal event enable)
0：禁止内部事件时间戳
1：使能内部事件时间戳

位 23 **COE**：校准输出使能 (Calibration output enable)
该位使能 RTC_CALIB 输出
0：禁止校准输出
1：使能校准输出

位 22:21 **OSEL[1:0]**：输出选择 (Output selection)
这些位用于选择要连接到 RTC_ALARM 输出的标志
00：禁止输出
01：使能闹钟 A 输出
10：使能闹钟 B 输出
11：使能唤醒输出

位 20 **POL**：输出极性 (Output polarity)
该位用于配置 RTC_ALARM 输出的极性
0：当 ALRAF/ALRBF/WUTF 置 1 时（取决于 OSEL[1:0]），该引脚为高电平
1：当 ALRAF/ALRBF/WUTF 置 1 时（取决于 OSEL[1:0]），该引脚为低电平

位 19 **COSEL**：校准输出选择 (Calibration output selection)
当 COE=1 时，该位可选择 RTC_CALIB 上输出的信号。
0：校准输出为 512 Hz（采用默认预分频器设置）
1：校准输出为 1 Hz（采用默认预分频器设置）
在 RTCCLK 为 32.768 kHz 且预分频器为其默认值（PREDIV_A=127 且 PREDIV_S=255）的条件下，这些频率有效。请参见*第 46.3.16 节：校准时钟输出*。

位 18 **BKP**：备份 (Backup)
用户可对此位执行写操作以记录是否已对夏令时进行更改。

位 17 **SUB1H**：减少 1 小时（冬季时间更改）(Subtract 1 hour (winter time change))
该位置 1 时，如果当前小时不是 0，则日历时间将减少 1 小时。此位始终读为 0。
当前小时为 0 时，将此位置 1 没有任何作用。
0：无影响
1：将当前时间减少 1 小时。这可用于冬季时间更改（在初始化模式以外）。

位 16 **ADD1H**：增加 1 小时（夏季时间更改）(Add 1 hour (summer time change))
该位置 1 时，日历时间将增加 1 小时。此位始终读为 0。
0：无影响
1：将当前时间增加 1 小时。这可用于夏季时间更改（在初始化模式以外）。

位 15 **TSIE**：时间戳中断使能 (Time-stamp interrupt enable)
0：禁止时间戳中断
1：使能时间戳中断

位 14 **WUTIE**：唤醒定时器中断使能 (Wakeup timer interrupt enable)
0：禁止唤醒定时器中断
1：使能唤醒定时器中断

位 13 **ALRBIE**：闹钟 B 中断使能 (Alarm B interrupt enable)
0：禁止闹钟 B 中断
1：使能闹钟 B 中断

位 12 **ALRAIE**：闹钟 A 中断使能 (Alarm A interrupt enable)
0：禁止闹钟 A 中断
1：使能闹钟 A 中断

位 11 **TSE**：时间戳使能 (timestamp enable)
0：禁止时间戳
1：使能时间戳

位 10 **WUTE**：唤醒定时器使能 (Wakeup timer enable)
0：禁止唤醒定时器
1：使能唤醒定时器

位 9 **ALRBE**：闹钟 B 使能 (Alarm B enable)
0：禁止闹钟 B
1：使能闹钟 B

位 8 **ALRAE**：闹钟 A 使能 (Alarm A enable)
0：禁止闹钟 A
1：使能闹钟 A

位 7 保留，必须保持复位值。

位 6 **FMT**：小时格式 (Hour format)
0：24 小时/天格式
1：AM/PM 小时格式

位 5 **BYPSHAD**：旁路影子寄存器 (Bypass the shadow registers)
0：日历值（从 RTC_SSR、RTC_TR 和 RTC_DR 读取时）取自影子寄存器，该影子寄存器每两个 RTCCLK 周期更新一次。
1：日历值（从 RTC_SSR、RTC_TR 和 RTC_DR 读取时）直接取自日历计数器。
注： 如果 APB 时钟的频率低于 7 倍的 RTCCLK 频率，则必须将 BYPSHAD 置“1”。

位 4 **REFCKON**：RTC_REFIN 参考时钟检测使能（50 Hz 或 60 Hz）(RTC_REFIN reference clock detection enable (50 or 60 Hz))

0：禁止 RTC_REFIN 检测
1：使能 RTC_REFIN 检测

注： PREDIV_S 必须为 0x00FF。

位 3 **TSEDGE**：时间戳事件有效边沿 (Timestamp event active edge)

0：RTC_TS 输入上升沿生成时间戳事件
1：RTC_TS 输入下降沿生成时间戳事件
TSEDGE 发生更改时，必须复位 TSE 以避免将 TSF 意外置 1。

位 2:0 **WUCKSEL[2:0]**：唤醒时钟选择 (Wakeup clock selection)

000：选择 RTC/16 时钟
001：选择 RTC/8 时钟
010：选择 RTC/4 时钟
011：选择 RTC/2 时钟
10x：选择 ck_spre 时钟（通常为 1 Hz）
11x：选择 ck_spre 时钟（通常为 1 Hz）并将 WUT 计数器值增加 2^{16}（见下面的注释）

注： *只能在初始化模式下 (RTC_ISR/INITF = 1) 对该寄存器的位 7、6 和 4 执行写操作。*

WUT = 唤醒单元计数器值。当 WUCKSEL[2:1 = 11] 时，WUT = (0x0000 到 0xFFFF) + 0x10000（增加的值）。

只能在 RTC_CR WUTE 位 = 0 且 RTC_ISR WUTWF 位 = 1 时对该寄存器的位 2 到 0 执行写操作。

建议不要在日历小时递增时更改小时，因为这样做会屏蔽日历小时的增量。

ADD1H 和 SUB1H 的更改在下一秒生效。

此寄存器受写保护。第 1752 页的RTC 寄存器写保护中介绍了写访问的过程。

注意： TSEDGE 发生更改时，必须复位 TSE 以避免对 TSF 的误操作。

46.6.4 RTC 初始化和状态寄存器 (RTC_ISR)

RTC initialization and status register

此寄存器受写保护（RTC_ISR[13:8] 位除外）。*第 1752 页的RTC 寄存器写保护*中介绍了写访问的过程。

偏移地址：0x0C

备份域复位值：0x0000 0007

系统复位：不受影响（INIT、INITF 和 RSF 位除外，它们在复位时被清零）

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ITSF	RECALPF
														rc_w0	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAMP3F	TAMP2F	TAMP1F	TSOVF	TSF	WUTF	ALRBF	ALRAF	INIT	INITF	RSF	INITS	SHPF	WUTWF	ALRBWF	ALRAWF
rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rc_w0	rw	r	rc_w0	r	r	r	r	r

位 31:18 保留，必须保持复位值

位 17 **ITSF**：内部时间戳标志 (Internal time-stamp flag)
发生内部时间戳事件时，由硬件将此标志置 1。
该标志由软件写零清零，且必须通过向两个位写零将此标志与 TSF 位一起清零。

位 16 **RECALPF**：重新校准挂起标志 (Recalibration pending Flag)
当软件对 RTC_CALR 寄存器执行写操作时，RECALPF 状态标志将自动置“1”，指示 RTC_CALR 寄存器已屏蔽。当采用新的校准设置时，该位恢复为“0”。请参见*动态重校准*。

位 15 **TAMP3F**：RTC_TAMP3 检测标志 (RTC_TAMP3 detection flag)
在 RTC_TAMP3 输入上检测到入侵检测事件时，由硬件将此标志置 1。
该标志由软件写零清除

位 14 **TAMP2F**：RTC_TAMP2 检测标志 (RTC_TAMP2 detection flag)
在 RTC_TAMP2 输入上检测到入侵检测事件时，由硬件将此标志置 1。
该标志由软件写零清除

位 13 **TAMP1F**：RTC_TAMP1 检测标志 (RTC_TAMP1 detection flag)
在 RTC_TAMP1 输入上检测到入侵检测事件时，由硬件将此标志置 1。
该标志由软件写零清除

位 12 **TSOVF**：时间戳溢出标志 (Time-stamp overflow flag)
当在 TSF 已置 1 的情况下发生时间戳事件时，由硬件将此标志置 1。
该标志由软件写零清除。建议仅在 TSF 位清零之后再检查并清零 TSOVF 位。否则，如果时间戳事件恰好在清零 TSF 位之前刚刚发生，则溢出事件可能会被漏掉。

位 11 **TSF**：时间戳标志 (Time-stamp flag)
发生时间戳事件时，由硬件将此标志置 1。
该标志由软件写零清除。如果 ITSF 标志已置 1，必须通过向两个位写零将 TSF 与 ITSF 一起清零。

位 10 **WUTF**：唤醒定时器标志 (Wakeup timer flag)

当唤醒自动重载计数器计数到 0 时，由硬件将此标志置 1。

该标志由软件写零清除。

软件必须在 WUTF 再次置 1 的 1.5 个 RTCCLK 周期之前将该标志清零。

位 9 **ALRBF**：闹钟 B 标志 (Alarm B flag)

当时间/日期寄存器（RTC_TR 和 RTC_DR）与闹钟 B 寄存器 (RTC_ALRMBR) 匹配时，由硬件将该标志置 1。

该标志由软件写零清除。

位 8 **ALRAF**：闹钟 A 标志 (Alarm A flag)

当时间/日期寄存器（RTC_TR 和 RTC_DR）与闹钟 A 寄存器 (RTC_ALRMAR) 匹配时，由硬件将该标志置 1。

该标志由软件写零清除。

位 7 **INIT**：初始化模式 (Initialization mode)

0：自由运行模式

1：初始化模式，用于编程时间和日期寄存器（RTC_TR 和 RTC_DR）以及预分频器寄存器 (RTC_PRER)。计数器停止计数，当 INIT 被复位后，计数器从新值开始计数。

位 6 **INITF**：初始化标志 (Initialization flag)

当此位置 1 时，RTC 处于初始化状态，此时可更新事件、日期和预分频器寄存器。

0：不允许更新日历寄存器

1：允许更新日历寄存器

位 5 **RSF**：寄存器同步标志 (Registers synchronization flag)

每次将日历寄存器的值复制到影子寄存器（RTC_SSRx、RTC_TRx 和 RTC_DRx）时，都会由硬件将此位置 1。在初始化模式下、平移操作挂起时 (SHPF=1) 或在旁路影子寄存器模式 (BYPSHAD=1) 下，该位由硬件清零。该位还可由软件清零。

在初始化模式下，该位可由软件或硬件清零。

0：日历影子寄存器尚未同步

1：日历影子寄存器已同步

位 4 **INITS**：初始化状态标志 (Initialization status flag)

当日历年份字段不为 0 时（备份域复位状态），由硬件将该位置 1。

0：日历尚未初始化

1：日历已经初始化

位 3 **SHPF**：平移操作挂起 (Shift operation pending)

0：没有平移操作挂起

1：某个平移操作挂起

只要通过对 RTC_SHIFTR 寄存器执行写操作来启动平移操作，此标志便由硬件置 1。执行完相应的平移操作后，此标志由硬件清零。对 SHPF 位执行写入操作不起作用。

位 2 **WUTWF**：唤醒定时器写标志 (Wakeup timer write flag)

此位在 RTC_CR 中的 WUTE 位清零后，并在 2 个 RTCCLK 周期后由硬件置 1，在 WUTE 位置 1 后并在 2 个 RTCCLK 周期后清零。当 WUTE 位清零且 WUTWF 位置 1 时，可更改唤醒定时器的值。

0：不允许更新唤醒定时器配置

1：允许更新唤醒定时器配置

位 1 **ALRBWF**：闹钟 B 写标志 (Alarm B write flag)

在 RTC_CR 寄存器中的 ALRBE 位置 0 之后，当闹钟 B 的值可更改时，由硬件将该位置 1。该位在初始化模式下由硬件清零。

0：不允许更新闹钟 B

1：允许更新闹钟 B

位 0 **ALRAWF**：闹钟 A 写标志 (Alarm A write flag)
在 RTC_CR 寄存器中的 ALRAE 位置 0 后，当闹钟 A 的值可更改时，由硬件将该位置 1。该位在初始化模式下由硬件清零。
0：不允许更新闹钟 A
1：允许更新闹钟 A

注： *将 ALRAF、ALRBF、WUTF 和 TSF 位编程为 0 之后 2 个 APB 时钟周期，清零生效。*

46.6.5 RTC 预分频器寄存器 (RTC_PRER)

RTC prescaler register

只能在初始化模式下对该寄存器执行写操作。必须通过两次独立的写访问执行初始化。请参见*第 1752 页的日历初始化和配置*。

此寄存器受写保护。*第 1752 页的RTC 寄存器写保护*中介绍了写访问的过程。

偏移地址：0x10

备份域复位值：0x007F 00FF

系统复位：不受影响

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

位 31:23 保留，必须保持复位值

位 22:16 **PREDIV_A[6:0]**：异步预分频系数 (Asynchronous prescaler factor)
下面是异步分频系数的公式：
ck_apre 频率 = RTCCLK 频率/(PREDIV_A+1)

位 15 保留，必须保持复位值。

位 14:0 **PREDIV_S[14:0]**：同步预分频系数 (Synchronous prescaler factor)
下面是同步分频系数的公式：
ck_spre 频率 = ck_apre 频率/(PREDIV_S+1)

46.6.6 RTC 唤醒定时器寄存器 (RTC_WUTR)

RTC wakeup timer register

仅当 RTC_ISR 中的 WUTWF 置 1 时才可对该寄存器执行写操作。

此寄存器受写保护。*第 1752 页的RTC 寄存器写保护*中介绍了写访问的过程。

偏移地址：0x14

备份域复位值：0x0000 FFFF

系统复位：不受影响

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

位 31:16 保留，必须保持复位值

位 15:0 **WUT[15:0]**：唤醒自动重载值位 (Wakeup auto-reload value bits)

当使能唤醒定时器时（WUTE 置 1），每 (WUT[15:0] + 1) 个 ck_wut 周期将 WUTF 标志置 1 一次。ck_wut 周期通过 RTC_CR 寄存器的 WUCKSEL[2:0] 位进行选择。

当 WUCKSEL[2] = 1 时，唤醒定时器变为 17 位，WUCKSEL[1] 等效为 WUT[16]，即要重载到定时器的最高有效位。

WUTF 第一次置 1 发生在 WUTE 置 1 之后 (WUT+1) 个 ck_wut 周期。禁止在 WUCKSEL[2:0]=011(RTCCLK/2) 时将 WUT[15:0] 设置为 0x0000。

46.6.7 RTC 闹钟 A 寄存器 (RTC_ALRMAR)

RTC alarm A register

仅当 RTC_ISR 中的 ALRAWF 置 1 时或在初始化模式下，才可以对该寄存器执行写操作。

此寄存器受写保护。*第 1752 页的RTC 寄存器写保护*中介绍了写访问的过程。

偏移地址：0x1C

备份域复位值：0x0000 0000

系统复位：不受影响

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MSK4	WDSEL	DT[1:0]		DU[3:0]				MSK3	PM	HT[1:0]		HU[3:0]			
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MSK2	MNT[2:0]			MNU[3:0]				MSK1	ST[2:0]			SU[3:0]			
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31 **MSK4**：闹钟 A 日期掩码 (Alarm A date mask)
0：如果日期/日匹配，则闹钟 A 置 1
1：在闹钟 A 比较中，日期/日无关

位 30 **WDSEL**：星期几选择 (Week day selection)
0：DU[3:0] 代表日期的个位
1：DU[3:0] 代表星期几 DT[1:0] 为无关位。

位 29:28 **DT[1:0]**：日期的十位（BCD 格式）(Date tens in BCD format)。

位 27:24 **DU[3:0]**：日期的个位或日（BCD 格式）(Date units or day in BCD format)。

位 23 **MSK3**：闹钟 A 小时掩码 (Alarm A hours mask)
0：如果小时匹配，则闹钟 A 置 1
1：在闹钟 A 比较中，小时无关

位 22 **PM**：AM/PM 符号 (AM/PM notation)
0：AM 或 24 小时制
1：PM

位 21:20 **HT[1:0]**：小时的十位（BCD 格式）(Hour tens in BCD format)。

位 19:16 **HU[3:0]**：小时的个位（BCD 格式）(Hour units in BCD format)。

位 15 **MSK2**：闹钟 A 分钟掩码 (Alarm A minutes mask)
0：如果分钟匹配，则闹钟 A 置 1
1：在闹钟 A 比较中，分钟无关

位 14:12 **MNT[2:0]**：分钟的十位（BCD 格式）(Minute tens in BCD format)。

位 11:8 **MNU[3:0]**：分钟的个位（BCD 格式）(Minute units in BCD format)。

位 7 **MSK1**：闹钟 A 秒掩码 (Alarm A seconds mask)
0：如果秒匹配，则闹钟 A 置 1
1：在闹钟 A 比较中，秒无关

位 6:4 **ST[2:0]**：秒的十位（BCD 格式）(Second tens in BCD format)。

位 3:0 **SU[3:0]**：秒的个位（BCD 格式）(Second units in BCD format)。

46.6.8 RTC 闹钟 B 寄存器 (RTC_ALRMBR)

RTC alarm B register

仅当 RTC_ISR 中的 ALRBWF 置 1 时或在初始化模式下，才可以对该寄存器执行写操作。

此寄存器受写保护。*第 1752 页的RTC 寄存器写保护*中介绍了写访问的过程。

偏移地址：0x20

备份域复位值：0x0000 0000

系统复位：不受影响

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MSK4	WDSEL	DT[1:0]		DU[3:0]				MSK3	PM	HT[1:0]		HU[3:0]			
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MSK2	MNT[2:0]			MNU[3:0]				MSK1	ST[2:0]			SU[3:0]			
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31 **MSK4**：闹钟 B 日期掩码 (Alarm B date mask)
0：如果日期和日匹配，则闹钟 B 置 1
1：在闹钟 B 比较中，日期和日无关

位 30 **WDSEL**：星期几选择 (Week day selection)
0：DU[3:0] 代表日期的个位
1：DU[3:0] 代表星期几 DT[1:0] 为无关位。

位 29:28 **DT[1:0]**：日期的十位（BCD 格式）(Date tens in BCD format)

位 27:24 **DU[3:0]**：日期个位或日（BCD 格式）(Date units or day in BCD format)

位 23 **MSK3**：闹钟 B 小时掩码 (Alarm B hours mask)
0：如果小时匹配，则闹钟 B 置 1
1：在闹钟 B 比较中，小时无关

位 22 **PM**：AM/PM 符号 (AM/PM notation)
0：AM 或 24 小时制
1：PM

位 21:20 **HT[1:0]**：小时的十位（BCD 格式）(Hour tens in BCD format)

位 19:16 **HU[3:0]**：小时的个位（BCD 格式）(Hour units in BCD format)

位 15 **MSK2**：闹钟 B 分钟掩码 (Alarm B minutes mask)
0：如果分钟匹配，则闹钟 B 置 1
1：在闹钟 B 比较中，分钟无关

位 14:12 **MNT[2:0]**：分钟的十位（BCD 格式）(Minute tens in BCD format)

位 11:8 **MNU[3:0]**：分钟的个位（BCD 格式）(Minute units in BCD format)

位 7 **MSK1**：闹钟 B 秒掩码 (Alarm B seconds mask)
0：如果秒匹配，则闹钟 B 置 1
1：在闹钟 B 比较中，秒无关

位 6:4 **ST[2:0]**：秒的十位（BCD 格式）(Second tens in BCD format)

位 3:0 **SU[3:0]**：秒的个位（BCD 格式）(Second units in BCD format)

46.6.9 RTC 写保护寄存器 (RTC_WPR)

RTC write protection register

偏移地址：0x24

复位值：0x0000 0000

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

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

位 31:8 保留，必须保持复位值。

位 7:0 **KEY**：写保护关键字 (Write protection key)
可通过软件对该字节执行写操作。
读取该字节时，始终返回 0x00。
有关如何解锁 RTC 寄存器写保护的介绍，请参见 *RTC 寄存器写保护*。

46.6.10 RTC 亚秒寄存器 (RTC_SSR)

RTC sub second register

偏移地址：0x28

备份域复位值：0x0000 0000

系统复位：当 BYPSHAD = 0 时为 0x0000 0000。当 BYPSHAD = 1 时不受影响。

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

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

位 31:16 保留，必须保持复位值

位 15:0 **SS**：亚秒值 (Sub seconds value)
SS[15:0] 是同步预分频器计数器的值。此亚秒值可根据以下公式得出：
亚秒值 = (PREDIV_S - SS) / (PREDIV_S + 1)
注： 仅当执行平移操作之后，SS 才能大于 PREDIV_S。在这种情况下，正确的时间/日期比 RTC_TR/RTC_DR 所指示的时间/日期慢一秒钟。

46.6.11 RTC 平移控制寄存器 (RTC_SHIFTR)

RTC shift control register

此寄存器受写保护。*第 1752 页的RTC 寄存器写保护*中介绍了写访问的过程。

偏移地址：0x2C

备份域复位值：0x0000 0000

系统复位：不受影响

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ADD1S	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
w															

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

位 31 **ADD1S**：增加一秒钟 (Add one second)

0：无影响

1：对时钟/日历增加一秒钟

此位为只写位且始终读为 0。当平移操作挂起（RTC_ISR 中的 SHPF=1）时，对该位执行写操作无作用。

此函数应与 SUBFS 配合使用（请参见下文介绍），以便有效地向原子操作机制的时钟添加亚秒值。

位 30:15 保留，必须保持复位值

位 14:0 **SUBFS**：减少亚秒值 (Subtract a fraction of a second)

此位为只写位且始终读为 0。当平移操作挂起（RTC_ISR 中的 SHPF=1）时，对该位执行写操作无作用。

写入 SUBFS 的值将加到同步预分频器计数器中。由于该计数器递减计数，此操作可有效地从时钟减去（延迟）以下时间：

延迟（秒）= SUBFS / (PREDIV_S + 1)

当 ADD1S 函数与 SUBFS 结合使用时，可有效地将亚秒值增加到时钟（提前时钟），使时钟提前以下时间：

提前（秒）= (1 - (SUBFS / (PREDIV_S + 1)))。

注：　对 SUBFS 执行写操作将使 RSF 清零。软件随后会等待至 RSF=1 以确定影子寄存器已更新为平移后的时间。

46.6.12 RTC 时间戳时间寄存器 (RTC_TSTR)

RTC timestamp time register

仅当 RTC_ISR 中的 TSF 置 1 时，该寄存器的内容才有效。当 TSF 位复位时，清零该寄存器。

偏移地址：0x30

备份域复位值：0x0000 0000

系统复位：不受影响

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PM	HT[1:0]		HU[3:0]			
									r	r	r	r	r	r	r

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	MNT[2:0]			MNU[3:0]				Res.	ST[2:0]			SU[3:0]			
	r	r	r	r	r	r	r		r	r	r	r	r	r	r

位 31:23 保留，必须保持复位值

位 22 **PM**：AM/PM 符号 (AM/PM notation)
0：AM 或 24 小时制
1：PM

位 21:20 **HT[1:0]**：小时的十位（BCD 格式）(Hour tens in BCD format)。

位 19:16 **HU[3:0]**：小时的个位（BCD 格式）(Hour units in BCD format)。

位 15 保留，必须保持复位值

位 14:12 **MNT[2:0]**：分钟的十位（BCD 格式）(Minute tens in BCD format)。

位 11:8 **MNU[3:0]**：分钟的个位（BCD 格式）(Minute units in BCD format)。

位 7 保留，必须保持复位值

位 6:4 **ST[2:0]**：秒的十位（BCD 格式）(Second tens in BCD format)。

位 3:0 **SU[3:0]**：秒的个位（BCD 格式）(Second units in BCD format)。

46.6.13 RTC 时间戳日期寄存器 (RTC_TSDR)

RTC timestamp date register

仅当 RTC_ISR 中的 TSF 置 1 时，该寄存器的内容才有效。当 TSF 位复位时，清零该寄存器。

偏移地址：0x34

备份域复位值：0x0000 0000

系统复位：不受影响

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WDU[1:0]			MT	MU[3:0]				Res.	Res.	DT[1:0]		DU[3:0]			
r	r	r	r	r	r	r	r			r	r	r	r	r	r

位 31:16 保留，必须保持复位值

位 15:13 **WDU[1:0]**：星期几的个位 (Week day units)

位 12 **MT**：月份的十位（BCD 格式）(Month tens in BCD format)

位 11:8 **MU[3:0]**：月份的个位（BCD 格式）(Month units in BCD format)

位 7:6 保留，必须保持复位值

位 5:4 **DT[1:0]**：日期的十位（BCD 格式）(Date tens in BCD format)

位 3:0 **DU[3:0]**：日期的个位（BCD 格式）(Date units in BCD format)

46.6.14 RTC 时间戳亚秒寄存器 (RTC_TSSSR)

RTC time-stamp sub second register

仅当 RTC_ISR/TSF 置 1 时，该寄存器的内容才有效。当 RTC_ISR/TSF 位复位时，清零该寄存器。

偏移地址：0x38

备份域复位值：0x0000 0000

系统复位：不受影响

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

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

位 31:16 保留，必须保持复位值

位 15:0 **SS**：亚秒值 (Sub seconds value)

当发生时间戳事件时，SS[15:0] 是同步预分频器计数器的值。

46.6.15 RTC 校准寄存器 (RTC_CALR)

RTC calibration register

此寄存器受写保护。*第 1752 页的RTC 寄存器写保护*中介绍了写访问的过程。

偏移地址：0x3C

备份域复位值：0x0000 0000

系统复位：不受影响

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CALP	CALW8	CALW16	Res.	Res.	Res.	Res.	CALM[8:0]								
rw	rw	rw					rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:16 保留，必须保持复位值

位 15 **CALP**：将 RTC 的频率增加 488.5 ppm (Increase frequency of RTC by 488.5 ppm)

0：不增加 RTCCLK 脉冲。

1：每 2^{11} 个脉冲有效插入一个 RTCCLK 脉冲（将频率增加 488.5 ppm）。

此功能应与 CALM 结合使用，后者在高分辨率下会降低日历的频率。如果输入频率为 32768 Hz，则在 32 秒窗口中增加的 RTCCLK 脉冲数按如下公式计算：(512 * CALP) - CALM。

请参见*第 46.3.13 节：RTC 精密数字校准*。

位 14 **CALW8**：使用 8 秒校准周期 (Use an 8-second calibration cycle period)

当 CALW8 置“1”时，选择 8 秒校准周期。

注： CALW8=“1”时，CALM[1:0] 将始终保持为“00”。请参见第 46.3.13 节：RTC 精密数字校准。

位 13 **CALW16**：使用 16 秒校准周期 (Use a 16-second calibration cycle period)

当 CALW16 置“1”时，选择 16 秒校准周期。如果 CALW8 = 1，则不能将该位置“1”。

注： 当 CALW16=“1”时，CALM[0] 将始终保持为“0”。请参见第 46.3.13 节：RTC 精密数字校准。

位 12:9 保留，必须保持复位值

位 8:0 **CALM[8:0]**：负校准 (Calibration minus)

在 2^{20} 个 RTCCLK 脉冲内屏蔽 CALM 个脉冲（如果输入频率为 32768 Hz，则为 32 秒）来降低日历的频率。其分辨率为 0.9537 ppm。

要提高日历的频率，则应将此功能与 CALP 结合使用。请参见*第 1755 页的第 46.3.13 节：RTC 精密数字校准*。

46.6.16 RTC 入侵配置寄存器 (RTC_TAMPCR)

RTC tamper configuration register

偏移地址：0x40

备份域复位值：0x0000 0000

系统复位：不受影响

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	TAMP3MF	TAMP3NOERASE	TAMP3IE	TAMP2MF	TAMP2NOERASE	TAMP2IE	TAMP1MF	TAMP1NOERASE	TAMP1IE
							rw	rw	rw	rw	rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAMPPUDIS	TAMPPRCH[1:0]		TAMPFLT[1:0]		TAMPFREQ[2:0]			TAMPTS	TAMP3TRG	TAMP3E	TAMP2TRG	TAMP2E	TAMPIE	TAMP1TRG	TAMP1E
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:25 保留，必须保持复位值。

位 24 **TAMP3MF**：入侵 3 屏蔽标志 (Tamper 3 mask flag)

0：入侵 3 事件生成触发事件，必须通过软件将 TAMP3F 清零来允许下一次入侵事件检测。

1：入侵 3 事件生成触发事件。屏蔽 TAMP3F 并由硬件在内部将其清零。不擦除备份寄存器和备份 SRAM。

注： TAMP3MF 置 1 时，不得使能入侵 3 中断。

位 23 **TAMP3NOERASE**：入侵 3 不擦除 (Tamper 3 no erase)

0：入侵 3 事件擦除备份寄存器和备份 SRAM。

1：入侵 3 事件不擦除备份寄存器和备份 SRAM。

位 22 **TAMP3IE**：入侵 3 中断使能 (Tamper 3 interrupt enable)

0：TAMPIE = 0 时禁止入侵 3 中断。

1：使能入侵 3 中断。

位 21 **TAMP2MF**：入侵 2 屏蔽标志 (Tamper 2 mask flag)

0：入侵 2 事件生成触发事件，如果要允许下一次入侵事件检测，必须通过软件将 TAMP2F 清零。

1：入侵 2 事件生成触发事件。屏蔽 TAMP2F 并由硬件在内部将其清零。不擦除备份寄存器。

注： TAMP2MF 置 1 时，不得使能入侵 2 中断。

位 20 **TAMP2NOERASE**：入侵 2 不擦除 (Tamper 2 no erase)

0：入侵 2 事件擦除备份寄存器和备份 SRAM。

1：入侵 2 事件不擦除备份寄存器和备份 SRAM。

位 19 **TAMP2IE**：入侵 2 中断使能 (Tamper 2 interrupt enable)

0：TAMPIE = 0 时禁止入侵 2 中断。

1：使能入侵 2 中断。

位 18 **TAMP1MF**：入侵 1 屏蔽标志 (Tamper 1 mask flag)

0：入侵 1 事件生成触发事件，必须通过软件将 TAMP1F 清零来允许下一次入侵事件检测。

1：入侵 1 事件生成触发事件。屏蔽 TAMP1F 并由硬件在内部将其清零。不擦除备份寄存器和备份 SRAM。

注： TAMP1MF 置 1 时，不得使能入侵 1 中断。

位 17 **TAMP1NOERASE**：入侵 1 不擦除 (Tamper 1 no erase)

0：入侵 1 事件擦除备份寄存器和备份 SRAM。

1：入侵 1 事件不擦除备份寄存器和备份 SRAM。

位 16 **TAMP1IE**：入侵 1 中断使能 (Tamper 1 interrupt enable)

0：TAMPIE = 0 时禁止入侵 1 中断。

1：使能入侵 1 中断。

位 15 **TAMPPUDIS**：RTC_TAMPx 上拉禁止 (RTC_TAMPx pull-up disable)

该位决定在每次采样之前是否对每个 RTC_TAMPx 引脚都进行预充电。

0：采样之前对 RTC_TAMPx 引脚进行预充电（使能内部上拉）

1：禁止对 RTC_TAMPx 引脚进行预充电。

位 14:13 **TAMPPRCH[1:0]**：RTC_TAMPx 预充电持续时间 (RTC_TAMPx precharge duration)

这些位决定了在每次采样之前激活上拉的持续时间。TAMPPRCH 对每个 RTC_TAMPx 输入都有效。

0x0：1 个 RTCCLK 周期

0x1：2 个 RTCCLK 周期

0x2：4 个 RTCCLK 周期

0x3：8 个 RTCCLK 周期

位 12:11 **TAMPFLT[1:0]**：RTC_TAMPx 过滤器计数 (RTC_TAMPx filter count)

这些位决定了为激活入侵事件所需的指定电平 (TAMP*TRG) 上的连续采样次数。TAMPFLT 对每个 RTC_TAMPx 输入都有效。

0x0：在 RTC_TAMPx 输入转变为有效电平的边沿激活入侵事件（RTC_TAMPx 输入上无内部上拉）。

0x1：在有效电平上连续执行 2 次采样后激活入侵事件。

0x2：在有效电平上连续执行 4 次采样后激活入侵事件。

0x3：在有效电平上连续执行 8 次采样后激活入侵事件。

位 10:8 **TAMPFREQ[2:0]**：入侵采样频率 (Tamper sampling frequency)

决定对每个 RTC_TAMPx 输入进行采样时的频率。

0x0：RTCCLK/32768（RTCCLK = 32768 Hz 时为 1 Hz）

0x1：RTCCLK/16384（RTCCLK = 32768 Hz 时为 2 Hz）

0x2：RTCCLK/8192（RTCCLK = 32768 Hz 时为 4 Hz）

0x3：RTCCLK/4096（RTCCLK = 32768 Hz 时为 8 Hz）

0x4：RTCCLK/2048（RTCCLK = 32768 Hz 时为 16 Hz）

0x5：RTCCLK/1024（RTCCLK = 32768 Hz 时为 32 Hz）

0x6：RTCCLK/512（RTCCLK = 32768 Hz 时为 64 Hz）

0x7：RTCCLK/256（RTCCLK = 32768 Hz 时为 128 Hz）

位 7 **TAMPTS**：发生入侵检测事件时激活时间戳 (Activate timestamp on tamper detection event)

0：发生入侵检测事件时不保存时间戳

1：发生入侵检测事件时保存时间戳

即便 RTC_CR 寄存器中的 TSE=0，TAMPTS 仍有效。

位 6 **TAMP3TRG**：RTC_TAMP3 输入的有效电平 (Active level for RTC_TAMP3 input)

如果 TAMPFLT ≠ 00：

0：RTC_TAMP3 输入保持低电平会触发入侵检测事件。

1：RTC_TAMP3 输入保持高电平会触发入侵检测事件。

如果 TAMPFLT = 00：

0：RTC_TAMP3 输入上升沿会触发入侵检测事件。

1：RTC_TAMP3 输入下降沿会触发入侵检测事件。

位 5 **TAMP3E**：RTC_TAMP3 检测使能 (RTC_TAMP3 detection enable)
0：禁止 RTC_TAMP3 输入检测
1：使能 RTC_TAMP3 输入检测

位 4 **TAMP2TRG**：RTC_TAMP2 输入的有效电平 (Active level for RTC_TAMP2 input)
如果 TAMPFLT != 00：
0：RTC_TAMP2 输入保持低电平会触发入侵检测事件。
1：RTC_TAMP2 输入保持高电平会触发入侵检测事件。
如果 TAMPFLT = 00：
0：RTC_TAMP2 输入上升沿会触发入侵检测事件。
1：RTC_TAMP2 输入下降沿会触发入侵检测事件。

位 3 **TAMP2E**：RTC_TAMP2 输入检测使能 (RTC_TAMP2 input detection enable)
0：禁止 RTC_TAMP2 检测
1：使能 RTC_TAMP2 检测

位 2 **TAMPIE**：入侵中断使能 (Tamper interrupt enable)
0：禁止入侵中断
1：使能入侵中断
注： 该位使能所有入侵引脚事件的中断，无论 TAMPxIE 电平如何。如果将该位清零，可以通过将 TAMPxIE 置 1 分别使能每个入侵事件中断。

位 1 **TAMP1TRG**：RTC_TAMP1 输入的有效电平 (Active level for RTC_TAMP1 input)
如果 TAMPFLT != 00
0：RTC_TAMP1 输入保持低电平会触发入侵检测事件。
1：RTC_TAMP1 输入保持高电平会触发入侵检测事件。
如果 TAMPFLT = 00：
0：RTC_TAMP1 输入上升沿会触发入侵检测事件。
1：RTC_TAMP1 输入下降沿会触发入侵检测事件。

位 0 **TAMP1E**：RTC_TAMP1 输入检测使能 (RTC_TAMP1 input detection enable)
0：禁止 RTC_TAMP1 检测
1：使能 RTC_TAMP1 检测

注意： 如果 TAMPFLT = 0，则更改 TAMPxTRG 时必须复位 TAMPxE，以避免将 TAMPxF 意外置 1。

46.6.17 RTC 闹钟 A 亚秒寄存器 (RTC_ALRMASSR)

RTC alarm A sub second register

仅当 RTC_CR 中的 ALRAE 复位时或在初始化模式下，才可以对该寄存器执行写操作。

此寄存器受写保护。*第 1752 页的RTC 寄存器写保护*中介绍了写访问的过程。

偏移地址：0x44

备份域复位值：0x0000 0000

系统复位：不受影响

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

位 31:28 保留，必须保持复位值。

位 27:24 **MASKSS[3:0]**：屏蔽从此位开始的最高有效位 (Mask the most-significant bits starting at this bit)

0：不对闹钟 A 的亚秒进行比较。当秒单元递增时设置闹钟（假定其余字段均匹配）。
1：在闹钟 A 比较中，SS[14:1] 为无关位。仅比较 SS[0]。
2：在闹钟 A 比较中，SS[14:2] 为无关位。仅比较 SS[1:0]。
3：在闹钟 A 比较中，SS[14:3] 为无关位。仅比较 SS[2:0]。
...
12：在闹钟 A 比较中，SS[14:12] 为无关位。比较 SS[11:0]。
13：在闹钟 A 比较中，SS[14:13] 为无关位。比较 SS[12:0]。
14：在闹钟 A 比较中，SS[14] 为无关位。比较 SS[13:0]。
15：所有 15 个 SS 位均进行比较，并且必须全部匹配才能激活闹钟。
同步计数器的溢出位（位 15）从不进行比较。仅当执行平移操作之后，此位才不为 0。

位 23:15 保留，必须保持复位值。

位 14:0 **SS[14:0]**：亚秒值 (Sub seconds value)
该值与同步预分频器计数器的内容进行比较以确定是否要激活闹钟 A。仅比较位 0 到 MASKSS-1。

46.6.18 RTC 闹钟 B 亚秒寄存器 (RTC_ALRMBSSR)

RTC alarm B sub second register

仅当 RTC_CR 中的 ALRBE 复位时或在初始化模式下，才可以对该寄存器执行写操作。

该寄存器受写保护。*RTC 寄存器写保护*一节中介绍了写访问的过程。

偏移地址：0x48

备份域复位值：0x0000 0000

系统复位：不受影响

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

位 31:28 保留，必须保持复位值。

位 27:24 **MASKSS[3:0]**：屏蔽从此位开始的最高有效位 (Mask the most-significant bits starting at this bit)

0x0：不对闹钟 B 的亚秒进行比较。当秒单元递增时设置闹钟（假定其余字段均匹配）。
0x1：在闹钟 B 比较中，SS[14:1] 为无关位。仅比较 SS[0]。
0x2：在闹钟 B 比较中，SS[14:2] 为无关位。仅比较 SS[1:0]。
0x3：在闹钟 B 比较中，SS[14:3] 为无关位。仅比较 SS[2:0]。
...
0xC：在闹钟 B 比较中，SS[14:12] 为无关位。比较 SS[11:0]。
0xD：在闹钟 B 比较中，SS[14:13] 为无关位。比较 SS[12:0]。
0xE：在闹钟 B 比较中，SS[14] 为无关位。比较 SS[13:0]。
0xF：所有 15 个 SS 位均进行比较，并且必须全部匹配才能激活闹钟。
同步计数器的溢出位（位 15）从不进行比较。仅当执行平移操作之后，此位才不为 0。

位 23:15 保留，必须保持复位值。

位 14:0 **SS[14:0]**：亚秒值 (Sub seconds value)

该值与同步预分频器计数器的内容进行比较以确定是否要激活闹钟 B。仅比较位 0 到 MASKSS-1。

46.6.19 RTC 选项寄存器 (RTC_OR)

RTC option register

偏移地址：0x4C

备份域复位值：0x0000 0000

系统复位：不受影响

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

位 31:2 保留，必须保持复位值。

位 1 **RTC_OUT_RMP**：RTC_OUT 重映射 (RTC_OUT remap)

将此位置 1 可将 RTC 输出重映射到 PB2，具体如下：

RTC_OUT_RMP = “0”：

如果 OSEL/= “00”：RTC_ALARM 输出到 PC13

如果 OSEL= “00”且 COE = “1”：RTC_CALIB 输出到 PC13

RTC_OUT_RMP = “1”：

如果 OSEL /= “00”且 COE = “0”：RTC_ALARM 输出到 PB2

如果 OSEL = “00”且 COE = “1”：RTC_CALIB 输出到 PB2

如果 OSEL /= “00”且 COE = “1”：RTC_CALIB 输出到 PB2 且 RTC_ALARM 输出到 PC13。

位 0 **RTC_ALARM_TYPE**：PC13 上的 RTC_ALARM 输出类型 (RTC_ALARM output type on PC13)

此位由软件置 1 和清零

0：RTC_ALARM 在映射到 PC13 上时为开漏输出

1：RTC_ALARM 在映射到 PC13 上时为推挽输出

46.6.20 RTC 备份寄存器 (RTC_BKPxR)

RTC backup registers

偏移地址：0x50 到 0xCC

备份域复位值：0x0000 0000

系统复位：不受影响

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

位 31:0 **BKP[31:0]**

应用可向/从这些寄存器写入/读取数据。

当 VDD 关闭时，这些寄存器由 V_{BAT} 供电，因而系统复位时，这些寄存器不会复位，并且当器件在低功耗模式下工作时，寄存器的内容仍然有效。

发生入侵检测事件时该寄存器会被复位，并且只要 TAMPxF=1，该寄存器就一直保持复位。

46.6.21 RTC 寄存器映射

表 358. RTC 寄存器映射和复位值

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	**RTC_TR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PM	HT [1:0]		HU[3:0]				Res.	MNT[2:0]			MNU[3:0]				Res.	ST[2:0]			SU[3:0]			
	Reset value										0	0	0	0	0	0	0		0	0	0	0	0	0	0		0	0	0	0	0	0	0
0x04	**RTC_DR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	YT[3:0]				YU[3:0]				WDU[2:0]			MT	MU[3:0]				Res.	Res.	DT [1:0]		DU[3:0]			
	Reset value									0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1			0	0	0	0	0	1
0x08	**RTC_CR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ITSE	COE	OSEL [1:0]		POL	COSEL	BKP	SUB1H	ADD1H	TSIE	WUTIE	ALRBE	ALRAIE	TSE	WUTE	ALRBE	ALRAE	Res.	FMT	BYPSHAD	REFCKON	TSEDGE	WUCKSEL[2:0]		
	Reset value								0	0	0	0	0		0	0	0	0	0	0	0	0	0	0	0		0	0	0	0	0	0	0
0x0C	**RTC_ISR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ITSF	RECALPF	TAMP3F	TAMP2F	TAMP1F	TSOVF	TSF	WUTF	ALRBF	ALRAF	INIT	INITF	RSF	INITS	SHPF	WUTWF	ALRBWF	ALRAWF
	Reset value															0	0		0	0	0	0	0	0	0	0	0	0	0	0	1	1	1
0x10	**RTC_PRER**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREDIV_A[6:0]							PREDIV_S[14:0]															
	Reset value										1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
0x14	**RTC_WUTR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WUT[15:0]															
	Reset value																	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0x1C	**RTC_ALRMAR**	MSK4	WDSEL	DT [1:0]		DU[3:0]				MSK3	PM	HT [1:0]		HU[3:0]				MSK2	MNT[2:0]			MNU[3:0]				MSK1	ST[2:0]			SU[3:0]			
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x20	**RTC_ALRMBR**	MSK4	WDSEL	DT [1:0]		DU[3:0]				MSK3	PM	HT [1:0]		HU[3:0]				MSK2	MNT[2:0]			MNU[3:0]				MSK2	ST[2:0]			SU[3:0]			
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x24	**RTC_WPR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	KEY							
	Reset value																									0	0	0	0	0	0	0	0
0x28	**RTC_SSR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SS[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x2C	**RTC_SHIFTR**	ADD1S	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SUBFS[14:0]														
	Reset value	0																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

表 358. RTC 寄存器映射和复位值（续）

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x30	**RTC_TSTR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PM	HT[1:0]		HU[3:0]				Res.	MNT[2:0]			MNU[3:0]				Res.	ST[2:0]			SU[3:0]			
	Reset value										0	0	0	0	0	0	0		0	0	0	0	0	0	0		0	0	0	0	0	0	0
0x34	**RTC_TSDR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WDU[1:0]			MT	MU[3:0]				Res.	Res.	DT [1:0]		DU[3:0]			
	Reset value																	0	0	0	0	0	0	0	0			0	0	0	0	0	0
0x38	**RTC_TSSSR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SS[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x3C	**RTC_ CALR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CALP	CALW8	CALW16	Res.	Res.	Res.	Res.	CALM[8:0]								
	Reset value																	0	0	0					0	0	0	0	0	0	0	0	0
0x40	**RTC_TAMPCR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TAMP3MF	TAMP3NOERASE	TAMP3IE	TAMP2MF	TAMP2NOERASE	TAMP2IE	TAMP1MF	TAMP1NOERASE	TAMP1IE	TAMPPUDIS	TAMPPRCH[1:0]		TAMPFLT[1:0]		TAMPFREQ[2:0]		TAMPTS	TAMP3TRG	TAMP3E	TAMP2TRG	TAMP2E	TAMPIE	TAMP1TRG	TAMP1E
	Reset value								0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x44	**RTC_ ALRMASSR**	Res.	Res.	Res.	Res.	MASKSS [3:0]				Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SS[14:0]														
	Reset value					0	0	0	0										0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x48	**RTC_ ALRMBSSR**	Res.	Res.	Res.	Res.	MASKSS [3:0]				Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SS[14:0]														
	Reset value					0	0	0	0										0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x4C	**RTC_ OR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RTC_OUT_RMP	RTC_ALARM_TYPE
	Reset value																															0	0
0x50 to 0xCC	**RTC_BKP0R**	BKP[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	to **RTC_BKP31R**	BKP[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

有关寄存器边界地址的信息，请参见*第 100 页的第 2.2.2 节*。

47 内部集成电路 (I2C) 接口

47.1 前言

I^2C（内部集成电路）总线接口处理微控制器与串行 I^2C 总线间的通信。它提供多主模式功能，可以控制所有 I^2C 总线特定的序列、协议、仲裁和时序。它支持标准模式 (Sm)、快速模式 (Fm) 和超快速模式 (Fm+)。

它还与 SMBus（系统管理总线）和 PMBus（电源管理总线）兼容。

可使用 DMA 来减轻 CPU 的工作量。

47.2 I2C 主要特性

- 兼容 I^2C 总线规范第 03 版：
  - 从模式和主模式
  - 多主模式功能
  - 标准速度模式（高达 100 kHz）
  - 快速模式（高达 400 kHz）
  - 超快速模式（高达 1 MHz）
  - 7 位和 10 位寻址模式
  - 多个 7 位从地址（2 个从设备地址寄存器，1 个具有可配置的掩码位段）
  - 所有 7 位地址应答模式
  - 广播呼叫
  - 总线上的数据建立和保持时间可软件配置
  - 方便易用的事件管理
  - 可选的时钟延长
  - 软件复位
- 带 DMA 功能的 1 字节缓冲
- 可编程模拟和数字噪声滤波器

还可额外提供以下特性，具体取决于产品实现（请参见*第 47.3 节：I2C 特性实现*）：

- 兼容 SMBus 规范第 2.0 版：
  - 具有 ACK 控制的硬件 PEC（数据包错误校验）生成和验证
  - 命令和数据应答控制
  - 支持地址解析协议 (ARP)
  - 支持主机和从设备
  - SMBus 报警
  - 超时和空闲条件检测
- 兼容 PMBus 第 1.1 版标准
- 独立时钟：选择独立时钟源可使 I2C 通信速度不受 i2c_pclk 重新编程的影响
- 地址匹配时从停止模式唤醒

47.3 I2C 特性实现

本手册介绍了 I2C 外设中实现的所有特性。对于 STM32H7xxx 器件，I2C1、I2C2、I2C3 和 I2C4 均实现了下表中列出的全部特性。

表 359. STM32H7x3 I2C 特性实现

I2C特性(1)	I2C1	I2C2	I2C3	I2C4
7 位寻址模式	X	X	X	X
10 位寻址模式	X	X	X	X
标准速度模式（高达 100 kbps）	X	X	X	X
快速模式（高达 400 kbps）	X	X	X	X
超快速模式 20mA 输出驱动I/O（高达1 Mbit/s）	X	X	X	X
独立时钟	X	X	X	X
SMBus	X	X	X	X
从停止模式唤醒	X	X	X	X

1. X = 支持。

47.4 I2C 功能说明

除了接收和发送数据之外，此接口还可以从串行格式转换为并行格式，反之亦然。中断由软件使能或禁止。该接口通过数据引脚 (SDA) 和时钟引脚 (SCL) 连接到 I^2C 总线。它可以连接到标准速度（高达 100 kHz）、快速（高达 400 kHz）或超快速（高达 1 MHz）I^2C 总线。

该接口也可通过数据引脚 (SDA) 和时钟引脚 (SCL) 连接到 SMBus。

如果支持 SMBus 功能：还可使用额外的可选 SMBus 报警引脚 (SMBA)。

47.4.1 I2C 框图

I2C 接口如*图 531* 所示。

图 531. I2C 框图

I2C 的时钟由独立时钟源提供，这使得 I2C 能够独立于 i2c_pclk 频率工作。

更多详细信息，请参见*图44：I2C 的内核时钟分配*。

I2C I/O 支持 20 mA 输出电流驱动器，用于驱动超快速操作。将 SCL 和 SDA 的驱动能力控制位置 1 可使能该功能，这些位位于*第 12 节：系统配置控制器 (SYSCFG)*。

47.4.2 I2C 时钟要求

I2C 内核的时钟由 i2c_ker_ck 提供。

i2c_ker_ck 周期 t_{I2CCLK} 必须遵循以下条件：

$t_{I2CCLK} < (t_{LOW} - t_{filters}) / 4$ 且 $t_{I2CCLK} < t_{HIGH}$

其中：

t_{LOW}：SCL 低电平时间；t_{HIGH}：SCL 高电平时间。

$t_{filters}$：滤波器使能时，该值为模拟滤波器和数字滤波器引入的延时总和。

模拟滤波器延时最大值为 260 ns。数字滤波器延时为 DNF x t_{I2CCLK}。

i2c_pclk 时钟周期 t_{PCLK} 必须遵循以下条件：

$t_{PCLK} < 4/3\ t_{SCL}$

其中，t_{SCL}：SCL 周期

注意： I2C 内核的时钟由 i2c_pclk 提供。该时钟必须遵循 t_{I2CCLK} 的条件。

47.4.3 模式选择

该接口在工作时可选用以下四种模式之一：

- 从发送器
- 从接收器
- 主发送器
- 主接收器

默认情况下，它以从模式工作。接口在生成起始位后会自动由从模式切换为主模式，并在出现仲裁丢失或生成停止位时从主模式切换为从模式，从而实现多主模式功能。

通信流程

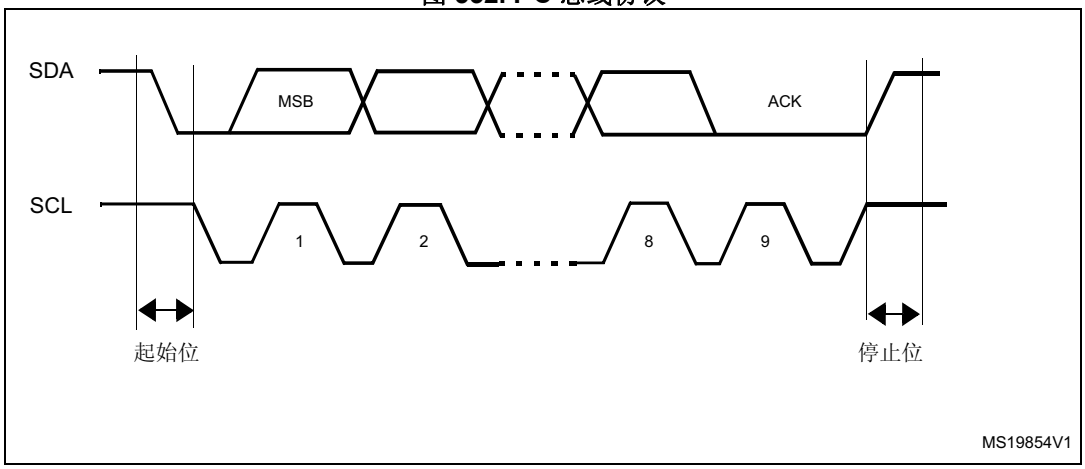
在主模式下，I2C 接口会启动数据传输并生成时钟信号。串行数据传输始终是在出现起始位时开始，在出现停止位时结束。起始位和停止位均在主模式下由软件生成。

在从模式下，该接口能够识别其自身地址（7 或 10 位）以及广播呼叫地址。广播呼叫地址检测可由软件使能或禁止。保留的 SMBus 地址也可由软件使能。

数据和地址均以 8 位字节传输，MSB 在前。起始位后紧随地址字节（7 位地址占据一个字节；10 位地址占据两个字节）。地址始终在主模式下传送。

在字节传输 8 个时钟周期后是第 9 个时钟脉冲，在此期间接收器必须向发送器发送一个应答位。请参见下图。

图 532. I²C 总线协议



应答位可由软件使能或禁止。I2C 接口地址可通过软件进行选择。

47.4.4 I2C 初始化

使能和禁止外设

I2C 外设时钟必须在时钟控制器中进行配置和使能（请参见*第 8 节：复位和时钟控制 (RCC)*）。

然后可通过将 I2C_CR1 寄存器中的 PE 位置 1 使能 I2C。

当禁止 I2C (PE=0) 时，I²C 将执行软件复位。更多详细信息，请参见*第 47.4.5 节：软件复位*。

噪声滤波器

通过将 I2C_CR1 寄存器中的 PE 位置 1 来使能 I2C 外设之前，如有必要，用户必须配置噪声滤波器。默认情况下，SDA 和 SCL 输入上集成了模拟噪声滤波器。该模拟滤波器符合 I²C 规范，此规范要求在快速模式和超快速模式下对脉宽在 50ns 以下的脉冲都要抑制。用户可通过将 ANFOFF 位置 1 来禁止该模拟滤波器，和/或通过配置 I2C_CR1 寄存器中的 DNF[3:0] 位来选择数字滤波器。

使能数字滤波器时，SCL 或 SDA 线的电平只有在电平稳定时间超过 DNF x I2CCLK 个周期后才会发生内部变化，从而可抑制的尖峰脉宽在 1 到 15 个 i2c_ker_ck 周期可编程。

表 360. 模拟滤波器与数字滤波器对比

	模拟滤波器	数字滤波器
抑制的脉冲宽度	≥ 50 ns	从 1 到 15 个 I2C 外设时钟的可编程长度
优点	停止模式中仍可用	– 长度可编程：额外的滤波能力与标准要求 – 稳定长度
缺点	随温度、电压和过程变化会发生变化	当使能数字滤波器后，无法在地址匹配时从停止模式唤醒

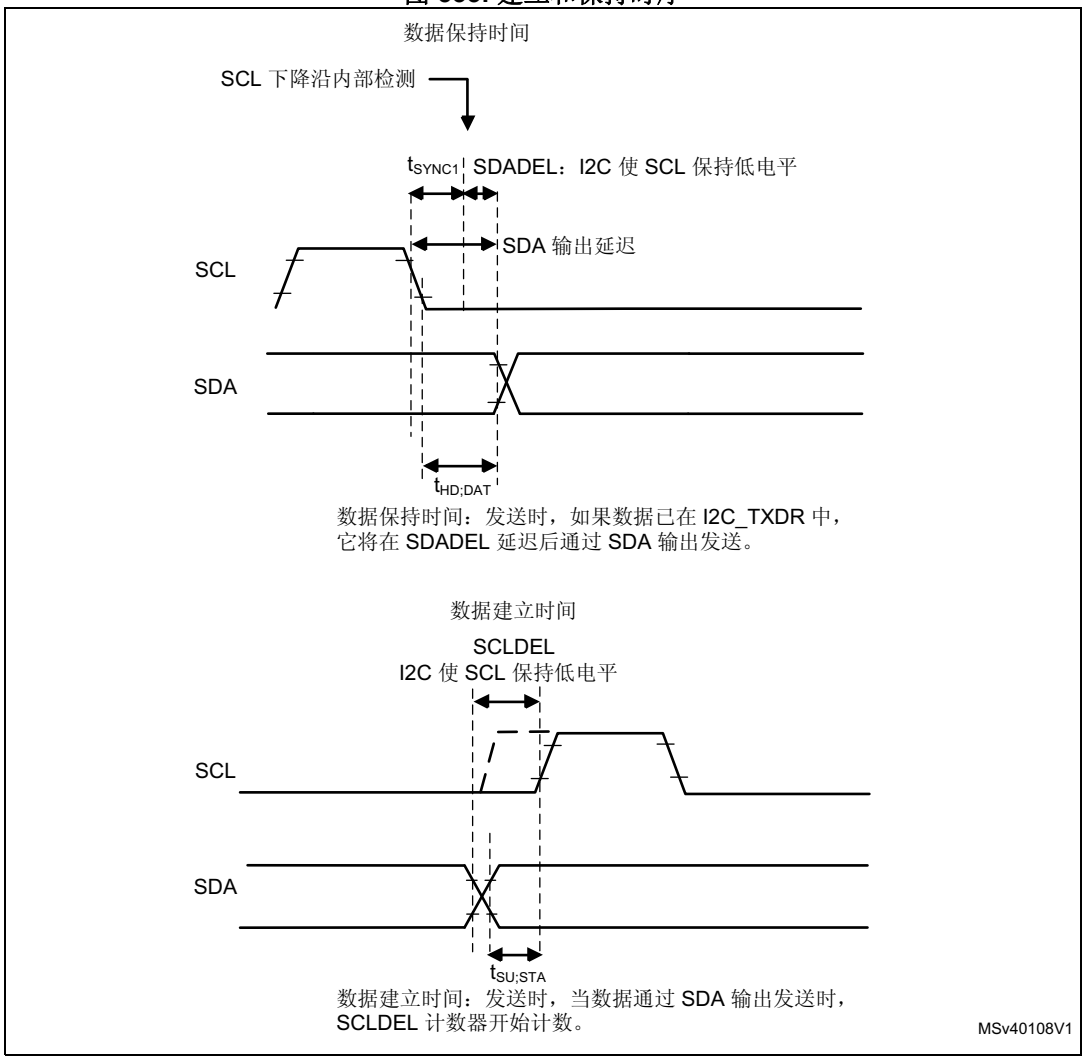
注意： 使能 I2C 时，不允许更改滤波器配置。

I2C 时序

必须配置时序，以便保证主模式和从模式下使用正确的数据保持和建立时间。配置方法是编程 I2C_TIMINGR 寄存器中的 PRESC[3:0]、SCLDEL[3:0] 和 SDADEL[3:0] 位。

STM32CubeMX 工具进行计算并将 I2C_TIMINGR 内容提供在 I2C 配置窗口中。

图 533. 建立和保持时序



- 当内部检测到 SCL 下降沿时，会在发送 SDA 输出之前插入一段延时。该延时为 t_{SDADEL} = SDADEL x t_{PRESC} + t_{I2CCLK}，其中 t_{PRESC} = (PRESC+1) x t_{I2CCLK}。
  T_{SDADEL} 会影响保持时间 $t_{HD;DAT}$。

SDA 总输出延时为：

t_{SYNC1} + {[SDADEL x (PRESC+1) + 1] x t_{I2CCLK} }

t_{SYNC1} 持续时间取决于以下参数：

- SCL 下降斜率
- 模拟滤波器（使能时）引入的输入延时：$t_{AF(min)}$ < t_{AF} < $t_{AF(max)}$ ns
- 数字滤波器（使能时）引入的输入延时：t_{DNF} = DNF x t_{I2CCLK}
- SCL 与 i2c_ker_ck 时钟建立同步而产生的延时（2 到 3 个 i2c_ker_ck 周期）

为了桥接 SCL 下降沿的未定义区域，用户编程 SDADEL 时必须遵循以下条件：

{$t_{f\ (max)}$ +$t_{HD;DAT\ (min)}$ -$t_{AF(min)}$ - [(*DNF* +3) x t_{I2CCLK}]} / {(PRESC +1) x t_{I2CCLK}} ≤ SDADEL

SDADEL ≤ {$t_{HD;DAT\ (max)}$ -$t_{AF(max)}$ - [(*DNF+4) x* t_{I2CCLK}]} / {(PRESC +1) x t_{I2CCLK}}

注： *只有使能模拟滤波器时，公式中才包含 $t_{AF(min)}$ / $t_{AF(max)}$。有关 t_{AF} 值的信息，请参见器件数据手册。*

标准模式、快速模式和超快速模式下的 $t_{HD;DAT}$ 最大值分别可达 3.45 µs、0.9 µs 和 0.45 µs，但必须小于 $t_{VD;DAT}$ 最大值（差值为跳变时间）。只有器件未延长 SCL 信号的低电平周期 (t_{LOW}) 时，才必须满足该最大值条件。如果时钟延长 SCL，数据必须在建立时间内保持有效，之后才能释放时钟。

SDA 上升沿通常为最坏情况，因此在这种情况下，上述公式变成如下形式：

SDADEL ≤ {$t_{VD;DAT\ (max)}$ -$t_{r\ (max)}$ -*260 ns* - [(*DNF+4) x* t_{I2CCLK}]} / {(PRESC +1) x t_{I2CCLK}}。

注： *NOSTRETCH=0 时会违反该条件，这是因为器件会根据 SCLDEL 值来延长 SCL 低电平时间，以保证建立时间。*

有关 t_f、t_r、$t_{HD;DAT}$ 和 $t_{VD;DAT}$ 标准值的信息，请参见*表 361：I2C-SMBUS 规范数据建立和保持时间*。

- 在 t_{SDADEL} 延时后，或在因数据未写入 I2C_TXDR 寄存器而导致从器件必须延长时钟的情况下发送 SDA 输出后，SCL 线会在建立时间内保持低电平。该建立时间为 t_{SCLDEL} = (SCLDEL+1) x t_{PRESC}，其中 t_{PRESC} = (PRESC+1) x t_{I2CCLK}。
  t_{SCLDEL} 会影响建立时间 $t_{SU;DAT}$。

为了桥接 SDA 跳变（上升沿通常为最坏情况）的未定义区域，编程 SCLDEL 时必须遵循以下条件：

{[$t_{r\ (max)}$ + $t_{SU;DAT\ (min)}$] / [(PRESC+1)] x t_{I2CCLK}]} - 1 <= SCLDEL

有关 t_r 和 $t_{SU;DAT}$ 标准值的信息，请参见*表 361：I2C-SMBUS 规范数据建立和保持时间*。

将使用的 SDA 和 SCL 跳变时间值就是应用中的值。使用最大值而非标准值会增加 SDADEL 和 SCLDEL 计算的约束条件，但能够确保任意应用的特性。

注： *在发送和接收模式下，对于每个时钟脉冲，检测到 SCL 下降沿后，I2C 主器件或从器件会至少在 [(SDADEL+SCLDEL+1) x (PRESC+1) + 1] x t_{I2CCLK} 期间内延长 SCL 低电平时间。在发送模式下，如果 SDADEL 计数器计数结束后数据还未写入 I2C_TXDR，则 I2C 会继续延长 SCL 低电平时间，直到写入下一个数据。随后，会将新数据 MSB 发送到 SDA 输出，SCLDEL 计数器将开始计数，同时会继续延长 SCL 低电平时间以确保提供充足的数据建立时间。*

如果从模式下 NOSTRETCH = 1，则 SCL 不会延长。因此，编程 SDADEL 时还必须确保提供充足的建立时间。

表 361. I²C-SMBUS 规范数据建立和保持时间

符号	参数	标准模式 (Sm)		快速模式 (Fm)		超快速模式 (Fm+)		SMBus		单位
		最小值	最大值	最小值	最大值	最小值	最大值	最小值	最大值	
$t_{HD;DAT}$	数据保持时间	0	-	0	-	0	-	0.3	-	µs
$t_{VD;DAT}$	数据有效时间	-	3.45	-	0.9	-	0.45	-	-	
$t_{SU;DAT}$	数据建立时间	250	-	100	-	50	-	250	-	ns
t_r	SDA 和 SCL 信号的上升时间	-	1000	-	300	-	120	-	1000	
t_f	SDA 和 SCL 信号的下降时间	-	300	-	300	-	120	-	300	

此外，在主模式下，必须通过编程 I2C_TIMINGR 寄存器中的 PRESC[3:0]、SCLH[7:0] 和 SCLL[7:0] 位来配置 SCL 时钟的高电平和低电平。

- 当内部检测到 SCL 下降沿时，会在释放 SCL 输出之前插入一段延时。该延时为 $t_{SCLL} = (SCLL+1) \times t_{PRESC}$，其中 $t_{PRESC} = (PRESC+1) \times t_{I2CCLK}$。
  t_{SCLL} 会影响 SCL 低电平时间 t_{LOW}。
- 当内部检测到 SCL 上升沿时，会在将 SCL 输出强制为低电平之前插入一段延时。该延时为 $t_{SCLH} = (SCLH+1) \times t_{PRESC}$，其中 $t_{PRESC} = (PRESC+1) \times t_{I2CCLK}$。$t_{SCLH}$ 会影响 SCL 高电平时间 t_{HIGH}。

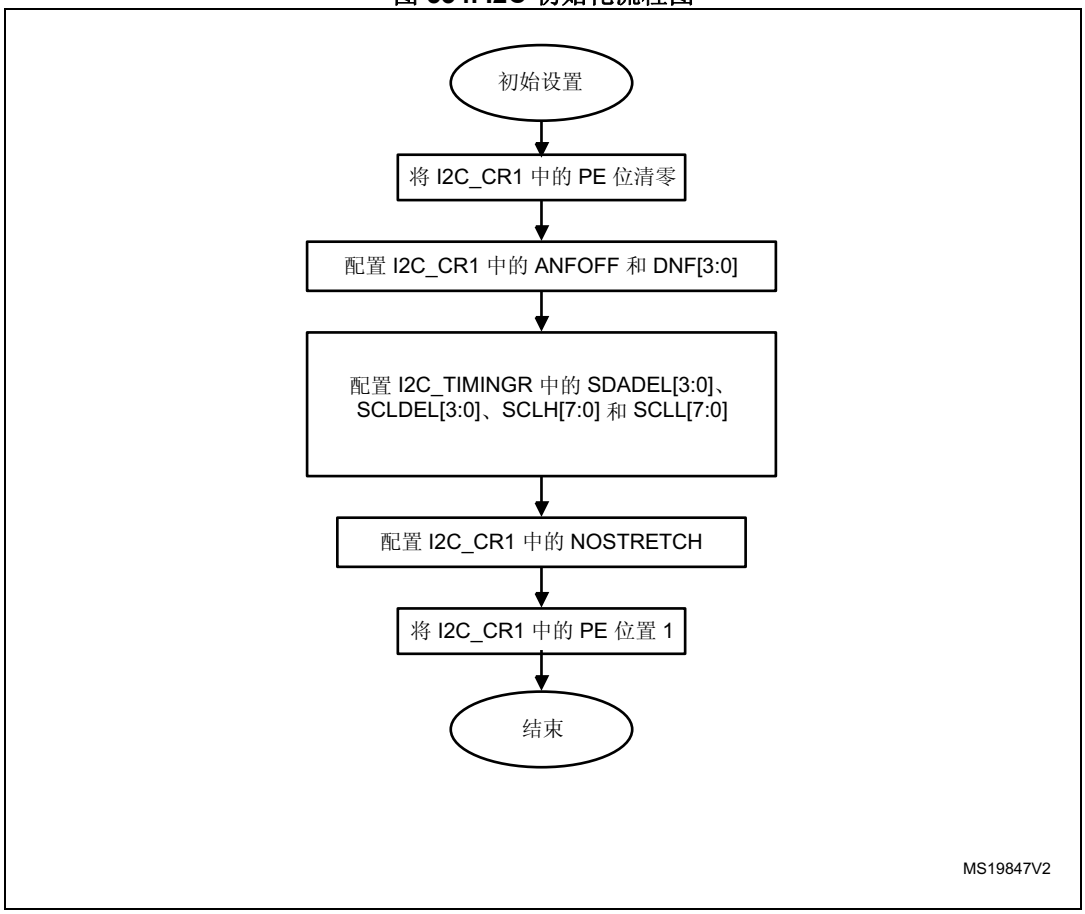
更多详细信息，请参见 *I2C 主模式初始化*。

注意： 使能 I2C 后，不允许更改时序配置。

此外，还必须在使能外设之前配置 I2C 从 NOSTRETCH 模式。更多详细信息，请参见 *I2C 从模式初始化*。

注意： 使能 I2C 后，不允许更改 NOSTRETCH 配置。

图 534. I2C 初始化流程图



47.4.5 软件复位

可通过将 I2C_CR1 寄存器中的 PE 位清零来执行软件复位。在这种情况下，I2C 线 SCL 和 SDA 被释放。内部状态机复位，通信控制位和状态位恢复为其复位值。配置寄存器不受影响。

下面列出了受影响的寄存器位：

1. I2C_CR2 寄存器：START、STOP 和 NACK
2. I2C_ISR 寄存器：BUSY、TXE、TXIS、RXNE、ADDR、NACKF、TCR、TC、STOPF、BERR、ARLO 和 OVR

支持 SMBus 功能时还会影响到以下寄存器位：

1. I2C_CR2 寄存器：PECBYTE
2. I2C_ISR 寄存器：PECERR、TIMEOUT 和 ALERT

必须使 PE 保持低电平持续至少 3 个 APB 时钟周期，才能成功执行软件复位。使用以下软件写序列可确保这一点：- 写入 PE=0 - 检查 PE=0 - 写入 PE=1

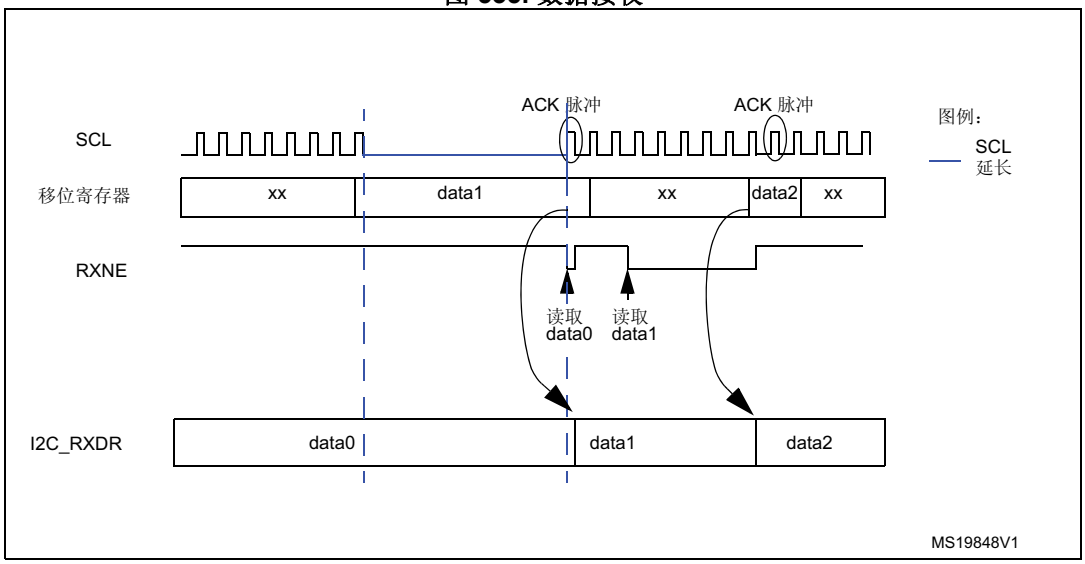
47.4.6 数据传输

数据传输由发送和接收数据寄存器以及移位寄存器来管理。

接收

SDA 输入填充移位寄存器。在第 8 个 SCL 脉冲后（接收到完整的数据字节时），如果 I2C_RXDR 寄存器为空 (RXNE=0)，则移位寄存器的内容会复制到其中。如果 RXNE=1（意味着尚未读取上一次接收到的数据字节），则将延长 SCL 线的低电平时间，直到读取了 I2C_RXDR 为止。在第 8 个和第 9 个 SCL 脉冲之间（应答脉冲之前）插入一段延长的时间。

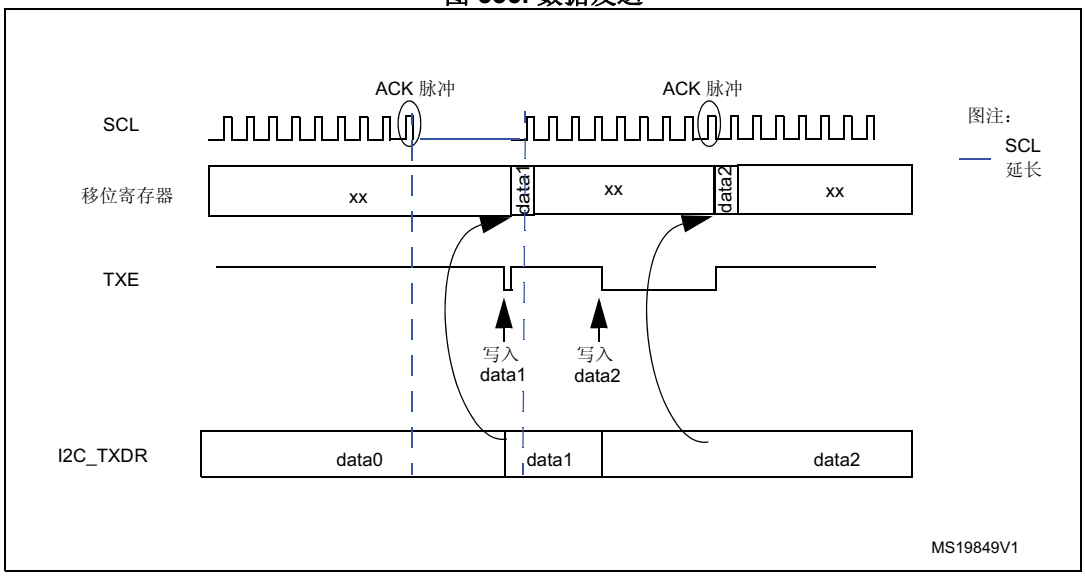
图 535. 数据接收



发送

如果 I2C_TXDR 寄存器不为空 (TXE=0)，则其内容会在第 9 个 SCL 脉冲（应答脉冲）后复制到移位寄存器中。然后移位寄存器的内容会移出到 SDA 线上。如果 TXE=1（意味着 I2C_TXDR 内尚未写入任何数据），则将延长 SCL 线的低电平时间，直到写入了 I2C_TXDR 为止。在第 9 个 SCL 脉冲后进行延长。

图 536. 数据发送



硬件传输管理

I2C 在硬件中内置了字节计数器，以便在下列各种模式下管理字节传输和结束通信：

- 主模式下生成 NACK、STOP 和 ReSTART
- 从接收器模式下控制 ACK 是否发出
- SMBus 模式下生成/校验 PEC

字节计数器通常在主模式下使用。在从模式下，字节计数器默认为禁止状态，但可以通过软件来使能，方法是将 I2C_CR2 寄存器中的 SBC（从字节控制）位置 1。

待传输的字节数在 I2C_CR2 寄存器的 NBYTES[7:0] 位域中进行编程。如果待传输的字节数 (NBYTES) 大于 255，或者接收方希望控制是否对接收到的数据字节进行应答，则必须选择重载模式，方法是将 I2C_CR2 寄存器的 RELOAD 位置 1。在该模式下，完成 NBYTES 中所编程字节数的数据传输之后，TCR 标志将置 1，并且 TCIE 置 1 时将生成中断。只要 TCR 标志置 1，SCL 便会延长。当往 NBYTES 写入一个非零值时，TCR 由软件清零。

在往 NBYTE 中设置最后一次传输的字节数前，必须把 RELOAD 位清零。

当主模式下 RELOAD=0 时，可在以下 2 种模式下使用计数器：

- **自动结束模式**（I2C_CR2 寄存器中的 AUTOEND = “1”）。在该模式下，一旦完成 NBYTES[7:0] 位域中所编程字节数的数据传输，主器件便会自动发送停止位。
- **软件结束模式**（I2C_CR2 寄存器中的 AUTOEND = “0”）在该模式下，一旦完成 NBYTES[7:0] 位域中所编程字节数的数据传输，TC 标志将置 1，并且 TCIE 置 1 时将生成中断。只要 TC 标志置 1，SCL 信号便会延长，需要软件介入操作。当软件把 I2C_CR2 寄存器中的起始位或停止位置 1 时，TC 标志将被清零。当主器件要发送重复起始位时，必须使用该模式。

注意： 当 RELOAD 位置 1 时，AUTOEND 位将不起作用。

表 362. I2C 配置表

功能	SBC 位	RELOAD 位	AUTOEND 位
主 Tx/Rx NBYTES + STOP	x	0	1
主 Tx/Rx + NBYTES + RESTART	x	0	0
从 Tx/Rx 接收的所有字节都要回复应答	0	x	x
具有 ACK 控制的从 Rx	1	1	x

47.4.7 I2C 从模式

I2C 从模式初始化

要在从模式下工作，用户必须至少使能一个从地址。可使用 I2C_OAR1 和 I2C_OAR2 这两个寄存器来编程自身从地址 OA1 和 OA2。

- OA1 既可配置为 7 位寻址模式（默认），也可通过将 I2C_OAR1 寄存器的 OA1MODE 位置 1 配置为 10 位寻址模式。

  通过将 I2C_OAR1 寄存器中的 OA1EN 位置 1 来使能 OA1。
- 如果需要额外的从地址，可配置第 2 个从地址 OA2。将 I2C_OAR2 寄存器的 OA2MSK[2:0] 位置 1 最多可屏蔽 7 个 OA2 LSB。因此，当 OA2MSK 配置为 1 到 6 时，将分别只有 OA2[7:2]、OA2[7:3]、OA2[7:4]、OA2[7:5]、OA2[7:6] 或 OA2[7] 与接收到的地址作比较。只要 OA2MSK 不等于 0， OA2 的地址比较器便会排除 I2C 保留地址 （0000 XXX 和 1111 XXX），这些地址将不会得到应答。如果 OA2MSK=7，接收到的所有 7 位地址（保留地址除外）均得到应答。 OA2 始终为 7 位地址。

  如果这些保留地址在 I2C_OAR1 或 I2C_OAR2 寄存器中进行了编程并且 OA2MSK=0，则它们可以在通过特定使能位使能后得到应答。

  通过将 I2C_OAR2 寄存器中的 OA2EN 位置 1 来使能 OA2。
- 通过将 I2C_CR1 寄存器中的 GCEN 位置 1 来使能广播呼叫地址。

当通过 I2C 的其中一个使能地址来寻址到该 I2C 设备时，ADDR 中断状态标志将置 1，并且 ADDRIE 位置 1 时将生成中断。

默认情况下，从器件使用其时钟延长功能（即必要时延长 SCL 信号的低电平时间）来为软件操作的执行提供时机。如果主器件不支持时钟延长，则必须对 I2C 进行如下配置：将 I2C_CR1 寄存器中 NOSTRETCH 位置 1。

接收到 ADDR 中断后，如果使能多个地址，则用户必须读取 I2C_ISR 寄存器中的 ADDCODE[6:0] 位，以确定是哪个地址匹配。还必须检查 DIR 标志，以获悉传输方向。

带时钟延长的从模式 (NOSTRETCH = 0)

在默认模式下，I2C 从器件会在以下情况下延长 SCL 时钟：

- ADDR 标志置 1 时：接收到的地址与其中一个使能的从地址匹配。通过软件将 ADDRCF 位置 1 以清零 ADRR 标志时，将释放该时钟延展。
- 发送时，前一次数据传输已完成但 I2C_TXDR 寄存器中未写入任何新数据，或者 ADDR 标志清零 (TXE=1) 时未写入第一个数据字节。往 I2C_TXDR 寄存器中写入数据时，将释放该时钟延展。
- 接收时，尚未读取 I2C_RXDR 寄存器但新的数据接收已完成。读取 I2C_RXDR 时，将释放该时钟延展。
- 当从器件字节控制模式和重载模式（SBC=1 且 RELOAD=1）下 TCR = 1 时，这意味着最后一个数据字节已完成传输。通过向 NBYTES[7:0] 字段写入一个非零值以将 TCR 清零时，将释放该时钟延展。
- 在 SCL 下降沿检测之后，I2C 会延长 SCL 的低电平时间（不超过 $[(SDADEL+SCLDEL+1) \times (PRESC+1) + 1] \times t_{I2CCLK}$）。

不带时钟延长的从模式 (NOSTRETCH = 1)

当 I2C_CR1 寄存器中的 NOSTRETCH = 1 时，I2C 从器件不会延长 SCL 信号。

- ADDR 标志置 1 时，不会延长 SCL 时钟。
- 发送时，必须在与发送数据对应的第一个 SCL 脉冲出现之前，向 I2C_TXDR 寄存器写入数据。否则，会发生下溢，I2C_ISR 寄存器中的 OVR 标志将置 1，如果 I2C_CR1 寄存器中的 ERRIE 位置 1，还将生成中断。当第一次数据发送开始而 STOPF 位仍置 1（尚未清零）时，OVR 标志也将置 1。因此，如果写入下一次传输要发送的第一个数据后才清零上一次传输的 STOPF 标志，则应提供 OVR 状态，甚至对于待发送的第一个数据也是如此。
- 接收时，必须在下一个数据字节的第 9 个 SCL 脉冲（ACK 脉冲）出现之前，从 I2C_RXDR 寄存器读取数据。否则，会发生上溢， I2C_ISR 寄存器中的 OVR 标志将置 1，如果 I2C_CR1 寄存器中的 ERRIE 位置 1，还将生成中断。

从器件字节控制模式

要在从接收模式下实现字节 ACK 控制，必须通过将 I2C_CR1 寄存器中的 SBC 位置 1 来使能从器件字节控制模式。这样符合 SMBus 标准。

要在从接收模式下实现字节 ACK 控制，必须选择重载模式 (RELOAD=1)。要控制每个字节，必须在 ADDR 中断子程序中将 NBYTES 初始化为 0x1，并在每接收一个字节后将 NBYTE 重载为 0x1。接收到字节后，TCR 位将置 1，从而延长 SCL 信号的第 8 个和第 9 个脉冲之间的低电平时间。用户可以从 I2C_RXDR 寄存器中读取数据，然后通过配置 I2C_CR2 寄存器中的 ACK 位来决定是否应答。通过将 NBYTES 编程为非零值来释放 SCL 延长：发送应答或不应答信号，然后可继续接收下一个字节。

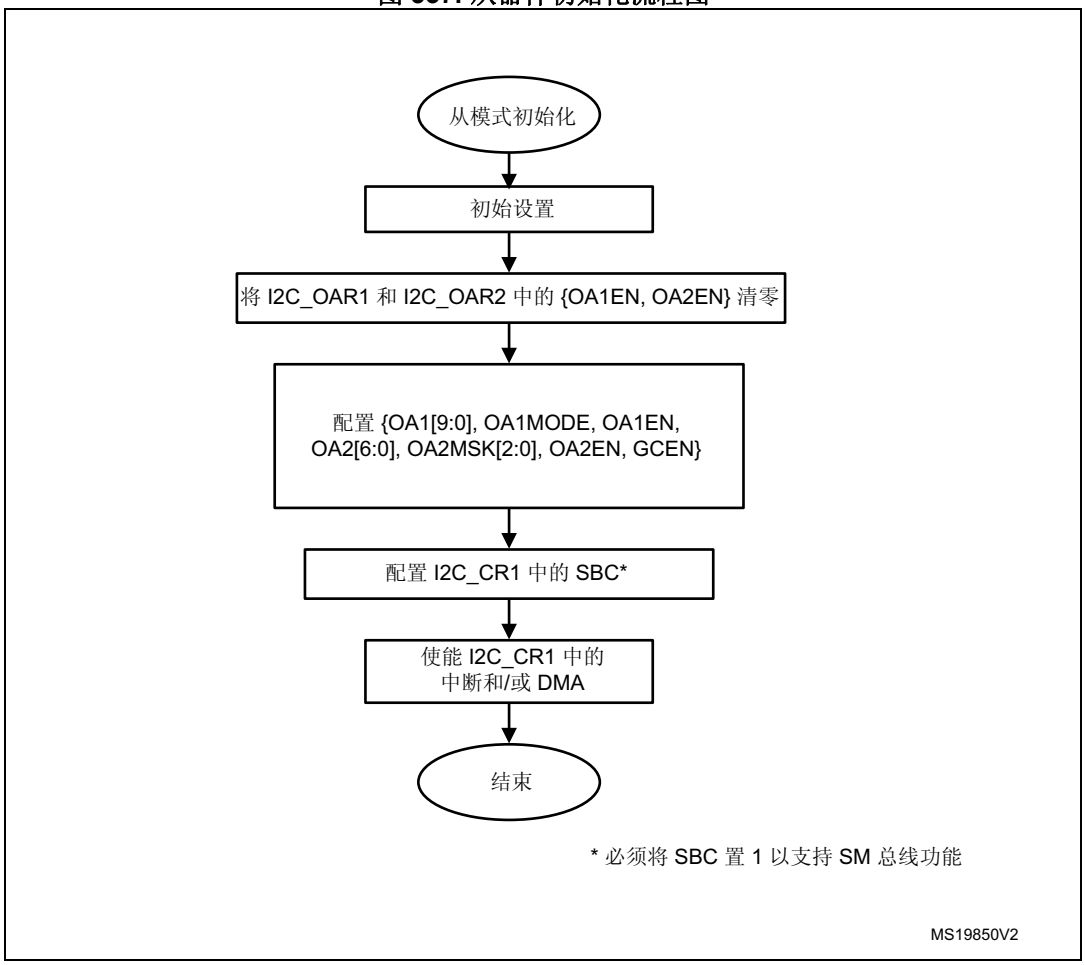
NBYTES 可加载大于 0x1 的值，在这种情况下，接收流在 NBYTES 个数据接收期间是连续的。

注： *SBC 位只能在 I2C 被禁止时、从器件不被寻址时或 ADDR=1 时配置。*

ADDR=1 或 TCR=1 时，可以更改 RELOAD 位的值。

注意： 从器件字节控制模式与 NOSTRETCH 模式不兼容。不允许在 NOSTRETCH=1 时将 SBC 位置 1。

图 537. 从器件初始化流程图



从发送器

当 I2C_TXDR 寄存器为空时，将生成发送中断状态 (TXIS)。如果 I2C_CR1 寄存器中的 TXIE 位置 1，将生成中断。

I2C_TXDR 寄存器中写入待发送的下一个数据字节时，TXIS 位将被清零。

接收到 NACK 时，I2C_ISR 寄存器中的 NACKF 位将置 1，如果 I2C_CR1 寄存器中的 NACKIE 位置 1，还将生成中断。从器件自动释放 SCL 和 SDA 线，以使主器件执行停止或重复起始位的发送。收到 NACK 时，TXIS 位不会置 1。

当接收到停止位且 I2C_CR1 寄存器中的 STOPIE 位置 1 时，I2C_ISR 寄存器中的 STOPF 标志将置 1 并且会生成中断。在大多数应用中，SBC 位通常编程为“0”。在这种情况下，如果接收到从地址 (ADDR=1) 时 TXE = 0，用户可以选择发送 I2C_TXDR 寄存器的内容作为第一个数据字节，也可以选择通过将 TXE 位置 1 来刷新 I2C_TXDR 寄存器以编程新的数据字节。

在从器件字节控制模式 (SBC=1) 下，必须在地址匹配中断子程序 (ADDR=1) 中向 NBYTE 写入待发送数据的个数。在这种情况下，传输期间 TXIS 事件的数量对应于 NBYTES 中编程的值。

注意： 如果 NOSTRETCH = 1，当 ADDR 标志置 1 时不会延长 SCL 时钟，因此用户无法在 ADDR 子程序中刷新 I2C_TXDR 寄存器的内容，从而编程第一个数据字节。必须在 I2C_TXDR 寄存器中预编程待发送的第一个数据字节：

- 该数据可以是前一个传输消息的最后一个 TXIS 事件中写入的数据。
- 如果该数据字节不是待发送的数据字节，可通过将 TXE 位置 1 来刷新 I2C_TXDR 寄存器，从而编程新的数据字节。必须仅在执行完这些操作后再清零 STOPF 位，以确保在第一次数据传输之后地址应答之前执行这些操作。

  如果第一次数据传输开始时 STOPF 仍置 1，则将生成下溢错误（OVR 标志置 1）。

  如果需要 TXIS 事件（发送中断或发送 DMA 请求），用户必须将 TXE 位和 TXIS 位均置 1，以便生成 TXIS 事件。

图 538. I2C 从发送器的传输序列流程图 (NOSTRETCH=0)

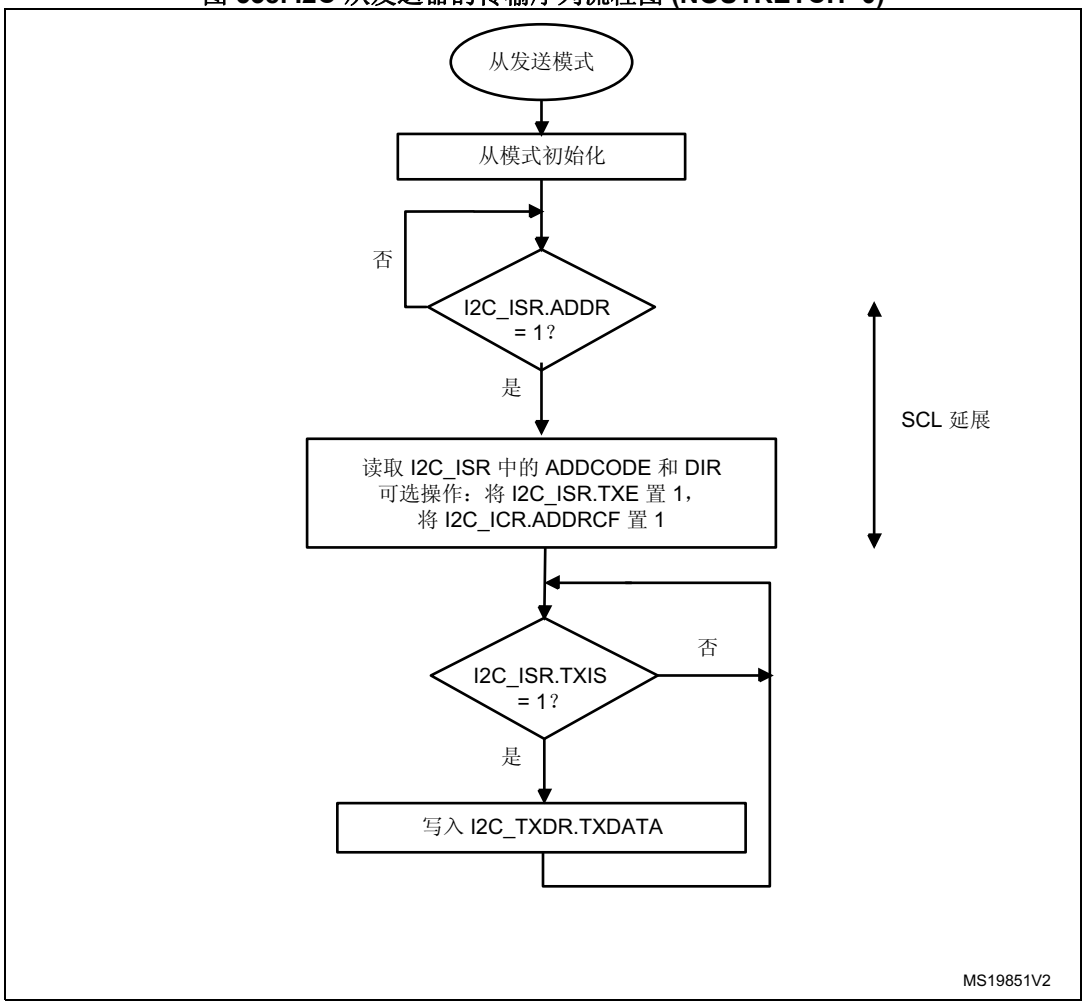


图 539. I2C 从发送器的传输序列流程图 (NOSTRETCH=1)

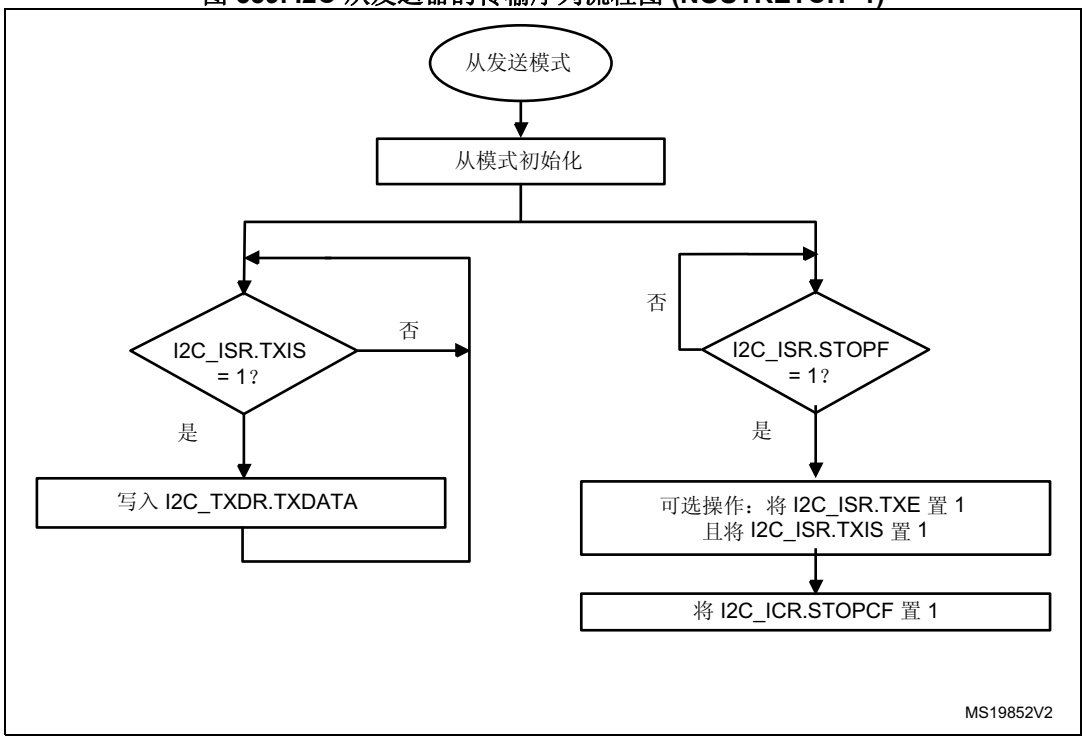
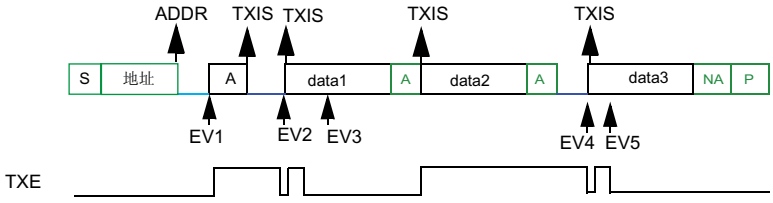


图 540. I2C 从发送器的传输总线图

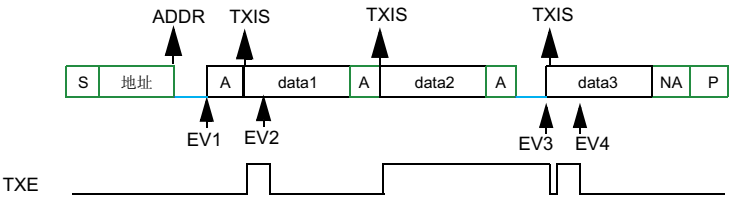
示例：I2C 从器件发送 3 个字节，刷新第 1 个数据，NOSTRETCH=0：



图注：发送 / 接收 / SCL 延长

EV1：ADDR ISR：检查 ADDCODE 和 DIR，将 TXE 置 1，将 ADDRCF 置 1
EV2：TXIS ISR：写入 data1
EV3：TXIS ISR：写入 data2
EV4：TXIS ISR：写入 data3
EV5：TXIS ISR：写入 data4（不发送）

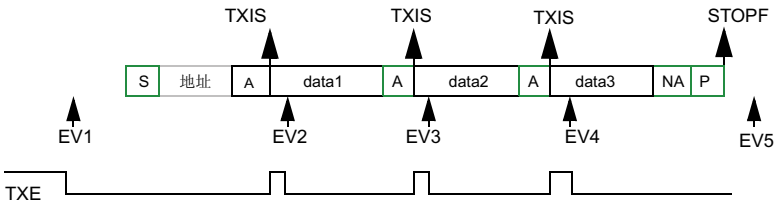
示例：I2C 从器件发送 3 个字节，不刷新第 1 个数据，NOSTRETCH=0：



图注：发送 / 接收 / SCL 延长

EV1：ADDR ISR：检查 ADDCODE 和 DIR，将 ADDRCF 置 1
EV2：TXIS ISR：写入 data2
EV3：TXIS ISR：写入 data3
EV4：TXIS ISR：写入 data4（不发送）

示例：I2C 从器件发送 3 个字节，NOSTRETCH=1：



图注：发送 / 接收 / SCL 延长

EV1：写入 data1
EV2：TXIS ISR：写入 data2
EV3：TXIS ISR：写入 data3
EV4：TXIS ISR：写入 data4（不发送）
EV5：STOPF ISR：（可选操作：将 TXE 和 TXIS 置 1），将 STOPCF 置 1

MS19853V1

从接收器

当 I2C_RXDR 满时，I2C_ISR 中的 RXNE 将置 1，如果 I2C_CR1 中的 RXIE 置 1，还将生成中断。读取 I2C_RXDR 时，将清零 RXNE。

接收到停止条件且 I2C_CR1 寄存器中的 STOPIE 置 1 时，I2C_ISR 中的 STOPF 将置 1 并且会生成中断。

图 541. 从接收器的传输序列流程图 (NOSTRETCH=0)

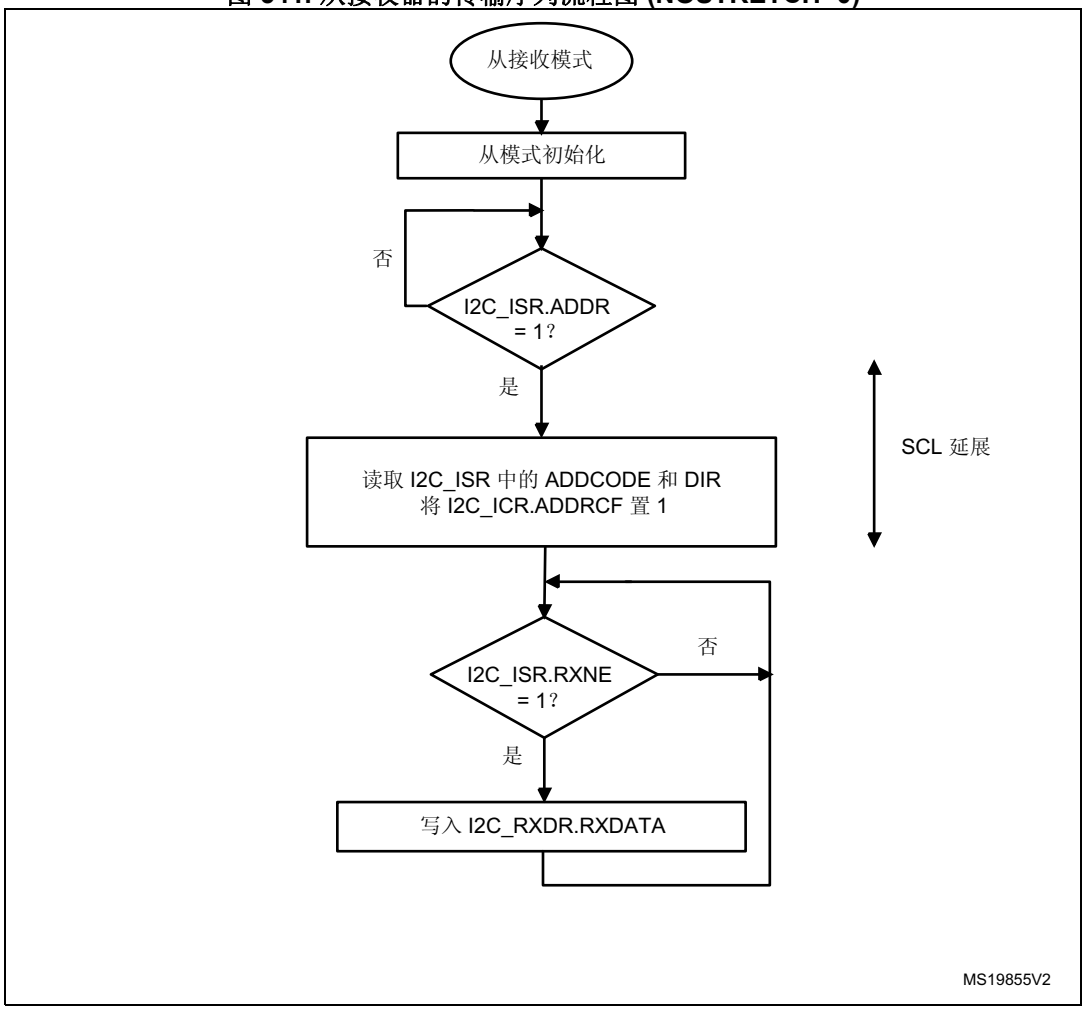


图 542. 从接收器的传输序列流程图 (NOSTRETCH=1)

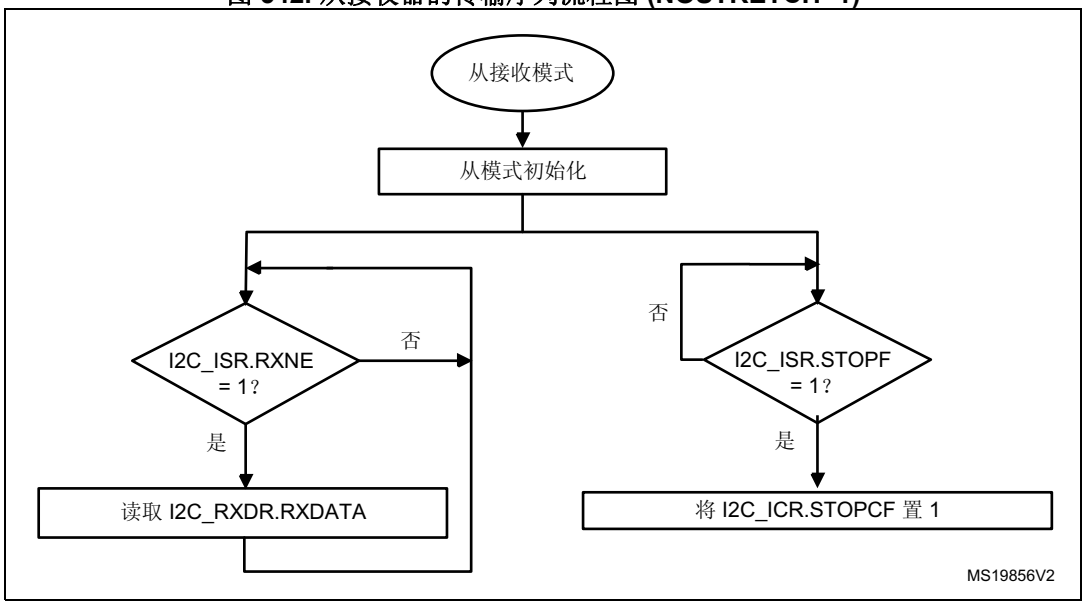
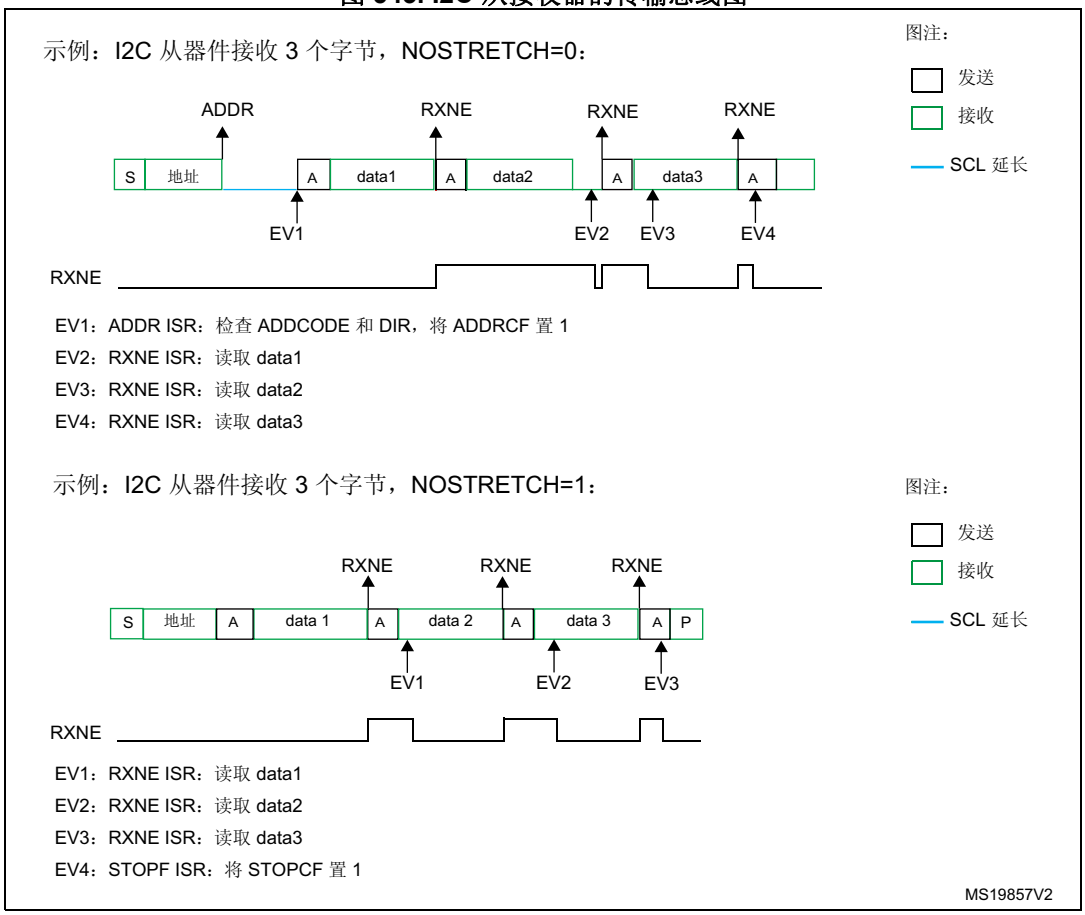


图 543. I2C 从接收器的传输总线图



47.4.8 I2C 主模式

I2C 主模式初始化

使能外设前，必须通过设置 I2C_TIMINGR 寄存器中的 SCLH 和 SCLL 位来配置 I2C 主时钟。

STM32CubeMX 工具进行计算并将 I2C_TIMINGR 内容提供在 I2C 配置窗口中。

为了支持多主环境和从时钟延长，I2C 实现了时钟同步机制。

为了实现时钟同步，需执行以下操作：

- 使用 SCLL 计数器从 SCL 低电平内部检测开始对时钟的低电平进行计数。
- 使用 SCLH 计数器从 SCL 高电平内部检测开始对时钟的高电平进行计数。

I2C 经过 t_{SYNC1} 延时后检测其自身的 SCL 低电平，该延时取决于 SCL 下降沿、SCL 输入噪声滤波器（模拟 + 数字）以及 SCL 与 I2CxCLK 时钟的同步。一旦 SCLL 计数器达到 I2C_TIMINGR 寄存器的 SCLL[7:0] 位中编程的值，I2C 便会将 SCL 释放为高电平。

I2C 经过 t_{SYNC2} 延时后检测其自身的 SCL 高电平，该延时取决于 SCL 上升沿、SCL 输入噪声滤波器（模拟 + 数字）以及 SCL 与 I2CxCLK 时钟的同步。一旦 SCLH 计数器达到 I2C_TIMINGR 寄存器的 SCLH[7:0] 位中编程的值，I2C 便会使 SCL 变为低电平。

因此，主时钟周期为：

$$t_{SCL} = t_{SYNC1} + t_{SYNC2} + \{[(SCLH+1) + (SCLL+1)] \times (PRESC+1) \times t_{I2CCLK}\}$$

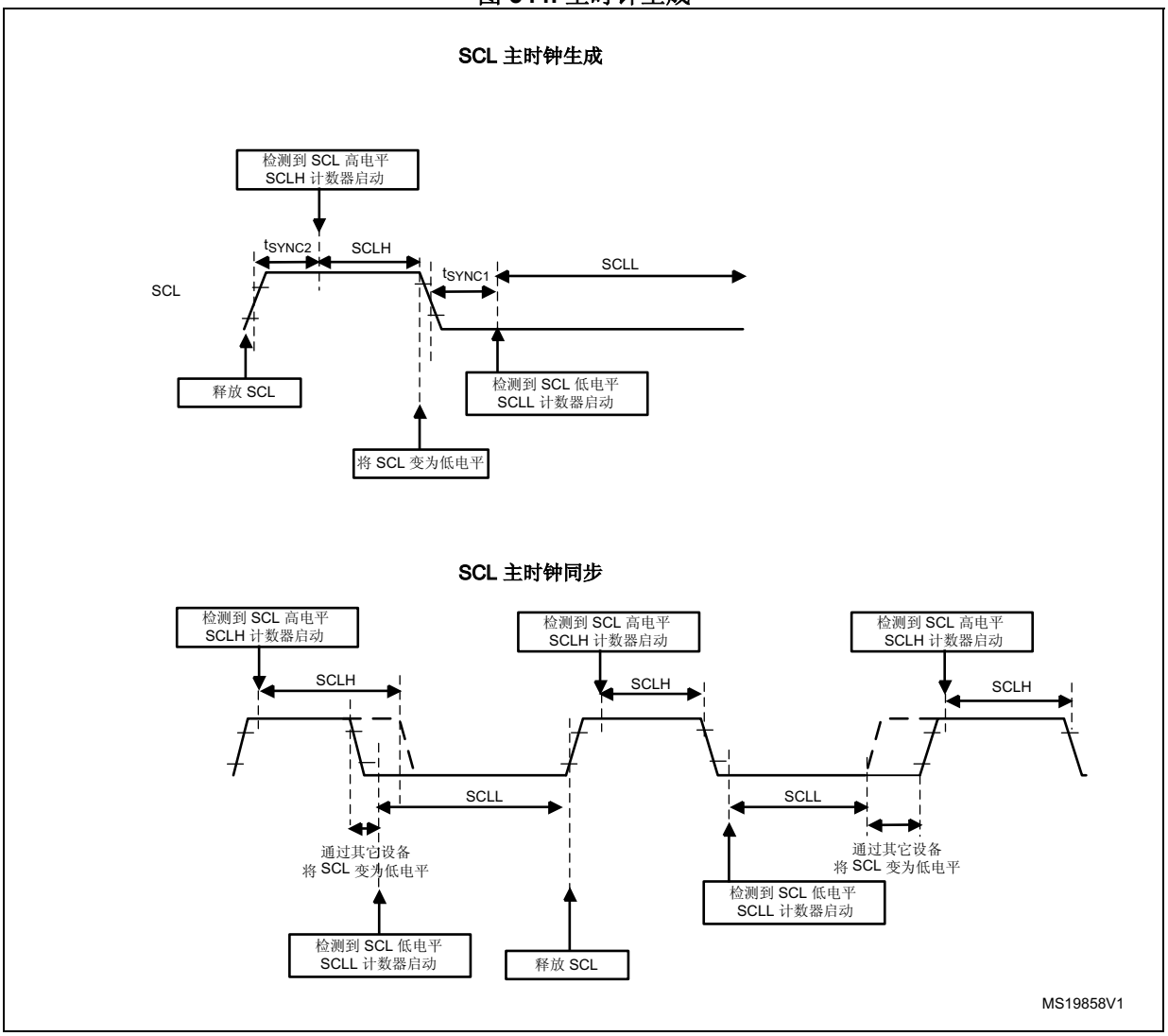
t_{SYNC1} 的持续时间取决于以下参数：

- SCL 下降斜率
- 模拟滤波器（使能时）引入的输入延时
- 数字滤波器（使能时）引入的输入延时：DNF x t_{I2CCLK}
- SCL 与 i2c_ker_ck 时钟建立同步而产生的延时（2 到 3 个 i2c_ker_ck 周期）

t_{SYNC2} 的持续时间取决于以下参数：

- SCL 上升斜率
- 模拟滤波器（使能时）引入的输入延时
- 数字滤波器（使能时）引入的输入延时：DNF x t_{I2CCLK}
- SCL 与 i2c_ker_ck 时钟建立同步而产生的延时（2 到 3 个 i2c_ker_ck 周期）

图 544. 主时钟生成



注意： 为了符合 I^2C 或 SMBus 规范，主时钟必须遵循下表中给出的时序：

表 363. I²C-SMBUS 规范时钟时序

符号	参数	标准模式 (Sm)		快速模式 (Fm)		超快速模式 (Fm+)		SMBus		单位
		最小值	最大值	最小值	最大值	最小值	最大值	最小值	最大值	
f_{SCL}	SCL 时钟频率	-	100	-	400	-	1000	-	100	kHz
$t_{HD:STA}$	（重复）起始条件的保持时间	4.0	-	0.6	-	0.26	-	4.0	-	µs
$t_{SU:STA}$	重复起始条件的建立时间	4.7	-	0.6	-	0.26	-	4.7	-	µs
$t_{SU:STO}$	停止条件的建立时间	4.0	-	0.6	-	0.26	-	4.0	-	µs
t_{BUF}	停止条件和起始条件之间的总线空闲时间	4.7	-	1.3	-	0.5	-	4.7	-	µs
t_{LOW}	SCL 时钟的低电平周期	4.7	-	1.3	-	0.5	-	4.7	-	µs
t_{HIGH}	SCL 时钟的高电平周期	4.0	-	0.6	-	0.26	-	4.0	50	µs
t_r	SDA 和 SCL 信号的上升时间	-	1000	-	300	-	120	-	1000	ns
t_f	SDA 和 SCL 信号的下降时间	-	300	-	300	-	120	-	300	ns

注： *SCLL 还用于生成 t_{BUF} 和 $t_{SU:STA}$ 时序。*

SCLH 还用于生成 $t_{HD:STA}$ 和 $t_{SU:STO}$ 时序。

有关 I2C_TIMINGR 设置与 i2c_ker_ck 频率的示例，请参见*第 47.4.9 节：I2C_TIMINGR 寄存器配置示例*。

主模式通信初始化（地址阶段）

要发起通信，用户必须在 I2C_CR2 寄存器中为寻址的从器件编程以下参数：

- 寻址模式（7 位或 10 位）：ADD10
- 待发送的从地址：SADD[9:0]
- 传输方向：RD_WRN
- 读取 10 位地址时：HEAD10R 位。必须对 HEAD10R 进行相应配置，以指示传输方向变化时必须发送完整的地址序列，还是只发送地址头。
- 待传输的字节数：NBYTES[7:0]。如果字节数等于或大于 255，则初始化时必须将 NBYTES[7:0] 填充为 0xFF。

然后，用户必须将 I2C_CR2 寄存器中的 START 位置 1。START 位置 1 时，不允许更改上述所有位。

之后，当主器件检测到总线空闲 (BUSY = 0) 时，它会在经过 t_{BUF} 的延时后自动发送起始位，随后发出从器件地址。

仲裁丢失时，主器件将自动切换回从模式，如果作为从器件被寻址，还可对其自身地址进行应答。

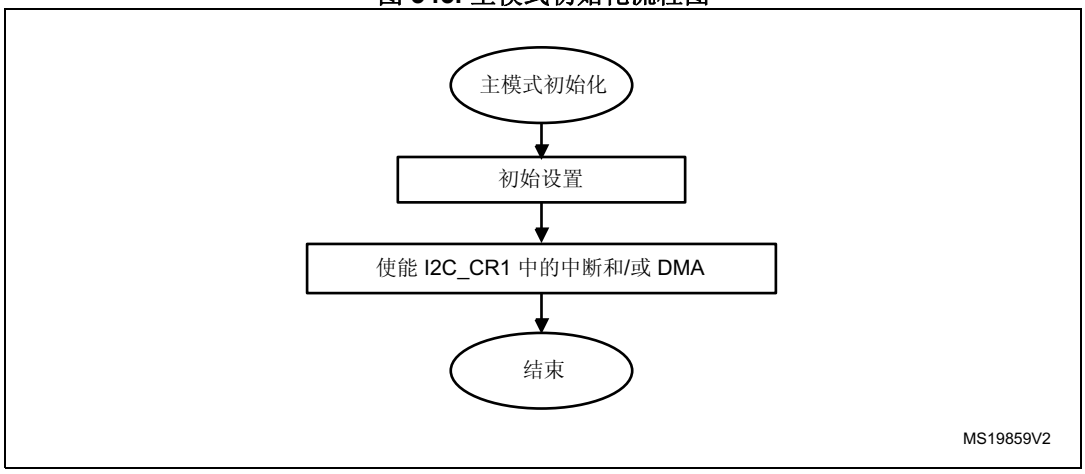
注： *无论接收到的应答值为何，只要已在总线上发送从地址，START 位便会由硬件复位。如果仲裁丢失，START 位也会由硬件复位。*

在 10 位寻址模式下，如果从器件不对从地址的前 7 位进行应答，则主器件将自动重新启动从地址发送，直至接收到 ACK。在这种情况下，如果从从器件接收到 NACK，则必须将 ADDRCF 置 1，以停止发送从地址。

如果当 START 位置 1 时，I2C 作为从器件 (ADDR=1) 被寻址，则 I2C 将切换为从模式，START 位将在 ADDRCF 位置 1 时清零。

注： *该步骤同样适用于重复起始位。在这种情况下，BUSY=1。*

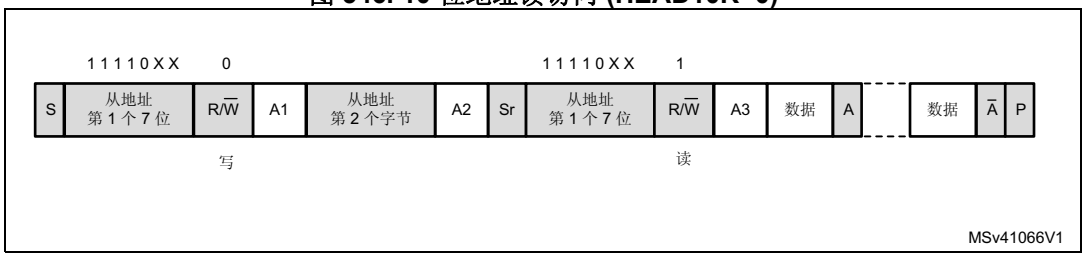
图 545. 主模式初始化流程图



主接收器寻址 10 位地址从器件的初始化过程

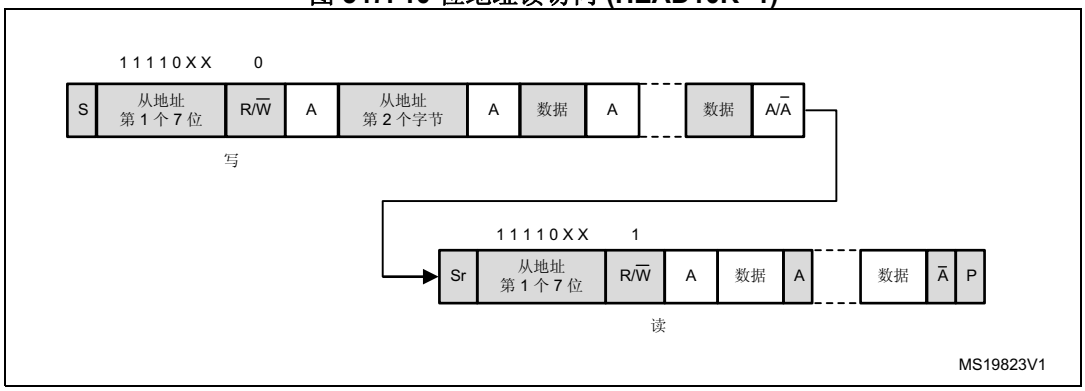
- 如果从地址采用 10 位格式，用户可选择将 I2C_CR2 寄存器中的 HEAD10R 位清零来发送完整的读序列。在这种情况下，主器件会在 START 位置 1 后自动发送以下完整序列：（重复）起始位 + 带写方向的从器件 10 位地址头字节 + 从器件地址第 2 个字节 + 重复起始位 + 带读方向的从器件 10 位地址头字节。

图 546. 10 位地址读访问 (HEAD10R=0)



- 如果主器件对 10 位地址从器件进行寻址、向该从器件发送数据、然后再从该从器件读取数据，则必须首先完成主器件发送过程。然后，重复起始位置 1，10 位从地址配置为 HEAD10R=1。在这种情况下，主器件发送以下序列：重复起始位 + 从地址 10 位头读取。

图 547. 10 位地址读访问 (HEAD10R=1)



主发送器

写传输时，在发送完每个字节（即第 9 个 SCL 脉冲（接收到 ACK 时））后，TXIS 标志将置 1。

如果 I2C_CR1 寄存器中的 TXIE 位置 1，TXIS 事件将生成中断。当 I2C_TXDR 寄存器中写入待发送的下一个数据字节时，该标志将被清零。

传输期间的 TXIS 事件的数量对应于 NBYTES[7:0] 中编程的值。如果待发送的数据字节总数大于 255，则必须通过将 I2C_CR2 寄存器中的 RELOAD 位置 1 来选择重载模式。在这种情况下，当 NBYTES 数据传输完成时，TCR 标志将置 1，并且 SCL 线的低电平将被延展，直到 NBYTES[7:0] 被写入非零值。

收到 NACK 时，TXIS 标志不会置 1。

- 当 RELOAD=0 且 NBYTES 数据传输完成时：
  - 在自动结束模式 (AUTOEND=1) 下，将自动发送停止位。
  - 在软件结束模式 (AUTOEND=0) 下，TC 标志将置 1 且 SCL 线的低电平将被延展，以便执行以下软件操作：

    可通过将 I2C_CR2 寄存器中的 START 位置 1 并配置适当的从地址和待传输字节数来请求发送重复起始位。将 START 位置 1 会将 TC 标志清零，并在总线上发送起始位。

    可通过将 I2C_CR2 寄存器中的 STOP 位置 1 来请求停止位。将 STOP 位置 1 会将 TC 标志清零，并在总线上发送停止位。
- 如果接收到 NACK：TXIS 标志不会置 1，并且接收到 NACK 后会自动发送停止位。I2C_ISR 寄存器中的 NACKF 标志置 1，如果 NACKIE 位置 1，还将生成中断。

图 548. I2C 主发送器的传输序列流程图（N≤255 字节）

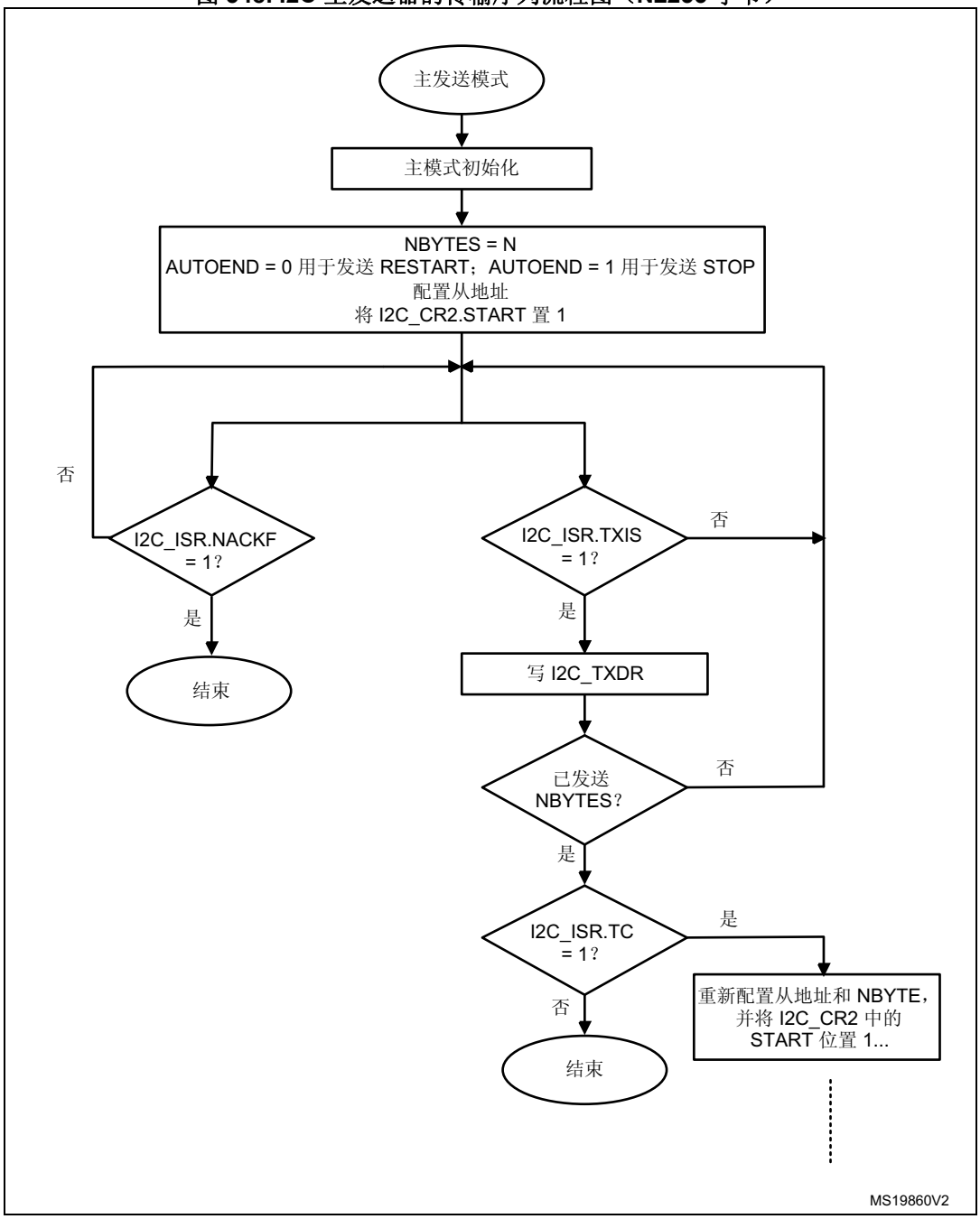


图 549. I2C 主发送器的传输序列流程图（N>255 字节）

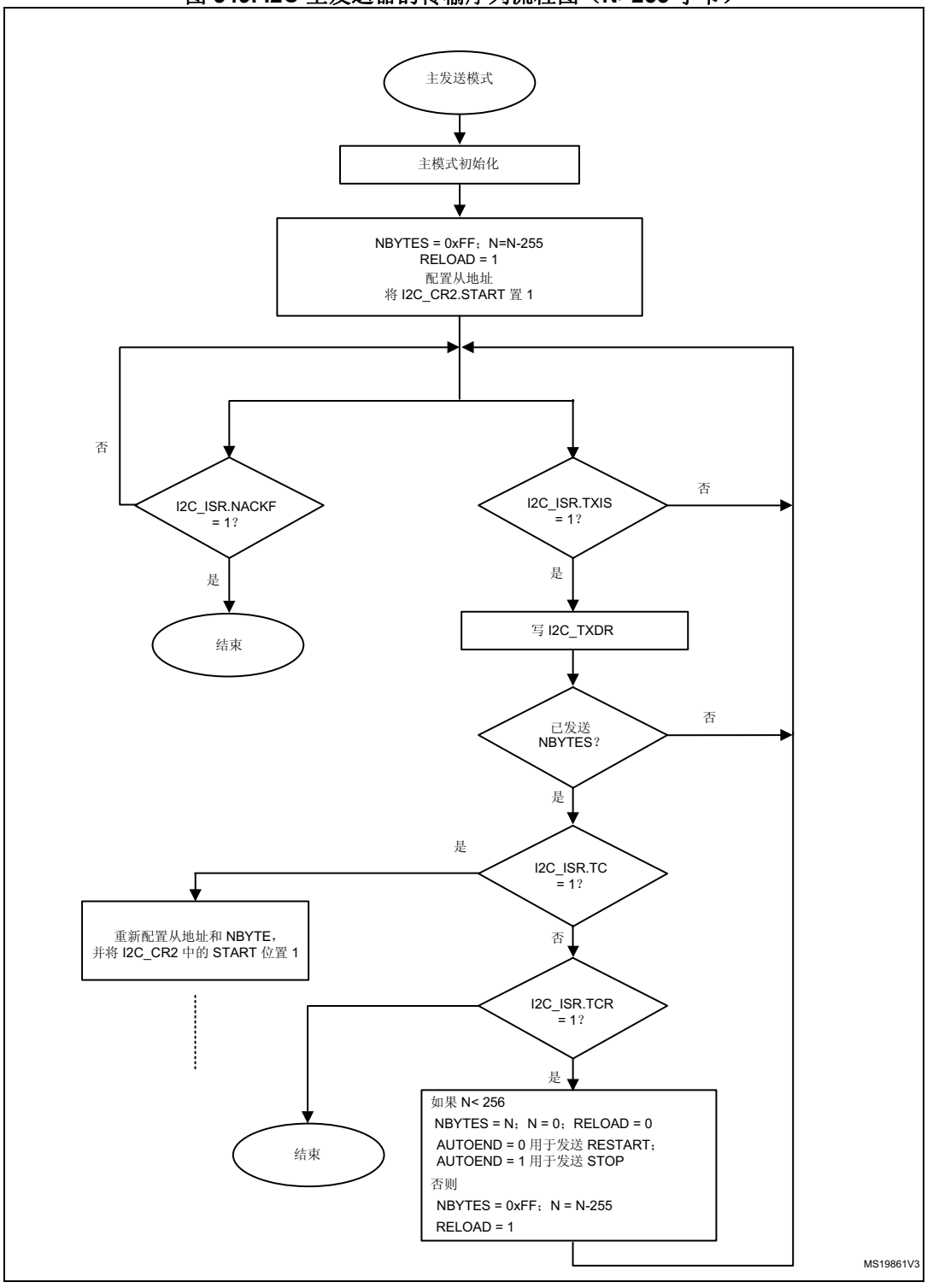
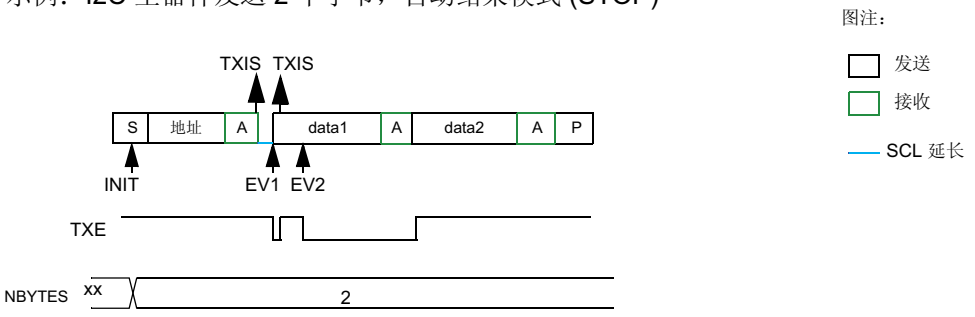


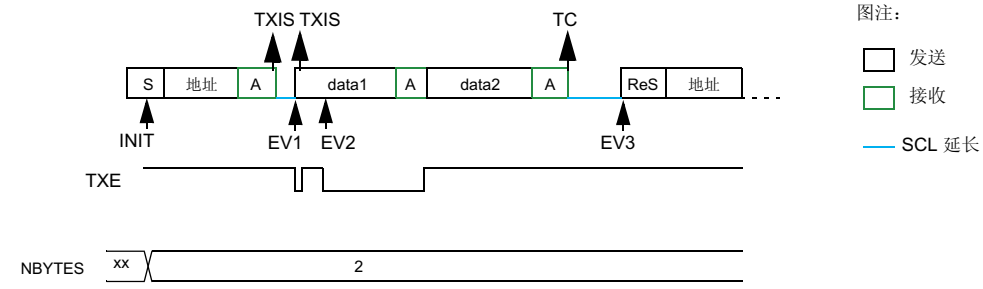
图 550. I2C 主发送器的传输总线图

示例：I2C 主器件发送 2 个字节，自动结束模式 (STOP)



INIT：设置从地址，设置 NBYTES = 2，AUTOEND=1，将 START 置 1
EV1：TXIS ISR：写入 data1
EV2：TXIS ISR：写入 data2

示例：I2C 主器件发送 2 个字节，软件结束模式 (RESTART)



INIT：设置从地址，设置 NBYTES = 2，AUTOEND=0，将 START 置 1
EV1：TXIS ISR：写入 data1
EV2：TXIS ISR：写入 data2
EV3：TC ISR：设置从地址，设置 NBYTES = N，将 START 置 1

MS19862V1

主接收器

读传输时，在接收到每个字节（即第 8 个 SCL 脉冲）后，RXNE 标志将置 1。如果 I2C_CR1 寄存器中的 RXIE 位置 1，RXNE 事件将生成中断。读取 I2C_RXDR 时，将清零该标志。

如果待接收的数据字节总数大于 255，则必须通过将 I2C_CR2 寄存器中的 RELOAD 位置 1 来选择重载模式。在这种情况下，当 NBYTES[7:0] 数据传输完成时，TCR 标志将置 1，并且 SCL 线的低电平将被延展，直到 NBYTES[7:0] 被写入非零值。

- 当 RELOAD=0 且 NBYTES[7:0] 数据传输完成时：
  - 在自动结束模式 (AUTOEND=1) 下，接收到最后一个字节后，将自动发送 NACK 和停止位。
  - 在软件结束模式 (AUTOEND=0) 下，接收到最后一个字节后，将自动发送 NACK，TC 标志将置 1 且 SCL 线的低电平将被延展，以便执行以下软件操作：

    可通过将 I2C_CR2 寄存器中的 START 位置 1 并配置适当的从地址和待传输字节数来请求发送重复起始位。将 START 位置 1 会将 TC 标志清零，并在总线上发送起始位，后跟从地址。

    可通过将 I2C_CR2 寄存器中的 STOP 位置 1 来请求停止位。将 STOP 位置 1 会将 TC 标志清零，并在总线上发送停止位。

图 551. I2C 主接收器的传输序列流程图（N≤255 字节）

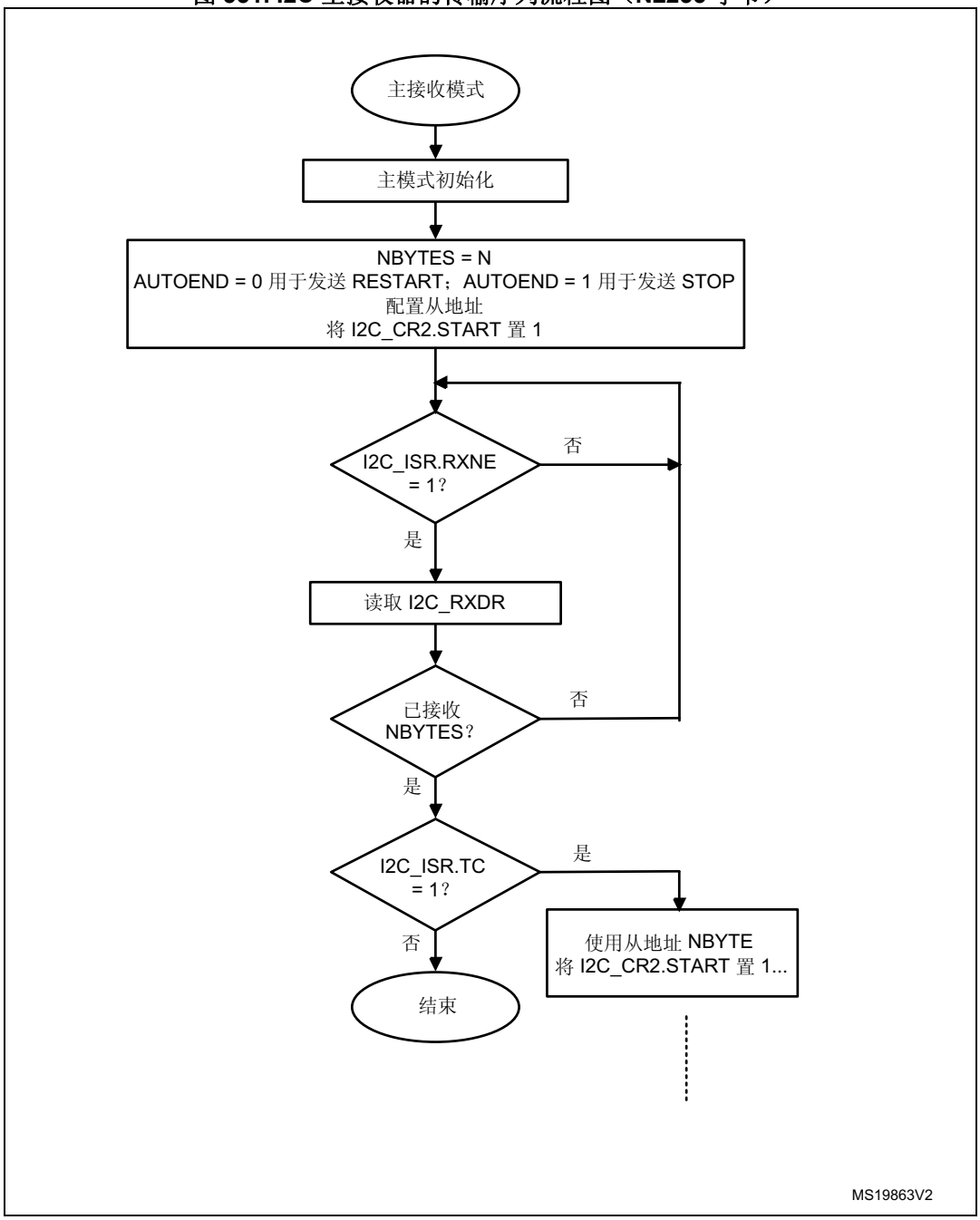


图 552. I2C 主接收器的传输序列流程图（N>255 字节）

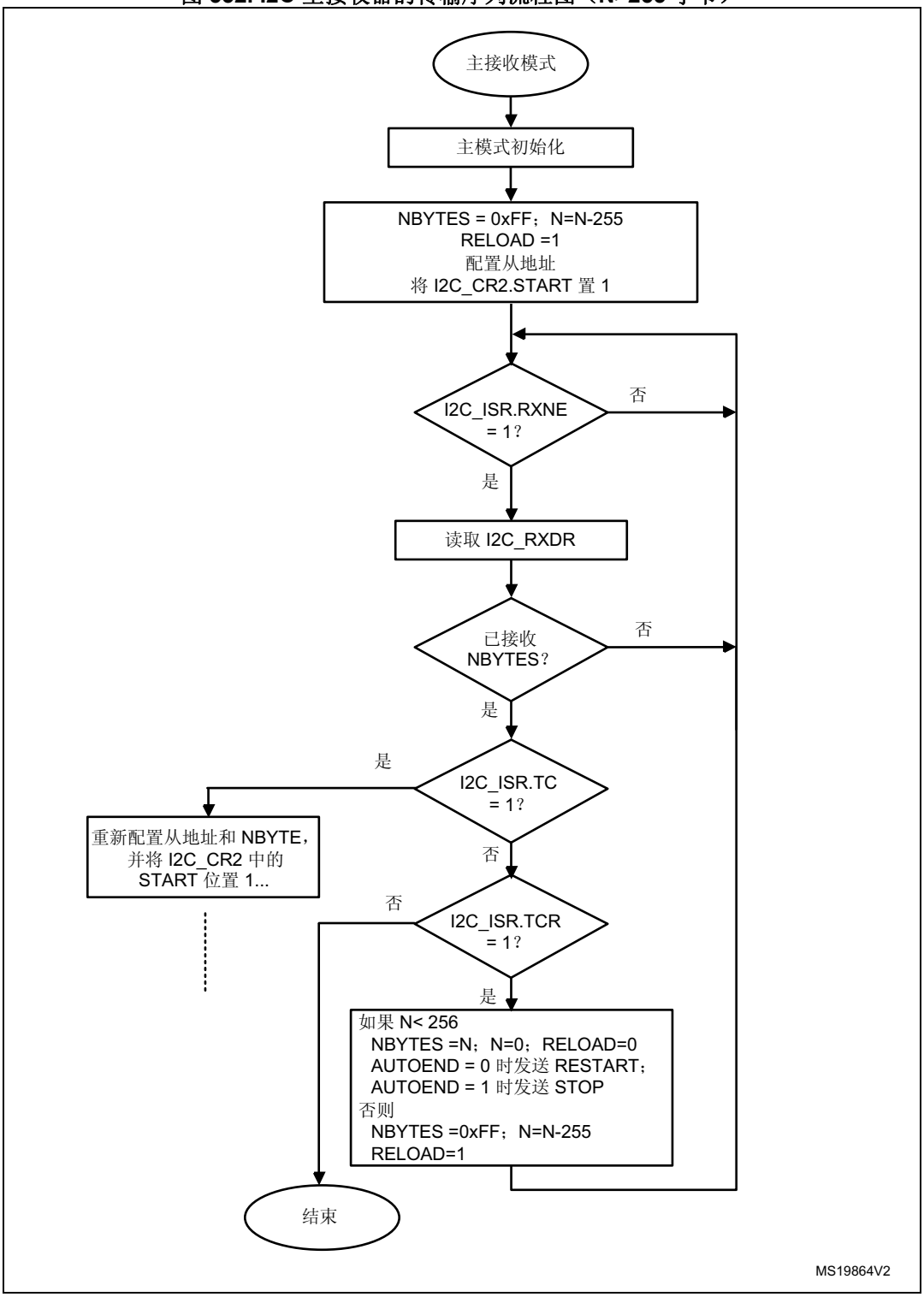
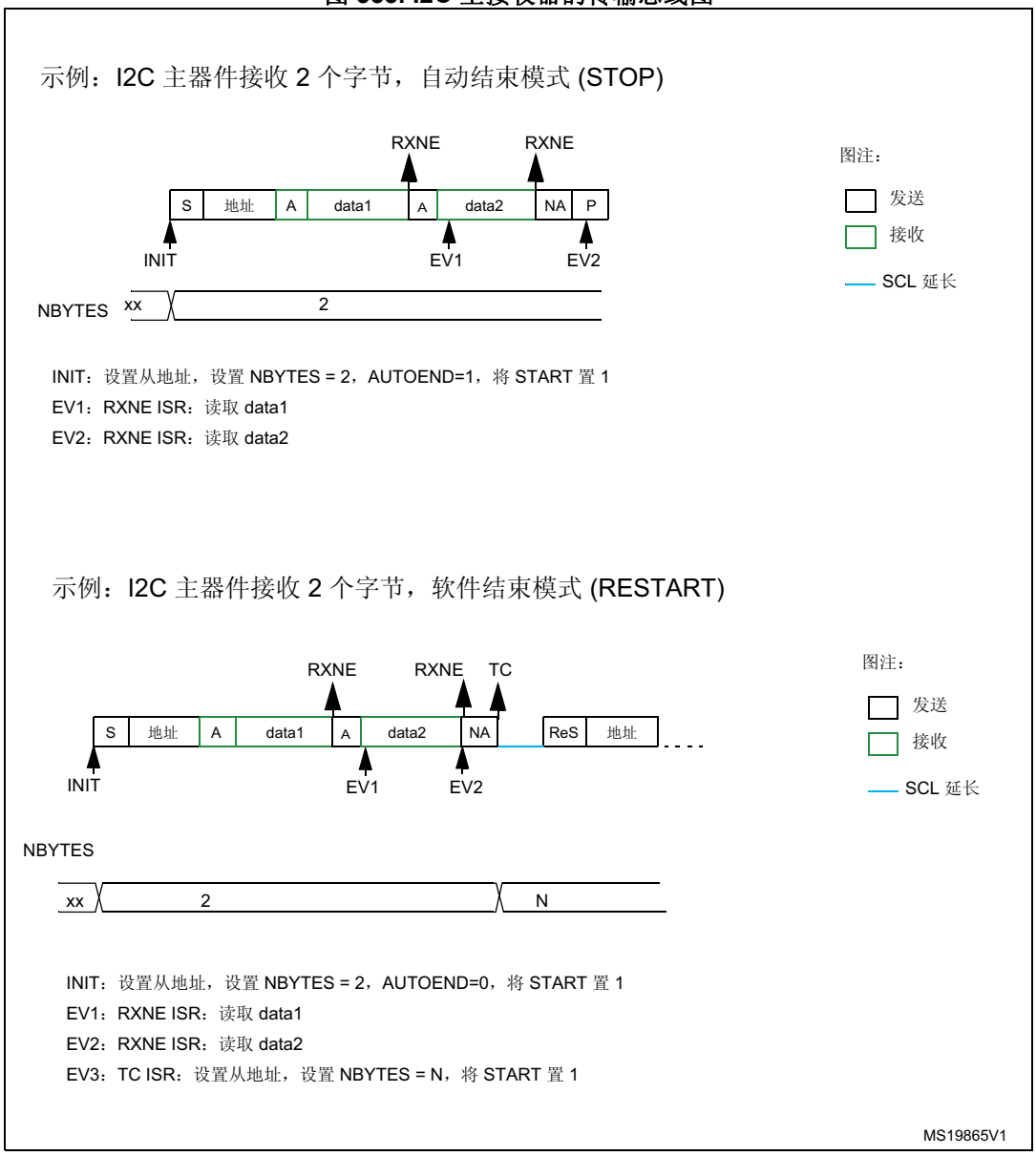


图 553. I2C 主接收器的传输总线图



47.4.9 I2C_TIMINGR 寄存器配置示例

下文各表提供了相应示例，以介绍如何编程 I2C_TIMINGR 才能获得符合 I^2C 规范的时序。要获取更准确的配置值，应使用 STM32CubeMX 工具（I2C 配置窗口）。

表 364. f_{I2CCLK} = 8 MHz 时的时序设置示例

参数	标准模式 (Sm)		快速模式 (Fm)	超快速模式 (Fm+)
	10 kHz	100 kHz	400 kHz	500 kHz
PRESC	1	1	0	0
SCLL	0xC7	0x13	0x9	0x6
t_{SCLL}	200x250 ns = 50 µs	20x250 ns = 5.0 µs	10x125 ns = 1250 ns	7x125 ns = 875 ns
SCLH	0xC3	0xF	0x3	0x3
t_{SCLH}	196x250 ns = 49 µs	16x250 ns = 4.0µs	4x125ns = 500ns	4x125 ns = 500 ns
t_{SCL}(1)	约 100 µs(2)	约 10 µs(2)	约 2500 ns(3)	约 2000 ns(4)
SDADEL	0x2	0x2	0x1	0x0
t_{SDADEL}	2x250 ns = 500 ns	2x250 ns = 500 ns	1x125 ns = 125 ns	0 ns
SCLDEL	0x4	0x4	0x3	0x1
t_{SCLDEL}	5x250 ns = 1250 ns	5x250 ns = 1250 ns	4x125 ns = 500 ns	2x125 ns = 250 ns

1. 由于 SCL 内部检测存在延时，SCL 周期 t_{SCL} 大于 t_{SCLL} + t_{SCLH}。为 t_{SCL} 提供的值仅用于举例说明。
2. t_{SYNC1} + t_{SYNC2} 最小值为 4 x t_{I2CCLK} = 500 ns。t_{SYNC1} + t_{SYNC2} = 1000 ns 时的示例。
3. t_{SYNC1} + t_{SYNC2} 最小值为 4 x t_{I2CCLK} = 500 ns。t_{SYNC1} + t_{SYNC2} = 750 ns 时的示例。
4. t_{SYNC1} + t_{SYNC2} 最小值为 4 x t_{I2CCLK} = 500 ns。t_{SYNC1} + t_{SYNC2} = 655 ns 时的示例。

表 365. f_{I2CCLK} = 16 MHz 时的时序设置示例

参数	标准模式 (Sm)		快速模式 (Fm)	超快速模式 (Fm+)
	10 kHz	100 kHz	400 kHz	1000 kHz
PRESC	3	3	1	0
SCLL	0xC7	0x13	0x9	0x4
t_{SCLL}	200 x 250 ns = 50 µs	20 x 250 ns = 5.0 µs	10 x 125 ns = 1250 ns	5 x 62.5 ns = 312.5 ns
SCLH	0xC3	0xF	0x3	0x2
t_{SCLH}	196 x 250 ns = 49 µs	16 x 250 ns = 4.0 µs	4 x 125 ns = 500 ns	3 x 62.5 ns = 187.5 ns
t_{SCL}(1)	约 100 µs(2)	约 10 µs(2)	约 2500 ns(3)	约 1000 ns(4)
SDADEL	0x2	0x2	0x2	0x0
t_{SDADEL}	2 x 250 ns = 500 ns	2 x 250 ns = 500 ns	2 x 125 ns = 250 ns	0 ns
SCLDEL	0x4	0x4	0x3	0x2
t_{SCLDEL}	5 x 250 ns = 1250 ns	5 x 250 ns = 1250 ns	4 x 125 ns = 500 ns	3 x 62.5 ns = 187.5 ns

1. 由于 SCL 内部检测存在延时，SCL 周期 t_{SCL} 大于 t_{SCLL} + t_{SCLH}。为 t_{SCL} 提供的值仅用于举例说明。
2. t_{SYNC1} + t_{SYNC2} 最小值为 4 x t_{I2CCLK} = 250 ns。t_{SYNC1} + t_{SYNC2} = 1000 ns 时的示例。
3. t_{SYNC1} + t_{SYNC2} 最小值为 4 x t_{I2CCLK} = 250 ns。t_{SYNC1} + t_{SYNC2} = 750 ns 时的示例。
4. t_{SYNC1} + t_{SYNC2} 最小值为 4 x t_{I2CCLK} = 250 ns。t_{SYNC1} + t_{SYNC2} = 500 ns 时的示例。

表 366. f_{I2CCLK} = 48 MHz 时的时序设置示例

参数	标准模式 (Sm)		快速模式 (Fm)	超快速模式 (Fm+)
	10 kHz	100 kHz	400 kHz	1000 kHz
PRESC	0xB	0xB	5	5
SCLL	0xC7	0x13	0x9	0x3
t_{SCLL}	200 x 250 ns = 50 µs	20 x 250 ns = 5.0 µs	10 x 125 ns = 1250 ns	4 x 125 ns = 500 ns
SCLH	0xC3	0xF	0x3	0x1
t_{SCLH}	196 x 250 ns = 49 µs	16 x 250 ns = 4.0 µs	4 x 125 ns = 500 ns	2 x 125 ns = 250 ns
t_{SCL}(1)	约 100 µs(2)	约 10 µs(2)	约 2500 ns(3)	约 875 ns(4)
SDADEL	0x2	0x2	0x3	0x0
t_{SDADEL}	2 x 250 ns = 500 ns	2 x 250 ns = 500 ns	3 x 125 ns = 375 ns	0 ns
SCLDEL	0x4	0x4	0x3	0x1
t_{SCLDEL}	5 x 250 ns = 1250 ns	5 x 250 ns = 1250 ns	4 x 125 ns = 500 ns	2 x 125 ns = 250 ns

1. 由于 SCL 内部检测存在延时，SCL 周期 t_{SCL} 大于 t_{SCLL} + t_{SCLH}。为 t_{SCL} 提供的值仅用于举例说明。
2. t_{SYNC1} + t_{SYNC2} 最小值为 4x t_{I2CCLK} = 83.3 ns。t_{SYNC1} + t_{SYNC2} = 1000 ns 时的示例。
3. t_{SYNC1} + t_{SYNC2} 最小值为 4x t_{I2CCLK} = 83.3 ns。t_{SYNC1} + t_{SYNC2} = 750 ns 时的示例。
4. t_{SYNC1} + t_{SYNC2} 最小值为 4x t_{I2CCLK} = 83.3 ns。t_{SYNC1} + t_{SYNC2} = 250 ns 时的示例。

47.4.10 SMBus 特性

仅当支持 SMBus 功能时，才涉及本节内容。请参见*第 47.3 节：I2C 特性实现*。

简介

系统管理总线 (SMBus) 是一个双线制接口，各器件可通过它在彼此之间或者与系统的其余部分进行通信。它以 I^2C 的工作原理为基础。SMBus 可针对系统和电源管理相关的任务提供控制总线。

该外设与 SMBUS 规范第 2.0 版兼容 (http://smbus.org)。

系统管理总线规范涉及三类器件。

- 从器件，用于接收或响应命令。
- 主器件，用于发出命令、生成时钟和中止传输。
- 主机，专用的主器件，可提供连接系统 CPU 的主接口。主机必须具有主 - 从器件功能，并且必须支持 SMBus 主机通知协议。系统中只允许存在一个主机。

该外设可配置为主器件或从器件，也可配置为主机。

SMBUS 以 I^2C 规范第 2.1 版为基础。

总线协议

任何给定器件都有十一种可用命令协议。器件既可以在这十一种协议中任选其一，也可以使用全部十一种协议进行通信。这十一种协议分别为快速命令、发送字节、接收字节、写入字节、写入字、读取字节、读取字、过程调用、块读取、块写入以及块写入-块读取过程调用。这些协议应通过用户软件实施。

有关这些协议的详细信息，请参见 SMBus 规范第 2.0 版 (http://smbus.org)。

地址解析协议 (ARP)

通过为各个从器件动态分配一个新的唯一地址可解决 SMBus 从地址冲突的问题。为了提供一种机制来针对地址分配隔离各个器件，各器件必须具有唯一的器件标识符 (UDID)。该 128 位数字由软件实现。

该外设支持地址解析协议 (ARP)。通过将 I2C_CR1 寄存器中的 SMBDEN 位置 1 来使能 SMBus 器件默认地址 (0b1100 001)。ARP 命令应通过用户软件实现。

此外，还将在从模式下执行仲裁以支持 ARP。

有关 SMBus 地址解析协议的详细信息，请参见 SMBus 规范第 2.0 版(http://smbus.org)。

接收的命令和数据应答控制

SMBus 接收器必须能够对接收到的每个命令或数据进行否定应答。要在从模式下实现 ACK 控制，必须通过将 I2C_CR1 寄存器中的 SBC 位置 1 来使能从字节控制模式。更多详细信息，请参见*第 1797 页的从器件字节控制模式*。

主机通知协议

该外设通过将 I2C_CR1 寄存器中的 SMBHEN 位置 1 来支持主机通知协议。在这种情况下，主机将应答 SMBus 主机地址 (0b0001 000)。

使用该协议时，器件用作主器件，而主机用作从器件。

SMBus 报警

器件支持 SMBus ALERT 可选信号。只具备从功能的器件可通过 SMBALERT# 引脚向主机发出信号，指示它想要通信。主机会处理该中断并通过报警响应地址 (0b0001 100) 同时访问所有 SMBALERT# 器件。只有那些将 SMBALERT# 拉到低电平的器件会应答报警响应地址。

如果配置为从器件 (SMBHEN=0)，则通过将 I2C_CR1 寄存器中的 ALERTEN 位置 1 来将 SMBA 引脚拉为低电平。这同时还会使能报警响应地址。

如果配置为主机 (SMBHEN=1)，则当 SMBA 引脚上检测到下降沿且 ALERTEN=1 时，I2C_ISR 寄存器中的 ALERT 标志置 1。如果 I2C_CR1 寄存器中的 ERRIE 位置 1，将生成中断。当 ALERTEN=0 时，即使外部 SMBA 引脚为低电平，ALERT 线也将被视为高电平。

如果无需 SMBus ALERT 引脚，则当 ALERTEN=0 时，SMBA *引脚可用作标准 GPIO*。

数据包错误校验

SMBus 规范中引入了数据包错误校验机制来提高可靠性和通信稳定性。数据包错误校验的实施方式是在每次消息传输结束时附加数据包错误代码 (PEC)。PEC 的计算方式是对所有消息字节（包括地址和读/写位）使用 CRC-8 多项式 $C(x) = x_8 + x^2 + x + 1$。

外设内置了硬件 PEC 计算器，可在接收到的字节与硬件计算的 PEC 不匹配时自动发送否定应答信号。

超时

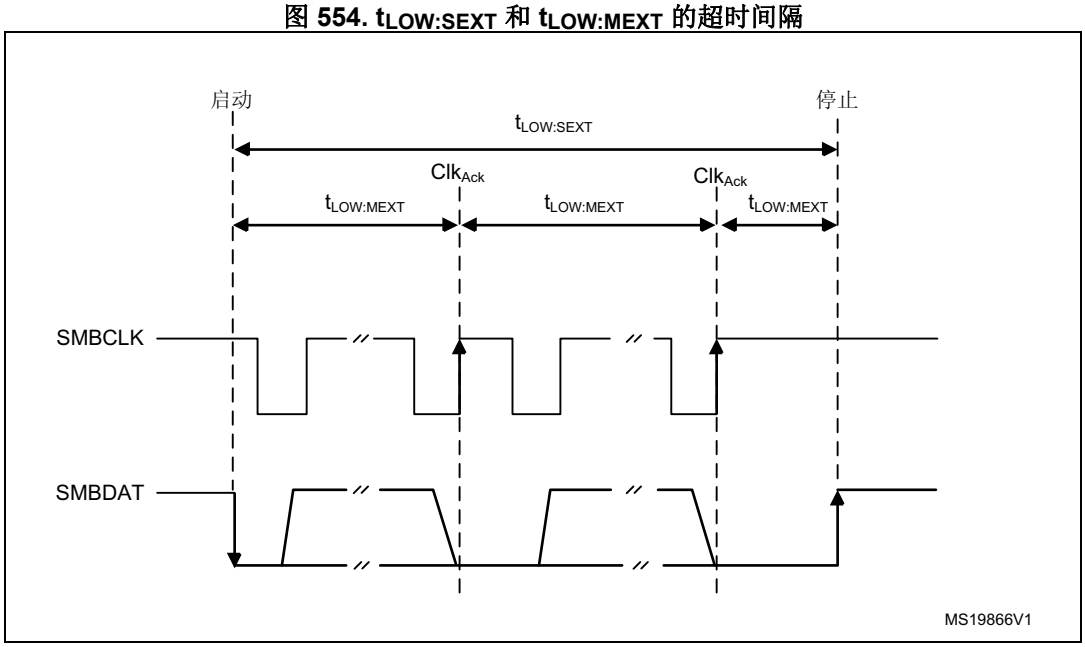
该外设内置了硬件定时器，以便符合 SMBus 规范第 2.0 版中定义的 3 个超时。

表 367. SMBus 超时规范

符号	参数	限值		单位
		最小值	最大值	
$t_{TIMEOUT}$	检测时钟低电平超时	25	35	ms
$t_{LOW:SEXT}$(1)	累积时钟低电平延长时间（从器件）	-	25	ms
$t_{LOW:MEXT}$(2)	累积时钟低电平延长时间（主器件）	-	10	ms

1. $t_{LOW:SEXT}$ 是一段累积时间，即给定从器件在一条消息的最初起始到停止期间时钟信号可延展的时间。其它从器件或主器件也可能延长时钟，进而导致时钟低电平总延长时间超过 $t_{LOW:SEXT}$。因此，测量该参数时该器件应该是全速主器件寻址的唯一器件。
2. $t_{LOW:MEXT}$ 是一段累积时间，即主器件在消息的每个字节（定义为 START 到 ACK、ACK 到 ACK 或 ACK 到 STOP）内时钟信号可延展的时间。从器件或其它主器件也可能延长时钟，进而导致时钟低电平总时间超过 $t_{LOW:MEXT}$（针对给定字节）。因此，测量该参数时该全速主器件只寻址一个从器件。

图 554. $t_{LOW:SEXT}$ 和 $t_{LOW:MEXT}$ 的超时间隔



总线空闲检测

如果主器件检测到时钟和数据信号的高电平时间已达 t_{IDLE}（超过 $t_{HIGH,MAX}$），则认为总线空闲（请参见*表 363：I2C-SMBUS 规范时钟时序*）。

该时序参数已考虑如下情况：主器件已动态添加至总线，但可能尚未检测到 SMBCLK 或 SMBDAT 线上的状态转换。在这种情况下，主器件必须等待足够长的时间，以确定当前未进行传输。外设支持硬件总线空闲检测。

47.4.11 SMBus 初始化

仅当支持 SMBus 功能时，才涉及本节内容。请参见*第 47.3 节：I2C 特性实现*。

除了 I2C 初始化之外，还必须进行一些其它的特定初始化，以便执行 SMBus 通信：

接收的命令和数据应答控制（从模式）

SMBus 接收器必须能够对接收到的每个命令或数据进行否定应答。要在从模式下实现 ACK 控制，必须通过将 I2C_CR1 寄存器中的 SBC 位置 1 来使能从字节控制模式。更多详细信息，请参见*第 1797 页的从器件字节控制模式*。

特定地址（从模式）

必要时应使能特定的 SMBus 地址。更多详细信息，请参见*第 1819 页的总线空闲检测*。

- 通过将 I2C_CR1 寄存器中的 SMBDEN 位置 1 来使能 SMBus 器件默认地址 (0b1100 001)。
- 通过将 I2C_CR1 寄存器中的 SMBHEN 位置 1 来使能 SMBus 主机地址 (0b0001 000)。
- 通过将 I2C_CR1 寄存器中的 ALERTEN 位置 1 来使能报警响应地址 (0b0001100)。

数据包错误校验

通过将 I2C_CR1 寄存器中的 PECEN 位置 1 来使能 PEC 的计算。然后，借助硬件字节计数器（I2C_CR2 寄存器中的 NBYTES[7:0]）来管理 PEC 传输。使能 I2C 之前，必须配置 PECEN 位。

PEC 传输由硬件字节计数器来管理，因此在从模式下连接 SMBus 时必须将 SBC 位置 1。当 PECBYTE 位置 1 且 RELOAD 位清零时，传输完 NBYTES-1 字节的数据后会传输 PEC。如果 RELOAD 置 1，PECBYTE 将不起作用。

注意： 使能 I2C 时，不允许更改 PECEN 配置。

表 368. 带 PEC 的 SMBUS 配置

模式	SBC 位	RELOAD 位	AUTOEND 位	PECBYTE 位
主 Tx/Rx NBYTES + PEC+ STOP	x	0	1	1
主 Tx/Rx NBYTES + PEC + ReSTART	x	0	0	1
从 Tx/Rx + PEC	1	0	x	1

超时检测

将 I2C_TIMEOUTR 寄存器中的 TIMOUTEN 和 TEXTEN 位置 1 来使能超时检测。定时器必须按如下方式编程：即在 SMBus 规范第 2.0 版规定的时间最大值之前检测出超时情况。

- $t_{TIMEOUT}$ 检查
  要使能 $t_{TIMEOUT}$ 检查，必须将 12 位 TIMEOUTA[11:0] 位编程为定时器重载值，以检查 $t_{TIMEOUT}$ 参数。必须将 TIDLE 位配置为“0”，以检测 SCL 低电平超时。
  然后，通过将 I2C_TIMEOUTR 寄存器中的 TIMOUTEN 位置 1 来使能定时器。
  如果 SCL 的低电平持续时间超过 (TIMEOUTA+1) x 2048 x t_{I2CCLK}，I2C_ISR 寄存器中的 TIMEOUT 标志将置 1。
  请参见*表 369：不同 i2c_ker_ck 频率下的 TIMEOUTA 设置示例（最大 $t_{TIMEOUT}$ = 25 ms）*。

注意： TIMEOUTEN 位置 1 时，不允许更改 TIMEOUTA[11:0] 位和 TIDLE 位的配置。

- $t_{LOW:SEXT}$ 和 $t_{LOW:MEXT}$ 检查
  必须根据外设配置为主器件还是从器件来配置 TIMEOUTB 定时器，以便为从器件校验 $t_{LOW:SEXT}$，为主器件校验 tLOW:MEXT。由于标准只规定了最大值，用户可以为这两个参数选择相同的值。
  然后，通过将 I2C_TIMEOUTR 寄存器中的 TEXTEN 位置 1 来使能定时器。
  如果 SMBus 外设延展 SCL 的累积时间超过 (TIMEOUTB+1) x 2048 x t_{I2CCLK}，并且达到*第 1819 页的总线空闲检测*一节给出的超时间隔，则 I2C_ISR 寄存器中的 TIMEOUT 标志将置 1。
  请参见*表 370：不同 i2c_ker_ck 频率下的 TIMEOUTB 设置示例*。

注意： TEXTEN 位置 1 时，不允许更改 TIMEOUTB 配置。

总线空闲检测

要使能 t_{IDLE} 检查，必须将 12 位 TIMEOUTA[11:0] 字段编程为定时器重载值，以获取 t_{IDLE} 参数。必须将 TIDLE 位配置为“1”，以检测 SCL 和 SDA 高电平超时。

然后，通过将 I2C_TIMEOUTR 寄存器中的 TIMOUTEN 位置 1 来使能定时器。

如果 SCL 和 SDA 线的高电平持续时间超过 (TIMEOUTA+1) x 4 x t_{I2CCLK}，I2C_ISR 寄存器中的 TIMEOUT 标志将置 1。

请参见*表 371：不同 i2c_ker_ck 频率下的 TIMEOUTA 设置示例（最大 t_{IDLE} = 50 µs）*。

注意： TIMEOUTEN 置 1 时，不允许更改 TIMEOUTA 和 TIDLE 配置。

47.4.12 SMBus：I2C_TIMEOUTR 寄存器配置示例

仅当支持 SMBus 功能时，才涉及本节内容。请参见*第 47.3 节：I2C 特性实现*。

- 将 $t_{TIMEOUT}$ 的最大持续时间配置为 25 ms：

表 369. 不同 i2c_ker_ck 频率下的 TIMEOUTA 设置示例（最大 $t_{TIMEOUT}$ = 25 ms）

f_{I2CCLK}	TIMEOUTA[11:0] 位	TIDLE 位	TIMEOUTEN 位	$t_{TIMEOUT}$
8 MHz	0x61	0	1	98 x 2048 x 125 ns = 25 ms
16 MHz	0xC3	0	1	196 x 2048 x 62.5 ns = 25 ms
48 MHz	0x249	0	1	586 x 2048 x 20.08 ns = 25 ms

- 将 $t_{LOW:SEXT}$ 和 $t_{LOW:MEXT}$ 的最大持续时间配置为 8 ms：

表 370. 不同 i2c_ker_ck 频率下的 TIMEOUTB 设置示例

f_{I2CCLK}	TIMEOUTB[11:0] 位	TEXTEN位	$t_{LOW:EXT}$
8 MHz	0x1F	1	32 x 2048 x 125 ns = 8 ms
16 MHz	0x3F	1	64 x 2048 x 62.5 ns = 8 ms
48 MHz	0xBB	1	188 x 2048 x 20.08 ns = 8 ms

- 将 t_{IDLE} 的最大持续时间配置为 50 µs

表 371. 不同 i2c_ker_ck 频率下的 TIMEOUTA 设置示例（最大 t_{IDLE} = 50 µs）

f_{I2CCLK}	TIMEOUTA[11:0] 位	TIDLE 位	TIMEOUTEN 位	t_{TIDLE}
8 MHz	0x63	1	1	100 x 4 x 125 ns = 50 µs
16 MHz	0xC7	1	1	200 x 4 x 62.5 ns = 50 µs
48 MHz	0x257	1	1	600 x 4 x 20.08 ns = 50 µs

47.4.13 SMBus 从模式

仅当支持 SMBus 功能时，才涉及本节内容。请参见*第 47.3 节：I2C 特性实现*。

除了 I2C 从模式传输管理（请参见*第 47.4.7 节：I2C 从模式*）之外，还提供了一些额外的软件流程图来支持 SMBus。

SMBus 从发送器

在 SMBus 模式下使用 IP 时，必须将 SBC 编程为“1”，以便在完成已编程数据字节数的传输后进行 PEC 传输。当 PECBYTE 位置 1 时，NBYTES[7:0] 中编程的字节数包含 PEC 传输。在这种情况下，总 TXIS 中断数为 NBYTES-1，如果主器件在完成 NBYTES-1 字节的数据传输后请求传输额外的字节，则将自动发送 I2C_PECR 寄存器的内容。

注意： 当 RELOAD 位置 1 时，PECBYTE 位将不起作用。

图 555. SMBus 从发送器的传输序列流程图（N 字节 + PEC）

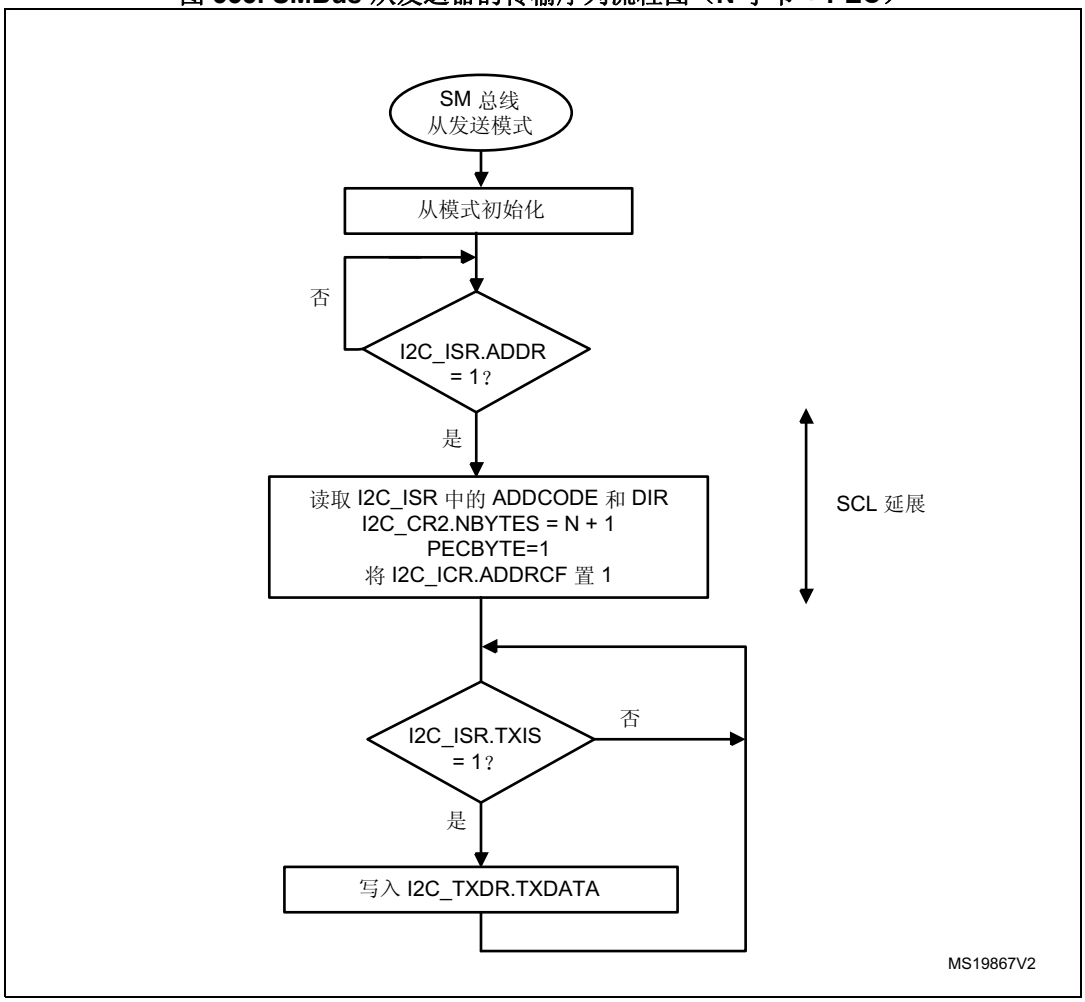
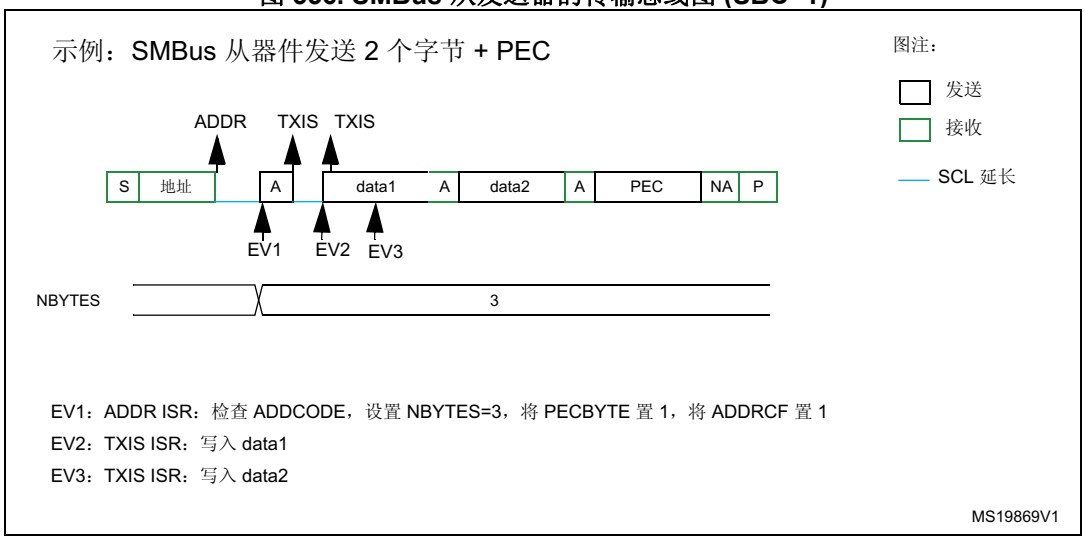


图 556. SMBus 从发送器的传输总线图 (SBC=1)



SMBus 从接收器

在 SMBus 模式下使用 I2C 时，必须将 SBC 编程为“1”，以便在完成已编程数据字节数的传输后进行 PEC 校验。要对每个字节进行 ACK 控制，必须选择重载模式 (RELOAD=1)。更多详细信息，请参见*第 1797 页的从器件字节控制模式*。

要校验 PEC 字节，必须将 RELOAD 位清零并将 PECBYTE 位置 1。在这种情况下，当接收到 NBYTES-1 字节的数据后，接收的下一个字节将与内部 I2C_PECR 寄存器的内容作比较。如果比较不匹配，则将自动生成 NACK 信号；如果比较匹配，则将自动生成 ACK 信号，而与 ACK 位的值无关。PEC 字节一经接收，便会像任何其它数据一样复制到 I2C_RXDR 寄存器中，并且 RXNE 标志将置 1。

当 PEC 不匹配时，PECERR 标志将置 1，如果 I2C_CR1 寄存器中的 ERRIE 位置 1，还将生成中断。

如果无需 ACK 软件控制，用户可编程 PECBYTE=1，在同一写操作下，将 NBYTES 编程为连续接收的字节数。接收到 NBYTES-1 字节的数据后，会将接收的下一个字节视为 PEC 进行校验。

注意： 当 RELOAD 位置 1 时，PECBYTE 位将不起作用。

图 557. SMBus 从接收器的传输序列流程图（N 字节 + PEC）

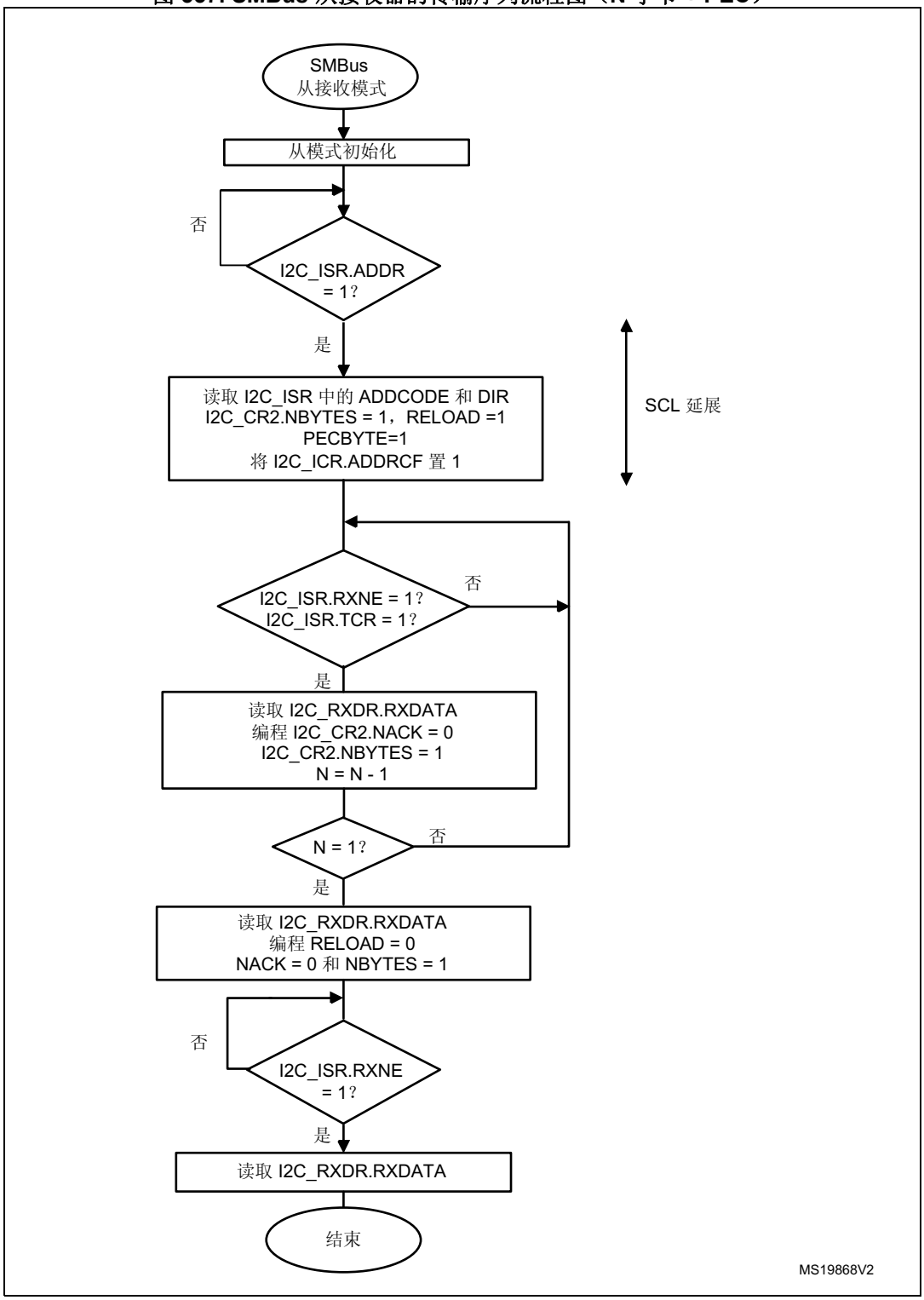
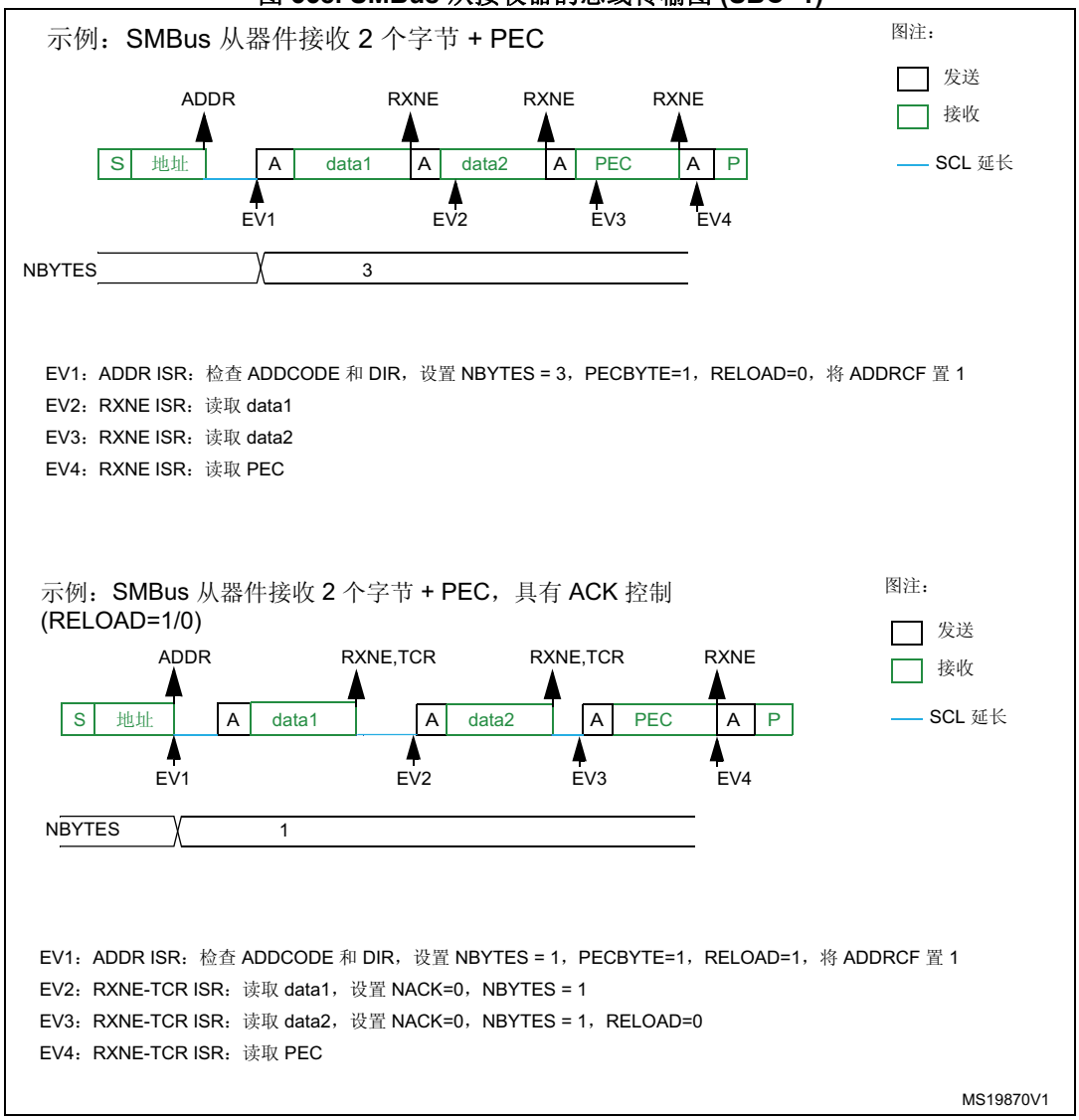


图 558. SMBus 从接收器的总线传输图 (SBC=1)



仅当支持 SMBus 功能时，才涉及本节内容。请参见*第 47.3 节：I2C 特性实现*。

除了 I2C 主模式传输管理（请参见*第 47.4.8 节：I2C 主模式*）之外，还提供了一些额外的软件流程图来支持 SMBus。

SMBus 主发送器

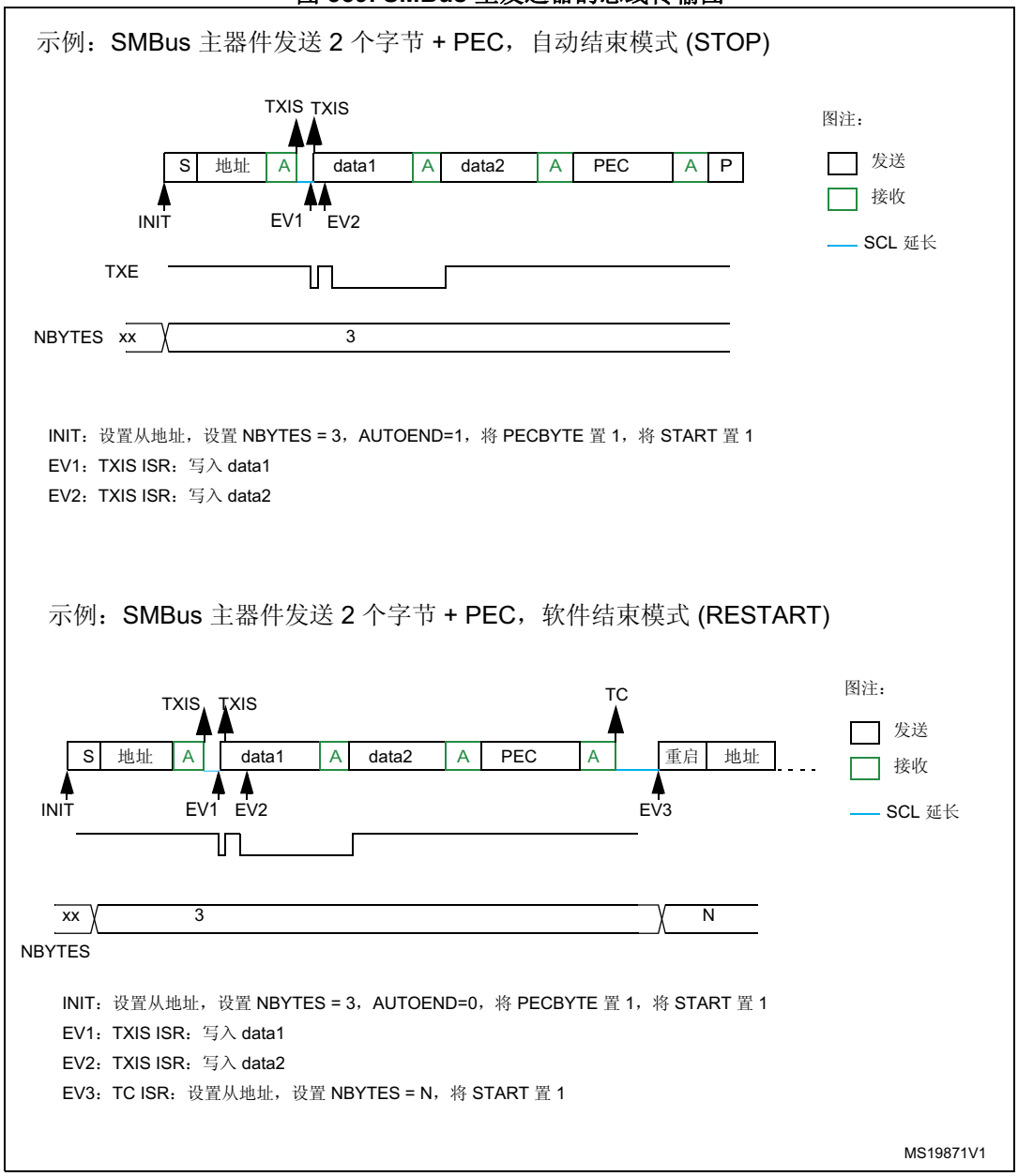
当 SMBus 主器件想要发送 PEC 时，必须在 START 位置 1 前，将 PECBYTE 位置 1 并在 NBYTES[7:0] 字段中设置字节数。在这种情况下，总 TXIS 中断数为 NBYTES-1。因此，如果 PECBYTE 位在 NBYTES=0x1 时置 1，则将自动发送 I2C_PECR 寄存器的内容。

如果 SMBus 主器件想要在 PEC 后发送停止位，则应选择自动结束模式 (AUTOEND=1)。在这种情况下，传输 PEC 后将自动发送停止位。

如果 SMBus 主器件想要在 PEC 后发送重复起始位，则必须选择软件模式 (AUTOEND=0)。在这种情况下，发送 NBYTES-1 字节的数据后，将发送 I2C_PECR 寄存器的内容，TC 标志将在传输完 PEC 之后置 1，SCL 线的低电平时间将延长。必须在 TC 中断子程序中设置重复起始位。

注意： 当 RELOAD 位置 1 时，PECBYTE 位将不起作用。

图 559. SMBus 主发送器的总线传输图



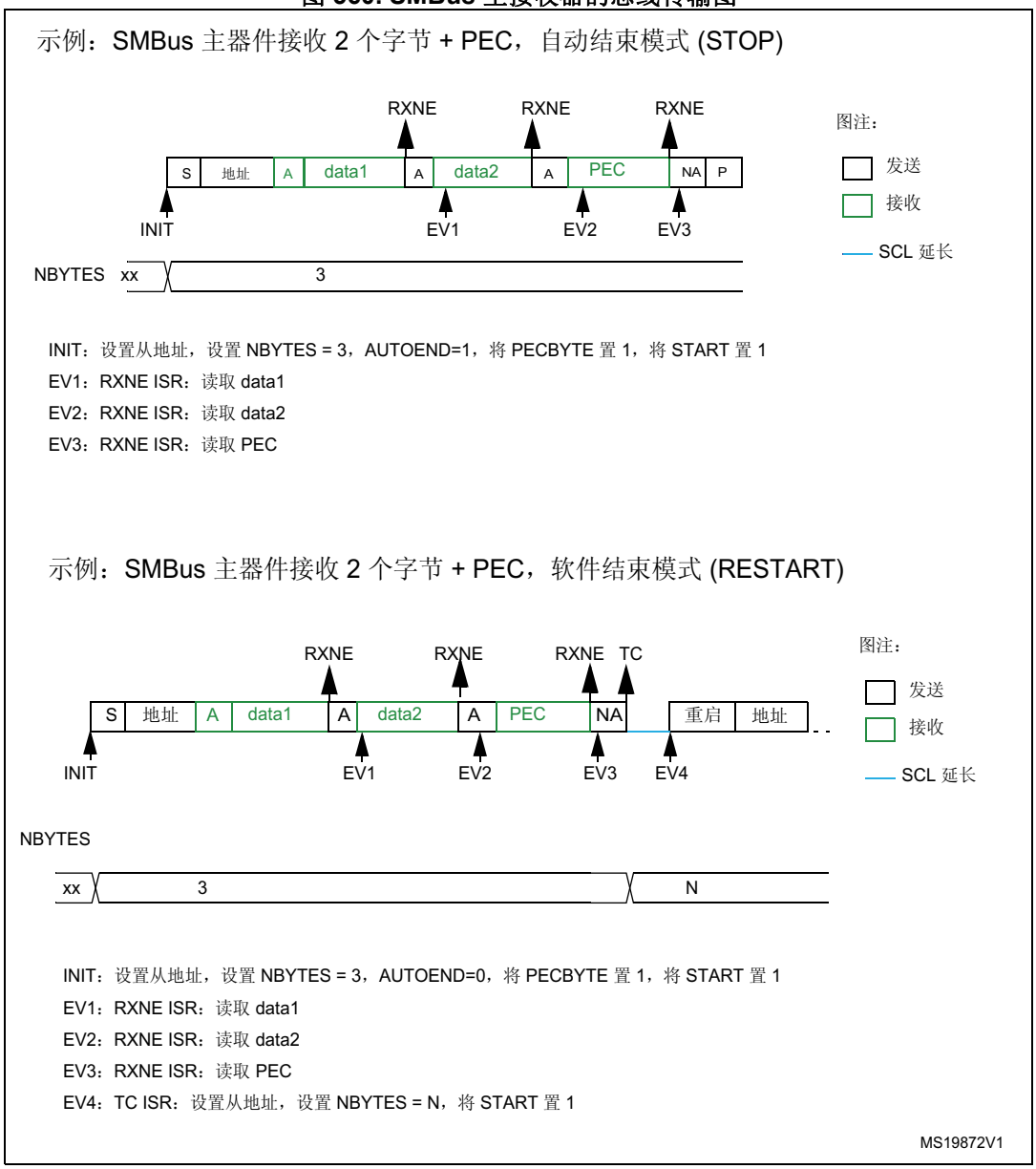
SMBus 主接收器

当 SMBus 主器件想要接收 PEC，并在传输结束后接收 STOP 时，可选择自动结束模式 (AUTOEND=1)。将 START 位置 1 之前，必须将 PECBYTE 位置 1 并设置从地址。在这种情况下，当接收到 NBYTES-1 字节的数据后，将自动使用 I2C_PECR 寄存器的内容对接收的下一个字节进行校验。PEC 字节（其后跟有停止位）将得到 NACK 响应。

当 SMBus 主接收器想要接收 PEC 字节，并且在传输结束后接收重复起始位时，必须选择软件模式 (AUTOEND=0)。将 START 位置 1 之前，必须将 PECBYTE 位置 1 并设置从地址。在这种情况下，当接收到 NBYTES-1 字节的数据后，将自动使用 I2C_PECR 寄存器的内容对接收的下一个字节进行校验。接收到 PEC 字节后，TC 标志将置 1，SCL 线的低电平时间将延长。可以在 TC 中断子程序中设置重复起始位。

注意： 当 RELOAD 位置 1 时，PECBYTE 位将不起作用。

图 560. SMBus 主接收器的总线传输图



47.4.14 地址匹配时从停止模式唤醒

仅当支持从停止模式唤醒功能时，才涉及本节内容。请参见*第 47.3 节：I2C 特性实现*。

被寻址时，I2C能够从停止模式中唤醒MCU（APB时钟关断）。支持所有寻址模式。

将 I2C_CR1 寄存器中的 WUPEN 位置 1，可以使能从停止模式唤醒功能。对于 i2c_ker_ck，必须选择 HSI 或 CSI 振荡器作为时钟源，以便从停止模式唤醒。

在停止模式下，内部振荡器关闭。当检测到 START 时，I2C 接口将内部振荡器接通，并延长 SCL 使其处于低电平直到唤醒内部振荡器。

内部振荡器随后用来接收地址。

地址匹配的情况下，MCU 唤醒时间内，I2C 延长 SCL 的低电平持续时间。通过软件清除 ADDR 标志后，此延长操作被释放，传输正常进行。

如果地址不匹配，内部振荡器再次关断，MCU 不被唤醒。

注： *如果 I2C 时钟是系统时钟，或者 WUPEN = 0，则接收到 START 后，内部振荡器不会接通。*

只有 ADDR 中断能够唤醒 MCU。因此，当 I2C 以主器件身份或在 ADDR 标志置 1 后以被寻址从器件身份执行传输时，不要进入停止模式。通过在 ADDR 中断程序中清除 SLEEPDEEP 位，然后仅在 STOPF 标志置 1 后再将其置 1，来对此进行管理。

注意： 数字滤波器与从停止模式唤醒功能不兼容。如果 DNF 位不等于 0，则将 WUPEN 位置 1 将不起任何作用。

注意： 只有当 I2C 时钟源为 HSI 或 CSI 振荡器时，该功能才可用。

注意： 必须使能时钟延长 (NOSTRETCH=0) 才能确保从停止模式唤醒功能正常工作。

注意： 如果禁止从停止模式唤醒 (WUPEN=0)，则在进入停止模式前必须禁止 I2C 外设 (PE=0)。

47.4.15 错误条件

以下错误条件可能导致通信失败。

总线错误 (BERR)

当检测到起始位或停止位但不位于第 9N 个 SCL 时钟脉冲之后时，会检测到总线错误。当 SDA 边沿出现且 SCL 为高电平时，会检测到起始或停止位。

只有当 I2C 在传输过程中用作主器件或被寻址为从器件时（即未处于从模式下的地址阶段），才会将总线错误标志置 1。

在从模式下检测到错位的起始位或重复起始位时，I2C 会像接收到正确的起始位一样进入地址识别状态。

检测到总线错误时，I2C_ISR 寄存器中的 BERR 标志将置 1，如果 I2C_CR1 寄存器中的 ERRIE 位置 1，还将生成中断。

仲裁丢失 (ARLO)

当 SDA 线上发送高电平但在 SCL 上升沿却采样到低电平时，会检测到仲裁丢失。

- 在主模式下，将在地址阶段、数据阶段和数据应答阶段检测到仲裁丢失。在这种情况下，SDA 线和 SCL 线被释放，起始控制位由硬件清零，主器件自动切换为从模式。
- 在从模式下，将在数据阶段和数据应答阶段检测到仲裁丢失。在这种情况下，传输停止，SCL 和 SDA 线被释放。

检测到仲裁丢失时，I2C_ISR 寄存器中的 ARLO 标志将置 1，如果 I2C_CR1 寄存器中的 ERRIE 位置 1，还将生成中断。

上溢/下溢错误 (OVR)

当满足 NOSTRETCH=1 和以下条件时，将在从模式下检测到上溢或下溢错误：

- 接收过程中接收到一个新字节但尚未读取 RXDR 寄存器。接收的新字节丢失，自动发送 NACK 来响应新字节。
- 在发送过程中：
  - 当 STOPF=1 且应发送第一个数据字节时。TXE=0 时发送 I2C_TXDR 寄存器的内容，否则发送 0xFF。
  - 应发送一个新字节但尚未向 I2C_TXDR 寄存器写入数据时，将发送 0xFF。

检测到上溢或下溢错误时，I2C_ISR 寄存器中的 OVR 标志将置 1，如果 I2C_CR1 寄存器中的 ERRIE 位置 1，还将生成中断。

数据包错误校验错误 (PECERR)

仅当支持 SMBus 功能时，才涉及本节内容。请参见*第 47.3 节：I2C 特性实现*。

当接收到的 PEC 字节与 I2C_PECR 寄存器的内容不匹配时，将检测到 PEC 错误。接收到错误的 PEC 后，将自动发送 NACK。

检测到 PEC 错误时，I2C_ISR 寄存器中的 PECERR 标志将置 1，如果 I2C_CR1 寄存器中的 ERRIE 位置 1，还将生成中断。

超时错误 (TIMEOUT)

仅当支持 SMBus 功能时，才涉及本节内容。请参见*第 47.3 节：I2C 特性实现*。

满足以下任何条件均会出现超时错误：

- TIDLE=0 且 SCL 的低电平持续时间达到 TIMEOUTA[11:0] 位中定义的时间：这用于检测 SMBus 超时。
- TIDLE=1 且 SDA 和 SCL 的高电平持续时间达到 TIMEOUTA[11:0] 位中定义的时间：这用于检测总线空闲情况。
- 主器件的累积时钟低电平延长时间达到了 TIMEOUTB[11:0] 位中定义的时间（SMBus $t_{LOW:MEXT}$ 参数）
- 从器件的累积时钟低电平延长时间达到了 TIMEOUTB[11:0] 位中定义的时间（SMBus $t_{LOW:SEXT}$ 参数）

当在主模式下检测到超时时，将自动发送停止位。

当在从模式下检测到超时时，将自动释放 SDA 和 SCL 线。

检测到超时错误时，I2C_ISR 寄存器中的 TIMEOUT 标志将置 1，如果 I2C_CR1 寄存器中的 ERRIE 位置 1，还将生成中断。

报警 (ALERT)

仅当支持 SMBus 功能时，才涉及本节内容。请参见*第 47.3 节：I2C 特性实现*。

当 I2C 接口配置为主机 (SMBHEN=1)、使能了报警引脚检测 (ALERTEN=1) 并且在 SMBA 引脚上检测到下降沿时，ALERT 标志将置 1。如果 I2C_CR1 寄存器中的 ERRIE 位置 1，将生成中断。

47.4.16 DMA 请求

使用 DMA 进行发送

将 I2C_CR1 寄存器中的 TXDMAEN 位置 1 可以使能 DMA（直接存储器访问）进行发送。当 TXIS 位置 1 时，数据将从由 DMA 外设配置的 SRAM 区（请参见*第 543 页的第 15 节：直接存储器访问控制器（DMA1、DMA2）*）装载进 I2C_TXDR 寄存器。

只有数据字节采用 DMA 进行传输。

- 在主模式下：初始化、从地址、方向、字节数和起始位均由软件编程（发送的从地址无法通过 DMA 传输）。当所有数据均通过 DMA 传输时，必须在起始位置 1 之前初始化 DMA。传输结束由 NBYTES 计数器来管理。请参见*第 1808 页的主发送器*。
- 在从模式下：
  - 当 NOSTRETCH=0 时，如果所有数据均通过 DMA 传输，则必须在地址匹配事件之前（或清零 ADDR 之前在 ADDR 中断子程序中）初始化 DMA。
  - 当 NOSTRETCH=1 时，必须在地址匹配事件之前初始化 DMA。
- 支持 SMBus 时：PEC 传输由 NBYTES 计数器管理。请参见*第 1822 页的SMBus 从发送器*和*第 1826 页的SMBus 主发送器*。

注：如果使用 DMA 进行发送，则无需使能 TXIE 位。

使用 DMA 进行接收

将 I2C_CR1 寄存器中的 RXDMAEN 位置 1 可以使能 DMA（直接存储器访问）进行接收。当 RXNE 位置 1 时，数据将从 I2C_RXDR 寄存器装载进由 DMA 外设配置的 SRAM 区（请参见*第 543 页的第 15 节：直接存储器访问控制器（DMA1、DMA2）*）。只有数据字节（包括 PEC）采用 DMA 进行传输。

- 在主模式下：初始化、从地址、方向、字节数和起始位均由软件编程。当所有数据均通过 DMA 传输时，必须在起始位置 1 之前初始化 DMA。传输结束由 NBYTES 计数器来管理。
- 在从模式下，当 NOSTRETCH=0 时，如果所有数据均通过 DMA 传输，则必须在地址匹配事件之前（或清零 ADDR 标志之前在 ADDR 中断子程序中）初始化 DMA。
- 如果支持 SMBus（请参见*第 47.3 节：I2C 特性实现*）：PEC 传输由 NBYTES 计数器管理。请参见*第 1824 页的SMBus 从接收器*和*第 1828 页的SMBus 主接收器*。

注：如果使用 DMA 进行接收，则无需使能 RXIE 位。

47.4.17 调试模式

当微控制器进入调试模式时（内核停止），SMBus 超时定时器会根据 DBG 模块中的 DBG_I2Cx_ 配置位选择继续正常工作或者停止工作。

47.5 I2C 低功耗模式

表 372. 低功耗模式

模式	说明
睡眠	对 I2C 通信没有任何影响 I2C 中断可使器件退出睡眠模式。
停止	I2C 寄存器的内容仍被保持。
待机	I2C 外设掉电，退出待机模式后必须重新初始化。

47.6 I2C 中断

在 I2C 中，可以根据下表中描述的事件生成两个中断（i2c_event_it 和 i2c_error_it）和一个唤醒事件信号 (i2c_wkup)：

表 373. I2C 中断请求

中断事件	事件标志	事件标志/ 中断清除方法	中断使能控制位	中断/唤醒是否已激活		
				i2c_event_it	i2c_error_it	i2c_wkup
接收缓冲区非空	RXNE	读取 I2C_RXDR 寄存器	RXIE	是	否	否
发送缓冲区中断状态	TXIS	写入 I2C_TXDR 寄存器	TXIE			
停止位检测中断标志	STOPF	写入 STOPCF=1	STOPIE			
传输完成等待重载	TCR	写入 I2C_CR2 (NBYTES[7:0] ≠ 0)	TCIE			
传输完成	TC	写入 START=1 或 STOP=1				
地址匹配	ADDR	写入 ADDRCF=1	ADDRIE			是(1)
接收到 NACK 应答	NACKF	写入 NACKCF=1	NACKIE			否
总线错误	BERR	写入 BERRCF=1	ERRIE	否	是	
仲裁丢失	ARLO	写入 ARLOCF=1				
上溢/下溢	OVR	写入 OVRCF=1				
PEC 错误	PECERR	写入 PECERRCF=1				
超时/t_{LOW} 错误	TIMEOUT	写入 TIMEOUTCF=1				
SMBus 报警	ALERT	写入 ALERTCF=1				

1. WUPEN 置 1 时。

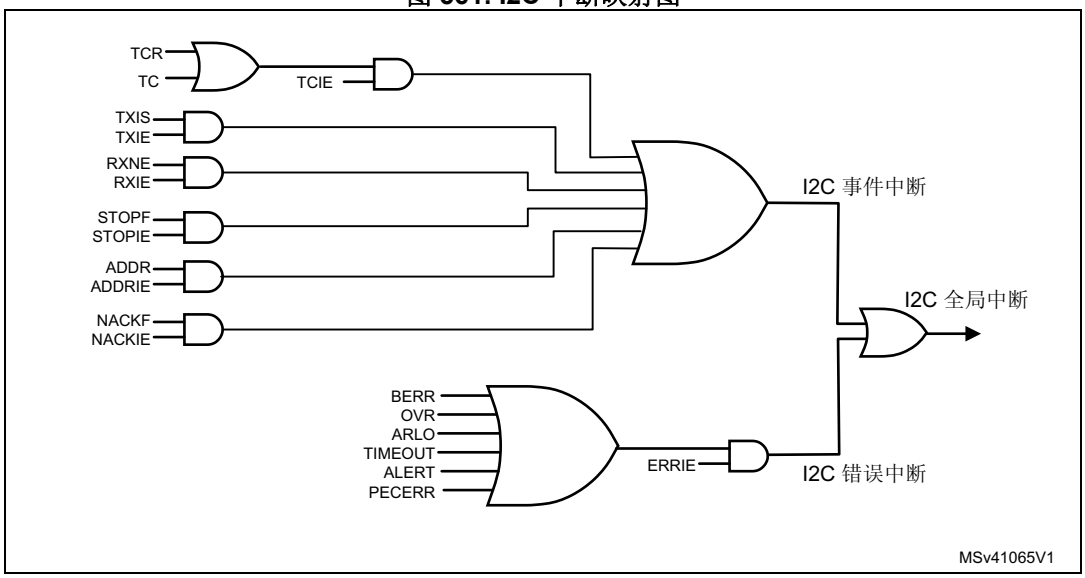
根据产品实现的不同，上述所有中断既可以共享同一个中断向量（I2C 全局中断），也可以分配到 2 个不同的中断向量上（I2C 事件中断和 I2C 错误中断）。有关详细信息，请参见*表 129：NVIC*。

要使能 I2C 中断，需按照以下顺序操作：

1. 配置 NVIC 中的 I2C IRQ 通道并将其使能。
2. 配置 I2C 以生成中断。

I2C 唤醒事件连接到 EXTI 控制器（请参见*第 20 节：扩展中断和事件控制器 (EXTI)*）。

图 561. I2C 中断映射图



47.7 I2C 寄存器

有关寄存器说明中使用的缩写，请参见*第 94 页的第 1.1 节*。

外设寄存器按字（32 位）进行访问。

47.7.1 控制寄存器 1 (I2C_CR1)

Control register 1

偏移地址：0x00

复位值：0x0000 0000

访问：无等待周期，正在对该寄存器执行写访问时出现另一个写访问的情况除外。在这种情况下，会在第二个写访问中插入等待周期，直到前一个写访问完成为止。第二个写访问的延时最长可达 2 x i2c_pclk + 6 x i2c_ker_ck。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PECEN	ALERT EN	SMBD EN	SMBH EN	GCEN	WUPE N	NOSTR ETCH	SBC
								rw	rw	rw	rw	rw		rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RXDMA EN	TXDMA EN	Res.	ANF OFF	DNF				ERRIE	TCIE	STOP IE	NACK IE	ADDR IE	RXIE	TXIE	PE
rw	rw		rw	rw				rw	rw	rw	rw	rw	rw	rw	rw

位 31:24 保留，必须保持复位值。

位 23 **PECEN**：PEC 使能 (PEC enable)

0：禁止 PEC 计算
1：使能 PEC 计算

注： 如果不支持 SMBus 功能，该位保留，并由硬件强制为“0”。请参见第 47.3 节：I2C 特性实现。

位 22 **ALERTEN**：SMBus 报警使能 (SMBus alert enable)

从机模式 (SMBHEN=0)：

0：将 SMBA 引脚释放为高电平并禁止报警响应地址头：0001100x 后跟 NACK。
1：将 SMBA 引脚驱动为低电平并使能报警响应地址头：0001100x 后跟 ACK。

主机模式 (SMBHEN=1)：

0：不支持 SMBus 报警引脚 (SMBA)。
1：支持 SMBus 报警引脚 (SMBA)。

注： 当 ALERTEN=0 时，SMBA 引脚可用作标准 GPIO。
如果不支持 SMBus 功能，该位保留，并由硬件强制为“0”。请参见第 47.3 节：I2C 特性实现。

位 21 **SMBDEN**：SMBus 器件默认地址使能 (SMBus Device Default address enable)

0：禁止器件默认地址。不对地址 0b1100001x 应答。
1：使能器件默认地址。对地址 0b1100001x 应答。

注： 如果不支持 SMBus 功能，该位保留，并由硬件强制为“0”。请参见第 47.3 节：I2C 特性实现。

位 20 **SMBHEN**：SMBus 主机地址使能 (SMBus Host address enable)

0：禁止主机地址。不对地址 0b0001000x 应答。
1：使能主机地址。对地址 0b0001000x 应答。

注： 如果不支持 SMBus 功能，该位保留，并由硬件强制为“0”。请参见第 47.3 节：I2C 特性实现。

位 19 **GCEN**：广播呼叫使能 (General call enable)

0：禁止广播呼叫。不对地址 0b00000000 应答。
1：使能广播呼叫。对地址 0b00000000 应答。

位 18 **WUPEN**：从停止模式唤醒使能 (Wakeup from Stop mode enable)

0：禁止从停止模式唤醒。
1：使能从停止模式唤醒。

注： 如果不支持从停止模式唤醒功能，该位保留并由硬件强制清零。请参见第 47.3 节：I2C 特性实现。

注： 只有当 DNF = “0000”时，WUPEN 才能置 1

位 17 **NOSTRETCH**：时钟延长禁止 (Clock stretching disable)

该位用于在从模式下禁止时钟延长。它在主模式下必须保持清零。

0：使能时钟延长
1：禁止时钟延长

注： 该位只能在 I2C 禁止时 (PE = 0) 编程。

位 16 **SBC**：从设备模式下的字节控制 (Slave byte control)

该位用于在从设备模式下使能硬件字节控制。

0：禁止从设备模式下的字节控制
1：使能从设备模式下的字节控制

位 15 **RXDMAEN**：DMA 接收请求使能 (DMA reception requests enable)

0：禁止 DMA 接收请求
1：使能 DMA 接收请求

位 14 **TXDMAEN**：DMA 发送请求使能 (DMA transmission requests enable)
0：禁止 DMA 发送请求
1：使能 DMA 发送请求

位 13 保留，必须保持复位值。

位 12 **ANFOFF**：模拟噪声滤波器关闭 (Analog noise filter OFF)
0：使能模拟噪声滤波器
1：禁止模拟噪声滤波器
注： 该位只能在 I2C 禁止时 (PE = 0) 编程。

位 11:8 **DNF[3:0]**：数字噪声滤波器 (Digital noise filter)
这些位用于配置 SDA 和 SCL 输入端的数字噪声滤波器。数字滤波器可滤除脉宽 DNF[3:0] * t_{I2CCLK} 以下的尖峰
0000：禁止数字滤波器
0001：使能数字滤波器，可滤除的噪声尖峰脉宽可达 1 t_{I2CCLK}
...
1111：使能数字滤波器，可滤除的噪声尖峰脉宽可达 15 t_{I2CCLK}
注： 如果模拟滤波器也已使能，数字滤波将叠加在模拟滤波之上。
该滤波器只能在 I2C 禁止时 (PE = 0) 编程。

位 7 **ERRIE**：错误中断使能 (Error interrupts enable)
0：禁止错误检测中断
1：使能错误检测中断
注： 以下任一错误均会生成中断：
仲裁丢失 (ARLO)
总线错误检测 (BERR)
上溢/下溢 (OVR)
超时检测 (TIMEOUT)
PEC 错误检测 (PECERR)
报警引脚事件检测 (ALERT)

位 6 **TCIE**：传输完成中断使能 (Transfer complete interrupt enable)
0：禁止传输完成中断
1：使能传输完成中断
注： 以下任一事件均会生成中断：
传输完成 (TC)
传输完成等待重载 (TCR)

位 5 **STOPIE**：停止位检测中断使能 (STOP detection Interrupt enable)
0：禁止停止位检测 (STOPF) 中断
1：使能停止位检测 (STOPF) 中断

位 4 **NACKIE**：接收到否定应答中断使能 (Not acknowledge received Interrupt enable)
0：禁止接收到否定应答 (NACKF) 中断
1：使能接收到否定应答 (NACKF) 中断

位 3 **ADDRIE**：地址匹配中断使能（仅从模式）(Address match Interrupt enable (slave only))
0：禁止地址匹配 (ADDR) 中断
1：使能地址匹配 (ADDR) 中断

位 2 **RXIE**：RX 中断使能 (RX Interrupt enable)
0：禁止接收 (RXNE) 中断
1：使能接收 (RXNE) 中断

位 1 **TXIE**：TX 中断使能 (TX Interrupt enable)

0：禁止发送 (TXIS) 中断

1：使能发送 (TXIS) 中断

位 0 **PE**：外设使能 (Peripheral enable)

0：禁止外设

1：使能外设

注： *当 PE=0 时，将释放 I2C SCL 线和 SDA 线。内部状态机和状态位均恢复为复位值。清零时，PE 必须保持低电平状态至少 3 个 APB 时钟周期。*

47.7.2 控制寄存器 2 (I2C_CR2)

Control register 2

偏移地址：0x04

复位值：0x0000 0000

访问：无等待周期，正在对该寄存器执行写访问时出现另一个写访问的情况除外。在这种情况下，会在第二个写访问中插入等待周期，直到前一个写访问完成为止。第二个写访问的延时最长可达 2 x i2c_pclk + 6 x i2c_ker_ck。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	PEC BYTE	AUTO END	RE LOAD	NBYTES[7:0]							
					rs	rw	rw	rw							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NACK	STOP	START	HEAD 10R	ADD10	RD_ WRN	SADD[9:0]									
rs	rs	rs	rw	rw	rw	rw									

位 31:27 保留，必须保持复位值。

位 26 **PECBYTE**：数据包错误校验字节 (Packet error checking byte)

此位由软件置 1，并可在 PEC 传输完成时、接收到停止位或匹配地址时、或者 PE=0 时由硬件清零。

0：不传输 PEC。

1：请求 PEC 发送/接收。

注： *向该位写入“0”不起作用。*

当 RELOAD 置 1 时，该位不起作用。

当 SBC=0 时，该位在从模式下不起作用。

如果不支持 SMBus 功能，该位保留，并由硬件强制为“0”。请参见第 47.3 节：I2C 特性实现。

位 25 **AUTOEND**：自动结束模式（主模式）(Automatic end mode (master mode))

此位由软件置 1 和清零。

0：软件结束模式：当 NBYTES 数据传输完成时，TC 标志将置 1,SCL 的低电平时间将延长直到相应软件操作结束。

1：自动结束模式：当 NBYTES 数据传输完成时，将自动发送停止位。

注： 在从模式下，或当 RELOAD 位置 1 时，该位不起作用。

位 24 **RELOAD**：NBYTES 重载模式 (NBYTES reload mode)

此位由软件置 1 和清零。

0：传输 NBYTES 数据（后跟停止位或重复起始位）之后即完成传输。

1：传输 NBYTES 数据之后未完成传输（将重载 NBYTES）。当 NBYTES 数据传输完成时，TCR 标志将置 1,SCL 的低电平时间将延长直到相应软件操作结束。

位 23:16 **NBYTES[7:0]**：字节数 (Number of bytes)

在此设置待发送/接收的字节数。在从模式下，当 SBC=0 时，该字段为无关字段。

注： START 位置 1 时，不允许更改这些位。

位 15 **NACK**：NACK 生成（从模式）(NACK generation (slave mode))

此位由软件置 1，并可在发送 NACK 时、接收到停止位或匹配地址时、或者 PE=0 时由硬件清零。

0：在当前接收的字节后发送 ACK。

1：在当前接收的字节后发送 NACK。

注： 向该位写入“0”不起作用。

该位仅在从模式下使用：在主接收器模式下，无论 NACK 位的值为何，最后一个字节（后跟停止位或重复起始位）后都将自动生成 NACK。

当从接收器 NOSTRETCH 模式下发生上溢时，无论 NACK 位的值为何，都将自动生成 NACK。

使能硬件 PEC 校验时 (PECBYTE=1)，PEC 应答值与 NACK 值无关。

位 14 **STOP**：停止位生成（主模式）(Stop generation (master mode))

该位由软件置 1，并可在检测到停止位时或 PE = 0 时由硬件清零。

在主模式下：

0：不生成停止位。

1：在当前字节传输完成后生成停止位。

注： 向该位写入“0”不起作用。

位 13 **START**：起始位生成 (Start generation)

该位由软件置 1，并可在发送起始位（后跟地址序列）之后、发生仲裁丢失时、出现超时错误时、或者 PE = 0 时由硬件清零。它也可由软件清零，方法是向 I2C_ICR 寄存器中的 ADDRCF 位写入“1”。

0：不生成起始位。

1：生成重复起始/起始位：

– 如果 I2C 已处于主模式下且 AUTOEND = 0，则将该位置 1 会在 NBYTES 传输结束后且 RELOAD=0 的情况下生成重复起始位。

– 否则，将该位置 1 会在总线释放后立即生成起始位。

注： 向该位写入“0”不起作用。

即使总线繁忙或 I2C 处于从模式，也可将 START 位置 1。

当 RELOAD 置 1 时，该位不起作用。在 10 位寻址模式下，如果在地址的第一部分接收到 NACK，则 START 位不会由硬件清零，主器件将重新发送地址序列（除非 START 位被软件清零）

位 12 **HEAD10R**：读方向传输时，只发送 10 位地址的前 7 位地址头字节（主接收器模式）(10-bit address header only read direction (master receiver mode))

0：主器件发送完整的 10 位从地址读序列：起始位 + 带写方向的 2 字节 10 位地址 + 重复起始位 + 带读方向的 10 位地址的前 7 位。

1：主器件只发送 10 位地址的前 7 位,后跟读方向。

注： START 位置 1 时，不允许更改此位。

位 11 **ADD10**：10 位寻址模式（主模式）(10-bit addressing mode (master mode))

0：主器件工作在 7 位寻址模式下

1：主器件工作在 10 位寻址模式下

注： START 位置 1 时，不允许更改此位。

位 10 **RD_WRN**：传输方向（主模式）(Transfer direction (master mode))

0：主器件请求写传输。

1：主器件请求读传输。

注： START 位置 1 时，不允许更改此位。

位 9:8 **SADD[9:8]**：从设备地址的第 8 到第 9 位（主模式）(Slave address bit 9:8 (master mode))

在 7 位寻址模式 (ADD10 = 0) 下：

这些位无意义

在 10 位寻址模式 (ADD10 = 1) 下：

这些位应写入待发送从地址的第 8 和第 9 位

注： START 位置 1 时，不允许更改这些位。

位 7:1 **SADD[7:1]**：从地址位 7:1（主模式）(Slave address bit 7:1 (master mode))

在 7 位寻址模式 (ADD10 = 0) 下：

这些位应写入待发送的 7 位从地址

在 10 位寻址模式 (ADD10 = 1) 下：

这些位应写入待发送从地址的第 1 到第 7 位。

注： START 位置 1 时，不允许更改这些位。

位 0 **SADD0**：从地址位 0（主模式）(Slave address bit 0 (master mode))

在 7 位寻址模式 (ADD10 = 0) 下：

该位无意义

在 10 位寻址模式 (ADD10 = 1) 下：

该位应写入待发送从地址的第 0 位

注： START 位置 1 时，不允许更改这些位。

47.7.3 设备自身地址 1 寄存器 (I2C_OAR1)

Own address 1 register

偏移地址：0x08

复位值：0x0000 0000

访问：无等待周期，正在对该寄存器执行写访问时出现另一个写访问的情况除外。在这种情况下，会在第二个写访问中插入等待周期，直到前一个写访问完成为止。第二个写访问的延时最长可达 2 x i2c_pclk + 6 x i2c_ker_ck。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OA1EN	Res.	Res.	Res.	Res.	OA1 MODE	OA1[9:8]		OA1[7:1]							OA1[0]
rw					rw	rw		rw							rw

位 31:16 保留，必须保持复位值。

位 15 **OA1EN**：设备自身地址 1 使能 (Own Address 1 enable)
0：禁止设备自身地址 1。不对接收的从地址 OA1 应答。
1：使能设备自身地址 1。对接收的从地址 OA1 应答。

位 14:11 保留，必须保持复位值。

位 10 **OA1MODE**：设备自身地址 1 10 位模式 (Own Address 1 10-bit mode)
0：设备自身地址 1 为 7 位地址。
1：设备自身地址 1 为 10 位地址。
注： 仅可在 OA1EN=0 时写入该位。

位 9:8 **OA1[9:8]**：接口地址 (Interface address)
7 位寻址模式：无关
10 位寻址模式：地址位 9:8
注： 仅可在 OA1EN=0 时写入这些位。

位 7:1 **OA1[7:1]**：接口地址 (Interface address)
7 位寻址模式：7 位地址
10 位寻址模式：10 位地址的位 7:1
注： 仅可在 OA1EN=0 时写入这些位。

位 0 **OA1[0]**：接口地址 (Interface address)
7 位寻址模式：无关
10 位寻址模式：地址位 0
注： 仅可在 OA1EN=0 时写入该位。

47.7.4 设备自身地址 2 寄存器 (I2C_OAR2)

Own address 2 register

偏移地址：0x0C

复位值：0x0000 0000

访问：无等待周期，正在对该寄存器执行写访问时出现另一个写访问的情况除外。在这种情况下，会在第二个写访问中插入等待周期，直到前一个写访问完成为止。第二个写访问的延时最长可达 2 x i2c_pclk + 6 x i2c_ker_ck。

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OA2EN	Res.	Res.	Res.	Res.	OA2MSK[2:0]			OA2[7:1]							Res.
rw					rw			rw							

位 31:16 保留，必须保持复位值。

位 15 **OA2EN**：设备自身地址 2 使能 (Own Address 2 enable)

0：禁止设备自身地址 2。不对接收的从地址 OA2 应答。

1：使能设备自身地址 2。对接收的从地址 OA2 应答。

位 14:11 保留，必须保持复位值。

位 10:8 **OA2MSK[2:0]**：设备自身地址 2 屏蔽位 (Own Address 2 masks)

000：无屏蔽

001：OA2[1] 被屏蔽，为无关位。仅比较 OA2[7:2]。

010：OA2[2:1] 被屏蔽，为无关位。仅比较 OA2[7:3]。

011：OA2[3:1] 被屏蔽，为无关位。仅比较 OA2[7:4]。

100：OA2[4:1] 被屏蔽，为无关位。仅比较 OA2[7:5]。

101：OA2[5:1] 被屏蔽，为无关位。仅比较 OA2[7:6]。

110：OA2[6:1] 被屏蔽，为无关位。仅比较 OA2[7]。

111：OA2[7:1] 被屏蔽，为无关位。不进行比较，对接收到的全部 7 位地址（保留位除外）应答。

注： *仅可在 OA2EN=0 时写入这些位。*

只要 OA2MSK 不等于 0，即使比较匹配，也不会对保留的 I2C 地址（0b0000xxx 和 0b1111xxx）应答。

位 7:1 **OA2[7:1]**：接口地址 (Interface address)

7 位寻址模式：7 位地址

注： *仅可在 OA2EN=0 时写入这些位。*

位 0 保留，必须保持复位值。

47.7.5 时序寄存器 (I2C_TIMINGR)

Timing register

偏移地址：0x10

复位值：0x0000 0000

访问：无等待周期

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

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

位 31:28 **PRESC[3:0]**：时序预分频因子 (Timing prescaler)

该字段用于对 i2c_ker_ck 进行预分频，以生成用于数据建立和保持计数器（请参见*第 1790 页的I2C 时序*）以及 SCL 高电平和低电平计数器（请参见*第 1804 页的I2C 主模式初始化*）的时钟周期 t_{PRESC}。

t_{PRESC} = (PRESC+1) x t_{I2CCLK}

位 27:24 保留，必须保持复位值。

位 23:20 **SCLDEL[3:0]**：数据建立时间 (Data setup time)

该字段用于在 SDA 边沿和 SCL 上升沿之间生成延时 t_{SCLDEL}。在主模式和从模式下，如果 NOSTRETCH = 0，则 SCL 线的低电平时间将在 t_{SCLDEL} 期间延长。

t_{SCLDEL} = (SCLDEL+1) x t_{PRESC}

注： t_{SCLDEL} 用于生成 $t_{SU:DAT}$ 时序。

位 19:16 **SDADEL[3:0]**：数据保持时间 (Data hold time)

该字段用于在 SCL 下降沿和 SDA 边沿之间生成延时 t_{SDADEL}。在主模式和从模式下，如果 NOSTRETCH = 0，则 SCL 线的低电平时间将在 t_{SDADEL} 期间延长。

t_{SDADEL}= SDADEL x t_{PRESC}

注： SDADEL 用于生成 $t_{HD:DAT}$ 时序。

位 15:8 **SCLH[7:0]**：SCL 高电平周期（主模式）(SCL high period (master mode))

在主模式下，该字段用于生成 SCL 高电平周期。

t_{SCLH} = (SCLH+1) x t_{PRESC}

注： SCLH 还用于生成 $t_{SU:STO}$ 和 $t_{HD:STA}$ 时序。

位 7:0 **SCLL[7:0]**：SCL 低电平周期（主模式）(SCL low period (master mode))

在主模式下，该字段用于生成 SCL 低电平周期。

t_{SCLL} = (SCLL+1) x t_{PRESC}

注： SCLL 还用于生成 t_{BUF} 和 $t_{SU:STA}$ 时序。

注： 该寄存器必须在 I2C 禁止时 (PE = 0) 进行配置。

注： STM32CubeMX 工具进行计算并将 I2C_TIMINGR 内容提供在 I2C 配置窗口中。

47.7.6 超时寄存器 (I2C_TIMEOUTR)

Timeout register

偏移地址：0x14

复位值：0x0000 0000

访问：无等待周期，正在对该寄存器执行写访问时出现另一个写访问的情况除外。在这种情况下，会在第二个写访问中插入等待周期，直到前一个写访问完成为止。第二个写访问的延时最长可达 2 x i2c_pclk + 6 x i2c_ker_ck。

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

位 31 **TEXTEN**：时钟信号延展超时使能 (Extended clock timeout enable)

0：禁止时钟信号延展超时检测

1：使能时钟信号延展超时检测 当 I2C 接口执行 SCL 延展的累积时间超过 $t_{LOW:EXT}$ 时，将检测到超时错误 (TIMEOUT=1)。

位 30:28 保留，必须保持复位值。

位 27:16 **TIMEOUTB[11:0]**：总线超时 B (Bus timeout B)

该字段用于配置累积时钟延展超时：

在主模式下，将检测主器件的累积时钟低电平延展时间 ($t_{LOW:MEXT}$)

在从模式下，将检测从器件的累积时钟低电平延展时间 ($t_{LOW:SEXT}$)

$t_{LOW:EXT}$ = (TIMEOUTB+1) x 2048 x t_{I2CCLK}

注： 仅可在 TEXTEN=0 时写入这些位。

位 15 **TIMOUTEN**：时钟超时使能 (Clock timeout enable)

0：禁止 SCL 超时检测

1：使能 SCL 超时检测：当 SCL 的低电平时间超过 $t_{TIMEOUT}$ (TIDLE=0)，或 SCL 的高电平时间超过 t_{IDLE} (TIDLE=1) 时，将检测到超时错误 (TIMEOUT=1)。

位 14:13 保留，必须保持复位值。

位 12 **TIDLE**：空闲时钟超时检测 (Idle clock timeout detection)

0：TIMEOUTA 用于检测 SCL 低电平超时

1：TIMEOUTA 用于检测 SCL 和 SDA 高电平超时（总线空闲条件）

注： 仅可在 TIMOUTEN=0 时写入该位。

位 11:0 **TIMEOUTA[11:0]**：总线超时 A (Bus Timeout A)

该字段用于配置：

– SCL 低电平超时条件 $t_{TIMEOUT}$（当 TIDLE=0 时）

$t_{TIMEOUT}$ = (TIMEOUTA+1) x 2048 x t_{I2CCLK}

– 总线空闲条件，即 SCL 和 SDA 高电平（当 TIDLE=1 时）

t_{IDLE} = (TIMEOUTA+1) x 4 x t_{I2CCLK}

注： 仅可在 TIMOUTEN=0 时写入这些位。

注： 如果不支持 SMBus 功能，该寄存器保留，并由硬件强制为“0x00000000”。请参见第 47.3 节：I2C 特性实现。

47.7.7 中断和状态寄存器 (I2C_ISR)

Interrupt and status register

偏移地址：0x18

复位值：0x0000 0001

访问：无等待周期

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ADDCODE[6:0]							DIR
								r							r

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BUSY	Res.	ALERT	TIME OUT	PEC ERR	OVR	ARLO	BERR	TCR	TC	STOPF	NACKF	ADDR	RXNE	TXIS	TXE
r		r	r	r	r	r	r	r	r	r	r	r	r	rs	rs

位 31:24 保留，必须保持复位值。

位 23:17 **ADDCODE[6:0]**：地址匹配代码（从模式）(Address match code (Slave mode))
发生地址匹配事件时 (ADDR = 1)，这些位更新为接收到的地址。
在 10 位地址的情况下，ADDCODE 提供 10 位地址的头字节，后跟地址的 2 个 MSB。

位 16 **DIR**：传输方向（从模式）(Transfer direction (Slave mode))
该标志在发生地址匹配事件时 (ADDR=1) 更新。
0：写传输，从器件进入接收器模式。
1：读传输，从器件进入发送器模式。

位 15 **BUSY**：总线繁忙 (Bus busy)
该标志用于指示总线上正在进行通信。当检测到起始位时，该位由硬件置 1。当检测到停止位或 PE = 0 时，该位由硬件清零。

位 14 保留，必须保持复位值。

位 13 **ALERT**：SMBus 报警 (SMBus alert)
当 SMBHEN=1（SMBus 主机配置）、ALERTEN=1 且在 SMBA 引脚上检测到 SMBALERT 事件（下降沿）时，该标志由硬件置 1。该位由软件清零，方法是将 ALERTCF 位置 1。
注： *当 PE=0 时，该位由硬件清零。*
如果不支持 SMBus 功能，该位保留，并由硬件强制为“0”。请参见第 47.3 节：I2C 特性实现。

位 12 **TIMEOUT**：超时或 t_{LOW} 检测标志 (Timeout or t_{LOW} detection flag)
发生超时或延长时钟超时时，该标志由硬件置 1。该位由软件清零，方法是将 TIMEOUTCF 位置 1。
注： *当 PE=0 时，该位由硬件清零。*
如果不支持 SMBus 功能，该位保留，并由硬件强制为“0”。请参见第 47.3 节：I2C 特性实现。

位 11 **PECERR**：接收期间的 PEC 错误 (PEC Error in reception)
当接收到的 PEC 与 PEC 寄存器的内容不匹配时，该标志由硬件置 1。接收到错误的 PEC 后，将自动发送 NACK。该标志由软件清零，方法是将 PECCF 位置 1。
注： *当 PE=0 时，该位由硬件清零。*
如果不支持 SMBus 功能，该位保留，并由硬件强制为“0”。请参见第 47.3 节：I2C 特性实现。

位 10 **OVR**：上溢/下溢（从模式）(Overrun/Underrun (slave mode))

在从模式下且 NOSTRETCH=1 时，如果发生上溢/下溢错误，该标志由硬件置 1。该标志由软件清零，方法是将 OVRCF 位置 1。

注： 当 PE=0 时，该位由硬件清零。

位 9 **ARLO**：仲裁丢失 (Arbitration lost)

发生仲裁丢失时，该标志由硬件置 1。该标志由软件清零，方法是将 ARLOCF 位置 1。

注： 当 PE=0 时，该位由硬件清零。

位 8 **BERR**：总线错误 (Bus error)

当检测到错位的起始位或停止位，而外设也参与传输时，该标志由硬件置 1。在从模式下的地址阶段，该标志不会置 1。该标志由软件清零，方法是将 *BERRCF 位置 1*。

注： 当 PE=0 时，该位由硬件清零。

位 7 **TCR**：传输完成等待重载 (Transfer Complete Reload)

当 RELOAD=1 且 NBYTES 数据传输完成时，该标志由硬件置 1。当 NBYTES 写入一个非零值时，该标志由软件清零。

注： 当 PE=0 时，该位由硬件清零。

该标志单独用于主模式，SBC 位置 1 时单独用于从模式。

位 6 **TC**：传输完成（主模式）(Transfer Complete (master mode))

当 RELOAD=0、AUTOEND=0 且 NBYTES 数据传输完成时，该标志由硬件置 1。当 START 位或 STOP 位置 1 时，该标志由软件清零。

注： 当 PE=0 时，该位由硬件清零。

位 5 **STOPF**：停止位检测标志 (Stop detection flag)

当在总线上检测到停止位，且外设也参与本次传输时，该标志由硬件置 1：

– 外设作为主器件，该位置位的前提是外设已经已经发出停止位。

– 外设作为从器件，该位置位的前提条件是此次传输的寻址对象就是该外设。

该标志由软件清零，方法是将 STOPCF 位置 1。

注： 当 PE=0 时，该位由硬件清零。

位 4 **NACKF**：接收到否定应答标志 (Not Acknowledge received flag)

传输完字节后接收到 NACK 时，该标志由硬件置 1。该标志由软件清零，方法是将 NACKCF 位置 1。

注： 当 PE=0 时，该位由硬件清零。

位 3 **ADDR**：地址匹配（从模式）(Address matched (slave mode))

接收到的地址与使能的从设备地址之一匹配时，该位由硬件置 1。该位由软件清零，方法是将 *ADDRCF 位置 1*。

注： 当 PE=0 时，该位由硬件清零。

位 2 **RXNE**：接收数据寄存器不为空（接收器）(Receive data register not empty (receivers))
当接收到的数据已复制到 I2C_RXDR 寄存器且准备就绪可供读取时，该位由硬件置 1。读取 I2C_RXDR 时，将清零该位。
注：　当 PE=0 时，该位由硬件清零。

位 1 **TXIS**：发送中断状态（发送器）(Transmit interrupt status (transmitters))
当 I2C_TXDR 寄存器为空时，该位由硬件置 1，待发送的数据必须写入 I2C_TXDR 寄存器。下一个待发送的数据写入 I2C_TXDR 寄存器时，该位被清零。
该位只能在 NOSTRETCH=1 时由软件写入“1”，以生成 TXIS 事件（TXIE=1 时为中断，TXDMAEN=1 时为 DMA 请求）。
注：　当 PE=0 时，该位由硬件清零。

位 0 **TXE**：发送数据寄存器为空（发送器）(Transmit data register empty (transmitters))
当 I2C_TXDR 寄存器为空时，该位由硬件置 1。下一个待发送的数据写入 I2C_TXDR 寄存器时，该位被清零。
该位可由软件写入“1”，以刷新发送数据寄存器 I2C_TXDR。
注：　当 PE=0 时，该位由硬件置 1。

47.7.8 中断清零寄存器 (I2C_ICR)

Interrupt clear register

偏移地址：0x1C

复位值：0x0000 0000

访问：无等待周期

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	ALERT CF	TIM OUTCF	PECCF	OVRCF	ARLO CF	BERR CF	Res.	Res.	STOP CF	NACK CF	ADDR CF	Res.	Res.	Res.
		w	w	w	w	w	w			w	w	w			

位 31:14 保留，必须保持复位值。

位 13 **ALERTCF**：报警标志清零 (Alert flag clear)
将 1 写入此位时，I2C_ISR 寄存器中的 ALERT 标志将清零。
注：　如果不支持 SMBus 功能，该位保留，并由硬件强制为“0”。请参见第 47.3 节：I2C 特性实现。

位 12 **TIMOUTCF**：超时检测标志清零 (Timeout detection flag clear)
将 1 写入此位时，I2C_ISR 寄存器中的 TIMEOUT 标志将清零。
注：　如果不支持 SMBus 功能，该位保留，并由硬件强制为“0”。请参见第 47.3 节：I2C 特性实现。

位 11 **PECCF**：PEC 错误标志清零 (PEC Error flag clear)
将 1 写入此位时，I2C_ISR 寄存器中的 PECERR 标志将清零。
注：　如果不支持 SMBus 功能，该位保留，并由硬件强制为“0”。请参见第 47.3 节：I2C 特性实现。

位 10 **OVRCF**：上溢/下溢标志清零 (Overrun/Underrun flag clear)
将 1 写入此位时，I2C_ISR 寄存器中的 OVR 标志将清零。

位 9 **ARLOCF**：仲裁丢失标志清零 (Arbitration Lost flag clear)
将 1 写入此位时，I2C_ISR 寄存器中的 ARLO 标志将清零。

位 8 **BERRCF**：总线错误标志清零 (Bus error flag clear)
将 1 写入此位时，I2C_ISR 寄存器中的 BERRF 标志将清零。

位 7:6 保留，必须保持复位值。

位 5 **STOPCF**：停止位检测标志清零 (Stop detection flag clear)
将 1 写入此位时，I2C_ISR 寄存器中的 STOPF 标志将清零。

位 4 **NACKCF**：否定应答标志清零 (Not Acknowledge flag clear)
将 1 写入此位时，I2C_ISR 寄存器中的 NACKF 标志将清零。

位 3 **ADDRCF**：地址匹配标志清零 (Address Matched flag clear)
将 1 写入此位时，I2C_ISR 寄存器中的 ADDR 标志将清零。将 1 写入此位时，I2C_CR2 寄存器中的 START 位也将清零。

位 2:0 保留，必须保持复位值。

47.7.9 PEC 寄存器 (I2C_PECR)

PEC register

偏移地址：0x20

复位值：0x0000 0000

访问：无等待周期

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

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

位 31:8 保留，必须保持复位值。

位 7:0 **PEC[7:0]**：数据包错误校验寄存器 (Packet error checking register)
当 PECEN=1 时，此字段包含内部 PEC。
当 PE=0 时，PEC 由硬件清零。

注： *如果不支持 SMBus 功能，该寄存器保留，并由硬件强制为“0x00000000”。请参见第 47.3 节: I2C 特性实现。*

47.7.10 接收数据寄存器 (I2C_RXDR)

Receive data register

偏移地址：0x24

复位值：0x0000 0000

访问：无等待周期

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

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

位 31:8 保留，必须保持复位值。

位 7:0 **RXDATA[7:0]**：8 位接收数据 (8-bit receive data)
从 I^2C 总线接收的数据字节。

47.7.11 发送数据寄存器 (I2C_TXDR)

Transmit data register

偏移地址：0x28

复位值：0x0000 0000

访问：无等待周期

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

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

位 31:8 保留，必须保持复位值。

位 7:0 **TXDATA[7:0]**：8 位发送数据 (8-bit transmit data)
待发送到 I^2C 总线的数据字节。
注：　仅可在 TXE=1 时写入这些位。

47.7.12 I2C 寄存器映射

下表提供了 I2C 寄存器映射和复位值。

表 374. I2C 寄存器映射和复位值

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x0	**I2C_CR1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PECEN	ALERTEN	SMBDEN	SMBHEN	GCEN	WUPEN	NOSTRETCH	SBC	RXDMAEN	TXDMAEN	Res.	ANFOFF	DNF[3:0]				ERRIE	TCIE	STOPIE	NACKIE	ADDRIE	RXIE	TXIE	PE
	Reset value									0	0	0	0	0	0	0	0	0	0		0	0	0	0	0	0	0	0	0	0	0	0	0
0x4	**I2C_CR2**	Res.	Res.	Res.	Res.	Res.	PECBYTE	AUTOEND	RELOAD	NBYTES[7:0]								NACK	STOP	START	HEAD10R	ADD10	RD_WRN	SADD[9:0]									
	Reset value						0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x8	**I2C_OAR1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OA1EN	Res.	Res.	Res.	Res.	OA1MODE	OA1[9:0]									
	Reset value																	0					0	0	0	0	0	0	0	0	0	0	0
0xC	**I2C_OAR2**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OA2EN	Res.	Res.	Res.	Res.	OA2MSK [2:0]	OA2[7:1]									Res.
	Reset value																	0					0	0	0	0	0	0	0	0	0	0	
0x10	**I2C_TIMINGR**	PRESC[3:0]				Res.	Res.	Res.	Res.	SCLDEL[3:0]				SDADEL[3:0]				SCLH[7:0]								SCLL[7:0]							
	Reset value	0	0	0	0					0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x14	**I2C_TIMEOUTR**	TEXTEN	Res.	Res.	Res.	TIMEOUTB[11:0]												TIMOUTEN	Res.		TIDLE	TIMEOUTA[11:0]											
	Reset value	0				0	0	0	0	0	0	0	0	0	0	0	0	0			0	0	0	0	0	0	0	0	0	0	0	0	0
0x18	**I2C_ISR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ADDCODE[6:0]							DIR	BUSY	Res.	ALERT	TIMEOUT	PECERR	OVR	ARLO	BERR	TCR	TC	STOPF	NACKF	ADDR	RXNE	TXIS	TXE
	Reset value									0	0	0	0	0	0	0	0	0		0	0	0	0	0	0	0	0	0	0	0	0	0	1
0x1C	**I2C_ICR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ALERTCF	TIMOUTCF	PECCF	OVRCF	ARLOCF	BERRCF	Res.	Res.	STOPCF	NACKCF	ADDRCF	Res.	Res.	Res.
	Reset value																			0	0	0	0	0	0			0	0	0			
0x20	**I2C_PECR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PEC[7:0]							
	Reset value																									0	0	0	0	0	0	0	0
0x24	**I2C_RXDR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RXDATA[7:0]							
	Reset value																									0	0	0	0	0	0	0	0

表 374. I2C 寄存器映射和复位值（续）

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x28	**I2C_TXDR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TXDATA[7:0]							
	Reset value																									0	0	0	0	0	0	0	0

有关寄存器边界地址的信息，请参见*第 100 页的第 2.2.2 节*。

48 通用同步异步收发器 (USART)

本节介绍通用同步/异步收发器 (USART)。

48.1 USART 简介

USART 能够灵活地与外部设备进行全双工数据交换，满足外部设备对工业标准 NRZ 异步串行数据格式的要求。USART 通过小数波特率发生器实现了多种波特率。

USART 不仅支持同步单向通信和半双工单线通信，以及 LIN（局域互连网络）、智能卡协议、IrDA（红外线数据协会）SIR ENDEC 规范和调制解调器操作 (CTS/RTS)，还支持多处理器通信。

通过配置多个缓冲区使用 DMA（直接存储器访问）可实现高速数据通信。

48.2 USART 主要特性

- 全双工异步通信
- NRZ 标准格式（标记/空格）
- 可配置为 16 倍过采样或 8 倍过采样，从而在速度容差与时钟容差之间取得最佳平衡
- 波特率发生器系统
- 两个用于收发数据的内部 FIFO
  每个 FIFO 均可由软件使能/禁止，并且均带有一个状态标志
- 通用可编程收发波特率
- 双时钟域，带有独立于 PCLK 的外设专用内核时钟
- 自动波特率检测
- 数据字长度可编程（7 位、8 位或 9 位）
- 可编程的数据顺序，最先移位 MSB 或 LSB
- 停止位可配置（支持 1 个或 2 个停止位）
- 用于同步通信的同步主/从模式和时钟输出/输入
- SPI 从发送下溢错误标志
- 单线半双工通信
- 使用 DMA 实现连续通信
- 使用中央 DMA 在预留的 SRAM 缓冲区中收/发字节
- 发射器和接收器有单独的使能位
- 发送和接收的单独信号极性控制
- Tx/Rx 引脚配置可交换
- 调制解调器和 RS-485 收发器的硬件流控制
- 通信控制/错误检测标志
- 奇偶校验控制：
  - 发送奇偶校验位
  - 检查接收的数据字节的奇偶性
- 具有标志的中断源
- 多处理器通信：从静默模式唤醒（通过空闲线检测或地址标记检测）

48.3 USART 扩展特性

- LIN 主模式同步中断发送功能和 LIN 从模式中断检测功能
  - 对 USART 进行 LIN 硬件配置时可生成 13 位停止符号和检测 10/11 位停止符号
- 正常模式下支持 3/16 位持续时间的 IrDA SIR 编解码器
- 智能卡模式
  - 对于 ISO/IEC 7816-3 标准中定义的智能卡，支持 T=“0”和 T=“1”异步协议
  - 智能卡工作模式下，支持 0.5 和 1.5 个停止位
- 支持 ModBus 通信
  - 超时功能
  - CR/LF 字符识别

48.4 USART 实现

表 375 介绍了 STM32H7x3 器件上的 USART 实现。

表 375. USART/LPUART 功能

USART 模式/特性(1)	USART1/2/3/6	UART4/5/7/8	LPUART
调制解调器的硬件流控	X	X	X
使用 DMA 进行连续通信	X	X	X
多处理器通信	X	X	X
同步模式（主/从）	X	-	-
智能卡模式	X	-	-
单线半双工通信	X	X	X
IrDA SIR ENDEC 模块	X	X	-
LIN 模式	X	X	-
双时钟域和从低功耗模式唤醒	X	X	X
接收器超时中断	X	X	-
Modbus 通信	X	X	-
自动波特率检测	X	X	-
驱动器使能	X	X	X
USART 数据长度	7 位、8 位和 9 位		
Tx/Rx FIFO	X	X	X
Tx/Rx FIFO 大小	16		

1. X = 支持。

48.5 USART 功能说明

48.5.1 USART 框图

图 562. USART 框图

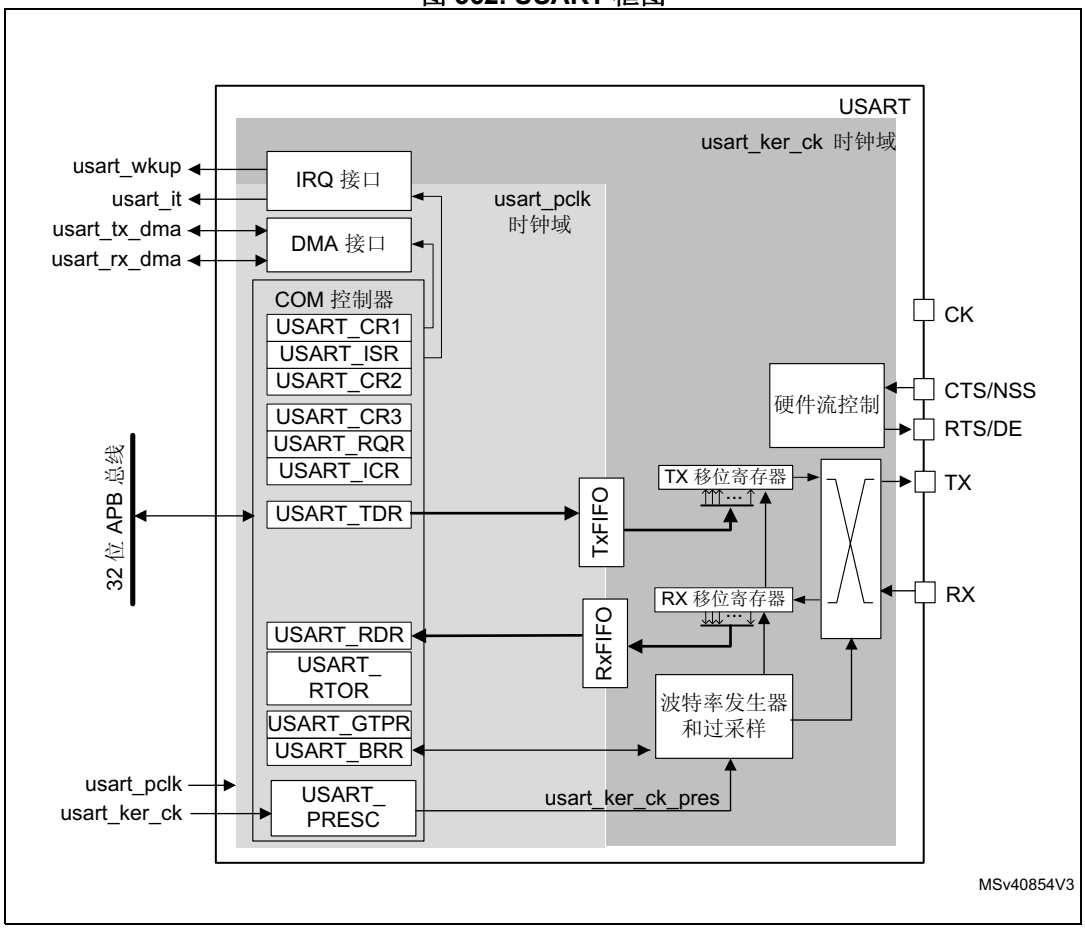


*图 562：USART 框图*中的简化框图显示的是两个完全独立的时钟域：

- **usart_pclk** 时钟域

  usart_pclk 时钟信号馈送外设总线接口。需要访问 USART 寄存器时，该信号必须有效。

- **usart_ker_ck** 内核时钟域。

  usart_ker_ck 是 USART 时钟源。它独立于 **usart_pclk**，由 RCC 提供。因此，即使 **usart_ker_ck** 时钟停止，也可以连续对 USART 寄存器进行读/写操作。

  禁用双时钟域功能时，**usart_ker_ck** 时钟与 **usart_pclk** 时钟相同。

usart_pclk 和 **usart_ker_ck** 之间无任何限制：**usart_ker_ck** 既可快于也可慢于 **usart_pclk**。唯一的限制是软件以足够快的速度管理通信的能力。

USART 工作在 SPI 从器件模式下时，会使用源自外部 SCLK 信号（由外部主 SPI 器件提供）的串行接口时钟来处理数据流。**usart_ker_ck** 时钟的速度必须至少 3 倍于 CK 输入上的时钟。

48.5.2 USART 信号

USART 双向通信

USART 双向通信需要至少两个引脚：接收数据输入引脚 (RX) 和发送数据输出引脚 (TX)：

- **RX**（接收数据输入引脚）

  RX 为串行数据输入引脚。采用过采样技术进行数据恢复，即区分有效输入数据和噪声。

- **TX**（发送数据输出引脚）

  如果关闭发送器，该输出引脚模式由其 I/O 端口配置决定。如果使能了发送器但没有需要发送的数据，则 TX 引脚处于高电平。在单线和智能卡模式下，该 I/O 用于发送和接收数据。

RS232 硬件流控制模式

在 RS232 硬件流控制模式下需要以下引脚：

- **CTS**（清除以发送）

  如果驱动为高电平，则该信号用于在当前传输结束时阻止数据发送。

- **RTS**（请求以发送）

  如果为低电平，则该信号用于指示 USART 已准备好接收数据。

RS485 硬件控制模式

在 RS485 硬件控制模式下需要以下引脚：

- **DE**（驱动器使能）

  该信号用于激活外部收发器的发送模式。

注： *DE 和 RTS 共用同一个引脚。*

同步主/从模式和智能卡模式

在同步主/从模式和智能卡模式下需要以下引脚：

- CK

  该引脚在同步主模式和智能卡模式下用作时钟输出。

  它在同步从模式下用作时钟输入。

  在同步主模式下，该引脚用于输出发送器数据时钟，以便按照 SPI 主器件模式进行同步发送（起始位和结束位上无时钟脉冲，可通过软件向最后一个数据位发送时钟脉冲）。RX 引脚上可同步接收并行数据。该机制可用于控制带移位寄存器的外设（如 LCD 驱动器）。时钟相位和极性可通过软件编程。

  在智能卡模式下，CK 输出向智能卡提供时钟。

- NSS

  该引脚在同步从模式下用作从器件选择输入。

注： *NSS 和 CTS 共用同一个引脚。*

48.5.3 USART 字符说明

可通过对 USART_CR1 寄存器中的 M 位（M0：位 12，M1：位 28）进行编程来将字长设置为 7 位、8 位或 9 位（请参见*图 563*）。

- 7 位字符长度：M[1:0] = “10”
- 8 位字符长度：M[1:0] = “00”
- 9 位字符长度：M[1:0] = “01”

注： *7 位数据长度模式下，不支持智能卡模式、LIN 主模式和自动波特率（0x7F 帧和 0x55 帧检测）。*

在默认情况下，信号（TX 或 RX）在起始位工作期间处于低电平状态。在停止位工作期间处于高电平状态。

通过极性配置控制，可以单独针对每个信号对这些值取反。

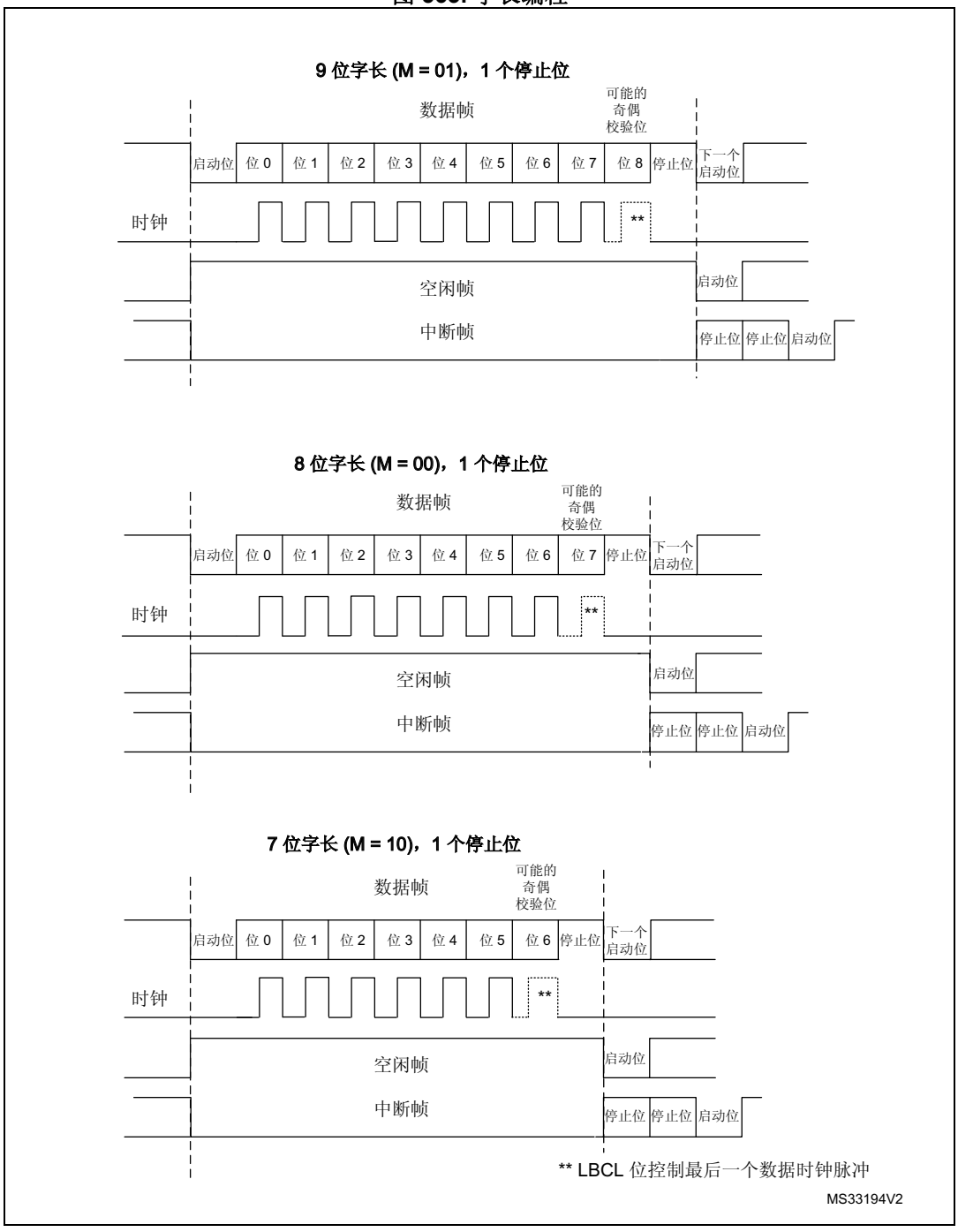
空闲字符可理解为整个帧周期内电平均为“1”（停止位的电平也是“1”）。

停止字符可理解为在一个帧周期内接收到的电平均为“0”。发送器在中断帧的末尾插入 2 个停止位。

发送和接收操作由通用波特率发生器驱动。当发送器和接收器的使能位置 1 时，将分别生成发送时钟和接收时钟。

下面给出了各个块的详细说明。

图 563. 字长编程


9 位字长 (M = 01)，1 个停止位
数据帧
可能的
奇偶
校验位
启动位
位 0
位 1
位 2
位 3
位 4
位 5
位 6
位 7
位 8
停止位
下一个
启动位
时钟
**
空闲帧
启动位
中断帧
停止位
停止位
启动位
8 位字长 (M = 00)，1 个停止位
数据帧
可能的
奇偶
校验位
启动位
位 0
位 1
位 2
位 3
位 4
位 5
位 6
位 7
停止位
下一个
启动位
时钟
**
空闲帧
启动位
中断帧
停止位
停止位
启动位
7 位字长 (M = 10)，1 个停止位
数据帧
可能的
奇偶
校验位
启动位
位 0
位 1
位 2
位 3
位 4
位 5
位 6
停止位
下一个
启动位
时钟
**
空闲帧
启动位
中断帧
停止位
停止位
启动位
** LBCL 位控制最后一个数据时钟脉冲
MS33194V2

48.5.4 USART FIFO 和阈值

USART 可工作在 FIFO 模式下。

USART 具有一个发送 FIFO (TXFIFO) 和一个接收 FIFO (RXFIFO)。可通过将 USART_CR1 寄存器中的 FIFOEN（位 29）置 1 使能 FIFO 模式。仅 UART、SPI 和智能卡模式下支持该模式。

最大数据字长度为 9 位，因此 TXFIFO 为 9 位宽。不过，RXFIFO 的默认宽度为 12 位。这是因为接收器不仅在 FIFO 中存储数据，而且还存储与每个字符相关的错误标志（奇偶校验错误、噪声错误和帧错误标志）。

注： *接收的数据与相应的标志一起存储在 RXFIFO 中，但读取 RDR 时仅读取数据。*

状态标志位于 USART_ISR 寄存器中。

可以配置触发 Tx 和 RX 中断的 TXFIFO 和 RXFIFO 阈值。这些阈值通过 USART_CR3 控制寄存器中的 RXFTCFG 和 TXFTCFG 位域进行编程。

在这种情况下：

- 当 RXFIFO 中接收到的数据量达到 RXFTCFG 位域中编程的阈值时，会生成 Rx 中断。此时，USART_ISR 寄存器中的 RXFT 标志置 1。这表示已接收到 RXFTCFG 数据：USART_RDR 中有 1 个数据，RXFIFO 中有 (RXFTCFG - 1) 个数据。例如，如果将 RXFTCFG 编程为“101”，则在接收到对应于 FIFO 大小的数据量（RXFIFO 中有（FIFO 大小 -1）个数据，USART_RDR 中有 1 个数据）时，RXFT 标志将置 1。因此，下一个接收到的数据不会将上溢标志置 1。
- 当 TXFIFO 中的空位置数达到在 TXFTCFG 位域中编程的阈值时，会生成 Tx 中断。

48.5.5 USART 发送器

发送器可发送 7 位、8 位或 9 位的数据字，具体取决于 M 位的状态。要激活发送器功能，必须将发送使能位 (TE) 置 1。发送移位寄存器中的数据在 TX 引脚输出，相应的时钟脉冲在 SCLK 引脚输出。

字符发送

USART 发送期间，首先通过 TX 引脚移出数据的最低有效位（默认配置）。在该模式下，USART_TDR 寄存器的缓冲区 (TDR) 位于内部总线和发送移位寄存器之间。

使能 FIFO 模式时，写入到发送数据寄存器 (USART_TDR) 中的数据会在 TXFIFO 中排队。

每个字符前面都有一个起始位，其对应于一个位周期的逻辑低电平。字符由可配置数量的停止位终止。

停止位的数量可配置为 0.5、1、1.5 或 2。

注： *向 USART_TDR 中写入要发送的数据前，TE 位必须先置 1。*

数据发送期间不应复位 TE 位。发送期间复位 TE 位会冻结波特率计数器，进而损坏 TX 引脚上的数据。当前发送的数据随即丢失。

使能 TE 位时，将发送空闲帧。

可配置的停止位

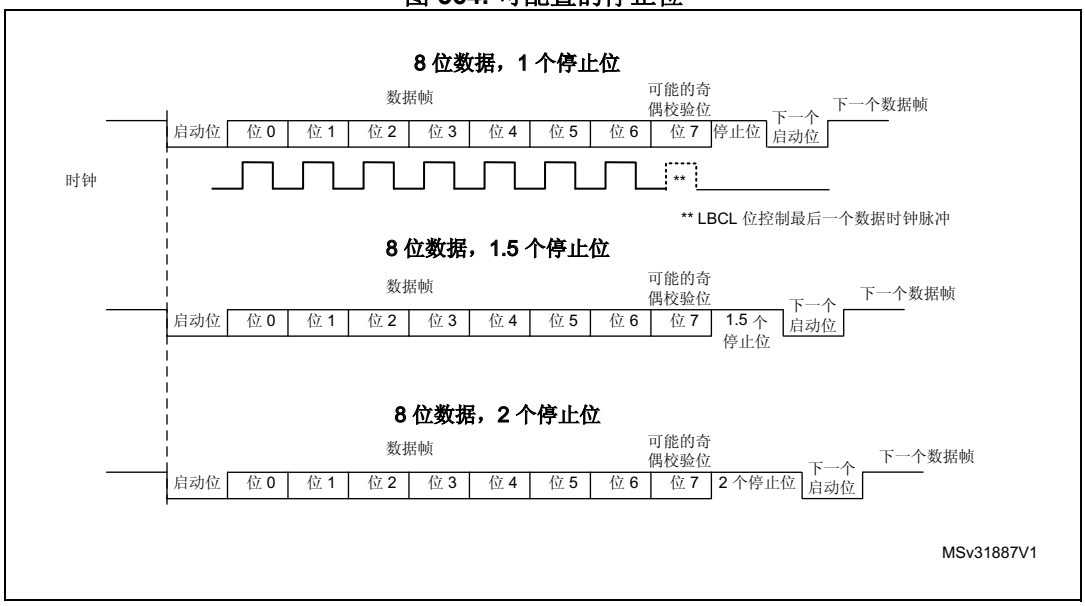
可以在 USART_CR2 的位 13 和位 12 中编程将随各个字符发送的停止位的数量。

- ***1 个停止位***：这是停止位数量的默认值。
- ***2 个停止位***：正常 USART 模式、单线模式和调制解调器模式支持该值。
- ***1.5 个停止位***：用于智能卡模式。

空闲帧发送将包括停止位。

break 发送是 10 个低电平位（M[1:0] = “00”时）、11 个低电平位（M[1:0] = “01”时）或 9 个低电平位（M[1:0] = “10”时），然后是 2 个停止位（请参见*图 564*）。无法传送长 break（break 长度大于 9/10/11 个低电平位）。

图 564. 可配置的停止位



字符发送步骤

要发送字符，需遵循以下步骤：

1. 对 USART_CR1 中的 M 位进行编程以定义字长。
2. 使用 USART_BRR 寄存器选择所需波特率。
3. 对 USART_CR2 中的停止位数量进行编程。
4. 通过向 USART_CR1 寄存器中的 UE 位写入 1 使能 USART。
5. 如果必须进行多缓冲区通信，请选择 USART_CR3 中的 DMA 使能 (DMAT)。按照*第 48.5.10 节：USART 多处理器通信*中的说明配置 DMA 寄存器。
6. 将 USART_CR1 中的 TE 位置 1 以便在首次发送时发送一个空闲帧。
7. 在 USART_TDR 寄存器中写入要发送的数据。为每个要在单缓冲区模式下发送的数据重复这一步骤。
   - 禁止 FIFO 模式时，向 USART_TDR 写入数据会将 TXE 标志清零。
   - 使能 FIFO 模式时，向 USART_TDR 写入数据会为 TXFIFO 增添一个数据。当 TXFNF 标志置 1 时，会对 USART_TDR 执行写操作。该标志会保持置 1，直到 TXFIFO 已满。
8. 将最后一个数据写入 USART_TDR 寄存器后，等待 TC = “1”。
   - 禁止 FIFO 模式时，这表示最后一个帧的发送已完成。
   - 使能 FIFO 模式时，这表示 TXFIFO 和移位寄存器均为空。

   当 USART 被禁止或进入暂停模式时，需要执行此检查来避免最后一次发送的崩溃。

单字节通信

- 禁止 FIFO 模式时

  对发送数据寄存器进行写操作始终会清零 TXE 位。TXE 标志由硬件置 1。它表示：

  – 数据已从 USART_TDR 寄存器移到移位寄存器中且数据发送已开始；

  – USART_TDR 寄存器为空；

  – USART_TDR 寄存器中可写入下一个数据，而不会覆盖前一个数据。

  TXEIE 位置 1 时该标志位会生成中断。

  发送时，要传入 USART_TDR 寄存器中的写指令会将数据存储在 TDR 缓冲区中。该数据随后会在当前发送结束时复制到移位寄存器中。

  未发送时，要传入 USART_TDR 寄存器的写指令会将数据置于移位寄存器中，数据发送开始时，TXE 位置 1。

- 使能 FIFO 模式时，TXFNF（TXFIFO 未满）标志由硬件置 1，以指示：

  – TXFIFO 未满；

  – USART_TDR 寄存器为空；

  – USART_TDR 寄存器中可写入下一个数据，而不会覆盖前一个数据。发送时，对 USART_TDR 寄存器的写操作会将数据存储在 TXFIFO 中。该数据将在当前发送结束时从 TXFIFO 复制到移位寄存器中。

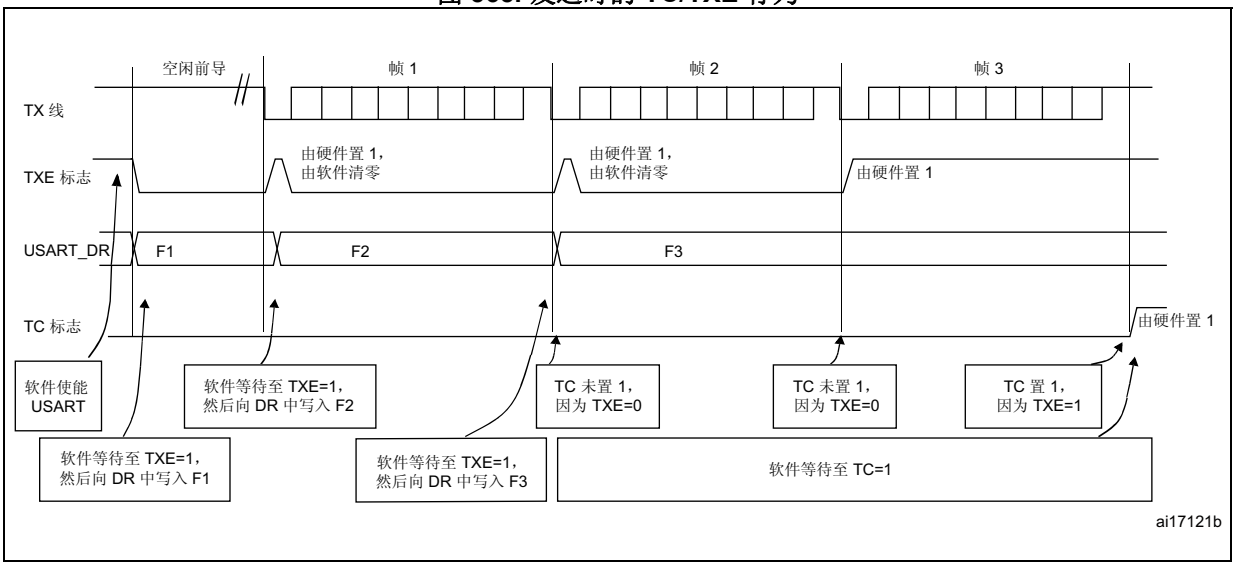
  TXFIFO 未满时，即使在对 USART_TDR 寄存器执行完写操作后，TXFNF 标志也保持为“1”。该标志在 TXFIFO 已满时清零。TXFNFIE 位置 1 时该标志位会生成中断。

  或者，当达到 TXFIFO 阈值时，会生成中断并将数据写入 FIFO。在这种情况下，CPU 可写入由编程的触发阈值定义的数据块。

  如果帧已发送（停止位后）且 TXE 标志（FIFO 模式下的 TXFE）置 1，TC 标志将变为高电平。如果 USART_CR1 寄存器中的 TCIE 位置 1，将生成中断。

向 USART_TDR 寄存器中写入最后一个数据后，必须等待至 TC 置 1，之后才可禁止 USART 或使微控制器进入低功耗模式（请参见*图 565：发送时的 TC/TXE 行为*）。

图 565. 发送时的 TC/TXE 行为



注：　使能 FIFO 管理时，TXFNF 标志将用于数据发送。

break 字符

将 SBKRQ 位置 1 将发送一个 break 字符。break 帧的长度取决于 M 位（请参见*图 563*）。

如果将“1”写入 SBKRQ 位，则当前字符发送完成后，将在 TX 线路上发送一个 break 字符。通过写操作将 SBKF 位置 1 并在 break 字符发送完成时（发送 break 字符后的停止位期间），该位由硬件复位。USART 在中断帧末尾的两位持续时间内插入一个逻辑“1”信号 (STOP)，以确保识别下个帧的起始位。

如果 SBKRQ 位置 1，则在当前发送结束时，会发送一个 break 字符。

使能 FIFO 模式时，即使 TXFIFO 已满，发送 break 字符的优先级也仍高于发送数据的优先级。

空闲字符

将 TE 位置 1 会驱动 USART 在第一个数据帧之前发送一个空闲帧。

48.5.6 USART 接收器

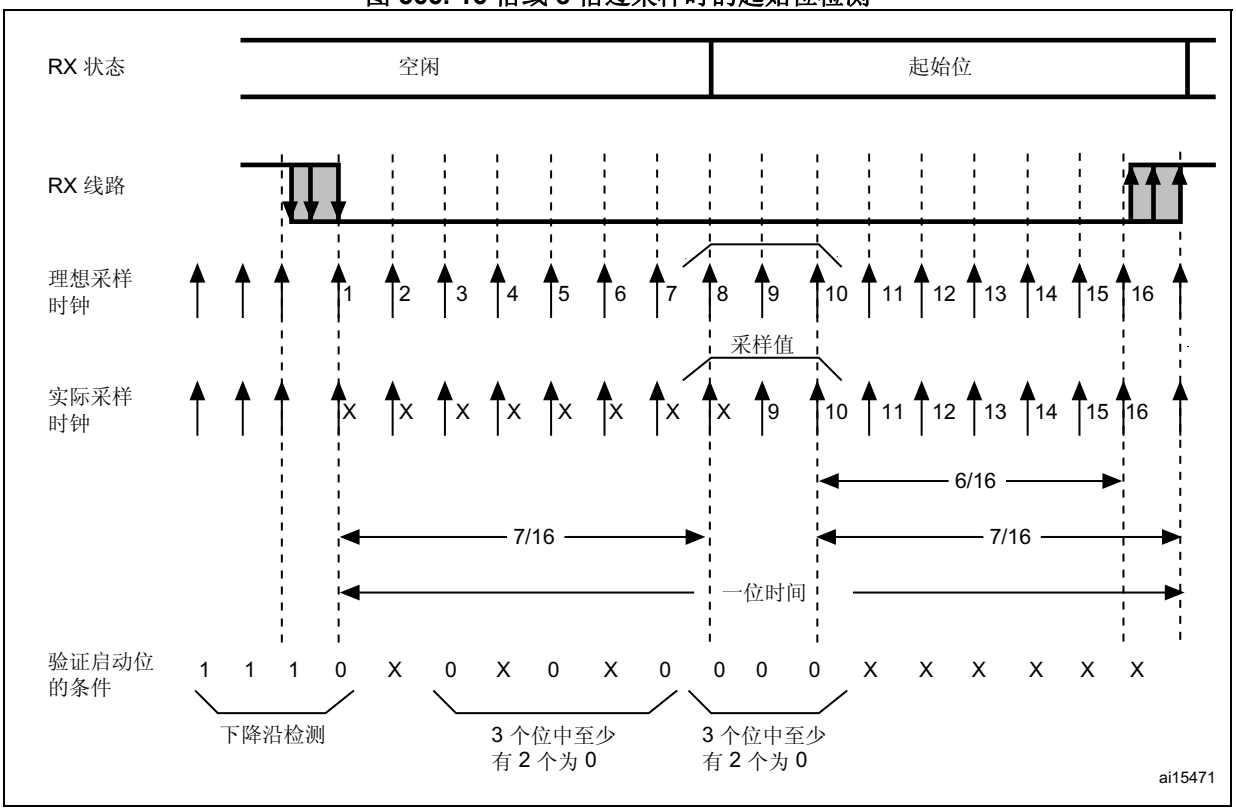
USART 可接收 7 位、8 位或 9 位的数据字，具体取决于 USART_CR1 寄存器中的 M 位。

起始位检测

16 倍或 8 倍过采样时，起始位检测序列相同。

在 USART 中，识别出特定序列的采样时会检测起始位。
此序列为：1 1 1 0 X 0 X 0X 0X 0 X 0X 0。

图 566. 16 倍或 8 倍过采样时的起始位检测



注： *如果序列不完整，起始位检测将中止，接收器将返回空闲状态（无标志位置 1）等待下降沿。*

如果 3 个采样位均为“0”（针对第 3 位、第 5 位和第 7 位进行首次采样时检测到这 3 位均为“0”；针对第 8 位、第 9 位和第 10 位进行第二次采样时仍检测到这 3 位均为“0”），可确认起始位（RXNE 标志置 1 且 RXNEIE=“1”时生成中断；如果使能 FIFO 模式，则 RXFNE 标志置 1 且 RXFNEIE=“1”时生成中断）。

满足以下条件时，可验证起始位但 NE 噪声标志置 1：

a) 对于两次采样，3 个采样位中有 2 位为“0”（针对第 3 位、第 5 位和第 7 位进行采样；针对第 8 位、第 9 位和第 10 位采样）

或

b) 如果其中一次采样时（对第 3 位、第 5 位和第 7 位进行采样或对第 8 位、第 9 位和第 10 位进行采样），3 个采样位中有 2 个为“0”。

如果上述条件均不满足，则启动检测中止，接收器返回空闲状态（无标志置 1）。

字符接收

USART 接收期间，首先通过 RX 引脚移出数据的最低有效位（默认配置）。

字符接收步骤

要接收字符，需遵循以下步骤：

1. 对 USART_CR1 中的 M 位进行编程以定义字长。
2. 使用波特率寄存器 USART_BRR 选择所需波特率。
3. 对 USART_CR2 中的停止位数量进行编程。
4. 通过向 USART_CR1 寄存器中的 UE 位写入“1”使能 USART。
5. 如果将进行多缓冲区通信，请选择 USART_CR3 中的 DMA 使能 (DMAR)。按照*第 48.5.10 节：USART 多处理器通信*中的说明配置 DMA 寄存器。
6. 将 RE 位 USART_CR1 置 1。这一操作将使能接收器开始搜索起始位。

接收到字符时：

- 如果已禁止 FIFO 模式，则 RXNE 位置 1，这表明移位寄存器的内容已传送到 RDR。也就是说，已接收到并可读取数据（及其相应的错误标志）。
- 如果已使能 FIFO 模式，则 RXFNE 位置 1，这表示 RXFIFO 非空。读取 USART_RDR 会返回输入到 RXFIFO 中的最早数据。接收到数据时，数据以及相应的错误位将一起被存储在 RXFIFO 中。
- 如果 RXNEIE（使能 FIFO 模式时为 RXFNEIE）位置 1，则会生成中断。
- 如果接收期间已检测到帧错误、噪声错误、奇偶校验错误或上溢错误，错误标志会置 1。
- 在多缓冲区通信模式下：
  - 如果禁止 FIFO 模式，则 RXNE 标志会在每次接收到字节后置 1。该标志在 DMA 读取接收数据寄存器时被清零。
  - 如果使能 FIFO 模式，则 RXFNE 标志在 RXFIFO 非空时置 1。每次收到 DMA 请求后，都会从 RXFIFO 检索数据。当 RXFIFO 非空时（即，当存在要从 RXFIFO 中读取的数据时），会触发 DMA 请求。
- 在单缓冲区模式下：
  - 如果禁止 FIFO 模式，则通过软件对 USART_RDR 寄存器进行读操作来将 RXNE 标志清零。也可以通过将 USART_RQR 寄存器中的 RXFRQ 位编程为“1”来清零 RXNE 标志。RXNE 标志必须在结束接收下一个字符前清零，以避免发生上溢错误。

– 如果使能 FIFO 模式，则 RXFNE 在 RXFIFO 非空时置 1。每次对 USART_RDR 执行完读操作后，都会从 RXFIFO 中检索数据。RXFIFO 为空时，RXFNE 标志将清零。也可以通过将 USART_RQR 中的 RXFRQ 位编程为“1”来清零 RXFNE 标志。RXFIFO 已满时，必须在结束接收下一个字符前读取 RXFIFO 中的第一个条目，以避免发生上溢错误。当 RXFNEIE 位置 1 时，RXFNE 标志会生成中断。或者，当达到 RXFIFO 阈值时，会生成中断并从 RXFIFO 中读取数据。在这种情况下，CPU 可读取由编程的阈值定义的数据块。

break 字符

接收到 break 字符时，USART 将会按照帧错误对其进行处理。

空闲字符

检测到空闲帧时，除了在 IDLEIE 位置 1 时会生成中断外，处理步骤与接收到数据字符的情况相同。

上溢错误

- 禁止 FIFO 模式

  如果在 RXNE 未清零时接收到字符，则会发生上溢错误。

  RXNE 位清零前，数据无法从移位寄存器传送到 RDR 寄存器。每接收到一个字节后，RXNE 标志都将置 1。

  当 RXNE 标志位是 1 时，如果在接收到下一个数据或尚未处理上一个 DMA 请求时，则会发生上溢错误。发生上溢错误时：

  – ORE 位置 1；
  – RDR 中的内容不会丢失。可通过读取 USART_RDR 寄存器获得之前的数据。
  – 移位寄存器将被覆盖。之后，上溢期间接收到的任何数据都将丢失。
  – 如果 RXNEIE 或 EIE 位置 1，则会生成中断。

- 使能 FIFO 模式

  移位寄存器准备好传送并且接收 FIFO 已满时，会发生上溢错误。

  在 RXFIFO 中出现一个空闲位置之前，数据无法从移位寄存器传送到 USART_RDR 寄存器。当 RXFIFO 非空时，RXFNE 标志置 1。

  如果 RXFIFO 已满且移位寄存器已准备好传送，则会发生上溢错误。发生上溢错误时：

  – ORE 位置 1。
  – RXFIFO 中的第一个条目不会丢失。通过读取 USART_RDR 寄存器可获得此条目。
  – 移位寄存器将被覆盖。之后，上溢期间接收到的任何数据都将丢失。
  – 如果 RXFNEIE 或 EIE 位置 1，则会生成中断。

通过将 USART_ICR 寄存器中的 ORECF 位置 1 来复位 ORE 位。

注： *ORE 位置 1 时表示至少 1 个数据丢失。*

禁止 FIFO 模式时，有以下两种可能

- *如果 RXNE=“1”，则最后一个有效数据存储于接收寄存器 (RDR) 中并且可进行读取；*
- *如果 RXNE=“0”，则最后一个有效数据已被读取，因此 RDR 寄存器中没有要读取的数据。接收到新（丢失）数据的同时已读取 RDR 寄存器中的最后一个有效数据时，会发生该情况。*

选择时钟源和合适的过采样方法

通过时钟控制系统选择时钟源（请参见*第 8 节：复位和时钟控制 (RCC)*）。在使能 USART 之前，必须通过 UE 位选择时钟源。

必须遵循以下两个条件选择时钟源：

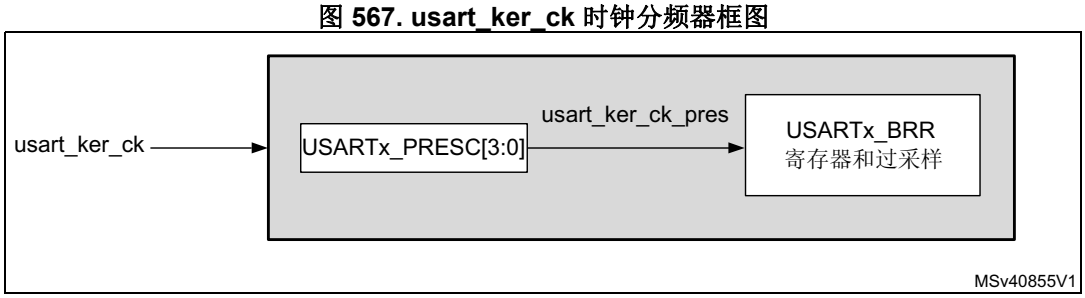
- 可在低功耗模式下使用 USART
- 通信速度。

时钟源频率为 usart_ker_ck。

如果支持双时钟域和从低功耗模式唤醒功能，则 usart_ker_ck 时钟源可在 RCC 中进行配置（请参见*第 8 节：复位和时钟控制 (RCC)*）。否则，usart_ker_ck 时钟与 usart_pclk 相同。

usart_ker_ck 时钟可根据可编程系数（在 USART_PRESC 寄存器中定义）进行分频。

图 567. usart_ker_ck 时钟分频器框图



某些 usart_ker_ck 时钟源允许 USART 在 MCU 处于低功耗模式时接收数据。需要时，USART 可基于所接收的数据和选择的唤醒模式来唤醒 MCU，以通过用软件读取 USART_RDR 寄存器的方式或通过 DMA 的方式传输已接收数据。

对于其他时钟源，系统必须激活才能进行 USART 通信。

时钟源还决定通信速度范围（尤其是最大通信速度）。

接收器采用不同的用户可配置过采样技术（除了同步模式下），可以从噪声中提取有效数据。这可在最大通信速度与抗噪声/时钟误差性能之间实现最佳平衡。

可通过编程 USART_CR1 寄存器中的 OVER8 位来选择采样方法，且采样时钟可以是波特率时钟的 16 倍或 8 倍（请参见*图 568* 和*图 569*）。

根据应用：

- 选择 8 倍过采样（OVER8=“1”）以获得更高的速度（高达 usart_ker_ck_pres/8）。这种情况下接收器对时钟偏差的最大容差将会降低（请参见*第 1867 页的第 48.5.8 节：USART 接收器对时钟偏差的容差*）
- 选择 16 倍过采样（OVER8=“0”）以增加接收器对时钟偏差的容差。在这种情况下，最大速度被限制为最大 usart_ker_ck_pres/16（其中，usart_ker_ck_pres 为 USART 输入时钟通过预分频器进行分频得到的值）。

可通过编程 USART_CR3 寄存器中的 ONEBIT 位选择用于评估逻辑电平的方法。有两种选择可供使用：

- 在已接收位的中心进行三次采样，从而进行多数表决。这种情况下，如果用于多数表决的 3 次采样结果不相等，NE 位置 1。
- 在已接收位的中心进行单次采样。

  根据应用：

  – 在噪声环境下工作时，请选择三次采样的多数表决法（ONEBIT=“0”）；在检测到噪声时请拒绝数据（请参见*图 376*），因为这表示采样过程中产生了干扰。

  – 线路无噪声时请选择单次采样法（ONEBIT=“1”）以增加接收器对时钟偏差的容差（请参见*第 1867 页的第 48.5.8 节：USART 接收器对时钟偏差的容差*）。这种情况下 NE 位始终不会置 1。

帧中检测到噪声时：

- 在 RXNE 位（使能 FIFO 模式时为 RXFNE 位）的上升沿时 NE 位置 1。
- 无效数据从移位寄存器传送到 USART_RDR 寄存器。
- 单字节通信时无中断产生。然而，在 RXNE 位（使能 FIFO 模式时为 RXFNE 位）生成中断时，该位出现上升沿。多缓冲区通信时，USART_CR3 寄存器中的 EIE 位置 1 时将发出中断。

通过设置 ICR 寄存器中的 NFCF 位复位 NE 位。

注： *SPI 模式不支持噪声错误。*

智能卡、IrDA 和 LIN 模式下不可采用 8 倍过采样。在这些模式下，OVER8 位由硬件强制清零。

图 568. 16 倍过采样时的数据采样

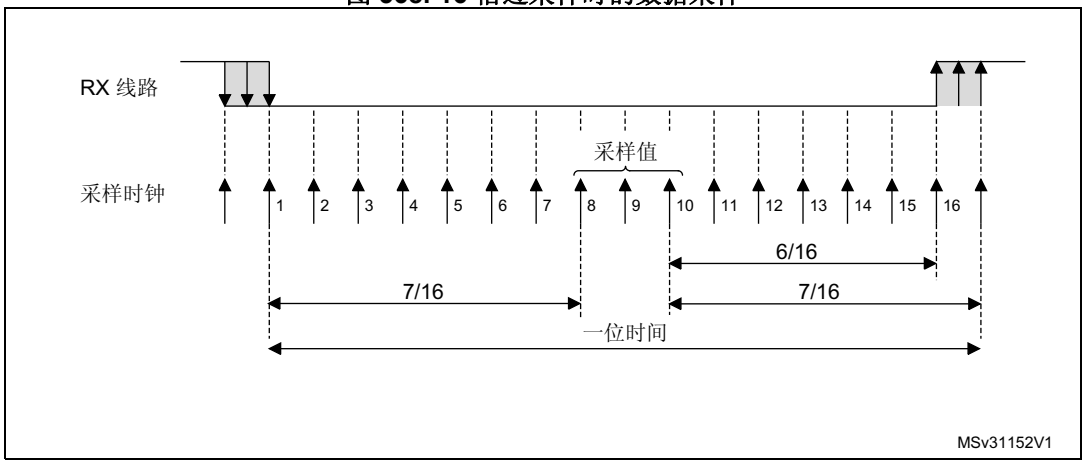


图 569. 8 倍过采样时的数据采样

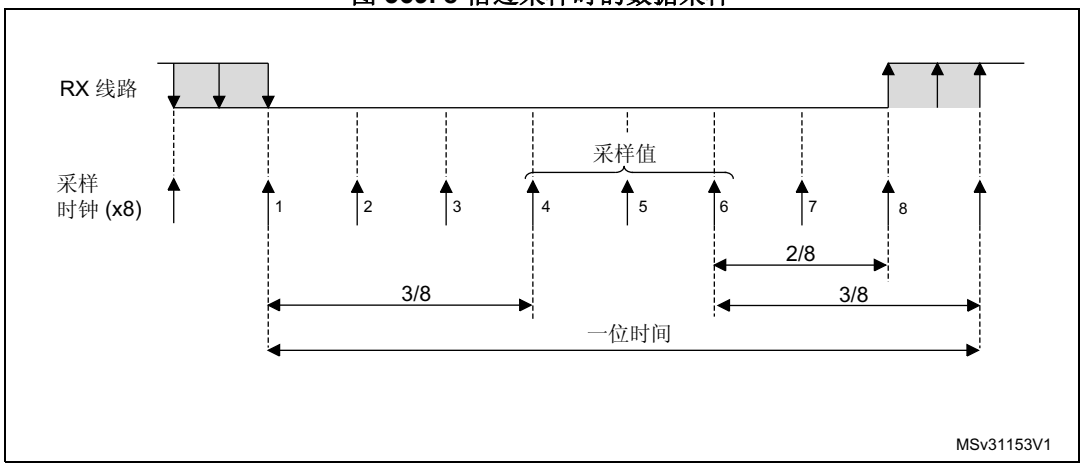


表 376. 通过采样数据进行噪声检测

采样值	NE 状态	接收的位值
000	0	0
001	1	0
010	1	0
011	1	1
100	1	0
101	1	1
110	1	1
111	0	1

帧错误

如果接收数据时未在预期时间内识别出停止位，从而出现同步失效或过度的噪声，则会检测到帧错误。

检测到帧错误时：

- FE 位由硬件置 1；
- 无效数据从移位寄存器传送到 USART_RDR 寄存器（使能 FIFO 模式时为 RXFIFO）。
- 单字节通信时无中断产生。然而，在 RXNE 位（使能 FIFO 模式时为 RXFNE 位）生成中断时，该位出现上升沿。多缓冲区通信时，USART_CR3 寄存器中的 EIE 位置 1 时将发出中断。

通过将“1”写入 USART_ICR 寄存器中的 FECF 位来复位 FE 位。

注： *SPI 模式不支持帧错误。*

接收期间可配置的停止位

可通过 USART_CR 的控制位配置要接收的停止位的数量：可以是 1 或 2 个（正常模式下），也可以是 0.5 或 1.5 个（智能卡模式下）。

- **0.5 个停止位（在智能卡模式下接收时）**：不会对 0.5 个停止位进行采样。结果，选择 0.5 个停止位时，无法检测到帧错误和中断帧。
- **1 个停止位**：将在第 8、第 9 和第 10 次采样时对 1 个停止位进行采样。
- 1.5 个停止位（在智能卡模式下）

  在智能卡模式下发送时，设备必须检查数据是否正确发送。因此，必须使能接收器块（USART_CR1 中的 RE = “1”）并检查停止位，以测试智能卡是否已检测到奇偶校验错误。

  发生奇偶校验错误时，智能卡会在采样时将数据信号强制为低电平（即 NACK 信号），该信号被标记为帧错误。之后，FE 标志在 1.5 个停止位的末尾由 RXNE 标志（使能 FIFO 模式时为 RXFNE）置 1。在第 16、第 17 和第 18 次采样时对 1.5 个停止位进行采样（停止位采样开始后维持 1 个波特时钟周期）。1.5 个停止位可分为 2 个部分：0.5 个波特时钟周期（未发生任何动作），然后是 1 个正常的停止位周期（一半时间处进行采样）（更多详细信息，请参见*第 1878 页的第 48.5.16 节：USART 接收器超时*）。
- 2 个停止位

  采样 2 个停止位时在第 8、第 9 和第 10 次采样时对第一个停止位进行采样。

  如果在第一个停止位期间检测到帧错误，则帧错误标志会置 1。

  发生帧错误时不检测第 2 个停止位。RXNE 标志（使能 FIFO 模式时为 RXFNE）将在第一个停止位结束时置 1。

48.5.7 USART 波特率生成

接收器和发送器（Rx 和 Tx）的波特率均设置为 USART_BRR 寄存器中编程的值。

公式 1：适用于标准 USART（包括 SPI 模式）的波特率（OVER8 = “0”或“1”）

在 16 倍过采样的情况下，波特率通过以下公式得出：

$$\text{Tx/Rx baud} = \frac{\text{usart_ker_ckpres}}{\text{USARTDIV}}$$

在 8 倍过采样的情况下，波特率通过以下公式得出：

$$\text{Tx/Rx baud} = \frac{2 \times \text{usart_ker_ckpres}}{\text{USARTDIV}}$$

公式 2：智能卡、LIN 和 IrDA 模式下的波特率（OVER8 = “0”）

波特率可根据以下公式得出：

$$\text{Tx/Rx baud} = \frac{\text{usart_ker_ckpres}}{\text{USARTDIV}}$$

USARTDIV 是一个存放在 USART_BRR 寄存器中的无符号定点数。

- 当 OVER8 = “0”时，BRR = USARTDIV。
- 当 OVER8 = “1”时
  - BRR[2:0] = USARTDIV[3:0]，右移 1 位。
  - BRR[3] 必须保持清零。
  - BRR[15:4] = USARTDIV[15:4]

注： *对 USART_BRR 执行写操作后，波特率计数器更新为波特率寄存器中的新值。因此，波特率寄存器的值不应在通信时发生更改。*

16 倍和 8 倍过采样时，USARTDIV 必须大于或等于 0d16。

如何从 USART_BRR 寄存器中获取 USARTDIV

示例 1

要通过 usart_ker_ck_pres = 8 MHz 获得 9600 波特：

- 16 倍过采样时：
  USARTDIV = 8 000 000/9600
  BRR = USARTDIV = 833d = 0341h
- 8 倍过采样时：
  USARTDIV = 2 * 8 000 000/9600
  USARTDIV = 1666,66 (1667d = 683h)
  BRR[3:0] = 3h >> 1 = 1h
  BRR = 0x681

示例 2

要通过 usart_ker_ck_pres = 48 MHz 获得 921.6 K 波特：

- 16 倍过采样时：
  USARTDIV = 48 000 000/921 600
  BRR = USARTDIV = 52d = 34h
- 8 倍过采样时：
  USARTDIV = 2 * 48 000 000/921 600
  USARTDIV = 104 (104d = 68h)
  BRR[3:0] = USARTDIV[3:0] >> 1 = 8h >> 1 = 4h
  BRR = 0x64

48.5.8 USART 接收器对时钟偏差的容差

仅当总时钟系统偏差小于 USART 接收器的容差时，USART 异步接收器才能正常工作。

影响总偏差的因素包括：

- DTRA：发送器误差引起的偏差（其中还包括发送器本地振荡器的偏差）
- DQUANT：接收器的波特率量化引起的误差
- DREC：接收器本地振荡器的偏差
- DTCL：传输线路引起的偏差（通常是由于收发器所引起，它可能会在低电平到高电平转换时序与高电平到低电平转换时序之间引入不对称）

DTRA + DQUANT + DREC + DTCL + DWU < USART 接收器的容差

其中

DWU 是使用从低功耗模式唤醒时因采样点偏差而产生的误差。

USART 接收器在*表 377* 和*表 378* 中指定的最大容许偏差下可正确接收数据，具体取决于以下设置：

- 由 USART_CR1 寄存器中的 M 位定义的 9 位、10 位或 11 位字符长度
- 由 USART_CR1 寄存器中的 OVER8 位定义的 8 倍或 16 倍过采样
- USART_BRR 寄存器的 BRR[3:0] 位等于或不等于 0000
- 使用 1 位或 3 位对数据进行采样，取决于 USART_CR3 寄存器中 ONEBIT 位的值

表 377. BRR [3:0] = 0000 时的 USART 接收器容差

M 位	OVER8 位 = “0”		OVER8 位 = “1”	
	ONEBIT= “0”	ONEBIT= “1”	ONEBIT= “0”	ONEBIT= “1”
00	3.75%	4.375%	2.50%	3.75%
01	3.41%	3.97%	2.27%	3.41%
10	4.16	4.86	2.77	4.16

表 378. BRR[3:0] 不等于 0000 时的 USART 接收器容差

M 位	OVER8 位 = “0”		OVER8 位 = “1”	
	ONEBIT= “0”	ONEBIT= “1”	ONEBIT= “0”	ONEBIT= “1”
00	3.33%	3.88%	2%	3%
01	3.03%	3.53%	1.82%	2.73%
10	3.7	4.31	2.22	3.33

注： *当接收的帧恰好包含 10 个（M 位 = “00”）、11 个（M 位 = “01”）或 9 个（M 位 = “10”）位时间的空闲帧时，*表 377 和表 378 中指定的数据可能与特例中的数据略微不同。

48.5.9 USART 自动波特率检测

USART 可根据接收一个字符检测并自动设置 USART_BRR 寄存器的值。自动波特率检测在以下两种情况下非常有用：

- 事先不知道系统的通信速度。
- 系统正在使用精确度相对较低的时钟源且该机制允许在不测量时钟偏差的情况下获得正确的波特率。

时钟源频率必须与预期通信速度兼容。

- 16 倍过采样时，波特率范围为 usart_ker_ck_pres/65535 到 usart_ker_ck_pres/16。
- 8 倍过采样时，波特率范围为 usart_ker_ck_pres/65535 到 usart_ker_ck_pres/8。

在激活自动波特率检测之前，必须通过 USART_CR2 寄存器中的 ABRMOD[1:0] 字段选择自动波特率检测模式。根据不同的字符模式，存在四种检测模式。在这些自动波特率模式下，波特率在同步接收数据期间被多次测量，每次测量的结果都与前一次进行比较。

这些模式如下：

- **模式 0**：以“1”位开头的任意字符。

  这种情况下，USART 会测量起始位的持续时间（下降沿到上升沿）。
- **模式 1**：以 10xx 位模式开头的任意字符。

  这种情况下，USART 会测量起始位和第一个数据位的持续时间。测量在下降沿到下降沿期间完成，可在信号斜率较小时确保较高的精度。
- **模式 2**：0x7F 字符帧（可以是 LSB 在前模式下的 0x7F 字符，也可以是 MSB 在前模式下的 0xFE 字符）。

  这种情况下，先在起始位结束时更新波特率 (BR)，然后在位 6 结束时更新波特率（根据从下降沿到下降沿执行的测量：BR6）。以 BR 对位 0 到位 6 进行采样，而以 BR6 对字符的其它位进行采样。
- **模式 3**：0x55 字符帧。

  这种情况下，先在起始位结束时更新波特率 (BR)，然后在位 0 结束时更新波特率（根据从下降沿到下降沿执行的测量：BR0），最后在位 6 结束时更新波特率 (BR6)。以 BR 对位 0 进行采样，以 BR0 对位 1 到位 6 进行采样，以 BR6 对字符的其它位进行采样。同时，对 RX 线路的各个中间转换执行其他检查。如果 RX 上的转换与接收器（基于根据位 0 计算的波特率的接收器）未充分同步，则生成错误。

激活自动波特率检测之前，必须先通过向 USART_BRR 寄存器写入非零的波特率值来初始化该寄存器。

通过将 USART_CR2 寄存器中的 ABREN 位置 1 来激活自动波特率检测。之后 USART 将等待 RX 线路上的第一个字符。通过将 USART_ISR 寄存器中的 ABRF 标志置 1 来指示自动波特率操作完成。如果线路繁忙，则无法保证正确的波特率检测。这种情况下，BRR 值可能会损坏，ABRE 错误标志位将置 1。如果通信速度不在自动波特率检测范围（位持续时间不在 16 个和 65536 个时钟周期（16 倍过采样时）之间，也不在 8 个和 65536 个时钟周期（8 倍过采样时）之间）内，也会出现这种情况。

稍后，可通过复位 ABRF 标志（通过写入“0”）重新启动自动波特率检测。

禁止 FIFO 管理且发生自动波特率错误时，ABRE 标志通过 RXNE 和 FE 位置 1。

使能 FIFO 管理且发生自动波特率错误时，ABRE 标志通过 RXFNE 和 FE 位置 1。

如果使能 FIFO 模式，则应使用第一个 RXFIFO 位置上的数据进行自动波特率检测。因此，在启动自动波特率检测之前，请通过检查 USART_ISR 寄存器的 RXFNE 标志来确保 RXFIFO 为空。

注： *如果在自动波特率操作期间禁止 USART（UE=“0”），则可能损坏 BRR 值。*

48.5.10 USART 多处理器通信

可以执行 USART 多处理器通信（多个 USART 连接在一个网络中）。例如，其中一个 USART 可以是主 USART，其 TX 输出与其他 USART 的 RX 输入相连；而其他 USART 为从 USART，其各自的 TX 输出在逻辑上通过与运算连在一起，并与主 USART 的 RX 输入相连。

在多处理器配置中，理想情况下通常只有预期的消息接收方主动接收完整的消息内容，从而减少由所有未被寻址的接收器造成的冗余 USART 服务开销。

可通过静默功能将未被寻址的器件置于静默模式下。为了使用静默模式功能，必须将 USART_CR1 寄存器中的 MME 位置 1。

注： *使能 FIFO 管理且 MME 已置 1 时，不得清零 MME 位再快速将其置 1（在两个 usart_ker_ck 周期内），否则静默模式可能保持有效。*

使能静默模式时：

- 不得将接收状态位置 1；
- 禁止任何接收中断；
- USART_ISR 寄存器中的 RWU 位置“1”。在某些情况下，RWU 可以由硬件或软件通过 USART_RQR 寄存器中的 MMRQ 位自动控制。

根据 USART_CR1 寄存器中 WAKE 位的设置，USART 可使用以下两种方法进入或退出静音模式：

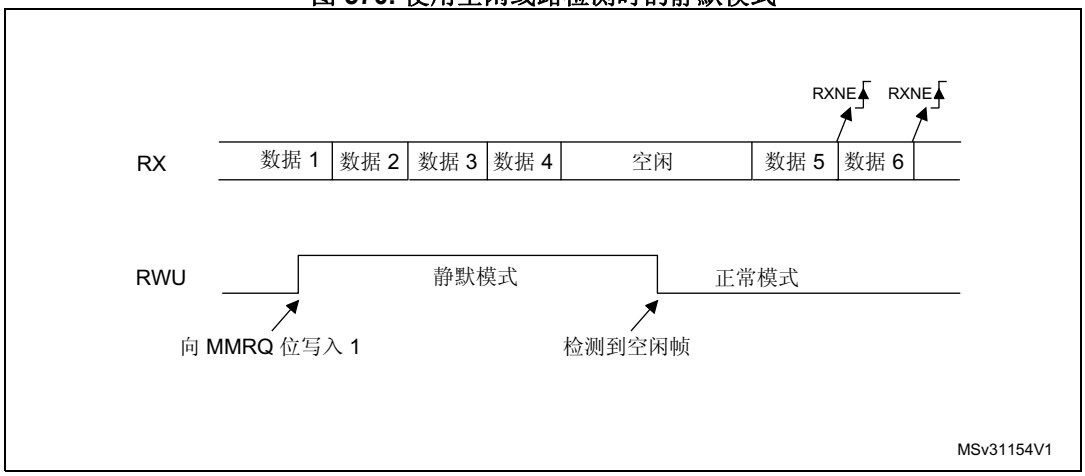
- 如果 WAKE 位被复位，则进行空闲线路检测，
- 如果 WAKE 位置 1，则进行地址标记检测。

空闲线路检测（WAKE=“0”）

向 MMRQ 位写入“1”且 RWU 位自动置 1 时，USART 进入静默模式。

检测到空闲帧时，USART 将唤醒。此时 RWU 位会由硬件清零，但 USART_ISR 寄存器中的 IDLE 位不会置 1。*图 570* 中给出了使用空闲线路检测时静默模式行为的示例。

图 570. 使用空闲线路检测时的静默模式



注： *如果在 IDLE 字符已经过去时将 MMRQ 位置 1，将不会进入静默模式（RWU 未置 1）。*

如果在线路处于空闲状态时激活 USART，在一个 IDLE 帧持续时间后（不只在接收一个字符帧后）会检测到空闲状态。

4 位/7 位地址标记检测（WAKE=“1”）

在此模式下，如果字节的 MSB 为 1，则将这些字节识别为地址，否则将其识别为数据。在地址字节中，目标接收器的地址位于 4 个或 7 个 LSB 中。7 位或 4 位地址检测通过 ADDM7 位来选择。接收器会将此 4 位/7 位字与其地址进行比较，该接收器的地址在 USART_CR2 寄存器的 ADD 位中进行设置。

注： *在 7 位和 9 位数据模式下，地址检测分别在 6 位和 8 位地址上完成（ADD[5:0] 和 ADD[7:0]）。*

当接收到与其编程地址不匹配的地址字符时，USART 会进入静默模式。此时，RWU 位将由硬件置 1。USART 进入静默模式后，RXNE 标志不会针对此地址字节置 1，也不会发出中断或 DMA 请求。使能 FIFO 管理时，软件应确保在进入静默模式之前 RXFIFO 中至少有一个空位置。

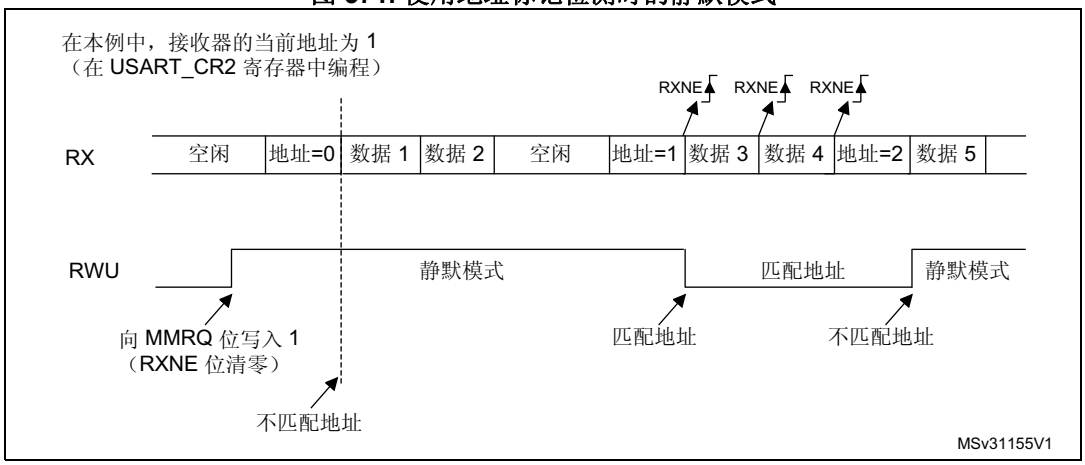
当向 MMRQ 位写入 1 时，USART 也会进入静默模式。这种情况下，RWU 位也自动置 1。

当接收到与编程地址匹配的地址字符时，USART 会退出静默模式。然后 RWU 位被清零，可以开始正常接收后续字节。由于 RWU 位已清零，RXNE/RXFNE 位会针对地址字符置 1。

注： *使能 FIFO 管理时，如果在接收器对数据的最后一位进行采样时 MMRQ 置 1，则可在有效进入静默模式之前接收该数据*

图 571 中给出了使用地址标记检测时静默模式行为的示例。

图 571. 使用地址标记检测时的静默模式



48.5.11 USART Modbus 通信

USART 为 Modbus/RTU 和 Modbus/ASCII 协议的实现提供基本支持。Modbus/RTU 是一个半双工块传输协议。该协议的控制部分（地址识别、块完整性控制和命令解析）必须用软件实现。

USART 为块结束检测提供基本支持，无需软件开销或其它资源。

Modbus/RTU

在此模式下，一个块的结束通过超过 2 个字符时间的“静默”（空闲线路）来识别。此功能通过可编程的超时功能实现。

超时功能和中断必须分别通过 USART_CR2 寄存器中的 RTOEN 位和 USART_CR1 寄存器中的 RTOIE 位激活。与 2 个字符时间（例如 22 个位时间）的超时相对应的值必须在 RTO 寄存器中编程。如果在此期间接收线路空闲，则在接收到最后一个停止位后，将生成中断，同时通知软件当前块接收已完成。

Modbus/ASCII

在此模式下，块结束通过特定 (CR/LF) 字符序列识别。USART 通过字符匹配功能管理此机制。

通过在 ADD[7:0] 字段中编程 LF ASCII 码以及激活字符匹配中断（CMIE=“1”），软件可在接收到 LF 时获得通知并检查 DMA 缓存区中的 CR/LF。

48.5.12 USART 极性控制

将 USART_CR1 寄存器中的 PCE 位置 1，可以使能奇偶校验控制（发送时生成奇偶校验位，接收时进行奇偶校验检查）。根据 M 位定义的帧长度，中列出了可能的 USART 帧格式。

表 379. USART 帧格式

M 位	PCE 位	USART 帧(1)
00	0	| SB | 8 位数据 | STB |
00	1	| SB | 7 位数据 | PB | STB |
01	0	| SB | 9 位数据 | STB |
01	1	| SB | 8 位数据 PB | STB |
10	0	| SB | 7 位数据 | STB |
10	1	| SB | 6 位数据 | PB | STB |

1. 图注：SB：起始位，STB：停止位，P：奇偶校验位。在数据寄存器中，PB 始终位于 MSB 位置（第 8 位或第 7 位，具体取决于 M 位的值）。

偶校验

对奇偶校验位进行计算，使帧和奇偶校验位中“1”的数量为偶数（帧由 6 个、7 个或 8 个 LSB 位组成，具体取决于 M 位的值）。

例如，如果数据= 00110101 且 4 个位置 1，则在选择偶校验（USART_CR1 中的 PS 位 = “0”）时，奇偶校验位为 0。

奇校验

对奇偶校验位进行计算，使帧和奇偶校验位中“1”的数量为奇数（帧由 6 个、7 个或 8 个 LSB 位组成，具体取决于 M 位的值）。

例如，如果数据 = 00110101 且 4 个位置 1，则在选择奇校验（USART_CR1 中的 PS 位 = “1”）时，奇偶校验位为 1。

接收时进行奇偶校验检查

如果奇偶校验检查失败，则 USART_ISR 寄存器中的 PE 标志置 1；如果 USART_CR1 寄存器中 PEIE 位置 1，则会生成中断。通过软件将 1 写入 USART_ICR 寄存器中的 PECF 位来清零 PE 标志。

发送时的奇偶校验生成

如果 USART_CR1 中的 PCE 位置 1，则在数据寄存器中所写入数据的 MSB 位会进行传送，但是会由奇偶校验位进行更改（如果选择偶校验（PS= “0”），则“1”的数量为偶数；如果选择奇校验（PS= “1”），则“1”的数量为奇数）。

48.5.13 USART LIN（局域互连网络）模式

仅在支持 LIN 模式时才与本节相关。请参见*第 1851 页的第 48.4 节：USART 实现*。

通过将 USART_CR2 寄存器中的 LINEN 位置 1 来选择 LIN 模式。在 LIN 模式下，必须将以下位清零：

- USART_CR2 寄存器中的 CLKEN 位，
- USART_CR3 寄存器中的 STOP[1:0]、SCEN、HDSEL 和 IREN 位。

LIN 发送

与正常的 USART 发送相比，在 LIN 主器件中发送时必须采用*第 48.5.4 节*中介绍的步骤，同时还具有以下区别：

- M 位清零以配置 8 位字长度。
- LINEN 位置 1 以进入 LIN 模式。此时，将 SBKRQ 位置 1 会发送 13 个“0”位作为 break 字符。然后会发送值为“1”的 2 个位以进行下一启动检测。

LIN 接收

使能 LIN 模式后，将激活中断检测电路。该检测完全独立于正常的 USART 接收器。在空闲状态或某个帧期间，只要发生中断即可检测出来。

接收器（USART_CR1 中的 RE= “1”）使能后，电路便开始监测启动信号的 RX 输入。检测起始位的方法与搜索 break 字符或数据的方法相同。检测到起始位后，电路会对接下来的位进行采样，方法与数据采样相同（第 8、第 9 和第 10 次采样）。如果 10 个（USART_CR2 中的 LBDL= “0”时）或 11 个（USART_CR2 中的 LBDL= “1”时）连续位均检测为“0”，且其后跟分隔符，则 USART_ISR 中的 LBDF 标志将置 1。如果 LBDIE 位 = “1”，则会生成中断。在验证中断前，会对分隔符进行检查，因为它表示 RX 线路已恢复到高电平。

如果在第 10 或第 11 次采样前已对“1”采样，则中断检测电路会取消当前检测，并重新搜索起始位。

如果禁止 LIN 模式（LINEN= “0”），接收器会作为正常的 USART 继续工作，不会再进行中断检测。

如果使能 LIN 模式 (LINEN=1)，只要发生帧错误（例如，在“0”处检测到停止位，这种情况可能出现在任何中断帧中），接收器即会停止，直到中断检测电路接收到“1”（中断字不完整时）或接收到分隔符（检测到中断时）为止。

*第 1873 页的图 572：LIN 模式下的中断检测（11 位中断长度——LBDL 位置 1）*中显示了中断检测器状态机和中断标志的行为。

*第 1874 页的图 573：LIN 模式下的中断检测与帧错误检测*中列出了中断帧的示例。

图 572. LIN 模式下的中断检测（11 位中断长度——LBDL 位置 1）

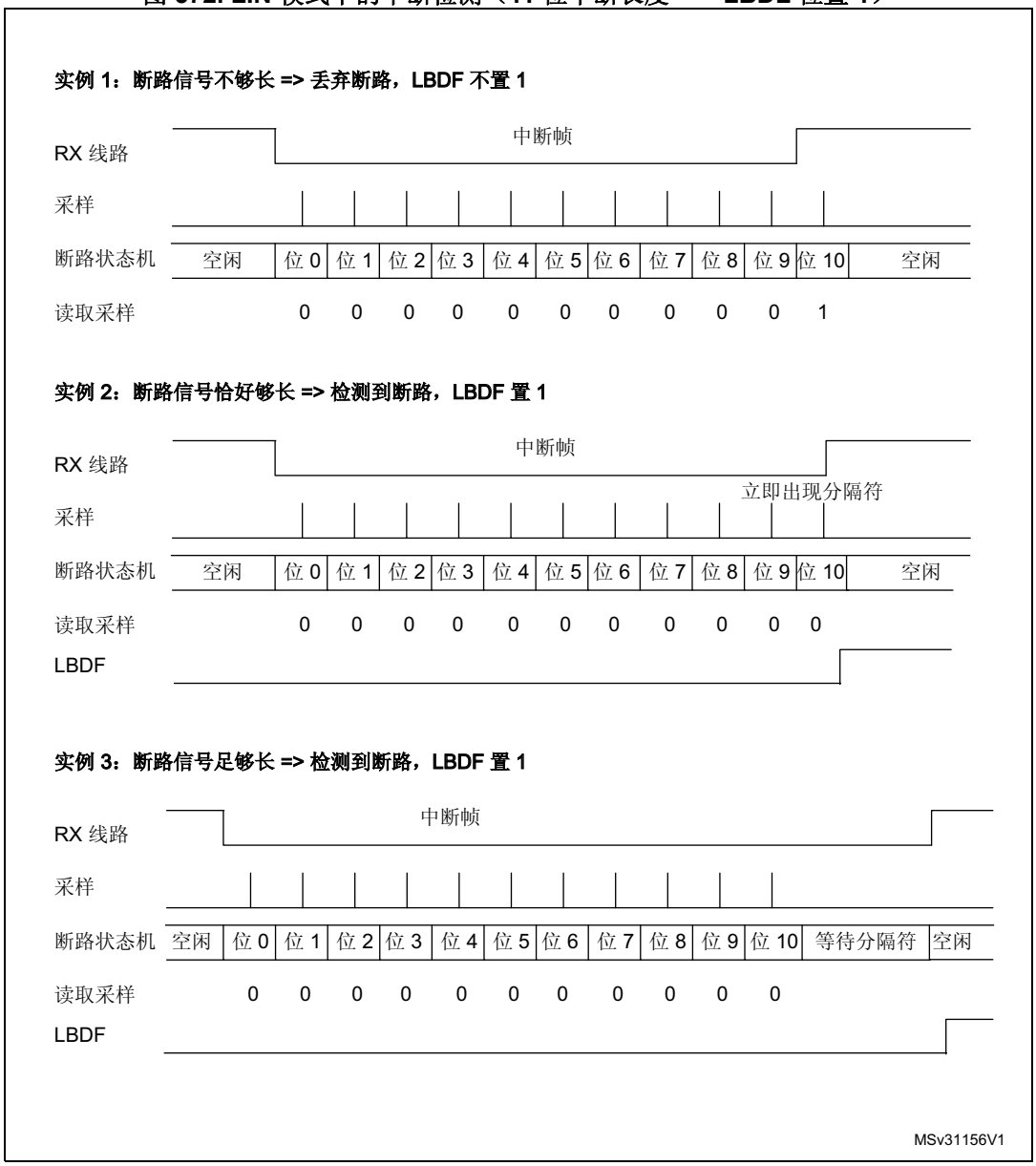
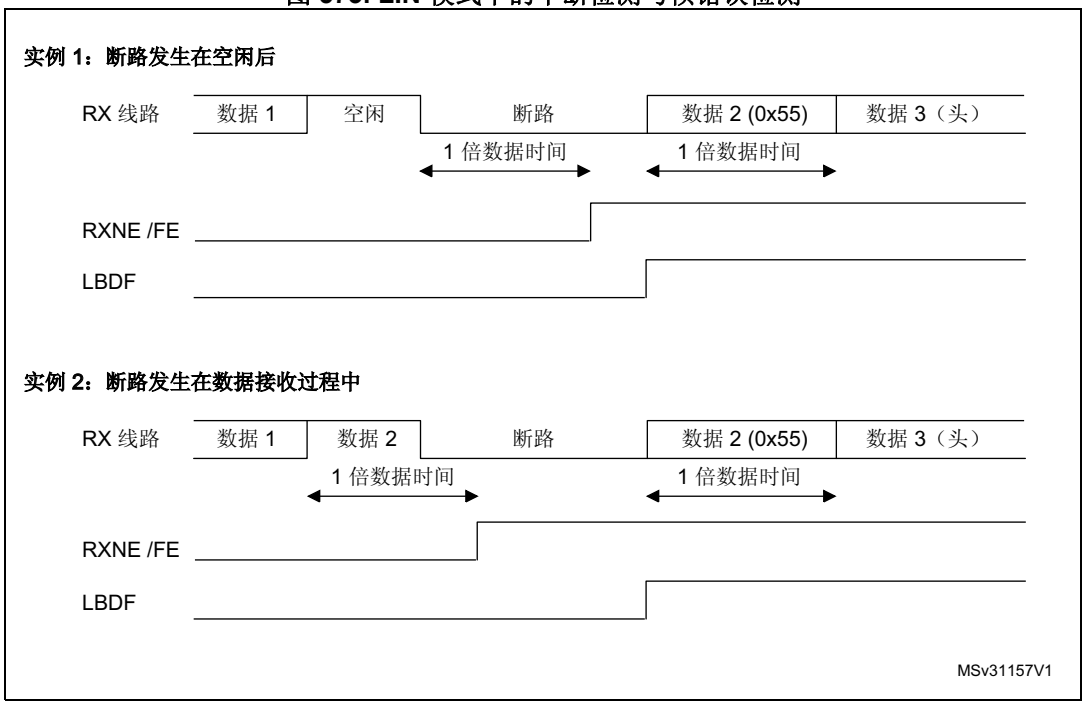


图 573. LIN 模式下的中断检测与帧错误检测



48.5.14 USART 同步模式

主模式

通过将 USART_CR2 寄存器中的 CLKEN 位编程为“1”来选择同步主模式。在同步模式下，必须将以下位清零：

- USART_CR2 寄存器中的 LINEN 位，
- USART_CR3 寄存器中的 SCEN、HDSEL 和 IREN 位。

在此模式下，USART 可用于在主模式下控制双向同步串行通信。SCLK 引脚是 USART 发送器时钟的输出。在起始位或停止位期间，不会向 SCLK 引脚发送时钟脉冲。在最后一个有效数据位（地址标记）期间，会（也可能不会）生成时钟脉冲，这取决于 USART_CR2 寄存器中 LBCL 位的状态。USART_CR2 寄存器中的 CPOL 位用于选择时钟极性；USART_CR2 寄存器中的 CPHA 位用于选择外部时钟的相位（请参见*图 574*、*图 575* 和*图 576*）。

在空闲状态、报头模式和发送中断期间，外部 SCLK 时钟处于未激活状态。

在同步主模式下，USART 发送器的工作方式与异步模式下完全相同。但是，由于 SCLK 与 TX 同步（根据 CPOL 和 CPHA），因此 TX 上的数据是同步的。

在同步主模式下，USART 接收器的工作方式与异步模式下不同。如果 RE 置“1”，则数据在 SCLK 上采样（上升或下降沿，具体取决于 CPOL 和 CPHA），而不会进行任何过采样。此时必须确保给定建立时间和保持时间（取决于波特率：1/16 位时间）。

注： *在主模式下，SCLK 引脚可与 TX 引脚结合使用。因此，仅当使能发送器（TE=“1”）且正在发送数据时（USART_TDR 数据寄存器已写入），才会提供时钟。这意味着，没有发送数据的情况下无法接收同步数据。*

图 574. USART 同步主发送示例

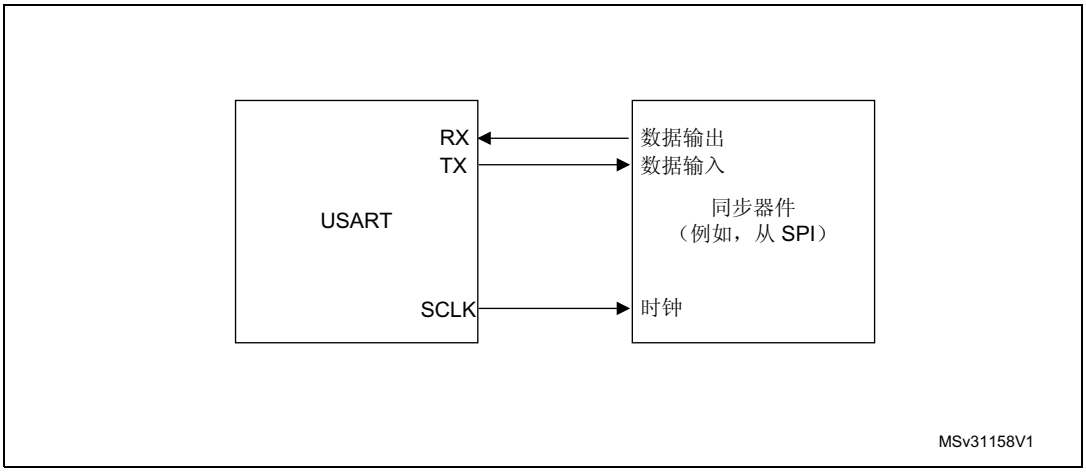


图 575. USART 在同步主模式下的数据时钟时序图（M 位 =“00”）

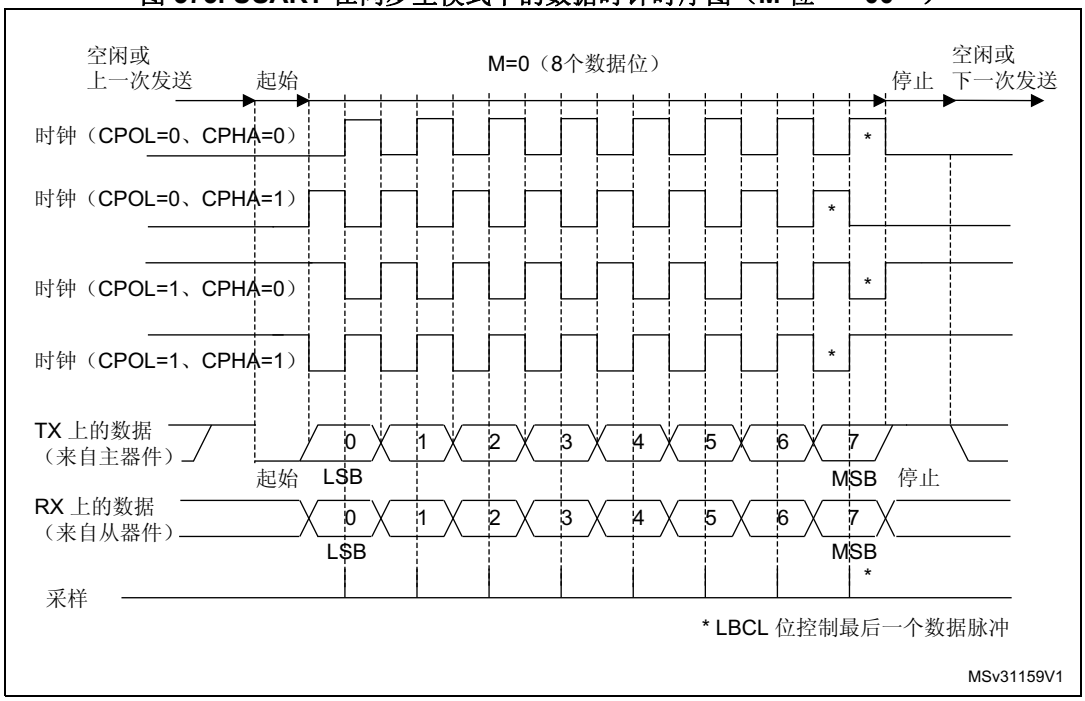
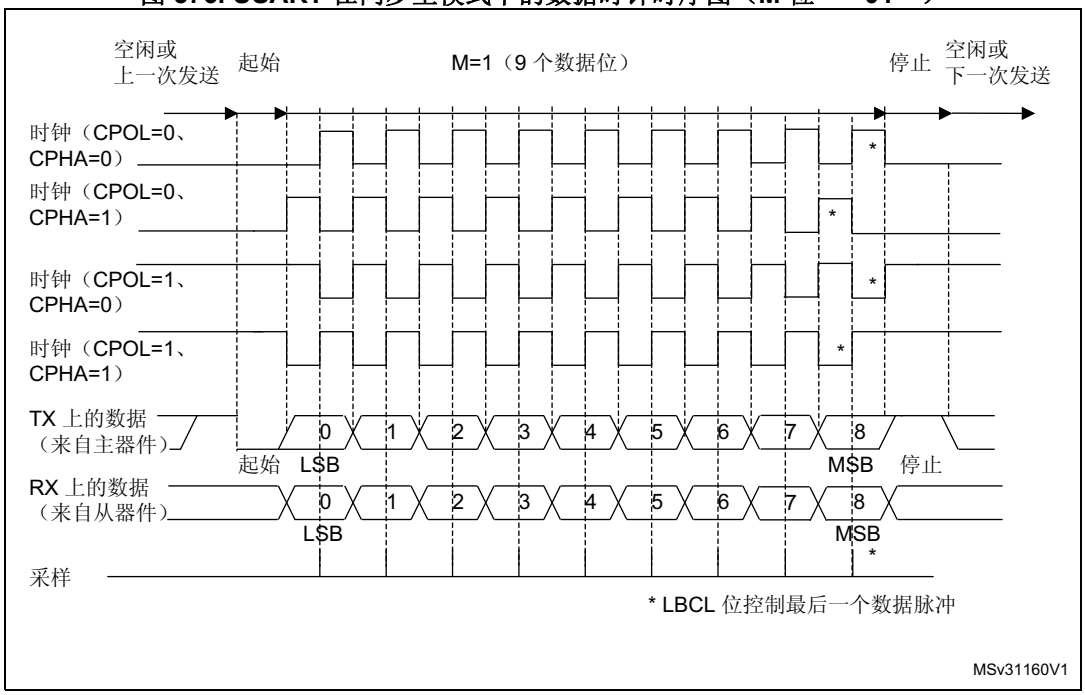


图 576. USART 在同步主模式下的数据时钟时序图（M 位 =“01”）



从模式

通过将 USART_CR2 寄存器中的 SLVEN 位编程为“1”来选择同步从模式。在同步从模式下，必须将以下位清零：

- USART_CR2 寄存器中的 LINEN 和 CLKEN 位，
- USART_CR3 寄存器中的 SCEN、HDSEL 和 IREN 位。

在此模式下，USART 可用于在从模式下控制双向同步串行通信。在从模式下，SCLK 引脚是 USART 的输入。

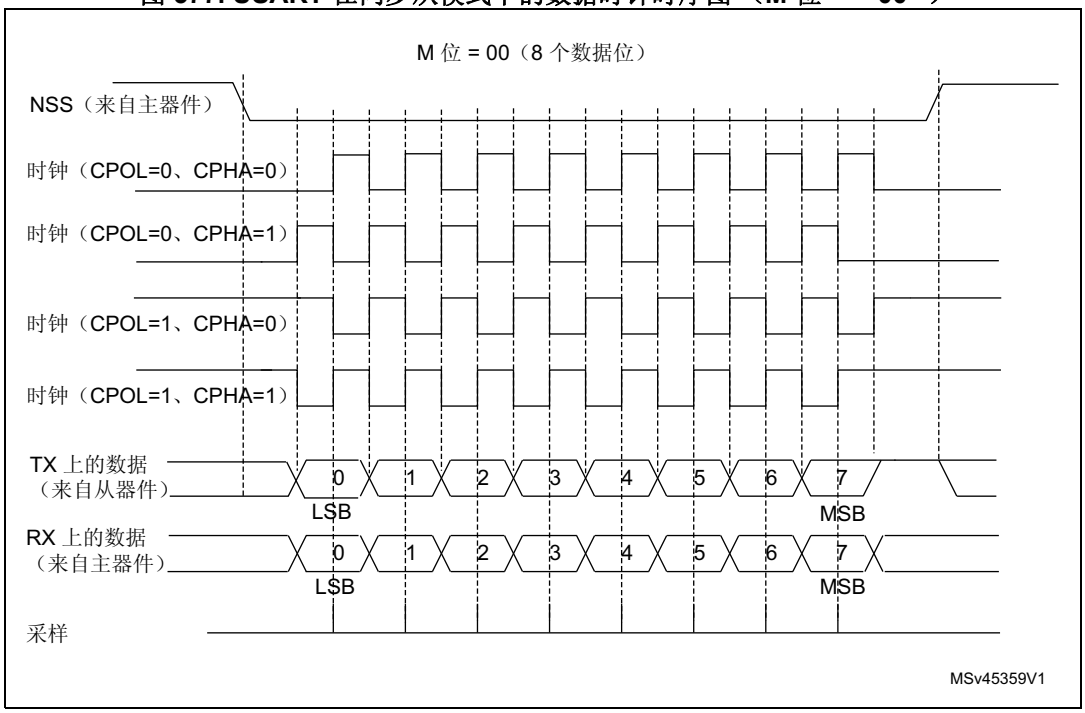
注： *当外设用于 SPI 从器件模式时，外设时钟源 (usart_ker_ck_pres) 的频率必须是 CK 输入频率的 3 倍以上。*

USART_CR2 寄存器中的 CPOL 位和 CPHA 位分别用于选择时钟极性和外部时钟的相位（请参见*图 575* 和*图 576*）。

从发送模式支持下溢错误标志。如果在软件尚未将任何值加载到 USART_TDR 之前出现第一个数据发送时钟脉冲，则此标志将置 1。

从器件支持硬件和软件 NSS 管理。

图 577. USART 在同步从模式下的数据时钟时序图 （M 位 = “00”）



从器件选择 (NSS) 引脚管理

可通过 USART_CR2 寄存器中的 DIS_NSS 位设置硬件或软件从器件选择管理：

- 软件 NSS 管理（DIS_NSS = “1”）

  将始终选择 SPI 从器件并忽略 NSS 输入引脚。

  外部 NSS 引脚空闲，可供其它应用使用。

- 硬件 NSS 管理（DIS_NSS = “0”）

  SPI 从器件选择取决于 NSS 输入引脚。NSS 为低电平时选择从器件，NSS 为高电平时取消选择从器件。

注： *禁止 USART 时（UE= “0”），必须选择 LBCL（仅用于 SPI 主器件模式）、CPOL 和 CPHA 位以确保时钟脉冲正常工作。*

在 SPI 从器件模式下，必须在启动主器件通信之前（或时钟稳定时在相邻帧之间）使能 USART。否则，如果 USART 从器件在主器件处于某个帧中间时使能，则它将与主器件失去同步。在通信时钟的第一个边沿到来之前或者正在进行的通信结束之前，从器件的数据寄存器就需要准备就绪，否则 SPI 从器件会发送零。

SPI 从器件下溢错误

发生下溢错误时，USART_ISR 寄存器中的 UDR 标志会置 1，SPI 从器件继续发送最后一个数据，直到下溢错误标志由软件清零。

下溢标志在帧开始时置 1。如果 USART_CR3 寄存器中的 EIE 位置 1，则会触发下溢错误中断。

通过将 USART_ICR 寄存器中的位 UDRCF 置 1 来清零下溢错误标志。

发生下溢错误时，仍然可以对 TDR 寄存器进行写操作。清除下溢错误将允许发送新数据。

如果发生下溢错误且没有新数据被写入 TDR 中，则 TC 标志在帧结束时置 1。

注： *如果向 USART_TDR 写入数据的时间点过于接近第一个 SCLK 发送边沿，则可能会发生下溢错误。为避免发生此下溢错误，应在第一个 SCLK 边沿的 3 个 usart_ker_ck 周期之前对 USART_TDR 进行写操作。*

48.5.15 USART 单线半双工通信

通过将 USART_CR3 寄存器中的 HDSEL 位置 1 来选择单线半双工模式。在此模式下，必须将以下位清零：

- USART_CR2 寄存器中的 LINEN 和 CLKEN 位，
- USART_CR3 寄存器中的 SCEN 和 IREN 位。

USART 可以配置为遵循单线半双工协议，其中 TX 和 RX 线路从内部相连接。使用 USART_CR3 寄存器中的控制位 HDSEL 可在半双工通信和全双工通信间进行选择。

向 HDSEL 位写入“1”后：

- TX 和 RX 线路从内部相连接。
- 不能再使用 RX 引脚。
- 无数据传输时，TX 引脚始终处于释放状态。因此，它在空闲状态或接收过程中用作标准 I/O。这意味着，必须对 I/O 进行配置，以便将 TX 配置为复用功能开漏并外接上拉电阻。

除此之外，通信协议与正常 USART 模式下的通信协议相似。此线路上的任何冲突都必须由软件管理（例如，使用中央仲裁器）。尤其要注意，发送过程永远不会被硬件阻塞，只要数据是在 TE 位置 1 的情况下写入，发送就会持续进行。

48.5.16 USART 接收器超时

接收器超时功能可通过将 USART_CR2 控制寄存器中的 RTOEN 位置 1 来使能。

超时间隔通过 USART_RTOR 寄存器中的 RTO 位域进行编程。

接收器超时计数器遵循以下规则开始计数：

- STOP =“00”或 STOP =“11”时从停止位结束时开始计数。
- STOP =“10”时从第二个停止位结束时开始计数。
- STOP =“01”时从停止位开始时开始计数。

经过超时间隔后，USART_ISR 寄存器中的 RTOF 标志置 1。如果 USART_CR1 寄存器中的 RTOIE 位置 1，则会产生超时。

48.5.17 USART 智能卡模式

仅在支持智能卡模式时才涉及本节内容。请参见*第 1851 页的第 48.4 节：USART 实现*。

通过将 USART_CR3 寄存器中的 SCEN 位置 1 选择智能卡模式。在智能卡模式下，必须将以下位清零：

- USART_CR2 寄存器中的 LINEN 位，
- USART_CR3 寄存器中的 HDSEL 和 IREN 位。

此外，还可将 CLKEN 位置 1，以便为智能卡提供时钟。

智能卡接口支持符合 ISO 7816-3 标准的异步智能卡协议。同时支持 T=“0”（字符模式）和 T=“1”（块模式）。

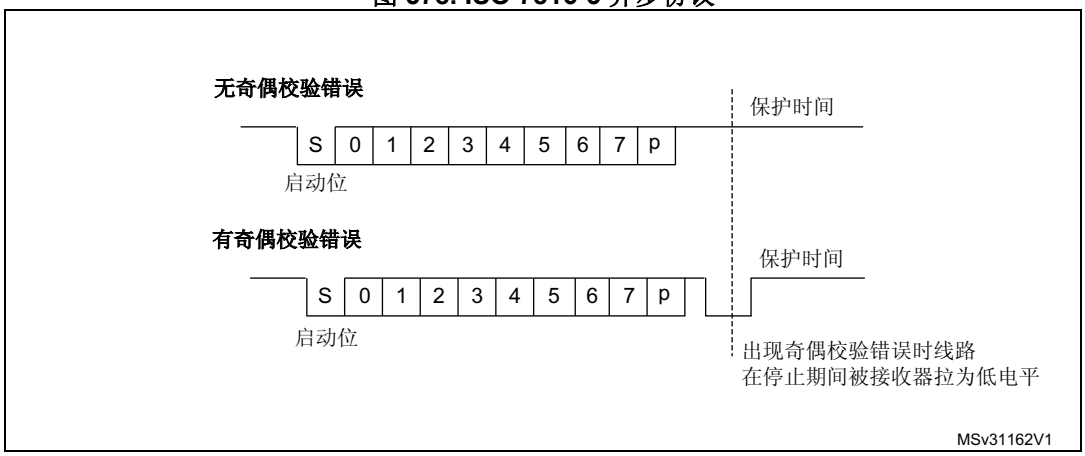
USART 应如下所示进行配置：

- 8 个位加奇偶校验：当 USART_CR1 寄存器中的 M=“1”且 PCE=“1”时
- 发送和接收数据时使用 1.5 个停止位：当 USART_CR2 寄存器中的 STOP=“11”时。接收时也可以选择 0.5 个停止位。

在 T=“0”（字符）模式下，奇偶校验错误在保护时间周期内的每个字符结束时指示。

图 578 显示了有奇偶校验错误和无奇偶校验错误时数据线上情况的示例。

图 578. ISO 7816-3 异步协议



连接到智能卡时，USART 的 TX 输出会驱动一条双向线（它也由智能卡驱动）。必须将 TX 引脚配置为开漏引脚。

智能卡模式采用单线半双工通信协议。

- 从发送移位寄存器发送数据会经过至少 1/2 个时钟周期的延迟。正常工作时，已满的发送移位寄存器会在下一个波特时钟边沿开始移位。在智能卡模式下，此发送过程还会进一步经过 1/2 波特时钟周期的延迟。
- 发送时，如果智能卡检测到奇偶校验错误，它会通过将线路驱动为低电平 (NACK) 告知 USART 此状态。此 NACK 信号（将发送线拉低 1 个时钟周期）会导致发送器端（配置为 1.5 个停止位）出现帧错误。USART 可根据协议自动重新发送数据。重试次数在 SCARCNT 位域中编程。如果经过编程次数的重试后，USART 继续收到 NACK，USART 会停止发送并将该错误以帧错误形式发出。TXE 位（使能 FIFO 模式时为 TXFNF 位）可使用 USART_RQR 寄存器中的 TXFRQ 位置 1。

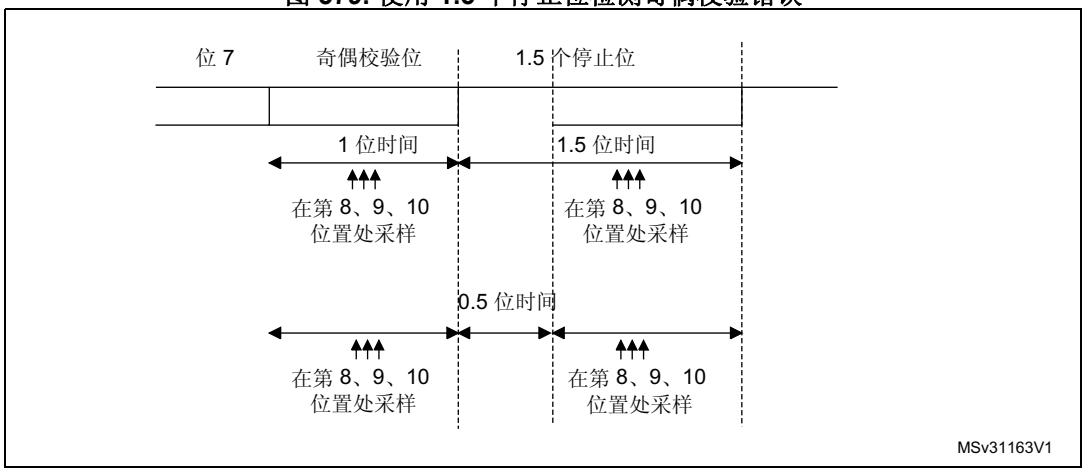
- 发送时智能卡自动重试：在 USART 检测 NACK 与重复字符的起始位之间插入 2.5 个波特率周期的延迟。接收最后一个重复字符后，立即将 TC 位置 1（无保护时间）。如果软件要重复此操作，必须确保标准要求的最短 2 个波特率周期。
- 如果在接收一个使用 1.5 个停止位编程的帧期间检测到奇偶校验错误，则在完成接收帧后，发送线会被拉低一个时钟周期。这是为了向智能卡指出发送到 USART 的数据尚未正确接收。如果 NACK 控制位置 1，接收器会向奇偶校验错误发送 “NACK” 信号；否则不会发送 NACK 信号 （将在 T=1 模式下会使用） 。如果接收到的字符错误，则不会激活 RXNE（使能 FIFO 模式时为 RXFNE）/接收 DMA 请求。根据协议规范，智能卡必须重新发送相同的字符。如果经过 SCARCNT 位域中指定的最大重试次数后接收到的字符仍然错误， USART 会停止发送 NACK 信号，并将错误以奇偶校验错误的形式发出。
- 接收时智能卡自动重试：如果 USART 向智能卡发送 NACK 信号，但智能卡不重复字符，则 BUSY 标志将保持置 1。
- 发送时，USART 会在两个连续字符之间插入保护时间（按照保护时间寄存器中编程的值）。由于保护时间在前一个字符的停止位后测量，因此必须将 GT[7:0] 寄存器编程为所需 CGT（字符保护时间，如 7816-3 规范所定义）减去 12（一个字符的持续时间）。
- 通过对保护时间寄存器进行编程，可以延迟 TC 标志的置位。正常工作时，当发送移位寄存器为空且没有新的发送请求出现时，会对 TC 标志进行置位。在智能卡模式下，空的发送移位寄存器会触发保护时间计数器，使其递增计数至保护时间寄存器中的值。在此期间，TC 标志被强制为低电平。当保护时间计数器达到设置值时，TC 置位为高电平。TCBGT 标志可用于检测数据传输是否结束，而无需等待保护时间结束。该标志在帧发送结束之后且未从智能卡接收到 NACK 时置 1。
- TC 标志的释放不受智能卡模式的影响。
- 如果在发送端检测到帧错误（由来自接收器的 NACK 信号引起），则发送端的接收块不会将 NACK 作为起始位进行检测。根据 ISO 协议，接收到的 NACK 信号的持续时间可以是 1 或 2 个时钟周期。
- 在接收端，如果检测到奇偶校验错误并发送了 NACK 信号，则接收端不会将 NACK 作为起始位进行检测。

注： *break 字符在智能卡模式下无效。带有帧错误的 0x00 数据被视为数据，而非中断。*

当翻转 TE 位时，不会发送空闲帧。空闲帧（在其它配置中进行了定义）在 ISO 协议中未进行定义。

图 579 详细介绍了 USART 如何对 NACK 信号采样。在本例中，USART 正在发送数据并配置了 1.5 个停止位。USART 的接收部分已被使能，以检查数据的完整性和 NACK 信号。

图 579. 使用 1.5 个停止位检测奇偶校验错误



USART 可以通过 SCLK 输出为智能卡提供时钟。在智能卡模式下，SCLK 仅通过一个 5 位预分频器由内部外设输入时钟提供。分频比在 USART_GTPR 寄存器中进行配置。SCLK 频率可在 usart_ker_ck_pres/2 到 usart_ker_ck_pres/62 之间进行编程，其中 usart_ker_ck_pres 为经过编程的预分频器分频的外设输入时钟。

块模式（T=“1”）

在 T=“1”（块）模式下，通过将 UART_CR3 寄存器中的 NACK 位清零可停用奇偶校验错误发送。

在块模式下请求从智能卡执行读操作时，软件必须将 RTOR 寄存器编程为 BWT（块等待时间 - 值 11）。如果在经过此时间段后未从智能卡接收到应答，将生成超时中断。如果该时间段后接收到第一个字符，则通过 RXNE/RXFNE 中断发出信号指示。

注： *即使在使用 DMA 模式下的 USART 从块模式下的智能卡读取时，也必须使能 RXNE/RXFNE 中断。同时，只有在接收到第一个字节后才可使能 DMA。*

接收到第一个字符（RXNE/RXFNE 中断）后，为允许自动检查两个连续字符间的最长等待时间，必须将 RTO 寄存器编程为 CWT（字符等待时间 - 值 11）。此时间以波特率时间单位表示。前一个字符结束后，如果智能卡未在小于 CWT 的时间段内发送新字符，USART 将通过 RTOF 标志和中断（当 RTOIE 位置 1 时）向软件指示此情况。

注： *按照智能卡协议定义，BWT/CWT 的值应从最后一个字符开始（起始位）时定义。必须将 RTO 寄存器分别编程为 BWT -11 或 CWT -11，并考虑最后一个字符本身的长度。*

块长度计数器用于对 USART 接收到的所有字符进行计数。当 USART 进行发送时，此计数器复位。块长度由智能卡在块的第三个字节（起始字段）中传达。必须将此值编程到 USART_RTOR 寄存器中的 BLEN 字段。使用 DMA 模式时，在块开始之前，必须将此寄存器字段编程为最小值 (0x0)。对于该值，在接收到第四个字符后生成中断。软件必须读取 LEN 字段（第三个字节），其值必须从接收缓存区中读取。

在中断驱动接收模式下，块长度可以由软件或者通过编程 BLEN 值来检查。但是在块开始前，可以编程 BLEN 的最大值 (0xFF)。接收到第三个字符后，将编程实际值。

如果块使用 LRC（纵向冗余校验，1 个结尾字节），则 BLEN=LEN。如果块使用 CRC 机制（2 个结尾字节），则必须编程 BLEN=LEN+1。块总长度（包括起始字段、结尾字段和信息字段）等于 BLEN+4。块结束的信号通过 EOBF 标志和中断（EOBIE 位置 1 时）发送给软件。

如果块长度出现错误，则通过 RTO 中断（字符等待时间上溢）发送块结束信号。

注： *错误检查代码 (LRC/CRC) 必须通过软件计算/验证。*

正向约定和反向约定

智能卡协议定义了两种约定：正向约定和反向约定。

正向约定定义为：LSB 在前，逻辑位值 1 对应于线路的 H 状态，奇偶校验为偶校验。要使用此约定，必须编程以下控制位：MSBFIRST=“0”，DATAINV=“0”（默认值）。

反向约定定义为：MSB 在前，逻辑位值 1 对应于信号线路的 L 状态，奇偶校验为偶校验。要使用此约定，必须编程以下控制位：MSBFIRST=“1”，DATAINV=“1”。

注： *将逻辑数据值取反（0=H，1=L）时，奇偶校验位将同样取反。*

为识别智能卡约定，智能卡会将初始字符 TS 作为 ATR（复位应答）的第一个字符发送。TS 支持两种格式：LHHL LLL LLH 和 LHHL HHH LLH。

- (H) LHHL LLL LLH 建立反向约定：状态 L 编码为值 1，时间分量 2 传送最高有效位（MSB 在前）。按反向约定解码时，传送的字节等于“3F”。
- (H) LHHL HHH LLH 建立正向约定：状态 H 编码为值 1，时间分量 2 传送最低有效位（LSB 在前）。按正向约定解码时，传送的字节等于“3B”。

在 2 到 10 的九个时间分量中，如果有偶数个位设置为 1，则字符的奇偶校验正确。

由于 USART 不了解智能卡使用哪种约定，因此 USART 需要能够识别任意一种模式并相应操作。模式识别不在硬件中完成，而是通过软件序列完成。此外，假设以正向约定配置 USART（默认）而智能卡以反向约定 (TS = LHHL LLL LLH) 应答，则 USART 接收到的字节将为“03”，奇偶校验将为奇校验。

因此，有两种方法可用于识别 TS 模式：

方法 1

将 USART 编程为标准智能卡模式/正向约定。这种情况下，TS 模式接收会向智能卡生成奇偶校验错误中断和错误信号。

- 奇偶校验错误中断通知软件智能卡未以正向约定正确应答。之后，软件重新将 USART 编程为反向约定
- 为响应错误信号，智能卡会重试同一 TS 字符，此时重新编程后的 USART 将正确接收该字符

或者，为应答奇偶校验错误中断，软件可决定重新编程 USART，同时向智能卡生成新的复位命令，然后再次等待 TS。

方法 2

将 USART 编程为 9 位/无奇偶校验模式，无位反向。在此模式下，按如下方式接收两种 TS 模式中的任一种：

(H) LHHL LLL LLH = 0x103 -> 选择反向约定

(H) LHHL HHH LLH = 0x13B -> 选择正向约定

软件根据这两种模式检查接收到的字符，如果其中任意一种匹配，则相应编程 USART 以接收下一个字符。

如果两种都未被识别，则可能复位智能卡以重新开始协商。

48.5.18 USART IrDA SIR ENDEC 模块

仅在支持 IrDA 模式时才涉及本节内容。请参见*第 1851 页的第 48.4 节：USART 实现*。

通过将 USART_CR3 寄存器中的 IREN 位置 1 来选择 IrDA 模式。在 IrDA 模式下，必须将以下位清零：

- USART_CR2 寄存器中的 LINEN、STOP 和 CLKEN 位，
- USART_CR3 寄存器中的 SCEN 和 HDSEL 位。

IrDA SIR 物理层规定使用反相归零 (RZI) 调制方案，它以红外光脉冲表示逻辑 0（参见*图 580*）。

SIR 发送编码器用于调制 USART 发出的非归零 (NRZ) 位流。输出脉冲流会发送到外部输出驱动器和红外线 LED。USART 支持的 SIR ENDEC 比特率最高为 115.2 Kbps。在正常模式下，所发送的脉冲宽度规定为一个位周期的 3/16。

SIR 接收解码器用于解调由红外探测器发出的归零位流，并将接收到的 NRZ 串行位流输出到 USART。在空闲状态下，解码器输入通常为高电平（标记状态）。发送编码器输出的极性与解码器输入相反。当解码器输入为低电平时，会检测到起始位。

- IrDA 是一个半双工通信协议。如果发送器忙（USART 正在向 IrDA 编码器发送数据时），则 IrDA 解码器会忽略 IrDA 接收线上的所有数据；如果接收器忙（USART 正在接收来自 USART 的解码数据时），则 IrDA 不会对 USART 发送到 IrDA 的 TX 上的数据进行编码。在接收数据时，应避免同时进行发送，因为这样做可能会破坏要发送的数据。
- “0”作为高电平脉冲发送，而“1”作为“0”发送。在正常模式下，脉冲宽度规定为所选位周期的 3/16（参见*图 581*）。
- SIR 解码器用于将兼容 IrDA 的接收信号转换为 USART 的位流。
- SIR 接收逻辑将高电平状态视为逻辑“1”，将低电平脉冲视为逻辑“0”。
- 发送编码器输出的极性与解码器输入相反。SIR 输出在空闲时处于低电平状态。
- IrDA 规范要求脉冲容忍值要大于 1.41 µs。可接受的脉冲宽度可通过寄存器设置。接收器端的干扰检测逻辑会滤除宽度小于 2 个 PSC 周期的脉冲（PSC 是在 USART_GTPR 中编程的预分频器值）。宽度小于 1 个 PSC 周期的脉冲都将被拒绝，但宽度大于 1 个而小于 2 个周期的脉冲可能被接受也可能被拒绝，而宽度大于 2 个周期的脉冲将被接受作为有效脉冲。当 PSC=“0”时，IrDA 编码器/解码器不工作。
- 接收器能够与低功耗发送器进行通信。
- 在 IrDA 模式下，USART_CR2 寄存器中的停止位必须配置为“1 个停止位”。

IrDA 低功耗模式

- 发送器

  在低功耗模式下，脉冲宽度不再保持为位周期的 3/16。此时的脉冲宽度为低功耗波特率的 3 倍，最小可为 1.42 MHz。通常此值是 1.8432 MHz (1.42 MHz < PSC < 2.12 MHz)。低功耗模式下的可编程分频器会对系统时钟进行分频，以达到此值。

- 接收器

  在低功耗模式下接收与在正常模式下接收类似。为进行干扰检测，USART 应丢弃持续时间短于 1/PSC 的脉冲。只有当持续时间长于 2 个 IrDA 低功耗波特时钟周期（USART_GTPR 的 PSC 值）时，才是有效低电平。

注： *宽度小于两个但大于一个 PSC 周期的脉冲可能被接受，也可能被拒绝。*

接收器的建立时间应由软件进行管理。IrDA 物理层规范规定发送和接收之间至少要经过 10 ms 的延迟（IrDA 是一个半双工协议）。

图 580. IrDA SIR ENDEC 框图

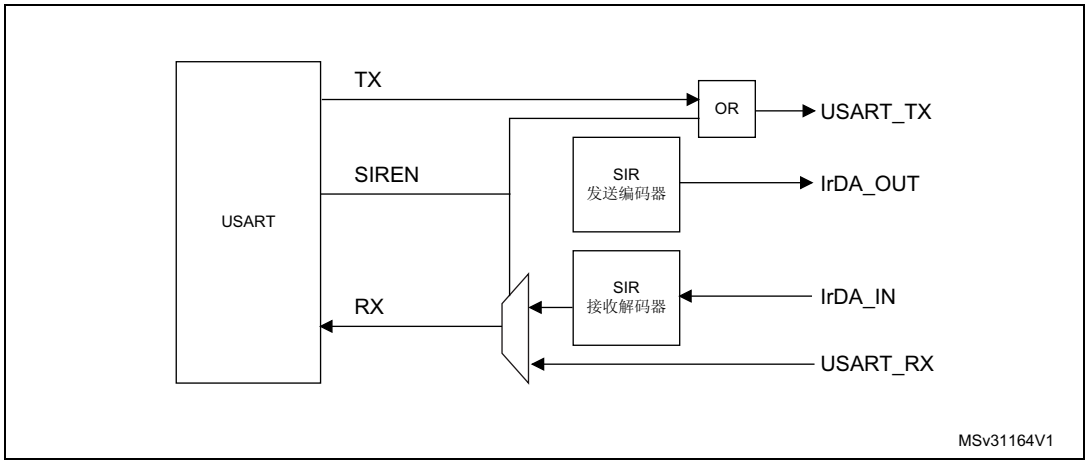
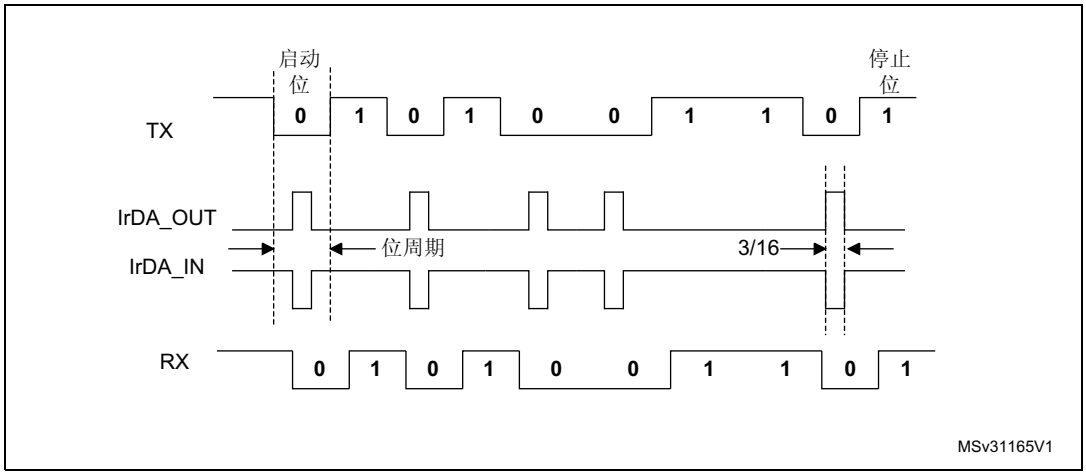


图 581. IrDA 数据调制 (3/16)——正常模式



48.5.19 使用 USART 和 DMA 进行连续通信

USART 能够使用 DMA 进行连续通信。接收缓冲区和发送缓冲区的 DMA 请求是独立生成的。

注： *要确定是否支持 DMA 模式，请参见第 1851 页的第 48.4 节：USART 实现。如果不支持 DMA 模式，请按照第 48.5.6 节中的说明使用 USART。可以将 USART_ISR 寄存器中的 TXE/RXNE 标志清零，从而在禁止 FIFO 时实现连续通信。*

使用 DMA 进行发送

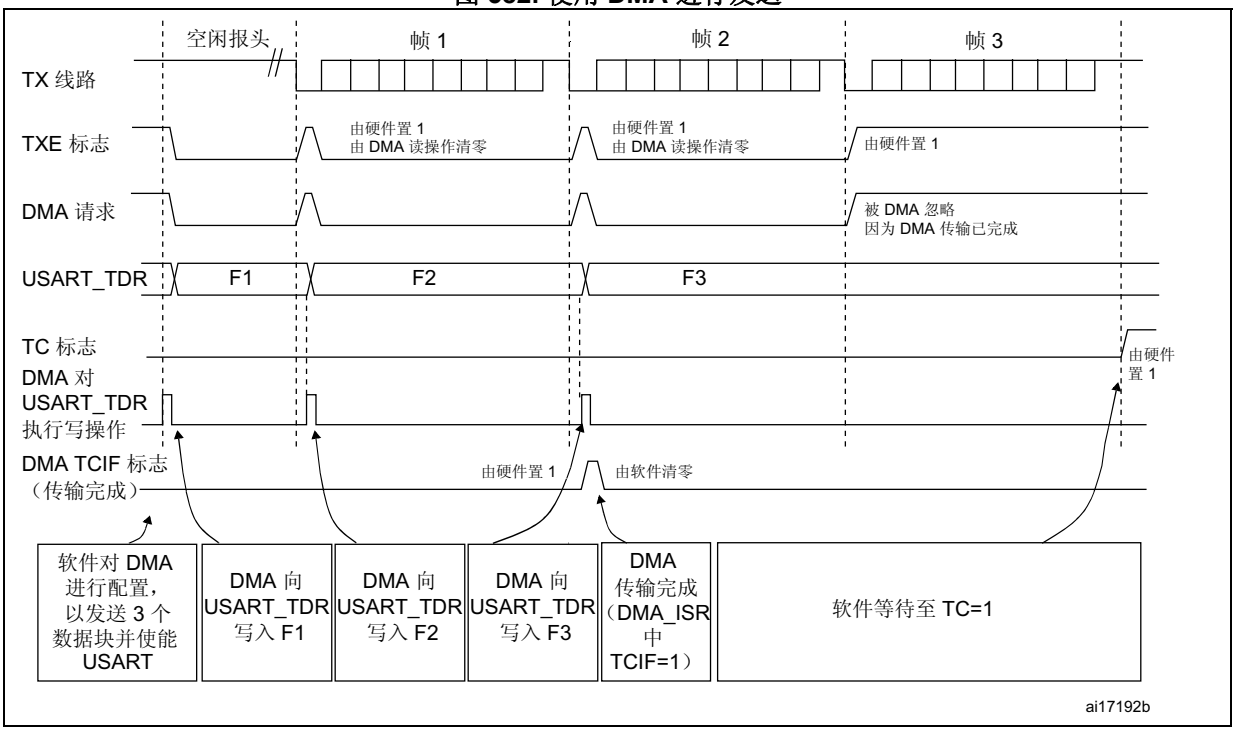
将 USART_CR3 寄存器中的 DMAT 位置 1 可以使能 DMA 模式进行发送。当 TXE 标志（使能 FIFO 模式时为 TXFNF 标志）置 1 时，可将数据从配置的 SRAM 区（通过 DMA 外设，请参见*第 15 节：直接存储器访问控制器（DMA1、DMA2）*和*第 16 节：基本直接存储器访问控制器 (BDMA)*）加载到 USART_TDR 寄存器。要映射一个 DMA 通道以进行 USART 发送，请按以下步骤操作（x 表示通道编号）：

1. 在 DMA 控制寄存器中写入 USART_TDR 寄存器地址，将其配置为传输的目标地址。每次发生 TXE（使能 FIFO 模式时为 TXFNF）事件后，数据都从存储器移动到此地址。
2. 在 DMA 控制寄存器中写入存储器地址，将其配置为传输的源地址。每次发生 TXE（使能 FIFO 模式时为 TXFNF）事件后，数据都从该存储区域加载到 USART_TDR 寄存器中。
3. 在 DMA 控制寄存器中配置要传输的总字节数。
4. 在 DMA 寄存器中配置通道优先级。
5. 根据应用的需求，在完成一半或全部传输后产生 DMA 中断。
6. 通过将 USART_ICR 寄存器中的 TCCF 位置 1，将 USART_ISR 寄存器中的 TC 标志清零。
7. 在 DMA 寄存器中激活该通道。

当达到在 DMA 控制器中设置的数据传输量时，DMA 控制器会在 DMA 通道的中断向量上产生一个中断。

在发送模式下，DMA 对所有要发送的数据执行了写操作（DMA_ISR 寄存器中的 TCIF 标志置 1）后，可以对 TC 标志进行监视，以确保 USART 通信已完成。在禁止 USART 之前或系统进入低功耗模式之前（禁止外设时钟时）必须执行此步骤，以避免损坏最后一次发送。软件必须等待至 TC=“1”。TC 标志在所有数据发送期间都保持清零状态，然后在最后一帧发送结束时由硬件置 1。

图 582. 使用 DMA 进行发送



注： *使能 FIFO 管理时，DMA 请求由发送 FIFO 未满（即 TXFNF = “1”）触发。*

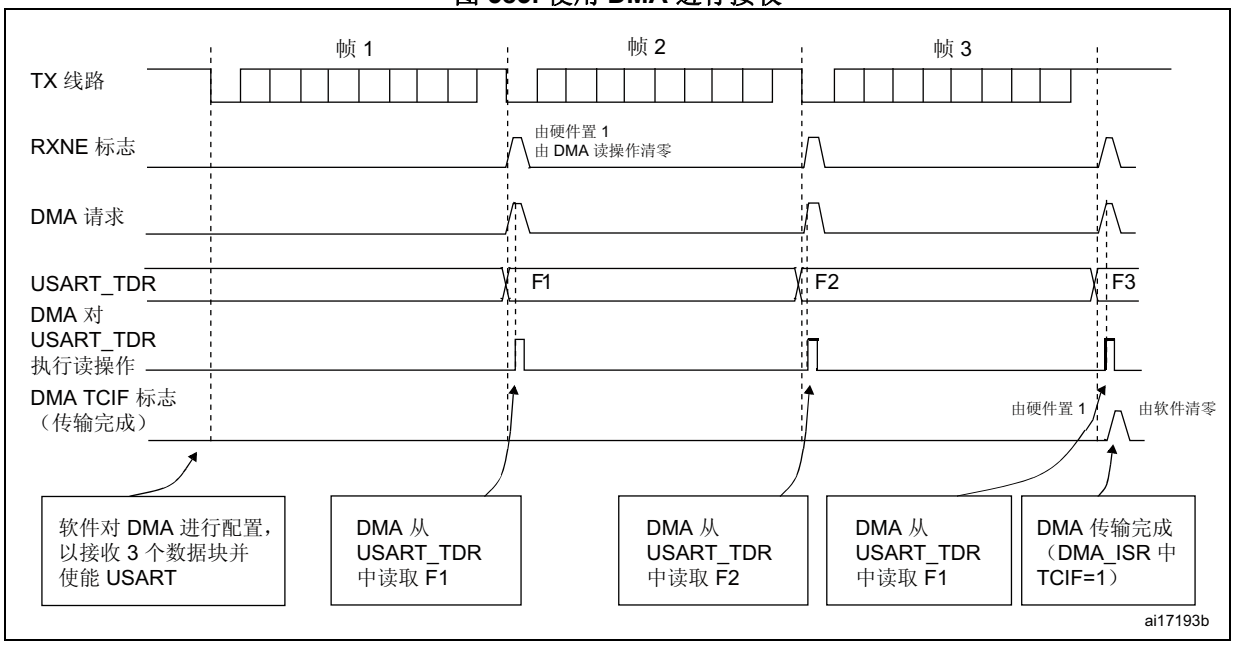
使用 DMA 进行接收

将 USART_CR3 寄存器中的 DMAR 位置 1 可以使能 DMA 模式进行接收。接收数据字节时，数据会从 USART_RDR 寄存器加载到配置的 SRAM 区域中（通过 DMA 外设，请参见 *第 15 节：直接存储器访问控制器（DMA1、DMA2）* 和 *第 16 节：基本直接存储器访问控制器 (BDMA)*）。要映射一个 DMA 通道以进行 USART 接收，请按以下步骤操作：

1. 在 DMA 控制寄存器中写入 USART_RDR 寄存器地址，将其配置为传输的源地址。每次发生 RXNE（使能 FIFO 模式时为 RXFNE）事件后，数据都从该地址移动到存储器。
2. 在 DMA 控制寄存器中写入存储器地址，将其配置为传输的目标地址。每次发生 RXNE（使能 FIFO 模式时为 RXFNE）事件后，数据都从 USART_RDR 加载到此存储区域。
3. 在 DMA 控制寄存器中配置要传输的总字节数。
4. 在 DMA 控制寄存器中配置通道优先级。
5. 根据应用的需求，在完成一半或全部传输后产生中断。
6. 在 DMA 控制寄存器中激活该通道。

当达到在 DMA 控制器中设置的数据传输量时，DMA 控制器会在 DMA 通道的中断向量上产生一个中断。

图 583. 使用 DMA 进行接收



注： *使能 FIFO 管理时，DMA 请求由接收 FIFO 非空（即 RXFNE = “1”）触发。*

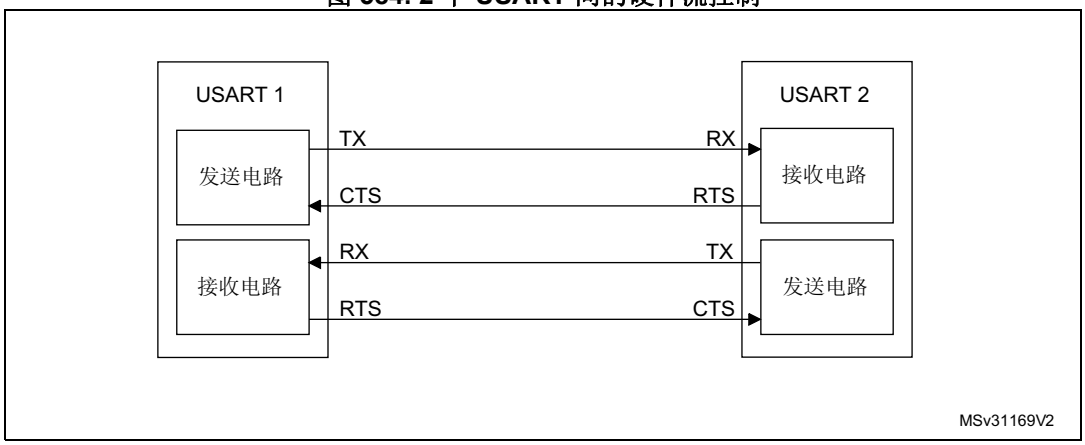
多缓冲区通信中的错误标志和中断生成

在多缓冲区通信模式下，如果事务中发生任何错误，都会在当前字节后将相应错误标志置位。如果中断使能标志置 1，则会产生中断。在单字节接收过程中，与 RXNE（使能 FIFO 模式时为 RXFNE）一同置位的帧错误、上溢错误和噪声标志具有单独的错误标志中断使能位（USART_CR3 寄存器中的 EIE 位）；如果该位置 1，在发生其中任何一个错误时，都会在当前字节后使能中断。

48.5.20 RS232 硬件流控制和 RS485 驱动器使能

使用 nCTS 输入和 nRTS 输出可以控制 2 个器件间的串行数据流。*图 584* 显示了在这种模式下如何连接 2 个器件：

图 584. 2 个 USART 间的硬件流控制

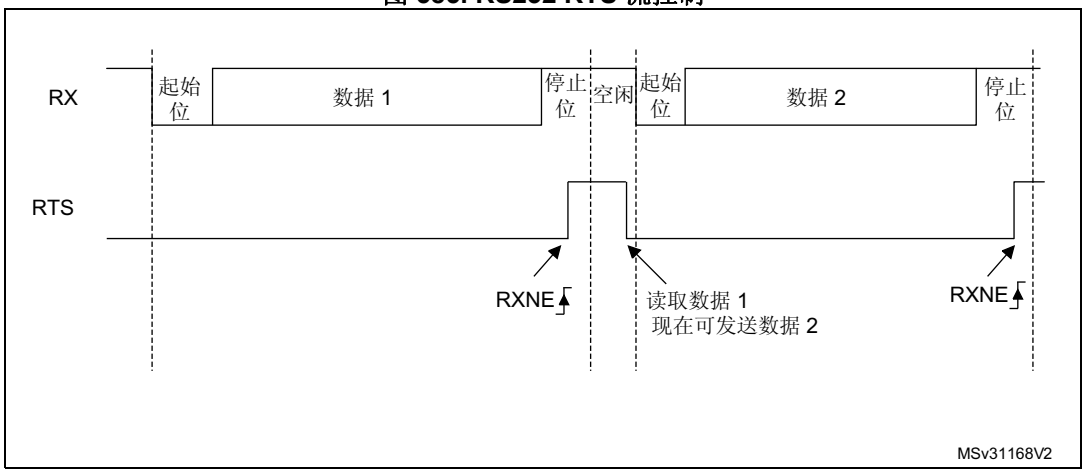


向 USART_CR3 寄存器中的 RTSE 位和 CTSE 位写入“1”，可分别使能 RS232 RTS 和 CTS 流控制。

RS232 RTS 流控制

如果使能 RTS 流控制 (RTSE=1)，只要 USART 接收器准备好接收新数据，便会将 nRTS 变为有效（连接到低电平）。当接收寄存器已满时，会将 nRTS 变为无效，表明发送过程会在当前帧结束后停止。*图 585* 显示了在使能 RTS 流控制的情况下进行通信的示例。

图 585. RS232 RTS 流控制



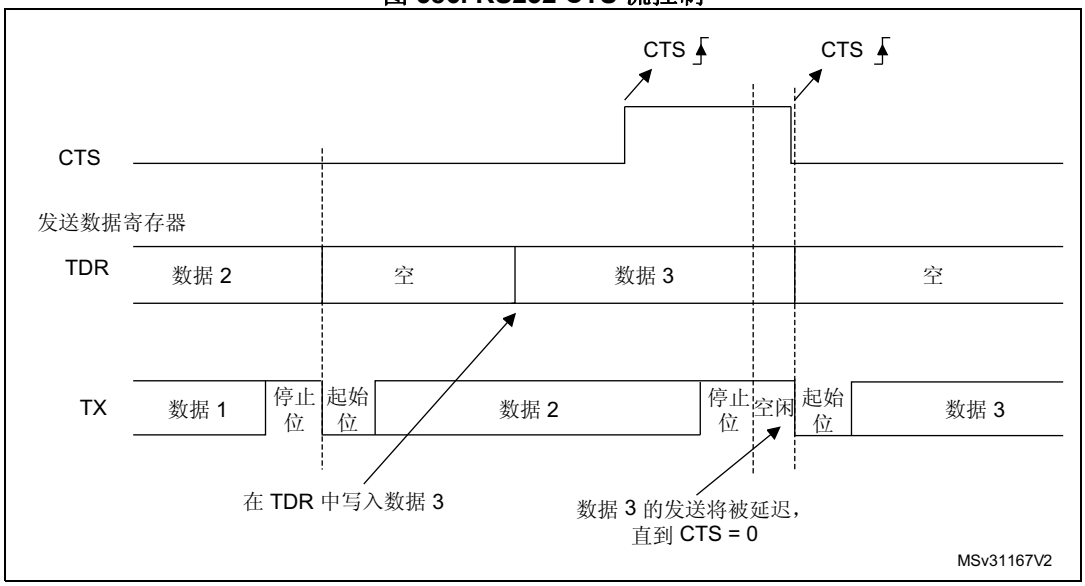
注： *如果使能 FIFO 模式，则仅当 RXFIFO 已满时，nRTS 才会变为无效。*

RS232 CTS 流控制

如果使能 CTS 流控制（CTSE=“1”），则发送器会在发送下一帧前检查 nCTS。如果 nCTS 有效（连接到低电平），则会发送下一数据（假设数据已准备好发送，即 TXE/TCFE='0'）；否则不会进行发送。如果在发送过程中 nCTS 变为无效，则当前发送完成之后，发送器停止。

当 CTSE=“1”时，只要 nCTS 发生变化，CTSIF 状态位便会由硬件自动置 1。这指示接收器是否已准备好进行通信。如果 USART_CR3 寄存器中的 CTSIE 位置 1，则会产生中断。*图 586* 显示了在使能 CTS 流控制的情况下进行通信的示例。

图 586. RS232 CTS 流控制



注： *为正常运行，必须在当前字符结束前至少 3 个 USART 时钟源周期内使能 nCTS。此外还应注意，当脉冲短于 2 个 PCLK 周期时无法将 CTSCF 标志置 1。*

RS485 驱动器使能

驱动器使能功能可通过将 USART_CR3 控制寄存器中的 DEM 位置 1 使能。这样用户便可通过 DE（驱动器使能）信号激活外部收发器控制。使能时间为激活 DE 信号与 START 位开始间的时间。可以通过 USART_CR1 控制寄存器中的 DEAT [4:0] 位域编程禁止时间。禁止时间为发送的消息中最后一个停止位结束与取消激活 DE 信号间的时间。可以通过 USART_CR1 控制寄存器中的 DEDT [4:0] 位域编程禁止时间。DE 信号的极性可使用 USART_CR3 控制寄存器中的 DEP 位配置。

在 USART 中，DEAT 和 DEDT 以采样时间单位表示（1/8 或 1/16 位时间，具体取决于过采样速率）。

48.5.21 USART 低功耗管理

USART 具有高级低功耗模式功能，即使禁止 usart_pclk 时钟，也能正常传输数据。

UESM 位置 1 时，USART 能够将 MCU 从低功耗模式唤醒。

对 usart_pclk 进行门控时，如果某些特定操作需要激活 **usart_pclk** 时钟，则 USART 会提供唤醒中断 (**usart_wkup**)：

- 禁止 FIFO 模式时

  必须激活 usart_pclk 时钟以清空 USART 数据寄存器。

  在这种情况下，usart_wkup 中断源为 RXNE 置“1”。RXNEIE 位必须在进入低功耗模式之前置 1。

- 使能 FIFO 模式时

  必须激活 usart_pclk 时钟以：

  – 填充 TXFIFO

  – 或清空 RXFIFO

  在这种情况下，usart_wkup 中断源可以为：

  – RXFIFO 非空。此时，RXFNEIE 位必须在进入低功耗模式之前置 1。

  – RxFIFO 已满。此时，RXFFIE 位必须在进入低功耗模式之前置 1，接收到的数据量对应于 RXFIFO 大小，并且 RXFF 标志未置 1。

  – TXFIFO 为空。此时，TXFEIE 位必须在进入低功耗模式之前置 1。

  这允许在低功耗模式下发送/接收 TXFIFO/RXFIFO 中的数据。

  为了避免发生上溢/下溢错误以及在低功耗模式下发送/接收数据，usart_wkup 中断源可以是以下事件之一：

  – 达到 TXFIFO 阈值。此时，TXFTIE 位必须在进入低功耗模式之前置 1。

  – 达到 RXFIFO 阈值。此时，RXFTIE 位必须在进入低功耗模式之前置 1。

  例如，如果唤醒时间短于在线路上接收单个字节所需的时间，则应用可将阈值设为最大 RXFIFO 大小。

  使用 RXFIFO 已满、TXFIFO 为空、RXFIFO 非空和 RXFIFO/TXFIFO 阈值中断将 MCU 从低功耗模式唤醒时，可在低功耗模式下尽可能多地进行 USART 传输，这样有助于优化功耗。

或者，也可通过 WUS 位域选择特定 **usart_wkup** 中断。

检测到唤醒事件后，WUF 标志会由硬件置 1 并在 WUFIE 位置 1 时生成一个 **usart_wkup** 中断。在这种情况下，无需 **usart_wkup** 中断，将 WUF 置 1 便足以将 MCU 从低功耗模式唤醒。

注： *在进入低功耗模式之前，请确保未进行任何 USART 传输。检查 BUSY 标志不能确保在进行数据接收时绝不会进入低功耗模式。*

WUF 标志在检测到唤醒事件时置 1，而与 MCU 处于低功耗模式还是工作模式无关。

如果在初始化和使能接收器后立即进入低功耗模式，则必须检查 REACK 位以确保 USART 确实已使能。

当 DMA 用于接收时，它必须在进入低功耗模式前禁止，并在退出低功耗模式后重新使能。

如果使能 FIFO，则只有在使能静默模式的情况下才能在地址匹配时从低功耗模式唤醒。

使用静默模式和低功耗模式

如果 USART 在进入低功耗模式前处于静默模式：

- 不得使用空闲检测时从静默模式唤醒，因为空闲检测无法在低功耗模式下工作。
- 如果使用地址匹配时从静默模式唤醒，则低功耗模式唤醒源也必须是地址匹配。如果 RXNE 标志在进入低功耗模式时置 1，则接口将在地址匹配时和从低功耗模式唤醒时保持静默模式。

注： *使能 FIFO 管理时，静默模式可与从低功耗模式唤醒搭配使用，而且没有任何限制（即，上述关于静默和低功耗模式的两点仅在 FIFO 管理禁止时有效）。*

USART 内核时钟 (usart_ker_ck) 在低功耗模式下关闭时从低功耗模式唤醒

在低功耗模式下，如果在 USART 接收线上检测到下降沿时 usart_ker_ck 时钟处于关闭状态，则 USART 接口会借助 usart_ker_ck_req 信号请求开启 usart_ker_ck 时钟。usart_ker_ck 随后用来接收帧。

如果唤醒事件通过验证，则 MCU 将从低功耗模式唤醒，并会正常进行数据接收。

如果唤醒事件未通过验证，则 usart_ker_ck 会再次关闭，MCU 不会被唤醒并保持在低功耗模式下，且内核时钟请求被释放。

以下示例显示了将唤醒事件编程为“地址匹配检测”并禁止 FIFO 管理的情况。

图 587 所示为唤醒事件通过验证时的 USART 行为。

图 587. 唤醒事件通过验证（唤醒事件 = 地址匹配，禁止 FIFO）

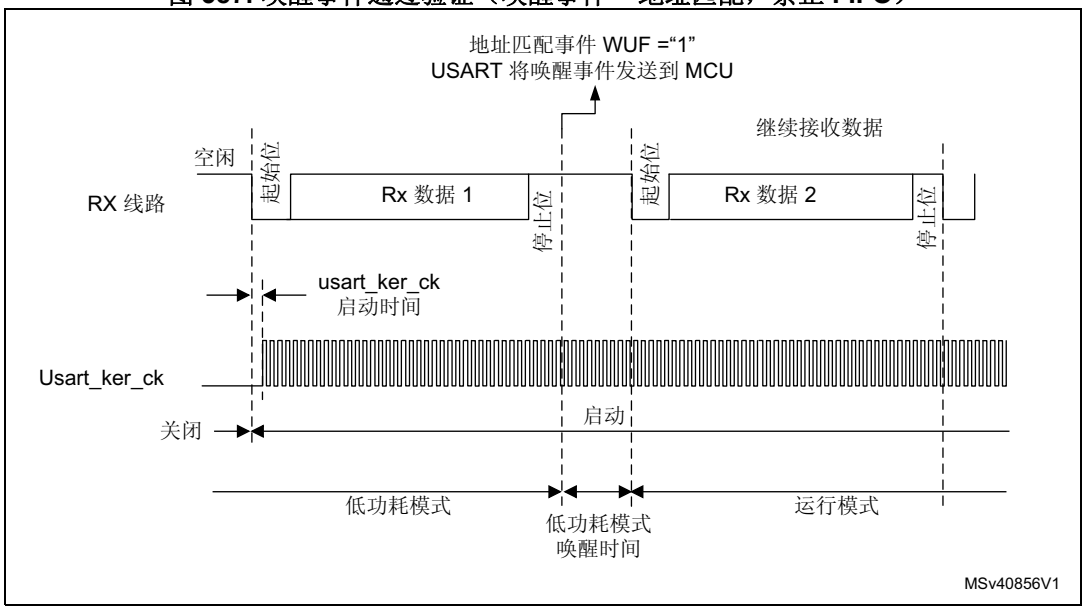
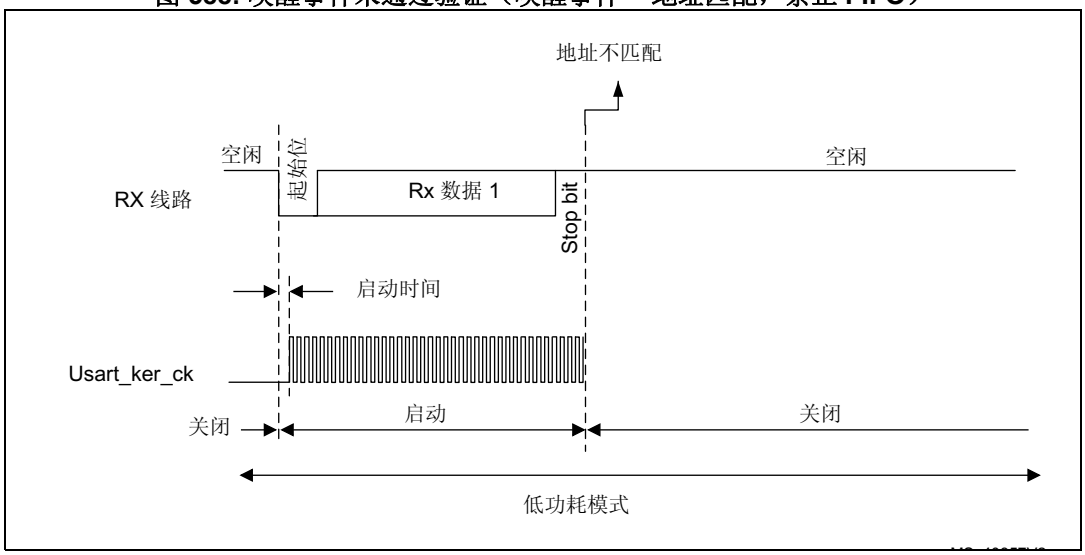


图 588 所示为唤醒事件未通过验证时的 USART 行为。

图 588. 唤醒事件未通过验证（唤醒事件 = 地址匹配，禁止 FIFO）



注： *将地址匹配或任一接收的帧用作唤醒事件时，如上两图适用。如果唤醒事件为起始位检测，则 USART 会在起始位结束时向 MCU 发送唤醒事件。*

48.6 USART 中断

在 USART 通信过程中，中断 (usart_it) 可由不同事件生成。USART 模块也可生成唤醒中断 (usart_wkup)。

有关所有 USART 中断请求的详细说明，请参见*表 380*。

表 380. USART 中断请求

中断事件	事件标志	使能控制位	中断清除方法	中断已激活	
				usart_it	usart_wkup
发送数据寄存器为空	TXE	TXEIE	TXE 在 TDR 中被写入数据时清零。	是	否
发送 FIFO 未满	TXFNF	TXFNFIE	TXFNF 在 TXFIFO 已满时清零。	是	否
发送 FIFO 为空	TXFE	TXFEIE	TXFE 在 TXFIFO 包含至少一个数据时清零，或通过将 TXFRQ 位置 1 来清零。	是	是
达到发送 FIFO 阈值	TXFT	TXFTIE	TXFT 在 TXFIFO 内容少于编程的阈值时由硬件清零。	是	是
CTS 中断	CTSIF	CTSIE	CTSIF 由软件通过将 CTSCF 位置 1 来清零。	是	否
发送完成	TC	TCIE	TC 在 TDR 中被写入数据时清零，或通过将 TCCF 位置 1 来清零。	是	否
保护时间前发送完成	TCBGT	TCBGTIE	TCBGT 在 TDR 中被写入数据时清零，或通过将 TCBGTCF 位置 1 来清零。	是	否

表 380. USART 中断请求（续）

中断事件	事件标志	使能控制位	中断清除方法	中断已激活	
				usart_it	usart_wkup
接收数据寄存器不为空（已准备好读取数据）	RXNE	RXNEIE	RXNE 通过读取 RDR 或通过将 RXFRQ 位置 1 来清零。	是	是
接收 FIFO 非空	RXFNE	RXFNEIE	RXFNE 在 RXFIFO 为空时清零，或通过将 RXFRQ 位置 1 来清零。	是	是
接收 FIFO 已满	RXFF(1)	RXFFIE	RXFF 在 RXFIFO 至少包含一个数据时清零。	是	是
达到接收 FIFO 阈值	RXFT	RXFTIE	RXFT 在 RXFIFO 内容少于编程的阈值时由硬件清零	是	是
检测到溢出错误	ORE	RXNEIE/ RXFNEIE	ORE 通过将 ORECF 位置 1 来清零。	是	否
检测到空闲线路	IDLE	IDLEIE	IDLE 通过将 IDLECF 位置 1 来清零。	是	否
奇偶校验错误	PE	PEIE	PE 通过将 PECF 位置 1 来清零。	是	否
LIN 中断	LBDF	LBDIE	LBDF 通过将 LBDCF 位置 1 来清零。	是	否
多缓冲区通信中的噪声标志、溢出错误和帧错误。	NE 或 ORE 或 FE	EIE	NE 通过将 NFCF 位置 1 来清零。 ORE 通过将 ORECF 位置 1 来清零。 FE 标志通过将 FECF 位置 1 来清零。	是	否
字符匹配	CMF	CMIE	CMF 通过将 CMCF 位置 1 来清零。	是	否
接收器超时	RTOF	RTOFIE	RTOF 通过将 RTOCCF 位置 1 来清零。	是	否
块结束	EOBF	EOBIE	EOBF 通过将 EOBCF 位置 1 来清零。	是	否
从低功耗模式唤醒	WUF(2)	WUFIE	WUF 通过将 WUCF 位置 1 来清零。	否	是
SPI 从器件下溢错误	UDR	EIE	UDR 通过将 UDRCF 位置 1 来清零。	是	否
达到发送 FIFO 阈值	TXFT	TXFTIE	TXFT 在 TXFIFO 内容少于编程的阈值时由硬件清零。	是	是
达到接收 FIFO 阈值	RXFT	RXFTIE	RXFT 在 RXFIFO 内容少于编程的阈值时由硬件清零。	是	是

1. 如果 USART 接收 n+1 个数据（n 表示 RXFIFO 大小，RXFIFO 中有 n 个数据，USART_RDR 中有 1 个数据），则 RXFF 标志置位。在停止模式下，未向 USART_RDR 提供时钟。因此，将不会对该寄存器进行写操作，一旦有 n 个数据被接收并写入到 RXFIFO，RXFF 中断将生效（RXFF 标志未置 1）。
2. WUF 中断仅在低功耗模式下有效。

48.7 USART 寄存器

有关寄存器说明中使用的缩写，请参见*第 94 页的第 1.1 节*。

48.7.1 USART 控制寄存器 1 (USART_CR1)

USART control register 1

偏移地址：0x00

复位值：0x0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RXFFIE	TXFEIE	FIFOEN	M1	EOBIE	RTOIE	DEAT[4:0]					DEDT[4:0]				
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OVER8	CMIE	MME	M0	WAKE	PCE	PS	PEIE	TXEIE/TXFNFIE	TCIE	RXNEIE/RXFNEIE	IDLEIE	TE	RE	UESM	UE
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31 **RXFFIE**：RXFIFO 变满时中断使能 (FIFO full interrupt enable)

此位由软件置 1 和清零。

0：禁止中断

1：当 USART_ISR 寄存器中的 RXFF=“1”时，生成 USART 中断

注：　禁止 FIFO 模式时，该位保留且必须保持复位值。

位 30 **TXFEIE**：TXFIFO 为空时中断使能 (TXFIFO empty interrupt enable)

此位由软件置 1 和清零。

0：禁止中断

1：当 USART_ISR 寄存器中的 TXFE=“1”时，生成 USART 中断

注：　禁止 FIFO 模式时，该位保留且必须保持复位值。

位 29 **FIFOEN**：FIFO 模式使能 (FIFO mode enable)

此位由软件置 1 和清零。

0：禁止 FIFO 模式。

1：使能 FIFO 模式。

只有在禁止 USART（UE=“0”）时才能写入此位域。

注：　FIFO 模式只能在标准 UART 通信、SPI 主/从器件模式和智能卡模式下使用，不得在 IrDA 和 LIN 模式下使能。

位 28 **M1**：字长 (Word length)

此位必须与位 12 (M0) 搭配使用来确定字长度。该位由软件置 1 或清零。

M[1:0] =“00”：1 个起始位，8 个数据位，n 个停止位

M[1:0] =“01”：1 个起始位，9 个数据位，n 个停止位

M[1:0] =“10”：1 个起始位，7 个数据位，n 个停止位

只有在禁止 USART（UE=“0”）时才能写入该位。

注：　在 7 位数据长度模式下，不支持智能卡模式、LIN 主模式和自动波特率（0x7F 帧和 0x55 帧检测）。

位 27 **EOBIE**：块结束中断使能 (End of Block interrupt enable)

此位由软件置 1 和清零。

0：禁止中断

1：USART_ISR 寄存器中的 EOBF 标志置 1 时生成 USART 中断。

注：　如果 USART 不支持智能卡模式，该位保留并由硬件强制清零。
请参见第 1851 页的第 48.4 节：USART 实现。

位 26 **RTOIE**：接收器超时中断使能 (Receiver timeout interrupt enable)

此位由软件置 1 和清零。

0：禁止中断

1：USART_ISR 寄存器中的 RTOF 位置 1 时生成 USART 中断。

注：　如果 USART 不支持接收器超时功能，该位保留并由硬件强制清零。
请参见第 1851 页的第 48.4 节：USART 实现。

位 25:21 **DEAT[4:0]**：驱动器使能使能时间 (Driver Enable assertion time)

该 5 位值用于定义激活 DE（启动器使能）信号与起始位开始间的时间。此时间以采样时间单位表示（1/8 或 1/16 位时间，具体取决于过采样速率）。

只有在禁止 USART（UE=“0”）时才能写入此位域。

注：　如果不支持驱动器使能功能，该位保留并且必须保持清零。
请参见第 1851 页的第 48.4 节：USART 实现。

位 20:16 **DEDT[4:0]**：驱动器使能禁止时间 (Driver Enable deassertion time)

该 5 位值用于定义发送的消息中最后一个停止位结束与取消激活 DE（驱动器使能）信号间的时间。此时间以采样时间单位表示（1/8 或 1/16 位时间，具体取决于过采样速率）。

如果在 DEDT 时间内对 USART_TDR 寄存器执行写操作，则新数据仅在经过 DEDT 和 DEAT 时间后才会发送。

只有在禁止 USART（UE=“0”）时才能写入此位域。

注：　如果不支持驱动器使能功能，该位保留并且必须保持清零。
请参见第 1851 页的第 48.4 节：USART 实现。

位 15 **OVER8**：过采样模式 (Oversampling mode)

0：16 倍过采样

1：8 倍过采样

只有在禁止 USART（UE=“0”）时才能写入该位。

注：　在 LIN、IrDA 和智能卡模式下，此位必须保持清零。

位 14 **CMIE**：字符匹配中断使能 (Character match interrupt enable)

此位由软件置 1 和清零。

0：禁止中断

1：USART_ISR 寄存器中的 CMF 位置 1 时生成 USART 中断。

位 13 **MME**：静默模式使能 (Mute mode enable)

此位可使能 USART 静默模式功能。此位置 1 时，USART 会按照 WAKE 位的定义在工作模式与静默模式之间切换。该位由软件置 1 和清零。

0：接收器永久处于工作模式

1：接收器可在静默模式和工作模式之间切换。

位 12 **M0**：字长 (Word length)

此位与位 28 (M1) 搭配使用来确定字长度。它由软件置 1 或清零（请参见位 28 (M1) 的说明）。

只有在禁止 USART（UE=“0”）时才能写入该位。

位 11 **WAKE**：接收器唤醒方法 (Receiver wakeup method)

此位用于确定 USART 静默模式的唤醒方法。该位由软件置 1 或清零。

0：空闲线路

1：地址标记

只有在禁止 USART（UE=“0”）时才能写入此位域。

位 10 **PCE**：奇偶校验控制使能 (Parity control enable)

该位选择硬件奇偶校验控制（生成和检测）。使能奇偶校验控制时，计算出的奇偶校验位被插入到 MSB 位置（如果 M=“1”，则为第 9 位；如果 M=“0”，则为第 8 位），并对接收到的数据检查奇偶校验位。此位由软件置 1 和清零。一旦该位置 1，PCE 在当前字节的后面处于活动状态（在接收和发送时）。

0：禁止奇偶校验控制

1：使能奇偶校验控制

只有在禁止 USART（UE=“0”）时才能写入此位域。

位 9 **PS**：奇偶校验选择 (Parity selection)

该位用于在使能奇偶校验生成/检测（PCE 位置 1）时选择奇校验或偶校验。该位由软件置 1 和清零。将在当前字节的后面选择奇偶校验。

0：偶校验

1：奇校验

只有在禁止 USART（UE=“0”）时才能写入此位域。

位 8 **PEIE**：PE 中断使能 (PE interrupt enable)

此位由软件置 1 和清零。

0：禁止中断

1：当 USART_ISR 寄存器中的 PE=“1”时，生成 USART 中断

位 7 **TXEIE/TXFNFIE**：发送数据寄存器为空/TXFIFO 未满中断使能 (Transmit data register empty/ TXFIFO not full interrupt enable)

此位由软件置 1 和清零。

0：禁止中断

1：当 USART_ISR 寄存器中的 TXE/TXFNF=“1”时，生成 USART 中断

位 6 **TCIE**：传输完成中断使能 (Transfer complete interrupt enable)

此位由软件置 1 和清零。

0：禁止中断

1：当 USART_ISR 寄存器中的 TC=“1”时，生成 USART 中断

位 5 **RXNEIE/RXFNEIE**：接收数据寄存器非空/RXFIFO 非空中断使能 (Receive data register not empty/RXFIFO not empty interrupt enable)

此位由软件置 1 和清零。

0：禁止中断

1：当 USART_ISR 寄存器中的 ORE=“1”或 RXNE/RXFNE=“1”时，生成 USART 中断

位 4 **IDLEIE**：IDLE 中断使能 (IDLE interrupt enable)

此位由软件置 1 和清零。

0：禁止中断

1：当 USART_ISR 寄存器中的 IDLE=“1”时，生成 USART 中断

位 3 **TE**：发送器使能 (Transmitter enable)

该位使能发送器。该位由软件置 1 和清零。

0：禁止发送器

1：使能发送器

注：　除了在智能卡模式下以外，传送期间 TE 位上的低电平脉冲（“0”后紧跟的是“1”）会在当前字的后面发送一个报头（空闲线路）。为生成空闲字符，TE 不能立即写入“1”。为确保所需的持续时间，软件可轮询 USART_ISR 寄存器中的 TEACK 位。

在智能卡模式下，当 TE 置 1 时，在发送开始前存在 1 位的时间延迟。

位 2 **RE**：接收器使能 (Receiver enable)

该位使能接收器。该位由软件置 1 和清零。

0：禁止接收器

1：使能接收器并开始搜索起始位

位 1 **UESM**：低功耗模式下的 USART 使能 (USART enable in low-power mode)

当此位清零时，USART 无法将 MCU 从低功耗模式唤醒。

当此位置 1 时，USART 可将 MCU 从低功耗模式唤醒。

此位由软件置 1 和清零。

0：USART 无法将 MCU 从低功耗模式唤醒。

1：USART 能够将 MCU 从低功耗模式唤醒。

注：　建议在进入低功耗模式前将 UESM 位置 1，并在退出低功耗模式时将其清零。

如果 USART 不支持从停止模式唤醒功能，则此位保留并由硬件强制清零。

请参见第 1851 页的第 48.4 节：USART 实现。

位 0 **UE**：USART 使能 (USART enable)

此位清零后，USART 预分频器和输出将立即停止，并丢弃所有当前操作。USART 配置会保留，而所有的 USART_ISR 状态标志均会复位。此位由软件置 1 和清零。

0：禁止 USART 预分频器和输出，低功耗模式

1：使能 USART

注：　为进入低功耗模式而不在线路上生成错误，之前必须复位 TE 位，并且软件必须等待 USART_ISR 中的 TC 位置 1 后才能复位 UE 位。

UE = “0”时也会复位 DMA 请求，因此必须在复位 UE 位前禁止 DMA 通道。

在智能卡模式下（SCEN = “1”），无论 UE 位值为何，当 CLKEN = “1”时，SCLK 始终可用。

48.7.2 USART 控制寄存器 2 (USART_CR2)

USART control register 2

偏移地址：0x04

复位值：0x0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ADD[7:4]				ADD[3:0]				RTOEN	ABRMOD[1:0]		ABREN	MSBFIRST	DATAINV	TXINV	RXINV
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SWAP	LINEN	STOP[1:0]		CLKEN	CPOL	CPHA	LBCL	Res	LBDIE	LBDL	ADDM7	DIS_NSS.	Res.	Res.	SLVEN.
rw	rw	rw	rw	rw	rw	rw	rw		rw	rw	rw	rw	rw		rw

位 31:28 **ADD[7:4]**：USART 节点的地址 (Address of the USART node)

此位域用于指定 USART 节点的地址或要识别的字符代码。

它在多处理器通信时于静默模式或低功耗模式下使用，以通过 7 位地址标记检测唤醒 MCU。发送器发送字符的 MSB 应为 1。此位域还可用于正常接收和静默模式无效时的字符检测（例如，ModBus 协议中的块结束检测）。这种情况下，接收到的整个字符（8 位）将与 ADD[7:0] 值进行比较，如果匹配，CMF 标志将置 1。

仅在禁止接收（RE =“0”）或禁止 USART（UE=“0”）时才能写入该位域

位 27:24 **ADD[3:0]**：USART 节点的地址 (Address of the USART node)

此位域用于指定 USART 节点的地址或要识别的字符代码。

它在多处理器通信时于静默模式或低功耗模式下使用，以通过地址标记检测进行唤醒。

仅在禁止接收（RE =“0”）或禁止 USART（UE=“0”）时才能写入该位域

位 23 **RTOEN**：接收器超时使能 (Receiver timeout enable)

此位由软件置 1 和清零。

0：禁止接收器超时功能。

1：使能接收器超时功能。

使能此功能后，如果 RX 线路在 RTOR（接收器超时寄存器）中编程的持续时间内处于空闲状态（无接收），则 USART_ISR 寄存器中的 RTOF 标志置 1。

注：　如果 USART 不支持接收器超时功能，该位保留并由硬件强制清零。
请参见第 1851 页的第 48.4 节：USART 实现。

位 22:21 **ABRMOD[1:0]**：自动波特率模式 (Auto baud rate mode)

这些位将由软件置 1 和清零。

00：通过测量起始位检测波特率。

01：下降沿到下降沿的测量（接收到的帧必须以一个等于 1 的位开头，即帧 = 10xxxxxx）

10：0x7F 帧检测。

11：0x55 帧检测。

仅在 ABREN =“0”时或禁止 USART（UE=“0”）时才能写入该位域。

注：　如果 DATAINV=“1”且/或 MSBFIRST=“1”，这些模式必须与在线路上时相同，例如 MSBFIRST 的 0xAA

如果 USART 不支持自动波特率功能，则此位保留并由硬件强制清零。
请参见第 1851 页的第 48.4 节：USART 实现。

位 20 **ABREN**：自动波特率使能 (Auto baud rate enable)

此位由软件置 1 和清零。

0：禁止自动波特率检测。

1：使能自动波特率检测。

注：　如果 USART 不支持自动波特率功能，则此位保留并由硬件强制清零。
请参见第 1851 页的第 48.4 节：USART 实现。

位 19 **MSBFIRST**：最高有效位在前 (Most significant bit first)

此位由软件置 1 和清零。

0：发送/接收数据时位 0 在前，跟在起始位后。

1：发送/接收数据时 MSB（位 7/8）在前，跟在起始位后。

只有在禁止 USART（UE=“0”）时才能写入此位域。

位 18 **DATAINV**：二进制数据反向 (Binary data inversion)

此位由软件置 1 和清零。

0：按正/正向逻辑发送/接收数据寄存器中的逻辑数据。（1=H，0=L）

1：按负/反向逻辑发送/接收数据寄存器中的逻辑数据。（1=L，0=H）。奇偶校验位也取反。

只有在禁止 USART（UE=“0”）时才能写入此位域。

位 17 **TXINV**：TX 引脚有效电平反向 (TX pin active level inversion)

此位由软件置 1 和清零。

0：TX 引脚信号使用标准逻辑电平（V_{DD} = 1/空闲，Gnd = 0/标记）工作

1：对 TX 引脚信号值取反。（V_{DD} = 0/标记，Gnd=1/空闲）

允许在 TX 线路上使用外部反相器。

只有在禁止 USART（UE=“0”）时才能写入此位域。

位 16 **RXINV**：RX 引脚有效电平反向 (RX pin active level inversion)

此位由软件置 1 和清零。

0：RX 引脚信号使用标准逻辑电平（V_{DD} = 1/空闲，Gnd = 0/标记）工作

1：对 RX 引脚信号值取反。（V_{DD} = 0/标记，Gnd=1/空闲）

允许在 RX 线路上使用外部反相器。

只有在禁止 USART（UE=“0”）时才能写入此位域。

位 15 **SWAP**：交换 TX/RX 引脚 (Swap TX/RX pins)

此位由软件置 1 和清零。

0：按标准引脚排列定义使用 TX/RX 引脚

1：交换 TX 和 RX 引脚功能。允许在与另一个 UART 的交叉连接时工作

只有在禁止 USART（UE=“0”）时才能写入此位域。

位 14 **LINEN**：LIN 模式使能 (LIN mode enable)

此位由软件置 1 和清零。

0：禁止 LIN 模式

1：使能 LIN 模式

LIN 模式可以使用 USART_CR1 寄存器中的 SBKRQ 位发送 LIN 同步中断（13 个低位），并可检测 LIN 同步中断。

只有在禁止 USART（UE=“0”）时才能写入此位域。

注：　如果 USART 不支持 LIN 模式，该位保留并由硬件强制清零。请参见第 1851 页的第 48.4 节：USART 实现。

位 13:12 **STOP[1:0]**：停止位 (STOP bits)

这些位用于编程停止位。

00：1 个停止位

01：0.5 个停止位

10：2 个停止位

11：1.5 个停止位

只有在禁止 USART（UE=“0”）时才能写入此位域。

位 11 **CLKEN**：时钟使能 (Clock enable)

该位允许用户使能 SCLK 引脚。

0：禁止 SCLK 引脚

1：使能 SCLK 引脚

只有在禁止 USART（UE=“0”）时才能写入该位。

注：　如果既不支持同步模式，也不支持智能卡模式，则该位保留并由硬件强制清零。
请参见第 1851 页的第 48.4 节：USART 实现。
在智能卡模式下，为了向智能卡正确提供 SCLK 时钟，必须按以下步骤操作：

- UE = “0”
- SCEN = “1”
- GTPR 配置
- CLKEN= “1”
- UE = “1”

位 10 **CPOL**：时钟极性 (Clock polarity)

该位允许用户在同步模式下选择 SCLK 引脚上时钟输出的极性。它与 CPHA 位结合使用可获得所需的时钟/数据关系

0：空闲时 SCLK 引脚为低电平

1：空闲时 SCLK 引脚为高电平

只有在禁止 USART（UE=“0”）时才能写入该位。

注：　如果不支持同步模式，该位保留并由硬件强制清零。请参见第 1851 页的第 48.4 节：USART 实现。

位 9 **CPHA**：时钟相位 (Clock phase)

此位用于在同步模式下选择 SCLK 引脚上时钟输出的相位。它与 CPOL 位结合使用可获得所需的时钟/数据关系（请参见*图 568* 和*图 569*）

0：从第一个时钟边沿开始采样数据

1：从第二个时钟边沿开始采样数据

只有在禁止 USART（UE=“0”）时才能写入该位。

注：　如果不支持同步模式，该位保留并由硬件强制清零。请参见第 1851 页的第 48.4 节：USART 实现。

位 8 **LBCL**：最后一个位时钟脉冲 (Last bit clock pulse)

此位用于在同步模式下选择与发送的最后一个数据位 (MSB) 关联的时钟脉冲是否必须在 SCLK 引脚上输出。

0：最后一个数据位的时钟脉冲不在 SCLK 引脚上输出

1：最后一个数据位的时钟脉冲在 SCLK 引脚上输出

注意：　最后一位为发送的第 7 个、第 8 个或第 9 个数据位，具体取决于 USART_CR1 寄存器中 M 位所选择的7位、8 位或 9 位格式。

只有在禁止 USART（UE=“0”）时才能写入该位。

注：　如果不支持同步模式，该位保留并由硬件强制清零。请参见第 1851 页的第 48.4 节：USART 实现。

位 7 保留，必须保持复位值。

位 6 **LBDIE**：LIN 中断检测中断使能 (LIN break detection interrupt enable)

中断中断屏蔽（使用中断分隔符进行中断检测）

0：禁止中断

1：当 USART_ISR 寄存器中的 LBDF =“1”时，生成中断

注：　如果 LIN 不支持智能卡模式，该位保留并由硬件强制清零。请参见第 1851 页的第 48.4 节：USART 实现。

位 5 **LBDL**：LIN 中断检测长度 (LIN break detection length)

该位用于选择 11 位中断检测或 10 位中断检测。

0：10 位中断检测

1：11 位中断检测

只有在禁止 USART（UE=“0”）时才能写入该位。

注：　如果 LIN 不支持智能卡模式，该位保留并由硬件强制清零。请参见第 1851 页的第 48.4 节：USART 实现。

位 4 **ADDM7**：7 位地址检测/4 位地址检测 (7-bit Address Detection/4-bit Address Detection)

此位用于选择 4 位地址检测或 7 位地址检测。

0：4 位地址检测

1：7 位地址检测（在 8 位数据模式下）

只有在禁止 USART（UE=“0”）时才能写入该位

注：　在 7 位和 9 位数据模式下，地址检测分别在 6 位和 8 位地址上完成（ADD[5:0] 和 ADD[7:0]）。

位 3 **DIS_NSS**

当 DSI_NSS 位置 1 时，忽略 NSS 引脚输入。

0：SPI 从器件选择取决于 NSS 输入引脚。

1：始终选择 SPI 从器件，忽略 NSS 输入引脚。

注： 不支持 SPI 从器件模式时，该位保留且必须保持复位值。
请参见第 1851 页的第 48.4 节：USART 实现。

位 2:1 保留，必须保持复位值

位 0 **SLVEN**：同步从模式使能 (Synchronous Slave mode enable)

SLVEN 位置 1 时，使能同步从模式。

0：禁止从模式。

1：使能从模式。

注： 不支持 SPI 从器件模式时，该位保留且必须保持复位值。
请参见第 1851 页的第 48.4 节：USART 实现。

注： 使能发送器时不应对 CPOL、CPHA 和 LBCL 位进行写操作。

48.7.3 USART 控制寄存器 3 (USART_CR3)

USART control register 3

偏移地址：0x08

复位值：0x0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TXFTCFG			RXF TIE	RXFTCFG			TCBG TIE	TXFTIE	WUFIE	WUS[2:0]		SCARCNT2:0]			Res.
rw			rw	rw			rw	rw	rw	rw	rw	rw	rw	rw	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DEP	DEM	DDRE	OVR DIS	ONE BIT	CTSIE	CTSE	RTSE	DMAT	DMAR	SCEN	NACK	HD SEL	IRLP	IREN	EIE
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:29 **TXFTCFG**：TXFIFO 阈值配置 (TXFIFO threshold configuration)
000：TXFIFO 达到其深度的 1/8
001：TXFIFO 达到其深度的 1/4
010：TXFIFO 达到其深度的 1/2
011：TXFIFO 达到其深度的 3/4
100：TXFIFO 达到其深度的 7/8
101：TXFIFO 变空
其余组合：保留

位 28 **RXFTIE**：RXFIFO 阈值中断使能 (RXFIFO threshold interrupt enable)
此位由软件置 1 和清零。
0：禁止中断
1：当接收 FIFO 达到 RXFTCFG 中编程的阈值时，生成 USART 中断。

位 27:25 **RXFTCFG**：接收 FIFO 阈值配置 (Receive FIFO threshold configuration)
000：接收 FIFO 达到其深度的 1/8
001：接收 FIFO 达到其深度的 1/4
010：接收 FIFO 达到其深度的 1/2
011：接收 FIFO 达到其深度的 3/4
100：接收 FIFO 达到其深度的 7/8
101：接收 FIFO 已满
其余组合：保留

位 24 **TCBGTIE**：保护时间前发送完成中断使能 (Transmission Complete before guard time, interrupt enable)
此位由软件置 1 和清零。
0：禁止中断
1：当 USART_ISR 寄存器中的 TCBGT=“1”时，生成 USART 中断
注：　如果 USART 不支持智能卡模式，该位保留并由硬件强制清零。请参见第 1851 页的第 48.4 节：USART 实现。

位 23 **TXFTIE**：TXFIFO 阈值中断使能 (TXFIFO threshold interrupt enable)
此位由软件置 1 和清零。
0：禁止中断
1：当 TXFIFO 达到 TXFTCFG 中编程的阈值时，生成 USART 中断。

位 22 **WUFIE**：从低功耗模式唤醒中断使能 (Wakeup from low-power mode interrupt enable)
此位由软件置 1 和清零。
0：禁止中断
1：当 USART_ISR 寄存器中的 WUF=“1”时，生成 USART 中断
注：　WUFIE 必须在进入低功耗模式前置 1。
WUF 中断仅在低功耗模式下有效。
如果 USART 不支持从停止模式唤醒功能，则此位保留并由硬件强制清零。
请参见第 1851 页的第 48.4 节：USART 实现。

位 21:20 **WUS[1:0]**：从低功耗模式唤醒中断标志选择 (Wakeup from low-power mode interrupt flag selection)

该位域用于指定激活 WUF（从低功耗模式唤醒标志）的事件。
00：WUF 在地址匹配时激活（按 ADD[7:0] 和 ADDM7 所定义）
01：保留。
10：WUF 在起始位检测时激活。
11：WUF 在 RXNE/RXFNE 时激活。
只有在禁止 USART（UE=“0”）时才能写入此位域。

如果 USART 不支持从停止模式唤醒功能，则此位保留并由硬件强制清零。
请参见第 1851 页的第 48.4 节：USART 实现。

位 19:17 **SCARCNT[2:0]**：智能卡自动重试计数 (Smartcard auto-retry count)

此位域用于指定智能卡模式下发送和接收的重试次数。
在发送模式下，此位域用于指定生成发送错误（FE 位置 1）前自动重新发送的重试次数。
在接收模式下，此位域用于指定生成接收错误（RXNE/RXFNE 位和 PE 位置 1）前错误接收尝试的次数。
只有在禁止 USART（UE=“0”）时才能编程此位域。
使能 USART（UE=“1”）时，此位域只能写入 0x0，以停止重新发送。
0x0：禁止重新发送——发送模式下不会自动重新发送。
0x1 到 0x7：自动重新发送的尝试次数（发出错误信号前）

注：　如果不支持智能卡模式，该位保留并由硬件强制清零。
请参见第 1851 页的第 48.4 节：USART 实现。

位 16 保留，必须保持复位值。

位 15 **DEP**：驱动器使能极性选择 (Driver enable polarity selection)

0：DE 信号高电平有效。
1：DE 信号低电平有效。
只有在禁止 USART（UE=“0”）时才能写入该位。

注：　如果不支持驱动器使能功能，该位保留并且必须保持清零。
请参见第 1851 页的第 48.4 节：USART 实现。

位 14 **DEM**：驱动器使能模式 (Driver enable mode)

此位用于通过 DE 信号激活外部收发器控制。
0：禁止 DE 功能。
1：使能 DE 功能。DE 信号在 RTS 引脚上输出。
只有在禁止 USART（UE=“0”）时才能写入该位。

注：　如果不支持驱动器使能功能，该位保留并且必须保持清零。
请参见第 1851 页的第 48.4 节：USART 实现.

位 13 **DDRE**：接收出错时的 DMA 禁止 (DMA Disable on Reception Error)

0：接收出错时不禁止 DMA。相应的错误标志置 1，但 RXNE 保持为 0 以防止上溢。因此，将不使能 DMA 请求，从而不会传送错误数据（无 DMA 请求），但会传送接收到的下一个正确数据。（用于智能卡模式）
1：接收出错后禁止 DMA。相应的错误标志以及 RXNE 均置 1。屏蔽 DMA 请求，直到错误标志清零。这意味着软件必须首先禁止 DMA 请求（DMAR = “0”）或者将 RXNE（使能 FIFO 模式时为 RXFNE）清零，然后再将错误标志清零。
只有在禁止 USART（UE=“0”）时才能写入该位。

注：　接收错误包括：奇偶校验错误、帧错误或噪声错误。

位 12 **OVRDIS**：上溢禁止 (Overrun Disable)

此位用于禁止接收上溢检测。

0：接收新数据前未读取已接收的数据时，上溢错误标志 ORE 置 1。

1：禁止上溢功能。如果在 RXNE 标志仍置 1 时接收到新数据，则 ORE 标志不会置 1，且新接收的数据会覆盖 USART_RDR 寄存器之前的内容。使能 FIFO 模式时，RXFIFO 将被旁路，数据将直接写入 USART_RDR 寄存器中。即使在使能 FIFO 管理时，也将使用 RXNE 标志。

只有在禁止 USART（UE=“0”）时才能写入该位。

注： 此控制位用于检查通信流而不会读取数据

位 11 **ONEBIT**：一个采样位方法使能 (One sample bit method enable)

该位允许用户选择采样方法。选择一个采样位方法后，将禁止噪声检测标志 (NE)。

0：三个采样位方法

1：一个采样位方法

只有在禁止 USART（UE=“0”）时才能写入该位。

位 10 **CTSIE**：CTS 中断使能 (CTS interrupt enable)

0：禁止中断

1：当 USART_ISR 寄存器中的 CTSIF =“1”时，生成中断

注： 如果不支持硬件流控制功能，该位保留并由硬件强制清零。
请参见第 1851 页的第 48.4 节：USART 实现。

位 9 **CTSE**：CTS 使能 (CTS enable)

0：禁止 CTS 硬件流控制

1：使能 CTS 模式，仅当 nCTS 输入有效（连接到 0）时才发送数据。如果在发送数据时使 nCTS 输入无效，会在停止之前完成发送。如果使 nCTS 有效时数据已写入数据寄存器，则将延迟发送，直到 nCTS 有效。

只有在禁止 USART（UE=“0”）时才能写入该位。

注： 如果不支持硬件流控制功能，该位保留并由硬件强制清零。
请参见第 1851 页的第 48.4 节：USART 实现。

位 8 **RTSE**：RTS 使能 (RTS enable)

0：禁止 RTS 硬件流控制

1：使能 RTS 输出，仅当接收缓冲区中有空间时才会请求数据。发送完当前字符后应停止发送数据。可以接收数据时使 nRTS 输出有效（拉至 0）。

只有在禁止 USART（UE=“0”）时才能写入该位。

注： 如果不支持硬件流控制功能，该位保留并由硬件强制清零。
请参见第 1851 页的第 48.4 节：USART 实现。

位 7 **DMAT**：DMA 使能发送器 (DMA enable transmitter)

该位由软件置 1/复位。

1：针对发送使能 DMA 模式

0：针对发送禁止 DMA 模式

位 6 **DMAR**：DMA 使能接收器 (DMA enable receiver)

该位由软件置 1/复位。

1：针对接收使能 DMA 模式

0：针对接收禁止 DMA 模式

位 5 **SCEN**：智能卡模式使能 (Smartcard mode enable)

该位用于使能智能卡模式。

0：禁止智能卡模式

1：使能智能卡模式

只有在禁止 USART（UE=“0”）时才能写入此位域。

注：　如果 USART 不支持智能卡模式，该位保留并由硬件强制清零。请参见第 1851 页的第 48.4 节：USART 实现。

位 4 **NACK**：智能卡 NACK 使能 (Smartcard NACK enable)

0：出现奇偶校验错误时禁止 NACK 发送

1：出现奇偶校验错误时使能 NACK 发送

只有在禁止 USART（UE=“0”）时才能写入此位域。

注：　如果 USART 不支持智能卡模式，该位保留并由硬件强制清零。请参见第 1851 页的第 48.4 节：USART 实现。

位 3 **HDSEL**：半双工选择 (Half-duplex selection)

选择单线半双工模式

0：未选择半双工模式

1：选择半双工模式

只有在禁止 USART（UE=“0”）时才能写入该位。

位 2 **IRLP**：IrDA 低功耗模式 (IrDA low-power)

该位用于选择正常模式和低功耗 IrDA 模式

0：正常模式

1：低功耗模式

只有在禁止 USART（UE=“0”）时才能写入该位。

注：　如果不支持 IrDA 模式，该位保留并由硬件强制清零。请参见第 1851 页的第 48.4 节：USART 实现。

位 1 **IREN**：IrDA 模式使能 (IrDA mode enable)

此位由软件置 1 和清零。

0：禁止 IrDA

1：使能 IrDA

只有在禁止 USART（UE=“0”）时才能写入该位。

注：　如果不支持 IrDA 模式，该位保留并由硬件强制清零。请参见第 1851 页的第 48.4 节：USART 实现。

位 0 **EIE**：错误中断使能 (Error interrupt enable)

如果出现帧错误、上溢错误、噪声标志或 SPI 从器件下溢错误（USART_ISR 寄存器中的 FE=“1”、ORE=“1”、NE=“1”或 UDR = “1”），则需要使用错误中断使能位来使能中断生成。

0：禁止中断

1：USART_ISR 寄存器中的 FE=“1”、ORE=“1”、NE=“1”或 UDR = “1”（在 SPI 从器件模式下）时，生成中断。

48.7.4 USART 波特率寄存器 (USART_BRR)

USART baud rate register

只有在禁止 USART（UE=“0”）时才能写入此寄存器。在自动波特率检测模式下，该位由硬件自动更新。

偏移地址：0x0C

复位值：0x0000

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

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

位 31:16 保留，必须保持复位值。

位 15:4 **BRR[15:4]**

BRR[15:4] = USARTDIV[15:4]

位 3:0 **BRR[3:0]**

当 OVER8 = “0”时，BRR[3:0] = USARTDIV[3:0]。
当 OVER8 = “1”时：
BRR[2:0] = USARTDIV[3:0]，右移 1 位。
BRR[3] 必须保持清零。

48.7.5 USART 保护时间和预分频器寄存器 (USART_GTPR)

USART guard time and prescaler register

偏移地址：0x10

复位值：0x0000

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

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

位 31:16 保留，必须保持复位值

位 15:8 **GT[7:0]**：保护时间值 (Guard time value)

此位域用于编程保护时间值（以波特时钟周期数为单位）。

该位用于智能卡模式。经过此保护时间后，发送完成标志置 1。

只有在禁止 USART（UE=“0”）时才能写入此位域。

注： 如果不支持智能卡模式，该位保留并由硬件强制清零。请参见第 1851 页的第 48.4 节：USART 实现。

位 7:0 **PSC[7:0]**：预分频器值 (Prescaler value)

在 IrDA 低功耗和正常的 IrDA 模式下：

PSC[7:0] = IrDA 正常和低功耗波特率

用于编程预分频器，进行 USART 源时钟分频以获得低功耗频率：

使用寄存器中给出的值（8 个有效位）对源时钟进行分频：

00000000：保留 - 不编程此值

00000001：源时钟 1 分频

00000010：源时钟 2 分频

...

在智能卡模式下：

PSC[4:0]：预分频器值 (Prescaler value)

用于编程预分频器，进行 USART 源时钟分频以提供智能卡时钟。

将寄存器中给出的值（5 个有效位）乘以 2 得出源时钟频率的分频系数：

00000：保留 - 不编程此值

00001：源时钟 2 分频

00010：源时钟 4 分频

00011：源时钟 6 分频

...

只有在禁止 USART（UE=“0”）时才能写入此位域。

注： 如果使用智能卡模式，则位 [7:5] 必须保持清零。

不支持智能卡和 IrDA 模式时，该位域保留并由硬件强制清零。请参见第 1851 页的第 48.4 节：USART 实现。

48.7.6 USART 接收器超时寄存器 (USART_RTOR)

USART receiver timeout register

偏移地址：0x14

复位值：0x0000

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

位 31:24 **BLEN[7:0]**：块长度 (Block Length)

此位域用于提供智能卡 T=“1”接收下的块长度。其值等于信息字符数 + 结尾字段的长度 (1-LEC/2-CRC) - 1。

例如：

BLEN = 0 -> 0 个信息字符 + LEC

BLEN = 1 -> 0 个信息字符 + CRC

BLEN = 255 -> 254 个信息字符 + CRC（总共 256 个字符）

在智能卡模式下，块长度计数器在 TXE=“0”（使能 FIFO 模式时为 TXFE =“0”）时复位。

此位域也可用于其他模式。在这种情况下，块长度计数器在 RE=“0”（禁止接收器）和/或 EOBCF 位写入 1 时复位。

注： 块接收开始后可编程此值（使用起始字段中 LEN 字符中的数据）。每个接收到的块只能对此值编程一次。

位 23:0 **RTO[23:0]**：接收器超时值 (Receiver timeout value)

此位域用于提供接收器的超时值（以波特时钟数为单位）。

在标准模式下，如果在接收到最后一个字符后，在 RTO 值对应的时间内未检测到新的起始位，则 RTOF 标志置 1。

在智能卡模式下，此值用于实施 CWT 和 BWT。有关更多详细信息，请参见智能卡章节。在标准模式下，从接收到的最后一个字符的起始位开始执行 CWT/BWT 测量。

注： 每个接收到的字符只能对此值编程一次。

注： 可以实时写入 RTOR。如果新值小于或等于计数器的值，RTOF 标志置 1。

如果不支持接收器超时功能，此寄存器保留并由硬件强制为“0x00000000”。

请参见第 1851 页的第 48.4 节：USART 实现。

48.7.7 USART 请求寄存器 (USART_RQR)

USART request register

偏移地址：0x18

复位值：0x0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TXFRQ	RXFRQ	MMRQ	SBKRQ	ABRRQ
											w_r0	w_r0	w_r0	w_r0	w_r0

位 31:5 保留，必须保持复位值

位 4 **TXFRQ**：发送数据刷新请求 (Transmit data flush request)

禁止 FIFO 模式时，向该位写入“1”会将 TXE 标志置 1。这可丢弃发送数据。由于错误 (NACK) 而未发送数据且 USART_ISR 寄存器中的 FE 标志有效时，只能在智能卡模式下使用此位。如果 USART 不支持智能卡模式，该位保留并由硬件强制清零。

使能 FIFO 时，TXFRQ 位置 1 以清空整个 FIFO。这会将 TXFE 标志（发送 FIFO 为空，USART_ISR 寄存器中的位 23）置位。在 UART 模式和智能卡模式下都支持清空发送 FIFO。

注： 在 FIFO 模式下，TXFNF 标志在清空请求期间复位，直到 TxFIFO 为空，以确保数据寄存器中没有写入数据。

位 3 **RXFRQ**：接收数据刷新请求 (Receive data flush request)

向该位写入“1”将清空整个接收 FIFO（即，将 RXFNE 位清零）。

这可以丢弃接收的数据而不对其执行读取操作，并避免发生上溢情况。

位 2 **MMRQ**：静默模式请求 (Mute mode request)

向此位写入“1”可将 USART 置于静默模式，并将 RWU 标志复位。

位 1 **SBKRQ**：发送中断请求 (Send break request)

向此位写入“1”可将 SBKF 标志置 1 并在发送设备可用后立即请求在线路上发送 BREAK。

注： 如果应用需要在之前插入的所有数据（包括尚未发送的数据）后发送 break 字符，软件应等到 TXE 标志使能后将 SBKRQ 位置 1。

位 0 **ABRRQ**：自动波特率请求 (Auto baud rate request)

向此位写入“1”可复位 USART_ISR 中的 ABRF 标志，并请求对下一个接收到的数据帧进行自动波特率测量。

注： 如果 USART 不支持自动波特率功能，则此位保留并由硬件强制清零。
请参见第 1851 页的第 48.4 节：USART 实现。

48.7.8 USART 中断和状态寄存器 (USART_ISR)

USART interrupt and status register

偏移地址：0x1C

复位值：0x0000 00C0（如果禁止 FIFO）。

复位值：0x0280 00C0（如果使能 FIFO/ 智能卡模式）。

复位值：0x0080 00C0（如果使能 FIFO 并禁止智能卡模式）。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	TXFT	RXFT	TCBGT	RXFF	TXFE	RE ACK	TE ACK	WUF	RWU	SBKF	CMF	BUSY
				r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ABRF	ABRE	UDR	EOBF	RTOF	CTS	CTSIF	LBDF	TXE/TX FNF	TC	RXNE/ RXFNE	IDLE	ORE	NE	FE	PE
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:28 保留，必须保持复位值。

位 27 **TXFT**：TXFIFO 阈值标志 (TXFIFO threshold flag)

当 TXFIFO 达到在 USART_CR3 寄存器的 TXFTCFG 中编程的阈值时（即 TXFIFO 包含 TXFTCFG 个空位置），该位由硬件置 1。如果 USART_CR3 寄存器中的 TXFTIE 位 = “1”（位 31），则会生成中断。

0：TXFIFO 未达到编程的阈值。

1：TXFIFO 已达到编程的阈值。

位 26 **RXFT**：RXFIFO 阈值标志 (RXFIFO threshold flag)

达到在 USART_CR3 寄存器的 RXFTCFG 中编程的阈值时，该位由硬件置 1。这意味着，接收 FIFO 中有 (RXFTCFG - 1) 个数据，USART_RDR 寄存器中有一个数据。如果 USART_CR3 寄存器中的 RXFTIE 位 = “1”（位 27），则会生成中断。

0：接收 FIFO 未达到编程的阈值。

1：接收 FIFO 已达到编程的阈值。

注： 当 RXFTCFG 阈值配置为 “101” 时，如果存在 16 个数据（即 RXFIFO 中有 15 个数据，USART_RDR 中有 1 个数据），则 RXFT 标志将置 1。因此，接收到的第 17 个数据不会导致上溢错误。接收到第 18 个数据后会发生上溢错误。

位 25 **TCBGT**：保护时间前发送完成标志 (Transmission complete before guard time flag)

当写入到 USART_TDR 中的最后一个数据已正确从移位寄存器中发出时，该位置 1。

如果包含数据的帧完成发送，并且智能卡未发回任何 NACK，则该位在智能卡模式下由硬件置 1。如果 USART_CR3 寄存器中的 TCBGTIE = “1”，则生成中断。

此位由软件清零，方法是向 USART_ICR 寄存器中的 TCBGTCF 写入 “1” 或向 USART_TDR 寄存器执行写操作。

0：发送未完成或发送未成功完成（即，从智能卡接收到 NACK）。

1：发送成功完成（在保护时间结束之前完成，智能卡未发送 NACK）。

注： 如果 USART 不支持智能卡模式，该位保留并由硬件强制清零。如果 USART 支持智能卡模式并使能智能卡模式，则 TCBGT 复位值为 “1”。请参见第 1851 页的第 48.4 节：USART 实现。

位 24 **RXFF**：RXFIFO 已满 (RXFIFO Full)

当接收到的数据量对应于 RXFIFO 大小 + 1（RXFIFO 已满 + USART_RDR 寄存器中的 1 个数据）时，该位由硬件置 1。

如果 USART_CR1 寄存器中的 RXFFIE 位 = “1”，则会生成中断。

0：RXFIFO 未满。

1：RXFIFO 已满。

位 23 **TXFE**：TXFIFO 为空 (TXFIFO Empty)

当 TXFIFO 为空时，该位由硬件置 1。当 TXFIFO 包含至少一个数据时，该标志被清零。也可以通过向 USART_RQR 寄存器中的位 TXFRQ（位 4）写入 “1” 将 TXFE 标志置 1。

如果 USART_CR1 寄存器中的 TXFEIE 位 = “1”（位 30），则会生成中断。

0：TXFIFO 非空。

1：TXFIFO 为空。

位 22 **REACK**：接收使能确认标志 (Receive enable acknowledge flag)

USART 采用接收使能值时，通过硬件将此位置 1/复位。

此位可用于验证 USART 是否准备好在进入低功耗模式前接收数据。

注： 如果 USART 不支持从停止模式唤醒功能，则此位保留并由硬件强制清零。请参见第 1851 页的第 48.4 节：USART 实现。

位 21 **TEACK**：发送使能确认标志 (Transmit enable acknowledge flag)

USART 采用发送使能值时，通过硬件将此位置 1/复位。

通过写入 TE= “0” 生成空闲帧请求，然后在 USART_CR1 寄存器中写入 TE= “1” 以遵循 TE= “0” 最短周期时，可使用此位。

位 20 **WUF**：从低功耗模式唤醒标志 (Wakeup from low-power mode flag)

当检测到唤醒事件时，此位由硬件置 1。事件通过 WUS 位域定义。此位由软件清零，方法是向 USART_ICR 寄存器中的 WUCF 写入 1。

如果 USART_CR3 寄存器中的 WUFIE=“1”，则会生成中断。

注：　当 UESM 清零时，WUF 标志也清零。

WUF 中断仅在低功耗模式下有效。

如果 USART 不支持从停止模式唤醒功能，则此位保留并由硬件强制清零。

请参见第 1851 页的第 48.4 节：USART 实现。

位 19 **RWU**：*接收器从静默模式唤醒 (Receiver wakeup from Mute mode)*

该位指示 USART 是否处于静默模式。当识别出唤醒/静默序列时，此位由硬件清零/置 1。静默模式控制序列（地址或 IDLE）通过 USART_CR1 寄存器中的 WAKE 位进行选择。

当选择 IDLE 模式下唤醒时，该位只能通过用软件向 USART_RQR 寄存器中的 MMRQ 位写入“1”的方式置 1。

0：接收器处于工作模式

1：接收器处于静默模式

注：　如果 USART 不支持从停止模式唤醒功能，则此位保留并由硬件强制清零。

请参见第 1851 页的第 48.4 节：USART 实现。

位 18 **SBKF**：发送中断标志 (Send break flag)

此位指示已请求发送 break 字符。此位由软件置 1，方法是向 USART_CR3 寄存器中的 SBKRQ 位写入“1”。此位在中断发送的停止位期间由硬件自动复位。

0：不发送 break 字符

1：将发送 break 字符

位 17 **CMF**：字符匹配标志 (Character match flag)

接收到由 ADD[7:0] 定义的字符后由硬件将此位置 1。此位由软件清零，方法是向 USART_ICR 寄存器中的 CMCF 写入“1”。

如果 USART_CR1 寄存器中的 CMIE=“1”，则会生成中断。

0：未检测到字符匹配

1：检测到字符匹配

位 16 **BUSY**：忙标志 (Busy flag)

此位由硬件置 1 和复位。当 RX 线路上正在进行通信（成功检测到起始位）时有效。在接收结束（成功或失败）时复位。

0：USART 处于空闲状态（无接收）

1：正在接收

位 15 **ABRF**：自动波特率标志 (Auto baud rate flag)

已设置自动波特率（RXNE 也将置 1，在 RXNEIE =“1”时生成中断），或者自动波特率操作未成功完成时，此位由硬件置 1（ABRE=“1”）（此时，ABRE、RXNE 和 FE 也置 1）

为请求新的自动波特率检测，通过向 USART_RQR 寄存器中的 ABRRQ 写入“1”，由软件将此位清零。

注：　如果 USART 不支持自动波特率功能，则此位保留并由硬件强制清零。

位 14 **ABRE**：自动波特率错误 (Auto baud rate error)

如果波特率测量失败（波特率超出范围或字符比较失败），此位由硬件置 1。

此位由软件清零，方法是向 USART_CR3 寄存器中的 ABRRQ 位写入“1”。

注：　如果 USART 不支持自动波特率功能，则此位保留并由硬件强制清零。

位 13 **UDR**：SPI 从器件下溢错误标志 (SPI slave underrun error flag)

在从发送模式下，如果在软件尚未将任何值加载到 USART_TDR 时出现第一个数据发送时钟脉冲，此标志将置 1。该标志通过将 USART_ICR 寄存器中的 UDRCF 位置 1 来复位。

0：无下溢错误

1：存在下溢错误

注：　如果 USART 不支持 SPI 从器件模式，则该位保留并由硬件强制清零。请参见第 1851 页的第 48.4 节：USART 实现。

位 12 **EOBF**：块结束标志 (End of block flag)

接收到完整块后，此位由硬件置 1（例如 T=“1”智能卡模式）。接收到的字节数（从块的起始处，包括起始字段）等于或大于 BLEN + 4 时执行检测。

如果 USART_CR2 寄存器中的 EOBIE =“1”，则会生成中断。

此位由软件清零，方法是向 USART_ICR 寄存器中的 EOBCF 写入“1”。

0：未达到块结束

1：已达到块结束（字符数）

注：　如果不支持智能卡模式，该位保留并由硬件强制清零。请参见第 1851 页的第 48.4 节：USART 实现。

位 11 **RTOF**：接收器超时 (Receiver timeout)

已经过在 RTOR 寄存器中编程的超时值后，如果无任何通信，此位由硬件置 1。此位由软件清零，方法是向 USART_ICR 寄存器中的 RTOCF 位写入“1”。

如果 USART_CR2 寄存器中的 RTOIE=“1”，则会生成中断。

在智能卡模式下，该超时对应于 CWT 或 BWT 时间。

0：未达到超值值

1：已达到超时值，未接收到任何数据

注：　如果 RTOR 寄存器中编程的时间值将 2 个字符隔开，则 RTOF 不置 1。如果此时间大于该值 + 2 个采样时间（2/16 或 2/8，具体取决于过采样方法），则 RTOF 标志置 1。即使 RE =“0”，计数器仍会计数，但 RTOF 仅在 RE =“1”时置 1。如果 RE 置 1 时已经超时，则 RTOF 将置 1。

如果 USART 不支持接收器超时功能，该位保留并由硬件强制清零。

位 10 **CTS**：CTS 标志 (CTS flag)

此位由硬件置 1/复位。此位是对 nCTS 输入引脚的状态取反。

0：nCTS 线置 1

1：nCTS 线复位

注：　如果不支持硬件流控制功能，该位保留并由硬件强制清零。

位 9 **CTSIF**：CTS 中断标志 (CTS interrupt flag)

如果 CTSE 位置 1，当 nCTS 输入切换时，此位由硬件置 1。此位由软件清零，方法是向 USART_ICR 寄存器中的 CTSCF 位写入“1”。

如果 USART_CR3 寄存器中的 CTSIE=“1”，则会生成中断。

0：nCTS 状态线上未发生变化

1：nCTS 状态线上发生变化

注：　如果不支持硬件流控制功能，该位保留并由硬件强制清零。

位 8 **LBDF**：LIN 中断检测标志 (LIN break detection flag)

检测到 LIN 中断时，该位由硬件置 1。此位由软件清零，方法是向 USART_ICR 中的 LBDCF 写入“1”。

如果 USART_CR2 寄存器中的 LBDIE =“1”，则会生成中断。

0：未检测到 LIN 中断

1：检测到 LIN 中断

注：　如果 USART 不支持 LIN 模式，该位保留并由硬件强制清零。请参见第 1851 页的第 48.4 节：USART 实现。

位 7 **TXE/TXFNF**：发送数据寄存器为空/TXFIFO 未满 (Transmit data register empty/TXFIFO not full)

如果禁止 FIFO 模式，则当 USART_TDR 寄存器的内容已传输到移位寄存器时，TXE 由硬件置 1。通过对 USART_TDR 寄存器执行写操作将该位清零。为丢弃数据（仅在智能卡 T=0 模式下出现发送故障时），也可以通过向 USART_RQR 寄存器中的 TXFRQ 写入“1”来将 TXE 标志置 1。

如果使能 FIFO 模式，TXFNF 会在 TXFIFO 未满时由硬件置 1，表示可向 USART_TDR 中写入数据。每次对 USART_TDR 进行写操作都会将数据置于 TXFIFO 中。该标志保持置 1，直到 TXFIFO 已满。当 TXFIFO 已满时，该标志清零，表示不能向 USART_TDR 中写入数据。

如果 USART_CR1 寄存器中的 TXEIE/TXFNFIE 位 =“1”，则会生成中断。

0：数据寄存器已满/发送 FIFO 已满。

1：数据寄存器/发送 FIFO 未满。

注： *在清空请求期间，TXFNF 保持复位，直到 TXFIFO 为空。发送清空请求（通过将 TXFRQ 位置 1）后，应先检查 TXFNF 标志，然后再写入 TXFIFO（TXFNF 和 TXFE 将同时置 1）。*

单缓冲区发送期间使用该位。

位 6 **TC**：发送完成 (Transmission complete)

该位指示写入到 USART_TDR 中的最后一个数据已从移位寄存器中发出。

如果已完成对包含数据的帧的发送并且 TXE/TXFE 置 1，则此位由硬件置 1。

如果 USART_CR1 寄存器中的 TCIE =“1”，则会生成中断。

TC 位由软件清零，方法是向 USART_ICR 寄存器中的 TCCF 写入“1”或向 USART_TDR 寄存器执行写操作。

0：传送未完成

1：传送已完成

注： *如果 TE 位复位且无任何发送正在进行，TC 位会立即置 1。*

位 5 **RXNE/RXFNE**：读取数据寄存器非空/RXFIFO 非空 (Read data register not empty/RXFIFO not empty)

当 USART_RDR 移位寄存器的内容已传输到 USART_RDR 寄存器时，RXNE 位由硬件置 1。通过对 USART_RDR 寄存器执行读取操作将该位清零。也可以通过向 USART_RQR 寄存器中的 RXFRQ 写入“1”将 RXNE 标志清零。

RXFIFO 非空时，RXFNE 位由硬件置 1，这表示可以从 USART_RDR 寄存器读取数据。每次对 USART_RDR 进行读操作都会在 RXFIFO 中释放一个位置。

RXNE/RXFNE 在 RXFIFO 为空时清零。也可以通过向 USART_RQR 寄存器中的 RXFRQ 写入“1”将 RXNE/RXFNE 标志清零。

如果 USART_CR1 寄存器中的 RXNEIE/RXFNEIE=“1”，则会生成中断。

0：未接收到数据

1：已准备好读取接收到的数据

位 4 **IDLE**：检测到空闲线路 (IDLE line detected)

检测到空闲线路时，该位由硬件置 1。如果 USART_CR1 寄存器中的 IDLEIE=“1”，则会生成中断。此位由软件清零，方法是向 USART_ICR 寄存器中的 IDLECF 写入“1”。

0：未检测到空闲线路

1：检测到空闲线路

注： *直到 RXNE 位已置 1 时（即，当出现新的空闲线路时）IDLE 位才会被再次置 1。*

使能静默模式（MME=“1”）后，如果 USART 未静默（RWU=“0”），则 IDLE 置 1，无论是否通过 WAKE 位选择了静默模式。如果 RWU=“1”，IDLE 不置 1。

位 3 **ORE**：溢出错误 (Overrun error)

在 RXNE =“1”（使能 FIFO 模式时为 RXFF =“1”）的情况下，当移位寄存器中当前正在接收的数据。

准备好传输到 USART_RDR 寄存器时，此位由硬件置 1。此位由软件清零，方法是向 USART_ICR 寄存器中的 ORECF 写入“1”。

如果 USART_CR1 寄存器中的 RXNEIE/RXFNEIE=“1”或 EIE =“1”，则会生成中断。

0：无溢出错误

1：检测到溢出错误

注： *当此位置 1 时，USART_RDR 寄存器的内容不会丢失，但移位寄存器会被覆盖。EIE 位置 1 后，如果在多缓冲区通信中 ORE 标志置 1，则会生成中断。*

USART_CR3 寄存器中的 OVRDIS 位置 1 时，此位将被永久强制清零（无上溢检测）。

位 2 **NE**：噪声检测标志 (Noise detection flag)

当在接收的帧上检测到噪声时，该位由硬件置 1。此位由软件清零，方法是向 USART_ICR 寄存器中的 NFCF 位写入“1”。

0：未检测到噪声

1：检测到噪声

注： *该位不会生成中断，因为该位出现的时间与本身生成中断的 RXNE/RXFNE 位出现的时间相同。EIE 位置 1 后，如果在多缓冲区通信中 NE 标志置 1，则会生成中断。*

当线路无噪声时，可以通过将 ONEBIT 位编程为 1 提高 USART 对偏差的容差来禁止 NE 标志（请参见第 1867 页的第 48.5.8 节：USART 接收器对时钟偏差的容差）。

在 FIFO 模式下，此错误与 USART_RDR 中的字符相关联。

位 1 **FE**：帧错误 (Framing error)

当检测到去同步化、过度的噪声或 break 字符时，该位由硬件置 1。此位由软件清零，方法是向 USART_ICR 寄存器中的 FECF 位写入“1”。

在智能卡模式下发送数据时，如果在达到最大发送尝试次数后仍未成功（智能卡向数据帧发送 NACK 信号），则此位置 1。

如果 USART_CR1 寄存器中 EIE = 1，则会生成中断。

0：未检测到帧错误

1：检测到帧错误或 break 字符

注： *在 FIFO 模式下，此错误与 USART_RDR 中的字符相关联。*

位 0 **PE**：奇偶校验错误 (Parity error)

当在接收器模式下发生奇偶校验错误时，该位由硬件置 1。此位由软件清零，方法是向 USART_ICR 寄存器中的 PECF 写入“1”。

如果 USART_CR1 寄存器中的 PEIE =“1”，则会生成中断。

0：无奇偶校验错误

1：奇偶校验错误

注： *在 FIFO 模式下，此错误与 USART_RDR 中的字符相关联。*

48.7.9 USART 中断标志清零寄存器 (USART_ICR)

USART interrupt flag clear register

偏移地址：0x20

复位值：0x0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WUCF	Res.	Res.	CMCF	Res.
											w			w	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	UDRCF	EOBCF	RTOCF	Res.	CTSCF	LBDCF	TCBGTCF	TCCF	TXFECF	IDLECF	ORECF	NECF	FECF	PECF
		w	w	w		w	w	w	w	w	w	w	w	w	w

位 31:21 保留，必须保持复位值。

位 20 **WUCF**：从低功耗模式唤醒清零标志 (Wakeup from low-power mode clear flag)

向此位写入“1”时，USART_ISR 寄存器中的 WUF 标志将清零。

注： 如果 USART 不支持从停止模式唤醒功能，则此位保留并由硬件强制清零。请参见第 1851 页的第 48.4 节：USART 实现。

位 19:18 保留，必须保持复位值。

位 17 **CMCF**：字符匹配清零标志 (Character match clear flag)

向此位写入“1”时，USART_ISR 寄存器中的 CMF 标志将清零。

位 16:14 保留，必须保持复位值。

位 13 **UDRCF**：SPI 从器件下溢清零标志 (SPI slave underrun clear flag)

向此位写入“1”时，USART_ISR 寄存器中的 UDRF 标志将清零。

注： 如果 USART 不支持 SPI 从器件模式，则该位保留并由硬件强制清零。请参考第 1851 页的第 48.4 节：USART 实现。

位 12 **EOBCF**：块结束清零标志 (End of block clear flag)

向此位写入“1”时，USART_ISR 寄存器中的 EOBF 标志将清零。

注： 如果 USART 不支持智能卡模式，该位保留并由硬件强制清零。请参见第 1851 页的第 48.4 节：USART 实现。

位 11 **RTOCF**：接收器超时清零标志 (Receiver timeout clear flag)

向此位写入“1”时，USART_ISR 寄存器中的 RTOF 标志将清零。

注： 如果 USART 不支持接收器超时功能，该位保留并由硬件强制清零。请参见第 1851 页的第 48.4 节：USART 实现。

位 10 保留，必须保持复位值。

位 9 **CTSCF**：CTS 清零标志 (CTS clear flag)

向此位写入“1”时，USART_ISR 寄存器中的 CTSIF 标志将清零。

注： 如果不支持硬件流控制功能，该位保留并由硬件强制清零。请参见第 1851 页的第 48.4 节：USART 实现。

位 8 **LBDCF**：LIN 中断检测清零标志 (LIN break detection clear flag)

向此位写入“1”时，USART_ISR 寄存器中的 LBDF 标志将清零。

注： 如果 LIN 不支持智能卡模式，该位保留并由硬件强制清零。请参见第 1851 页的第 48.4 节：USART 实现。

位 7 **TCBGTCF**：保护时间前发送完成清零标志 (Transmission complete before Guard time clear flag)

向此位写入“1”时，USART_ISR 寄存器中的 TCBGT 标志将清零。

位 6 **TCCF**：发送完成清零标志 (Transmission complete clear flag)

向此位写入“1”时，USART_ISR 寄存器中的 TC 标志将清零。

位 5 **TXFECF**：TXFIFO 为空清零标志 (TXFIFO empty clear flag)

向此位写入“1”时，USART_ISR 寄存器中的 TXFE 标志将清零。

位 4 **IDLECF**：检测到空闲线路清零标志 (Idle line detected clear flag)
向此位写入“1”时，USART_ISR 寄存器中的 IDLE 标志将清零。

位 3 **ORECF**：上溢错误清零标志 (Overrun error clear flag)
向此位写入“1”时，USART_ISR 寄存器中的 ORE 标志将清零。

位 2 **NECF**：检测到噪声清零标志 (Noise detected clear flag)
向此位写入“1”时，USART_ISR 寄存器中的 NE 标志将清零。

位 1 **FECF**：帧错误清零标志 (Framing error clear flag)
向此位写入“1”时，USART_ISR 寄存器中的 FE 标志将清零。

位 0 **PECF**：奇偶校验错误清零标志 (Parity error clear flag)
向此位写入“1”时，USART_ISR 寄存器中的 PE 标志将清零。

48.7.10 USART 接收数据寄存器 (USART_RDR)

USART receive data register

偏移地址：0x24

复位值：未定义

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

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

位 31:9 保留，必须保持复位值。

位 8:0 **RDR[8:0]**：接收数据值 (Receive data value)
包含接收到的数据字符。
RDR 寄存器在输入移位寄存器和内部总线之间提供了并行接口（请参见*图 562*）。
在使能奇偶校验位的情况下进行接收时，从 MSB 位中读取的值为接收到的奇偶校验位。

48.7.11 USART 发送数据寄存器 (USART_TDR)

USART transmit data register

偏移地址：0x28

复位值：未定义

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

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

位 31:9 保留，必须保持复位值。

位 8:0 **TDR[8:0]**：发送数据值 (Transmit data value)

包含要发送的数据字符。

USART_TDR 寄存器在内部总线和输出移位寄存器之间提供了并行接口（请参见*图 562*）。

在使能奇偶校验位的情况下（USART_CR1 寄存器中的 PCE 位被置 1）进行发送时，由于 MSB 的写入值（位 7 或位 8，具体取决于数据长度）会被奇偶校验位所取代，因此该值不起任何作用。

注： 只能在 TXE/TXFNF=“1”时写入此寄存器。

48.7.12 USART 预分频器寄存器 (USART_PRESC)

USART prescaler register

只有在禁止 USART（UE=“0”）时才能写入此寄存器。

偏移地址：0x2C

复位值：0x0000

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

位 31:4 保留，必须保持复位值。

位 3:0 **PRESCALER[3:0]**：时钟预分频器 (Clock prescaler)

USART 输入时钟可通过预分频系数进行分频：

0000：输入时钟未分频
0001：输入时钟 2 分频
0010：输入时钟 4 分频
0011：输入时钟 6 分频
0100：输入时钟 8 分频
0101：输入时钟 10 分频
0110：输入时钟 12 分频
0111：输入时钟 16 分频
1000：输入时钟 32 分频
1001：输入时钟 64 分频
1010：输入时钟 128 分频
1011：输入时钟 256 分频
其余组合：保留

注： 为 PRESCALER 编程不允许的值时，编程的预分频值将为“1011”，即输入时钟除以 256。

48.7.13 USART 寄存器映射

下表提供了 USART 寄存器映射和复位值。

表 381. USART 寄存器映射和复位值

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	USART_CR1	RXFFIE	TXFEIE	FIFOEN	M1	EOBIE	RTOIE	DEAT4	DEAT3	DEAT2	DEAT1	DEAT0	DEDT4	DEDT3	DEDT2	DEDT1	DEDT0	OVER8	CMIE	MME	M	WAKE	PCE	PS	PEIE	TXEIE	TCIE	RXNEIE	IDLEIE	TE	RE	UESM	UE
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x04	USART_CR2	ADD[7:4]				ADD[3:0]				RTOEN	ABRMOD1	ABRMOD0	ABREN	MSBFIRST	DATAINV	TXINV	RXINV	SWAP	LINEN	STOP [1:0]		CLKEN	CPOL	CPHA	LBCL	Res.	LBDIE	LBDL	ADDM7	DIS_NSS	Res.	Res.	SLVEN
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		0	0	0	0			0
0x08	USART_CR3	TXFT CFG			RXFTIE	RXFT CFG			TCBGTIE	TXFTIE	WUFIE	WUS [1:0]		SCAR CNT [2:0]			Res.	DEP	DEM	DDRE	OVRDIS	ONEBIT	CTSIE	CTSE	RTSE	DMAT	DMAR	SCEN	NACK	HDSEL	IRLP	IREN	EIE
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0C	USART_BRR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BRR[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x10	USART_GTPR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	GT[7:0]								PSC[7:0]							
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x14	USART_RTOR	BLEN[7:0]								RTO[23:0]																							
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x18	USART_RQR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TXFRQ	RXFRQ	MMRQ	SBKRQ	ABRRQ
	Reset value																												0	0	0	0	0
0x1C	USART_ISR	Res.	Res.	Res.	Res.	TXFT	RXFT	TCBGT	RXFF	TXFE	REACK	TEACK	WUF	RWU	SBKF	CMF	BUSY	ABRF	ABRE	UDR	EOBF	RTOF	CTS	CTSIF	LBDF	TXE	TC	RXNE	IDLE	ORE	NE	FE	PE
	Reset value					0	0	X	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
0x20	USART_ICR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WUCF	Res.	Res.	CMCF	Res.	Res.	Res.	UDRCF	EOBCF	RTOCF	Res.	CTSCF	LBDCF	TCBGTCF	TCCF	TXFECF	IDLECF	ORECF	NECF	FECF	PECF
	Reset value												0			0				0	0	0		0	0	0	0	0	0	0	0	0	0
0x24	USART_RDR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RDR[8:0]								
	Reset value																								0	0	0	0	0	0	0	0	0
0x28	USART_TDR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TDR[8:0]								
	Reset value																								0	0	0	0	0	0	0	0	0

表 381. USART 寄存器映射和复位值（续）

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x2C	USART_PRESC	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PRESCALER[3:0]			
	Reset value																													0	0	0	0

有关寄存器边界地址的信息，请参见*第 100 页的第 2.2.2 节*。

49 低功耗通用异步接收器 (LPUART)

本节介绍低功耗通用异步收发器 (LPUART)。

49.1 LPUART 简介

LPUART 是一种 UART，允许在有限功耗下双向 UART 通信。仅需 32.768 kHz LSE 时钟即可进行高达 9600 baud/s 的 UART 通信。当LPUART使用不同于LSE的时钟源，如HSI，系统时钟作为时钟源时，可以达到更高的波特率。

即使当微控制器处于低功耗模式，能耗极低时，LPUART 也会等待 UART 帧的到来。LPUART 包含所有必要的硬件支持，使在最小功耗下可以进行异步串行通信。

它支持半双工单线通信和调制解调器操作 (CTS/RTS)。

还支持多处理器通信。

DMA（直接存储器访问）可用于数据发送/接收。

49.2 LPUART 主要特性

- 全双工异步通信
- NRZ 标准格式（标记/空格）
- 可编程波特率
- 使用 32.768 kHz 时钟源时波特率为 300 baud/s 到 9600 baud/s
- 使用高频时钟源可实现更高的波特率
- 两个用于收发数据的内部 FIFO
  每个 FIFO 均可由软件使能/禁止，并且均带有用于指示 FIFO 状态的状态标志
- 双时钟域，带有独立于 PCLK 的外设专用内核时钟
- 数据字长度可编程（7 位、8 位或 9 位）
- 可编程的数据顺序，最先移位 MSB 或 LSB
- 停止位可配置（支持 1 个或 2 个停止位）
- 单线半双工通信
- 使用 DMA 实现连续通信
- 使用中央 DMA 在预留的 SRAM 缓冲区中收/发字节
- 发射器和接收器有单独的使能位
- 发送和接收的单独信号极性控制
- Tx/Rx 引脚配置可交换
- 调制解调器和 RS-485 收发器的硬件流控制
- 传输检测标志：
  - 接收缓冲区已满
  - 发送缓冲区为空
  - BUSY 标志和发送结束标志
- 奇偶校验控制：
  - 发送奇偶校验位
  - 检查接收的数据字节的奇偶性
- 四个错误检测标志：
  - 上溢错误
  - 噪声检测
  - 帧错误
  - 奇偶校验错误
- 具有标志的中断源
- 多处理器通信：从静默模式唤醒（通过空闲线检测或地址标记检测）

49.3 LPUART 功能说明

49.3.1 LPUART框图

图 589. LPUART框图

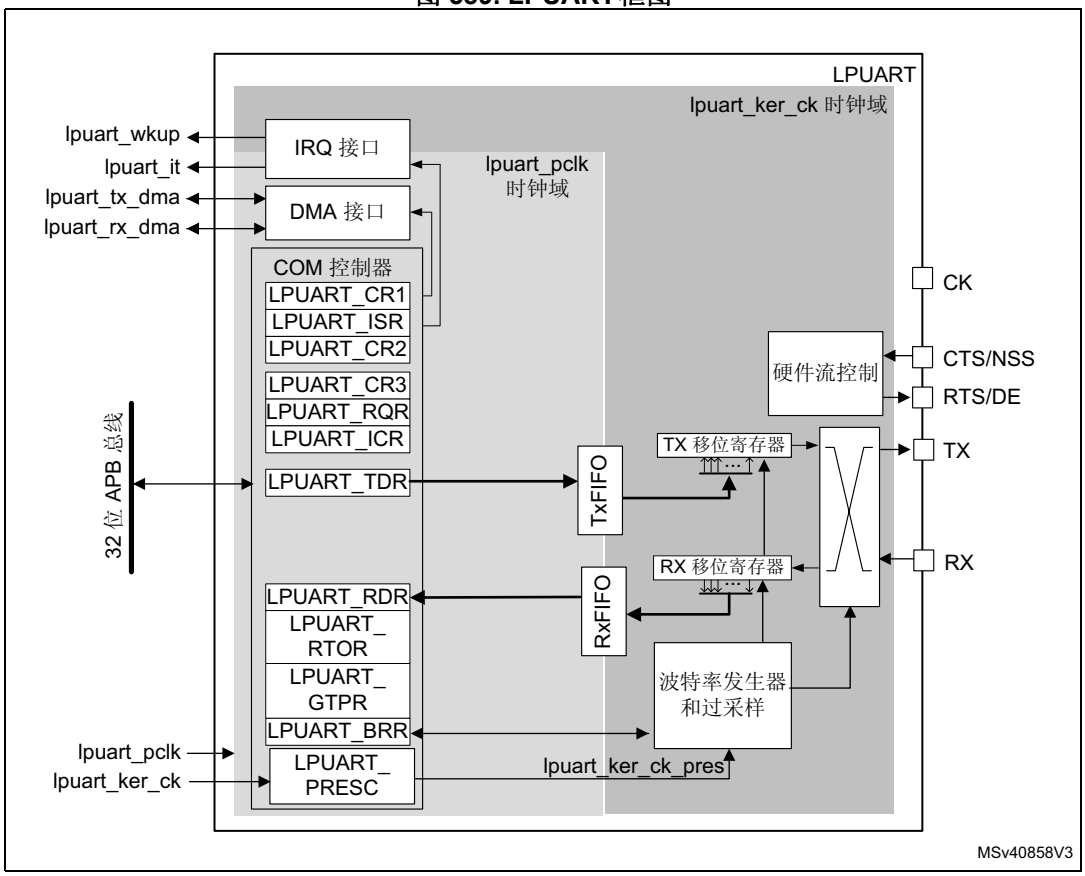


*图 589*中的简化框图显示的是两个完全独立的时钟域：

- **lpuart_pclk** 时钟域

  lpuart_pclk 时钟信号馈送外设总线接口。需要访问 LPUART 寄存器时，该信号必须有效。

- **lpuart_ker_ck** 内核时钟域

  lpuart_ker_ck 是 LPUART 时钟源。它独立于 **lpuart_pclk**，由 RCC 提供。因此，即使 **lpuart_ker_ck** 停止，也可以对 LPUART 寄存器进行读/写操作。

  禁用双时钟域功能时，**lpuart_ker_ck** 与 lpuart_pclk 时钟相同。

lpuart_pclk 和 **lpuart_ker_ck** 之间无任何限制：**lpuart_ker_ck** 既可快于也可慢于 **lpuart_pclk**，唯一的限制是软件以足够快的速度管理通信的能力。

49.3.2 LPUART 信号

LPUART 双向通信需要至少两个引脚：接收数据输入引脚 (RX) 和发送数据输出引脚 (TX)：

- **RX**（接收数据输入引脚）
  RX 为串行数据输入引脚。
- **TX**（发送数据输出引脚）
  如果关闭发送器，该输出引脚模式由其 I/O 端口配置决定。如果使能了发送器但没有待发送的数据，则 TX 引脚处于高电平。在单线模式下，该 I/O 用于发送和接收数据。

RS232 硬件流控制模式

在 RS232 硬件流控制模式下需要以下引脚：

- **CTS**（清除以发送）
  如果驱动为高电平，则该信号用于在当前传输结束时阻止数据发送。
- **RTS**（请求以发送）
  如果为低电平，则该信号用于指示 USART 已准备好接收数据。

RS485 硬件流控制模式

在 RS485 硬件控制模式下需要以下引脚：

- DE 驱动器使能
  该信号用于激活外部收发器的发送模式。

注： *DE 和 RTS 共用同一个引脚。*

49.3.3 LPUART 字符说明

可通过对 LPUART_CR1 寄存器中的 M 位（M0：位 12，M1：位 28）进行编程来将字长设置为 7 位、8 位或 9 位（请参见*图 563*）。

- 7 位字符长度：M[1:0] = “10”
- 8 位字符长度：M[1:0] = “00”
- 9 位字符长度：M[1:0] = “01”

在默认情况下，信号（TX 或 RX）在起始位工作期间处于低电平状态。在停止位工作期间处于高电平状态。

通过极性配置控制，可以单独针对每个信号对这些值取反。

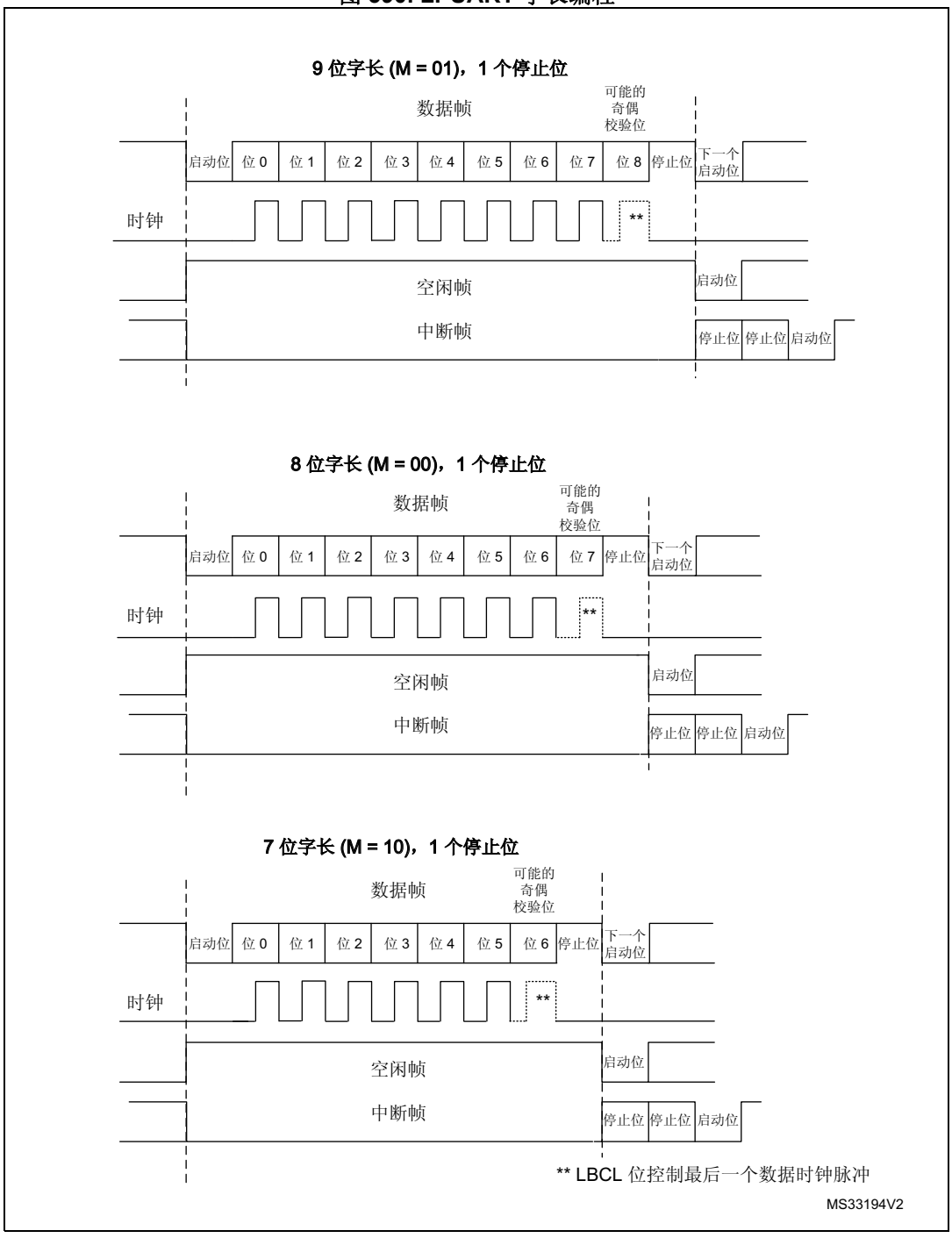
*空闲字符*可理解为整个帧周期内电平均为“1”。（停止位的电平也是“1”）。

*停止字符*可理解为在一个帧周期内接收到的电平均为“0”。发送器在中断帧的末尾插入 2 个停止位。

发送和接收操作由通用波特率发生器驱动。发送器和接收器的使能位置 1 时，将分别生成发送时钟和接收时钟。

下面给出了各个块的详细信息。

图 590. LPUART 字长编程



49.3.4 LPUART FIFO 和阈值

LPUART 可工作在 FIFO 模式下。

LPUART 具有一个发送 FIFO (TXFIFO) 和一个接收 FIFO (RXFIFO)。可通过将 LPUART_CR1 寄存器中的 FIFOEN 位（位 29）置 1 使能 FIFO 模式。

最大数据字长度为 9 位，因此 TXFIFO 为 9 位宽。不过，RXFIFO 的默认宽度为 12 位。这是因为接收器不仅在 FIFO 中存储数据，而且还存储与每个字符相关的错误标志（奇偶校验错误、噪声错误和帧错误标志）。

注： *接收的数据与相应的标志一起存储在 RXFIFO 中，但读取 RDR 时仅读取数据。*

状态标志位于 LPUART_ISR 寄存器中。

可以定义触发 Tx 和 RX 中断的 TXFIFO 和 RXFIFO 阈值。这些阈值通过 LPUART_CR3 控制寄存器中的 RXFTCFG 和 TXFTCFG 位域进行编程。

在这种情况下：

- 当 RXFIFO 中接收到的数据量达到 RXFTCFG 位域中编程的阈值时，会生成 Rx 中断。此时，LPUART_ISR 寄存器中的 RXFT 标志置 1。这表示已接收到 RXFTCFG 数据：LPUART_RDR 中有 1 个数据，RXFIFO 中有 (RXFTCFG - 1) 个数据。例如，如果将 RXFTCFG 编程为“101”，则在接收到对应于 FIFO 大小的数据量（RXFIFO 中有 FIFO 大小 - 1 个数据，LPUART_RDR 中有 1 个数据）时，RXFT 标志将置 1。因此，下一个接收到的数据不会将上溢标志置 1。
- 当 TXFIFO 中的空位置数达到在 TXFTCFG 位域中编程的阈值时，会生成 Tx 中断。

49.3.5 LPUART 发送器

发送器可发送 7 位、8 位或 9 位的数据字，具体取决于 M 位的状态。要激活发送器功能，必须将发送使能位 (TE) 置 1。发送移位寄存器中的数据输出到 TX 引脚。

字符发送

LPUART 发送期间，首先通过 TX 引脚移出数据的最低有效位 （默认配置） 。该模式下，LPUART_TDR 寄存器的缓冲区 (TDR) 位于内部总线和发送移位寄存器之间 （请参见*图 589*）。

使能 FIFO 模式时，写入到 LPUART_TDR 寄存器中的数据会在 TXFIFO 中排队。

每个字符前面都有一个起始位，其对应于一个位周期的逻辑低电平。字符由可配置数量的停止位终止。

停止位的数量可为 1 或 2。

注： *向 LPUART_TDR 中写入要发送的数据前，TE 位必须先置 1。*

数据发送期间不应复位 TE 位。发送期间复位 TE 位会冻结波特率计数器，进而损坏 TX 引脚上的数据。当前发送的数据将会丢失。

使能 TE 位后，将会发送空闲帧。

可配置的停止位

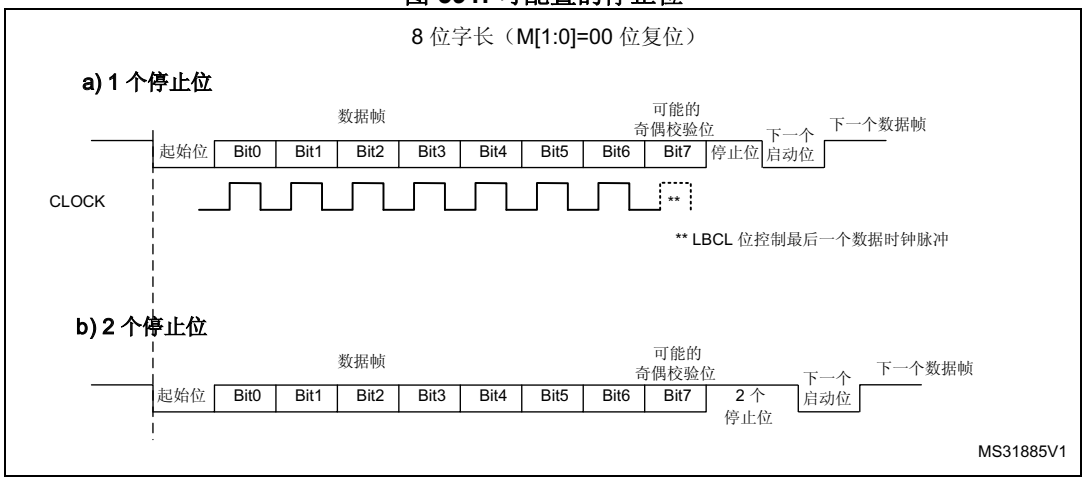
可以在 LPUART_CR2 的位 13 和位 12 中编程将随各个字符发送的停止位的数量。

- ***1 个停止位***：这是停止位数量的默认值。
- ***2 个停止位***：正常 LPUART 模式、单线模式和调制解调器模式支持该值。

空闲帧发送将包括停止位。

break 发送是 10 个低电平位（M[1:0] = “00”时）、11 个低电平位（M[1:0] = “01”时）或 9 个低电平位（M[1:0] = “10”时），然后是 2 个停止位。无法传送长 break（break 长度大于 9/10/11 个低电平位）。

图 591. 可配置的停止位



字符发送步骤

要发送字符，需遵循以下步骤：

1. 对 LPUART_CR1 中的 M 位进行编程以定义字长。
2. 使用 LPUART_BRR 寄存器选择所需波特率。
3. 对 LPUART_CR2 中的停止位数量进行编程。
4. 通过向 LPUART_CR1 寄存器中的 UE 位写入“1”使能 LPUART。
5. 如果将进行多缓冲区通信，请选择 LPUART_CR3 中的 DMA 使能 (DMAT)。按照*第 48.5.10 节：USART 多处理器通信*中的说明配置 DMA 寄存器。
6. 将 LPUART_CR1 中的 TE 位置 1 以便在首次发送时发送一个空闲帧。
7. 在 LPUART_TDR 寄存器中写入要发送的数据。为每个要在单缓冲区模式下发送的数据重复该操作。
   - 禁止 FIFO 模式时，向 LPUART_TDR 写入数据会将 TXE 标志清零。
   - 使能 FIFO 模式时，向 LPUART_TDR 写入数据会为 TXFIFO 增添一个数据。当 TXFNF 标志置 1 时，会对 LPUART_TDR 执行写操作。该标志会保持置 1，直到 TXFIFO 已满。
8. 将最后一个数据写入 LPUART_TDR 寄存器后，等待 TC = “1”。这表明最后一个帧的传送已完成。
   - 禁止 FIFO 模式时，这表示最后一个帧的发送已完成。
   - 使能 FIFO 模式时，这表示 TXFIFO 和移位寄存器均为空。

   当 LPUART 被禁止或进入暂停模式时，需要执行此检查来避免最后一次发送的崩溃。

单字节通信

- 禁止 FIFO 模式时：

  对发送数据寄存器进行写操作始终会清零 TXE 位。TXE 标志由硬件置 1 以指示：

  – 数据已从 LPUART_TDR 寄存器移到移位寄存器中且数据发送已开始；
  – LPUART_TDR 寄存器为空；
  – LPUART_TDR 寄存器中可写入下一个数据，而不会覆盖前一个数据。

  当 TXEIE 位置 1 时，TXE 标志会生成中断。

  发送时，要传入 LPUART_TDR 寄存器的写指令中存有 TDR 寄存器中的数据，该数据将在当前发送结束时复制到移位寄存器中。

  未发送时，要传入 LPUART_TDR 寄存器的写指令会将数据置于移位寄存器中，数据发送开始时，TXE 位置 1。

- 使能 FIFO 模式时，TXFNF（TXFIFO 未满）标志由硬件置 1，以指示：
  – TXFIFO 未满；
  – LPUART_TDR 寄存器为空；
  – LPUART_TDR 寄存器中可写入下一个数据，而不会覆盖前一个数据。发送时，对 LPUART_TDR 寄存器的写操作会将数据存储在 TXFIFO 中。该数据将在当前发送结束时从 TXFIFO 复制到移位寄存器中。

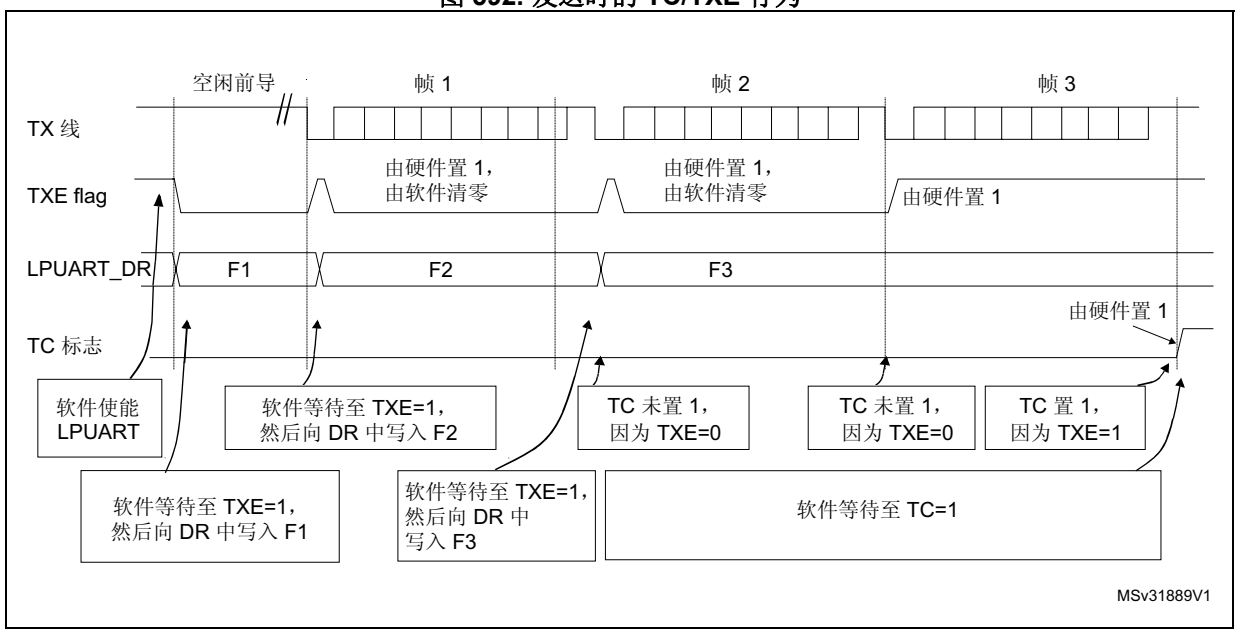
  TXFIFO 未满时，即使在对 LPUART_TDR 执行完写操作后，TXFNF 标志也保持为“1”。该标志在 TXFIFO 已满时清零。TXFNEIE 位置 1 时，该标志会生成中断。

  或者，当达到 TXFIFO 阈值时，会生成中断并将数据写入 TXFIFO。在这种情况下，CPU 可写入由编程的阈值定义的数据块。

  如果帧已发送（停止位后）且 TXE 标志（FIFO 模式下的 TXFE）置 1，TC 位将变为高电平。如果 LPUART_CR1 寄存器中的 TCIE 位置 1，将生成中断。

  向 LPUART_TDR 寄存器中写入最后一个数据后，必须等待至 TC=“1”，之后才可禁止 LPUART 或使微控制器进入低功耗模式（请参见*图 592：发送时的 TC/TXE 行为*）。

图 592. 发送时的 TC/TXE 行为



注： *使能 FIFO 管理时，TXFNF 标志将用于数据发送。*

break 字符

将 SBKRQ 位置 1 将发送一个 break 字符。break 帧的长度取决于 M 位（请参见*图 590*）。

如果将“1”写入 SBKRQ 位，则当前字符发送完成后，将在 TX 线路上发送一个 break 字符。通过写操作将 SBKF 位置 1 并在 break 字符发送完成时（发送 break 字符后的停止位期间），该位由硬件复位。LPUART 在 break 帧末尾的两位持续时间内插入一个逻辑“1”信号 (STOP)，以确保识别下个帧的起始位。

如果 SBKRQ 位置 1，则在当前发送结束时，会发送一个 break 字符。

使能 FIFO 模式时，即使 TXFIFO 已满，发送 break 字符的优先级也仍高于发送数据的优先级。

空闲字符

将 TE 位置 1 会驱动 LPUART 在第一个数据帧之前发送一个空闲帧。

49.3.6 LPUART 接收器

LPUART 可接收 7 位、8 位或 9 位的数据字，具体取决于 LPUART_CR1 寄存器中的 M 位。

起始位检测

在 LPUART 中，先在 Rx 线路上出现下降沿时检测起始位，然后在起始位中间采样以确认电平是否仍为“0”。如果起始采样为“1”，则噪声错误标志 (NE) 置 1，起始位将被丢弃，接收器将等待新的起始位。否则，接收器将继续正常采样所有传入位。

字符接收

LPUART 接收期间，首先通过 RX 引脚移入数据的最低有效位（默认配置）。该模式下，LPUART_RDR 寄存器的缓冲区 (RDR) 位于内部总线和接收移位寄存器之间。

字符接收步骤

要接收字符，需遵循以下步骤：

1. 对 LPUART_CR1 中的 M 位进行编程以定义字长。
2. 使用波特率寄存器 LPUART_BRR 选择所需波特率
3. 对 LPUART_CR2 中停止位个数进行编程。
4. 通过向 LPUART_CR1 寄存器中的 UE 位写入“1”使能 LPUART。
5. 如果将进行多缓冲区通信，请选择 LPUART_CR3 中的 DMA 使能 (DMAR)。
   按照*第 48.5.10 节：USART 多处理器通信*中的说明配置 DMA 寄存器。
6. 将 RE 位 LPUART_CR1 置 1。这一操作将使能接收器开始搜索起始位。

接收到字符时

- 禁止 FIFO 模式时，RXNE 位置 1。这表明移位寄存器的内容已传送到 RDR。也就是说，已接收到并可读取数据（及其相应的错误标志）。
- 如果已使能 FIFO 模式，则 RXFNE 位置 1，这表示 RXFIFO 非空。读取 LPUART_RDR 会返回输入到 RXFIFO 中的最早数据。接收到数据时，数据以及相应的错误位将一起被存储在 RXFIFO 中。
- 如果 RXNEIE（FIFO 模式下时为 RXFNEIE）位置 1，则会生成中断。
- 如果接收期间已检测到帧错误、噪声错误或上溢错误，错误标志位可置 1。
- 在多缓冲区通信模式下：
  - 禁止 FIFO 模式时，每接收到一个字节后 RXNE 标志均置 1，然后通过 DMA 对接收数据寄存器执行读操作清零。
  - 如果使能 FIFO 模式，则 RXFNE 标志在 RXFIFO 非空时置 1。每次收到 DMA 请求后，都会从 RXFIFO 检索数据。在 RXFIFO 非空时（即，当 RXFIFO 中存在要读取的数据时），会触发 DMA 请求。
- 在单缓冲区模式下：
  - 如果禁止 FIFO 模式，则通过软件对 LPUART_RDR 寄存器进行读操作来将 RXNE 标志清零。也可以通过向 LPUART_RQR 寄存器中的 RXFRQ 写入“1”来将 RXNE 标志清零。RXNE 位必须在结束接收下一个字符前清零，以避免发生上溢错误。
  - 如果使能 FIFO 模式，则 RXFNE 标志在 RXFIFO 非空时置 1。每次对 LPUART_RDR 寄存器执行完读操作后，都会从 RXFIFO 中检索数据。RXFIFO 为空时，RXFNE 标志将清零。也可以通过将 LPUART_RQR 寄存器中的 RXFRQ 位置“1”将 RXFNE 标志清零。RXFIFO 已满时，必须在结束接收下一个字符前读取 RXFIFO 中的第一个条目，以避免发生上溢错误。当 RXFNEIE 位置 1 时，RXFNE 标志会生成中断。或者，当达到 RXFIFO 阈值时，会生成中断并从 RXFIFO 中读取数据。在这种情况下，CPU 可读取由编程的阈值定义的数据块。

break 字符

接收到 break 字符时，USART 将会按照帧错误对其进行处理。

空闲字符

检测到空闲帧时，除了在 IDLEIE 位置 1 时会生成中断外，处理步骤与接收到数据字符的情况相同。

上溢错误

- 禁止 FIFO 模式

  如果在 RXNE 未清零时接收到字符，则会发生上溢错误。

  RXNE 位清零前，数据无法从移位寄存器传送到 RDR 寄存器。每接收到一个字节后，RXNE 标志位都将置 1。

  当 RXNE 标志位是 1 时，如果在接收到下一个数据或尚未处理上一个 DMA 请求时，则会发生上溢错误。发生上溢错误时：

  – ORE 位置 1。
  – RDR 中的内容不会丢失。对 LPUART_RDR 执行读操作时可使用先前的数据。
  – 移位寄存器将被覆盖。之后，上溢期间接收到的任何数据都将丢失。
  – 如果 RXNEIE 位置 1 或 EIE 位置 1，则会生成中断。

- 使能 FIFO 模式

  移位寄存器准备好传送且接收 FIFO 已满时，会发生上溢错误。

  在 RXFIFO 中存在一个空闲位置之前，数据无法从移位寄存器传送到 LPUART_RDR 寄存器。当 RXFIFO 非空时，RXFNE 标志置 1。

  如果 RXFIFO 已满且移位寄存器已准备好传送，则会发生上溢错误。发生上溢错误时：

  – ORE 位置 1。
  – RXFIFO 中的第一个条目不会丢失。可通过对 LPUART_RDR 执行读操作来获取。
  – 移位寄存器将被覆盖。之后，上溢期间接收到的任何数据都将丢失。
  – 如果 RXFNEIE 位置 1 或 EIE 位置 1，则会生成中断。

通过设置 ICR 寄存器中的 ORECF 位复位 ORE 位。

注： *ORE 位置 1 时表示至少 1 个数据丢失。*

禁止 FIFO 模式时，有以下两种可能

- *如果 RXNE=“1”，则最后一个有效数据存储于接收寄存器 (RDR) 中并且可进行读取；*
- *如果 RXNE=“0”，则表示最后一个有效数据已被读取，因此 RDR 中没有要读取的数据。接收到新（和丢失）数据的同时已读取 RDR 中的最后一个有效数据时，会发生该情况。*

选择时钟源

时钟源的选择通过时钟控制系统完成（请参见 *复位和时钟控制器 (RCC) 部分*）。在使能 LPUART 之前，必须通过 UE 位选择时钟源。

必须遵循以下两个条件选择时钟源：

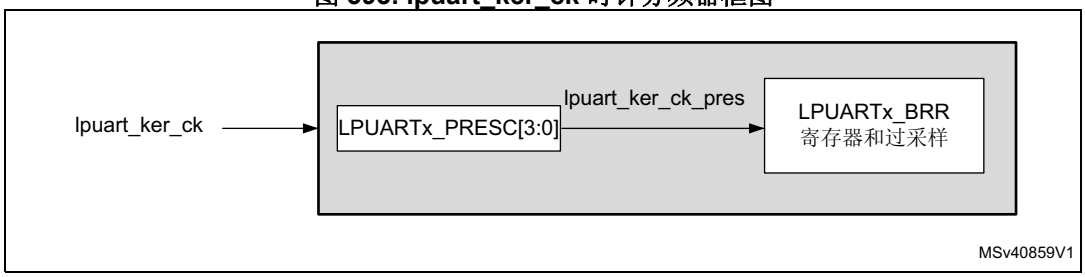
- 可在低功耗模式下使用 LPUART
- 通信速度。

时钟源频率为 lpuart_ker_ck。

如果支持双时钟域和从低功耗模式唤醒功能，则 lpuart_ker_ck 时钟源可在 RCC 中进行配置（请参见 *复位和时钟控制器 (RCC) 部分*）。否则，lpuart_ker_ck 与 lpuart_pclk 相同。

lpuart_ker_ck 可根据 LPUART_PRESC 寄存器中的可编程系数进行分频。

图 593. lpuart_ker_ck 时钟分频器框图



某些 lpuart_ker_ck 时钟源允许 LPUART 在 MCU 处于低功耗模式时接收数据。
必要时，LPUART 可基于所接收的数据和唤醒模式的选择来唤醒 MCU，以通过用软件读取 LPUART_RDR 寄存器的方式或通过 DMA 的方式传输已接收数据。

对于其他时钟源，系统必须激活才能进行 LPUART 通信。

时钟源还决定通信速度范围（尤其是最大通信速度）。

接收器在尽可能靠近波特周期中间的位置采样每个传入波特。每个传入波特仅进行一次采样。

注： *不进行数据噪声检测。*

帧错误

如果接收数据时未在预期时间内识别出停止位，从而出现同步失效或过度的噪声，则会检测到帧错误。

检测到帧错误时：

- FE 位由硬件置 1。
- 无效数据从移位寄存器传送到 LPUART_RDR 寄存器。
- 单字节通信时无中断产生。然而，在 RXNE 位产生中断时，该位出现上升沿。多缓冲区通信时，LPUART_CR3 寄存器中的 EIE 位置 1 时将发出中断。

通过向 LPUART_ICR 寄存器中的 FECF 位写入“1”来复位 FE 位。

接收期间可配置的停止位

可通过 LPUART_CR2 的控制位配置要接收的停止位的数量：可以是 1 个或 2 个（正常模式下）。

- **1 个停止位**：将在第 8、第 9 和第 10 次采样时对 1 个停止位进行采样。
- **2 个停止位**：将在第二个停止位的中间对 2 个停止位进行采样。RXNE 和 FE 标志在此采样期间（例如，在第二个停止位期间）置 1。发生帧错误时不检测第一个停止位。

49.3.7 LPUART 波特率生成

接收器和发送器（Rx 和 Tx）的波特率均设置为 LPUART_BRR 寄存器中编程的值。

$$\text{Tx/Rx baud} = \frac{256 \times \text{lpuartckpres}}{\text{LPUARTDIV}}$$

LPUARTDIV 在 LPUART_BRR 寄存器中定义。

注： *对 LPUART_BRR 执行写操作后，波特率计数器更新为波特率寄存器中的新值。因此，波特率寄存器的值不应在通信时发生更改。*

禁止在 LPUART_BRR 寄存器中写入小于 0x300 的值。

f_{CK} 必须介于 3 x 波特率到 4096 x 波特率之间。

LPUART 时钟源为 LSE 时可达到的最大波特率为 9600 波特。当 LPUART 由与 LSE 时钟不同的时钟源计时，可以达到更高的波特率。例如，如果 LPUART 时钟源频率为 100 MHz，则可达到的最大波特率约为 33 M 波特。

表 382. lpuart_ker_ck_pres = 32,768 KHz 时编程的波特率的误差计算

波特率		lpuart_ker_ck_pres = 32,768 KHz		
序号	所需值	实际值	波特率寄存器中编程的值	误差 % = (计算值 - 所需值) / 所需波特率
1	0.3 KBps	0.3 KBps	0x6D3A	0
2	0.6 KBps	0.6 KBps	0x369D	0
3	1200 Bps	1200.087 Bps	0x1B4E	0.007
4	2400 Bps	2400.17 Bps	0xDA7	0.007
5	4800 Bps	4801.72 Bps	0x6D3	0.035
6	9600 KBps	9608.94 Bps	0x369	0.093

表 383. 采用 16 倍过采样 (OVER8 = 0) 时，在 f_{CK} = 100 MHz 下

波特率		f_{CK} = 100 MHz		
序号	所需值	实际值	波特率寄存器中编程的值	误差 % = (计算值 - 所需值) / 所需波特率
1	38400 波特	38400,04 波特	A2C2A	0,0001
2	57600 波特	57600,06 波特	6C81C	0,0001
3	115200 波特	115200,12 波特	3640E	0,0001
4	230400 波特	230400,23 波特	1B207	0,0001
5	460800 波特	460804,61 波特	D903	0,001
6	921600 波特	921625,81 波特	6C81	0,0028
7	4000 K 波特	4000000,00 波特	1900	0
8	10000 K 波特	10000000,00 波特	A00	0
9	20000 K 波特	20000000,00 波特	500	0
10	30000 K 波特	33032258,06 波特	307	0,1

49.3.8 LPUART 接收器对时钟偏差的容差

仅当总时钟系统偏差小于 LPUART 接收器的容差时，LPUART 异步接收器才能正常工作。影响总偏差的因素包括：

- DTRA：发送器误差引起的偏差（其中还包括发送器本地振荡器的偏差）
- DQUANT：接收器的波特率量化引起的误差
- DREC：接收器本地振荡器的偏差
- DTCL：传输线路引起的偏差（通常是由于收发器所引起，它可能会在低电平到高电平转换时序与高电平到低电平转换时序之间引入不对称）

DTRA + DQUANT + DREC + DTCL + DWU < LPUART 接收器的容差

其中

DWU 是使用从低功耗模式唤醒时因采样点偏差而产生的误差。

LPUART 接收器在*表 384* 中指定的最大容许偏差下可正确接收数据：

- 通过 LPUART_CR2 寄存器中的 STOP[1:0] 位定义的停止位数。
- LPUART_BRR 寄存器值。

表 384. LPUART 接收器的容差

M 位	768 < BRR < 1024	1024 < BRR < 2048	2048 < BRR < 4096	4096 ≤ BRR
8 位（M=“00”），1 个停止位	1.82%	2.56%	3.90%	4.42%
9 位（M=“01”），1 个停止位	1.69%	2.33%	2.53%	4.14%
7 位（M=“10”），1 个停止位	2.08%	2.86%	4.35%	4.42%
8 位（M=“00”），2 个停止位	2.08%	2.86%	4.35%	4.42%
9 位（M=“01”），2 个停止位	1.82%	2.56%	3.90%	4.42%
7 位（M=“10”），2 个停止位	2.34%	3.23%	4.92%	4.42%

注： *当接收的帧恰好包含 10 个（M 位 = “00”）、11 个（M 位 = “01”）或 9 个（M 位 = “10”）位时间的空闲帧时，表 384 中指定的数据可能与特例中的数据略微不同。*

49.3.9 LPUART 多处理器通信

可以执行 LPUART 多处理器通信（多个 LPUART 连接在一个网络中）。例如，其中一个 LPUART 可以是主器件，其 TX 输出与其他 LPUART 的 RX 输入相连接。其他 LPUART 为从器件，其各自的 TX 输出在逻辑上通过与运算连在一起，并与主 LPUART 的 RX 输入相连。

在多处理器配置中，理想情况下通常只有预期的消息接收方主动接收完整的消息内容，从而减少由所有未被寻址的接收器造成的冗余 LPUART 服务开销。

可通过静默功能将未被寻址的器件置于静默模式下。为了使用静默模式功能，必须将 LPUART_CR1 寄存器中的 MME 位置 1。

注： *使能 FIFO 管理且 MME 已置 1 时，不得清零 MME 位再快速将其置 1（在两个 lpuart_ker_ck 周期内），否则静默模式可能保持有效。*

使能静默模式时：

- 不得将接收状态位置 1。
- 禁止任何接收中断。
- LPUART_ISR 寄存器中的 RWU 位置“1”。在某些情况下，RWU 可以由硬件或软件通过 LPUART_RQR 寄存器中的 MMRQ 位自动控制。

根据 LPUART_CR1 寄存器中 WAKE 位的设置，LPUART 可使用以下两种方法进入或退出静默模式：

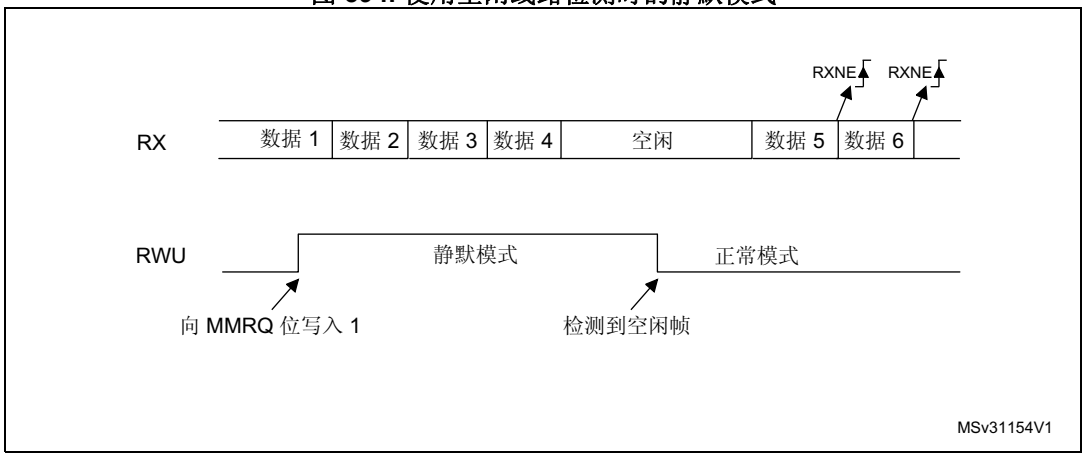
- 如果 WAKE 位被复位，则进行空闲线路检测，
- 如果 WAKE 位置 1，则进行地址标记检测。

空闲线路检测（WAKE=“0”）

当向 MMRQ 位写入 1 且 RWU 位自动置 1 时，LPUART 进入静默模式。

检测到空闲帧时，LPUART 将唤醒。此时 RWU 位会由硬件清零，但 LPUART_ISR 寄存器中的 IDLE 位不会置 1。*图 594* 中给出了使用空闲线路检测时静默模式行为的示例。

图 594. 使用空闲线路检测时的静默模式



注： *如果在 IDLE 字符已经过去时将 MMRQ 位置 1，将不会进入静默模式（RWU 未置 1）。*

如果在线路处于空闲状态时激活 LPUART，在一个 IDLE 帧持续时间后（不只在接收一个字符帧后）会检测到空闲状态。

4 位/7 位地址标记检测（WAKE=“1”）

在此模式下，如果字节的 MSB 为 1，则将这些字节识别为地址，否则将其识别为数据。在地址字节中，目标接收器的地址位于 4 个或 7 个 LSB 中。7 位或 4 位地址检测通过 ADDM7 位来选择。接收器会将此 4 位/7 位字与其地址进行比较，该接收器的地址在 LPUART_CR2 寄存器的 ADD 位中进行设置。

注： *在 7 位和 9 位数据模式下，地址检测分别在 6 位和 8 位地址上完成（ADD[5:0] 和 ADD[7:0]）。*

当接收到与其编程地址不匹配的地址字符时，LPUART 会进入静默模式。此时，RWU 位将由硬件置 1。LPUART 进入静默模式后，RXNE 标志不会针对此地址字节置 1，也不会发出中断或 DMA 请求。

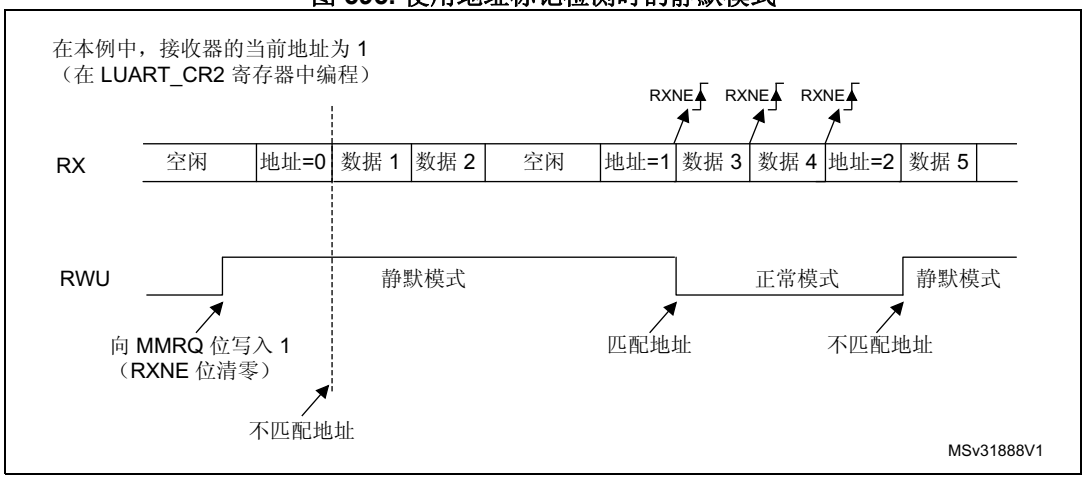
当向 MMRQ 位写入“1”时，LPUART 也会进入静默模式。这种情况下，RWU 位也自动置 1。

当接收到与编程地址匹配的地址字符时，LPUART 会退出静默模式。然后 RWU 位被清零，可以开始正常接收后续字节。由于 RWU 位已清零，RXNE/RXFNE 位会针对地址字符置 1。

注： *使能 FIFO 管理时，如果在接收器对数据的最后一位进行采样时 MMRQ 位置 1，则可在有效进入静默模式之前接收该数据。*

图 595 中给出了使用地址标记检测时静默模式行为的示例。

图 595. 使用地址标记检测时的静默模式



49.3.10 LPUART 极性控制

将 LPUART_CR1 寄存器中的 PCE 位置 1，可以使能奇偶校验控制（发送时生成奇偶校验位，接收时进行奇偶校验检查）。根据 M 位定义的帧长度，*表 385* 中列出了可能的 LPUART 帧格式。

表 385：LPUART 帧格式

M 位	PCE 位	LPUART 帧(1)
00	0	| SB | 8 位数据 | STB |
00	1	| SB | 7 位数据 | PB | STB |
01	0	| SB | 9 位数据 | STB |
01	1	| SB | 8 位数据 PB | STB |
10	0	| SB | 7 位数据 | STB |
10	1	| SB | 6 位数据 | PB | STB |

1. 图注：SB：起始位，STB：停止位，P：奇偶校验位。
2. 在数据寄存器中，PB 始终位于 MSB 位置（第 8 位或第 7 位，具体取决于 M 位的值）。

偶校验

对奇偶校验位进行计算，使帧和奇偶校验位中“1”的数量为偶数（帧由 6 个、7 个或 8 个 LSB 位组成，具体取决于 M 位的值）。

例如，如果数据=00110101 且 4 个位置 1，则在选择偶校验（LPUART_CR1 中的 PS 位 = “0”）时，校验位为 0。

奇校验

对奇偶校验位进行计算，使帧和奇偶校验位中“1”的数量为奇数（帧由 6 个、7 个或 8 个 LSB 位组成，具体取决于 M 位的值）。

例如，如果数据=00110101 且 4 个位置 1，则在选择奇校验（LPUART_CR1 寄存器中的 PS 位 =“1”）时，校验位为 1。

接收时进行奇偶校验检查

如果奇偶校验检查失败，则 LPUART_ISR 寄存器中的 PE 标志置 1；如果 LPUART_CR1 寄存器中 PEIE 位置 1，则会生成中断。通过软件向 LPUART_ICR 寄存器中的 PECF 位写入“1”来清零 PE 标志。

发送时的奇偶校验生成

如果 LPUART_CR1 中的 PCE 位置 1，则在数据寄存器中所写入数据的 MSB 位会进行传送，但是会由奇偶校验位进行更改（如果选择偶校验（PS=“0”），则“1”的数量为偶数；如果选择奇校验（PS=“1”），则“1”的数量为奇数）。

49.3.11 LPUART 单线半双工通信

通过将 LPUART_CR3 寄存器中的 HDSEL 位置 1 来选择单线半双工模式。在此模式下，必须将以下位清零：

- LPUART_CR2 寄存器中的 LINEN 和 CLKEN 位。
- LPUART_CR3 寄存器中的 SCEN 和 IREN 位。

LPUART 可以配置为遵循单线半双工协议，其中 TX 和 RX 线路从内部相连接。
使用 LPUART_CR3 寄存器中的控制位 HDSEL 可在半双工通信和全双工通信间进行选择。

向 HDSEL 位写入“1”后：

- TX 和 RX 线路从内部相连接。
- 不能再使用 RX 引脚。
- 无数据传输时，TX 引脚始终处于释放状态。因此，它在空闲状态或接收过程中用作标准 I/O。这意味着，必须对 I/O 进行配置，以便将 TX 配置为复用功能开漏并外接上拉电阻。

除此之外，通信协议与正常 LPUART 模式下的通信协议相似。此线路上的任何冲突都必须由软件管理（例如，使用中央仲裁器）。尤其要注意，发送过程永远不会被硬件阻塞，只要数据是在 TE 位置 1 的情况下写入，发送就会持续进行。

注： *在 LPUART 通信中，当进行 1 个停止位配置时，RXNE 标志在停止位中间置 1。*

49.3.12 使用 DMA 和 LPUART 进行连续通信

LPUART 能够使用 DMA 进行连续通信。接收缓冲区和发送缓冲区的 DMA 请求是独立生成的。

注： *要确定是否支持 DMA 模式，请参见第 1851 页的第 48.4 节: USART 实现。*如果不支持 DMA，请按照*第 48.5.6 节中的说明使用 LPUSRT。当 FIFO 禁止时，可以将 LPUART_ISR 寄存器中的 TXE/RXNE 标志清零，从而实现连续通信。*

使用 DMA 进行发送

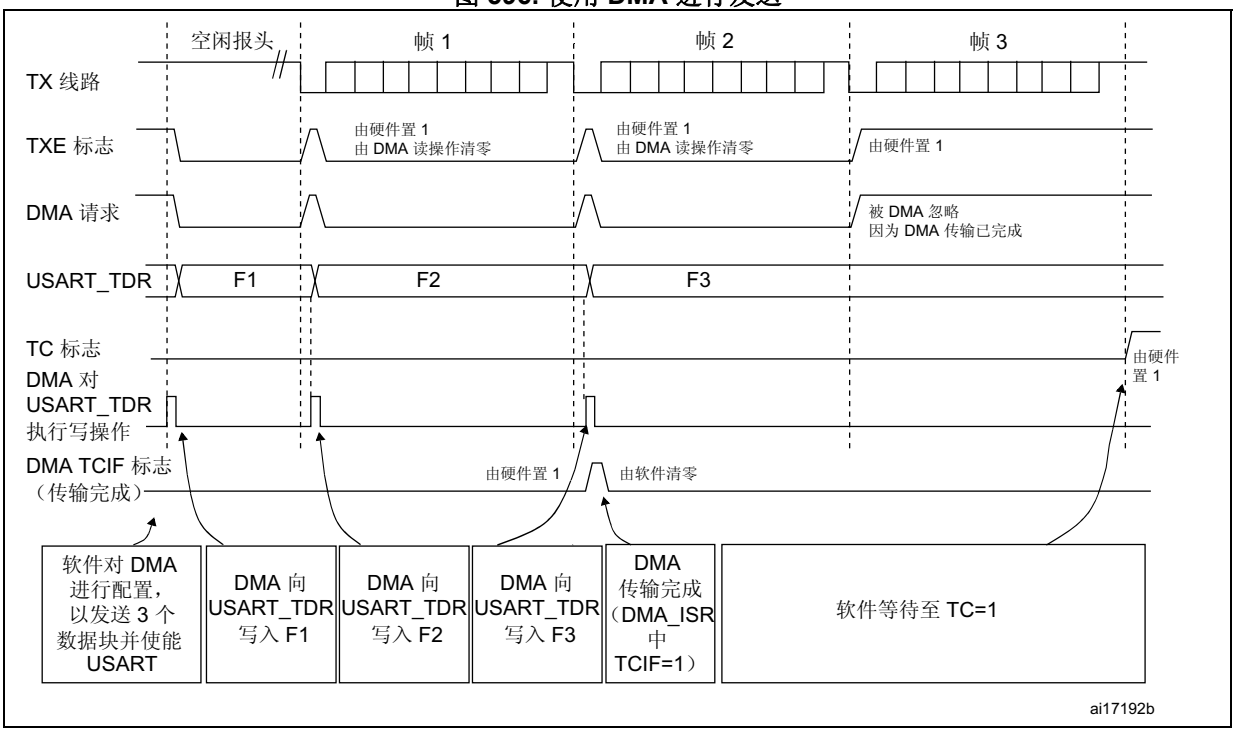
将 LPUART_CR3 寄存器中的 DMAT 位置 1 可以使能 DMA 模式进行发送。当 TXE 标志（使能 FIFO 模式时为 TXFNF 标志）置 1 时，可将数据从配置的 SRAM 区（通过 DMA 外设，请参见*第 15 节：直接存储器访问控制器（DMA1、DMA2）*和*第 16 节：基本直接存储器访问控制器 (BDMA)*）加载到 LPUART_TDR 寄存器。要映射一个 DMA 通道以进行 LPUART 发送，请按以下步骤操作（x 表示通道编号）：

1. 在 DMA 控制寄存器中写入 LPUART_TDR 寄存器地址，将其配置为传输的目标地址。每次发生 TXE（使能 FIFO 模式时为 TXFNF）事件后，数据都从存储器移动到此地址。
2. 在 DMA 控制寄存器中写入存储器地址，将其配置为传输的源地址。每次发生 TXE（使能 FIFO 模式时为 TXFNF）事件后，数据都从该存储区域加载到 LPUART_TDR 寄存器中。
3. 在 DMA 控制寄存器中配置要传输的总字节数。
4. 在 DMA 寄存器中配置通道优先级。
5. 根据应用的需求，在完成一半或全部传输后产生 DMA 中断。
6. 通过将 LPUART_ICR 寄存器中的 TCCF 位置 1，将 LPUART_ISR 寄存器中的 TC 标志清零。
7. 在 DMA 寄存器中激活该通道。

当达到在 DMA 控制器中设置的数据传输量时，DMA 控制器会在 DMA 通道的中断向量上产生一个中断。

在发送模式下，DMA 对所有要发送的数据执行了写操作（DMA_ISR 寄存器中的 TCIF 标志置 1）后，可以对 TC 标志进行监视，以确保 LPUART 通信已完成。在禁止 LPUART 或进入低功耗模式前必须执行此步骤，以避免损坏最后一次发送。软件必须等待至 TC=“1”。TC 标志在所有数据发送期间都保持清零状态，然后在最后一帧发送结束时由硬件置 1。

图 596. 使用 DMA 进行发送



注：　使能 FIFO 管理时，DMA 请求由发送 FIFO 未满（即 TXFNF = “1”）触发。

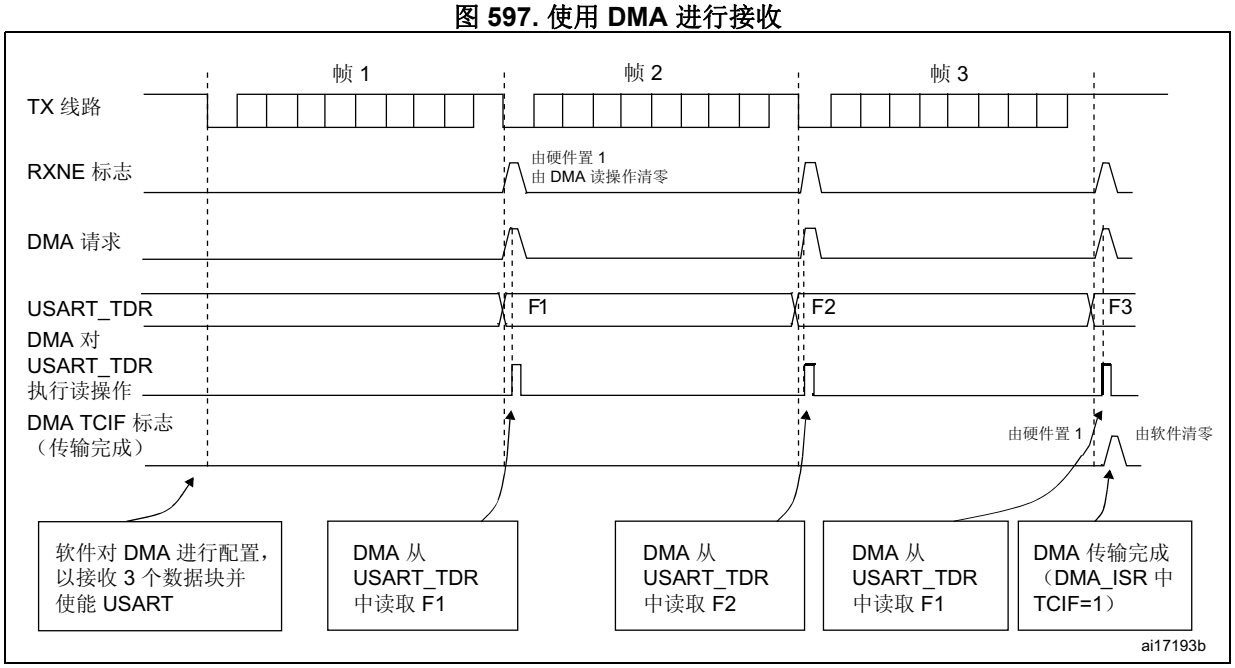
使用 DMA 进行接收

将 LPUART_CR3 寄存器中的 DMAR 位置 1 可以使能 DMA 模式进行接收。接收数据字节时，数据会从 LPUART_RDR 寄存器加载到配置的 SRAM 区域中（通过 DMA 外设，请参见 *第 15 节：直接存储器访问控制器（DMA1、DMA2）*和*第 16 节：基本直接存储器访问控制器 (BDMA)*）。要映射一个 DMA 通道以进行 LPUART 接收，请按以下步骤操作：

1. 在 DMA 控制寄存器中写入 LPUART_RDR 寄存器地址，将其配置为传输的源地址。每次发生 RXNE（使能 FIFO 模式时为 RXFNE）事件后，数据都从该地址移动到存储器。
2. 在 DMA 控制寄存器中写入存储器地址，将其配置为传输的目标地址。每次发生 RXNE（使能 FIFO 模式时为 RXFNE）事件后，数据都从 LPUART_RDR 加载到此存储区域。
3. 在 DMA 控制寄存器中配置要传输的总字节数。
4. 在 DMA 控制寄存器中配置通道优先级。
5. 根据应用的需求，在完成一半或全部传输后产生中断。
6. 在 DMA 控制寄存器中激活该通道。

当达到在 DMA 控制器中设置的数据传输量时，DMA 控制器会在 DMA 通道的中断向量上产生一个中断。

图 597. 使用 DMA 进行接收



注： *使能 FIFO 管理时，DMA 请求由接收 FIFO 非空（即 RXFNE = “1”）触发。*

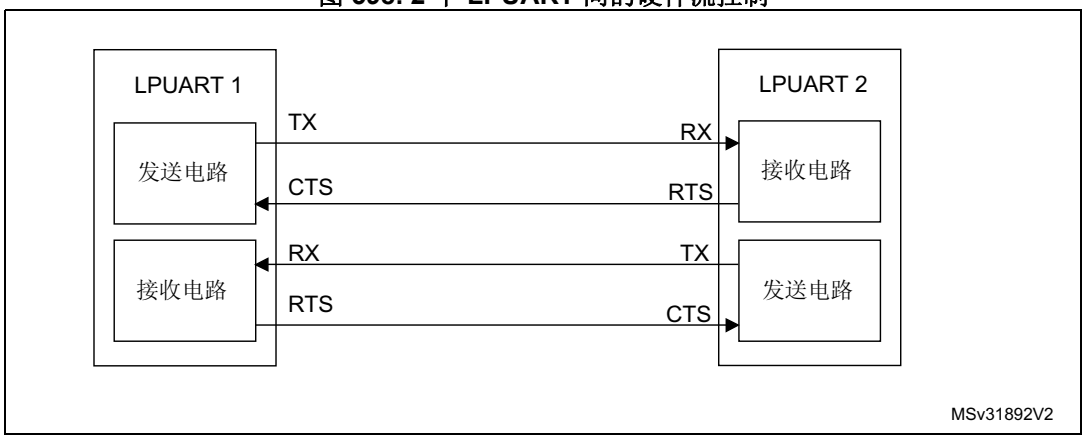
多缓冲区通信中的错误标志和中断生成

在多缓冲区通信模式下，如果事务中发生任何错误，都会在当前字节后将相应错误标志置位。如果中断使能标志置 1，则会产生中断。在单字节接收过程中，与 RXNE（使能 FIFO 模式时为 RXFNE）一同置位的帧错误、上溢错误和噪声标志具有单独的错误标志中断使能位（LPUART_CR3 寄存器中的 EIE 位）；如果该位置 1，在发生其中任何一个错误时，都会在当前字节后使能中断。

49.3.13 RS232 硬件流控制和 RS485 驱动器使能

使用 nCTS 输入和 nRTS 输出可以控制 2 个器件间的串行数据流。*图 584* 显示了在这种模式下如何连接 2 个器件：

图 598. 2 个 LPUART 间的硬件流控制

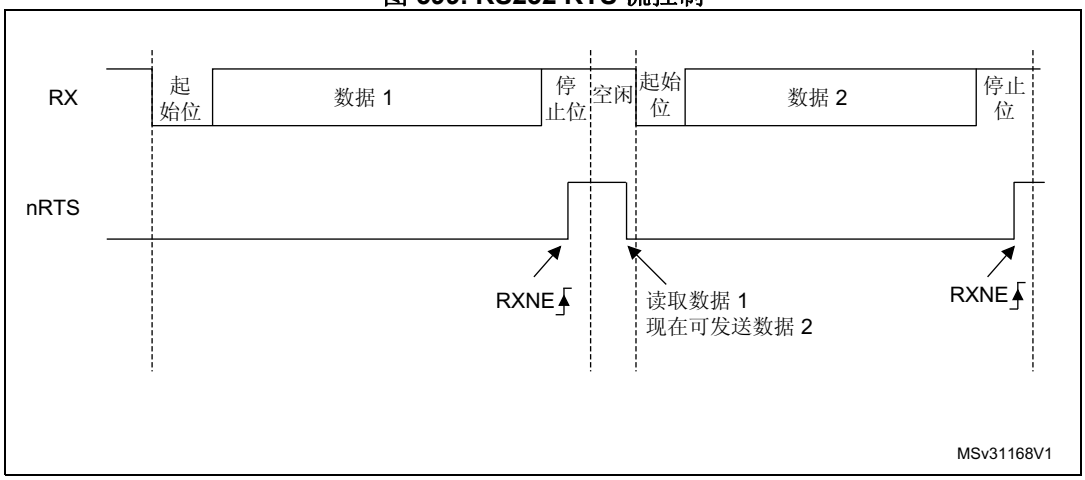


分别向 LPUART_CR3 寄存器中的 RTSE 位和 CTSE 位写入 1，可以分别使能 RS232 RTS 和 CTS 流控制。

RS232 RTS 流控制

如果使能 RTS 流控制（RTSE=“1”），只要 LPUART 接收器准备好接收新数据，便会将 nRTS 变为有效（连接到低电平）。当接收寄存器已满时，会将 nRTS 变为无效，表明发送过程会在当前帧结束后停止。*图 599* 显示了在使能 RTS 流控制的情况下进行通信的示例。

图 599. RS232 RTS 流控制



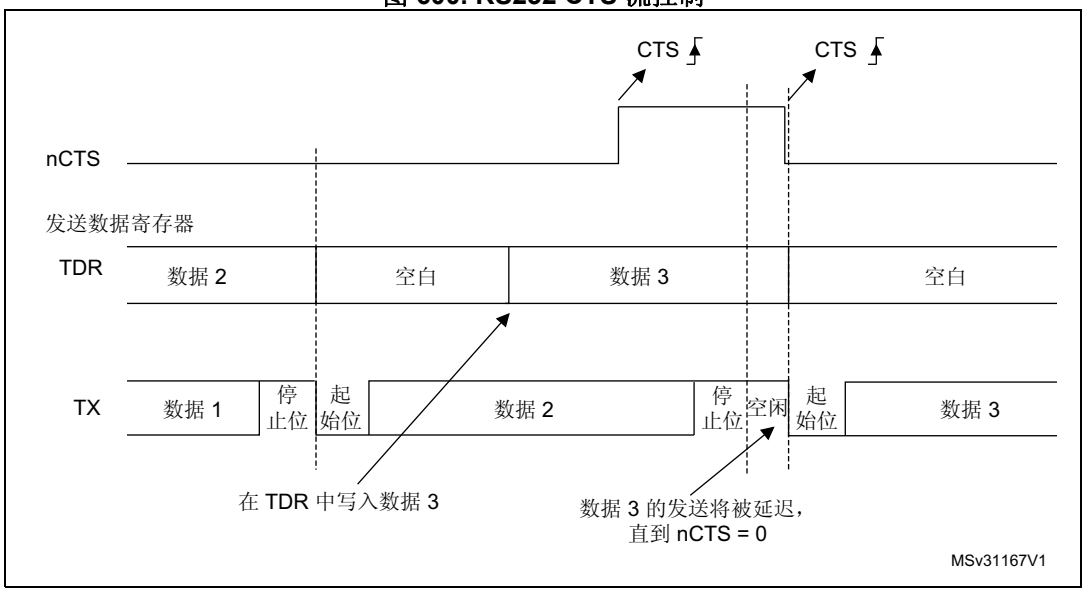
注： *如果使能 FIFO 模式，则仅当 RXFIFO 已满时，nRTS 才会变为无效。*

RS232 CTS 流控制

如果使能 CTS 流控制（CTSE=“1”），则发送器会在发送下一帧前检查 nCTS。如果 nCTS 有效（连接到低电平），则会发送下一数据（假设数据已准备好发送，即 TXE/TXFE=“0”）；否则不会进行发送。如果在发送过程中 nCTS 变为无效，则当前发送完成之后，发送器停止。

当 CTSE=“1”时，只要 nCTS 发生变化，CTSIF 状态位便会由硬件自动置 1。这指示接收器是否已准备好进行通信。如果 LPUART_CR3 寄存器中的 CTSIE 位置 1，则会产生中断。*图 600* 显示了在使能 CTS 流控制的情况下进行通信的示例。

图 600. RS232 CTS 流控制



注： *为正常运行，必须在当前字符结束前至少 3 个 LPUART 时钟源周期内使能 nCTS。此外还应注意，当脉冲短于 2 个 PCLK 周期时无法将 CTSCF 标志置 1。*

RS485 驱动器使能

驱动器使能功能可通过将 LPUART_CR3 控制寄存器中的 DEM 位置 1 使能。这样便可通过 DE（驱动器使能）信号激活外部收发器控制。使能时间为激活 DE 信号与 START 位开始间的时间。可以通过 LPUART_CR1 控制寄存器中的 DEAT [4:0] 位域编程禁止时间。禁止时间为发送的消息中最后一个停止位结束与取消激活 DE 信号间的时间。可以通过 LPUART_CR1 控制寄存器中的 DEDT [4:0] 位域编程禁止时间。DE 信号的极性可使用 LPUART_CR3 控制寄存器中的 DEP 位配置。

LPUART 的 DEAT 和 DEDT 用 LPUART 时钟源 (f_{CK}) 周期表示：

- 驱动器使能有效时间等于
  - (1 + (DEAT x P)) x f_{CK}（如果 P # 0）
  - (1 + DEAT) x f_{CK}（如果 P = 0）
- 驱动器使能禁止时间等于
  - (1 + (DEDT x P)) x f_{CK}（如果 P # 0）
  - (1 + DEDT) x f_{CK}（如果 P = 0）

其中 P = BRR[20:11]

49.3.14 LPUART 低功耗管理

LPUART 具有高级低功耗模式功能，即使禁止 lpuart_pclk 时钟，也能正常传输数据。
UESM 位置 1 时，LPUART 能够将 MCU 从低功耗模式唤醒。
对 usart_pclk 进行门控时，如果某些特定操作需要激活 **usart_pclk** 时钟，则 LPUART 会提供唤醒中断 (**usart_wkup**)：

- 禁止 FIFO 模式时

  必须激活 luart_pclk 时钟以清空 LPUART 数据寄存器。

  在这种情况下，lpuart_wkup 中断源为 RXNE 置“1”。RXNEIE 位必须在进入低功耗模式之前置 1。

- 使能 FIFO 模式时

  必须激活 luart_pclk 时钟以

  – 填充 TXFIFO
  – 或清空 RXFIFO

  在这种情况下，lpuart_wkup 中断源可以为：

  – RXFIFO 非空。此时，RXFNEIE 位必须在进入低功耗模式之前置 1。
  – RxFIFO 已满。此时，RXFFIE 位必须在进入低功耗模式之前置 1，接收到的数据量对应于 RXFIFO 大小，并且 RXFF 标志未置 1。
  – TXFIFO 为空。此时，TXFEIE 位必须在进入低功耗模式之前置 1。

  这允许在低功耗模式下发送/接收 TXFIFO/RXFIFO 中的数据。

  为了避免发生上溢/下溢错误以及在低功耗模式下发送/接收数据，lpuart_wkup 中断源可以是以下事件之一：

  – 达到 TXFIFO 阈值。此时，TXFTIE 位必须在进入低功耗模式之前置 1。
  – 达到 RXFIFO 阈值。此时，RXFTIE 位必须在进入低功耗模式之前置 1。

  例如，如果唤醒时间短于在线路上接收单个字节所需的时间，则应用可将阈值设为最大 RXFIFO 大小。

  使用 RXFIFO 已满、TXFIFO 为空、RXFIFO 非空和 RXFIFO/TXFIFO 阈值中断将 MCU 从低功耗模式唤醒时，可在低功耗模式下尽可能多地进行 LPUART 传输，这样有助于优化功耗。

或者，也可通过 WUS 位域选择特定 **lpuart_wkup** 中断。

检测到唤醒事件后，WUF 标志会由硬件置 1 并在 WUFIE 位置 1 时生成 **lpuart_wkup** 中断。在这种情况下，无需 **lpuart_wkup** 中断即可唤醒。将 WUF 置 1 便足以将 MCU 从低功耗模式唤醒。

注： *在进入低功耗模式之前，请确保未进行任何 LPUART 传输。检查 BUSY 标志不能确保在进行数据接收时绝不会进入低功耗模式。*

WUF 标志在检测到唤醒事件时置 1，而与 MCU 处于低功耗模式还是工作模式无关。

如果在初始化和使能接收器后立即进入低功耗模式，则必须校验 REACK 位以确保 LPUART 确实已使能。

当 DMA 用于接收时，它必须在进入低功耗模式前禁止，并在退出低功耗模式后重新使能。

如果使能 FIFO，则只有在使能静默模式的情况下才能在地址匹配时从低功耗模式唤醒。

使用静默模式和低功耗模式

如果 LPUART 在进入低功耗模式前处于静默模式：

- 不得使用空闲检测时从静默模式唤醒，因为空闲检测无法在低功耗模式下工作。
- 如果使用地址匹配时从静默模式唤醒，则低功耗模式唤醒源也必须是地址匹配。如果 RXNE 标志在进入低功耗模式时置 1，则接口将在地址匹配时和从低功耗模式唤醒时保持静默模式。

注： *使能 FIFO 管理时，静默模式可与从低功耗模式唤醒搭配使用，而且没有任何限制（即，上述关于静默和低功耗模式的两点仅在 FIFO 管理禁止时有效）。*

LPUART 内核时钟 lpuart_ker_ck 在低功耗模式下关闭时，从低功耗模式唤醒

在低功耗模式下，如果在 LPUART 接收线上检测到下降沿时 lpuart_ker_ck 时钟处于关闭状态，则 LPUART 接口会借助 lpuart_ker_ck_req 信号请求开启 lpuart_ker_ck 时钟。lpuart_ker_ck 随后用来接收帧。

如果唤醒事件通过验证，则 MCU 将从低功耗模式唤醒，并会正常进行数据接收。

如果唤醒事件未通过验证，则 usart_ker_ck 会再次关闭，MCU 不会被唤醒并保持在低功耗模式下，且内核时钟请求被释放。

以下示例显示了将唤醒事件编程为“地址匹配检测”并禁止 FIFO 管理的情况。

图 601 所示为唤醒事件通过验证时的行为。

图 601. 唤醒事件通过验证（唤醒事件 = 地址匹配，禁止 FIFO）

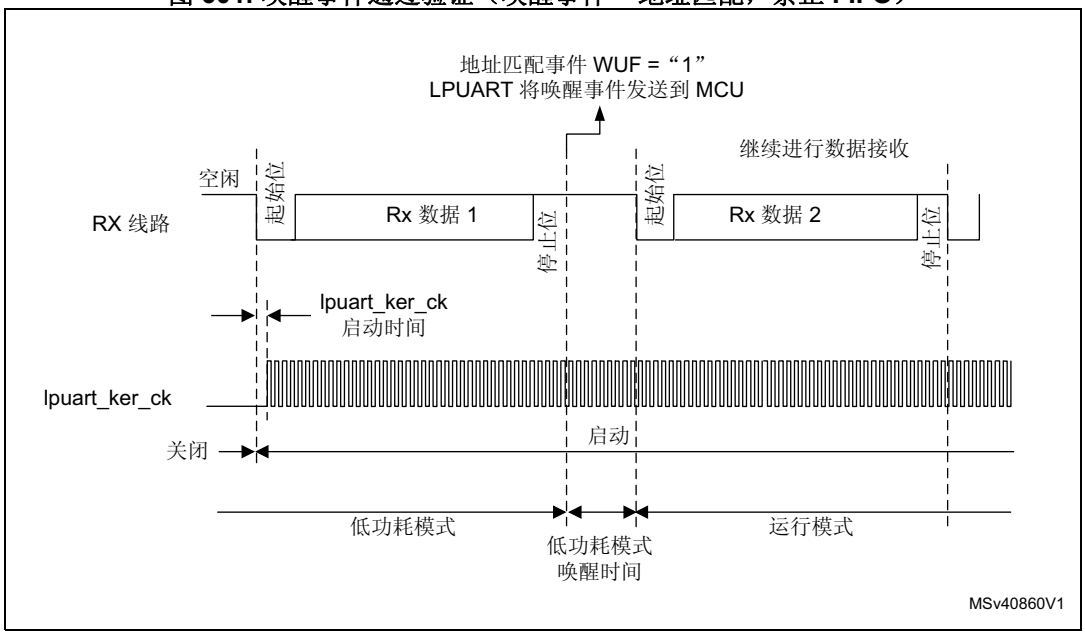
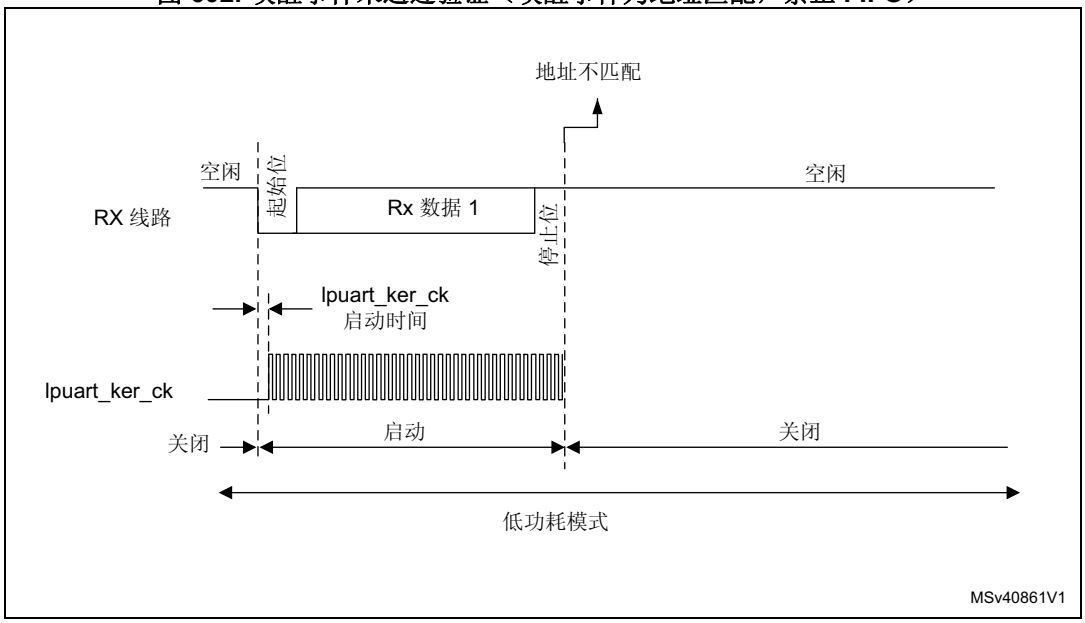


图 602 所示为唤醒事件未通过验证时的行为。

图 602. 唤醒事件未通过验证（唤醒事件为地址匹配，禁止 FIFO）



注： *将地址匹配或任一接收的帧用作唤醒事件时，如上两图适用。如果唤醒事件为起始位检测，则 LPUART 会在起始位结束时向 MCU 发送唤醒事件。*

49.4 LPUART 中断

有关所有 LPUART 中断请求的详细说明，请参见*表 386*。

表 386. LPUART 中断请求

中断事件	事件标志	使能控制位	中断清除方法	中断已激活	
				lpuart_it	lpuart_wkup
发送数据寄存器为空	TXE	TXEIE	TXE 在 TDR 中被写入数据时清零。	是	否
发送 FIFO 未满	TXFNF	TXFNFIE	TXFNF 在 TXFIFO 已满时清零。	是	否
发送 FIFO 为空	TXFE	TXFEIE	TXFE 在 TXFIFO 包含至少一个数据时清零，或通过将 TXFRQ 位置 1 来清零。	是	是
达到发送 FIFO 阈值	TXFT	TXFTIE	TXFT 在 TXFIFO 内容少于编程的阈值时由硬件清零。	是	是
CTS 中断	CTSIF	CTSIE	CTSIF 由软件通过将 CTSCF 位置 1 来清零。	是	否
发送完成	TC	TCIE	TC 在 TDR 中被写入数据时清零，或通过将 TCCF 位置 1 来清零。	是	否
接收数据寄存器不为空（已准备好读取数据）	RXNE	RXNEIE	RXNE 通过读取 RDR 或通过将 RXFRQ 位置 1 来清零。	是	是

表 386. LPUART 中断请求（续）

中断事件	事件标志	使能控制位	中断清除方法	中断已激活	
				lpuart_it	lpuart_wkup
接收 FIFO 非空	RXFNE	RXFNEIE	RXFNE 在 RXFIFO 为空时清零，或通过将 RXFRQ 位置 1 来清零。	是	是
接收 FIFO 已满	RXFF(1)	RXFFIE	RXFF 在 RXFIFO 至少包含一个数据时清零。	是	是
达到接收 FIFO 阈值	RXFT	RXFTIE	RXFT 在 RXFIFO 内容少于编程的阈值时由硬件清零。	是	是
检测到溢出错误	ORE	RXNEIE/ RXFNEIE	ORE 通过将 ORECF 位置 1 来清零。	是	否
检测到空闲线路	IDLE	IDLEIE	IDLE 通过将 IDLECF 位置 1 来清零。	是	否
奇偶校验错误	PE	PEIE	PE 通过将 PECF 位置 1 来清零。	是	否
多缓冲区通信中的噪声标志、溢出错误和帧错误。	NE 或 ORE 或 FE	EIE	NE 通过将 NFCF 位置 1 来清零。 ORE 通过将 ORECF 位置 1 来清零。 FE 标志通过将 FECF 位置 1 来清零	是	否
字符匹配	CMF	CMIE	CMF 通过将 CMCF 位置 1 来清零。	是	否
从低功耗模式唤醒	WUF(2)	WUFIE	WUF 通过将 WUCF 位置 1 来清零。	否	是
达到发送 FIFO 阈值	TXFT	TXFTIE	TXFT 在 TXFIFO 内容少于编程的阈值时由硬件清零	是	是
达到接收 FIFO 阈值	RXFT	RXFTIE	RXFT 在 RXFIFO 内容少于编程的阈值时由硬件清零	是	是

1. 如果 LPUART 接收 n+1 个数据（n 表示 RXFIFO 大小，RXFIFO 中有 n 个数据，LPUART_RDR 中有 1 个数据），则 RXFF 标志置位。在停止模式下，未向 LPUART_RDR 提供时钟。因此，将不会对该寄存器进行写操作，一旦有 n 个数据被接收并写入到 RXFIFO，RXFF 中断将生效（RXFF 标志未置 1）。
2. WUF 中断仅在低功耗模式下有效。

49.5 LPUART寄存器

有关寄存器说明中使用的缩写，请参见*第 94 页的第 1.1 节*。

49.5.1 控制寄存器 1 (LPUART_CR1)

Control register 1

偏移地址：0x00

复位值：0x0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RXFFIE	TXFEIE	FIFOEN	M1	Res.	Res.	DEAT[4:0]					DEDT[4:0]				
rw	rw	rw	rw			rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	CMIE	MME	M0	WAKE	PCE	PS	PEIE	TXEIETXFNFIE	TCIE	RXNEIERXFNEIE	IDLEIE	TE	RE	UESM	UE
	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31 **RXFFIE**：RXFIFO 变满时中断使能 (RXFIFO full interrupt enable)

此位由软件置 1 和清零。

0：禁止中断

1：当 LPUART_ISR 寄存器中的 RXFF=“1”时，生成 LPUART 中断

注： 禁止 FIFO 模式时，该位保留且必须保持复位值。

位 30 **TXFEIE**：TXFIFO 为空时中断使能 (TXFIFO empty interrupt enable)

此位由软件置 1 和清零。

0：禁止中断

1：当 LPUART_ISR 寄存器中的 TXFE=“1”时，生成 LPUART 中断

注： 禁止 FIFO 模式时，该位保留且必须保持复位值。

位 29 **FIFOEN**：FIFO 模式使能 (FIFO mode enable)

此位由软件置 1 和清零。

0：禁止 FIFO 模式。

1：使能 FIFO 模式。

位 28 **M1**：字长 (Word length)

此位必须与位 12 (M0) 搭配使用来确定字长度。该位由软件置 1 或清零。

M[1:0] = “00”： 1 个起始位，8 个数据位，n 个停止位

M[1:0] = “01”： 1 个起始位，9 个数据位，n 个停止位

M[1:0] = “10”： 1 个起始位，7 个数据位，n 个停止位

只有在禁止 LPUART（UE=“0”）时才能写入该位。

注： 7 位数据长度模式下，不支持智能卡模式、LIN 主模式和自动波特率（0x7F 帧和 0x55 帧检测）。

位 27:26 保留，必须保持复位值

位 25:21 **DEAT[4:0]**：驱动器使能 assertion time (Driver Enable assertion time)

该 5 位值用于定义激活 DE（启动器使能）信号与起始位开始间的时间。以 lpuart_ker_ck 时钟周期数表示。有关详细信息，请参见*第 48.5.20 节：RS232 硬件流控制和 RS485 驱动器使能*。

只有在禁止 LPUART（UE=”0 “）时才能写入此位域。

位 20:16 **DEDT[4:0]**：驱动器使能 deassertion time (Driver Enable deassertion time)

该 5 位值用于定义发送的消息中最后一个停止位结束与取消激活 DE（驱动器使能）信号间的时间。以 lpuart_ker_ck 时钟周期数表示。有关详细信息，请参见*第 49.3.13 节：RS232 硬件流控制和 RS485 驱动器使能*。

如果在 DEDT 时间内对 LPUART_TDR 寄存器执行写操作，则新数据仅在经过 DEDT 和 DEAT 时间后才会发送。

只有在禁止 LPUART（UE=”0“）时才能写入此位域。

位 15 保留，必须保持复位值

位 14 **CMIE**：字符匹配中断使能 (Character match interrupt enable)

此位由软件置 1 和清零。

0：禁止中断

1：如果 LPUART_ISR 寄存器中的 CMF 位置 1，则生成 LPUART 中断。

位 13 **MME**：静默模式使能 (Mute mode enable)

此位用于激活 LPUART 的静默模式功能，此位置 1 时，LPUART 可按 WAKE 位定义的方式在工作模式与静默模式之间切换。该位由软件置 1 和清零。

0：接收器永久处于工作模式

1：接收器可在静默模式和工作模式之间切换。

位 12 **M0**：字长 (Word length)

此位与位 28 (M1) 搭配使用来确定字长度。它由软件置 1 或清零（请参见位 28 (M1) 的说明）。

只有在禁止 LPUART（UE=“0”）时才能写入该位。

位 11 **WAKE**：接收器唤醒方法 (Receiver wakeup method)

此位用于确定 LPUART 静默模式的唤醒方法。该位由软件置 1 或清零。

0：空闲线路

1：地址标记

只有在禁止 LPUART（UE=”0“）时才能写入此位域。

位 10 **PCE**：奇偶校验控制使能 (Parity control enable)

该位选择硬件奇偶校验控制（生成和检测）。使能奇偶校验控制时，计算出的奇偶校验位被插入到 MSB 位置（如果 M=“1”，则为第 9 位；如果 M=“0”，则为第 8 位），并对接收到的数据检查奇偶校验位。此位由软件置 1 和清零。一旦该位置 1，PCE 在当前字节的后面处于活动状态（在接收和发送时）。

0：禁止奇偶校验控制

1：使能奇偶校验控制

只有在禁止 LPUART（UE=”0“）时才能写入此位域。

位 9 **PS**：奇偶校验选择 (Parity selection)

该位用于在使能奇偶校验生成/检测（PCE 位置 1）时选择奇校验或偶校验。该位由软件置 1 和清零。将在当前字节的后面选择奇偶校验。

0：偶校验

1：奇校验

只有在禁止 LPUART（UE=”0“）时才能写入此位域。

位 8 **PEIE**：PE 中断使能 (PE interrupt enable)

此位由软件置 1 和清零。

0：禁止中断

1：当 LPUART_ISR 寄存器中的 PE=“1”时，生成 LPUART 中断

位 7 **TXEIE/TXFNFIE**：发送数据寄存器为空/TXFIFO 未满中断使能 (Transmit data register empty/TXFIFO not full interrupt enable)

此位由软件置 1 和清零。

0：禁止中断

1：当 LPUART_ISR 寄存器中的 TXE/TXFNF =“1”时，生成 LPUART 中断

位 6 **TCIE**：传输完成中断使能 (Transfer complete interrupt enable)

此位由软件置 1 和清零。

0：禁止中断

1：当 LPUART_ISR 寄存器中的 TC=“1”时，生成 LPUART 中断

位 5 **RXNEIE/RXFNEIE**：接收数据寄存器非空/RXFIFO 非空中断使能 (Receive data register not empty/RXFIFO not empty interrupt enable)

此位由软件置 1 和清零。

0：禁止中断

1：当 LPUART_ISR 寄存器中的 ORE=“1”或 RXNE/RXFNE=“1”时，生成 LPUART 中断

位 4 **IDLEIE**：IDLE 中断使能 (IDLE interrupt enable)

此位由软件置 1 和清零。

0：禁止中断

1：当 LPUART_ISR 寄存器中的 IDLE=“1”时，生成 LPUART 中断

位 3 **TE**：发送器使能 (Transmitter enable)

该位使能发送器。该位由软件置 1 和清零。

0：禁止发送器

1：使能发送器

注：　传送期间 TE 位上的低电平脉冲（“0”后紧跟的是“1”）会在当前字的后面发送一个报头（空闲线路）。为生成空闲字符，TE 不能立即写入 1。为确保所需的持续时间，软件可轮询 LPUART_ISR 寄存器中的 TEACK 位。

当 TE 置 1 时，在发送开始前存在 1 位的时间延迟。

位 2 **RE**：接收器使能 (Receiver enable)

该位使能接收器。该位由软件置 1 和清零。

0：禁止接收器

1：使能接收器并开始搜索起始位

位 1 **UESM**：停止模式下的 LPUART 使能 (LPUART enable in Stop mode)

当此位清零时，LPUART 无法将 MCU 从低功耗模式唤醒。

当此位置 1 时，LPUART 能够将 MCU 从低功耗模式唤醒，前提是 LPUART 时钟选择为 HSI 或 LSE（在 RCC 中）。

此位由软件置 1 和清零。

0：LPUART 无法将 MCU 从低功耗模式唤醒。

1：LPUART 能够将 MCU 从低功耗模式唤醒。当该功能激活时，LPUART 的时钟源必须是 HSI 或 LSE（请参见 RCC 一章）。

注：　建议在进入低功耗模式前将 UESM 位置 1，并在退出低功耗模式时将其清零。

位 0 **UE**：LPUART 使能 (LPUART enable)

此位清零后，LPUART 预分频器和输出将立即停止，并丢弃当前操作。LPUART 的配置保留，但 LPUART_ISR 中的所有状态标志将被复位。此位由软件置 1 和清零。

0：禁止 LPUART 预分频器和输出，低功耗模式

1：使能 LPUART

注：　为进入低功耗模式而不在线路上生成错误，之前必须复位 TE 位，并且软件必须等待 LPUART_ISR 寄存器中的 TC 位置 1 后才能复位 UE 位。

UE = “0”时也会复位 DMA 请求，因此必须在复位 UE 位前禁止 DMA 通道。

49.5.2 控制寄存器 2 (LPUART_CR2)

Control register 2

偏移地址：0x04

复位值：0x0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ADD[7:4]				ADD[3:0]				Res.	Res.	Res.	Res.	MSBFIRST	DATAINV	TXINV	RXINV
rw	rw	rw	rw	rw	rw	rw	rw					rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SWAP	Res.	STOP[1:0]		Res.	Res.	Res.	Res.	Res.	Res.	Res.	ADDM7	Res.	Res.	Res.	Res.
rw		rw	rw								rw				

位 31:28 **ADD[7:4]**：LPUART 节点的地址 (Address of the LPUART node)

此位域用于指定 LPUART 节点的地址或要识别的字符代码。

它在多处理器通信时于静默模式或停止模式下使用，以通过 7 位地址标记检测唤醒 MCU。发送器发送字符的 MSB 应为 1。此位域还可用于正常接收和静默模式无效时的字符检测（例如，ModBus 协议中的块结束检测）。这种情况下，接收到的整个字符（8 位）将与 ADD[7:0] 值进行比较，如果匹配，CMF 标志将置 1。

仅在禁止接收（RE =“0”）或禁止 LPUART（UE=“0”）时才能写入该位域。

位 27:24 **ADD[3:0]**：LPUART 节点的地址 (Address of the LPUART node)

此位域用于指定 LPUART 节点的地址或要识别的字符代码。

它在多处理器通信时于静默模式或低功耗模式下使用，以通过地址标记检测进行唤醒。

仅在禁止接收（RE =“0”）或禁止 LPUART（UE=“0”）时才能写入该位域。

位 23:20 保留，必须保持复位值

位 19 **MSBFIRST**：最高有效位在前 (Most significant bit first)

此位由软件置 1 和清零。

0：发送/接收数据时位 0 在前，后跟起始位。

1：发送/接收数据时 MSB（位 7/8）在前，后跟起始位。

只有在禁止 LPUART（UE=”0“）时才能写入此位域。

位 18 **DATAINV**：二进制数据反向 (Binary data inversion)

此位由软件置 1 和清零。

0：按正/正向逻辑发送/接收数据寄存器中的逻辑数据。（1=H，0=L）

1：按负/反向逻辑发送/接收数据寄存器中的逻辑数据。（1=L，0=H）。奇偶校验位也取反。

只有在禁止 LPUART（UE=“0”）时才能写入此位域。

位 17 **TXINV**：TX 引脚有效电平反向 (TX pin active level inversion)

此位由软件置 1 和清零。

0：TX 引脚信号使用标准逻辑电平（V_{DD} = 1/空闲，Gnd = 0/标记）工作。

1：对 TX 引脚信号值取反。（V_{DD} = 0/标记，Gnd=1/空闲）。

允许在 TX 线路上使用外部反相器。

只有在禁止 LPUART（UE=“0”）时才能写入此位域。

位 16 **RXINV**：RX 引脚有效电平反向 (RX pin active level inversion)
此位由软件置 1 和清零。
0：RX 引脚信号使用标准逻辑电平（V_{DD} = 1/空闲，Gnd = 0/标记）工作。
1：对 RX 引脚信号值取反。（V_{DD} = 0/标记，Gnd=1/空闲）。
允许在 RX 线路上使用外部反相器。
只有在禁止 LPUART（UE=“0”）时才能写入此位域。

位 15 **SWAP**：交换 TX/RX 引脚 (Swap TX/RX pins)
此位由软件置 1 和清零。
0：按标准引脚排列定义使用 TX/RX 引脚。
1：交换 TX 和 RX 引脚功能。允许在与另一个 UART 的交叉连接时工作。
只有在禁止 LPUART（UE=“0”）时才能写入此位域。

位 14 保留，必须保持复位值

位 13:12 **STOP[1:0]**：停止位 (STOP bits)
这些位用于编程停止位。
00：1 个停止位
01：保留
10：2 个停止位
11：保留
只有在禁止 LPUART（UE=“0”）时才能写入此位域。

位 11:5 保留，必须保持复位值

位 4 **ADDM7**：7 位地址检测/4 位地址检测 (7-bit Address Detection/4-bit Address Detection)
此位用于选择 4 位地址检测或 7 位地址检测。
0：4 位地址检测
1：7 位地址检测（在 8 位数据模式下）
只有在禁止 LPUART（UE=“0”）时才能写入该位
注：　在 7 位和 9 位数据模式下，地址检测分别在 6 位和 8 位地址上完成（ADD[5:0] 和 ADD[7:0]）。

位 3:0 保留，必须保持复位值。

49.5.3 控制寄存器 3 (LPUART_CR3)

Control register 3

偏移地址：0x08

复位值：0x0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DEP	DEM	DDRE	OVRDIS	Res.	CTSIE	CTSE	RTSE	DMAT	DMAR	Res.	Res.	HDSEL	Res.	Res.	EIE
rw	rw	rw	rw		rw	rw	rw	rw	rw			rw			rw

位 31:29 **TXFTCFG**：TXFIFO 阈值配置 (TXFIFO threshold configuration)
000：TXFIFO 达到其深度的 1/8。
001：TXFIFO 达到其深度的 1/4。
110：TXFIFO 达到其深度的 1/2。
011：TXFIFO 达到其深度的 3/4。
100：TXFIFO 达到其深度的 7/8。
101：TXFIFO 变空。
其余组合：保留。

位 28 **RXFTIE**：RXFIFO 阈值中断使能 (RXFIFO threshold interrupt enable)。
此位由软件置 1 和清零。
0：禁止中断
1：当接收 FIFO 达到 RXFTCFG 中编程的阈值时，生成 LPUART 中断。

位 27:25 **RXFTCFG**：接收 FIFO 阈值配置 (Receive FIFO threshold configuration)
000：接收 FIFO 达到其深度的 1/8。
001：接收 FIFO 达到其深度的 1/4。
110：接收 FIFO 达到其深度的 1/2。
011：接收 FIFO 达到其深度的 3/4。
100：接收 FIFO 达到其深度的 7/8。
101：接收 FIFO 已满。
其余组合：保留。

位 24 保留，必须保持复位值。

位 23 **TXFTIE**：TXFIFO 阈值中断使能 (TXFIFO threshold interrupt enable)
此位由软件置 1 和清零。
0：禁止中断
1：当 TXFIFO 达到 TXFTCFG 中编程的阈值时，生成 LPUART 中断。

位 22 **WUFIE**：从低功耗模式唤醒中断使能 (Wakeup from low-power mode interrupt enable)
此位由软件置 1 和清零。
0：禁止中断
1：当 LPUART_ISR 寄存器中的 WUF=“1”时，生成 LPUART 中断
注： *WUFIE 必须在进入低功耗模式前置 1。*
WUF 中断仅在低功耗模式下有效。
如果 LPUART 不支持从停止模式唤醒功能，则此位保留并由硬件强制清零。
请参见第 48.4 节：USART 实现。

位 21:20 **WUS[1:0]**：从低功耗模式唤醒中断标志选择 (Wakeup from low-power mode interrupt flag selection)
该位域用于指定激活 WUF（从低功耗模式唤醒标志）的事件。
00：WUF 在地址匹配时激活（按 ADD[7:0] 和 ADDM7 所定义）
01：保留
10：WUF 在起始位检测时激活
11：WUF 在 RXNE 时激活
只有在禁止 LPUART（UE=“0”）时才能写入此位域。
注： *如果 LPUART 不支持从停止模式唤醒功能，则此位保留并由硬件强制清零。*
请参见第 48.4 节：USART 实现。

位 19:16 保留，必须保持复位值。

位 15 **DEP**：驱动器使能极性选择 (Driver enable polarity selection)

0：DE 信号高电平有效。

1：DE 信号低电平有效。

只有在禁止 LPUART（UE=“0”）时才能写入该位。

位 14 **DEM**：驱动器使能模式 (Driver enable mode)

此位用于通过 DE 信号激活外部收发器控制。

0：禁止 DE 功能。

1：使能 DE 功能。DE 信号在 RTS 引脚上输出。

只有在禁止 LPUART（UE=“0”）时才能写入该位。

位 13 **DDRE**：接收出错时的 DMA 禁止 (DMA Disable on Reception Error)

0：接收出错时不禁止 DMA。相应的错误标志置 1，但 RXNE 保持为 0 以防止上溢。因此，将不使能 DMA 请求，从而不会传送错误数据（无 DMA 请求），但会传送接收到的下一个正确数据。

1：接收出错后禁止 DMA。相应的错误标志以及 RXNE 均置 1。屏蔽 DMA 请求，直到错误标志清零。这意味着软件必须首先禁止 DMA 请求（DMAR =“0”）或者将 RXNE 清零，然后再将错误标志清零。

只有在禁止 LPUART（UE=“0”）时才能写入该位。

注： 接收错误包括：奇偶校验错误、帧错误或噪声错误。

位 12 **OVRDIS**：上溢禁止 (Overrun Disable)

此位用于禁止接收上溢检测。

0：接收新数据前未读取已接收的数据时，上溢错误标志 ORE 置 1。

1：禁止上溢功能。如果在 RXNE 标志仍置 1 时接收到新数据，

则 ORE 标志不会置 1，且新接收的数据会覆盖 LPUART_RDR 寄存器之前的内容。

只有在禁止 LPUART（UE=“0”）时才能写入该位。

注： 此控制位用于检查通信流而不会读取数据。

位 11 保留，必须保持复位值。

位 10 **CTSIE**：CTS 中断使能 (CTS interrupt enable)

0：禁止中断

1：当 LPUART_ISR 寄存器中的 CTSIF =“1”时，生成中断

位 9 **CTSE**：CTS 使能 (CTS enable)

0：禁止 CTS 硬件流控制

1：使能 CTS 模式，仅当 nCTS 输入有效（连接到 0）时才发送数据。如果在发送数据时使 nCTS 输入无效，会在停止之前完成发送。如果使 nCTS 有效时数据已写入数据寄存器，则将延迟发送，直到 nCTS 有效。

只有在禁止 LPUART（UE=“0”）时才能写入该位。

位 8 **RTSE**：RTS 使能 (RTS enable)

0：禁止 RTS 硬件流控制

1：使能 RTS 输出，仅当接收缓冲区中有空间时才会请求数据。发送完当前字符后应停止发送数据。可以接收数据时使 nRTS 输出有效（拉至 0）。

只有在禁止 LPUART（UE=“0”）时才能写入该位。

位 7 **DMAT**：DMA 使能发送器 (DMA enable transmitter)

该位由软件置 1/复位。

1：针对发送使能 DMA 模式

0：针对发送禁止 DMA 模式

位 6 **DMAR**：DMA 使能接收器 (DMA enable receiver)

该位由软件置 1/复位。

1：针对接收使能 DMA 模式

0：针对接收禁止 DMA 模式

位 5:4 保留，必须保持复位值。

位 3 **HDSEL**：半双工选择 (Half-duplex selection)

选择单线半双工模式
0：未选择半双工模式
1：选择半双工模式
只有在禁止 LPUART（UE=“0”）时才能写入该位。

位 2:1 保留，必须保持复位值。

位 0 **EIE**：错误中断使能 (Error interrupt enable)

如果出现帧错误、上溢错误或噪声标志（LPUART_ISR 寄存器中的 FE = “1”、ORE = “1”或 NE = “1”），则需要使用错误中断使能位来使能中断生成。
0：禁止中断
LPUART_ISR 寄存器中的 FE=“1”、ORE=“1”或 NE=“1”时生成中断。

49.5.4 波特率寄存器 (LPUART_BRR)

Baud rate register

只有在禁止 LPUART（UE=“0”）时才能写入此寄存器。在自动波特率检测模式下，该位由硬件自动更新。

偏移地址：0x0C

复位值：0x0000

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

位 31:20 保留，必须保持复位值。

位 19:0 **BRR[19:0]**

注： *禁止在 LPUART_BRR 寄存器中写入小于 0x300 的值。*

如果 LPUART_BRR 必须 ≥ 0x300 且 LPUART_BRR 为 20 位，则使用高 fck 值生成高波特率时应十分谨慎。fck 必须在 [3 x 波特率到 4096 x 波特率] 的范围内。

49.5.5 请求寄存器 (LPUART_RQR)

Request register

偏移地址：0x18

复位值：0x0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TXFRQ	RXFRQ	MMRQ	SBKRQ	Res.
											w	w	w	w	

位 31:5 保留，必须保持复位值

位 4 **TXFRQ**：发送数据刷新请求 (Transmit data flush request)

该位在 FIFO 模式使能时使用。TXFRQ 位置 1 以清空整个 FIFO。这会将标志 TXFE（TXFIFO 为空，LPUART_ISR 寄存器中的位 23）置位。

注： 在 FIFO 模式下，TXFNF 标志在清空请求期间复位，直到 TxFIFO 为空，以确保数据寄存器中没有写入数据。

位 3 **RXFRQ**：接收数据刷新请求 (Receive data flush request)

向该位写入“1”时会将 RXNE 标志清零。

这可以丢弃接收的数据而不对其执行读取操作，并避免发生上溢。

位 2 **MMRQ**：静默模式请求 (Mute mode request)

向此位写入“1”可将 LPUART 置于静默模式，并将 RWU 标志复位。

位 1 **SBKRQ**：发送中断请求 (Send break request)

向此位写入“1”可将 SBKF 标志置 1 并在发送设备可用后立即请求在线路上发送 BREAK。

注： 如果应用需要在之前插入的所有数据（包括尚未发送的数据）后发送 break 字符，软件应等到 TXE 标志使能后将 SBKRQ 位置 1。

位 0 保留，必须保持复位值

49.5.6 中断和状态寄存器 (LPUART_ISR)

Interrupt & status register

偏移地址：0x1C

复位值：0x00C0（如果禁止 FIFO）

复位值：0x08000C0（如果使能 FIFO）

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	TXFT	RXFT	Res.	RXFF	TXFE	RE ACK	TE ACK	WUF	RWU	SBKF	CMF	BUSY
				r	r		r	r	r	r	r	r	r	r	r

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	CTS	CTSIF	Res.	TXE	TC	RXNE	IDLE	ORE	NE	FE	PE
					r	r		r	r	r	r	r	r	r	r

位 31:28 保留，必须保持复位值。

位 27 **TXFT**：TXFIFO 阈值标志 (TXFIFO threshold flag)

当 TXFIFO 达到在 LPUART_CR3 寄存器的 TXFTCFG 中编程的阈值时（即 TXFIFO 包含 TXFTCFG 个空位置），该位由硬件置 1。如果 LPUART_CR3 寄存器中的 TXFTIE 位 =“1”（位 31），则会生成中断。

0：TXFIFO 未达到编程的阈值。

1：TXFIFO 已达到编程的阈值。

位 26 **RXFT**：RXFIFO 阈值标志 (RXFIFO threshold flag)

当 RXFIFO 达到在 LPUART_CR3 寄存器的 RXFTCFG 中编程的阈值时（即接收 FIFO 包含 RXFTCFG 个数据），该位由硬件置 1。如果 LPUART_CR3 寄存器中的 RXFTIE 位 =“1”（位 27），则会生成中断。

0：接收 FIFO 未达到编程的阈值。

1：接收 FIFO 已达到编程的阈值。

位 25 保留，必须保持复位值。

位 24 **RXFF**：RXFIFO 已满 (RXFIFO Full)

当接收到的数据量对应于 RXFIFO 大小 + 1（RXFIFO 已满 + LPUART_RDR 寄存器中的 1 个数据）时，该位由硬件置 1。

如果 LPUART_CR1 寄存器中的 RXFFIE 位 =“1”，则会生成中断。

0：RXFIFO 未满。

1：RXFIFO 已满

位 23 **TXFE**：TXFIFO 为空 (TXFIFO Empty)

当 TXFIFO 为空时，该位由硬件置 1。当 TXFIFO 包含至少一个数据时，该标志被清零。也可以通过向 LPUART_RQR 寄存器中的位 TXFRQ（位 4）写入“1”将 TXFE 标志置 1。

如果 LPUART_CR1 寄存器中的 TXFEIE 位 =“1”（位 30），则会生成中断。

0：TXFIFO 非空。

1：TXFIFO 为空。

位 22 **REACK**：接收使能确认标志 (Receive enable acknowledge flag)

LPUART 采用接收使能值时，通过硬件将此位置 1/复位。

此位可用于验证 LPUART 是否准备好在进入低功耗模式前接收数据。

注： 如果 LPUART 不支持从停止模式唤醒功能，则此位保留并由硬件强制清零。

位 21 **TEACK**：发送使能确认标志 (Transmit enable acknowledge flag)

LPUART 采用发送使能值时，通过硬件将此位置 1/复位。

通过写入 TE=“0”生成空闲帧请求，然后在 LPUART_CR1 寄存器中写入 TE=“1”以遵循 TE=“0”最短周期时，可使用此位。

位 20 **WUF**：从低功耗模式唤醒标志 (Wakeup from low-power mode flag)

当检测到唤醒事件时，此位由硬件置 1。事件通过 WUS 位域定义。通过向 LPUART_ICR 寄存器中的 WUCF 写入 1，此位由软件清零。

如果 LPUART_CR3 寄存器中的 WUFIE=“1”，则会生成中断。

注： 当 UESM 清零时，WUF 标志也清零。

WUF 中断仅在低功耗模式下有效。

如果 LPUART 不支持从停止模式唤醒功能，则此位保留并由硬件强制清零。

位 19 **RWU**：*接收器从静默模式唤醒 (Receiver wakeup from Mute mode)*

此位指示 LPUART 是否处于静默模式。当识别出唤醒/静默序列时，此位由硬件清零/置 1。静默模式控制序列（地址或 IDLE）通过 LPUART_CR1 寄存器中的 WAKE 位选择。

当选择 IDLE 模式下唤醒时，该位只能通过用软件向 LPUART_RQR 寄存器中的 MMRQ 位写“1”的方式置 1。

0：接收器处于工作模式

1：接收器处于静默模式

注： 如果 LPUART 不支持从停止模式唤醒功能，则此位保留并由硬件强制清零。

位 18 **SBKF**：发送 break 标志 (Send break flag)

此位指示已请求发送 break 字符。此位由软件置 1，方法是向 LPUART_CR3 寄存器中的 SBKRQ 位写入“1”。此位在中断发送的停止位期间由硬件自动复位。

0：不发送 break 字符

1：将发送 break 字符

位 17 **CMF**：字符匹配标志 (Character match flag)

接收到由 ADD[7:0] 定义的字符后由硬件将此位置 1。此位由软件清零，方法是向 LPUART_ICR 寄存器中的 CMCF 写入“1”。

如果 LPUART_CR1 寄存器中的 CMIE=“1”，则会生成中断。

0：未检测到字符匹配

1：检测到字符匹配

位 16 **BUSY**：忙标志 (Busy flag)

此位由硬件置 1 和复位。当 RX 线路上正在进行通信（成功检测到起始位）时有效。在接收结束（成功或失败）时复位。

0：LPUART 处于空闲状态（无接收）

1：正在接收

位 15:11 保留，必须保持复位值。

位 10 **CTS**：CTS 标志 (CTS flag)

此位由硬件置 1/复位。此位是对 nCTS 输入引脚的状态取反。

0：nCTS 线置 1

1：nCTS 线复位

注： 如果不支持硬件流控制功能，该位保留并由硬件强制清零。

位 9 **CTSIF**：CTS 中断标志 (CTS interrupt flag)

如果 CTSE 位置 1，当 nCTS 输入切换时，此位由硬件置 1。此位由软件清零，方法是向 LPUART_ICR 寄存器中的 CTSCF 位写入“1”。

如果 LPUART_CR3 寄存器中的 CTSIE=“1”，则会生成中断。

0：nCTS 状态线上未发生变化

1：nCTS 状态线上发生变化

注： 如果不支持硬件流控制功能，该位保留并由硬件强制清零。

位 8 保留，必须保持复位值。

位 7 **TXE/TXFNF**：发送数据寄存器为空/TXFIFO 未满 (Transmit data register empty/TXFIFO not full)

如果禁止 FIFO 模式，则当 LPUART_TDR 寄存器的内容已传输到移位寄存器时，TXE 由硬件置 1。通过对 LPUART_TDR 寄存器执行写入操作将该位清零。

如果使能 FIFO 模式，TXFNF 在 TXFIFO 未满时由硬件置 1，并因此可向 LPUART_TDR 中写入数据。每次对 LPUART_TDR 进行写操作都会将数据置于 TXFIFO 中。该标志保持置 1，直到 TXFIFO 已满。当 TXFIFO 已满时，该标志清零，表示不能向 LPUART_TDR 写入数据。

注：在清空请求期间，TXFNF 保持复位，直到 TXFIFO 为空。发送清空请求（通过将 TXFRQ 位置 1）后，应先检查 TXFNF 标志，然后再写入 TXFIFO（TXFNF 和 TXFE 将同时置 1）。

如果 LPUART_CR1 寄存器中的 TXEIE/TXFNFIE 位 =“1”，则会生成中断。

0：数据寄存器已满/发送 FIFO 已满。

1：数据寄存器/发送 FIFO 未满。

注： 单缓冲区发送期间使用该位。

位 6 **TC**：发送完成 (Transmission complete)

如果已完成对包含数据的帧的发送并且 TXE/TXFF 置 1，则该位由硬件置 1。

如果 LPUART_CR1 寄存器中的 TCIE =“1”，则会生成中断。此位由软件清零，方法是向 LPUART_ICR 寄存器中的 TCCF 写入“1”或向 LPUART_TDR 寄存器执行写操作。

如果 LPUART_CR1 寄存器中的 TCIE =“1”，则会生成中断。

0：传送未完成

1：传送已完成

注： 如果 TE 位复位且无任何发送正在进行，TC 位会立即置 1。

位 5 **RXNE/RXFNE**：读取数据寄存器非空/RXFIFO 非空 (Read data register not empty/RXFIFO not empty)

当 LPUART_RDR 移位寄存器的内容已传输到 LPUART_RDR 寄存器时，RXNE 位由硬件置 1。通过对 LPUART_RDR 寄存器执行读入操作将该位清零。

也可以通过向 LPUART_RQR 寄存器中的 RXFRQ 写入“1”来将 RXNE 标志清零。

RXFIFO 非空时，RXFNE 位由硬件置 1，并因此可以从 LPUART_RDR 寄存器读取数据。每次对 LPUART_RDR 进行读操作都会在 RXFIFO 中释放一个位置。它在 RXFIFO 为空时清零。

也可以通过向 LPUART_RQR 寄存器中的 RXFRQ 写入“1”将 RXNE/RXFNE 标志清零。

如果 LPUART_CR1 寄存器中的 RXNEIE/RXFNEIE=“1”，则会生成中断。

0：未接收到数据

1：已准备好读取接收到的数据

位 4 **IDLE**：检测到空闲线路 (IDLE line detected)

检测到空闲线路时，该位由硬件置 1。如果 LPUART_CR1 寄存器中的 IDLEIE=“1”，则会生成中断。此位由软件清零，方法是向 LPUART_ICR 寄存器中的 IDLECF 写入“1”。

0：未检测到空闲线路

1：检测到空闲线路

注： 直到 RXNE 位已置 1 时（即，当出现新的空闲线路时）IDLE 位才会被再次置 1。

使能静默模式（MME=“1”）后，如果 LPUART 未静默（RWU=“0”），则 IDLE 置 1，无论是否通过 WAKE 位选择了静默模式。如果 RWU=“1”，IDLE 不置 1。

位 3 **ORE**：溢出错误 (Overrun error)

在 RXNE =“1”（使能 FIFO 模式时为 RXFF =“1”）的情况下，当移位寄存器中当前正在接收的数据准备好传输到 LPUART_RDR 寄存器时，此位由硬件置 1。此位由软件清零，方法是向 LPUART_ICR 寄存器中的 ORECF 写入“1”。

如果 LPUART_CR1 寄存器中的 RXNEIE/RXFNEIE=“1”或 EIE =“1”，则会生成中断。

0：无溢出错误

1：检测到溢出错误

注：　当此位置 1 时，LPUART_RDR 寄存器的内容不会丢失，但移位寄存器会被覆盖。EIE 位置 1 后，如果在多缓冲区通信中 ORE 标志置 1，则会生成中断。

LPUART_CR3 寄存器中的 OVRDIS 位置 1 时，此位将被永久强制清零（无上溢检测）。

位 2 **NE**：START 位噪声检测标志 (START bit Noise detection flag)

当在接收的帧的起始位上检测到噪声时，此位由硬件置 1。此位由软件清零，方法是向 LPUART_ICR 寄存器中的 NFCF 位写入“1”。

0：未检测到噪声

1：检测到噪声

注：　该位不会生成中断，因为该位出现的时间与本身生成中断的 RXNE/RXFNE 位出现的时间相同。EIE 位置 1 后，如果在多缓冲区通信中 NE 标志置 1，则会生成中断。

在 FIFO 模式下，此错误与 LPUART_RDR 中的字符相关联。

位 1 **FE**：帧错误 (Framing error)

当检测到去同步化、过度的噪声或 break 字符时，该位由硬件置 1。此位由软件清零，方法是向 LPUART_ICR 寄存器中的 FECF 位写入“1”。

在智能卡模式下发送数据时，如果在达到最大发送尝试次数后仍未成功（智能卡向数据帧发送 NACK 信号），则此位置 1。

如果 LPUART_CR1 寄存器中 EIE = 1，则会生成中断。

0：未检测到帧错误

1：检测到帧错误或 break 字符

注：　在 FIFO 模式下，此错误与 LPUART_RDR 中的字符相关联。

位 0 **PE**：奇偶校验错误 (Parity error)

当在接收器模式下发生奇偶校验错误时，该位由硬件置 1。此位由软件清零，方法是向 LPUART_ICR 寄存器中的 PECF 写入“1”。

如果 LPUART_CR1 寄存器中的 PEIE =“1”，则会生成中断。

0：无奇偶校验错误

1：奇偶校验错误

注：　在 FIFO 模式下，此错误与 LPUART_RDR 中的字符相关联。

49.5.7 中断标志清零寄存器 (LPUART_ICR)

Interrupt flag clear register

偏移地址：0x20

复位值：0x0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WUCF	Res.	Res.	CMCF	Res.
											w_r0			w_r0	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	CTSCF	Res.	Res.	TCCF	Res.	IDLECF	ORECF	NECF	FECF	PECF
						w_r0			w_r0		w_r0	w_r0	w_r0	w_r0	w_r0

位 31:21 保留，必须保持复位值。

位 20 **WUCF**：从低功耗模式唤醒清零标志 (Wakeup from low-power mode clear flag)
向此位写入“1”时，LPUART_ISR 寄存器中的 WUF 标志将清零。
注： 如果 LPUART 不支持从停止模式唤醒功能，则此位保留并由硬件强制清零。请参见第 48.4 节：USART 实现。

位 19:18 保留，必须保持复位值。

位 17 **CMCF**：字符匹配清零标志 (Character match clear flag)
向此位写入“1”时，LPUART_ISR 寄存器中的 CMF 标志将清零。

位 16:10 保留，必须保持复位值。

位 9 **CTSCF**：CTS 清零标志 (CTS clear flag)
向此位写入“1”时，LPUART_ISR 寄存器中的 CTSIF 标志将清零。

位 8:7 保留，必须保持复位值。

位 6 **TCCF**：发送完成清零标志 (Transmission complete clear flag)
向此位写入“1”时，LPUART_ISR 寄存器中的 TC 标志将清零。

位 5 保留，必须保持复位值。

位 4 **IDLECF**：检测到空闲线路清零标志 (Idle line detected clear flag)
向此位写入“1”时，LPUART_ISR 寄存器中的 IDLE 标志将清零。

位 3 **ORECF**：上溢错误清零标志 (Overrun error clear flag)
向此位写入“1”时，LPUART_ISR 寄存器中的 ORE 标志将清零。

位 2 **NECF**：检测到噪声清零标志 (Noise detected clear flag)
向此位写入“1”时，LPUART_ISR 寄存器中的 NE 标志将清零。

位 1 **FECF**：帧错误清零标志 (Framing error clear flag)
向此位写入“1”时，LPUART_ISR 寄存器中的 FE 标志将清零。

位 0 **PECF**：奇偶校验错误清零标志 (Parity error clear flag)
向此位写入“1”时，LPUART_ISR 寄存器中的 PE 标志将清零。

49.5.8 接收数据寄存器 (LPUART_RDR)

Receive data register

偏移地址：0x24

复位值：未定义

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

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

位 31:9 保留，必须保持复位值。

位 8:0 **RDR[8:0]**：接收数据值 (Receive data value)

包含接收到的数据字符。

RDR 寄存器在输入移位寄存器和内部总线之间提供了并行接口（请参见*图 589*）。

在使能奇偶校验位的情况下进行接收时，从 MSB 位中读取的值为接收到的奇偶校验位。

49.5.9 发送数据寄存器 (LPUART_TDR)

Transmit data register

偏移地址：0x28

复位值：未定义

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

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

位 31:9 保留，必须保持复位值。

位 8:0 **TDR[8:0]**：发送数据值 (Transmit data value)

包含要发送的数据字符。

TDR 寄存器在内部总线和输出移位寄存器之间提供了并行接口（请参见*图 589*）。

在使能奇偶校验位的情况下（LPUART_CR1 寄存器中的 PCE 位被置 1）进行发送时，由于 MSB 的写入值（位 7 或位 8，具体取决于数据长度）会被奇偶校验位所取代，因此该值不起任何作用。

注：　只能在 TXE/TXFNF=“1”时写入此寄存器。

49.5.10 预分频器寄存器 (LPUART_PRESC)

Prescaler register

只有在禁止 LPUART（UE=“0”）时才能写入此寄存器。

偏移地址：0x2C

复位值：0x0000

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

位 31:4 保留，必须保持复位值。

位 3:0 **PRESCALER[3:0]**：时钟预分频器 (Clock prescaler)

LPUART 输入时钟可通过预分频器进行分频：

0000：输入时钟未分频
0001：输入时钟 2 分频
0010：输入时钟 4 分频
0011：输入时钟 6 分频
0100：输入时钟 8 分频
0101：输入时钟 10 分频
0110：输入时钟 12 分频
0111：输入时钟 16 分频
1000：输入时钟 32 分频
1001：输入时钟 64 分频
1010：输入时钟 128 分频
1011：输入时钟 256 分频
其余组合：保留。

注： 为PRESCALER 编程不允许的值时，编程的预分频值将为«1011»，即输入时钟除以 256。

49.5.11 LPUART 寄存器映射

下表提供了 LPUART 寄存器映射和复位值。

表 387. LPUART 寄存器映射和复位值

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	LPUART_CR1	RXFFIE	TXFEIE	FIFOEN	M1	Res.	Res.	DEAT4	DEAT3	DEAT2	DEAT1	DEAT0	DEDT4	DEDT3	DEDT2	DEDT1	DEDT0	Res.	CMIE	MME	M	WAKE	PCE	PS	PEIE	TXEIE	TCIE	RXNEIE	IDLEIE	TE	RE	UESM	UE
	Reset value	0	0	0	0			0	0	0	0	0	0	0	0	0	0		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x04	LPUART_CR2	ADD[7:4]				ADD[3:0]				Res.	Res.	Res.	Res.	MSBFIRST	DATAINV	TXINV	RXINV	SWAP	Res.	STOP [1:0]		Res.	Res.	Res.	Res.	Res.	Res.	Res.	ADDM7	Res.	Res.	Res.	Res.
	Reset value	0	0	0	0	0	0	0	0					0	0	0	0	0		0	0								0				
0x08	LPUART_CR3	TXFT CFG			RXFTIE	RXFT CFG			Res.	TXFTIE	WUFIE	WUS [1:0]		Res.	Res.	Res.	Res.	DEP	DEM	DDRE	OVRDIS	Res.	CTSIE	CTSE	RTSE	DMAT	DMAR	Res.	Res.	HDSEL	Res.	Res.	EIE
	Reset value	0	0	0	0	0	0	0		0	0	0	0					0	0	0	0		0	0	0	0	0			0			0
0x0C	LPUART_BRR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BRR[19:0]																			
	Reset value																		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x10-0x14	Reserved																																
0x18	LPUART_RQR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TXFRQ	RXFRQ	MMRQ	SBKRQ	Res.
	Reset value																												0	0	0	0	
0x1C	LPUART_ISR	Res.	Res.	Res.	Res.	TXFT	RXFT	Res.	RXFF	TXFF	REACK	TEACK	WUF	RWU	SBKF	CMF	BUSY	Res.	Res.	Res.	Res.	Res.	CTS	CTSIF	Res.	TXE	TC	RXNE	IDLE	ORE	NE	FE	PE
	Reset value					0	0		0	0	0	0	0	0	0	0	0						0	0		1	0	0	0	0	0	0	0
0x20	LPUART_ICR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WUCF	Res.	Res.	CMCF	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CTSCF	Res.	Res.	TCCF	Res.	IDLECF	ORECF	NECF	FECF	PECF
	Reset value												0			0								0			0		0	0	0	0	0
0x24	LPUART_RDR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RDR[8:0]								
	Reset value																								0	0	0	0	0	0	0	0	0
0x28	LPUART_TDR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TDR[8:0]								
	Reset value																								0	0	0	0	0	0	0	0	0
0x2C	LPUART_PRESC	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PRESCALER[3:0]			
	Reset value																													0	0	0	0

有关寄存器边界地址的信息，请参见*第 100 页的第 2.2.2 节*。

50 串行外设接口 (SPI)

50.1 前言

串行外设接口 (SPI) 可使用特定同步协议与外部器件进行通信。(SPI) 接口支持与外部器件进行半双工、全双工和单工同步串行通信。该接口可配置为主模式或从模式，并且能够在多从或多主配置中运行。在配置为主器件时，它为外部从器件提供通信时钟 (SCK)。从器件选择信号可以由主器件提供，也可以选择由从器件接收。默认情况下使用 Motorola 数据格式，但也支持某些其他特定模式。

50.2 SPI 主要特性

- 基于三条线的全双工同步传输
- 基于双线的半双工同步传输，其中一条可作为双向数据线
- 基于双线的单工同步传输，其中一条可作为单向数据线
- 数据大小可从 4 位到 32 位
- 多主或多从模式功能
- 双时钟域，外设内核时钟可以独立于 PCLK
- 8 个主模式波特率预分频器，可达内核频率的一半
- 从模式频率可达 IP 内核频率的一半
- 保护配置和设置
- 对于主模式和从模式都可以通过硬件或者软件进行 SS 管理
- 数据之间的最小延时，以及 SS 与数据流之间的最小延时均可调
- 可配置的 SS 信号极性和时序，MISO x MOSI 交换功能
- 可调节的主器件接收器采样时间
- 可编程的时钟极性和相位
- 可编程的数据顺序，最先移位 MSB 或 LSB
- 可以对传输的数据量进行编程，以控制 SS 和 CRC
- 可触发中断的专用发送和接收标志
- 停止模式（未向外设 IP 提供时钟）下从器件发送和/或接收功能，以及唤醒功能
- SPI Motorola 和 TI 格式支持
- 用于确保可靠通信的硬件 CRC 功能：
  - 在发送模式下可以添加 CRC 校验值
  - 在接收模式下，自动进行 CRC 错误校验
- 可触发中断的主模式故障、上溢或下溢标志以及 CRC 错误检测
- 具有 DMA 功能的两个 16x 或 8x 8 位的内置 Rx 和 Tx FIFO
- 可编程的传输数据量
- 可配置的 FIFO 阈值（数据打包）
- 从模式下，下溢条件可配置（支持级联循环缓冲区）

50.3 SPI 实现

表 388. STM32F7xx SPI 特性

SPI 模式/特性	SPI2S1	SPI2S2	SPI2S3	SPI4	SPI5	SPI6
Rx 和 TxFIFO 大小 (N) [x 8 位]	16	16	16	8	8	8
最大可配置数据大小 [位]	32	32	32	16	16	16
I2S 功能	有	有	有	无	无	无

50.4 SPI 功能说明

50.4.1 SPI 框图

SPI 支持在 MCU 与外部器件之间进行同步串行通信。应用软件可通过轮询状态标志或使用专用 SPI 中断对通信进行管理。SPI 的主要组件及其交互方式如以下框图所示（*图 603*）。

图 603. SPI2S框图

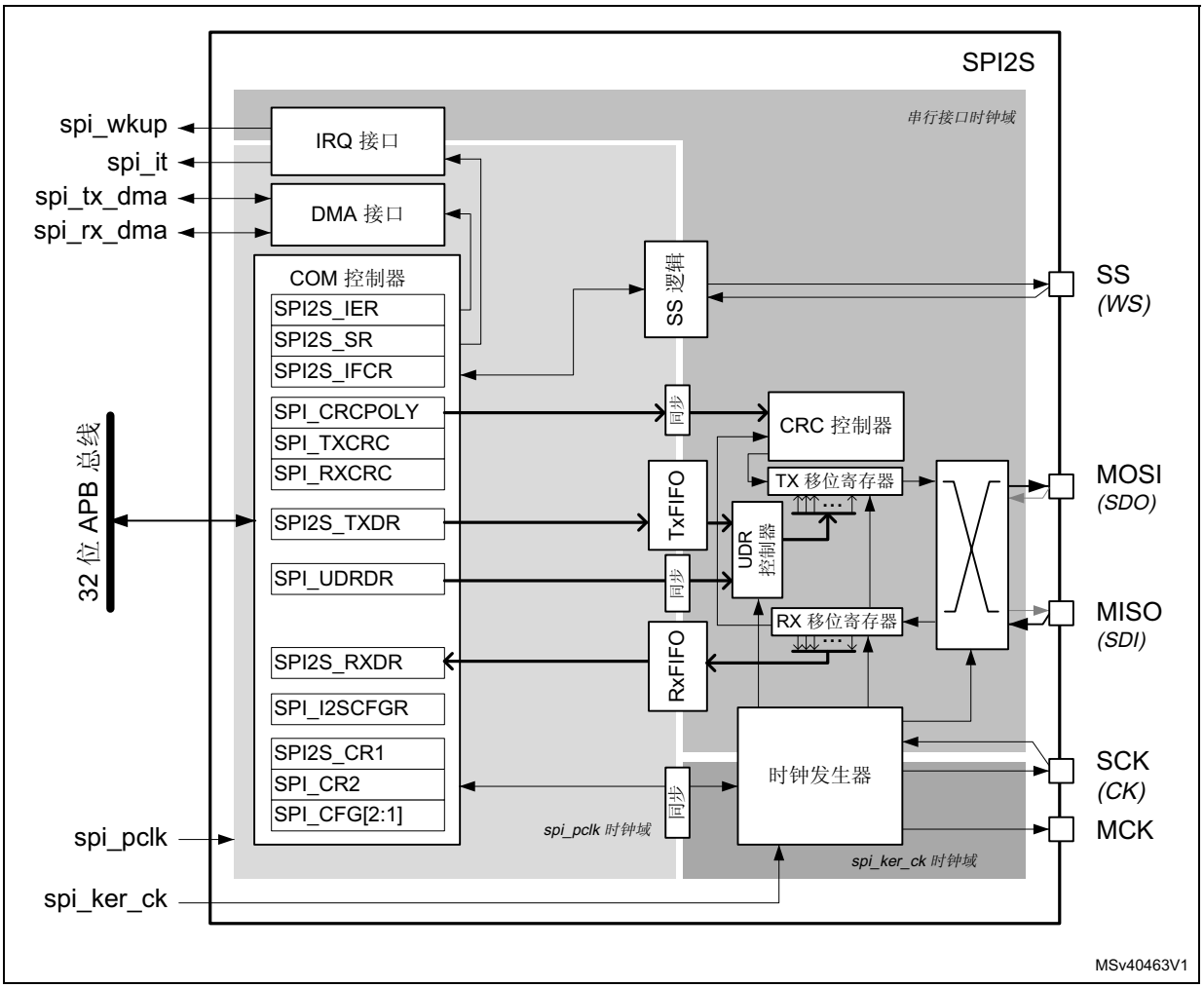


图 603 的简图显示了三个完全独立的时钟域：

- **spi_pclk** 时钟域
- **spi_ker_ck** 内核时钟域
- 串行接口时钟域

这些域之间的所有控制和状态信号均严格同步。这些时钟信号之间的频率比没有特定限制。用户必须考虑使用与数据流速度兼容的比率，以避免数据下溢或上溢。

spi_pclk 时钟信号给外设总线（APB 总线）供时钟。需要访问 SPI 寄存器时，该时钟必须有效。

SPI 工作在从模式下进行数据传输时，由 SPI 主器件提供时钟 SCK 信号。在这种情况下，即使 **spi_pclk** 和 **spi_ker_ck** 时钟信号无效，SPI 从器件也能接收和发送数据。

这不适用于 SPI 配置为主器件时的情形，因为此时 SPI 主器件需要来自 RCC 的 **spi_ker_ck** 内核时钟来提供其工作时钟。另一方面，工作在从模式下的 SPI，如果需要置寄存器的标志位，那么需要提供外设总线时钟（例如：出现 FIFO 上溢或者下溢的时候）。总线进入空闲状态时，不能进行上述过程。在特定情况下，从器件甚至需要时钟发生器工作（请参见 *第 50.5.1 节：TI 模式*）。

50.4.2 SPI 信号

四个 I/O 引脚专用于与外部器件进行 SPI 通信。

- **MISO**：主输入/从输出数据。通常情况下，此引脚用于在从模式下发送数据和在主模式下接收数据。
- **MOSI**：主输出/从输入数据。通常情况下，此引脚用于在主模式下发送数据和在从模式下接收数据。
- **SCK**：SPI 主器件的串行时钟输出引脚以及 SPI 从器件的串行时钟输入引脚。
- **SS**：从器件选择引脚。根据 SPI 和 SS 设置，该引脚可用于：
  - 选择单个从器件以进行通信
  - 同步数据帧或
  - 检测多个主器件之间是否存在冲突

  详细信息，请参见*第 50.4.6 节：多主通信*。

SPI 总线支持一个主器件与一个或多个从器件之间进行通信。该总线至少由两条线构成：一条用于时钟信号，另一条用于同步数据传输。其它信号可以根据 SPI 节点间的数据交换及其从器件选择信号管理进行添加。MOSI 和 MISO 引脚之间的功能可以在任意 SPI 模式下反相（请参见 SPI_CFG2 寄存器的 IOSWP 位）。

50.4.3 SPI 通信一般情况

SPI 支持 MCU 基于目标器件和应用要求使用不同的配置进行通信。这些配置使用 2 条或 3 条线（通过软件 SS 管理），也可以使用 3 条或 4 条线（通过硬件 SS 管理）。通信始终由主器件发起和控制。主器件基于 SCK 线提供时钟信号，并在由硬件进行管理时通过 SS 线选择或同步从器件以实现通信。主从器件之间通过 MOSI 和/或 MISO 线实现数据流动。

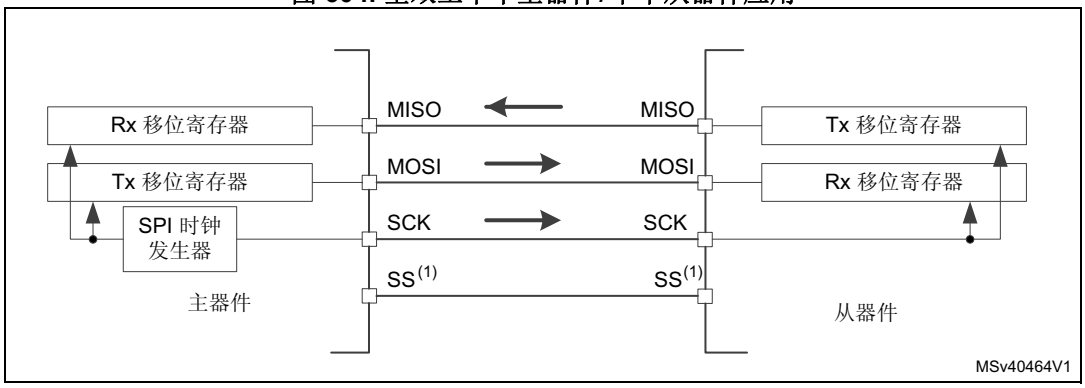
50.4.4 一个主器件和一个从器件之间的通信

通信流可使用以下 3 种模式之一：全双工（3 线）、半双工（2 线）或单工（2 线）。SS 信号在单一主从器件配置下是可选信号，并且一般不使用。不过，SS 信号在此配置下可有助于同步数据流，并在默认情况下用于某些特定 SPI 模式（例如，TI 模式）。

全双工通信

默认情况下，SPI 配置为全双工通信（SPI_CFG2 寄存器中的位 COMM[1:0] = 00）。在这种配置下，主器件和从器件的移位寄存器通过 MOSI 和 MISO 引脚之间的两条单向线连接。在 SPI 通信过程中，数据随主器件提供的 SCK 时钟边沿同步移位。主器件通过 MOSI 线将待发送的数据发送给从器件，同时通过 MISO 线从从器件接收数据。当数据帧传输完成时（所有位均移出），主器件和从器件之间即完成信息交换。

图 604. 全双工单个主器件 / 单个从器件应用

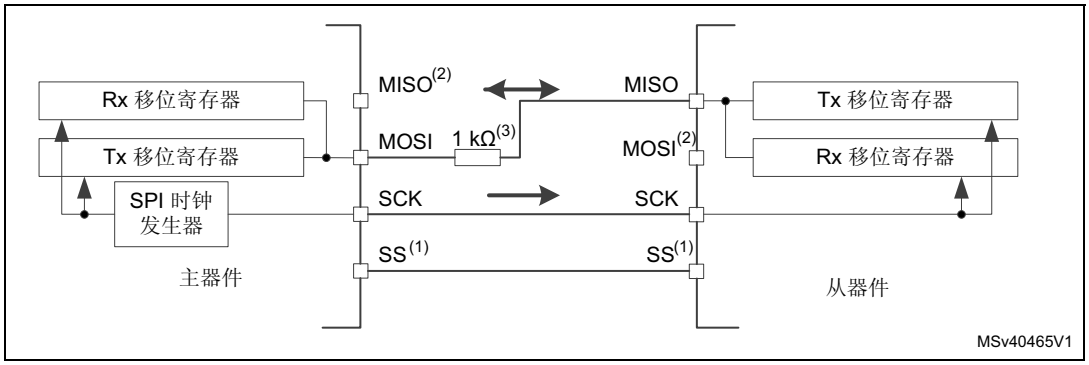


1. 主器件的 SS 引脚配置为输出模式。SS 引脚可保持未连接状态，之后主器件和从器件的 SS 都由软件在内部进行管理。

半双工通信

通过设置 SPI_CFG2 寄存器中的 COMM[1:0] = 11，SPI 可采用半双工模式进行通信。在这种配置下，使用一条交叉连接线将主器件和从器件的移位寄存器连接起来。在此通信过程中，数据随 SCK 时钟边沿在移位寄存器之间进行移位，传输方向由主器件和从器件通过各自 SPI_CR1 寄存器中的 HDDIR 位进行选择。请注意，更改通信方向并且主器件传输正在进行，则只发送模式下的主器时，必须禁止 SPI。在该配置下，主器件的 MISO 引脚和从器件的 MOSI 引脚空闲，可在其他应用中用作 GPIO。

图 605. 半双工单个主器件 / 单个从器件应用



1. 主器件的 SS 引脚配置为输出模式。引脚可保持未连接状态，之后主器件和从器件的 SS 都由软件在内部进行管理。
2. 在该配置下，主器件的 MISO 引脚和从器件的 MOSI 引脚可用作 GPIO
3. 当以双向模式工作的两个节点间的通信方向不是同步变化时，会出现临界情况，新发送器访问共用数据线，而前一个发送器仍保持线路上的相反值（值取决于 SPI 配置和通信数据）。两个节点会出现冲突，在共用线上短暂提供相反的输出电平，直到下一个节点也相应地改变其方向设置。建议此模式下在 MISO 和 MOSI 引脚之间插入串行电阻以在这种情况下保护输出并限制电流在二者之间流过。

单工通信

通过 SPI_CFG2 寄存器中的 COMM[1:0] 字段将 SPI 设置为只发送模式或只接收模式，可使 SPI 以单工模式进行通信。在这种配置下，仅使用一条线在主器件和从器件的移位寄存器之间进行传输。其余 MISO 或 MOSI 引脚对不用于通信，可用作标准 GPIO。

- 只发送模式：COMM[1:0] = 01

  只要 TxFIFO 中存在可用数据，并且主器件传输正在进行，只发送模式下的主器件就会生成时钟。

  如果只发送模式下的从器件接收到 SCK 引脚上的时钟并且 SS 引脚（或软件管理的内部信号）有效，则该从器件会发送数据（请参见 *50.4.6：多主通信*）。

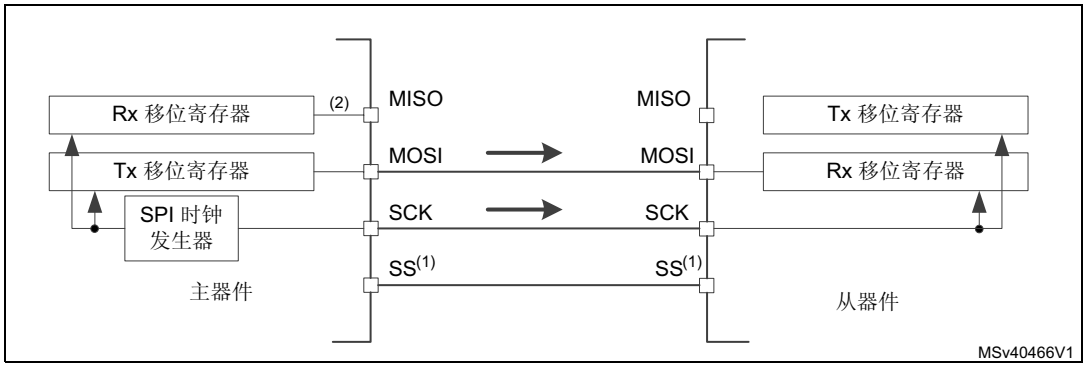
- **只接收模式：COMM[1:0] = 10**

  在主模式下，MOSI 输出被禁止，并且可用作 GPIO。只要 SPI 处于使能状态，且 SPI_CR1 寄存器的 CSTART 位置 1，就会不断生成时钟信号。时钟既可以由软件通过将 SPI_CR1 寄存器的 CSUSP 位置 1 进行明确请求来停止，也可以在 RxFIFO 已满时（SPI_CR1 的 MASRX 位置 1 的情况下）自动停止。

  在从器件配置下，MISO 输出被禁止，该引脚可用作 GPIO。当从器件选择信号有效时，从器件继续从 MOSI 引脚接收数据（请参见 *50.4.6：多主通信*）。基于数据缓冲区的配置产生接收数据事件。

注： *无论在何种主模式和从模式下，通过更改 SPI_CFG2 寄存器中的 IOSWP 位值，可将专用于发送的数据引脚替换为专用于接收的数据引脚，反之亦然（该位只能在 SPI 禁止时进行修改）。任何单工通信均可以通过半双工通信的一种变型来替换，该变型中设置的数据传输方向不变（在 HDDIR 位保持不变的同时，双向模式处于使能状态）。*

图 606. 单工单个主器件/单个从器件应用（主器件为只发送模式/从器件为只接收模式）

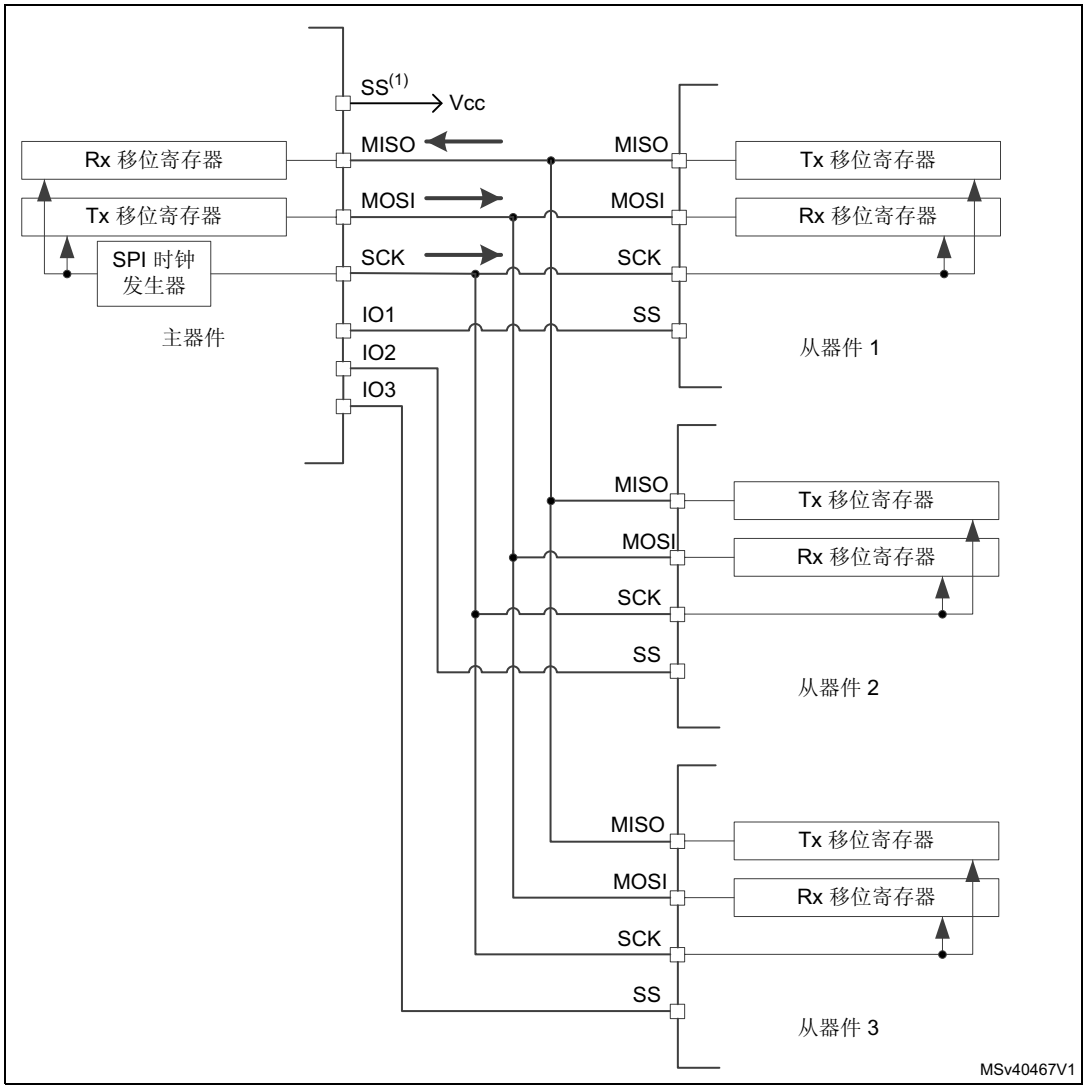


1. 主器件的 SS 引脚配置为输出模式。引脚可保持未连接状态，之后主器件和从器件的 SS 都由软件在内部进行管理。
2. 输入信息在移位寄存器中捕获，在标准的只发送模式下必须忽略此信息（例如，OVF 标志）
3. 在这种配置下，两个 MISO 引脚均可用作 GPIO。

50.4.5 标准多从器件通信

在具有两个或多个独立从器件的配置下，主器件使用星型拓扑和专用的 GPIO 引脚来分别管理每个从器件的片选线（请参见*图 607*）。主器件必须通过拉低与从器件 SS 输入相连的 GPIO 的电平来单独选择一个从器件（在同一时刻，只有一个从器件可以控制共用 MISO 线上的数据）。完成上述操作后，随即会在主器件和所选从器件之间建立通信。这种拓扑除了具有简便性优势外，还支持对各个从器件应用特定 SPI 配置，因为所有通信会话均在一个主器件-从器件对内单独进行。此外，如果无需从从器件中读取任何信息，主器件可以将相同的信息发送到多个从器件。

图 607. 使用星形拓扑的主器件和三个独立从器件

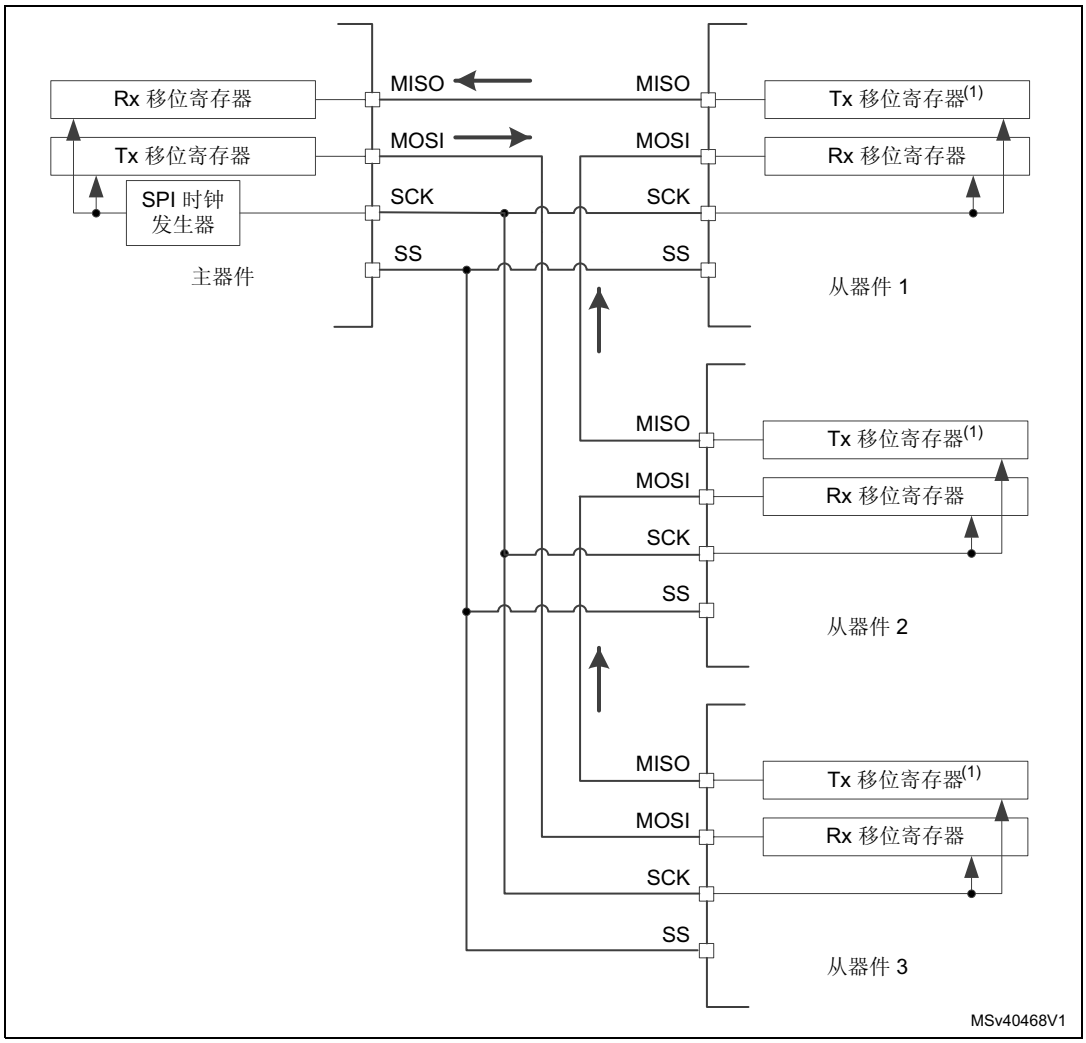


1. 此配置的主器件侧不使用 SS 引脚。该引脚必须在内部管理 (SSM = 1，SSI = 1) 以避免任何 MODF 错误。
2. 由于从器件的 MISO 引脚连在一起，所有从器件 MISO 引脚的 GPIO 配置必须设置为复用功能的开漏模式（请参见*第 450 页的第 11.3.7 节：I/O 复用功能输入/输出*）。

当采用环形拓扑时，主器件可以与所有从器件进行 SPI 通信（请参见*图 608*）。所有从器件类似于串行菊花链上的简单移位寄存器，通过共用的从器件选择信号和时钟进行控制。所有信息在环上同时移位，然后返回到主器件。会话的长度是固定的，主器件与从器件数处理的数据帧数相同。在菊花链中处理第一个数据帧时，主器件只通过第一个从节点输入向菊花链中最后一个从节点发送信息，主器件接收的第一条信息来自最后一个节点输出。相应地，用来结束会话的最后处理的数据发送给第一个从节点，同时其最先传出的数据在会话期间通过菊花链上所有其他从器件正好到达主器件输入。采用该拓扑时，菊花链中所有节点的数据格式配置和时钟设置都必须相同。由于接收和发送移位寄存器在内部是彼此分开的，因此当在接收器和发射器之间通过硬件交互信息时，必须使用从器件 TxFIFO 的下溢功能来完成。在这种情况下，传输下溢功能被配置为重复最后接收到的数据帧的模式 (UDRCFG[1:0] = 01)。在会话开始之前，每个从器件把单个数据写入 TxFIFO，然后发起会话传输（通常用到从器件状态信息）。在这种情况下，处理完该第一个数据帧后会实际发生下溢（数据处理结束时

必须在对从器件设置下溢检测：UDRDET[1:0] = 01）。为了能够立即清除内部下溢条件，并再次通过 TxFIFO 内容重新启动会话，用户必须先禁止再使能会话之间的 SPI，以及通过新的单个数据模式填充 TxFIFO（以克服清除时造成的传播延时，通过 UDRC 位以标准方式清除下溢时会引起这种延时）。

图 608. 一主三从环形（菊花链）拓扑



1. 在该配置下，在从器件上使用了下溢特性，一旦从器件的 TxFIFO 变空，从器件可以将先前接收的数据发送到 Rx 移位寄存器中。

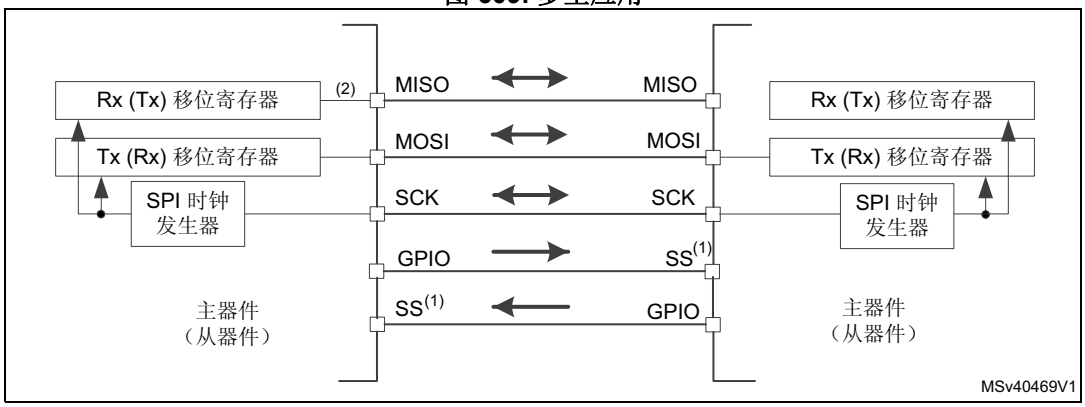
50.4.6 多主通信

如果 SPI 总线没有多主功能，用户可使用内置功能来检测尝试同时控制总线的两个节点间是否存在潜在冲突。对于该检测，SS 引脚配置为硬件输入模式。由于此时只有一个节点可将其输出施加到共用数据线上，因此无法同时连接超过两个此模式的 SPI 节点。

当节点无效时，默认情况下均保持从模式。一旦一个节点要接管对总线的控制，它会将自身切换到主模式，然后通过专用 GPIO 引脚向其它节点的从器件选择输入施加有效电平。会话完成后，有效的从器件选择信号将被释放，控制总线的节点会短暂切换回被动从模式，等待下一个会话开始。

如果两个节点同时发出各自的控制请求，则会出现总线冲突（请参见模式故障 MODF 事件）。随后，用户可应用某个简单的仲裁过程（例如，在两个节点上施加不同的预定义超时来推迟下一个尝试）。

图 609. 多主应用



1. 在两个节点上，SS 引脚都配置为硬件输入模式。当被动节点配置为从器件时，其有效电平将使能 MISO 线输出。

50.4.7 从器件选择 (SS) 引脚管理

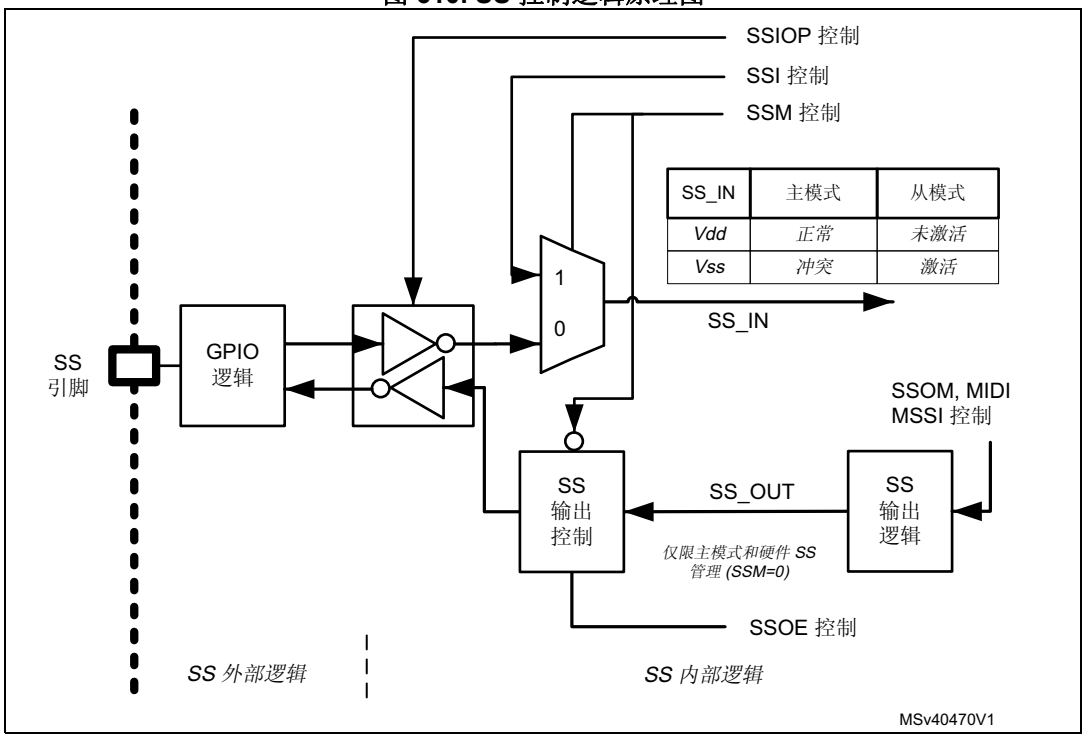
在从模式下，SS 用作标准的“片选”输入，使从器件与主器件进行通信。在主模式下，SS 可用作输出或输入。用作输入时，可防止多主模式总线冲突；用作输出时，可驱动单个从器件的从器件选择信号。SS 信号可在内部进行管理（SS 输入信号的软件管理），或在 SS 输入和输出均与 SS 引脚相关联时在外部进行管理（硬件 SS 管理）。用户可以通过 SSIOP 位设置来配置（SS 引脚上）输入/输出外部信号的有效电平。在内部将低电平视为有效电平时，SSIOP = 1 设置可以针对外界将该逻辑反转。

可以使用 SPI_CFG2 寄存器中的 SSM 位设置硬件或软件从器件选择管理：

- **软件 SS 管理 (SSM = 1)**：在这种配置下，由 SPI_CR1 寄存器中的 SSI 位的值内部驱动从器件选择信息。外部 SS 引脚空闲，可供其他应用使用（用作 GPIO 或其他复用功能）。
- **硬件 SS 管理 (SSM = 0)**：在这种情况下，可行的配置有两种：所用配置取决于 SS 输出配置（SPI_CFG2 寄存器中的 SSOE 位）。
  - **SS 输出使能 (SSOE = 1)**：仅在将 MCU 设置为主器件时才使用该配置。SS 引脚由硬件管理。
    a) 如果 SSOM = 0 且 SP = 000，则在主器件传输开始 (CSTART = 1) 时，SS 信号会被驱动至有效电平，并始终保持有效，直到其 EOT 标志置 1 或传输进入挂起状态。
    b) 当 SP = 001 时，根据 TI 模式定义生成脉冲。
    c) 当 SSOM = 1、SP = 000 且 MIDI>1 时，SS 在各个数据帧之间的脉冲无效，并在 MIDI 值减 1 得到的值所定义的 SPI 时钟周期（1 到 14）内保持无效。
  - **SS 输出禁止（SSM = 0，SSOE = 0）**：
    a) 如果微控制器在总线上用作主器件，则此配置可实现多主模式功能。如果在该模式下将 SS 引脚拉至有效电平，SPI 将进入主模式故障状态，SPI 器件将在从模式 (MASTER = 0) 下自动进行重新配置。
    b) 在从模式下，SS 引脚用作标准的“片选”输入，当 SS 线为有效电平时将选择此从器件。

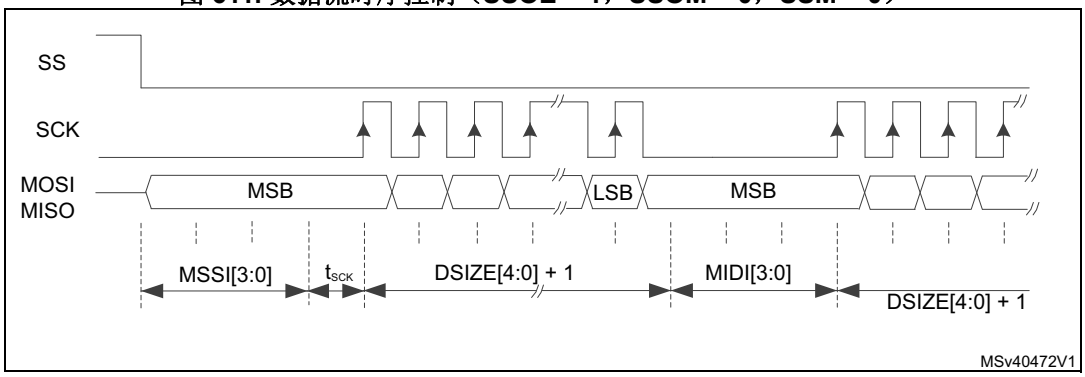
注： *在模式故障条件下自动切换到从模式可避免数据和时钟线上发生潜在冲突。SPE 不会自动复位，因为这会自动清空 RX 和 TXFIFO，且当前数据可能会丢失。在 MODF 事件之后，软件必须正确管理 FIFO 读取/清空，并正确重新编程 SPI 配置，以接管系统中的从器件角色。*

图 610. SS 控制逻辑原理图



应用硬件输出 SS 控制（SSM = 0，SSOE = 1）时，用户可通过配置 MIDI[3:0] 和 MSSI[3:0] 位域来控制数据帧之间的 SS 信号时序，以及插入每次事务开始时的额外延时（以分隔 SS 和时钟启动）。从器件需要减慢数据流速以获得充足的空间来正确处理数据时，这十分有用（请参见*图 611：数据流时序控制（SSOE = 1，SSOM = 0，SSM = 0）*）

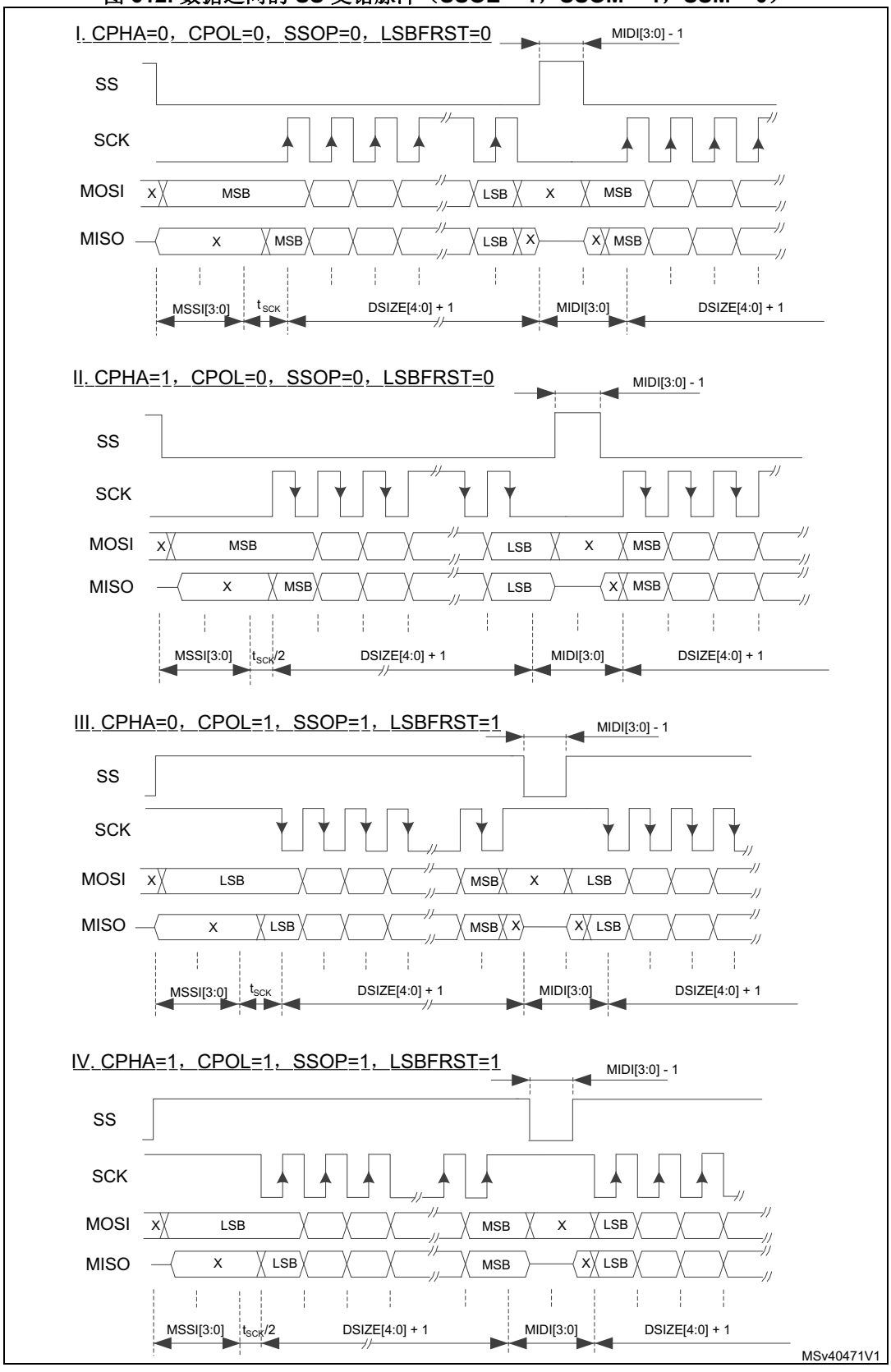
图 611. 数据流时序控制（SSOE = 1，SSOM = 0，SSM = 0）



1. MSSI[3:0] = 0011，MIDI[3:0] = 0011（MIDI[3;0] =0 时，SCK 流连续）。
2. CPHA = 0，CPOL = 0，SSOP = 0，LSBFRST = 0。

此外，位 SSOM =1 设置会调用特定模式，在该模式下，如果有足够的空间可用则会在数据帧之间加入交错脉冲（必须将 MIDI[3:0] 设置为大于一个 SPI 周期的值）。*图 612：数据之间的 SS 交错脉冲（SSOE = 1，SSOM = 1，SSM = 0）*给出了一些配置示例。

图 612. 数据之间的 SS 交错脉冲（SSOE = 1，SSOM = 1，SSM = 0）



1. MSSI[3:0] = 0010，MIDI[3:0] = 0010。
2. MIDI[3:0] > 1 时数据之间的 SS 交错。

50.4.8 通信格式

SPI 通信过程中，将同时执行接收和发送操作。串行时钟 (SCK) 对数据线上的信息的移位和采样进行同步。通信格式取决于时钟相位、时钟极性和数据帧格式。为了能够在彼此间进行通信，主器件和从器件必须遵循相同的通信格式并且必须正确同步。

时钟相位和极性控制

通过 SPI_CFG2 寄存器中的 CPOL 和 CPHA 位，可以用软件选择四种可能的时序关系。CPOL（时钟极性）位控制不传输任何数据时时钟的空闲状态值。此位对主器件和从 器件都有作用。如果复位 CPOL，SCK 引脚在空闲状态处于低电平。如果将 CPOL 置 1，SCK 引脚在空闲状态处于高电平。

如果将 CPHA 位置 1，则会在 SCK 引脚的第二个边沿捕获传输的第一个数据位（如果复位 CPOL 位，则为下降沿；如果将 CPOL 位置 1，则为上升沿）。即，在每次出现该时钟边沿时锁存数据。如果将 CPHA 位复位，则会在 SCK 引脚的第一个边沿捕获传输的第一个数据位（如果将 CPOL 位置 1，则为下降沿；如果将 CPOL 位复位，则为上升沿）。即，在每次出现该时钟边沿时锁存数据。

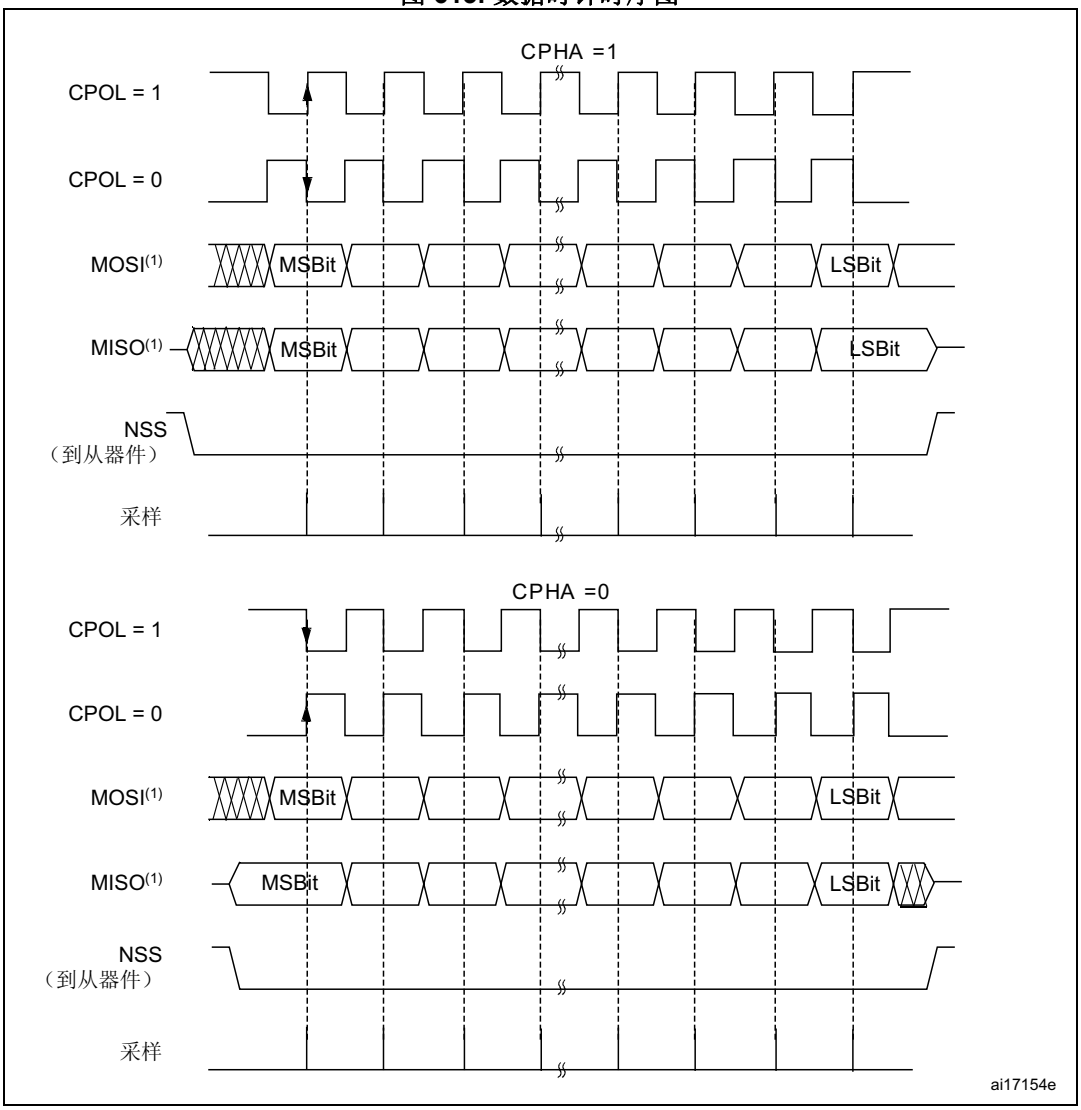
CPOL（时钟极性）和 CPHA（时钟相位）位的组合用于选择数据捕获时钟边沿。

图 613 给出了在 CPHA 和 CPOL 位的四种组合下的 SPI 全双工传输。

注： *在切换 CPOL/CPHA 位之前，必须通过复位 SPE 位来禁止 SPI。*

SCK 的空闲状态必须与 SPI_CFG2 寄存器中选择的极性相对应（如果 CPOL=1，则上拉 SCK；如果 CPOL=0，则下拉 SCK）。

图 613. 数据时钟时序图



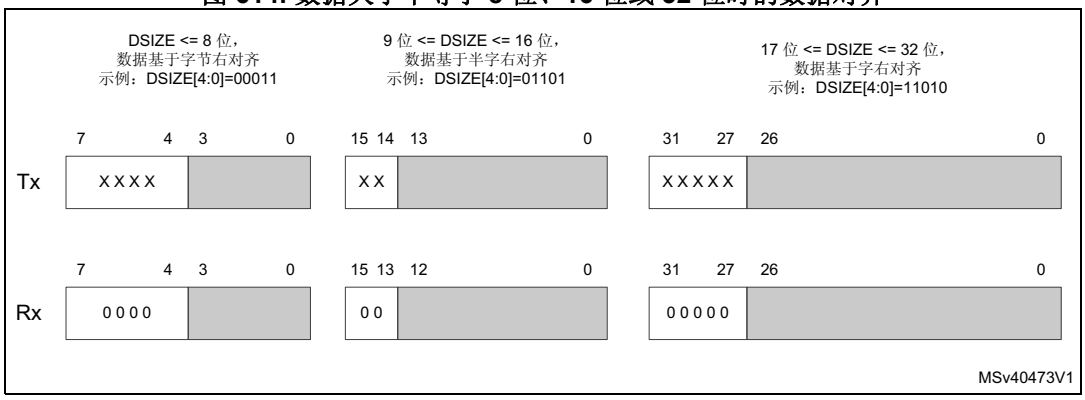
1. 数据位的顺序取决于 LSBFRST 位的设置。

数据帧格式

SPI 移位寄存器可设置为以 MSB 在前或 LSB 在前的方式移出数据，具体取决于 SPI_CFG2 寄存器的 LSBFRST 位的值。数据帧的长度通过 DSIZE[4:0] 位进行选择。数据帧的长度可设置为 4 位到 32 位，且此设置对于发送和接收均适用。访问 SPI_TXDR/SPI_RXDR 寄存器时，数据帧始终按字节（数据不超过一个字节时）、半字或字进行右对齐（请参见*图 614*）。

如果访问大小是单个数据帧所需的最小数据大小的倍数，则会将 2 个或 4 个数据帧置于该寄存器中。在通信过程中，只会为数据帧内的位提供时钟并传输这些位。

图 614. 数据大小不等于 8 位、16 位或 32 位时的数据对齐



注： *数据长度至少为 4 位。如果所选的数据长度不足 4 位，则会将数据帧长度强制为 4 位。*

50.4.9 SPI 配置

主器件和从器件的配置步骤几乎相同。对于具体的模式设置，请遵从相应章节的内容。若要对标准通信进行初始化，请执行以下步骤：

1. 对适当的 GPIO 寄存器执行写操作：将 MOSI、MISO 和 SCK 引脚配置为 GPIO。
2. 对 SPI_CFG1 和 SPI_CFG2 寄存器进行写操作，以便为所有非保留位以及相应的位域设置合适的值，但以下内容除外：
   a) 仅在主模式下才需要 SSOM、SSOE、MBR[2:0]、MIDI[3:0] 和 MSSI[3:0]，MSSI 位在 SSOE 置 1 时有效，TI 模式下的从器件需要 MBR 设置。
   b) 仅在从模式下才需要 UDRDET[1:0] 和 UDRCFG[1:0]。
   c) CRCEN 置 1 时需要配置 CRCSIZE[4:0]。
   d) TI 模式下不需要 CPOL、CPHA、LSBFRST、SSOM、SSOE、SSIOP 和 SSM。
   e) SPI_CFG2 寄存器的 AFCNTR 位置 1 后，无论外设使能如何，所有 SPI 输出都会开始传播到相关的 GPIO 引脚，因此，之后对 SPI_CFG1 和 SPI_CFG2 寄存器的配置进行的任何更改都会影响这些引脚的信号电平。
   f) SPI_I2SCGFR 寄存器的 I2SMOD 位必须保持清零，以防偶然 I2S 配置的意外影响。
3. 对 SPI_CR2 寄存器进行写操作以选择传输的长度，如果值未知，则必须将 TSIZE 编程为零。
4. 对 SPI_CRCPOLY 进行写操作，并对 SPI2S_CR1 寄存器的 TCRCINI、RCRCINI 和 CRC33_17 位进行写操作，以根据需要配置 CRC 多项式和 CRC 计算。
5. 如果使用 DMA 数据流，请在 DMA 寄存器中配置 SPI Tx 和 Rx 专用的 DMA 数据流（请参见“使用 DMA 进行通信”一章）。
6. 如果需要配置保护，请对 SPI_CFG1 寄存器的 IOLOCK 位进行编程（安全起见）。

50.4.10 使能 SPI 的步骤

建议在主器件发送时钟之前，配置和使能 SPI 从器件，如果在总线上正在进行通信时执行配置和使能过程，则无任何影响。从器件发送器的数据寄存器应包含待发送的数据，这样主器件才能开始提供时钟。在由 SS 选择 SPI 从器件之前，SCK 信号必须稳定到所选极性所对应的空闲状态电平，否则传输可能会不同步。

SPI 从器件在硬件 SS 管理模式下使能时，即使在发现 SS 处于有效电平的情况下，也会忽略所有通信，直到从器件检测到 SS 信号的开始（从无效电平变为有效电平的传输启动信号），以将从器件与主器件同步。因此，外部 SS 引脚固定时不能使用硬件管理模式。SS 软件管理不提供此类保护功能。总线上无通信时，SSI 位应发生更改，在传输之间 SCK 信号处于空闲状态电平（仅限这种情况）。

当 SPI 处于使能状态，CSTART 位置 1 且 TxFIFO 非空时，或者对 TxFIFO 执行下一次写操作时，全双工模式（或任何只发送模式）下的主器件开始通信。

在任何主器件仅接收模式下，在 SPI 使能并且 CSTART 位置 1 后，主器件开始通信，时钟开始运行。

有关处理 DMA 的信息，请参见*第 50.4.14 节：使用 DMA（直接存储器寻址）进行通信*。

50.4.11 SPI 数据发送和接收过程

RXFIFO 和 TXFIFO

所有 SPI 数据传输均通过按字节组织的嵌入式 FIFO（N x 8 位）。FIFO 的大小 (N) 与产品和外设实例相关。这使得 SPI 能够以连续流工作，并能防止在数据帧长度较短或中断/DMA 延时过长时发生上溢。每个方向都有其自身的 FIFO，分别被称为 TxFIFO 和 RxFIFO。

FIFO 的处理取决于数据交换模式（双工和单工）、数据帧格式（帧中的位数）、FIFO 数据寄存器中的访问大小（8 位、16 位或 32 位）以及数据在数据包中的组织方式。

对 SPI2S_RXDR 寄存器执行读访问时，会返回尚未读取的 RxFIFO 中存储的最早的值。对 SPI2S_TXDR 执行写访问时，会在发送队列结束时将写入的数据存储到 TxFIFO 中。

SPI2S_RXDR 寄存器的读访问必须通过 RXP 事件进行管理。接收处于激活状态时，如果接收 FIFO 中至少有一个完整数据包（通过 SPI_CFG1 寄存器的 FTHVL[3:0] 位定义为接收器阈值）可用，则该标志由硬件置 1。通过软件或 DMA 读取 SPI2S_RXDR 时，如果 RXFIFO 中的数据不足，则会清零 RXP。

RXP 会在 RXPIE 位置 1 时触发中断，或在 RXDMAEN 置 1 时触发 DMA 请求。

RXP 标志置 1 时，应用软件会执行适当数量的 SPI 数据寄存器读取操作，以下载一个数据包的内容。下载完整数据包后，应用软件会检查 RXP 值，以查看接收 FIFO 中是否存在待决的其他数据包，如果存在，则逐个数据包对其进行下载，直到 RXP 读为 0。RxFIFO 最多可存储 N 个数据帧（帧大小 =< 8 位时）、N/2 个数据帧（8 位 < 帧大小 =< 16 位时）、N/3 个数据帧（16 位 < 帧大小 =< 24 位时）或 N/4 个数据帧（数据帧大小 > 24 位时），其中 N 表示 FIFO 的大小（以字节计）。

接收结束时，可能会发生 RxFIFO 中仍然有某些数据可用，而不会达到 FTHVL 级别的情况，因此 RXP 不会置 1。在这种情况下，FIFO 中的剩余 RX 数据帧数将由 SPI_SR 寄存器中的 RXWNE 和 RXPLVL 字段指示。传输大小和数据包大小未对齐时，如果在传输中接收到的最后数据数目不能完全达到配置的数据包大小，则会发生上述情况。不过，应用软件仍可从 RxFIFO 中进行用于先前完整数据包的标准次数的读取操作，而不会产生不利影响：只会从 RxFIFO 中弹出一致数据（完整的数据帧），而冗余读取次数（或任何不完整的数据）将为读数 0。通过这样的方式，应用软件可以使用相同的方式处理传输中的所有数据，并且

可减轻负担来预知传输中的最后一个数据的接收，可减轻对要从 RxFIFO 中弹出的数据的适当读取次数进行计算的负担。

同样地，待发送数据帧的写访问通过 TXP 事件进行管理。发送处于激活状态时，如果应用软件有足够的空间来将至少一个完整的数据包（通过 SPI_CFG1 寄存器的 FTHVL[3:0] 位进行定义）推入到发送 FIFO 中，则该标志由硬件置 1。TxFIFO 由软件和/或 DMA 进行填充后，TXP 会被清零，当前可用于下一完整数据包的空间将丢失。逐帧存储新数据包时，如果从 TxFIFO 中释放数据，则这会导致 TXP 信号的振荡。TXTF 置 1 或禁止 SPI 时，如果没有足够的空间来存储至少一个数据帧（未遵循 TXP 事件），则对 TxFIFO 进行的任何写操作都将被忽略。

TXP 会在 TXPIE 位置 1 时触发中断，或在 TXDMAEN 置 1 时触发 DMA 请求。TXPIE 屏蔽在 TXTF 标志置 1 时由硬件清零。

TXP 标志置 1 时，应用软件会执行适当数量的 SPI 数据寄存器写入操作，以上传一整个数据包的内容。上传完新的完整数据包后，应用软件会检查 TXP 值，以查看是否可将其他数据包推入 TxFIFO 中，如果可以，则逐个数据包对其进行上传，直到在数据包加载结束时 TXP 读为 0。

传输大小和数据包大小未对齐时，传输中的最后一个数据量可短于配置的数据包大小。不过，应用软件仍可进行用于先前数据包的标准次数的数据寄存器写入操作，而不会产生不利影响：只会将一致数据推入 TxFIFO 中，而冗余写入次数将被忽略。因此，应用软件可以使用相同的方式处理传输中的所有数据，并且可减轻负担来预知传输中的最后一个数据的发送，可减轻对要推入 TxFIFO 中的最后一个数据的适当写入次数进行计算的负担。仅在最后一个数据情况下，如果 TxFIFO 有足够的空间来存储剩余数据进而完成当前传输，TXP 事件将由 SPI 置为有效。

TXP 和 RXP 事件均可通过中断进行轮询或处理。在全双工模式下，DXP 位可作为通用 TXP 和 RXP 事件进行监视。

将 DXP 标志置 1 时，应用软件会向 SPI 数据寄存器执行适当数量的写入操作，来上传一个用于发送的完整数据包的内容，然后从 SPI 数据寄存器进行相同数量的读取操作来下载一个数据包的内容。上传并下载一个数据包后，应用软件会检查 DXP 值，以查看是否可按顺序推入和弹出其他数据包，如果可以，则逐个数据包对其进行上传/下载，直到 DXP 读为 0。

DXP 会在 DXPIE 位置 1 时触发中断，或在 TXDMAEN 和 RXDMAEN 置 1 时触发 DMA 请求。DXPIE 屏蔽在 TXTF 标志置 1 时由硬件清零。

DXP 对于全双工通信十分有用，可优化数据上传/下载性能，并减少支持 SPI 传输所需的中断数，从而最大限度地减少对 CPU 带宽和系统电源的需求，特别是在 SPI 工作在停止模式时，尤为如此。

另一种管理数据交换的方式是使用 DMA（请参见*第 15 节：直接存储器访问控制器（DMA1、DMA2）*和*第 16 节：基本直接存储器访问控制器 (BDMA)*）。

如果在 RXFIFO 已满时收到下一个数据，将发生上溢事件（请参见*第 50.5.2 节：SPI 错误标志*中的 OVR 标志说明）。上溢事件可通过中断来轮询或处理。

这可能会发生在从模式下或主模式下（全双工或仅接收模式，MASRX = 0）。在主器件仅接收模式下，MASRX = 1 时，如果 RxFIFO 已满，则生成的时钟将自动停止，因此可防止发生上溢。

禁止 SPI (SPE = 0) 时，RxFIFO 和 TxFIFO 内容保持清空状态。

序列处理

可通过一个序列传送一些数据帧从而完成一条消息。用户可通过存储在 TSIZE 和 TSER 字段中的值，处理一条消息中的多个数据。原则上，消息的传输在通过将 CSTART 位置 1 来使能 SPI 时开始，并在处理完所需数据量后结束。如适用，事务的结束可控制 CRC 和硬件 SS 管理。如果在 CSTART 置 1 时 TSIZE 保持为零，则会初始化无限事务（不应用数据大小控制）。凭借设置可清零 CSTART 位的 CSUSP 位，可随时挂起事务。

在主模式下，用户可增加当前会话中的数据量。处理完 TSIZE 中的数据量时，如果 TSER 包含非零值，则会将 TSER 的内容复制到 TSIZE 中，并自动清零 TSER 值。传输随即会增加重新加载到 TSIZE 中的值所对应的数据量。在这种情况下，不会出现 EOT 事件，因为事务继续进行。完成重新加载操作后，如果 TSERFIE 置 1，则 TSERF 标志置 1 并会发生中断。用户可在下次重新加载之前将下一个非零值写入 TSER 中，因此在重复此过程时可处理无数个数据。

应用任一数据扩展时，数据扩展都始终以对齐的数据包开始。因此，建议将要扩展的数据量始终与数据包大小保持对齐，否则，在应用扩展之前的最后一个数据包必须作为不完整数据包进行处理（请参见数据封装一章）。如果数据总数未对齐，则用户应在最后一个扩展周期内将其余未对齐的数据量写到 TSER 中，然后在 EOT 事件处理程序中以标准方式处理数据的最后一个不完整数据包。

例如，如果用户想要在基于 8 位数据大小的配置应用数据量扩展时传输 23 个字节，则使用被设为 4 个数据的数据包和 32 位至 FIFO 的访问，这与下一序列是否正确无关

- TSIZE = 16 TSER = 7；
- TSIZE = 12 TSER = 8；最后一个扩展 TSER = 3；

因为在最后一次（第 6 次）访问 FIFO 时，会忽略最后一个未对齐的 MSB 字节。

如果对诸如以下要扩展的数据应用未对齐的序列：

- TSIZE = 15 TSER = 8 或
- TSIZE = 8 TSER = 7；最后一个扩展 TSER = 8；

则 MSB 字节在第 4 次访问 FIFO 时被忽略，而其他访问会始终处理 FIFO 中的 4 个数据。

使能发送后，序列即开始，只要主器件的 TxFIFO 中存在数据便一直继续。时钟信号由主器件永久性提供，直至 TxFIFO 变为空，之后时钟信号停止，等待其他数据。

在仅接收模式、半双工（COMM[1:0] = 11，HDDIR = 0）或单工 (COMM[1:0] = 10) 模式下，主器件在使能 SPI 时启动序列并通过将 CSTART 位置 1 来启动传输。时钟信号由主器件提供，且直至主器件禁止/挂起 SPI 或仅接收模式才会停止。在此之前，主器件会永久性接收数据帧。可通过软件控制以及通过向 SPI_CR1 寄存器的 CSUSP 位写入 1 将接收挂起，或者可在 MASRX = 1 且 RxFIFO 已满时自动将接收挂起。已完成在 SPI_CR2 寄存器的 TSIZE 和 TSER 字段中编程的帧数时，接收也会自动停止。

为禁止主器件仅接收模式，必须先挂起 SPI。SPI 被挂起时，会在更改配置前完成当前帧。

注意： 如果在主模式下向 SPE 写入 0，而接收正在进行且不会挂起，则时钟将停止，且不会完成当前帧，并会清空 RxFIFO。

当主器件能够以连续模式（SCK 信号连续）提供所有交互时，任何时候都必须根据从器件功能来处理数据流及其内容。必要时，主器件必须降低通信速度，提供较慢的时钟或带有足够延时的单独帧或数据会话（通过将 MIDI[3:0] 位置 1），或者，提供初始延时（通过将 MSSI[1:0] 置 1），这会延迟传输的开始，从而为从器件提供充分的空间来准备数据。请注意，来自从器件的数据始终由主器件传输和处理，即使从器件无法及时正确地准备数据也是如此。从器件最好使用 DMA，尤其是数据帧较短，按字节访问 FIFO 且 SPI 总线速率较高时。

为了对来自从器件发送器节点的 SPI 通信流添加软件控制功能，可使用写入到 SPI_UDRDR（SPI 下溢数据寄存器）中的特定值。在从器件侧，当 TxFIFO 变为空时，该值将作为下一个数据自动送出，并且在主器件接收器侧可通过软件进行解析（丢弃或解析为类似 XOFF 的命令，以便通过软件挂起主器件接收器）。

在多从系统中，每个序列必须通过 SS 脉冲进行使能，从而只选择其中一个从器件进行通信。在单个从器件系统中，无需通过 SS 来控制从器件，但此时提供此脉冲通常会更好，以在每个数据序列开始时同步从器件。SS 可通过软件和硬件进行管理（请参见*第 50.4.6 节：多主通信*）。

50.4.12 禁止 SPI 的步骤

当禁止 SPI 时，必须按照本段中介绍的禁止步骤进行操作。

在主模式下，在外设时钟停止，系统进入低功耗模式之前执行该操作十分重要。否则，在这种情况下，会损坏正在进行的传输。

在从模式下，SPI 通信可在 **spi_pclk** 和 **spi_ker_ck** 时钟停止时继续不间断进行，直至达到结束通信或数据服务请求条件。通常，可通过将系统置于停止模式来停止 **spi_pclk**。更多相关信息，请参见 RCC 部分。

当处于全双工或仅发送模式下的主器件停止提供待发送的数据时，可结束任何事务。在这种情况下，时钟在最后一个数据传输后停止。可轮询 TXC 标志（或通过 EOTIE = 1 使能中断），以等待发送最后一个数据帧。

为停止外设，当主器件处于仅接收模式下时，必须首先通过将 CSUSP 置 1 挂起 SPI 通信。

当 SPI 被挂起时，已接收但未读取的数据始终存储在 RxFIFO 中。

禁止 SPI 时，RxFIFO 会被清空。为了防止丢失未读取的数据，用户必须通过读取所有剩余的数据（如 SPI_SR 寄存器的 RXP、RXWNE 和 RXPLVL 字段所示），来确保在禁止 SPI 时 RxFIFO 为空。

标准禁止步骤通过轮询 EOT 和/或 TXC 状态来检查发送会话是否（完全）结束。还可以在必须识别正在处理的传输是否结束的特定情况下完成这种检查，例如：

- 当 SS 信号由软件管理且主器件必须为从器件提供 SS 脉冲结束时，或者
- 最后一个数据帧或 CRC 帧传输仍在外设总线中处理时，来自 DMA 或 FIFO 的传输数据流完成。

主模式下的正确禁止步骤如下（使用仅接收模式时除外）：

1. 等待 TXC = 1 和/或 EOT = 1（不再有待发送的数据和发送最后一个数据帧）。使用 CRC 时，会在块中的最后一个数据被处理后自动发送。在这种情况下，TXC/EOT 会在 CRC 帧完成时置 1。当发送挂起时，软件必须等到 CSTART 位清零。
2. 读取所有 RxFIFO 数据（直到 RXWNE = 0 和 RXPLVL = 00）。
3. 禁止 SPI (SPE = 0)。

主器件仅接收模式的正确禁止步骤如下：

1. 等待 EOT 或通过挂起 SPI (CSUSP = 1) 中断接收流。
2. 如果接收流已挂起，等待至 SUSP = 1（最后一个数据帧已处理完）。
3. 读取所有 RxFIFO 数据（直到 RXWNE = 0 和 RXPLVL = 00）。
4. 禁止 SPI (SPE = 0)。

在从模式下，禁止 SPI 时，任何正在进行的数据都将丢失。

50.4.13 数据打包

从用户的角度来看，数据封装方式有两种，这两种方式可彼此覆盖：

- 向 TxFIFO 中写入数据或从 RxFIFO 中读取数据时的访问类型。
  如果数据大小明显小于对 SPI2S_TXDR 或 SPI2S_RXDR 寄存器执行的访问大小，则可通过单次访问有效地推入或提取多个数据。
- 在单一软件服务期间要处理的数据量。
  可方便地将数据分组到数据包中，并将所有数据包内容累积到 FIFO 服务，而不是分别逐帧处理数据。用户可通过 FIFO 阈值设置定义数据包。之后，所有 FIFO 占用事件均与该阈值相关，而所需服务通过具有中断和/或唤醒功能的适当标志来反映。

若数据帧大小不足一个字节（小于或等于 8 位），则在 SPI2S_RXDR/SPI2S_TXDR 寄存器上执行任何 16 位或 32 位读写访问时都将自动使用数据封装。在这种情况下将并行处理多数据帧类型。最初，SPI 以所访问字的 LSB 中存储的模式工作，然后以 MSB 中存储的另一种模式来工作。*图 615* 给出了数据封装模式序列处理的示例。此时 DSIZE[3:0] 被配置为 4 位，在对发送器的 SPI2S_TXDR 寄存器进行单次 16 位或 32 位访问后，会向 TxFIFO 中写入两个或四个数据帧。

如果数据帧大小介于 9 位和 16 位之间，则在完成一次 32 位访问后，会自动使用数据封装。将首先使用最低有效半字。（与 LSBFRST 值无关）

如果将 RxFIFO 阈值设为 1 个帧（且以帧为单位读取数据，即解封形式），则该序列可在接收器中生成两个或四个 RXP 事件，或者，如果 SPI_CFG1 寄存器中的 FTHLV[3:0] 字段被配置为以封装模式读取的多个帧（16 位或 32 位读访问），则可生成一个 RXP 事件。

数据根据*图 614：数据大小不等于 8 位、16 位或 32 位时的数据对齐*进行对齐。有效位仅在总线上执行。未使用的位在发送器侧无关紧要，而在接收器侧则用零填充。

将短数据帧（<8 位或 < 16 位）与大型数据访问模式（16 位或 32 位）搭配使用时，FTHVL 值必须被配置为帧/数据访问数的倍数（即，如果将 32 位访问用于最高 8 位的帧则为 4 的倍数，或者，如果将 16 位访问用于最高为 8 位的帧或者将 32 位访问用于最高 16 位的帧，则为 2 的倍数）。

RxFIFO 阈值设置必须始终高于后续读访问大小，否则将读取额外的伪数据。

不允许小于配置的数据大小的 FIFO 数据访问。始终须确保至少访问一个完整的数据帧。

如果在 FIFO 中存在不完整的数据包（即，少于 4x8 位帧或一个单独的 16 位帧可用），则会出现特定问题。

解决该问题的方法有两种：

A. 不使用 TSIZE 字段

在发送器侧，只需将任意奇序列的最后一个数据帧以 8 位/16 位访问的方式写入 SPI2S_TXDR 即可。

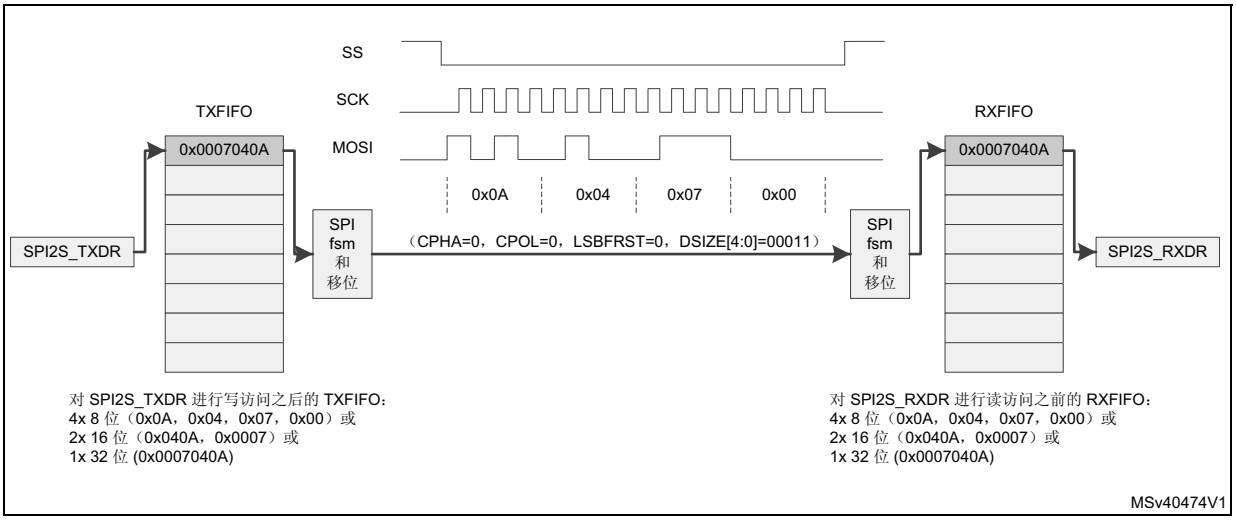
在接收器侧，可通过任意访问读取剩余数据，任何额外的数据读取都会用零填充。可通过轮询 RXWNE 和 RXPLVL 来检测 RxFIFO 中何时存在可用的 RX 数据。（可在系统级使用定时器超时来检测轮询）

B. 使用 TSIZE 字段

在发送器侧，事务在遇到 EOT 事件时由主器件停止。

在接收过程中，RXP 标志不会在 EOT 置 1 时置 1。要接收的数据数 (TSIZE) 不是数据包大小的倍数时，剩余数据的数目通过 SPI_SR 寄存器的 RXWNE 和 RXPLVL 字段指示。可通过任意访问读取剩余数据，任何额外的读取都会用零填充。

图 615. 将数据封装到 FIFO 中以进行发送和接收



1. DSIZE[3:0] 配置为 4 位，数据右对齐，有效位仅在总线上执行，其顺序取决于 LSBFRST，总线上 LSB 字节的内容在前。

50.4.14 使用 DMA（直接存储器寻址）进行通信

为了以最大速度工作并且方便避免上溢所需的数据寄存器读/写过程，SPI 提供了 DMA 功能，该功能采用了简单的请求/应答协议。

将 SPI_CFG1 寄存器的使能位 TXDMAEN 或 RXDMAEN 置 1 时，将请求 DMA 访问。必须向发送缓冲区和接收缓冲区发出单独的请求。

- 在发送过程中，每次 TXP 置 1 都会触发一系列 DMA 请求。然后，DMA 将对 SPI2S_TXDR 寄存器执行一系列写操作。
- 在接收过程中，每次 RXP 置 1 都会触发一系列 DMA 请求。然后，DMA 将对 SPI2S_RXDR 寄存器执行一系列读操作。如果 EOT 在事务结束时置 1，且最后一个数据包不完整，则会根据 RXWNE 和 RXPLVL[1:0] 设置自动激活 DMA 请求来读取剩余数据。

当 SPI 仅用于接收数据时，可以只使能 SPI Rx DMA 通道。

如果将 SPI 编程为仅接收模式，则 UDR 将永不置 1。

如果将 SPI 编程为发送模式，如果发送数据未准备好，则 TXP 和 UDR 最终可在从器件侧置 1。在这种情况下，将根据 UDR 管理选择在 TX 线上发送一些数据。

当 SPI 仅用于发送数据时，可以只使能 SPI Tx DMA 通道。

如果将 SPI 编程为仅发送模式，则 RXP 和 OVR 将永不置 1。

如果将 SPI 编程为全双工模式，如果接收数据未读取，则 RXP 和 OVR 最终会置 1。

在发送模式下，DMA 或用户写入所有要发送的数据（SPI2C_SR 寄存器中的 TXTF 标志置 1）后，可以对 TXC 标志进行监视，以确保 SPI 通信已完成。在关闭 SPI 之前或在主模式下关掉 **spi_pclk** 之前必须执行此步骤，以避免损坏最后一次发送。软件必须首先等待至 EOT = 1 和/或 TXC = 1。

通过 DMA 开始通信时，为防止 DMA 通道管理引发错误事件，必须按顺序执行以下步骤：

1. 如果使用 DMA Rx，通过 SPI_CFG1 寄存器中的 RXDMAEN 位来使能 DMA 接收缓冲区。
2. 如果使用 DMA，则通过 DMA 寄存器来使能 Tx 和 Rx 的 DMA 请求。
3. 如果使用 DMA Tx，通过 SPI_CFG1 寄存器中的 TXDMAEN 位来使能 DMA 发送缓冲区。
4. 通过将 SPE 位置 1 使能 SPI。

要关闭通信，必须按顺序执行以下步骤：

1. 如果使用 DMA，则通过 DMA 寄存器来禁止 Tx 和 Rx 的 DMA 请求。
2. 通过后续 SPI 禁止步骤来禁止 SPI。
3. 如果使用 DMA Tx 和/或 DMA Rx，通过将 SPI_CFG1 寄存器中的 TXDMAEN 和 RXDMAEN 位清零来禁止 DMA 发送缓冲区和接收缓冲区。

使用 DMA 时的数据封装

如果由 DMA 管理传输（SPI_CFG1 寄存器中的 TXDMAEN 和 RXDMAEN 位置 1），则将根据为 SPI TX 和 SPI RX 的 DMA 通道配置的 PSIZE 值自动使能/禁止封装模式。

如果 DMA 通道的 PSIZE 值等于 16 位或 32 位，且 SPI 数据大小小于或等于 8 位，则将使能封装模式。类似地，如果 DMA 通道的 PSIZE 值等于 32 位，且 SPI 数据大小小于或等于 16 位，则将使能封装模式。然后，DMA 将自动管理对 SPI2S_TXDR 寄存器的写操作。

无论使用何种数据封装模式，也无论要传输的数据量是否为 DMA 数据大小（16 位或 32 位）的倍数，如果帧大小较小，则 DMA 会根据 TSIZE 字段设置自动完成传输。

此外，最后几个数据帧还可由软件以单一/解封模式写入。

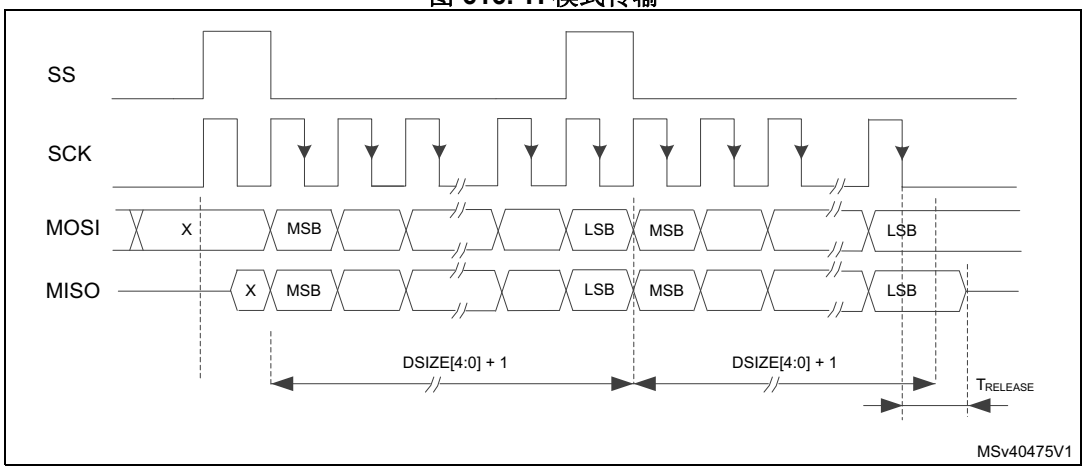
要配置 DMA，禁止小于配置的数据大小的 DMA 数据访问。始终须确保至少访问一个完整的数据帧。

50.5 SPI 特定模式和控制

50.5.1 TI 模式

通过对 SPI_CFG2 寄存器的 SP[2:0] 位域进行特定设置，可将 SPI 配置为符合 TI 协议。SCK 和 SS 信号极性、相位和流以及位顺序均固定，因此在 SPI 处于 TI 模式配置下时，无需 CPOL、CPHA、LSBFRST、SSOM、SSOE、SSIOP 和 SSM 的设置。SS 信号通过 LSB 数据位的脉冲同步协议，如*图 616：TI 模式传输*所示。

图 616. TI 模式传输



在从模式下，时钟发生器用于定义当前事务完成时，MISO 引脚的从器件输出变为高阻态的时间。主波特率设置适用，并且可以使用任意波特率，可以非常灵活地确定此时刻。MISO 信号变为高阻态的延时 ($T_{RELEASE}$) 还取决于内部再同步。具体公式如下：

$$\frac{Tbaud}{2} + 2 \times Tcom < Trelease < \frac{Tbaud}{2} + 4 \times Tcom$$

如果从器件在进行数据事务时检测到错位的 SS 脉冲，TIFRE 标志将置 1。

50.5.2 SPI 错误标志

如果以下其中一个错误标志置 1 且已通过将相应的中断使能位置 1 使能了中断，则将生成 SPI 中断。

上溢标志 (OVR)

当主器件或从器件接收了数据但 RxFIFO 没有足够的空间来存储接收的数据时，将出现上溢的情况。如果软件或 DMA 没有足够的时间来读取之前接收到的数据（存储在 RxFIFO 中），则可能会发生这种情况。

当出现上溢的情况时，OVR 标志将置 1，新接收的值不会覆盖 RxFIFO 中之前的值。新接收的值将被丢弃，之后发送的所有数据都将丢失。OVRIE 位置 1 时 OVR 标志会触发中断。通过向 SPI_IFCR 中的 OVRC 位写入 1 来清零 OVR 位。为了防止之后出现上溢事件，应在通过软件读取清空后 RxFIFO 进行清零操作。在主模式下，用户可通过自动通信挂起（MASRX 位）来防止出现 RxFIFO 上溢。

下溢标志 (UDR)

在从器件发送模式下，如果在 UDRDET 位指定的时刻，从器件 TxFIFO 中没有数据可用于发送，则会由硬件在内部捕获下溢情况。UDR 标志设置随后通过硬件传播到状态寄存器（请参见下面的注）。UDRIE 位置 1 时 UDR 会触发中断。

捕获到下溢情况后，下一个为发送提供的数据将取决于 UDRCFG 位。从器件可提供最后存储到其 TxFIFO 中的数据、之前从主器件接收到的数据，或者由用户存储到 UDRDR 寄存器中的常数模式。第二种配置可用于环形拓扑结构（请参见*图 608*）。软件将 UDR 标志清零后会重新使能标准发送，此清零操作会由硬件传播到 SPI 逻辑。用户应向 TxFIFO 中写入一些数据，然后再清零 UDR 标志，以防捕获到下一个下溢情况。

从器件处理的数据不可预测，尤其是在 TxFIFO 为空，且尚未捕获或刚清除完下溢条件时，开始或继续事务的情况下，更为如此。通常，UDRDET[1:0] = 00 或刚刚使能 SPI 时，或者在刚刚开始已规定大小的事务时，会出现这种情况。在这种情况下，前几个位可能会损坏，此外，在数据事务开始之前过于紧密地向空 TxFIFO 中写入第一个数据（到 TxFIFO 的数据传播会占用几个 APB 时钟周期）时，也会出现这种情况。如果用户不能确保适时将数据写入空 TxFIFO 中，应避免 UDRDET[1:0] = 00 设置。

要正确处理下溢控制功能，用户尤其应避免以下严重干扰：

- 主器件开始提供时钟时对空 TxFIFO 进行的任何填充操作（特别是当 UDRDET[1:0] = 00 时）
- 当 TxFIFO 为空时对 UDR 标志进行的任何清零操作
- 同时设置 UDRDET[1:0] = 00 和 UDRCFG[1:0] = 10
- 当帧与帧之间 SS 信号不翻转的情况下，应该在每帧检测数据下溢，此时设置 UDRDET[1:0] = 10
- 在 SS 由软件管理时设置 UDRDET[1:0] = 10

注： *UDR 标志更改的硬件传播需要在总线上进行额外通信。当更改事件发生后，硬件传播通常需要占用 3 个 SPI 时钟周期（由硬件捕获下溢或由软件清零 UDR 标志）。*

模式故障 (MODF)

当主器件的内部 SS 信号（SS 硬件模式下为 SS 引脚，SS 软件模式下为 SSI 位）被拉低时，将发生模式故障。这将以以下方式自动影响 SPI 接口：

- 如果 MODFIE 位置 1，MODF 位将置 1，并生成 SPI 中断。
- SPE 位清零。这将关闭器件的所有输出，并禁止 SPI 接口。
- MASTER 位清零，从而强制器件进入从模式。

向 SPI_IFCR 的 MODFC 位写入 1 可以将 MODF 清零。

为避免在包含多个 MCU 的系统中发生多从冲突，在重新使能 SPI 之前，必须将 SPE 位置 1，以将 SS 引脚拉至无效电平。

为安全起见，硬件不允许在 MODF 位置 1 时将 SPE 位置 1。在从器件中，MODF 位不可置 1，但由前一次多主模式冲突引起时除外。

在多主系统中，要主器件接管总线，正确的软件程序应如下所示：

- 在 SSOE = 0 时切换到主模式
  （当另一个主器件占用总线时，可能会出现潜在冲突。出现这种情形时，MODF 置 1，以防止任何以下节点切换到主模式）
- 将专用于另一个主器件 SS 控制的 GPIO 引脚置于有效电平
- 执行数据事务
- 将专用于另一个主器件 SS 控制的 GPIO 引脚置于非有效电平
- 切换回从模式

CRC 错误 (CRCE)

当 SPI_CFG1 寄存器中的 CRCEN 位置 1 时，此标志用于验证接收数据的有效性。在接收了最后一个数据后（由 TSIZE 定义），如果移位寄存器中接收的值与接收器 SPI_RXCRC 的值不匹配，则 SPI_SR 寄存器中的 CRCE 标志将置 1。CRCIE 位置 1 时 CRCE 标志会触发中断。向 SPI_IFCR 的 CRCEC 位写入 1 可以将 CRCE 位清零。

TI 模式帧格式错误 (TIFRE)

如果 SPI 在从模式下工作，并配置为符合 TI 模式协议，则在通信进行期间出现 SS 脉冲时，将检测到 TI 模式帧格式错误。出现此错误时，SPI_SR 寄存器中的 TIFRE 标志将置 1。发生错误时不会禁止 SPI，但会忽略 SS 脉冲，并且 SPI 会等待下一个 SS 脉冲，然后再开始新的传输。由于错误检测可能导致丢失几个数据字节，因此数据可能会损坏。

向 SPI_IFCR 的 TIFREC 位写入 1 来可以将 TIFRE 标志清零。如果 TIFREIE 位置 1，则会在检测到 SS 错误时生成中断。由于无法再保证数据一致性，因此应由软件重新启动主从器件之间的通信。

50.5.3 CRC 计算

为检查发送数据和接收数据的可靠性，使用两个独立的 33 位或 17 位 CRC 计算器。对于最大数据大小为 32 位的情况，SPI 可提供 5 到 33 位的 CRC 多项式长度；对于最大数据大小被限制为 16 位的外设实例，则可提供 5 到 17 位的多项式长度。多项式的长度由存储在 CRCPOLY 寄存器中的值的最高有效位定义。必须将其设置为大于 DSIZE 字段中定义的数据帧长度的值。应用最大数据大小时，必须额外将 CRC33_17 位置 1，以定义多项式字符串的最高有效位，同时保持其大小始终大于数据大小。SPI_CFG1 中的 CRCSIZE 字段随后会定义 CRC 计算寄存器中被处理并与 CRC 帧进行比较的最高有效位数。定义过程与数据帧长度无关，但是其必须等于数据帧大小或是数据帧大小的整数倍。

CRC 原理

在使能 SPI (SPE = 1) 前，通过将 SPI_CFG1 寄存器中的 CRCEN 位置 1 来使能 CRC 计算。然后使用由 CRCPOLY 寄存器和 CRC33_17 位定义的 CRC 多项式计算 CRC 值。如果使能了 SPI，则可以更改 CRC 多项式，但只有在总线上无通信时才可进行更改。

按照由 SPIx_CR1 寄存器的 CPHA 位和 CPOL 位定义的采样时钟边沿，逐位进行 CRC 计算。会在 SPI_CR2 寄存器专门定义的数据块末尾，自动检查计算得到的 CRC 值。

如果在接收到的数据内部计算得出的 CRC 与从器件发送器接收到的 CRC 之间检测到不匹配，则 CRCERR 标志将置 1，以指示存在数据损坏错误。CRC 的正确处理步骤取决于 SPI 配置和所选的传输管理。

CRC 传输管理

通信开始后将一直持续到必须发送完或接收完 SPI_DR 寄存器中的最后一个数据帧。

传输的长度必须由 TSIZE 和 TSER 定义。处理完所需的数据量后，将发送 TXCRC，并会将在线路上接收到的数据与 RXCRC 值进行比较。

如果使能了 CRC，则不能将 TSIZE 设置为值 0xFFFF。例如，发送 65535 个具有 CRC 的数据的正确方式是设置：

- TSIZE = 0xFFFE 和 TSER = 1（数据包配置为保存一个数据时）。
- TSIZE = 0xFFFC 和 TSER = 3（数据包保存 4 个数据时——以确保 TSIZE 值与使用了扩展位的数据包大小对齐）。

在发送过程中，CRC 计算在 CRC 事务期间被冻结，TXCRC 将以帧的形式进行发送，帧的长度等于 CRCSIZE 字段值。

在接收过程中，处理完所需的数据量后，RXCRC 也会被冻结。随后以帧的形式接收要与 RXCRC 寄存器内容进行比较的信息，帧长等于 CRCSIZE 值。

完成 CRC 帧后，将执行自动校验，将接收的 CRC 值与 SPIx_RXCRC 寄存器中计算的值进行比较。软件必须校验 SPI_SR 寄存器中的 CRCERR 标志，以确定数据传输是否损坏。软件通过向 CRCERRC 写入 1 将 CRCERR 标志清零。

用户无需考虑刷新冗余 CRC 信息，该过程将自动完成。

复位 SPIx_TXCRC 和 SPIx_RXCRC 值

在 CRC 阶段后对新数据进行采样时，SPI_TXCRC 和 SPI_RXCRC 值将自动初始化。借此，可使用 DMA 循环模式来不间断地传输数据（中间 CRC 校验阶段会涉及一些数据块）。接收器和发送器的初始化模式可配置为零或全 1，具体取决于对 SPI2S_CR1 寄存器的位 TCRCINI 和 RCRCINI 进行的设置。

禁止 SPI 时，CRC 值复位。

50.6 低功耗模式管理

SPI 具有先进的低功耗模式功能，即使禁止 **spi_pclk** 时钟，也仍能在 FIFO 和串行接口之间正常传输数据。

在主模式下，需要 **spi_ker_ck** 内核时钟来提供串行接口的时序。

在从模式下，还可在 FIFO 和串行接口之间进行数据传输时删除 **spi_ker_ck** 时钟。在该模式下，时钟由外部 SPI 器件提供。

对 **spi_pclk** 时钟进行门控（如果 SPI 处于从模式下，则还对 **spi_ker_ck** 时钟进行门控）时，如果诸如以下某个特定操作需要激活 **spi_pclk** 时钟，则 SPI 会提供唤醒事件信号 (**spi_wkup**)：

- 填充 TxFIFO
- 清空 RxFIFO
- 发出其他信号：传输结束、错误...

spi_ker_ck 和 **spi_pclk** 时钟的生成由 RCC 块根据寄存器设置和处理器模式进行控制。更多详细信息，请参见 RCC 部分。

应用程序应在将 SPI 切换到低功耗模式（即删除 **spi_pclk**）之前确认所有挂起的中断事件。

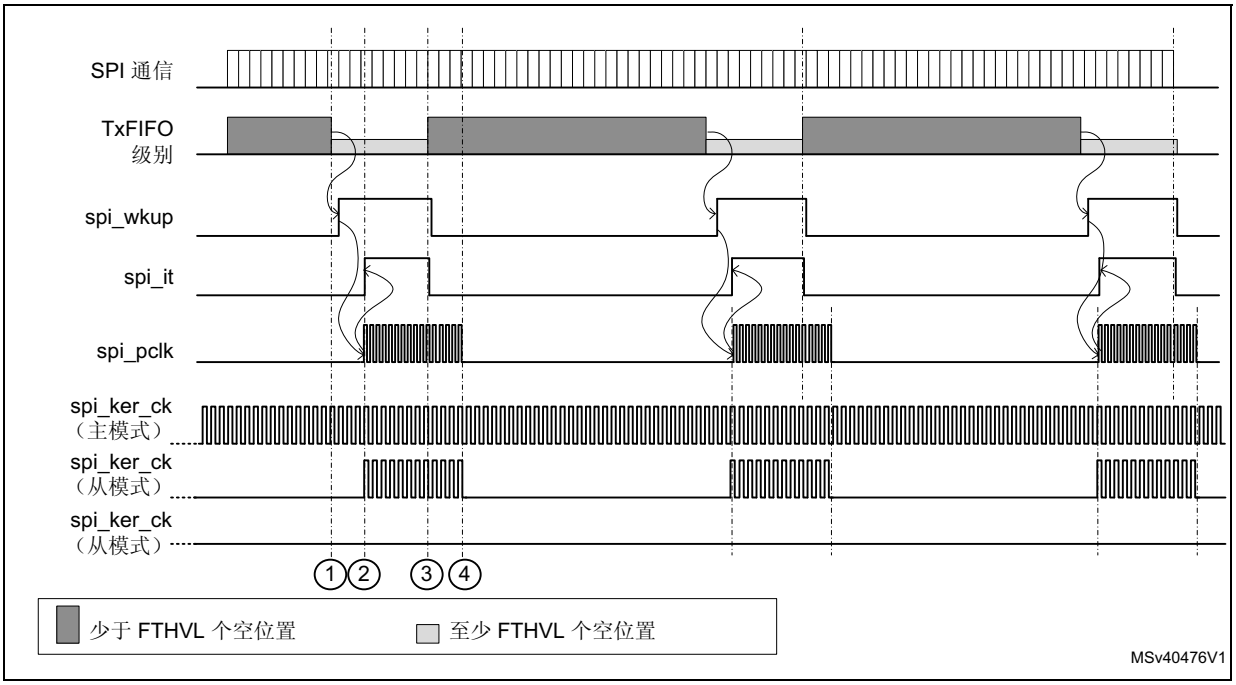
图 617 所示为 SPI2S 工作在低功耗模式时的时钟处理示例。给出的示例适用于发送模式。

在主模式下，生成时序需要 **spi_ker_ck** 时钟。

图 617 所示为从模式支持的两种 **spi_ker_ck** 内核时钟处理情形：

- 在大多数从模式下，均可禁止 **spi_ker_ck** 内核时钟。
- 对于某些产品，可能要根据系统状态激活 **spi_ker_ck** 内核时钟。

图 617. 低功耗模式应用示例



该图清楚地显示，需要将数据从存储器传输到 SPI2S TxFIFO 时，必须向 SPI2S 提供 **spi_pclk**。下面对最重要步骤进行了说明：

- **第 1 步：**
  TxFIFO 级别低于编程的阈值，此事件 (TXP) 会激活 **spi_wkup** 信号。该信号通常用于从低功耗模式唤醒系统，从而激活总线时钟 (**spi_pclk**)。
- **第 2 步：**
  spi_pclk 激活时，**spi_it** 也会激活，产品已准备好通过 DMA 或软件来填充 TxFIFO。另请注意，对于某些产品，系统唤醒还会自动使能 **spi_ker_ck** 内核时钟。
- **第 3 步：**
  TxFIFO 中的空位置数小于 FTHVL 时，将禁用 **spi_wkup** 和 **spi_it** 信号，但 TxFIFO 填充过程可能仍会继续。请注意，**spi_wkup** 下降沿与串行接口时钟域对齐，**spi_it** 的下降沿与 **spi_pclk** 时钟域对齐。
- **第 4 步：**
  TxFIFO 填充已完成；软件可将系统切换回低功耗模式，直至出现下一个 **spi_wkup**。

50.7 SPI 唤醒和中断

表 389 概述了能够生成中断事件 (**spi_it**) 的 SPI 事件。其中的某些事件还具有低功耗模式唤醒功能 (**spi_wkup**)。

大多数事件均可在使用特定中断使能控制位时单独使能和禁止。

表 389. SPI 唤醒和中断请求

中断事件	事件标志(1)	使能控制位	事件清除方法	中断/唤醒是否已激活	
				spi_it	spi_wkup
TxFIFO 准备就绪，可装载数据（存在可供一个数据包使用的空间——FIFO 阈值）	TXP	TXPIE	TxFIFO 包含的空位置数少于 FTHVL 个时，TXP 由硬件清零	是	是
在 RxFIFO 中接收到数据（一个数据包可用——FIFO 阈值）	RXP	RXPIE	RxFIFO 包含的采样数少于 FTHVL 个时，RXP 由硬件清零		是
TXP 和 RXP 均有效	DXP	DXPIE	TXP 或 RXP 被清零时		是
发送传输已填充	TXTF	TXTFIE	向 TXTFC 写入 1		否
下溢	UDR	UDRIE	向 UDRC 写入 1		是
上溢	OVR	OVRIE	向 OVRC 写入 1		是
CRC 错误	CRCE	CRCEIE	向 CRCEC 写入 1		是
TI 帧格式错误	TIFRE	TIFREIE	向 TIFREC 写入 1		否
模式故障 (Mode fault)	MODF	MODFIE	向 MODFC 写入 1		否
传输结束（完整传输序列完成——基于 TSIZE 值）	EOT	EOTIE	向 EOTC 写入 1		是
主模式处于挂起状态	SUSP		向 SUSPC 写入 1		是
TxFIFO 发送完成（TxFIFO 为空）	TXC		总线上开始进行发送活动时，TXC 由硬件清零		否
TSER 值传输到 TSIZE（可将新值装载到 TSER）	TSERF	TSERFIE	向 TSERFC 写入 1		否

1. 有关事件标志的更多详细信息，请参见 SPI2S 寄存器说明。

50.8 I2S 主要特性

- 全双工通信
- 半双工通信（仅作为发送器或接收器）
- 主模式或从模式操作
- 8 位可编程线性预分频器
- 数据长度可以是 16 位、24 位或 32 位
- 在主模式下通道长度可以是 16 位或 32 位，在从模式下则可为任一值
- 可编程时钟极性
- 提高可靠性的错误标志信号：下溢、上溢和帧错误
- 嵌入式 Rx 和 TxFIFO
- 支持的 I^2S 协议：
  - I^2S Philips 标准
  - MSB 对齐标准（左对齐）
  - LSB 对齐标准（右对齐）
  - PCM 标准（支持短帧同步和长帧同步）
- 数据排序可编程（LSb 在前或 MSb 在前）
- 用于发送和接收的 DMA 功能
- 可输出主时钟以驱动外部音频元件。比率固定为 256 x F_{WS}（其中 F_{WS} 为音频采样频率）

50.9 I2S 功能说明

50.9.1 I2S 一般说明

图 603 所示的框图同样适用于 I2S 模式。

I2SMOD 位置 1 时，SPI/I2S 块可工作在 I2S/PCM 模式下。专用寄存器 (SPI_I2SCFGR) 可用于配置专用 I2S 参数，包括时钟发生器和串行链路接口。

将 SPI/I2S 置于主模式下时，I2S/PCM 功能使用时钟发生器生成通信时钟。此时钟发生器也是主时钟输出 (MCK) 的源。

诸如 RxFIFO、TxFIFO、DMA 等资源和部分中断信号与 SPI 功能共享。低功耗模式功能也可用于 I2S 模式，请参见*第 50.6 节：低功耗模式管理*和*第 50.10 节：I2S 唤醒和中断*。

50.9.2 与 SPI 功能共享的引脚

I2S 与 SPI 共享以下四个共用引脚：

- SDO：串行数据输出（映射到 MOSI 引脚上），用于在主模式下发送音频采样以及在从模式下接收音频采样。请参见*第 1999 页的串行数据线交换*一节。
- SDI：串行数据输入（映射到 MISO 引脚上），用于在主模式下接收音频采样以及在从模式下发送音频采样。请参见*第 1999 页的串行数据线交换*一节。
- WS：字选择（映射到 SS 引脚上），表示帧同步。它在主模式下被配置为输出，在从模式下被配置为输入。
- CK：串行时钟（映射到 SCK 引脚上），表示串行位时钟。它在主模式下被配置为输出，在从模式下被配置为输入。

当某些外部音频设备需要使用主时钟输出时，可以使用其它引脚：

- MCK：主时钟（单独映射），适用于 I2S 配置为主模式的情况。主时钟频率固定为 256 x F_{WS}（其中 F_{WS} 为音频采样频率）。

50.9.3 I2S/PCM 模式下可用的位和位域

选择 I2S/PCM 模式（I2SMOD = “1”）时，一些位域不再可用，必须将其强制设为特定值，以便保证 I2S/PCM 功能正常工作。*表 390* 列出了 I2S/PCM 模式下可用的位和位域，并指示了其中哪些必须强制设为特定值。

表 390. PCM/I2S 模式下可用的位域

寄存器名称	PCM/I2S 模式下可用的位域	关于其他位域的限制
SPI2S 控制寄存器 1 (SPI/I2S_CR1)	IOLOCK、CSUSP、CSTART	其他字段设为其各自的复位值
SPI 控制寄存器 2 (SPI_CR2)	-	设为复位值
SPI 配置寄存器 1 (SPI_CFG1)	TXDMAEN、RXDMAEN、FTHVL	其他字段设为其各自的复位值
SPI 配置寄存器 2 (SPI_CFG2)	AFCNTR、LSBFRST、IOSWP	其他字段设为其各自的复位值
SPI/I2S 中断使能寄存器 (SPI2S_IER)	TIFREIE、OVRIE、UDRIE、TXPIE、RXPIE	
SPI/I2S 状态寄存器 (SPI2S_SR)	RXWNE、RXPLVL、SUSP、TIFRE、OVR、UDR、TXP、RXP	其他标志不相关
SPI/I2S 中断/状态标志清零寄存器 (SPI2S_IFCR)	SUSPC、TIFREC、OVRC、UDRC	其他字段设为其各自的复位值
SPI/I2S 发送数据寄存器 (SPI2S_TXDR)	整个寄存器	-
SPI/I2S 接收数据寄存器 (SPI2S_RXDR)	整个寄存器	-
SPI 多项式寄存器 (SPI_CRCPOLY)	-	设为复位值
SPI 发送器 CRC 寄存器 (SPI_TXCRC)	-	
SPI 接收器 CRC 寄存器 (SPI_RXCRC)	-	
SPI 下溢数据寄存器 (SPI_UDRDR)	-	
SPI/I2S 配置寄存器 (SPI_I2SCGFR)	整个寄存器	-

50.9.4 从模式和主模式

SPI/I2S 块都支持采用 I2S 或 PCM 协议的主模式和从模式。
在主模式下，CK、WS 和 MCK 信号均被设置为输出。

在从模式下，CK 和 WS 信号被设置为输入。从模式下不使用信号 MCK。

为了改善从模式下 SPI/I2S 块的稳健性，外设根据 WS 信号在每次接收和发送时再同步。这意味着：

- 使用 I2S Philips 标准时，在每个 WS 沿跳变之后，每个数据的移入或移出都会被触发一个时钟周期。
- 使用 I2S MSB 对齐标准时，如果检测到 WS 沿跳变，则会触发各个数据的移入或移出。
- 使用 PCM 标准时，在 WS 上升沿之后，每个数据的移入或移出都会被触发一个时钟周期。

注： *这种再同步机制不适用于 I2S LSB 对齐标准。*

注： *另请注意，将 SPI/I2S 配置为从模式时，无需提供内核时钟。*

50.9.5 支持的音频协议

I2S/PCM 接口支持四种音频标准，可使用 SPIx_I2SCFGR 寄存器中的 I2SSTD[1:0] 和 PCMSYNC 位对其进行配置。

使用 I2S 协议时，在左右两个通道上时分复用音频数据。WS 信号用于指示哪个通道应被视为左通道，以及哪个通道应被视为右通道。

在 I2S 主模式下，支持以下四种帧格式：

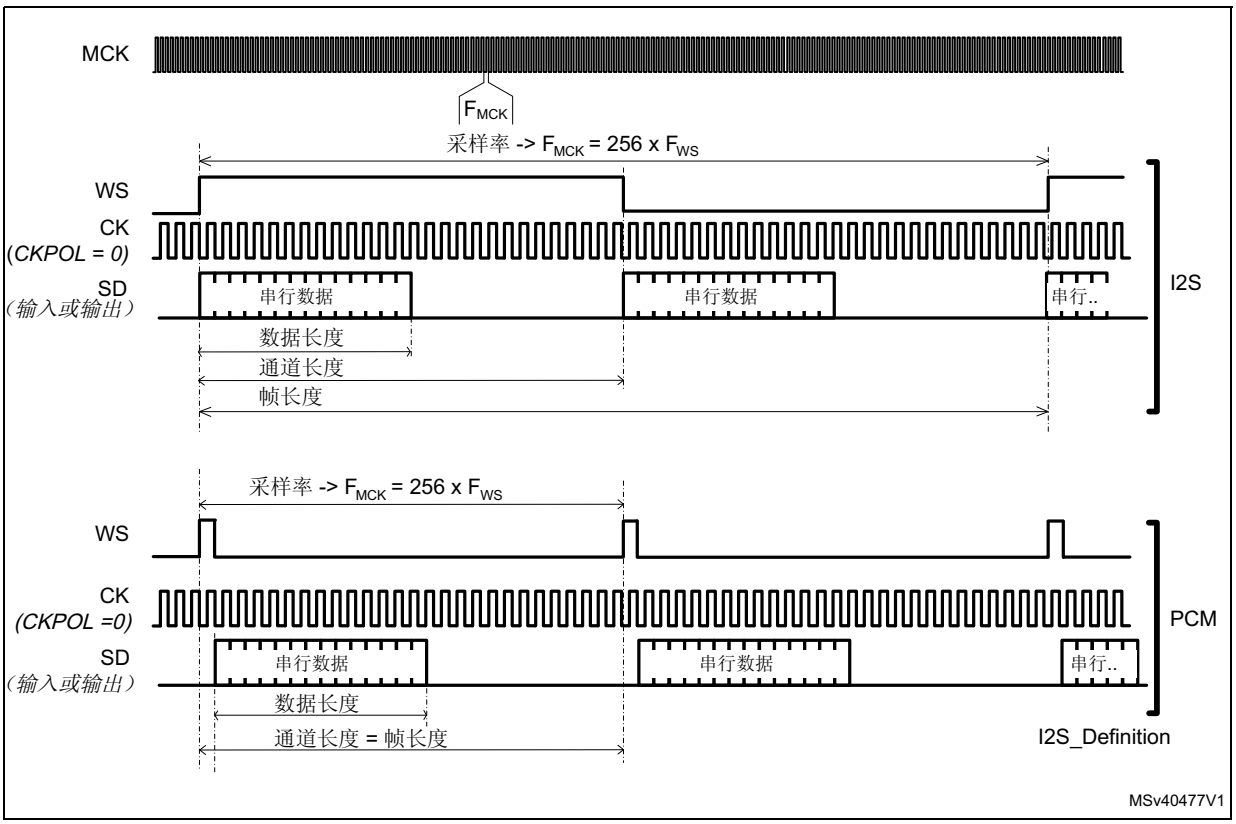
- 将 16 位数据封装在 16 位通道中
- 将 16 位数据封装在 32 位通道中
- 将 24 位数据封装在 32 位通道中
- 将 32 位数据封装在 32 位通道中

在 PCM 主模式下，支持以下三种帧格式：

- 将 16 位数据封装在 16 位通道中
- 将 16 位数据封装在 32 位通道中
- 将 24 位数据封装在 32 位通道中

下图显示了本节中使用的主要定义：数据长度、通道长度和帧长度。

图 618. 波形示例

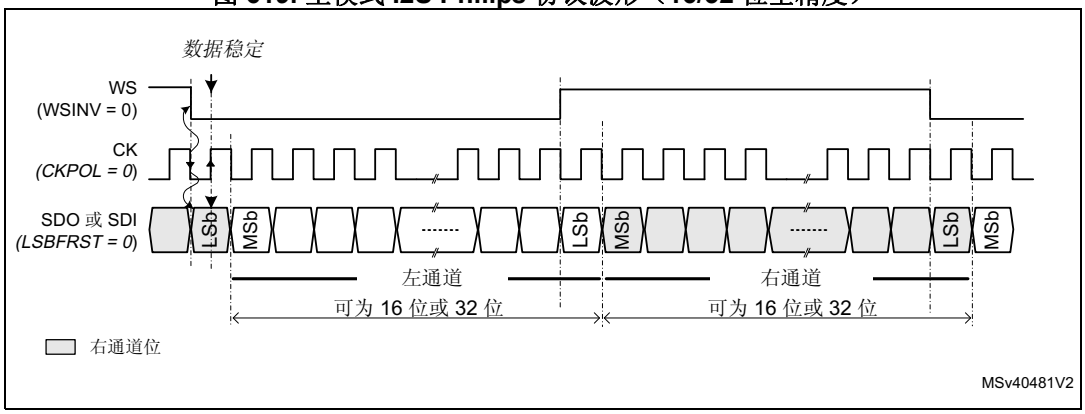


I2S Philips 标准

要选择 I2S Philips 标准，将 I2SSTD 设为 0b00 即可。主模式和从模式均支持该标准。

对于该标准，在第一个位（对于 I2S Philips 标准，为 MSb）可用之前 WS 信号将翻转一个 CK 时钟周期。WS 的下降沿跳变表示下一个传输的数据将用于左通道，上升沿跳变则表示下一个传输的数据用于右通道。

图 619. 主模式 I2S Philips 协议波形（16/32 位全精度）

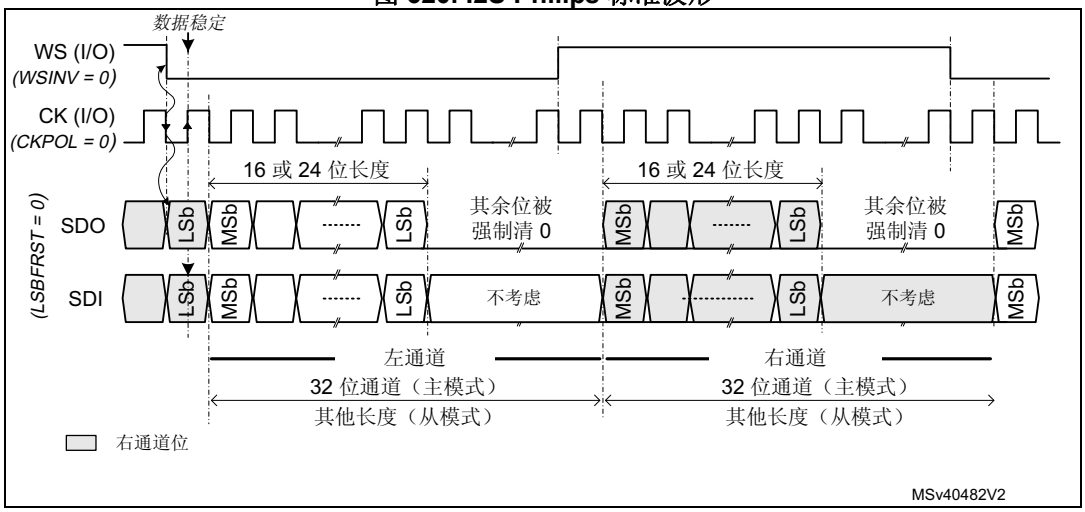


为与 I2S Philips 协议匹配，要将 CKPOL 设为 0。有关 WS 信号处理的信息，请参见 *CK 采样边沿的选择*。

图 619 所示为通道长度等于数据长度时 SPI/I2S 所生成的波形示例。更确切地说，这适用于 CHLEN = 0 且 DATLEN = 0b00 或 CHLEN = 1 且 DATLEN = 0b10 的情况。

有关 WS 信号处理的信息，请参见 *WS 反转控制*。

图 620. I2S Philips 标准波形

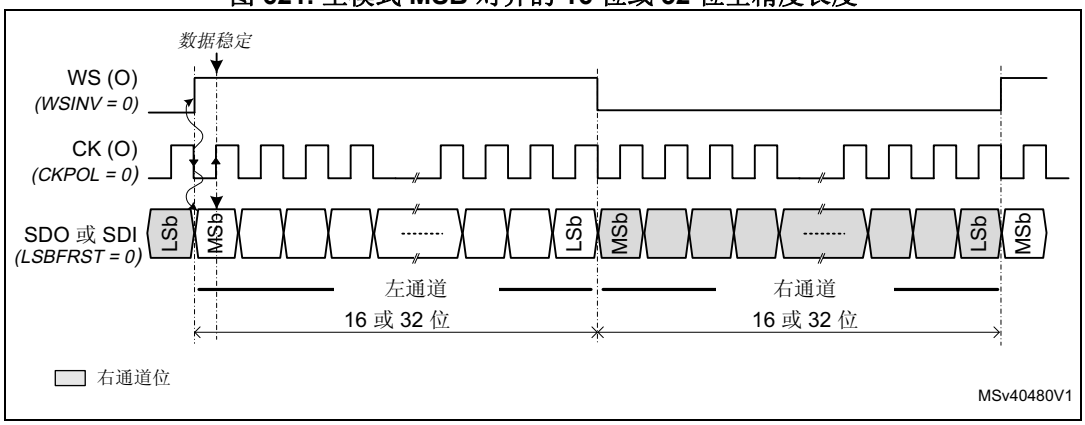


如果通道长度大于数据长度，则将 SPI/I2S 配置为发送模式时，其余位被强制清零。这同时适用于主模式和从模式。

MSB 对齐标准

对于该标准，提供第一个数据位时，WS 信号翻转。如果 WS 为高电平，则传输的数据表示左通道；如果 WS 为低电平则表示右通道。

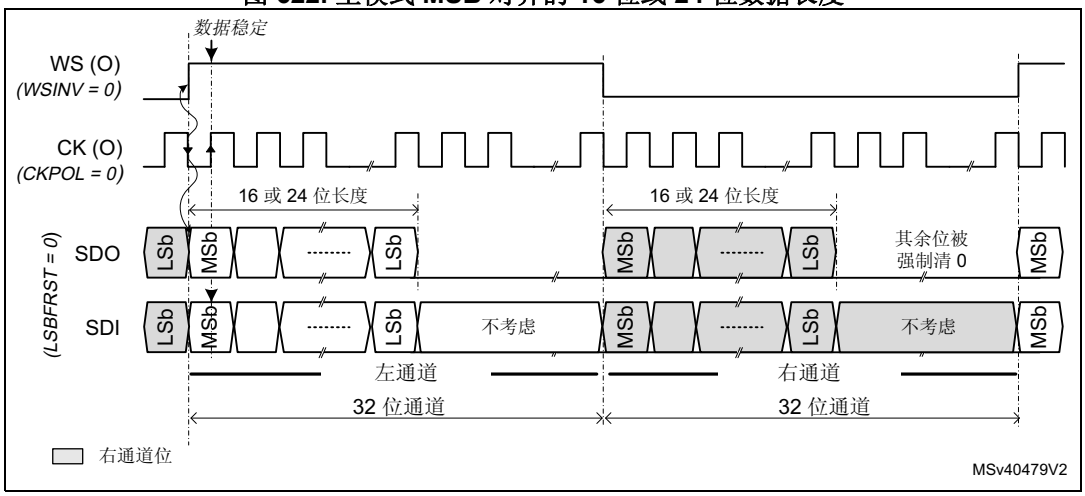
图 621. 主模式 MSB 对齐的 16 位或 32 位全精度长度



为与 I2S MSB 对齐协议匹配，要将 CKPOL 设为 0。有关 WS 信号处理的信息，请参见 *CK 采样边沿的选择*。

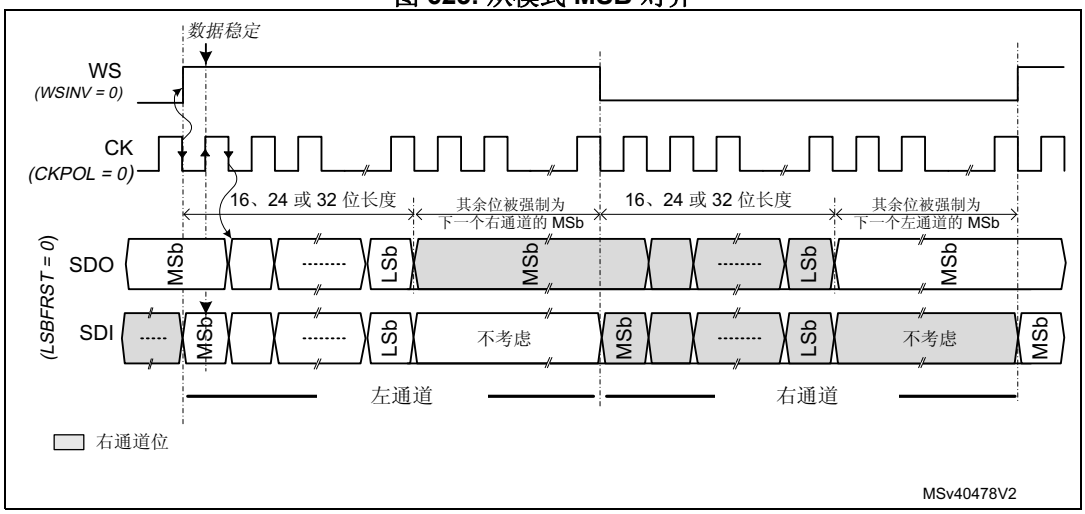
有关 WS 信号处理的信息，请参见 *WS 反转控制*。

图 622. 主模式 MSB 对齐的 16 位或 24 位数据长度



如果通道长度大于数据长度，则将 SPI/I2S 配置为主器件发送模式时，其余位被强制清零。在从器件发送模式下，其余位被强制为要生成的下一个数据的第一位的值，以避免出现时序问题（请参见*图 623*）。

图 623. 从模式 MSB 对齐

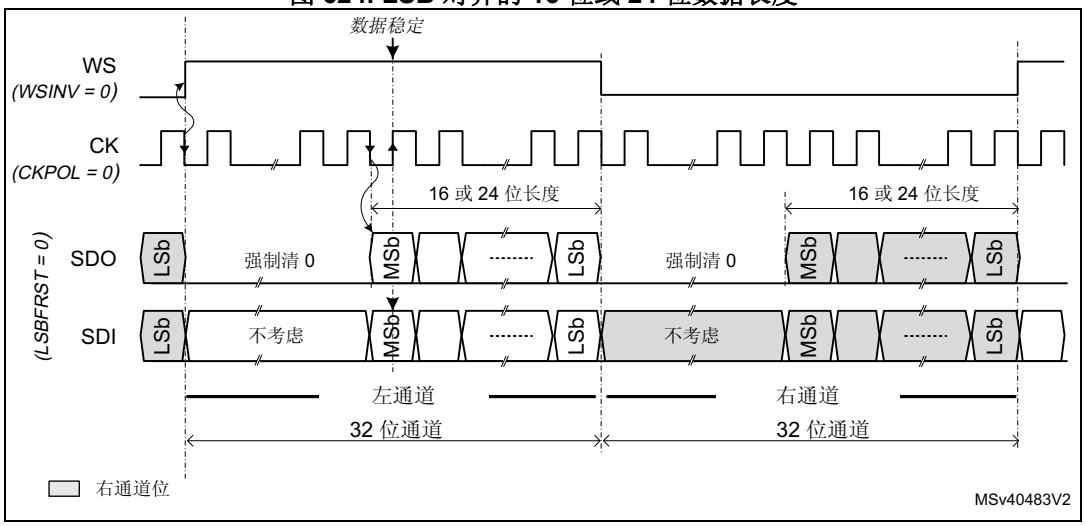


LSB 对齐标准

该标准与主模式下的 MSB 对齐标准类似（对于 16 位和 32 位全精度帧格式，没有任何不同）。LSB 对齐的 16 位或 32 位全精度格式的波形与 MSB 对齐模式下的波形类似（请参见*图 621*），因为通道和数据的长度相同。

注： *在 LSB 对齐格式下，主模式和从模式仅支持 16 位和 32 位通道长度。这是因为，如果发送器和接收器侧不清楚通道长度，则无法正确传输数据。*

图 624. LSB 对齐的 16 位或 24 位数据长度



为与 I2S LSB 对齐协议匹配，要将 CKPOL 设为 0。有关 WS 信号处理的信息，请参见 *CK 采样边沿的选择*。

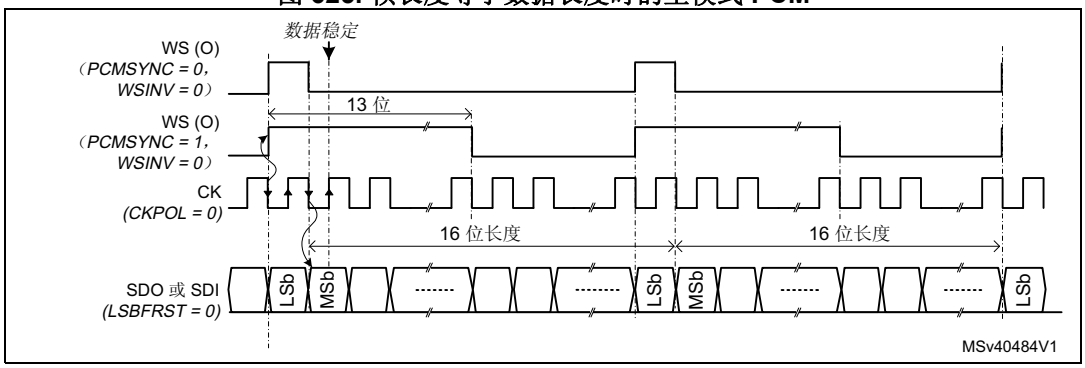
有关 WS 信号处理的信息，请参见 *WS 反转控制*。

PCM 标准

对于 PCM 标准，无需使用通道信息。可使用两种 PCM 模式（短帧和长帧），并且可使用 SPI_I2SCFGR 寄存器中的 PCMSYNC 位来配置。

注： *PCM 长帧和短帧之间的区别仅仅是帧同步的宽度：对于两种协议，帧的有效边沿在第一个位的一个 CK 时钟周期之前被生成（在从模式下则为预期）。*

图 625. 帧长度等于数据长度时的主模式 PCM



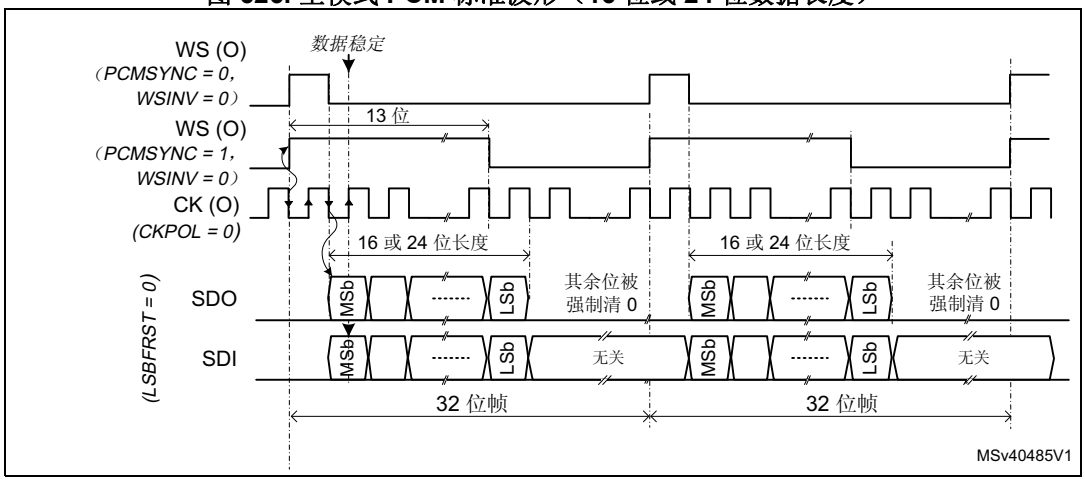
对于长帧同步，在主模式下会将 WS 信号持续 13 个周期。

将通道长度设置为 32 位时，可以使用 16 位或 24 位数据大小。

对于短帧同步，WS 同步信号的持续时间仅为一个周期。

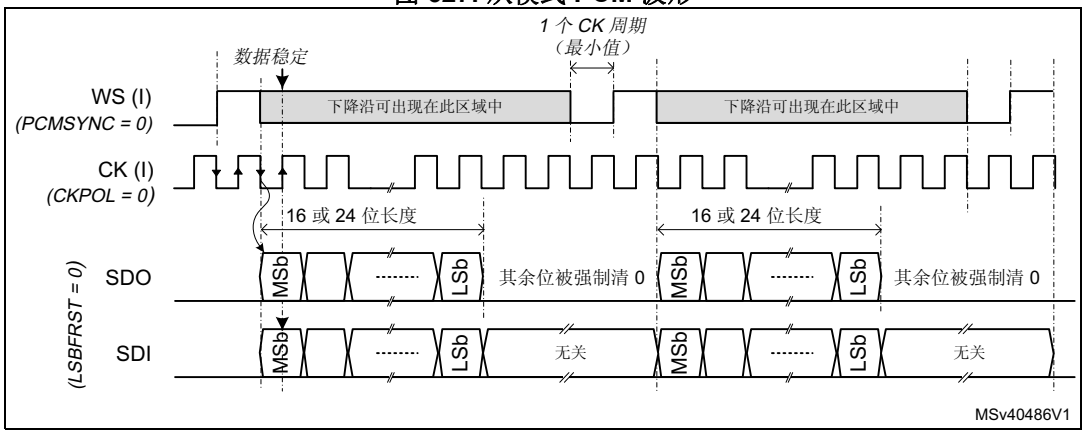
有关 WS 信号处理的信息，请参见 *WS 反转控制*。

图 626. 主模式 PCM 标准波形（16 位或 24 位数据长度）



如果将 PCM 协议用于从模式，则帧长度可以不是 16 位或 32 位。如*图 627* 所示，在从模式下，可以接受各种脉冲宽度的 WS，因为通过 WS 的上升沿检测帧起始。唯一的限制是，WS 必须返回至无效状态并至少持续一个 CK 周期。

图 627. 从模式 PCM 波形



为与 PCM 协议匹配，要将 CKPOL 设为 0。有关 WS 信号处理的信息，请参见*CK 采样边沿的选择*。

50.9.6 更灵活的串行接口

从模式下的帧长度可变

在从模式下，只要通道长度大于数据长度，便可接受非 16 位或 32 位的通道长度。这适用于除 I2S LSB 对齐协议外的所有协议。

数据排序

对于所有数据格式和通信标准，可通过 *SPI 配置寄存器 2 (SPI_CFG2)* 的 LSBFRST 位来选择数据排序（MSb 在前或 LSb 在前）。

CK 采样边沿的选择

SPI/I2S 配置寄存器 (SPI_I2SCGFR) 的位 CKPOL 允许用户为从模式和主模式选择 CK 的采样边沿极性，这适用于所有协议。

- CKPOL = 0 时，在 CK 的下降沿更改串行数据 SDO 和 WS（主模式下），在上升沿读取串行数据 SDI 和 WS（从模式下）。
- CKPOL = 1 时，在 CK 的上升沿更改串行数据 SDO 和 WS（主模式下），在下降沿读取串行数据 SDI 和 WS（从模式下）。

WS 反转控制

可通过将 WSINV 置 1，为主模式和从模式反转默认 WS 信号极性，这适用于所有协议。默认情况下，WS 极性如下：

- 使用 I2S Philips 标准时，WS 对于左通道为低电平，对于右通道为高电平
- 在 MSB/LSB 对齐模式下，WS 对于左通道为高电平，对于右通道为低电平
- 在 PCM 模式下，帧起始通过 WS 的上升沿表示

WSINV 置 1 时，WS 极性将反转，因此：

- 使用 I2S Philips 标准时，WS 对于左通道为高电平，对于右通道为低电平
- 在 MSB/LSB 对齐模式下，WS 对于左通道为低电平，对于右通道为高电平
- 在 PCM 模式下，帧起始通过 WS 的下降沿表示

WSINV 位于 *SPI/I2S 配置寄存器 (SPI_I2SCGFR)* 中。

IO 的控制

SPI/I2S 块借助 *SPI 配置寄存器 2 (SPI_CFG2)* 的 AFCNTR 位，允许在使能 SPI/I2S 之前将 WS 和 CK 信号稳定在无效状态。

可通过使用以下序列编程 CKPOL 和 WSINV 来完成上述过程：

假设 AFCNTR 最初被设为 0

- 将 I2SMOD 置 1（以通知硬件通过 CKPOL 和 WSINV 控制 CK 和 WS 极性）。
- 将位 CKPOL 和 WSINV 设为所需值。
- 将 AFCNTR 置 1。
  随即会根据 CKPOL 和 WSINV 值设置 CK 和 WS IOs 的无效电平，即使尚未使能 SPI/I2S，亦如此。
- 然后执行 I2S/PCM 的激活序列。

表 391 所示为将 AFCNTR 位置 1 时以及在使能 SPI/I2S 块之前的 WS 和 CK 信号电平（即，无效电平）。请注意，WS 电平还取决于所选协议。

表 391. AFCNTR = 1 时在使能 SPI/I2S 之前的 WS 和 CK 电平

WSINV	I2SSTD		使能 SPI/I2S 之前的 WS 电平
0	I2S 标准 (00)	→	高
	其它	→	低
1	I2S 标准 (00)	→	低
	其它	→	高

CKPOL		使能 SPI/I2S 之前的 CK 电平
0	→	低
1	→	高

注： *SPI2S 处于从模式下时，位 AFCNTR 不应置 1。*

串行数据线交换

SPI/I2S 能通过 *SPI 配置寄存器 2 (SPI_CFG2)* 的 IOSWP 位来交换 SDI 和 SDO 线的功能。*表 392* 给出了该功能的详细信息。

表 392. 串行数据线交换

配置	IOSWP	SDI 方向	SDO 方向
主/从 RX	0	输入	-
	1	-	输入
主/从 TX	0	-	输出
	1	输出	-
主/从全双工	0	输入	输出
	1	输出	输入

简单起见，*I2S 功能说明*部分所示的波形都是基于 IOSWP = 0。

50.9.7 启动序列

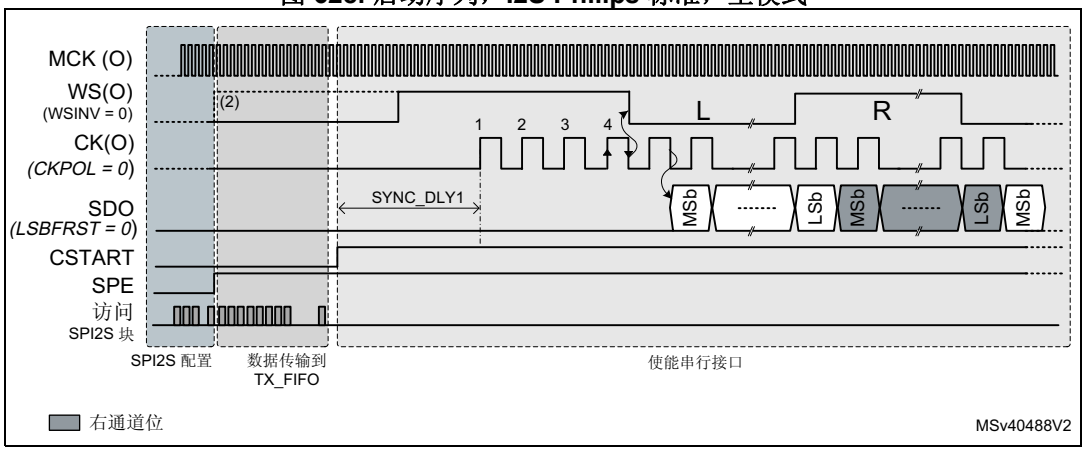
将 SPE 位设为 0 时，不允许用户对 SPI2S_RXDR 和 SPI2S_TXDR 寄存器进行读操作和写操作，但允许访问其他寄存器。

如果应用要使用 SPI/I2S 块，则用户必须按以下步骤进行操作：

1. 确保将 SPE 设为 0，否则会向 SPE 写入 0。
2. 根据所需配置去编程所有配置和控制寄存器。有关详细的编程示例，请参见*第 50.9.16 节*。
3. 将 SPE 位置 1，以激活 SPI/I2S 块。该位置 1 时，串行接口仍然被禁止，但 DMA 和中断服务处于工作状态，因而可将数据传输到 TxFIFO。
4. 将位 CSTART 置 1，以激活串行接口。

如*图 628* 所示，对于符合 I2S Philips 标准的主 TX，一旦将 CSTART 位置 1 且 TxFIFO 非空，就会开始生成 WS、MCK 和 CK 信号。请注意，在 WS 下降沿之前的 4 个上升沿，位时钟 CK 处于激活状态，以确保外部从器件可正确检测 WS 沿跳变。其他标准的相应行为与此类似。

图 628. 启动序列，I2S Philips 标准，主模式



1. 在该图中，在将 SPE 位置 1 之前使能了 MCK。更多信息，请参见 *MCK 生成*。
2. 请注意，如果将 AFCNTR 位置 1，则在配置阶段，WS 和 CK 信号的电平将由 SPI/I2S 块进行控制。

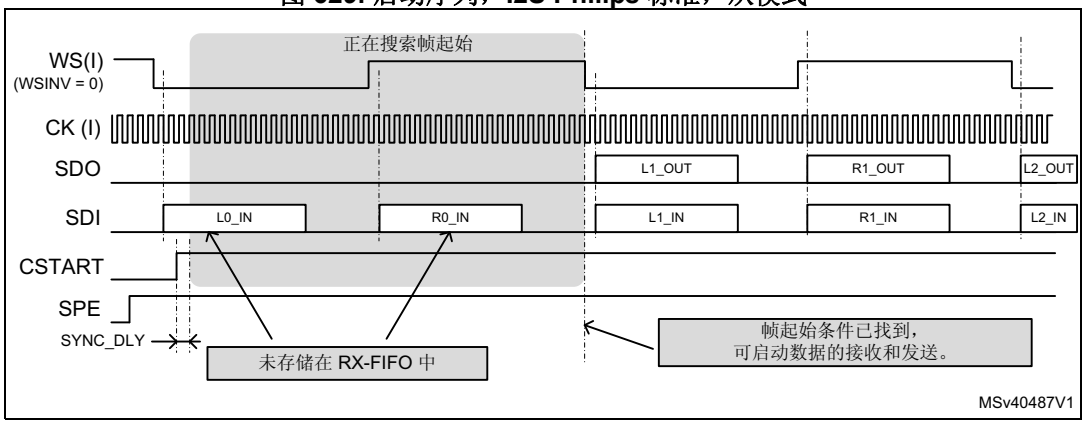
注： *由于时钟域再同步，因此在约 3 个 CK 时钟周期 (SYNC_DLY1) 后硬件会考虑 CSTART 位。*

在从模式下，如果将 CSTART 位置 1，则在满足帧起始条件时将开始数据传输：

- 对于 I2S Philips 标准，帧起始条件为 WS 信号下降沿。将在延迟一个时钟周期后开始发送/接收。
  如果 WSINV = 1，则帧起始条件为上升沿。
- 对于其他协议，帧起始条件为 WS 信号上升沿。对于 MSB 对齐协议，在 WS 上升沿开始发送/接收。对于 PCM 协议，将在延迟一个时钟周期后开始发送/接收。
  如果 WSINV = 1，则帧起始条件为下降沿。

图 629 所示为采用 I2S Philips 标准的从模式的启动序列示例。

图 629. 启动序列，I2S Philips 标准，从模式



注： *由于时钟域再同步，因此在 2 个 CK 时钟周期 (SYNC_DLY) 后硬件会考虑 CSTART 位。*

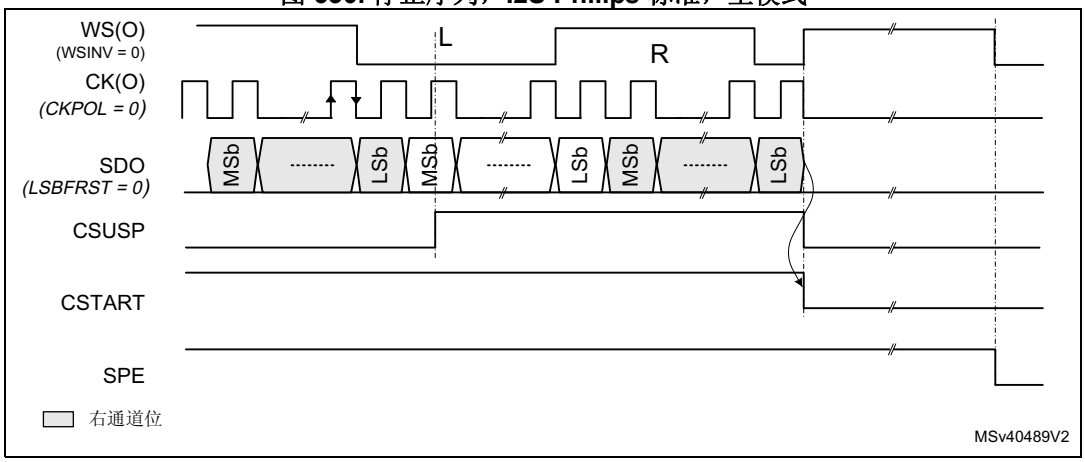
50.9.8 停止序列

应用可通过将 SPE 位置 0 来停止 I2S/PCM 传输。在这种情况下，通信将立即停止，而不会等到当前帧结束。

在主模式下，还可以在当前帧结束时停止 I2S/PCM 传输。为此，用户必须将 CSUSP 位置 1，并轮询 CSTART 位，直到其值为 0。当前立体声（如果选择了 I2S 模式）或单声道采样完全移入或移出时，CSTART 位将变为 0。之后可将 SPE 位设为 0。

图 630 所示为主模式下的停止序列示例。CSUSP 位置 1，在传送左采样期间，传输继续进行，直到传输右采样的最后一位。随后 CSTART 和 CSUSP 返回至 0，CK 和 WS 信号返回到无效状态，用户可以将 SPE 设为 0。

图 630. 停止序列，I2S Philips 标准，主模式

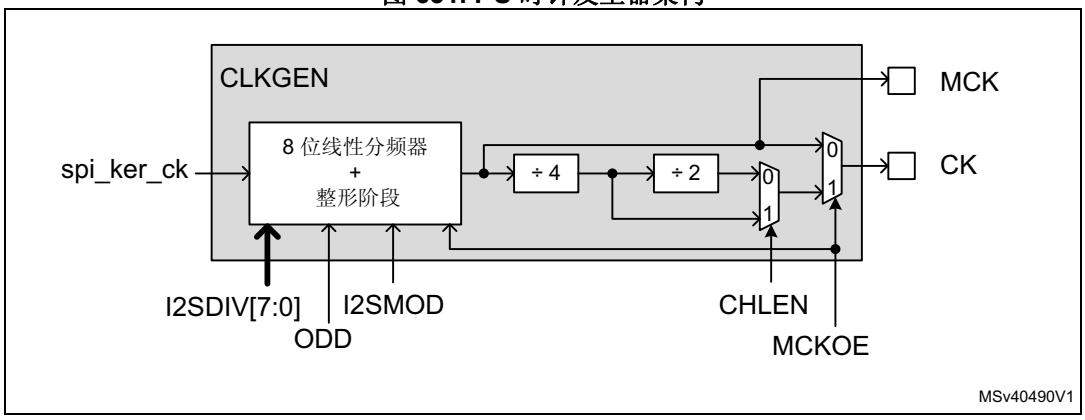


注： *在从模式下，停止序列仅由 SPE 位控制。*

50.9.9 时钟发生器

将 I2S 或 PCM 配置为主模式时，用户需要对时钟发生器进行编程，以便基于所需频率产生帧同步 (WS)、位时钟 (CK) 和主时钟 (MCK)。
如果将 I2S 或 PCM 用于从模式下，则无需配置时钟发生器。

图 631. I^2S 时钟发生器架构



MCK、CK 和 WS 上产生的频率主要取决于 I2SDIV、ODD、CHLEN 和 MCKOE。MCKOE 位用于指示是否需要生成主时钟。主时钟的频率比帧同步的频率高 256 倍。如果要为外部音频编解码器提供参考时钟，则通常需要该主时钟。

注： *在主模式下，总线时钟频率 (F_{pclk}) 和位时钟频率 (F_{CK}) 之比无特定限制。总线时钟频率必须足够高，这样才能支持数据吞吐量。*

在 I2S 模式下，生成主时钟 (MCKOE = 1) 时，可通过以下公式得出帧同步的频率：

$$F_{WS} = \frac{F_{i2s_clk}}{256 \times \{(2 \times I2SDIV) + ODD\}}$$

在 PCM 模式下通过以下公式得出频率：

$$F_{WS} = \frac{F_{i2s_clk}}{128 \times \{(2 \times I2SDIV) + ODD\}}$$

此外，可通过下列公式得出 MCK 频率 (F_{MCK})：

$$F_{MCK} = \frac{F_{i2s_clk}}{\{(2 \times I2SDIV) + ODD\}}$$

在 I2S 模式下，禁止主时钟 (MCKOE = 0) 时，可通过以下公式得出帧同步的频率：

$$F_{WS} = \frac{F_{i2s_clk}}{32 \times (CHLEN + 1) \times \{(2 \times I2SDIV) + ODD\}}$$

在 PCM 模式下通过以下公式得出频率：

$$F_{WS} = \frac{F_{i2s_clk}}{16 \times (CHLEN + 1) \times \{(2 \times I2SDIV) + ODD\}}$$

其中，F_{WS} 表示帧同步的频率，F_{i2s_clk} 表示提供给 SPI/I2S 块的内核时钟的频率。

注： *CHLEN 和 ODD 既可以为 0 也可以为 1。*
ODD = 0 时，I2SDIV 可以为 0 到 255 之间的任一值；但当 ODD = 1 时，不允许值 I2SDIV = 1。
I2SDIV = 0 时，{(2 x I2SDIV) + ODD} 强制为 1。

注： *{(2 x I2SDIV) + ODD} 为奇数时，MCK 或 CK 信号的占空比不会为 50%。必须谨慎使用奇数比：因为它会影响建立和保持时间的裕量。例如，如果 {(2 x I2SDIV) + ODD} = 5，则占空比可为 40%。*

表 393 所示为适用于 I2S 模式的时钟发生器编程示例。

MCK 生成

无论 SPE 位如何，都可生成主时钟 MCK。MCK 的生成由以下位控制：

- I2SMOD 必须等于 1，
- I2SCFG 必须选择主模式，
- MCKOE 必须设置为 1。

表 393. 适合常见 I2S 频率的 CLKGEN 编程示例

i2s_clk (MHz)	通道长度（位）	I2SDIV	ODD	MCK	采样率：F_{WS} (kHz)
12.288	16	12	0	无	16
12.288	32	6	0		16
12.288	16	6	0		32
12.288	32	3	0		32
49.152	16	16	0		48
49.152	32	8	0		48
49.152	16	8	0		96
49.152	32	4	0		96
49.152	16	4	0		192
49.152	32	2	0		192

表 393. 适合常见 I2S 频率的 CLKGEN 编程示例（续）

i2s_clk (MHz)	通道长度（位）	I2SDIV	ODD	MCK	采样率：F_{WS} (kHz)
4.096	16 或 32	0	-	有	16
24.576	16 或 32	3	0		
49.152	16 或 32	3	0		32
12.288	16 或 32	0	-		48
49.152	16 或 32	2	0		
61.44	16 或 32	2	1		
98.304	16 或 32	2	0		96
196.608	16 或 32	2	0		192

50.9.10 内部 FIFO

I2S 接口可将专用 FIFO 用于 RX 和 TX 路径。可通过 SPI2S_TXDR 寄存器将要发送的采样写入 TxFIFO。通过 SPI2S_RXDR 寄存器读取 RxFIFO。

数据对齐和排序

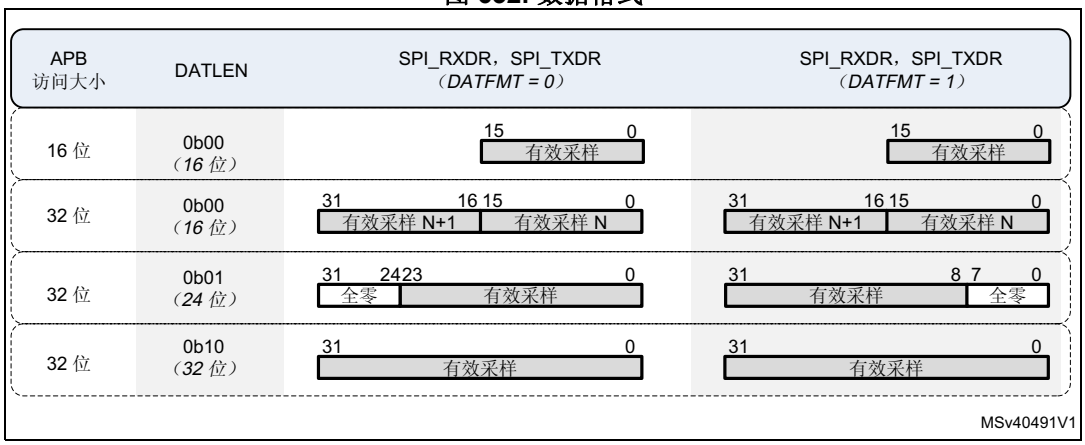
可通过 DATFMT 位选择对齐到 SPI2S_RXDR 和 SPI2S_TXDR 寄存器的数据。

另请注意，SPI2S_RXDR 或 SPI2S_TXDR 中数据的格式还取决于通过 APB 总线对这些寄存器进行访问的方式。

图 632 显示了 APB 访问大小、DATFMT 和 DATLEN 之间的允许设置。

注： *APB 访问大小为 32 位且 DATLEN = 0 时应格外注意。对于读操作，RxFIFO 必须至少包含两个数据，否则读取的数据将无效。同样，对于写操作，TxFIFO 必须至少有两个空位置，否则数据会丢失。*

图 632. 数据格式



1. 在 I2S 模式下，采样 N 表示左采样，采样 N+1 表示右采样。

可根据可编程的 FIFO 阈值生成中断或 DMA 请求。FIFO 阈值对 RX 通用，TxFIFO 可通过 FTHVL 进行调整。

在 I2S 模式下，左音频采样和右音频采样在 FIFO 中交错。这表示，对于发送操作，用户必须开始先用左采样然后再用右采样来填充 TxFIFO，依此类推。对于接收模式，从 RxFIFO 中读取的第一个数据用于表示左通道，下一个数据用于表示右通道，依此类推。

请注意，SPE 位置 0 时，FIFO 的读指针和写指针将复位。

更多相关信息，请参见*第 50.9.11 节*和*第 50.9.15 节*。

FIFO 大小优化

字节是 FIFO 的基本元素。这样可以优化 FIFO 位置数。例如，数据大小固定为 24 位时，每个音频采样都会占用 3 个基本 FIFO 元素。

举例来说，具有 16 个基本元素的 FIFO 的深度可为：

- 8 个采样，适合 DATLEN = 0 的情况（16 位）
- 5 个采样，适合 DATLEN = 1 的情况（24 位）
- 4 个采样，适合 DATLEN = 2 的情况（32 位）

50.9.11 FIFOs 状态标志

应用程序可通过两个状态标志来全面监视 I2S 接口的状态。两个标志均可生成一个中断请求。如果使能 RXPIE 位则会生成接收中断，如果使能 TXPIE 位则会生成发送中断。这些位均位于 SPI_IER 寄存器中。

达到了 TxFIFO 阈值 (TXP)

该标志置 1 时，即表示 TxFIFO 至少包含 FTHVL 个空位置。因此，可向 SPI2S_TXDR 中写入 FTHVL 个待发送的新数据。TXP 标志在空位置数少于 FTHVL 时复位。请注意，禁止 I2S（SPE 位复位）时 TXP = 1。

达到了 RxFIFO 阈值 (RXP)

该标志置 1 时，即表示 RxFIFO 中至少有 FTHVL 个有效数据，因此用户可通过 SPI2S_RXDR 读取这些数据。RxFIFO 包含的数据少于 FTHVL 时，它将复位。

有关 I2S 模式下中断功能的更多信息，请参见*第 50.10 节*。

50.9.12 下溢情况的处理

在发送模式下，需要将新数据载入移位寄存器中，而 TxFIFO 已空时，UDR 标志置 1。在这种情况下，至少会丢失一个数据。

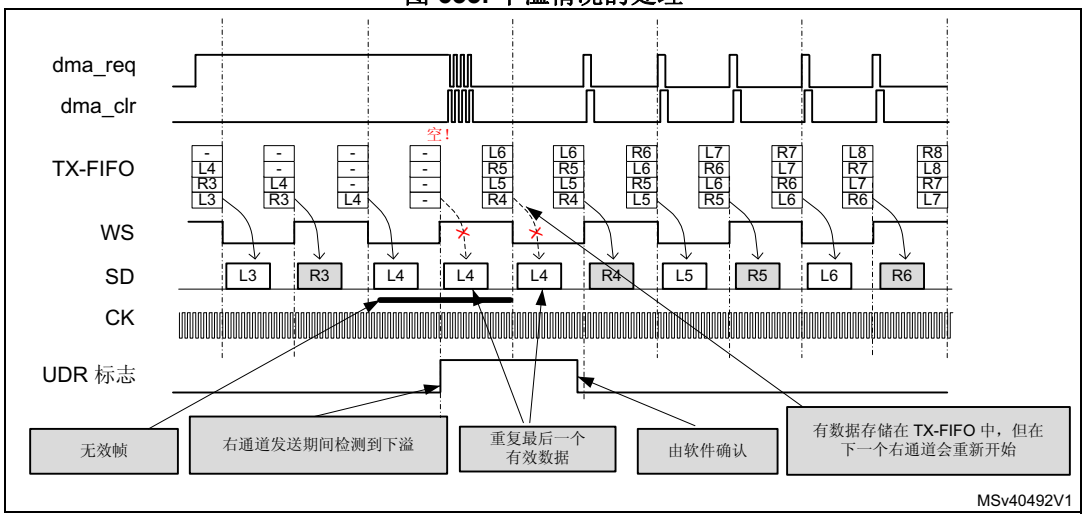
在 I2S 模式下，存在一种硬件机制，可防止出现偏离情况（左右通道互换）。如下图所示，发生下溢时，只要不满足重新启动条件，外设就会重复左右通道上的最后一个有效数据。将在以下情况下重新启动发送：

- TxFIFO 中存在足够的数据时，
- UDR 标志由软件清零时。

之后，下一个要发送的数据将是：

- 右通道（适合需要发送右通道数据时发生下溢的情况），或
- 左通道（适合需要发送左通道数据时发生下溢的情况）。

图 633. 下溢情况的处理



如果 SPI_IER 寄存器的 UDRIE 位置 1，则 UDR 标志可触发中断。通过向 SPI_IFCR 寄存器的 UDRC 位写入 1 来将 UDR 位清零。

将块配置为 PCM 模式时，不会激活此重新对齐机制。

注： *主模式或从模式下都可能发生下溢情况。在主模式下，发生下溢时，不会对 WS、CK 和 MCK 信号进行门控。*

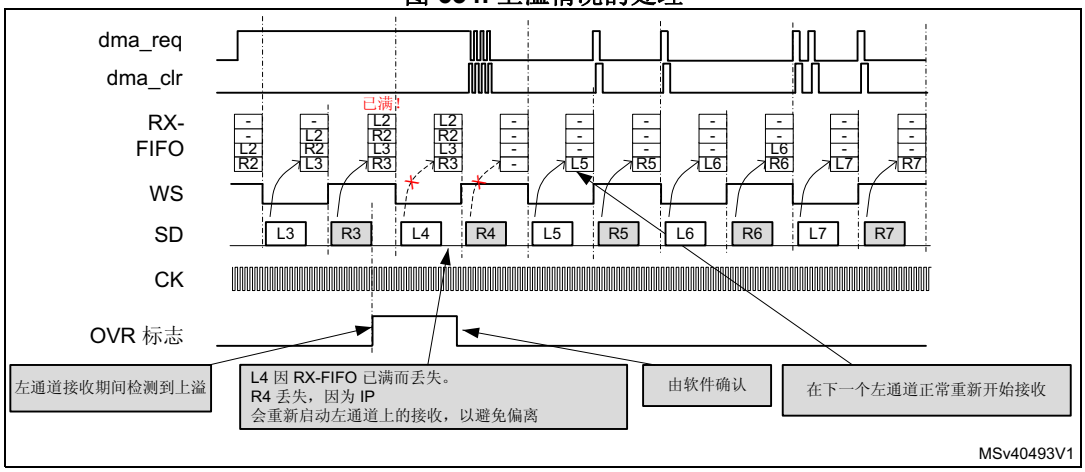
出于再同步的原因，在经过至少 2 个 CK 时钟周期后，硬件将考虑 UDR 标志的变化。

50.9.13 上溢情况的处理

需要将接收到的数据写入 RxFIFO 中，而 RxFIFO 已满时，OVR 标志置 1。因此，某些传入的数据会丢失。

在 I2S 模式下，存在一种硬件机制，可防止出现偏离情况（左右通道互换）。如下图所示，发生上溢时，只要不满足重新启动条件，外设就会停止向 RxFIFO 中写入数据。RxFIFO 中有足够的空间，且 OVR 标志清零时，如果在接收右通道数据时发生上溢，则块会通过向 RxFIFO 中写入下一个右通道开始；或者，如果在接收左通道数据时发生上溢，则块会通过写入下一个左通道开始。

图 634. 上溢情况的处理



如果 SPI_IER 寄存器中的 OVRIE 位置 1，可产生中断。通过向 SPI_IFCR 寄存器的 OVRC 位写入 1 来将 OVR 位清零。

将块配置为 PCM 模式时，不会激活此重新对齐机制

注： *主模式或从模式下都可能发生上溢情况。在主模式下，发生上溢时，不会对 WS、CK 和 MCK 信号进行门控。*

50.9.14 帧错误检测

配置为从模式时，SPI/I2S 块会检测以下两种帧错误：

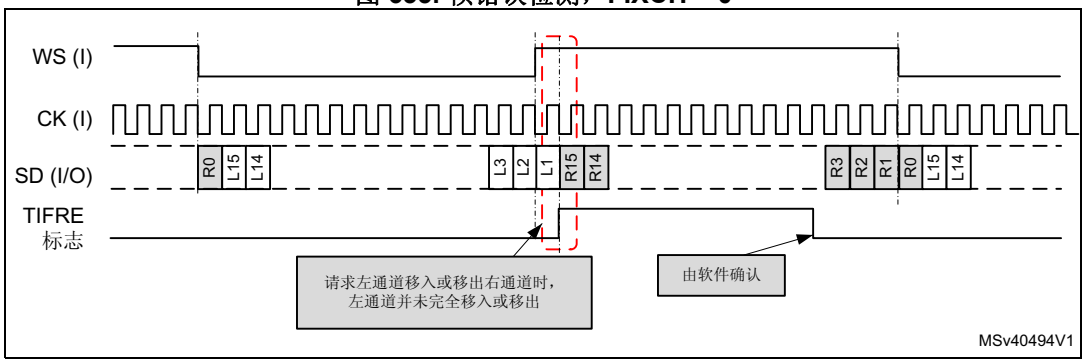
- 在先前数据的移入或移出未完成时接收到帧同步（早期帧错误）。通过 FIXCH = 0 选择此模式。
- 在意外位置发生帧同步。通过 FIXCH = 1 选择此模式。

在从模式下，如果由外部主器件提供的帧长度不是 32 位或 64 位，则用户必须将 FIXCH 设为 0。由于 SPI/I2S 会将每次传输均与 WS 同步，因此不存在偏离风险，但是在噪声环境下，如果 CK 信号出现毛刺，则采样可能会受到影响，而应用不会了解这一情况。

如果外部主器件提供的帧长度等于 32 位或 64 位，则用户可将 FIXCH 置 1，并相应调整 CHLEN。由于 SPI/I2S 会将每次传输均与 WS 同步，因此仍不存在偏离风险，但如果每个通道边界之间的位时钟数不同于 CHLEN，则帧错误标志 (TIFRE) 将置 1。

图 635 所示为帧错误检测示例。SPI/I2S 块处于从模式下，左通道的位时钟周期数不足以用于移入或移出数据。该图显示的是，正在进行的传输被中断，下一次传输已启动，以保持与 WS 信号对齐。

图 635. 帧错误检测，FIXCH = 0

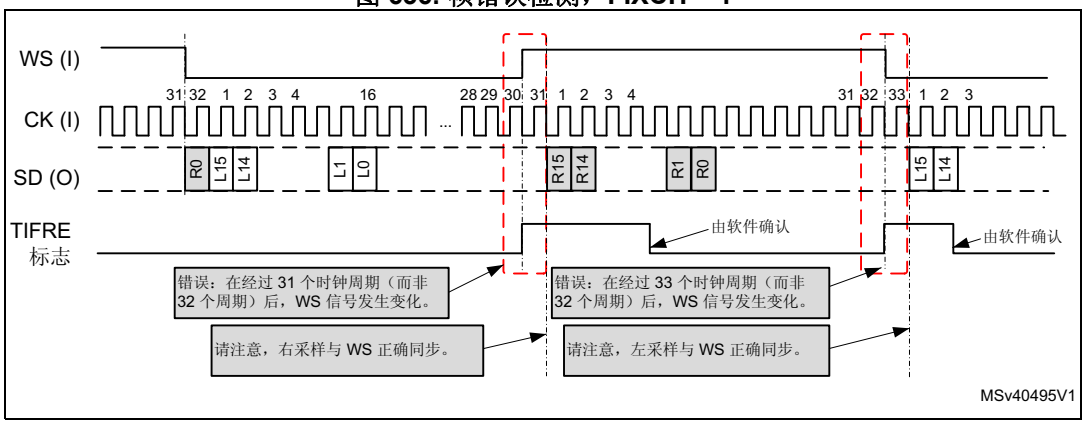


如果 TIFREIE 位置 1，则可生成中断。通过向 SPI_IFCR 寄存器的 TIFREC 位写入 1 来将帧错误标志 (TIFRE) 清零。

可以通过将位 FIXCH 置 1 来扩大帧错误标志的覆盖范围。该位置 1 时，SPI/I2S 可预计在从模式下的固定通道长度。这表示，预计通道长度可为 16 位或 32 位，具体取决于 CHLEN。如*图 636* 所示，在此模式下，SPI/I2S 块能够检测 WS 信号是否在预期的时间点（过早或过晚）发生变化。

注： *图 635* 和*图 636* 所示为适用于从器件发送模式的机制，但该机制还适用于从器件接收和从全双工模式。

图 636. 帧错误检测，FIXCH = 1



通常会因噪声环境干扰了 WS 或 CK 信号的良好接收而进行帧错误检测。

注： *如果在同一帧内出现上溢和早期帧，则 SPI/I2S 无法正常恢复。在这种情况下，用户必须禁止并重新使能 SPI/I2S。*

50.9.15 DMA 接口

I2S/PCM 模式与 SPI 功能共用相同的 DMA 请求线。TX 和 RX 路径有单独的 DMA 通道。每个 DMA 通道均可通过 SPI_CFG1 寄存器的 RXDMAEN 和 TXDMAEN 位使能。

在接收模式下，DMA 接口的工作如下：

1. 硬件评估 RxFIFO 级别。
2. 如果 RxFIFO 至少包含 FTHVL 个采样，则会生成 FTHVL DMA 请求。
   – FTHVL DMA 请求完成时，硬件返回到第 1 步，如此循环往复。
3. 如果 RxFIFO 包含的采样数少于 FTHVL 个，则不会生成 DMA 请求，硬件返回到第 1 步，如此循环往复。

在发送模式下，DMA 接口的工作如下：

1. 硬件评估 TxFIFO 级别。
2. 如果 TxFIFO 至少包含 FTHVL 个空位置，则会生成 FTHVL DMA 请求。
   – FTHVL DMA 请求完成时，硬件返回到第 1 步，如此循环往复。
3. 如果 TxFIFO 包含的空位置数少于 FTHVL 个，则不会生成 DMA 请求，硬件返回到第 1 步，如此循环往复。

50.9.16 编程示例

主模式 I2S Philips 标准，发送

该示例显示如何使用主时钟以 48 kHz 采样率对接口进行编程，以便支持主器件发送模式下的 Philips I2S 标准。假定 SPI/I2S 正在从电路的时钟控制器接收 61.44 MHz 的内核时钟 (i2s_clk)。

启动步骤

1. 使能总线接口时钟（pclk 或 hclk），必要时释放复位信号，以便能够对 SPI/I2S 块进行编程。
2. 确保 SPI/I2S 块正确接收内核频率（在本示例中为 61.44 MHz）。
3. 确保 SPE 置 0。
4. 对时钟发生器进行编程，以提供 MCK 时钟，以及获得恰好为 48 kHz 的帧同步率。因此，I2SDIV = 2，ODD = 1，且 MCKOE = 1。
5. 编程串行接口协议：CKPOL = 0，WSINV = 0，LSBFRST = 0，CHLEN = 1（每通道 32 位），DATLEN = 1（24 位），I2SSTD = 0（Philips 标准），I2SCFG = 2（主器件发送模式），I2SMOD = 1（针对 I2S/PCM 模式）。在进行后续步骤之前必须更新寄存器 SPI_I2SCFGR。
6. 通过对 FTHVL 设置所需值来调整 FIFO 阈值。例如，如果需要 2 个音频采样阈值，则 FTHVL = 1。
7. 清除所有状态标志寄存器。
8. 使能会生成中断的标志（例如 UDRIE）。请注意，TIFRE 在主模式下无效。
9. 如果数据传输使用 DMA，则要：
   - 编程 DMA 外设
   - 使用有效的音频采样初始化存储器缓冲区
   - 使能 DMA 通道
10. 如果要通过中断进行数据传输，则用户必须通过将 TXPIE 位置 1 来使能中断。
11. 将 SPE 置 1，该位置 1 后，可能会发生以下操作：
    - 如果使能中断生成，SPI/I2S 将生成中断请求，允许中断处理程序填充 TxFIFO。
    - 如果使能 DMA 传输 (TXDMAEN = 1)，SPI/I2S 将生成 DMA 请求来填充 TxFIFO
12. 最后，在使能串行接口之前，用户必须确保 TxFIFO 非空。这对于避免在使能接口时出现下溢条件至关重要。然后可通过将 CSTART 位置 1 来使能 SPI/I2S 块。CSTART 位位于 SPI_CR1 寄存器中。

在主模式下的停止步骤

1. 将 CSUSP 位置 1，以停止正在进行的传输
2. 检查 CSTART 位的值，直至其变为 0
3. 停止 DMA 外设、总线时钟...
4. 将 SPE 位置 0 以禁止 SPI/I2S 块

主模式 I2S MSB 对齐，全双工

该示例显示如何在不使用主时钟的情况下，以 48 kHz 采样率对接口进行编程，以便支持主全双工模式下的 I2S MSB 对齐协议。我们假定 SPI/I2S 正在从电路的时钟控制器接收 12.288 MHz 的内核时钟 (i2s_clk)。

步骤

1. 使能总线接口时钟（pclk 或 hclk），必要时释放复位信号，以便能够对 SPI/I2S 块进行编程。
2. 确保 SPI/I2S 块正确接收内核频率（在本示例中为 12.288 MHz）。
3. 确保 SPE 置 0。
4. 对时钟发生器进行编程，以提供 MCK 时钟，以及获得恰好为 48 kHz 的帧同步率。因此，I2SDIV = 2，ODD = 0，且 MCKOE = 0。
5. 编程串行接口协议：CKPOL = 0，WSINV = 0，LSBFRST = 0，CHLEN = 1（每通道 32 位），DATLEN = 1（24 位），I2SSTD = 1（MSB 对齐），I2SCFG = 5（主全双工模式），I2SMOD = 1（针对 I2S/PCM 模式）。在进行后续步骤之前必须更新寄存器 SPI_I2SCFGR。
6. 通过对 FTHVL 设置所需值来调整 FIFO 阈值。例如，如果需要 2 个音频采样阈值，则 FTHVL = 1。
7. 清除所有状态标志寄存器。
8. 使能会生成中断的标志（例如 UDRIE）。请注意，TIFRE 在主模式下无效。
9. 如果数据传输使用 DMA，则要：
   - 编程 DMA 外设：两个通道，一个通道用于 RX，另一个通道用于 TX
   - 针对 TX 路径使用有效的音频采样初始化存储器缓冲区
   - 使能 DMA 通道
   - 在 SPI/I2S 块中，通过将 TXDMAEN 和 RXDMAEN 位置 1 来使能 DMA。这些位置 1 后，SPI/I2S 将通过发送 DMA 请求开始填充 TxFIFO
10. 如果要通过中断进行数据传输，则用户必须通过将 TXPIE 和 RXPIE 位置 1 来使能中断。
11. 将 SPE 置 1，该位置 1 后，可能会发生以下操作：
    - 如果使能中断生成，SPI/I2S 将生成中断请求，允许中断处理程序填充 TxFIFO。
    - 如果使能 DMA 传输，SPI/I2S 将生成 DMA 请求来填充 TxFIFO
12. 最后，在使能串行接口之前，用户必须确保 TxFIFO 非空。这对于避免在使能接口时出现下溢条件至关重要。然后可通过将 CSTART 位置 1 来使能 SPI/I2S 块。CSTART 位位于 SPI_CR1 寄存器中。

有关停止步骤的详细信息，请参见*在主模式下的停止步骤*。

50.9.17 从模式 I2S Philips 标准，接收

该示例显示如何以 48 kHz 采样率对接口进行编程，以便支持从器件接收器模式下的 I2S Philips 标准协议。请注意，在从模式下，SPI/I2S 块不能控制接收采样的采样率。在本示例中，我们假定外部主器件具有通道长度为 24 位的 I2S 帧结构。因此，FIXCH 置 1 时，我们无法使用为帧错误检测提供的功能。

步骤

1. 使能总线接口时钟（pclk 或 hclk），必要时释放复位信号，以便能够对 SPI/I2S 块进行编程。
2. 确保 SPE 置 0。
3. 编程串行接口协议：CKPOL = 0，WSINV = 0，LSBFRST = 0，FIXCH = 0（因为通道长度不为 16 位和 32 位），DATLEN = 0（16 位），I2SSTD = 0（Philips 协议），I2SCFG = 1（从器件接收模式），I2SMOD = 1（针对 I2S 模式）。在进行后续步骤之前必须对寄存器 SPI_I2SCFGR 进行正确编程。
4. 通过对 FTHVL 设置所需值来调整 FIFO 阈值。例如，如果需要 2 个音频采样阈值，则 FTHVL = 1。
5. 清除所有状态标志寄存器。
6. 使能会生成中断的标志，例如 UDRIE 和 TIFRE。
7. 如果数据传输使用 DMA，则要：
   – 编程 DMA 外设：一个 RX 通道
   – 使能 DMA 通道
   – 在 SPI/I2S 块中，通过将 RXDMAEN 位置 1 来使能 DMA
8. 如果要通过中断进行数据传输，则用户必须通过将 RXPIE 位置 1 来使能中断。
9. Set SPE to 1
10. 最后，用户可将位 CSTART 置 1 来使能串行接口。SPI/I2S 将在下次出现外部主器件发送的左通道数据时，开始将数据存储到 RxFIFO 中。

在从模式下的停止步骤

1. 将 SPE 位置 0 以禁止 SPI/I2S 块
2. 停止 DMA 外设、总线时钟...

50.10 I2S 唤醒和中断

在 PCM/I2S 模式下，可根据*表 394* 中描述的事件生成中断 (**spi_it**) 或唤醒事件信号 (**spi_wkup**)。

中断事件可分别进行使能和禁止。

表 394. I2S 中断请求

中断事件	事件标志	使能控制位	事件清除方法	中断/唤醒是否已激活	
				spi_it	spi_wkup
达到 TxFIFO 阈值	TXP	TXPIE	TxFIFO 包含的空位置数少于 FTHVL 个时，TXP 标志清零	是	是
达到 RxFIFO 阈值	RXP	RXPIE	RxFIFO 包含的采样数少于 FTHVL 个时，RXP 标志清零		
上溢错误	OVR	OVRIE	通过向 OVRC 写入 1 来清零 OVR		
下溢错误	UDR	UDRIE	通过向 UDRC 写入 1 来清零 UDR		
帧错误	TIFRE	TIFREIE	通过向 TIFREC 写入 1 来清零 TIFRE		否

50.11 SPI/I2S 寄存器

50.11.1 SPI2S 控制寄存器 1 (SPI/I2S_CR1)

SPI/I2S control register 1

偏移地址：0x00

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	IOLOCK
															rs
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TCRCINI	RCRCINI	CRC33_17	SSI	HDDIR	CSUSP	CSTART	MASRX	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SPE
rw	rw	rw	rw	rw	w	rs	rw								rw

位 31:17 保留

位 16 **IOLOCK**：锁定相关 IO 的 AF 配置 (Locking the AF configuration of associated IOs)
此位由软件置 1，并通过硬件在下一次器件复位时清零
0：AF 配置未锁定
1：AF 配置已锁定
该位置 1 时，不能再修改 SPI_CFG2 寄存器内容。
应在 SPI 禁止时配置此位。当 SPE = 1 时，该位受写保护。

位 15 **TCRCI**：发送器 CRC 计算初始化模式控制 (CRC calculation initialization pattern control for transmitter)
0：应用全零模式
1：应用全 1 模式

位 14 **RCRCI**：接收器 CRC 计算初始化模式控制 (CRC calculation initialization pattern control for receiver)
0：应用全零模式
1：应用全 1 模式

位 13 **CRC33_17**：32 位 CRC 多项式配置 (32-bit CRC polynomial configuration)
0：不使用全尺寸（33 位或 17 位）CRC 多项式
1：使用全尺寸（33 位或 17 位）CRC 多项式

位 12 **SSI**：内部 SS 信号输入电平 (Internal SS signal input level)
仅当 SSM 位置 1 时，此位才有效。此位的值将作用到外设 SS 输入上，并忽略 SS 引脚的 I/O 值。

位 11 **HDDIR**：半双工模式下的 Rx/Tx 方向 (Rx/Tx direction at Half-duplex mode)
在半双工配置下，HDDIR 位确立了数据传输的 Rx/Tx 方向。在全双工或任何单工配置下，忽略该位。
0：SPI 为接收器
1：SPI 为发送器

位 10 **CSUSP**：主挂起请求 (Master SUSPend request)

该位读为零。

在主模式下，该位由软件置 1 时，CSTART 位将在当前帧结束时复位，SPI 通信将被挂起。用户必须检查 SUSP 标志以检查帧事务的结束。

在禁止 SPI 或进入低功耗模式之前，主模式通信必须挂起（通过该位或将 TXDR 保持为空来实现）。

位 9 **CSTART**：主传输开始 (Master transfer start)

此位由软件置 1，用于开始在主模式下进行 SPI 传输。传输结束 (EOT) 标志置 1 或接受 CSUSPend 请求时，该位由硬件清零。

0：主传输处于空闲状态

1：主传输正在进行，或通过自动挂起被临时挂起

在 SPI 模式下，仅当 SPE = 1 且 MASTER = 1 时，CSTART 才能置 1。
在 SPI 模式下，如果使能了主器件发送，则只有当发送 FIFO 中存在数据时，才能启动或继续通信。

在 I2S/PCM 模式下，CSTART 可在 SPE = 1 时置 1。

位 8 **MASRX**：接收模式下主器件自动挂起 (Master automatic SUSP in Receive mode)

该位由软件置 1 和清零，用于控制主器件接收器模式下的连续 SPI 传输以及自动管理，以避免出现上溢情况。

0：无论是否出现上溢情况，SPI 流/时钟生成都会连续。（数据将丢失）

1：在达到上溢情况之前，SPI 流在 RxFIFO 已满状态下暂时挂起。SUSP 标志将在 SPI 通信挂起时置 1。

SPI 通信挂起以防出现上溢情况时，可能会因内部同步延迟而导致同步输出下一帧的几个位。读取 RxFIFO 后，通信将恢复并继续进行后续的位传输，不受任何限制。

出于同样的原因，数据大小低于 8 位时，自动挂起并不十分可靠。在这种情况下，可通过结合在 MIDI 参数保持非零值时所应用的数据帧之间插入的延迟，来实现安全挂起；数据大小和交错 SPI 周期之和应始终至少产生长度为 8 个 SPI 时钟周期的间隔。

位 7:1 保留

位 0 **SPE**：串行外设使能 (Serial Peripheral Enable)

该位由软件置 1 和清零。

0：禁止串行外设。

1：使能串行外设。

当 SPE =1 时，将使能 SPI 数据传输，配置寄存器和 SPI_CR1 中的 IOLOCK 位受写保护，只有在 SPE = 0 时，才能更改。
当 SPI = 0 时，所有 SPI 操作被停止和禁止，内部状态机被复位，所有 FIFO 内容被清空，CRC 计算进行初始化，接收数据寄存器读为零。

MODF 错误标志有效时，SPE 不能置 1。

50.11.2 SPI 控制寄存器 2 (SPI_CR2)

SPI control register 2

偏移地址：0x04

复位值：0x0000 0000

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

位 31:16 **TSER[15:0]**：处理完存储在 TSIZE 中的先前数据量时就将数据传输扩展量重新载入到 TSIZE (Number of data transfer extension to be reload into TSIZE just when a previous number of data stored at TSIZE is transacted)

该寄存器只能在其内容被清空时由软件置 1。在完成 TSIZE 重载后，它由硬件清零。必须在 CTSIZE 计数器达到零之前，提前编程 TSER 值，否则将不会考虑重载，且通信将以正常 EOT 事件终止。

位 15:0 **TSIZE[15:0]**：当前传输的数据量 (Number of data at current transfer)

这些位通过软件进行更改。CSTART 位置 1 时，不应更改该值。

TSIZE 位是 0，且 CSTART 位置 1 时，会启动不终止的传输。使能 CRC 时，不能将 TSIZE 设为 0xFFFF 值。

50.11.3 SPI 配置寄存器 1 (SPI_CFG1)

SPI configuration register 1

偏移地址：0x08

复位值：0x0007 0007

使能 SPI 时，该寄存器的内容受写保护

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res	MBR[2:0]			Res	Res	Res	Res	Res	CRCEN	Res	CRCSIZE[4:0]				
	rw								rw		rw				
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TXDMAEN	RXDMAEN	Res	UDRDET[1:0]		UDRCFG[1:0]		FTHLV[3:0]				DSIZE[4:0]				
							rw				rw				

位 31 保留

位 30:28 **MBR [2:0]**：主波特率 (Master baud rate)

000：SPI 主时钟/2
001：SPI 主时钟/4
010：SPI 主时钟/8
011：SPI 主时钟/16
100：SPI 主时钟/32
101：SPI 主时钟/64
110：SPI 主时钟/128
111：SPI 主时钟/256

位 27:23 保留

位 22 **CRCEN**：硬件 CRC 计算使能 (Hardware CRC computation enable)

0：禁止 CRC 计算
1：使能 CRC 计算

位 21 保留

位 20:16 **CRCSIZE [4:0]**：要处理和比较的 CRC 帧的长度 (Length of CRC frame to be transacted and compared)

处理或比较 CRC 结果时，多项式计算会考虑最高有效位。多项式的长度不受该设置的影响。

00000：未使用
00001：未使用
00010：未使用
00011：4 位
00100：5 位
00101：6 位
00110：7 位
00111：8 位
.....
11101：30 位
11110：31 位
11111：32 位

该值必须等于 DSIZE 长度或是此长度的倍数

注： 对于数据大小被限制为 16 位的外设实例，CRCSIZE 位域的最高有效位将保留。

位 15 **TXDMAEN**：Tx DMA 数据流使能 (Tx DMA stream enable)

0：禁止 Tx DMA
1：使能 Tx DMA

位 14 **RXDMAEN**：Rx DMA 数据流使能 (Rx DMA stream enable)

0：禁止 Rx-DMA
1：使能 Rx-DMA

位 13 保留

位 12:11 **UDRDET [1:0]**：从器件发送器下溢条件检测 (Detection of underrun condition at slave transmitter)

00：数据帧开始时检测到下溢（无第 1 位保护）
01：在最后一个数据帧结束时检测到下溢
10：在有效 SS 信号开始时检测到下溢
11：保留

用户可在此处定义从器件接收器下溢条件检测的时间和方式

位 10:9 **UDRCFG [1:0]**：下溢条件时从器件发送器行为 (Behavior of slave transmitter at underrun condition)

00：从器件发送一个常数，该常数由用户在 SPI_UDRDR 寄存器中定义

01：从器件重复上一次从主器件接收到的数据帧

10：从器件重复其上一次发送的数据帧

11：保留

TxFIFO 为空时，发送移位寄存器保持 SPI_UDRDR 寄存器的值。该寄存器可被锁定从而发送由用户定义的值，或通过从主器件接收的最后处理的帧自动刷新，或者，通过最后存储在从 TxFIFO 中的数据（通过向 TXDR 进行写操作）来进行更新。

位 8:5 **FTHVL[3:0]**：FIFO 阈值级别 (FIFO threshold level)

定义单个数据包的数据帧数。数据包的大小不应超过 FIFO 空间的 1/2。

0000：1 个数据帧

0001：2 个数据帧

0010：3 个数据帧

0011：4 个数据帧

0100：5 个数据帧

0101：6 个数据帧

0110：7 个数据帧

0111：8 个数据帧

1000：9 个数据帧

1001：10 个数据帧

1010：11 个数据帧

1011：12 个数据帧

1100：13 个数据帧

1101：14 个数据帧

1110：15 个数据帧

1111：16 个数据帧

如果配置的数据包大小与数据寄存器访问并行位数对齐，则 SPI 接口的效率更高：

– 如果将 SPI 数据寄存器作为 16 位寄存器进行访问，且 DSIZE <= 8 位，则最好选择 FTHVL = 2、4、6 等值。

– 如果将 SPI 数据寄存器作为 32 位寄存器进行访问，且 DSIZE > 8 位，则最好选择 FTHVL = 2、4、6 等值；如果 DSIZE <= 8 位，则最好选择 FTHVL = 4、8、12 等值

位 4:0 **DSIZE [4:0]**：单个 SPI 数据帧的位数 (Number of bits in at single SPI data frame)

00000：未使用

00001：未使用

00010：未使用

00011：4 位

00100：5 位

00101：6 位

00110：7 位

00111：8 位

.....

11101：30 位

11110：31 位

11111：32 位

注： 对于数据大小被限制为 16 位的外设实例，DSIZE 位域的最高有效位将保留。

50.11.4 SPI 配置寄存器 2 (SPI_CFG2)

SPI configuration register 2

偏移地址：0x0C

复位值：0x0000 0000

SPI 使能或 SPI2S_CR1 寄存器中的 IOLOCK 位置 1 时，该寄存器的内容受写保护。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
AFCNTR	SSOM	SSOE	SSIOP	Res	SSM	CPOL	CPHA	LSBFRST	MASTER	SP[2:0]			COMM[1:0]		Res
rw	rw	rw	rw		rw	rw	rw	rw	rw	rw			rw		
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IOSWP	Res	Res	Res	Res	Res	Res	Res	MIDI [3:0]					MSSI [3:0]		
rw								rw	rw	rw	rw	rw	rw	rw	rw

位 31 **AFCNTR**：复用功能 GPIO 控制 (Alternate function GPIOs control)

仅当 SPE = 0 时，才考虑该位。

0：外设在禁止时不对 GPIO 进行控制

1：外设始终控制所有相关 GPIO

出于特定配置原因（例如 CRC 复位、CPHA 或 HDDIR 更改）而必须暂时禁止 SPI 主器件时，将该位置 1 可将配置为复用功能模式的相关输出强制为当前 SPI 配置所对应的状态，以防止出现毛刺。在从模式下，该位不应该使用，因为任何从器件发送器一旦禁止 SPI，就不得强制其 MISO 输出。

注： 该位也可用于 PCM 和 I2S 模式。

位 30 **SSOM**：主模式 SS 输出管理 (SS output management in master mode)

SSOE 使能时，该位用于主模式。它允许在两个连续数据传输之间配置 SS 输出。

0：SS 保持为有效电平，直到数据传输完成，随即会通过 EOT 标志变为无效状态

1：当 MIDI[3:0] > 1 时，SPI 数据帧与 SS 非有效脉冲交错

位 29 **SSOE**：SS 输出使能 (SS output enable)

该位仅在主模式下使用

0：禁止 SS 输出，SPI 可在多主配置下工作

1：使能 SS 输出。SPI 不能在多主环境下工作。它使用 SSOM、MIDI、MSSI 和 SSIOP 位设置在传输后将 SS 引脚强制为无效电平

位 28 **SSIOP**：SS 输入/输出极性 (SS input/output polarity)

0：SS 信号低电平有效

1：SS 信号高电平有效

位 27 保留

位 26 **SSM**：SS 信号输入的软件管理 (Software management of SS signal input)

0：SS 输入值由 SS PAD 决定

1：SS 输入值由 SSI 位决定

在主模式下使能 SS 输出模式 (SSOE = 1) 时，SS 信号输入必须由软件管理（SSM = 1，SSI = 1）。

位 25 **CPOL**：时钟极性 (Clock Polarity)

0：空闲时 SCK 信号为低电平

1：空闲时 SCK 信号为高电平

位 24 **CPHA**：时钟相位 (Clock phase)
0：从第一个时钟边沿开始采样数据
1：从第二个时钟边沿开始采样数据

位 23 **LSBFRST**：数据帧格式 (Data frame format)
0：先发送 MSB
1：先发送 LSB
注： 该位也可用于 PCM 和 I2S 模式。

位 22 **MASTER**：SPI 主模式 (SPI Master)
0：SPI 从模式
1：SPI 主模式

位 21:19 **SP[2:0]**：串行协议 (Serial Protocol)
000：SPI Motorola
001：SPI(TI)
其他：保留，不得使用

位 18:17 **COMM**：SPI 通信模式 (SPI Communication Mode)
00：全双工
01：单工发送器
10：单工接收器
11：半双工

位 16 保留

位 15 **IOSWP**：交换 MISO 和 MOSI 引脚的功能 (Swap functionality of MISO and MOSI pins)
0：不交换
1：交换 MOSI 和 MISO 引脚功能
该位置 1 时，MISO 和 MOSI 引脚复用功能互换。
原 MISO 引脚变为 MOSI 引脚，原 MOSI 引脚变为 MISO 引脚。
请注意，该位也可用于 PCM 和 I2S 模式。

位 14:8 保留

位 7:4 **MIDI [3:0]**：主模式数据间空闲 (Master Inter-Data Idleness)
指定主模式下在两个连续数据帧之间插入的最小时间延迟（以 SPI 时钟周期表示）。
0000：无延迟
0001：1 个时钟周期延迟
...
1111：15 个时钟周期延迟
TI 模式不支持该功能。

位 3:0 **MSSI [3:0]**：主模式 SS 空闲 (Master SS Idleness)
指定使能 SSOE 时在主模式下额外插入到 SS 有效边沿和第一个数据事务开始之间的额外延迟（用 SPI 时钟周期数表示）。
0000：无额外延迟
0001：添加了 1 个时钟周期
...
1111：添加了 15 个时钟周期
TI 模式不支持该功能。

50.11.5 SPI/I2S 中断使能寄存器 (SPI2S_IER)

SPI/I2S Interrupt Enable Register

偏移地址：0x10

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	TSERFIE	MODFIE	TIFREIE	CRCEIE	OVRIE	UDRIE	TXTFIE	EOTIE	DPXPIE	TXPIE	RXPIE
					rw	rw	rw	rw	rw	rw	rw	rw	rs	rs	rw

位 31:11 保留

位 10 **TSERFIE**：附加事务数重载中断使能 (Additional number of transactions reload interrupt enable)
0：禁止 TSERF 中断
1：使能 TSERF 中断

位 9 **MODFIE**：模式故障中断使能 (Mode Fault interrupt enable)
0：禁止 MODF 中断
1：使能 MODF 中断

位 8 **TIFREIE**：TIFRE 中断使能 (TIFRE interrupt enable)
0：禁止 TIFRE 中断
1：使能 TIFRE 中断

位 7 **CRCIE**：CRC 中断使能 (CRC Interrupt enable)
0：禁止 CRC 中断
1：使能 CRC 中断

位 6 **OVRIE**：OVR 中断使能 (OVR interrupt enable)
0：禁止 OVR 中断
1：使能 OVR 中断

位 5 **UDRIE**：UDR 中断使能 (UDR interrupt enable)
0：禁止 UDR 中断
1：使能 UDR 中断

位 4 **TXTFIE**：TXTFIE 中断使能 (TXTFIE interrupt enable)
0：禁止 TXTF 中断
1：使能 TXTF 中断

位 3 **EOTIE**：EOT、SUSP 和 TXC 中断使能 (EOT, SUSP and TXC interrupt enable)
0：禁止 EOT/SUSP/TXC 中断
1：使能 EOT/SUSP/TXC 中断

位 2 **DXPIE**：DXP 中断使能 (DXP interrupt enabled)
DXPIE 由软件置 1，并由 TXTF 标志置 1 事件清零。
0：禁止 DXP 中断
1：使能 DXP 中断

位 1 **TXPIE**：TXP 中断使能 (TXP interrupt enable)
TXPIE 由软件置 1，并由 TXTF 标志置 1 事件清零。
0：禁止 TXP 中断
1：使能 TXP 中断

位 0 **RXPIE**：RXP 中断使能 (RXP Interrupt Enable)
0：禁止 RXP 中断
1：使能 RXP 中断

50.11.6 SPI/I2S 状态寄存器 (SPI2S_SR)

SPI/I2S Status Register

偏移地址：0x14

复位值：0x0000 1002

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CTSIZE[15:0]															
r															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RXWNE	RXPLVL[1:0]		TXC	SUSP	TSERF	MODF	TIFRE	CRCE	OVR	UDR	TXTF	EOT	DPXP	TXP	RXP
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:16 **CTSIZE[15:0]**：当前 TSIZE 会话中剩余的数据帧数 (Number of data frames remaining in current TSIZE session)
当总线上有数据传输时候，该值并不十分可靠

位 15 **RXWNE**：RxFIFO 字非空 (RxFIFO Word Not Empty)
RxFIFO 包含多个 32 位数据
0：RxFIFO 中的数据帧小于 32 位
1：RxFIFO 中至少有一个 32 位数据

位 14:13 **RXPLVL[1:0]**：RxFIFO 封装级别 (RxFIFO Packing LeVeL)
RxFIFO 的最后一个 32 位字中已封装的帧的数量。当 RXWNE=0 时，会从 RxFIFO 中读取该帧数。
当帧大小小于或等于 8 位时
00：RxFIFO 中的帧数为 0 或 4 的倍数。
01：RXWNE = 0 时有 1 个帧
10：RXWNE = 0 时有 2 个帧
11：RXWNE = 0 时有 3 个帧
当帧大小小于或等于 16 位，但大于 8 位时，RXPLV[1:0] 可取值 0 或 1。
00：RxFIFO 中的帧数为 0 或 2 的倍数
01：RxFIFO 中的帧数为奇数。当 RXWNE = 0 时，有一个 16 位数据
其他值被禁止。
当帧大小大于 16 位时，这些位读为 00。
00：-
其他值被禁止。

位 12 **TXC**：TxFIFO 发送完成 (TxFIFO transmission complete)

该标志由硬件进行更改。

如果 TSIZE = 0，则只要 TxFIFO 变为空，TXC 就置 1，总线上无任何活动。

如果 TSIZE <>0，则无论 TxFIFO 占用情况如何，在传输结束时 TXC 都会置 1。

该标志置 1 时，主器件发送完成。对于主器件接收结束，该标志不可靠。当使能了 CRC 模式时，TXC 将仅在 CRC 发送后置 1。EOTIE 置 1 时，TXC 会生成中断。

0：当前数据事务仍在进行中、TxFIFO 中存在数据或者正在进行最后一个帧发送（包括 CRC）。

1：最后一个 TxFIFO 或 CRC 帧发送已完成

位 11 **SUSP**：挂起 (Suspend)

在主模式下，只要当前帧已完成，或者在 RxFIFO 已满后进入主器件自动挂起接收模式（SPI2S_CR1 寄存器的 MASRX 位置 1）时，在执行 CSUSP 后，SUSP 就由硬件置 1。

当 EOTIE 置 1 时，SUSP 会生成中断。

向 SPI2S_IFCR 的 SUSPC 位写入 1 可将该位清零

0：SPI 未挂起（主模式有效或其他模式）。

1：主模式挂起（当前帧已完成）

位 10 **TSERF**：已重载要处理的额外 SPI 数据量 (Additional number of SPI data to be transacted was reload)

通过向 SPI2S_IFCR 的 TSERFC 位写入 1 或通过对 TSER[15:0] (SPI_CR2) 寄存器进行写操作来将该位清零

0：不接收

1：接收了额外数据量，当前事务继续进行

位 9 **MODF**：模式故障 (Mode fault)

0：无模式故障

1：检测到模式故障

向 SPI2S_IFCR 的 MODFC 位写入 1 可将该位清零

位 8 **TIFRE**：TI 帧格式错误 (TI frame format error)

0：无 TI 帧错误

1：检测到 TI 帧错误

向 SPI2S_IFCR 的 TIFREC 位写入 1 可将该位清零

位 7 **CRCE**：CRC 错误 (CRC error)

0：无CRC错误

1：检测到 CRC 错误

向 SPI2S_IFCR 的 CRCEC 位写入 1 可将该位清零

位 6 **OVR**：溢出 (Overrun)

0：无上溢

1：检测到上溢

向 SPI2S_IFCR 的 OVRC 位写入 1 可将该位清零

位 5 **UDR**：从器件发送模式下溢 (Underrun at slave transmission mode)

0：无下溢

1：检测到下溢

向 SPI2S_IFCR 的 UDRC 位写入 1 可将该位清零

注：　UDR 标志仅适用于从模式

位 4 **TXTF**：发送传输已填充 (Transmission Transfer Filled)

0：TxFIFO 上传正在进行或者未启动

1：TxFIFO 上传已完成

某次传输中的所有数据包均由应用软件或 DMA 提交以用于发送时，即，TSIZE 数据量已推入到 TxFIFO 中时，TXTF 将由硬件置 1。

该位可由软件通过向 SPI2S_IFCR 的 TXTFC 位写入 1 来清零。

TXTFIE 位置 1 时 TXTF 标志会触发中断。

TXTF 设置可清除 TXIE 和 DPXIE 屏蔽，因此可在计算禁止 TXP 和 DXP 中断的时间时减轻应用软件的负担。

位 3 **EOT**：传输结束 (End Of Transfer)

完整传输完成后，即，基于 SPI 发送和/或接收 TSIZE 数据量时，EOT 由硬件置1。EOT 由软件通过向 SPI2S_IFCR 的 EOTC 位写入 1 来清零。

EOTIE 位置 1 时 EOT 标志会触发中断。

如果在 TXTF 标志置 1 以及 DXPIE 清零之前始终使用 DXP 标志，则可使用 EOT 一次性下载 RxFIFO 中包含的最后数据包。

0：传输正在进行或未启动

1：传输完成

在主模式下，EOT 事件会终止数据事务，并且可选择性处理 SS 输出。在主模式和从模式下，事件均会处理 CRC 事务。

位 2 **DXP**：双工数据包 (Duplex Packet)

0：TxFIFO 已满和/或 RxFIFO 为空

1：TxFIFO 具有用于写操作的空间，且 RxFIFO 至少包含一个用于读操作的数据包

TXP 和 RXP 标志均置 1 时，DXP 标志置 1。

位 1 **TXP**：可用的 Tx 数据包空间 (Tx-Packet space available)

0：TxFIFO 中没有足够的空间来放置下一个数据包

1：TxFIFO 具有足够的可用位置来容纳 1 个数据包

TXP 标志由硬件进行更改。如果使能 SPI，它将监视 TxFIFO 中当前可用的总体空间。在 TxFIFO 中存储完整的数据包后，必须对其进行检查。

位 0 **RXP**：可用的 Rx 数据包 (Rx-Packet available)

0：RxFIFO 为空或接收到的数据包不完整

1：RxFIFO 至少包含 1 个数据包

RXP 标志由硬件进行更改。如果使能 SPI，它将监视 RxFIFO 中当前可用的整体数据量。从 RxFIFO 中完整读取一个数据包后，必须对其进行检查。

50.11.7 SPI/I2S 中断/状态标志清零寄存器 (SPI2S_IFCR)

SPI/I2S Interrupt/Status Flags Clear Register

偏移地址：0x18

复位值：0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res	Res	Res	Res	SUSPC	TSERFC	MODFC	TIFREC	CRCEC	OVRC	UDRC	TXTFC	EOTC	Res	Res	Res
				w	w	w	w	w	w	w	w	w			

位 31:12 保留，必须保持复位值

位 11 **SUSPC**：SUSPend 标志清零 (SUSPend flag clear)
向该位写入 1 可将 SPI2S_SR 寄存器的 SUSP 标志清零

位 10 **TSERFC**：TSERFC 标志清零 (TSERFC flag clear)
向该位写入 1 可将 SPI2S_SR 寄存器的 TSERF 标志清零
注：*TSERF 还可通过对 TSER[15:0] (SPI_CR2) 寄存器进行写操作来复位*

位 9 **MODFC**：模式故障标志清零 (Mode Fault flag clear)
向该位写入 1 可将 SPI2S_SR 寄存器的 MODF 标志清零

位 8 **TIFREC**：TI 帧格式错误标志清零 (TI frame format error flag clear)
向该位写入 1 可将 SPI2S_SR 寄存器的 TIFRE 标志清零

位 7 **CRCEC**：CRC 错误标志清零 (CRC Error flag clear)
向该位写入 1 可将 SPI2S_SR 寄存器的 CRCE 标志清零

位 6 **OVRC**：上溢标志清零 (Overrun flag clear)
向该位写入 1 可将 SPI2S_SR 寄存器的 OVR 标志清零

位 5 **UDRC**：下溢标志清零 (Underrun flag clear)
向该位写入 1 可将 SPI2S_SR 寄存器的 UDR 标志清零

位 4 **TXTFC**：发送传输填充标志清零 (Transmission Transfer Filled flag clear)
向该位写入 1 可将 SPI2S_SR 寄存器的 TXTF 标志清零

位 3 **EOTC**：传输结束标志清零 (End Of Transfer flag clear)
向该位写入 1 可将 SPI2S_SR 寄存器的 EOT 标志清零

位 2:0 保留，必须保持复位值

50.11.8 SPI/I2S 发送数据寄存器 (SPI2S_TXDR)

SPI/I2S Transmit Data Register

偏移地址：0x20

复位值：0x0000 0000

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

位 31:0 **TXDR[31:0]**：发送数据寄存器 (Transmit data register)

该寄存器用作与 TxFIFO 的接口。对其进行写操作可访问 TxFIFO。

注： 数据始终右对齐。写入寄存器时将忽略未使用位，读取寄存器时会将未使用位读为 0。

注： DR 可按字节进行访问（8 位访问）：在这种情况下，通过单次访问只能写入一个数据字节。

按半字进行访问（16 位访问）：在这种情况下，通过单次访问可写入 2 个数据字节或 1 个半字数据。

逐字进行访问（32 位访问）：在这种情况下，通过单次访问可写入 4 个数据字节或者 2 个半字数据或字数据。

50.11.9 SPI/I2S 接收数据寄存器 (SPI2S_RXDR)

SPI/I2S Receive Data Register

偏移地址：0x30

复位值：0x0000 0000

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

位 31:0 **RXDR[31:0]**：接收数据寄存器 (Receive data register)

该寄存器用作与 RxFIFO 的接口。对其进行读操作时将访问 RxFIFO。

注： 数据始终右对齐。读取该寄存器时，未使用的位将读为 0。对该寄存器进行的写操作将被忽略。

注： DR 可按字节进行访问（8 位访问）：在这种情况下，通过单次访问只能读取一个数据字节。

按半字进行访问（16 位访问）：在这种情况下，通过单次访问可读取 2 个数据字节或 1 个半字数据。

逐字进行访问（32 位访问）：在这种情况下，通过单次访问可读取 4 个数据字节或者 2 个半字数据或字数据。

50.11.10 SPI 多项式寄存器 (SPI_CRCPOLY)

SPI Polynomial Register

偏移地址：0x40

复位值：0x0000 0107

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

位 31:0 **CRCPOLY[31:0]**：CRC 多项式寄存器 (CRC polynomial register)

此寄存器包含用于 CRC 计算的多项式。

默认的 9 位多项式设置 0x107 对应于 DSIZE 的默认 8 位设置。它与多项式字符串（字符串的最高有效位始终被隐藏）长度固定的某些其他 ST 产品使用的设置 0x07 兼容。

多项式的长度由存储在该寄存器中的值的最高有效位给出。必须将其设置为大于 DSIZE 的值。此外，将 DSIZE 配置为最大 32 位或 16 位大小并使能 CRC 时，还必须通过 CRCPOLY 寄存器将 CRC33_17 位置 1（以使多项式长度大于数据大小）。

对于数据大小被限制为 16 位的外设实例，位 31-16 保留。对这些地址执行 32 位访问时，不受限制。保留的位 31-16 始终读为零，并忽略所有写操作。

50.11.11 SPI 发送器 CRC 寄存器 (SPI_TXCRC)

SPI Transmitter CRC Register

偏移地址：0x44

复位值：0x0000 0000

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

位 31:0 **TXCRC[31:0]**：用于发送器的 CRC 寄存器 (CRC register for transmitter)

使能 CRC 计算后，TXCRC[31:0] 位将包含后续发送字节在计算后所得到的 CRC 值。向 SPI_CR1 的 CRCEN 位写入 1 或完整处理数据块时，将初始化 CRC 计算。CRC 通过 SPI_CRCPOLY 寄存器中编程的多项式连续计算。

计算时考虑的位数取决于 SPI_CRCPOLY 寄存器和 SPI_CFG1 寄存器的 CRCSIZE 位设置。

注： *进行通信时读取该寄存器会返回一个错误值。*

不适用于 I2S 模式。

对于数据大小被限制为 16 位的外设实例，位 31-16 保留。对这些地址执行 32 位访问时，不受限制。保留的位 31-16 始终读为零，并忽略所有写操作。

50.11.12 SPI 接收器 CRC 寄存器 (SPI_RXCRC)

SPI Receiver CRC Register

偏移地址：0x48

复位值：0x0000 0000

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

位 31:0 **RXCRC[31:0]**：用于接收器的 CRC 寄存器 (CRC register for receiver)

使能 CRC 计算后，RXCRC[31:0] 位将包含后续接收字节在计算后所得到的 CRC 值。向 SPI_CR1 的 CRCEN 位写入 1 或完整处理数据块时，将初始化 CRC 计算。CRC 通过 SPI_CRCPOLY 寄存器中编程的多项式连续计算。

计算时考虑的位数取决于 SPI_CRCPOLY 寄存器和 SPI_CFG1 寄存器的 CRCSIZE 位设置。

注： *进行通信时读取该寄存器会返回一个错误值。*

不适用于 I^2S 模式。

对于数据大小被限制为 16 位的外设实例，位 31-16 保留。对这些地址执行 32 位访问时，不受限制。保留的位 31-16 始终读为零，并忽略所有写操作。

50.11.13 SPI 下溢数据寄存器 (SPI_UDRDR)

SPI Underrun Data Register

偏移地址：0x4C

复位值：0x0000 0000

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

位 31:0 **UDRDR[31:0]**：从模式下溢时的数据 (Data at slave underrun condition)

仅在从模式下和下溢条件下考虑该寄存器。所考虑的位数取决于 SPI_CFG1 寄存器的 DSIZE 位设置。下溢条件的处理取决于 SPI_CFG1 寄存器的 UDRDET 和 UDRCFG 位是否置 1。

对于数据大小被限制为 16 位的外设实例，位 31-16 保留。对这些地址执行 32 位访问时，不受限制。保留的位 31-16 始终读为零，并忽略所有写操作。

50.11.14 SPI/I2S 配置寄存器 (SPI_I2SCGFR)

SPI/I2S configuration register

偏移地址：0x50

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res	Res	Res	Res	Res	Res	MCKOE (1)	ODD (1)	I2SDIV[7:0] (1)							
						rw	rw	rw							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res	DATFMT (1)	WSINV (1)	FIXCH (1)	CKPOL (1)	CHLEN (1)	DATLEN[1:0] (1)		PCMSYNC (1)	Res	I2SSTD[1:0] (1)		I2SCFG[2:0] (1)			I2SMOD (1)
	rw	rw	rw	rw	rw	rw		rw		rw		rw			rw

1. 这些位必须在 I2S 禁止时 (SPE = 0) 进行配置。
除了需要在 SPI 模式下置 0 的 I2SMOD 外，这些字段不用于 SPI 模式。

位 31:26 保留：由硬件强制为 0

位 25 **MCKOE**：主时钟输出使能 (Master clock output enable)
0：禁止主时钟输出
1：使能主时钟输出

位 24 **ODD**：预分频器的奇数因子 (Odd factor for the prescaler)
0：实际分频值为 = I2SDIV *2
1：实际分频值为 = (I2SDIV * 2)+1
有关详细信息，请参见*第 50.9.9 节：时钟发生器*。

位 23:16 **I2SDIV**：I^2S 线性预分频器 (I^2S linear prescaler)
当 ODD 也等于 1 时，I2SDIV 可以取除了值 1 之外的任何值。
有关详细信息，请参见*第 50.9.9 节：时钟发生器*。

位 15 保留：由硬件强制为 0

位 14 **DATFMT**：数据格式
0：SPI2S_RXDR 或 SPI2S_TXDR 内的数据右对齐。
1：SPI2S_RXDR 或 SPI2S_TXDR 内的数据左对齐。

位 13 **WSINV**：字选择反转 (Word select inversion)
该位用于反转 WS 信号的默认极性。
0：对于 I2S Philips 标准，WS 为低电平时传输左通道，WS 为高电平时传输右通道。
在 MSB 或 LSB 对齐模式下，WS 为高电平时传输左通道，WS 为低电平时传输右通道。
在 PCM 模式下，帧起始通过上升沿表示。
1：对于 I2S Philips 标准，WS 为高电平时传输左通道，WS 为低电平时传输右通道。
在 MSB 或 LSB 对齐模式下，WS 为低电平时传输左通道，WS 为高电平时传输右通道。
在 PCM 模式下，帧起始通过下降沿表示。

位 12 **FIXCH**：从模式下的固定通道长度 (Fixed channel length in slave)
0：从器件模式下的通道长度不为 16 位或 32 位（不取决于 CHLEN）
1：从器件模式下的通道长度应为 16 位或 32 位（取决于 CHLEN）

位 11 **CKPOL**：串行音频时钟极性 (Serial audio clock polarity)

0：由 SPI/I2S 生成的信号（即，SDO 和 WS）在 CK 的下降沿发生更改，由 SPI/I2S 接收的信号（即，SDI 和 WS）在 CK 的上升沿进行读取。

1：由 SPI/I2S 生成的信号（即，SDO 和 WS）在 CK 的上升沿发生更改，由 SPI/I2S 接收的信号（即，SDI 和 WS）在 CK 的下降沿进行读取。

位 10 **CHLEN**：通道长度（每个音频通道的位数）(Channel length (number of bits per audio channel))

0：16 位

1：32 位

只有在 DATLEN = 00 时，此位的写操作才有意义，否则无论写入何值，通道长度始终由硬件固定为 32 位。

位 9:8 **DATLEN**：要传输的数据长度 (Data length to be transferred)

00：16 位数据长度

01：24 位数据长度

10：32 位数据长度

11：不允许

位 7 **PCMSYNC**：PCM 帧同步 (PCM frame synchronization)

0：短帧同步

1：长帧同步

位 6 保留：由硬件强制为 0

位 5:4 **I2SSTD**：I^2S 标准选择 (I^2S standard selection)

00：I^2S Philips 标准

01：MSB 对齐标准（左对齐）

10：LSB 对齐标准（右对齐）

11：PCM 标准

有关 I^2S 标准的详细信息，请参见*第 50.9.5 节：支持的音频协议*。

位 3:1 **I2SCFG**：I2S 配置模式 (I2S configuration mode)

000：从模式 - 发送

001：从模式 - 接收

010：主模式 - 发送

011：主模式 - 接收

100：从模式 - 全双工

101：主模式 - 全双工

其他，未使用

位 0 **I2SMOD**：I2S 模式选择 (I2S mode selection)

0：选择 SPI 模式

1：选择 I2S/PCM 模式

50.12 SPI 寄存器映射和复位值

表 395. SPI 寄存器映射和复位值

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	**SPI2S_CR1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	IOLOCK	TCRCINI	RCRCINI	CRC33_17	SSI	HDDIR	CSUSP	CSTART	MASRX	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SPE
	Reset value																0	0	0	0	0	0	0	0	0								0
0x04	**SPI_CR2**	TSER[15:0]																TSIZE[15:0]															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x08	**SPI_CFG1**	Res.	MBR[2:0]			Res.	Res.	Res.	Res.	Res.	CRCEN	Res.	CRCSIZE[4:0]					TXDMAEN	RXDMAEN	Res.	UDRDET [1:0]		UDRCFG [1:0]		FTHLV[3:0]				DSIZE[4:0]				
	Reset value		0	0	0						0		1	1	1	1	1	0	0		0	0	0	0	0	0	0	0	0	0	1	1	1
0x0C	**SPI_CFG2**	AFCNTR	SSOM	SSOE	SSIOP	Res.	SSM	CPOL	CPHA	LSBFRST	MASTER	SP[2:0]			COMM [1:0]		Res.	IOSWP	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MIDI[3:0]				MSSI[3:0]				
	Reset value	0	0	0	0		0	0	0	0	0	0	0	0	0	0		0								0	0	0	0	0	0	0	0
0x10	**SPI2S_IER**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TSERFIE	MODFIE	TIFREIE	CRCEIE	OVRIE	UDRIE	TXTFIE	EOTIE	DPXPIE	TXPIE	RXPIE
	Reset value																						0	0	0	0	0	0	0	0	0	0	0
0x14	**SPI2S_SR**	CTSIZE[15:0]																RXWNE	RXPLVL [1:0]		TXC	SUSP	TSERF	MODF	TIFRE	CRCERR	OVR	UDR	TXTF	EOT	DPXP	TXP	RXP
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0
0x18	**SPI2S_IFCR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SUSPC	TSERFC	MODFC	TIFREC	CRCEC	OVRC	UDRC	TXTFC	EOTC	Res.	Res.	Res.
	Reset value																					0	0	0	0	0	0	0	0	0			
0x20	**SPI2S_TXDR**	TXDR[31:16]																TXDR[15:0]															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x24 0x2C	Reserved	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	Reset value																																
0x30	**SPI2S_RXDR**	RXDR[31:16]																RXDR[15:0]															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x34 0x3C	Reserved	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	Reset value																																
0x40	**SPI_CRCPOLY**	CRCPOLY[31:16](1)																CRCPOLY[15:0]															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	1	1
0x44	**SPI_TXCRC**	TXCRC[31:16](1)																TXCRC[15:0]															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x48	**SPI_RXCRC**	RXCRC[31:16](1)																RXCRC[15:0]															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x4C	**SPI_UDRDR**	UDRDR[31:16](1)																UDRDR[15:0]															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

表 395. SPI 寄存器映射和复位值（续）

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x50	**SPI_I2SCGFR**	Res.	Res.	Res.	Res.	Res.	Res.	MCKOE	ODD	I2SDIV[7:0]								Res.	Res.	WSINV	FIXCH	CKPOL	CHLEN	DATLEN		PCMSYNC	Res.	I2SSTD[1:0]		I2SCFG[2:0]			I2SMOD
	Reset value							0	0	0	0	0	0	0	0	0	0			0	0	0	0	0	0	0		0	0	0	0	0	0

1. 对于数据大小被限制为 16 位的外设实例，位 31-16 保留。对这些地址执行 32 位访问时，不受限制。保留的位 31-16 始终读为零，并忽略所有写操作。 *请参见寄存器边界地址表。*

51 串行音频接口 (SAI)

51.1 简介

SAI 接口（串行音频接口）灵活性高、配置多样，可支持多种音频协议。该接口适用于许多立体声或单声道应用。例如，它可配置为支持 I2S 标准、LSB 或 MSB 对齐、PCM/DSP、TDM 和 AC'97 等协议。将音频模块配置为发送器时，SAI 接口可提供 SPDIF 输出。

SAI 通过两个完全独立的音频子模块来实现这种灵活性和可配置性。每个模块都有自己的时钟发生器和 I/O 线控制器。

SAI 可以配置为主模式或配置为从模式。音频子模块既可作为接收器，又可作为发送器；既可与另一模块同步，又可以不同步。

SAI 可与其它 SAI 相连接来同步运行。

51.2 SAI 主要特性

- 具有两个独立的音频子模块，子模块既可作为接收器，也可作为发送器，并带有自身的 FIFO。
- 每个音频子模块集成多达 8 个字，每个字 32 位的 FIFO。
- 两个音频子模块间可以是同步或异步模式。
- 多个 SAI 之间可实现同步。
- 两个音频子模块的主/从配置相互独立。
- 当两个音频子模块都配置为以主模式工作时，每个子模块的时钟发生器采用独立的音频采样频率。
- 数据大小可配置：8 位、10 位、16 位、20 位、24 位或 32 位。
- 音频协议：I2S、LSB 或 MSB 对齐、PCM/DSP、TDM、AC'97。
- PDM 接口，支持多达 4 对麦克风。
- 如果需要，可提供 SPDIF 输出。
- 高达 16 个大小可配置的 Slot。
- 每帧的位数可配置。
- 帧同步有效电平可配置（偏移、位长、电平）。
- 可配置 Slot 中第一个有效位的位置。
- 支持 LSB 或 MSB 数据传输。
- 支持静音模式。
- 具有立体声/单声道音频帧功能。
- 通信时钟选通边沿可配置 (SCK)。
- 错误标志对应相应中断（分别使能时）。
  - 上溢和下溢检测
  - 从模式下的帧同步信号提前检测
  - 从模式下的帧同步信号滞后检测
  - 接收时编码解码器未针对 AC'97 模式就绪

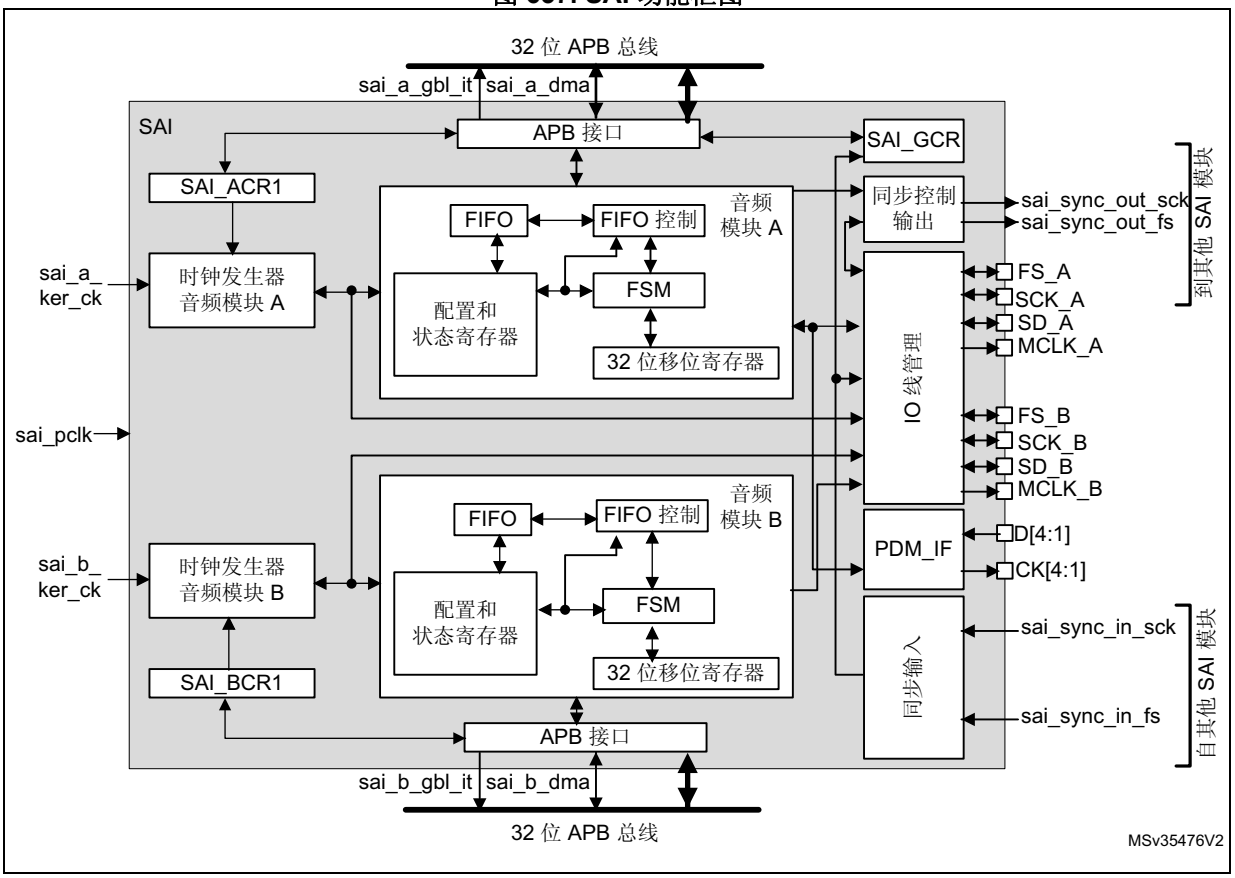
- 支持如下中断源（使能时）：
  - 错误
  - FIFO 请求
- DMA 接口有 2 个通道。

51.3 SAI 功能说明

51.3.1 SAI 框图

图 637 给出了 SAI 框图，*表 396* 和*表 397* 列出了 SAI 的内部信号和外部信号。

图 637. SAI 功能框图



SAI 主要由两个各自带有时钟发生器的音频子模块组成。每个音频模块集成一个 32 位移位寄存器，该寄存器由模块自身的功能状态机控制。数据存储和读取都是通过专用的 FIFO 来完成。FIFO 可通过 CPU 访问，也可通过 DMA 访问以减轻 CPU 的通信负担。每个音频模块是独立的。这两个音频子模块可彼此同步。

I/O 线控制器管理 SAI 中指定音频模块的 4 个专用引脚（SD、SCK、FS、MCLK）。如果将两个子模块声明为同步模块，则可以共用其中某些引脚，从而留出一些引脚用作通用 I/O。MCLK 引脚是否可用作输出引脚取决于实际应用和解码要求以及音频模块是否配置为主模式。

如果将一个 SAI 配置为与其它 SAI 同步运行，可以释放更多 I/O（引脚 SD_x 除外）。

可配置功能状态机来处理多种音频协议。一些寄存器用于设置所需协议（音频帧波形发生器）。

音频子模块在主模式或从模式下均可用作发送器或接收器。主模式意味着从 SAI 生成 SCK_x 位时钟和帧同步信号，而从模式则意味着 SCK_x 位时钟和帧同步信号来自另一外部或内部主器件。在特殊情况下，FS 信号方向与主模式或从模式定义不直接相关。在 AC'97 协议中，即使 SAI（链接控制器）设置为消耗 SCK 时钟，FS 信号也会是 SAI 输出（从模式下也是如此）。

注： *为方便阅读此部分，符号 SAI_x 指 SAI_A 或 SAI_B，其中“x”代表 SAI A 子模块或 SAI B 子模块。*

51.3.2 SAI 引脚和内部信号

表 396. SAI 内部输入/输出信号

内部信号名称	信号类型	说明
sai_a_gbl_it/ sai_b_gbl_it	输出	音频模块 A 和 B 全局中断。
sai_a_dma、 sai_b_dma	输入/输出	音频模块 A 和 B DMA 确认与请求。
sai_sync_out_sck、 sai_sync_out_fs	输出	与其他 SAI 模块交换的内部时钟和帧同步输出信号。
sai_sync_in_sck、 sai_sync_in_fs	输入	与其他 SAI 模块交换的内部时钟和帧同步输入信号。
sai_a_ker_ck/ sai_b_ker_ck	输入	音频模块 A/B 内核时钟。
sai_pclk	输入	APB 时钟。

表 397. SAI 输入/输出引脚

名称	信号类型	注释
SAI_SCK_A/B	输入/输出	音频模块 A/B 位时钟。
SAI_MCLK_A/B	输出	音频模块 A/B 主时钟。
SAI_SD_A/B	输入/输出	用于模块 A/B 的数据线。
SAI_FS_A/B	输入/输出	用于音频模块 A/B 的帧同步线。
SAI_CK[4:1]	输出	PDM 比特流时钟。
SAI_D[4:1]	输入	PDM 比特流数据。

51.3.3 SAI 的主要模式

SAI 的每个音频子模块均可通过所选音频模块的 SAI_xCR1 寄存器中的 MODE 位配置为主模式或从模式。

主模式

在主模式下，SAI 会向连接的外部器件提供时钟信号：

- 位时钟和帧同步分别在引脚 SCK_x 和 FS_x 上输出。
- 如果需要，SAI 也可以在 MCLK_x 引脚上生成主时钟。

SCK_x、FS_x 和 MCLK_x 均配置为输出。

从模式

SAI 从外部器件接收时钟信号。

- 如果将 SAI 子模块配置为异步模式，则 SCK_x 引脚和 FS_x 引脚将被配置为输入。
- 如果将 SAI 子模块配置为与另一个 SAI 接口或第二个音频子模块同步运行，则会释放相应的 SCK_x 引脚和 FS_x 引脚以用作通用 I/O。

在从模式下，不使用 MCLK_x 引脚，可将其分配给其它功能。

建议在使能主器件前先使能从器件。

配置和使能 SAI 模式

可通过相应音频模块的 SAI_xCR1 寄存器中的 MODE 位将每个音频子模块独立定义为发送器或接收器。为此，SAI_SD_x 引脚将分别被配置为输出或输入。

可使用两个不同的 MCLK 和 SCK 时钟频率配置同一 SAI 中的两个主音频模块。在这种情况下，必须将这两个音频模块配置为异步模式。

SAI 中的每个音频模块均通过 SAI_xCR1 寄存器中的 SAIEN 位使能。在从模式下，此位一经激活，发送器或接收器便会对时钟线、数据线和同步线上的活动敏感。

在主 TX 模式下，即使 FIFO 中没有数据，使能音频模块也会立即为外部从模块产生位时钟，但 FS 信号的产生受 FIFO 中是否存在数据的控制。FIFO 接收到要发送的第一个数据后，此数据将输出到外部从模块。如果 FIFO 中没有要发送的数据，则随后将在音频帧中传送值 0，并会产生一个下溢标志。

在从模式下，使能音频模块时和检测到 SOF 位时开始音频帧。

在从 TX 模式下，使能音频模块后的第一个帧上不可能出现下溢事件，因为此时的强制操作顺序如下：

1. 通过软件或 DMA 写入 SAI_xDR。
2. 等待至 FIFO 阈值 (FLH) 标志与 000b（FIFO 为空）不同。
3. 使能音频模块为从发送模式。

51.3.4 SAI 同步模式

同步模式存在两个同步级别，即音频子模块级别和 SAI 级别。

内部同步

音频子模块可以和同一 SAI 中的另一个音频子模块同步运行。在这种情况下，二者将共用位时钟和帧同步信号，以减少通信时占用的外部引脚数。配置为同步模式的音频模块将释放其 SCK_x、FS_x 和 MCLK_x 引脚以用作 GPIO，而配置为异步模式的音频模块将使用其 FS_x、SCK_x 和 MCLK_x I/O 引脚（如果该音频模块被配置为主模块）。

通常，同步模式下的音频模块可用于在全双工模式下配置 SAI。两个音频模块中的一个可配置为主模块，另一个为从模块；也可将两个均配置为从模块；一个模块为异步模块（SAI_xCR1 中的相应位 SYNCEN[1:0] 设置为 00），另一个为同步模块（SAI_xCR1 中的相应位 SYNCEN[1:0] 设置为 01）。

注： *由于存在内部重新同步阶段，因此 PCLK APB 频率必须大于比特率时钟频率的二倍。*

外部同步

音频子模块也可以配置为与其它 SAI 同步运行，具体方法如下：

1. 配置为其它 SAI 的同步源的 SAI 必须定义其音频子模块中的哪个子模块应向其它 SAI 提供 FS 和 SCK 信号。此操作通过编程 SYNCOUT[1:0] 位完成。
2. 接收同步信号的 SAI 必须通过为 SYNCIN[1:0] 位设置合适的值来选择哪个 SAI 将提供同步。之后，对于这两个 SAI 音频子模块的每个子模块，用户都必须通过 SYNCEN 位指定其是否与其它 SAI 同步运行。

注： *SYNCIN[1:0] 位和 SYNCOUT[1:0] 位位于 SAI_GCR 寄存器中，SYNCEN 位位于 SAI_xCR1 寄存器中。*

如果指定 SAI 中的两个音频子模块都需要与另一个 SAI 同步，则可以选择以下配置之一：

- 通过 SYNCEN[1:0] 位将每个音频子模块均配置为与另一个 SAI 模块同步。
- 通过 SYNCEN[1:0] 位将一个音频子模块配置为与另一个 SAI 模块同步。随后，通过 SYNCEN[1:0] 位将其它音频子模块配置为与第二个 SAI 音频子模块同步。

下表显示了如何根据所使用的 SAI 模块选择合适的同步信号。例如，SAI2 可通过将 SAI2 SYNCIN 设置为 0 来选择通过 SAI1 进行同步。如果 SAI1 要选择通过 SAI2 进行同步，则必须将 SAI1 SYNCIN 置 1。标注为“保留”的位不可使用。

表 398. 外部同步选择

模块实例	SYNCIN= 3	SYNCIN= 2	SYNCIN= 1	SYNCIN= 0
SAI1	SAI4 同步	SAI3 同步	SAI2 同步	保留
SAI2	SAI4 同步	SAI3 同步	保留	SAI1 同步
SAI3	SAI4 同步	保留	SAI2 同步	SAI1 同步
SAI4	保留	SAI3 同步	SAI2 同步	SAI1 同步

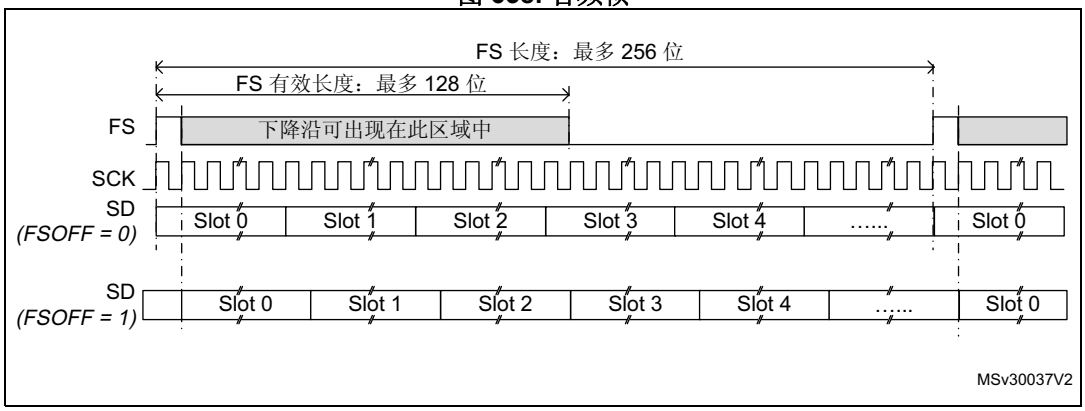
51.3.5 音频数据大小

通过配置 SAI_xCR1 寄存器中的 DS[2:0] 位，配置音频帧的数据大小。数据大小可以是 8 位、10 位、16 位、20 位、24 位或 32 位。在传输期间，将首先发送数据的 MSB 或 LSB，具体取决于 SAI_xCR1 寄存器中的 LSBFIRST 位的配置。

51.3.6 帧同步

FS 信号用作音频帧中的帧同步信号（SOF）。此信号的波形完全可配置，以在帧同步时，支持各种具有特殊规格的音频协议。这一可重新配置性通过寄存器 SAI_xFRCR 来实现。*图 638* 说明了这种灵活性。

图 638. 音频帧



在 AC’97 模式或 SPDIF 模式下（SAI_xCR1 寄存器中的位 PRTCFG[1:0] = 10 或 PRTCFG[1:0] = 01），帧同步信号的波形被强制配置为支持 AC’97 协议。SAI_xFRCR 寄存器值被忽略。

每个音频模块相互独立，因此均需要特定的配置。

帧长度

- 主模式

  将 SAI_xFRCR 寄存器中的 FRL[7:0] 位域置 1，可将音频帧的长度配置为最长 256 个位时钟周期。

  如果帧长度大于为该帧声明的 Slot 数，则要发送的剩余位将用 0 填充，或者 SD 线将释放为高阻态，具体取决于 SAI_xCR2 寄存器中的位 TRIS 的状态（见*FS 信号的作用*一节）。在接收模式下，剩余位被忽略。

  如果将位 NOMCK 清零，(FRL+1) 必须等于 2 的几次幂（从 8 到 256）以确保音频帧的每个位时钟周期都包含整数个 MCLK 脉冲。

  如果将位 NOMCK 置 1，则 (FRL+1) 字段可采用从 8 到 256 的任意值。
  请参见*第 51.3.8 节：SAI 时钟发生器*。

- 从模式

  音频帧的长度主要用于指定由外部主模块向从模块发送的每个音频帧的位时钟周期数。它主要用于从主模块中检测音频帧传输期间出现的提前或滞后帧同步信号。这种情况下会出现错误。有关详细信息，请参见*第 51.3.14 节：错误标志*。

  在从模式下，SAI_xFRCR 寄存器中的 FRL[7:0] 位的配置不受任何限制。

帧中的位数等于 FRL[7:0] + 1。

音频帧中要传输的最小位数为 8。

帧同步极性

SAI_xFRCR 寄存器中的 FSPOL 位用于设置 FS 引脚的有效极性，通过该极性来启动帧。SOF 信号对边沿敏感。

在从模式下，音频模块等待一个有效帧来启动发送或接收。SOF 信号与此信号同步。只有通信期间未检测到 SOF 信号并且 SOF 信号与预期 SOF 信号相同时，帧同步极性才有效（请参见*第 51.3.14 节：错误标志*）。

在主模式下，每次音频帧完成时均会发送帧同步信号，直至 SAI_xCR1 寄存器中的 SAIEN 位清零。如果前一个音频帧结束时 FIFO 中不存在数据，则将按照*第 51.3.14 节：错误标志*所述管理下溢条件，但音频通信流不会中断。

帧同步有效电平长度

SAI_xFRCR 寄存器的 FSALL[6:0] 位用于配置帧同步信号的有效电平的长度。该长度可设置为 1 到 128 个位时钟周期。

例如，有效长度在 I2S、LSB 或 MSB 对齐模式下为帧长的一半，在 PCM/DSP 或 TDM 模式下为 1 位。

帧同步偏移

基于应用中支持的音频协议（例如 I2S 标准协议和 MSB 对齐协议），可以在发送音频帧的最后一位或第一位时将帧同步信号置为有效。通过 SAI_xFRCR 寄存器中的 FSOFF 位选择两个配置之一。

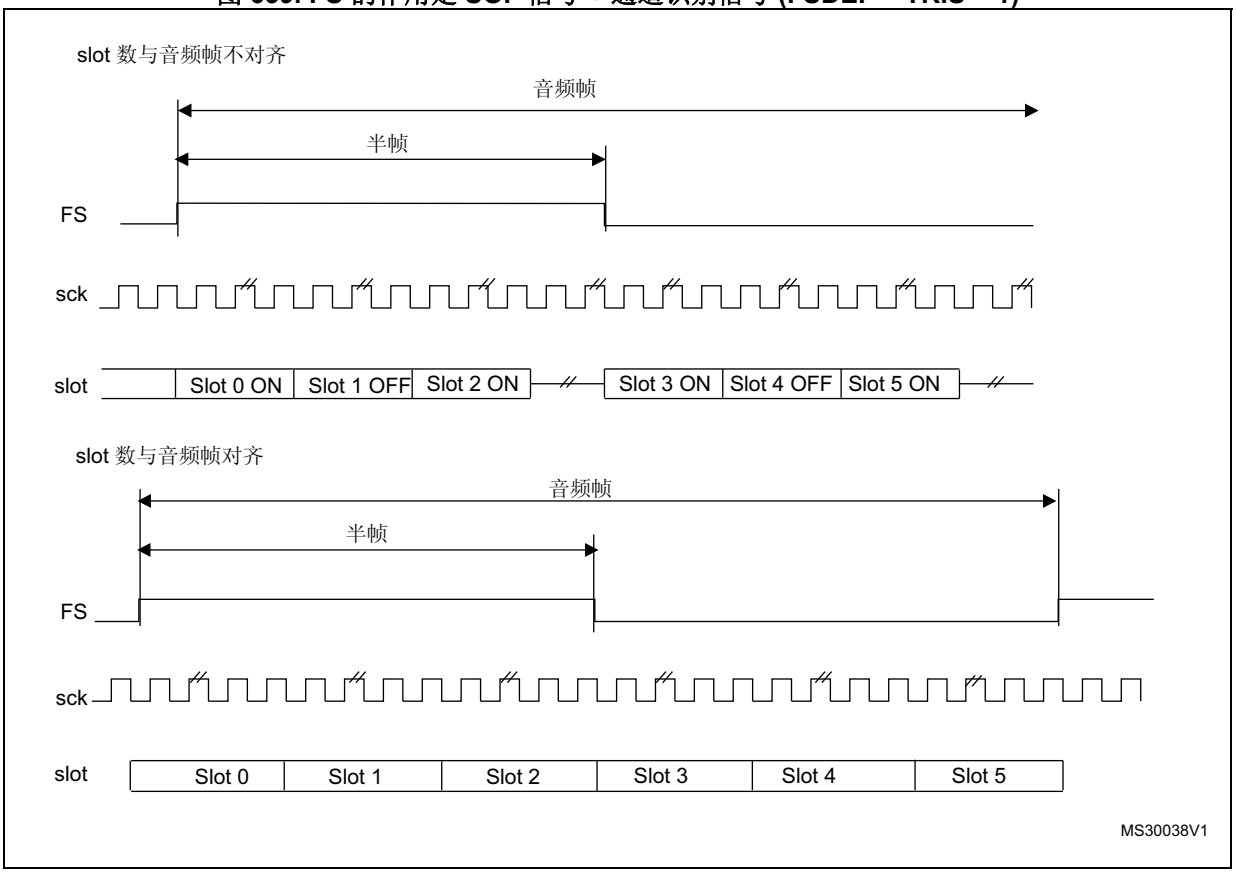
FS 信号的作用

FS 信号可以有不同的含义，具体取决于 FS 的功能。SAI_xFRCR 寄存器中的 FSDEF 位用于选择 FS 信号的含义。

- 0：SOF 信号，例如 PCM/DSP、TDM、AC’97 和音频协议，
- 1：音频帧内的 SOF 信号和通道识别信号，例如 I2S、MSB 或 LSB 对齐协议。

当 FS 信号被视为帧内的 SOF 信号和通道识别信号时，声明的 Slot 数必须是一半用于左通道，一半用于右通道。如果半个音频帧上的位时钟周期数大于某个通道的专用 Slot 数，若 TRIS = 0，则 SAI_xCR2 寄存器中的剩余位时钟周期将发送 0。否则若 TRIS = 1，SD 线将释放为高阻态。接收时，直到通道发生变化，才会考虑剩余位时钟。

图 639. FS 的作用是 SOF 信号 + 通道识别信号 (FSDEF = TRIS = 1)

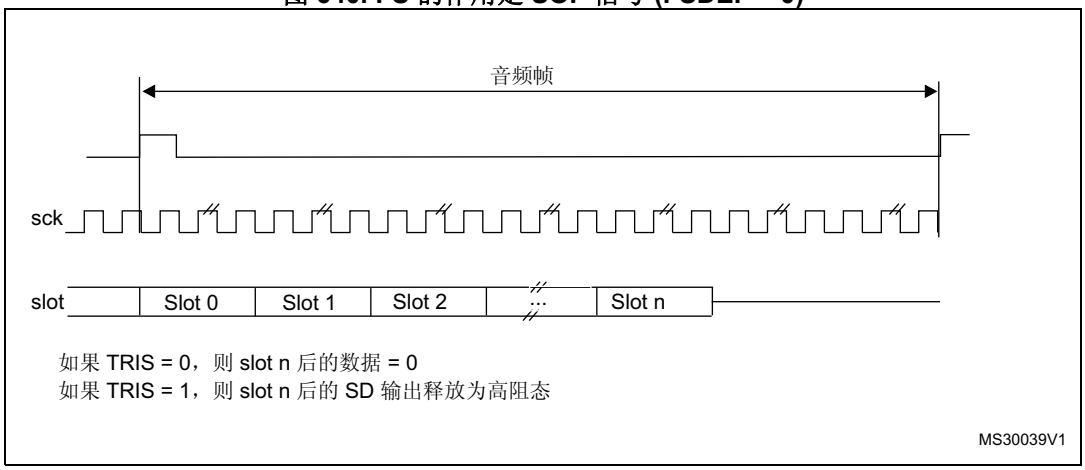


1. 帧长度应为偶数。

如果 SAI_xFRCR 中的 FSDEF 位保持清零状态，则 FS 信号等效于 SOF 信号，如果 SAI_xSLOTR 中的 NBSLOT[3:0] 位定义的 Slot 数乘以 SAI_xSLOTR 中的 SLOTSZ[1:0] 位配置的 Slot 位数所得的结果小于帧大小（SAI_xFRCR 寄存器中的 FRL[7:0] 位），则：

- 如果 SAI_xCR2 寄存器中的 TRIS = 0，则最后一个 Slot 后的剩余位将强制为 0，直至发送器的帧结束。
- 如果 TRIS = 1，则传输这些剩余位时，数据线将释放为高阻态。在接收模式下，这些位被丢弃。

图 640. FS 的作用是 SOF 信号 (FSDEF = 0)



在发送模式下将音频模块配置为获取 SD 线上的 SPDIF 输出时，将不使用 FS 信号。相应的 FS I/O 会被释放，留作他用。

51.3.7 Slot 配置

Slot 是音频帧中的基本元素。音频帧中的 Slot 数等于 NBSLOT[3:0] + 1。

每个音频帧的最大 Slot 数固定为 16。

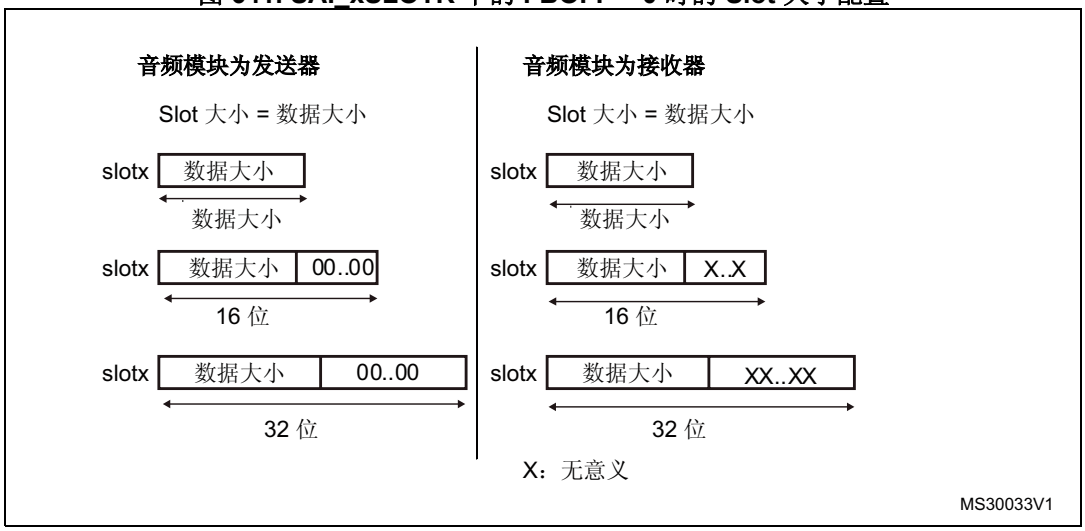
对于 AC'97 协议或 SPDIF 协议（位 PRTCFG[1:0] = 10 或 PRTCFG[1:0] = 01 时），Slot 数自动按照协议规范设置，NBSLOT[3:0] 的值被忽略。

通过设置 SAI_xSLOTR 寄存器的 SLOTEN[15:0] 位，可将各个 Slot 定义为有效 Slot 或无效 Slot。

传输一个无效 Slot 时，SD 数据线将强制为 0 或释放为高阻态，具体取决于 TRIS 位在发送模式下的配置（请参见*无效 Slot 上的输出数据线管理*一节）。在接收模式下，从该 Slot 结束后接收的数据被忽略。结果，不会有 FIFO 访问，也不会有与此无效 Slot 状态相关的 FIFO 读/写请求。

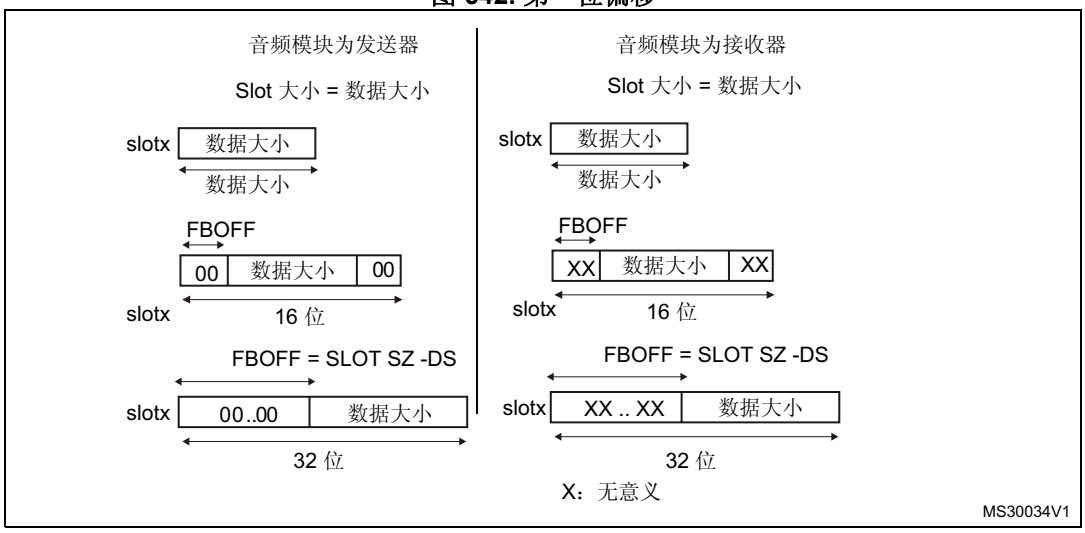
Slot 大小也可配置，如*图 641* 所示。通过将 SAI_xSLOTR 寄存器中的 SLOTSZ[1:0] 位置 1 来选择 Slot 大小。该大小适用于音频帧中的每个 Slot。

图 641. SAI_xSLOTR 中的 FBOFF = 0 时的 Slot 大小配置



可以选择 Slot 内要传输的第一个数据位的位置， 此偏移通过 SAI_xSLOTR 寄存器中的 FBOFF[4:0] 位配置。在发送模式下，将从 Slot 开始时注入 0 值，直至到达此偏移位置。接收时，偏移阶段中的位被忽略。此特性适用于 LSB 对齐协议（如果偏移等于 Slot 大小减去数据大小）。

图 642. 第一位偏移



要避免出现故障 SAI 行为，必须遵循以下条件：

FBOFF ≤ (SLOTSZ - DS)，

DS ≤ SLOTSZ，

NBSLOT x SLOTSZ ≤ FRL（帧长度）

SAI_xFRCR 寄存器中的 FSDEF 位置 1 时，Slot 数必须为偶数。

在 AC’97 和 SPDIF 协议（位 PRTCFG[1:0] = 10 或 PRTCFG[1:0] = 01）中，Slot 大小按照 *第 51.3.11 节：AC’97 链路控制器*中的定义自动设置。

51.3.8 SAI 时钟发生器

每个音频子模块都有自己的时钟发生器，旨在使这两个模块完全独立。这两个时钟发生器的功能没有任何区别。

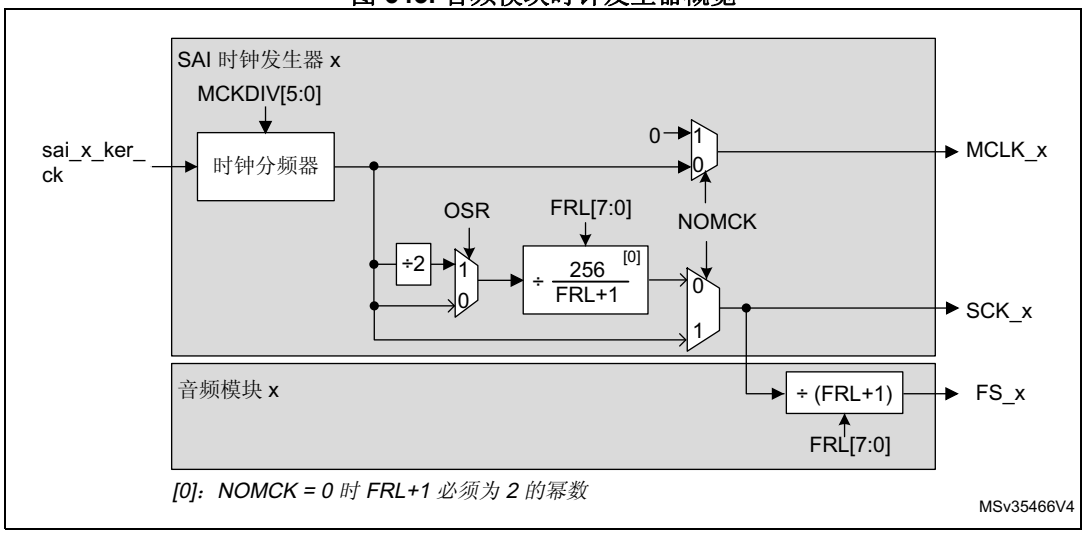
将音频模块配置为主模式时，时钟发生器将提供位时钟 (SCK_x) 以及用于外部解码器的主时钟 (MCLK_x)。帧同步 (FS_x) 还可源自时钟发生器提供的信号。SAI 时钟发生器的时钟源 (sai_x_ker_ck) 由芯片时钟控制器 (RCC) 提供。

将音频模块定义为从模式时，时钟发生器将关闭。NOMCK、MCKDIV 和 OSR 位的值将被忽略。此外，MCLK_x I/O 引脚也会被释放，可用作通用 I/O。

(SCK_x) 的位时钟选通边沿可通过 SAI_xCR1 寄存器中的 CKSTR 位配置。此位对主模式和从模式均有效。

图 643 给出了音频模块时钟发生器的架构。

图 643. 音频模块时钟发生器概览



SAI_xCR1 寄存器的 NOMCK 位用于定义是否生成主时钟。

将 SAI 用于主模式下时，时钟发生器配置有所不同，具体取决于是否需要提供主时钟 (MCLK_x)。

如果 NOMCK 置 1，则不生成主时钟，用户可以更灵活地选择帧长度和帧同步频率。此外，如果该引脚被配置为 GPIO 外设中的 SAI 引脚，则 MCLK_x 信号将驱动为低电平。MCKDIV 仍可用于将 SCK_x 时钟调整为所需频率。

如果 NOMCK 置 0，则会生成主时钟，并且可将主时钟用作外部解码器的参考时钟。在这种情况下，帧同步与主时钟之间的频率比为固定值 512 或 256，帧长度必须为 2 的幂数。下面进行了详细说明。

存在 MCLK (NOMCK = 0) 时的时钟发生器编程

此时，MCLK_x 频率将为：

$$F_{MCLK_x} = 256 \times F_{FS_x}$$ （当 OSR=0）

$$F_{MCLK_x} = 512 \times F_{FS_x}$$ （当 OSR=1）

MCKDIV 不为 0 时，MCLK_x 频率由如下公式计算得出：

$$F_{MCLK_x} = \frac{F_{sai_x_ker_ck}}{MCKDIV}$$

帧同步频率由如下公式计算得出：

$$F_{FS_x} = \frac{F_{sia_x_ker_ck}}{MCKDIV \times (OSR + 1) \times 256}$$

位时钟 (SCK_x) 的频率由以下表达式计算得出：

$$F_{SCK_x} = \frac{F_{MCLK_x} \times (FRL + 1)}{(OSR + 1) \times 256}$$

注： *如果 NOMCK = 0，则 (FRL+1) 必须为 2 的幂数。此外，(FRL+1) 必须介于 8 到 256 之间（请参见 FS 信号的作用一节）。*

MCKDIV 分频比为奇数时，MCLK 的占空比不会为 50%。如果 MCKDIV 为奇数、OSR 等于 0 或 (FRL+1) = 2^8，位时钟信号 (SCK_x) 的占空比也可以不是 50%。

建议将 MCKDIV 配置为偶数或较大的值（大于 10）。

请注意，MCKDIV = 0 时得到的结果与 MCKDIV = 1 时得到的结果相同。

无 MCLK (NOMCK = 1) 时的时钟发生器编程

MCKDIV 不为 0 时，SCK_x 频率如下公式计算得出：

$$F_{SCK_x} = \frac{F_{sai_x_ker_ck}}{MCKDIV}$$

帧同步 (FS_x) 的频率由如下公式计算得出：

$$F_{FS_x} = \frac{F_{sai_x_ker_ck}}{(FRL + 1) \times MCKDIV}$$

注： *NOMCK = 0 时，(FRL+1) 可采用 8 到 256 之间的任意值。*

请注意，MCKDIV = 0 时得到的结果与 MCKDIV = 1 时得到的结果相同。

时钟发生器编程示例

表 399 列出了一些针对 48 kHz、96 kHz 和 192 kHz 频率的编程示例。

表 399. 时钟发生器编程示例

输入 sai_x_ker_ck 时钟频率	MCLK	F_{MCLK}/F_{FS}	FRL (1)	OSR	NOMCK	MCKDIV[5:0]	音频采样频率 (F_{FS})
98.304 MHz	是	512	2^N-1	1	0	0 或 1	192 kHz
		512	2^N-1	1	0	2	96 kHz
		512	2^N-1	1	0	4	48 kHz
		256	2^N-1	0	0	2	192 kHz
		256	2^N-1	0	0	4	96 kHz
		256	2^N-1	0	0	8	48 kHz
	否	-	63	-	1	8	192 kHz
		-	63	-	1	16	96 kHz
		-	63	-	1	32	48 kHz

1. N 表示介于 3 到 8 之间的整数值。

51.3.9 内部 FIFO

SAI 中的每个音频模块都有自己的 FIFO。根据模块是定义为发送器还是接收器，可相应的读取或写入其 FIFO。因此只存在一个 FIFO 请求与 SAI_xSR 寄存器中的 FREQ 位相关。

如果 SAI_xIM 寄存器中的 FREQIE 位使能，将产生中断。这取决于：

- FIFO 阈值设置（SAI_xCR2 中的 FLVL 位）
- 通信方向（发送器或接收器）。请参见*在发送模式下产生中断*一节和*在接收模式下产生中断*一节。

在发送模式下产生中断

根据发送模式下的 FIFO 配置产生中断：

- 当 SAI_xCR2 寄存器中的 FIFO 阈值位配置为 FIFO 为空时（FTH[2:0] 置为 000b），如果寄存器中没有数据（SAI_xSR 中的 FLVL[2:0] 位小于 001b），将产生中断（SAI_xSR 寄存器中的 FREQ 位由硬件置 001）。当 FIFO 不再为空时（SAI_xSR 中的 FLVL[2:0] 位不是 000b），即 FIFO 中存储一个或多个数据时，此中断（SAI_xSR 寄存器中的 FREQ 位）由硬件清零。
- 当 SAI_xCR2 寄存器中的 FIFO 阈值位配置为 FIFO 四分之一满时（FTH[2:0] 置为 001b），如果不到四分之一的 FIFO 包含数据（SAI_xSR 中的 FLVL[2:0] 位小于 010b），将产生中断（SAI_xSR 寄存器中的 FREQ 位由硬件置 1）。当至少四分之一的 FIFO 包含数据时（SAI_xSR 中的 FLVL[2:0] 位大于等于 010b），此中断（SAI_xSR 寄存器中的 FREQ 位）由硬件清零。
- 当 SAI_xCR2 寄存器中的 FIFO 阈值位配置为 FIFO 半满时（FTH[2:0] 置为 010b），如果不到一半的 FIFO 包含数据（SAI_xSR 中的 FLVL[2:0] 位小于 011b），将产生中断（SAI_xSR 寄存器中的 FREQ 位由硬件置 1）。当至少一半的 FIFO 包含数据时（SAI_xSR 中的 FLVL[2:0] 位大于或等于 011b），此中断（SAI_xSR 寄存器中的 FREQ 位）由硬件清零。

- 当 SAI_xCR2 寄存器中的 FIFO 阈值位配置为 FIFO 四分之三满时（FTH[2:0] 置为 011b），如果不到四分之三的 FIFO 包含数据（SAI_xSR 中的 FLVL[2:0] 位小于 100b），将产生中断（SAI_xSR 寄存器中的 FREQ 位由硬件置 1）。当至少四分之三的 FIFO 包含数据时（SAI_xSR 中的 FLVL[2:0] 位大于或等于 100b），此中断（SAI_xSR 寄存器中的 FREQ 位）由硬件清零。
- 当 SAI_xCR2 寄存器中的 FIFO 阈值位配置为 FIFO 为满时（FTH[2:0] 置为 100b），如果 FIFO 不满（SAI_xSR 中的 FLVL[2:0] 位小于 101b），将产生中断（SAI_xSR 寄存器中的 FREQ 位由硬件置 1）。当 FIFO 已满时（SAI_xSR 中的 FLVL[2:0] 位等于值 101b），此中断（SAI_xSR 寄存器中的 FREQ 位）由硬件清零。

在接收模式下产生中断

根据接收模式下的 FIFO 配置产生中断：

- 当 SAI_xCR2 寄存器中的 FIFO 阈值位配置为 FIFO 为空时（FTH[2:0] 置为 000b），如果 SAI_xDR 寄存器中至少有一个数据（SAI_xSR 中的 FLVL[2:0] 位大于或等于 001b），将产生中断（SAI_xSR 寄存器中的 FREQ 位由硬件置 001）。当 FIFO 变为空时（SAI_xSR 中的 FLVL[2:0] 位等于 000b），即 FIFO 中未存储数据时，此中断（SAI_xSR 寄存器中的 FREQ 位）由硬件清零。
- 当 SAI_xCR2 寄存器中的 FIFO 阈值位配置为 FIFO 四分之一满时（FTH[2:0] 置为 001b），如果至少有四分之一的 FIFO 数据单元可用（SAI_xSR 中的 FLVL[2:0] 位大于或等于 010b），将产生中断（SAI_xSR 寄存器中的 FREQ 位由硬件置 1）。当不到四分之一的 FIFO 数据单元可用时（SAI_xSR 中的 FLVL[2:0] 位小于 010b），此中断（SAI_xSR 寄存器中的 FREQ 位）由硬件清零。
- 当 SAI_xCR2 寄存器中的 FIFO 阈值位配置为 FIFO 半满时（FTH[2:0] 置为 010b），如果至少有一半的 FIFO 数据单元可用（SAI_xSR 中的 FLVL[2:0] 位大于或等于 011b），将产生中断（SAI_xSR 寄存器中的 FREQ 位由硬件置 1）。当不到一半的 FIFO 数据单元可用时（SAI_xSR 中的 FLVL[2:0] 位小于 011b），此中断（SAI_xSR 寄存器中的 FREQ 位）由硬件清零。
- 当 SAI_xCR2 寄存器中的 FIFO 阈值位配置为 FIFO 四分之三满时（FTH[2:0] 置为 011b），如果至少有四分之三的 FIFO 数据单元可用（SAI_xSR 中的 FLVL[2:0] 位大于或等于 100b），将产生中断（SAI_xSR 寄存器中的 FREQ 位由硬件置 1）。当不到四分之三的 FIFO 数据单元可用时（SAI_xSR 中的 FLVL[2:0] 位小于 100b），此中断（SAI_xSR 寄存器中的 FREQ 位）由硬件清零。
- 当 SAI_xCR2 寄存器中的 FIFO 阈值位配置为 FIFO 满时（FTH[2:0] 置为 100b），如果 FIFO 已满（SAI_xSR 中的 FLVL[2:0] 位等于 101b），将产生中断（SAI_xSR 寄存器中的 FREQ 位由硬件置 1）。当 FIFO 不满时（SAI_xSR 中的 FLVL[2:0] 位小于 101b），此中断（SAI_xSR 寄存器中的 FREQ 位）由硬件清零。

与中断的产生类似，如果 SAI_xCR1 寄存器中的 DMAEN 位置 1，则 SAI 可使用 DMA。FREQ 位的有效机制与上述 FREQIE 的中断产生机制相同。

每个 FIFO 均是一个 8 字 FIFO。无论访问大小为何，每次对 FIFO 进行读写时，均针对 1 个字的 FIFO 单元进行操作。每个 FIFO 字包含一个音频 Slot。每次访问 SAI_xDR 寄存器后，FIFO 指针递增一个字。

数据应以右对齐方式写入 SAI_xDR。

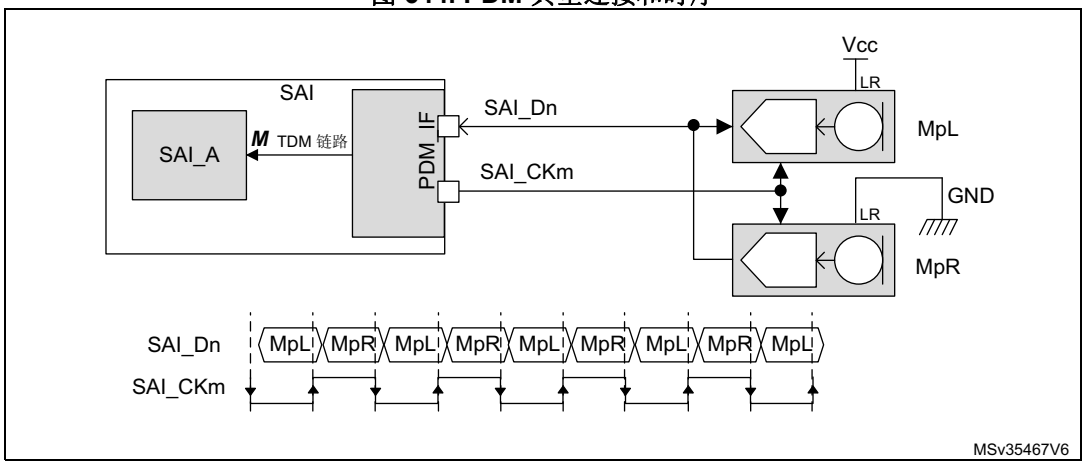
接收到的数据将以右对齐方式存储到 SAI_xDR。

将 SAI_xCR2 寄存器中的 FFLUSH 位置 1 禁止 SAI 后，可重新初始化 FIFO 指针。如果 FFLUSH 在 SAI 使能情况下置 1，则 FIFO 中的数据将自动清除。

51.3.10 PDM 接口

PDM（脉冲密度调制）接口旨在支持数字麦克风， 最多可并联 4 对数字麦克风。*图 644* 所示为一对数字麦克风通过 PDM 接口实现的典型连接。两个麦克风共用同一比特流时钟和数据线。借助配置引脚 (LR)，其中一个麦克风可在 SAI_CK [m] 上升沿提供有效数据，而另一个麦克风则在 SAI_CK [m] 下降沿提供有效数据（m 表示时钟线的数量）。

图 644. PDM 典型连接和时序



1. **n** 表示数据线的数量，**p** 表示麦克风对的数量。

PDM 功能旨在与配置为 TDM 主模式的 SAI_A 子模块配合使用。它不能与 SAI_B 子模块搭配使用。PDM 接口使用由 SAI_A 的 TDM 接口提供的时序信号，并对其进行调整来生成时钟 (SAI_CK[m])。

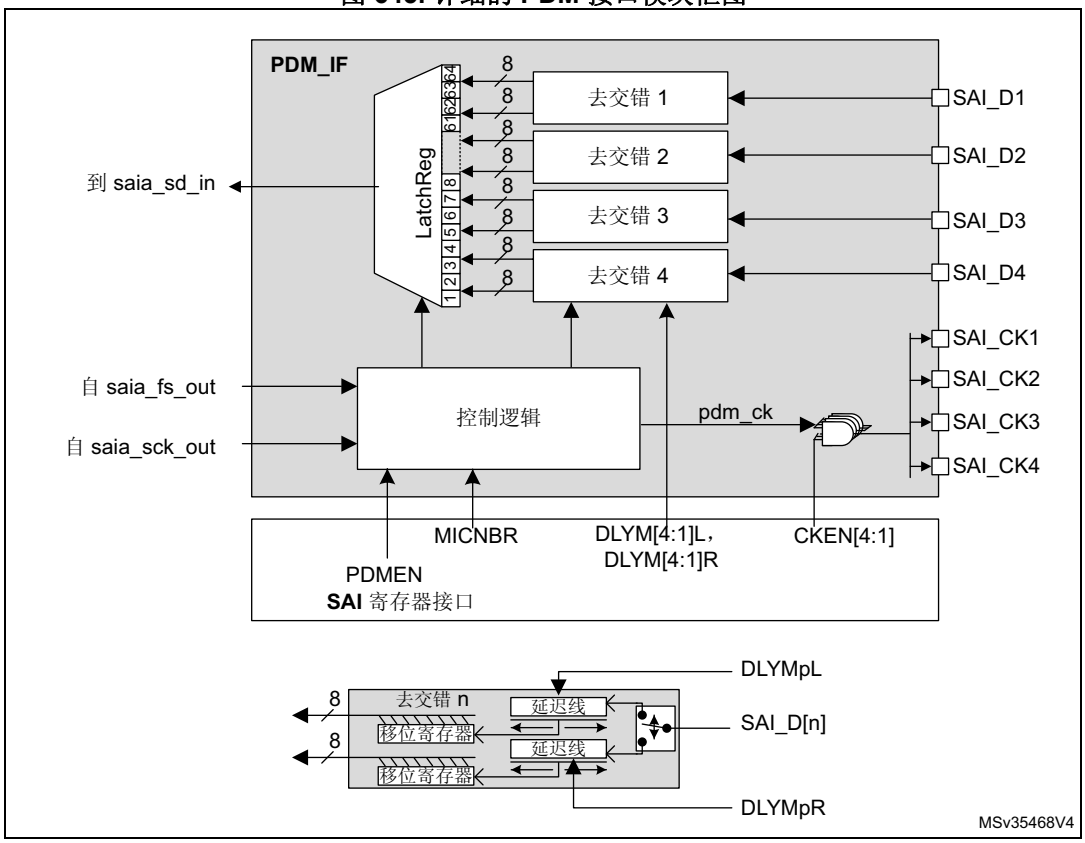
PDM 中的数据处理序列如下：

1. PDM 接口通过从 SAI_A 的 TDM 接口接收的位时钟构建比特流时钟。
2. 接收自麦克风的比特流数据 (SAI_D[n]) 进行去交错，然后经过 7 位延迟线，以根据比特流时钟的精度对每个麦克风的延迟进行微调。
3. 移位寄存器将每个串行比特流转换为字节。
4. 最后通过 TDM 接口的串行数据线将得到的字节移出到 SAI_A。

下面的*图 645* 给出了 PDM 接口框图，其中包括去交错的详细视图。

注： *PDM 接口未嵌入抽取滤波器，因此无法根据比特流构建 PCM 音频采样， 需要借助应用软件来执行这一操作。*

图 645. 详细的 PDM 接口模块框图



1. **n** 表示数据线的数量，**p** 表示麦克风对的数量。

可通过 SAI_PDMCR 寄存器中的 PDMEN 位使能 PDM 接口。不过，必须在使能 SAI_A 模块之前使能 PDM 接口。

为减少存储器占用量，用户可根据应用需求选择麦克风的数量。这可以通过 MICNBR[1:0] 位来实现。用户可以选择 2、4、6 或 8 个麦克风。例如，如果应用需要使用 3 个麦克风，则用户必须选择 4 个麦克风。

使能 PDM 接口

要使能 PDM 接口，需遵循以下序列：

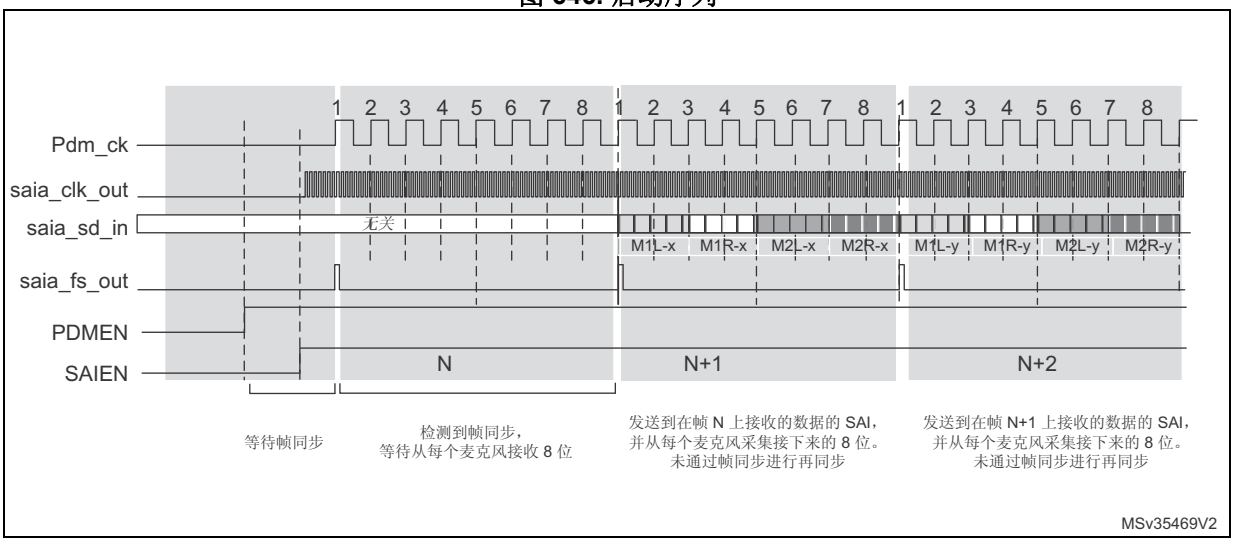
1. 将 SAI_A 配置为 TDM 主模式（请参见*表 400*）。
2. 按以下步骤配置 PDM 接口：
   a) 通过 MICNBR 定义数字麦克风的数量。
   b) 通过将 CKEN 的相应位置 1 来使能应用所需的比特流时钟。
3. 通过 PDMEN 位使能 PDM 接口。
4. 使能 SAI_A。

注： *使能 PDM 接口和 SAI_A 后，在 SAI_ADR 上接收到的前 2 个 TDMA 帧将是无效帧，应将其丢弃。*

启动序列

图 646 显示了启动序列：PDM 接口使能后，会等待帧同步事件，然后再开始采集麦克风采样。经过 8 个 SAI_CK 时钟周期之后，来自每个麦克风的数据字节将变为可用，并会通过 TDM 接口传输到 SAI。

图 646. 启动序列



SAI_ADR 数据格式

在 SAI_ADR 寄存器中，将根据以下参数对来自麦克风的数据进行排列：

- 麦克风的数量
- 所选 Slot 宽度
- 左对齐第一位

Slot 宽度定义了 SAI_ADR 寄存器中每个字的有效位数。

如果选择 32 位的 Slot 宽度，则 SAI_ADR 中的每个数据将包含 32 个有用位。这样可减少存储到存储器中的字数。不过，软件必须相应地执行某些操作来对每个麦克风的数据进行去交错。

另一方面，如果将 Slot 宽度设为 8 位，则 SAI_ADR 中的每个数据将包含 8 个有用位。这样会增加存储到存储器中的字数。不过，这样有助于避免额外的处理操作，因为每个字包含一个麦克风的信息。

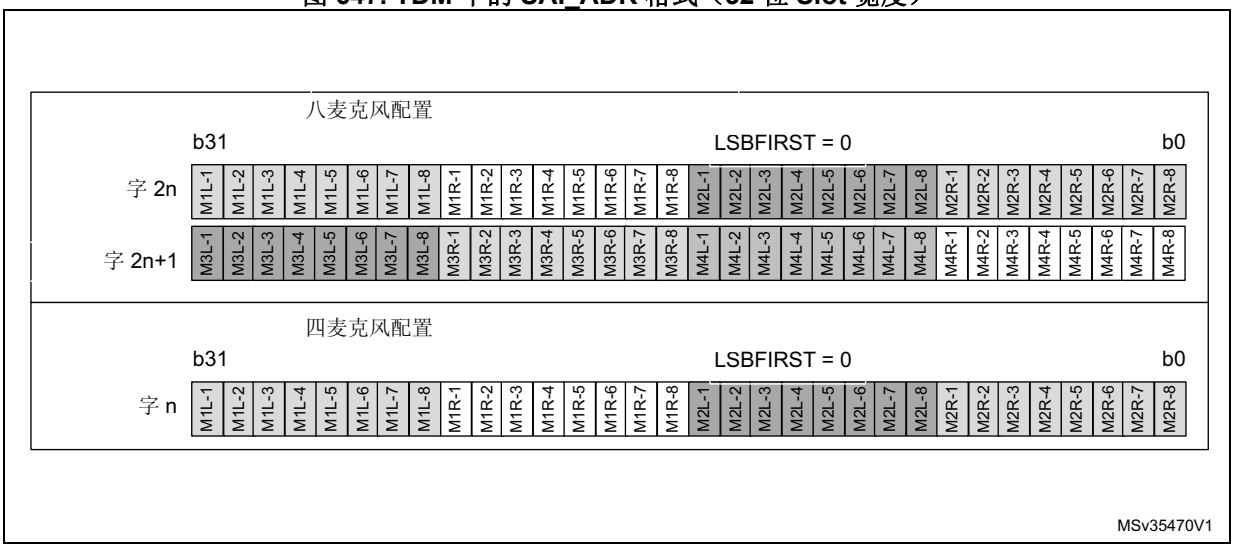
SAI_ADR 数据格式示例

- **32 位 Slot 宽度**（DS = 0b111 且 SLOTSZ = 0）。请参见*图 647*。

  对于 8 麦克风配置，连续对 SAI_ADR 寄存器进行两次字读操作可获取所有麦克风的数据字节。

  对于 4 麦克风配置，每次对 SAI_ADR 寄存器进行字读操作均可获取所有麦克风的数据字节。

图 647. TDM 中的 SAI_ADR 格式（32 位 Slot 宽度）

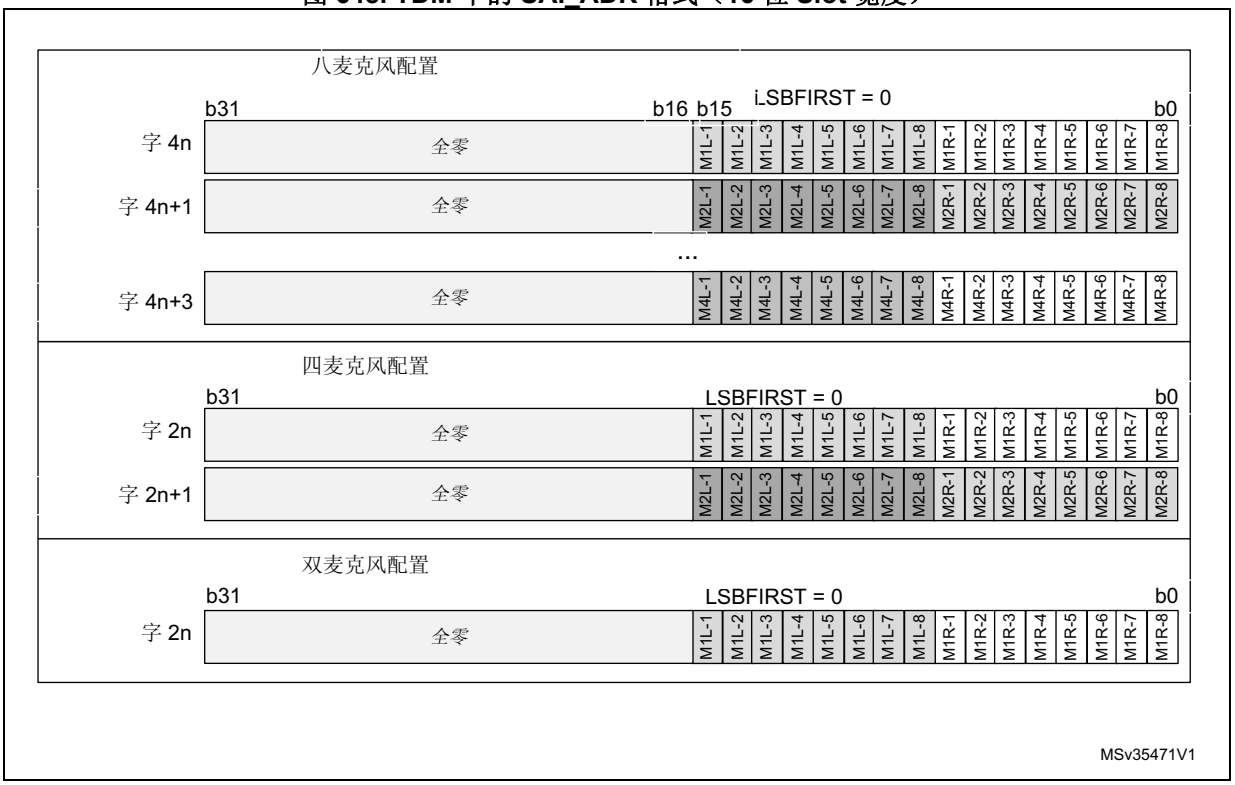


- **16 位 Slot 宽度**（DS = 0b100 且 SLOTSZ = 0）。请参见*图 648*。

  对于 8 麦克风配置，连续对 SAI_ADR 寄存器进行四次字读操作可获取所有麦克风的数据字节。请注意，SAI_ADR 的 16 位数据为右对齐。

  对于 4 麦克风配置或 2 麦克风配置，SAI 行为类似于 8 麦克风配置。获取 4 个麦克风的数据字节需要多达 2 个 16 位字；而获取 2 个麦克风的数据字节需要 1 个 16 位字。

图 648. TDM 中的 SAI_ADR 格式（16 位 Slot 宽度）

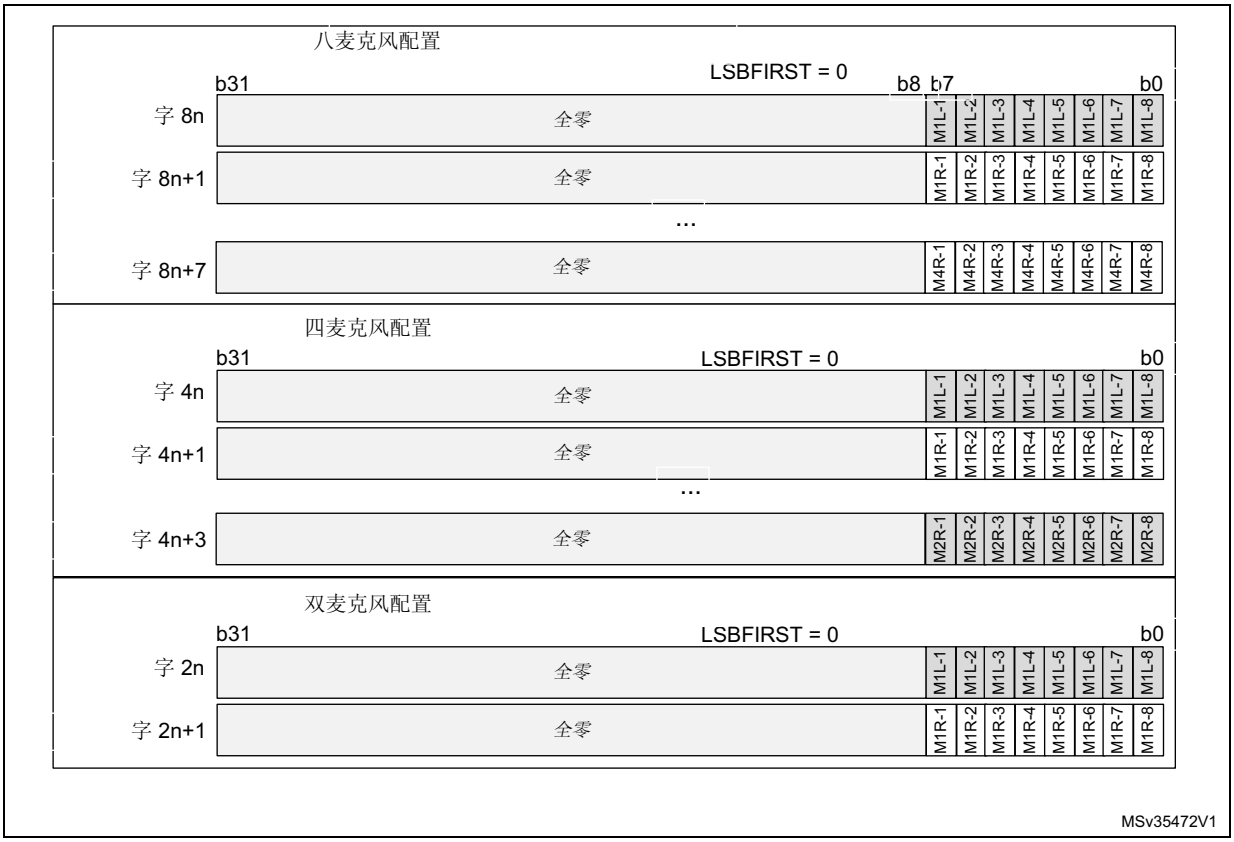


- **使用 8 位 Slot 宽度**（DS = 0b010 且 SLOTSZ = 0）。请参见*图 649*。

  对于 8 麦克风配置，连续对 SAI_ADR 寄存器进行 8 次字读操作可获取所有麦克风的数据字节。请注意，SAI_ADR 的 8 位数据为右对齐。

  对于 4 麦克风配置或 2 麦克风配置，SAI 行为类似于 8 麦克风配置。获取 4 个麦克风的数据字节需要多达 4 个 8 位字；而获取 2 个麦克风的数据字节需要 2 个 8 位字。

图 649. TDM 中的 SAI_ADR 格式（8 位 Slot 宽度）



PDM 接口的 TDM 配置

SAI_A TDM 接口从内部连接到 PDM 接口，以获取麦克风采样。用户应用程序必须按照 *表 400* 所示配置 PDM 接口，以确保与 PDM 接口的连接良好。

表 400. TDM 设置

位域	值	注释
MODE	0b01	必须为主接收模式
PRTCFG	0b00	TDM 的自由协议
DS	X	根据所需数据格式，通过帧长度和 Slot 数（FRL 和 NBSLOT）进行调整。请参见*表 401*。
LSBFIRST	X	可根据所需数据格式使用该参数
CKSTR	0	信号在 SCK_A 位时钟的上升沿发生跳变， 在位时钟的下降沿达到稳定状态。
MONO	0	立体声模式
FRL	X	根据麦克风数量 (MICNBR) 进行调整。请参见*表 401*。
FSALL	0	脉宽为一个位时钟周期
FSDEF	0	FS 信号为帧起始信号
FSPOL	1	FS 为高电平有效

表 400. TDM 设置（续）

位域	值	注释
FSOFF	0	在 Slot 0 的第一位上使能 FS
FBOFF	0	Slot 上无偏移
SLOTSZ	0	Slot 大小 = 数据大小
NBSLOT	X	根据所需数据格式，通过 Slot 大小和帧长度（FRL 和 DS）进行调整。请参见*表 401*。
SLOTEN	X	根据 NBSLOT 进行调整
NOMCK	1	无需生成主时钟 MCLK
MCKDIV	X	取决于提供给 sai_a_ker_ck 输入的频率。 应对该参数进行调整，以生成合适的比特流时钟频率。请参见*表 401*。

调整比特流时钟频率

为正确编程 SAI TDM 接口，用户应用程序必须考虑*表 400* 中给出的设置，并遵循以下规则：

1. 使用以下公式，根据 PDM 比特流时钟所需的频率调整位时钟频率 (F_{SCK_A})：

$$F_{SCK_A} = F_{PDM_CK} \times (MICNBR + 1) \times 2$$

MICNBR 可以是 0、1、2 或 3（0 = 2 个麦克风，请参见*第 51.5.10 节*）

2. 使用以下公式设置帧长度 (FRL)

$$FRL = (16 \times (MICNBR + 1)) - 1$$

3. 将 Slot 大小 (DS) 配置为 (FRL+1) 的倍数。

表 401. 允许的 TDM 帧配置(1)

麦克风采样率	麦克风数量	所需 SAI_CKn 频率 (2)	位时钟 (SCK_A) 频率	帧同步 (FS_A) 频率	FRL	DS	NBSLOT	注释
48 kHz	最多 8 个	3.072 MHz	24.576 MHz	384 kHz	63	0b111	1	每个帧 2 个 32 位 Slot
		3.072 MHz	24.576 MHz	384 kHz	63	0b100	3	每个帧 4 个 16 位 Slot
		3.072 MHz	24.576 MHz	384 kHz	63	0b010	7	每个帧 8 个 8 位 Slot
	最多 6 个	3.072 MHz	18.432 MHz	384 kHz	47	0b110	1	每个帧 2 个 24 位 Slot
		3.072 MHz	18.432 MHz	384 kHz	47	0b100	2	每个帧 3 个 16 位 Slot
		3.072 MHz	18.432 MHz	384 kHz	47	0b010	5	每个帧 6 个 8 位 Slot
	最多 4 个	3.072 MHz	12.288 MHz	384 kHz	31	0b111	0	每个帧 1 个 32 位 Slot
		3.072 MHz	12.288 MHz	384 kHz	31	0b100	1	每个帧 2 个 16 位 Slot
		3.072 MHz	12.288 MHz	384 kHz	31	0b010	3	每个帧 4 个 8 位 Slot
	最多 2 个	3.072 MHz	6.144 MHz	384 kHz	15	0b100	0	每个帧 1 个 16 位 Slot
		3.072 MHz	6.144 MHz	384 kHz	15	0b010	1	每个帧 2 个 8 位 Slot
16 kHz	最多 8 个	1.024 MHz	8.192 MHz	128 kHz	63	0b111	1	每个帧 2 个 32 位 Slot
		1.024 MHz	8.192 MHz	128 kHz	63	0b100	3	每个帧 4 个 16 位 Slot
		1.024 MHz	8.192 MHz	128 kHz	63	0b010	7	每个帧 8 个 8 位 Slot
	最多 6 个	1.024 MHz	6.144 MHz	128 kHz	47	0b110	1	每个帧 2 个 24 位 Slot
		1.024 MHz	6.144 MHz	128 kHz	47	0b010	5	每个帧 6 个 8 位 Slot
	最多 4 个	1.024 MHz	4.096 MHz	128 kHz	31	0b111	0	每个帧 1 个 32 位 Slot
		1.024 MHz	4.096 MHz	128 kHz	31	0b100	1	每个帧 2 个 16 位 Slot
		1.024 MHz	4.096 MHz	128 kHz	31	0b010	3	每个帧 4 个 8 位 Slot
	最多 2 个	1.024 MHz	2.048 MHz	128 kHz	15	0b100	0	每个帧 1 个 16 位 Slot
		1.024 MHz	2.048 MHz	128 kHz	15	0b010	1	每个帧 2 个 8 位 Slot

1. 有关 TDM 配置的更多信息，请参见*表 400：TDM 设置*。提供给 SAI 的 sai_a_ker_ck 时钟频率应是 SCK_A 频率的倍数，并且应相应地编程 MCKDIV。
2. 上表给出了抽取率为 64 时所允许的设置。

调整延迟线

使能 PDM 接口时，应用程序可通过 SAI_PDMDLY 寄存器实时调整每个麦克风输入的延迟单元数。

新的延迟值将在两个 TDM 帧后生效。

51.3.11 AC’97 链路控制器

SAI 可用作 AC’97 链路控制器。在此协议中：

- Slot 数和 Slot 大小固定。
- 帧同步信号定义完善并且波形固定。

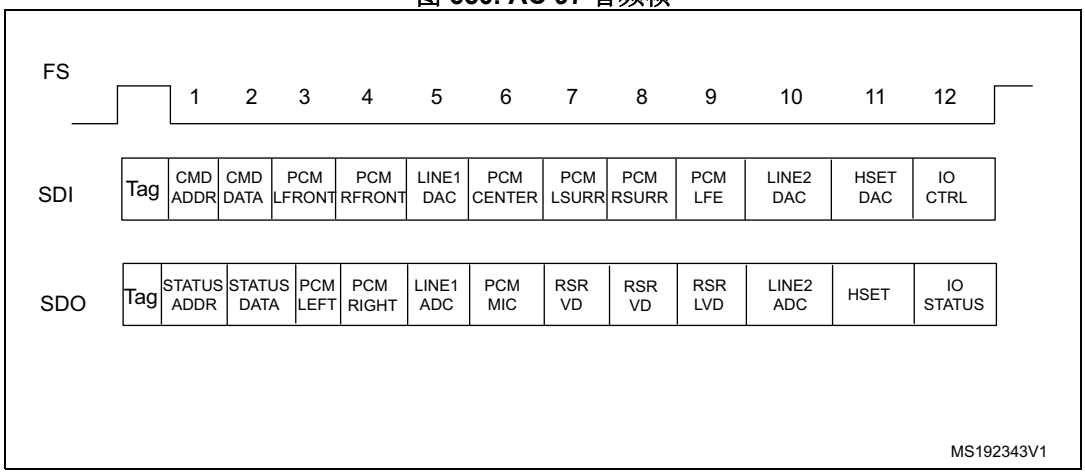
要选择此协议，可将 SAI_xCR1 寄存器中的 PRTCFG[1:0] 位置为 10。选择 AC’97 模式时，只能使用 16 位或 20 位的数据大小，否则将无法保证 SAI 运行正常。

- 因此，NBSLOT[3:0] 和 SLOTSZ[1:0] 位将被忽略。
- Slot 数固定为 13。第一个 Slot 为 16 位宽，所有其它 Slot 均为 20 位宽（数据 Slot）。
- SAI_xSLOTR 寄存器中的 FBOFF[4:0] 位被忽略。
- SAI_xFRCR 寄存器被忽略。
- 未使用 MCLK。

无论采用主配置还是从配置，由于 AC’97 链路控制器驱动 FS 信号，异步模块发出的 FS 信号将自动配置为输出。

图 650 给出了 AC’97 音频帧的结构。

图 650. AC’97 音频帧



注： *在 AC’97 协议中，TAG 的位 2 将保留（始终为 0），因此无论 SAI FIFO 中写入何值，TAG 的位 2 均强制为 0。*

有关 TAG 表示的详细信息，请参见 AC’97 协议标准。

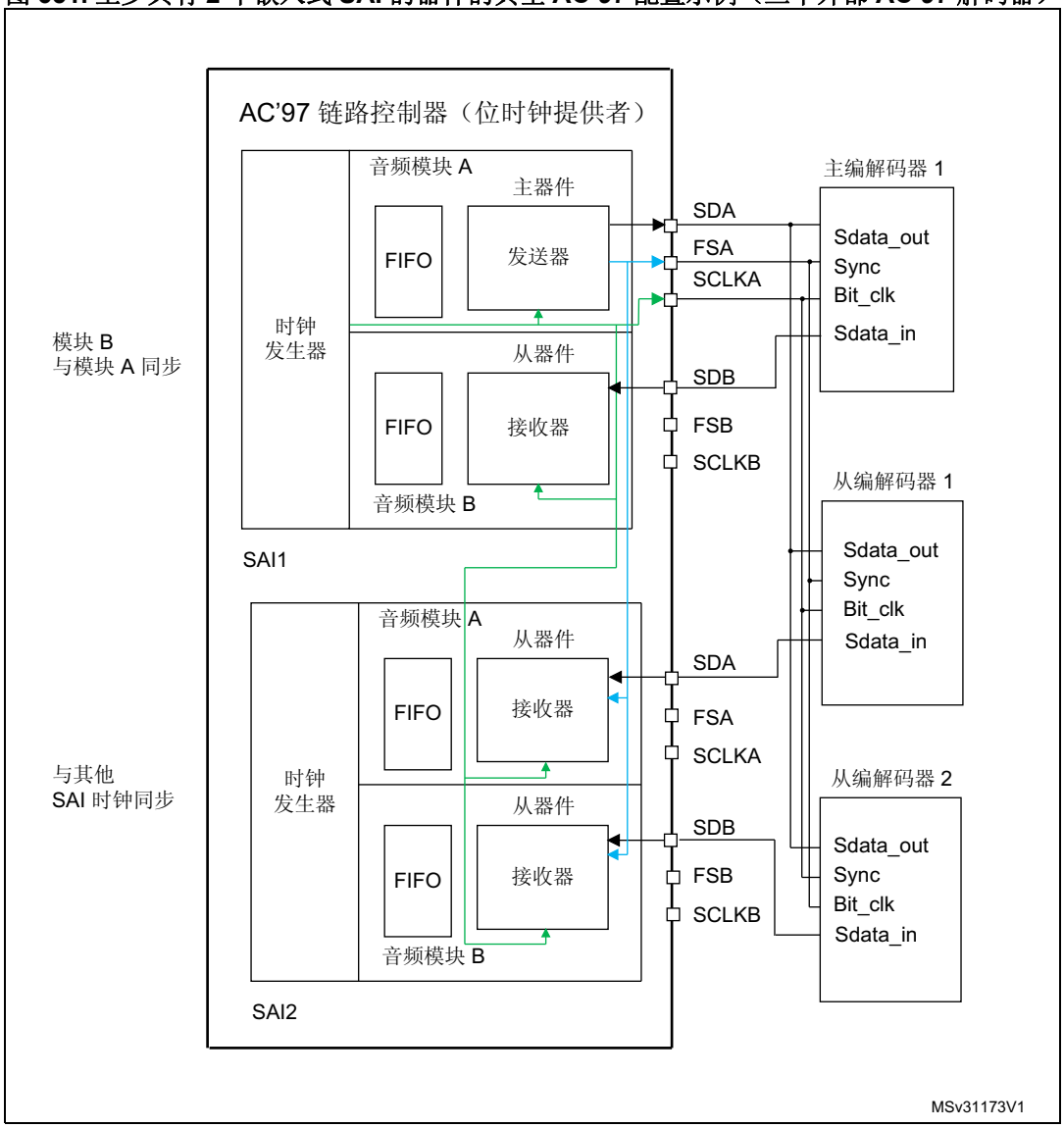
可将一个 SAI 用于 AC’97 点对点通信。

使用两个 SAI（针对具有两个嵌入式 SAI 的器件）可以控制三个外部 AC’97 解码器，如*图 651* 所示。

在 SAI1 中，必须将音频模块 A 声明为异步主发送器，而将音频模块 B 定义为从接收器，并在内部与音频模块 A 同步。

在从接收器模式下，同时针对音频模块 A 和音频模块 B 将 SAI2 配置为与外部 SAI1 同步。

图 651. 至少具有 2 个嵌入式 SAI 的器件的典型 AC’97 配置示例（三个外部 AC’97 解码器）



在接收器模式下，用作 AC’97 链路控制器的 SAI 无需 FIFO 请求，因此当 Slot 0 中的编码解码器就绪位解码为低电平时，FIFO 中不存储任何数据。如果 SAI_xIM 寄存器中的 CNRDYIE 位使能，则 SAI_xSR 寄存器中的标志 CNRDY 将置 1 并会产生一个中断。此标志专用于 AC’97 协议。

AC’97 模式下的时钟发生器编程

在 AC’97 模式下，帧长度固定为 256 位，其频率应设置为 48 kHz。*第 51.3.8 节：SAI 时钟发生器*中给出的规则应在 FRL = 255 时使用，以生成适当的帧速率 (F_{FS_x})。

51.3.12 SPDIF 输出

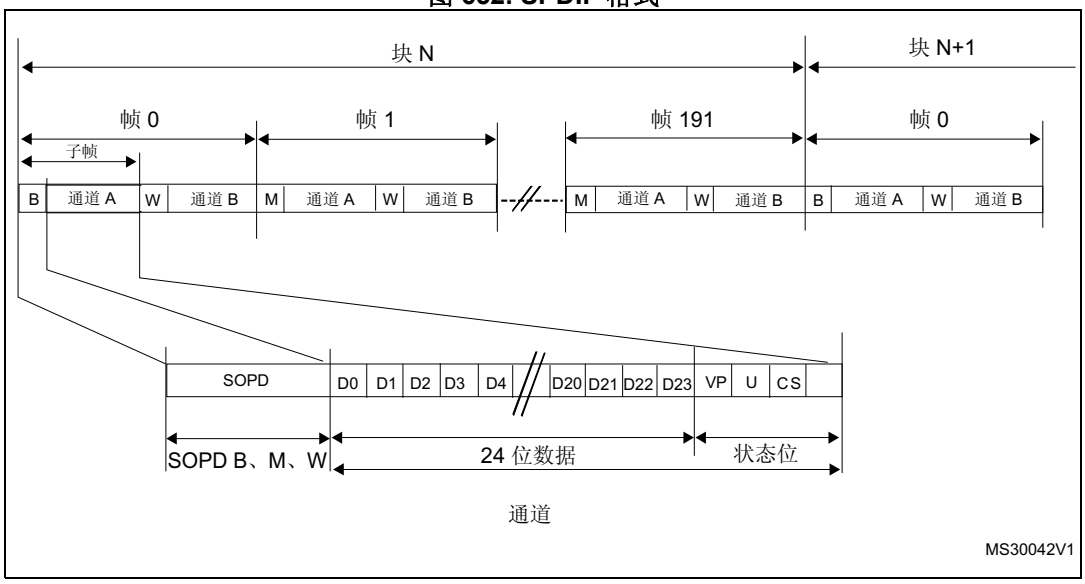
SPDIF 接口仅在发送模式下可用。它支持 IEC60958 音频标准。

要选择 SPDIF 模式，需将 SAI_xCR1 寄存器中的 PRTCFG[1:0] 位置为 01。

对于 SPDIF 协议：

- 仅使能 SD 数据线。
- 释放 FS、SCK、MCLK I/O 引脚。
- 为使能 SAI 的时钟发生器并管理 SD 线上的数据速率，将 MODE[1] 位强制设置为 0 以选择主模式。
- 数据大小强制设置为 24 位。忽略在 SAI_xCR1 寄存器的 DS[2:0] 位中设置的值。
- 必须配置时钟发生器以定义符号率，已知位时钟应为符号率的两倍。数据采用曼彻斯特协议进行编码。
- 忽略 SAI_xFRCR 和 SAI_xSLOTR 寄存器。在内部配置 SAI 使其符合 SPDIF 协议要求，如*图 652* 所示。

图 652. SPDIF 格式



SPDIF 块包含 192 个帧。每个帧由两个 32 位的子帧构成，通常一个子帧用于左通道，一个用于右通道。每个子帧由一个 SOPD 模式（4 位）构成，用于指定该子帧是否为块的开始（并用于识别通道 A），或者是否标识块中某处的通道 A，或者是否指代通道 B（请参见*表 402*）。接下来的 28 位是通道信息，由 24 个数据位和 4 个状态位组成。

表 402. SOPD 模式

SOPD	报头编码		说明
	最后一位是 0	最后一位是 1	
B	11101000	00010111	块开始处的通道 A 数据
W	11100100	00011011	块中某处的通道 B 数据
M	11100010	00011101	通道 A 数据

必须按如下方式填充 SAI_xDR 中存储的数据：

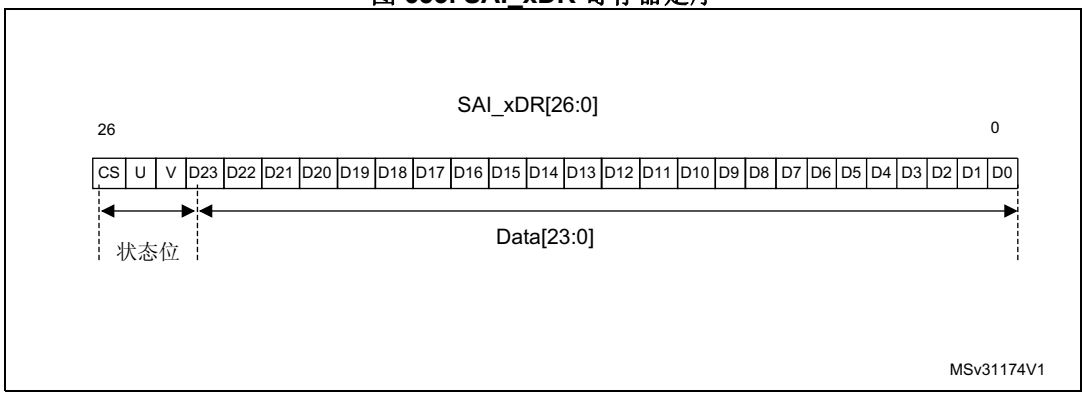
- SAI_xDR[26:24] 包含通道状态位、用户位和有效性位。
- SAI_xDR[23:0] 包含所考虑通道的 24 位数据。

如果数据大小为 20 位，应将数据映射到 SAI_xDR[23:4] 上。

如果数据大小为 16 位，应将数据映射到 SAI_xDR[23:8] 上。

SAI_xDR[23] 始终代表 MSB。

图 653. SAI_xDR 寄存器定序



注： *执行传输时，LSB 始终优先。*

SAI 首先在块中发送每个子帧的适当报头。随后在 SD 线上发送 SAI_xDR（以曼彻斯特协议进行编码）。SAI 通过传输按 *表 403* 所述计算的奇偶校验位来结束子帧。

表 403. 奇偶校验位计算

SAI_xDR[26:0]	传输奇偶校验位 P 值
奇数个 0	0
奇数个 1	1

在 SPDIF 模式下，下溢是在 SAI_xSR 寄存器唯一个用到的错误标志。因为 SAI 只能在发送模式下工作。因此，要从通过下溢中断或下溢状态位检测到的下溢错误中恢复，应按顺序执行以下步骤：

1. 如果已使用 DMA，则禁止 DMA 流（通过 DMA 外设）。
2. 禁止 SAI 并通过轮询 SAI_xCR1 寄存器中的 SAIEN 位确认已从物理上禁止外设。
3. 将 SAI_xCLRFR 寄存器中的 COVRUNDR 标志清零。
4. 通过将 SAI_xCR2 寄存器中的 FFLUSH 位置 1 来刷新 FIFO。
   软件需要指向与新块开始（报头 B 的数据）相对应的后续数据的地址。如果已使用 DMA，应相应地更新 DMA 源起始地址指针。
5. 如果用于管理数据的 DMA 根据新的源起始地址来传输数据，则再次使能 DMA 流（DMA 外设）。
6. 通过将 SAI_xCR1 寄存器中的 SAIEN 位置 1 再次使能 SAI。

SPDIF 发生器模式下的时钟发生器编程

对于 SPDIF 发生器，SAI 应提供与符号率相等的位时钟。下表给出了符号率相对于音频采样率的通常示例。

表 404. 音频采样频率与符号率

音频采样频率 (F_S)	符号率
44.1 kHz	2.8224 MHz
48 kHz	3.072 MHz
96 kHz	6.144 MHz
192 kHz	12.288 MHz

通常，音频采样率 (F_S) 与位时钟速率 (F_{SCK_X}) 之间的关系由以下公式给出：

51.3.13 特性

根据所选的音频协议，SAI 接口内嵌了一些特定的实用功能。这些功能可通过 SAI_xCR2 寄存器的特定位来访问。

静音模式

当音频子模块用作发送器或接收器时，可使用静音模式。

发送模式下的音频子模块

在发送模式下，可随时选择静音模式。静音模式对于全部音频帧均有效。在帧传输过程中将 SAI_xCR2 寄存器的 MUTE 置 1，将使能静音模式。

该静音模式位仅在帧结束时选通。如果帧结束时置 1，静音模式将在新的音频帧开始时激活，并持续整个帧长度，直至下次帧结束； 然后将选通该位，以确定下一帧是否仍为静音帧。

如果通过 SAI_xSLOTR 寄存器的 NBSLOT[3:0] 位设置的 Slot 数小于或等于 2，可指定静音模式下发送的值是否为 0 或该值是否为每个 Slot 的最后一个值。通过 SAI_xCR2 寄存器中的 MUTEVAL 位进行选择。

如果在 SAI_xSLOTR 寄存器的 NBSLOT[3:0] 位中设置的 Slot 数大于 2，由于在各 Slot 的每位上都发送值 0，因此 SAI_xCR2 中的 MUTEVAL 位没有意义。

在静音模式下，FIFO 指针仍递增， 这意味着将丢弃 FIFO 中请求在静音模式下传输的数据。

接收模式下的音频子模块

在接收模式下，对于给定数量的连续音频帧（SAI_xCR2 寄存器中的 MUTECNT[5:0] 位），如果在音频帧所有声明的有效 Slot 接收到 0，则可检测到从外部发送器发来的静音模式。

检测到相应数量的静音帧时，SAI_xSR 寄存器中的 MUTEDET 标志置 1 并会在 SAI_xCR2 中的 MUTEDETIE 位置 1 情况下产生中断。

在音频子模块禁止时或在有效 Slot 内至少接收到音频帧中的一个数据时，静音帧计数器清零。计数器达到 MUTECNT[5:0] 位中指定的值时，仅产生一次中断。随后中断事件在计数器清零时重新初始化。

注： *静音模式不可用于 SPDIF 音频模块。*

单声道/立体声模式

在发送器模式下，如果 Slot 数等于 2（SAI_xSLOTR 中的 NBSLOT[3:0] = 0001），则可寻址单声道模式，而无需在存储器中进行任何数据预处理。在这种情况下，由于发送时 Slot 0 的数据被复制到数据 Slot 1 中，FIFO 的访问时间将减少一半。

要使能单声道模式，

1. 将 SAI_xCR1 寄存器中的 MONO 位置 1。
2. 将 SAI_xSLOTR 中的 NBSLOT 置 1，并将 SLOTEN 设置为 3。

在接收模式下，MONO 位可置 1 并且仅在 Slot 数等于 2（与发送模式下相同）时才有意义。当该位置 1 时，只有 Slot 0 的数据将存储到 FIFO 中。Slot 1 的数据由于被认为与前一个 Slot 的数据相同而被丢弃。如果接收模式下的数据流量是左声道数据和右声道数据明显不同的真立体声音频流，则 MONO 位没有意义。由软件完成从输出立体声文件到等效单声道文件的转换。

压扩模式

移动通信应用可能需要通过数据压扩算法处理待发送或待接收的数据。

应用可根据 SAI_xCR2 寄存器中的 COMP[1:0] 位（仅当选择 TDM 模式时使用），选择在 SD 串行输出线（压缩）发送数据前是否处理数据，以及是否在 SD 串行输入线（扩展）接收数据后扩展数据，如 *图 654* 所示。所支持的两个压扩模式是 μ-Law 和 A-Law，二者是 CCITT G.711 推荐标准的一部分。

美国和日本采用的压扩标准是 μ-Law， 该标准允许 14 位动态范围（SAI_xCR2 寄存器中的 COMP[1:0] = 10）。

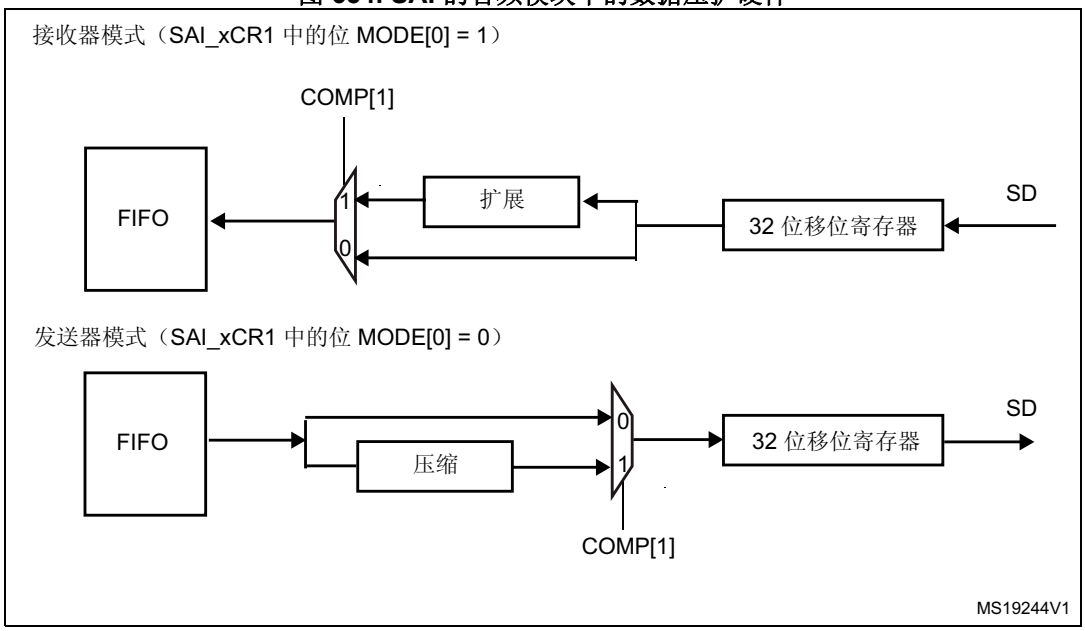
欧洲压扩标准是 A-Law，该标准支持 13 位动态范围（SAI_xCR2 寄存器中的 COMP[1:0] = 11）。

可根据 1 的补码或 2 的补码表示来计算 μ-Law 或 A-Law 压扩标准，具体取决于 SAI_xCR2 寄存器中的 CPL 位设置。

在 μ-Law 和 A-Law 标准中，数据将编码为采用 MSB 对齐的 8 位。压扩数据始终为 8 位宽。因此，当 SAI 音频模块使能（SAI_xCR1 寄存器中的位 SAIEN = 1）并且通过 COMP[1:0] 位选择这两个压扩模式之一时，SAI_xCR1 寄存器中的 DS[2:0] 位将强制为 010。

如果无需压扩处理，则 COMP[1:0] 位应保持清零。

图 654. SAI 的音频模块中的数据压扩硬件



1. 选择 AC'97 或 SPDIF 时不适用。

通过 SAI_xCR2 自动选择扩展模式和压缩模式：

- 如果 SAI 音频模块配置为发送器，且 SAI_xCR2 寄存器中的 COMP[1] 位置 1，将采用压缩模式。
- 如果 SAI 音频模块声明为接收器，将采用扩展算法。

无效 Slot 上的输出数据线管理

在发送模式下，当在数据线上发送无效 Slot 时，可选择 SD 线输出的行为（通过 TRIS 位实现）。

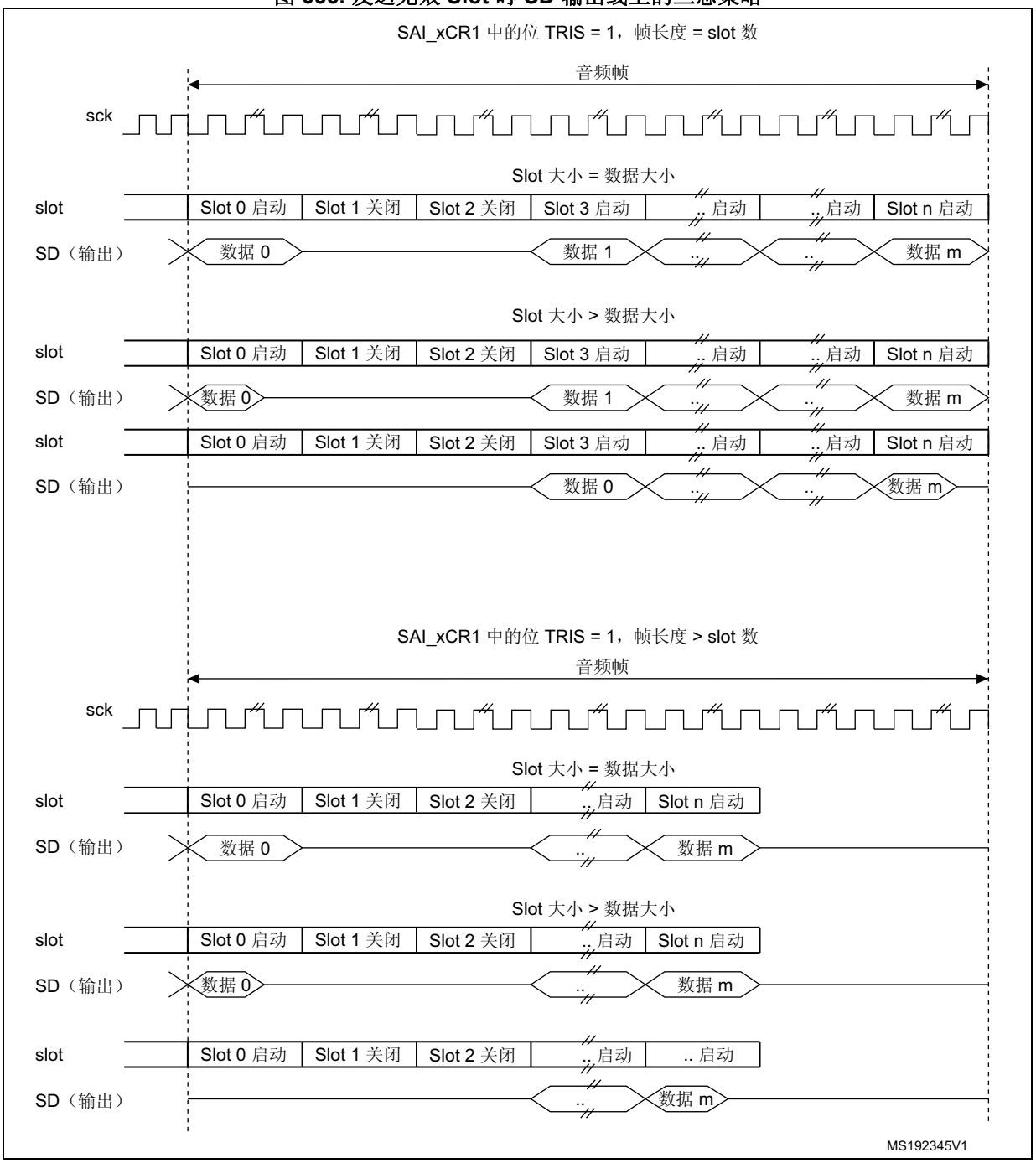
- 发送无效 Slot 时，SAI 将 SD 输出线强制为 0。
- 该输出线在最有一个数据位传输结束时释放为高阻态，为另一个连接此节点的发送器释放该数据线。

切记不要让两个发送器同时驱动同一个 SD 输出引脚，否则会导致短路。为确保存在发送间隙，如果数据低于 32 位，可通过在 SAI_xSLOTR 寄存器中设置 SLOTSZ[1:0] = 10 将数据扩展到 32 位。随后，如果下一 Slot 声明为无效，则 SD 输出引脚将在有效 Slot 的 LSD 结束时（将数据扩展到 32 位的填 0 阶段）置为三态。

此外，如果 Slot 数乘以 Slot 大小所得结果小于帧长度，则在通过填 0 来补充音频帧结束时，SD 输出线置为三态。

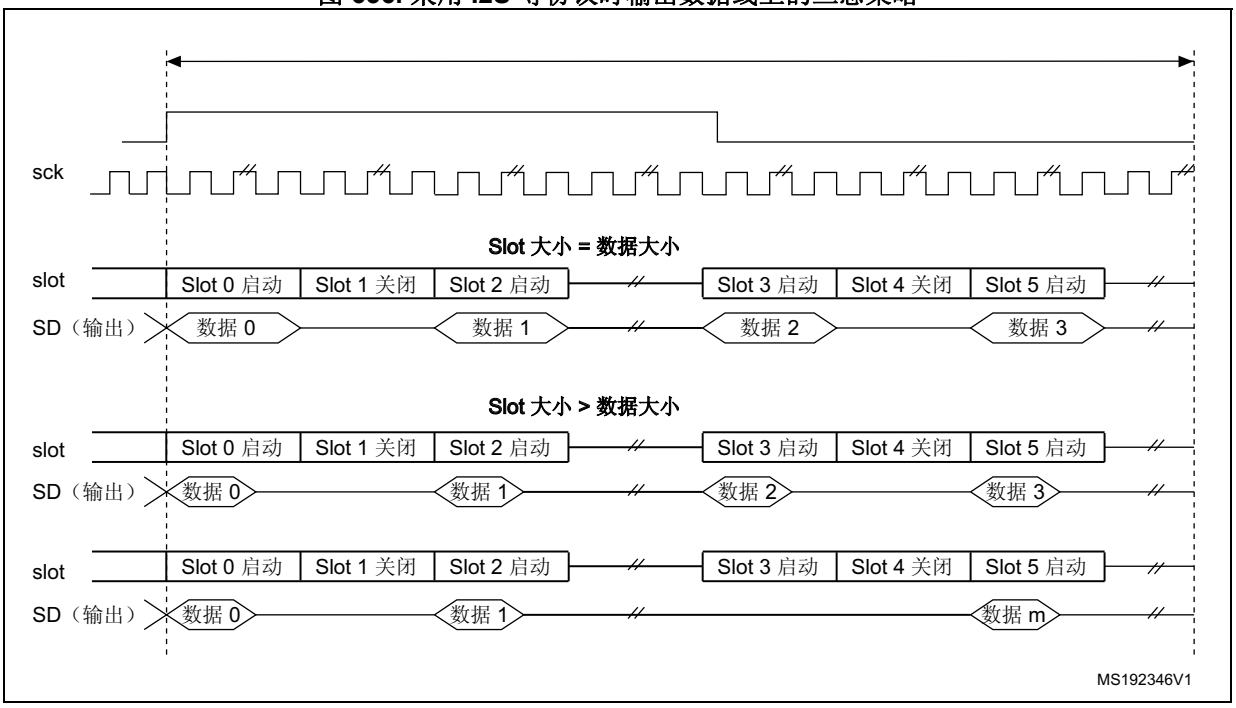
图 655 说明了这些行为。

图 655. 发送无效 Slot 时 SD 输出线上的三态策略



当所选音频协议使用 FS 信号作为 SOF 信号或通道识别信号（SAI_xFRCR 寄存器中的位 FSDEF = 1）时，将按照*图 656* 管理三态模式（其中，SAI_xCR1 寄存器中的位 TRIS = 1，FSDEF = 1，半帧长大于 Slot 数/2 且 NBSLOT = 6）。

图 656. 采用 I2S 等协议时输出数据线上的三态策略



如果 SAI_xCR2 寄存器中的 TRIS 位清零，*图 655* 和*图 656* 上的 SD 输出线上的所有高阻态将替换为使用值 0 驱动。

51.3.14 错误标志

SAI 使用以下错误标志：

- FIFO 上溢/下溢
- 帧同步提前检测
- 帧同步滞后检测
- 编解码器未就绪（仅限 AC'97）
- 主模式时钟配置错误。

FIFO 上溢/下溢 (OVRUDR)

FIFO 上溢/下溢位是 SAI_xSR 寄存器中的 OVRUDR 位。

由于音频模块既可作为接收器，又可作为发送器，并且指定 SAI 中的每个音频模块都具有自己的 SAI_xSR 寄存器，因此上溢或下溢错误共用同一位。

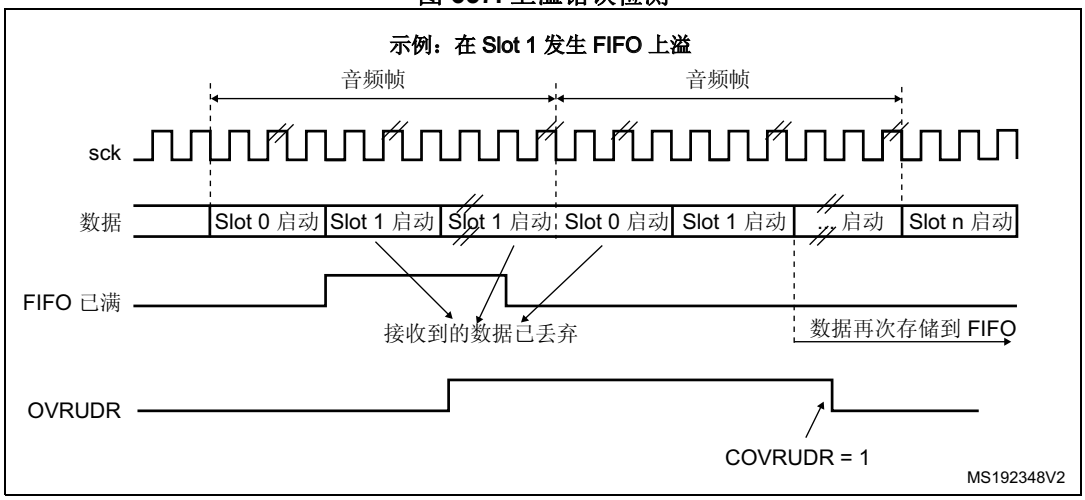
上溢

若音频模块配置为接收器，则在 FIFO 已满且无法再存储接收数据的情况下又收到音频帧数据时，将出现上溢情况。这种情况下，接收数据将丢失，SAI_xSR 寄存器中的 OVRUDR 标志置 1；如果 SAI_xIM 寄存器中的 OVRUDRIE 位置 1，还将生成中断。内部将存储发生上溢时的 Slot 编号。FIFO 无法再存储更多数据，直至释放出空间存储新数据为止。在 FIFO 释放了至少一个数据的空间时，SAI 音频模块接收器将从检测到上溢后内部记录的 Slot 编号

开始接收来自新音频帧的新数据， 这样可避免目标存储器中出现数据 Slot 不对齐的情况（请参见*图 657*）。

SAI_xCLRFR 寄存器中的 COVRUDR 位置 1 时将清除 OVRUDR 标志。

图 657. 上溢错误检测



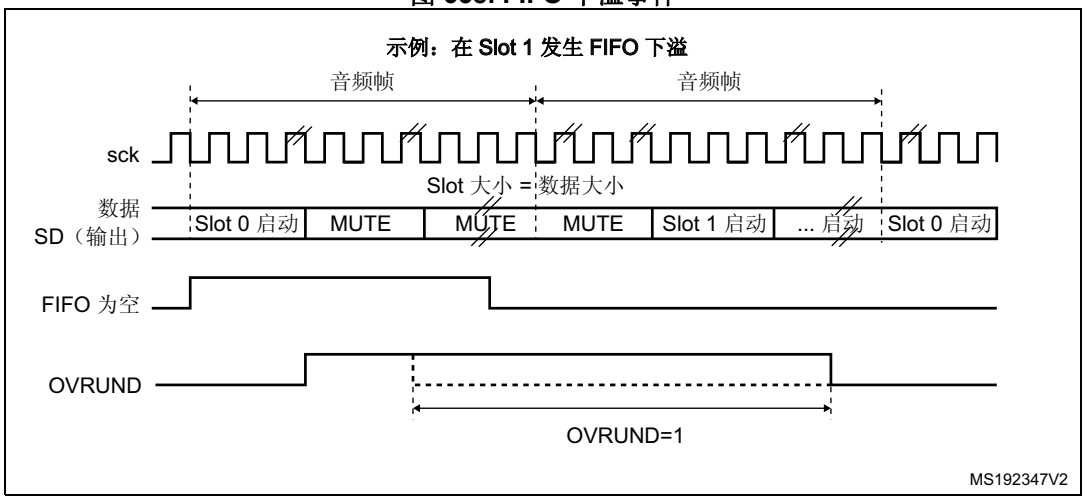
下溢

当 SAI 中的音频模块用作发送器时，如果需要发送数据时 FIFO 为空，则可能出现下溢。如果检测到下溢，则将存储发生事件的 Slot 编号并发送 MUTE 值 (00)，直到 FIFO 准备好发送与检测到下溢的 Slot 对应的数据（请参见*图 658*）。这样可避免存储器指针与音频帧中的 Slot 之间发生同步失效。

下溢事件会使 SAI_xSR 寄存器中的 OVRUDR 标志置 1，如果 SAI_xIM 寄存器中的 OVRUDRIE 位置 1，还将生成中断。要清除该标志，可将 SAI_xCLRFR 寄存器中的 COVRUDR 位置 1。

将音频子模块配置为主模式或从模式时，会发生下溢事件。

图 658. FIFO 下溢事件



帧同步提前检测 (AFSDET)

AFSDET 标志仅在从模式下使用。主模式下不会使能该标志。由于帧长度、帧极性和帧偏移已定义且已知，该标志用于指示是否比预期更早检测到帧同步 (FS) 信号。

出现帧检测提前时，SAI_xSR 寄存器中的 AFSDET 标志将置 1。

对 FS 提前不敏感的当前音频帧不受该检测影响。也就是说 FS 信号的“寄生”事件将被标记但不干扰当前音频帧。

如果 SAI_xIM 寄存器中的 AFSDETIE 位置 1，将生成中断。要清除 AFSDET 标志，必须将 SAI_xCLRFR 寄存器中的 CAFSDET 位置 1。

为了在出现帧检测提前错误后重新与主模块同步，需要执行以下四个步骤：

1. 通过复位 SAI_xCR1 寄存器中的 SAIEN 位禁止 SAI 模块。为确保禁止 SAI，读回 SAIEN 位并检查其是否已设置为 0。
2. 通过 SAI_xCR2 寄存器的 FFLUS 位刷新 FIFO。
3. 重新使能 SAI 外设（SAIEN 位置 1）。
4. SAI 模块将等待 FS 使能以重新开始与主模块同步。

注： *AC’97 模式下不使能 SAIEN 标志，原因是 SAI 音频模块作为链路控制器，即使在声明为从模块时也会生成 FS 信号。由于未使用 FS 信号，因此它在 SPDIF 模式下无意义。*

帧同步滞后检测

只有当 SAI 音频模块用作从模块时，SAI_xSR 寄存器中的 LFSDET 标志才可置 1。SAI_xFRCR 寄存器中，帧长度、帧极性和帧偏移配置均已知。

如果外部主模块未在预定时间发送 FS 信号生成过晚），LFSDET 标志将置 1，如果 SAI_xIM 寄存器中的 LFSDETIE 位置 1，还将生成中断。

SAI_xCLRFR 寄存器中的 CLFSDET 位置 1 时将清除 LFSDET 标志。

在检测到相应错误时帧同步滞后检测标志置 1， SAI 需要重新与主模块同步（请参见*帧同步提前检测 (AFSDET)*一节 中所述的顺序）。

在噪声环境中，音频模块的状态机可能会错误检测到对 SCK 时钟的干扰，并将 SAI 数据移位到错误的帧位置。SAI 会检测到此事件，并将其报告为帧同步滞后检测错误。

如果外部主模块不是在连续模式下管理音频数据帧发送，则不会对帧造成破坏，大多数应用都不会出现这种情况。这种情况下 LFSDET 标志将置 1。

注： *AC’97 模式下不使能 LFSDET 标志，原因是 SAI 音频模块作为链路控制器，即使在声明为从模块时也会生成 FS 信号。由于协议未使用 FS 信号，因此该位在 SPDIF 模式下无意义。*

编解码器未就绪 (CNRDY AC’97)

仅当 SAI 音频模块配置为在 AC’97 模式下工作时（SAI_xCR1 寄存器中的 PRTCFG[1:0] = 10），SAI_xSR 寄存器中的 CNRDY 标志才有意义。如果 SAI_xIM 寄存器中的 CNRDYIE 位置 1，则在 CNRDY 标志置 1 时将生成中断。

在接收 AC’97 音频帧的 TAG 0 (slot0) 期间，当编解码器未准备好进行通信时，将使能 CNRDY。这种情况下，在 TAG 0 指示编解码器就绪之前，数据都不会自动存储到 FIFO，原因是编解码器未就绪。编解码器就绪后将捕获 SAI_xSLOTR 寄存器中定义的所有有效 Slot 。

要清除 CNRDY 标志，必须将 SAI_xCLRFR 寄存器中的 CCNRDY 位置 1。

主模式时钟配置错误 (NOMCK = 0)

当音频模块在主模式下工作（MODE[1] = 0）且 NOMCK 位等于 0 时，如果满足以下条件，则只要使能 SAI，WCKCFG 标志便会置 1：

- (FRL+1) 不是 2 的几次幂，并且
- (FRL+1) 不在 8 和 256 之间

MODE 位、NOMCK 位和 SAIEN 位属于 SAI_xCR1 寄存器，FRL 位属于 SAI_xFRCR 寄存器。

如果 WCKCFGIE 位置 1，则当 SAI_xSR 寄存器中的 WCKCFG 标志置 1 时将生成中断。要清除该标志，可将 SAI_xCLRFR 寄存器中的 CWCKCFG 位置 1。

当 WCKCFG 位置 1 时，音频模块会自动禁止，因此将对 SAIEN 位执行硬件清零。

51.3.15 禁止 SAI

可随时通过清零 SAI_xCR1 寄存器中的 SAIEN 位禁止 SAI 音频模块。所有已开始的帧将在 SAI 停止工作前自动完成。SAIEN 位将保持高电平，直到当前音频帧传输结束时 SAI 完全关闭。

如果 SAI 中有与另一个音频模块同步工作的音频模块，则必须先禁止以主模式工作的音频模块。

51.3.16 SAI DMA 接口

为减轻 CPU 负担和优化总线带宽，每个 SAI 音频模块都具有独立的 DMA 接口以便对 SAI_xDR 寄存器进行读/写操作（访问内部 FIFO）。每个音频子模块都有一个支持基本 DMA 请求/应答协议的 DMA 通道。

要为 DMA 传输配置音频子模块，可将 SAI_xCR1 寄存器中的 DMAEN 位置 1。DMA 请求直接由 FIFO 控制器管理，具体取决于 FIFO 阈值（更多详细信息，请参见*第 51.3.9 节：内部 FIFO*）。DMA 传输方向与 SAI 音频子模块配置相关：

- 如果音频模块用作发送器，则音频模块的 FIFO 控制器将输出 DMA 请求以向 FIFO 加载 SAI_xDR 寄存器中写入的数据。
- 如果音频模块用作接收器，则 DMA 请求与来自 SAI_xDR 寄存器的读取操作相关。

按照下面的顺序将 SAI 接口配置为 DMA 模式：

1. 配置 SAI 和 FIFO 阈值以指定何时启动 DMA 请求。
2. 配置 SAI DMA 通道。
3. 使能 DMA。
4. 使能 SAI 接口。

注： *配置 SAI 模块前，必须禁止 SAI DMA 通道。*

51.4 SAI 中断

SAI 支持 7 个中断源，如*表 405* 所示。

表 405. SAI 中断源

中断源	中断组	音频模块模式	中断使能	中断清零
FREQ	FREQ	主或从 接收器或发送器	SAI_xIM 寄存器中的 FREQIE	取决于： – FIFO 阈值设置（SAI_xCR2 中的 FLVL 位） – 通信方向（发送器或接收器） 更多详细信息， 请参见*第 51.3.9 节：内部 FIFO*
OVRUDR	ERROR	主或从 接收器或发送器	SAI_xIM 寄存器中的 OVRUDRIE	SAI_xCLRFR 寄存器中的 COVRUDR = 1
AFSDET	ERROR	从 （不适用于 AC’97 模式和 SPDIF 模式）	SAI_xIM 寄存器中的 AFSDETIE	SAI_xCLRFR 寄存器中的 CAFSDET = 1
LFSDET	ERROR	从 （不适用于 AC’97 模式和 SPDIF 模式）	SAI_xIM 寄存器中的 LFSDETIE	SAI_xCLRFR 寄存器中的 CLFSDET = 1
CNRDY	ERROR	从 （仅限 AC’97 模式）	SAI_xIM 寄存器中的 CNRDYIE	SAI_xCLRFR 寄存器中的 CCNRDY = 1
MUTEDET	MUTE	主或从 仅限接收模式	SAI_xIM 寄存器中的 MUTEDETIE	SAI_xCLRFR 寄存器中的 CMUTEDET = 1
WCKCFG	ERROR	主模式且 SAI_xCR1 寄存器中的 NOMCK = 0	SAI_xIM 寄存器中的 WCKCFGIE	SAI_xCLRFR 寄存器中的 CWCKCFG = 1

按照以下顺序使能中断：

1. 禁止 SAI 中断。
2. 配置 SAI。
3. 配置 SAI 中断源。
4. 使能 SAI。

51.5 SAI 寄存器

51.5.1 全局配置寄存器 (SAI_GCR)

Global configuration register

偏移地址：0x00

复位值：0x0000 0000

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

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

位 31:6 保留，必须保持复位值

位 5:4 **SYNCOUT[1:0]**：同步输出 (Synchronization outputs)

这些位将由软件置 1 和清零。

00：无同步输出信号。将音频模块配置为 SPDIF 模式时，应将 SYNCOUT[1:0] 配置为“无同步输出信号”

01：模块 A 用于其它 SAI 的进一步同步

10：模块 B 用于其它 SAI 的进一步同步

11：保留。必须在音频模块（A 和 B）禁止的情况下设置这些位

位 3:2 保留，必须保持复位值

位 1:0 **SYNCIN[1:0]**：同步输入 (Synchronization inputs)

这些位将由软件置 1 和清零。

有关如何编程此位域的信息，请参见。

必须在音频模块（A 和 B）禁止的情况下配置这些位。

如果将两个音频模块之一定义为与外部 SAI 在同步模式下工作（SAI_ACR1 或 SAI_BCR1 寄存器中的 SYNCEN[1:0] = 10），这些位便有意义。

51.5.2 配置寄存器 1 (SAI_ACR1/SAI_BCR1)

Configuration register 1

偏移地址：模块 A 为 0x004

偏移地址：模块 B 为 0x024

复位值：0x0000 0040

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	OSR	MCKDIV[5:0]						NOMCK	Res.	DMAEN	SAI EN
					rw	rw	rw	rw	rw	rw	rw	rw		rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	OUTDRIV	MONO	SYNCEN[1:0]		CKSTR	LSBFIRST	DS[2:0]			Res.	PRTCFG[1:0]		MODE[1:0]	
		rw	rw	rw	rw	rw	rw	rw	rw	rw		rw	rw	rw	rw

位 31:27 保留，必须保持复位值

位 26 **OSR**：主时钟的过采样率 (Oversampling ratio for master clock)
0：主时钟频率 = F_{FS} x 256
1：主时钟频率 = F_{FS} x 512

位 25:20 **MCKDIV[5:0]**：主时钟分频器 (Master clock divider)。
这些位将由软件置 1 和清零。
000000：内核时钟输入 (sai_x_ker_ck) 1 分频。
否则，主时钟频率将根据*第 51.3.8 节：SAI 时钟发生器*中的公式计算。
音频模块为从模块时，这些位没有意义。
必须在音频模块禁止的情况下配置这些位。

位 19 **NOMCK**：无分频器 (No divider)
此位由软件置 1 和清零。
0：使能主时钟发生器。
1：禁止主时钟发生器。由 MCKDIV 控制的时钟分频器仍可用于生成位时钟。

位 18 保留，必须保持复位值

位 17 **DMAEN**：DMA 使能 (DMA enable)
此位由软件置 1 和清零。
0：禁止 DMA
1：使能 DMA
注：　在接收模式下，必须在 DMAEN 位置 1 前配置 MODE[1:0] 位，以避免 DMA 请求，原因是复位后音频模块将默认以发送模式工作。

位 16 **SAIEN**：音频模块使能 (Audio block enable)
该位由软件置 1。
要关闭音频模块，必须由应用软件将该位编程为 0 并轮询该位，直到该位读回 0，这表示该模块已完全禁止。将该位置 1 前，先检查该位是否已置 0，否则不会执行此使能命令。
该位可用于控制 SAI 音频模块的状态。如果在音频帧传输期间禁止该位，则正在进行的传输将完成并且单元在该音频帧传输结束时完全禁止。
0：禁止 SAIx 音频模块
1：使能 SAIx 音频模块。
注：　当 SAI 模块（A 或 B）配置为主模式时，SAI 模块输入中必须有时钟，然后才能将 SAIEN 位置 1。

位 15:14 保留，必须保持复位值

位 13 **OUTDRIV**：输出驱动 (Output drive)
此位由软件置 1 和清零。
0：当 SAIEN 置 1 时驱动音频模块输出
1：在该位置 1 后立即驱动音频模块输出。
注：　该位必须在音频模块配置后的使能前置 1。

位 12 **MONO**：单声道模式 (Mono mode)
此位由软件置 1 和清零。仅当 Slot 数为 2 时该位才有意义。如果选择了单声道模式，则当音频模块用作发送器时，Slot 0 的数据将复制到 Slot 1 上。在接收模式下，将丢弃 Slot 1 并仅存储从 Slot 0 接收的数据。更多详细信息，请参见*单声道/立体声模式*一节。
0：立体声模式。
1：单声道模式。

位 11:10 **SYNCEN[1:0]**：同步使能 (Synchronization enable)

这些位将由软件置 1 和清零。必须在音频子模块禁止的情况下配置这些位。

00：音频子模块处于异步模式。

01：音频子模块与另一个内部音频子模块同步。这种情况下，必须将该音频子模块配置为从模式。

10：音频子模块与外部 SAI 的嵌入式外设同步。这种情况下，应将音频子模块配置为从模式。

11：保留

注： 使能 SPDIF 模式后，应将音频子模块配置为异步。

位 9 **CKSTR**：时钟选通边沿 (Clock strobing edge)

此位由软件置 1 和清零。必须在音频模块禁止的情况下配置该位。SPDIF 音频协议下该位没有意义。

0：在 SCK 上升沿更改 SAI 生成的信号，而在 SCK 下降沿对 SAI 接收的信号进行采样。

1：在 SCK 下降沿更改 SAI 生成的信号，而在 SCK 上升沿对 SAI 接收的信号进行采样。

位 8 **LSBFIRST**：最低有效位优先 (Least significant bit first)

此位由软件置 1 和清零。必须在音频模块禁止的情况下配置该位。AC'97 音频协议下该位没有意义，原因是传输 AC'97 数据时，数据的 MSB 位优先。SPDIF 音频协议下该位没有意义，原因是传输 SPDIF 数据时，数据的 LSB 位优先。

0：优先传输数据的 MSB

1：优先传输数据的 LSB

位 7:5 **DS[2:0]**：数据大小 (Data size)

这些位将由软件置 1 和清零。选择 SPDIF 协议时（位 PRTCFG[1:0]），将忽略这些位，原因是在这种情况下帧和数据大小是固定的。通过 COMP[1:0] 位选择压扩模式时，将忽略 DS[1:0]，原因是算法已将数据大小固定为 8 位。

必须在音频模块禁止的情况下配置这些位。

000：保留

001：保留

010：8 位

011：10 位

100：16 位

101：20 位

110：24 位

111：32 位

位 4 保留，必须保持复位值

位 3:2 **PRTCFG[1:0]**：协议配置 (Protocol configuration)

这些位将由软件置 1 和清零。必须在音频模块禁止的情况下配置这些位。

00：自由协议。通过设置大部分配置寄存器位以及帧配置寄存器，自由协议允许用户使用音频模块这一强大的配置功能来处理特定的音频协议（如 I2S、LSB/MSB 对齐、TDM、PCM/DSP...）。

01：SPDIF 协议

10：AC'97 协议

11：保留

位 1:0 **MODE[1:0]**：SAIx 音频模块模式 (SAIx audio block mode)

这些位将由软件置 1 和清零。必须在 SAIx 音频模块禁止的情况下配置这些位。

00：主发送器

01：主接收器

10：从发送器

11：从接收器

注： 如果将音频模块配置为 SPDIF 模式，则将强制设置主发送模式 (MODE[1:0] = 00)。在主发送模式下，音频模块将立即开始生成 FS 和时钟。

51.5.3 配置寄存器 2 (SAI_ACR2/SAI_BCR2)

Configuration register 2

偏移地址：模块 A 为 0x008

偏移地址：模块 B 为 0x028

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
COMP[1:0]		CPL	MUTECNT[5:0]						MUTE VAL	MUTE	TRIS	F FLUSH	FTH		
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	w	rw	rw	rw

位 31:16 保留，必须保持复位值

位 15:14 **COMP[1:0]**：压扩模式 (Companding mode)。

这些位将由软件置 1 和清零。μ-Law 和 A-Law 算法是 CCITT G.711 建议的一部分，要使用何种补码类型取决于 *CPL 位*。

数据扩展还是数据压缩由 MODE[0] 位的状态确定。

如果将音频模块配置为发送器，则应用数据压缩。

如果将音频模块配置为接收器，则自动应用数据扩展。

更多详细信息，请参见*压扩模式*一节。

00：不支持压扩算法

01：保留。

10：μ-Law 算法

11：A-Law 算法

注： 仅当选择了 TDM 协议时才能使用压扩模式。

位 13 **CPL**：补码位 (Complement bit)。

此位由软件置 1 和清零。

该位定义用于压扩模式的补码类型。

0：1 的补码表示。

1：2 的补码表示。

注： 仅当压扩模式为 μ-Law 算法或 A-Law 算法时该位才有效。

位 12:7 **MUTECNT[5:0]**：静音计数器 (Mute counter)。

这些位将由软件置 1 和清零。这些位仅用于接收模式。

这些位中所设置的值将与接收模式下检测到的连续静音帧数量进行比较。当静音帧数量与该值相等时，MUTEDET 标志置 1，并且在 MUTEDETIE 位置 1 的情况下，还将生成中断。

更多详细信息，请参见*静音模式*一节。

位 6 **MUTEVAL**：静音值 (Mute value)。

该位由软件置 1 和清零。必须在使能音频模块 (SAIEN) 前写入。仅当音频模块用作发送器、Slot 数小于或等于 2 并且 MUTE 位置 1 时，该位才有意义。

如果声明了 2 个以上的 Slot，则无论 MUTEVAL 位的值为何，静音模式下发送的位值都将等于 0。

如果 Slot 数小于或等于 2 且 MUTEVAL = 1，则为每个 Slot 发送的 MUTE 值将是上一帧期间发送的值。

更多详细信息，请参见*静音模式*一节。

0：静音模式期间发送位值 0。

1：静音模式期间发送上一个值。

注： 该位对 SPDIF 音频模块无意义，从而也不使用。

位 5 **MUTE**：静音 (Mute)。

此位由软件置 1 和清零。仅当音频模块用作发送器时该位才有意义。Slot 数小于或等于 2 时，MUTE 值与 MUTEVAL 值相关；Slot 数大于 2 时，MUTE 值等于 0。

更多详细信息，请参见*静音模式*一节。

0：禁止静音模式。

1：使能静音模式。

注： 该位对 SPDIF 音频模块无意义，从而也不使用。

位 4 **TRIS**：数据线的三态管理 (Tristate management on data line)。

此位由软件置 1 和清零。仅当将音频模块配置为发送器时该位才有意义。当音频模块配置为 SPDIF 模式时不使用该位。应在 SAI 禁止时配置此位。

更多详细信息，请参见*无效 Slot 上的输出数据线管理*一节。

0：Slot 无效时，SD 输出线仍由 SAI 驱动。

1：SD 输出线将在上一个有效 Slot（下一个 Slot 无效）的最后一个数据位传输结束时释放（高阻态）。

位 3 **FFLUSH**：FIFO 刷新 (FIFO flush)。

该位由软件置 1， 始终读为 0。应在 SAI 禁止时配置此位。

0：禁止 FIFO 刷新。

1：FIFO 刷新。将此位编程为 1 可触发 FIFO 刷新。所有的内部 FIFO 指针（读和写）将清零。这种情况下，仍存留在 FIFO 中的数据丢失（发送或接收数据不会继续丢失）。刷新 SAI 前，必须禁止 DMA 数据流 / 中断。

位 2:0 **FTH**：FIFO 阈值 (FIFO threshold)。

此位由软件置 1 和清零。

000：FIFO 为空

001：¼ FIFO

010：½ FIFO

011：¾ FIFO

100：FIFO 已满

101：保留

110：保留

111：保留

51.5.4 帧配置寄存器 (SAI_AFRCR/SAI_BFRCR)

Frame configuration register

偏移地址：模块 A 为 0x00C

偏移地址：模块 B 为 0x02C

复位值：0x0000 0007

注： *该寄存器对于 AC'97 和 SPDIF 音频协议无意义*

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FSOFF	FSPOL	FSDEF
													rw	rw	r

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

位 31:19 保留，必须保持复位值

位 18 **FSOFF**：帧同步偏移 (Frame synchronization offset)。

此位由软件置 1 和清零。该位对 AC'97 或 SPDIF 音频模块配置无意义，从而也不使用。必须在音频模块禁止的情况下配置该位。

0：在 Slot 0 的第一位上使能 FS。

1：在 Slot 0 第一位的前一位上使能 FS。

位 17 **FSPOL**：帧同步极性 (Frame synchronization polarity)。

此位由软件置 1 和清零。该位用于配置 FS 信号上的 SOF 电平。该位对 AC'97 或 SPDIF 音频模块配置无意义，从而也不使用。

必须在音频模块禁止的情况下配置该位。

0：FS 为低电平有效（下降沿）

1：FS 为高电平有效（上升沿）

位 16 **FSDEF**：帧同步定义 (Frame synchronization definition)。

此位由软件置 1 和清零。

0：FS 信号为起始帧信号

1：FS 信号为 SOF 信号 + 通道识别信号

此位置 1 时，SAI_xSLOTR 寄存器中定义的 Slot 数必须为偶数。这意味着有半数 Slot 将用于左通道，其它 Slot 用于右通道（例如，对于 I2S 或 MSB/LSB 对齐等协议，该位必须置 1）。

此位对 AC'97 或 SPDIF 音频模块配置无意义，从而也不使用。必须在音频模块禁止的情况下配置该位。

位 15 保留，必须保持复位值

位 14:8 **FSALL[6:0]**：帧同步有效电平长度 (Frame synchronization active level length)。
这些位将由软件置 1 和清零。这些位用于指定音频帧中 FS 信号的有效电平长度，以位时钟数 (SCK) + 1 (FSALL[6:0] + 1) 表示这些位对 AC’97 或 SPDIF 音频模块配置无意义，从而也不使用。
必须在音频模块禁止的情况下配置这些位。

位 7:0 **FRL[7:0]**：帧长度 (Frame length)。
这些位将由软件置 1 和清零。这些位用于定义以 SCK 时钟周期数表示的音频帧长度：帧中的位数等于 FRL[7:0] + 1。
音频帧中发送的位数必须大于或等于 8，否则音频模块将出现操作异常。数据大小为 8 位且在 SAI_xSLOTR 寄存器的 NBSLOT[4:0] 中只定义了一个 Slot (NBSLOT[3:0] = 0000) 时便属于这种情况。
在主模式下，如果使用主时钟（MCLK_x 引脚上提供），则帧长度应为 8 到 256 之间的一个等于 2 的几次幂的数。不使用主时钟 (NOMCK = 1) 时，建议将帧长度编程为 8 到 256 之间的值。
这些位对 AC’97 或 SPDIF 音频模块配置无意义，从而也不使用。

51.5.5 Slot 寄存器 (SAI_ASLOTR/SAI_BSLOTR)

Slot register

偏移地址：模块 A 为 0x010

偏移地址：模块 B 为 0x030

复位值：0x0000 0000

注：　该寄存器对于 AC’97 和 SPDIF 音频协议无意义

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

位 31:16 **SLOTEN[15:0]**：Slot 使能 (Slot enable)。
这些位将由软件置 1 和清零。
每个 SLOTEN 位都对应于从 0 到 15 的一个 Slot 位置（最多 16 个 Slot）。
0：无效 Slot。
1：有效 Slot。
必须在音频模块禁止的情况下使能 Slot。
在 AC’97 或 SPDIF 模式下会忽略这些位。

位 15:12 保留，必须保持复位值

位 11:8 **NBSLOT[3:0]**：音频帧中的 Slot 数 (Number of slots in an audio frame)。
这些位将由软件置 1 和清零。
此位域中设置的值表示音频帧中的 Slot 数 + 1（包括无效 Slot 数）。Slot 数最大值为 16。
SAI_xFRCR 寄存器中的 FSDEF 位置 1 时，Slot 数应为偶数。
必须在音频模块禁止的情况下配置 Slot 数。
在 AC’97 或 SPDIF 模式下会忽略这些位。

位 7:6 **SLOTSZ[1:0]**：Slot 大小 (Slot size)

此位由软件置 1 和清零。

Slot 大小必须大于或等于数据大小。如果不满足该条件，SAI 的行为将不确定。

有关如何驱动 SD 线的信息，请参见*无效 Slot 上的输出数据线管理*一节。

必须在音频模块禁止的情况下配置这些位。

在 AC'97 或 SPDIF 模式下会忽略这些位。

00：Slot 大小与数据大小（在 SAI_xCR1 寄存器的 DS[3:0] 位中指定）相当。

01：16 位

10：32 位

11：保留

位 5 保留，必须保持复位值

位 4:0 **FBOFF[4:0]**：第一个位偏移 (First bit offset)

这些位将由软件置 1 和清零。

此位域中设置的值定义 Slot 中第一个数据传输位的位置。它表示一个偏移值。在发送模式下，此数据位域以外的位将强制清零。在接收模式下，将丢弃额外接收的位。

必须在音频模块禁止的情况下配置这些位。

在 AC'97 或 SPDIF 模式下会忽略这些位。

51.5.6 中断屏蔽寄存器 2 (SAI_AIM/SAI_BIM)

Interrupt mask register 2

偏移地址：模块 A 为 0x014

偏移地址：模块 B 为 0x034

复位值：0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LFSDETIE	AFSDETIE	CNRDYIE	FREQIE	WCKCFGIE	MUTEDETIE	OVRUDRIE
									rw	rw	rw	rw	rw	rw	rw

位 31:7 保留，必须保持复位值

位 6 **LFSDETIE**：帧同步滞后检测中断使能 (Late frame synchronization detection interrupt enable)。

此位由软件置 1 和清零。

0：禁止中断

1：使能中断

该位置 1 时，若 SAI_xSR 寄存器中的 LFSDET 位置 1，则生成中断。

该位对于 AC'97、SPDIF 模式无意义；若音频模块为主模块，该位也无意义。

位 5 **AFSDETIE**：帧同步提前检测中断使能 (Anticipated frame synchronization detection interrupt enable)。

此位由软件置 1 和清零。

0：禁止中断

1：使能中断

该位置 1 时，若 SAI_xSR 寄存器中的 AFSDET 位置 1，则生成中断。

该位对于 AC'97、SPDIF 模式无意义；若音频模块为主模块，该位也无意义。

位 4 **CNRDYIE**：编解码器未就绪中断使能 (Codec not ready interrupt enable) (AC’97)。
此位由软件置 1 和清零。
0：禁止中断
1：使能中断
若使能该中断，音频模块将在 AC’97 帧的 Slot 0 (tag0) 中检测连接到该线路的编解码器是否就绪。
如果未就绪，SAI_xSR 寄存器中的 CNRDY 标志置 1，并生成中断。
仅当通过 PRTCFG[1:0] 位选择了 AC’97 模式且音频模块用作接收器时，该位才有意义。

位 3 **FREQIE**：FIFO 请求中断使能 (FIFO request interrupt enable)。
此位由软件置 1 和清零。
0：禁止中断
1：使能中断
该位置 1 时，若 SAI_xSR 寄存器中的 FREQ 位置 1，则生成中断。
在接收模式下，必须在 FREQIE 位置 1 前配置 MODE 位，以避免寄生中断，原因是复位后音频模块将默认用作发送器。

位 2 **WCKCFGIE**：时钟配置错误中断使能 (Wrong clock configuration interrupt enable)。
此位由软件置 1 和清零。
0：禁止中断
1：使能中断
仅在将音频模块配置为主模式 (MODE[1] = 0) 且 NOMCK = 0 时才执行此位。
该位在 SAI_xSR 寄存器中的 WCKCFG 标志置 1 时生成中断。
注：　该位仅用于 TDM 模式，其它模式下没有意义。

位 1 **MUTEDETIE**：静音检测中断使能 (Mute detection interrupt enable)。
此位由软件置 1 和清零。
0：禁止中断
1：使能中断
该位置 1 时，若 SAI_ASR 寄存器中的 MUTEDET 位置 1，则生成中断。
仅当音频模块配置为以发送模式工作时该位才有意义。

位 0 **OVRUDRIE**：上溢/下溢中断使能 (Overrun/underrun interrupt enable)。
此位由软件置 1 和清零。
0：禁止中断
1：使能中断
该位置 1 时，若 SAI_ASR 寄存器中的 OVRUDR 位置 1，则生成中断。

51.5.7 状态寄存器 (SAI_ASR/SAI_BSR)

Status register

偏移地址：模块 A 为 0x018

偏移地址：模块 B 为 0x038

复位值：0x0000 0008

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LFSDET	AFSDET	CNRDY	FREQ	WCKCFG	MUTEDET	OVRUDR
									r	r	r	r	r	r	r

位 31:19 保留，必须保持复位值

位 18:16 **FLVL**：FIFO 阈值 (FIFO level threshold)。

该位为只读。FIFO 阈值标志只通过硬件管理，其设置取决于 SAI 模块的配置（发送器或接收器模式）。

如果将 SAI 模块配置为发送器：

000：FIFO 为空

001：FIFO <= ¼，但非空

010：¼ < FIFO <= ½

011：½ < FIFO <= ¾

100：¾ < FIFO，但未满

101：FIFO 已满

如果将 SAI 模块配置为接收器：

000：FIFO 为空

001：FIFO < ¼，但非空

010：¼ <= FIFO < ½

011：½ =< FIFO < ¾

100：¾ =< FIFO，但未满

101：FIFO 已满

位 15:7 保留，必须保持复位值

位 6 **LFSDET**：帧同步滞后检测 (Late frame synchronization detection)。

该位为只读。

0：无错误。

1：帧同步信号未在正确的时刻出现。

仅当音频模块配置为以从模式工作时，此标志才能置 1。

不适用于 AC’97 或 SPDIF 模式。

它可在 SAI_xIM 寄存器中的 LFSDETIE 位置 1 时生成中断。

在软件将 SAI_xCLRFR 寄存器中的 CLFSDET 位置 1 时清除该标志

位 5 **AFSDET**：帧同步提前检测 (Anticipated frame synchronization detection)。

该位为只读。

0：无错误。

1：提前检测到帧同步信号。

仅当音频模块配置为以从模式工作时，此标志才能置 1。

不适用于 AC’97 或 SPDIF 模式。

它可在 SAI_xIM 寄存器中的 AFSDETIE 位置 1 时生成中断。

在软件将 SAI_xCLRFR 寄存器中的 CAFSDET 位置 1 时清除该标志。

位 4 **CNRDY**：编解码器未就绪 (Codec not ready)。

该位为只读。

0：外部 AC’97 编解码器已就绪

1：外部 AC’97 编解码器未就绪

仅当在 SAI_xCR1 寄存器中选择了 AC’97 音频模块并且音频模块配置为接收器模式时，才使用该位。

它可在 SAI_xIM 寄存器中的 CNRDYIE 位置 1 时生成中断。

在软件将 SAI_xCLRFR 寄存器中的 CCNRDY 位置 1 时清除该标志。

位 3 **FREQ**：FIFO 请求 (FIFO request)。
该位为只读。
0：无 FIFO 请求。
1：FIFO 请求读取或写入 SAI_xDR。
请求内容取决于音频模块的配置：
– 如果模块配置为发送模式，则 FIFO 请求与向 SAI_xDR 中写入相关。
– 如果音频模块配置为接收模式，则 FIFO 请求与从 SAI_xDR 中读取相关。
此标志可在 SAI_xIM 寄存器中的 FREQIE 位置 1 时生成中断。

位 2 **WCKCFG**：时钟配置错误标志 (Wrong clock configuration flag)。
该位为只读。
0：时钟配置正确
1：时钟配置不符合*第 51.3.6 节：帧同步*中定义的帧长度规范（SAI_xFRCR 寄存器中 FRL[7:0] 位的配置）
该位仅在音频模块工作在主模式 (MODE[1] = 0) 下且 NOMCK = 0 时使用。
它可在 SAI_xIM 寄存器中的 WCKCFGIE 位置 1 时生成中断。
在软件将 SAI_xCLRFR 寄存器中的 CWCKCFG 位置 1 时清除该标志。

位 1 **MUTEDET**：静音检测 (Mute detection)。
该位为只读。
0：SD 输入线上未检测到 MUTE 值
1：在SD 输入线上检测到指定数量的连续音频帧中的 MUTE 值（0 值）
如果在指定音频帧的每个 Slot 或在一定数量（在 SAI_xCR2 寄存器中的 MUTECNT 位中设置）的连续音频帧中接收到连续的 0 值，则该标志置 1。
它可在 SAI_xIM 寄存器中的 MUTEDETIE 位置 1 时生成中断。
在软件将 SAI_xCLRFR 寄存器中的 CMUTEDET 位置 1 时清除该标志。

位 0 **OVRUDR**：上溢/下溢 (Overrun/underrun)。
该位为只读。
0：无上溢/下溢错误。
1：检测到上溢/下溢错误。
仅当音频模块分别被配置为接收器和发送器时才会发生上溢和下溢情况。
它可在 SAI_xIM 寄存器中的 OVRUDRIE 位置 1 时生成中断。
在软件将 SAI_xCLRFR 寄存器中的 COVRUDR 位置 1 时清除该标志。

51.5.8 清除标志寄存器 (SAI_ACLRFR/SAI_BCLRFR)

Clear flag register

偏移地址：模块 A 为 0x01C

偏移地址：模块 B 为 0x03C

复位值：0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CLFSDET	CAFSDET	CCNRDY	Res.	CWCKCFG	CMUTEDET	COVRUDR
									w	w	w		w	w	w

位 31:7 保留，必须保持复位值

位 6 **CLFSDET**：清除帧同步滞后检测标志 (Clear late frame synchronization detection flag)。
该位为只写位。
将此位编程为 1 可清除 SAI_xSR 寄存器中的 LFSDET 标志。
该位不适用于 AC'97 或 SPDIF 模式。
读取该位将始终返回值 0。

位 5 **CAFSDET**：清除帧同步提前检测标志 (Clear anticipated frame synchronization detection flag)。
该位为只写位。
将此位编程为 1 可清除 SAI_xSR 寄存器中的 AFSDET 标志。
不适用于 AC'97 或 SPDIF 模式。
读取该位将始终返回值 0。

位 4 **CCNRDY**：清除编解码器未就绪标志 (Clear codec not ready flag)。
该位为只写位。
将此位编程为 1 可清除 SAI_xSR 寄存器中的 CNRDY 标志。
仅当在 SAI_xCR1 寄存器中选择了 AC'97 音频协议时，才使用该位。
读取该位将始终返回值 0。

位 3 保留，必须保持复位值

位 2 **CWCKCFG**：清除时钟配置错误标志 (Clear wrong clock configuration flag)。
该位为只写位。
将此位编程为 1 可清除 SAI_xSR 寄存器中的 WCKCFG 标志。
仅当音频模块设置为主模块时 (MODE[1] = 0) 且 SAI_xCR1 寄存器中的 NOMCK = 0 时，才使用此位。
读取该位将始终返回值 0。

位 1 **CMUTEDET**：静音检测标志 (Mute detection flag)。
该位为只写位。
将此位编程为 1 可清除 SAI_xSR 寄存器中的 MUTEDET 标志。
读取该位将始终返回值 0。

位 0 **COVRUDR**：清除上溢/下溢标志 (Clear overrun/underrun)。
该位为只写位。
将此位编程为 1 可清除 SAI_xSR 寄存器中的 OVRUDR 标志。
读取该位将始终返回值 0。

51.5.9 数据寄存器 (SAI_ADR/SAI_BDR)

Data register

偏移地址：模块 A 为 0x020

偏移地址：模块 B 为 0x040

复位值：0x0000 0000

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

位 31:0 **DATA[31:0]**：数据 (Data)

若 FIFO 未满，写入该寄存器可向 FIFO 加载数据。

若 FIFO 非空，读取该寄存器可清空 FIFO 。

51.5.10 PDM 控制寄存器 (SAI_PDMCR)

PDM control register

偏移地址：0x0044

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	CKEN4 (1)	CKEN3 (1)	CKEN2 (1)	CKEN1 (1)	Res.	Res.	MICNBR[1:0] (1)		Res.	Res.	Res.	PDMEN
				rw	rw	rw	rw			rw	rw				rw

1. 建议不要在 PDMEN = 1 时配置这些位域

位 31:16 保留，必须保持复位值

位 15:12 保留，必须保持复位值

位 11 **CKEN4**：4 号比特流时钟使能 (Clock enable of bitstream clock number 4)

此位由软件置 1 和清零。

0：禁止 SAI_CK4 时钟

1：使能 SAI_CK4 时钟

位 10 **CKEN3**：3 号比特流时钟使能 (Clock enable of bitstream clock number 3)

此位由软件置 1 和清零。

0：禁止 SAI_CK3 时钟

1：使能 SAI_CK3 时钟

位 9 **CKEN2**：2 号比特流时钟使能 (Clock enable of bitstream clock number 2)
此位由软件置 1 和清零。
0：禁止 SAI_CK2 时钟
1：使能 SAI_CK2 时钟

位 8 **CKEN1**：1 号比特流时钟使能 (Clock enable of bitstream clock number 1)
此位由软件置 1 和清零。
0：禁止 SAI_CK1 时钟
1：使能 SAI_CK1 时钟

位 7:6 保留，必须保持复位值

位 5:4 **MICNBR**：麦克风数量 (Number of microphones)
此位由软件置 1 和清零。
0b00：配置有 2 个麦克风
0b01：配置有 4 个麦克风
0b10：配置有 6 个麦克风
0b11：配置有 8 个麦克风

位 3:1 保留，必须保持复位值

位 0 **PDMEN**：PDM 使能 (PDM enable)
此位由软件置 1 和清零。该位可用于控制 PDM 接口模块的状态。
在使能 PDM 接口之前，确保 SAI 已工作在 TDM 主模式下。
0：禁止 PDM 接口
1：使能 PDM 接口

51.5.11 PDM 延迟寄存器 (SAI_PDMDLY)

PDM delay register

偏移地址：0x0048

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	DLYM4R[2:0]			Res.	DLYM4L[2:0]			Res.	DLYM3R[2:0]			Res.	DLYM3L[2:0]		
	rw				rw				rw				rw		
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	DLYM2R[2:0]			Res.	DLYM2L[2:0]			Res.	DLYM1R[2:0]			Res.	DLYM1L[2:0]		
	rw				rw				rw				rw		

位 31 保留，必须保持复位值

位 30:28 **DLYM4R**：**第 4 对**的第二个麦克风的延迟线 (Delay line for second microphone of **pair 4**)
此位由软件置 1 和清零。
0b000：无延迟
0b001：延迟 1 个 T_{SAI_CK} 周期
0b010：延迟 2 个 T_{SAI_CK} 周期
...
0b111：延迟 7 个 T_{SAI_CK} 周期

该位域可实时更改。

位 27 保留，必须保持复位值

位 26:24 **DLYM4L**：第 4 对的第一个麦克风的延迟线 (Delay line for first microphone of pair 4)
此位由软件置 1 和清零。
0b000：无延迟
0b001：延迟 1 个 T_{SAI_CK} 周期
0b010：延迟 2 个 T_{SAI_CK} 周期
...
0b111：延迟 7 个 T_{SAI_CK} 周期

该位域可实时更改。

位 23 保留，必须保持复位值

位 22:20 **DLYM3R**：第 3 对的第二个麦克风的延迟线 (Delay line for second microphone of pair 3)
此位由软件置 1 和清零。
0b000：无延迟
0b001：延迟 1 个 T_{SAI_CK} 周期
0b010：延迟 2 个 T_{SAI_CK} 周期
...
0b111：延迟 7 个 T_{SAI_CK} 周期

该位域可实时更改。

位 19 保留，必须保持复位值

位 18:16 **DLYM3L**：第 3 对的第一个麦克风的延迟线 (Delay line for first microphone of pair 3)
此位由软件置 1 和清零。
0b000：无延迟
0b001：延迟 1 个 T_{SAI_CK} 周期
0b010：延迟 2 个 T_{SAI_CK} 周期
...
0b111：延迟 7 个 T_{SAI_CK} 周期

该位域可实时更改。

位 15 保留，必须保持复位值

位 14:12 **DLYM2R**：第 2 对的第二个麦克风的延迟线 (Delay line for second microphone of pair 2)
此位由软件置 1 和清零。
0b000：无延迟
0b001：延迟 1 个 T_{SAI_CK} 周期
0b010：延迟 2 个 T_{SAI_CK} 周期
...
0b111：延迟 7 个 T_{SAI_CK} 周期

该位域可实时更改。

位 11 保留，必须保持复位值

位 10:8 **DLYM2L**：第 2 对的第一个麦克风的延迟线 (Delay line for first microphone of pair 2)
此位由软件置 1 和清零。
0b000：无延迟
0b001：延迟 1 个 T_{SAI_CK} 周期
0b010：延迟 2 个 T_{SAI_CK} 周期
...
0b111：延迟 7 个 T_{SAI_CK} 周期

该位域可实时更改。

位 7 保留，必须保持复位值

位 6:4 **DLYM1R**：第 1 对的第二个麦克风的延迟线调整 (Delay line adjust for second microphone of pair 1)
此位由软件置 1 和清零。
0b000：无延迟
0b001：延迟 1 个 T_{SAI_CK} 周期
0b010：延迟 2 个 T_{SAI_CK} 周期
...
0b111：延迟 7 个 T_{SAI_CK} 周期

该位域可实时更改。

位 3 保留，必须保持复位值

位 2:0 **DLYM1L**：第 1 对的第一个麦克风的延迟线调整 (Delay line adjust for first microphone of pair 1)
此位由软件置 1 和清零。
0b000：无延迟
0b001：延迟 1 个 T_{SAI_CK} 周期
0b010：延迟 2 个 T_{SAI_CK} 周期
...
0b111：延迟 7 个 T_{SAI_CK} 周期

该位域可实时更改。

51.5.12 SAI 寄存器映射

下表对 SAI 寄存器进行了汇总。

表 406. SAI 寄存器映射和复位值

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x0000	SAI_GCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SYNCOUT[1:0]		Res.	Res.	SYNCIN[1:0]	
	Reset value																											0	0			0	0

表 406. SAI 寄存器映射和复位值（续）

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x0004 或 0x0024	SAI_xCR1	Res.	Res.	Res.	Res.	Res.	OSR	MCKDIV[3:0]						NOMCK	Res.	DMAEN	SAIEN	Res.	Res.	OUTDRIV	MONO	SYNCEN[1:0]		CKSTR	LSBFIRST	DS[2:0]			Res.	PRTCFG[1:0]		MODE[1:0]	
	Reset value						0	0	0	0	0	0	0	0		0	0			0	0	0	0	0	0	0	1	0		0	0	0	0
0x0008 或 0x0028	SAI_xCR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	COMP[1:0]		CPL	MUTECN[5:0]							MUTE VAL	MUTE	TRIS	FFLUS	FTH			
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x000C 或 0x002C	SAI_xFRCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FSOFF	FSPOL	FSDEF	Res.	FSALL[6:0]							FRL[7:0]							
	Reset value														0	0	0		0	0	0	0	0	0	0	0	0	0	0	0	1	1	1
0x0010 或 0x0030	SAI_xSLOTR	SLOTEN[15:0]																Res.	Res.	Res.	Res.	NBSLOT[3:0]				SLOTSZ[1:0]		Res.	FBOFF[4:0]				
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0					0	0	0	0	0	0		0	0	0	0	0
0x0014 或 0x0034	SAI_xIM	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LFSDET	AFSDETIE	CNRDYIE	FREQIE	WCKCFG	MUTEDET	OVRUDRIE
	Reset value																										0	0	0	0	0	0	0
0x0018 或 0x0038	SAI_xSR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FLVL[2:0]			Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LFSDET	AFSDET	CNRDY	FREQ	WCKCFG	MUTEDET	OVRUDR
	Reset value														0	0	0										0	0	0	0	1	0	0
0x001C 或 0x003C	SAI_xCLRFR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LFSDET	CAFSDET	CNRDY	Res.	WCKCFG	MUTEDET	OVRUDR
	Reset value																										0	0	0		0	0	0
0x0020 或 0x0040	SAI_xDR	DATA[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0044	SAI_PDMCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CKEN4	CKEN3	CKEN2	CKEN1	Res.	Res.	MICNBR[1:0]		Res.	Res.	Res.	PDMEN
	Reset value																					0	0	0	0			0	0				0
0x0048	SAI_PDMDLY	Res.	DLYM4R[2:0]			Res.	DLYM4L[2:0]			Res.	DLYM3R[2:0]			Res.	DLYM3L[2:0]			Res.	DLYM2R[2:0]			Res.	DLYM2L[2:0]			Res.	DLYM1R[2:0]			Res.	DLYM1L[2:0]		
	Reset value		0	0	0		0	0	0		0	0	0		0	0	0		0	0	0		0	0	0		0	0	0		0	0	0

有关寄存器边界地址的信息，请参见*第 100 页的第 2.2.2 节*。

52 SPDIF 接收器接口 (SPDIFRX)

52.1 SPDIFRX 接口简介

SPDIFRX 接口处理 S/PDIF 音频协议。

52.2 SPDIFRX 主要特性

- 提供多达 4 路输入
- 自动符号率检测
- 最大符号率：12.288 MHz
- 支持 8 kHz 到 192 kHz 的立体声
- 支持 IEC-60958 和 IEC-61937 音频标准，消费类应用
- 可在 S/PDIF 流内插入 SOPD B、M 和 W
- 奇偶校验位管理
- 使用 DMA 通信进行音频采样
- 对于控制和用户信息采用 DMA 通信
- 中断功能

52.3 SPDIFRX 功能说明

SPDIFRX 外设用于接收符合 IEC-60958 和 EC-61937 标准的 S/PDIF 流。这些标准支持高采样率的简单立体声以及压缩的多通道环绕声，例如由 Dolby 或 DTS 定义的音频。

接收器提供所有必要的功能来检测符号率并解码传入的数据。这使用户可以使用专用通路和通道信息来简化接口的处理操作。*图 659* 给出了简化框图。

SPDIFRX_DC 模块负责解码从 SPDIFRX_IN[4:1] 输入接收的 S/PDIF 数据流。该模块重新采样传入的信号，解码曼彻斯特数据流，并识别帧、子帧和块元素。它传送到 REG_IF 部分、解码数据和相关的状态标志。

该外设可通过 APB1 总线完全控制并且能够处理两个 DMA 通道：

- 一个专用于传输音频采样的 DMA 通道
- 一个专用于传输 IEC60958 通道状态和用户信息的 DMA 通道

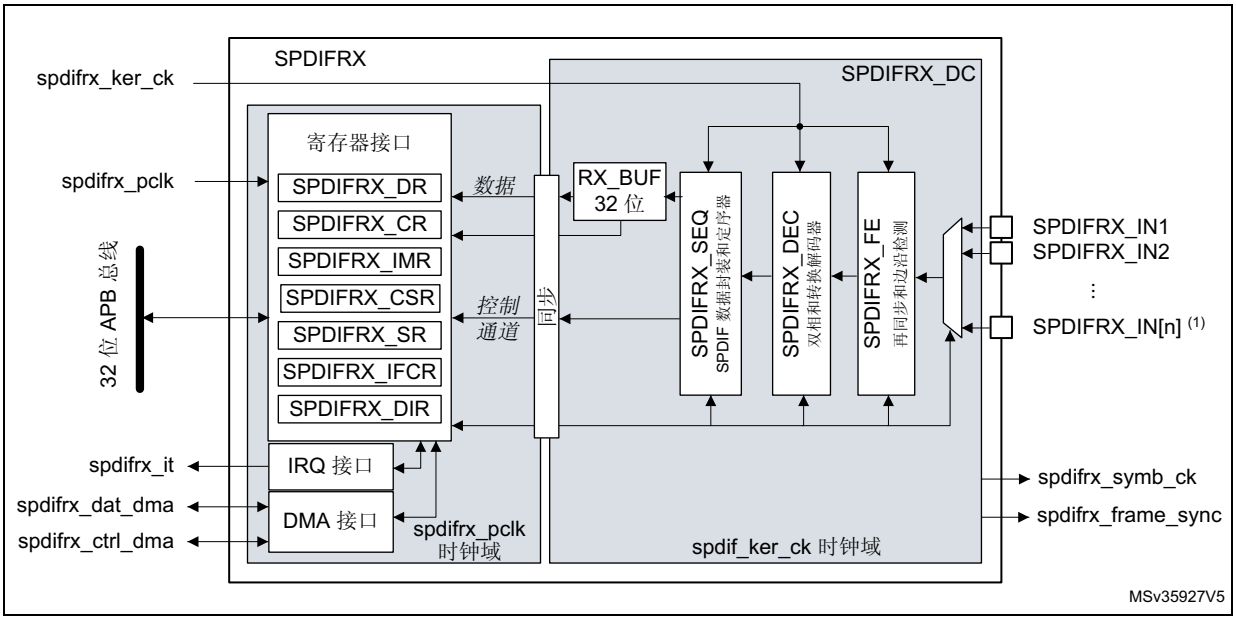
此外，还提供了中断服务，既可用作 DMA 的复用功能，也可用来指示外设的错误或关键状态。

SPDIFRX 还提供名为 **spdifrx_frame_sync** 的信号，该信号会在每次检测到子帧的报头时翻转。因此占空比为 50%，频率等于帧速率。

该信号可与定时器事件相连，以计算频率偏移。

此外，SPDIFRX 还提供了一个以符号率翻转的信号，名为 **spdifrx_symb_ck**。

图 659. SPDIFRX 框图



1. “n”固定为 4

52.3.1 SPDIFRX 引脚和内部信号

表 407 列出了 SPDIFRX 内部输入/输出信号，*表 408* 列出了 SPDIFRX 引脚（复用功能）。

表 407. SPDIFRX 内部输入/输出信号

信号名称	信号类型	说明
spdifrx_ker_ck	数字输入	SPDIFRX 内核时钟
spdifrx_pclk	数字输入	SPDIFRX 寄存器接口时钟
spdifrx_it	数字输出	SPDIFRX 全局中断
spdifrx_dat_dma	数字输入/输出	SPDIFRX DMA 数据传输请求（和应答）
spdifrx_ctrl_dma	数字输入/输出	SPDIFRX DMA 通道状态和用户信息传输请求（和应答）
spdifrx_frame_sync	数字输出	SPDIFRX 帧速率同步信号
spdifrx_symb_ck	数字输出	SPDIFRX 通道符号时钟

表 408. SPDIFRX 引脚

信号名称	信号类型	说明
SPDIFRX_IN1	数字输入	S/PDIF 信号输入 1
SPDIFRX_IN2	数字输入	S/PDIF 信号输入 2
SPDIFRX_IN3	数字输入	S/PDIF 信号输入 3
SPDIFRX_IN4	数字输入	S/PDIF 信号输入 4

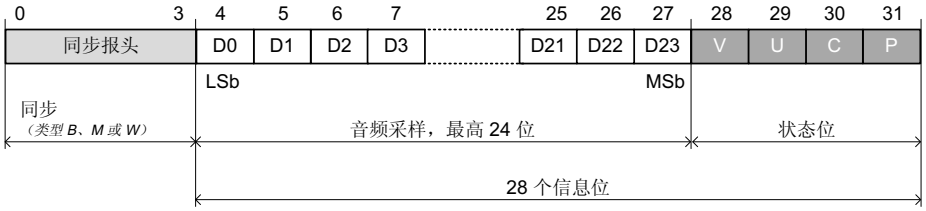
52.3.2 S/PDIF 协议 (IEC-60958)

S/PDIF 块

一个 S/PDIF 帧由两个子帧组成（见*图 661*）。每个子帧包含 32 位（或时隙）：

- 位 0 到位 3 包含同步报头之一。
- 位 4 到位 27 包含以线性 2 的补码表示的音频采样字。最高有效位 (MSB) 为位 27。使用 20 位编码范围时，位 8 到位 27 包含音频采样字，其中位 8 为 LSB。
- 位 28（有效性位“V”）表示数据是否有效（例如转换为模拟数据）
- 位 29（用户数据位“U”）包含用户数据信息，如光盘的音轨编号。
- 位 30（通道状态位“C”）包含通道状态信息，如采样率和复制保护。
- 位 31（奇偶检验位“P”）包含奇偶校验位，这样位 4 到位 31（含）将包含偶数个“1”和偶数个“0”（偶校验）。

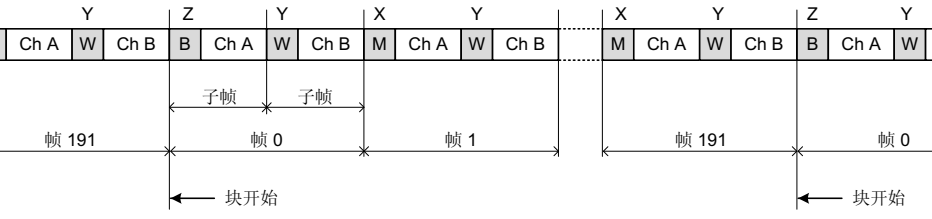
图 660. S/PDIF 子帧格式



对于线性编码音频应用，第一个子帧（立体声操作中的左声道或“A”通道以及单声道操作中的主通道）通常以报头“M”开始。但是，报头每 192 帧切换为报头“B”一次，以识别用于组织通道状态和用户信息的块结构的开始。第二个子帧（立体声操作中的右声道或“B”通道以及单声道操作中的辅助通道）始终以报头“W”开始。

一个 S/PDIF 块包含 192 对 32 位子帧。

图 661. S/PDIF 块格式

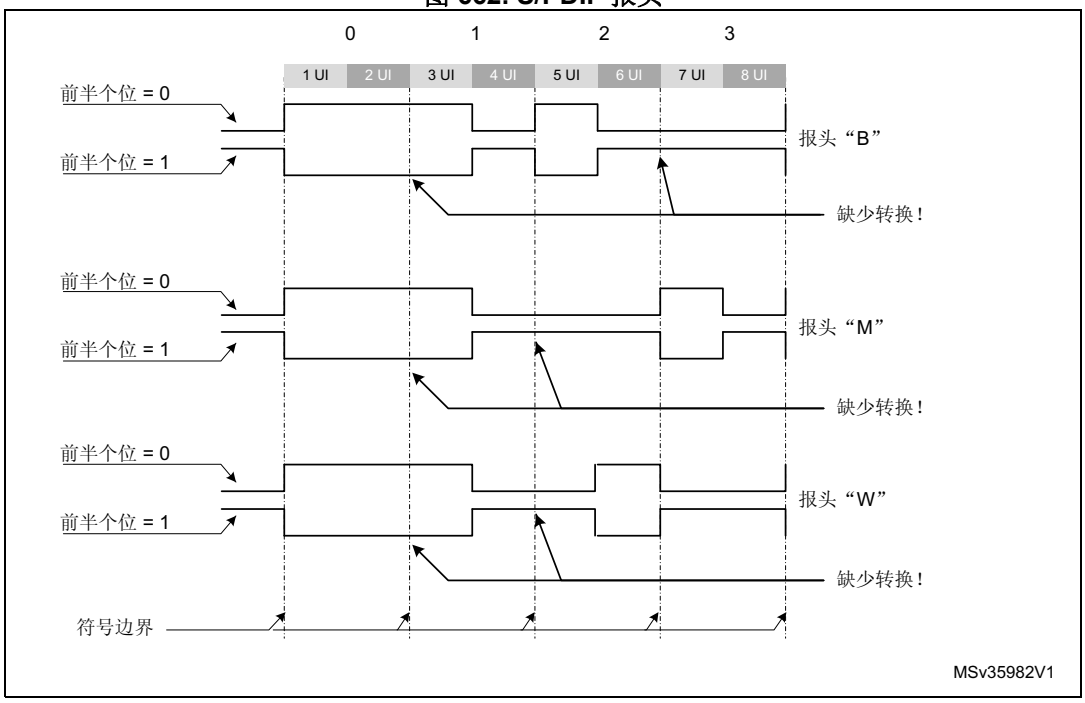


注
出于历史原因，报头“B”、“M”和“W”在专业应用中使用时分别称为“Z”、“X”以及“Y”。

同步报头

报头模式反转或者不符合前半位值。使能第一个帧的第一个“B”报头的传输前，此前半位值为线路的电平。对于其它报头，此前半位值为之前子帧的奇偶校验位的第二个半位。*图 662* 中介绍了报头模式 B、M 和 W。

图 662. S/PDIF 报头



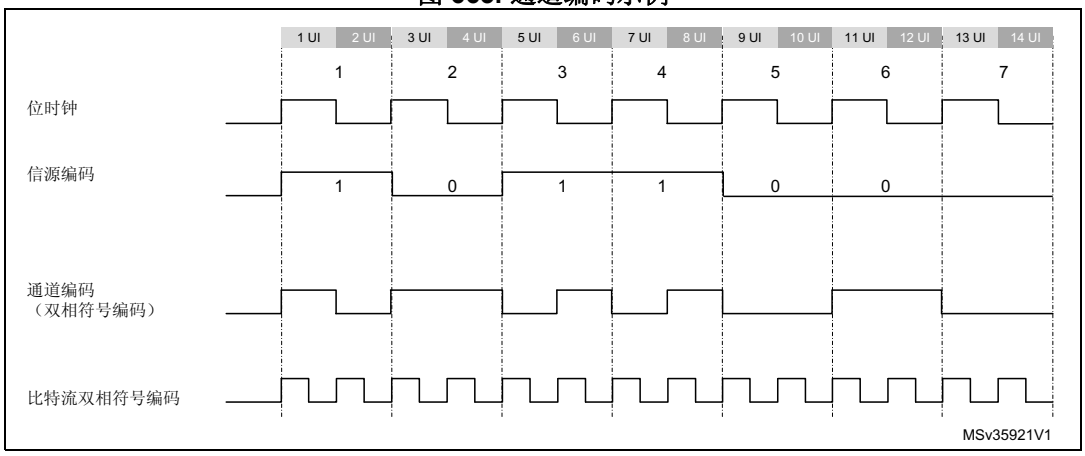
信息位编码

为最大程度减小传输线路上的直流分量值，并加快时钟从数据流中恢复的速度，位 4 到位 31 采用双相符号编码。

要传输的各个位通过由两个连续二进制状态构成的符号表示。符号的第一个状态始终不同于前一个符号的第二个状态。如果要传输的位为逻辑 0，则符号的第二个状态与第一个状态相同。但如果该位为逻辑 1，则两个状态不同。在 IEC-60958 规范中，这些状态称为“UI”（单位间隔）。

首先传输 24 个数据位中的 LSB。

图 663. 通道编码示例



52.3.3 SPDIFRX 解码器 (SPDIFRX_DC)

主要原则

SPDIFRX 用来解码 S/PDIF 数据流的技术以测量两个连续边沿之间时间间隔为基础。可在 S/PDIF 数据流中找到三类时间间隔：

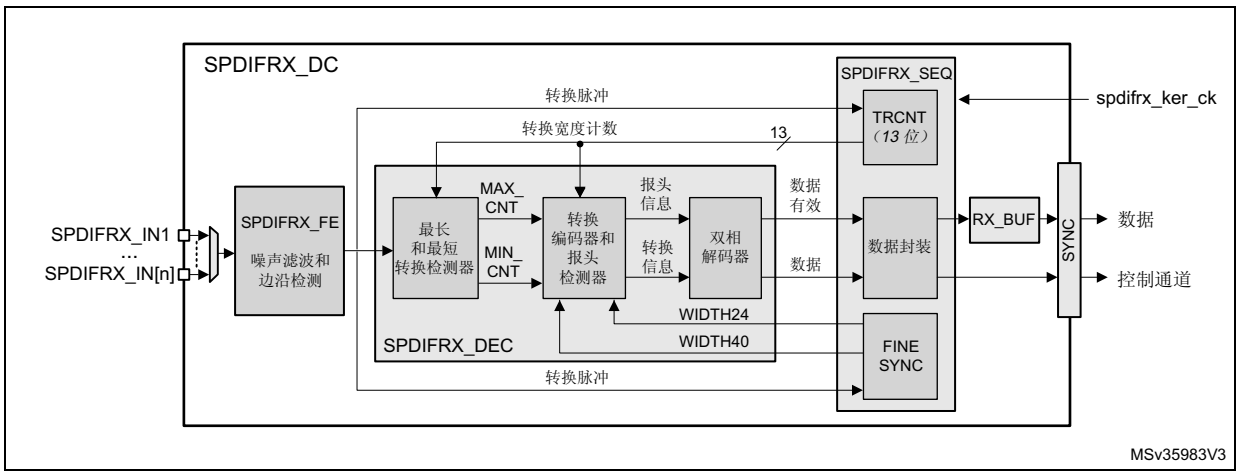
- 长时间间隔，持续 3 个 UI，称为 TL。仅在报头期间出现。
- 中时间间隔，持续 2 个 UI，称为 TM。在一些报头或信息域中出现。
- 短时间间隔，持续 1 个 UI，称为 TS。在一些报头或信息域中出现。

SPDIFRX_DC 模块负责解码接收到的 S/PDIF 数据流。它负责以下功能：

- 对传入信号进行重新采样和滤波
- 估算时间间隔
- 估算符号率和同步性
- 解码串行数据并验证完整性
- 检测块和子帧报头
- 连续跟踪符号率

图 664 详细说明了 SPDIFRX 解码器。

图 664. SPDIFRX 解码器

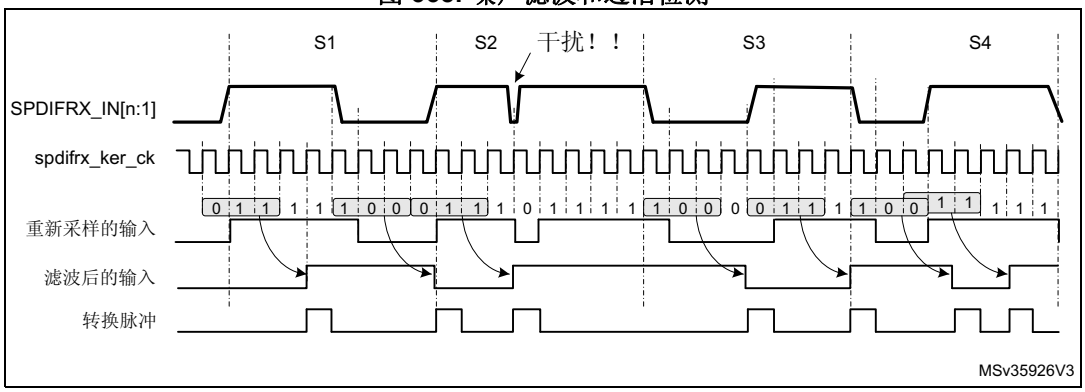


噪声滤波和上升/下降沿检测

使用 spdifrx_ker_ck 时钟（采集时钟）重新采样所选的 SPDIFRX_IN 上接收到的 S/PDIF 信号。采用简单滤波来去除杂波。通过检测边沿转换的阶段执行此过程。边沿转换的检测步骤如下：

- 当采样到序列 0 后跟两个 1 时，检测到上升沿。
- 当采样到序列 1 后跟两个 0 时，检测到下降沿。
- 在上升沿后，预期为下降沿序列。
- 在下降沿后，预期为上升沿序列。

图 665. 噪声滤波和边沿检测



最长和最短转换检测器

最长和最短转换检测器模块可检测两次转换间的最大 (MAX_CNT) 和最小 (MIN_CNT) 持续时间。TRCNT 计数器用于测量时间间隔的持续时间。它由 spdifrx_ker_ck 时钟信号驱动。对于每个转换脉冲，都会存储计数器值并复位计数器以重新开始计数。

最大持续时间通常出现在报头周期内。最大持续时间将作为 MAX_CNT 发出。最小持续时间将作为 MIN_CNT 发出。

当转换定时器过期时，最长和最短转换的搜索将停止。转换定时器类似于每转换 70 次传入信号就触发一次的看门狗定时器。请注意，计数 70 次转换可确保延迟略长于一个子帧。

请注意，当 TRCNT 因两个脉冲间的时间间隔过长而发生上溢时，SPDIFRX 将停止并且 SPDIFRX_SR 寄存器的标志 TERR 将置 1。

转换编码器和报头检测器

转换编码器和报头检测器块接收 MAX_CNT 和 MIN_CNT。它还接收 TRCNT 计数器中的当前转换宽度（请参见）。*图 664* 该块通过将当前转换宽度与两个不同阈值（TH_{HI} 和 TH_{LO}）进行比较来编码当前转换宽度。

- 如果当前转换宽度小于 (TH_{LO} - 1)，则接收到的数据为数据位“1”的一半，并编码为 TS。
- 如果当前转换宽度大于 (TH_{LO} - 1) 且小于 TH_{HI}，则接收到的数据为数据位“0”，并编码为 TM。
- 如果当前转换宽度大于 TH_{HI}，则接收到的数据为报头的长脉冲，并编码为 TL。
- 否则将生成错误代码（FERR 标志置 1）。

使用两种不同方法来详细阐述阈值 TH_{HI} 和 TH_{LO}。

如果外设正在执行其初始同步（“粗同步”），则阈值的计算方法如下：

- TH_{LO} = MAX_CNT / 2。
- TH_{HI} = MIN_CNT + MAX_CNT / 2。

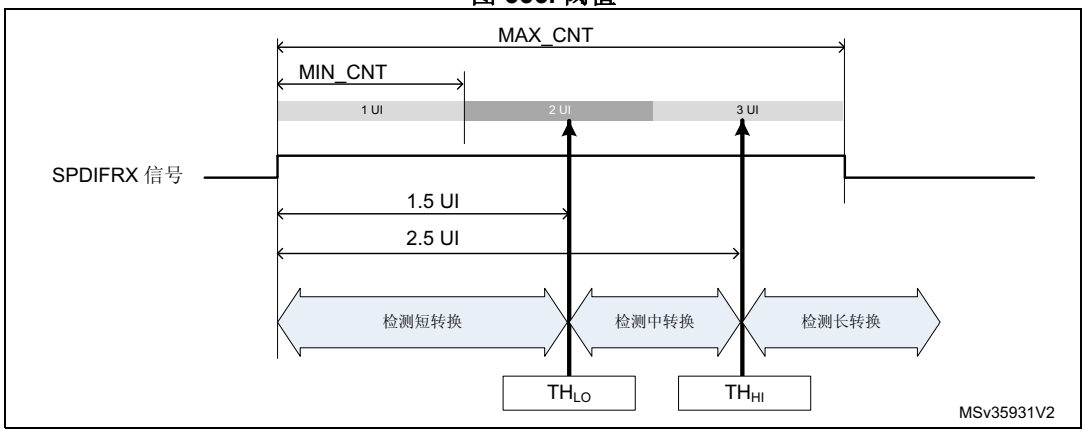
“粗同步”完成后，SPDIFRX 将使用更精确的参考来详细阐述阈值。SPDIFRX 将测量 24 个信号 (WIDTH24) 的长度来定义 TH_{LO}，测量 40 个信号 (WIDTH40) 的长度来定义 TH_{HI}。TH_{HI} 和 TH_{LO} 的计算方法如下：

- TH_{LO} = (WIDTH24) / 32
- TH_{HI} = (WIDTH40) / 32

第二个同步阶段称为“精同步”。有关更多信息，请参见*图 668*。

如后面的图所示，理想情况下 TH_{LO} 等于 1.5 个 UI，TH_{HI} 等于 2.5 个 UI。

图 666. 阈值



报头检测器检查特定序列的四个连续转换来确定它们是否构成报头的一部分。假设 TRANS0、TRANS1、TRANS2 和 TRANS3 表示四个连续的转换，编码方式如上面所述。*表 409* 给出了这四个转换的值以构成报头。缺少该模式表示这些转换构成子帧中数据的一部分并且双相解码器将对它们进行解码。

表 409. 报头的转换序列

报头类型	双相数据模式	TRANS3	TRANS2	TRANS1	TRANS0
报头 B	11101000	TL	TS	TS	TL
报头 M	11100010	TL	TL	TS	TS
报头 W	11100100	TL	TM	TS	TM

双相解码器

双相解码器使用由**转换编码器和报头检测器**块提供的转换信息来解码输入的双相符号数据流。首先，它等待报头检测信息。在报头检测后，它将解码以下转换信息：

- 如果传入的转换信息为 TM，则解码为“0”。
- 两个连续的 TS 解码为“1”。
- 任何其它转换序列将生成错误信号（FERR 置 1）。

通过这种方式解码 28 个数据位后，此模块将查找紧跟的报头数据。如果新报头与预期不同，则该模块将生成一个错误信号（FERR 置 1）。有关错误标志的更多信息，请参见*第 52.3.9 节：接收错误*。

数据封装

该模块负责解码 IEC-60958 帧和块。它还处理向 RX_BUF 或 SPDIFRX_CSR 寄存器写入数据。

52.3.4 SPDIFRX 对时钟偏差的容差

SPDIFRX 对时钟偏差的容差取决于一个位时隙中的采样时钟周期数。spdifrx_ker_ck 越快，接收越稳定。spdifrx_ker_ck 频率和信号率间的比率必须至少为 11。

有两种现象（至少！）会降低接收质量：

- 反映两个连续转换之间长度差异的周期间抖动。
- 反映周期间抖动积累效果的长期抖动。它可被看作低频符号调制。

52.3.5 SPDIFRX 同步

将 SPDIFRXEN 设置为 0b01 或 0b11 时，同步阶段开始。*图 667* 给出了同步过程。

如果 SPDIFRX_CR 寄存器的 WFA 位置 1，则在开始同步过程前，外设必须首先检测所选 SPDIFRX_IN 线路上的活动。通过检测所选 SPDIFRX_IN 上的四个转换来执行活动检测。外设保持该状态，直到未检测到转换。因为仅当所选 SPDIFRX_IN 输入上出现活动时，IP 才切换为粗同步模式，因此该功能在避免同步错误时特别有用。更多信息，请参见*第 52.4 节：编程步骤*。
用户仍然可以通过将 SPDIFRXEN 设置为 0 来使 SPDIFRX 进入 STATE_IDLE。如果 WFA 设置为 0，外设将直接启动粗同步而不检查活动。

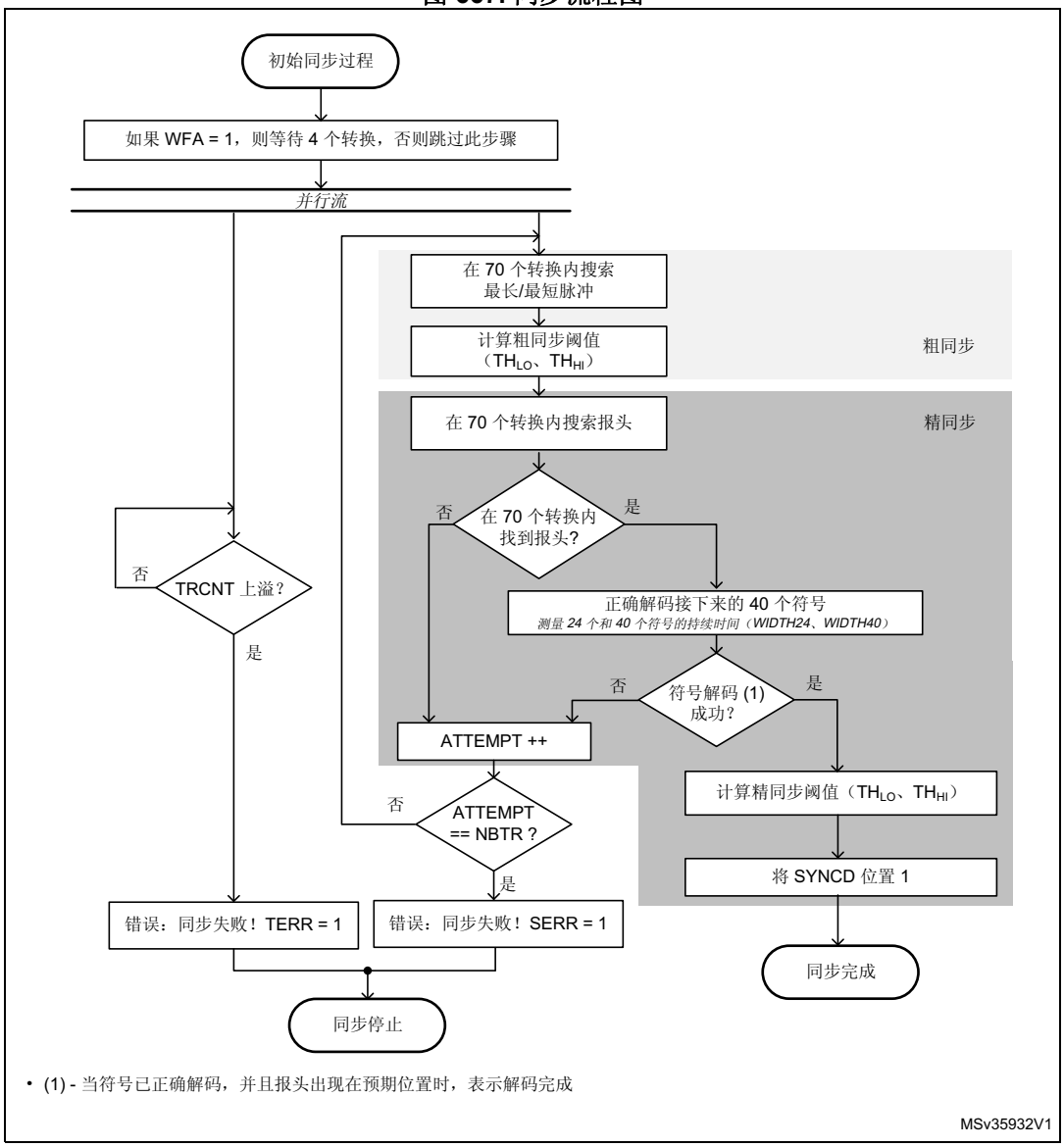
下一步将初步估算阈值（粗同步），以便执行精细同步（精同步）。由于 SPDIFRX 线路上的干扰，第一次可能无法正确执行该过程。为此，用户可在将 SERR 错误标志置 1 前编程允许的重试次数 (NBTR)。
当 SPDIFRX 能够正确测量 24 个和 40 个连续符号的持续时间并完成精同步时，阈值将更新并且标志 SYNCD 置 1。有关更多信息，请参见*转换编码器和报头检测器*一节。

可以检测到两类错误：

- TRCNT 的上溢，这通常意味着输入线路上没有有效的 S/PDIF 数据流。该上溢由 TERR 标志指示。
- 重试次数达到了设定的值。这意味着 S/PDIF 信号上存在较强的抖动。该错误由 SERR 标志指示。

完成第一次精同步后，如果检测到下一个“B”报头，将开始接收通道状态 (C) 和用户数据 (U)（请参见*图 671*）。随后，用户可通过 SPDIFRX_CSR 寄存器读取 IEC-60958 C 和 U 位。根据此信息，用户可以为 DRFMT 和 RXSTEO 选择合适的设置。例如，如果用户检测到当前音频流传输编码数据，则可在开始数据接收之前将 RXSTEO 设置为 0，并将 DRFMT 设置为 0b10。请注意，当 SPDIFRXEN = 0b11 时，不能修改 DRFMT 和 RXSTEO。如果 SPDIFRXEN 已经设置为 0b11，则忽略对这些位域的写操作，尽管可使用导致 SPDIFRXEN 变为 0b11 的相同写指令来修改这些位域。
然后在开始保存音频采样前，SPDIFRX 将等待 SPDIFRXEN = 0b11 和“B”报头。

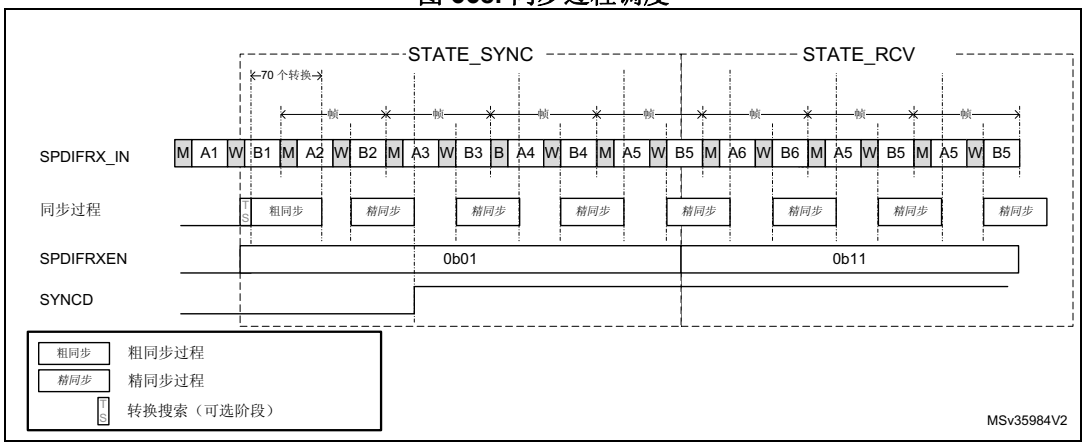
图 667. 同步流程图



有关 TRCNT 上溢的更多信息，请参见*帧结构和同步错误*。

如*图 668* 中所示，每一帧都将重新触发精同步过程来更新阈值，以便连续跟踪 S/PDIF 同步。

图 668. 同步过程调度



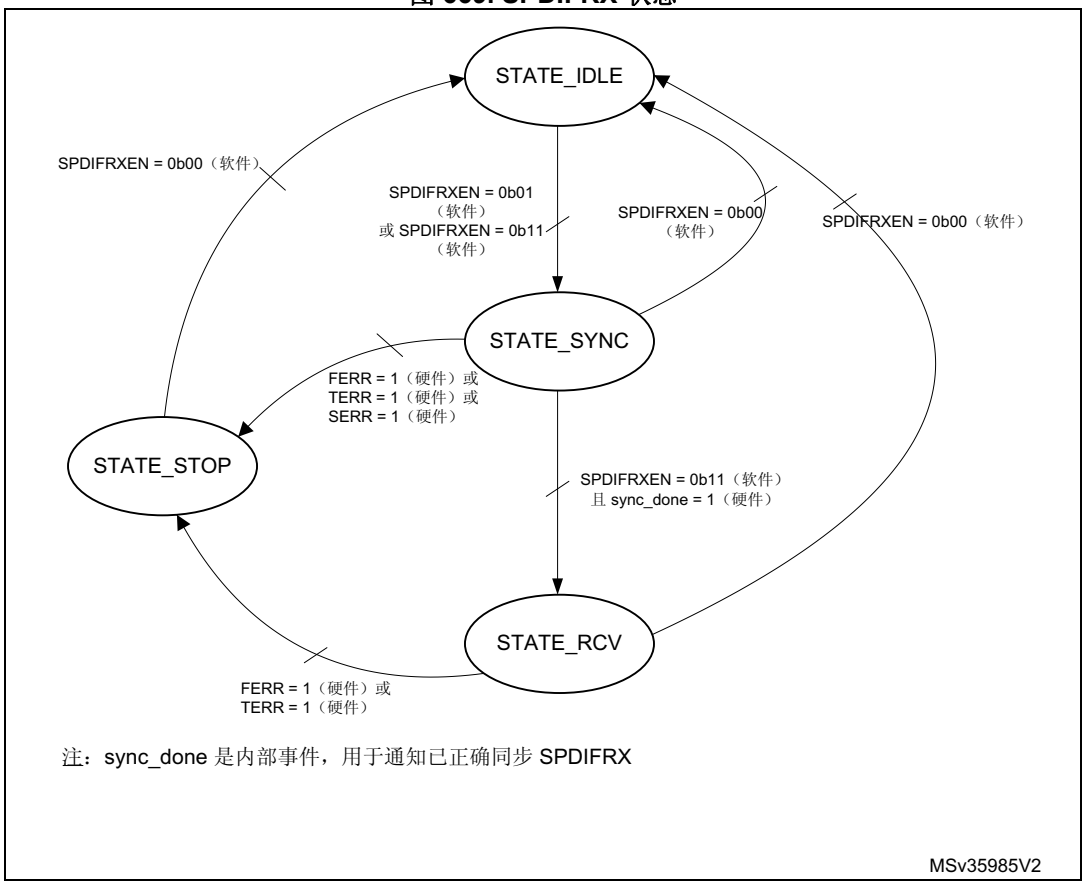
52.3.6 SPDIFRX 处理

软件可以通过 SPDIFRXEN 位域控制 SPDIFRX 的状态。SPDIFRX 可进入以下状态之一：

- STATE_IDLE：
  禁止外设，spdifrx_ker_ck 域复位。spdifrx_pclk 域功能正常。
- STATE_SYNC：
  外设与数据流同步，阈值定期更新，可通过 DMA 的中断读取用户和通道状态。音频采样未提供给接收缓冲区。
- STATE_RCV：
  外设与数据流同步，阈值定期更新，可通过中断或 DMA 通道读取用户、通道状态和音频采样。当 SPDIFRXEN 变为 0b11 时，SPDIFRX 将在开始保存音频采样前一直等待“B”报头。
- STOP_STATE：
  外设不再同步，用户、通道状态和音频采样的接收将停止。预期软件将重启 SPDIFRX。

图 669 给出了 SPDIFRX 可能的状态以及如何从一个状态转换为其它状态。受软件控制的位后跟说明“(SW)”，受 IP 控制的位后跟说明“(HW)”。

图 669. SPDIFRX 状态



当 SPDIFRX 处于 STATE_IDLE 时：

- 通过将 SPDIFRXEN 设置为 0b01 或 0b11，软件可转换为 STATE_SYNC。

当 SPDIFRX 处于 STATE_SYNC 时：

- 如果同步失败或者接收的数据未正确解码且无法在不进行再同步的情况下恢复（FERR 或 SERR 或 TERR = 1），则 SPDIFRX 将进入 STATE_STOP 状态并且等待软件应答。
- 当同步阶段完成时，如果 SPDIFRXEN = 0b01，外设将保持该状态。
- 软件可随时将 SPDIFRXEN 设置为 0，随后 SPDIFRX 立刻返回至 STATE_IDLE。如果 DMA 传输正在进行，则将正确完成。
- 如果 SPDIFRXEN = 0b11 且 SYNCD = 1，则 SPDIFRX 进入 STATE_RCV。

当 SPDIFRX 处于 STATE_RCV 时：

- 如果接收的数据未正确解码且无法在不进行再同步的情况下恢复（FERR 或 SERR 或 TERR = 1），则 SPDIFRX 将进入 STATE_STOP 状态并且等待软件应答。
- 软件可随时将 SPDIFRXEN 设置为 0，随后 SPDIFRX 立刻返回至 STATE_IDLE。如果 DMA 传输正在进行，则将正确完成。

当 SPDIFRX 处于 STATE_STOP 时：

- SPDIFRX 停止接收和同步，并等待软件将 SPDIFRXEN 位设置为 0，以清零错误标志。

当 SPDIFRXEN 设置为 0 时，IP 被禁止，这意味着所有状态机被复位，并且 RX_BUF 被刷新。还要注意，标志 FERR、SERR 和 TERR 也会复位。

52.3.7 数据接收管理

SPDIFRX 为音频采样接收提供了双缓冲区。32 位缓冲区位于 spdifrx_ker_ck 时钟域 (RX_BUF) 和 SPDIFRX_DR 寄存器中。如果 SPDIFRX_DR 为空，则包含在 RX_BUF 中的有效数据将立刻传输至 SPDIFRX_DR。

当满足以下两个条件时，包含在 RX_BUF 中的有效数据将传输至 SPDIFRX_DR：

- 检测到奇偶校验位 (P) 和下一个报头之间的转换（这表示字已经完全接收到）。
- SPDIFRX_DR 为空。

双字缓冲区为延迟限制提供了更高的灵活性。

允许的最大延迟为 T_{SAMPLE} - $2T_{PCLK}$ - $2T_{spdifrx_ker_ck}$

其中，T_{SAMPLE} 是接收的立体声音频采样的音频采样率，T_{PCLK} 是 spdifrx_pclk 时钟的周期，$T_{spdifrx_ker_ck}$ 是 spdifrx_ker_ck 时钟的周期。

SPDIFRX 可以使用 DMA（pdifrx_dat_dma 和 spdifrx_ctrl_dma）或中断将音频采样传输到存储器中。建议选择 DMA，有关更多信息，请参见*第 52.3.12 节：DMA 接口*。

SPDIFRX 提供多种处理接收数据的方法。用户可以分别处理控制信息流和音频采样流，或将二者一起处理。

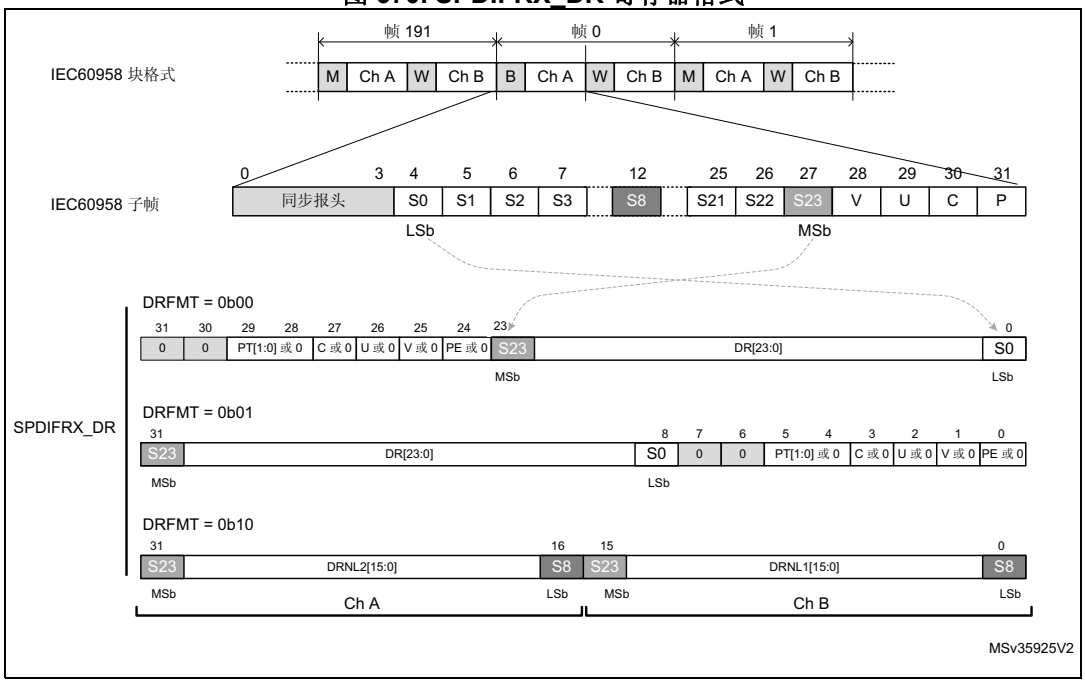
对于各子帧，数据接收寄存器 SPDIFRX_DR 包含 24 个数据位和可选的 V、U、C、PE 状态位以及 PT（请参见*混合数据和控制流*）。

请注意，PE 位表示奇偶校验错误位，该位将在解码的子帧中检测到奇偶校验错误时置 1。PT 位域包含报头类型（B、M 或 W）。

V、U 和 C 是从 S/PDIF 接口接收的值的直接副本。

DRFMT 位允许选择*图 670* 中所示的 3 种音频格式。

图 670. SPDIFRX_DR 寄存器格式



将 DRFMT 设置为 0b00 或 0b01，可以使数据在 SPDIFRX_DR 寄存器中左对齐或右对齐。可以根据软件所需的处理方式启用状态信息或将其强制设置为零。

通过 DRFMT = 0b10 得出的格式在非线性模式下相关，因为每个子帧仅使用 16 位。使用此格式，两个连续子帧的数据存储到 SPDIFRX_DR 中，存储器占用量将除以二。请注意，当 RXSTEO = 1 时，不存在偏离风险（即，来自通道 A 的数据始终存储在 SPDIFRX_DR[31:16] 中）。如果 RXSTEO = 0，则在上溢情况下存在偏离风险。此时，SPDIFRX_DR[31:16] 始终包含最早的值，SPDIFRX_DR[15:0] 则包含较新的值（请参见*图 672*）。

在此格式中，状态信息不能与数据混合，但用户仍可通过 SPDIFRX_CSR 寄存器获取这些数据，并使用专用的 DMA 通道或中断将它们传输到存储器中（请参见*第 52.3.8 节：专用控制流*）

混合数据和控制流

用户可以选择使用此模式，以完全灵活地处理控制流。用户可以选择应保存到数据寄存器 (SPDIFRX_DR) 中的位域。

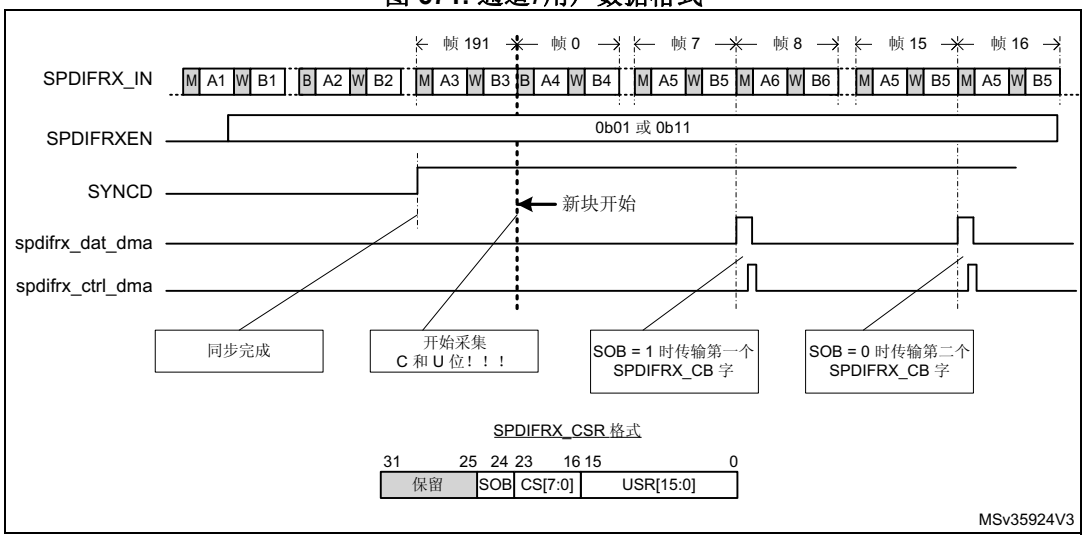
- 当位 PMSK = 1 时，将屏蔽奇偶校验错误信息（设置为 0），否则将其复制到 SPDIFRX_DR 中。
- 当位 VMSK = 1 时，将屏蔽有效性信息（设置为 0），否则将其复制到 SPDIFRX_DR 中。
- 当位 CUMSK = 1 时，将屏蔽通道状态和使用的数据信息（设置为 0），否则将其复制到 SPDIFRX_DR 中。
- 当位 PTMSK = 1 时，将屏蔽报头类型信息（设置为 0），否则将其复制到 SPDIFRX_DR 中。

52.3.8 专用控制流

SPDIFRX 可以通过专用 DMA 通道捕获用户数据和通道状态信息。该功能允许 SPDIFRX 连续采集通道状态和用户信息。采集将在 IEC 60958 块开始时启动。可使用两个位域控制该路径：CBDMAEN 和 SPDIFRXEN。当 SPDIFRXEN 设置为 0b01 或 0x11 时，采集将在同步阶段完成后开始。当接收到 8 个通道状态和 16 个用户数据位时，会将它们封装并存储到 SPDIFRX_CSR 寄存器中。如果 CBDMAEN 位置 1，则触发 DMA 请求（请参见*图 671*）。

如果 CS[0] 对应新块的第一位，则 SOB 位将置 1。请参见*第 52.5.8 节：通道状态寄存器 (SPDIFRX_CSR)*。可使用 CHSEL 位选择用户想要从通道 A 还是通道 B 选择通道状态信息 (C)。

图 671. 通道/用户数据格式



注： *检测到第一个块开始（B 报头）后，SPDIFRX 将每 8 个帧检测一次报头类型。*

注： *SPDIFRX_DR 寄存器中的上溢错误不影响该路径。*

52.3.9 接收错误

帧结构和同步错误

发生下列情况之一时，SPDIFRX 将检测到错误：

- 出现以下情况时，FERR 位置 1：
  - 对于 28 个信息位中的每个位，如果一个符号转换序列不正确：例如短脉冲未按对分组。
  - 如果报头出现在异常位置，或未接收到预期的报头。
- 当同步失败时，SERR 位置 1，原因是重试次数超过了设定的值。
- 当用于估算两次传输的时间间隔的计数器 (TRCNT) 产生上溢时，TERR 位置 1。
  在 8192 个 spdifrx_ker_ck 时钟周期内未检测到转换时，将发生上溢。这表示最大时间间隔为 11.6 帧。

当这些标志之一变为 1 时，将忽略所选 SPDIFRX_IN 上的通信，并将在 SPDIFRX_CR 寄存器的 IFEIE 位置 1 时生成中断。

发生这些错误之一时的正常步骤为：

- 将 SPDIFRXEN 设置为 0 以清零错误标志
- 将 SPDIFRXEN 设置为 0b01 或 0b11 以重新启动 IP

有关更多信息，请参见*图 669*。

奇偶校验错误

对于各子帧，28 个信息位中预期包含偶数个 0 和 1。否则，SPDIFRX_SR 寄存器的 PERR 位将被置位，并且如果 SPDIFRX_CR 寄存器中的奇偶校验中断使能 PERRIE 位置 1，则将生成中断。不会暂停输入数据的接收，并且即使中断仍处于挂起状态，SPDIFRX 也会继续将数据传输到 SPDIFRX_DR 中。

通过 PERRCF 位清零 PERR 标志来确认中断。

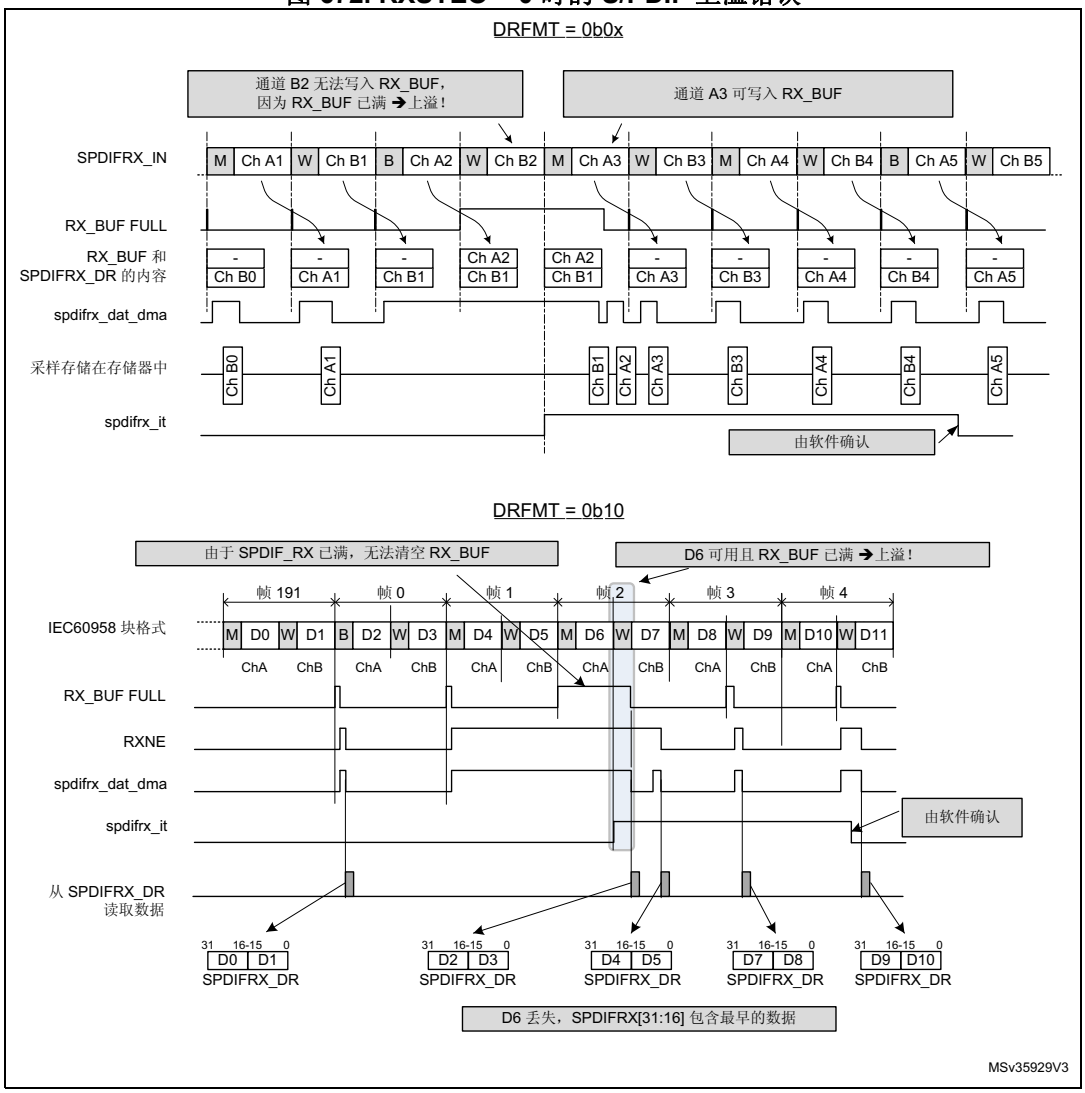
如果软件要确保 SPDIFRX_DR 寄存器中读取的数据与 PERR 位的值保持一致，必须将 PMSK 位设置为 0。

上溢错误

如果 SPDIFRX_DR 和 RX_BUF 均已满，而 SPDIFRX_DC 又需要向 RX_BUF 中写入新采样，则将丢弃这一新采样，并触发上溢条件。SPDIFRX_SR 寄存器中的上溢错误标志 OVR 置 1，并且如果 SPDIFRX_CR 寄存器的 OVRIE 位置 1，则将生成中断。

如果 RXSTEO 位设置为 0，则只要 RX_BUF 为空，IP 便会存储接下来传入的数据，即使 OVR 标志仍然处于挂起状态。主要目的是尽可能地减少丢失的采样数。请注意，无论 DRFMT 的值为何，这种行为都是类似的。请参见*图 672*。

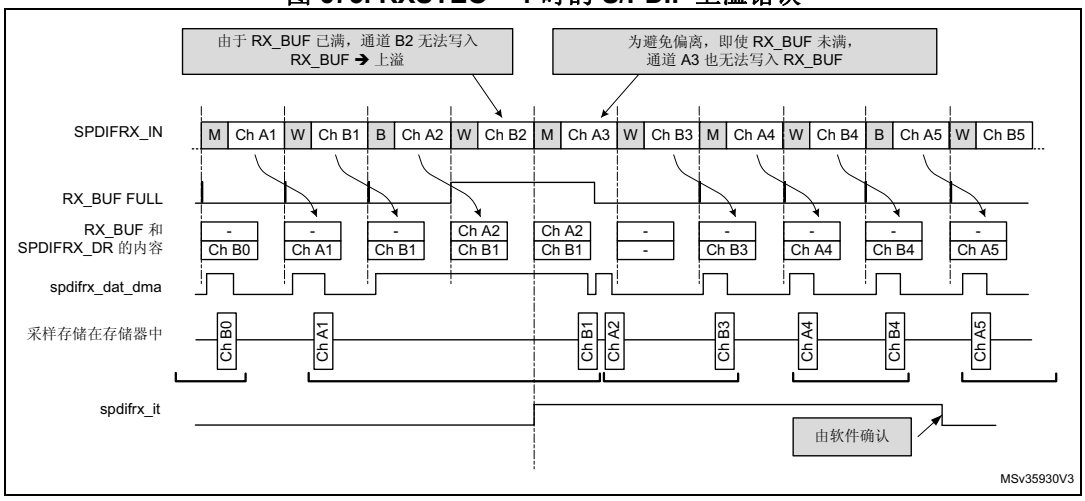
图 672. RXSTEO = 0 时的 S/PDIF 上溢错误



如果 RXSTEO 位置 1，则意味着传输的是立体声数据，此时 SPDIFRX 必须避免左右通道之间的偏离。因此，即使在 RX_BUF 内仍有空间，外设也必须丢弃一秒的采样以避免发生偏离。随后，传入的采样可正常写入到 RX_BUF 中，即使 OVR 标志仍处于挂起状态。请参见 *图 673*。

在 OVRCF 位置 1 时，OVR 标志由软件清零。

图 673. RXSTEO = 1 时的 S/PDIF 上溢错误



52.3.10 时钟策略

SPDIFRX 块需要两个不同的时钟：

- APB 时钟 (spdifrx_pclk)，用于寄存器接口。
- spdifrx_ker_ck，主要由 SPDIFRX_DC 部分使用。这些时钟不应锁相，因此需要重新同步这些时钟域中的所有信号（*图 659* 中的 SYNC 块）。

为正确解码传入的 S/PDIF 数据流，SPDIFRX_DC 应以至少比最大符号率高 11 倍或者比音频采样率高 704 倍的时钟重新采样接收的数据。例如，如果用户预期接收到最高 12.288 MHz 的符号率，则采样率至少为 135.2 MHz。SPDIFRX_DC 使用的时钟为 spdifrx_ker_ck。

spdifrx_pclk 的频率必须至少等于符号率。

表 410. spdifrx_ker_ck 的最小频率与音频采样率

符号率	spdifrx_ker_ck 的最小频率	注释
3.072 MHz	33.8 MHz	用于 48 kHz 数据流
6.144 MHz	67.6 MHz	用于 96 kHz 数据流
12.288 MHz	135.2 MHz	用于 192 kHz 数据流

52.3.11 符号时钟生成

SPDIFRX 模块在 **spdifrx_symb_ck** 信号上提供了符号时钟，不但可用作其他音频器件（例如，SAI 或 SPI/I2S）的参考内核时钟，而且可用于 SPDIFRX 转 I2S 桥接功能。

符号时钟使用 WIDTH24、WIDTH40 和符号边界的值构建。

- 在接收子帧同步报头期间，符号时钟的下降沿和上升沿通过 WIDTH24 和 WIDTH40 的值形成。请注意，如果 SPDIFRX 为 STATE_STOP 或 STATE_IDLE，WIDTH24 和 WIDTH40 也用于生成符号时钟。有关详细信息，请参见*表 411*。
- 在接收子帧有效负载期间，SPDIFRX 使用符号边界来生成上升沿，WIDTH24 和 WIDTH40 的值则用于生成下降沿。

在接收子帧有效负载期间，符号时钟的占空比接近 50%。不过，SPDIFRX 从使用 WIDTH24 和 WIDTH40 生成的符号时钟转换到由符号时钟边界生成的时钟时，占空比会发生改变，反之亦然。

符号时钟会产生大幅抖动的主要原因是：

- 使用 spdifrx_ker_ck 时钟重新采样 S/PDIF 信号
- 符号时钟生成模式发生转换

因此，如果将符号时钟用作 A/D 或 D/A 转换器的参考时钟，则应用程序应考虑质量下降问题。

该符号时钟的生成由 CKSEN 位控制。如果 CKSEN = “1”，则在 SPDIFRX 成功完成首次精同步 (SYNCD = 1) 时，以及从所选 SPDIFRX 输入接收正确的数据时，会生成符号时钟。

当 SPDIFRX 进入 STATE_STOP 或 STATE_IDLE 时，如果位 CKSBKPEN = “0”，则会对符号时钟进行门控。如果 CKSBKPEN = “1”，则在 SPDIFRX 正确同步（即，WIDTH24 和 WIDTH40 的值有效）时，仍会生成备份符号时钟。*表 411* 详细介绍了有关控制符号时钟生成的条件。

表 411. spdifrx_symb_ck 的生成条件

SPDIFRX 状态和条件	CKSEN	CKSBKPEN	spdifrx_symb_ck state
任意状态	0	X	输出关闭
– SPDIFRX 处于 STATE_SYNC，成功完成精同步（SYNCD = “1”） – SPDIFRX 处于 STATE_RCV，通过所选 SPDIFRX 输入接收到有效数据	1	0	启用
– SPDIFRX 处于 STATE_IDLE – SPDIFRX 处于 STATE_STOP – SPDIFRX 未完成精同步（正在进行） – SPDIFRX 处于 STATE_RCV，但在所选 SPDIFRX 输入上未检测到数据（转换）		0	输出关闭
– SPDIFRX 处于 STATE_IDLE，但存在 WIDTH40 和 WIDTH24 的有效值 – SPDIFRX 处于 STATE_SYNC，成功完成精同步（SYNCD = “1”） – SPDIFRX 处于 STATE_SYNC，正在进行的精同步尚未完成，但 WIDTH40 和 WIDTH24 包含来自上一次同步的有效值 – SPDIFRX 处于 STATE_RCV，通过所选 SPDIFRX 输入接收到有效数据 – SPDIFRX 处于 STATE_STOP，但存在 WIDTH40 和 WIDTH24 的有效值	1	1	启用
– SPDIFRX 处于 IDLE，WIDTH40 和 WIDTH24 的值无效 – SPDIFRX 处于 STOP，WIDTH40 和 WIDTH24 的值无效（SERR = “1”） – SPDIFRX 处于 STATE_SYNC，WIDTH40 和 WIDTH24 的值无效，且正在进行的精同步尚未完成 – SPDIFRX 处于 STATE_RCV，在所选 SPDIFRX 输入上未检测到转换			输出关闭

请注意，当标志 SERR 置“1”时，无法生成符号时钟和备份时钟，因为不存在同步。

请注意，如果 CKSEN 和 CKSBKPEN 均置“1”，则在 SPDIFRX 从 STATE_SYNC 或 STATE_RCV 切换到 STATE_STOP 或 STATE_IDLE 时，符号时钟会丢失某些转换。

CKSEN 和 CKSBKPEN 位在*控制寄存器 (SPDIFRX_CR)* 中。

52.3.12 DMA 接口

SPDIFRX 能够使用 DMA 执行通信。

注： *有关 DMA 控制器可用性的信息，请参见产品规范。*

SPDIFRX 提供两个独立的 DMA 通道：

- 一个专门用于数据传输的 DMA 通道
- 一个专用于传输通道状态和用户信息的 DMA 通道

可通过将 SPDIFRX_CR 寄存器中的 RXDMAEN 位置 1 来启用数据的 DMA 模式。这种情况下，只要 SPDIFRX_DR 不为空，SPDIFRX 接口便会将传输请求发送给 DMA。DMA 通过 SPDIFRX_DR 寄存器读取接收的数据，无需 CPU 干预。

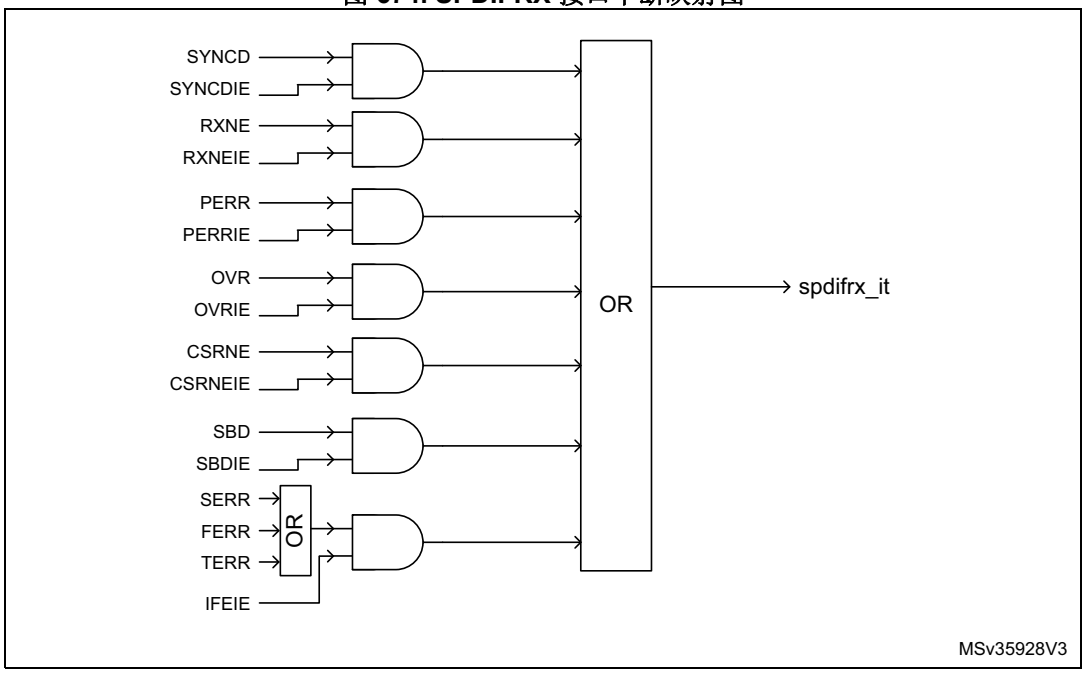
有关控制数据的 DMA 的用法，请参见*第 52.3.8 节：专用控制流*。

52.3.13 中断生成

下列各项共享一条中断线：

- 数据流的接收事件 (RXNE)
- 控制流的接收事件 (CSRNE)
- 数据损坏检测 (PERR)
- 传输流中断 (OVR)
- 帧结构和同步错误（SERR、TERR 和 FERR）
- 新块中段的开始 (SBD)
- 同步完成 (SYNCD)

图 674. SPDIFRX 接口中断映射图



清零中断源

- 读取 SPDIFRX_DR 寄存器时，将清零 RXNE。
- 读取 SPDIFRX_CSR 寄存器时，将清零 CSRNE。
- 当 SPDIFRXEN 被设置为 0 时，将清零 FERR。
- 当 SPDIFRXEN 被设置为 0 时，将清零 SERR。
- 当 SPDIFRXEN 被设置为 0 时，将清零 TERR。
- 其它中断源通过 SPDIFRX_IFCR 寄存器清零。

注： *仅当 SPDIFRX 与输入数据流同步时 (SYNCD = 1)，才会发生 SBD 事件。*
当包含 B 报头的子帧因上溢而丢失时，将无法保证 SBD 标志行为。

52.3.14 寄存器保护

SPDIFRX 块内嵌某种硬件保护功能以避免错误使用控制寄存器。下面的表格根据 SPDIFRX 的状态说明了位域属性。

表 412. 位域属性与 SPDIFRX 状态

寄存器	位域	SPDIFRXEN		
		0b00 (STATE_IDLE)	0b01 (STATE_SYNC)	0b11 (STATE_RCV)
SPDIFRX_CR	INSEL	rw	r	r
	WFA	rw	r	r
	NBTR	rw	r	r
	CHSEL	rw	r	r
	CBDMAEN	rw	rw	rw
	PTMSK	rw	rw	rw
	CUMSK	rw	rw	rw
	VMSK	rw	rw	rw
	PMSK	rw	rw	rw
	DRFMT	rw	rw	r
	RXSTEO	rw	rw	r
	RXDMAEN	rw	rw	rw
SPDIFRX_IMR	所有位域	rw	rw	rw

此表格明确说明了 INSEL 等位域必须在 IP 处于 STATE_IDLE 时进行编程。在其它 IP 状态下，硬件将阻止写入该位域。

注： *即使硬件允许“实时”写入 CBDMAEN 和 RXDMAEN，也不推荐在 IP 接收数据时启用 DMA。*

注： *请注意，各屏蔽位（PMSK、VMSK…）可以在任意 IP 状态下进行“实时”更改，但任何更改都不影响已经保存到 SPDIFRX_DR 中的数据。*

52.4 编程步骤

下列示例将说明 SPDIFRX 块的完整激活顺序。数据路径以及通道状态和用户信息都将使用专用的 DMA 通道。激活顺序随即分为下面几个步骤：

- 等待所选 SPDIFRX_IN 输入上的有效数据
- 与 S/PDIF 数据流同步
- 读取通道状态和用户数据以建立完整的音频路径
- 开始采集数据

一种检查有效数据是否可以进入 SPDIFRX_IN 线路的简单方法是通过将 WFA 位置 1 将 SPDIFRX 切换为 STATE_SYNC。后面将重点对检测进行说明。此外，也可以按如下方式实现此功能：

- 软件必须不时检查（例如，每隔 100 ms）SPDIFRX 是否可找到同步。可以通过检查 TERR 位是否置 1 来执行此操作。当该位置 1 时，表示没有发现活动。
- 将 SPDIFRX_IN 输入连接到外部中断事件块以便检测 SPDIFRX_IN 线路的转换。当检测到活动时，可将 SPDIFRXEN 设置为 0b01 或 0b11。

对于这两种实现，WFA 位设置为 0。

52.4.1 初始化阶段

- 初始化功能具有如下形式：
- 为音频采样和 IEC60958 通道状态和用户信息配置 DMA 传输（DMA 通道选择和活动、优先级、要传输的数据数量、循环/非循环模式和 DMA 中断）
- 配置目标地址：
  - 将 SPDIFRX_CSR 寄存器的地址配置为 IEC60958 通道状态和用户信息的源地址
  - 将 SPDIFRX_DR 寄存器的地址配置为音频采样的源地址
  - 使能 spdifrx_ker_ck 的生成。要定义最小时钟频率与支持的音频采样率，请参见 *表 410*
    请注意，无法提前获知已接收数据流的音频采样率。这意味着用户必须选择至少比应用应处理的最大音频采样率高 704 倍的 spdifrx_ker_ck 频率：例如，如果应用能够处理采样率最大为 96 kHz 的数据流，则 $F_{spdifrx_ker_ck}$ 至少应为 704 x 96 kHz = 67.6 MHz
- 使能错误和事件信号的中断（IFEIE = SYNCDIE = OVRIE，PERRIE = 1，其它位设置为 0）。注意，SYNCDIE 可设置为 0。
- 配置 SPDIFRX_CR 寄存器：
  - INSEL 将选择所需输入
  - NBTR = 2，WFA = 1（允许重试 16 次，等待活动，直至进入同步阶段）
  - PTMSK = CUMSK = 1（报头，C 和 U 位不与数据混合）
  - VMSK = PMSK = 0（奇偶校验错误和有效性位与数据混合）
  - CHSEL = 0 （从子帧 A 读取通道状态）
  - DRFMT = 0b01（数据左对齐）
  - RXSTEO = 1（预期立体声模式为线性）
  - CBDMAEN = RXDMAEN = 1（使能 DMA 通道）
  - SPDIFRXEN = 0b01（将 SPDIFRX 切换为 STATE_SYNC）
- CPU 可以进入 WFI 模式

随后 CPU 将收到来自 DMA 或 SPDIFRX 的中断。

52.4.2 处理来自 SPDIFRX 的中断

当收到来自 SPDIFRX 的中断时，软件必须通过读取 SPDIFRX_SR 寄存器来确认中断源。

- 如果 SYNCD 置 1，则意味着同步已正常完成。此示例中无需执行任何操作，因为 DMA 已进行编程。软件仅需要等待 DMA 中断以读取通道状态信息。
  必须通过将 SPDIFRX_IFCR 寄存器的 SYNCDCF 位置 1 来清零 SYNCD 标志。
- 如果 TERR 或 SERR 或 FERR 设置为 1，则软件必须将 SPDIFRXEN 设置为 0，并从初始化阶段重新启动。
  - TERR 表示在同步阶段期间或之后发生超时。
  - SERR 表示因为达到允许的最大重试次数而同步失败。
  - FERR 表示同失败步后读取信息（异常报头、错误数据解码...）。
- 如果 PERR 置 1，则意味着已检测到奇偶校验错误，从而表明接收到的音频采样、通道状态或用户数据位中的某一位损坏。这里采取的操作取决于应用：一个操作是丢弃当前通道状态块，原因是其不可靠。无需从初始化阶段重新启动，因为同步不会丢失。
  必须通过将 SPDIFRX_IFCR 寄存器的 PERRCF 位置 1 来清零 PERR 标志。

52.4.3 处理来自 DMA 的中断

如果中断来自通道状态所用的 DMA 通道 (SPDIFRX_CSR)：

如果未发生错误（即 PERR），则 CPU 可以开始解码通道信息。例如，通道状态的位 1 通知用户当前数据流是否为线性。该信息对于建立正常处理链十分重要。同样，通道状态的位 24 到位 27 给出了传入数据流的采样频率。

借助该信息，用户可在启动数据接收之前配置 RXSTEO 位和 DRFMT 位域。例如，如果当前数据流为非线性 PCM，则 RXSTEO 将被设置为 0，而 DRFMT 将被设置为 0b10。随后用户可通过将 SPDIFRXEN 设置为 0b11 来使能数据接收。

SOB 位置 1 时表示新块开始。该信息有助于软件识别通道状态的位 0。请注意，如果每次将 24 个值传输到存储器 DMA 便生成一个中断，则第一个字将始终对应新块的开始位置。

如果中断来自音频采样所用的 DMA 通道 (SPDIFRX_DR)：

这里执行的过程取决于数据类型（线性或非线性）以及所选的数据格式。

例如，在线性模式中，如果 PE 或 V 位置 1，则将在本地执行一个特殊过程以避免输出上出现杂波。在非线性模式下，这些位并不重要，因此数据帧具有各自的校验和。

52.5 SPDIFRX 接口寄存器

52.5.1 控制寄存器 (SPDIFRX_CR)

Control register

偏移地址：0x00

复位值：0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CKSBKPEN	CKSEN	Res.	INSEL[2:0](1)		
										rw	rw		rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	WFA(1)	NBTR[1:0](1)		CHSEL(1)	CBDMAEN(1)	PTMSK(1)	CUMSK(1)	VMSK(1)	PMSK(1)	DRFMT[1:0](1)		RXSTEO(1)	RXDMAEN(1)	SPDIFRXEN[1:0](1)	
	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

1. 有关位域属性的更多信息，请参见*第 52.3.14 节：寄存器保护*。

位 31:22 保留，由硬件强制清零。

位 21 **CKSBKPEN**：备份符号时钟使能 (Backup Symbol Clock Enable)
该位由软件置 1/复位。
1：CKSEN = “1” 时，SPDIFRX 会生成备份符号时钟
0：SPDIFRX 不生成备份符号时钟

位 20 **CKSEN**：符号时钟使能 (Symbol Clock Enable)
该位由软件置 1/复位。
1：SPDIFRX 生成符号时钟
0：SPDIFRX 不生成符号时钟

位 19 保留，由硬件强制清零。

位 18:16 **INSEL[2:0]**：SPDIFRX 输入选择 (SPDIFRX input selection)
0b000：选择 SPDIFRX_IN1
0b001：选择 SPDIFRX_IN2
0b010：选择 SPDIFRX_IN3
0b011：选择 SPDIFRX_IN4
其它保留

位 15 保留，由硬件强制清零。

位 14 **WFA**：等待活动 (Wait For Activity)
该位由软件置 1/复位。
1：执行同步前，SPDIFRX 等待 SPDIFRX_IN 线路上的活动（4 次转换）
0：执行同步前，SPDIFRX 不等待 SPDIFRX_IN 线路上的活动

位 13:12 **NBTR[1:0]**：同步阶段期间允许的最大重试次数 (Maximum allowed re-tries during synchronization phase)
0b00：不允许重试（仅尝试一次）
0b01：允许重试 3 次
0b10：允许重试 15 次
0b11：允许重试 63 次

位 11 **CHSEL**：通道选择 (Channel Selection)
该位由软件置 1/复位。
1：控制流从通道 B 获取通道状态
0：控制流从通道 A 获取通道状态

位 10 **CBDMAEN**：控制流的控制缓冲区 DMA 使能 (Control Buffer DMA ENable for control flow)
该位由软件置 1/复位。
1：使能 DMA 模式以接收通道状态和使用的数据信息。
0：禁止 DMA 模式接收通道状态和使用的数据信息。
当此位置 1 时，每当 CSRNE 标志置 1 时，即产生 DMA 请求。

位 9 **PTMSK**：报头类型的屏蔽位 (Mask of Preamble Type bits)
该位由软件置 1/复位。
1：不将报头类型位复制到 SPDIFRX_DR 中，写入 0 来替代
0：将报头类型位复制到 SPDIFRX_DR 中

位 8 **CUMSK**：通道状态和用户的屏蔽位 (Mask of channel status and user bits)
该位由软件置 1/复位。
1：不将通道状态和用户位复制到 SPDIFRX_DR 中，写入 0 来替代
0：将通道状态和用户位复制到 SPDIFRX_DR 中

位 7 **VMSK**：有效性的屏蔽位 (Mask of Validity bit)
该位由软件置 1/复位。
1：不将有效性位复制到 SPDIFRX_DR 中，写入 0 来替代
0：将有效性位复制到 SPDIFRX_DR 中

位 6 **PMSK**：屏蔽奇偶校验错误位 (Mask Parity error bit)
该位由软件置 1/复位。
1：不将奇偶校验错误位复制到 SPDIFRX_DR 中，写入 0 来替代
0：将奇偶校验错误位复制到 SPDIFRX_DR 中

位 5:4 **DRFMT[1:0]**：RX 数据格式 (RX Data format)
该位由软件置 1/复位。
0b11：保留
0b10：通过将两个 16 位采样置于一个 32 位字中来封装数据采样
0b01：数据采样采用左对齐 (MSB)
0b00：数据采样采用右对齐 (LSB)

位 3 **RXSTEO**：立体声模式 (STerEO Mode)
该位由软件置 1/复位。
1：外设处于立体声模式
0：外设处于单声道模式
该位用于在发生上溢时处理偏离

位 2 **RXDMAEN**：数据流的接收器 DMA 使能 (Receiver DMA ENable for data flow)
该位由软件置 1/复位。
1：针对接收使能 DMA 模式
0：针对接收禁止 DMA 模式
当此位置 1 时，每当 RXNE 标志置 1 时，即产生 DMA 请求。

位 1:0 **SPDIFRXEN[1:0]**：外设模块使能 (Peripheral Block Enable)
此位域由软件修改。
该位用于在三种可能的状态（STATE_IDLE、STATE_SYNC 和 STATE_RCV）间更改外设阶段。

0b00：禁止 SPDIFRX (STATE_IDLE)。
0b01：仅使能 SPDIFRX 同步
0b10：保留
0b11：使能 SPDIF 接收器

注： *1 无法从 STATE_RCV 转换为 STATE_SYNC，用户应首先进入 STATE_IDLE。*
2 可以从 STATE_IDLE 转换为 STATE_RCV：在这种情况下，外设从 STATE_IDLE 转换为 STATE_SYNC，并且一旦执行完同步便进入 STATE_RCV。

52.5.2 中断屏蔽寄存器 (SPDIFRX_IMR)

Interrupt mask register

偏移地址：0x04

复位值：0x00000000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	IFEIE	SYNCDIE	SBLKIE	OVRIE	PERRIE	CSRNEIE	RXNEIE
									rw	rw	rw	rw	rw	rw	rw

位 31:7 保留，由硬件强制清零。

位 6 **IFEIE**：串行接口错误中断使能 (Serial Interface Error Interrupt Enable)
此位由软件置 1 和清零。
0：禁止中断
1：当 SPDIFRX_SR 寄存器中 SERR = 1、TERR = 1 或 FERR = 1 时，生成 SPDIFRX 接口中断

位 5 **SYNCDIE**：同步完成 (Synchronization Done)
此位由软件置 1 和清零。
0：禁止中断
1：当 SPDIFRX_SR 寄存器中 SYNCD = 1 时，生成 SPDIFRX 接口中断

位 4 **SBLKIE**：同步块检测中断使能 (Synchronization Block Detected Interrupt Enable)
此位由软件置 1 和清零。
0：禁止中断
1：当 SPDIFRX_SR 寄存器中 SBD = 1 时，生成 SPDIFRX 接口中断

位 3 **OVRIE**：上溢错误中断使能 (Overrun error Interrupt Enable)
此位由软件置 1 和清零。
0：禁止中断
1：当 SPDIFRX_SR 寄存器中 OVR = 1 时，生成 SPDIFRX 接口中断

位 2 **PERRIE**：奇偶校验错误中断使能 (Parity error interrupt enable)
此位由软件置 1 和清零。
0：禁止中断
1：当 SPDIFRX_SR 寄存器中 PERR = 1 时，生成 SPDIFRX 接口中断

位 1 **CSRNEIE：控制缓冲区就绪中断使能 (Control Buffer Ready Interrupt Enable)**
此位由软件置 1 和清零。
0：禁止中断
1：当 SPDIFRX_SR 寄存器中 CSRNE = 1 时，生成 SPDIFRX 接口中断。

位 0 **RXNEIE**：RXNE 中断使能 (RXNE interrupt enable)
此位由软件置 1 和清零。
0：禁止中断
1：当 SPDIFRX_SR 寄存器中 RXNE = 1 时，生成 SPDIFRX 接口中断

52.5.3 状态寄存器 (SPDIFRX_SR)

Status register

偏移地址：0x08

复位值：0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	WIDTH5[14:0]														
	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	TERR	SERR	FERR	SYNCD	SBD	OVR	PERR	CSRNE	RXNE
							r	r	r	r	r	r	r	r	r

位 31 保留，由硬件强制清零。

位 30:16 **WIDTH5[14:0]**：使用 spdifrx_ker_ck 计数 5 个符号的持续时间 (Duration of 5 symbols counted with spdifrx_ker_ck)

该值表示 5 个连续符号的时间内包含的 spdifrx_ker_ck 时钟周期数。该值可用于估算 S/PDIF 符号率。其精度受 spdifrx_ker_ck 的频率限制。

例如，如果 spdifrx_ker_ck 固定为 84 MHz，则 WIDTH5 = 147d。S/PDIF 数据流的采样率估算值为：

$Fs = 5 \times F_{spdifrx_ker_ck} / (WIDTH5 \times 64) \sim 44.6$ kHz，因此最接近的标准采样率为 44.1 kHz。

请注意，当 SYNCD 变为高电平时，WIDTH5 将由硬件更新，随后每帧都更新。

位 15:9 保留，由硬件强制清零。

位 8 **TERR**：超时错误 (Time-out error)

当计数器 TRCNT 值达到最大值时，该位由硬件置 1。它表示两次转换之间的时间间隔过长。这通常意味着 SPDIFRX_IN 输入上没有有效信号。

该标志通过将 SPDIFRXEN 写为 0 来清零。

如果 SPDIFRX_IMR 寄存器中 IFEIE=1，则会生成中断。

0：未检测到序列错误

1：检测到序列错误

位 7 **SERR**：同步错误 (Synchronization error)

当同步出于 NBTR 重试次数方面的原因而失败时，该位由硬件置 1。

该标志通过将 SPDIFRXEN 写为 0 来清零。

如果 SPDIFRX_IMR 寄存器中 IFEIE=1，则会生成中断。

0：未检测到同步错误

1：检测到同步错误

位 6 **FERR**：帧错误 (Framing error)

当在接收数据期间发生错误时，该位由硬件置 1：报头未出现在预期位置，短转换未按对分组...

仅当同步完成时 (SYNCD = 1)，该位才能由硬件置 1。

该标志通过将 SPDIFRXEN 写为 0 来清零。

如果 SPDIFRX_IMR 寄存器中 IFEIE=1，则会生成中断。

0：未检测到曼彻斯特编码违例

1：检测到曼彻斯特编码违例

位 5 **SYNCD**：同步完成 (Synchronization Done)
当初始同步阶段正确完成时，该位由硬件置 1。
向 SPDIFRX_CLR_SR 寄存器中的相应位写入 1 即可将该标志清零。
如果 SPDIFRX_IMR 寄存器中 SYNCDIE = 1，则会生成中断
0：同步挂起
1：同步完成

位 4 **SBD**：检测到同步块 (Synchronization Block Detected)
当检测到“B”报头时，该位由硬件置 1。
向 SPDIFRX_CLR_SR 寄存器中的相应位写入 1 即可将该标志清零。
如果 SPDIFRX_IMR 寄存器中 SBLKIE = 1，则会生成中断
0：未检测到“B”报头
1：已检测到“B”报头

位 3 **OVR**：上溢错误 (Overrun error)
在 RXNE = 1 且 SPDIFRX_DR 和 RX_BUF 均已满的情况下，当 SPDIFRX_DR 寄存器中的接收数据准备好传输时，该位由硬件置 1。
向 SPDIFRX_CLR_SR 寄存器中的相应位写入 1 即可将该标志清零。
如果 SPDIFRX_IMR 寄存器中 OVRIE=1，则会生成中断。
0：无上溢错误
1：检测到上溢错误
注： 当该位置 1 时，SPDIFRX_DR 寄存器的内容不会丢失，但最后接收的数据会丢失。

位 2 **PERR**：奇偶校验错误 (Parity error)
当所接收子帧的数据和状态位中包含奇数个 0 和 1 时，该位由硬件置 1。
向 SPDIFRX_CLR_SR 寄存器中的相应位写入 1 即可将该标志清零。
如果 SPDIFRX_IMR 寄存器中 PIE = 1，则会生成中断。
0：无奇偶校验错误
1：奇偶校验错误

位 1 **CSRNE**：控制缓冲区寄存器不为空 (The Control Buffer register is not empty)
当有效的控制信息就绪时，该位由硬件置 1。
当读取 SPDIFRX_CSR 寄存器时，该标志清零。
如果 SPDIFRX_IMR 寄存器中 CBRDYIE = 1，则会生成中断。
0：SPDIFRX_CSR 寄存器中无控制字可用
1：SPDIFRX_CSR 寄存器中有控制字可用

位 0 **RXNE**：读取数据寄存器不为空 (Read data register not empty)
当有效数据进入 SPDIFRX_DR 寄存器时，该位由硬件置 1。
可通过读取 SPDIFRX_DR 寄存器清零该标志。
如果 SPDIFRX_IMR 寄存器中 RXNEIE=1，则会生成中断。
0：未接收到数据
1：已准备好读取接收到的数据

52.5.4 中断标志清零寄存器 (SPDIFRX_IFCR)

Interrupt flag clear register

偏移地址：0x0C

复位值：0x00000000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SYNCDCF	SBDCF	OVRCF	PERRCF	Res.	Res.
										w	w	w	w		

位 31:6 保留，由硬件强制清零。

位 5 **SYNCDCF**：清零同步完成标志 (Clears the Synchronization Done flag)
向该位写入 1 可清零 SPDIFRX_SR 寄存器中的 SYNCD 标志。
读取该位将始终返回值 0。

位 4 **SBDCF**：清零同步块检测标志 (Clears the Synchronization Block Detected flag)
向该位写入 1 可清零 SPDIFRX_SR 寄存器中的 SBD 标志。
读取该位将始终返回值 0。

位 3 **OVRCF**：清零上溢错误标志 (Clears the Overrun error flag)
向该位写入 1 可清零 SPDIFRX_SR 寄存器中的 OVR 标志。
读取该位将始终返回值 0。

位 2 **PERRCF**：清零奇偶校验错误标志 (Clears the Parity error flag)
向该位写入 1 可清零 SPDIFRX_SR 寄存器中的 PERR 标志。
读取该位将始终返回值 0。

位 1:0 保留

52.5.5 数据输入寄存器 (SPDIFRX_DR)

Data input register

偏移地址：0x10

复位值：0x00000000

根据 DRFMT 的值，该寄存器可采用 3 种不同格式。当 DRFMT = 0b00 时，格式如下：

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	PT[1:0]		C	U	V	PE	DR[23:16]							
		r	r	r	r	r	r	r	r	r	r	r	r	r	r

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

位 31:30 保留：由硬件强制清零

位 29:28 **PT**：报头类型 (Preamble Type)
这些位表示接收到的报头。
00：未使用
01：接收到报头 B
10：接收到报头 M
11：接收到报头 W
请注意，如果 PTMSK = 1，该位域将强制被设置为零

位 27 **C**：通道状态位 (Channel Status bit)
如果 CUMSK = 0，则包含接收到的通道状态位，否则被强制设置为 0

位 26 **U**：用户位 (User bit)
如果 CUMSK = 0，则包含接收到的用户位，否则被强制设置为 0

位 25 **V**：有效性位 (Validity bit)
如果 VMSK = 0，则包含接收到的有效性位，否则被强制设置为 0

位 24 **PE**：奇偶校验错误位 (Parity Error bit)
如果 PMSK = 0，则包含 PERR 位的副本，否则被强制设置为 0

位 23:0 **DR**：数据值 (Data value)
包含 24 个接收到的数据位，与 D[23] 对齐

52.5.6 数据输入寄存器 (SPDIFRX_DR)

Data input register

偏移地址：0x10

复位值：0x00000000

根据 DRFMT 的值，该寄存器可采用 3 种不同格式。当 DRFMT = 0b01 时，格式如下：

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DR[23:8]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DR[7:0]								Res.	Res.	PT[1:0]		C	U	V	PE
r	r	r	r	r	r	r	r			r	r	r	r	r	r

位 31:8 **DR**：数据值 (Data value)
包含 24 个接收到的数据位，与 D[23] 对齐

位 7:6 保留：由硬件强制清零

位 5:4 **PT**：报头类型 (Preamble Type)
这些位表示接收到的报头。
00：未使用
01：接收到报头 B
10：接收到报头 M
11：接收到报头 W
请注意，如果 PTMSK = 1，该位域将强制被设置为零

位 3 **C**：通道状态位 (Channel Status bit)
如果 CUMSK = 0，则包含接收到的通道状态位，否则被强制设置为 0

位 2 **U**：用户位 (User bit)
如果 CUMSK = 0，则包含接收到的用户位，否则被强制设置为 0

位 1 **V**：有效性位 (Validity bit)
如果 VMSK = 0，则包含接收到的有效性位，否则被强制设置为 0

位 0 **PE**：奇偶校验错误位 (Parity Error bit)
如果 PMSK = 0，则包含 PERR 位的副本，否则被强制设置为 0

52.5.7 数据输入寄存器 (SPDIFRX_DR)

Data input register

偏移地址：0x10

复位值：0x00000000

根据 DRFMT 的值，该寄存器可采用 3 种不同格式。

DRFMT = 0b10 时的数据格式专用于非线性模式，因为仅使用了 16 位（S/PDIF 子帧中的位 23 到位 8）。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DRNL2[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DRNL1[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:16 **DRNL2**：数据值 (Data value)
该位域包含通道 A

位 15:0 **DRNL1**：数据值 (Data value)
该位域包含通道 B

52.5.8 通道状态寄存器 (SPDIFRX_CSR)

Channel status register

偏移地址：0x14

复位值：0x00000000

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

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

位 31:25 保留

位 24 **SOB**：块开始 (Start Of Block)
该位表示位 CS[0] 是否对应于新块的第一位
0：CS[0] 不是新块的第一位
1：CS[0] 是新块的第一位

位 23:16 **CS[7:0]**：通道 A 状态信息 (Channel A status information)
位 CS[0] 是最早的值

位 15:0 **USR[15:0]**：用户数据信息 (User data information)
位 USR[0] 是最早的值，来自通道 A，USR[1] 来自通道 B。
因此，n 为偶数时，USR[n] 位来自通道 A，否则来自通道 B。

52.5.9 调试信息寄存器 (SPDIFRX_DIR)

Debug Information register

偏移地址：0x18

复位值：0x00000000

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

位 31:29 保留，由硬件强制清零。

位 28:16 **TLO**：阈值下限 (Threshold LOW) (TLO = 1.5 x UI / $T_{spdifrx_ker_ck}$)

该位域包含当前阈值下限的估算值。该值可用于估算接收数据流的采样率。TLO 的精度受限于 spdifrx_ker_ck 的周期。采样率按如下公式估算：

采样率 = [2 x TLO x $T_{spdifrx_ker_ck}$ +/- $T_{spdifrx_ker_ck}$] x 2/3

请注意，当 SYNCD 变为高电平时，TLO 将由硬件更新，随后每帧都更新。

位 15:13 保留，由硬件强制清零。

位 12:0 **THI**：阈值上限 (Threshold HIGH) (THI = 2.5 x UI / $T_{spdifrx_ker_ck}$)

该位域包含当前阈值上限的估算值。该值可用于估算接收数据流的采样率。THI 的精度受限于 spdifrx_ker_ck 的周期。采样率按如下公式估算：

采样率 = [2 x THI x $T_{spdifrx_ker_ck}$ +/- $T_{spdifrx_ker_ck}$] x 2/5

请注意，当 SYNCD 变为高电平时，THI 将由硬件更新，随后每帧都更新。

52.5.10 SPDIFRX 版本寄存器 (SPDIFRX_VERR)

SPDIFRX version register

偏移地址：0x03F4

复位值：0x0000 0012

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

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

位 31:8 保留，必须保持复位值。

位 7:4 **MAJREV[3:0]**：主版本 (Major revision)
这些位返回 SPDIFRX 主版本。
主版本为 1。

位 3:0 **MINREV[3:0]**：次版本 (Minor revision)
这些位返回 SPDIFRX 次版本。
次版本为 2。

52.5.11 SPDIFRX 标识寄存器 (SPDIFRX_IDR)

SPDIFRX identification register

偏移地址：0x03F8

复位值：0x0013 0041

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

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

位 31:0 **ID[31:0]**：SPDIFRX 标识符 (SPDIFRX identifier)
这些位返回 SPDIFRX 标识符值。

52.5.12 SPDIFRX 大小标识寄存器 (SPDIFRX_SIDR)

SPDIFRX size identification register

偏移地址：0x03FC

复位值：0xA3C5 DD01

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

位 31:0 **SID[31:0]**：大小标识 (Size identification)
这些位返回分配给 SPDIFRX 寄存器的存储区域的大小。
该存储区域的大小为 1 KB。

52.5.13 SPDIFRX 接口寄存器映射

表 413 给出了 SPDIFRX 接口寄存器映射和复位值。

表 413. SPDIFRX 接口寄存器映射和复位值

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	**SPDIFRX_CR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CKSBKPEN	CKSEN	Res.	INSEL[2:0]			Res.	WFA	NBTR[1:0]		CHSEL	CBDMAEN	PTMSK	CUMSK	VMSK	PMSK	DRFMT[1:0]		RXSTEO	RXDMAEN	SPDIFRXEN[1:0]	
	Reset value											0	0		0	0	0		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x04	**SPDIFRX_IMR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	IFEIE	SYNCDIE	SBLKIE	OVRIE	PERRIE	CSRNEIE	RXNEIE
	Reset value																										0	0	0	0	0	0	0
0x08	**SPDIFRX_SR**	Res.	WIDTH5[14:0]															Res.	Res.	Res.	Res.	Res.	Res.	Res.	TERR	SERR	FERR	SYNCD	SBD	OVR	PERR	CSRNE	RXNE
	Reset value		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0								0	0	0	0	0	0	0	0	0
0x0C	**SPDIFRX_IFCR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SYNCDCF	SBDCF	OVRCF	PERRCF	Res.	Res.
	Reset value																											0	0	0	0		
0x10	**SPDIFRX_DR**	Res.	Res.	PT[1:0]		C	U	V	PE	DR[23:0]																							
		DR[23:0]																								Res.	PT[1:0]		C	U	V	PE	
		DRNL2[15:0]																DRNL1[15:0]															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x14	**SPDIFRX_CSR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SOB	CS[7:0]								USR[15:0]															
	Reset value								0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x18	**SPDIFRX_DIR**	Res.	Res.	Res.	TLO[12:0]													Res.	Res.	Res.	THI[12:0]												
	Reset value				0	0	0	0	0	0	0	0	0	0	0	0	0				0	0	0	0	0	0	0	0	0	0	0	0	0
0x03F4	**SPDIFRX_VERR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MAJREV[3:0]				MINREV[3:0]			
	Reset value																									0	0	0	1	0	0	1	0
0x03F8	**SPDIFRX_IDR**	ID[31:16]																ID[15:0]															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1
0x03FC	**SPDIFRX_SIDR**	SID[31:16]																SID[15:0]															
	Reset value	1	0	1	0	0	0	1	1	1	1	0	0	0	1	0	1	1	1	0	1	1	1	0	1	0	0	0	0	0	0	0	1

53 单线协议主接口 (SWPMI)

53.1 简介

单线协议主接口 (SWPMI) 是与 ETSI TS 102 613 技术规范中定义的非接触式前端 (CLF) 相对应的主接口。

单线协议 (SWP) 的基本原理是在全双工模式下发送数字信息：

- S1 信号（从主器件到从器件）通过电压域中的数字调制（L 或 H）发送（请参见*图 675：S1 信号编码*）。
- S2 信号（从从器件到主器件）通过电流域中的数字调制（L 或 H）发送（请参见*图 676：S2 信号编码*）。

图 675. S1 信号编码

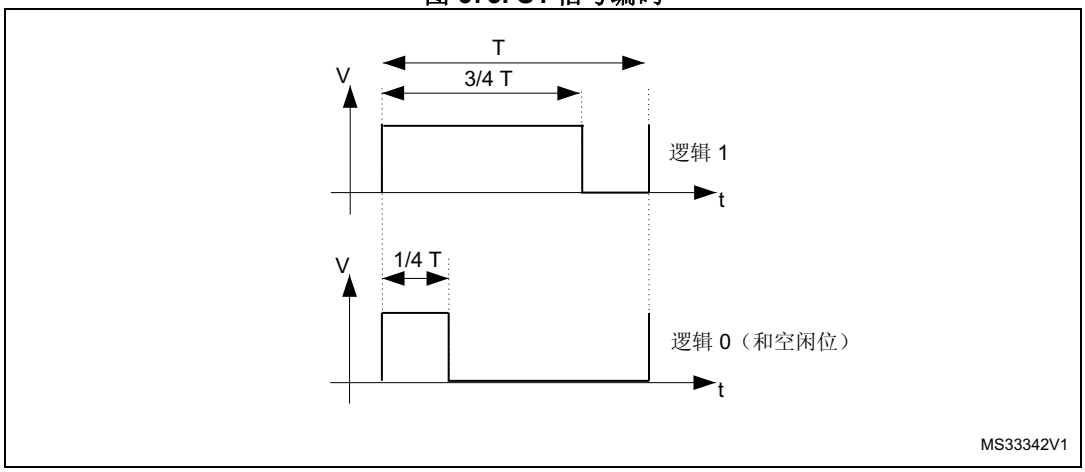
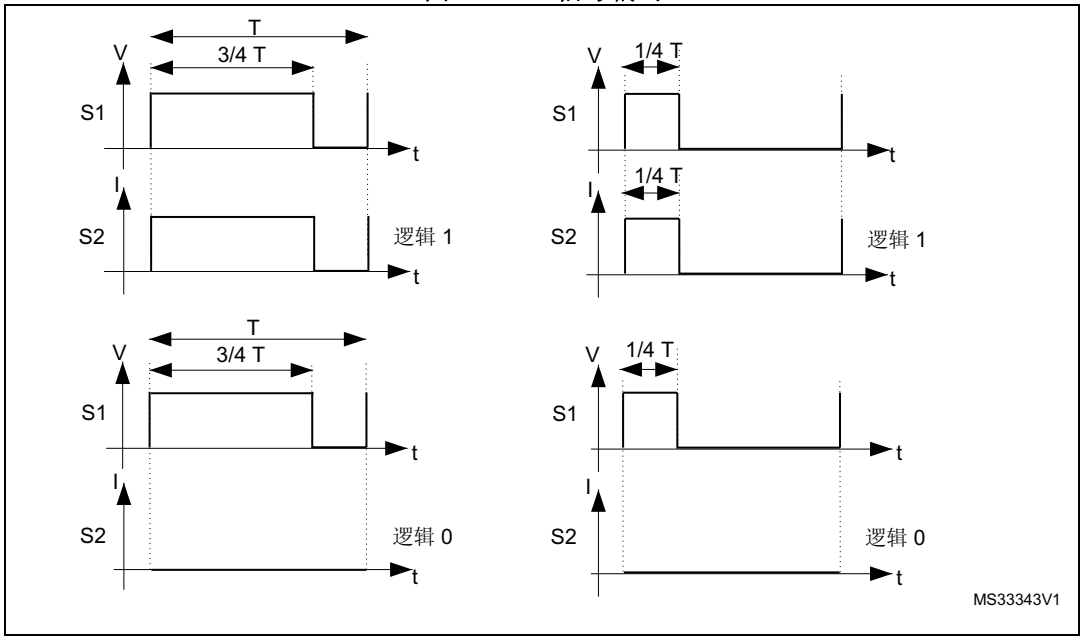


图 676. S2 信号编码



53.2 SWPMI 主要特性

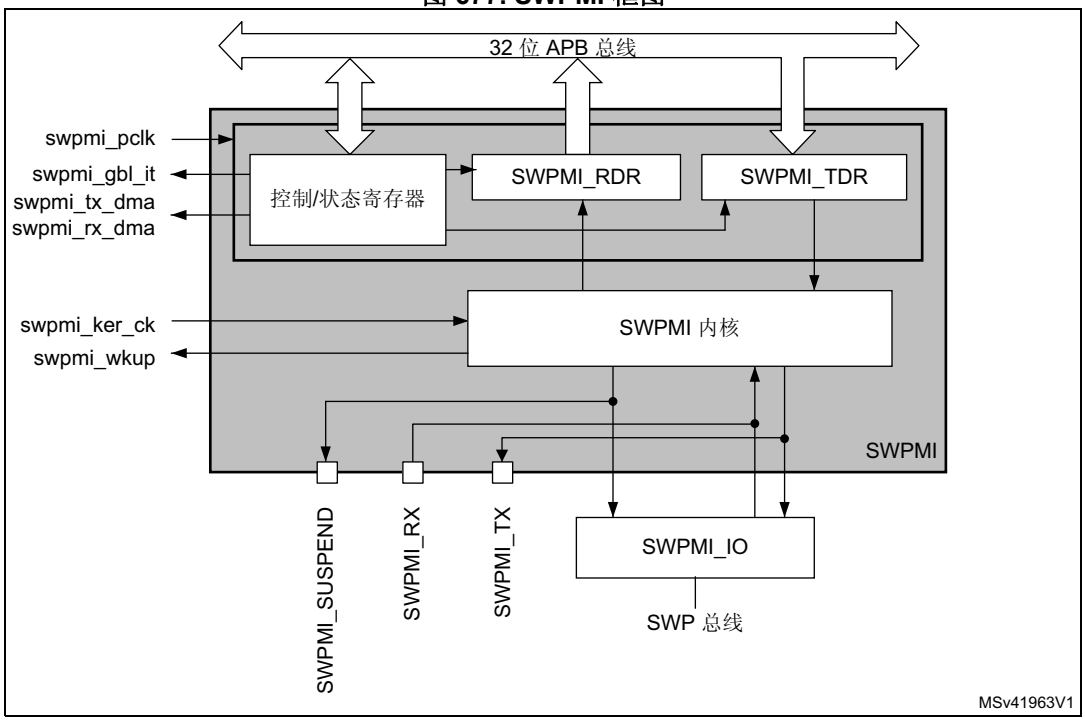
SWPMI 模块的主要特性如下（请参见*图 53.3.4：SWP 总线状态*）：

- 全双工通信模式
- 自动 SWP 总线状态管理
- 自动处理帧起始 (SOF)
- 自动处理帧结束 (EOF)
- 自动处理填充位
- 在发送期间自动进行 CRC-16 计算和生成
- 在接收期间自动进行 CRC-16 计算和校验
- 32 位发送数据寄存器
- 32 位接收数据寄存器
- 多软件缓冲区模式，可实现高效 DMA 传输和多帧缓冲
- 可配置比特率，最高可达 2 Mbit/s
- 可配置中断
- CRC 错误、下溢和上溢标志
- 帧接收和发送完成标志
- 从器件恢复检测标志
- 用于测试的回送模式
- 符合 ETSI TS 102 613 技术规范的嵌入式 SWPMI_IO 收发器
- 专用模式，可在外接收发器时在 GPIO 上输出 SWPMI_RX、SWPMI_TX 和 SWPMI_SUSPEND 信号

53.3 SWPMI 功能说明

53.3.1 SWPMI 框图

图 677. SWPMI 框图



请参见*第 8.7.18 节：RCC 域 2 内核时钟配置寄存器 (RCC_D2CCIP1R)* 中的 SWPSRC 位来选择 swpmi_ker_ck（SWPMI 内核时钟源）。

注： *为支持通过从器件 RESUME 来退出停止模式，必须为 swpmi_ker_ck 选择 HSI。如果无需此功能，则可选择 swpmi_pclk，并且必须在进入停止模式之前禁止 SWPMI。*

53.3.2 SWPMI 引脚和内部信号

表 414 列出了连接到封装引脚或焊球的 SWPMI 从输入与输出信号，*表 415* 则列出了内部 SWPMI 信号。

表 414. 连接到封装引脚或焊球的 SWPMI 输入/输出信号

信号名称	信号类型	说明
SWPMI_SUSPEND	数字输出	SWPMI 挂起信号
SWPMI_TX	数字输出	SWPMI 发送信号
SWPMI_RX	数字输入	SWPMI 接收信号

表 415. SWPMI 内部输入/输出信号

信号名称	信号类型	说明
swpmi_pclk	数字输入	APB 时钟
swpmi_ker_ck	数字输入	SWPMI 内核时钟
swpmi_wkup	数字输出	SWPMI 唤醒信号
swpmi_gbl_it	数字输出	SWPMI 中断信号
swpmi_tx_dma	数字输出	SWPMI DMA 发送请求
swpmi_rx_dma	数字输出	SWPMI DMA 接收请求

53.3.3 SWP 初始化和激活

初始化和激活会将 SWPMI_IO 状态从低电平设为高电平。

使用内部收发器时，相关过程如下：

1. 根据 VDD 电压（3 V 或 1.8 V）配置 SWPMI_OR 寄存器中的 SWP_CLASS 位。
2. 将 SWPMI_CR 寄存器中的 SWPTEN 置 1 以使能 SWPMI_IO 收发器，并将 SWPMI_IO 设为低电平（SWP 总线“已禁用”）。
3. 等待 SWPMI_SR 寄存器中的 RDYF 标志置 1（轮询标志或使用 SWPMI_IER 寄存器中的 RDYIE 位使能中断）。
4. 将 SWPMI_CR 寄存器中的 SWPACT 位置 1 以激活 SWP，即从“已禁用”状态转换到“已挂起”状态。

53.3.4 SWP 总线状态

SWP 总线可处于以下状态：“已禁用”、“已挂起”和“已激活”。

有以下几种状态转换：

- 激活：从“已禁用”状态转换到“已挂起”状态。
- 挂起：从“已激活”状态转换到“已挂起”状态。
- 主器件恢复：由主器件发起从“已挂起”状态到“已激活”状态的转换。
- 从器件恢复：由从器件发起从“已挂起”状态到“已激活”状态的转换。
- 禁用：从“已挂起”状态转换到“已禁用”状态。

激活

在复位期间以及在刚刚完成复位后，SWPMI_IO 配置为模拟模式。请参见*第 53.3.3 节：SWP 初始化和激活*了解有关激活 SWP 总线的信息。

挂起

无论是在发送模式下还是在接收模式下，只要与从器件之间存在通信，SWP 总线就会保持为“已激活”状态。如果后续没有发送或接收活动，则在 7 个空闲位后，SWP 总线将切换回“已挂起”状态。

主器件恢复

SWPMI 使能后，用户可请求发送 SWPMI 帧。SWPMI 先发送转换序列和 8 个空闲位（主器件恢复），然后再开始帧发送。在“主器件恢复”后，SWP 从“已挂起”状态转换到“已激活”状态（请参见*图 678：SWP 总线状态*）。

从器件恢复

SWPMI 使能后，如果 SWPMI 接收到来自从器件的恢复状态，SWP 也会从“已挂起”状态转换到“已激活”状态。“从器件恢复”会将 SWPMI_ISR 寄存器中的 SRF 标志置 1。

禁用

禁用请求

如果无需进一步通信，并且 SWP 处于“已挂起”模式，则用户可通过禁止 SWPMI 外设来请求将 SWP 切换到“已禁用”模式。软件必须将 SWPMI_CR 寄存器中的 DEACT 位置 1 才能请求切换到“已禁用”模式。如果 SWPMI 未检测到“从器件恢复”状态，则 SWPMI_ISR 寄存器中的 DEACTF 标志将置 1，SWPMI_ICR 寄存器中的 SWPACT 位将清零。如果在软件将 DEACT 位置 1 的情况下 SWPMI 检测到“从器件恢复”状态，则 SWPMI_ISR 寄存器中的 SRF 标志会置 1，DEACTF 保持清零，SWPACT 保持置 1，DEACT 位清零。

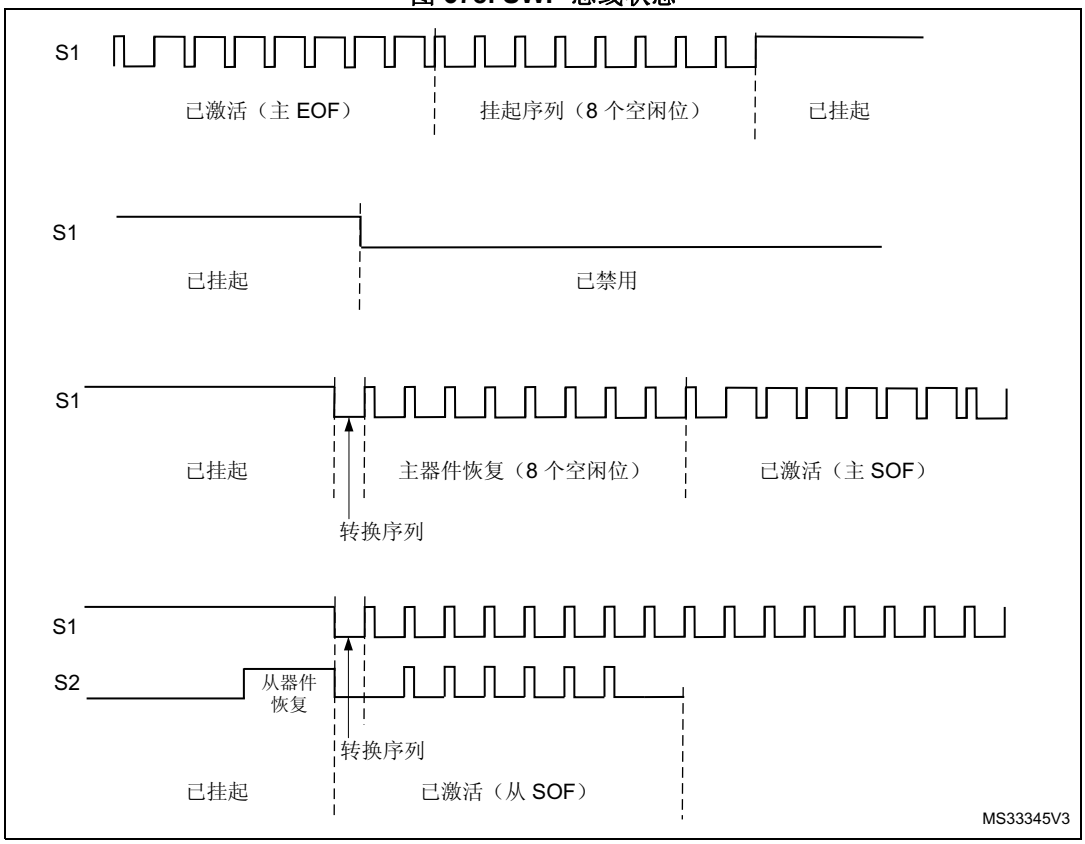
如要再次激活 SWP，软件必须先将 SWPMI_CR 寄存器中的 DEACT 位清零，然后再将 SWPACT 位置 1。

禁用模式

为了将 SWP 立即切换到“已禁用”模式，忽略可能传入的“从器件恢复”状态，用户必须将 SWPMI_CR 寄存器中的 SWPACT 位清零。

注： *为了进一步降低电流消耗，请将 GPIO 控制器中的 SWPMI_IO 端口配置为推挽输出低电平状态，然后将 SWPMI_CR 寄存器中的 SWPTEN 位清零（请参见第 11 节：通用 I/O (GPIO)）。*

图 678. SWP 总线状态



53.3.5 SWPMI_IO（内部收发器）旁路

微控制器中嵌入了符合 ETSI TS 102 613 技术规范的 SWPMI_IO（收发器）。不过，可通过将 SWPMI_OR 寄存器中的 SWP_TBYP 位置 1 将其旁路。在这种情况下，SWPMI_IO 被禁止，SWPMI_RX、SWPMI_TX 和 SWPMI_SUSPEND 信号作为这三个 GPIO 上的复用功能可供使用（请参见产品数据手册中的“引脚排列和引脚说明”）。通过选择此配置，可以连接外部收发器。

注： *在 SWPMI_IO 旁路模式下，SWPMI_CR 寄存器中的 SWPTEN 位必须保持清零。*

53.3.6 SWPMI 比特率

必须根据以下公式在 SWPMI_BRR 寄存器中设置比特率：

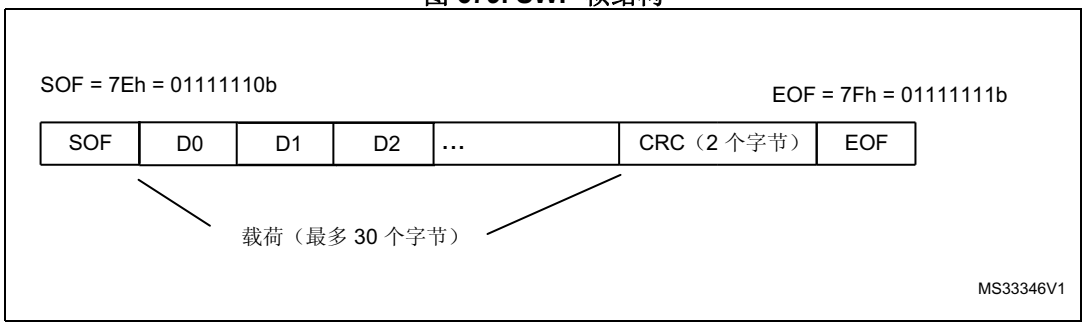
$F_{SWP} = F_{swpmi_ker_ck} / ((BR[7:0]+1)x4)$

注： *最大比特率为 2 Mbit/s。*

53.3.7 SWPMI 帧处理

SWP 帧包括以下部分：帧起始 (SOF)、1 到 30 字节的有效负载、16 位 CRC 和帧结束 (EOF)（请参见*图 679：SWP 帧结构*）。

图 679. SWP 帧结构



SWPMI 嵌入了一个用于发送的 32 位数据寄存器 (SWPMI_TDR)，和一个用于接收的 32 位数据寄存器 (SWPMI_RDR)。

在发送模式下，SOF 插入、CRC 计算和插入以及 EOF 插入均由 SWPMI 自动管理。用户仅需提供有效负载的内容和大小。向 SWPMI_TDR 寄存器中写入数据后，将立即开始帧发送。发送数据寄存器为空和完成帧发送事件会由专用标志来指示。

在接收模式下，SOF 删除、CRC 计算和检查以及 EOF 删除均由 SWPMI 自动管理。用户仅需读取有效负载的内容和大小。接收数据寄存器为满、完成帧接收和可能的 CRC 错误事件会由专用标志来指示。

填充位插入（发送模式下）和填充位删除（接收模式下）由 SWPMI 内核自动管理。这些操作对用户是透明的。

53.3.8 发送过程

在开始帧发送之前，用户必须激活 SWP。请参见*第 53.3.3 节：SWP 初始化和激活*。

对于帧发送，可能的软件实现有以下几种：无软件缓冲区模式、单软件缓冲区模式和多软件缓冲区模式。

使用软件缓冲区时，需要通过 DMA 通道将数据从 RAM 存储器的软件缓冲区传输到 SWPMI 外设中的发送数据寄存器。

无软件缓冲区模式

在该模式下，无需使用 DMA。SWP 帧发送的处理通过在主循环或 SWPMI 中断程序内轮询状态标志来完成。SWPMI 有一个 32 位发送数据寄存器 (SWPMI_TDR)，因此向该寄存器进行写操作会触发多达 4 个字节的发送操作。

通过将 SWPMI_CR 寄存器中的 TXDMA 位清零来选择无软件缓冲区模式。

第一次写入 SWPMI_TDR 寄存器时会启动帧发送。写入 SWPMI_TDR 寄存器中的第一个 32 位字的低有效字节（位 [7:0]）表示有效负载中的数据字节数，该字的其他 3 个字节必须包含有效负载的前 3 个字节（位 [15:8] 包含有效负载的第一个字节，位 [23:16] 包含第二个字节，位 [31:24] 包含第三个字节）。之后对 SWPMI_TDR 寄存器进行的写操作将仅包含后续的有效负载数据字节（每次写操作最多 4 个字节）。

注： *写入 SWPMI_TDR 寄存器的第一个 32 位字的低有效字节用于对有效负载中的数据字节数进行编码。该数字可为 1 到 30。如果低有效字节中的值不在此范围内，则会被忽略，并且将不会启动发送。*

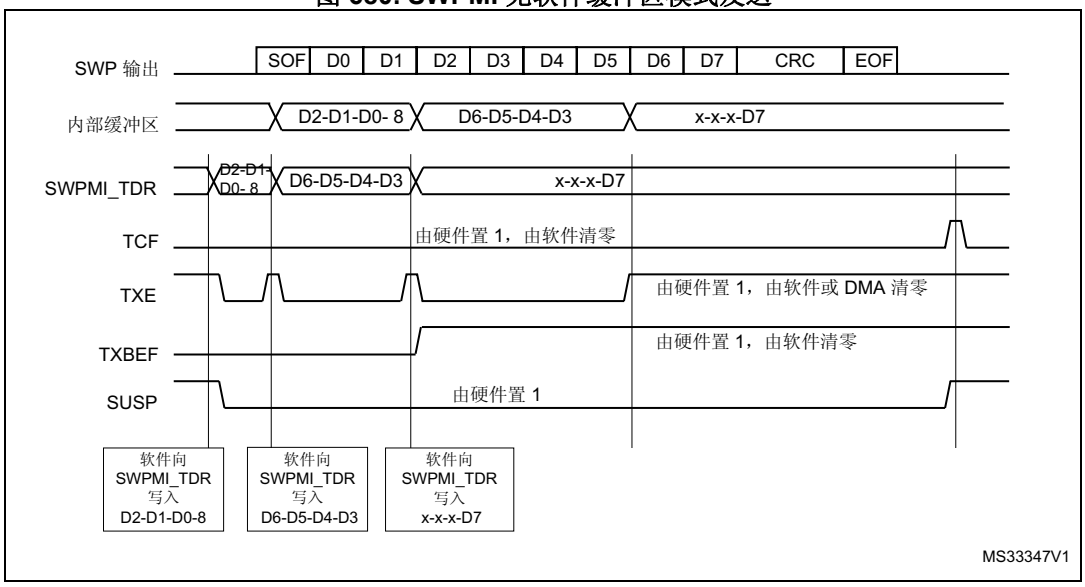
对 SWPMI_TDR 寄存器进行写操作将引发以下操作：

- 如果 SWP 总线状态为“已挂起”，则会发送转换序列和 8 个空闲位（主器件恢复）（如果 SWP 总线状态为“已激活”，则不会引发该操作）。
- 发送帧起始 (SOF)。
- 根据 SWPMI_TRD 寄存器内容发送有效负载。如果有效负载中的字节数大于 3，则只要 SWPMI_ISR 寄存器中的 TXBEF 标志未置 1，每次 SWPMI_ISR 寄存器的 TXE 标志置 1 时，SWPMI_TDR 就都需要由软件重新填充。
- 发送 16 位 CRC（由 SWPMI 内核自动计算）。
- 发送帧结束 (EOF)。

软件对 SWPMI_TDR 寄存器执行写操作时，将自动清零 TXE 标志。

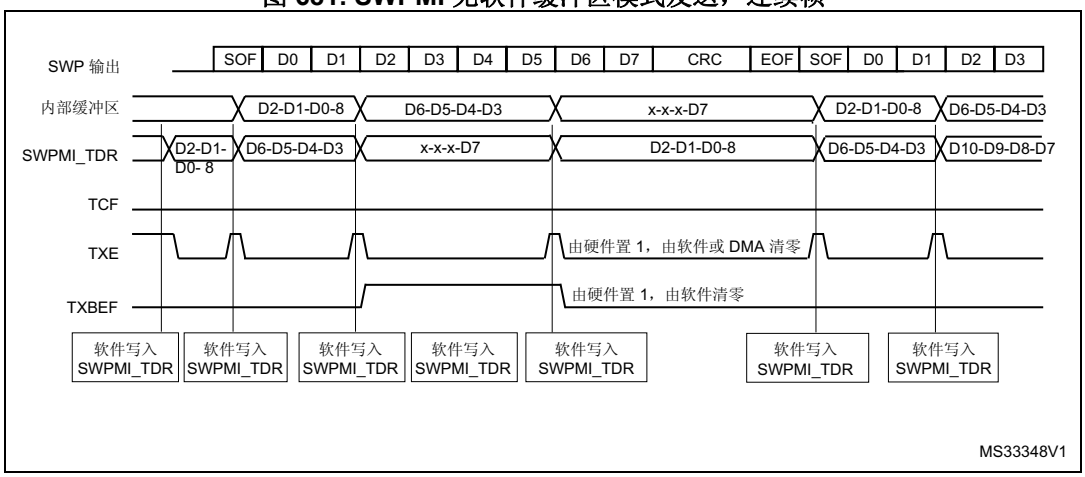
完整的帧完成发送后，如果尚未请求其他帧发送（即，在 TXBEF 标志置 1 后未再次对 SWPMI_TDR 进行写操作），则在 EOF 发送完成后再经过 7 个空闲位，SWFMI_ISR 寄存器中的 TCF 和 SUSP 标志会置 1，并且如果将 SWPMI_IER 寄存器中的 TCIE 位置 1，还将生成中断（请参见*图 680：SWPMI 无软件缓冲区模式发送*）。

图 680. SWPMI 无软件缓冲区模式发送



如果在 EOF 发送结束之前再一次请求了帧发送，则 TCF 标志不会置 1，且该帧将与前一个帧衔接（二者之间只有一个空闲位，请参见*图 681：SWPMI 无软件缓冲区模式发送，连续帧*）。

图 681. SWPMI 无软件缓冲区模式发送，连续帧



单软件缓冲区模式

在该模式下，可使用 DMA 发送完整 SWP 帧，而无需 CPU 干预。DMA 将重新填充 32 位 SWPMI_TDR 寄存器，软件可使用 SWPMI_TXBEF 标志轮询帧发送是否结束。

通过将 SWPMI_CR 寄存器中的 TXDMA 位置 1、TXMODE 位清零来选择单软件缓冲区模式。

DMA 通道或数据流必须配置为以下模式（请参见 DMA 部分）：

- 禁止存储器到存储器模式。
- 使能存储器递增模式。
- 存储器大小设为 32 位。
- 外设大小设为 32 位。
- 禁止外设递增模式。
- 禁止循环模式。
- 数据传输方向设为从存储器读取。
- 要传输的字数必须根据 SWP 帧长度进行设置。
- 源地址是 RAM 中的 SWP 帧缓冲区。
- 目标地址为 SWPMI_TDR 寄存器。

之后，用户必须：

1. 将 SWPMI_CR 寄存器中的 TXDMA 位置 1。
2. 将 SWPMI_IER 寄存器中的 TXBEIE 位置 1。
3. 对 RAM 存储器中的缓冲区进行填充（有效负载中的数据字节数由第一个字的最低有效字节表示）。
4. 使能 DMA 模块中的数据流或通道，以启动 DMA 传输和帧发送。

当 SWPMI_ISR 中的 TXE 标志置 1 时，SWPMI 会发出 DMA 请求。DMA 对 SWPMI_TDR 寄存器执行写操作时，将自动清零 TXE 标志。

在 SWPMI 中断程序中，用户必须检查 SWPMI_ISR 寄存器中的 TXBEF 位。如果该位置 1，且有其他帧需要发送，则用户必须：

1. 禁止 DMA 模块中的数据流或通道
2. 使用要发送的下一帧的内容更新 RAM 存储器中的缓冲区
3. 在 DMA 模块中配置要传输的总字数
4. 使能 DMA 模块中的数据流或通道，以启动下一次帧发送
5. 将 SWPMI_ICR 寄存器中的 CTXBEF 位置 1，以清零 TXBEF 标志

多软件缓冲区模式

该模式允许在 RAM 存储器中使用多个帧缓冲区，以确保连续发送、保持极低 CPU 负载，以及提供更多延迟以供软件进行缓冲区更新（使用 DMA）。软件可随时检查 DMA 计数器，并相应地在 RAM 存储器中更新 SWP 帧。

多软件缓冲区模式必须与循环模式下的 DMA 结合使用。

无论 SWP 帧有效负载中的字节数是多少，RAM 存储器中的每个发送缓冲区都必须具有八个 32 位字的固定长度。RAM 存储器中的发送缓冲区必须由软件填充，确保在两个连续缓冲区之间保持 8 字偏移。缓冲区的第一个数据字节表示帧有效负载的字节数。请参见*图 682：SWPMI 多软件缓冲区模式发送*中的缓冲区示例

通过将 SWPMI_CR 寄存器中的 TXDMA 和 TXMODE 位置 1 来选择多软件缓冲区模式。

例如，若要使用 4 个发送缓冲区，用户必须对 DMA 进行如下配置：

DMA 通道或数据流必须配置为以下模式（请参见 DMA 部分）：

- 禁止存储器到存储器模式。
- 使能存储器递增模式。
- 存储器大小设为 32 位。
- 外设大小设为 32 位。
- 禁止外设递增模式。
- 使能循环模式。
- 数据传输方向设为从存储器读取。
- 必须将要传输的字数设为 32（每个缓冲区 8 个字）。
- 源地址是 RAM 中的缓冲区 1。
- 目标地址为 SWPMI_TDR 寄存器。

之后，用户必须：

1. 将 SWPMI_CR 寄存器中的 TXDMA 置 1。
2. 将 SWPMI_IER 寄存器中的 TXBEIE 置 1。
3. 对 RAM 存储器中的缓冲区 1、缓冲区 2、缓冲区 3 和缓冲区 4 进行填充（有效负载中的数据字节数由第一个字的最低有效字节表示）。
4. 使能 DMA 模块中的数据流或通道，以启动 DMA。

在 SWPMI 中断程序中，用户必须检查 SWPMI_ISR 寄存器中的 TXBEF 位。如果该位置 1，则用户必须将 SWPMI_ICR 寄存器中的 CTXBEF 位置 1 以清零 TXBEF 标志，并且可更新 RAM 存储器中的缓冲区 1。

在出现下一个 SWPMI 中断程序时，用户将更新缓冲区 2，依此类推。

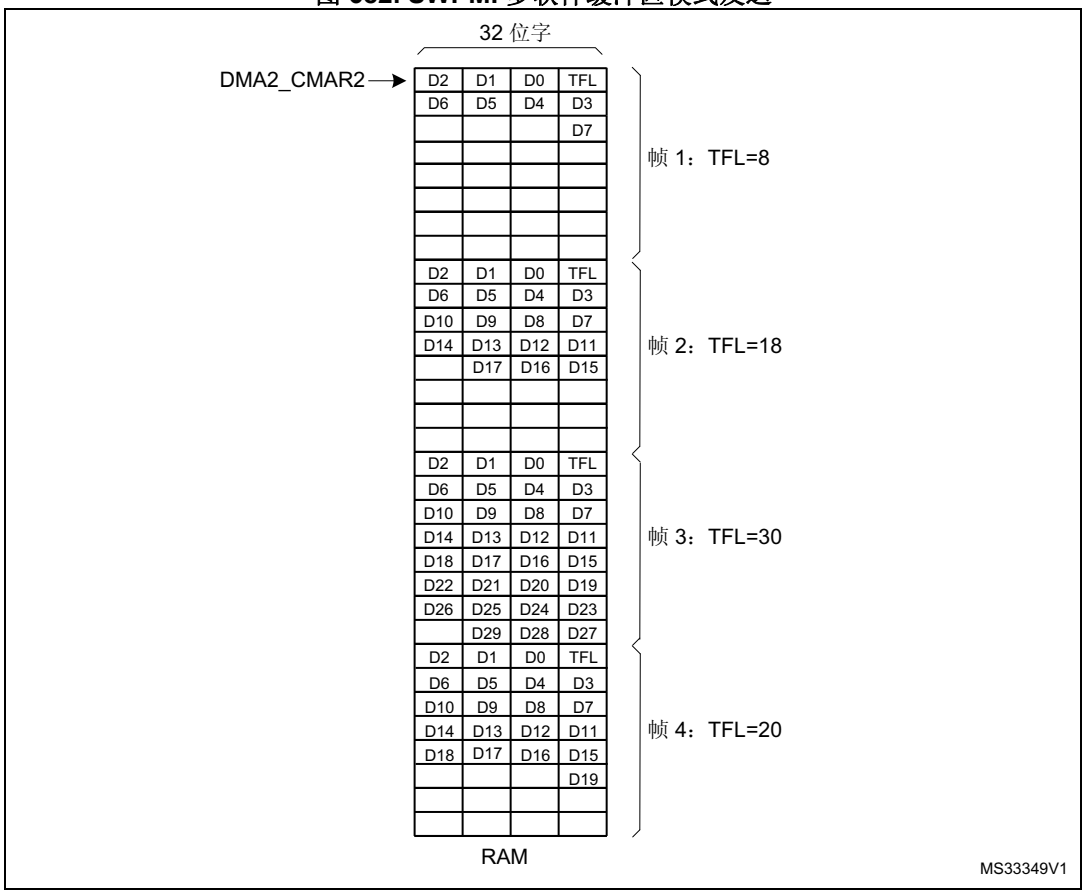
软件还可在 DMA 寄存器中读取 DMA 计数器（要传输的数据量），以检索已从 RAM 存储器传输并发送的帧。例如，如果软件使用 4 个发送缓冲区，并且 DMA 计数器值等于 17，则表示 RAM 区域中有两个缓冲区已准备好进行更新。这一功能很有用，它可以解决需要先发送几个帧之后软件才能处理 SWPMI 中断的情况。如果发生这种情况，软件必须更新多个缓冲区。

如果之后没有帧需要发送，用户必须禁止 DMA 模块的循环模式。发送将在缓冲区 4 发送结束时停止。

如果需要提前停止发送（例如，在缓冲区 2 结束时），用户必须将缓冲区 3 和缓冲区 4 中第一个字的低有效字节设为 0。

在第一个字的最低有效字节中读取到有效负载中的数据字节数为 0 时，SWPMI_CR 寄存器中的 TXDMA 位将由硬件清零。

图 682. SWPMI 多软件缓冲区模式发送



53.3.9 接收过程

在开始帧接收之前，用户必须激活 SWP（请参见*第 53.3.3 节：SWP 初始化和激活*）。

SWPMI_CR 寄存器中的 SWPACT 位置 1 后，“从器件恢复”状态会将 SWPMI_ISR 寄存器中的 SRF 标志置 1，并自动使能 SWPMI 以进行帧接收。

如果 SWP 总线处于“已激活”状态（例如，因正在进行帧发送），则 SWPMI 内核无需“从器件恢复”状态，可立即进行接收。

对于帧接收，可能的软件实现有以下几种：

- 无软件缓冲区模式。
- 单软件缓冲区模式。
- 多软件缓冲区模式。

使用软件缓冲区时，需要通过 DMA 通道将数据从 SWPMI 外设中的接收数据寄存器传输到 RAM 存储器中的软件缓冲区。

无软件缓冲区模式

在该模式下，无需使用 DMA。SWP 帧接收的处理通过在主循环或 SWPMI 中断程序内轮询状态标志来完成。SWPMI 中存在一个 32 位接收数据寄存器 (SWPMI_RDR)，在读取该寄存器之前允许最多接收 4 个字节。

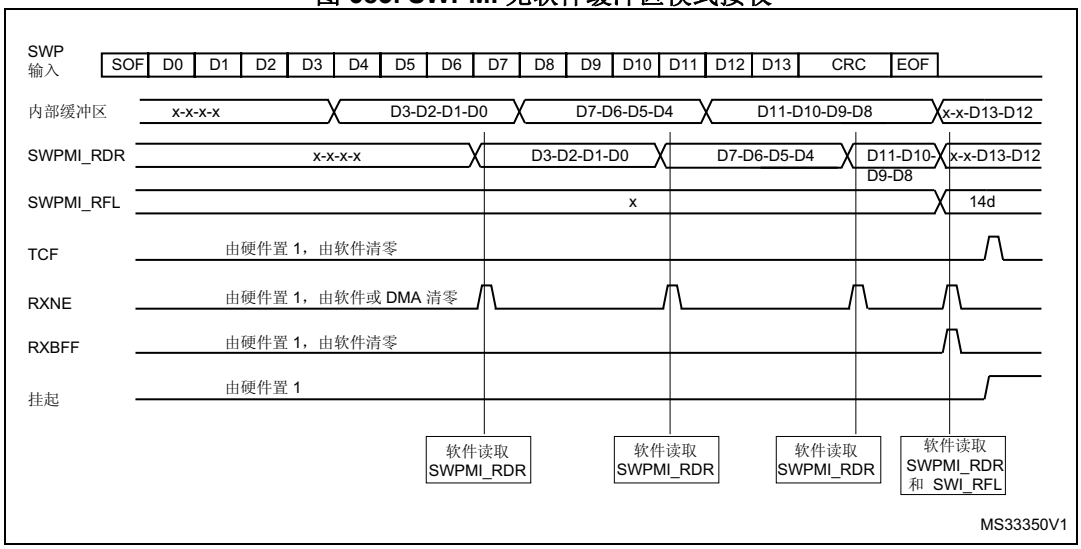
通过将 SWPMI_CR 寄存器中的 RXDMA 位复位来选择无软件缓冲区模式。

接收到帧起始 (SOF) 后，以下字节（有效负载）会存储在 SWPMI_RDR 寄存器中。在 SWPMI_RDR 已满时，SWPMI_ISR 中的 RXNE 标志将置 1，并且如果 SWPMI_IER 寄存器中的 RIE 位置 1，还将生成中断。用户可读取 SWPMI_RDR 寄存器，RXNE 标志会在软件读取 SWPMI_RDR 寄存器时自动清零。

一旦接收到完整的帧（包括 CRC 和帧结束 (EOF)），SWPMI_ISR 寄存器中的 RXNE 和 RXBFF 标志便会置 1。用户必须读取 SWPMI_RDR 寄存器中有效负载的最后一个（几个）字节，并将 SWPMI_ICR 中的 CRXBFF 标志置 1，以将 RXBFF 标志清零。可在 SWPMI_RFL 寄存器中获取有效负载中的数据字节数。当软件读取 SWPMI_RDR 寄存器时，RXNE 标志会再次自动复位（请参见*图 683：SWPMI 无软件缓冲区模式接收*）。

在 RXNE 清零时读取 SWPMI_RDR 寄存器将返回 0。

图 683. SWPMI 无软件缓冲区模式接收



单软件缓冲区模式

在该模式下，可使用 DMA 接收完整 SWP 帧，而无需 CPU 干预。DMA 会将接收到的数据从 32 位 SWPMI_RDR 寄存器传输到 RAM 存储器，软件可使用 SWPMI_RBFF 标志轮询帧接收是否结束。

通过将 SWPMI_CR 寄存器中的 RXDMA 位置 1、RXMODE 位清零来选择单软件缓冲区模式。

DMA 必须进行如下配置：

DMA 通道或数据流必须配置为以下模式（请参见 DMA 部分）：

- 禁止存储器到存储器模式。
- 使能存储器递增模式。
- 存储器大小设为 32 位。
- 外设大小设为 32 位。
- 禁止外设递增模式。
- 禁止循环模式。
- 数据传输方向设为从外设读取。
- 必须将要传输的字数设为 8。
- 源地址是 SWPMI_RDR 寄存器。
- 目标地址是 RAM 中的 SWP 帧缓冲区。

之后，用户必须：

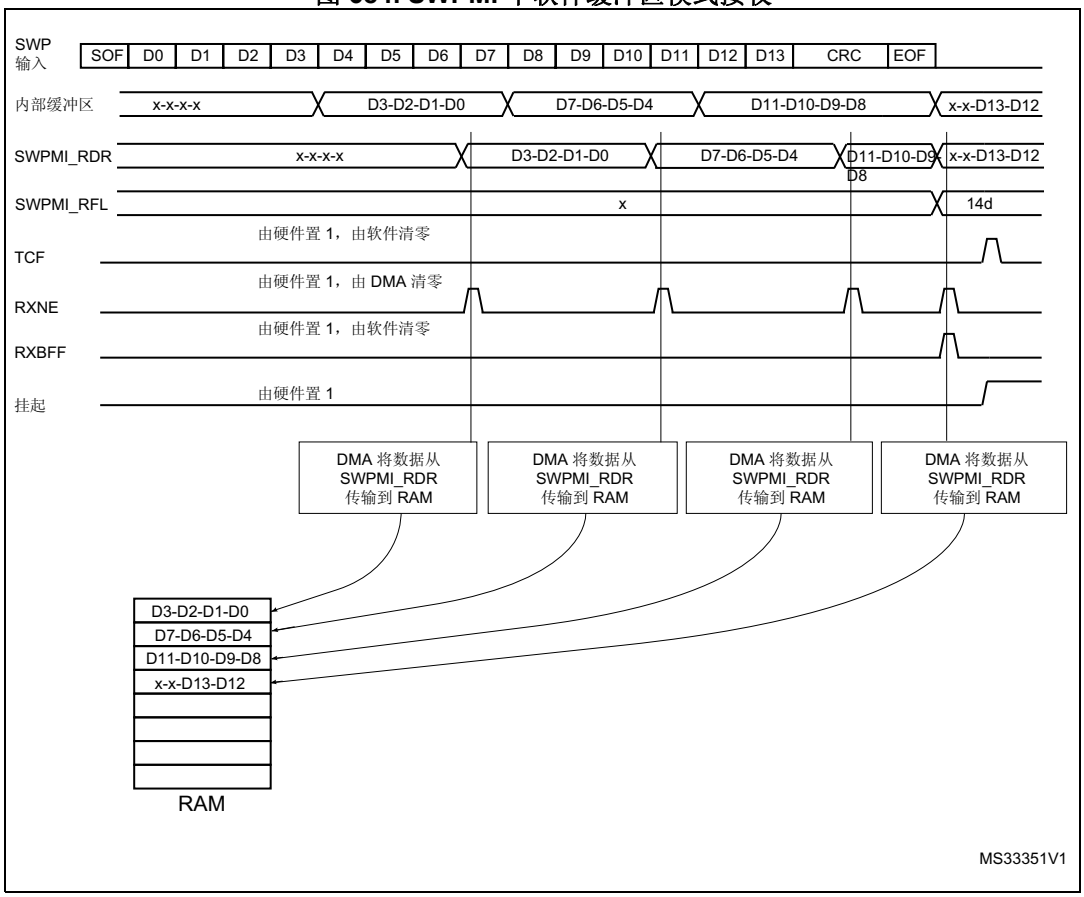
1. 将 SWPMI_CR 寄存器中的 RXDMA 位置 1。
2. 将 SWPMI_IER 寄存器中的 RXBFIE 位置 1。
3. 使能 DMA 模块中的数据流或通道。

当 SWPMI_ISR 中的 RXNE 标志置 1 时，SWPMI 会发出 DMA 请求。当 DMA 读取 SWPMI_RDR 寄存器时，RXNE 标志将自动清零。

在 SWPMI 中断程序中，用户必须检查 SWPMI_ISR 寄存器中的 RXBFF 位。如果该位置 1，则用户必须：

1. 禁止 DMA 模块中的数据流或通道。
2. 读取 SWPMI_RFL 寄存器中已接收的帧有效负载的字节数。
3. 读取 RAM 缓冲区中的帧有效负载。
4. 使能 DMA 模块中的数据流或通道。
5. 将 SWPMI_ICR 寄存器中的 CRXBFF 位置 1，以清零 RXBFF 标志（请参见*图 684：SWPMI 单软件缓冲区模式接收*）。

图 684. SWPMI 单软件缓冲区模式接收



多软件缓冲区模式

该模式允许在 RAM 存储器中使用多个帧缓冲区，以确保连续接收，保持极低 CPU 负载（使用 DMA）。帧有效负载与帧状态标志一起存储在 RAM 存储器中。软件可随时检查 DMA 计数器和状态标志，以处理 RAM 存储器中接收到的 SWP 帧。

多软件缓冲区模式必须与循环模式下的 DMA 结合使用。

通过将 SWPMI_CR 寄存器中的 RXDMA 和 RXMODE 位置 1 来选择多软件缓冲区模式。

为在 RAM 中使用 n 个接收缓冲区，DMA 通道或数据流必须配置为以下模式（请参见 DMA 部分）：

- 禁止存储器到存储器模式。
- 使能存储器递增模式。
- 存储器大小设为 32 位。
- 外设大小设为 32 位。
- 禁止外设递增模式。
- 使能循环模式。
- 数据传输方向设为从外设读取。
- 必须将要传输的字数设为 8 x n（每个缓冲区 8 个字）。
- 源地址是 SWPMI_TDR 寄存器。
- 目标地址是 RAM 中的缓冲区 1 地址。

之后，用户必须：

1. 将 SWPMI_CR 寄存器中的 RXDMA 置 1。
2. 将 SWPMI_IER 寄存器中的 RXBFIE 置 1。
3. 使能 DMA 模块中的数据流或通道。

在 SWPMI 中断程序中，用户必须检查 SWPMI_ISR 寄存器中的 RXBFF。如果该位置 1，用户必须将 SWPMI_ICR 寄存器中的 CRXBFF 位置 1 以清零 RXBFF 标志，并且可读取在第一个缓冲区（在 DMA2_CMAR1 中设置的 RAM 地址）中接收到的第一个帧有效负载。

可在倒数第 8 个字的位 [23:16] 中获取有效负载的数据字节数。

在出现下一个 SWPMI 中断程序时，用户将读取在第二个缓冲区（在 DMA2_CMAR1 中设置的地址 + 8）中接收到的第二个帧，依此类推（请参见*图 685：SWPMI 多软件缓冲区模式接收*）。

如果应用软件无法确保在下一次帧接收之前处理 SMPMI 中断，则会在第 8 个缓冲区字的最高有效字节中提供每个缓冲区状态：

- CRC 错误标志（相当于 SWPMI_ISR 寄存器中的 RXBERF 标志）通过第 8 个字的位 24 表示。有关 CRC 错误的说明，请参见*第 53.3.10 节：错误管理*。
- 接收上溢标志（相当于 SWPMI_ISR 寄存器中的 RXOVRF 标志）通过第 8 个字的位 25 表示。有关上溢错误的说明，请参见*第 53.3.10 节：错误管理*。
- 接收缓冲区已满标志（相当于 SWPMI_ISR 寄存器中的 RXBFF 标志）通过第 8 个字的位 26 表示。

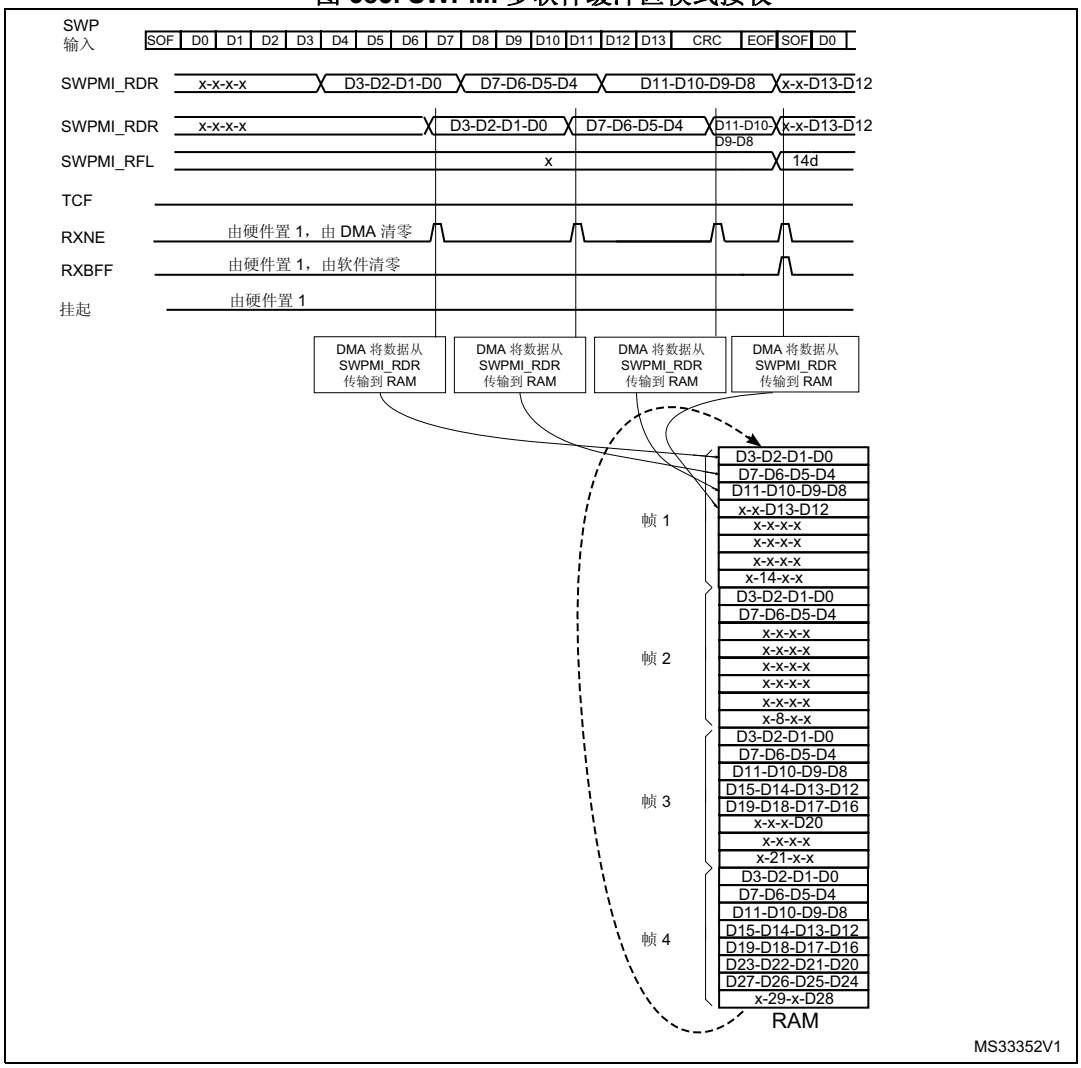
出现 CRC 错误时，RXBFF 和 RXBERF 标志均置 1，因此位 24 和位 26 均置 1。

发生上溢时，上溢标志置 1，因此位 25 置 1。只有在接收最后一个字时发生上溢的情况下，接收缓冲区已满标志才会置 1；之后，会针对当前和下一次帧接收将位 25 和位 26 置 1。

软件还可读取 DMA 寄存器中的 DMA 计数器（要传输的数据量），以便检索已接收并通过 DMA 传输到 RAM 存储器中的帧。例如，如果软件使用 4 个接收缓冲区，并且 DMA 计数器值等于 17，则表示在 RAM 区域中有两个缓冲区已准备就绪可供读取。

在多软件缓冲区接收模式下，如果软件正在读取第 8 个字的位 24、25 和 26，则在每次接收帧后都无需将 RXBERF、RXOVRF 和 RXBFF 标志清零。

图 685. SWPMI 多软件缓冲区模式接收



53.3.10 错误管理

有效负载发送期间发生下溢

在发送帧有效负载期间，发送下溢由 SWPMI_ISR 寄存器的 TXUNRF 标志指示。
如果 SWPMI_IER 寄存器中的 TXBUNREIE 位置 1，则会生成中断。

如果发生发送下溢，则 SWPMI 会停止有效负载发送，并发送损坏的 CRC（发送的第一个 CRC 字节的第一个位被反转），后跟 EOF。如果使用 DMA，则会自动清零 SWPMI_CR 寄存器中的 TXDMA 位。

TXUNRF 置 1 时，对 SWPMI_TDR 寄存器进行的后续写操作均会被忽略。
用户必须将 SWPMI_ICR 寄存器中的 CTXUNRF 位置 1 以清零 TXUNRF 标志。

有效负载接收期间发生上溢

在接收帧有效负载期间，接收上溢由 SWPMI_ISR 寄存器中的 RXOVRF 标志指示。如果发生接收上溢，则 SWPMI 不会使用传入的数据更新 SWPMI_RDR。传入的数据将丢失。

在 EOF 之前始终进行接收，如果上溢条件消失，则 RXBFF 标志会置 1。RXBFF 标志置 1 时，用户可检查 RXOVRF 标志。用户必须将 SWPMI_ICR 寄存器中的 CRXOVRF 位置 1 以清零 RXBOVRF 标志。

如果用户要立即检测上溢，则可将 SWPMI_IER 寄存器中的 RXBOVREIE 位置 1，以便在发生上溢时立即生成中断。

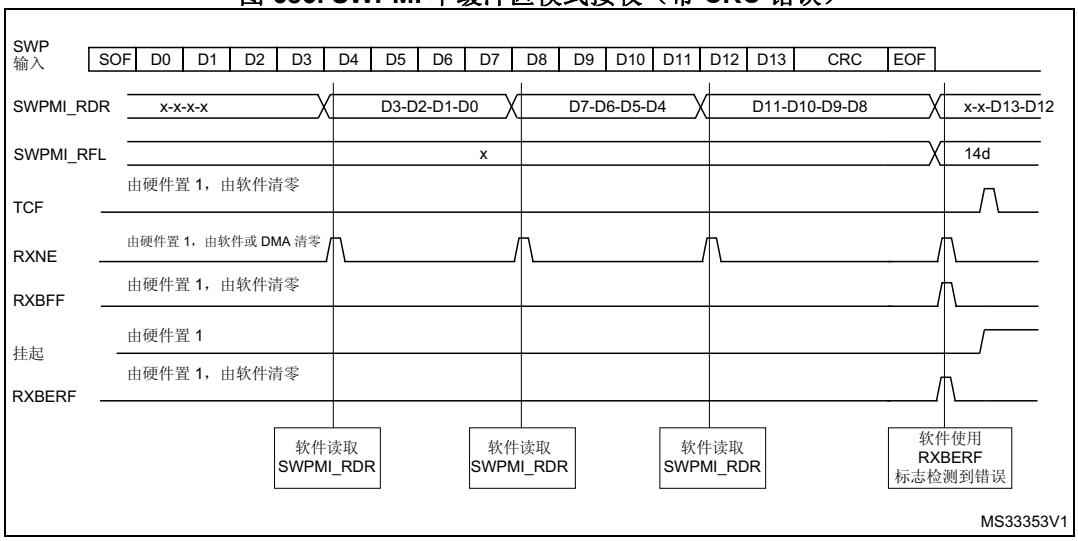
RXOVRF 标志与 RXNE 标志同时置 1，在发生上溢事件后会进行两次 SWPMI_RDR 读操作。这表示，至少有一个接收到的字节已丢失，并且 SWPMI_RDR 中加载的字包含发生上溢前刚接收到的字节。

在多软件缓冲区模式下，如果针对已接收帧的最后一个字将 RXOVRF 标志置 1，则会针对当前帧和下一帧将上溢位（第 8 个字的位 25）置 1。

有效负载接收期间发生 CRC 错误

接收到两个 CRC 字节后，如果 CRC 错误，则在完成 EOF 接收后，SWPMI_ISR 寄存器中的 RXBERF 标志会置 1。如果 SWPMI_IER 寄存器中的 RXBEIE 位置 1，则会生成中断（请参见*图 686：SWPMI 单缓冲区模式接收（带 CRC 错误）*）。用户必须将 SWPMI_ICR 中的 CRXBERF 位置 1 以清零 RXBERF 标志。

图 686. SWPMI 单缓冲区模式接收（带 CRC 错误）



有效负载接收期间丢失或损坏填充位

当有效负载中的填充位丢失或损坏时，在完成 EOF 接收后，SWPMI_ISR 中的 RXBERF 和 RXBFF 标志会置 1。

接收损坏的 EOF

接收到 SOF 后，SWPMI 会累计接收到的字节，直至接收到 EOF（忽略任何可能的 SOF）。接收到 EOF 后，SWPMI 已准备好开始新的帧接收并等待 SOF。

如果 EOF 损坏，则在接收到下一个 EOF 后，SWPMI_ISR 寄存器中的 RXBERF 和 RXBFF 标志将置 1。

注： *当接收到损坏的 EOF 时，仍会继续接收有效负载，因此如果接收到的字节数大于 30，则有效负载中的字节数可取值 31。有效负载中的字节数在 SWPMI_RFL 寄存器中读取，或者在 RAM 存储器缓冲区的第 8 个字的位 [23:16] 中读取，具体取决于工作模式。*

53.3.11 回送模式

可使用回送模式进行测试。用户必须将 SWPMI_CR 寄存器中的 LPBK 位置 1 才能使能回送模式。

使能回送模式时，SWPMI_TX 和 SWPMI_RX 信号连接在一起。因此，将收回 SWPMI 发送的所有帧。

53.4 SWPMI 低功耗模式

表 416. 低功耗模式对 SWPMI 的作用

模式	说明
睡眠	无影响。SWPMI 中断可使器件退出睡眠模式。
停止	如果 swpmi_ker_ck 为 HSI，则从器件发出的“从已挂起模式恢复”状态可将器件从停止模式唤醒（请参见*第 53.3.1 节：SWPMI 框图*）。
待机	SWPMI 会停止。

53.5 SWPMI 中断

所有 SWPMI 中断均连接至 NVIC。

要使能 SWPMI 中断，需按照以下顺序操作：

1. 配置 NVIC 中的 SWPMI 中断通道并将其使能。
2. 配置 SWPMI 以生成 SWPMI 中断（请参见 SWPMI_IER 寄存器）。

表 417. 中断控制位

中断事件	事件标志	启用控制位	退出睡眠模式	退出停止模式	退出待机模式
接收缓冲区已满	RXBFF	RXBFIE	是	否	否
发送缓冲区为空	TXBEF	TXBEIE	是	否	否
接收缓冲区错误（CRC 错误）	RXBERF	RXBEIE	是	否	否
接收缓冲区上溢	RXOVRF	RXBOVEREIE	是	否	否
发送缓冲区下溢	TXUNRF	TXBUNREIE	是	否	否
接收数据寄存器非空	RXNE	RIE	是	否	否
发送数据寄存器已满	TXE	TIE	是	否	否
传输完成标志	TCF	TCIE	是	否	否
从器件恢复标志	SRF	SRIE	是	是 (1)	否
收发器就绪标志	RDYF	RDYIE	是	否	否

1. 为 swpmi_ker_ck 选择 HSI 时。

53.6 SWPMI 寄存器

有关寄存器说明中使用的缩写，请参见参考手册中的*第 1.1 节*。

外设寄存器可按字（32 位）进行访问。

53.6.1 SWPMI 配置/控制寄存器 (SWPMI_CR)

SWPMI Configuration/Control register

偏移地址：0x00

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	SWPEN	DEACT	Res.	Res.	Res.	Res.	SWPACT	LPBK	TXMODE	RXMODE	TxDMA	RXDMA
				rw	rw					rw	rw	rw	rw	rw	rw

位 31:12 保留，必须保持复位值

位 11 **SWPTEN**：单线协议主收发器使能 (Single wire protocol master transceiver enable)

该位用于使能收发器以及通过 SWPMI 控制 SWPMI_IO（请参见*第 53.3.3 节：SWP 初始化和激活*）。

0：SPWMI_IO 引脚由 GPIO 控制器控制

1：SWPMI_IO 收发器由 SWPMI 控制

位 10 **DEACT**：单线协议主接口禁用 (Single wire protocol master interface deactivate)

该位用于请求 SWP“已禁用”状态。除了可能传入的“从器件恢复”状态会使 SWP 保持为“已激活”状态外，将该位置 1 与清零 SWPACT 的作用相同。

位 9:6 保留，必须保持复位值

位 5 **SWPACT**：单线协议主接口激活 (Single wire protocol master interface activate)

该位用于激活 SWP 总线（请参见*第 53.3.3 节：SWP 初始化和激活*）。

0：SWPMI_IO 被下拉至地，SWP 总线切换到“已禁用”状态

1：SWPMI_IO 被释放，SWP 总线切换到“已挂起”状态

为了能将 SWPACT 位置 1，必须先清零 DEACT 位。

位 4 **LPBK**：回送模式使能 (Loopback mode enable)

此位用于使能回送模式

0：禁止回送模式

1：使能回送模式

注： SWPACT 位置 1 时，该位无法写入。

位 3 **TXMODE**：发送缓冲模式 (Transmission buffering mode)

此位用于选择发送缓冲模式。只有在 TXDMA 位置 1 时，该位才相关（请参见*表 418：针对发送/接收选择缓冲区模式*）。

0：针对发送将 SWPMI 配置为单软件缓冲区模式。

1：针对发送将 SWPMI 配置为多软件缓冲区模式。

注： SWPACT 位置 1 时，该位无法写入。

位 2 **RXMODE**：接收缓冲模式 (Reception buffering mode)

此位用于选择接收缓冲模式。只有在 TXDMA 位置 1 时，该位才相关（请参见*表 418：针对发送/接收选择缓冲区模式*）。

0：针对接收将 SWPMI 配置为单软件缓冲区模式。

1：针对接收将 SWPMI 配置为多软件缓冲区模式。

注： SWPACT 位置 1 时，该位无法写入。

位 1 **TXDMA**：发送 DMA 使能 (Transmission DMA enable)

该位用于针对发送使能 DMA 模式

0：针对发送禁止 DMA

1：针对发送使能 DMA

注： 如果将已发送帧的有效负载大小指定为 0x00 （在帧的第一个字的 TDR 最低有效字节中），TXDMA 将自动清零。TXDMA 在发生下溢事件时 （SWP_ISR 寄存器的 TXUNRF 标志置 1 时）也会自动清零

位 0 **RXDMA**：接收 DMA 使能 (Reception DMA enable)

此位用于针对接收使能 DMA 模式

0：针对接收禁止 DMA

1：针对接收使能 DMA

表 418. 针对发送/接收选择缓冲区模式

缓冲区模式	无软件缓冲区	单软件缓冲区	多软件缓冲区
RXMODE/TXMODE	x	0	1
RXDMA/TXDMA	0	1	1

53.6.2 SWPMI 比特率寄存器 (SWPMI_BRR)

SWPMI Bitrate register

偏移地址：0x04

复位值：0x0000 0001

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

位 31:8 保留，必须保持复位值

位 7:0 **BR[7:0]**：比特率预分频器 (Bitrate prescaler)

必须按照以下公式，在考虑到 RCC（复位和时钟控制）中编程的 $F_{swpmi_ker_ck}$ 的情况下编程该位域以设置 SWP 总线比特率：

$F_{SWP} = F_{swpmi_ker_ck} / ((BR[7:0]+1)x4)$

注： 编程的比特率必须处于以下范围内：100 kbit/s 到 2 Mbit/s。

SWPMI_CR 寄存器中的 SWPACT 位置 1 时，不能写入 BR[7:0]。

53.6.3 SWPMI 中断和状态寄存器 (SWPMI_ISR)

SWPMI Interrupt and Status register

偏移地址：0x0C

复位值：0x0000 02C2

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	RDYF	DEACTF	SUSP	SRF	TCF	TXE	RXNE	TXUNRF	RXOVRF	RXBERF	TXBEF	RXBFF
				r	r	r	r	r	r	r	r	r	r	r	r

位 31:12 保留，必须保持复位值

位 11 **RDYF**：收发器就绪标志 (transceiver ready flag)

当收发器就绪时，该位立即由硬件置 1。通过将 SWPMI_CR 寄存器中的 SWPTEN 位置 1 以使能 SWPMI_IO 收发器后，软件必须等待此标志置 1 后再将 SWPACT 位置 1 以激活 SWP 总线。

0：收发器未就绪

1：收发器已就绪

位 10 **DEACTF**：已禁用标志 (DEACTIVATED flag)

该位是一个状态标志，用于确认进入“已禁用”模式的请求。

0：SWP 总线处于“已激活”状态或“已挂起”状态

1：SWP 总线处于“已禁用”状态

如果在 DEACT 位由软件置 1 时 SWPMI 检测到“从器件恢复”状态，则 SRF 标志会置 1，DEACTF 不会置 1，SWP 会进入“已激活”状态。

位 9 **SUSP**：挂起标志 (SUSPEND flag)

该位是一个状态标志，用于报告 SWP 总线状态

0：SWP 总线处于“已激活”状态

1：SWP 总线处于“已挂起”或“已禁用”状态

位 8 **SRF**：从器件恢复标志 (Slave resume flag)

该位由硬件置 1，用于指示检测到“从器件恢复”状态。该位由软件清零，方法是向 SWPMI_ICR 寄存器中的 CSRF 位写入 1。

0：未检测到“从器件恢复”状态

1：在 SWP 总线处于“已挂起”状态时检测到“从器件恢复”状态

位 7 **TCF**：传输完成标志 (Transfer complete flag)

当发送和接收均已完成并且 SWP 切换到“已挂起”状态时，该标志将立即由硬件置 1。该位由软件清零，方法是向 SWPMI_ICR 寄存器中的 CTCF 位写入 1。

0：发送或接收未完成

1：发送和接收均已完成并且 SWP 切换到“已挂起”状态

位 6 **TXE**：发送数据寄存器为空 (Transmit data register empty)

该标志指示发送数据寄存器状态

0：尚未发送写入到发送数据寄存器 SWPMI_TDR 中的数据

1：已发送写入到发送数据寄存器 SWPMI_TDR 中的数据，可再次对 SWPMI_TDR 进行写操作

位 5 **RXNE**：接收数据寄存器非空 (Receive data register not empty)

该标志指示接收数据寄存器状态

0：SWPMI_RDR 寄存器中未接收到数据

1：SWPMI_RDR 寄存器中接收到的数据已准备就绪可供读取

位 4 **TXUNRF**：发送下溢错误标志 (Transmit underrun error flag)

该标志由硬件置 1，用于指示在有效负载发送期间发生下溢，即，软件或 DMA 未及时对 SWPMI_TDR 进行写操作。该标志由软件清零，方法是向 SWPMI_ICR 寄存器中的 CTXUNRF 位写入 1。

0：发送期间无下溢错误

1：在发送期间检测到下溢错误

位 3 **RXOVRF**：接收上溢错误标志 (Receive overrun error flag)

该标志由硬件置 1，用于指示在有效负载接收期间发生上溢，即，软件或 DMA 未及时对 SWPMI_RDR 进行读操作。该标志由软件清零，方法是向 SWPMI_ICR 寄存器中的 CRXOVRF 位写入 1。

0：接收期间无上溢错误

1：在接收期间检测到上溢错误

位 2 **RXBERF**：接收 CRC 错误标志 (Receive CRC error flag)

该标志由硬件置 1，用于指示接收到的帧中存在 CRC 错误。它与 RXBFF 标志同步置 1。该位由软件清零，方法是向 SWPMI_ICR 寄存器中的 CRXBERF 位写入 1。

0：接收期间无 CRC 错误

1：在接收期间检测到 CRC 错误

位 1 **TXBEF**：发送缓冲区为空标志 (Transmit buffer empty flag)

该标志由硬件置 1，用于指示无需进一步进行 SWPMI_TDR 更新来完成当前帧发送。该位由软件清零，方法是向 SWPMI_ICR 寄存器中的 CTXBEF 位写入 1。

0：帧发送缓冲区尚未清空

1：帧发送缓冲区已清空

位 0 **RXBFF**：接收缓冲区已满标志 (Receive buffer full flag)

当 SWPMI_RDR 中包含接收帧的最后一个字时，该标志由硬件置 1。该标志由软件清零，方法是向 SWPMI_ICR 寄存器中的 CRXBFF 位写入 1。

0：接收帧的最后一个字尚未传到 SWPMI_RDR

1：接收帧的最后一个字已传到 SWPMI_RDR

53.6.4 SWPMI 中断标志清零寄存器 (SWPMI_ICR)

SWPMI Interrupt Flag Clear register

偏移地址：0x10

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	CRDYF	Res.	Res.	CSRF	CTCF	Res.	Res.	CTXUNRF	CRXOVRF	CRXBERF	CTXBEF	CRXBFF
				rc_w1			rc_w1	rc_w1			rc_w1	rc_w1	rc_w1	rc_w1	rc_w1

位 31:12 保留，必须保持复位值

位 11 **CRDYF**：清零收发器就绪标志 (Clear transceiver ready flag)
将 1 写入此位时，SWPMI_ISR 寄存器中的 RDYF 标志将清零
向该位写入 0 无任何影响

位 10:9 保留，必须保持复位值

位 8 **CSRF**：清零从器件恢复标志 (Clear slave resume flag)
将 1 写入此位时，SWPMI_ISR 寄存器中的 SRF 标志将清零
向该位写入 0 无任何影响

位 7 **CTCF**：清零传输完成标志 (Clear transfer complete flag)
将 1 写入此位时，SWPMI_ISR 寄存器中的 TCF 标志将清零
向该位写入 0 无任何影响

位 6:5 保留，必须保持复位值

位 4 **CTXUNRF**：清零发送下溢错误标志 (Clear transmit underrun error flag)
将 1 写入此位时，SWPMI_ISR 寄存器中的 TXUNRF 标志将清零
向该位写入 0 无任何影响

位 3 **CRXOVRF**：清零接收上溢错误标志 (Clear receive overrun error flag)
将 1 写入此位时，SWPMI_ISR 寄存器中的 RXBOCREF 标志将清零
向该位写入 0 无任何影响

位 2 **CRXBERF**：清零接收 CRC 错误标志 (Clear receive CRC error flag)
将 1 写入此位时，SWPMI_ISR 寄存器中的 RXBERF 标志将清零
向该位写入 0 无任何影响

位 1 **CTXBEF**：清零发送缓冲区为空标志 (Clear transmit buffer empty flag)
将 1 写入此位时，SWPMI_ISR 寄存器中的 TXBEF 标志将清零
向该位写入 0 无任何影响

位 0 **CRXBFF**：清零接收缓冲区已满标志 (Clear receive buffer full flag)
将 1 写入此位时，SWPMI_ISR 寄存器中的 RXBFF 标志将清零
向该位写入 0 无任何影响

53.6.5 SWPMI 中断使能寄存器 (SMPMI_IER)

SWPMI Interrupt Enable register

偏移地址：0x14

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	RDYIE	Res.	Res.	SRIE	TCIE	TIE	RIE	TXUNRIE	RXOVRIE	RXBEIE	TXBERIE	RXBFIE
				rw			rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:12 保留，必须保持复位值

位 11 **RDYIE**：收发器就绪中断使能 (Transceiver ready interrupt enable)
0：禁止中断
1：SWPMI_ISR 寄存器中的 RDYF 标志置 1 时生成 SWPMI 中断

位 10:9 保留，必须保持复位值

位 8 **SRIE**：从器件恢复中断使能 (Slave resume interrupt enable)
0：禁止中断
1：SWPMI_ISR 寄存器中的 SRF 标志置 1 时生成 SWPMI 中断

位 7 **TCIE**：发送完成中断使能 (Transmit complete interrupt enable)
0：禁止中断
1：SWPMI_ISR 寄存器中的 TCF 标志置 1 时生成 SWPMI 中断

位 6 **TIE**：发送中断使能 (Transmit interrupt enable)
0：禁止中断
1：SWPMI_ISR 寄存器中的 TXE 标志置 1 时生成 SWPMI 中断

位 5 **RIE**：接收中断使能 (Receive interrupt enable)
0：禁止中断
1：SWPMI_ISR 寄存器中的 RXNE 标志置 1 时生成 SWPMI 中断

位 4 **TXUNRIE**：发送下溢错误中断使能 (Transmit underrun error interrupt enable)
0：禁止中断
1：SWPMI_ISR 寄存器中的 TXBUNRF 标志置 1 时生成 SWPMI 中断

位 3 **RXOVRIE**：接收上溢错误中断使能 (Receive overrun error interrupt enable)
0：禁止中断
1：SWPMI_ISR 寄存器中的 RXBOVRF 标志置 1 时生成 SWPMI 中断

位 2 **RXBERIE**：接收 CRC 错误中断使能 (Receive CRC error interrupt enable)
0：禁止中断
1：SWPMI_ISR 寄存器中的 RXBERF 标志置 1 时生成 SWPMI 中断

位 1 **TXBEIE**：发送缓冲区为空中断使能 (Transmit buffer empty interrupt enable)
0：禁止中断
1：SWPMI_ISR 寄存器中的 TXBEF 标志置 1 时生成 SWPMI 中断

位 0 **RXBFIE**：接收缓冲区已满中断使能 (Receive buffer full interrupt enable)
0：禁止中断
1：SWPMI_ISR 寄存器中的 RXBFF 标志置 1 时生成 SWPMI 中断

53.6.6 SWPMI 接收帧长度寄存器 (SWPMI_RFL)

SWPMI Receive Frame Length register

偏移地址：0x18

复位值：0x0000 0000

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

位 31:5 保留，必须保持复位值

位 4:0 **RFL[4:0]**：接收帧长度 (Receive frame length)

RFL[4:0] 表示已接收帧的有效负载中的数据字节数。两个最低有效位 RFL[1:0] 表示最后一次 SWPMI_RDR 寄存器读取操作的相关字节数。

53.6.7 SWPMI 发送数据寄存器 (SWPMI_TDR)

SWPMI Transmit data register

偏移地址：0x1C

复位值：0x0000 0000

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

位 31:0 **TD[31:0]**：发送数据 (Transmit data)

包含要发送的数据。

对该寄存器进行写操作会触发 SOF 发送或下一次有效负载数据发送，并会清零 TXE 标志。

53.6.8 SWPMI 接收数据寄存器 (SWPMI_RDR)

SWPMI Receive data register

偏移地址：0x20

复位值：0x0000 0000

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

位 31:0 **RD[31:0]**：已接收数据 (received data)
包含接收到的数据
读取该寄存器会清零 RXNE 标志。

53.6.9 SWPMI 选项寄存器 (SWPMI_OR)

SWPMI Option register

偏移地址：0x24

复位值：0x0000 0000

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

位 31:2 保留，必须保持复位值

位 1 **SWP_CLASS**：SWP 类别选择 (SWP class selection)
该位用于选择 SWP 类别（请参见*第 53.3.3 节：SWP 初始化和激活*）。
0：C 类：SWPMI_IO 直接使用 VDD 电压来工作在 C 类下。
VDD 处于 [1.62 V 到 1.98 V] 范围内时，必须选择此配置
1：B 类：SWPMI_IO 使用内部稳压器来工作在 B 类下。
VDD 处于 [2.70 V 到 3.30 V] 范围内时，必须选择此配置

位 0 **SWP_TBYP**：SWP 收发器旁路 (SWP transceiver bypass)
该位用于旁路内部收发器 (SWPMI_IO)，以及连接外部收发器。
0：使能内部收发器。SWPMI 的外部接口为 SWPMI_IO（SWPMI_RX、SWPMI_TX 和 SWPMI_SUSPEND 信号在 GPIO 上不可用）
1：禁止内部收发器。SWPMI_RX、SWPMI_TX 和 SWPMI_SUSPEND 信号作为 GPIO 上的复用功能可供使用。通过选择此配置，可以连接外部收发器

53.6.10 SWPMI 寄存器映射和复位值表

表 419. SWPMI 寄存器映射和复位值

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	SWPMI_CR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SWPTEN	DEACT	Res.	Res.	Res.	Res.	SWPACT	LPBK	TXMODE	RXMODE	TxDMA	RXDMA
	Reset value																					0	0					0	0	0	0	0	0
0x04	SWPMI_BRR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BRR[7:0]							
	Reset value																									0	0	0	0	0	0	0	1
0x08	RESERVED	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	Reset value																																
0x0C	SWPMI_ISR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RDYF	DEACTF	SUSP	SRF	TCF	TXE	RXNE	TXUNRF	RXOVRF	RXBERF	TXBEF	RXBFF
	Reset value																					0	0	1	0	1	1	0	0	0	0	1	0
0x10	SWPMI_ICR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CRDYF	Res.	Res.	CSRF	CTCF	Res.	Res.	CTXUNRF	CRXOVRF	CRXBERF	CTXBEF	CRXBFF
	Reset value																					0			0	0			0	0	0	0	0
0x14	SWPMI_IER	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RDYIE	Res.	Res.	SRIE	TCIE	TIE	RIE	TXUNRIE	RXOVRIE	RXBERIE	TXBEIE	RXBFIE
	Reset value																					0			0	0	0	0	0	0	0	0	0
0x18	SWPMI_RFL	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RFL[4:0]				
	Reset value																												0	0	0	0	0
0x1C	SWPMI_TDR	TD[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x20	SWPMI_RDR	RD[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x24	SWPMI_OR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SWP_CLASS	SWP_TBYP
	Reset value																															0	0

有关寄存器边界地址的信息，请参见*第 100 页的第 2.2.2 节*。

54 管理数据输入/输出 (MDIOS)

54.1 MDIOS 简介

如果系统中的主芯片需要管理（配置和获取状态数据）一个或多个从芯片，MDIO 总线会十分有用。该总线协议只使用两个信号：

- MDC：管理数据时钟
- MDIO：携带操作码（写入或读取）、从（端口）地址、MDIOS 寄存器地址和数据的数据线

在每个事务中，主器件会读取其中一个从器件的 MDIOS 寄存器内容，或者将数据写入其中一个从器件的 MDIOS 寄存器。

MDIOS 外设用作 MDIO 总线的从接口。MDIO 主器件可以使用 MDC/MDIO 线来写入和读取 32 个 MDIOS 中保存的 16 位 MDIOS 寄存器。这些 MDIOS 寄存器由固件管理，从而允许 MDIO 主器件配置在 STM32 上运行的应用程序，并从中获取状态信息。

MDIOS 可以在停止模式下工作，如果 MDIO 主器件对其中一个 MDIOS 寄存器进行读取或写入，则可以选择唤醒 STM32。

54.2 MDIOS 主要特性

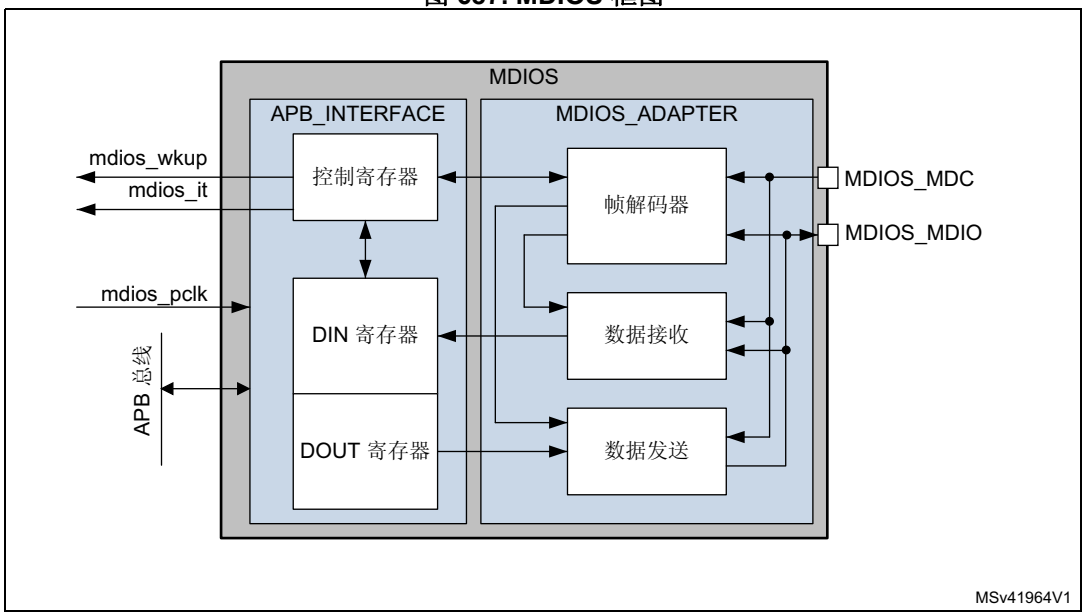
MDIOS 包括下列特性：

- 32 个 MDIOS 寄存器地址，每个地址使用单独的输入和输出数据寄存器进行管理：
  - 32 x 16 位固件读/写、MDIOS 只读输出数据寄存器
  - 32 x 16 位固件只读、MDIOS 只写输入数据寄存器
- 可配置从（端口）地址
- 可单独屏蔽的中断/事件：
  - MDIOS 寄存器写入
  - MDIOS 寄存器读取
  - MDIOS 协议错误
- 能够在停止模式下工作以及从停止模式唤醒

54.3 MDIOS 功能说明

54.3.1 MDIOS 框图

图 687. MDIOS 框图



54.3.2 MDIOS 引脚和内部信号

表 420 列出了连接到封装引脚或焊球的 MDIOS 输入与输出信号，*表 421* 则列出了内部 PWR 信号。

表 420. 连接到封装引脚或焊球的 MDIOS 输入/输出信号

信号名称	信号类型	说明
MDIOS_MDC	数字输入	MDIO 主时钟
MDIOS_MDIO	数字输入/输出	MDIO 信号（操作码、地址和输入/输出数据）

表 421. MDIOS 内部输入/输出信号

信号名称	信号类型	说明
mdios_wkup	数字输出	MDIOS 唤醒信号
mdios_it	数字输出	MDIOS 中断信号
mdios_pclk	数字输入	APB 时钟

54.3.3 MDIOS 协议

MDIOS 协议使用两个信号：

1. MDIOS_MDC：时钟，始终由主器件驱动
2. MDIOS_MDIO：携带操作码、地址和双向数据的信号

每个事务都使用“帧”来执行。每个帧包含 32 位：14 个控制位、2 个周转位以及 16 个数据位，每个位均以串行方式传送。

- 14 个控制位，由主器件驱动
  - 2 个起始位：始终为“01”
  - 2 个操作码位：读=“10”，写=“01”
  - 5 个端口地址位，指示哪个从器件正在进行寻址
  - 5 个 MDIOS 寄存器地址位，每个从器件中可寻址多达 32 个 MDIOS 寄存器
- 2 个周转状态位
  - 写操作时，主器件驱动“10”
  - 读操作时，第一个位为高阻抗，第二个位由从器件驱动为“0”
- 16 个数据位
  - 写操作时，写入从器件的 MDIOS 寄存器的数据由主器件驱动
  - 读操作时，从从器件的 MDIOS 寄存器读取的数据由从器件驱动

每个帧前面通常都有一个报头，其中 MDIOS 在 32 个 MDC 时钟内保持为“1”。当主器件没有帧要发送时，可继续使 MDIO 保持为“1”，表示“空闲”状态。

当主器件驱动 MDIO 信号时，MDIOS 将使用 MDC 的上升沿采样该信号。当 MDIOS 驱动 MDIO 时，输出将在 MDC 的上升沿发生变化。

图 688. MDIO 协议写入帧波形

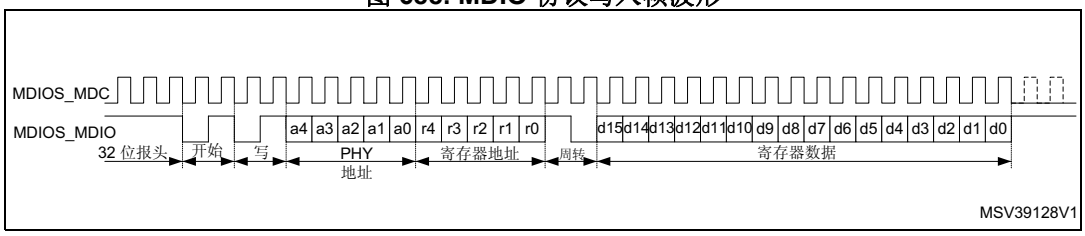
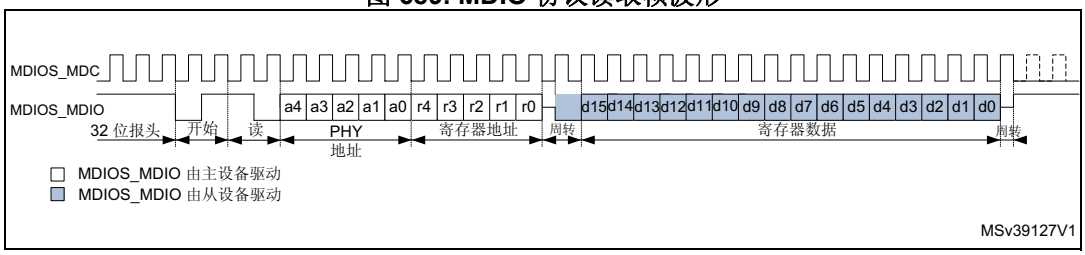


图 689. MDIO 协议读取帧波形



54.3.4 MDIOS 使能和禁止

通过将 MDIOS_CR 寄存器中的 EN 位置 1 来使能 MDIOS。当 EN = 1 时，MDIOS 将对 MDIO 总线以及寻址到其中一个 MDIOS 寄存器的服务帧进行监视。

当使能 MDIOS（将 EN 置“1”）时，必须通过对 MDIOS_CR 寄存器执行的相同写操作正确设置 PORT_ADDRESS[4:0] 位域以指示从端口地址。如果某个帧的端口地址与 PORT_ADDRESS[4:0] 不同（推测是用于另一个从器件），则 MDIOS 将忽略该帧。

当 EN = 0 时，MDIOS 将忽略在 MDC/MDIO 线路上传输的帧，并且 IP 处于低功耗模式。将 EN 清零也会将所有 DIN 寄存器清零。如果 EN 在 MDIOS 驱动读取数据时清零，则它将立即释放总线，而不会驱动其余的数据。如果 EN 在 MDIOS 接收帧时清零，则帧将被中止，数据将丢失。

如果 MDIOS 先使能、接着禁止、随后重新使能，则状态标志不会清零。为了实现正确操作，固件应在重新使能 MDIOS 之前将状态标志清零。

54.3.5 MDIOS 数据

从 MDIO 主器件的角度来看，MDIOS 中有 32 个 16 位 MDIOS 寄存器可供写入和读取。实际上，每个 MDIOS 寄存器“n”都对应有两组寄存器：DINn[15:0]和 DOUTn[15:0]。

输入数据

当 MDIO 主器件发送一个帧，而该帧要写入 MDIOS 中的 MDIOS 寄存器“n”时，传入数据要更新的是 DINn[15:0] 寄存器。DIN 寄存器 (DIN0 - DIN31) 可由固件读取，但只能由 MDIO 主器件通过 MDIO 总线写入。

当采样最后一个数据位时，DINn 的内容会在 MDC 上升沿之后立即发生变化。

如果固件碰巧读取正在更新的 DINn 的内容，则读取的值有可能损坏（旧值和新值之间逐位交叉）。因此，**固件应确保对同一 DINn 寄存器的两次后续读操作给出相同的值，从而保证之前读取的数据是稳定的。**在最坏的情况下，固件需要读取 DINn 四次：第一次获取旧值，第二次获取非相干值（在寄存器变化时读取），第三次获取新值，第四次确认新值。

如果固件使用 WRF 中断，并且可以保证在任何新的 MDIOS 写入帧完成之前读取 DINn 寄存器，则固件可执行单次读操作。

如果 MDIO 主器件执行写操作且寄存器地址大于 31，则 MDIOS 将忽略相应帧（数据未保存且没有标志置 1）。

输出数据

当 MDIOS 接收到请求读取寄存器“n”的帧时，它将返回 DOUTn[15:0] 寄存器中的值。因此，如果 MDIO 主器件期望读取先前写入 MDIOS 寄存器“n”的值，固件必须在每次将新数据写入 DINn 时便将数据从 DINn 复制到 DOUTn。为了实现正确操作，固件必须在一个报头（如果主器件在每个帧之前都发送报头）加上 15 个周期的时间内将数据复制到 DOUTn 寄存器。

当通过 MDIO 总线读取 MDIOS 寄存器时，MDIOS 在读取帧的第 15 个周期中将 16 位值（来自相应的 DOUTn 寄存器）传送到 MDIOS 时钟域。如果固件尝试在 MDIO 主器件正读取 MDIOS 寄存器“n”时写入 DOUTn 寄存器，并且该固件写操作发生在帧的第 15 个周期内（在一个 MDC 周期窗口内），则将被忽略。因此，**在写入 DOUTn 寄存器后，固件应对同一 DOUTn 寄存器执行回读操作，以确认值确实已写入**。如果 DOUTn 寄存器不包含写入的值，则固件只需尝试再次写入并重新读取确认。

如果 MDIOS 频率与 mdios_pclk 频率相比极慢，最好不要连续写入和重新读取 DOUT 寄存器，以免过度占用 CPU。请注意，只要 DOUTn 值传送到 MDIOS 时钟域，读标志（RDFn）便会置 1。因此，当忽略对 DOUTn 执行的写操作时（当读回的值不是刚刚写入的值时），固件可以使用读取中断来获悉何时能够写入 DOUTn。

以下是 MDC 时钟极慢时可供使用的操作步骤：

1. 写入 DOUTn。
2. 确保所有读取标志均为 0 (MDIOS_RDFR = 0x0000)。必要时使用 MDIOS_CRDFR 将标志清零。
3. 对同一 DOUTn 寄存器执行回读操作，并将读取的值与步骤 1 中写入的值进行比较。
4. 如果这两个值相同，则该过程已完成。否则，继续执行步骤 5。
5. 通过将 MDIOS_CR1 中的 RDIE 位置 1 来使能读取中断。
6. 在中断程序中，确保 RDFn 置 1。（其他任何读取标志都不会在位 n 之前置 1）。
7. 请确保有一段“31 个周期 + 报头”的时间窗口（如果主器件在每个帧之前都发送一个报头）可供安全写入 DOUTn，而无需执行回读和比较操作。如果无法保证此最大延时，请返回步骤 #1。

如果 MDIO 主器件执行读操作且寄存器地址大于 31，则 MDIOS 将返回全零的数据值。

54.3.6 MDIOS APB 频率

每当固件读取 MDIOS_DINRn 寄存器或写入 MDIOS_DOUTRn 寄存器时，APB 总线的频率必须至少为 MDC 频率的 1.5 倍。例如，如果 MDC 为 20MHz，则 APB 必须为 30MHz 或更高频率。

54.3.7 写入/读取标志和中断

当通过 MDIO 总线写入 MDIOS 寄存器“n”时，MDIOS_WRFR 寄存器中的 WRFn 位将置 1。WRFn 在 DINn 更新时（即采样写入帧上的最后一个数据位时）变为“1”。如果 MDIOS_CR 寄存器中 WRIEN=1，则会生成中断。WRFn 由软件清零，方法是向 MDIOS_CWRFR 寄存器中的 CWRFn 位写入“1”。

当通过 MDIO 总线读取 MDIOS 寄存器 “n”时， MDIOS_RDFR 寄存器中的 RDFn 位将置 1。RDFn 在 DOUTn 复制到 MDC 时钟域时（即读取帧的第 15 个周期）变为“1”。如果 MDIOS_CR 寄存器中 RDIEN=1，则会生成中断。 RDFn 由软件清零，方法是向 MDIOS_CRDFR 寄存器中的 CRDFn 位写入 “1”。

54.3.8 MDIOS 错误管理

下面列出了三种错误及其相应的错误标志：

- 报头错误：PERF（MDIOS_SR 寄存器的位 0）
- 起始错误：SERF（MDIOS_SR 寄存器的位 1）
- 周转错误：TERF（MDIOS_SR 寄存器的位 2）

各个错误标志都将在出现相应的错误条件时由硬件置 1。可以通过向清零标志寄存器 (MDIOS_CLRFR) 中的相应位写入“1”将各个标志清零。

如果 EIE = 1 (MDIOS_CR) 时三个错误标志中的任何一个置 1，则会发生中断。

除了将错误标志置 1 外，MDIOS 不会对检测到错误的帧执行任何操作：数据阶段既不更新 DINn 寄存器也不强制 MDIO 线。

对于给定的帧，错误不会累积。例如，如果检测到报头错误，则不会检查当前帧的其余部分是否存在起始错误或周转错误。

当 DPC = 0 时，在检测到错误之后，所有新帧和错误将被忽略，直到检测到完整的报头。

当 DPC = 1（禁止报头检查，MDIOS_CR[7]）时，只要其中一个错误标志置 1，所有帧和新错误都将被忽略。错误位清零后，MDIOS 将立即开始寻找起始序列。因此，只有当确定没有帧正在接受处理时，应用程序才能够将错误标志清零。否则，MDIOS 可能会误解正在发送的位，并与主器件失去同步。

报头错误

报头错误出现在起始序列开始（MDIO 采样为“0”）时，而不会紧接在报头之后（MDIO 采样为“1”并至少持续 32 个连续时钟）。

在接收到完整的报头之前，首次使能 MDIOS（MDIOS_CR 中的 EN=1）之后不会报告报头错误。这样一来，即使在报头或帧正在接受处理时使能外设帧检测，也可以避免出现错误条件。在这种情况下，MDIOS 会忽略第一个帧（因为它先前没有检测到完整的报头），而不会将 PERF 置 1。

如果 DPC 位（禁止报头检查，MDIOS_CR[7]）置 1，则 MDIO 主器件可发送不带报头的帧，并且不会报告报头错误。当 DPC = 1 时，应用程序必须确保主器件未在 MDIOS 使能（EN 置 1）时发送帧。否则，从器件可能会与主器件失去同步。

起始错误

当出现非法起始序列或者发出非法命令时，会出现起始错误。起始序列必须始终为“01”，命令必须为“01”（写）或“10”（读）。

与报头错误一样，在接收到完整的报头之前，不会报告起始错误。

周转错误

当在写入帧的周转位中检测到错误时，会出现周转错误。写入帧的第 15 位必须为“1”，第 16 位必须为“0”。

仅当满足以下条件时才会报告周转错误：接收到完整的报头、没有起始错误、当前帧中的端口地址匹配、并且寄存器地址处于受支持的范围内（0 到 31）。

54.3.9 停止模式下的 MDIOS

MDIOS 可以在停止模式下运行，响应所有读操作、执行所有写操作，并在 MDIOS 中断时将 STM32 从停止模式唤醒。

54.3.10 MDIOS 中断

有三种类型的中断（写入、读取和错误）对应同一个中断向量。这些中断源中的任何一个都可以将 STM32 从停止模式唤醒。如要清除中断线，需将所有中断标志均清零。

表 422. 中断控制位

中断事件	事件标志	使能控制位
中断	WRF[31:0]	WRIE
中断	RDF[31:0]	RDIE
错误中断	PERF（报头），SERF（起始），TERF（周转）	EIE

54.4 MDIOS 寄存器

54.4.1 MDIOS 配置寄存器 (MDIOS_CR)

MDIOS configuration register

偏移地址：0x00

复位值：0x0000 0000

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

位 31:13 保留，必须保持复位值

位 12:8 **PORT_ADDRESS[4:0]**：从器件地址 (Slave's address)
只能在外设已禁止时 (EN=0) 写入。
如果 MDIO 主器件提供的地址与 PORT_ADRESS[4:0] 匹配，则 MDIOS 将处理帧。否则将忽略帧。

位 7 **DPC**：禁止报头检查 (Disable Preamble Check)
0：MDIO 主器件必须在每个帧之前提供报头。
1：MDIO 主器件发送每个帧时可以不带报头，MDIOS 不会报告报头错误。
当该位置 1 时，应用程序必须确保 MDIOS 使能时未在处理任何帧。否则，MDIOS 可能会与主器件失去同步。
除非 EN = 0，否则该位不能更改（但它可以在 EN 置 1 的同时进行更改）。

位 6:4 保留，必须保持复位值

位 3 **EIE**：错误中断使能 (Error interrupt enable)。
0：禁止中断
1：使能中断
该位置 1 时，如果任一错误标志（MDIOS_SR 寄存器中的 PERF、SERF 或 TERF）置 1，则会生成中断。

位 2 **RDIE**：寄存器读取中断使能 (Register Read Interrupt Enable)
0：禁止中断
1：使能中断
该位置 1 时，如果任一读取标志（MDIOS_RDFR 寄存器中的 RDF[31:0]）置 1，则会生成中断。

位 1 **WRIE**：寄存器写入中断使能 (Register write interrupt enable)
0：禁止中断
1：使能中断
该位置 1 时，如果任一读取标志（MDIOS_WRFR 寄存器中的 WRF[31:0]）置 1，则会生成中断。

位 0 **EN**：外设使能 (Peripheral enable)。
0：禁止 MDIOS
1：MDIOS 使能且正在监视 MDIO 总线 (MDC/MDIO)

54.4.2 MDIOS 写入标志寄存器 (MDIOS_WRFR)

MDIOS write flag register

偏移地址：0x04

上电复位值：0x0000 0000

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

位 31:0 **WRF[31:0]**：MDIOS 寄存器 0 到 31 的写入标志 (Write flags for MDIOS registers 0 to 31)

各个位在 MDIO 主器件对相应的 MDIOS 寄存器执行写操作时由硬件置 1。如果 MDIOS_CR 中的 WRIE 置 1，则会生成中断。

各个位由软件清零，方法是向 MDIOS_CWRFR 寄存器中相应的 CWRF 位写入“1”。

对于 WRFn：

0：MDIO 主器件尚未对 MDIOS 寄存器“n”进行写操作

1：MDIO 主器件已对 MDIOS 寄存器“n”进行写操作，数据可从 MDIOS_DINRn 寄存器中的 DINn[15:0] 中获取

54.4.3 MDIOS 清零写入标志寄存器 (MDIOS_CWRFR)

MDIOS clear write flag register

偏移地址：0x08

上电复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CWRF[31:16]															
w_r0	w_r0	w_r0	w_r0	w_r0	w_r0	w_r0	w_r0	w_r0	w_r0	w_r0	w_r0	w_r0	w_r0	w_r0	w_r0
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CWRF[15:0]															
w_r0	w_r0	w_r0	w_r0	w_r0	w_r0	w_r0	w_r0	w_r0	w_r0	w_r0	w_r0	w_r0	w_r0	w_r0	w_r0

位 31:0 **CWRF[31:0]**：清零写入标志 (Clear the write flag)

向 CWRFn 写入“1”会将 MDIOS_WRF 寄存器中的 WRFn 位清零。

54.4.4 MDIOS 读取标志寄存器 (MDIOS_RDFR)

MDIOS read flag register

偏移地址：0x0C

上电复位值：0x0000 0000

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

位 31:0 **RDF[31:0]**：MDIOS 寄存器 0 到 31 的读取标志 (Read flags for MDIOS registers 0 to 31)
各个位在 MDIO 主器件对相应的 MDIOS 寄存器执行读操作时由硬件置 1。如果 MDIOS_CR 中的 RDIE 置 1，则会生成中断。
各个位由软件清零，方法是向 MDIOS_CRDFR 寄存器中相应的 CRDF 位写入“1”。
对于 RDFn：
0：MDIO 主器件尚未对 MDIOS 寄存器“n”进行读操作
0：MDIO 主器件已对 MDIOS 寄存器“n”进行读操作

54.4.5 MDIOS 清零读取标志寄存器 (MDIOS_CRDFR)

MDIOS clear read flag register

偏移地址：0x10

上电复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CRDF[31:16]															
w_r0	w_r0	w_r0	w_r0	w_r0	w_r0	w_r0	w_r0	w_r0	w_r0	w_r0	w_r0	w_r0	w_r0	w_r0	w_r0
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CRDF[15:0]															
w_r0	w_r0	w_r0	w_r0	w_r0	w_r0	w_r0	w_r0	w_r0	w_r0	w_r0	w_r0	w_r0	w_r0	w_r0	w_r0

位 31:0 **CRDF[31:0]**：清零读取标志 (Clear the read flag)
向 CRDFn 写入“1”会将 MDIOS_RDF 寄存器中的 RDFn 位清零。

54.4.6 MDIOS 状态寄存器 (MDIOS_SR)

MDIOS status register

偏移地址：0x14

上电复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TERF	SERF	PERF
													r	r	r

位 31:3 保留，必须保持复位值

位 2 TERF：周转错误标志 (Turnaround error flag)
0：未出现周转错误
1：已出现周转错误
向 CTERF (MDIOS_CLRFR) 写入“1”会将该位清零。

位 1 SERF：起始错误标志 (Start error flag)
0：未出现起始错误
1：已出现起始错误
向 CSERF (MDIOS_CLRFR) 写入“1”会将该位清零。

位 0 PERF：报头错误标志 (Preamble error flag)
0：未出现报头错误
1：已出现报头错误
向 CPERF (MDIOS_CLRFR) 写入“1”会将该位清零。
如果 DPC（禁用报头检查，MDIOS_CR[7]）置 1，则该位将不会置 1。

注： *对 MDIOS_SR 执行写入操作不起作用。*

54.4.7 MDIOS 清零标志寄存器 (MDIOS_CLRFR)

MDIOS clear flag register

偏移地址：0x18

上电复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CTERF	CSERF	CPERF
													w_r0	w_r0	w_r0

位 31:3 保留，必须保持复位值

位 2 **CTERF**：清零周转错误标志 (Clear the turnaround error flag)
向该位写入“1”时会将 TERF 标志 (MDIOS_SR) 清零。
当 DPC =“1”(MDIOS_CR[7]) 时，TERF 标志只能在未处理任何帧时清零。

位 1 **CSERF**：清零起始错误标志 (Clear the start error flag)
向该位写入“1”时会将 SERF 标志 (MDIOS_SR) 清零。
当 DPC =“1”(MDIOS_CR[7]) 时，SERF 标志只能在未处理任何帧时清零。

位 0 **CPERF**：清零报头错误标志 (Clear the preamble error flag)
向该位写入“1”时会将 PERF 标志 (MDIOS_SR) 清零。

注： *读取 MDIOS_CLRFR 将返回全零。*

54.4.8 MDIOS 输入数据寄存器 (MDIOS_DINR0-MDIOS_DINR31)

MDIOS input data register

偏移地址：0x100-0x17C

复位值：0x0000_0000

31	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

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

位 31:16 保留，必须保持复位值

位 15:0 **DINn[15:0]**：在写入帧期间从 MDIO 主器件接收的输入数据 (Input data received from MDIO Master during write frames)

该位域由硬件写入 16 位数据，而写入的 16 位数据是在寻址到 MDIOS 寄存器“n”的写入帧中接收到的。

54.4.9 MDIOS 输出数据寄存器 (MDIOS_DOUTR0-MDIOS_DOUTR31)

MDIOS output data register

偏移地址：0x180-0x1FC

复位值：0x0000_0000

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

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

位 31:16 保留，必须保持复位值

位 15:0 **DOUTn[15:0]**：读取帧期间发送到 MDIO 主器件的输出数据 (Output data sent to MDIO Master during read frames)

该位域由软件写入。在寻址 MDIOS 寄存器“n”的读取帧期间，这 16 位在 MDIO 总线上以串行方式输出。

54.4.10 MDIOS 寄存器映射

表 423. MDIOS 寄存器映射和复位值

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	**MDIOS_CR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PORT_ADDRESS[4:0]					Res.	Res.	Res.	Res.	EEI	RDIE	WRIE	EN
	Reset value																				0	0	0	0	0					0	0	0	0
0x04	**MDIOS_WRFR**	WRF[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x08	**MDIOS_CWRFR**	CWRF[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0C	**MDIOS_RDFR**	RDF[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x10	**MDIOS_CRDFR**	CRDF[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x14	**MDIOS_SR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TERF	SERF	PERF
	Reset value																														0	0	0
0x18	**MDIOS_CLRFR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CTERF	CSERF	CPERF
	Reset value																														0	0	0
0x1C - 0xFC	**Reserved**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
0x100	**MDIOS_DINR0**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DIN0[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x104	**MDIOS_DINR1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DIN1[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
...																																	
0x17C	**MDIOS_DINR31**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DIN31[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x180	**MDIOS_DOUTR0**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DOUT0[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x184	**MDIOS_DOUTR1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DOUT1[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
...																																	

表 423. MDIOS 寄存器映射和复位值（续）

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x1FC	**MDIOS_DOUTR31**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DOUT31[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

55 安全数字输入/输出多媒体卡接口 (SDMMC)

55.1 SDMMC 主要特性

SD/SDIO 多媒体卡 (MMC) 主机接口 (SDMMC) 提供 AHB 总线与 SD 存储卡、SDIO 卡以及 MMC 设备之间的接口。

多媒体卡协会网站 www.mmca.org 中提供了由 MMCA 技术委员会发布的多媒体卡系统规范。

SD 卡协会网站 www.sdcard.org 中提供了 SD 存储卡和 SD I/O 卡系统规范。

SDMMC 具有以下特性：

- 完全兼容*多媒体卡系统规范版本 4.51*。
  卡支持三种不同数据总线模式：1 位（默认）、4 位和8 位。
- 完全兼容先前版本的多媒体卡（向后兼容性）。
- 完全兼容 *SD 存储卡规范版本 4.1*。
  （SDR104 SDMMC_CK 速度限制为允许的最大 I/O 速度，不支持 SPI 模式和 UHS-II 模式）。
- 完全兼容 *SDIO 卡规范版本 4.0*。
  卡支持两种不同数据总线模式：1 位（默认）和 4 位。
  （SDR104 SDMMC_CK 速度限制为允许的最大 I/O 速度，不支持 SPI 模式和 UHS-II 模式）。
- 对于 8 位模式，数据传输高达 208 MB/s。
  （取决于允许的最大 I/O 速度）。
- 数据和命令输出使能信号，控制外部双向驱动程序。

多媒体卡/SD 总线将卡连接到主机。

当前版本的 SDMMC 在同一时间只支持一个协议栈版本为 MMC V4.51 或先前版本的 SD/SDIO/MMC 卡。

55.2 SDMMC 总线拓扑

总线的通信基于命令/响应和数据传输。

SD/SDIO/MMC 总线上的基本事务是命令/响应事务。这些类型的总线事务直接在命令或响应结构中传输其信息。此外，某些操作具有数据令牌。

数据传输通过以下方式完成：

- 块模式：块大小为 2^N 字节的数据块，N 为 0-14 范围内的值。
- SDIO 多字节模式：块大小为 1-512 字节的单一数据块。
- MMC 流模式：连续数据流。

与 MMC 卡的数据相互传输在数据块或流中执行。

图 690. SDMMC“无响应”和“无数据”操作

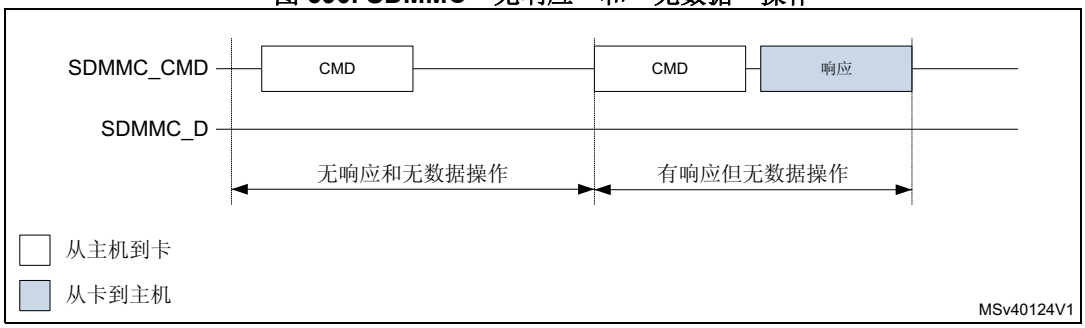
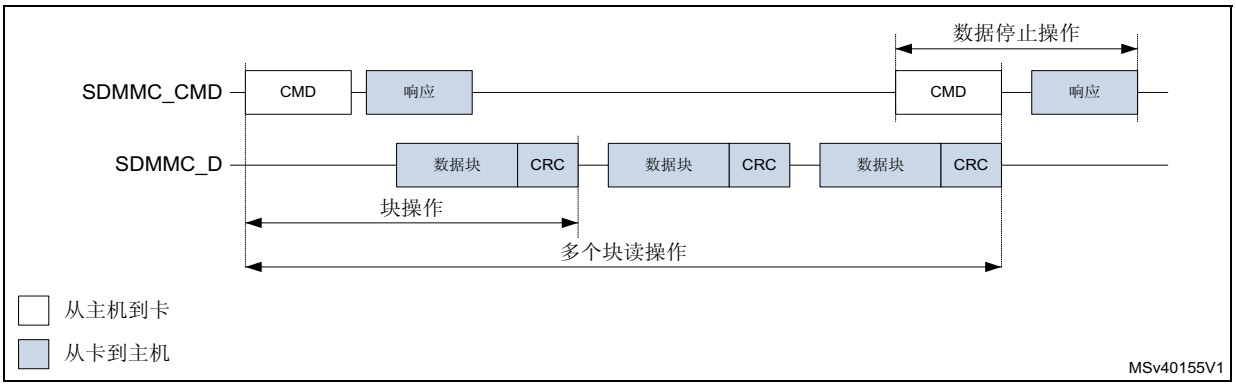
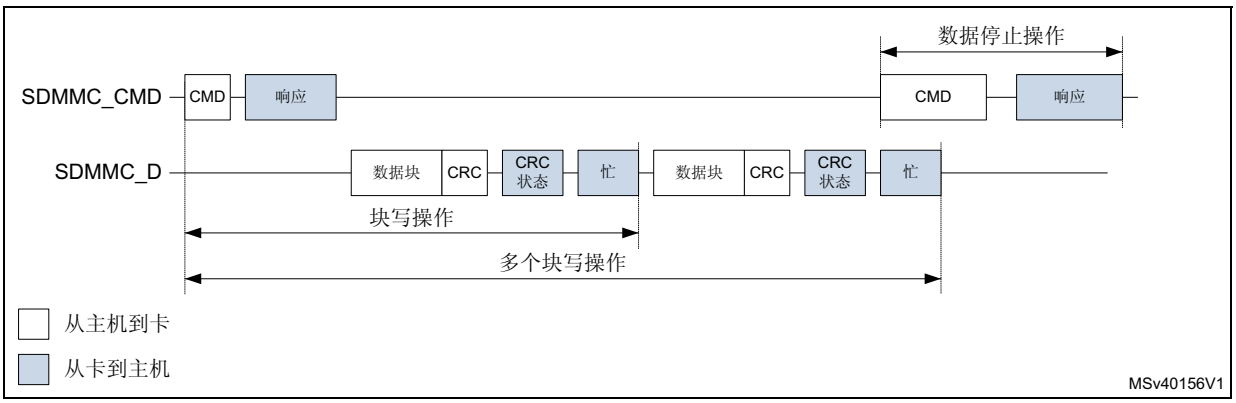


图 691. SDMMC（多个）块读取操作



注：　在具有预定义块数的 MMC 多块读取操作结束时，不需要停止传输命令。

图 692. SDMMC（多个）块写入操作



注：　在具有预定义块数的 MMC 多块写入操作结束时，不需要停止传输命令。

注：　只要发出“繁忙”信号（SDMMC_D0 被拉到低电平），SDMMC 便不会发送任何数据。

图 693. SDMMC（连续）流读取操作

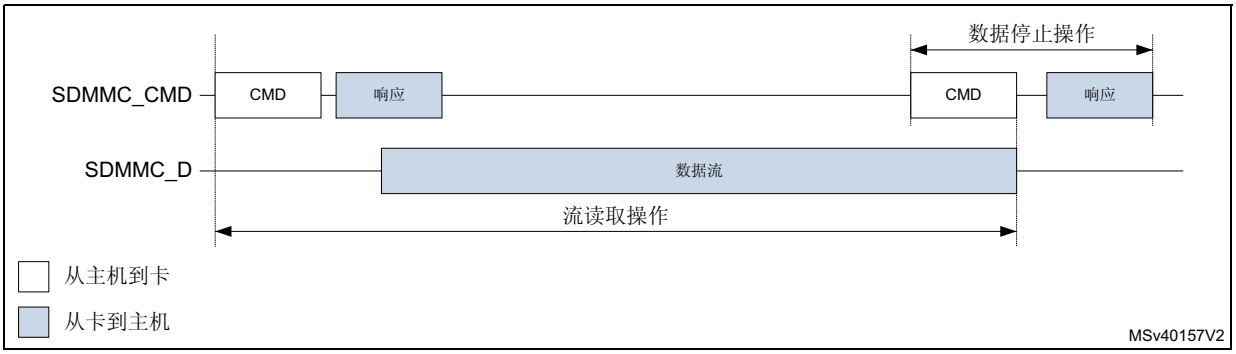
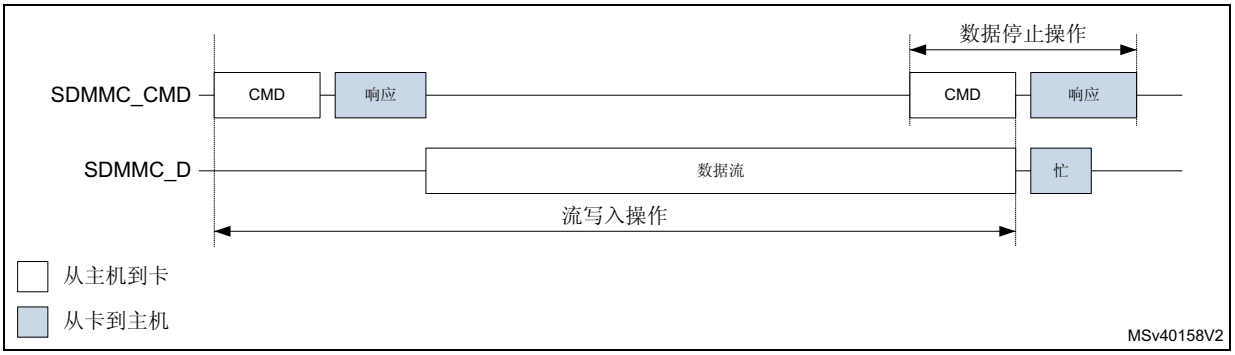


图 694. SDMMC（连续）流写入操作



数据流传输操作只适用于总线配置位宽为 1，且工作模式为单倍速率模式（DS、HS 和 SDR）下的 SDMMC_D0 上。

55.3 SDMMC 工作模式

表 424. SDMMC 工作模式 SD 和 SDIO

SDIO 总线速度模式(1)(2)	最大总线速度 (3) [MB/s]	最大时钟频率 [MHz]	信号电压 [V]
DS（默认速度）	12.5	25	3.3
HS（高速）	25	50	3.3
SDR12	12.5	25	1.8
SDR25	25	50	1.8
DDR50	50	50	1.8
SDR50	50	100	1.8
SDR104	104	208(4)	1.8

1. SDR 单倍数据速率信号传输。
2. DDR 双倍数据速率信号传输。（在 SDMMC_CK 的两种时钟边沿上均采样数据。）
3. SDIO 总线速度，4 位总线宽度。
4. 最大频率取决于允许的最大 I/O 速度。

SDR104 模式需要使用一个可变延迟来支持采样点的调节，而对于 SDR50 模式来说，这个可变延迟则是可选的。

表 425. SDMMC 工作模式 eMMC

eMMC 总线速度模式 (1)(2)	最大总线速度 (3) [MB/s]	最大时钟频率 [MHz]	信号电压 [V]
传统兼容型	26	26	3/1.8/1.2V
高速 SDR	52	52	3/1.8/1.2V
高速 DDR	104	52	3/1.8/1.2V
高速 HS200	200	200(4)	1.8/1.2V

1. SDR 单倍数据速率信号传输。
2. DDR 双倍数据速率信号传输。（在 SDMMC_CK 的两种时钟边沿上均采样数据。）
3. eMMC 总线速度，8 位总线宽度。
4. 最大频率取决于允许的最大 I/O 速度。

55.4 SDMMC 功能说明

SDMMC 由三部分组成：

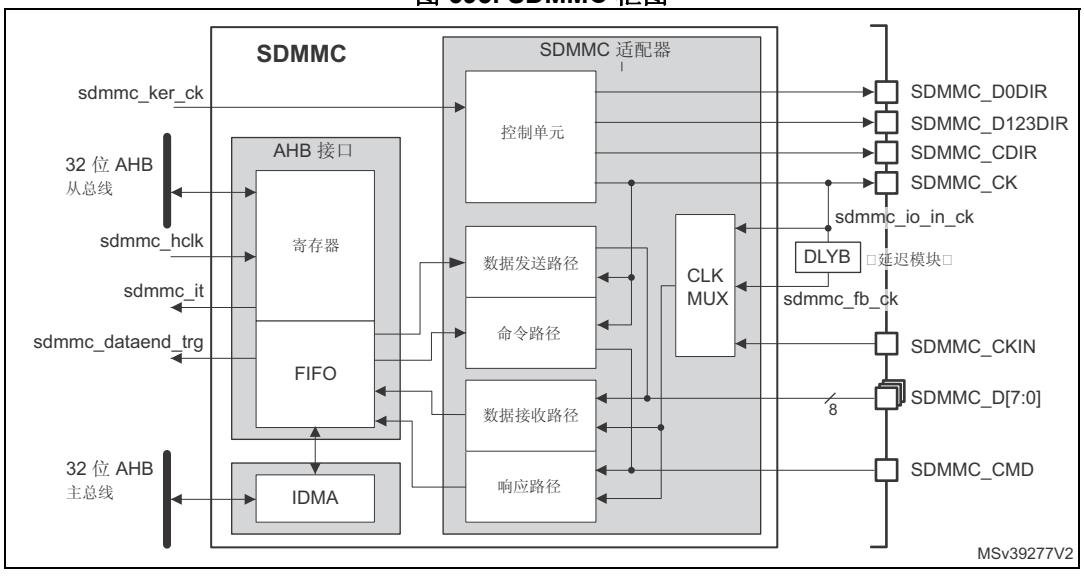
- AHB 从接口访问 SDMMC 适配器寄存器，并且生成中断信号和 IDMA 控制信号。
- SDMMC 适配器块提供特定于 MMC/SD/SD I/O 卡的所有功能，如时钟生成单元、命令和数据传输。
- 内部 DMA（IDMA）模块及其 AHB 主接口。

器件的 DLYB 延迟模块可用于对齐 SDMMC 接收到的数据的采样时钟。若要使 SDMMC 支持 SDR104 模式，此延迟模块是必须的。

55.4.1 SDMMC 图

图 695 显示了 SDMMC 框图。

图 695. SDMMC 框图



55.4.2 SDMMC 引脚和内部信号

表 426 列出了 SDMMC 内部输入/输出信号，*表 427* 列出了 SDMMC 引脚（复用功能）。

表 426. SDMMC 内部输入/输出信号

信号名称	信号类型	说明
sdmmc_ker_ck	数字输入	SDMMC 内核时钟
sdmmc_hclk	数字输入	AHB 时钟
sdmmc_it	数字输出	SDMMC 全局中断
sdmmc_dataend_trg	数字输出	MDMA 的 SDMMC 数据结束触发信号
sdmmc_io_in_ck	数字输入	SD/SDIO/MMC 卡反馈时钟。此信号在内部连接到 SDMMC_CK 引脚（针对 DS 和 HS 模式）。
sdmmc_fb_ck	数字输入	DLYB 延迟模块后的 SD/SDIO/MMC 卡调节反馈时钟（用于 SDR50、DDR50、SDR104）

表 427. SDMMC 引脚

信号名称	信号类型	说明
SDMMC_CK	数字输出	SD/SDIO/MMC 卡的时钟
SDMMC_CKIN	数字输入	来自 SD/SDIO/MMC 卡的外部驱动器的时钟反馈。（用于 SDR12、SDR25、SDR50 和 DDR50）
SDMMC_CMD	数字输入/输出	SD/SDIO/MMC 卡双向命令/响应信号。
SDMMC_CDIR	数字输出	SDMMC_CMD 信号的 SD/SDIO/MMC 卡 I/O 方向指示。
SDMMC_D[7:0]	数字输入/输出	SD/SDIO/MMC 卡双向数据线。
SDMMC_D0DIR	数字输出	SDMMC_D0 数据线的 SD/SDIO/MMC 卡 I/O 方向指示。
SDMMC_D123DIR	数字输出	SDMMC_D[3:1] 数据线的 SD/SDIO/MMC 卡 I/O 方向指示。

55.4.3 概述

SDMMC_D[7:0] 线具有不同的工作模式：

- 默认情况下，SDMMC_D0 线用于数据传输。初始化后，主机可以更改数据总线宽度。
- 对于 MMC，可以使用 1 位 (SDMMC_D0)、4 位 (SDMMC_D[3:0]) 或 8 位 (SDMMC_D[7:0]) 的数据总线宽度。
- 对于 SD 或 SDIO 卡，可以使用 1 位 (SDMMC_D0) 或 4 位 (SDMMC_D[3:0]) 的数据总线宽度。所有数据线均以推挽模式运行。

为了连接外部驱动器（电压切换收发器），使用 I/O 方向信号指示数据线上的数据流方向。**SDMMC_D0DIR** 信号指示 SDMMC_D0 数据线的 I/O 方向，**SDMMC_D123DIR** 则指示 SDMMC_D[3:1] 数据线的方向。

SDMMC_CMD 仅在推挽模式下运行：

为了连接外部驱动器（电压切换收发器），使用 I/O 方向信号 **SDMMC_CDIR** 指示 SDMMC_CMD 线上的数据流方向。

卡的 **SDMMC_CK** 时钟源自 **sdmmc_ker_ck**：

- 当 sdmmc_ker_ck 时钟的占空比为 50% 时，即使在旁路模式 (CLKDIV = 0) 下也可以使用。
- 当 sdmmc_ker_ck 占空比不是 50% 时，必须使用 CLKDIV 以 2 或更大值 (CLKDIV > 0) 对其进行分频。
- SDMMC_CMD/SDMMC_D[7:0] 输出和 SDMMC_CK 之间的相位关系可通过 NEGEDGE 位进行选择。相位关系取决于 CLKDIV、NEGEDGE 和 DDR 设置。请参见*图 696*。

图 696. SDMMC 命令和数据相位关系

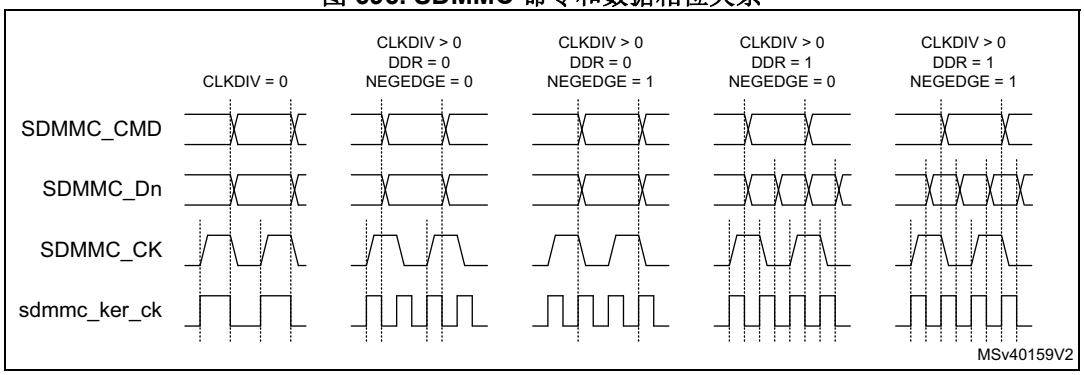


表 428. SDMMC 命令和数据相位选择

CLKDIV	DDR	NEGEDGE	SDMMC_CK	命令输出	数据输出
0	x	x	= sdmmc_ker_ck	在 sdmmc_ker_ck 下降沿生成	
>0	0	0	在 sdmmc_ker_ck 上升沿生成	在 SDMMC_CK 上升沿后的 sdmmc_ker_ck 下降沿生成。	
		1		在生成 SDMMC_CK 下降沿的同一 sdmmc_ker_ck 上升沿生成。	
	1	0		在 SDMMC_CK 上升沿后的 sdmmc_ker_ck 下降沿生成。	在 SDMMC_CK 边沿后的 sdmmc_ker_ck 下降沿生成。
		1		在生成 SDMMC_CK 下降沿的同一 sdmmc_ker_ck 上升沿生成。	

默认情况下，选择 **sdmmc_io_in_ck** 反馈时钟输入来采样 SDMMC 接收路径中的传入数据。它源自 SDMMC_CK 引脚。

为调整采样时钟的相位以适应接收数据时序，可将器件上的 DLYB 延迟模块连接在 **sdmmc_io_in_ck** 信号（DLYB 输入 dlyb_in_ck）和 SDMMC 的 **sdmmc_fb_ck** 时钟输入（DLYB 输出 dlyb_out_ck）之间。然后选择将 **sdmmc_fb_ck** 时钟输入用于接收路径即可对传入的数据使用相位经过调整的采样时钟。若要使 SDMMC 支持 SDR104 工作模式以及可选择支持 SDR50 和 DDR50 模式，这是必须项。

当使用外部驱动器（电压切换收发器）时，可以选择 SDMMC_CKIN 反馈时钟输入来对接收数据进行采样。

对于 SD/SDIO/MMC 卡，时钟频率可在 0 到 208 MHz 之间变化（受限于最大 I/O 速度）。

根据选择的总线模式（SDR 或 DDR），每个时钟周期在 SDMMC_D[7:0] 线上传输一位或两位。SDMMC_CMD 线每个时钟周期只传输一位。

55.4.4 SDMMC 适配器

SDMMC 适配器（参见*图 695：SDMMC 框图*）是一个多媒体卡/安全数字存储卡总线主设备，提供与多媒体卡堆栈或安全数字存储卡的接口。它由以下子单元组成：

- 控制单元
- 数据发送路径
- 命令路径
- 数据接收路径
- 响应路径
- 接收数据路径时钟复用器。
- 适配器寄存器模块
- 数据 FIFO
- 内部 DMA (IDMA)

注： *适配器寄存器和 FIFO 使用 AHB 时钟域 (sdmmc_hclk)。控制单元、命令路径和数据发送路径使用 SDMMC 适配器时钟域 (sdmmc_ker_ck)。响应路径和数据接收路径使用来自 sdmmc_io_in_ck、SDMMC_CKIN 或 sdmmc_fb_ck（由 DLYB 生成）的 SDMMC 适配器反馈时钟域。*

器件上的 DLYB 延迟模块可与 SDMMC 适配器搭配用来调整 SDMMC 接收模式下传入数据的采样时钟相位。若要使 SDMMC 支持 SDR104 工作模式以及可选择支持 SDR50 和 DDR50 模式，这是必须项。

适配器寄存器模块

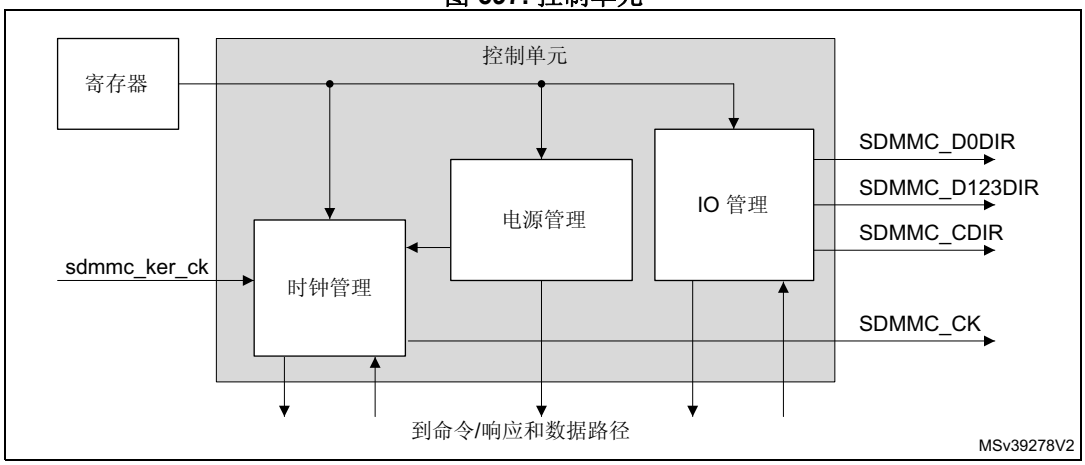
适配器寄存器模块包含所有系统控制寄存器、SDMMC 命令和响应寄存器以及数据 FIFO。

该模块还从 SDMMC 清零寄存器中的相应位单元生成信号，该寄存器用于清零 SDMMC 适配器中的静态标志。

控制单元

控制单元如*图 697* 所示，其包含电源管理功能、带有分频器的 SDMMC_CK 时钟管理功能和 I/O 方向管理功能。

图 697. 控制单元



电源管理子单元会在断电阶段和上电阶段中禁止卡总线输出信号。

有三个电源阶段：

- 掉电
- 上电
- 通电

时钟管理子单元使用 sdmmc_ker_ck 来生成 SDMMC_CK，并提供分频控制。它还负责停止 SDMMC_CK 以进行流控制等。

时钟输出在以下情况下无效：

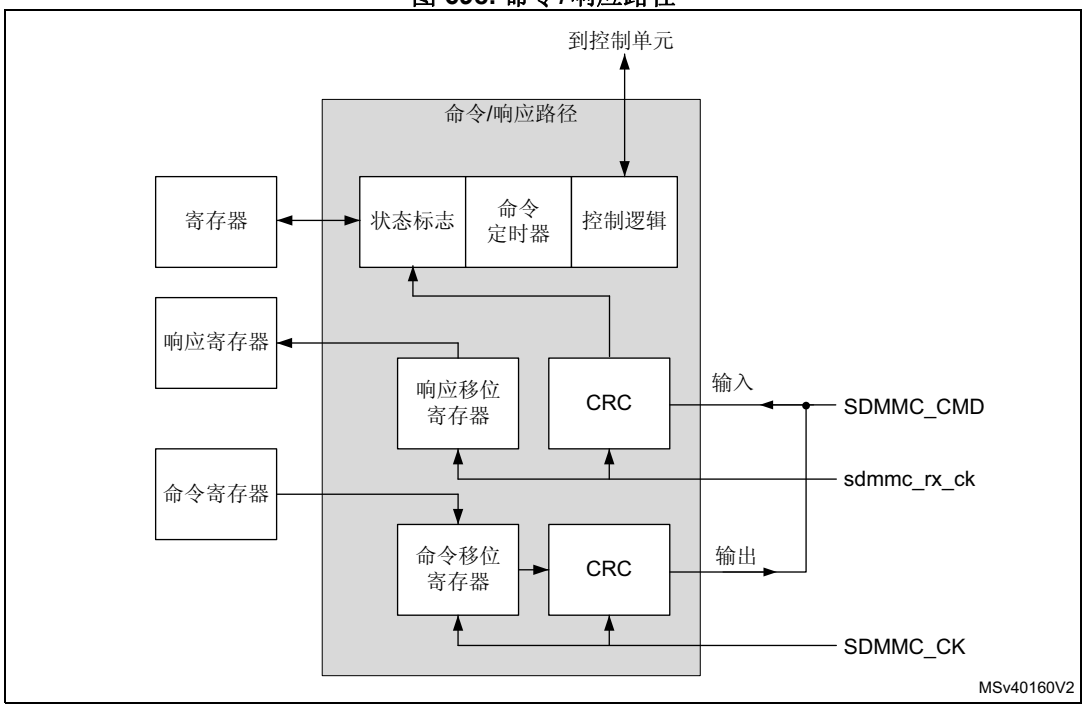
- 复位后
- 在断电或上电阶段中
- 节能模式（寄存器位 PWRSAV）已使能并且卡总线处于空闲状态的时间持续 8 个时钟周期。在命令/响应 CPSM 和数据路径 DPSM 子单元进入空闲阶段后的八个周期后，时钟停止。当命令/响应 CPSM 或数据路径 DPSM 激活（使能）时，时钟将重新启动。

I/O 管理子单元负责处理用于控制外部电压收发器的 SDMMC_Dn 和 SDMMC_CMD I/O 方向信号。

命令/响应路径

命令/响应路径子单元在 SDMMC_CMD 线上传送命令和响应。命令路径由 SDMMC_CK 提供时钟，并向卡发送命令。响应路径由 sdmmc_rx_ck 提供时钟，并从卡接收响应。

图 698. 命令/响应路径

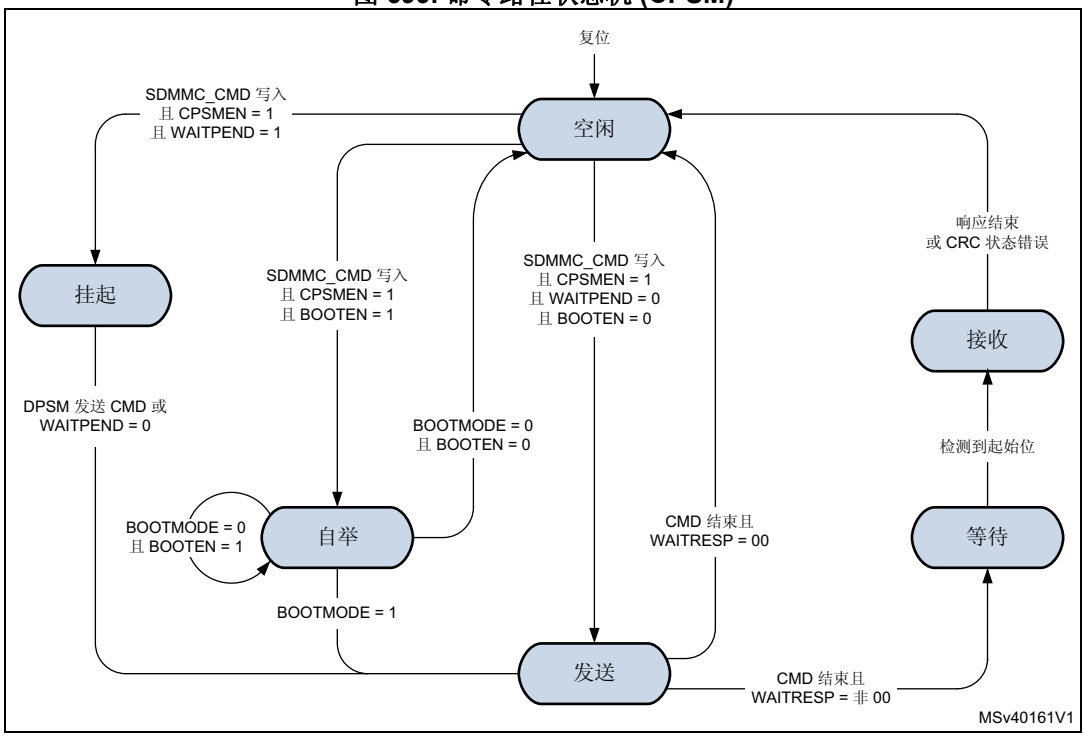


命令/响应路径状态机 (CPSM)

- 写入命令寄存器并且将使能位置 1 后，命令传输将开始。命令发送完成后，将附加 CRC，命令路径状态机 (CPSM) 将设置状态标志，并且：
  - 如果不需要响应，则进入空闲状态。
  - 如果需要响应，则等待响应。
- 接收到响应时，
  - 对于带有 CRC 的响应，将比较接收到的 CRC 代码和内部生成代码，并且会根据结果将相应的状态标志置 1。
  - 对于没有 CRC 的响应，不会进行 CRC 校验，并且不会将相应的状态标志置 1。

当 CPSM 有效（即不处于空闲状态）时，CPSMACT 位置 1。

图 699. 命令路径状态机 (CPSM)



- **空闲**：命令路径无效。当写入命令控制寄存器并且使能位 (CPSMEN) 置 1 后，CPSM 将激活 SDMMC_CK 时钟（因节能 PWRSAV 位而停止）并且
  - 在 WAITPEND = 0 且 BOOTEN = 0 时变为发送状态。
  - 在 WAITPEND = 1 时变为挂起状态。
  - 在 BOOTEN = 1 时变为启动状态。
- **发送**：发送命令并附加 CRC。
  - 当 CMDTRANS 位置 1 或 BOOTEN 位置 1、BOOTMODE 为备用启动状态且 DTDIR = 接收时，CPSM 数据使能信号将在命令结束时发送给 DPSM。
  - 当 CMDTRANS 位置 1 且 CMDSUSPEND 位为 0 时，中断周期将在命令结束时终止。
  - 当 CMDSTOP 位置 1 时，CPSM 中止信号将在命令结束时发送给 DPSM。
  - 如果不需要响应 (WAITRESP = 00)，CPSM 将变为空闲状态并生成 CMDSENT 标志。当 BOOTMODE = 1 且 BOOTEN = 0 时，CMDSENT 标志将在命令结束位后延迟 56 个周期生成，否则 CMDSENT 标志将在命令结束位后立即生成。
    不会修改 RESPCMDR 和 RESPxR 寄存器。
  - 如果需要命令响应（WAITRESP 不为 00），CPSM 将变为等待状态并启动响应超时。

- **等待**：命令路径等待响应。
  - 当 WAITINT 位为 0 时，命令定时器开始运行，CPSM 等待起始位。
  a) 如果在超时之前检测到起始位，CPSM 将变为接收状态。
  b) 如果在 CPSM 检测到响应起始位之前便已超时，则超时标志 (CTIMEOUT) 将置 1，CPSM 将变为空闲状态。
  不会修改 RESPCMDR 和 RESPxR 寄存器。
  - 当 WAITINT 位为 1 时，定时器将禁止并且 CPSM 将等待来自其中一个卡的中断请求（响应起始位）。
  a) 如果检测到起始位，CPSM 将变为接收状态。
  b) 向 WAITINT 写入 0（中断模式中止）时，主机将自行发送响应，并且检测到起始位时，CPSM 将变为接收状态。
- **接收**：将收到命令响应。响应可短可长，可带 CRC 可不带 CRC，具体取决于命令控制寄存器中的响应模式位 WAITRESP。将根据内部生成的 CRC 代码对接收到的 CRC 代码（存在时）进行校验。
  - 当 CMDSUSPEND 位置 1 且 SDIO 响应位 BS = 0（响应位 [39]）时，中断周期将在响应后启动。
  当 CMDSUSPEND 位清零或 CMDSUSPEND 位为 1 且 SDIO 响应位 BS = 1（响应位 [39]）时，不会有中断周期启动。
  - 当 CMDTRANS 位置 1、CMDSUSPEND 位置 1 且 SDIO 响应位 DF= 1（响应位 [32]）时，中断周期将在响应后终止。
  - 当 CRC 状态通过校验或不存在 CRC 时，CMDREND 标志将置 1，CPSM 变为空闲状态。
  RESPCMDR 和 RESPxR 寄存器通过接收到的响应进行更新。
  - 当 BOOTMODE = 1 且 BOOTEN = 0 时，CMDREND 标志将在响应结束位后延迟 56 个周期生成，否则 CMDREND 标志将在响应结束位后立即生成。
  - 当 CMDTRANS 位置 1 且 DTDIR = 发送时，CPSM 数据使能信号将在命令响应结束时发送给 DPSM。
  - 当 CRC 状态未通过校验时，CCRCFAIL 标志将置 1，CPSM 变为空闲状态。
  RESPCMDR 和 RESPxR 寄存器通过接收到的响应进行更新。
- **挂起**：CPSM 根据命令寄存器中的挂起 WAITPEND 位进入挂起状态。
  - 当 DATALENGTH =< 5 个字节时，CPSM 变为发送状态，并生成数据使能信号以启动与 CMD12 停止传输命令对齐的数据传输。
  - 当 DATALENGTH > 5 个字节时，将向 DPSM 发出 CPSM 数据使能信号以启动数据传输。CPSM 会等待来自 DPSM 的 sendCMD 信号，然后再变为发送状态。这样一来，便能够发送与数据对齐的 CMD12 停止传输命令。
  - 向 WAITPEND 写入 0 时，CPSM 将变为发送状态。
- **启动**：如果命令寄存器中的 BOOTEN 位置 1，则 CPSM 将进入启动状态，并且：
  - 当 BOOTMODE = 0 时，SDMMC_CMD 线驱动为低电平；当 CMDTRANS 位置 1 且 DTDIR = 接收时，CPSM 数据使能信号将发送到 DPSM，从而进入正常启动操作。启动过程结束后，将寄存器位 BOOTEN 清零以退出启动状态，此时会将 SDMMC_CMD 线驱动为高电平并将 CPSM 中止信号发送到 DPSM，然后再进入空闲状态。CMDSENT 标志在 SDMMC_CMD 线变为高电平后的 56 个周期后生成。
  - 当 BOOTMODE = 1 时，变为发送状态。此时可以发送 CMD0（启动）。将 BOOTEN 清零不起作用。

注： *CPSM 保持空闲状态至少八个 SDMMC_CK 周期以满足 N_{CC} 和 N_{RC} 时序限制。N_{CC} 是两个主机命令之间的最小延迟，N_{RC} 是主机命令和卡响应之间的最小延迟。*

注： *响应超时为固定值：64 个 SDMMC_CK 时钟周期。*

命令是用于启动操作的令牌。命令从主机发送到单个卡（编址命令），或发送到所有已连接的卡（MMC V3.31 或更低版本可以使用广播命令）。命令在 SDMMC_CMD 线上以串行方式传输。所有命令都为固定长度 48 位。*表 429* 中显示了 SD 存储卡、SDIO 卡和 MMC 卡的命令令牌的常规格式。

命令令牌数据取自 2 个寄存器，一个包含 32 位参数，另一个包含 6 位命令索引（发送到卡的 6 位）。

表 429. 命令令牌格式

位的位置	宽度	值	说明
47	1	0	起始位
46	1	1	传输位
[45:40]	6	x	命令索引
[39:8]	32	x	参数
[7:1]	7	x	CRC7
0	1	1	结束位

命令数据旁为命令类型 (WAITRESP) 位，用于控制命令路径状态机 (CPSM)。这些位还用于确定命令是否需要响应，响应是短（48 位）还是长（136 位）以及是否存在 CRC。

响应是一个令牌，它作为对先前接收命令的应答，从编址卡或从所有已连接的卡同步发送到主机。所有响应都是通过命令线 SDMMC_CMD 来发送。响应传输始终从对应于响应代码字的位字符串的左侧位开始。代码长度取决于响应类型。响应令牌 R1、R2、R3、R4、R5 和 R6 具有多种编码方案，具体取决于其内容。*表 430*、*表 431* 和 *表 432* 分别给出了 SD 存储卡、SDIO 卡和 MMC 卡的响应令牌的常规格式。

响应始终从起始位（始终为 0）开始，后跟用于指示传输方向的位 (card = 0)。下表中标有 x 的值表示变量项。大多数响应均受 CRC 保护。每个命令代码字由结束位（始终为 1）来终止。

响应令牌数据存储在 5 个寄存器中，其中 4 个寄存器包含 32 位卡状态、OCR 寄存器、参数或包括内部 CRC 的 127 位 CID 或 CSD 寄存器，另一个寄存器包含 6 位命令索引。

表 430. 带 CRC 的短响应令牌格式

位的位置	宽度	值	说明
47	1	0	起始位
46	1	0	传输位
[45:40]	6	x	命令索引（或保留为 111111）
[39:8]	32	x	参数
[7:1]	7	x	CRC7
0	1	1	结束位

表 431. 不带 CRC 的短响应令牌格式

位的位置	宽度	值	说明
47	1	0	起始位
46	1	0	传输位
[45:40]	6	x	命令索引（或保留为 111111）
[39:8]	32	x	参数
[7:1]	7	1111111	（保留为 1111111）
0	1	1	结束位

表 432. 带 CRC 的长响应令牌格式

位的位置	宽度	值	说明
135	1	0	起始位
134	1	0	传输位
[133:128]	6	111111	保留
[127:1]	127:8	x	CID 或 CSD 片段
	7:1	x	CRC7（包含在 CID 或 CSD 中）
0	1	1	结束位

命令/响应路径以半双工模式运行，因此既可发送命令，也可接收响应。如果 CPSM 未处于发送状态，则 SDMMC_CMD 输出处于高阻抗状态。在 SDMMC_CMD 上发送的数据将根据 NEGEDGE 寄存器位与 SDMMC_CK 进行同步，请参见*图 696*。

对于命令和带 CRC 的短响应，CRC 生成器计算 CRC 代码前面全部 40 个位的 CRC 校验和。这包括起始位、传输位、命令索引和命令参数（或卡状态）。

对于长响应，仅计算 R2 CID 或 CSD 的 120 个位的 CRC 校验和。请注意，CRC 计算中不使用起始位、发送位和 6 个保留位。

CRC 校验和是一个 7 位值：

CRC[6:0] = 余数 [(M(x) * x^7) / G(x)]

G(x) = $x^7 + x^3 + 1$

M(x) = (第 1 个位) * x^n + (第 2 个位) * x^{n-1}+... + (CRC 前的最后一位) * x^0

其中，n = 39 或 119。

CPSM 可在 CPSMEN 置 1 时发送一些特定命令来处理各种工作模式，请参见*表 433*。

表 433. 特定命令概述

VSWITCH	BOOTEN	BOOTMODE	CMDTRANS	WAITPEND	CMDSTOP	WAITINT	说明
1	x	x	x	x	x	x	启动电压切换序列
0	1	x	x	x	x	x	开始正常启动
0	1	1	x	x	x	x	开始备用启动
0	0	1	x	x	x	x	停止备用启动
0	0	0	1	x	x	x	发送包含相关数据传输的命令。
0	0	0	0	1	1	x	MMC 流数据传输，命令 (STOP_TRANSMISSION) 挂起至数据传输结束。
0	0	0	0	1	0	x	MMC 流数据传输，(STOP_TRANSMISSION) 以外的命令挂起至数据传输结束。
0	0	0	0	0	1	x	发送命令 (STOP_TRANSMISSION)，停止任何正在进行的数据传输。
0	0	0	0	0	0	1	进入 MMC 等待中断 (Wait-IRQ) 模式。
0	0	0	0	0	0	0	任何其他非特定命令

命令/响应路径实现了*表 434* 所示的状态标志和相关清零位：

表 434. 命令路径状态标志

标志	说明
CMDSENT	在命令（无响应）结束时置 1。（CPSM 从发送状态变为空闲状态）
CMDREND	当 CRC 正常时，在命令响应结束时置 1。（CPSM 从接收状态变为空闲状态）
CCRCFAIL	当 CRC 失败时，在命令响应结束时置 1。（CPSM 从接收状态变为空闲状态）
CTIMEOUT	如果未在超时前接收到响应起始位，则在命令后置 1。（CPSM 从等待状态变为空闲状态）
CKSTOP	当 CRC 正常并且 SDMMC_CK 停止时，在电压切换 (VSWITCHEN = 1) 命令响应后置 1。（对 CPSM 无影响）
VSWEND	在 5 ms + 1 ms 的电压切换 (VSWITCH = 1) 超时后置 1。（对 CPSM 无影响）
CPSMACT	正在进行命令传输。（CPSM 不处于空闲状态）

命令路径错误处理如*表 435* 所示：

表 435. 命令路径错误处理

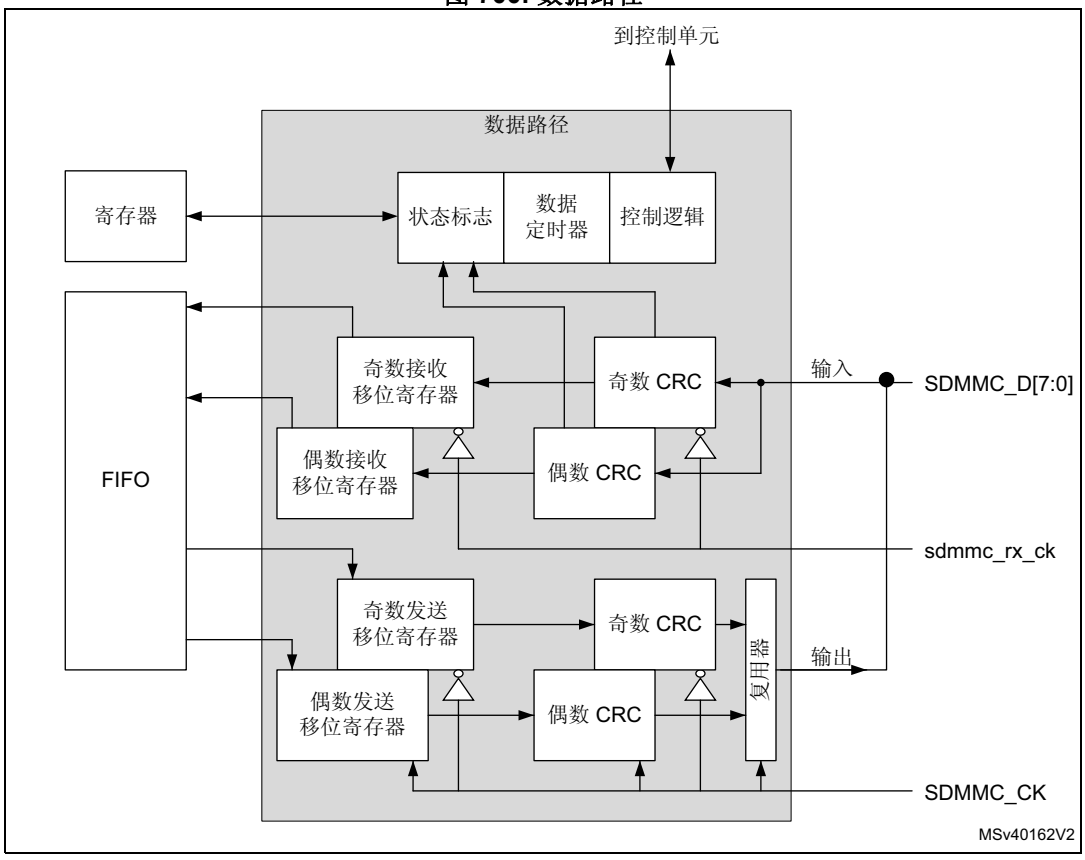
误差	CPSM 状态	原因	卡操作	主机操作	CPSM 操作
超时	等待	未及时出现起始位	未知	对卡进行复位或掉电再上电(1)	变为空闲状态
CRC状态	接收	否定状态	忽略命令	重新发送命令(1)	变为空闲状态
		传输错误	接受命令	重新发送命令(1)	

1. 当 CMDTRANS 置 1 时，还需要发送 stop_transmission 命令才能使 DPSM 变为空闲状态。

数据路径

数据路径子单元负责通过 SDMMC_D[7:0] 线与卡相互传输数据。数据传输路径由 SDMMC_CK 提供时钟，并向卡发送数据。数据接收路径由 sdmmc_rx_ck 提供时钟，并从卡接收数据。*图 700* 显示了数据路径框图。

图 700. 数据路径



卡数据总线宽度可以在时钟控制寄存器位 WIDBUS 中进行编程。支持的数据总线宽度模式如下：

- 如果未使能宽总线模式，则仅通过 SDMMC_D0 传输一位。
- 如果使能了 4 位宽度的总线模式，则通过 SDMMC_D[3:0] 传输 4 个数据位。
- 如果使能了 8 位宽度的总线模式，则通过 SDMMC_D[7:0] 传输 8 个数据位。

与数据总线宽度相似，数据采样模式可以在时钟控制寄存器位 DDR 中进行编程。支持的数据采样模式如下：

- 单倍数据速率信号传输 (SDR)，在时钟的上升沿驱动数据。
- 双倍数据速率信号传输 (DDR)，在时钟的两个边沿驱动数据。只有宽总线模式（4 位宽和 8 位宽）才支持 DDR 模式。

注： *数据采样模式仅适用于 SDMMC_D[7:0] 线。（不适用于 SDMMC_CMD 线。）*

在 DDR 模式下，根据以下规则在 SDMMC_CK 的两个边沿对数据进行采样，另请参见*图 701*和*图 702*：

- 在时钟的上升沿，对奇数字节进行采样。
- 在时钟的下降沿，对偶数字节进行采样。
- 数据有效负载大小始终为 2 字节的倍数。
- 每个数据线计算两个 CRC16。
  - 奇数位 CRC16，在时钟的下降沿驱动。
  - 偶数位 CRC16，在时钟的上升沿驱动。
- 起始位、结束位和空闲条件均为完整周期。
- CRC 状态/启动确认和繁忙信号均为完整周期，仅在时钟的上升沿进行采样。

在 DDR 模式下，SDMMC_CK 时钟分频比应 >= 2。

图 701. DDR 模式数据包时钟驱动

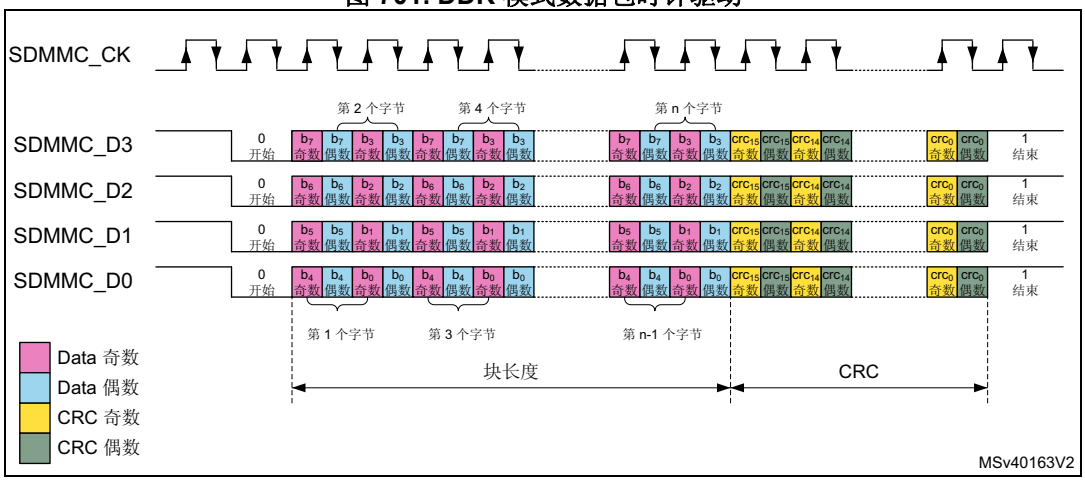
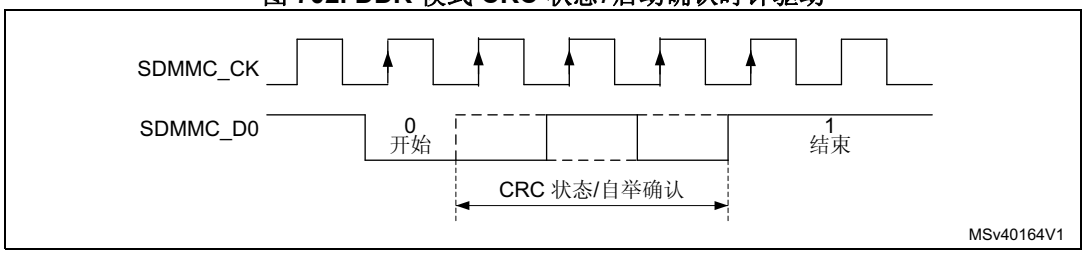


图 702. DDR 模式 CRC 状态/启动确认时钟驱动



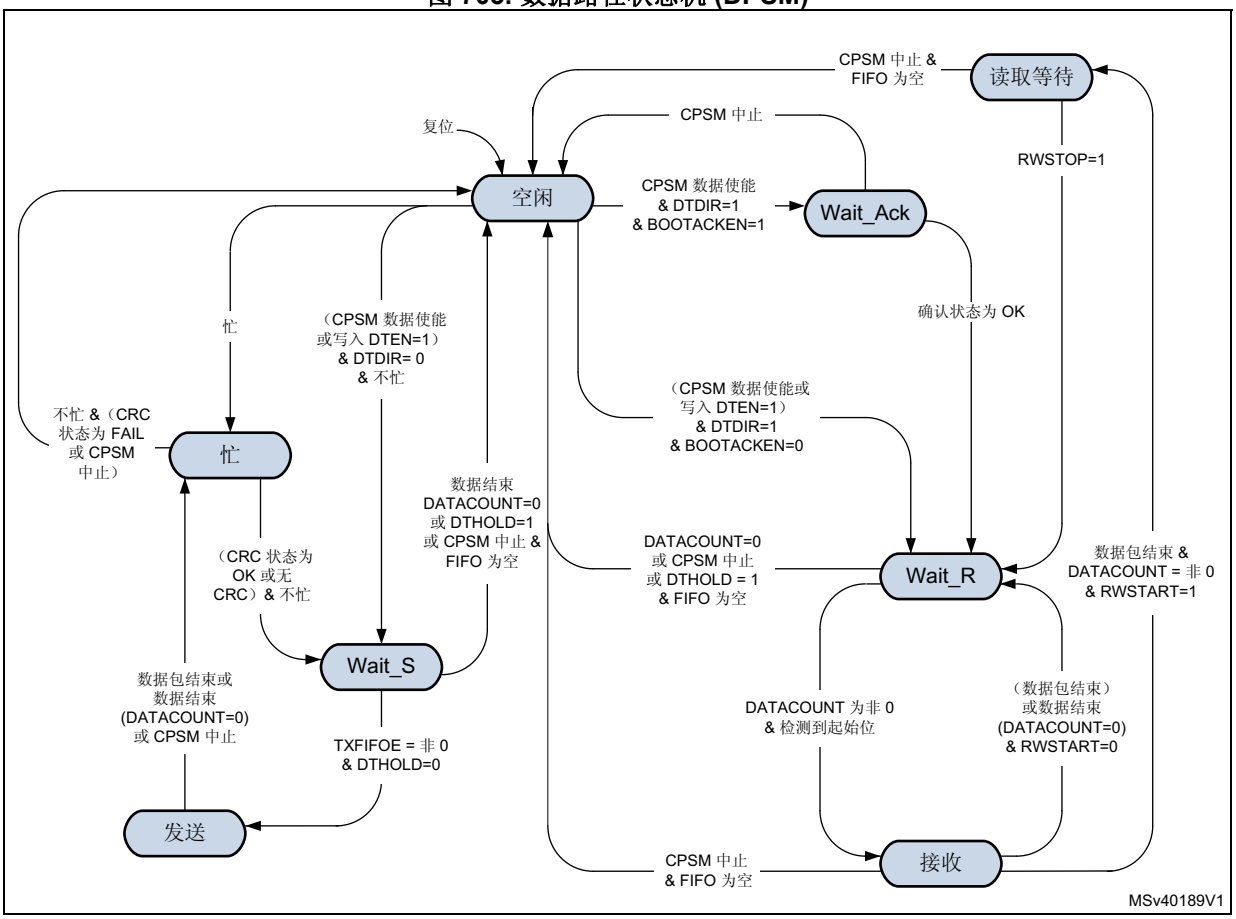
数据路径状态机 (DPSM)

根据传输方向（发送或接收），数据路径状态机 (DPSM) 将变为 Wait_S 或 Wait_R 状态（如果已使能）：

- 发送：DPSM 变为 Wait_S 状态。如果传输 FIFO 中有数据，则 DPSM 变为发送状态并且数据路径子单元开始向卡发送数据。
- 接收：DPSM 变为 Wait_R 状态并等待起始位。在 DPSM 收到起始位时，将变为接收状态，并且数据路径子单元开始从卡中接收数据。
  对于包含确认的启动操作，DPSM 变为 Wait_Ack 状态并等待启动确认，然后再变为 Wait_R 状态。

DPSM 以 SDMMC_CK 工作。DPSM 具有以下状态，如*图 703* 所示。当 DPSM 有效（即不处于空闲状态）时，DPSMACT 位置 1。

图 703. 数据路径状态机 (DPSM)



- **空闲**状态：数据路径无效，SDMMC_D[7:0] 输出取决于 PWRCTRL 设置。DPSM 将通过以下方式激活：发送将 CMDTRANS 位置 1 的命令或将 DTEN 位置 1，或者检测 SDMMC_D0 上的繁忙状态（即，在包含 R1b 响应的命令之后）。

  当不忙时，DPSM 将激活 SDMMC_CK 时钟（因节能 PWRSAV 位而停止时），并使用新的 (DATALENGTH) 值加载数据计数器，并且：

  – 当数据方向位 (DTDIR) 指示发送时，变为 Wait_S。
  – 当数据方向位 (DTDIR) 指示接收时，
    - 如果 BOOTACKEN 寄存器位清零，则变为 Wait_R。
    - 如果 BOOTACKEN 寄存器位置 1 且启动确认超时，则变为 Wait_Ack。

  当繁忙时，DPSM 将使 SDMMC_CK 时钟保持有效并变为繁忙状态。

注： *DTEN 不能用于启动与 SD、SDIO 和 eMMC 卡之间的数据传输。*

- **Wait_Ack** 状态：数据路径等待启动确认令牌。
  – 如果 DPSM 在超时之前收到无错确认，则它将变为 Wait_R 状态。
  – 当接收到确认以外的模式时，将生成确认状态错误，并且确认失败状态标志 (ACKFAIL) 将置 1。DPSM 保持 Wait_Ack 状态。
  – 如果在检测到起始位前便已超时 (ACKTIME)，则它会将超时状态标志 (ACKTIMEOUT) 置 1。DPSM 保持 Wait_Ack 状态。
  – 当 CPSM 中止信号置 1 时，它将变为空闲状态并将 DABORT 标志置 1。
- **Wait_R** 状态：如果数据计数器不为零并且数据未保存，则数据路径将等待 SDMMC_D[n:0] 上的起始位。如果数据计数器为零或数据已保存，则数据路径将等待 FIFO 变为空。
  – 在块模式下，如果在超时之前接收到起始位，DPSM 将变为接收状态，并使用 DBLOCKSIZE 加载数据块计数器。
  – 在 SDIO 多字节模式下，如果在超时之前接收到起始位，DPSM 将变为接收状态，并使用 DATALENGTH 加载数据块计数器。
  – 在流模式下，如果在超时之前接收到起始位，DPSM 将变为接收状态，并使用 DATALENGTH 加载数据计数器。
  – 如果数据计数器 (DATACOUNT) 等于零（数据结束），当接收 FIFO 为空并且 DATAEND 标志置 1 时，DPSM 将变为空闲状态。
  – 如果在检测到起始位前便已超时 (DATATIME)，则它会将超时状态标志 (DTIMEOUT) 置 1 且 DPSM 会保持 WAIT_R 状态。
  – 如果 CPSM 中止信号置 1：
    如果 DATACOUNT > 0，则当 FIFO 为空时，DPSM 变为空闲状态，当 IDMAEN = 0 时，使用 FIFORST 复位，并将 DABORT 标志置 1。
    如果 DATACOUNT 为零，则继续正常操作，由于传输已经正常完成，将不会有 DABORT 标志。
  – 如果 DTHOLD 位置 1：
    - 如果 DATACOUNT > 0，则当接收 FIFO 为空时，DPSM 变为空闲状态，当 IDMAEN = 0 时，使用 FIFORST 复位，并发出 DHOLD 标志。保持期间禁止超时。如果在保持期间接收到 CPSM 中止信号，则传输中止。
    - 当 DATACOUNT = 0 时，传输正常完成，不会有 DHOLD 标志。
  – 如果 DPSM 通过 DTEN 启动，在发生错误 (DTIMEOUT) 之后，当 FIFO 为空时，DPSM 将变为空闲状态，当 IDMAEN = 0 时，将使用 FIFORST 复位。

- **读取等待**状态：数据路径对总线进行读取等待。
  - 当读取等待停止位 (RWSTOP) 置 1 时，DPSM 将变为 Wait_R 状态，并且启动接收超时。
  - 如果 CPSM 中止信号置 1，则等待 FIFO 变为空，并且当 IDMAEN = 0 时，使用 FIFORST 复位，然后变为空闲状态并将 DABORT 标志置 1。
- **接收**状态：数据路径从卡接收串行数据。将数据以字节为单位打包，并写入到数据 FIFO。根据数据控制寄存器中选择的传输模式 (DTMODE)，数据传输模式可以是块或流：
  - 在块模式中，当接收到的数据字节数达到数据块大小 (DBLOCKSIZE) 时，DPSM 将等待直至其收到 CRC 代码。
  - 在 SDIO 多字节模式中，当接收到的数据字节数达到数据块大小 (DATALENGTH) 时，DPSM 将等待直至其收到 CRC 代码。

  a) 如果收到的 CRC 代码与内部生成的 CRC 代码相匹配，则
  - 当 RWSTART = 0 且启动接收超时时，DPSM 将变为 Wait_R 状态。
  - 当 RWSTART = 1 且 DATACOUNT > 0 时，DPSM 将变为读取等待状态，并会生成 DBCKEND 标志。

  b) 如果收到的 CRC 代码与内部生成的 CRC 代码不匹配，则将阻止任何后续的数据接收操作。
  - 当并未接收到所有数据 (DATACOUNT > 0) 时，CRC 失败状态标志 (DCRCFAIL) 将置 1，DPSM 将保持接收状态。
  - 当接收到所有数据 (DATACOUNT = 0) 时，将等待 FIFO 变为空，之后 CRC 失败状态标志 (DCRCFAIL) 将置 1，DPSM 将变为空闲状态。
  - 在流模式中，DPSM 在数据计数器 DATACOUNT > 0 时接收数据。当计数器为零时，会将移位寄存器中其余数据写入到数据 FIFO，并且 DPSM 将变为 Wait_R 状态。
  - 如果发生 FIFO 溢出错误，则 DPSM 将 FIFO 溢出错误标志 (RXOVERR) 置 1 并且会阻止任何后续的数据接收操作。DPSM 保持接收状态。
  - 收到 CPSM_Abort 信号时：
    - 如果 DATACOUNT = 0 时在数据的最后 2 位之前接收到 CPSM_Abort 信号，则传输中止。移位寄存器中的剩余数据将写入数据 FIFO，等待 FIFO 变为空，当 IDMAEN = 0 时，使用 FIFORST 复位，DPSM 随后变为空闲状态且 DABORT 标志置 1。
    - 如果 DATACOUNT = 0 时在传输的最后 2 位期间或之后接收到 CPSM_Abort 信号，则传输正常完成。DPSM 保持接收状态，不生成 DABORT 标志。
  - 如果 DPSM 通过 DTEN 启动，在发生错误（DATACOUNT > 0 时出现 DCRCFAIL，或者出现 RXOVERR）之后，当 FIFO 为空时，DPSM 将变为空闲状态，当 IDMAEN = 0 时，将使用 FIFORST 复位。
- **Wait_S** 状态：数据路径等待 FIFO 中有数据可用。
  - 如果数据计数器 DATACOUNT > 0，则等待直到数据 FIFO 为空标志 (TXFIFOE) 置为无效并且 DTHOLD 未置 1，然后变为发送状态。
  - 如果数据计数器 (DATACOUNT) = 0，DPSM 将变为空闲状态并生成 DATAEND 标志。
  - 当 DTHOLD 置 1 并且 DATACOUNT > 0 时
    - 如果使能 IDMA，FIFO 将被清空，之后 DPSM 将变为空闲状态并发出 DHOLD 标志。（DBCKEND 标志也将置 1，并且可以与 DHOLD 标志同时清零。）
    - 如果禁止 IDMA，则会生成 DBCKEND 标志。等待 FIFO 使用 FIFORST 复位，之后 DPSM 将变为空闲状态并发出 DHOLD 标志。
  - 当 DTHOLD 置 1 且 DATACOUNT = 0 时，传输正常完成。

– 当收到 CPSM 中止信号时
- 如果 DATACOUNT = 0 时在数据的最后 2 位之前接收到 CPSM_Abort 信号，则传输将中止，等待 FIFO 变为空并且当 IDMAEN = 0 时，使用 FIFORST 复位，之后 DPSM 变为空闲状态并将 DABORT 标志置 1。
- 如果 DATACOUNT = 0 时在传输的最后 2 位期间或之后接收到 CPSM_Abort 信号，则继续正常操作，并且由于传输正常完成，将没有 DABORT 标志。

注： *DPSM 在 Wait_S 状态下需要至少保持两个时钟周期以满足 N_{WR} 计时要求，N_{WR} 是从收到卡响应到开始从主机传输数据这一期间的时钟周期数。*

- **发送**状态：DPSM 开始向卡发送数据。根据数据控制寄存器中的传输模式，数据传输模式可以是块、SDIO 多字节或流：
  – 在块模式中，当发送的数据字节数达到数据块大小 (DBLOCKSIZE) 时，DPSM 将发送一个内部生成的 CRC 代码和结束位，然后变为繁忙状态并启动发送超时。
  – 在 SDIO 多字节模式下，当发送的数据字节数达到数据块大小 (DATALENGTH) 时，DPSM 将发送一个内部生成的 CRC 代码和结束位，然后变为繁忙状态并启动发送超时。
  – 在流模式中，DPSM 在数据计数器 DATACOUNT > 0 时向卡发送数据。当数据计数器达到 0 时，DPSM 变为繁忙状态并启动发送超时。
  在根据 DATACOUNT 发送最后一个流字节之前，DPSM 会触发 sendCMD 信号。CPSM 使用该信号来发送任何挂起的命令。（即 CMD12 停止传输命令）
  – 如果发生 FIFO 下溢错误，DPSM 会将 FIFO 下溢错误标志 (TXUNDERR) 置 1。DPSM 保持发送状态。
  – 当收到 CPSM 中止信号时
  - 如果 DATACOUNT = 0 时在传输的最后 2 位之前接收到 CPSM_Abort 信号，则传输中止。DPSM 将发送最后一个数据位，后跟一个结束位。FIFO 将被禁止/清空，并且 DPSM 将变为繁忙状态，等待至不忙状态后将 DABORT 标志置 1。
  - 如果 DATACOUNT = 0 时在传输的最后 2 位期间或之后接收到 CPSM_Abort 信号，则传输正常完成，将不会有 DABORT 标志。
- **繁忙**状态：DPSM 等待 CRC 状态令牌（需要时），并等待不忙信号：
  – 如果需要 CRC 状态令牌并且该令牌指示“无错误传输”或者不需要 CRC 时：
  - 如果 SDMMC_D0 未处于低电平（卡未处于繁忙状态），则它变为 Wait_S 状态。
  - 当卡处于繁忙状态时，SDMMC_D0 处于低电平，它将保持繁忙状态。
  – 如果需要 CRC 状态令牌并且该令牌指示“错误传输”。
  - 当并未发送所有数据 (DATACOUNT > 0) 时。DPSM 等待至不忙状态后将 CRC 失败状态标志 (DCRCFAIL) 置 1。FIFO 将被禁止/清空并且 DPSM 保持繁忙状态。
  - 当已发送所有数据 (DATACOUNT = 0) 时。DPSM 等待至不忙状态后将 CRC 失败状态标志 (DCRCFAIL) 置 1，并且 DPSM 将变为空闲状态。
  – 如果 DPSM 处于繁忙状态时发生 CRC 状态 (Ncrc) 超时，则它会将数据超时标志 (DTIMEOUT) 置 1 并保持繁忙状态。
  – 如果 DPSM 处于繁忙状态时发生繁忙超时，则它会将数据超时标志 (DTIMEOUT) 置 1 并保持繁忙状态。
  – 当在繁忙状态下接收到 CPSM 中止信号时：
  - 如果 DATACOUNT > 0 时在 CRC 响应的最后 2 位之前接收到 CPSM_Abort 信号，则数据传输中止。DPSM 等待至不忙状态且 FIFO 被禁止/清空后变为空闲状态，并将 DABORT 标志置 1。
  - 如果 DATACOUNT = 0 时在 CRC 响应的最后 2 位期间或之后接收到 CPSM_Abort 信号，或者当不需要 CRC、DATACOUNT = 0 且没有 DTIMEOUT 错误时，DPSM 将保持繁忙状态，并且由于传输可正常完成，将不会生成 DABORT 标志。
  - 如果在已出现 DTIMEOUT 错误时接收到 CPSM_Abort 信号，则 DPSM 将等待至不忙状态且 FIFO 禁止/清空后变为空闲状态，并将 DABORT 标志置 1。

  - 当由于在发送状态下中止而进入繁忙状态时，DPSM 将等待至不忙状态后变为空闲状态，并将 DABORT 标志置 1。
  - 如果 DPSM 通过 DTEN 启动，在发生错误（DATACOUNT > 0 时出现 DCRCFAIL，或者出现 DTIMEOUT）之后，当 FIFO 复位时，DPSM 将变为空闲状态。
  - 当 DPSM 因 SDMMC_D0 上繁忙而启动时，它将等待至不忙状态后将繁忙结束状态标志 (BUSYD0END) 置 1，并且 DPSM 将变为空闲状态。

当 DPSM 在数据块结束位、数据读取命令结束位或 R1b 响应后 2 个周期后进入 Wait_R 或繁忙状态时，将使能数据定时器 (DATATIME)，并会生成数据超时错误 (DTIMEOUT)：

- 发送数据时，发生超时
  - 当需要 CRC 状态并且在 8 个 SDMMC_CK 周期内未接收到起始位时，DTIMEOUT 标志将置 1。
  - 当繁忙状态的时长超过编程的超时周期时，DTIMEOUT 标志将置 1。
- 接收数据时，发生超时
  - 当仍然有数据要接收、DATACOUNT > 0 并且未在编程的超时周期前接收到起始位时，DTIMEOUT 标志将置 1。
- 在 R1b 响应后，发生超时
  - 当繁忙状态的时长超过编程的超时周期时，DTIMEOUT 标志将置 1。

DATATIME = 0 时，

- 接收期间，起始位应在数据块结束位或数据读取命令结束位后的 2 个周期后出现。
- 发送期间，繁忙状态在 CRC 令牌结束位或流数据结束位后超时 2 个周期。
- 在 R1b 响应之后，繁忙状态在响应结束位后超时 2 个周期。

可以将数据从卡传输到主机（发送），或从主机传输到卡（接收）。数据通过 SDMMC_Dn 数据线传输，并且存储在 FIFO 中。

表 436. 数据令牌格式

说明	起始位	数据(1)	CRC16	结束位	DTMODE
块数据	0	（DBLOCKSIZE 和 DATALENGTH）	是	1	00
SDIO 多字节	0	(DATALENGTH)	是	1	01
MMC 数据流	0	(DATALENGTH)	否	1	10

1. 要传输的数据总量由 DATALENGTH 给出。对于块数据而言，每个块中的数据量由 DBLOCKSIZE 给出。

数据令牌格式通过相应的寄存器位 DTMODE 选择。

数据路径实现了*表 437* 所示的状态标志和相关清零位：

表 437. 数据路径状态标志和清零位

标志		说明
DATAEND	TX	如果 CRC 正常且繁忙状态已结束，则在完整数据传输结束时置 1。(DATACOUNT = 0)。（DPSM 从 WAIT_S 状态变为空闲状态）
	RX	如果 CRC 正常且所有数据读取完毕，则在完整数据传输结束时置 1（DATACOUNT = 0 且 FIFO 为空）。（DPSM 从 WAIT_R 状态变为空闲状态）
	BOOT	
DCRCFAIL	TX	如果 CRC 失败且繁忙状态已结束，则在 CRC 结束时置 1。（当仍然有数据要发送时，DPSM 保持繁忙状态并等待中止）（当所有数据发送完毕或 DPSM 已通过 DTEN 启动时，DPSM 将从繁忙状态变为空闲状态）
	RX	如果 CRC 失败且 FIFO 为空，则在 CRC 结束时置 1。（当仍然有数据要接收时，DPSM 保持接收状态并等待中止）（当所有数据接收完毕或 DPSM 已通过 DTEN 启动时，DPSM 将从接收状态变为空闲状态）
	BOOT	
ACKFAIL	BOOT	如果失败，则在 BOOT ACK 结束时置 1。（DPSM 保持 Wait_Ack 状态并等待中止）
DTIMEOUT	CMD R1b	如果未在超时前接收到繁忙结束信号，则在命令响应后置 1。（DPSM 保持繁忙状态并等待中止）
	TX	当未在 Ncrc 内接收到 CRC 令牌起始位或未在超时前接收到繁忙结束信号时置 1。（DPSM 保持繁忙状态并等待中止）（当 DPSM 已通过 DTEN 启动时，将变为空闲状态）注意：在繁忙超时之前，如果 CRC 失败，则还会将 DCRCFAIL 标志置 1。
	RX	如果未在超时前接收到起始位，则置 1。（DPSM 保持 WAIT_R 状态并等待中止）（当 DPSM 已通过 DTEN 启动时，将变为空闲状态）
	BOOT	
ACKTIMEOUT	BOOT	如果未在超时前接收到起始位，则置 1。（DPSM 保持 Wait_Ack 状态并等待中止）
DBCKEND	TX	当 DTHOLD = 1 时：在数据传输未完成的情况下 (DATACOUNT >0)，如果 CRC 正常且繁忙状态已结束，则在数据块传输结束时置 1。（DPSM 从 WAIT_S 状态变为空闲状态）
	RX	当 RWSTART = 1 时：在数据传输未完成的情况下 (DATACOUNT >0)，如果 CRC 正常，则在数据块传输结束时置 1。（DPSM 从接收状态变为读取等待状态）
	BOOT	
DHOLD	TX	当 DTHOLD = 1 时：在数据传输未完成的情况下 (DATACOUNT >0)，如果 CRC 正常且繁忙状态已结束，则在数据块传输结束时置 1。（DPSM 从 WAIT_S 状态变为空闲状态）
	RX	当 DTHOLD = 1 时：在数据传输未完成的情况下 (DATACOUNT >0)，如果 CRC 正常且所有数据读取完毕（FIFO 为空），则在数据块传输结束时置 1。（DPSM 从 WAIT_R 状态变为空闲状态）
DABORT	CMD R1b	当 CPSM 已发送中止事件且繁忙状态结束时。（DPSM 从繁忙状态变为空闲状态）
	TX	当 CPSM 在传输的最后 2 位之前已发送中止事件时。（DPSM 从任何状态变为空闲状态）
	RX	
	BOOT	
BUSYD0END	CMD R1b	如果繁忙状态在超时前结束，则在命令响应后置 1。（DPSM 从繁忙状态变为空闲状态）
DPSMACT		正在进行数据传输。（DPSM 不处于空闲状态）

数据路径错误处理如*表 438* 所示：

表 438. 数据路径错误处理

误差	DPSM 状态	原因	卡操作	主机操作	DPSM 操作
超时	Wait_Ack	未及时确认	未知	对卡进行掉电再上电	保持 Wait_Ack 状态（通过 RCC.SDMMCxRST 寄存器位复位 SDMMC）
	Wait_R	未及时出现起始位	未知	停止数据接收 发送停止传输命令	出现 CPSM_Abort 时，变为空闲状态
			未知	停止启动过程	
	繁忙	繁忙状态太久（由于数据传输）	未知	停止数据接收 发送停止传输命令	
		繁忙状态太久（由于 R1b）	未知	发送复位命令	
CRC	接收	传输错误	发送后续数据	停止数据接收 发送停止传输命令	出现 CPSM_Abort 时，变为空闲状态
CRC 状态	繁忙	否定状态	忽略后续数据	停止数据发送 发送停止传输命令	出现 CPSM_Abort 时，变为空闲状态
		传输错误	等待后续数据		
确认状态	Wait_Ack	传输错误	发送启动数据	停止启动过程	出现 CPSM_Abort 时，变为空闲状态
上溢	接收	FIFO 已满	发送后续数据	停止数据接收 发送停止传输命令	出现 CPSM_Abort 时，变为空闲状态
下溢	发送	FIFO 为空	接收后续数据	停止数据发送 发送停止传输命令	出现 CPSM_Abort 时，变为空闲状态

数据 FIFO

数据 FIFO（先进先出）子单元包含发送和接收数据缓冲器。单个 FIFO 用于由 DTDIR 位选择的发送或接收操作。FIFO 包含一个宽度为 32 位且深度为 16 字的数据缓冲器和控制逻辑。由于数据 FIFO 在 AHB 时钟域 (sdmmc_hclk) 中运行，因此来自 SDMMC 时钟域 (SDMMC_CK/sdmmc_rx_ck) 的子单元中的所有信号都将重新同步。

FIFO 可进入以下状态之一：

- 向卡发送数据时，发送 FIFO 引用发送逻辑和数据缓冲器。(DTDIR = 0)
- 从卡接收数据时，接收 FIFO 引用接收逻辑和数据缓冲器。(DTDIR = 1)

当来自 FIFO 的 SDMMC 数据传输正确完成时，将由数据路径子单元驱动 DATAEND 标志来指示。当来自 FIFO 的 SDMMC 数据传输出错（中止）时，将由数据路径子单元驱动其中一个错误标志（DCRCFAIL、DTIMEOUT 和 DABORT）或由 FIFO 控制驱动其中一个 FIFO 错误标志（TXUNDERR 和 RXOVERR）来指示。

可以通过以下方式访问数据 FIFO，请参见*表 439*。

表 439. 数据 FIFO 访问

数据 FIFO 访问	IDMAEN
在固件的控制下通过 AHB 从接口访问	0
在 IDMA 的控制下通过 AHB 主接口访问	1

发送 FIFO：

当 DPSM 已激活 (DPSMACT = 1) 时，数据可以写入发送 FIFO。

当 IDMAEN = 1 时，FIFO 由 IDMA 完全处理。

当 IDMAEN = 0 时，FIFO 由固件通过 AHB 从接口控制。可以通过连续地址来访问发送 FIFO。传输 FIFO 包含一个数据输出寄存器，其中存储了读取指针所指向的数据字。在数据路径子单元已加载了其移位寄存器之后，该子单元会将读取指针递增并将新数据推出。发送 FIFO 的处理方式如下：

1. 将数据长度写入 DATALENGTH，将块长度写入 DBLOCKSIZE。
   - 对于块数据传输 (DTMODE = 0)，DATALENGTH 应为 DBLOCKSIZE 的倍数。
2. 将 SDMMC 设置为发送模式（DTDIR = 0）。
   - 将 FIFO 配置为发送模式。
3. 使能数据传输。
   - 通过将 CMDTRANS 位置 1 从 CPSM 发送一条命令。
   - 通过将 DTEN 位置 1。
4. 当 (DPSMACT = 1) 将数据写入 FIFO 时。
   - DPSM 将保持 Wait_S 状态，直到 FIFO 已满 (TXFIFOF = 1) 或者达到 DATALENGTH 指示的数字。
   - 只要 FIFO 不为空，SDMMC 就会开始发送数据。
5. 当 FIFO 为半空（TXFIFOHE 标志）时，将数据写入 FIFO，直到 FIFO 已满 (TXFIFOF = 1)，或者写入最后一个数据。
6. 当写入最后一个数据后，等待数据结束（DATAEND 标志）
   - SDMMC 已发送完所有数据且 DPSM 处于禁止状态 (DPSMACT = 0)。

如果在 IDMAEN = 0 时发生数据传输错误或保持传输，固件将停止写入 FIFO，并使用 FIFORST 寄存器位清空和复位 FIFO。

表 440 中列出了发送 FIFO 状态标志。

表 440. 传输 FIFO 状态标志

标志	说明
TXFIFOF	当所有发送 FIFO 字都包含有效数据时，设置为高电平。
TXFIFOE	当传输 FIFO 不包含有效数据时，设置为高电平。
TXFIFOHE	当一半或更多个发送 FIFO 字为空时，设置为高电平。
TXUNDERR	发生下溢错误时，设置为高电平。通过写入到 SDMMC 清零寄存器可将该位清零。

接收 FIFO：

当 DPSM 已激活 (DPSMACT = 1) 时，可以从接收 FIFO 读取数据。

当 IDMAEN = 1 时，FIFO 由 IDMA 完全处理。

当 IDMAEN = 0 时，FIFO 由固件通过 AHB 从接口控制。当数据路径子单元接收到数据字时，它会驱动写入数据总线上的数据。在写入操作完成后，写入指针递增。在读取端，读取指针的当前值指向的 FIFO 字内容被推入读取数据总线中。可以通过连续地址来访问接收 FIFO。接收 FIFO 的处理方式如下：

1. 将数据长度写入 DATALENGTH，将块长度写入 DBLOCKSIZE。
   – 对于块数据传输 (DTMODE = 0)，DATALENGTH 应为 DBLOCKSIZE 的倍数。
2. 将 SDMMC 设置为接收模式（DTDIR = 1）。
   – 将 FIFO 配置为接收模式。
3. 使能 DPSM 传输。
   – 通过将 CMDTRANS 位置 1 从 CPSM 发送一条命令。
   – 或者通过将 DTEN 位置 1。
4. 当 (DPSMACT = 1) FIFO 准备好接收数据时。
   – DPSM 会将接收到的数据写入 FIFO。
5. 当 FIFO 为半满（RXFIFOHF 标志）时，从 FIFO 读取数据，直到 FIFO 为空 (RXFIFOE = 1)。
6. 如果在数据结束（DATAEND 标志）时接收到最后一个数据，则从 FIFO 读取数据，直到 FIFO 为空 (RXFIFOE = 1)。
   – SDMMC 已接收完所有数据且 DPSM 处于禁止状态 (DPSMACT = 0)。

如果在 IDMAEN = 0 时保持数据传输，固件应读取剩余数据，直到 FIFO 为空，并使用 FIFORST 寄存器位复位 FIFO。这将导致 DPSM 进入空闲状态 (DPSMACT = 0)。

如果在 IDMAEN = 0 时出现数据传输错误，固件应停止读取 FIFO，并使用 FIFORST 寄存器位清空和复位 FIFO。这将导致 DPSM 进入空闲状态 (DPSMACT = 0)。

表 441 中列出了接收 FIFO 状态标志。

表 441. 接收 FIFO 状态标志

标志	说明
RXFIFOF	当所有接收 FIFO 字都包含有效数据时，设置为高电平。
RXFIFOE	当接收 FIFO 不包含有效数据时，设置为高电平。
RXFIFOHF	当一半或更多个接收 FIFO 字包含有效数据时，设置为高电平。
RXOVERR	发生溢出错误时，设置为高电平。通过写入到 SDMMC 清零寄存器可将该位清零。

CLKMUX 单元

CLKMUX 选择用于与接收到的数据和命令响应搭配使用的 sdmmc_rx_ck 时钟源。可以通过时钟控制寄存器位 SELCLKRX 从以下几项中选择接收数据时钟源：

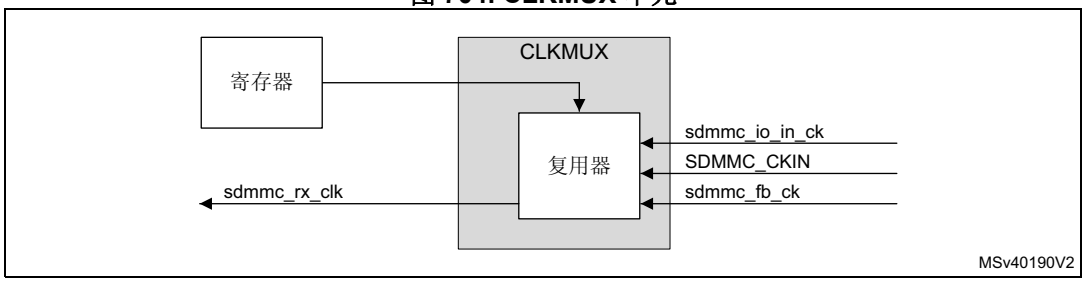
- sdmmc_io_in_ck 总线主设备主反馈时钟。
- SDMMC_CKIN 外部总线反馈时钟。
- sdmmc_fb_ck 总线调节反馈时钟。

当没有外部驱动器且使用 DS 和 HS 时，选择 sdmmc_io_in_ck。

当有外部驱动器且使用 SDR12、SDR25、SDR50 和 DDR50 时，选择 SDMMC_CKIN。

如果器件上的 DLYB 模块与 SDR104 模式搭配使用并且可选择与 SDR50 和 DDR50 模式搭配使用，则必须选择 sdmmc_fb_ck 时钟输入。

图 704. CLKMUX 单元



当 CPSM 和 DPSM 处于空闲状态时，应更改 sdmmc_rx_ck 源。

55.4.5 SDMMC AHB 从接口

AHB 从接口生成中断请求，并且访问 SDMMC 适配器寄存器和数据 FIFO。该接口由数据路径、寄存器解码器和中断逻辑构成。

SDMMC FIFO

FIFO 访问仅限于字访问：

- 在发送 FIFO 模式下
  - 数据按字（32 位）写入 FIFO，直到 DATALENGTH 所指示的所有数据均传输完毕。当 DATALENGTH 不是 4 的倍数时，最后剩余的数据（1、2 或 3 字节）通过字传输的方式写入。
- 在接收 FIFO 模式下
  - 数据按字（32 位）从 FIFO 读取，直到 DATALENGTH 所指示的所有数据均传输完毕。当 DATALENGTH 不是 4 的倍数时，最后剩余的数据（1、2 或 3 字节）通过以 0 值字节填充的字传输读取。

当通过半字或字节访问的方式访问 FIFO 时，会生成 AHB 总线故障。

SDMMC 中断

中断逻辑生成一个中断请求信号，当至少其中一个非屏蔽状态标志有效时，便发出此信号。提供了一个屏蔽寄存器，用于选择将生成中断的条件。如果对应的屏蔽标志置 1，则状态标志将生成中断请求。一些状态标志需要在清零寄存器中隐式清零。

55.4.6 SDMMC AHB 主接口

AHB 主接口用于在存储器和 FIFO 之间使用 SDMMC IDMA 传输数据。

SDMMC IDMA

直接存储器访问 (DMA) 用于在 SDMMC FIFO 与存储器之间提供高速传输。AHB 主设备可优化系统总线的带宽。SDMMC 内部 DMA (IDMA) 提供一个用于发送或接收的通道。

IDMA 通过 IDMAEN 位使能，并支持 8 节拍的突发传输。

- 在发送突发传输模式下：
  - 只要 FIFO 中的突发传输次数为空，就会以突发形式从存储器中获取数据，直到 DATALENGTH 所指示的所有数据均已传输完毕。如果 DATALENGTH 不是突发大小的倍数，则不足突发大小的剩余数据将使用单次传输模式进行传输。
    当 DATALENGTH 不是 4 的倍数时，最后剩余的数据（1、2 或 3 字节）通过字传输的方式获取。
- 在接收突发传输模式下：
  - 只要 FIFO 中包含突发传输次数，就会以突发形式将数据存储到存储器中，直到 DATALENGTH 所指示的所有数据均已传输完毕。
    如果 DATALENGTH 不是突发传输的倍数，则不足突发大小的剩余数据将使用单次传输模式进行传输。
    当 DATALENGTH 不是 4 的倍数时，最后剩余的数据（1、2 或 3 字节）通过半字或字节传输的方式存储。

此外，IDMA 还提供有两种通道配置（通过位 IDMABMODE 选择）：

- 单缓冲通道
- 双缓冲通道

在单缓冲区配置中，存储器端的数据从基址 IDMABASE0 开始以线性方式进行访问。当 IDMA 完成所有数据的传输并且 DPSM 也完成传输后，DATAEND 标志将置 1。

在双缓冲区配置中，存储器端的数据从 2 个缓冲区进行访问，一个从基址 IDMABASE0 开始，另一个从基址 IDMABASE1 开始。这样一来，当 IDMA 访问其中一个存储器缓冲区时，固件可以处理另一个存储器缓冲区。存储器缓冲区的大小由 IDMABSIZE 定义。缓冲区大小应为突发大小的倍数。使能通道时，可以即时更新缓冲区的基址，适用的规则如下：

- 当 IDMABACT 位为“0”时，IDMA 硬件使用 IDMABASE0 来访问存储器。当尝试通过固件写入该寄存器时，写操作将被丢弃，IDMABASE0 数据将不会发生变化。允许固件写入 IDMABASE1。
- 当 IDMABACT 位为“1”时，IDMA 硬件使用 IDMABASE1 来访问存储器。当尝试通过固件写入该寄存器时，写操作将被丢弃，IDMABASE1 数据将不会发生变化。允许固件写入 IDMABASE0。

当 IDMA 完成其中一个缓冲区的数据传输时，缓冲区传输完成标志 (IDMABTC) 将置 1 且 IDMABACT 位将发生翻转，之后 IDMA 继续从另一个缓冲区传输数据。当 IDMA 完成所有数据的传输并且 DPSM 也完成传输后，DATAEND 标志将置 1。

IDMABASEn 地址应按字对齐。

错误管理

当对保留的地址空间执行读写操作时，将出现 IDMA 传输错误。在出现 IDMA 传输错误时，后续的 IDMA 传输将禁止，IDMATE 标志将置 1。IDMATE 标志的行为取决于 SDMMC 传输期间出现 IDMA 传输错误的时间：

- 在出现任何 SDMMC 传输错误（TXUNDERR、RXOVERR、DCRCFAIL 或 DTIMEOUT）之前检测到 IDMA 传输错误：
  - IDMATE 标志与 SDMMC 传输错误标志同时置 1。
  - 生成 TXUNDERR、RXOVERR、DCRCFAIL 或 DTIMEOUT 中断。
- 在 STOP_TRANSNMISSION 命令期间检测到 IDMA 传输错误：
  - IDMATE 标志与 DABORT 标志同时置 1。
  - 生成 DABORT 中断。
- 在 SDMMC 传输（HOLD 或 DATAEND）结束时检测到 IDMA 传输错误。
  - IDMATE 标志在 SDMMC 传输结束时置 1。
  - 生成 SDMMC 传输结束中断且 HOLD 或 DATAEND 标志置 1。

IDMATE 将在出现其他 SDMMC 传输中断（TXUNDERR、RXOVERR、DCRCFAIL、DTIMEOUT、DABORT、HOLD 或 DATAEND）时生成。

55.4.7 MDMA 请求生成

来自 SDMMC 的内部触发线可向 MDMA 控制器发送直接请求，以在不使用 CPU 的情况下实现自/至不同内部 RAM 地址的连续传输。

自/至卡的数据传输成功完成后，状态寄存器的 DATAEND 标志将置 1。通过 sdmmc_dataend_trg 输出向 MDMA 请求输入指示此事件。它可触发 DATAEND 和 CMDREND 标志的清零，并最终通过对 SDMMC 控制和配置寄存器进行 MDMA 直接访问来开始新的传输，而无需 CPU 干预。

下表给出了根据 SDMMC 请求在 MDMA 中进行编程的操作：

表 442. SDMMC 与 MDMA 的连接

触发信号	指示事件	事件发生的条件	MDMA 传输配置	MDMA 操作
sdmmc_ dataend_trg	成功数据传输结束	DATAEND = 1	单独	将 DATAENDC 置 1

55.4.8 AHB 和 SDMMC_CK 时钟的关系

AHB 带宽应至少为 SDMMC 总线带宽的 3 倍，即对于 SDR50 4 位模式 (50 MB/s)，最小 sdmmc_hclk 频率为 37.5 MHz (150 MB/s)。

表 443. AHB 和 SDMMC_CK 时钟频率的关系

SDMMC 总线模式	SDMMC 总线带宽	最大 SDMMC_CK [MHz]	最小 AHB 时钟 [MHz]
MMC DS	8	26	19.5
MMC HS	8	52	39
MMC DDR52	8	52	78
MMC HS200	8	200	150

表 443. AHB 和 SDMMC_CK 时钟频率的关系

SDMMC 总线模式	SDMMC 总线带宽	最大 SDMMC_CK [MHz]	最小 AHB 时钟 [MHz]
SD DS / SDR12	4	25	9.4
SD HS / SDR25	4	50	18.8
SD DDR50	4	50	37.5
SD SDR50	4	100	37.5
SD SDR104	4	208	78

55.5 卡功能说明

55.5.1 SD I/O 模式

以下功能是专用于 SDMMC 的操作：

- SDIO 中断
- SDIO 挂起/恢复操作（写入和读取挂起）
- 通过停止时钟执行的 SDIO 读取等待操作
- 通过触发 SDMMC_D2 执行的 SDIO 读取等待操作

表 444. SDIO 特殊操作控制

操作模式	SDIOEN	RWMOD	RWSTOP	RWSTART	DTDIR
中断检测	1	X	X	X	X
挂起/恢复操作	X	X	X	X	X
读取等待 SDMMC_CK 时钟停止 (START)	X	1	0	1	1
读取等待 SDMMC_CK 时钟停止 (STOP)	X	1	1	1	1
读取等待 SDMMC_D2 信号 (START)	X	0	0	1	1
读取等待 SDMMC_D2 信号 (STOP)	X	0	1	1	1

SD I/O 中断

为使 SD I/O 卡能够中断主机，在 SD 接口的引脚 8（4 位模式下与 SDMMC_D1 共用）上提供了一个中断功能。每个卡或者卡中的功能都可以使用中断。SD I/O 中断为电平敏感型，这意味着，在主机识别出中断线并对其采取操作或者由于中断周期结束而使其失效之前，中断线必须保持有效状态（低电平）。在主机处理完中断后，将通过在 SD I/O 卡内部寄存器中的相应位中进行 I/O 写入来清除中断状态位。所有 SD I/O 卡的中断输出均为低电平有效，并且应用程序必须在所有数据线 (SDMMC_D[3:0]) 上提供外部上拉电阻。

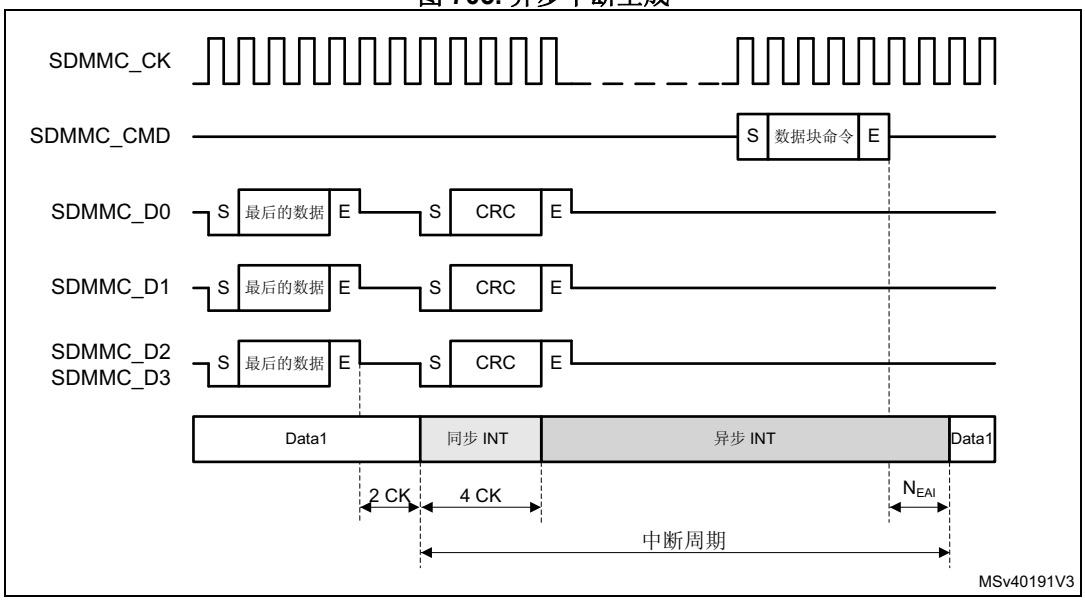
在 SD 1 位模式下，引脚 8 专用于中断功能 (IRQ)，并且中断没有时序限制。

在 SD 4 位模式下，只有在中断周期内，主机才会将引脚 8 (SDMMC_D1/IRQ) 的电平采样到中断检测器中。而在所有其他时候，主机中断将忽略此值。当卡上使能中断且 SDIOEN 位置 1 时，中断周期开始，请参见见*表 444* 中的寄存器设置。

在 4 位模式下，卡可以生成同步或异步中断，分别通过卡 CCCR 寄存器中的 SAI 和 EAI 位指示。

- 同步中断，要求 SDMMC_CK 有效。
- 异步中断，可以在 SDMMC_CK 停止时生成，即在最后一个数据块后的卡中断周期开始后的 4 个周期后生成。

图 705. 异步中断生成



中断周期的时序取决于总线速度模式：

在 DS、HS、SDR12 和 SDR25 模式（通过寄存器位 BUSSPEED 选择）下，中断周期与 SD 时钟同步。

- 中断周期在数据块传输命令（在 CMDTRANS 位置 1 时发送命令）的结束位的下一个时钟结束或在 DTEN 位置 1 时结束。
- 中断周期在数据块完成传输后的 2 个 SDMMC_CK 后恢复。
- 在数据块间隙，中断周期被限制为 2 个 SDMMC_CK 周期。

注： *DTEN 不能用于启动与 SD 和 eMMC 卡之间的数据传输。*

图 706. 同步中断周期数据读取

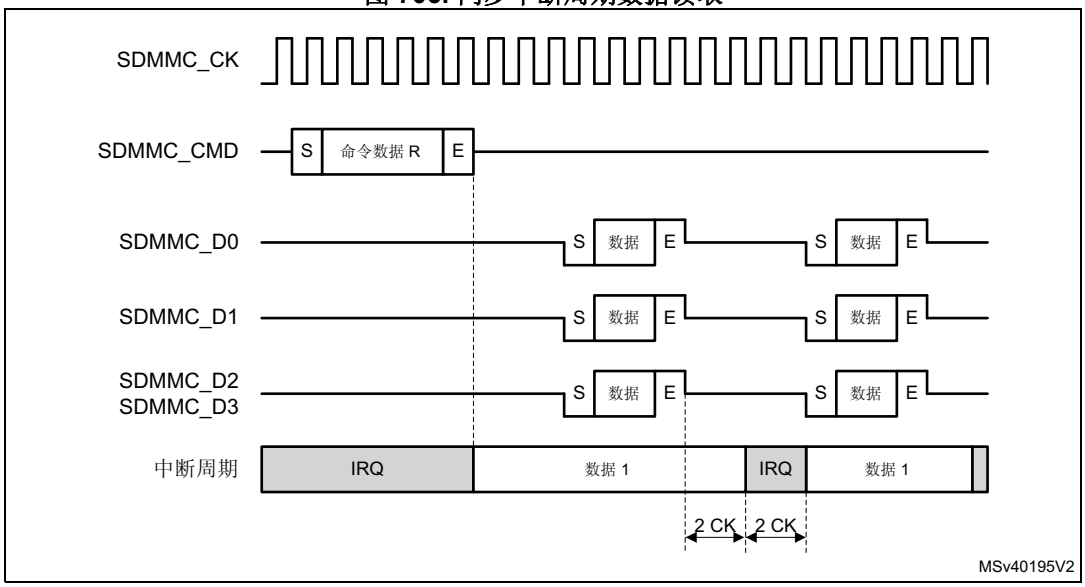
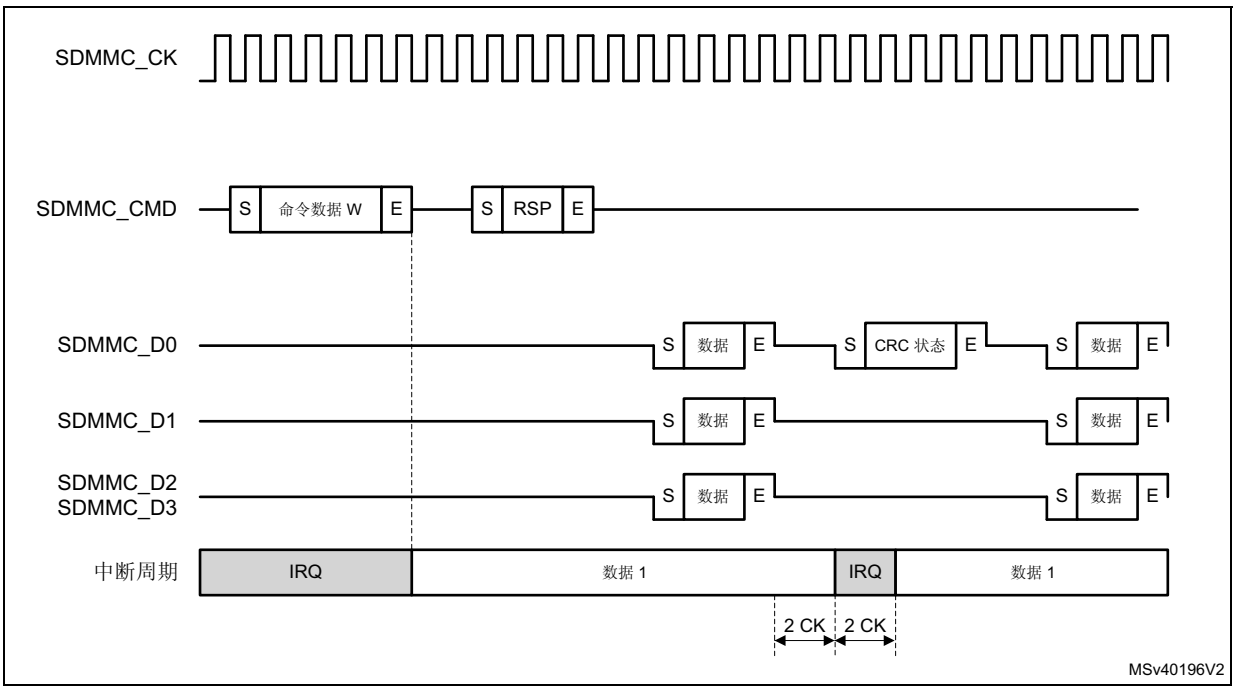


图 707. 同步中断周期数据写入



在 SDR50、SDR104 和 DDR50（通过寄存器位 BUSSPEED 选择）下，由于卡与主机之间存在传播延迟，因此中断周期是异步的。

- 卡中断周期在数据块传输命令（在 CMDTRANS 位置 1 时发送命令）的结束位后的 0 到 2 个 SDMMC_CK 周期后结束或在 DTEN 位置 1 时结束。在主机中，中断周期在数据块传输命令的结束位之后结束。如果在此中断周期内主机未检测到命令结束位，则将在 1 到 2 个周期后发出卡中断。
- 卡中断周期在最后一个数据块完成传输后的 2 到 4 个 SDMMC_CK 后恢复。主机将始终在最后一个数据块之后的 2 个周期后恢复中断周期。
- 数据块间隙内没有中断周期。

注： *DTEN 不能用于启动与 SD 和 eMMC 卡之间的数据传输。*

图 708. 异步中断周期数据读取

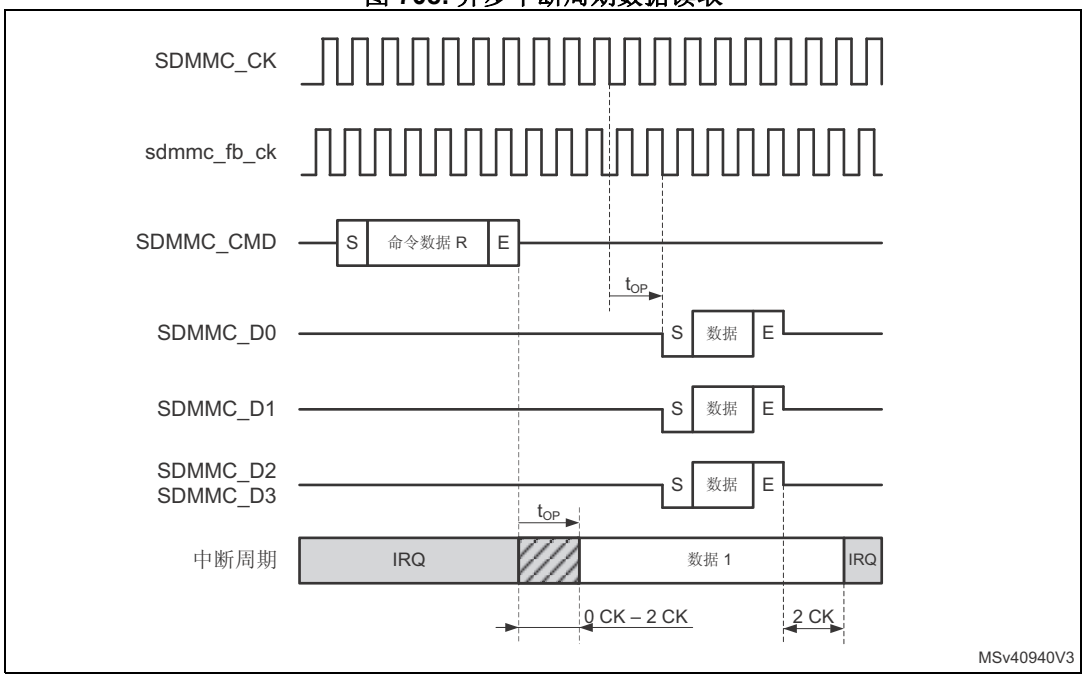
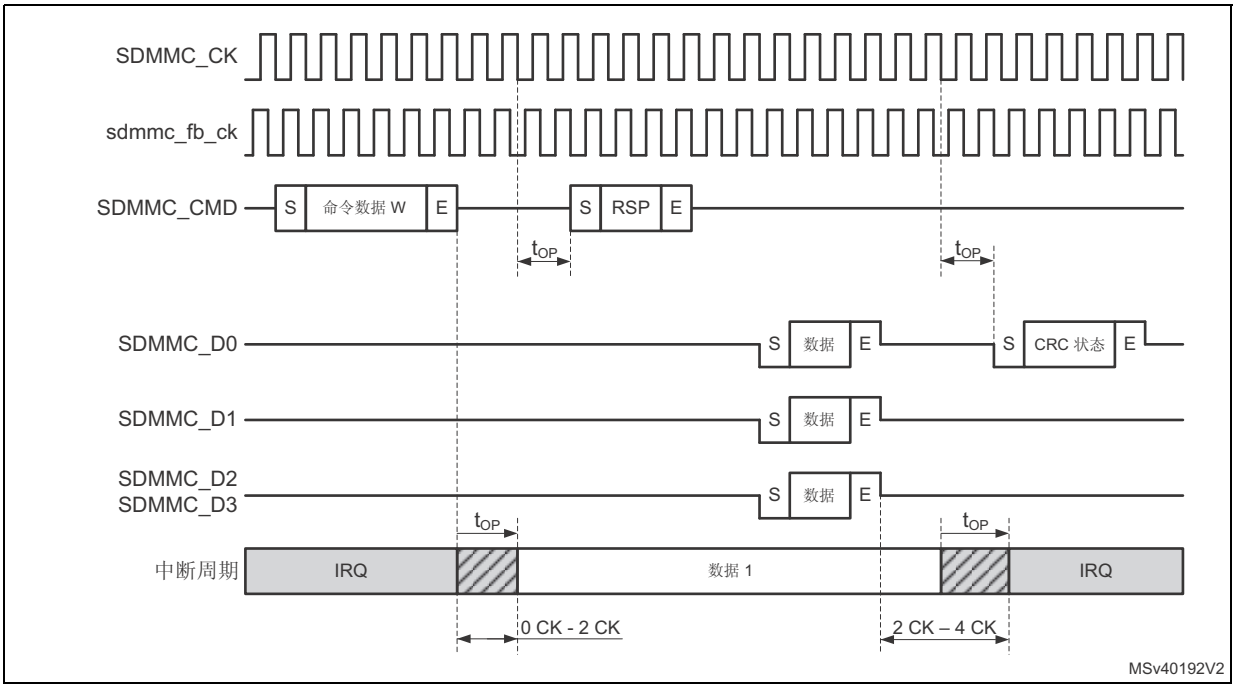


图 709. 异步中断周期数据写入



当传输开放式多块数据并使用 DTMODE “以 STOP_TRANSMISSION 命令结束的块数据传输”时， SDMMC 将在最后一个数据块之后屏蔽中断周期，直到 CMD12 STOP_TRANSMISSION 命令结束。

中断周期适用于存储器和 I/O 操作。

在 4 位模式下，可以根据*表 445* 来区分中断与其他信号。

表 445. 4 位模式启动、中断和 CRC 状态信号检测

SDMMC 数据线	启动	中断	CRC 状态
SDMMC_D0	0	1 或 CRC 状态	0
SDMMC_D1	0	0	X
SDMMC_D2	0	1 或读取等待	X
SDMMC_D3	0	1	X

SD I/O 挂起和恢复

SDIO V4.00 或更高版本不支持此功能。

在多功能 SD I/O 中或者在同时具备 I/O 和存储器功能的卡中，有多个设备（I/O 和存储器）共享对 MMC/SD 总线的访问权限。要在多个设备之间共享对主机的访问权限，SD I/O 和组合卡可以有选择性地实施“挂起/恢复”这一概念。如果卡支持挂起/恢复，则主机可以临时停止（挂起）对某一功能或存储器的数据传输操作以释放总线，从而将总线用于其他功能或存储器的更高优先级传输。此更高优先级传输完成后，原始传输将从中断位置重新开始（恢复）。

要在总线上执行挂起/恢复操作，主机将执行以下步骤：

1. 确定当前使用 SDMMC_D [3:0] 线的功能
2. 请求将优先级较低或速度较慢的事务挂起
3. 等待事务挂起完成
4. 开始更高优先级的事务
5. 等待更高优先级事务完成
6. 恢复挂起的事务

接收到挂起命令的卡将使用其当前总线状态进行响应。只有当总线通过卡挂起后，总线状态才会指示挂起完成。

有多种挂起情况：

- 在数据传输开始之前接受挂起请求。
- 未接受挂起请求（由于同时在传输数据），主机持续检查请求，直到接受请求。（数据传输已停止）
- 写繁忙时的挂起请求。
- 多次写入时的挂起请求。
- 读取等待期间的挂起请求。

为了让主机获知总线是否已释放，应检查挂起请求的状态（挂起完成）。

当挂起请求响应的总线状态指示挂起完成时，表示卡已释放总线。此时，应保存挂起操作的状态，之后可启动其他操作。

挂起命令应在 CMDSUSPEND 位置 1 时发送。这样，当总线挂起时（响应位 BS = 0），可在挂起命令响应之后启动中断周期。

硬件不会保存挂起的操作恢复后还需传输的剩余数据量。固件负责确定已传输的数据，从而保证恢复后的剩余字节数是正确的。

从卡中接收数据时，SDMMC 可以在读取数据块结束（DPSM 处于 Wait_R 状态）之后挂起读取操作。在收到卡的挂起确认响应后，固件应执行以下步骤：

1. 应通过将 DTHOLD 位置 1 来停止正常接收过程。
   a) 应读取 FIFO 中的剩余数据字节数，直到接收 FIFO 为空（RXFIFOE 标志置 1），当 IDMAEN = 0 时，FIFO 应使用 FIFORST 复位。
2. 通过 DHOLD 标志指示对已从 FIFO 中读取所有数据以及挂起完成的确认。
   a) 恢复操作后仍需读取的剩余数据字节数（数据块的倍数）应通过 DATACOUNT 指示的剩余字节数确定。

注： *挂起过程中出现的 DTIMEOUT 标志将被忽略。*

要继续从卡中接收数据，固件应执行以下步骤：

1. 应在 DATALENGTH 中编程剩余的数据字节数（数据块的倍数）。
2. 应在 DTDIR 位中将 DPSM 配置为接收数据。
3. 应从 CPSM 发送恢复命令，此时 CMDTRANS 位置 1 且 CMDSUSPEND 位置 1，该命令将在数据传输恢复（响应位 DF = 1）且 DPSM 使能时结束中断周期，之后卡将继续发送数据。

在向卡发送数据时，SDMMC 可以在写入数据块 CRC 状态结束（DPSM 处于繁忙状态）之后挂起写入操作。在向卡发送挂起命令前，固件应执行以下步骤：

1. 使能 DHOLD 标志（当 IDMAEN = 0 时使能 DBCKEND 标志）
2. 应防止 DPSM 通过将 DTHOLD 置 1 开始发送新的数据块。
3. 当 IDMAEN = 0 时：如果接收到 DBCKEND 标志，将停止数据传输。固件可以停止填充 FIFO，之后应使用 FIFORST 复位 FIFO。恢复操作后，仍保留在 FIFO 中的任何字节都需要重写。
4. 当接收到 DHOLD 标志时，将停止数据传输。恢复操作后仍需写入的剩余数据字节数应通过 DATACOUNT 指示的剩余字节数确定。
5. 要挂起卡，应在 CMDSUSPEND 位置 1 时通过 CPSM 发送挂起命令。这样，当总线挂起时（响应位 BS = 0），可在挂起命令响应之后启动中断周期。

要继续向卡发送数据，固件应执行以下步骤：

1. 应在 DATALENGTH 中编程剩余的数据字节数。
2. 应通过设置 DTDIR 来配置 DPSM 的传输方向，通过 CPSM 发送恢复命令（CMDTRANS 位置 1 且 CMDSUSPEND 位置 1）的方式来使能 DPSM。这将结束中断周期并开始数据传输。DPSM 将在 SDMMC_D0 未发出繁忙信号时进入 Wait_S 状态，而在 SDMMC_D0 发出繁忙信号时进入繁忙状态。
3. 当 IDMAEN = 1 时：需要针对要传输的剩余字节重新编程 DMA。
4. 当 IDMAEN = 0 时：固件应开始使用剩余的数据填充 FIFO。

SD I/O 读取等待

有两种方法可以在块间隙内暂停数据传输：

1. 停止 SDMMC_CK。
2. 在 SDMMC_D2 上使用读取等待信号。

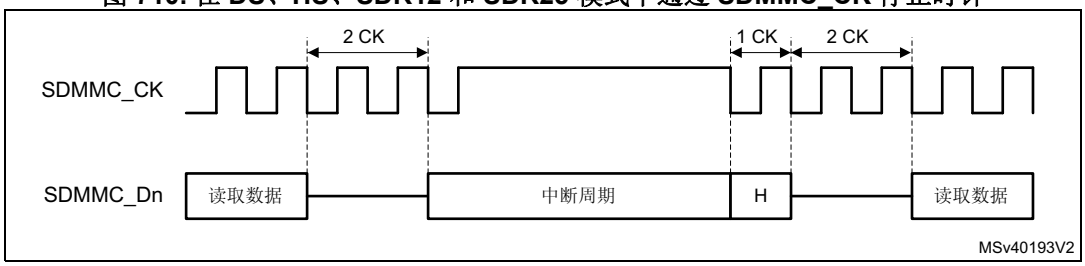
SDMMC 可以根据*表 444* 通过寄存器设置执行读取等待。

根据 SDMMC 工作模式（DS、HS、SDR12 和 SDR25）或（SDR50、SDR104 和 DDR），每种方法都有不同的特性。

在 DS、HS、SDR12 和 SDR25 模式下通过停止 SDMMC_CK 实现暂停读取操作的时序中，SDMMC_CK 可在结束位之后的 2 个 SDMMC_CK 周期后停止。主机就绪后将通过重启时钟

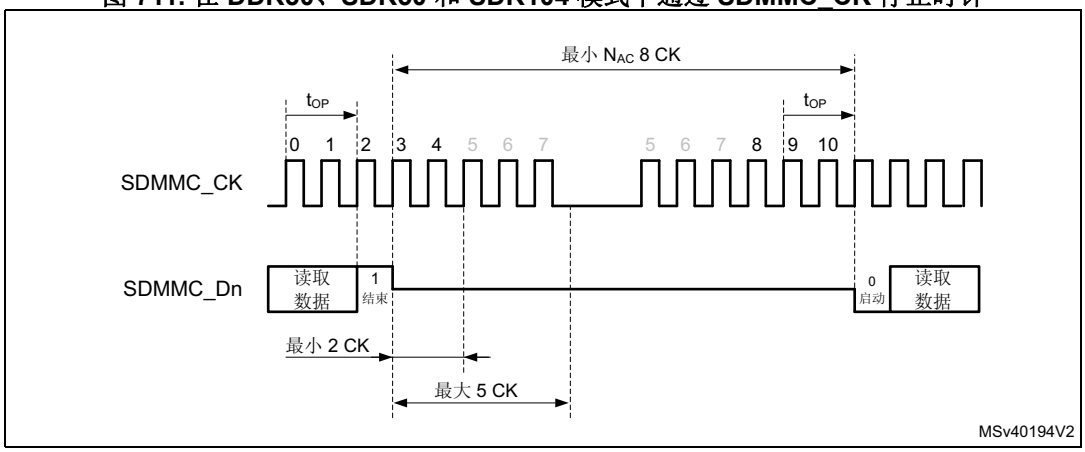
进行恢复，请参见*图 710*。

图 710. 在 DS、HS、SDR12 和 SDR25 模式下通过 SDMMC_CK 停止时钟



在 SDR50、SDR104 和 DDR50 模式下通过停止 SDMMC_CK 实现暂停读取操作的时序中，SDMMC_CK 可在结束位之后的 2-5 个 SDMMC_CK 周期后停止。主机就绪后将通过重启时钟进行恢复，请参见*图 711*。（在 DDR50 模式下，如果时钟线为低电平，则只应在下降沿后停止 SDMMC_CK）。

图 711. 在 DDR50、SDR50 和 SDR104 模式下通过 SDMMC_CK 停止时钟



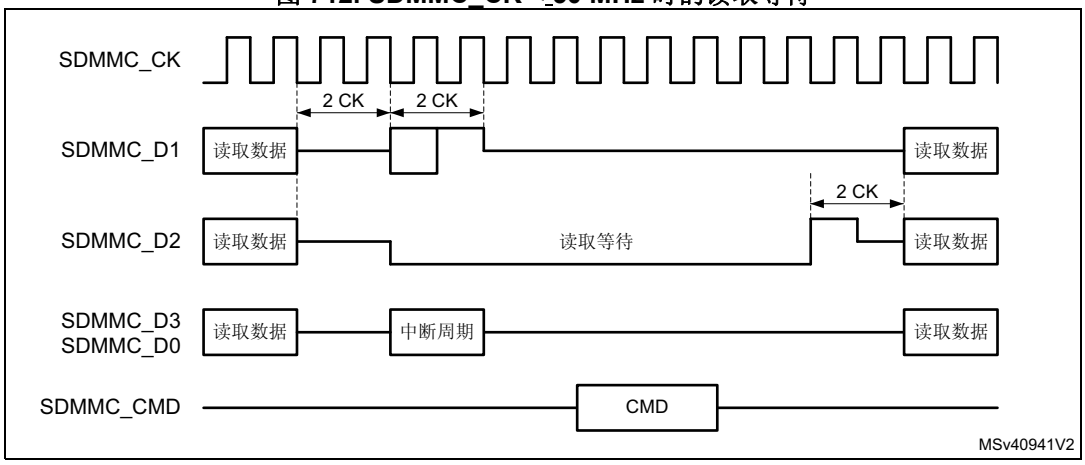
在读取等待 SDMMC_CK 时钟停止期间，如果 RWSTART 置 1，DSPM 会在当前接收到的数据块 CRC 的结束位之后停止时钟。时钟将在向 RWSTOP 位写入 1 后再次启动，之后 DPSM 将等待来自卡的起始位。

SDMMC_CK 停止时，无法向卡发出任何命令。在读取等待间隔内，SDMMC 仍可在 SDMMC_D1 上检测 SDIO 中断。

SDMMC_D2 上可选的读取等待信号 (RW) 操作仅针对 SD 的 1 位和 4 位模式进行了定义。“读取等待”操作允许主机发出如下信号：卡正在读取多个寄存器 (IO_RW_EXTENDED, CMD53) 以临时停止数据传输，同时允许主机向 SD I/O 设备中的任何功能发送命令。要确定卡是否支持“读取等待”协议，主机必须测试内部卡寄存器中的功能位。

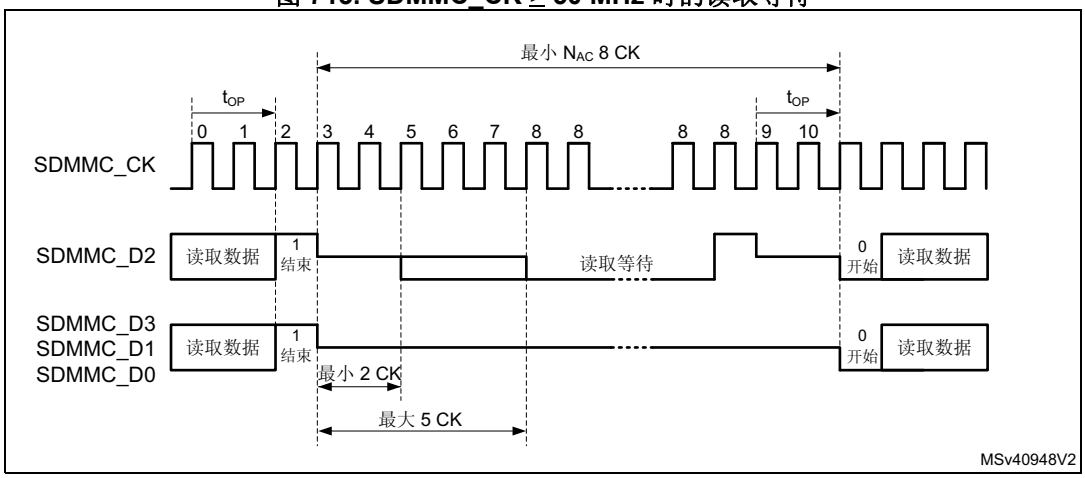
SDMMC_CK 小于 50 MHz（DS、HS、SDR12 和 SDR25）的读取等待的时序基于卡在 SDMMC_D1 上生成的中断周期。主机在中断周期内将 SDMMC_D2 置为低电平，请求卡进入读取等待状态。要退出读取等待状态，主机应在将 SDMMC_D2 设置为高阻态之前的一个 SDMMC_CK 周期内将 SDMMC_D2 升为高电平，请参见*图 712*。

图 712. SDMMC_CK < 50 MHz 时的读取等待



对于 SDR50、SDMMC_CK 大于 50 MHz 的 SDR104 以及 DDR50，卡会将 SDMMC_D2 上的读取等待请求视为异步事件。主机在 2-5 个 SDMMC_CK 周期后将 SDMMC_D2 置为低电平，请求卡进入读取等待状态。要退出读取等待状态，主机应在将 SDMMC_D2 设置为高阻态之前的一个 SDMMC_CK 周期内将 SDMMC_D2 升为高电平。主机应基于 SDMMC_CK 时钟将 SDMMC_D2 升为高电平。请参见*图 713*。

图 713. SDMMC_CK ≥ 50 MHz 时的读取等待



在读取等待 SDMMC_D2 信号期间，如果 RWSTART 置 1，DPSM 将在当前接收的数据块 CRC 的结束位后驱动 SDMMC_D2。当将 1 写入 RWSTOP 位时，SDMMC_D2 上的读取等待信号将被删除。DPSM 保持读取等待状态的时间将延长两个 SDMMC_CK 时钟周期，以将 SDMMC_D2 驱动为 1 并保持一个时钟周期（根据 SDIO 规范），之后 DPSM 将等待来自卡的起始位。

在 SDMMC_D2 上的读取等待信号传输期间，可以向卡发出命令。在读取等待间隔内，SDMMC 可以在 SDMMC_D1 上检测 SDIO 中断。

55.5.2 CMD12 发送时序

CMD12 用于停止/中止数据传输，卡数据传输在停止传输命令的结束位后的两个时钟周期后终止。

表 446. 应用案例

数据操作	停止传输命令 CMD12 的说明
MMC 流写入	通过发送停止传输命令来停止/中止数据传输。
MMC 开放式多块写入	通过发送停止传输命令来停止/中止数据传输。 如果卡检测到错误，主机必须通过发送停止传输命令来中止操作。
预定义块数的 MMC 块写入	在该类型的多块写入操作结束时，不需要停止传输命令。（在卡接收到最后一个块后发送停止传输命令被视为非法命令。） 如果卡检测到错误，主机必须通过发送停止传输命令来中止操作。
MMC 流读取	通过发送停止传输命令来停止/中止数据传输。
MMC 开放式多块读取	通过发送停止传输命令来停止/中止数据传输。 如果卡检测到错误，主机必须通过发送停止传输命令来中止操作。
预定义块数的 MMC 块读取	在该类型的多块读取操作结束时，不需要停止传输命令。（在卡发送最后一个块后发送停止传输命令被视为非法命令。） 通过发送停止传输命令可中止事务。 如果卡检测到错误，主机必须通过发送停止传输命令来中止操作。

所有数据写入和读取命令都可以随时通过停止传输命令 CMD12 中止。如果正在进行数据传输，请按照以下数据中止步骤操作：

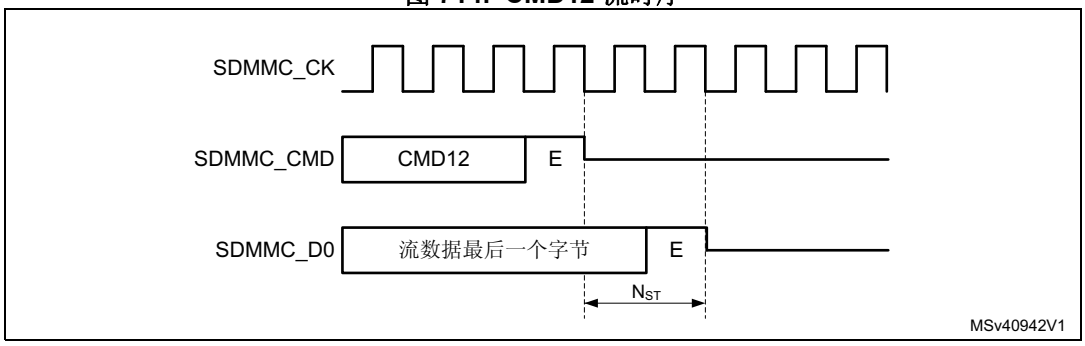
1. 在寄存器中加载 CMD12 停止传输命令并将 CMDSTOP 位置 1。
   a) 当命令发送到 DPSM 时，这将导致 CPSM 生成中止信号。
2. 将 CPSM 配置为立即发送命令（将 WAITPEND 位清零）。
   a) 发送数据时，卡将在停止传输命令结束位后的 2 个周期后停止数据传输。
   未发送数据时，卡将不会开始发送任何新数据。
   b) 发送数据时，主机将在停止传输命令结束位之后发送最后一个数据位，后跟一个结束位。
   未发送数据时，主机将不会开始发送任何新数据。
3. 当 IDMAEN = 0 时，需要使用 FIFORST 复位 FIFO。
   a) 向卡写入数据时。出现 CMDREND 标志时，固件将停止向 FIFO 写入数据。随后，将使用 FIFORST 复位 FIFO，这将清空 FIFO。
   b) 从卡读取数据时。出现 CMDREND 标志时，固件将从 FIFO 读取剩余数据。随后，将使用 FIFORST 复位 FIFO。
4. 当 IDMAEN = 1 时，硬件将处理 FIFO。
   a) 向卡写入数据时。出现中止信号时，硬件将停止 IDMA，随后 FIFO 将被清空。
   b) 从卡读取数据时。出现中止信号时，硬件将指示 IDMA 将剩余数据从 FIFO 传输到 RAM。
5. 当 FIFO 为空/复位时，将生成 DABORT 标志。

流操作和 CMD12

要在待传输的最后一个字节之后停止流传输，应在数据流的最后一个字节结束时发送 CMD12 结束位时序。请按照以下流数据写入步骤操作：

1. 在 DPSM 中初始化流数据，DTMODE = MCC 流数据传输。
2. 从 CPSM 发送 WRITE_DATA_STREAM 命令 (CMDTRANS = 1)。
3. 在命令寄存器中预加载 CMD12，CMDSTOP 位应置 1。
4. 将 CPSM 配置为仅在最后一个数据（根据 DATALENGTH）的等待挂起 (WAITPEND = 1) 结束后发送命令。
5. 使能 CPSM 发送 STOP_TRANSMISSION 命令，流数据结束位和命令结束位将对齐。
   a) 当 DATALENGTH > 5 字节时，CPSM 中将等待命令 CMD12 与数据传输结束位对齐。
   b) 当 DATALENGHT < 5 字节时，命令 CMD12 将提前启动，DPSM 将保持 Wait_S 状态，以将数据传输结束位与 CMD12 结束位对齐。
6. 可通过将 WAITPEND 位清零随时中止流数据写入过程。这将导致预加载的 CMD12 立即发送，并停止数据流写入过程。

图 714. CMD12 流时序



要在最后一个字节之后停止流传输读取过程，应在数据流最后一个字节后发送 CMD12 结束位时序。请按照以下流数据读取步骤操作：

1. 等待 DPSM 接收到所有数据（DATAEND 标志）。
   a) 即使卡发送更多数据，DPSM 接收的数据也不会超过 DATALENGTH 所指示的数据量。
2. 通过 CPSM 发送 CMD12。
   a) CMD12 将使卡停止发送数据。

注： *即使卡继续发送数据，SDMMC 也不会在 DATACOUNT = 0 时从卡中继续接收任何数据。*

块操作和 CMD12

要在数据结束时停止块传输，应在最后一个块结束位之后发送 CMD12 结束位。

向卡写入数据时，将在写入数据块 CRC 令牌结束位之后发送 CMD12 结束位。这要求 CMD12 发送过程遵循数据块传输时序。要停止开放式多块写入操作，请按照以下步骤操作：

1. 在开始数据传输之前，将 DTMODE 设置为“以 STOP_TRANSMISSION 命令结束块数据传输”。
2. 等待 DPSM 发送完所有数据并等待接收到 CRC 令牌（DATAEND 标志）。
   a) DPSM 发送的数据不会超过 DATALENGTH 所指示的数据量。
3. 通过 CPSM 发送 CMD12。
   a) CMD12 会将卡设置为空闲模式。

当从卡读取数据时，CMD12 结束位应最早发送，即在卡读取数据块最后一个数据位时发送。这要求 CMD12 发送过程遵循数据块接收时序。请按照以下停止开放式多块读取数据块步骤操作：

1. 在开始数据传输之前，将 DTMODE 设置为“以 STOP_TRANSMISSION 命令结束块数据传输”。
2. 等待 DPSM 接收到所有数据（DATAEND 标志）。
   a) 即使卡发送更多数据，DPSM 接收的数据也不会超过 DATALENGTH 所指示的数据量。
3. 通过 CPSM 发送 CMD12 并将 CMDSTOP 位置 1。
   a) CMD12 将使卡停止发送更多数据，并将卡设置为空闲模式。卡将中止正在进行的任何块传输操作。

注： *即使卡继续发送数据，SDMMC 也不会在 DATACOUNT = 0 时从卡中继续接收任何数据。*

55.5.3 睡眠 (CMD5)

MMC 卡可以通过 CMD5 在睡眠状态和待机状态之间转换。在睡眠状态下，卡的功耗降至最低，此时可关闭 Vcc 电源。

CMD5 (SLEEP) 用于启动从待机状态到睡眠状态的状态转换。在转换阶段，卡指示繁忙，同时会将 SDMMC_D0 拉为低电平。当卡停止将 SDMMC_DO 线拉为低电平时，表示已进入睡眠状态。

要将卡设置为睡眠状态，请按照以下步骤操作：

1. 使能 BUSYD0END 中断。
2. 发送 CMD5 (SLEEP)。
3. 出现 BUSYD0END 中断时，卡处于睡眠状态。
4. 允许关闭 Vcc 电源。

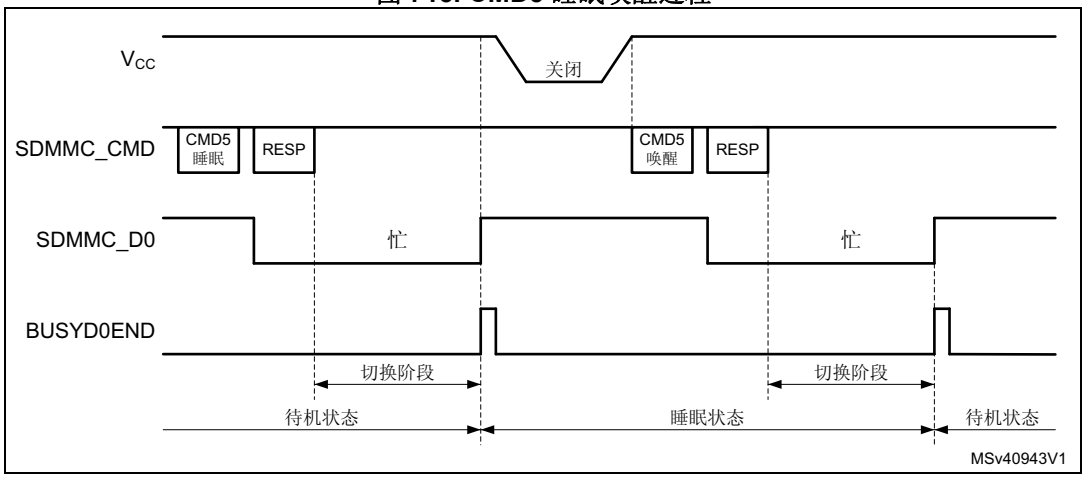
CMD5 (AWAKE) 用于启动从睡眠状态到待机状态的状态转换。在转换阶段，卡指示繁忙，同时会将 SDMMC_D0 拉为低电平。当卡停止将 SDMMC_DO 线拉为低电平时，表示已进入待机状态。

要将卡设置为睡眠状态，请按照以下步骤操作：

1. 接通 Vcc 电源，等待其达到最低工作电压。
2. 使能 BUSYD0END 中断。
3. 发送 CMD5 (AWAKE)。
4. 出现 BUSYD0END 中断时，卡处于待机状态。

只有在进入睡眠状态后，才允许关闭 Vcc 电源。在发送 CMD5 (AWAKE) 之前，应重新安装 Vcc 电源。

图 715. CMD5 睡眠唤醒过程



55.5.4 中断模式 (Wait-IRQ)

主机和卡同时进入和退出中断模式 (Wait-IRQ)。在中断模式下，不会进行任何数据传输。唯一允许的消息是来自卡或主机的中断服务请求响应。为了使中断模式正常工作，应根据开漏模式下可实现的 SDMMC_CMD 数据速率设置 SDMMC_CK 频率，具体取决于电容负载和上拉电阻。CLKDIV 应设置为 > 1，SETCLKRX 应选择 sdmmc_io_in_ck 或 SDMMC_CLKin 源。

主机必须确保发出 CMD40 (GO_IRQ_STATE) 之前卡处于待机状态。在等待中断响应时，SDMMC_CK 时钟信号必须保持有效。

中断模式下的卡（IRQ 状态）：

- 等待内部卡中断事件。出现中断事件后，卡开始发送中断服务请求响应。该响应在开漏模式下发送。
- 等待内部卡中断事件时，卡还会监视 SDMMC_CMD 线上的起始位。检测到启动位后，卡将中止中断模式并转换到待机状态。

中断模式下的主机（CPSM 等待状态，等待中断）：

- 等待卡中断服务请求响应（起始位）。
- 等待卡中断服务请求响应时，主机可中止中断模式（通过将 WAITINT 寄存器位清零），这会导致主机在开漏模式下发送 RCA = 0x0000 的中断服务请求响应 R5。

发送中断服务请求响应时，发送方逐位监视 SDMMC_CMD 位流。如果发送方的中断服务请求响应位与 SDMMC_CMD 线上的位不对应，则将停止发送。当存在多个发送方时，只有一个发送方将成功发送其完整的中断服务请求响应。即，如果主机与其中一个发送方同时发送，则主机将在传输位之后停止发送。

要处理中断模式，请按照以下步骤操作：

1. 根据开漏模式下可实现的 SDMMC_CMD 数据速率设置 SDMMC_CK 频率，CLKDIV 应设置为 > 1，SETCLKRX 应选择 sdmmc_io_in_ck。
2. 在命令寄存器中加载 CMD40 (GO_IRQ_STATE)。
3. 通过将 WAITINT 寄存器位置 1 使能等待中断。
4. 将 CPSM 配置为立即发送命令。
   a) 这将会发送 CMD40，并且使 CPSM 在等待状态下停止，等待中断服务请求响应。
5. 要退出等待中断状态（CPSM 等待状态）：
   a) 检测到中断服务请求响应起始位后，CPSM 将变为接收状态以接收响应。是否成功接收到响应通过 CMDREND 或命令 CRC 错误标志来指示。
   b) 为中止中断模式，主机会将 WAITINT 寄存器位清零。这将导致主机自行发送中断服务请求响应，从而使 CPSM 进入接收状态。是否成功接收到响应通过 CMDREND 或命令 CRC 错误标志来指示。

注： *当同步发送中断服务请求响应起始位发生冲突时，主机将在传输位之后丢失总线访问。*

55.5.5 启动操作

在启动操作模式下，主机可以通过 2 种启动操作功能从卡中读取启动数据：

1. 正常启动。（使 CMD 线保持低电平）
2. 备用启动（发送具有参数 0xFFFFFFFA 的 CMD0）

可以根据以下配置选项读取启动数据，具体取决于卡寄存器设置：

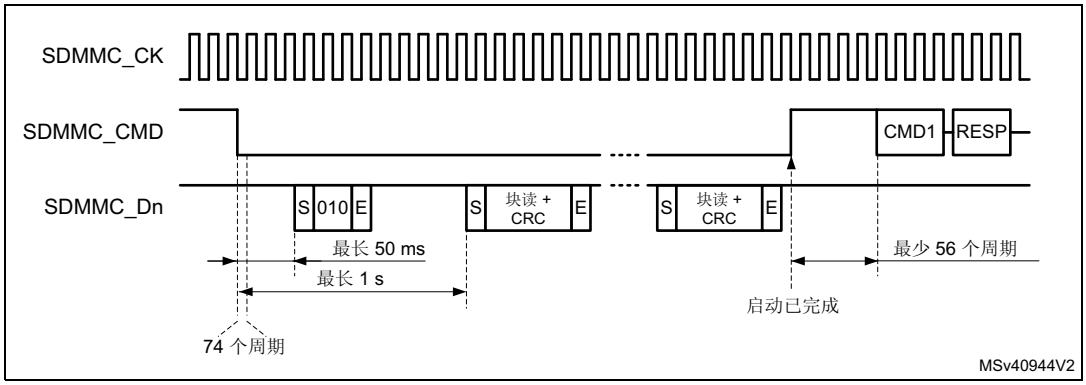
- 从中读取启动数据的分区（EXT_CSD 字节 [179]）
- 启动数据大小（EXT_CSD 字节 [226]）
- 启动期间的总线配置（EXT_CSD 字节 [177]）
- 从卡接收启动确认。（EXT_CSD 字节 [179]）

如果使能启动确认，则卡将在请求启动模式（通过将 CMD 线置为低电平）或发送具有参数 0xFFFFFFFA 的 CMD0 后的 50 ms 内在 SDMMC_D0 上发送模式 010，并将提供启动确认超时 (ACKTIMEOUT) 和确认状态（ACKFAIL）。

正常启动操作

如果在卡上电或复位后 SDMMC_CMD 线的低电平状态持续至少 74 个时钟周期，则卡将在发出第一个命令之前识别出正在发起启动模式。在 CMD 线变为低电平后的 1 秒内，卡开始在 SDMMC_Dn 线上发送第一个启动代码数据。主机必须使 SDMMC_CMD 线保持低电平，直到读取完所有启动数据为止。主机可通过将 SDMMC_CMD 线拉高来终止启动模式。

图 716. 正常启动模式操作



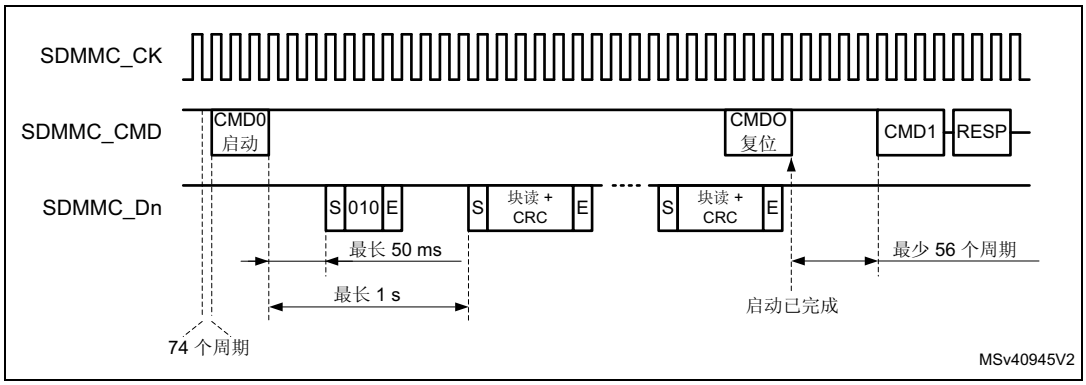
要执行正常的启动过程，需按照以下步骤操作：

1. 将卡复位。
2. 如果请求启动确认，需使能 BOOTACKEN、设置 ACKTIME 并使能 ACKFAIL 和 ACKTIMEOUT 中断。
3. 将 DPSM 设置为接收模式 (DTDIR) 并在 DATALENGTH 中设置要接收的数据字节数，以使能数据接收。
4. 使能 DTIMEOUT、DATAEND 和 CMDSENT 中断，用于完成启动命令确认。
5. 通过 BOOTMODE 选择正常启动操作模式，并通过 BOOTEN 使能启动。启动过程的启动方法是通过 CPSMEN 使能 CPSM。这将导致：
   - SDMMC_CMD 被驱动为低电平。（BOOTMODE = 正常启动）。
   - ACK 超时被启动。
   - DPSM 被使能。
6. 可通过 ACKFAIL 标志或 ACKTIMEOUT 标志（使能时）检测是否接收到不正确的启动确认。
   - 当接收到不正确的启动确认时，将出现 ACKFAIL 标志。
   - 当未及时接收到启动确认时，将出现 ACKTIMEOUT 标志。
7. 当接收到所有启动数据时，将出现 DATAEND 标志。
   - 当数据 CRC 失败时，还会生成 DCRCFAIL 标志。
   - 当数据发生超时时，还会生成 DTIMEOUT 标志。
8. 当接收到最后一个数据时，从 FIFO 读取数据，直到 FIFO 为空 (RXFIFOE = 1)，之后将生成数据结束 DATAEND 标志。
   - SDMMC 已接收完所有数据且 DPSM 处于禁止状态。
9. 通过固件清零 BOOTEN 将终止启动过程，这会导致 SDMMC_CMD 线变为高电平。56 个周期后将生成 CMDSENT 标志，以指示可以发送新命令。
   a) 如果在接收到所有数据之前通过固件中止启动过程，CPSM 中止信号将停止数据接收并禁止 DPSM，此时将生成 DABORT 标志（使能时）。
10. CMDSENT 标志用于指示启动过程结束，卡已准备好接收新命令。

备用启动操作

卡上电或复位后，如果主机在 74 个时钟周期后发送具有参数 0xFFFFFFFA 的 CMD0，则卡将在发出 CMD0 之前识别出正在发起启动模式。在发送具有参数 0xFFFFFFFA 的 CMD0 后的 1 秒内，卡开始在 SDMMC_Dn 线上发送第一个启动代码数据。主机将通过发送 CMD0（复位）来终止启动操作。

图 717. 备用启动模式操作



要执行备用启动过程，需按照以下步骤操作：

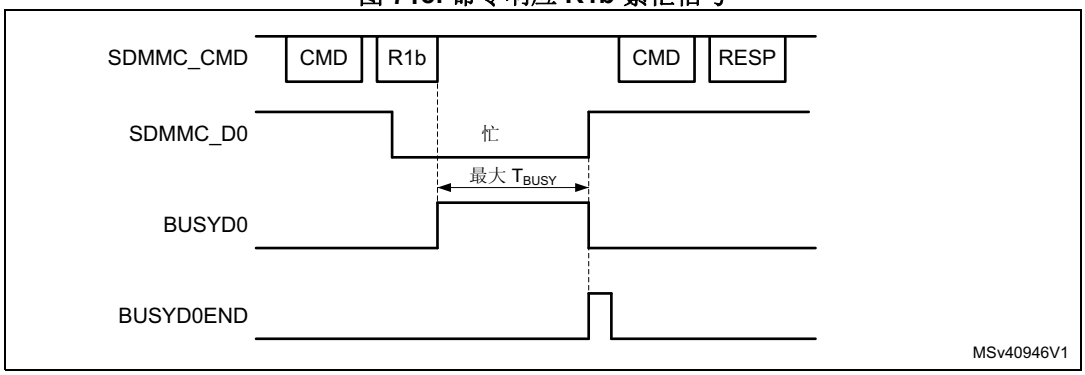
1. 使 SDMMC 变为掉电状态，并将卡复位
2. 使 SDMMC 变为上电状态。这将保证在任何命令之前提供 74 个 SCDMMC_CK 周期的时钟驱动。
3. 如果请求启动确认，需使能 BOOTACKEN、设置 ACKTIME 并使能 ACKTIMEOUT 标志。
4. 将 DPSM 设置为接收模式 (DTDIR) 并在 DATALENGTH 中设置要接收的数据字节数，以使能数据接收。使能 DTIMEOUT 和 DATAEND 标志。
5. 通过 BOOTMODE 选择备用启动操作模式，在命令寄存器中加载具有参数 0xFFFFFFFA 的 CMD0。使能 CMDSENT 标志以指示完成启动命令确认，并通过 BOOTEN 使能启动。启动过程的启动方法是通过 CPSMEN 使能 CPSM。这将导致：
   - 加载的命令和参数被发送出去。（BOOTMODE = 备用启动）。
   - ACK 超时被启动。
   - DPSM 被使能。
6. 发送命令后，将生成 CMDSENT 标志，此时 BOOTEN 位应清零。
7. 可通过 ACKFAIL 标志（使能时）检测是否接收到启动确认。
   - 当未及时接收到启动确认时，将出现 ACKTIMEOUT 标志。
8. 当接收到所有启动数据时，将出现 DATAEND 标志。
   - 当数据 CRC 失败时，还会生成 DCRCFAIL 标志。
   - 当数据发生超时时，还会生成 DTIMEOUT 标志。
9. 当接收到最后一个数据时，从 FIFO 读取数据，直到 FIFO 为空 (RXFIFOE = 1)，之后将生成数据结束 DATAEND 标志。
   - SDMMC 已接收完所有数据且 DPSM 处于禁止状态。

10. 在通过发送 BOOTMODE = 备用启动的 CMD0（复位）以中止启动过程之前，应将 BOOTEN 位清零。这将导致在命令后的 56 个周期后出现 CMDSENT 标志。
    – 如果在接收到所有数据之前通过固件中止启动过程，CPSM 中止信号将停止数据传输并禁止 DPSM，此时将生成 DABORT 标志（使能时）。
11. CMDSENT 标志用于指示启动过程结束，卡已准备好接收新命令。当 RESET 命令成功发送后，必须将 BOOTMODE 控制位清零才能终止启动操作。

55.5.6 响应 R1b 的处理

当发送具有 R1b 响应的命令时，繁忙信号将反映在 BUSYD0 寄存器位上，而繁忙释放信号将通过 BUSYD0END 标志反映。SDMMC_D0 线在 R1b 响应结束时进行采样，在 BUSYD0 寄存器位中进行指示。当 SDMMC_D0 线释放繁忙状态时，BUSYD0 寄存器位将复位为不忙，同时将生成 BUSYD0END 标志。

图 718. 命令响应 R1b 繁忙信号



发送命令之前，应在 DATATIME 寄存器中设置预期的最长繁忙时间。当 R1b 响应繁忙状态保持有效的时间超过编程的时间时，DTIMEOUT 标志（使能时）将置 1。

发送具有 R1b 响应的命令时，如果要检测 SDMMC_D0 繁忙信号，请按照以下步骤操作：

- 使能 CMDREND 标志。
- 通过 CPSM 发送命令。
- 出现 CMDREND 标志时，检查 BUSYD0 寄存器位。
  – 如果 BUSYD0 指示不忙，则向固件发出繁忙释放信号
  – 如果 BUSYD0 指示繁忙，则等待 BUSYD0END 标志
- 出现 BUSYD0END 标志时，向固件发出繁忙已释放信号。
- 出现 DTIMEOUT 标志时，表示繁忙状态保持有效的时间超过编程的时间。

55.5.7 复位和卡掉电再上电

复位

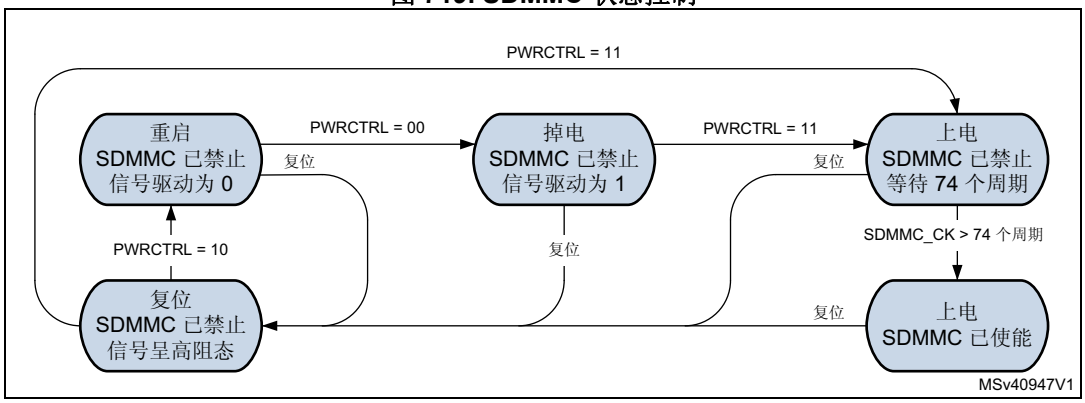
复位后，SDMMC 将处于复位状态。在此状态下，SDMMC 将被禁止，不能传输任何命令和数据。SDMMC_D[7:0] 和 SDMMC_CMD 处于高阻态，SDMMC_CK 驱动为低电平。

SDMMC 应在变为上电状态之前进行配置。

在上电状态下，SDMMC_CK 时钟处于运行状态。首先将提供 74 个 SDMMC_CK 周期的时钟驱动，之后将使能 SDMMC，此时即可传输命令和数据。

SDMMC 状态由固件通过 PWRCTL 寄存器位进行控制，如*表 719* 所示。

图 719. SDMMC 状态控制

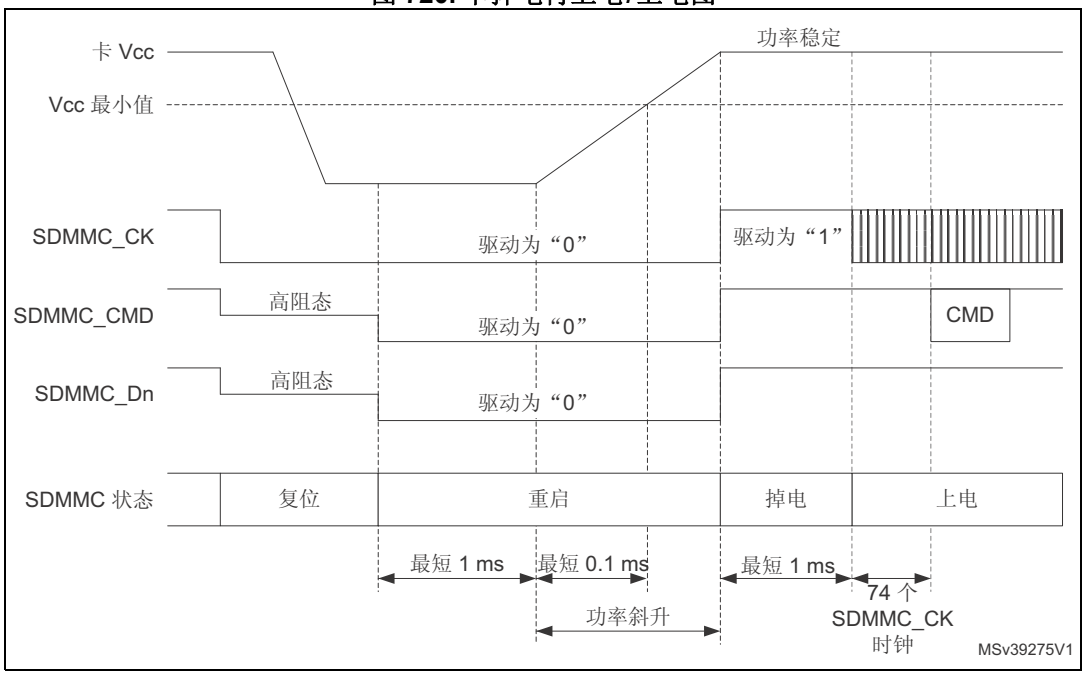


对卡进行掉电再上电

要对卡进行掉电再上电，请按照以下步骤进行操作：

1. 通过 RCC.SDMMCxRST 寄存器位复位 SDMMC。这将使 SDMMC 复位为复位状态，CPSM 和 DPSM 复位为空闲状态。
2. 禁止卡的 Vcc 电源。
3. 将 SDMMC 设置为掉电再上电状态。这将使 SDMMC_D[7:0]、SDMMC_CMD 和 SDMMC_CK 被驱动为低电平，以防止卡通过信号线供电。
4. 至少等待 1 ms 后，才能使能卡的 Vcc 电源。
5. 在电源斜升周期后，将 SDMMC 设置为掉电状态并至少持续 1 ms。SDMMC_D[7:0]、SDMMC_CMD 和 SDMMC_CK 将设置为“1”。
6. 经过 1 ms 的延迟后，将 SDMMC 设置为上电状态，在此期间将使能 SDMMC_CK 时钟。
7. 经过 74 个 SDMMC_CK 周期后，可以向卡发送第一个命令。

图 720. 卡掉电再上电/上电图



55.6 硬件流控制

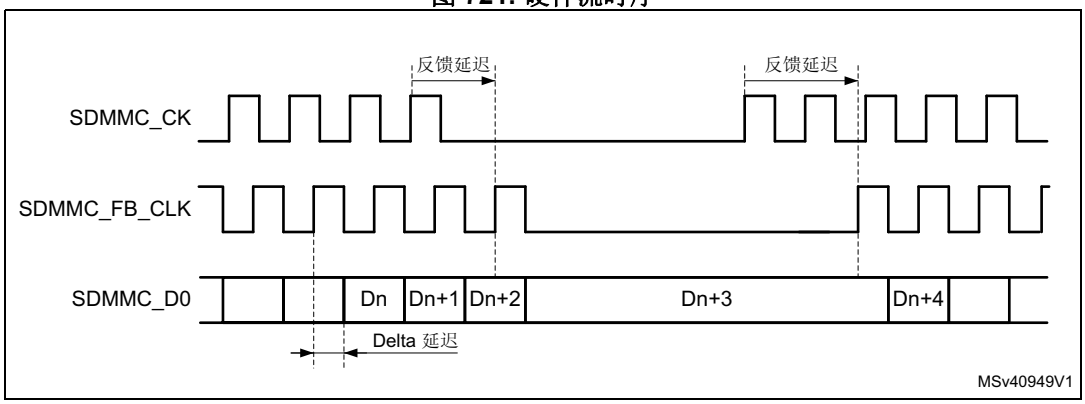
硬件流控制功能用于避免 FIFO 下溢（发送模式）和上溢（接收模式）错误。

具体行为是在数据传输过程中停止 SDMMC_CK 并冻结 SDMMC 状态机。当 FIFO 无法发送或接收数据时，将停止数据传输。数据传输一直保持停止状态，直到发送 FIFO 半满或已存储 DATALENGHT 所指示的所有数据，或者直到接收 FIFO 半空。只有由 SDMMC_CK 提供时钟的状态机才会冻结，AHB 接口仍保持活动状态。因此，即使激活流控制，仍可填充或清空 FIFO。

要使能硬件流控制，HWFC_EN 寄存器位必须置 1。复位后，硬件流控制将禁止。

仅当 SDMMC_Dn 数据与 SDMMC_CK 循环对齐时，才应使用硬件流控制。只要使用来自 DLYB 延迟模块的 sdmmc_fb_ck（即在 t_{OP} 和 Dt_{OP} 延迟 > 1 个周期的 SDR104 模式下），便不能使用硬件流控制。

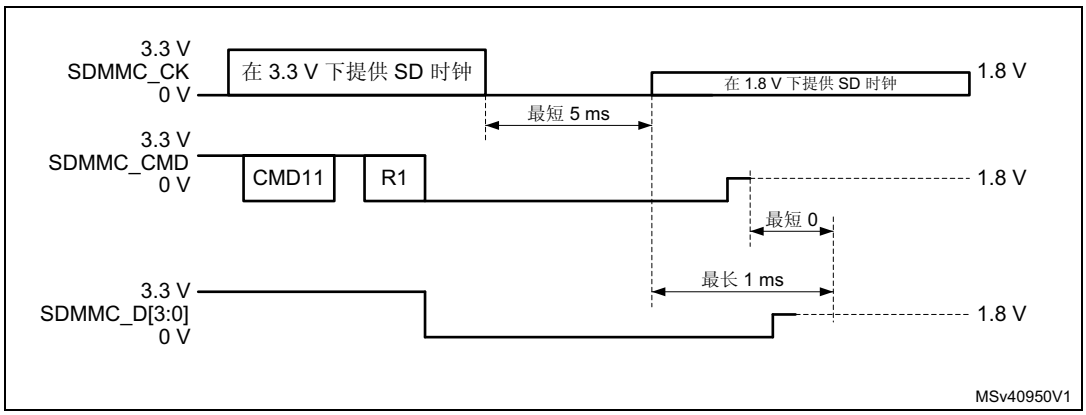
图 721. 硬件流时序



55.7 超高速 I 相 (UHS-I) 电压切换

UHS-I 模式（SDR12、SDR25、SDR50、SDR104 和 DDR50）需要 1.8V 信号支持。上电后，卡将以 3.3V 模式启动。CMD11 将调用电压切换序列以切换到 1.8V 模式。当电压序列成功完成时，卡将以默认 SDR12 进入 UHS-I 模式，卡输入和输出时序将发生变化。

图 722. CMD11 信号电压切换序列



要执行信号电压切换序列，需按以下步骤进行操作：

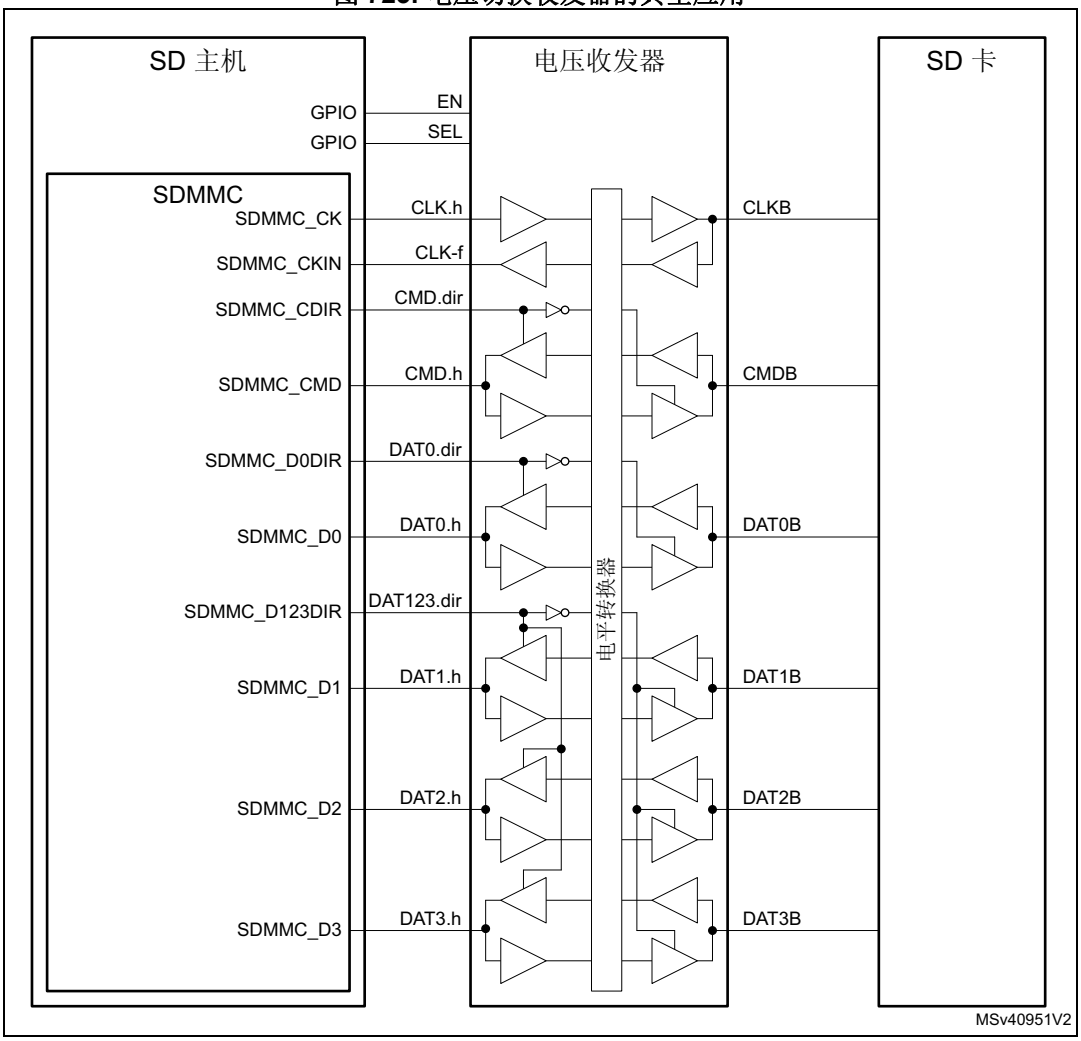
1. 在启动电压切换过程之前，应将 SDMMC_CK 频率设置在 100 kHz - 400 kHz 范围内。
2. 主机通过在发送 CMD11 之前将 VSWITCHEN 位置 1 来启动电压切换过程。
3. 卡返回 R1 响应。
   - 如果响应 CRC 通过，电压切换过程将继续执行，主机将不再驱动 CMD 和 SDMMC_D[3:0] 信号，直到电压切换序列完成。在响应后再经过若干周期，SDMMC_CK 将停止，CKSTOP 标志将置 1。
   - 如果响应 CRC 失败（CCRCFAIL 标志）或未在超时（CTIMEOUT 标志）之前收到响应，则电压切换过程将停止。
4. 在 R1 响应之后的下一个时钟，卡将 CMD 和 SDMMC_D[3:0] 驱动为低电平。
5. 收到 R1 响应后，主机可以使用 BUSYD0 寄存器位监视 SDMMC_D0 线。SDMMC_D0 线在响应后的两个 SDMMC_CK 时钟周期后进行采样。固件可以读取 CKSTOP 标志后的 BUSYD0 寄存器位。
   - 当检测到 BUSYD0 为低电平时，主机固件会将稳压器切换到 1.8V，然后通过将寄存器位 VSWITCH 置 1 来指示 SDMMC 启动电压切换序列的时序关键部分。硬件将继续停止 SDMMC_CK，方法是使 SDMMC_CK 保持低电平并持续至少 5 ms。
   - 当检测到 BUSYD0 为高电平时，主机将中止电压切换序列并对卡进行掉电再上电。
6. 检测到 SDMMC_CK 为低电平后，卡会开始将信号电压切换到 1.8V。
7. 主机 SDMMC 硬件将在至少 5 ms 后重新启动 SDMMC_CK。
8. 在检测到 SDMMC_CK 切换后的 1 ms 内，卡会将 CMD 和 DAT[3:0] 驱动为高电平并至少持续 1 个 SDMMC_CK 周期，然后停止驱动 CMD 和 DAT[3:0]。
9. 在 SDMMC_CK 重启后的 1 ms 后，主机 SDMMC 硬件会将 SDMMC_D0 采样到 BUSYD0 中并生成 VSWEND 标志。
10. 出现 VSWEND 标志时，主机将使用 BUSYD0 寄存器位检查 SDMMC_D0 线，以确认电压切换序列是否完成：
    - 当检测到 BUSYD0 为高电平时，表示电压切换成功完成。
    - 当检测到 BUSYD0 为低电平时，表示电压切换失败，主机将对卡执行掉电再上电。

停止 SDMMC_CK 所需的 5 ms 最短时间来自内部非门控 SDMMC_CK 时钟，其最大频率为 25 MHz（SD 模式），该频率由时钟分频比 CLKDIV 设置。> 5 ms 的时间将以 2^12 个周期（400 kHz 时为 10.24 ms）计数。如果通过时钟分频比 CLKDIV 选择较低的 SDMMC_CK 频率，则停止 SDMMC_CK 时钟所需的时间将会更长。

卡将 SDMMC_Dn 和 SDMMC_CMD 线驱动为高电平所需的 1 ms 最长时间来自内部非门控 SDMMC_CK 时钟，其最大频率为 25 MHz（SD 模式），该频率由时钟分频比 CLKDIV 设置。SDMMC 将在 > 1 ms 的时间后检查线路，该时间将以 2^9 个周期（25 MHz 时为 1.28 ms）计数。如果通过时钟分频比 CLKDIV 选择较低的 SDMMC_CK 频率，则检查线路所需的时间将会更长。

信号电压通过外部电压切换收发器（即 ST6G3244ME）来提供支持。

图 723. 电压切换收发器的典型应用



为了连接标准信号旁的外部驱动器（电压切换收发器），SDMMC 使用以下信号：

SDMMC_CKIN 反馈输入时钟

SDMMC_CDIR CMD 的 I/O 方向控制信号。

SDMMC_D0DIR SDMMC_D0 的 I/O 方向控制信号。

SDMMC_D123DIR SDMMC_D1、SDMMC_D2 和 SDMMC_D3 的 I/O 方向控制信号。

电压收发器信号 **EN** 和 **SEL** 将通过通用 I/O 进行处理。

SDMMC_CDIR、SDMMC_D0DIR 和 SDMMC_D123DIR 信号的极性可通过 SDMMC_POWER.DIRPOL 控制位进行选择。

55.8 SDMMC 寄存器

器件通过 32 位控制寄存器（可通过 AHB 从接口访问）与系统进行通信。

外设寄存器必须按字（32 位）进行访问。按字节（8 位）和半字（16 位）访问会生成 AHB 总线错误。

55.8.1 SDMMC 电源控制寄存器 (SDMMC_POWER)

SDMMC power control register

偏移地址：0x000

复位值：0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DIR POL	V SWITCH EN	V SWITCH	PWRCTRL[1:0]	
											rw	rw	rw	rw	rw

位 31:5 保留，必须保持复位值。

位 4 **DIRPOL**：数据和命令方向信号极性选择 (Data and command direction signals polarity selection)。
只有在 SDMMC 处于掉电状态时 (PWRCTRL = 00) 才能写入该位。
0：当方向信号为低电平时，电压收发器 IO 驱动为输出。
1：当方向信号为高电平时，电压收发器 IO 驱动为输出。

位 3 **VSWITCHEN**：电压切换过程使能 (Voltage switch procedure enable)。
只有在禁止 CPSM (CPSMEN=0) 时才能通过固件写入此位。
该位用于在电压切换命令响应后停止 SDMMC_CK：
0：SDMMC_CK 时钟在成功接收到命令响应后保持不变。
1：SDMMC_CK 时钟在成功接收到命令响应后停止。

位 2 **VSWITCH**：电压切换序列启动 (Voltage switch sequence start)。
该位用于启动电压切换序列的时序关键部分：
0：电压切换序列未启动且未激活。
1：电压切换序列已启动或已激活。

位 1:0 **PWRCTRL[1:0]**：SDMMC 状态控制位 (SDMMC state control bits)。
只有在 SDMMC 未处于上电状态时 (PWRCTRL ≠ 11) 才能写入这些位。
这些位用于定义 SDMMC 信号的功能状态：
00：复位后，复位：SDMMC 禁止，卡的时钟停止，SDMMC_D[7:0] 和 SDMMC_CMD 处于高阻态，SDMMC_CK 驱动为低电平。
写入 00 时，掉电：SDMMC 禁止，卡的时钟停止，SDMMC_D[7:0]、SDMMC_CMD 和 SDMMC_CK 驱动为高电平。
01：保留。（写入 01 时，PWRCTRL 值不会改变）
10：掉电再上电，SDMMC 禁止，卡的时钟停止，SDMMC_D[7:0]、SDMMC_CMD 和 SDMMC_CK 驱动为低电平。
11：上电：卡由时钟驱动，前 74 个 SDMMC_CK 周期内，SDMMC 仍禁止。74 个周期后，使能 SDMMC，并根据 SDMMC 操作控制 SDMMC_D[7:0]、SDMMC_CMD 和 SDMMC_CK。
任何后续写操作都将被忽略，PWRCTRL 值将保持为 11。

55.8.2 SDMMC 时钟控制寄存器 (SDMMC_CLKCR)

SDMMC clock control register

偏移地址：0x004

复位值：0x0000 0000

SDMMC_CLKCR 寄存器控制 SDMMC_CK 输出时钟、sdmmc_rx_ck 接收时钟和总线宽度。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SELCLKRX[1:0]		BUS SPEED	DDR	HWFC_ EN	NEG EDGE
										rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WID BUS[1:0]		Res.	PWR SAV	Res.	Res.	CLKDIV[9:0]									
rw	rw		rw			rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:22 保留，必须保持复位值。

位 21:20 **SELCLKRX**：接收时钟选择 (Receive clock selection)。

只有在 CPSM 和 DPSM 未激活时（CPSMACT = 0 且 DPSMACT = 0），才能写入这些位

00：选择 sdmmc_io_in_ck 作为接收时钟

01：选择 SDMMC_CKIN 反馈时钟作为接收时钟

10：选择 sdmmc_fb_ck 调节反馈时钟作为接收时钟

11：保留（选择 sdmmc_io_in_ck）

位 19 **BUSSPEED**：在 DS、HS、SDR12、SDR25 以及 SDR50、DDR50、SDR104 总线速度模式间进行选择 (Bus speed mode selection between DS, HS, SDR12, SDR25 and SDR50, DDR50, SDR104)。

只有在 CPSM 和 DPSM 未激活时（CPSMACT = 0 且 DPSMACT = 0），才能写入该位

0：选择 DS、HS、SDR12、SDR25 总线速度模式

1：选择 SDR50、DDR50、SDR104 总线速度模式

位 18 **DDR**：数据速率信号传输选择 (Data rate signaling selection)

只有在 CPSM 和 DPSM 未激活时（CPSMACT = 0 且 DPSMACT = 0），才能写入该位

只能在 4 位或 8 位宽的总线模式下选择 DDR 速率。(WIDBUS > 00)。当 WIDBUS = 00（1 位宽总线）时，DDR = 1 不起作用。

只能在时钟分频比 > 1 的情况下选择 DDR 速率。(CLKDIV > 0)

0：SDR 单倍数据速率信号传输。

1：DDR 双倍数据速率信号传输。

位 17 **HWFC_EN**：硬件流控制使能 (Hardware flow control enable)

只有在 CPSM 和 DPSM 未激活时（CPSMACT = 0 且 DPSMACT = 0），才能写入该位

0：禁止硬件流控制

1：使能硬件流控制

如果使能硬件流控制，TXFIFOE 和 RXFIFOF 标志的含义将发生变化，请参见*第 55.8.11 节*中的 SDMMC 状态寄存器定义。

位 16 **NEGEDGE**：数据和命令的 SDMMC_CK 移相选择位 (SDMMC_CK dephasing selection bit for data and Command)。

只有在 CPSM 和 DPSM 未激活时（CPSMACT = 0 且 DPSMACT = 0），才能写入该位。

当时钟分频比 = 1 (CLKDIV = 0) 时，该位不起作用。数据和命令在 SDMMC_CK 下降沿发生变化。

当时钟分频比 > 1 (CLKDIV > 0) 且 DDR = 0 时：

0：- 命令和数据在 SDMMC_CK 上升沿后的 sdmmc_ker_ck 下降沿发生变化。
- SDMMC_CK 边沿出现在 sdmmc_ker_ck 上升沿。

1：- 命令和数据在生成 SDMMC_CK 下降沿的同一 sdmmc_ker_ck 上升沿发生变化。

当时钟分频比 > 1 (CLKDIV > 0) 且 DDR = 1 时：

0：- 命令在 SDMMC_CK 上升沿后的 sdmmc_ker_ck 下降沿发生变化。
- 数据在 SDMMC_CK 边沿后的 sdmmc_ker_ck 下降沿发生变化。
- SDMMC_CK 边沿出现在 sdmmc_ker_ck 上升沿。

1：- 命令在生成 SDMMC_CK 下降沿的同一 sdmmc_ker_ck 上升沿发生变化。
- 数据在 SDMMC_CK 边沿后的 SDMMC_CK 下降沿发生变化。
- SDMMC_CK 边沿出现在 sdmmc_ker_ck 上升沿。

位 15:14 **WIDBUS[1:0]**：宽总线模式使能位 (Wide bus mode enable bit)

只有在 CPSM 和 DPSM 未激活时（CPSMACT = 0 且 DPSMACT = 0），才能写入该位

00：默认 1 位宽总线模式：使用 SDMMC_D0（不支持 DDR）

01：4 位宽总线模式：使用 SDMMC_D[3:0]

10：8 位宽总线模式：使用 SDMMC_D[7:0]

位 13 保留，必须保持复位值。

位 12 **PWRSAV**：节能模式配置位 (Power saving configuration bit)

只有在 CPSM 和 DPSM 未激活时（CPSMACT = 0 且 DPSMACT = 0），才能写入该位

要实现节能模式，可在总线空闲时通过将 PWRSAV 置 1 来禁止 SDMMC_CK 时钟输出：

0：始终使能 SDMMC_CK 时钟

1：仅在总线激活时使能 SDMMC_CK

位 11:10 保留，必须保持复位值。

位 9:0 **CLKDIV[9:0]**：时钟分频系数 (Clock divide factor)

只有在 CPSM 和 DPSM 未激活时（CPSMACT = 0 且 DPSMACT = 0），才能写入该位。

该位域定义输入时钟 (sdmmc_ker_ck) 与输出时钟 (SDMMC_CK) 之间的分频系数：SDMMC_CK 频率 = sdmmc_ker_ck / [2 * CLKDIV]。

000：SDMMC_CK 频率 = sdmmc_ker_ck / 1（不支持 DDR）

001：SDMMC_CK 频率 = sdmmc_ker_ck / 2

002：SDMMC_CK 频率 = sdmmc_ker_ck / 4

0xx：...

080：SDMMC_CK 频率 = sdmmc_ker_ck / 256

xxx：...

3FF：SDMMC_CK 频率 = sdmmc_ker_ck / 2046

注： *1 当 SD/SDIO 卡或 MMC 处于识别模式时，SDMMC_CK 频率必须小于 400 kHz。*

2 如果为所有卡分配相对卡地址，则可以将时钟频率更改为卡总线最大频率。

3 此寄存器的两次写访问至少需要相隔七个 sdmmc_hclk 时钟周期。在 SD I/O 卡的读取等待间隔期间也可以停止 SDMMC_CK：在此情况下，SDMMC_CLKCR 寄存器不对 SDMMC_CK 进行控制。

55.8.3 SDMMC 参数寄存器 (SDMMC_ARGR)

SDMMC argument register

偏移地址：0x008

复位值：0x0000 0000

SDMMC_ARGR 寄存器包含一个 32 位命令参数，该参数作为命令消息的一部分发送到卡。

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

位 31:0 **CMDARG[31:0]**：命令参数 (Command argument)。

只有在禁止 CPSM (CPSMEN=0) 时才能通过固件写入这些位。

作为命令消息的一部分发送给卡的命令参数。如果命令包含参数，则在将命令写入到命令寄存器之前，必须将参数加载到此寄存器中。

55.8.4 SDMMC 命令寄存器 (SDMMC_CMDR)

SDMMC command register

偏移地址：0x00C

复位值：0x0000 0000

SDMMC_CMDR 寄存器包含命令索引和命令类型位。命令索引作为命令消息的一部分而发送给卡。命令类型位控制命令路径状态机 (CPSM)。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CMD SUS PEND
															rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BOOT EN	BOOT MODE	DT HOLD	CPSM EN	WAITP END	WAIT INT	WAITRESP[1:0]		CMD STOP	CMD TRANS	CMDINDEX[5:0]					
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:17 保留，必须保持复位值。

位 16 **CMDSUSPEND**：CPSM 将命令视为挂起或恢复命令并发出中断周期开始/结束信号 (The CPSM treats the command as a Suspend or Resume command and signals interrupt period start/end)。

只有在禁止 CPSM (CPSMEN=0) 时才能通过固件写入此位。

CMDSUSPEND = 1 且 CMDTRANS = 0 时为挂起命令，响应位 BS = 0 时开始中断周期。

CMDSUSPEND = 1 且 CMDTRANS = 1 时为含数据的恢复命令，响应位 DF = 1 时结束中断周期。

位 15 **BOOTEN**：使能启动模式过程 (Enable boot mode procedure)。

0：禁止启动模式过程

1：使能启动模式过程

位 14 **BOOTMODE**：选择要使用的启动模式过程 (Select the boot mode procedure to be used)。
只有在禁止 CPSM (CPSMEN = 0) 时才能写入该位
0：选择正常启动模式过程
1：选择备用启动模式过程

位 13 **DTHOLD**：在 DPSM 中保持新的数据块发送和接收 (Hold new data block transmission and reception in the DPSM)。
如果该位置 1，DPSM 将不会从 Wait_S 状态变为发送状态或从 Wait_R 状态变为接收状态。

位 12 **CPSMEN**：命令路径状态机 (CPSM) 使能位 (Command path state machine (CPSM) Enable bit)
该位由固件写 1，并在 CPSM 进入空闲状态时由硬件清零。
如果此位置 1，则使能 CPSM。
当 DTEN = 1 时，不会传输命令，也不会开始启动过程。CPSMEN 将清零。
在读取等待期间，当 SDMMC_CK 停止时，不会发送任何命令，CPSMEN 保持为 0。

位 11 **WAITPEND**：CPSM 等待来自 DPSM 的数据传输结束（CmdPend 内部信号）(CPSM Waits for end of data transfer (CmdPend internal signal) from DPSM)。
如果此位置 1，则 CPSM 将等到数据传输结束触发信号后才开始发送命令。
仅当 DTMODE = MMC 流数据传输、WIDBUS = 1 位宽总线模式、DPSMACT = 1 且 DTDIR = 从主机到卡时，才考虑 WAITPEND。

位 10 **WAITINT**：CPSM 等待中断请求 (CPSM waits for interrupt request)。
如果此位置 1，则 CPSM 禁止命令超时并等待卡中断请求（响应）。
如果该位在 CPSM 等待状态下清零，将导致中断模式中止。

位 9:8 **WAITRESP[1:0]**：等待响应位 (Wait for response bits)。
只有在禁止 CPSM (CPSMEN=0) 时才能通过固件写入此位。
这些位用于配置 CPSM 是否等待响应，如果等待，将等待哪种类型的响应。
00：无响应，将出现 CMDSENT 标志
01：短响应，将出现 CMDREND 或 CCRCFAIL 标志
10：短响应，将出现 CMDREND 标志（无 CRC）
11：长响应，将出现 CMDREND 或 CCRCFAIL 标志

位 7 **CMDSTOP**：CPSM 将命令视为停止传输命令并向 DPSM 发出中止信号 (The CPSM treats the command as a Stop Transmission command and signals Abort to the DPSM)。
只有在禁止 CPSM (CPSMEN=0) 时才能通过固件写入此位。
如果该位置 1，当发送命令时，CPSM 将向 DPSM 发出中止信号。

位 6 **CMDTRANS**：CPSM 将命令视为数据传输命令、停止中断周期并向 DPSM 发出数据使能信号 (The CPSM treats the command as a data transfer command, stops the interrupt period, and signals DataEnable to the DPSM)
只有在禁止 CPSM (CPSMEN=0) 时才能通过固件写入此位。
如果该位置 1，当发送命令时，CPSM 将发出中断周期结束信号并向 DPSM 发出数据使能信号。

位 5:0 **CMDINDEX[5:0]**：命令索引 (Command index)。
只有在禁止 CPSM (CPSMEN=0) 时才能通过固件写入此位。
命令索引作为命令消息的一部分发送给卡。

注： *1 此寄存器的两次写访问至少需要相隔七个 sdmmc_hclk 时钟周期。*

2 多媒体卡可以发送两种类型的响应：短响应（48 位）或长响应（136 位）。SD 卡和 SD I/O 卡则只能发送短响应，参数因响应类型而异：软件将根据发送的命令区分响应类型。

55.8.5 SDMMC 命令响应寄存器 (SDMMC_RESPCMDR)

SDMMC command response register

偏移地址：0x010

复位值：0x0000 0000

SDMMC_RESPCMDR 寄存器包含接收到的最后一个命令响应的命令索引位域。如果命令响应传输不包含命令索引位域（长响应或 OCR 响应），则 RESPCMD 位域为未知，但该位域必须包含 111111b（响应中保留位域的值）。

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

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

位 31:6 保留，必须保持复位值。

位 5:0 **RESPCMD[5:0]**：响应命令索引 (Response command index)
只读位域。包含接收到的最后一个命令响应的命令索引。

55.8.6 SDMMC 响应 1..4 寄存器 (SDMMC_RESPxR) (x = 1..4)

SDMMC response 1..4 register

偏移地址：(0x010 + (4 × x))

复位值：0x0000 0000

SDMMC_RESP1/2/3/4R 寄存器包含卡的状态，该状态来自接收的响应。

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

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

位 31:0 **CARDSTATUSx[31:0]**：请参见*表 447*。

卡状态为 32 位或 128 位，具体取决于响应类型。

表 447. 响应类型和 SDMMC_RESPxR 寄存器

寄存器(1)	短响应	长响应
SDMMC_RESP1R	卡状态 [31:0]	卡状态 [127:96]
SDMMC_RESP2R	全 0	卡状态 [95:64]
SDMMC_RESP3R	全 0	卡状态 [63:32]
SDMMC_RESP4R	全 0	卡状态 [31:0](2)

1. 首先接收卡状态的最高有效位。
2. SDMMC_RESP4R 寄存器 LSB 始终为 0。

55.8.7 SDMMC 数据定时器寄存器 (SDMMC_DTIMER)

SDMMC data timer register

偏移地址：0x024

复位值：0x0000 0000

SDMMC_DTIMER 寄存器包含以卡总线时钟周期表示的数据超时周期。

计数器加载 SDMMC_DTIMER 寄存器的值，并在数据路径状态机进入 Wait_R 或繁忙状态后开始递减。如果定时器达到 0 而 DPSM 处于上述一种状态，则超时状态标志置 1。

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

位 31:0 **DATATIME[31:0]**：数据和 R1b 繁忙超时周期 (Data and R1b busy timeout period)
只有在 CPSM 和 DPSM 未激活时（CPSMACT = 0 且 DPSMACT = 0），才能写入该位。
以卡总线时钟周期表示的数据和 R1b 繁忙超时周期。

注： *数据传输必须首先写入到数据计时器寄存器和数据长度寄存器，然后才写入到数据控制寄存器。*

55.8.8 SDMMC 数据长度寄存器 (SDMMC_DLENR)

SDMMC data length register

偏移地址：0x028

复位值：0x0000 0000

SDMMC_DLENR 寄存器包含要传输的数据字节数。当数据传输开始时，值将加载到数据计数器中。

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

位 31:25 保留，必须保持复位值。

位 24:0 **DATALENGTH[24:0]**：数据长度值 (Data length value)

只有在未激活 DPSM (DPSMACT = 0) 时才能通过固件写入此寄存器。

要传输的数据字节数量。

当 DDR = 1 时，DATALENGTH 被截断为 2 的倍数。（不传输最后一个奇数字节）

当 DATALENGTH = 0 时，不会传输任何数据，当通过 CPSMEN 和 CMDTRANS = 1 请求时，也不会传输任何命令。DTEN 和 CPSMEN 均清零。

注： *对于块数据传输，数据长度寄存器中的值必须是块大小的倍数（请参见 SDMMC_DCTRL）。数据传输必须首先写入到数据计时器寄存器和数据长度寄存器，然后才写入到数据控制寄存器。*

对于 SDMMC 多字节传输，数据长度寄存器中的值必须在 1 到 512 之间。

55.8.9 SDMMC 数据控制寄存器 (SDMMC_DCTRL)

SDMMC data control register

偏移地址：0x02C

复位值：0x0000 0000

SDMMC_DCTRL 寄存器控制数据路径状态机 (DPSM)。

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	FIFO RST	BOOT ACKEN	SDIO EN	RW MOD	RW STOP	RW START	DBLOCKSIZE[3:0]				DTMODE		DTDIR	DTEN
		rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:14 保留，必须保持复位值。

位 13 **FIFORST**：FIFO 复位，将清空任何剩余的数据 (FIFO reset, will flush any remaining data)。
只有在 IDMAEN = 0 且未激活 DPSM (DPSMACT = 1) 时才能通过固件写入该位。只有在发生传输错误或传输保持时，该位才会生效。
0：FIFO 不受影响。
1：清空任何剩余的数据并复位 FIFO 指针。当 DPSM 变为非激活状态 (DPSMACT = 0) 时，该位由硬件自动清零。

位 12 **BOOTACKEN**：使能启动确认接收 (Enable the reception of the boot acknowledgment)。
只有在未激活 DPSM (DPSMACT = 0) 时才能通过固件写入此位。
0：禁止启动确认，将不会接收
1：使能启动确认，将会接收

位 11 **SDIOEN**：SD I/O 中断使能功能 (SD I/O interrupt enable functions)
只有在未激活 DPSM (DPSMACT = 0) 时才能通过固件写入此位。
如果将该位置 1，则 DPSM 使能特定于 SD I/O 卡的中断操作。

位 10 **RWMOD**：读取等待模式 (Read wait mode)。
只有在未激活 DPSM (DPSMACT = 0) 时才能通过固件写入此位。
0：使用 SDMMC_D2 进行读取等待控制
1：通过停止 SDMMC_CK 进行读取等待控制

位 9 **RWSTOP**：读取等待停止 (Read wait stop)
该位由固件写入，并在 DPSM 从 READ_WAIT 状态变为 WAIT_R 或空闲状态时由硬件自动清零。
0：无读取等待停止。
1：当 DPSM 处于 READ_WAIT 状态时，使能读取等待停止。

位 8 **RWSTART**：读取等待开始 (Read wait start)。
如果将该位置 1，则读取等待操作开始。

位 7:4 **DBLOCKSIZE[3:0]**：数据块大小 (Data block size)

只有在未激活 DPSM (DPSMACT = 0) 时才能通过固件写入此位。

定义在选择了块数据传输模式时数据块的长度：

0000：（十进制数 0）块长度 = 2^0 = 1 字节

0001：（十进制数 1）块长度 = 2^1 = 2 字节

0010：（十进制数 2）块长度 = 2^2 = 4 字节

0011：（十进制数 3）块长度 = 2^3 = 8 字节

0100：（十进制数 4）块长度 = 2^4 = 16 字节

0101：（十进制数 5）块长度 = 2^5 = 32 字节

0110：（十进制数 6）块长度 = 2^6 = 64 字节

0111：（十进制数 7）块长度 = 2^7 = 128 字节

1000：（十进制数 8）块长度 = 2^8 = 256 字节

1001：（十进制数 9）块长度 = 2^9 = 512 字节

1010：（十进制数 10）块长度 = 2^{10} = 1024 字节

1011：（十进制数 11）块长度 = 2^{11} = 2048 字节

1100：（十进制数 12）块长度 = 2^{12} = 4096 字节

1101：（十进制数 13）块长度 = 2^{13} = 8192 字节

1110：（十进制数 14）块长度 = 2^{14} = 16384 字节

1111：（十进制数 15）保留

当 DATALENGTH 不是 DBLOCKSIZE 的倍数时，传输的数据将截断为 DBLOCKSIZE 的倍数。（将不会传输任何剩余数据。）

当 DDR = 1 时，不应使用 DBLOCKSIZE = 0000。（将不会传输任何数据）

位 3:2 **DTMODE**：数据传输模式选择 (Data transfer mode selection)。

只有在未激活 DPSM (DPSMACT = 0) 时才能通过固件写入此位。

00：按块数结束的块数据传输。

01：SDIO 多字节数据传输。

10：MMC 流数据传输。（WIDBUS 应选择 1 位宽总线模式）

11：以 STOP_TRANSMISSION 命令结束的块数据传输（不与 DTEN 启动的数据传输一起使用）。

位 1 **DTDIR**：数据传输方向选择 (Data transfer direction selection)

只有在未激活 DPSM (DPSMACT = 0) 时才能通过固件写入此位。

0：从主机到卡。

1：从卡到主机。

位 0 **DTEN**：数据传输使能位 (Data transfer enable bit)

只有在未激活 DPSM (DPSMACT = 0) 时才能通过固件写入此位。当数据传输完成时，该位由硬件清零。

该位仅在不使用相关数据传输命令时才用于传输数据，即不能与 SD 或 eMMC 卡一起使用。

0：不使用 CPSM 数据传输命令时不启动数据传输。

1：不使用 CPSM 数据传输命令时启动数据传输。

55.8.10 SDMMC 数据计数器寄存器 (SDMMC_DCNTR)

SDMMC data counter register

偏移地址：0x030

复位值：0x0000 0000

当 DPSM 从空闲状态变为 Wait_R 或 Wait_S 状态时，SDMMC_DCNTR 寄存器将从数据长度寄存器中加载值（请参见 SDMMC_DLENR）。在传输数据时，计数器将值递减直至计数器达到 0。然后 DPSM 将变为空闲状态，并且在无错误时将数据状态结束标志 (DATAEND) 置 1。

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

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

位 31:25 保留，必须保持复位值。

位 24:0 **DATACOUNT[24:0]**：数据计数值 (Data count value)

读取时，将返回要传输的剩余数据字节数。写入没有任何效果。

注： *仅当数据传输完成或保持后才应读取该寄存器。在错误事件后读取时，读取的数据计数值可能与传输的实际数据字节数不同。*

55.8.11 SDMMC 状态寄存器 (SDMMC_STAR)

SDMMC status register

偏移地址：0x034

复位值：0x0000 0000

SDMMC_STAR 寄存器是一个只读寄存器。它包含两种类型的标志：

- 静态标志（位 [29,21,11:0]）：在通过写入到 SDMMC 中断清零寄存器来清零这些位之前，会一直保持这些位（请参见 SDMMC_ICR）。
- 动态标志（位 [20:12]）：这些位根据底层逻辑的状态来更改状态（例如，随着数据写入到 FIFO，发出和停止发出 FIFO 满和空标志）

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	IDMA BTC	IDMA TE	CK STOP	VSW END	ACK TIME OUT	ACK FAIL	SDIOIT	BUSY D0END	BUSY D0	RX FIFOE	TX FIFOE	RX FIFOF	TX FIFOF
			r	r	r	r	r	r	r	r	r	r	r	r	r

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RX FIFO HF	TX FIFO HE	CPSM ACT	DPSM ACT	D ABORT	DBCK END	DHOLD	DATA END	CMD SENT	CMDR END	RX OVERR	TX UNDER R	D TIME OUT	C TIME OUT	DCRC FAIL	CCRC FAIL
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:29 保留，必须保持复位值。

位 28 **IDMABTC**：IDMA 缓冲区传输完成 (IDMA buffer transfer complete)。通过写入 SDMMC_ICR 中的相应中断清除位来清除中断标志。

位 27 **IDMATE**：IDMA 传输错误 (IDMA transfer error)。通过写入 SDMMC_ICR 中的相应中断清除位来清除中断标志。

位 26 **CKSTOP**：SDMMC_CK 在电压切换过程中停止 (SDMMC_CK stopped in Voltage switch procedure)。通过写入 SDMMC_ICR 中的相应中断清除位来清除中断标志。

位 25 **VSWEND**：电压切换时序关键部分完成 (Voltage switch critical timing section completion)。通过写入 SDMMC_ICR 中的相应中断清除位来清除中断标志。

位 24 **ACKTIMEOUT**：启动确认超时 (Boot acknowledgment timeout)。通过写入 SDMMC_ICR 中的相应中断清除位来清除中断标志。

位 23 **ACKFAIL**：接收到启动确认（启动确认校验失败）(Boot acknowledgment received (boot acknowledgment check fail))。通过写入 SDMMC_ICR 中的相应中断清除位来清除中断标志。

位 22 **SDIOIT**：收到了 SDIO 中断 (SDIO interrupt received) 通过写入 SDMMC_ICR 中的相应中断清除位来清除中断标志。

位 21 **BUSYD0END**：检测到 CMD 响应后 SDMMC_D0 繁忙状态结束 (end of SDMMC_D0 Busy following a CMD response detected)。

该位仅指示 CMD 响应后繁忙状态结束，而不会指示因数据传输引起的繁忙状态。通过写入 SDMMC_ICR 中的相应中断清除位来清除中断标志。

0：卡 SDMMC_D0 信号未从繁忙状态变为不忙状态。

1：卡 SDMMC_D0 信号从繁忙状态变为不忙状态。

位 20 **BUSYD0**：SDMMC_D0 线的反转值（繁忙）(Inverted value of SDMMC_D0 line (Busy))，在 CMD 响应结束时进行采样，并在 CMD 响应后的 2 个 SDMMC_CK 周期后进行二次采样。

当 SDMMCD0 线从繁忙状态变为不忙状态时，该位复位为不忙状态。该位不会指示因数据传输引起的繁忙状态。它只是一个硬件状态标志，不会生成中断。

0：卡在 SDMMC_D0 上发出不忙信号。

1：卡在 SDMMC_D0 上发出繁忙信号。

位 19 **RXFIFOE**：接收 FIFO 为空 (Receive FIFO empty)

它只是一个硬件状态标志，不会生成中断。该位在一个 FIFO 单元已满时清零。

位 18 **TXFIFOE**：发送 FIFO 为空 (Transmit FIFO empty)

该位在一个 FIFO 单元已满时清零。

位 17 **RXFIFOF**：接收 FIFO 已满 (Receive FIFO full)

该位在一个 FIFO 单元为空时清零。

位 16 **TXFIFOF**：发送 FIFO 已满 (Transmit FIFO full)

它只是一个硬件状态标志，不会生成中断。该位在一个 FIFO 单元为空时清零。

位 15 **RXFIFOHF**：接收 FIFO 半满 (Receive FIFO half full)

FIFO 中至少有一半字数。该位在 FIFO 为“半 + 1”空时清零。

位 14 **TXFIFOHE**：发送 FIFO 半空 (Transmit FIFO half empty)

FIFO 中至少可写入一半字数。该位在 FIFO 为“半 + 1”满时清零。

位 13 **CPSMACT**：命令路径状态机激活，即不处于空闲状态 (Command path state machine active, i.e. not in Idle state)。

它只是一个硬件状态标志，不会生成中断。

位 12 **DPSMACT**：数据路径状态机激活，即不处于空闲状态 (Data path state machine active, i.e. not in Idle state)。

它只是一个硬件状态标志，不会生成中断。

位 11 **DABORT**：数据传输由 CMD12 中止 (Data transfer aborted by CMD12)。

通过写入 SDMMC_ICR 中的相应中断清除位来清除中断标志。

位 10 **DBCKEND**：已发送/接收数据块 (Data block sent/received)。

（CRC 校验已通过）且 DPSM 变为读取等待状态。通过写入 SDMMC_ICR 中的相应中断清除位来清除中断标志。

位 9 **DHOLD**：数据传输保持 (Data transfer Hold)。

通过写入 SDMMC_ICR 中的相应中断清除位来清除中断标志。

位 8 **DATAEND**：数据传输正确结束 (Data transfer ended correctly)。

（数据计数器，DATACOUNT 为零，未出现错误）。通过写入 SDMMC_ICR 中的相应中断清除位来清除中断标志。

位 7 **CMDSENT**：命令已发送（不需要响应）(Command sent (no response required))

通过写入 SDMMC_ICR 中的相应中断清除位来清除中断标志。

位 6 **CMDREND**：已接收命令响应（CRC 校验通过或无 CRC）(Command response received (CRC check passed, or no CRC))。

通过写入 SDMMC_ICR 中的相应中断清除位来清除中断标志。

位 5 **RXOVERR**：已接收 FIFO 上溢错误或 IDMA 写入传输错误 (Received FIFO overrun error or IDMA write transfer error)。

通过写入 SDMMC_ICR 中的相应中断清除位来清除中断标志。

位 4 **TXUNDERR**：发送 FIFO 下溢错误或 IDMA 读取传输错误 (Transmit FIFO underrun error or IDMA read transfer error)。
通过写入 SDMMC_ICR 中的相应中断清除位来清除中断标志。

位 3 **DTIMEOUT**：数据超时 (Data timeout)。
通过写入 SDMMC_ICR 中的相应中断清除位来清除中断标志。

位 2 **CTIMEOUT**：命令响应超时 (Command response timeout)。
通过写入 SDMMC_ICR 中的相应中断清除位来清除中断标志。
命令超时周期为固定值 64 个 SDMMC_CK 时钟周期。

位 1 **DCRCFAIL**：已发送/接收数据块（CRC 校验失败）(Data block sent/received (CRC check failed))。
通过写入 SDMMC_ICR 中的相应中断清除位来清除中断标志。

位 0 **CCRCFAIL**：已接收命令响应（CRC 校验失败）(Command response received (CRC check failed))。
通过写入 SDMMC_ICR 中的相应中断清除位来清除中断标志。

注： *使用 IDMA 模式时，SDMMC_MASKR 中应屏蔽 FIFO 中断标志。*

55.8.12 SDMMC 中断清零寄存器 (SDMMC_ICR)

SDMMC interrupt clear register

偏移地址：0x038

复位值：0x0000 0000

SDMMC_ICR 寄存器是一个只写寄存器。向某个位写入 1 会将 SDMMC_STAR 状态寄存器中的对应位清零。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	IDMABTCC	IDMATEC	CKSTOPC	VSWENDC	ACKTIMEOUTC	ACKFAILC	SDIOITC	BUSYD0ENDC	Res.	Res.	Res.	Res.	Res.
			rw	rw	rw	rw	rw	rw	rw	rw					

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	DABORTC	DBCKENDC	DHOLDC	DATAENDC	CMDSENTC	CMDRENDC	RXOVERRC	TXUNDERRC	DTIMEOUTC	CTIMEOUTC	DCRCFAILC	CCRCFAILC
				rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:29 保留，必须保持复位值。

位 28 **IDMABTCC**：IDMA 缓冲区传输完成清零位 (IDMA buffer transfer complete clear bit)
由软件置 1，用于将 IDMABTC 标志清零。
0：未将 IDMABTC 清零
1：已将 IDMABTC 清零

位 27 **IDMATEC**：IDMA 缓冲区错误清零位 (IDMA transfer error clear bit)
由软件置 1，用于将 IDMATE 标志清零。
0：未将 IDMATE 清零
1：已将 IDMATE 清零

位 26 **CKSTOPC**：CKSTOP 标志清零位 (CKSTOP flag clear bit)
由软件置 1，用于将 CKSTOP 标志清零。
0：未将 CKSTOP 清零
1：已将 CKSTOP 清零

位 25 **VSWENDC**：VSWEND 标志清零位 (VSWEND flag clear bit)
由软件置 1，用于将 VSWEND 标志清零。
0：未将 VSWEND 清零
1：已将 VSWEND 清零

位 24 **ACKTIMEOUTC**：ACKTIMEOUT 标志清零位 (ACKTIMEOUT flag clear bit)
由软件置 1，用于将 ACKTIMEOUT 标志清零。
0：未将 ACKTIMEOUT 清零
1：已将 ACKTIMEOUT 清零

位 23 **ACKFAILC**：ACKFAIL 标志清零位 (ACKFAIL flag clear bit)
由软件置 1，用于将 ACKFAIL 标志清零。
0：未将 ACKFAIL 清零
1：已将 ACKFAIL 清零

位 22 **SDIOITC**：SDIOIT 标志清零位 (SDIOIT flag clear bit)
由软件置 1，用于将 SDIOIT 标志清零。
0：未将 SDIOIT 清零
1：已将 SDIOIT 清零

位 21 **BUSYD0ENDC**：BUSYD0END 标志清零位 (BUSYD0END flag clear bit)
由软件置 1，用于将 BUSYD0END 标志清零。
0：未将 BUSYD0END 清零
1：已将 BUSYD0END 清零

位 20:12 保留，必须保持复位值。

位 11 **DABORTC**：DABORT 标志清零位 (DABORT flag clear bit)
由软件置 1，用于将 DABORT 标志清零。
0：未将 DABORT 清零
1：已将 DABORT 清零

位 10 **DBCKENDC**：DBCKEND 标志清零位 (DBCKEND flag clear bit)
由软件置 1，用于将 DBCKEND 标志清零。
0：未将 DBCKEND 清零
1：已将 DBCKEND 清零

位 9 **DHOLDC**：DHOLD 标志清零位 (DHOLD flag clear bit)
由软件置 1，用于将 DHOLD 标志清零。
0：未将 DHOLD 清零
1：已将 DHOLD 清零

位 8 **DATAENDC**：DATAEND 标志清零位 (DATAEND flag clear bit)
由软件置 1，用于将 DATAEND 标志清零。
0：未将 DATAEND 清零
1：已将 DATAEND 清零

位 7 **CMDSENTC**：CMDSENT 标志清零位 (CMDSENT flag clear bit)
由软件置 1，用于将 CMDSENT 标志清零。
0：未将 CMDSENT 清零
1：已将 CMDSENT 清零

位 6 **CMDRENDC**：CMDREND 标志清零位 (CMDREND flag clear bit)
由软件置 1，用于将 CMDREND 标志清零。
0：未将 CMDREND 清零
1：已将 CMDREND 清零

位 5 **RXOVERRC**：RXOVERR 标志清零位 (RXOVERR flag clear bit)
由软件置 1，用于将 RXOVERR 标志清零。
0：未将 RXOVERR 清零
1：已将 RXOVERR 清零

位 4 **TXUNDERRC**：TXUNDERR 标志清零位 (TXUNDERR flag clear bit)
由软件置 1，用于将 TXUNDERR 标志清零。
0：未将 TXUNDERR 清零
1：已将 TXUNDERR 清零

位 3 **DTIMEOUTC**：DTIMEOUT 标志清零位 (DTIMEOUT flag clear bit)
由软件置 1，用于将 DTIMEOUT 标志清零。
0：未将 DTIMEOUT 清零
1：已将 DTIMEOUT 清零

位 2 **CTIMEOUTC**：CTIMEOUT 标志清零位 (CTIMEOUT flag clear bit)
由软件置 1，用于将 CTIMEOUT 标志清零。
0：未将 CTIMEOUT 清零
1：已将 CTIMEOUT 清零

位 1 **DCRCFAILC**：DCRCFAIL 标志清零位 (DCRCFAIL flag clear bit)
由软件置 1，用于将 DCRCFAIL 标志清零。
0：未将 DCRCFAIL 清零
1：已将 DCRCFAIL 清零

位 0 **CCRCFAILC**：CCRCFAIL 标志清零位 (CCRCFAIL flag clear bit)
由软件置 1，用于将 CCRCFAIL 标志清零。
0：未将 CCRCFAIL 清零
1：已将 CCRCFAIL 清零

55.8.13 SDMMC 屏蔽寄存器 (SDMMC_MASKR)

SDMMC mask register

偏移地址：0x03C

复位值：0x0000 0000

中断屏蔽寄存器通过将对应的位置 1 来确定哪一个状态标志位可以产生中断请求。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	IDMABTCIE	Res.	CKSTOPIE	VSWENDIE	ACKTIMEOUTIE	ACKFAILIE	SDIOITIE	BUSYD0ENDIE	Res.	Res.	TXFIFOEIE	RXFIFOFIE	Res.
			rw		rw	rw	rw	rw	rw	rw			rw	rw	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RXFIFOHFIE	TXFIFOHEIE	Res.	Res.	DABORTIE	DBCKENDIE	DHOLDIE	DATAENDIE	CMDSENTIE	CMDRENDIE	RXOVERRIE	TXUNDERRIE	DTIMEOUTIE	CTIMEOUTIE	DCRCFAILIE	CCRCFAILIE
rw	rw			rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:29 保留，必须保持复位值。

位 28 **IDMABTCIE**：IDMA 缓冲区传输完成中断使能 (IDMA buffer transfer complete interrupt enable)
由软件置 1 和清零，用于使能/禁止 IDMA 传输完存储器缓冲区内的所有数据时生成的中断。
0：禁止 IDMA 缓冲区传输完成中断
1：使能 IDMA 缓冲区传输完成中断

位 27 保留，必须保持复位值。

位 26 **CKSTOPIE**：电压切换时钟停止中断使能 (Voltage Switch clock stopped interrupt enable)
由软件置 1 和清零，用于使能/禁止由电压切换时钟停止引起的中断。
0：禁止电压切换时钟停止中断
1：使能电压切换时钟停止中断

位 25 **VSWENDIE**：电压切换时序关键部分完成中断使能 (Voltage switch critical timing section completion interrupt enable)
由软件置 1 和清零，用于使能/禁止电压切换时序关键部分完成时生成的中断。
0：禁止电压切换时序关键部分完成中断
1：使能电压切换时序关键部分完成中断

位 24 **ACKTIMEOUTIE**：确认超时中断使能 (Acknowledgment timeout interrupt enable)
由软件置 1 和清零，用于使能/禁止由确认超时引起的中断。
0：禁止确认超时中断
1：使能确认超时中断

位 23 **ACKFAILIE**：确认失败中断使能 (Acknowledgment Fail interrupt enable)
由软件置 1 和清零，用于使能/禁止由确认失败引起的中断。
0：禁止确认失败中断
1：使能确认失败中断

位 22 **SDIOITIE**：接收到 SDIO 模式中断时中断使能 (SDIO mode interrupt received interrupt enable)
由软件置 1 和清零，用于使能/禁止收到 SDIO 模式中断时生成中断。
0：禁止接收到 SDIO 模式中断时中断
1：使能接收到 SDIO 模式中断时中断

位 21 **BUSYD0ENDIE**：BUSYD0END 中断使能 (BUSYD0END interrupt enable)
由软件置 1 和清零，用于使能/禁止 CMD 响应后 SDMMC_D0 信号从繁忙状态变为不忙状态时生成的中断。
0：禁止 BUSYD0END 中断
1：使能 BUSYD0END 中断

位 20:19 保留，必须保持复位值。

位 18 **TXFIFOEIE**：Tx FIFO 为空时中断使能 (Tx FIFO empty interrupt enable)
由软件置 1 和清零，用于使能/禁止由 Tx FIFO 为空引起的中断。
0：禁止 Tx FIFO 为空时中断
1：使能 Tx FIFO 为空时中断

位 17 **RXFIFOFIE**：Rx FIFO 变满时中断使能 (Rx FIFO full interrupt enable)
由软件置 1 和清零，用于使能/禁止由 Rx FIFO 变满引起的中断。
0：禁止 Rx FIFO 变满时中断
1：使能 Rx FIFO 变满时中断

位 16 保留，必须保持复位值。

位 15 **RXFIFOHFIE**：Rx FIFO 半满时中断使能 (Rx FIFO half full interrupt enable)
由软件置 1 和清零，用于使能/禁止由 Rx FIFO 半满引起的中断。
0：禁止 Rx FIFO 半满时中断
1：使能 Rx FIFO 半满时中断

位 14 **TXFIFOHEIE**：Tx FIFO 半空时中断使能 (Tx FIFO half empty interrupt enable)
由软件置 1 和清零，用于使能/禁止由 Tx FIFO 半空引起的中断。
0：禁止 Tx FIFO 半空时中断
1：使能 Tx FIFO 半空时中断

位 13:12 保留，必须保持复位值。

位 11 **DABORTIE**：数据传输中止中断使能 (Data transfer aborted interrupt enable)
由软件置 1 和清零，用于使能/禁止由数据传输中止引起的中断。
0：禁止数据传输中止中断
1：使能数据传输中止中断

位 10 **DBCKENDIE**：数据块结束中断使能 (Data block end interrupt enable)
由软件置 1 和清零，用于使能/禁止由数据块结束引起的中断。
0：禁止数据块结束中断
1：使能数据块结束中断

位 9 **DHOLDIE**：数据保持中断使能 (Data hold interrupt enable)
由软件置 1 和清零，用于使能/禁止发送过程中新数据保持 DPSM Wait_S 状态时生成的中断。
0：禁止数据保持中断
1：使能数据保持中断

位 8 **DATAENDIE**：数据结束中断使能 (Data end interrupt enable)
由软件置 1 和清零，用于使能/禁止由数据结束引起的中断。
0：禁止数据结束中断
1：使能数据结束中断

位 7 **CMDSENTIE**：命令发送中断使能 (Command sent interrupt enable)
由软件置 1 和清零，用于使能/禁止由发送命令引起的中断。
0：禁止命令发送中断
1：使能命令发送中断

位 6 **CMDRENDIE**：命令响应接收中断使能 (Command response received interrupt enable)
由软件置 1 和清零，用于使能/禁止由接收命令响应引起的中断。
0：禁止命令响应接收中断
1：使能命令响应接收中断

位 5 **RXOVERRIE**：Rx FIFO 上溢错误中断使能 (Rx FIFO overrun error interrupt enable)
由软件置 1 和清零，用于使能/禁止由 Rx FIFO 上溢错误引起的中断。
0：禁止 Rx FIFO 上溢错误中断
1：使能 Rx FIFO 上溢错误中断

位 4 **TXUNDERRIE**：Tx FIFO 下溢错误中断使能 (Tx FIFO underrun error interrupt enable)
由软件置 1 和清零，用于使能/禁止由 Tx FIFO 下溢错误引起的中断。
0：禁止 Tx FIFO 下溢错误中断
1：使能 Tx FIFO 下溢错误中断

位 3 **DTIMEOUTIE**：数据超时中断使能 (Data timeout interrupt enable)
由软件置 1 和清零，用于使能/禁止由数据超时引起的中断。
0：禁止数据超时中断
1：使能数据超时中断

位 2 **CTIMEOUTIE**：命令超时中断使能 (Command timeout interrupt enable)
由软件置 1 和清零，用于使能/禁止由命令超时引起的中断。
0：禁止命令超时中断
1：使能命令超时中断

位 1 **DCRCFAILIE**：数据 CRC 失败中断使能 (Data CRC fail interrupt enable)
由软件置 1 和清零，用于使能/禁止由数据 CRC 失败引起的中断。
0：禁止数据 CRC 失败中断
1：使能数据 CRC 失败中断

位 0 **CCRCFAILIE**：命令 CRC 失败中断使能 (Command CRC fail interrupt enable)
由软件置 1 和清零，用于使能/禁止由命令 CRC 失败引起的中断。
0：禁止命令 CRC 失败中断
1：使能命令 CRC 失败中断

55.8.14 SDMMC 确认定时器寄存器 (SDMMC_ACKTIMER)

SDMMC acknowledgment timer register

偏移地址：0x040

复位值：0x0000 0000

SDMMC_ACKTIMER 寄存器包含以 SDMMC_CK 总线时钟周期表示的确认超时周期。

计数器加载 SDMMC_ACKTIMER 寄存器的值，并在数据路径状态机 (DPSM) 进入 Wait_Ack 状态后开始递减。如果定时器达到 0 而 DPSM 处于上述状态，则确认超时状态标志置 1。

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

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

位 31:25 保留，必须保持复位值。

位 24:0 **ACKTIME[24:0]**：启动确认超时时间 (Boot acknowledgment timeout period)
只有在禁止 CPSM (CPSMEN=0) 时才能通过固件写入此位。
以卡总线时钟周期表示的启动确认超时周期。

注： *数据传输必须首先写入到确认定时器寄存器，然后才写入到数据控制寄存器。*

55.8.15 SDMMC 数据 FIFO 寄存器 (SDMMC_FIFOR)

SDMMC data FIFO register

偏移地址：0x080 到 0x0BC

复位值：0x0000 0000

接收和发送 FIFO 只能被当作 1 字（32 位）宽的寄存器来读出或写入。FIFO 在连续地址上包含 16 个入口。这使 CPU 可以使用其加载 (load) 和存储 (store) 多操作符命令来读写 FIFO。

当通过半字或字节访问的方式访问 SDMMC_FIFOR 时，会生成 AHB 总线故障。

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

位 31:0 **FIFODATA[31:0]**：接收和发送 FIFO 数据 (Receive and transmit FIFO data)
只有在激活 DPSM (DPSMACT = 1) 时才能通过固件读取或写入此寄存器。
FIFO 数据占用 32 位字的 16 个入口。

55.8.16 SDMMC DMA 控制寄存器 (SDMMC_IDMACTRLR)

SDMMC DMA control register

偏移地址：0x050

复位值：0x0000 0000

接收和发送 FIFO 可以被当做 32 位宽的寄存器来读出或写入。FIFO 在 32 个连续地址上包含 32 个入口。这使 CPU 可以使用其加载 (load) 和存储 (store) 多操作符命令来读写 FIFO。

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

位 31:3 保留，必须保持复位值。

位 2 **IDMABACT**：双缓冲区模式激活缓冲区指示 (Double buffer mode active buffer indication)

只有在未激活 DPSM (DPSMACT = 0) 时才能通过固件写入此位。使能 IDMA 时，该位由硬件切换。

0：使能 IDMA 时，使用缓冲区 0，禁止固件对 IDMABASE0 进行写访问。

1：使能 IDMA 时，使用缓冲区 1，禁止固件对 IDMABASE1 进行写访问。

位 1 **IDMABMODE**：缓冲区模式选择 (Buffer mode selection)。

只有在未激活 DPSM (DPSMACT = 0) 时才能通过固件写入此位。

0：单缓冲区模式。

1：双缓冲区模式。

位 0 **IDMAEN**：IDMA 使能 (IDMA enable)

只有在未激活 DPSM (DPSMACT = 0) 时才能通过固件写入此位。

0：禁止 IDMA

1：使能 IDMA

55.8.17 SDMMC IDMA 缓冲区大小寄存器 (SDMMC_IDMABSIZER)

SDMMC IDMA buffer size register

偏移地址：0x054

复位值：0x0000 0000

SDMMC_IDMABSIZER 寄存器包含双缓冲区配置下的缓冲区大小。

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

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

位 31:13 保留，必须保持复位值。

位 12:5 **IDMABNDT[7:0]**：每个缓冲区的传输次数 (Number of transfers per buffer)。
该 8 位值应乘以 8（可得到用 32 位字表示的缓冲区大小）和 32（可得到用字节表示的缓冲区大小）。
示例：IDMABNDT = 0x01：缓冲区大小 = 8 字 = 32 字节。
只有在未激活 DPSM (DPSMACT = 0) 时才能通过固件写入这些位。

位 4:0 保留，必须保持复位值。

55.8.18 SDMMC IDMA 缓冲区 0 基址寄存器 (SDMMC_IDMABASE0R)

SDMMC IDMA buffer 0 base address register

偏移地址：0x058

复位值：0x0000 0000

SDMMC_IDMABASE0R 寄存器包含单缓冲区配置下的存储器缓冲区基址和双缓冲区配置下的缓冲区 0 基址。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IDMABASE0[31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IDMABASE0[15:0]															
rw	rw	rw		rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	r	r

位 31:0 **IDMABASE0[31:0]**：缓冲区 0 存储器基址位 [31:2]，应按字对齐（位 [1:0] 始终为 0 且为只读位）。
当 DPSM 未激活 (DPSMACT = 0) 时，可通过固件写入该寄存器，当 DPSM 激活 (DPSMACT = 1) 且存储器缓冲区 0 未激活（IDMABACT = “1”）时，可通过固件动态写入该寄存器。

55.8.19 SDMMC IDMA 缓冲区 1 基址寄存器 (SDMMC_IDMABASE1R)

SDMMC IDMA buffer 1 base address register

偏移地址：0x05C

复位值：0x0000 0000

SDMMC_IDMABASE1R 寄存器包含双缓冲区配置下的第二个缓冲区存储器基址。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
IDMABASE1[31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IDMABASE1[15:0]															
rw	rw	rw		rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	r	r

位 31:0 **IDMABASE1[31:0]**：缓冲区 1 存储器基址，应按字对齐（位 [1:0] 始终为 0 且为只读位）。当 DPSM 未激活 (DPSMACT = 0) 时，可通过固件写入该寄存器，当 DPSM 激活 (DPSMACT = 1) 且存储器缓冲区 1 未激活（IDMABACT = “0”）时，可通过固件动态写入该寄存器。

55.8.20 SDMMC 寄存器映射

下表对 SDMMC 寄存器进行了汇总。

表 448. SDMMC 寄存器映射

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	**SDMMC_ POWER**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DIRPOL	VSWITCHEN	VSWITCH	PWRCTRL[1:0]	
	Reset value																												0	0	0	0	0
0x04	**SDMMC_ CLKCR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SELCLKRX[1:0]		BUSSPEED	DDR	HWFC_EN	NEGEDGE	WIDBUS[1:0]		Res.	PWRSAV	Res.	Res.	CLKDIV[9:0]									
	Reset value											0	0	0	0	0	0	0	0		0			0	0	0	0	0	0	0	0	0	0
0x08	**SDMMC_ ARGR**	CMDARG[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0C	**SDMMC_ CMDR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CMDSUSPEND	BOOTEN	BOOTMODE	DTHOLD	CPSMEN	WAITPEND	WAITINT	WAITRESP[1:0]		CMDSTOP	CMDTRANS	CMDINDEX[5:0]					
	Reset value																																
0x10	**SDMMC_ RESPCMDR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RESPCMD[5:0]					
	Reset value																											0	0	0	0	0	0
0x14	**SDMMC_ RESP1R**	CARDSTATUS1[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x18	**SDMMC_ RESP2R**	CARDSTATUS2[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x1C	**SDMMC_ RESP3R**	CARDSTATUS3[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x20	**SDMMC_ RESP4R**	CARDSTATUS4[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x24	**SDMMC_ DTIMER**	DATATIME[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x28	**SDMMC_ DLENR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DATALENGTH[24:0]																								
	Reset value								0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x2C	**SDMMC_ DCTRLR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FIFORST	BOOTACKEN	SDIOEN	RWMOD	RWSTOP	RWSTART	DBLOCK SIZE[3:0]				DTMODE	DTDIR	DTEN
	Reset value																			0	0	0	0	0	0	0	0	0	0	0	0	0	0

表 448. SDMMC 寄存器映射 （续）

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x30	SDMMC_ DCNTR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DATACOUNT[24:0]																								
	Reset value								0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x34	SDMMC_ STAR	Res.	Res.	Res.	IDMABTC	IDMATE	CKSTOP	VSWEND	ACKTIMEOUT	ACKFAIL	SDIOIT	BUSYD0END	BUSYD0	RXFIFOE	TXFIFOE	RXFIFOF	TXFIFOF	RXFIFOHF	TXFIFOHE	CPSMACT	DPSMACT	DABORT	DBCKEND	DHOLD	DATAEND	CMDSENT	CMDREND	RXOVERR	TXUNDERR	DTIMEOUT	CTIMEOUT	DCRCFAIL	CCRCFAIL
	Reset value				0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x38	SDMMC_ ICR	Res.	Res.	Res.	IDMABTCC	IDMATEC	CKSTOPC	VSWEDNDC	ACKTIMEOUTC	ACKFAILC	SDIOITC	BUSYD0ENDC	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DABORTC	DBCKENDC	DHOLDC	DATAENDC	CMDSENTC	CMDRENDC	RXOVERRC	TXUNDERRC	DTIMEOUTC	CTIMEOUTC	DCRCFAILC	CCRCFAILC
	Reset value				0	0	0	0	0	0	0	0										0	0	0	0	0	0	0	0	0	0	0	0
0x3C	SDMMC_ MASKR	Res.	Res.	Res.	IDMABTCIE	Res.	CKSTOPIE	VSWENDIE	ACKTIMEOUTIE	ACKFAILIE	SDIOITIE	BUSYD0ENDIE	Res.	Res.	TXFIFOEIE	RXFIFOFIE	Res.	RXFIFOHFIE	TXFIFOHEIE	Res.	Res.	DABORTIE	DBCKENDIE	DHOLDIE	DATAENDIE	CMDSENTIE	CMDRENDIE	RXOVERRIE	TXUNDERRIE	DTIMEOUTIE	CTIMEOUTIE	DCRCFAILIE	CCRCFAILIE
	Reset value				0		0	0	0	0	0	0			0	0		0	0			0	0	0	0	0	0	0	0	0	0	0	0
0x40	SDMMC_ ACKTIMER	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ACKTIME[24:0]																								
	Reset value								0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x44 - 0x4C	Reserved	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
0x50	SDMMC_ IDMACTRLR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	IDMABACT	IDMABMODE	IDMAEN
	Reset value																														0	0	0
0x54	SDMMC_ IDMABSIZER	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	IDMABNDT[7:0]								Res.	Res.	Res.	Res.	Res.
	Reset value																				0	0	0	0	0	0	0	0					
0x58	SDMMC_ IDMABASE0R	IDMABASE0[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x5C	SDMMC_ IDMABASE1R	IDMABASE1[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x60 - 0x7C	Reserved	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
0x80	SDMMC_ FIFOR	FIF0Data[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x84 - x3FC	Reserved	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

有关寄存器边界地址的信息，请参见*第 100 页的第 2.2.2 节*。

56 FD 控制器局域网络 （FDCAN）

56.1 简介

控制器局域网络 (CAN) 子系统（请参见*图 724*）由两个 CAN 模块、一个共享消息 RAM 存储器和一个时钟校准单元组成。关于这四个组成部分的基址，请参见存储器映射。

两个 CAN 模块（FDCAN1 和 FDCAN2）均符合 ISO 11898-1: 2015（CAN 协议规范第 2.0 版 A、B 部分）和 CAN FD 协议规范第 1.0 版。

此外，第一个 CAN 模块 FDCAN1 支持 ISO 11898-4 中规定的时间触发 CAN (TTCAN)，包括事件同步时间触发通信、全局系统时间和时钟漂移补偿。FDCAN1 还额外包含专供时间触发功能使用的寄存器。CAN FD 选项可与事件触发和时间触发 CAN 通信一起使用。

10 KB 的消息 RAM 存储器可实现过滤器、接收 FIFO、接收缓冲区、发送事件 FIFO、发送缓冲器（TTCAN 触发）功能。该消息 RAM 在 FDCAN1 和 FDCAN2 模块之间共用。

通用时钟校准单元是可选的。通过评估 FDCAN1 接收到的 CAN 消息，该单元可基于 HSI 内部 RC 振荡器和 PLL 为 FDCAN1 和 FDCAN2 生成经过校准的时钟。

表 449 和*表 450* 分别对 CAN 子系统 I/O 信号和引脚进行了详细介绍。

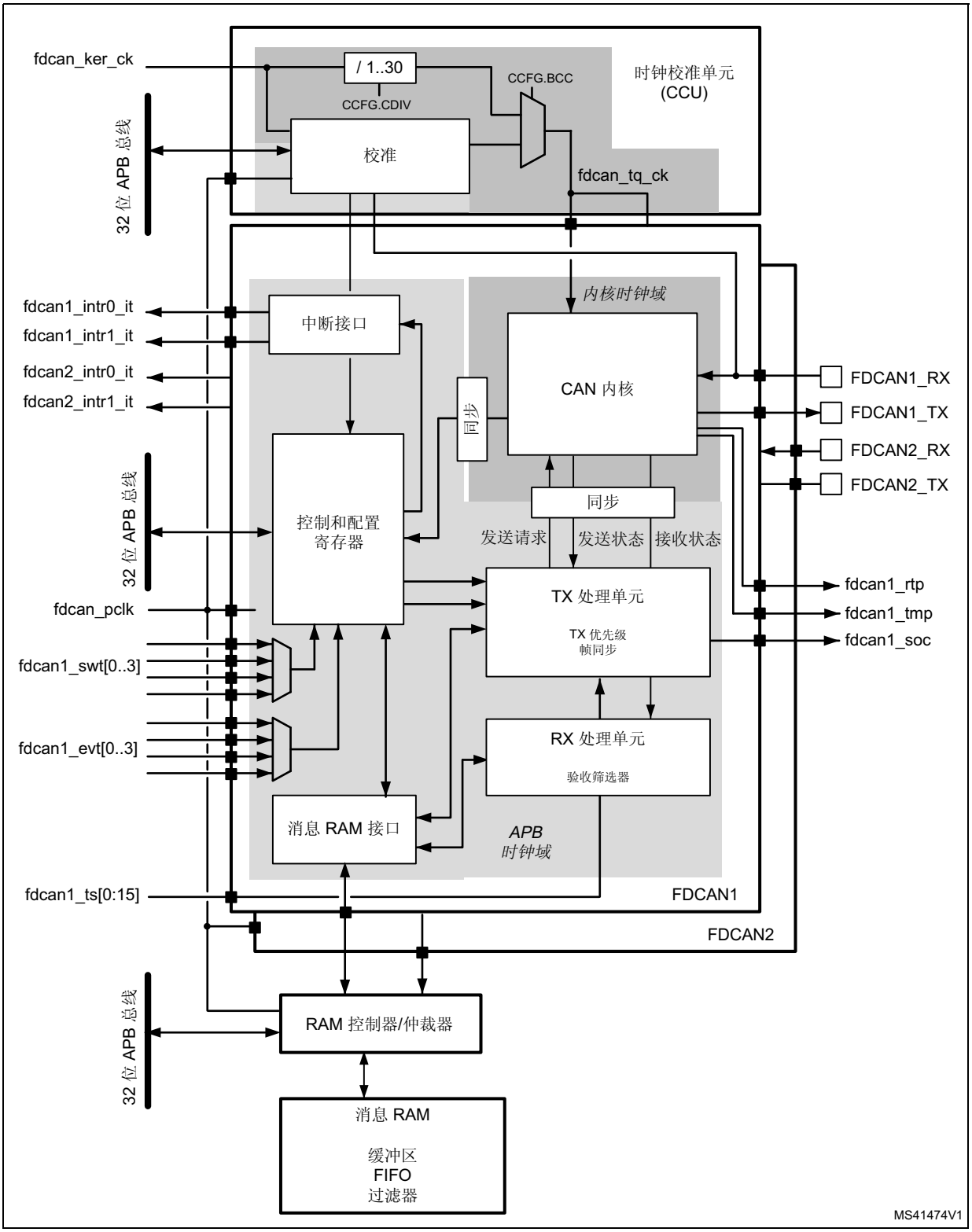
表 449. CAN 子系统 I/O 信号

名称	类型	说明
fdcan_ker_ck	数字输入	CAN 子系统内核时钟输入
fdcan_pclk		CAN 子系统 APB 接口时钟输入
fdan1_intr0_it	数字输出	FDCAN1 中断 0
fdan1_intr1_it		FDCAN1 中断 1
fdan2_intr0_it		FDCAN2 中断 0
fdan2_intr1_it		FDCAN2 中断 1
fdcan1_swt[0:3]	数字输入	停止监视触发输入
fdcan1_evt[0:3]		事件触发输入
fdcan1_ts[0:15]		外部时间戳向量
fdcan1_soc	数字输出	周期开始脉冲
fdcan1_rtp		寄存器时间标记脉冲
fdcan1_tmp		触发时间标记脉冲

表 450. CAN 子系统 I/O 引脚

名称	类型	说明
FDCAN1_RX	数字输入	FDCAN1 接收引脚
FDCAN1_TX	数字输出	FDCAN1 发送引脚
FDCAN2_RX	数字输入	FDCAN2 接收引脚
FDCAN2_TX	数字输出	FDCAN2 发送引脚

图 724. CAN 子系统

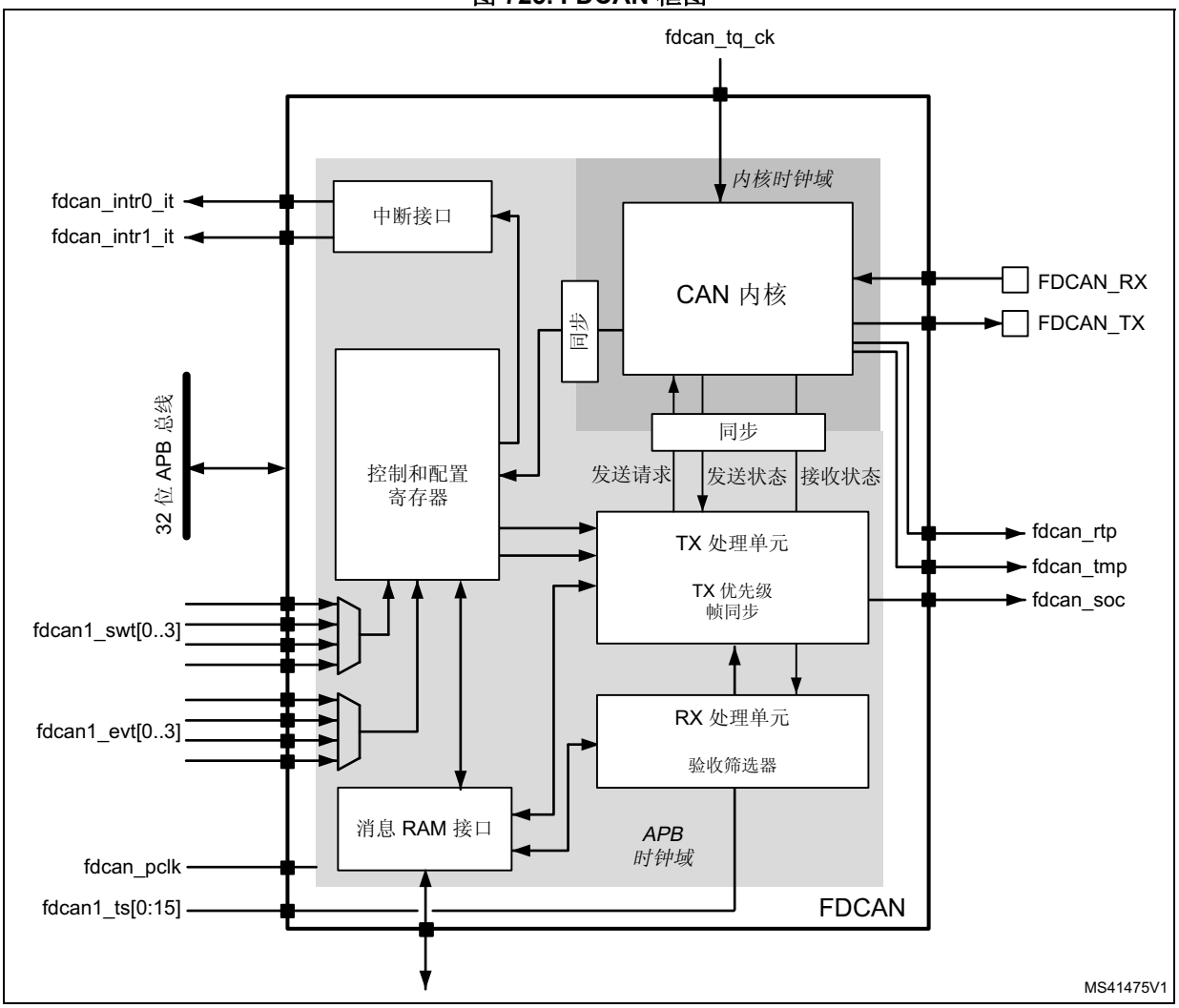

fdcan_ker_ck
/ 1..30
CCFG.CDIV
CCFG.BCC
时钟校准单元
(CCU)
校准
fdcan_tq_ck
32 位 APB 总线
内核时钟域
fdcan1_intr0_it
fdcan1_intr1_it
fdcan2_intr0_it
fdcan2_intr1_it
中断接口
同步
CAN 内核
FDCAN1_RX
FDCAN1_TX
FDCAN2_RX
FDCAN2_TX
32 位 APB 总线
控制和配置
寄存器
同步
发送请求
发送状态
接收状态
fdcan_pclk
TX 处理单元
TX 优先级
帧同步
fdcan1_rtp
fdcan1_tmp
fdcan1_soc
fdcan1_swt[0..3]
fdcan1_evt[0..3]
RX 处理单元
验收筛选器
消息 RAM 接口
APB
时钟域
fdcan1_ts[0:15]
FDCAN1
FDCAN2
32 位 APB 总线
RAM 控制器/仲裁器
消息 RAM
缓冲区
FIFO
过滤器
MS41474V1

56.2 FDCAN 主要特性

- 符合 CAN 协议第 2.0 版 A、B 部分和 ISO 11898-1: 2015, -4
- CAN FD 最多支持收发 64 个数据字节
- 硬件完全支持 TTCAN 协议级别 1 和级别 2（仅限 FDCAN1）
- 支持事件同步时间触发通信（仅限 FDCAN1）
- CAN 错误记录
- 支持 AUTOSAR 和 J1939
- 改进了接收过滤
- 两个可配置接收 FIFO
- 接收到高优先级消息时单独发出信号指示
- 多达 64 个专用接收缓冲区
- 多达 32 个专用发送缓冲区
- 可配置发送 FIFO/队列
- 可配置发送事件 FIFO
- FDCAN1 和 FDCAN2 两个模块共用同一消息 RAM
- 可编程环回测试模式
- 可屏蔽模块中断
- 两个时钟域：APB 总线接口和 CAN 内核时钟
- 支持掉电

56.3 FDCAN 功能说明

图 725. FDCAN 框图



双中断线

FDCAN 外设提供 fdcan_intr0_it 和 fdcan_intr1_it 两条中断线。通过对 FDCAN_ILE 寄存器中的 EINT0 和 EINT1 位进行编程，可分别使能或禁止这两条中断线。

CAN 内核

CAN 内核包含协议控制器和接收/发送移位寄存器。它可处理所有 ISO 11898-1: 2015 协议功能，并支持 11 位和 29 位标识符。

同步

同步模块将来自 APB 时钟域的信号同步到 CAN 内核时钟域，以及将来自 CAN 时钟域的信号同步到 APB 内核时钟域。

发送处理单元

该处理单元控制从消息 RAM 到 CAN 内核的消息传输。最多可配置 32 个发送缓冲区进行发送。发送缓冲区可用作专用发送缓冲区、发送 FIFO（发送队列的组成部分）或二者的组合。发送事件 FIFO 会将发送时间戳与相应的消息 ID 存储在一起，另外还支持取消发送。

在 FDCAN1 上，发送处理单元还会实现帧同步实体 (FSE)，按照 ISO11898-4 的规定控制时间触发通信。它会将自身与 CAN 总线上的参考消息进行同步，控制周期时间和全局时间，并根据预先定义的消息调度（系统矩阵）处理发送操作。它还会处理与消息 RAM 中的消息相关联的系统矩阵的时间标记。停止监视触发、事件触发和时间标记中断均为同步接口。

接收处理单元

该处理单元控制着从 CAN 内核接收的消息到外部消息 RAM 的传输。接收处理单元支持两个接收 FIFO（每个 FIFO 的大小均可配置）以及最多 64 个专用接收缓冲区（用于存储所有通过验收过滤的消息）。专用接收缓冲区仅用于存储具有特定标识符的消息，与接收 FIFO 有所不同。每条消息均与其接收时间戳存储在一起。对于 11 位 ID，最多可定义 128 个过滤器；对于 29 位 ID，最多可定义 64 个过滤器。

APB 接口

将 FDCAN 连接至 APB 总线。

消息 RAM 接口

通过 RAM 控制器/仲裁器将 FDCAN 访问连接到外部 10 KB 消息 RAM。

56.3.1 工作模式

软件初始化

要进行软件初始化，首先应通过软件或硬件复位、或者进入 Bus_Off 状态将 FDCAN_CCCR 寄存器中的 INIT 位置 1。当 FDCAN_CCCR 寄存器中的 INIT 位置 1 时，从 CAN 总线传入和传出的消息都将停止，并且 CAN 总线输出 FDCAN_TX 的状态为隐性（高）。错误管理逻辑 (EML) 的计数器保持不变。将 FDCAN_CCCR 中的 INIT 位置 1 不会更改任何配置寄存器。将 FDCAN_CCCR 中的 INIT 位清零会结束软件初始化。随后，位流处理单元 (BSP) 会等待出现 11 个连续隐性位 (Bus_Idle) 序列，以此将其自身与 CAN 总线上的数据传输进行同步，然后才能参与总线活动并开始消息传输。

仅当 FDCAN_CCCR 寄存器中的 INIT 位以及 FDCAN_CCCR 寄存器中的 CCE 位均置 1 时，才能访问 FDCAN 配置寄存器。

FDCAN_CCCR 寄存器中的 CCE 位只能在 FDCAN_CCCR 中的 INIT 位置 1 时才能置 1/清零。FDCAN_CCCR 寄存器中的 CCE 位会在 FDCAN_CCCR 中的 INIT 位清零时自动清零。

FDCAN_CCCR 寄存器中的 CCE 位置 1 后，以下寄存器会复位：

- FDCAN_HPMS——高优先级消息状态
- FDCAN_ RXF0S——接收 FIFO 0 状态
- FDCAN_RXF1S——接收 FIFO 1 状态
- FDCAN_TXFQS——发送 FIFO/队列状态
- FDCAN_TXBRP——发送缓冲区请求挂起
- FDCAN_TXBTO——发送缓冲区发送已发生
- FDCAN_TXBCF——发送缓冲区取消完成
- FDCAN_TXEFS——发送事件 FIFO 状态
- FDCAN_TTOST——TT 工作状态（仅限 FDCAN1）
- FDCAN_TTLGT——TT 本地和全局时间，仅会复位全局时间 TTLGT.GT （仅限 FDCAN1）
- FDCAN_TTCTC——TT 周期时间和计数（仅限 FDCAN1）
- FDCAN_TTCSM——TT 周期同步标记（仅限 FDCAN1）

FDCAN_CCCR 中的 CCE 位置 1 后，FDCAN_TOCV 寄存器中的超时计数器值 TOC 位会预设为由 FDCAN_TOCC 寄存器中的 TOP 位配置的值。

此外，FDCAN_CCCR 中的 CCE 位置 1 时，发送处理单元和接收处理单元中的状态机会保持空闲状态。

仅当 FDCAN_CCCR 寄存器中的 CCE 位清零时，才能写入以下寄存器：

- TXBAR——发送缓冲区添加请求
- TXBCR——发送缓冲区取消请求

仅当 CCCR 寄存器中的 INIT 位和 CCE 位均已置 1 时，才能通过软件将 FDCAN_CCCR 中的 TEST 位和 MON 位置 1。这两个位可随时复位。仅当 FDCAN_CCCR 中的 INIT 位和 CCE 位均已置 1 时，才能将 FDCAN_CCCR 中的 DAR 位置 1/清零。

正常工作

FDCAN1 在硬件复位后的默认工作模式是无时间触发的事件驱动 CAN 通信（TTOCF[OM] = “00”）。要更改 TT 工作模式，需要先将 FDCAN_CCCR 寄存器中的 INIT 位和 CCE 位置 1。

FDCAN 完成初始化且 FDCAN_CCCR 寄存器中的 INIT 位清零后，FDCAN 会将其自身与 CAN 同步并准备好进行通信。

通过接收滤波后，接收到的包含消息 ID 和 DLC 的消息会存储到专用的接收缓冲区中，或者存储到接收 FIFO 0 或接收 FIFO 1 中。

要发送消息，可初始化或更新发送缓冲区和/或发送 FIFO 或发送队列。不支持在接收到远程帧后自动发送。

CAN FD 操作

FDCAN 协议中有两种类型，第一种是长帧模式 (LFM)，在该模式下，CAN 帧的数据字段可以超过八个字节。第二种是快速帧模式 (FFM)，在该模式下，CAN 帧的控制字段、数据字段和 CRC 字段的发送比特率要高于帧开始和结束的发送比特率。快速帧模式可与长帧模式结合使用。

具有 11 位标识符的 CAN 帧中之前保留的位和具有 29 位标识符的 CAN 帧中之前保留的第一个位现在可解码为 FDF 位。FDF 隐性表示 CAN FD 帧，而 FDF 显性则表示典型 CAN 帧。在 CAN FD 帧中，FDF 之后的两个位 res 和 BRS 决定是否切换这一 CAN FD 帧中的比特率。CAN FD 比特率切换通过 res 显性和 BRS 隐性指示。res 隐性的编码保留，供将来协议扩展使用。如果 M_TTCAN 接收到的帧为 FDF 隐性和 res 隐性，则会通过将 PSR.PXE 位置 1 的方式发出协议异常事件。如果使能协议异常处理（CCCR.PXHD = “0”），下一采样点的工作状态会从接收器（PSR.ACT = “10”）变为同步中（PSR.ACT = “00”）。如果禁止协议异常处理（CCCR.PXHD = “1”），FDCAN 会将隐性 res 位当作格式错误处理，并将以错误帧进行响应。

CAN FD 操作通过编程 CCCR.FDOE 来使能。如果 CCCR.FDOE = “1”，则会使能 CAN FD 帧的发送和接收。始终可进行典型 CAN 帧的发送和接收。是否可发送 CAN FD 帧或典型 CAN 帧可通过相应发送缓冲区元素中的 FDF 位配置。如果 CCCR.FDOE = “0”，接收到的帧会被当作典型 CAN 帧，从而会在接收到 CAN FD 帧时发送错误帧。如果 CAN FD 操作已禁止，即使发送缓冲区元素的 FDF 位置 1，也不会发送 CAN FD 帧。仅当 CCCR.INIT 和 CCCR.CCE 均置 1 时，才能更改 CCCR.FDOE 和 CCCR.BRSE。

如果 CCCR.FDOE = “0”，则会忽略 FDF 和 BRS 位的设置，并会以典型 CAN 格式发送帧。如果 CCCR.FDOE = “1”且 CCCR.BRSE = “0”，则仅会评估发送缓冲区元素的 FDF 位。如果 CCCR.FDOE = “1”且 CCCR.BRSE = “1”，则会使能通过比特率切换发送 CAN FD 帧。所有 FDF 和 BRS 位置 1 的发送缓冲区元素均会通过比特率切换以 CAN FD 格式发送。

建议仅当满足以下条件下在 CAN 操作期间切换模式：

- CAN FD 数据阶段的故障率显著高于 CAN FD 仲裁阶段的故障率。在这种情况下，应禁止发送的 CAN FD 比特率切换选项。
- 系统启动期间，所有节点在被验证能够以 CAN FD 格式进行通信之前，都将发送典型 CAN 消息。如果通过验证，所有节点会切换为 CAN FD 操作。
- CAN 局部网络中的唤醒消息必须以典型 CAN 格式进行发送。
- 并非所有节点均能进行 CAN FD 操作时采用行结束编程。非 CAN FD 节点在编程完成之前会一直保持静默模式。随后，所有节点会切换回典型 CAN 通信。

在 FDCAN 格式中，DLC 的编码与标准 CAN 格式不同。DLC 代码 0 到 8 的编码与标准 CAN 相同，代码 9 到 15（在标准 CAN 中，所有代码的数据字段均为 8 个字节）是按照 *表 451* 进行编码的。

表 451. FDCAN 中的 DLC 编码

DLC	9	10	11	12	13	14	15
数据字节数	12	16	20	24	32	48	64

在 CAN FD 快速帧中，如果该位为隐性，则在 BRS（比特率切换）位后，位时长将在帧内切换。在 BRS 位之前，在 FDCAN 仲裁阶段，标准 CAN 位时长按照位时长和预分频器寄存器 BTP 定义的方式使用。在接下来的 FDCAN 数据阶段，快速 CAN 位时长按照快速位时长和预分频器寄存器 FBTP 定义的方式使用。位时长会在出现 CRC 分隔符或检测到错误时从快速时长切换回来，以先发生的事件为准。

CAN FD 数据阶段的最大可配置比特率取决于 FDCAN 内核时钟频率。举例来说，如果 FDCAN 内核时钟频率为 20 MHz，最短可配置位时间为四个时间片 (tq)，则数据阶段的比特率为 5 Mb/s。

在 CAN FD 长帧和 CAN FD 快速帧这两种数据帧格式中，位 ESI（错误状态指示符）的值是由发送开始时的发送器错误状态决定的。如果发送器为错误被动状态，则 ESI 会隐性传送，否则会显性传送。在 CAN FD 远程帧中，ESI 位始终显性传送，与发送器错误状态无关。CAN FD 远程帧的数据长度代码以 0 的形式发送。

为 DLC > 8 的 FDCAN 发送配置 FDCAN 发送缓冲区时，前 8 个字节会按照发送缓冲区中的配置发送，而数据字段的剩余部分会用 0xCC 填充。当 FDCAN 接收到 DLC > 8 的 FDCAN 帧时，该帧的前 8 个字节会存储到匹配的接收缓冲区或接收 FIFO 中。剩余字节会被丢弃。

收发器延迟补偿

在 FDCAN 发送的数据阶段，只有一个节点会进行发送，所有其他节点都是接收器。总线的长度没有影响。通过引脚 FDCAN_TX 进行发送时，协议控制器会通过引脚 FDCAN_RX 接收到其本地 CAN 收发器发送的数据。接收数据的延迟为 CAN 收发器环路延迟。如果该延迟大于 TSEG1（采样点之前的时间段），则会检测到位错误。如果不进行收发器延迟补偿，FDCAN 帧数据阶段的比特率会受到收发器环路延迟的限制。

FDCAN 会实施延迟补偿机制来补偿 CAN 收发器环路延迟，从而可在 FDCAN 数据阶段以较高的比特率进行发送，而不受特定 CAN 收发器延迟的影响。

为了检查发送节点的数据阶段是否存在位错误，会将延迟发送数据与在第二采样点 SSP 接收到的数据进行比较。如果检测到位错误，发送器将在下一常规采样点对这一位错误作出响应。在仲裁阶段，始终会禁止延迟补偿。

发送器延迟补偿会使能数据位时间短于发送器延迟的配置，新版 ISO11898-1 中进行了详细介绍。发送器延迟补偿是通过将 DBTP.TDC 位置 1 使能的。

接收到的位会与在 SSP 发送的位进行比较。SSP 位置定义为从 FDCAN 发送输出引脚 FDCAN_TX 通过收发器到接收输入引脚 FDCAN_RX 测得的延迟与由 TDCR.TDCO 配置的发送器延迟补偿偏移之和。发送器延迟补偿偏移用于调整已接收位中 SSP 的位置（例如数据阶段的半个位时间）。第二采样点的位置会向下舍入到下一 mtq 整数（最小时间片，即一个 fdcan_tq_ck 时钟周期）。

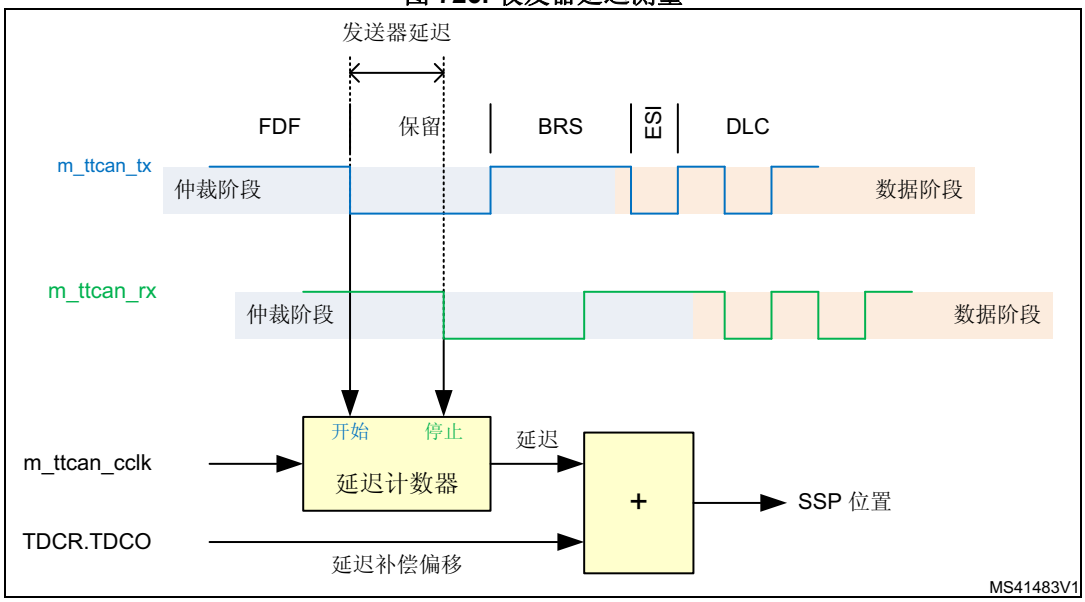
PSR.TDCV 显示了实际发送器延迟补偿值。CCCR.INIT 置 1 时，PSR.TDCV 会清零；并且如果 DBTP.TDC 置 1，每次发送 FD 帧时，PSR.TDCV 都会更新。

对于在 FDCAN 中实施的发送器延迟补偿，必须考虑以下边界条件：

- 测得的 m_ttcan_tx 到 m_ttcan_rx 的延迟与配置的发送器延迟补偿偏移 TDCR.TDCO 之和必须小于数据阶段的 6 个位时间。
- 测得的 m_ttcan_tx 到 m_ttcan_rx 的延迟与配置的发送器延迟补偿偏移 TDCR.TDCO 之和必须小于或等于 127 mtq。如果二者之和超过 127 mtq，则为发送器延迟补偿使用最大值 (127 mtq)。
- 数据阶段在 CRC 分隔符的采样点结束，此时会停止检查在 SSP 接收到的位

如果通过编程 DBTP.TDC = “1” 使能发送器延迟补偿，则会在每个 FDF 位到 res 位的下降沿处发送的 CAN FD 帧中开始测量。在发送器的接收输入引脚 FDCAN-RX 上观察到该边沿时，会停止测量。该测量的分辨率为一个 mtq。

图 726. 收发器延迟测量



为了避免接收到的 FDF 位中的显性毛刺信号导致在接收到的 res 位下降沿之前结束延迟补偿测量（造成 SSP 位置过早），可通过编程 TDCR.TDCF 来使能发送器延迟补偿过滤器窗口，从而定义最小 SSP 位置值。进行发送器延迟测量时，会忽略可能导致 SSP 位置更加提前的 m_ttcan_rx 上的显性边沿。当 SSP 位置至少为 TDCR.TDCF 且 FDCAN_RX 为低电平时，会停止测量。

受限工作模式

在受限工作模式下，节点能够接收数据帧和远程帧，并能对有效帧进行确认，但不会发送数据帧、远程帧、有效错误帧或过载帧。如果存在错误条件或过载条件，则不会发送显性位，而会等待出现总线空闲条件，以便使自身与 CAN 通信重新同步。错误记录 (ECR.CEL) 激活时，错误计数器（ECR.REC、ECR.TEC）会冻结。软件可通过将 CCCR.ASM 位置 1 的方式将 FDCAN 设置为进入受限工作模式。仅当 CCCR.CCE 和 CCCR.INIT 均设为“1”时，才能通过软件将此位置 1。此位可随时通过软件清零。

当发送处理单元无法及时从消息 RAM 中读取数据时，会自动进入受限工作模式。要退出受限工作模式，软件必须复位 CCCR.ASM。

受限工作模式可用于根据不同 CAN 比特率对自身进行调整的应用中。在这种情况下，应用会测试不同比特率，并在收到有效帧后退出受限工作模式。

CCCR.ASM 也由时钟校准单元控制。时钟校准过程使能后，会进入受限工作模式，CCR.ASM 位会置 1。校准完成后，CCR.ASM 位会清零。

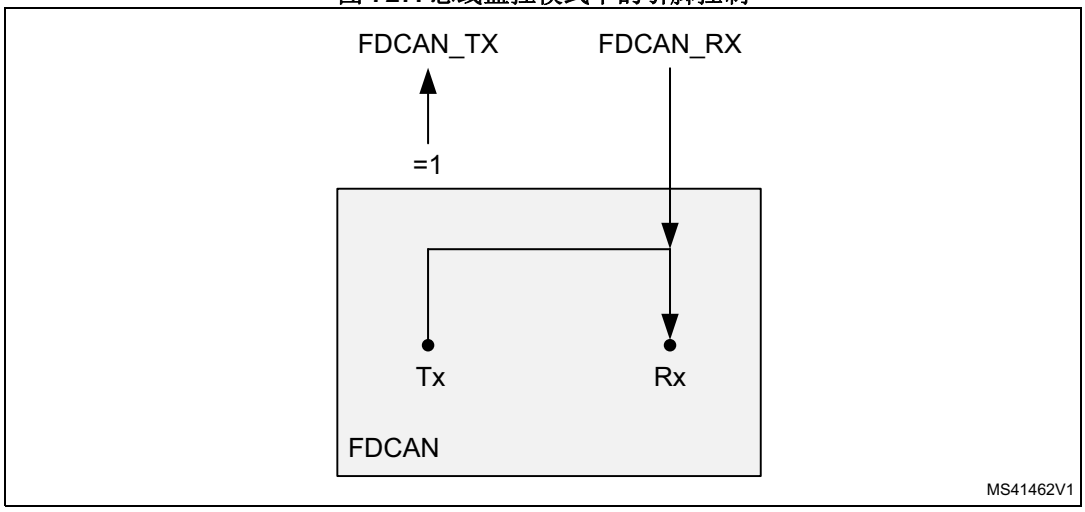
注： *受限工作模式不得与环回模式（内部或外部）结合使用。*

总线监控模式

将 CCCR.MON 位置 1 或进入错误级别 S3（TTOST[EL] = “11”）时，FDCAN 会被置于总线监控模式。在总线监控模式下（更多详细信息，请参见 ISO11898-1，10.12 总线监控），FDCAN 能够接收有效数据帧和有效远程帧，但不能启动发送过程。在该模式下，FDCAN 仅会在 CAN 总线上发送隐性位，如果 FDCAN 必须发送一个显性位（ACK 位、过载标志、活动错误标志），该位将在内部被改道发送，以便 FDCAN 可以监视该显性位，但 CAN 总线可以保持隐性状态。在总线监控模式下，寄存器 TXBRP 会保持复位状态。

总线监控模式可用于分析 CAN 总线上的流量，同时又不会因发送显性位对其造成影响。*图 727* 显示了总线监控模式下 FDCAN_TX 和 FDCAN_RX 信号与 FDCAN 的连接。

图 727. 总线监控模式下的引脚控制



禁止自动重发送 (DAR) 模式

根据 CAN 规范（请参见 ISO11898-1，6.3.3 恢复管理），FDCAN 提供的方法可自动重发已丢失仲裁的帧或在发送期间受到错误干扰的帧。默认使能自动重发送。

为了支持 ISO 11898-1: 2015 章节 9.2 中介绍的时间触发通信，可通过 CCCR[DAR] 禁止自动重发送。

禁止自动重发送 (DAR) 模式下的帧发送

在 DAR 模式下，所有发送在 CAN 总线上启动后都会自动取消。发送缓冲区发送请求挂起位 TXBRP.TRPx 会在成功发送后复位，如果取消时发送尚未开始、发送已因仲裁丢失而中止、或者帧发送期间出错，该位也会复位。

- 成功发送：
  - 相应的发送缓冲区发送已发生位 TXBTO[TOx] 置 1
  - 相应的发送缓冲区取消完成位 TXBCF[CFx] 不置 1
- 即使已取消发送也会成功进行发送：
  - 相应的发送缓冲区发送已发生位 TXBTO[TOx] 置 1
  - 相应的发送缓冲区取消完成位 TXBCF[CFx] 置 1
- 仲裁丢失或帧发送受到干扰：
  - 相应的发送缓冲区发送已发生位 TXBTO[TOx] 不置 1
  - 相应的发送缓冲区取消完成位 TXBCF[CFx] 置 1

对于帧发送成功的情况，如果使能了发送事件存储，则会写入发送事件 FIFO 元素，写入的事件类型 ET = “10”（即使已取消发送也会进行发送）。

掉电（休眠模式）

FDCAN 可通过 CC 控制寄存器 CCCR[CSR] 设为由时钟停止请求输入控制的掉电模式。只要时钟停止请求有效，CCCR[CSR] 位就读为 1。

所有挂起发送请求完成后，FDCAN 会一直等待至检测到总线空闲状态。检测到总线空闲状态之后，FDCAN 会置 1，随后，CCCR[INIT] 会置 1，以免继续进行 CAN 传输。现在，FDCAN 会通过将 CCCR[CSA] 置 1 确认其准备好进入掉电状态。在该模式下，时钟关闭之前，可继续访问寄存器。对 CCCR[INIT] 的写访问不起作用。现在，可以关闭模块时钟输入。

退出掉电模式之前，应用必须在复位 CC 控制寄存器标志 CCCR.CSR 之前开启模块时钟。FDCAN 将通过复位 CCCR[CSA] 确认时钟开启。之后，应用可通过复位 CCCR[INIT] 位重新开始 CAN 通信。

测试模式

要使能对 FDCAN 测试寄存器的写访问（请参见*第 2297 页的第 56.4.4 节*），必须将 CCCR.TEST 位置 1，从而能够配置测试模式和功能。

可通过编程 TEST.TX 将四种输出功能用于 CAN 发送引脚 m_can_tx。除了串行数据输出这一默认功能之外，还可驱动 CAN 采样点信号来监控 FDCAN 位时长，并可驱动常量显性或隐性值。引脚 m_can_rx 的实际值可从 TEST.RX 读取。这两种功能均可用于检查 CAN 总线物理层。

由于 CAN 内核时钟与 APB 时钟域之间的同步机制，在 FDCAN_TX 输出引脚上显示新配置之前，对 TEST.TX 的写操作之间可能有几个 APB 时钟周期的延迟。这一点还适用于通过 TEST.RX 读取 FDCAN_RX 输入引脚的情况。

注： *测试模式仅限用于生产测试和自检。对 FDCAN_TX 引脚进行软件控制会干扰所有 CAN 协议功能。不建议为应用使用测试模式。*

外部环回模式

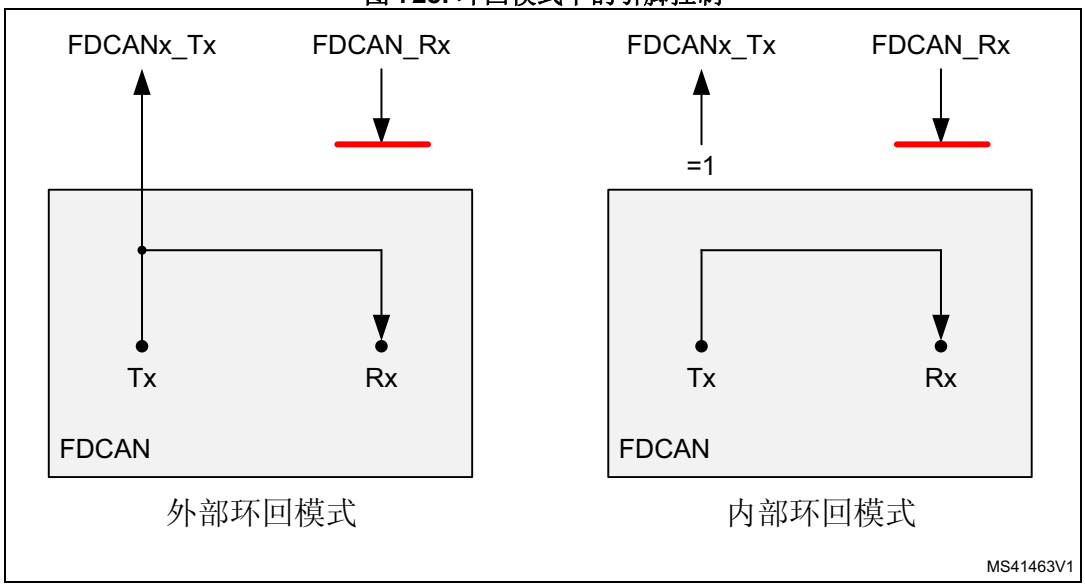
在外部环回模式下，可通过将 TEST.LBCK 编程为 1 来将 FDCAN 置 1。在环回模式下，FDCAN 将其自身发送的消息作为接收的消息来处理，并将消息存储（如果这些消息通过了验收过滤）在接收 FIFO 中。*图 728* 显示了外部环回模式下发送和接收信号 FDCAN_TX 和 FDCAN_RX 与 FDCAN 的连接。

该模式为硬件自检提供。为了不受外部仿真的影响，FDCAN 在环回模式下将忽略确认错误（在数据/远程帧的确认时隙对隐性位采样）。在此模式下，FDCAN 将执行从发送输出到其接收输入的内部反馈。FDCAN 将忽略 FDCAN_RX 输入引脚的实际值。从 FDCAN_TX 发送引脚可以监视发送的消息。

内部环回模式

将 TEST.LBCK 和 CCCR.MON 位编程为 1 会进入内部环回模式。该模式可用于“热自检”，也就是说，FDCAN 可以进行检测，同时又不会影响与 FDCAN_TX 和 FDCAN_RX 引脚相连接的运行中的 CAN 系统。在此模式下，FDCAN_RX 引脚与 FDCAN 断开连接，FDCAN_TX 引脚则保持隐性。*图 728* 显示了内部环回模式下 FDCAN_TX 和 FDCAN_RX 引脚与 FDCAN 的连接。

图 728. 环回模式下的引脚控制



应用看门狗（仅限 FDCAN1）

应用看门狗通过读取寄存器 TTOST 进行处理。如果未及时处理应用看门狗，位 TTOST.AWE 会置 1，所有 TTCAN 通信都会停止，FDCAN1 会设为进入总线监控模式。

将应用看门狗限值 TTOCF[AWL] 编程为 0x00 可禁止 TT 应用看门狗。不应在 TTCAN 应用程序中禁止 TT 应用看门狗。

时间戳生成

FDCAN 提供一个 16 位回卷计数器来生成时间戳。预分频器 TSCC.TCP 可配置为以 CAN 位时间的倍数 (1…16) 为计数器提供时钟。计数器值可通过 TSCV[TCV] 读取。对寄存器 TSCV 进行写访问会将计数器复位为 0。时间戳计数器回卷时，中断标志 IR[TSW] 会置 1。

开始接收/发送帧时，会捕获计数器值，并将该值存储在接收缓冲区/接收 FIFO (RXTS[15:0]) 或发送事件 FIFO (TXTS[15:0]) 元素的时间戳部分。

通过编程 TSCC.TSS 位，可以使用 16 位时间戳。

超时计数器

FDCAN 提供 16 位超时计数器来指示接收 FIFO 0、接收 FIFO 1 和发送事件 FIFO 的超时条件。它作为递减计数器工作，并且与时间戳计数器一样，使用由 TSCC[TCP] 控制的预分频器。超时计数器通过寄存器 TOCC 配置。实际计数器值可从 TOCV[TOC] 读取。仅当 CCCR[INIT] = “0” 时，才会启动超时计数器。当 CCCR[INIT] = “1” 时，超时计数器停止计数，例如 FDCAN 进入 Bus_Off 状态。

工作模式是通过 TOCC[TOS] 选择的。在连续模式下工作时，计数器会在 CCCR[INIT] 复位时开始计数。对 TOCV 进行写操作会将计数器预设为由 TOCC[TOP] 配置的值并继续递减计数。

超时计数器由其中一个 FIFO 控制时，空 FIFO 会将计数器预设为由 TOCC[TOP] 配置的值。存储了第一个 FIFO 元素后，会开始递减计数。对 TOCV 执行写入操作不起作用。

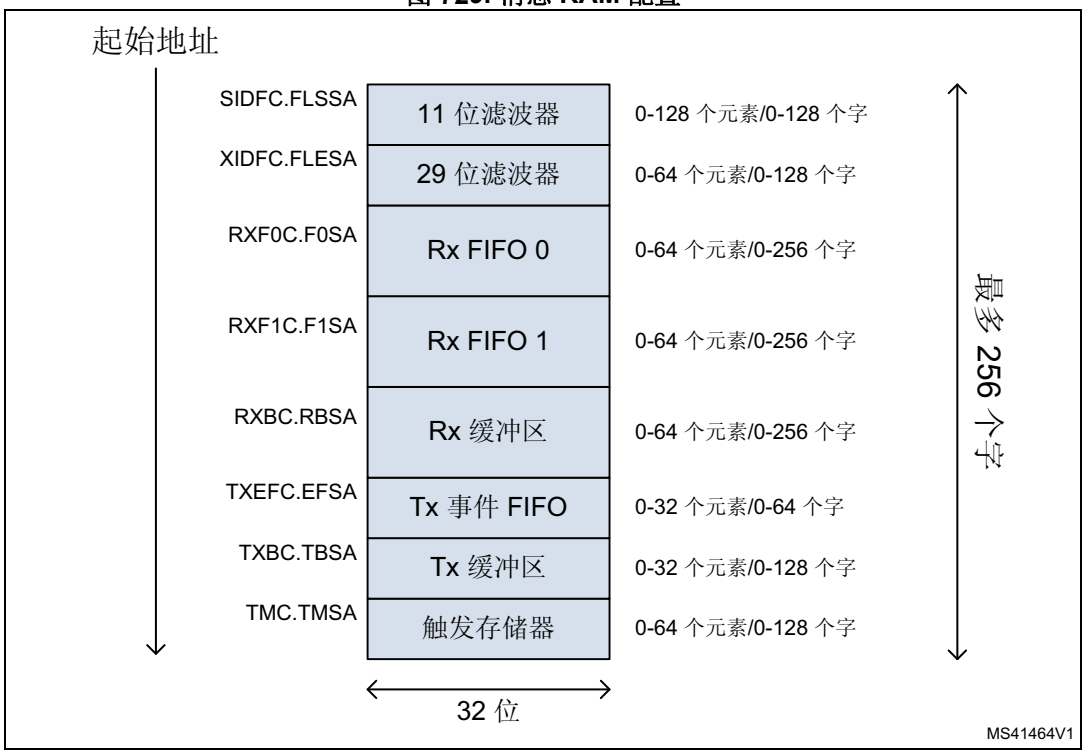
计数器达到 0 时，中断标志 IR[TOO] 会置 1。在连续模式下，计数器会在达到 TOCC[TOP] 后立即重启。

注： *超时计数器的时钟信号来自 CAN 内核采样点信号。因此，由于 CAN 内核同步/重新同步机制的原因，超时计数器递减的时间点可能有所不同。如果使用 FDCAN 中的波特率切换功能，则超时计数器在仲裁字段和数据字段中使用不同的时钟。*

56.3.2 消息 RAM

消息 RAM 的宽度为 32 位。FDCAN 模块最多可在消息 RAM 中分配 2560 个字。不需要一一配置*图 729* 中列出的每个部分，各部分之间的顺序也没有限制。

图 729. 消息 RAM 配置



当 FDCAN 对消息 RAM 进行寻址时，会寻址 32 位字，而不是单独字节。配置的起始地址为 32 位字地址，也就是说，仅会评估位 15 到位 2，两个最低有效位会被忽略。

注： *FDCAN 不会检查消息 RAM 的配置是否存在错误。因此配置不同部分的起始地址以及各部分的元素数时必须多加留意，以免造成数据入侵或丢失。*

接收处理

接收处理单元控制着验收过滤、已接收消息到接收缓冲区或其中一个接收 FIFO（共两个）的传输以及接收 FIFO 放入和获取索引。

验收过滤器

FDCAN 能够配置两组验收过滤器，一组供标准标识符使用，另一组供扩展标识符使用。这两组过滤器可分配给接收缓冲区、接收 FIFO 0 或接收 FIFO 1。进行验收过滤时，会从第 0 个元素开始执行每个过滤器列表，直至过滤出第一个匹配元素。过滤出第一个匹配元素后，验收过滤会停止。不会为该消息评估以下过滤器元素。

主要特性有：

- 每个过滤器元素可配置为
  - 范围过滤器（起止范围）
  - 用于一个或两个专用 ID 的过滤器
  - 典型位屏蔽过滤器
- 每个过滤器元素均可配置为实现验收过滤或拒绝过滤
- 每个过滤器元素均可单独使能/禁止
- 会按顺序检查过滤器，过滤出第一个匹配的过滤器元素后会停止执行

相关配置寄存器位：

- 全局过滤器配置 (GFC)
- 标准 ID 过滤器配置 (SIDFC)
- 扩展 ID 过滤器配置 (XIDFC)
- 扩展 ID 与掩码 (XIDAM)

根据过滤器元素的配置 (SFEC/EFEC)，发生匹配时会触发下列操作之一：

- 将接收到的帧存储在 FIFO 0 或 FIFO 1 中
- 将接收到的帧存储在接收缓冲区中
- 将接收到的帧存储在接收缓冲区中，并在过滤器事件引脚上生成脉冲
- 拒绝接收到的帧
- 将高优先级消息中断标志 IR[HPM] 置 1
- 将高优先级消息中断标志 IR[HPM] 置 1，并将接收到的帧存储在 FIFO 0 或 FIFO 1 中
- 将高优先级消息中断标志 IR.HPM 置 1，并将接收到的帧存储在 FIFO 0 或 FIFO 1 中

接收到完整的标识符后，开始验收过滤。验收过滤完成后，如果已找到匹配的接收缓冲区或接收 FIFO，消息处理单元便会开始将接收到的消息数据以 32 位的形式写入匹配的接收缓冲区或接收 FIFO 中。如果 CAN 协议控制器已检测到错误条件（例如 CRC 错误），则会丢弃此消息，丢弃后的影响如下：

- **接收缓冲区**
  匹配接收缓冲区的新数据标志不会置 1，但接收缓冲区（部分）会被接收到的数据覆盖。有关错误类型，请参见 PSR.LEC 和 PSR.DLEC。
- **接收 FIFO**
  匹配接收 FIFO 的放入索引不会更新，但相关接收 FIFO 元素（部分）会被接收到的数据覆盖。有关错误类型，请参见 PSR.LEC 和 PSR.DLEC。如果匹配的接收 FIFO 在覆盖模式下工作，必须考虑*接收 FIFO 覆盖模式*中介绍的边界条件。

注： *当接受的消息写入两个接收 FIFO 之一、或写入接收缓冲区中时，未修改的已接收标识符会独立于所用过滤器进行存储。验收过滤过程的结果很大程度上取决于所配置的过滤器元素的顺序。*

范围过滤器

该过滤器会将所有接收到的帧与由 SF1ID/SF2ID 和 EF1ID/EF2ID 定义的范围内的消息 ID 进行匹配检查。

范围过滤与扩展帧一起使用时，有两种可能的情况：

- EFT = “00”：应用范围过滤器之前，会将已接收帧的消息 ID 与扩展 ID 与掩码 (XIDAM) 进行与运算
- EFT = “11”：不会为范围过滤使用扩展 ID 与掩码 (XIDAM)

专用 ID 过滤器

可将过滤器元素配置为过滤一个或两个特定的消息 ID。要过滤一个特定的消息 Id，过滤器元素必须配置为 SF1ID=SF2ID 和 EF1ID=EF2ID。

典型位屏蔽过滤器

典型位屏蔽过滤用于通过屏蔽已接收消息 ID 的单个位来过滤消息 ID 组。进行典型位屏蔽过滤时，SF1ID/EF1ID 用作消息 ID 过滤器，SF2ID/EF2ID 则用作过滤器屏蔽位。

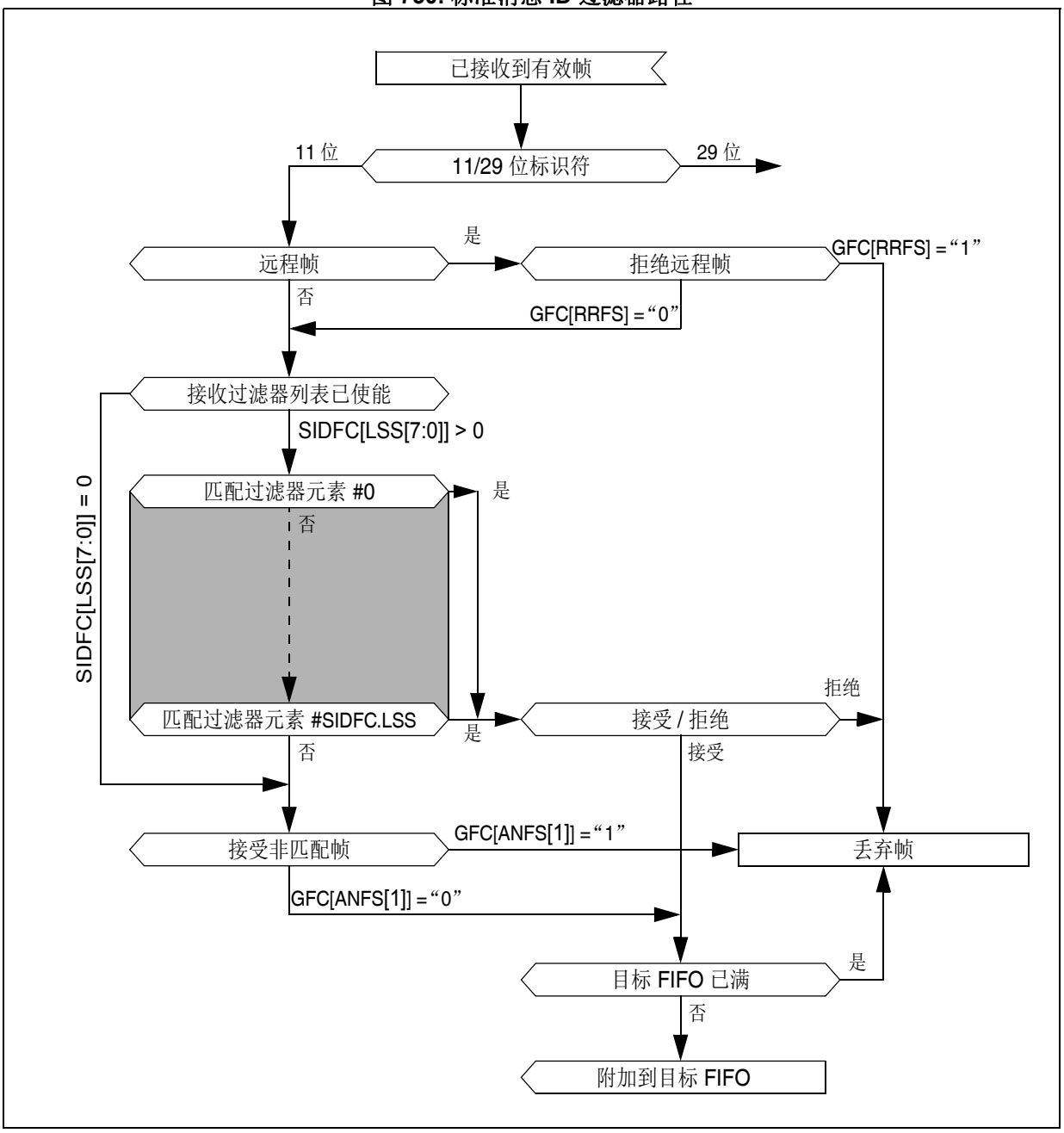
过滤器屏蔽的 0 位将屏蔽掉已配置 ID 过滤器的相应位位置，例如，已接收消息 ID 在该位位置的值与验收过滤无关。只有相应屏蔽位为 1 的已接收消息 ID 的位才与验收过滤有关。

如果所有屏蔽位均为 1，则仅当接收到的消息 ID 和消息 ID 过滤器完全相同时，才会出现匹配。如果所有屏蔽位均为 0，则所有消息 ID 都匹配。

标准消息 ID 过滤

图 730 介绍了标准消息 ID（11 位标识符）的过滤流程。*第 2291 页的第 56.3.19 节*中介绍了标准消息 ID 过滤器元素。

图 730. 标准消息 ID 过滤器路径

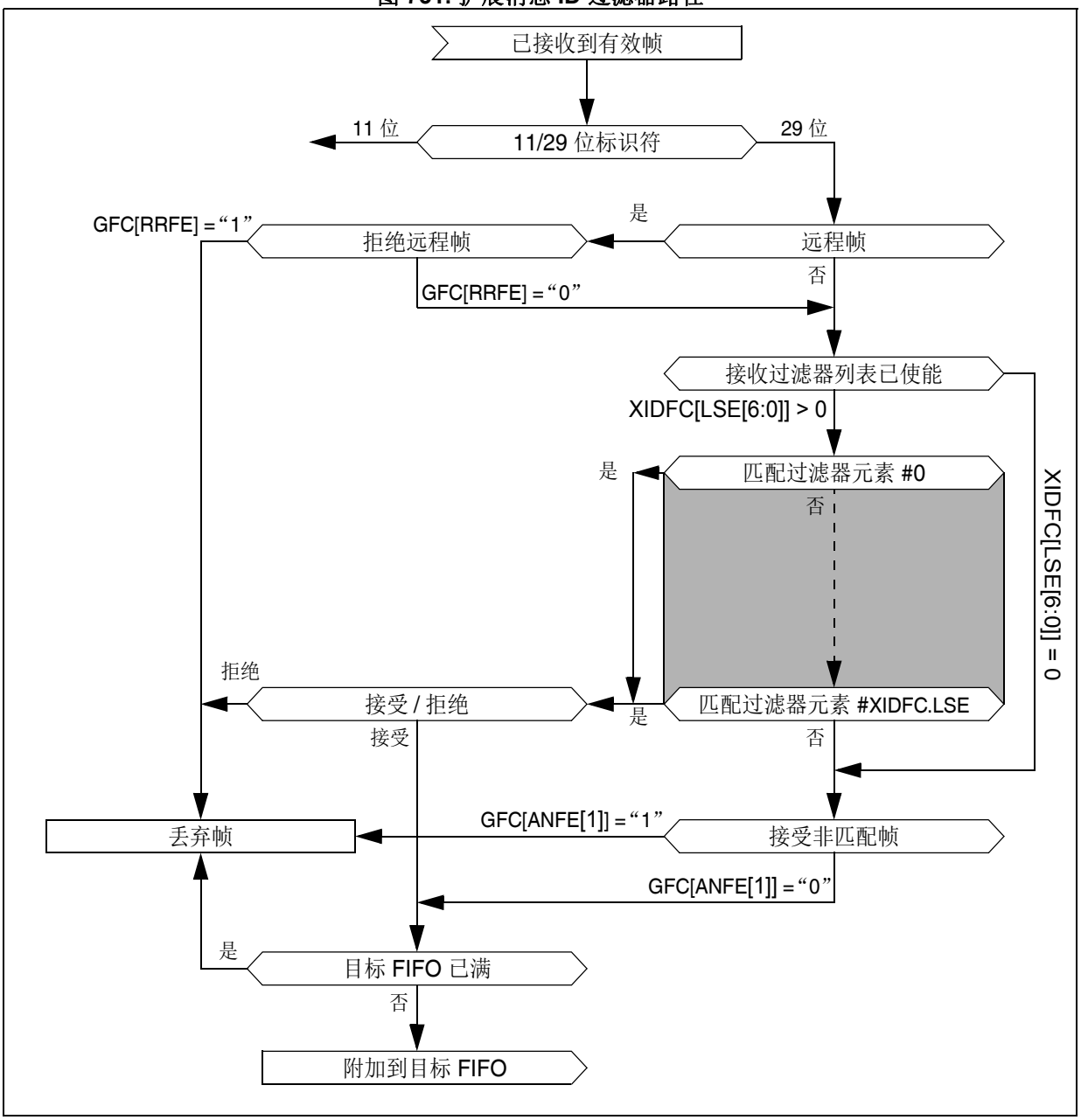


已接收帧的远程发送请求位 (RTR) 和标识符扩展位 (IDE) 会在全局过滤器配置 (GFC) 和标准 ID 过滤器配置 (SIDFC) 消息 ID 的控制下与已配置过滤器元素列表进行比较。

扩展消息 ID 过滤

图 731 介绍了扩展消息 ID（29 位标识符）的过滤流程。*第 2293 页的第 56.3.20 节*中介绍了扩展消息 ID 过滤器元素。

图 731. 扩展消息 ID 过滤器路径



已接收帧的远程发送请求位 (RTR) 和标识符扩展位 (IDE) 会在全局过滤器配置 GFC 和扩展 ID 过滤器配置 XIDFC 消息 ID 的控制下与已配置过滤器元素列表进行比较。

扩展 ID 与掩码 (XIDAM) 会在执行过滤器列表之前与接收到的标识符进行与运算。

接收 FIFO

接收 FIFO 0 和接收 FIFO 1 最多可分别保存 64 个元素。两个接收 FIFO 的配置是通过寄存器 RXF0C 和 RXF1C 完成的。

通过验收过滤的已接收消息会传输到由匹配过滤器元素配置的接收 FIFO 中。有关可用于接收 FIFO 0 和接收 FIFO 1 的过滤器机制说明，请参见*验收过滤器*。*第 56.3.16 节：FDCAN 接收缓冲区和 FIFO 元素*中介绍了接收缓冲区和 FIFO 元素。

当 IR[RFnF] 指示接收 FIFO 已满条件时，在至少已读取一条消息且接收 FIFO 获取索引递增之前，不会继续向相应的接收 FIFO 写入消息。如果在相应的接收 FIFO 已满时收到消息，此消息会被丢弃，中断标志 IR[RFnL] 会置 1。

为了避免接收 FIFO 溢出，可使用接收 FIFO 水印。当接收 FIFO 填充级别达到由 RXFnC[FnWM] 配置的接收 FIFO 水印时，中断标志 IR[RFnW] 会置 1。

从接收 FIFO 中读取消息时，接收 FIFO 获取索引 RXFnS[FnGI] + FIFO 元素大小必须与相应的接收 FIFO 起始地址 RXFnC[FnSA] 相加。

接收 FIFO 阻止模式

接收 FIFO 阻止模式是通过 RXFnC.FnOM = “0” 配置的。该模式是接收 FIFO 的默认工作模式。

达到接收 FIFO 已满条件时 (RXFnS.FnPI = RXFnS.FnGI)，在至少已读取一条消息且接收 FIFO 获取索引递增之前，不会继续向相应的接收 FIFO 写入消息。接收 FIFO 已满条件是通过 RXFnS.FnF = “1” 指示的。此外，中断标志 IR.RFnF 也会置 1。

如果在相应的接收 FIFO 已满时收到消息，此消息会被丢弃，并会通过 RXFnS.RFnL = “1” 指示消息丢失条件。此外，中断标志 IR.RFnL 也会置 1。

接收 FIFO 覆盖模式

接收 FIFO 覆盖模式是通过 RXFnC.FnOM = “1” 配置的。

当 RXFnS.FnF = “1” 指示接收 FIFO 已满条件时 (RXFnS.FnPI = RXFnS.FnGI)，FIFO 接收的下一条消息将覆盖最原始的 FIFO 消息。放入索引和获取索引都会加 1。

如果接收 FIFO 在覆盖模式下工作，并且指示了接收 FIFO 已满条件，则应至少从获取索引 + 1 处开始读取接收 FIFO 元素。这是因为 CPU 从消息 RAM（获取索引）读取消息时可能会将接收到的消息写入消息 RAM（放入索引）。在这种情况下，从相应的接收 FIFO 元素中读取的数据可能不一致。从接收 FIFO 读取数据时向获取索引中添加偏移可避免此问题。偏移取决于 CPU 访问接收 FIFO 的速度。*图 733：混合配置专用发送缓冲区/发送队列示例*显示了读取接收 FIFO 时与放入索引的偏移为 2。在这种情况下，存储在元素 1 和 2 中的两条消息会丢失。

从接收 FIFO 读取数据后，读取的最后一个元素的编号必须写入接收 FIFO 确认索引 RXFnA.FnA 中。这样会使获取索引增加到该元素编号。如果放入索引未增加到该接收 FIFO 元素，接收 FIFO 已满条件会复位 （RXFnS.FnF = “0”）。

专用接收缓冲区

FDCAN 支持多达 64 个专用接收缓冲区。专用接收缓冲区部分的起始地址是通过 RXBC.RBSA 配置的。

对于每个接收缓冲区，必须配置 SFEC/EFEC= “111” 和 SFID2/EFID2[10:9]= “00” 的标准或扩展消息 ID 过滤器元素（请参见*第 56.3.19 节：FDCAN 标准消息 ID 过滤器元素*和*第 56.3.20 节：FDCAN 扩展消息 ID 过滤器元素*）。

接收到的消息被过滤器元素接受后，会存储在过滤器元素引用的消息 RAM 中的接收缓冲区中。存储格式与接收 FIFO 元素的格式相同。此外，中断寄存器中的标志 IR.DRX（专用接收缓冲区中存储的消息）也会置 1。

表 452. 接收缓冲区过滤器配置示例

过滤器元素	SFID1[10:0] EFID1[28:0]	SFID2[10:9] EFID2[10:9]	SFID2[5:0] EFID2[5:0]
0	ID 消息 1	00	00 0000
1	ID 消息 2	00	00 0001
2	ID 消息 3	00	00 0010

匹配接收消息的最后一个字写入消息 RAM 中后，寄存器 NDAT1,2 中相应的新数据标志会置 1。只要新数据标志保持置 1 状态，相应的接收缓冲区就会锁定，以免被接收到的匹配帧更新。新数据标志必须由主机向相应的位位置写入“1”进行复位。

接收缓冲区新数据标志置 1 时，引用该特定接收缓冲区的消息 ID 过滤器元素将不匹配，因此会继续进行验收过滤。以下消息 ID 过滤器元素可能导致接收到的消息存储到另一接收缓冲区或接收 FIFO 中，或者导致消息被拒绝，具体取决于过滤器配置。

接收缓冲区处理

- 复位中断标志 IR.DRX
- 读取新数据寄存器
- 从消息 RAM 中读取消息
- 复位已处理消息的新数据标志

过滤调试消息

要过滤调试消息，需为三个调试消息分别配置一个标准/扩展消息 ID 过滤器元素。要使能过滤器元素过滤调试消息，必须将 SFEC/EFEC 编程为“111”。在这种情况下，SFID1/SFID2 和 EFID1/EFID2 的含义不同。SFID2/EFID2[10:9] 控制着调试消息处理状态机，而 SFID2/EFID2[5:0] 控制着已接收调试消息的存储位置。

存储调试消息时，相应的新数据标志和 IR.DRX 均不会置 1。可通过 RXF1S.DMS 监视调试消息的接收情况。

表 453. 调试消息过滤器配置示例

过滤器元素	SFID1[10:0] EFID1[28:0]	SFID2[10:9] EFID2[10:9]	SFID2[5:0] EFID2[5:0]
0	ID 调试消息 A	01	11 1101
1	ID 调试消息 B	10	11 1110
2	ID 调试消息 C	11	11 1111

发送处理

发送处理单元处理专用发送缓冲区、发送 FIFO 和发送队列的发送请求。它控制着发送消息到 CAN 内核的传输、放入和获取索引以及发送事件 FIFO，最多可为消息发送设置 32 个发送缓冲区（请参见 *专用发送缓冲区*）。根据元素大小 (RXESC) 的配置，会使用 2 到 16 个 32 位字 (Rn = 3 ..17) 来存储 CAN 消息数据字段。

表 454. 帧发送的可能配置

CCCR		发送缓冲区元素		帧发送
BRSE	FDOE	FDF	BRS	
忽略	0	忽略	忽略	典型 CAN
0	1	0	忽略	典型 CAN
0	1	1	忽略	无比特率切换的 FD
1	1	0	忽略	典型 CAN
1	1	1	0	无比特率切换的 FD
1	1	1	1	有比特率切换的 FD

注： *AUTOSAR 至少需要三个发送队列缓冲区，并支持发送取消。*

当发送缓冲区请求挂起寄存器 TXBRP 更新时，或者发送已开始时，发送处理单元会启动发送扫描来检查优先级最高的挂起发送请求（消息 ID 最小的发送缓冲区）。

发送暂停

发送暂停功能用于 CAN 消息标识符（永久性地）被指定为特定值、并且无法轻易更改的 CAN 系统中。这些消息标识符的 CAN 仲裁优先级可能高于其他已定义的消息，而在特定的应用中，其相对仲裁优先级应反转。这样可能会导致如下情况：一个 ECU 发送的 CAN 消息突发造成 CAN 仲裁优先级较低的另一个 ECU CAN 消息被延迟。

举例来说，如果 CAN ECU-1 启用了此功能，并且其应用软件请求发送四条消息，成功发送第一条消息后，将等待两个总线空闲的 CAN 位时间，然后才允许开始发送下一条请求的消息。如果其他 ECU 有挂起的消息，则会在空闲时间开始发送这些消息，不需要与 ECU-1 的下一条消息进行仲裁。接收到消息后，ECU-1 可在接收到的消息释放 CAN 总线后立即开始进行下一次发送。

此功能是通过 CCCR 寄存器中的 TXP 位控制的。如果此位置 1，则每次成功发送消息后，FDCAN 都将暂停两个 CAN 位时间，然后再开始进行下一次发送。这样一来，即使网络中其他节点的优先级标识符较低，也能发送消息。默认禁止此功能（CCCR.TXP = “0”）。

此功能可使来自单个节点的突发发送变得松散一些，避免出现应用程序错误地请求过多传输的“胡乱指示”情况。

专用发送缓冲区

专用发送缓冲区可完全在 CPU 的控制下发送消息。每个专用发送缓冲区都配置了特定的消息 ID。如果多个发送缓冲区配置为使用同一消息 ID，则会先发送缓冲区编号最小的发送缓冲区中的消息。

如果数据部分已更新，则会通过添加请求的 TXBAR[ARn] 位请求发送。请求的消息在内部会与可选发送 FIFO 或发送队列中的消息进行仲裁，在外部会与 CAN 总线上的消息进行仲裁，并会根据其消息 ID 发送出去。

专用发送缓冲区会在消息 RAM 中分配四个 32 位字。因此消息 RAM 中专用发送缓冲区的起始地址是通过将发送缓冲区索引 (0…31) 乘以四再与发送缓冲区起始地址 TXBC[TBSA] 相加计算得出的。

表 455. 发送缓冲区/FIFO - 队列元素大小

TXESC，TBDS[2;0]	数据字段（字节）	元素大小（RAM 字）
000	8	4
001	12	5
010	16	6
011	20	7
100	24	8
101	32	10
110	48	14
111	64	18

发送 FIFO

发送 FIFO 操作是通过将 TXBC[TFQM] 编程为“0”配置的。发送 FIFO 中存储的消息是先从获取索引 TXFQS[TFGI] 引用的消息开始发送的。每次发送后，获取索引会循环递增，直至发送 FIFO 为空。发送 FIFO 可按消息写入发送 FIFO 的顺序发送来自不同发送缓冲区但消息 ID 相同的消息。FDCAN 会计算获取索引和放入索引之差作为发送 FIFO 空闲级别 TXFQS[TFFL]，用于指示可用（空闲）的发送 FIFO 元素数。

新的发送消息必须写入以放入索引 TXFQS[TFQPI] 引用的发送缓冲区开始的发送 FIFO 中。添加请求会将放入索引增加到下一空闲发送 FIFO 元素。放入索引达到获取索引后，会指示发送 FIFO 已满（TXFQS[TFQF]=“1”）。在这种情况下，下一条消息已发送且获取索引已递增之前，不应继续向发送 FIFO 写入消息。

如果有一条消息添加到发送 FIFO，则会通过向与发送 FIFO 放入索引引用的发送缓冲区相关的 TXBAR 位写入“1”来请求发送消息。

如果有多条 (n) 消息添加到发送 FIFO，则会写入以放入索引开始的 n 个连续发送缓冲区中。随后会通过 TXBAR 请求发送。放入索引随后会循环递增 n。请求的发送缓冲区数不应超过发送 FIFO 空闲级别指示的空闲发送缓冲区数。

如果获取索引引用的发送缓冲区的发送请求取消，获取索引会增加到下一个具有挂起发送请求的发送缓冲区，并会重新计算发送 FIFO 空闲级别。如果取消对其他任何发送缓冲区的发送，获取索引和 FIFO 空闲级别保持不变。

发送 FIFO 元素会在消息 RAM 中分配四个 32 位字。因此下一可用（空闲）发送 FIFO 缓冲区的起始地址是通过将放入索引 TXFQS[TFQPI] (0…31) 乘以四再与发送缓冲区起始地址 TXBC[TBSA] 相加计算得出的。

发送队列

发送队列操作是通过将 TXBC[TFQM] 编程为“1”配置的。发送队列中存储的消息是先从消息 ID 最小（优先级最高）的消息开始发送的。如果多个队列缓冲区配置为使用同一消息 ID，则会先发送缓冲区编号最小的队列缓冲区。

新消息必须写入放入索引 TXFQS[TFQPI] 引用的发送缓冲区中。添加请求会将放入索引循环增加到下一空闲发送缓冲区。如果发送队列已满（TXFQS[TFQF]=“1”），则放入索引无效，并且在至少有一个请求的消息已发出或挂起的发送请求已取消之前，不应继续向发送队列写入消息。

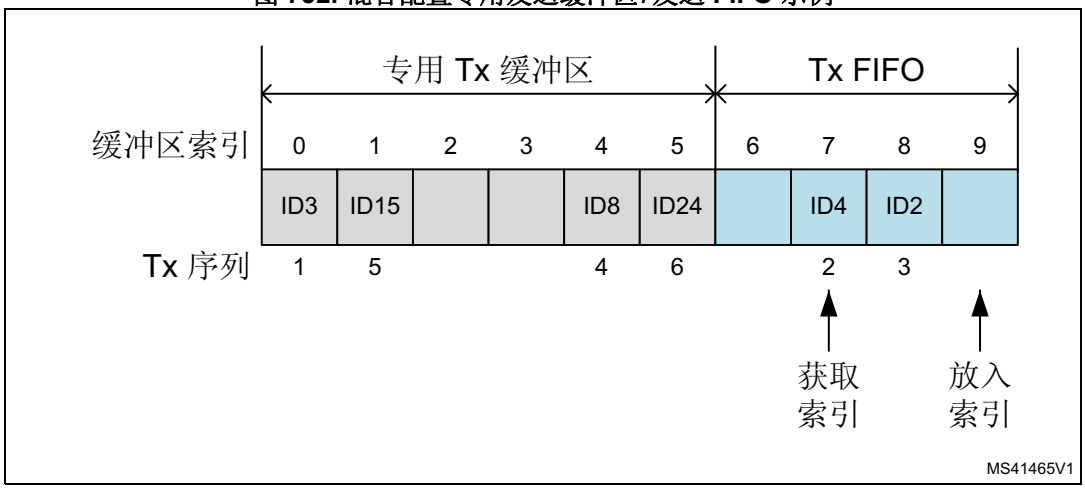
应用可使用寄存器 TXBRP 来代替放入索引，并可将消息放入任何没有挂起传输请求的发送缓冲区中。

发送队列缓冲区会在消息 RAM 中分配四个 32 位字。因此下一可用（空闲）发送队列缓冲区的起始地址是通过将发送队列放入索引 TXFQS[TFQPI] (0…31) 乘以四再与发送缓冲区起始地址 TXBC[TBSA] 相加计算得出的。

混合专用发送缓冲区 / 发送 FIFO

在这种情况下，消息 RAM 中的发送缓冲区部分会被划分为一组专用发送缓冲区和一个发送 FIFO。专用发送缓冲区的数量是通过 TXBC[NDTB] 配置的。分配给发送 FIFO 的发送缓冲区数量是通过 TXBC[TFQS] 配置的。如果 TXBC[TFQS] 编程为 0，则仅会使用专用发送缓冲区。

图 732. 混合配置专用发送缓冲区 / 发送 FIFO 示例



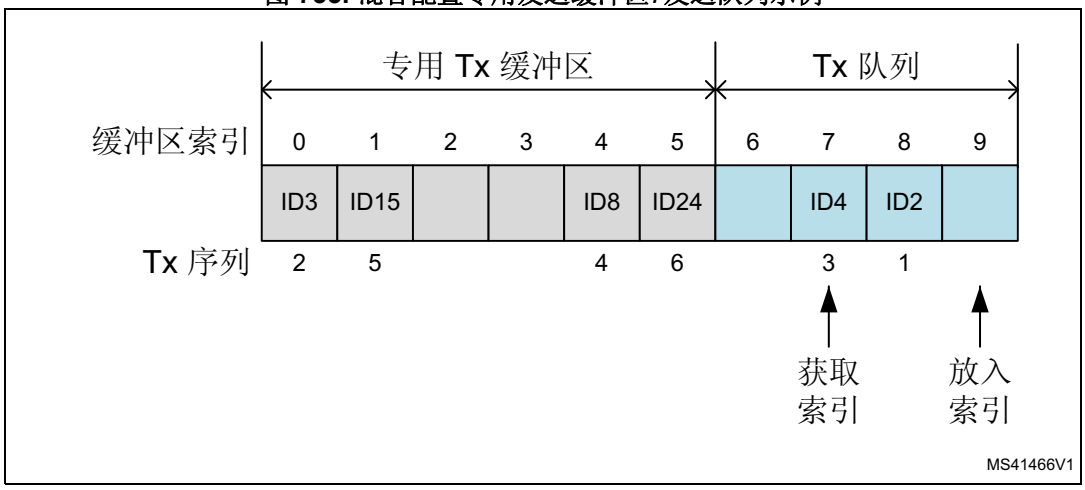
发送优先次序：

- 扫描专用发送缓冲区和时间最久的挂起发送 FIFO 缓冲区（TXFS[TFGI] 引用的缓冲区）
- 消息 ID 最小的缓冲区优先级最高，下次将发送该缓冲区的数据

混合专用发送缓冲区 / 发送队列

在这种情况下，消息 RAM 中的发送缓冲区会被划分为一组专用发送缓冲区和一个发送队列。专用发送缓冲区的数量是通过 TXBC[NDTB] 配置的。发送队列缓冲区的数量是通过 TXBC[TFQS] 配置的。如果 TXBC[TFQS] 编程为 0，则仅会使用专用发送缓冲区。

图 733. 混合配置专用发送缓冲区/发送队列示例



发送优先级设置：

- 扫描所有激活了发送请求的发送缓冲区
- 消息 ID 最小的发送缓冲区优先级最高，下次将发送该缓冲区的数据

发送取消

FDCAN 支持发送取消功能。要取消专用发送缓冲区或发送队列缓冲区请求的发送，主机必须向寄存器 TXBCR 中相应的位位置（= 发送缓冲区的数量）写入“1”。发送取消不适用于发送 FIFO 操作。

成功取消后，寄存器 TXBCF 的相应位会设为“1”。

如果已经在从发送缓冲区进行发送时请求发送取消，则只要发送正在进行中，相应的 TXBRP 位就会保持置 1 状态。如果发送成功，相应的 TXBTO 和 TXBCF 位会置 1。如果发送不成功，则不会重复发送，只会将相应的 TXBCF 位置 1。

注： *如果在挂起的发送尚未开始之前立即取消此次发送，即使该节点中有另一消息挂起，取消后也会有一个短暂的时间窗，在此期间不会开始任何发送。这样一来，另一节点便可以发送优先级可能比该节点中第二条消息低的消息。*

发送事件处理

为了支持发送事件处理，FDCAN 实现了发送事件 FIFO。FDCAN 在 CAN 总线上发送消息后，消息 ID 和时间戳会存储在发送事件 FIFO 元素中。为了将发送事件关联到发送事件 FIFO 元素，已发送的发送缓冲区中的消息标志会被复制到发送事件 FIFO 元素中。

发送事件 FIFO 最多可配置为 32 个元素。*发送 FIFO* 中介绍了发送事件 FIFO 元素。根据元素大小 (TXESC) 的配置，会使用 2 到 16 个 32 位字 (Tn = 3 ..17) 来存储 CAN 消息数据字段。

发送事件 FIFO 的用途是将处理发送状态信息与处理发送消息分开，也就是让发送缓冲区仅保存要发送的消息，而将发送状态单独存储在发送事件 FIFO 中。这样做有很大的优势，尤其是在处理动态管理的发送队列时，发送缓冲区可在发送成功后立即用于新消息。覆盖发送缓冲区之前，不需要保存该发送缓冲区的发送状态信息。

当 IR[TEFF] 指示发送事件 FIFO 已满条件时，在至少已读取一个元素且发送事件 FIFO 获取索引递增之前，不会继续向发送事件 FIFO 写入元素。如果在发送事件 FIFO 已满时发生发送事件，此事件会被丢弃，中断标志 IR[TEFL] 会置 1。

为了避免发送事件 FIFO 溢出，可使用发送事件 FIFO 水印。当发送事件 FIFO 填充级别达到由 TXEFC[EFWM] 配置的发送事件 FIFO 水印时，中断标志 IR[TEFW] 会置 1。

从发送事件 FIFO 读取数据时，必须将发送事件 FIFO 获取索引 TXEFS[EFGI] 乘以二再与发送事件 FIFO 起始地址 TXEFC[EFSA] 相加。

56.3.3 FIFO 确认处理

接收 FIFO 0、接收 FIFO 1 和发送事件 FIFO 是通过向相应的 FIFO 确认索引进行写操作控制的，请参见*第 56.4.28 节：FDCAN 接收 FIFO 0 确认寄存器 (FDCAN_RXF0A)*、*第 56.4.32 节：FDCAN 接收 FIFO 1 确认寄存器 (FDCAN_RXF1A)* 和*第 56.4.44 节：FDCAN 发送事件 FIFO 配置寄存器 (FDCAN_TXEFC)*。对 FIFO 确认索引进行写操作会将 FIFO 获取索引设置为 FIFO 确认索引加 1，进而会更新 FIFO 填充级别。有两种用例：

1. 如果只从 FIFO 读取了一个元素（获取索引指向的元素），则该获取索引值会写入 FIFO 确认索引中。
2. 如果已从 FIFO 中读取一系列元素，则仅在该读取序列结束时写入一次 FIFO 确认索引（数值：读取的最后一个元素的索引）即可更新 FIFO 获取索引。

由于 CPU 可自由访问 FDCAN 消息 RAM，以任意顺序（不考虑获取索引）读取 FIFO 元素时需要特别留意。这一点在从两个接收 FIFO 之一读取高优先级消息时会很有用。在这种情况下，不应写入 FIFO 确认索引，否则会将获取索引设为错误的位置，还会改变 FIFO 填充级别。在这种情况下，一些时间较久的 FIFO 元素会丢失。

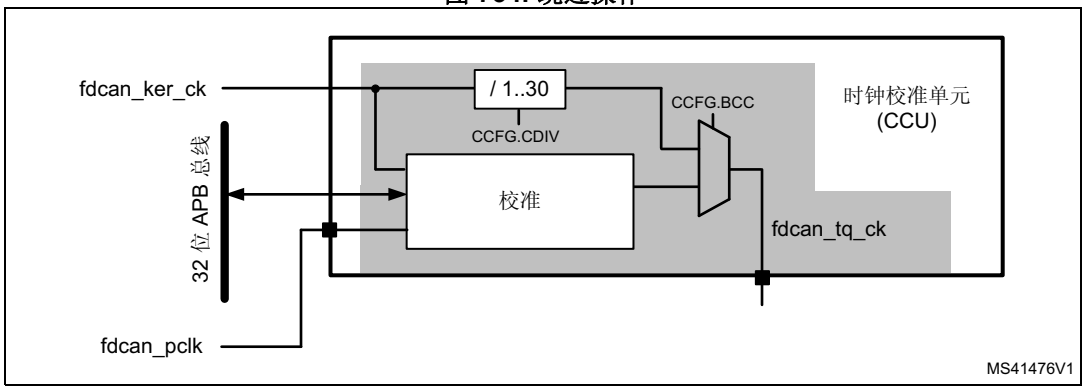
注： *应用必须确保 FIFO 确认索引中已写入有效值。FDCAN 不会检查数值是否正确。*

56.3.4 CAN 时钟校准

器件复位后，时钟校准单元 (CCU) 不会向 FDCAN1 和 FDCAN2 提供有效的时钟信号。必须通过 CCFG 寄存器对 CCU 进行初始化。仅当 FDCAN1 的 CCCR.CCE 和 CCCR.INIT 位均置 1 时，才能向 CCFG 寄存器写入数据。因此，需要先完成 CCU 和 FDCAN1 初始化，任何 FDCAN1 和/或 FDCAN2 模块才能够工作。

当 CCFG.BCC = “1” 时，会绕过时钟校准（请参见*图 734*）。

图 734. 绕过操作



工作条件

CAN 时钟校准单元的工作条件如下：

- CAN 内核时钟频率 fdcan_ker_ck 在 80 MHz 与 500 MHz 之间
- FDCAN 比特率在 125 Kb/s 与 1 Mb/s 之间

FDCAN 时钟校准单元会生成介于 0.5 Mhz 和 25 MHz 范围内的经校准时间片时钟 fdcan_tq_ck。

注： *FDCAN 要求 CAN 时间片时钟始终小于或等于 APB 时钟 (fdcan_tq_ck < fdcan_pclk)。绕过 CAN 时钟校准单元时（CCFG.BCC = “1”），必须考虑这一点。*

校准精度

Precision_Calibrated 状态下的校准精度取决于下列因素。

- CAN 内核时钟输入 fdcan_ker_ck 的动态时钟容差。
- 测量误差。对于用于校准测量的每个位序列，存在最大一个 fdcan_pclk 周期的误差。位时间测量使用的位数为 32 位或 64 位，具体取决于 CCFG.CFL 的配置。
- 校准机制中可容忍的误差。

选择校准消息之间的距离时，必须满足 FDCAN1 模块的时钟容差要求。

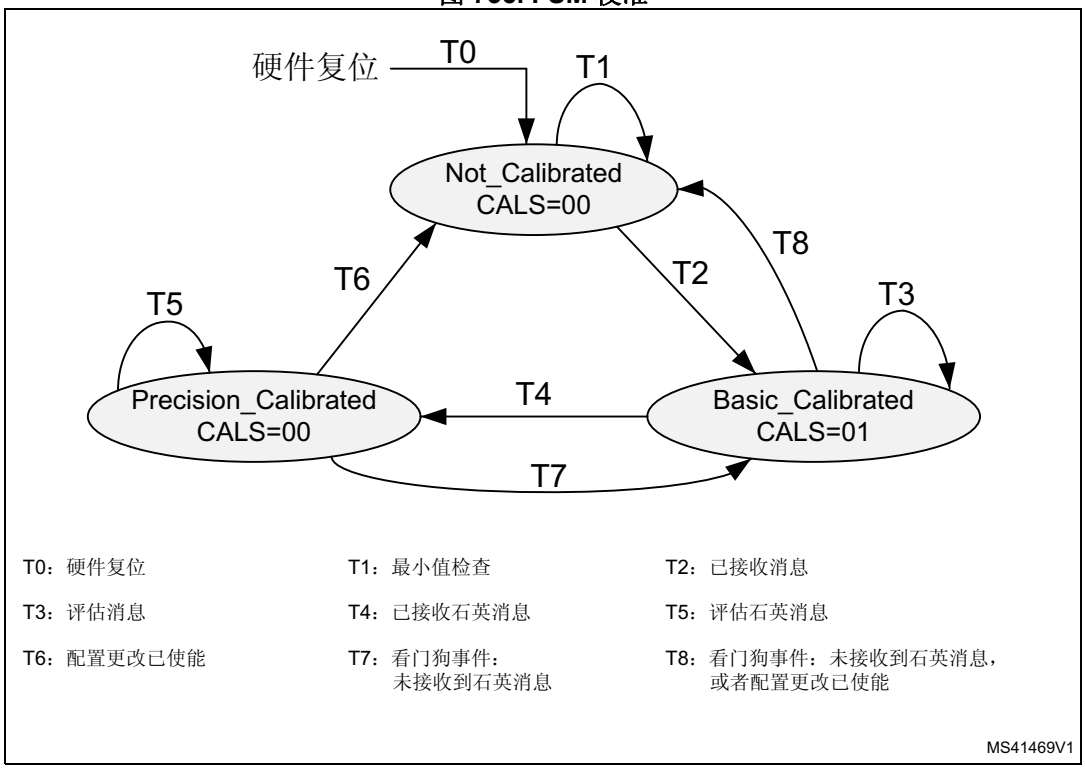
注： *动态时钟容差是因温度或工作电压变化等原因造成的两条校准消息之间的时钟频率变化。*

功能说明

通过 CAN 消息执行时间片时钟 fdcan_tq_ck 校准时，应调整用于通过时钟 fdcan_ker_ck 生成 CAN 协议时间片 tq 的时钟分频器。

1. 第一步：基本校准
   测量从隐性到显性的两个边沿之间的最短距离，该时间将假定为两个 CAN 位时间，以 PLL 时钟周期计数。每次新测量发现两个边沿间的距离变短时，都会更新时钟分频器。当 CAN 协议控制器检测到有效 CAN 消息时，基本校准完成。
2. 第二步：精确校准
   校准状态机通过计算 fdcan_ker_ck 周期数来测量 CAN 帧中较长的位序列长度。该位序列长度可通过 CCFG.CLF 配置为 32 位或 64 位。如果校准字段长度为 32 位/64 位，需要使用数据字段至少为 2/6 字节的校准消息。精确校准基于通过测量更长的位序列计算出的新时钟分频器值。

图 735. FSM 校准



校准状态改变还会将中断标志 CUIR.CSC 置 1。如果中断线 cu_int 通过 CUIE.CSCE 使能，则会被激活（设为高电平）。中断标志 CUIR.CSC 复位之前，中断线 cu_int 会保持有效状态。

在精确校准完成之前，FDCAN1 和 FDCAN2 会在受限模式下工作（不发送帧、不发送错误或过载标志、不计算错误数量）。如果 PLL 校准是通过软件评估寄存器 CSTAT 中的校准状态来完成的，则 FDCAN1 和 FDCAN2 必须设为受限工作模式（CCCR.ASM = “1”），直到 CAN 校准单元进入 Precision_Calibrated 状态（请参见*应用*）。

只能对使用石英控制的稳定时钟的节点所发送的有效 CAN 帧执行精确校准。校准帧是通过 FDCAN1 验收过滤检测的。必须在 FDCAN1 中配置过滤器元素和接收缓冲区，以识别和存储校准消息。接收到校准消息后，必须复位接收缓冲区的新数据标志，从而可指示下一条校准消息。

如果网络中只有一个使用石英时钟的 CAN 发送器，则该节点必须在通过数据字段或标识符中至少一个“1010”序列启动后发送其第一条消息。这样可确保非石英节点能够进入 Basic_Calibrated 状态，随后确认石英节点消息。

必须在由校准看门狗监视的预定义最大时间间隔内重复执行精确校准。

注： *当 CAN 时钟校准单元从 Precision_Calibrated 状态切换回 Basic_Calibrated 状态时，会禁止校准正常信号，FDCAN1 会完成当前进行的发送过程，然后进入受限工作模式（不发送帧、不发送错误或过载标志、不计算错误数量）。*

配置

CAN 时钟校准单元是通过寄存器 CCFG 配置的，也就在 FDCAN1 将 CCCR.CCE 和 CCCR.INIT 位置 1。

对于基本校准，将会测量引脚 FDCAN1_RX 上两个连续下降沿之间的最小振荡器周期数。时钟周期数取决于对输入 fdcan_ker_ck 应用的时钟频率。如果测得的时钟周期数因 FDCAN1_RX 上存在毛刺信号等原因小于由 CCFG.OCPM 配置的最小值，该值会被丢弃，测量会继续进行。

建议将 CCFG.OCPM 配置为略小于两个 CAN 位时间：

CCFG.OCPM < ((2 个 CAN 位时间) / fdcan_ker_ck 周期) / 32

精确校准使用的位域的长度可通过 CCFG.CFL 配置为 32 位或 64 位。精确校准使用的位数会影响校准精确度以及两条校准消息之间的最大距离。

通过 CCFG.TQBT 配置的每个位时间的时间片数与测得的振荡器时钟周期 CSTAT.OCPC 数共同定义每个位时间的振荡器时钟数。

如果通过配置 CCFG.BCC =“1”绕过时钟校准，则需要通过 CCFG.CDIV 配置内部时钟分频器，以满足 fdcan_tq_ck < fdcan_pclk 的条件。

注： *当 CAN 时钟校准有效时（CCFG.BCC =“0”），必须将 FDCAN1 和 FDCAN2 的波特率预分频器配置为无效。*

状态指示

CAN 时钟校准单元的状态可通过读取寄存器 CSTAT 进行监视。处于 Precision_Calibrated 时，CSTAT.OCPC 会指示校准字段中的振荡器时钟周期数，而 CSTAT.TQC 则会指示校准字段中的时间片数。

校准状态是通过 CSTAT.CALS 进行监视的。cu_csc 脉冲的持续时间为一个 cu_hclk 周期。校准状态改变还会将中断标志 CUIR.CSC 置 1。如果中断线 cu_int 通过 CUIE.CSCE 使能，则会被激活（设为高电平）。中断标志 CUIR.CSC 复位之前，中断线 cu_int 会保持有效状态。

校准看门狗事件还会将中断标志 CUIR.CWE 置 1。如果中断线 cu_int 通过 CUIE.CWEE 使能，则会被激活（设为高电平）。中断标志 CUIR.CWE 复位之前，中断线 cu_int 会保持有效状态。

应用

绕过时钟校准

CCU 内部时钟分频器配置为除以 1（CCFG.CDIV =“0000”）。CCU 输出信号 cu_cok 固定为“1”。在该工作模式下，输入时钟 fdcan_ker_ck 会直接发送到时钟输出 fdcan_tq_ck。在这种情况下，fdcan_tq_ck 与连接到 CCU 的 FDCAN1 和 FDCAN2 的配置和状态无关。如果 fdcan_ker_ck 为 20/40/80 MHz，可执行 CAN FD 操作。

注： *CCFG.BCC 的复位值通过通用参数在合成时配置为“1”，此模式为复位后的默认工作模式。*

软件校准

CAN 时钟校准单元还支持通过软件调整片上振荡器的方式对 fdcan_ker_ck 进行校准。要计算调整值，主机必须从 CSTAT 读取 CCU 状态。在该工作模式下，CCU 输出信号 cu_cok 固定为“1”，来自 fdcan_ker_ck 的时钟会发送到输出 fdcan_tq_ck（CCFG.BCC =“1”）。

输入时钟 fdcan_ker_ck 必须在 80 MHz 到 500 MHz 范围内。必须通过 CCFG.CDIV 配置 CCU 的时钟分频器，以便将 fdcan_tq_ck 调整到有效范围内。所有其他配置参数都必须通过 CCFG 进行设置。为了使 tFDCAN1 和 FDCAN2 正常工作，APB 时钟 fdcan_pclk 需要等于或大于时间片时钟 (fdcan_tq_ck)。不能进行 CAN FD 操作。

启动时，FDCAN1 和 FDCAN2 必须在 CCCR.INIT 复位之前均配置为受限工作模式（CCCR.ASM =“1”）。必须调整输入时钟 fdcan_ker_ck，直到 CAN 时钟校准单元达到 Precision_Calibrated 状态。现在，软件可以复位 CCCR.ASM，CANFD1 和 CANFD2 可以开始正常工作。

工作期间，软件必须定期检查 CAN 时钟校准单元是否仍处于 Precision_Calibrated 状态。如果 CAN 时钟校准单元已因 fdcan_ker_ck 偏移而退出 Precision_Calibrated 状态，则必须将 CCCR.INIT、CCCR.CCE 和 CCCR.ASM 编程为“1”，从而使 FDCAN1 和 FDCAN2 进入受限工作模式。成功调整 fdcan_ker_ck 后（CAN 时钟校准单元处于 Precision_Calibrated 状态），FDCAN1 和 FDCAN2 可恢复正常工作。

注： *调整精度需要满足已配置比特率的 CAN 时钟容差要求。*

时钟校准有效

将 CCFG.BCC 复位为“0”可进入此工作模式。在该工作模式下，CCU 输出信号 cu_cok 和 fdcan_ker_ck 是由 CCU 控制的。如果 CCU 不处于 Precision_Calibrated 状态，cu_cok 为“0”。

CCU 输出信号 fdcan_tq_ck 和 cu_cok 的生成取决于 FDCAN1 的状态。输入时钟 fdcan_ker_ck 必须在 80 Mhz 到 500 MHz 范围内。需要配置 CCU 和 FDCAN1。不能进行 CAN FD 操作。

如果 FDCAN1 变为 Bus_Off 状态，或者其 INIT 位通过主机命令置 1（CCCR.INIT =“1”），CCU 会进入 Not_Calibrated 状态，输出 cu_cok 会复位为“0”。CANFD1 和 CANFD2 进入受限工作模式。

注： *CCFG.BCC 的复位值通过通用参数在合成时配置为“0”，此模式为复位后的默认工作模式。*

56.3.5 TTCAN 操作（仅限 FDCAN1）

参考消息

参考消息是具有特定 CAN 标识符的数据帧。该消息会被除时间主控节点（参考消息发送方）以外的所有节点接收和接受。

对于 1 级参考消息，数据长度必须至少为 1；对于 0 级和 2 级参考消息，数据长度必须至少为 4；否则消息不会作为参考消息被接受。参考消息可通过其他数据最多扩展为总共八个 CAN 数据字节。标识符中除三个 LSB 以外的所有位会将消息定性为参考消息。最后三位指定最多八个潜在时间主控节点的优先级。保留的位会以逻辑 0 的形式发送，接收器会忽略这些位。参考消息是通过寄存器 TTRMC 配置的。

时间主控节点会发送参考消息。如果参考消息受到错误的干扰，则会立即重发。重发时，已发送的 Master_Ref_Mark 会更新。参考消息会定期发送，但允许停止定期发送（Next_is_Gap 位）并在下一基本周期开始时通过当前时间主控节点或其他潜在时间主控节点之一发起事件同步发送。

发送参考消息的节点是当前时间主控节点。时间主控节点可以发送其他消息。如果当前时间主控节点发生故障，优先级最高的潜在时间主控节点会代替它执行操作。如果某个节点既不是时间主控节点，也不是潜在时间主控节点，则该节点为时间接收节点。

1 级

1 级操作是通过 TTOCF[OM] =“01”以及 TTOCF[GEN] 配置的。1 级参考消息中不能进行外部时钟同步。与参考消息相关的信息存储在第一个数据字节中，如下图所示。Cycle_Count 为可选位。

表 456. 1 级参考消息的第一个字节

位	0	1	2	3	4	5	6	7
第一个字节	Next_is_Gap	保留	Cycle_Count[5;0]					

2 级

2 级操作是通过 TTOCF[OM] =“10”和 TTOCF[GEN] 配置的。与参考消息相关的信息存储在前四个数据字节中，如下图所示。Cycle_Count 和 NTU_Res 的低四位为可选位。TTCAN 不会对接收到的参考消息中的 NTU_Res[3:0] 进行评估，始终会将这些位作为 0 发送。

表 457. 2 级参考消息的前四个字节

位	0	1	2	3	4	5	6	7
第一个字节	Next_is_Gap	保留	Cycle_Count[5;0]					
第二个字节	NTU_Res[6:4]			NTU_Res[3:0]				Disc_Bit
第三个字节	Master_Ref_Mark[7:0]							
第四个字节	Master_Ref_Mark[15:8]							

0 级

0 级操作是通过 TTOCF[OM] = “11” 配置的。0 级参考消息中不能实现外部事件同步时间触发操作。与参考消息相关的信息存储在前四个数据字节中，如下表所示。在 0 级参考消息中，Next_is_Gap 始终为 0。Cycle_Count 和 NTU_Res 的低四位为可选位。TTCAN 不会对接收到的参考消息中的 NTU_Res[3:0] 进行评估，始终会将这些位作为 “0” 发送。

表 458. 0 级参考消息的前四个字节

位	0	1	2	3	4	5	6	7
第一个字节	Next_is_Gap	保留	Cycle_Count[5;0]					
第二个字节	NTU_Res[6:4]			NTU_Res[3:0]				Disc_Bit
第三个字节	Master_Ref_Mark[7:0]							
第四个字节	Master_Ref_Mark[15:8]							

56.3.6 TTCAN 配置

TTCAN 定时

网络时间单位 (NTU) 是所有时间的测量单位。NTU 在整个网络内是恒定不变的，被网络系统设计人员定义为优先项。在 TTCAN 1 级参考消息中，NTU 是标称 CAN 位时间。在 TTCAN 0 级和 2 级参考消息中，NTU 是零点几个物理秒。

NTU 是本地时间的时基。每经过一个 NTU，本地时间的整数部分（16 位值）会递增 1 次。周期时间和全局时间均源自本地时间。本地时间、周期时间和全局时间的小数部分（3 位值）不可读。

在 TTCAN 0 级和 2 级参考消息中，NTU 的长度是由时间单位比 TUR 定义的。TUR 通常为非整数，计算公式为：TUR = TURNA[NAV] / TURCF[DC]。NTU 长度的计算公式为：NTU = CAN 时钟周期 x TUR。

TUR 分子配置 NC 为 18 位数，TURCF[NCL[15:0]] 可设定在 0x0000 - 0xFFFF 范围内。TURCF[NCH[17:16]] 通过硬接线的方式写入 0b01。如果将 0xnnnn 写入 TURCF[NCL[15:0]]，TURNA[NAV] 的起始值为 0x10000 + 0x0nnnn = 0x1nnnn。TUR 分母配置 TURCF[DC] 为 14 位数。TURCF[DC] 可设定在 0x0001 - 0x3FFF 范围内（0x0000 为非法值）。

在 1 级参考消息中，NC 必须 ≥ 4 x TURCF[DC]。在 0 级和 2 级参考消息中，要使 NTU 内部小数部分达到 3 位分辨率，NC 必须 ≥ 8 x TURCF[DC]。

硬件复位会将 TURCF[DC] 预设为 0x1000，将 TURCF[NCL] 预设为 0x10000，因此 NTU 会包含 16 个 CAN 时钟周期。在 CCCR[INIT] 复位或 TURCF[ELT] 置 1 之前，本地时间和应用看门狗都不会启动。配置 NTU 之前，TURCF[ELT] 不会置 1。将 TURCF[ELT] 置 “1” 还会锁定对寄存器 TURCF 的写访问。

启动时，如果 TURCF[ELT] 置 1，TURNA[NAV] 会从 NC (= TURCF[NCL] + 0x10000) 进行更新。TTCAN 1 级参考消息中没有偏移补偿。TURNA.NAV 在工作期间不会改变，始终等于 NC。

在 TTCAN 0 级和 2 级参考消息中，TURNA[NAV] 会在两种情况下发生改变。作为时间被控节点或备份时间主控节点工作且 TTOCF[ECC] 位置 1 时，只要 TTCAN 处于同步状态 In_Schedule 或 In_Gap，TURNA[NAV] 就会自动更新为通过监视的全局时间速度计算出的值。如果同步丢失，则会恢复为 NC。作为实际时间主控节点工作且 TTOCF[EECS] 置 1 时，主机可更新 TURCF[NCL]。当主机将 TTOCN[ECS] 置 1 时，TURNA[NAV] 将从下一参考消息的 NC 新值进行更新。状态标志 TTOST[WECS] 会在 TTOCN[ECS] 置 1 时置 1，并会在 TURNA[NAV] 更新时清零。当 TTOST[WECS] 置 1 时，TURCF[NCL] 会锁定，无法执行写操作。

在 TTCAN 0 级和 2 级参考消息中，时钟校准过程会在 NC+/- 2(TTOCF[LDSDL]+5) 的同步偏差限值范围 SDL 内调整 TURNA[NAV]。TURCF[NCL] 应编程为最大的适用数字值，以便在计算 TURNA[NAV] 时达到最佳精度。

同步偏差 SD 是 NC 与 TURNA[NAV] 之差 (SD = |NC - TURNA[NAV]|)。该值受同步偏差限值 SDL 的限制，SDL 是通过其双对数 TTOCF[LDSDL] (SDL = 2(TTOCF[LDSDL]+5)) 配置的，不应超过 CAN 位定时配置提供的时钟容差。每个新的基本周期都会计算 SD。如果计算出的 TURNA[NAV] 与 NC 的偏差大于 SDL，或者参考消息中的 Disc_Bit 置 1，则偏移补偿会暂停，TTIR[GTE] 会置 1，TTOSC[QCS] 会复位，或者在 Disc_Bit = “1” 的情况下，TTIR[GTD] 会置 1。

表 459. TUR 配置示例

TUR	8	10	24	50	510	125000	32.5	100/12	529/17
NC	0x1FFF8	0x1FFFE	0x1FFF8	0x1FFEA	0x1FFFE	0x1E848	0x1FFE0	0x19000	0x10880
TURCF.DC	0x3FFF	0x3333	0x1555	0x0A3D	0x0101	0x0001	0x0FC0	0x3000	0x0880

TTOCN[ECS] 通过下一条参考消息调度 NC 进行激活。TTOCN[SGT] 通过下一条参考消息调度 TTGTP[TP] 进行激活。当 FDCAN 为实际时间主控节点时，设置 TTOCN[ECS] 和 TTOCN[SGT] 时需要将 TTOCF[EECS] 置 1（使能外部时钟同步）。

TTCAN 模块提供应用看门狗来验证应用程序的功能。主机必须定期处理该看门狗，否则所有 CAN 总线活动都会停止。应用看门狗限值 TTOCF[AWL] 指定了两次处理看门狗的时间间隔（以 NTU 数为单位）。NTU 数最大值为 256。通过读取寄存器 TTOST 来处理应用看门狗。TTOST[AWE] 指示是否已及时处理看门狗。如果应用无法处理应用看门狗，中断标志 TTIR[AW] 将置 1。进行软件开发时，可通过将 TTOCF[AWL] 编程为 0x00 禁止应用看门狗，请参见*第 56.4.49 节：FDCAN TT 操作配置寄存器 (FDCAN_TTOCF)*。

接口信号定时

在输出 FDCAN 触发时间标记中断脉冲 m_ttcan_tmp 和 FDCAN 寄存器时间标记中断脉冲 m_ttcan_rtp 上产生脉冲的定时事件在 CAN 时钟域中生成。需考虑在 APB 时钟域中出现这一事件之前（TTIR[TTMI] 置 1 或 TTIR[RTMI] 置 1）使用一个时钟域跨越延迟。举例来说，信号可连接至另一 FDCAN 节点的定时输入 (fdcan_swt/fdcan_evt)，以便自动同步两个 TTCAN 网络。

参考消息完成（发送或接收）时，输出 FDCAN 周期开始脉冲 m_ttcan_soc 会激活。输出在 APB 时钟域中进行控制。

56.3.7 消息调度

TTOCF[TM] 控制着 TTCAN 作为潜在时间主控节点还是时间被控节点工作。如果是潜在时间主控节点，参考消息标识符 TTRMC[RID] 的三个 LSB 定义主控节点优先级，0 代表最高优先级，7 代表最低优先级。网络中不能存在两个使用相同主控优先级的节点。TTRMC[RID] 用于识别参考消息。TTRMC[RMPS] 与时间被控节点无关。

初始参考触发偏移 TTOCF[IRTO] 是 7 位值，定义了在预期发送参考消息、但总线仍处于空闲状态时，备份时间主控节点等待多久（以 NTU 数为单位）后才会开始发送参考消息。TTOCF[IRTO] 的建议值为主控节点优先级乘以相应的系数，具体系数值取决于网络中潜在时间主控节点之间的预期时钟偏移。在当前时间主控节点发生故障的情况下，各备份时间主控节点启动参考消息的顺序应与其主控节点优先级相对应，即使时钟偏移达到最大值时也不例外。

TTOCF[OM] 决定了节点在 TTCAN 0 级、1 级还是 2 级中工作。在一个网络中，所有潜在时间主控节点都必须在同一级别工作。时间被控节点可工作在 2 级网络中的 1 级，但反过来不可以。最后的设置步骤是通过 TTOCF[OM] 配置 TTCAN 工作模式。如果 TTOCF[OM] = “00”（事件驱动 CAN 通信），FDCAN 会按照 ISO 11898-1: 2015 的规定工作，没有时间触发。如果 TTOCF[OM] = “01”（1 级），FDCAN 会按照 ISO 11898-4 的规定工作，但不能将基本周期同步到外部事件，参考消息中的 Next_is_Gap 位会被忽略。如果 TTOCF[OM] = “10”（2 级），TTCAN 会按照 ISO 11898-4 的规定工作，包括基本周期的事件同步启动。如果 TTOCF[OM] = “11”（0 级），FDCAN 会作为事件驱动 CAN 工作，但会保留 2 级中已校准的全局时基。

TTOCF[EECS] 可实现外部时钟同步，从而允许当前时间主控节点的应用程序在时间触发操作期间更新 TUR 配置，根据外部参考值调整时钟速度和（仅限 0 级和 2 级中）全局时钟相位。

TT 矩阵限值寄存器中的 TTMLM[ENTT] 指定了系统矩阵中预期的 Tx_Trigger 数。该值为独占、单独仲裁和合并仲裁窗口对应的 Tx_Trigger 之和，不包括 Tx_Ref_Trigger。需要注意的是，该值通常不是 Tx_Trigger 存储器元素数；必须考虑系统矩阵中的基本周期数和触发重复系数。如果 TTMLM[ENTT] 的配置不准确，则会导致发送计数下溢（TTIR[TXU] = “1”且 TTOST[EL] = “01”，严重程度 1）或者发送计数上溢（TTIR[TXO] = “1”且 TTOST[EL] = “10”，严重程度 2）。

注： *如果节点观察到的第一条参考消息没有 Cycle_Count 0，则该节点可能会使用其发送计数完成第一个矩阵周期，从而会造成发送计数下溢条件。只要节点处于同步状态，其 Tx_Trigger 就不会触发发送。*

TTMLM[CCM] 指定系统矩阵中最后一个基本周期的编号。基本周期计数从 0 开始。如果系统矩阵包含 8 个基本周期，TTMLM[CCM] 将为 7。TTMLM[CCM] 会被时间被控节点忽略，参考消息的接收器会将接收到的周期计数视为实际基本周期的有效周期计数。

TTMLM[TXEW] 指定发送使能窗口的长度（以 NTU 数为单位）。发送使能窗口是可开始发送的时间窗口开始处的时间段。如果由于与之前的时间窗口消息稍有重叠等原因而无法在发送使能窗口内启动消息发送，则无法在该时间窗口内启动发送。选择 TTMLM[TXEW] 时必须考虑网络同步质量以及时间窗口长度与消息长度之间的关系。

触发存储器

触发存储器是 TTCAN 连接到的外部消息 RAM 的一部分（请参见*第 56.4.26 节：FDCAN 接收 FIFO 0 配置寄存器 (FDCAN_RXF0C)*），最多存储 64 个触发元素。触发存储器元素包括时间标记 TM、周期代码 CC、触发类型 TYPE、过滤器类型 FTYPE、消息编号 MNR、消息状态计数 MSC、时间标记事件内部 TMIN、时间标记事件外部 TMEX（请参见*第 56.3.21 节：FDCAN 触发存储器元素*）。

时间标记定义了触发在哪一周期时间生效。触发存储器中的触发必须按其时间标记进行排序。时间标记最小的触发元素会写入第一个触发存储器字中。对于类型为 Tx_Ref_Trigger、Tx_Ref_Trigger_Gap、Watch_Trigger、Watch_Trigger_Gap 和 End_of_List 的触发，会忽略消息编号和周期代码。

如果周期时间达到实际触发的时间标记，FSE 会切换到下一触发，并开始从触发存储器中读取以下触发。如果是发送触发，一旦 FSE 切换到其触发，发送处理单元就会开始从消息 RAM 中读取消息。RAM 访问速度定义了发送触发与它之前的触发之间的最短时间步，发送处理单元必须能够在达到发送触发时间标记之前做好发送准备。RAM 访问速度还限制了两个匹配触发之间的非匹配（对于其周期代码）触发数，必须在达到其时间标记之前读取下一匹配触发。如果参考消息长度为 n 个 NTU，则时间标记小于 n 的触发将永不会激活，并将被视为配置错误处理。

周期时间的起始点是参考消息帧起始位的采样点。当周期时间达到 Tx_Ref_Trigger 时间标记时，会请求下一条参考消息。FDCAN 会在下一采样点对发送请求作出响应。会在帧起始位捕获新的 Sync_Mark，但周期时间会增加，直到参考消息成功发送（或接收），Sync_Mark 被当作新的 Ref_Mark。此时，周期时间会重新开始。因此，周期时间值绝不会小于 n（初始化时例外），n 表示参考消息的长度（以 NTU 数为单位）。

基本周期长度：Tx_Ref_Trigger 时间标记 + 1 个 NTU + 1 个 CAN 位时间。

FDCAN 网络中所有节点的触发列表都是不同的。每个节点只知道自己的发送消息对应的 Tx_Trigger、自己处理的接收消息对应的 Rx_Trigger 以及与参考消息相关的触发。

触发类型

Tx_Ref_Trigger （TYPE = “0000”）和 Tx_Ref_Trigger_Gap （TYPE = “0001”）会触发通过时间主控节点发送参考消息。如果时间被控节点在其触发存储器中遇到 Tx_Ref_Trigger(_Gap)，则会检测到配置错误（TTOST[EL] = “11”，严重程度 3）。Tx_Ref_Trigger_Gap 仅用于外部事件同步时间触发工作模式。在该模式下，当 FDCAN 同步状态为 In_Gap（TTOST[SYS] = “10”）时，会忽略 Tx_Ref_Trigger。

Tx_Trigger_Single）TYPE = “0010”）、Tx_Trigger_Continuous（TYPE = “0011”）、Tx_Trigger_Arbitration（TYPE = “0100”）和 Tx_Trigger_Merged（TYPE = “0101”）会触发启动发送过程。这些触发类型定义了时间窗口起点。

如果消息缓冲区发送请求挂起位置 1，Tx_Trigger_Single 会在独占时间窗口中启动单次发送。成功发送后，发送请求挂起位会复位。

如果消息缓冲区发送请求挂起位置 1，Tx_Trigger_Continuous 会在独占时间窗口中启动发送。成功发送后，发送请求挂起位保持置 1 状态，并且消息缓冲区会在下一匹配的时间窗口中再次发送。

Tx_Trigger_Arbitration 启动仲裁时间窗口，Tx_Trigger_Merged 启动合并仲裁时间窗口。合并仲裁时间窗口的最后一个 Tx_Trigger 的类型必须为 Tx_Trigger_Arbitration。如果类型为 Tx_Trigger_Merged 的触发之后不是类型为 Tx_Trigger_Merged 或 Tx_Trigger_Arbitration 的 Tx_Trigger，则会检测到配置错误（TTOST[EL] = “11”，严重程度 3）。可为同一发送消息缓冲区定义多个 Tx_Trigger，某些基本周期中可能会忽略它们，具体取决于周期代码。计算预计 Tx_Trigger 数 (TTMLM[ENTT]) 时，必须考虑周期代码。

Watch_Trigger （TYPE = “0110”）和 Watch_Trigger_Gap （TYPE = “0111”）会检查缺失的参考消息。时间主控节点和时间被控节点都会使用它们。Watch_Trigger_Gap 仅用于外部事件同步时间触发工作模式。在该模式下，当 FDCAN 同步状态为 In_Gap （TTOST[SYS] = “10” ）时，会忽略 Watch_Trigger。

Rx_Trigger（TYPE = “1000”）用于检查独占时间窗口中周期性消息的接收情况。在达到 In_Schedule 或 In_Gap 状态之前，Rx_Trigger 不会激活。Rx_Trigger 的时间标记应放置在消息发送结束之后，与时间窗口边界无关。某些基本周期中可能会忽略 Rx_Trigger，具体取决于周期代码。在 Rx_Trigger 的时间标记处，会检查在该时间标记之前以及周期开始之后或上一 Rx_Trigger 之后接收的最后一条消息是否与 MNR 引用的验收过滤器元素相匹配。接受的消息会根据验收过滤结果存储在两个接收 FIFO 之一，与 Rx_Trigger 无关。Rx_Trigger 引用的验收过滤器元素应放置在过滤器列表开始处，以确保过滤会在达到 Rx_Trigger 时间标记之前完成。

Time_Base_Trigger（TYPE = “1001”）用于根据 TMIN 和 TMEX 的配置生成内部/外部事件。

End_of_List（TYPE = “1010…1111”）属于非法触发类型，如果在触发存储器中出现 Watch_Trigger 和 Watch_Trigger_Gap 之前出现了 End_of_List 触发，则会检测到配置错误（TTOST[EL] = “11”，严重程度 3）。

节点触发列表的限制

同一周期时间和周期计数不能同时激活两个触发，但在不同基本周期中激活的触发（周期代码不同）可共用同一时间标记。

Rx_Trigger 和 Time_Base_Trigger 不能置于 Tx_Trigger_Single/Continuous/Arbitration 的发送使能窗口中，但可置于 Tx_Trigger_Merged 之后。

如果触发置于 Watch_Trigger 之后（或者在 TTOST[SYS] = “10”时置于 Watch_Trigger_Gap 之后），触发将永远不会激活。如果参考消息按时发送，监视触发本身将不会激活。

所有未使用的触发存储器字 （在 Watch_Trigger 之后；TTOST[SYS] = “10”时则在 Watch_Trigger_Gap 之后）必须设为触发类型 End_of_List。

潜在时间主控节点的典型触发列表将以 Tx_Trigger 和 Rx_Trigger 数目开头，后接 Tx_Ref_Trigger 和 Watch_Trigger。对于进行外部事件同步时间触发通信的网络，后接 Tx_Ref_Trigger_Gap 和 the Watch_Trigger_Gap。时间被控节点的触发列表与之相同，但不包含 Tx_Ref_Trigger 和 Tx_Ref_Trigger_Gap。

每个基本周期开始时，也就是每次接收或发送参考消息时，都会从第一个触发存储器元素开始对触发列表进行处理。FSE 会查找第一个周期代码与当前周期计数匹配的触发。FSE 会等待周期时间达到触发时间标记并激活触发。随后，FSE 会在列表中查找下一个周期代码与当前周期计数匹配的触发。

对于 Tx_Ref_Trigger 和 Tx_Ref_Trigger_Gap 附近的时间，需要做特殊考虑。在竞争主控的时间主控节点中，Tx_Ref_Trigger 的有效时间标记可能会递减，以便成为第一个启动参考消息的节点。在备份时间主控节点中，Tx_Ref_Trigger 或 Tx_Ref_Trigger_Gap 的有效时间标记是其配置的时间标记与参考触发偏移 TTOCF[IRTO] 之和。如果达到 2 级错误（TTOST[EL] = “10”），有效时间标记则为其时间标记与 0x127 之和。此范围内不应放入其他触发元素，否则时间标记的出现顺序可能会混乱并标记为配置错误。只要参考消息及时到达，Tx_Ref_Trigger 之后到达的触发元素就可能永远也不会生效。

以下参数之间具有相关性：

- APB 时钟频率
- 触发 RAM 访问的速度和等待时间
- 验收过滤器列表的长度
- 触发元素数量
- 触发元素周期代码过滤的复杂程度
- 触发元素时间标记之间的偏移

触发处理示例

下例介绍了如何通过节点系统矩阵生成触发列表。假设节点 A 是第一个时间主控节点，并已知道*表 460* 所示的系统矩阵的部分。

表 460. 系统矩阵，节点 A

周期计数	时间标记						
	1	2	3	4	5	6	7
0	Tx7	-	-	-	-	TxRef	错误
1	Rx3	-	Tx2、Tx4		-	TxRef	错误
2	-	-	-	-	-	TxRef	错误
3	Tx7	-	Rx5	-	-	TxRef	错误
4	Tx7	-	-	Rx6	-	TxRef	错误

周期计数从 0 开始，计数到 0、1、3、7、15、31、63 为止（系统矩阵中对应的基本周期编号为 1、2、4、8、16、32、64）。最大周期计数是通过 TTMLM.CCM 配置的。周期代码 CC 由重复系数（= 最高有效位“1”的值）以及系统矩阵中的第一个基本周期编号（= 最高有效位“1”之后的位域）组成。

示例：如果周期代码为 0b0010011（重复系数 = 16，第一个基本周期 = 3），并且最大周期计数 TTMLM.CCM =“0x3F”，则会在周期计数达到 3、19、35、51 时出现匹配。

触发元素由时间标记 TM、周期代码 CC、触发类型 TYPE 和消息编号 MNR 组成。进行发送时，MNR 会引用发送缓冲区编号 (0..31)。进行接收时，MNR 会引用验收过滤期间匹配的过滤器元素 (0...127) 的编号。根据过滤器类型 FTYPE 的配置，将引用 11 位或 29 位消息 ID 过滤器列表。

此外，还可以配置触发元素，以生成时间标记事件内部 TMIN 和时间标记事件外部 TMEX。消息状态计数 MSC 保存计数器值 (0..7)，用于在触发元素时间标记激活时立即调度独占时间窗口中周期性消息的错误。

表 461. 触发列表，节点 A

触发	时间标记 TM[15:0]	周期代码 CC [6:0]	触发类型 TYPE [3:0]	消息编号 MNR [6:0]
0	Mark1	0b0000100	Tx_Trigger_Single	7
1	标记 1	0b1000000	Rx_Trigger	3
2	标记 1	0b1000011	Tx_Trigger_Single	7
3	标记 3	0b1000001	Tx_Trigger_Merged	2

表 461. 触发列表，节点 A（续）

触发	时间标记 TM[15:0]	周期代码 CC [6:0]	触发类型 TYPE [3:0]	消息编号 MNR [6:0]
4	标记 3	0b1000011	Rx_Trigger	5
5	标记 4	0b1000001	Tx_Trigger_Arbitration	4
6	标记 4	0b1000100	Rx_Trigger	6
7	标记 6	N/A	Tx_Ref_Trigger	0（参考）
8	标记 7	N/A	Watch_Trigger	N/A
9	N/A	N/A	End_of_List	N/A

Tx_Trigger_Single、Tx_Trigger_Continuous、Tx_Trigger_Merged、Tx_Trigger_Arbitration、Rx_Trigger 和 Time_Base_Trigger 仅对指定的周期代码有效。对于所有其他触发类型，会忽略周期代码。

FSE 启动基本周期时，会从 0 开始扫描触发列表，直至达到时间标记大于周期时间且其周期代码 CC 与实际周期计数相匹配的触发，或者遇到了类型为 Tx_Ref_Trigger、Tx_Ref_Trigger_Gap、Watch_Trigger 或 Watch_Trigger_Gap 的触发。

当周期时间达到时间标记 TM 时，会启动由触发类型 TYPE 和消息编号 MNR 定义的操作。达到 End_of_List 时，配置中会出错。

在 Mark6 处，会发送参考消息（始终为 TxRef）。发送参考消息后，FSE 会返回到触发列表开头。达到 Mark7 处的监视触发时，节点不能发送参考消息；此时会启动错误处理过程。

配置错误检测

下列情况下，会通过 TTOST[EL] =“11”（严重程度 3）指示配置错误：

- FSE 到达列表中周期代码与当前周期计数匹配、但时间标记小于周期时间的触发。
- 之前激活的触发为 Tx_Trigger_Merged，FSE 到达列表中周期代码与当前周期计数匹配、但既不是 Tx_Trigger_Merged、也不是 Tx_Trigger_Arbitration、Time_Base_Trigger 或 Rx_Trigger 的触发。
- TTOCF[TM]=“0”的节点（时间被控节点）的 FSE 遇到 Tx_Ref_Trigger 或 Tx_Ref_Trigger_Gap。
- 在具有匹配周期代码的 Tx_Trigger 的发送使能窗口（由 TTMLM[TXEW] 定义）中放置的任何时间标记处。
- 时间标记放置在 Tx_Ref_Trigger 的时间标记附近，参考触发偏移 TTOST[RTO] 会导致在周期时间中的测量顺序反转。

TTCAN 调度初始化

CCCR[INIT] 复位时，会开始同步到 TTCAN 消息调度。TTCAN 可严格按照时间触发（TTOCF[GEN] =“0”）或外部事件同步时间触发（TTOCF[GEN] =“1”）操作。TTOST[SYS] =“00”时（非同步），所有节点都会在其触发列表开头以周期时间 0 启动，除参考消息外，不会使能任何传输。处于外部事件同步时间触发工作模式下的节点将忽略 Tx_Ref_Trigger 和 Watch_Trigger，并将使用 Tx_Ref_Trigger_Gap 和 Watch_Trigger_Gap 代替，直到第一条参考消息决定是否激活间隙。

时间被控节点

完成配置后，如果时间被控节点在达到 Watch_Trigger 之前未接收到任何消息，则会忽略其 Watch_Trigger 和 Watch_Trigger_Gap。当时间被控节点达到 Init_Watch_Trigger 时，中断标志 TTIR[IWT] 置 1，FSE 冻结，周期时间失效，但节点仍可参与 CAN 总线通信（进行确认或发送错误标志）。接收到的第一条参考消息将重新启动 FSE 和周期时间。

注： *Init_Watch_Trigger 不属于触发列表。它作为内部计数器执行操作，会递增计数到 0xFFFF（= 最大周期时间）。*

如果时间被控节点在达到 Watch_Trigger 之前接收到了任何除参考消息以外的消息，则会判定发生了致命错误（TTOST[EL] = “11”，严重程度 3），并将中断标志 TTIR[WT] 置 1，然后切断其 CAN 总线输出并进入总线监控模式（CCCR[MON] 设为 “1”）。在总线监控模式下，仍可以接收消息，但不能发送任何显性位，因此不能提供确认。

注： *要退出致命错误状态，主机必须设置 CCCR[INIT] = “1”。CCCR[INIT] 复位后，节点会重新开始 TTCAN 通信。*

如果同步过程中未出错，第一条参考消息会设置 TTOST[SYS] = “01”（同步），第二条参考消息会将 FDCAN 同步状态（具体取决于 Next_is_Gap 位）设为 TTOST[SYS] = “11” (In_Schedule) 或 TTOST[SYS] = “10” (In_Gap)，从而使能所有 Tx_Trigger 和 Rx_Trigger。

潜在时间主控节点

完成配置后，潜在时间主控节点将在达到其 Tx_Ref_Trigger（外部事件同步时间触发操作时为 Tx_Ref_Trigger_Gap）后开始发送参考消息。如果潜在时间主控节点在达到 Watch_Trigger 之前未接收到任何消息或未成功发送参考消息（假设的原因：所有其他节点仍处于复位或配置状态，不能提供确认），则会忽略其 Watch_Trigger 和 Watch_Trigger_Gap。当潜在时间主控节点达到 Init_Watch_Trigger 时，尝试进行的发送会中止，中断标志 TTIR[IWT] 置 1，FSE 冻结，周期时间失效，但节点仍可参与 CAN 总线通信（提供确认或发送错误标志）。复位 TTIR[IWT] 后，将在下一次满足 Init_Watch_Trigger 条件或接收另一 CAN 消息时重新使能参考消息发送。接收参考消息不会重新启动 FSE。

如果潜在时间主控节点在接收到除参考消息以外的消息后达到 Watch_Trigger，则会判定发生了致命错误（TTOST[EL] = “11”，严重程度 3），并将中断标志 TTIR[WT] 置 1，然后切断其 CAN 总线输出并进入总线监控模式（CCCR[MON] 设为 “1”）。在总线监控模式下，仍可以接收消息，但不能发送任何显性位，因此不能提供确认。

如果检测到初始化过程中存在错误，第一条参考消息会设置 TTOST[SYS] = “01”（同步），第二条参考消息会将 FDCAN 同步状态（具体取决于 Next_is_Gap 位）设为 TTOST[SYS] = “11” (In_Schedule) 或 TTOST[SYS] = “10” (In_Gap)，从而使能所有 Tx_Trigger 和 Rx_Trigger。

如果潜在时间主控节点是上一条参考消息的发送方，则该潜在时间主控节点就是当前时间主控节点（TTOST[MS] = “11”），否则是备份时间主控节点（TTOST[MS] = “10”）。

当所有潜在时间主控节点均已完成配置后，网络中时间主控优先级最高的节点将成为当前时间主控节点。

56.3.8 TTCAN 间隙控制

所有与间隙控制相关的功能仅在 FDCAN 工作于外部事件同步时间触发模式下时（TTOCF[GEN] = “1”）才能使用。在该工作模式下，由于会在系统矩阵的基本周期之间插入间隙，因此 FDCAN 消息调度可能被中断。所有连接到 CAN 网络的节点必须针对外部事件同步时间触发操作进行配置。

在间隙期间，所有发送都会停止，CAN 总线保持空闲状态。间隙在下一条参考消息启动新的基本周期时结束。间隙在在本身由参考消息启动（例如 Next_is_Gap = “1”）的基本周期结束时开始。间隙是通过当前时间主控节点启动的。

当前时间主控节点可通过两种方式启动间隙。当应用程序写入 TTOCN[NIG] = “1”时，可在软件控制下启动时隙。Next_is_Gap 位将以“1”的形式与下一参考消息一起发送。当应用程序通过写入 TTOCN[GCS] = “1”的方式使能事件触发输入引脚 fdcan_evt 时，也可在硬件控制下启动时隙。当参考消息启动且 TTOCN[GCS] 置 1 时，事件触发引脚 fdcan_evt 上的高电平会将 Next_is_Gap 置“1”。

参考消息完成后，TTOST[WFE] 位将立即向时间主控节点以及时间被控节点发布时隙。当前的基本周期将继续，直至达到其最后的时间窗口。最后的时间窗口之后的时间就是间隙时间。

对于实际时间主控节点和潜在时间主控节点，当最后一个基本周期结束并且间隙时间开始时，TTOST[GSI] 将置 1。在作为时间被控节点的节点中，位 TTOST[GSI] 将保持为“0”。

当潜在时间主控节点处于同步状态 In_Gap（TTOST[SYS] = “10”）时，可通过四种方法有计划地结束间隙：

在软件控制下写入 TTOCN[FGP] = “1”。

在硬件控制下（TTOCN[GCS] = “1”），如果输入引脚 fdcan_evt 上出现从高电平变为低电平的边沿，则 TTOCN[FGP] 会置 1，并会重新启动调度。

第三种方法是时间触发重启。当 TTOCN[TMG] = “1”时，下一寄存器时间标记中断（TTIR[RTMI] = “1”）会将 TTOCN[FGP] 置 1 并启动参考消息。

最后一种方法是：任何潜在时间主控节点会在达到其 Tx_Ref_Trigger_Gap 后结束间隙，前提是要同步的事件并未及时发生。

以上任何方法都不会导致基本周期因参考消息而中断。

在间隙时间开始后将 TTOCN[FGP] 置 1 会立即开始发送参考消息，继而会同步消息调度。如果 TTOCN[FGP] 在间隙时间开始之前置 1（基本周期仍在进行中），则将在基本周期结束时（Tx_Ref_Trigger 时）启动下一参考消息，消息调度中将不存在间隙时间。

在严格的时间触发操作中，参考消息中的位 Next_is_Gap = “1”将被忽略，同时事件触发输入引脚 fdcan_evt 以及位 TTOCN[NIG]、TTOCN[FGP] 和 TTOCN[TMG] 也会被忽略。

56.3.9 停止监视

停止监视功能可捕获由外部事件触发的 FDCAN 内部时间值（本地时间、周期时间或全局时间）。

要使能停止监视功能，应用程序必须先通过 TTOCN[SWS] 将本地时间、周期时间或全局时间定义为停止监视源。如果 TTOCN[SWS] 不是“00”，并且 TT 中断寄存器标志 TTIR[SWE] 为“0”，则 TTTOCN[SWS] 选择的实际时间值将在停止监视触发引脚 fdcan_swt 的下一上升/下降沿（通过 TTOCN[SWP] 配置）复制到 TTCPT[SWV]。此操作会将中断标志 TTIR[SWE] 置 1。应用程序读取 TTCPT[SWV] 后，可通过将 TTIR[SWE] 复位为“0”的方式使能下一停止监视事件。

56.3.10 本地时间、周期时间、全局时间和外部时钟同步

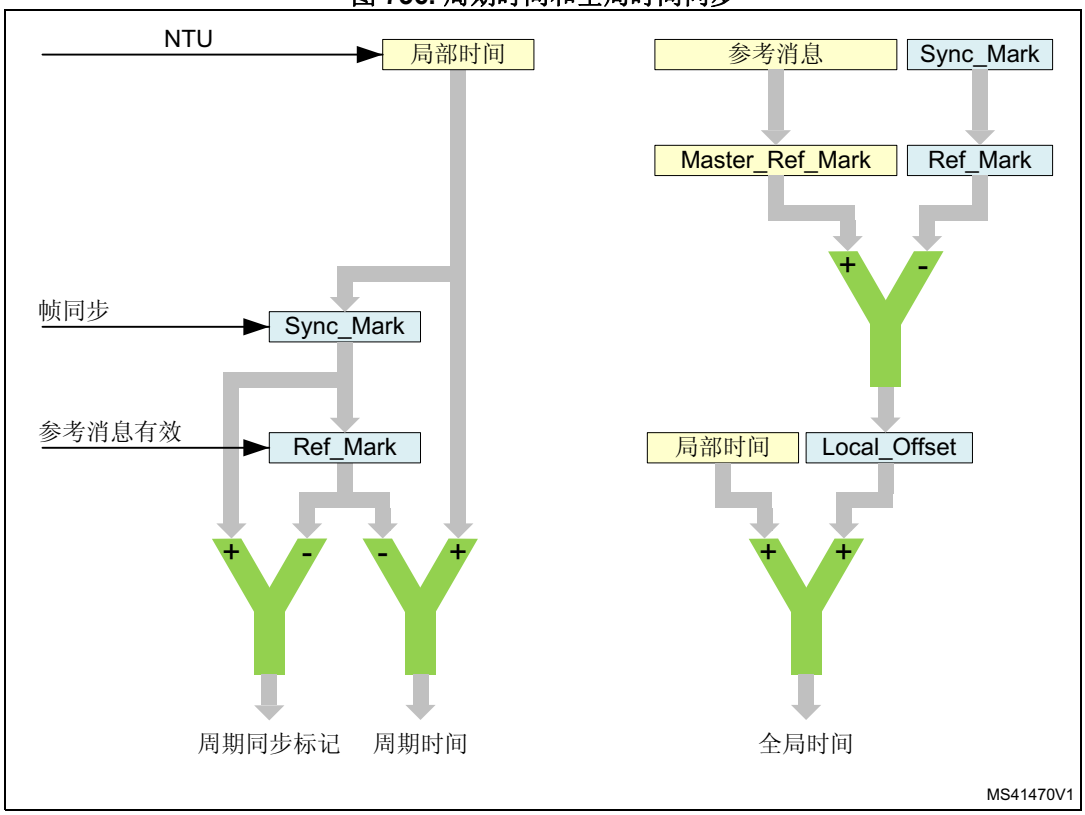
时间触发 CAN 可分为两个级别：1 级和 2 级。1 级仅使用周期时间提供时间触发操作。2 级提供增强的同步质量、全局时间和外部时钟同步。在两种级别下，所有定时特性均基于本地时基，即本地时间。

本地时间为 16 位的循环计数器，每个 NTU 都会递增。在内部，NTU 由 3 位计数器表示，可被视为本地时间的小数部分（三个二进制数字）。一般情况下，3 位 NTU 计数器每隔一个 NTU 会递增 8 次。如果 NTU 的长度小于 8 个 CAN 时钟周期（也许在等级 1 中是这样配置的，或者是级别 2 中的时钟校准造成的），则会调整 NTU 分数的长度，调整后的 NTU 计数器每隔一个 NTU 仅会递增 4 次。

图 736 介绍了包括时间主控节点在内的所有 FDCAN 节点以同一方式执行的周期时间和全局时间的同步。如果接收或发送的消息调用对消息本地时间的捕获，则该消息为帧同步事件。这一帧同步事件发生在每个帧起始 (SoF) 位的采样点，并会使本地时间以 Sync_Mark 形式存储。将会捕获 Sync_Marks 和 Ref_Marks，包括 3 位小数部分。

只要成功发送或接收有效的参考消息，内部 Ref_Mark 就会通过 Sync_Mark 进行更新。Ref_Mark 与 Sync_Mark 之差是存储在寄存器 TTCSM 中的周期同步标记（周期同步标记 = Sync_Mark - Ref_Mark）。Ref_Mark 与本地时间实际值之差的 16 个最高有效位为周期时间（周期时间 = 本地时间 - Ref_Mark）。

图 736. 周期时间和全局时间同步



可从 TTCTC[CT] 读取的周期时间是节点本地时间与 Ref_Mark 之差，两者均同步到 APB 时钟域并截断为 16 位。

全局时间仅存在于 TTCAN 0 级和 2 级中，在 1 级中无效。全局时间的节点视图为全局时间的本地图像（以本地 NTU 数为单位）。完成配置后，潜在时间主控节点将使用自己的本地时间作为全局时间。时间主控节点通过在参考消息中发送自己的 Ref_Mark 作为 Master_Ref_Mark（字节 3 和 4），以将自己的本地时间建立为全局时间。可从 TTLGT[GT] 读取的全局时间是节点本地时间与其本地偏移之和，两者均同步到 APB 时钟域并截断为 16 位。小数部分仅用于时钟同步。

接收参考消息的节点会通过将其本地 Ref_Mark 与接收到的 Master_Ref_Mark 进行比较来计算其本地偏移（请参见*图 737*）。全局时间的节点视图为本地时间 + 本地偏移。在从未接收到另一时间主控节点参考消息的潜在时间主控节点中，Local_Offset 将为 0。如果某个节点在首次接收到其他参考消息后成为当前时间主控节点，Local_Offset 将冻结为其上一个值。在时间接收节点中，由于存在时钟偏移，在另一节点成为时间主控节点时、或者出现全局时间不连续的情况下（由参考消息中的 Disc_Bit 指示），可能会对 Local_Offset 稍作调整。除全局时间不连续的情况之外，由寄存器 TTLGT 提供给应用程序的全局时间会通过低通滤波进行平滑处理，以得到连续单调值。

图 737. TTCAN 0 级和 2 级偏移补偿

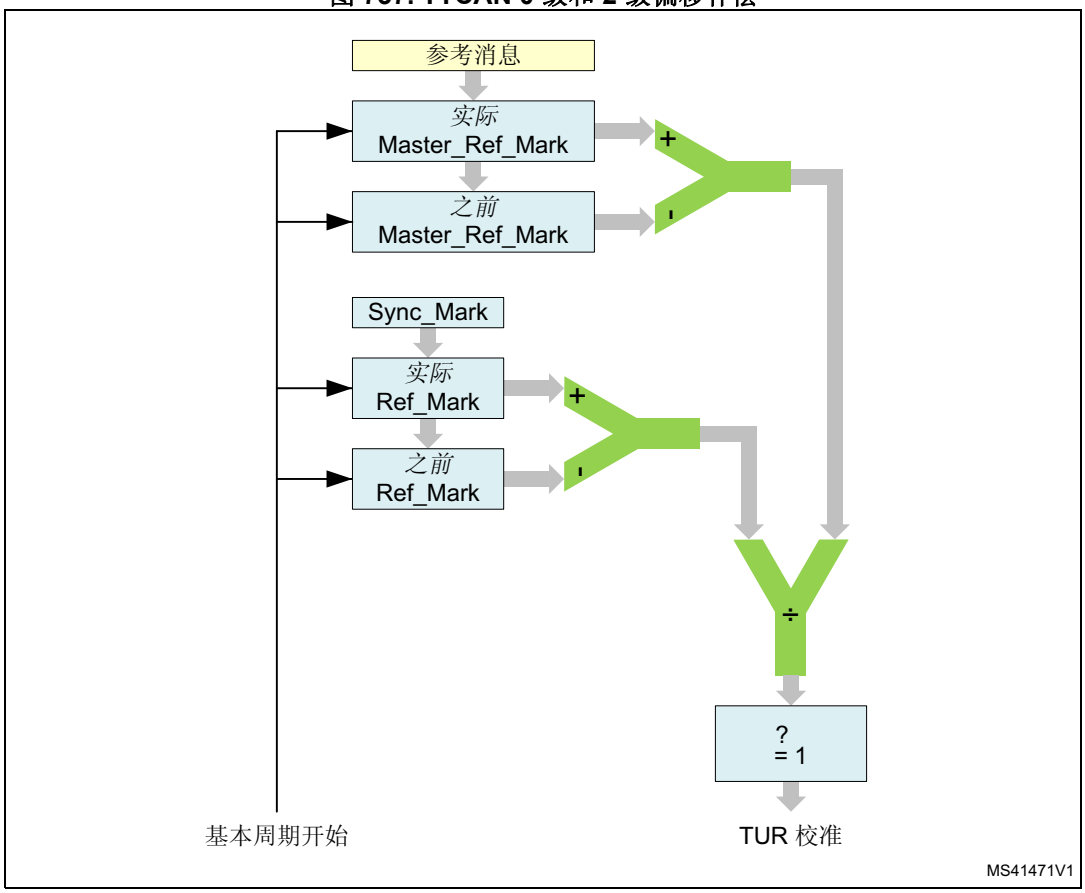


图 737 介绍了 TTCAN 0 级和 2 级中，接收节点每次是怎样通过比较本地时间与全局时间中基本周期的长度来补偿其自身的本地时钟与时间主控节点时钟之间的偏移的。如果两个值之间存在偏差，参考消息中的 Disc_Bit 不会置 1，并且会计算 TURNA[NAV] 的新值。如果同步偏差 SD = | NC - TURNA[NAV] | ≤ SDL（同步偏差限值），则 TURNA[NAV] 的新值会生效，否则会暂停自动偏移补偿。

在 TTCAN 0 级和 2 级中，TTOST[QCS] 指示自动偏移补偿处于激活状态还是暂停状态。在 TTCAN 1 级中，TTOST[QCS] 始终为“1”。

当前时间主控节点可将其本地时钟速度和全局时间相位同步到外部时钟源。此操作是通过位 TTOCF[EECS] 使能的。

可使用停止监视功能（请参见*第 56.3.9 节：停止监视*）测量本地时钟与外部时钟的时钟速度之差。调整本地时钟速度时，要先将新计算的分子配置下限写入 TURCF[NCL]（操作过程中不能更新 TURCF[DC]）。向 TTOCN[ECS] 写入“1”后，新值会生效。

调整全局时间相位时，要先将相位偏移写入 TT 全局时间预设寄存器 TTGTP。向 TTOCN[SGT] 写入“1”后，新值会生效。在调整完全局时间相位后发送的第一条参考消息会将 Disc_Bit 设为“1”。

TTOST[QGTP] 指示节点全局时间是否与时间主控节点全局时间同相。在 TTCAN 1 级中以及 TTCAN 0、2 级中超过同步偏差限值（TTOST[QCS] = “0”）的情况下，TTOST[QGTP] 永久为“0”。通过对全局时间进行低通滤波为应用提供连续单调值时，此位暂时为“0”。如果上一参考消息包含 Disc_Bit = “1”或者 TTOST[QCS]= “0”，则不会进行低通滤波。

56.3.11 TTCAN 错误级别

ISO 11898-4 规定了四种错误严重等级：

- S0——无错。
- S1——警告——仅通知应用，响应特定于应用。
- S2——错误——通知应用。将会禁止在独占或仲裁时间窗口中的所有发送（也就是说，可能不会启动任何数据帧或远程帧）。潜在时钟主控节点仍会发送参考触发偏移 TTOST[RTO] 设为最大值 127 的参考消息。
- S3——严重错误。
- 通知应用。所有 CAN 总线操作都会停止，也就是说不允许发送显性位，且 CCCR[MON] 会置 1。应用更新配置（将 CCCR[CCE] 置 1）之前，S3 错误条件都会保持有效。

如果同时检测到多个错误，则会处理优先级最高的错误。检测到错误时，会通过 TTIR[ELC] 通知应用。错误级别通过 TTOST[EL] 进行监控。

按照 ISO 11898-4 的要求，TTCAN 会指示以下错误条件：

- Config_Error (S3)
- 如果合并仲裁时间窗口未正确关闭或者 Tx_Trigger 的时间标记超过 Tx_Ref_Trigger，则将错误级别 TTOST[EL] 设为“11”。
- Watch_Trigger_Reached (S3)
- 如果因参考消息丢失而到达监视触发，则将错误级别 TTOST[EL] 设为“11”。
- Application_Watchdog (S3)
- 如果应用无法处理应用看门狗，则将错误级别 TTOST[EL] 设为“11”。应用看门狗通过 TTOCF[AWL] 进行配置。应用看门狗通过读取寄存器 TTOST 进行处理。如果未及时处理看门狗，位 TTOST[AWE] 和中断标志 TTIR[AW] 会置 1，所有 FDCAN 通信停止，FDCAN 设为进入总线监控模式（CCCR[MON] 设为“1”）。
- CAN_Bus_Off (S3)
- 进入 CAN_Bus_Off 状态会将错误级别 TTOST[EL] 设为“11”。CAN_Bus_Off 状态通过 PSR[BO] = “1”以及 CCCR[INIT] = “1”指示。
- Scheduling_Error_2 (S2)

- 如果其中一个 Tx_Trigger 的 MSC 已达到 7，则将错误级别 TTOST[EL] 设为“10”。此外，中断标志 TTIR[SE2] 也会置 1。如果之前的矩阵周期中没有任何 Tx_Trigger 的 MSC 为 7，则会在矩阵周期开始时将错误级别 TTOST[EL] 复位为“00”。
- Tx_Overflow (S2)
- 如果发送计数等于或大于预期的 Tx_Trigger 数 TTMLM[ENTT]，且发生了 Tx_Trigger 事件，则会将错误级别 TTOST[EL] 设为“10”。此外，中断标志 TTIR[TXO] 也会置 1。如果新矩阵周期开始时发送计数大于 TTMLM[ENTT]，则会将错误级别 TTOST[EL] 复位为“00”。
- Scheduling_Error_1 (S1)
- 如果一个矩阵周期内（独占时间窗口中的）所有触发存储器元素最大 MSC 与最小 MSC 之差大于 2，或者独占 Rx_Trigger 的其中一个 MSC 达到了 7，则会将错误级别 TTOST[EL] 设为“01”。此外，中断标志 TTIR[SE1] 也会置 1。如果一个矩阵周期内以上条件均无效，则会将错误级别 TTOST[EL] 复位为“00”。
- Tx_Underflow (S1)
- 如果新矩阵周期开始时发送计数小于预期的 Tx_Trigger 数 TTMLM[ENTT]，则会将错误级别 TTOST[EL] 设为“01”。此外，中断标志 TTIR[TXU] 也会置 1。如果新矩阵周期开始时发送计数至少为 TTMLM[ENTT]，则会将错误级别 TTOST[EL] 复位为“00”。

56.3.12 TTCAN 消息处理

参考消息

对于潜在时间主控节点，参考消息的标识符是通过 TTRMC[RID] 配置的。不需要使用专用发送缓冲区来发送参考消息。发送参考消息时，TTCAN 1 级的第一个数据字节（也就是 TTCAN 0 级的前四个数据字节和 TTCAN 2 级的前四个数据字节）将由 FSE 提供。

如果参考消息有效负载选择 TTRMC[RMPS] 置 1，其余参考消息负载（1 级：字节 2-8；0、2 级：字节 5-6）将从发送缓冲区 0 获取。在这种情况下，会使用消息缓冲区 0 中的数据长度 DLC 代码。

表 462. 随参考消息一起发送的数据字节数

TTRMC.RMPS	TXBRP.TRP0	0 级	1 级	2 级
0	0	4	1	4
0	1	4	1	4
1	0	4	1	4
1	1	4+MBO	1+MBO	4+MBO

要随参考消息一起发送额外的有效负载，对于 1 级，必须配置 DLC > 1；对于 0 级和 2 级，需要配置 DLC > 4。此外，消息缓冲区 0 的发送请求挂起位 TXBRP[TRP0] 必须置 1（请参见*表 462*）。如果参考消息启动时位 TXBRP[TRP0] 未置 1，则参考消息仅会与由 FSE 提供的数据字节一起发送。

对参考消息进行验收过滤时，会使用参考标识符 TTRMC[RID]。

消息接收

对于事件驱动的 CAN 通信，消息接收是通过同样的方法在接收 FIFO 中完成的（请参见*接收处理单元*）。

消息状态计数 MSC 是相应的触发存储器元素的组成部分，配置过程中必须初始化为 0。当 TTCAN 处于同步状态 In_Gap 或 In_Schedule 时，会更新该值。接收到消息 Rx_Trigger 时，会进行更新。此时，会检查在该基本周期中接收到的最新消息在哪一验收过滤器元素发生了匹配。匹配过滤器编号会作为验收过滤器结果存储起来。如果此编号与该触发存储器元素中定义的过滤器编号相同，则 MSC 会递减 1。如果验收过滤器结果与为该过滤器元素定义的过滤器编号不同，或者验收过滤器结果清零，则 MSC 会递增 1。每次出现 Rx_Trigger 以及每次周期开始时，上一验收过滤器结果都会清零。

设置 Rx_Trigger 的时间标记值时，应确保目标消息的接收和验收过滤已完成。此时必须考虑 RAM 访问时间以及过滤器列表的顺序。建议将用于 Rx_Trigger 的过滤器放置在过滤器列表开头。不建议为参考消息使用 Rx_Trigger。

消息发送

为了发送时间触发消息，TTCAN 提供 32 个专用发送缓冲区（请参见*发送暂停*）。当 FDCAN 配置为进行时间触发操作时（TTOCF[OM] = “01” 或 “10”），发送 FIFO 或发送队列不可用。

触发存储器中的每个 Tx_Trigger 都指向包含特定消息的发送缓冲区。如果某一给定的发送缓冲区包含要在基本周期或矩阵周期中发送多次的消息，则该发送缓冲区可能有多个 Tx_Trigger。

应用程序必须定期在同步到周期时间的时间点更新数据。主机 CPU 负责确保不会发送未完全更新的消息。为了确保这一点，主机必须按以下方式进行操作：

Tx_Trigger_Single/Tx_Trigger_Merged/Tx_Trigger_Arbitration

- 通过读取 TXBTO 检查之前的发送是否已完成
- 更新发送缓冲区配置和/或有效负载
- 发出添加请求，将发送缓冲区请求挂起位置 1

Tx_Trigger_Continous

- 发出取消请求，将发送缓冲区请求挂起位复位
- 通过读取 TXBCF 检查取消是否已完成
- 更新发送缓冲区配置和/或有效负载
- 发出添加请求，将发送缓冲区请求挂起位置 1

与相应 Tx_Trigger 一起存储的 MSC 会提供关于发送是否成功的信息。

如果发送因 CAN 总线在相应的发送使能窗口内未处于空闲状态而无法开始，或者消息已启动且无法成功完成，则 MSC 会递增 1。如果消息成功发送，或者消息本可以在其发送使能窗口中启动、但因发送被禁止的原因而未启动（错误级别为 S2 的 TTCAN 或主机禁用了这一特殊消息），MSC 会递减 1。

发送缓冲区可进行动态管理，也就是说，多条标识符不同的消息可共用同一发送缓冲区元素。在这种情况下，主机必须通过检查 TXBRP 来确保要重新配置的发送缓冲区元素没有挂起的发送请求。

如果应更新具有挂起的发送请求的发送缓冲区，主机必须先发出取消请求，并通过读取 TXBCF 检查取消是否完成，然后才会开始更新。

发送处理单元会将消息从消息 RAM 传输到其中间输出缓冲区，消息会放入刚好在定义发送窗口起点的 Tx_Trigger 之前激活的触发元素处。传输时间期间以及传输时间之后，发送消息可能无法更新，其 TXBRP 位可能不会更改。要控制该传输时间，可在 Tx_Trigger 之前放置附加的触发元素。举例来说，触发元素可能是不需要进行其他任何操作的 Time_Base_Trigger。Tx_Trigger 与上一次触发的时间标记之差必须足够大，以确保即使其他模块对 RAM 的访问负载很高时，发送处理单元也能从消息 RAM 中读取四个字。

在独占时间窗口中发送

当周期时间达到 Tx_Trigger_Single 或 Tx_Trigger_Continuous 的时间标记时，会开始时间触发发送。不会在总线上与来自其他节点的消息进行仲裁。MSC 会根据发送尝试结果进行更新。成功完成通过 Tx_Trigger_Single 启动的发送后，相应的发送缓冲区请求挂起位会复位。成功完成通过 Tx_Trigger_Continuous 启动的发送后，相应的发送缓冲区请求挂起位会保持置 1。如果发送因受到干扰未成功完成，将在下一次（其中一个）Tx_Trigger 激活时重复发送。

在仲裁时间窗口中发送

当周期时间达到 Tx_Trigger_Arbitration 的时间标记时，会开始时间触发发送。多个节点可同时开始发送。在这种情况下，消息必须与来自其他节点的消息进行仲裁。MSC 未更新。如果发送不成功（失去仲裁或受到干扰），将在下一次（其中一个）Tx_Trigger 激活时重复发送。

在合并仲裁时间窗口中发送

合并仲裁时间窗口的用途是使多个节点能够按照 CAN 仲裁提供的顺序即时发送有限数量的帧，而不是通过一个节点进行突发发送。由于节点在该时间窗口中没有独占访问权限，因此可能出现部分请求的发送不成功的情况。

失去仲裁或受到错误干扰的消息可在同一合并仲裁时间窗口中重新发送。如果相应的发送请求挂起标志通过成功的发送取消复位，则不会启动重新发送过程。

在单独发送窗口中，发送处理单元会发送由触发元素的消息编号指示的消息。在合并仲裁时间窗口中，最多可处理触发列表中的三个消息编号，并且将按照触发列表定义的顺序尝试对其进行发送。如果在发送第一条消息之前达到第四条消息的时间标记（或者由主机取消），将会忽略第四个请求。

合并仲裁时间窗口中的发送不是通过时间触发的。消息发送可在其时间标记之前开始，如果总线不处于空闲状态，也可以在时间标记之后开始。

由特定节点在合并仲裁时间窗口中发送的消息将按照其 Tx_Trigger 的顺序开始发送，因此，如果 CAN 优先级较低的消息竞争总线流量，可能会阻碍以下优先级较高的消息成功发送。配置触发列表时必须考虑这一点。Time_Base_Trigger 可放置在相邻的 Tx_Trigger 之间，以定义相应发送缓冲区数据需要更新的截止时间。

56.3.13 TTCAN 中断和错误处理

TT 中断寄存器 TTIR 由四部分组成。每个中断均可单独通过 TT 中断使能寄存器 TTIE 中的相应位使能。主机将标志清零之前，标志仍保持置 1 状态。通过向对应位位置写入“1”将标志清零。

第一部分包括标志 CER、AW、WT 和 IWT。每个标志都指示会停止 CAN 通信的致命错误条件。除 IWT 之外，这些错误条件需要先重新配置 FDCAN 模块，然后才能重新开始进行通信。

第二部分包括标志 ELC、SE1、SE2、TXO、TXU 和 GTE。每个标志都指示 CAN 通信受到干扰的错误条件。如果 CAN 通信中断是瞬时故障造成的（例如 CAN 总线上的干扰），则会通过 FDCAN 协议故障处理进行处理，不需要应用程序干预。

第三部分由标志 GTD、GTW、SWE、TTMI 和 RTMI 组成。前两个标志是由全局时间事件（仅限 0 级、2 级）控制的，需要应用程序作出响应。如果通过引脚 fdcan_swt 的上升沿触发停止监视事件，则会捕获内部时间值。触发时间标记中断会通知应用已达到特定 Time_Base_Trigger。寄存器时间标记中断指示 TTOCN[TMC] 引用的时间（周期时间、本地时间或全局时间）等于时间标记 TTTMK[TM]，也可用于结束间隙。

第四部分由标志 SOG、CSM、SMC 和 SBC 组成。这些标志可用于将应用程序同步到通信调度。

56.3.14 0 级

TTCAN 0 级不是 ISO11898-4 的一部分。该工作模式会使处于 TTCAN 2 级的硬件保持已校准的全局时基，根据 ISO11898-1 的规定，该工作模式还可用于事件驱动 CAN。

0 级操作是通过 TTOCF[OM] = “11”配置的。在该模式下，FDCAN 在事件驱动 CAN 通信中工作，没有固定的调度，TTOCF[GEN] 的配置会被忽略。0 级参考消息中不能实现外部事件同步操作。同步时基通过发送参考消息来保持。

在 0 级参考消息中，触发存储器未激活，因此不需要进行配置。当周期时间达到 TTOCF[IRTO] 0x200 时，时间标记中断标志 (TTIR[TTMI]) 置 1，它会提醒主机为消息缓冲区 0 设置发送请求。当周期时间达到 0xFF00 时，Watch_Trigger 中断标志 (TTIR[WT]) 会置 1。选择这些值后，会有足够的裕量进行稳定的时钟校准。不会进行后续的 TT 错误检查。

此外，还可以使用寄存器时间标记中断 (TTIR[RTMI])。

参考消息针对 2 级操作进行配置。已接收的参考消息通过在寄存器 TTRMC 中配置的标识符来识别。要发送参考消息，仅可使用消息缓冲区 0。主机为消息缓冲区 0 设置发送请求后，节点可随时发送参考消息，不存在参考触发偏移。

0 级操作是通过以下寄存器位配置的：

- TTRMC
- TTOCF，但 EVTP、AWL、GEN 位除外
- TTMLM，但 ENTT、TXEW 位除外
- TURCF

0 级操作是通过以下寄存器位控制的：

- TTOCN，但 NIG、TMG、FGP、GCS、TTMIE 位除外
- TTGTP
- TTTMK
- TTIR，不包括 CER、AW、IWT SE2、SE1、TXO、TXU、SOG 位（无功能）
- TTIR 以下位的功能发生了变化
  - 不是由触发存储器定义的 TTMI - 在周期时间 TTOCF[IRTO] 0x200 激活
  - 不是由触发存储器定义的 WT - 在周期时间 0xFF00 激活

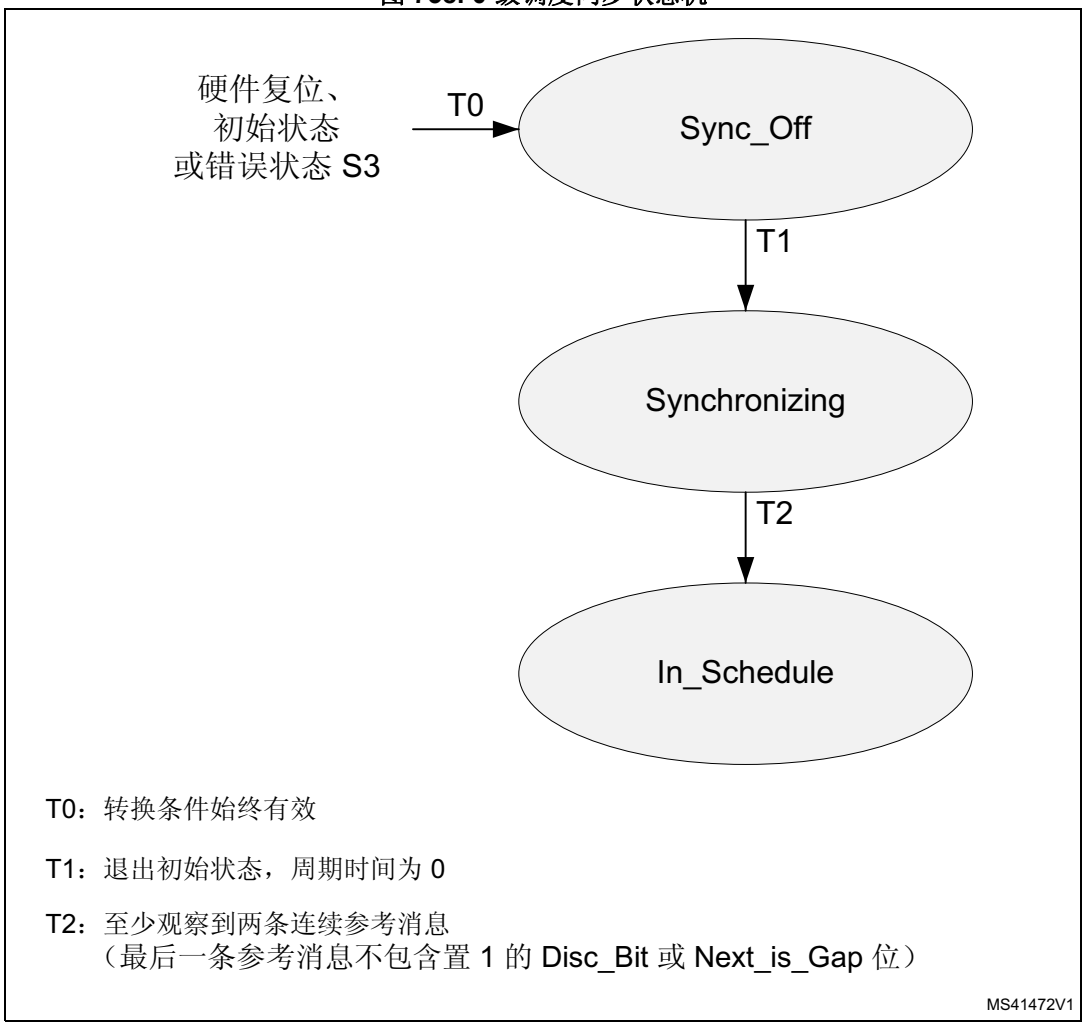
0 级操作是通过以下寄存器位指示的：

- TTOST，不包括 AWE、WFE、GSI、GFI、RTO 位（无功能）

同步

图 738 介绍了 TTCAN 0 级操作中的状态和状态转换。0 级没有 In_Gap 状态。

图 738. 0 级调度同步状态机



错误级别处理

0 级操作期间，仅可能出现以下错误条件：

- Watch_Trigger_Reached (S3)，已达到周期时间 0xFF00
- CAN_Bus_Off (S3)

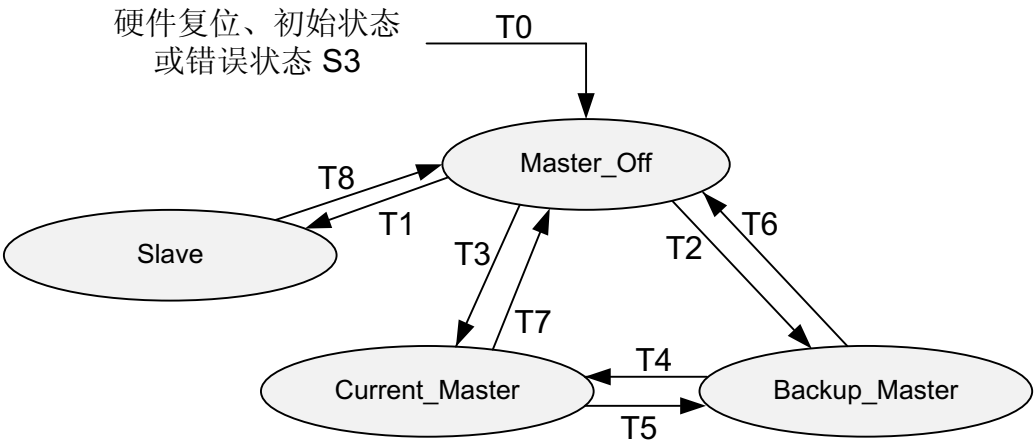
由于不会发生 S1 和 S2 错误，因此错误级别仅可在 S0（无错）和 S3（严重错误）之间切换。在 TTCAN 0 级中，对 S3 错误的处理方式有所不同。达到错误级别 S3 时，TTOST[SYS] 和 TTOST[MS] 均会复位，中断标志 TTIR[GTE] 和 TTIR[GTD] 会置 1。

进入错误级别 S3 (TTOST[EL] = “11”）时，不会进入总线监控模式（与 TTCAN 1 级和 2 级不同）。发送（时间主控节点）或接收（时间被控节点）下一参考消息后，会自动退出 S3 错误级别。

主控/被控节点关系

图 739 介绍了 TTCAN 0 级中主控/被控节点的关系。如果发生 S3 错误，FDCAN 会恢复到状态 Master_Off。

图 739. 0 级主控节点与被控节点的关系



T0：转换条件始终有效

T1：观察到参考消息（非潜在时间主器件时）

T2：观察到参考消息（主器件优先级与自身主器件优先级不同），错误状态 ≠ S3

T3：观察到参考消息（主器件优先级与自身主器件优先级相同），错误状态 ≠ S3

T4：观察到参考消息（自身主器件优先级）

T5：观察到参考消息（主器件优先级高于自身主器件优先级）

T6：错误状态 S3

T7：错误状态 S3

T8：错误状态 S3

MS41473V1

56.3.15 与外部时间调度同步

可使用此功能将 FDCAN 调度的相位同步到外部调度（例如另一 TTCAN 网络或 FlexRay 网络的调度）。仅当 FDCAN 为当前时间主控节点时（TTOST[MS] = “11”），才可使用此功能。

外部同步是通过事件触发输入引脚 fdcan_evt 控制的。如果 TTOCN[ESCN] 位置 1，则当事件触发引脚 fdcan_evt 上出现上升沿时，FDCAN 会将其实际周期时间与通过 TTGTP[CTP] 配置的目标相位值进行比较。

将 TTOCN[ESCN] 置 1 之前，主机需要使用 FDCAN 间隙控制等功能（请参见*第 56.3.8 节：TTCAN 间隙控制*）调整两个时间调度的相位。当主机将 TTOCN[ESCN] 置 1 后，TTOST[SPL] 会置 1。

如果事件触发引脚 fdcan_evt 上升沿处的周期时间与目标相位值 TTGTP[CTP] 之差大于 9 个 NTU，相位锁定位 TTOST[SPL] 会复位，中断标志 TTIR[CSM] 置 1。当另一节点变为时间主控节点时，TTOST[SPL] 也会复位（且 TTIR[CSM] 会置 1）。

如果 TTOST[SPL] 和 TTOCN[ESCN] 均置 1，并且事件触发引脚 fdcan_evt 上升沿处的周期时间与目标相位值 TTGTP[CTP] 之差小于或等于 9 个 NTU，相位锁定位 TTOST[SPL] 保持置 1，测得的差值会作为参考触发偏移值，用于调整下一次发送的参考消息的相位。

注： *每次基本周期开始时，都会对事件触发引脚 fdcan_evt 使能上升沿检测。第一个上升沿会触发实际周期时间与 TTGTP[CTP] 的比较。下一基本周期开始之前的所有后续边沿都会被忽略。*

56.3.16 FDCAN 接收缓冲区和 FIFO 元素

最多可在消息 RAM 中配置 64 个接收缓冲区和 2 个接收 FIFO。每个接收 FIFO 部分均可配置为最多存储 64 个已接收消息。接收缓冲区/FIFO 元素的结构如*表 463* 所示，相关说明请参见*表 464*。

表 463. 接收缓冲区和 FIFO 元素

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R0	ESI	XTD	RTR	ID[28:0]																												
R1	ANMF	FIDX[6:0]							Res.			FDF	BRS	DLC[3:0]			RXTS[15:0]															
R2	DB3[7:0]								DB2[7:0]								DB1[7:0]								DB0[7:0]							
R3	DB7[7:0]								DB6[7:0]								DB5[7:0]								DB4[7:0]							
⋮	⋮								⋮								⋮															
Rn	DBm[7:0]								DBm-1[7:0]								DBm-2[7:0]								DBm-3[7:0]							

可对元素大小进行配置，以通过寄存器 RXESC 存储数据字段多达 64 字节的 CAN FD 消息。

表 464. 接收缓冲区和 FIFO 元素说明

位域	说明
R0 位 31 ESI	错误状态指示符 (Error State Indicator) 0：发送节点为错误主动状态 1：发送节点为错误被动状态
R0 位 30 XTD	扩展标识符 (Extended Identifier) 向主机指示接收到的帧具有标准标识符还是扩展标识符。 0：11 位标准标识符 1：29 位扩展标识符
R0 位 29 RTR	远程发送请求 (Remote Transmission Request) 向主机指示接收到的帧属于数据帧还是远程帧。 0：接收到的帧为数据帧 1：接收到的帧为远程帧
R0 位 28:0 ID[28:0]	标识符 (Identifier) 标准标识符或扩展标识符取决于 XTD 位。标准标识符存储在 ID[28:18] 中。
R1 位 31 ANMF	接受非匹配帧 (Accepted Non-matching Frame) 可通过 GFC[ANFS] 和 GFC[ANFE] 使能接受非匹配帧。 0：接收到的帧与过滤器索引 FIDX 匹配 1：接收到的帧与任何接收过滤器元素都不匹配
R1 位 30:24 FIDX[6:0]	过滤器索引 (Filter Index) 0-127 = 匹配接收验收过滤器元素的索引（ANMF = “1” 时无效）。 范围为 0 到 SIDFC[LSS] - 1 或 XIDFC[LSE] - 1。
R1 位 21 FDF	FD 格式 (FD Format) 0：标准帧格式 1：FDCAN 帧格式（新 DLC 编码和 CRC）
R1 位 20 BRS	比特率切换 (Bit Rate Switch) 0：接收帧时不切换比特率 1：接收帧时切换比特率
R1 位 19:16 DLC[3:0]	数据长度代码 (Data Length Code) 0-8：CAN + CAN FD：接收到的帧包含 0-8 个数据字节 9-15：CAN：接收到的帧包含 8 个数据字节 9-15：CAN FD：接收到的帧包含 12/16/20/24/32/48/64 个数据字节
R1 位 15:0 RXTS[15:0]	接收时间戳 (Rx Timestamp) 开始接收帧时捕获的时间戳计数器值。分辨率取决于时间戳计数器预分频器 TSCC[TCP] 的配置。
R2 位 31:24 DB3[7:0]	数据字节 3 (Data Byte 3)
R2 位 23:16 DB2[7:0]	数据字节 2 (Data Byte 2)
R2 位 15:8 DB1[7:0]	数据字节 1 (Data Byte 1)

表 464. 接收缓冲区和 FIFO 元素说明（续）

位域	说明
R2 位 7:0 DB0[7:0]	数据字节 0 (Data Byte 0)
R3 位 31:24 DB7[7:0]	数据字节 7 (Data Byte 7)
R3 位 23:16 DB6[7:0]	数据字节 6 (Data Byte 6)
R3 位 15:8 DB5[7:0]	数据字节 5 (Data Byte 5)
R3 位 7:0 DB4[7:0]	数据字节 4 (Data Byte 4)
⋮	⋮
Rn 位 31:24 DBm[7:0]	数据字节 m (Data Byte m)
Rn 位 23:16 DBm-1[7:0]	数据字节 m-1 (Data Byte m-1)
Rn 位 15:8 DBm-2[7:0]	数据字节 m-2 (Data Byte m-2)
Rn 位 7:0 DBm-3[7:0]	数据字节 m-3 (Data Byte m-3)

56.3.17 FDCAN 发送缓冲区元素

发送缓冲区部分可配置为保存专用发送缓冲区以及发送 FIFO/发送队列的内容。如果发送缓冲区部分由专用发送缓冲区和发送 FIFO/发送队列共享，则专用发送缓冲区会在发送缓冲区部分的起始处开始，后面是分配给发送 FIFO 或发送队列的缓冲区。发送处理单元会评估发送缓冲区配置 TXBC.TFQS 和 TXBC.NDTB，以区分专用发送缓冲区与发送 FIFO/发送队列。可对元素大小进行配置，以通过寄存器 TXESC 存储数据字段多达 64 字节的 CAN FD 消息。

表 465. 发送缓冲区和 FIFO 元素

位	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
T0	ESI	XTD	RTR	ID[28:0]																												
T1	MM[7:0]								EFC	Res.	FDF	BPS	DLC[3:0]				Res.															
T2	DB3[7:0]								DB2[7:0]								DB1[7:0]								DB0[7:0]							
T3	DB7[7:0]								DB6[7:0]								DB5[7:0]								DB4[7:0]							
⋮	⋮								⋮								⋮															
Tn	DBm[7:0]								DBm-1[7:0]								DBm-2[7:0]								DBm-3[7:0]							

表 466. 发送缓冲区元素说明

位域	说明
T0 位 31 ESI(1)	错误状态指示符 (Error State Indicator) 0：CAN FD 格式的 ESI 位仅取决于错误被动标志 1：CAN FD 格式的 ESI 位会进行隐性发送
T0 位 30 XTD	扩展标识符 (Extended Identifier) 0：11 位标准标识符 1：29 位扩展标识符
T0 位 29 RTR(2)	远程发送请求 (Remote Transmission Request) 0：发送数据帧 1：发送远程帧
T0 位 28:0 ID[28:0]	标识符 (Identifier) 标准标识符或扩展标识符取决于 XTD 位。标准标识符必须写入 ID[28:18]。
T1 位 31:24 MM[7:0]	消息标记 (Message Marker) 在发送缓冲区配置期间由 CPU 写入。复制到发送事件 FIFO 元素中，用于标识发送消息状态。
T1 位 23 EFC	事件 FIFO 控制 (Event FIFO Control) 0：不存储发送事件 1：存储发送事件
T1 位 21 FDF	FD 格式 (FD Format) 0：以典型 CAN 格式发送帧 1：以 CAN FD 格式发送帧
T1 位 20 BRS(3)	比特率切换 (Bit Rate Switching) 0：发送 CAN FD 帧时不切换比特率 1：发送 CAN FD 帧时切换比特率
T1 位 19:16 DLC[3:0]	数据长度代码 (Data Length Code) 0-8：CAN + CAN FD：接收到的帧包含 0-8 个数据字节 9-15：CAN：接收到的帧包含 8 个数据字节 9-15：CAN FD：接收到的帧包含 12/16/20/24/32/48/64 个数据字节
T2 位 31:24 DB3[7:0]	数据字节 3 (Data Byte 3)
T2 位 23:16 DB2[7:0]	数据字节 2 (Data Byte 2)
T2 位 15:8 DB1[7:0]	数据字节 1 (Data Byte 1)
T2 位 7:0 DB0[7:0]	数据字节 0 (Data Byte 0)
T3 位 31:24 DB7[7:0]	数据字节 7 (Data Byte 7)
T3 位 23:16 DB6[7:0]	数据字节 6 (Data Byte 6)

表 466. 发送缓冲区元素说明（续）

位域	说明
T3 位 15:8 DB5[7:0]	数据字节 5 (Data Byte 5)
T3 位 7:0 DB4[7:0]	数据字节 4 (Data Byte 4)
⋮	⋮
Tn 位 31:24 DBm[7:0]	数据字节 m (Data Byte m)
Tn 位 23:16 DBm-1[7:0]	数据字节 m-1 (Data Byte m-1)
Tn 位 15:8 DBm-2[7:0]	数据字节 m-2 (Data Byte m-2)
Tn 位 7:0 DBm-3[7:0]	数据字节 m-3 (Data Byte m-3)

1. 发送缓冲区的 ESI 位会与错误被动标志进行或运算，以确定已发送 FD 帧中 ESI 位的值。按照 CAN FD 协议规范的要求，错误主动节点可选择隐性发送 ESI 位，但错误被动节点将始终隐性发送 ESI 位。
2. 当 RTR = 1 时，即使 CCCR.FDOE 使能以 CAN FD 格式进行发送，FDCAN 也会按照 ISO11898-1 的规定发送远程帧。
3. 仅当 CAN FD 操作使能时（CCCR.FDOE =“1”），才会评估 ESI、FDF 和 BRS 位。仅当额外设置 CCCR.BRSE =“1”时，才会评估 BRS 位。

56.3.18 FDCAN 发送事件 FIFO 元素

每个元素均存储与已发送消息相关的消息。通过读取发送事件 FIFO，主机 CPU 会按消息的发送顺序获取此类信息。关于发送事件 FIFO 的状态信息，可从寄存器 TXEFS 中获取。

表 467. 发送事件 FIFO 元素

位	31			24	23					16	15	8	7	0
E0	ESI	XTD	RTR		ID[28:0]									
E1	MM[7:0]				ET[1:0]		EDL	BRS	DLC[3:0]		TXTS[15:0]			

表 468. 发送事件 FIFO 元素说明

位域	说明
E0 位 31 ESI	错误状态指示符 (Error State Indicator) 0：发送节点为错误主动状态 1：发送节点为错误被动状态
E0 位 30 XTD	扩展标识符 (Extended Identifier) 0：11 位标准标识符 1：29 位扩展标识符

表 468. 发送事件 FIFO 元素说明（续）

位域	说明
E0 位 29 RTR	远程发送请求 (Remote Transmission Request) 0：发送数据帧 1：发送远程帧
E0 位 28:0 ID[28:0]	标识符 (Identifier) 标准标识符或扩展标识符取决于 XTD 位。标准标识符必须写入 ID[28:18]。
E1 位 31:24 MM[7:0]	消息标记 (Message Marker) 从发送缓冲区复制到发送事件 FIFO 元素中，用于标识发送消息状态。
E1 位 23:22 EFC	事件类型 (Event Type) 00：保留 01：发送事件 10：取消后仍发送（在 DAR 模式下发送时始终设为该值） 11：保留
E1 位 21 EDL	扩展数据长度 (Extended Data Length) 0：标准帧格式 1：FDCAN 帧格式（新 DLC 编码和 CRC）
E1 位 20 BRS	比特率切换 (Bit Rate Switching) 0：发送帧时不切换比特率 1：发送帧时切换比特率
T1 位 19:16 DLC[3:0]	数据长度代码 (Data Length Code) 0-8：发送包含 0-8 个数据字节的帧 9-15：发送包含 8 个数据字节的帧
E1 位 15:0 TXTS[15:0]	发送时间戳 (Tx Timestamp) 开始发送帧时捕获的时间戳计数器值。分辨率取决于时间戳计数器预分频器 TSCC[TCP] 的配置。

56.3.19 FDCAN 标准消息 ID 过滤器元素

最多可为 11 位标准 ID 配置 128 个过滤器元素。访问标准消息 ID 过滤器元素时，其地址为过滤器列表标准起始地址 SIDFC.FLSSA 加上过滤器元素的索引 (0…127)。

表 469. 标准消息 ID 过滤器元素

位	31		24	23	16	15	8	7	0
S0	SFT[1:0]	SFEC[2:0]		SFID1[10:0]		Res.		SFID2[10:0]	

表 470. 标准消息 ID 过滤器元素位域说明

位域		说明
位 31:30 SFT[1:0](1)		标准过滤器类型 (Standard Filter Type) 00：从 SFID1 到 SFID2 的范围过滤器 01：用于 SFID1 或 SFID2 的双 ID 过滤器 10：典型过滤器：SFID1 = 过滤器，SFID2 = 掩码 11：禁止过滤器元素
位 29:27 SFEC[2:0]		标准过滤器元素配置 (Standard Filter Element Configuration) 所有使能的过滤器元素都用于对标准帧进行验收过滤。当发现第一个匹配的已使能过滤器元素时，或者到达过滤器列表结尾处时，会停止验收过滤。如果 SFEC = “100”、“101”或“110”，则出现匹配时，会将中断标志 IR.HPM 置 1，并会生成中断（若使能）。在这种情况下，寄存器 HPMS 会更新为优先级匹配的状态。 000：禁止过滤器元素 001：如果过滤器匹配，则存储在接收 FIFO 0 中 010：如果过滤器匹配，则存储在接收 FIFO 1 中 011：如果过滤器匹配，则拒绝 ID 100：如果过滤器匹配，则设置优先级 101：如果过滤器匹配，则设置优先级并存储在 FIFO 0 中 110：如果过滤器匹配，则设置优先级并存储在 FIFO 1 中 111：存储在接收缓冲区中或作为调试消息，忽略 SFT[1:0] 的配置
位 26:16 SFID1[10:0]		标准过滤器 ID 1 (Standard Filter ID 1) 标准 ID 过滤器元素的第一个 ID。 过滤接收缓冲区或调试消息时，此位域定义要存储的标准消息的 ID。接收的标识符必须完全匹配，不使用屏蔽机制。
位 15:0	SFID2[15:10]	标准过滤器 ID 2 (Standard Filter ID 2) 该位域具有不同含义，具体视 SFEC 的配置而定： – SFEC = “001” ... “110”：标准 ID 过滤器元素的第二个 ID – SFEC = “111”：用于接收缓冲区或调试消息的过滤器
	SFID2[10:9]	确定接收的消息存储在接收缓冲区中，还是作为调试消息序列中的消息 A、B 或 C 进行处理。 00：将消息存储在接收缓冲区中 01：调试消息 A 10：调试消息 B 11：调试消息 C
	SFID2[8:6]	用于控制扩展接口的过滤器事件引脚。如果相应的位位置为“1”，则当发生过滤器匹配时，会在相关过滤器事件引脚上产生脉冲，脉冲持续时间为一个 m_ttcan_hclk 周期。 SFID2[8] 由校准单元使用。
	SFID2[5:0]	定义与用于存储匹配消息的接收缓冲区起始地址 RXBC.RBSA 的偏移。

1. 当 SFT = “11”时，会禁止过滤器元素，验收过滤会继续进行（与 SFEC = “000”时的行为相同）。

注：　　如果配置了保留值，则会将过滤器元素视为已禁止。

56.3.20 FDCAN 扩展消息 ID 过滤器元素

最多可为 29 位扩展 ID 配置 64 个过滤器元素。访问扩展消息 ID 过滤器元素时，其地址为过滤器列表扩展起始地址 XIDFC[FLESA] 加上过滤器元素的索引 (0…63) 的二倍。

表 471. 扩展消息 ID 过滤器元素

位	31		24	23 16	15 8	7 0
F0	EFEC[2:0]		EFID1[28:0]			
F1	EFTI[1:0]	Res.	EFID2[28:0]			

表 472. 扩展消息 ID 过滤器元素位域说明

位域	说明
F0 位 31:29 EFEC[2:0]	扩展过滤器元素配置 (Extended Filter Element Configuration) 所有使能的过滤器元素都用于对扩展帧进行验收过滤。当发现第一个匹配的已使能过滤器元素时，或者到达过滤器列表结尾处时，会停止验收过滤。如果 EFEC =“100”、“101”或“110”，则出现匹配时，会将中断标志 IR[HPM] 置 1，并会生成中断（若使能）。在这种情况下，寄存器 HPMS 会更新为优先级匹配的状态。 000：禁止过滤器元素 001：如果过滤器匹配，则存储在接收 FIFO 0 中 010：如果过滤器匹配，则存储在接收 FIFO 1 中 011：如果过滤器匹配，则拒绝 ID 100：如果过滤器匹配，则设置优先级 101：如果过滤器匹配，则设置优先级并存储在 FIFO 0 中 110：如果过滤器匹配，则设置优先级并存储在 FIFO 1 中 111：存储在接收缓冲区中，忽略 EFT[1:0] 的配置
F0 位 28:0 EFID1[28:0]	扩展过滤器 ID 1 (Extended Filter ID 1) 扩展 ID 过滤器元素的第一个 ID。 过滤接收缓冲区或调试消息时，此位域定义要存储的扩展消息的 ID。接收的标识符必须完全匹配，仅使用 XIDAM 屏蔽机制。
F1 位 31:30 EFT[1:0]	扩展过滤器类型 (Extended Filter Type) 00：从 EF1ID 到 EF2ID 的范围过滤器 (EF2ID ≥ EF1ID) 01：用于 EF1ID 或 EF2ID 的双 ID 过滤器 10：典型过滤器：EF1ID = 过滤器，EF2ID = 掩码 11：从 EF1ID 到 EF2ID 的范围过滤器 (EF2ID ≥ EF1ID)，未应用 XIDAM 掩码

表 472. 扩展消息 ID 过滤器元素位域说明（续）

位域		说明
F1 位 28:0	EFID2[10:0]	扩展过滤器 ID 2 (Extended Filter ID 2) 该位域具有不同含义，具体视 EFEC 的配置而定： – SFEC =“001”...“110”：扩展 ID 过滤器元素的第二个 ID – SFEC =“111”：用于接收缓冲区或调试消息的过滤器
	EFID2[10:9]	确定接收的消息存储在接收缓冲区中，还是作为调试消息序列中的消息 A、B 或 C 进行处理。 00：将消息存储在接收缓冲区中 01：调试消息 A 10：调试消息 B 11：调试消息 C
	EFID2[8:6]	用于控制扩展接口的过滤器事件引脚。如果相应的位位置为“1”，则当发生过滤器匹配时，会在相关过滤器事件引脚上产生脉冲，脉冲持续时间为一个 m_ttcan_hclk 周期。 EFID2[8] 接口由校准单元使用。
	EFID2[5:0]	定义与用于存储匹配消息的接收缓冲区起始地址 RXBC.RBSA 的偏移。

56.3.21 FDCAN 触发存储器元素

最多可配置 64 个触发存储器元素。访问触发存储器元素时，其地址为触发存储器起始地址 TTTMC[TMSA] 加上触发存储器元素的索引 (0…63)。

表 473. 触发存储器元素

位	31 24	23 16	15 8	7 0
T0	TM[15:0]		Res. CC[6:0]	Res. TMIN TMEX TYPE[3:0]
T1	Res.	FTYPE MNR[6:0]	Res.	MSC[2:0]

表 474. 触发存储器元素说明

位域	说明
T0 位 31:16 TM[15:0]	时间标记 (Time Mark) 触发激活的周期时间。
T0 位 14:8 CC[6:0]	周期代码 (Cycle Code) 触发有效的周期计数。触发类型为 Tx_Ref_Trigger、Tx_Ref_Trigger_Gap、Watch_Trigger、Watch_Trigger_Gap、End_of_List 时会忽略。 0b000000x 对所有周期有效 周期计数 mod2 = c 时，每 2 个周期，0b000001c 有效 周期计数 mod4 = cc 时，每 4 个周期，0b00001cc 有效 周期计数 mod8 = ccc 时，每 8 个周期，0b0001ccc 有效 周期计数 mod16 = cccc 时，每 16 个周期，0b001cccc 有效 周期计数 mod32 = ccccc 时，每 32 个周期，0b01ccccc 有效 周期计数 mod64 = cccccc 时，每 64 个周期，0b1cccccc 有效

表 474. 触发存储器元素说明（续）

位域	说明
T0 位 5 TMIN	内部时间标记事件 (Time Mark Event Internal) 0：不执行任何操作 1：当触发存储器元素激活时，TTIR.TTMI 会置 1
T0 位 4 TMEX	外部时间标记事件 (Time Mark Event External) 0：不执行任何操作 1：当触发存储器元素的时间标记激活且 TTOCN.TTMIE =“1”时，会在输出 m_ttcan_tmp 上产生长度为 1 个周期的脉冲。
T0 位 3:0 TYPE[3:0]	触发类型 (Trigger Type) 0000 Tx_Ref_Trigger——不处于间隙中时有效 0001 Tx_Ref_Trigger——处于间隙中时有效 0010 Tx_Trigger_Single——在独占时间窗口中开始单次发送 0011 Tx_Trigger_Continuous——在独占时间窗口中开始连续发送 0100 Tx_Trigger_Arbitration——在仲裁时间窗口中开始发送 0101 Tx_Trigger_Merged——启动合并仲裁窗口 0110 Watch_Trigger——不处于间隙中时有效 0111 Watch_Trigger_Gap——处于间隙中时有效 1000 Rx_Trigger——检查接收情况 1001 Time_Base_Trigger——仅控制 TMIN、TMEX 1010…1111=End_of_List——非法类型，会导致配置错误
T1 位 23FTYPE	过滤器类型 (Filter Type) 0：11 位标准消息 ID 1：29 位扩展消息 ID
T1 位 22:16 MNR[6:0](1)	消息编号 (Message Number) – 发送：触发对配置的发送缓冲区编号有效。有效值为 0 到 31。 – 接收：触发对标准/扩展消息 ID 过滤器元素编号有效。有效值分别为 0 到 63 和 0 到 127。
T1 位 2:0 MSC[2:0]	消息状态计数 (Message Status Count) 计算独占时间窗口中周期性消息的调度错误数。不包含用于仲裁消息的功能和事件驱动 CAN 通信中的功能 (ISO11898-1)。 0-7 = 实际状态

1. 当 FDCAN 处于 INIT 状态时，必须写入触发存储器元素。不允许对未处于 INIT 状态的触发存储器元素进行写访问。但如果 TMIN 和 TMEX 被定义为 TYPE Tx_Ref_Trigger 触发存储器元素的组成部分，则属于例外情况。在这种情况下，TMIN 和 TMEX 会在达到由实际参考触发偏移 (TTOST[RTO]) 修改的时间标记时激活。

56.4 FDCAN 寄存器

56.4.1 FDCAN 内核释放寄存器 (FDCAN_CREL)

FDCAN Core Release Register

偏移地址：0x0000

复位值：0xrrd dddd

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
REL[3:0]				STEP[3:0]				SUBSTEP[3:0]				YEAR[3:0]			
r												d			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MON[7:0]								DAY[7:0]							
d															

位 31:28 **REL**：内核释放 (Core release)
一位，BCD

位 27:24 **STEP**：内核释放步 (Step of Core release)
一位，BCD

位 23:20 **SUBSTEP**：内核释放子步 (Sub-step of Core release)
一位，BCD

位 19:16 **YEAR**：时间戳年 (Timestamp Year)
一位，BCD

位 15:8 **MONTH**：时间戳月 (Timestamp Month)
两位，BCD

位 7:0 **DAY**：时间戳日 (Timestamp Day)
两位，BCD

56.4.2 FDCAN 字节序寄存器 (FDCAN_ENDN)

FDCAN Endianness Register

偏移地址：0x0004

复位值：0x8765_4321

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

位 31:0 **ETV**：字节序测试值 (Endianness Test Value)
字节序测试值为 0x8765 4321。

56.4.3 FDCAN 数据位定时和预分频器寄存器 (FDCAN_DBTP)

FDCAN Data Bit Timing and Prescaler Register

偏移地址：0x000C

复位值：0x0000 0A33

仅当 CCCR.CCE 和 CCCR.INIT 位置 1 时，才可以向此寄存器写入数据。CAN 位时间的可编程范围为 4 到 25 个时间片。CAN 时间片的可编程范围为 1 到 1024 个 FDCAN 时钟周期。tq = (DBRP + 1) 个 FDCAN 时钟周期。

DTSEG1 为 Prop_Seg 与 Phase_Seg1 之和。DTSEG2 为 Phase_Seg2。因此，位时间的长度为（编程值）[DTSEG1 + DTSEG2 + 3] tq 或（函数值）[Sync_Seg + Prop_Seg + Phase_Seg1 + Phase_Seg2] tq。

信息处理时间 (IPT) 为零，这意味着下一位的数据在采样点后的第一个时钟边沿可用。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TDC	Res.	Res.	DBRP[4:0]				
r	r	r	r	r	r	r	r	r	r	r	rw				
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	DTSEG1[4:0]					DTSEG2[3:0]			DSJW[3:0]				
r	r	r	rw												

位 31:24 保留

位 23 **TDC**：收发器延迟补偿 (Transceiver Delay Compensation)
0：禁止收发器延迟补偿
1：使能收发器延迟补偿

位 22:21 保留

位 20:16 **DBRP**：数据比特率预分频器 (Data Bit Rate Prescaler)
生成位时间片时将振荡器频率除以的值。位时间为该时间片的倍数。比特率预分频器的有效值为 0 到 1023。硬件将该值解析为编程值加 1。

位 15:13 保留

位 12:8 **DTSEG1**：采样点之前的数据时间段 (Data time segment before sample point)
有效值为 1 到 15。实际上，硬件会将该值解析为编程值加 1。

位 7:4 **DTSEG1**：采样点之后的数据时间段 (Data time segment after sample point)
有效值为 1 到 7。实际上，硬件会将该值解析为编程值加 1。

位 3:0 **DSJW**：同步跳转宽度 (Synchronization Jump Width)
有效值为 0 到 15。实际上，硬件会将该值解析为编程值加 1。

注： *如果 FDCAN 时钟为 8 MHz，复位值 0x00000A33 会将 FDCAN 的比特率配置为较快的 500 kb/s。*

56.4.4 FDCAN 测试寄存器 (FDCAN_TEST)

FDCAN Test Register

必须通过将 CCCR[TEST] 位置“1”的方式使能对测试寄存器的写访问。CCCR[TEST] 位复位时，所有测试寄存器功能都会设为其复位值。

接收引脚 FDCANx_TX 的环回模式和软件控制属于硬件测试模式。将 TX 编程为“00”以外的值可能会干扰 CAN 总线上的消息传输。

地址：0x0010

复位值：0x0000 0000

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

位 31:8 保留

位 7 **RX**：接收引脚 (Receive Pin)
监视发送引脚 FDCANx_RX 的实际值
0：CAN 总线为显性（FDCANx_RX = “0”）
1：CAN 总线为隐性（FDCANx_RX = “1”）

位 6:5 **TX**：发送引脚控制 (Control of Transmit Pin)
00：复位值，FDCANx_TX 由 CAN 内核控制，会在 CAN 位时间结束时更新
01：可在引脚 FDCANx_TX 监控采样点
10：引脚 FDCANx_TX 上为显性（“0”）电平
11：引脚 FDCANx_TX 上为隐性（“1”）电平

位 4 **LBCK**：环回模式（Loop Back mode)
0：复位值，禁止环回模式
1：使能环回模式（请参见*测试模式*）

位 3:0 保留

56.4.5 FDCAN RAM 看门狗寄存器 (FDCAN_RWD)

FDCAN RAM Watchdog Register

RAM 看门狗监视消息 RAM 的 READY 输出。对消息 RAM 进行访问会启动消息 RAM 看门狗计数器，计数器的起始值由 RWD[WDC] 位配置。

当消息 RAM 通过激活其 READY 输出指示访问成功完成时，计数器会重新载入 RWD[WDC] 位的值。如果计数器递减为 0 之前消息 RAM 没有发出响应，则计数器会停止计数，中断标志 IR[WDI] 位置 1。RAM 看门狗计数器由 fdcan_pclk 时钟驱动。

地址：0x0014

复位值：0x0000 0000

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

位 31:16 保留

位 15:8 **WDV**：看门狗值 (Watchdog value)
实际消息 RAM 看门狗计数器值。

位 7:0 **WDC**：看门狗配置 (Watchdog configuration)
消息 RAM 看门狗计数器的起始值。如果使用复位值“00”，计数器会禁止。
这些位受写保护 (P)，仅当 CCCR 寄存器的位 1 [CCE] 和位 0 [INIT] 置“1”时，才能进行写访问。

56.4.6 FDCAN CC 控制寄存器 (FDCAN_CCCR)

FDCAN CC Control Register

地址：0x0018

复位值：0x0000 0001

有关各个位置 1 和复位的详细信息，请参见*软件初始化*。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
r															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NISO	TXP	EFBI	PHXD	Res.	Res.	BRSE	FDOE	TEST	DAR	MON	CSR	CSA	ASM	CCE	INIT
rw	rw	rw	rw			rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:16 保留

位 15 **NISO**：非 ISO 操作 (Non ISO Operation)
如果此位置 1，FDCAN 会使用 Bosch CAN FD 规范 V1.0 规定的 CAN FD 帧格式。
0：符合 ISO11898-1 规定的 CAN FD 帧格式
1：符合 Bosch CAN FD 规范 V1.0 规定的 CAN FD 帧格式

位 14 **TXP**
如果此位置 1，成功发送帧后，FDCAN 会先暂停两个 CAN 位时间，然后再开始进行下一次发送。
0：禁止
1：使能

位 13 **EFBI**：总线同步期间的边沿过滤 (Edge Filtering during Bus Integration)
0：禁止边沿过滤
1：需要两个连续显性 tq 才能检测硬同步边沿

位 12 **PXHD**：协议异常处理禁止 (Protocol Exception Handling Disable)
0：使能协议异常处理
1：禁止协议异常处理

位 11:10 保留

位 9 **BSE**：FDCAN 比特率切换 (FDCAN Bit Rate Switching)
0：禁止发送时进行比特率切换
1：使能发送时进行比特率切换

位 8 **FDOE**：FD 操作使能 (FD Operation Enable)
0：禁止 FD 操作
1：使能 FD 操作

位 7 **TEST**：测试模式使能 (Test Mode Enable)
0：正常操作，寄存器 TEST 保存复位值
1：测试模式，使能对寄存器 TEST 的写访问

位 6 **DAR**：禁止自动重发送 (Disable Automatic Retransmission)
0：使能自动重发送未成功发送的消息
1：禁止自动重发送

位 5 **MON**：总线监控模式 (Bus Monitoring Mode)
仅当 CCE 和 INIT 均置“1”时，才能通过软件将 MON 位置 1。此位可随时通过主机复位。
0：禁止总线监控模式
1：使能总线监控模式

位 4 **CSR**：时钟停止请求 (Clock Stop Request)
0：未请求时钟停止
1：已请求时钟停止。如果请求时钟停止，则在所有挂起的传输请求均已完成且 CAN 总线达到空闲状态之后，INIT 会先置 1，然后 CSA 会置 1。

位 3 **CSA**：时钟停止确认 (Clock Stop Acknowledge)
0：未确认时钟停止
1：可通过停止 APB 时钟和内核时钟将 FDCAN 设为掉电状态

位 2 **ASM**：ASM 受限工作模式 (ASM Restricted Operation Mode)
受限工作模式可用于根据不同 CAN 比特率对自身进行调整的应用中。应用会测试不同比特率，并在收到有效帧后退出受限工作模式。在可选的受限工作模式下，节点能够发送和接收数据帧和远程帧，并能对有效帧进行确认，但不会发送有效错误帧或过载帧。如果存在错误条件或过载条件，则不会发送显性位，而会等待出现总线空闲条件，以便使自身与 CAN 通信重新同步。错误计数器不会递增。仅当 CCE 和 INIT 均置“1”时，才能通过软件将 ASM 位置 1。此位可随时通过软件复位。
如果 FDCAN 连接到 CAN 时钟校准单元，只要校准未完成，ASM 位就会由硬件置 1。
0：正常 CAN 操作
1：受限工作模式激活

位 1 **CCE**：配置更改使能 (Configuration Change Enable)
0：CPU 对受保护的配置寄存器没有写访问权限
1：CPU 对受保护的配置寄存器有写访问权限（CCCR.INIT =“1”时）

位 0 **INIT**：初始化 (Initialization)
0：正常工作
1：启动初始化

注： *由于两个时钟域之间存在同步机制，可能会经过一定的延迟之后才能读取写入 INIT 的值。因此，编程人员必须确保之前写入 INIT 中的值已通过读取 INIT 的方式被接受，然后再将 INIT 设为新值。*

56.4.7 FDCAN 标称位定时和预分频器寄存器 (FDCAN_NBTP)

FDCAN Nominal Bit Timing and Prescaler Register

地址：0x001C

复位值：0x0000 0A33

仅当 CCCR[CCE] 和 CCCR[INIT] 位置 1 时，才可以向此寄存器写入数据。CAN 位时间的可编程范围为 4 到 81 个 tq。CAN 时间片的可编程范围为 1 到 1024 个 FDCAN 内核时钟周期。

tq = (BRP + 1) 个 FDCAN 时钟周期 m_ttcan_cclk

NTSEG1 为 Prop_Seg 与 Phase_Seg1 之和。NTSEG2 为 Phase_Seg2。因此，位时间的长度为（编程值）[NTSEG1 + NTSEG2 + 3] tq 或（函数值）[Sync_Seg + Prop_Seg + Phase_Seg1 + Phase_Seg2] tq。

信息处理时间 (IPT) 为零，这意味着下一位的数据在采样点后的第一个时钟边沿可用。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NSJW[6:0]							NBRP[8:0]								
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NTSEG1[7:0]								Res.	TSEG2[6:0]						
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:25 **NSJW**：标称（重新）同步跳转宽度 (Nominal (Re)Synchronization Jump Width)
有效值为 1 到 127。实际上，硬件会将该值解析为编程值加 1。
这些位受写保护 (P)，仅当 CCCR 寄存器的位 1 [CCE] 和位 0 [INIT] 置“1”时，才能进行写访问。

位 24:16 **NBRP**：比特率预分频器 (Bit Rate Prescaler)
生成位时间片时将振荡器频率除以的值。位时间为该时间片的倍数。有效值为 0 到 511。实际上，硬件会将该值解析为编程值加 1。
这些位受写保护 (P)，仅当 CCCR 寄存器的位 1 [CCE] 和位 0 [INIT] 置“1”时，才能进行写访问。

位 15:8 **NTSEG1**：采样点之前的标称时间段 (Nominal time segment before sample point)
有效值为 1 到 255。实际上，硬件会将该值解析为编程值加 1。
这些位受写保护 (P)，仅当 CCCR 寄存器的位 1 [CCE] 和位 0 [INIT] 置“1”时，才能进行写访问。

位 7 保留

位 6:0 **TSEG2**：采样点之后的标称时间段 (Nominal time segment after sample point)
有效值为 1 到 127。实际上，硬件会将该值解析为编程值加 1。

注： *如果 CAN 内核时钟为 8 MHz，复位值 0x00000A33 会将 FDCAN 的比特率配置为 500 kb/s。*

56.4.8 FDCAN 时间戳计数器配置寄存器 (FDCAN_TSCC)

FDCAN Timestamp Counter Configuration Register

地址：0x0020

复位值：0x0000 0000

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

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

位 31:20 保留

位 19:16 **TCP**：时间戳计数器预分频器 (Timestamp Counter Prescaler)

将时间戳和超时计数器时间单位配置为 CAN 位时间的倍数 [1…16]。

实际上，硬件会将该值解析为编程值加 1。

在 CAN FD 模式下，由于仲裁阶段与数据阶段之间的 CAN 位时间不同，内部时间戳计数器 TCP 不会提供恒定的时基。因此，CAN FD 需要使用外部计数器生成时间戳（TSS = “10”）。

这些位受写保护 (P)，仅当 CCCR 寄存器的位 1 [CCE] 和位 0 [INIT] 置“1”时，才能进行写访问。

位 15:2 保留

位 1:0 **TSS**：采样点之前的标称时间段 (Nominal time segment before sample point)

有效值为 1 到 255。实际上，硬件会将该值解析为编程值加 1。

这些位受写保护 (P)，仅当 CCCR 寄存器的位 1 [CCE] 和位 0 [INIT] 置“1”时，才能进行写访问。

56.4.9 FDCAN 时间戳计数器值寄存器 (FDCAN_TSCV)

FDCAN Timestamp Counter Value Register

地址：0x0024

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TSC[15:0]															
w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c

位 31:16 保留

位 15:0 **TSC**：时间戳计数器 (Timestamp Counter)

内部/外部时间戳计数器值是在（接收和发送）帧开始时捕获的。当 TSCC[TSS] = “01”时，时间戳计数器会以 CAN 为时间的倍数 [1…16] 递增，具体取决于 TSCC[TCP] 的配置。回卷操作会将中断标志 IR[TSW] 置 1。写访问会将计数器复位为 0。当 TSCC.TSS = “10”时，TSC 会反映外部时间戳计数器值。写访问没有影响。

注： *“回卷”是指时间戳计数器值不是由于对 TSCV 进行写访问的原因而由非零值变为 0。*

56.4.10 FDCAN 超时计数器配置寄存器 (FDCAN_TOCC)

FDCAN Timeout Counter Configuration Register

地址：0x0028

复位值：0xFFFF 0000

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

位 31:16 **TOP**：超时时长 (Timeout Period)

超时计数器（递减计数器）的起始值。配置超时时长。

位 15:3 保留

位 2:1 **TOS**：超时选择 (Timeout Select)

在连续模式下工作时，对 TOCV 进行写访问会将计数器预设为由 TOCC[TOP] 配置的值并继续递减计数。超时计数器由其中一个 FIFO 控制时，空 FIFO 会将计数器预设为由 TOCC[TOP] 配置的值。存储了第一个 FIFO 元素后，会开始递减计数。

00：连续工作

01：超时由发送事件 FIFO 控制

10：超时由接收 FIFO 0 控制

11：超时由接收 FIFO 1 控制

这些位受写保护 (P)，仅当 CCCR 寄存器的位 1 [CCE] 和位 0 [INIT] 置“1”时，才能进行写访问。

位 0 **ETOC**：使能超时计数器 (Enable Timeout Counter)

0：禁止超时计数器

1：使能超时计数器

此位受写保护 (P)，仅当 CCCR 寄存器的位 1 [CCE] 和位 0 [INIT] 置“1”时，才能进行写访问。

有关更多详细信息，请参见*超时计数器*。

56.4.11 FDCAN 超时计数器值寄存器 (FDCAN_TOCV)

FDCAN Timeout Counter Value Register

地址：0x002C

复位值：0x0000 FFFF

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TOC[15:0]															
w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c

位 31:16 保留

位 15:0 **TOC**：超时计数器 (Timeout Counter)

超时计数器会以 CAN 位时间的倍数 [1…16] 递减，具体取决于 TSCC.TCP 的配置。当计数器递减至 0 时，中断标志 IR.TOO 会置 1，超时计数器会停止计数。开始和复位/重启条件是通过 TOCC.TOS 配置的。

56.4.12 FDCAN 错误计数器寄存器 (FDCAN_ECR)

FDCAN Error Counter Register

地址：0x0040

复位值：0x0000 0000

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

位 31:24 保留

位 23:16 **CEL**：CAN 错误记录 (CAN Error Logging)

每当 CAN 协议错误导致发送错误计数器或接收错误计数器递增时，计数器都会递增。对 CEL 进行读访问时，计数器会复位。计数器值达到 0xFF 时，会停止计数；TEC 或 REC 下一次递增时，中断标志 IR[ELO] 会置 1。

访问类型为 RX：进行读访问时复位。

位 15 **RP**：接收错误被动 (Receive Error Passive)

0：接收错误计数器低于错误被动级别 128

1：接收错误计数器已达到错误被动级别 128

位 14:8 **TREC**：接收错误计数器 (Receive Error Counter)

接收错误计数器的实际状态，取值范围为 0 到 127。

位 7:0 **TEC**：发送错误计数器 (Transmit Error Counter)

发送错误计数器的实际状态，取值范围为 0 到 255。

如果 CCCR.ASM 置 1，则检测到 CAN 协议错误时，CAN 协议控制器不会使 TEC 和 REC 递增，但 CEL 仍会递增。

56.4.13 FDCAN 协议状态寄存器 (FDCAN_PSR)

FDCAN Protocol Status Register

地址：0x0044

复位值：0x0000 0707

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TDCV[6:0]						
									rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	PXE	REDL	RBRS	RESI	DLEC[2:0]			BO	EW	EP	ACT[1:0]		LEC[2:0]		
	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:23 保留

位 22:16 **TDCV**：发送器延迟补偿值 (Transmitter Delay Compensation Value)
第二采样点的位置，由从 m_can_tx 到 m_can_rx 之间测得的延迟与 TDCR.TDCO 之和定义。在数据阶段，SSP 位置为已发送位起点与第二采样点之间的最小时间片 (mtq) 数。有效值为 0 到 127 个 mtq。

位 15 保留

位 14 **PXE**：协议异常事件 (Protocol Exception Event)
0：自上次读访问起未发生协议异常事件
1：已发生协议异常事件

位 13 **REDL**：已接收 FDCAN 消息 (Received FDCAN Message)
此位的设置与验收过滤无关。
0：由于此位是通过 CPU 复位的，因此尚未接收到 FDCAN 消息
1：已接收到 EDL 标志置 1 的 FDCAN 格式的消息
访问类型为 RX：进行读访问时复位。

位 12 **RBRS**：上次接收的 FDCAN 消息的 BRS 标志 (BRS flag of last received FDCAN Message)
此位与 REDL 一起设置，与验收过滤无关。
0：上次接收的 FDCAN 消息的 BRS 标志未置 1
1：上次接收的 FDCAN 消息的 BRS 标志已置 1
访问类型为 RX：进行读访问时复位。

位 11 **RESI**：上次接收的 FDCAN 消息的 ESI 标志 (ESI flag of last received FDCAN Message)
此位与 REDL 一起设置，与验收过滤无关。
0：上次接收的 FDCAN 消息的 ESI 标志未置 1
1：上次接收的 FDCAN 消息的 ESI 标志已置 1
访问类型为 RX：进行读访问时复位。

位 10:8 **DLEC**：数据上一错误代码 (Data Last Error Code)
在 BRS 标志置 1 的 FDCAN 格式帧数据阶段发生的上一错误类型。代码与 LEC 相同。当 BRS 标志置 1 的 FDCAN 格式帧已无错传输（接收或发送）时，此位域将被清零。
访问类型为 RS：进行读访问时置 1。

位 7 **BO**：Bus_Off 状态 (Bus_Off Status)
0：FDCAN 未处于 Bus_Off 状态
1：FDCAN 处于 Bus_Off 状态

位 6 **EW**：警告状态 (Warning Status)

0：两个错误计数器的值均小于 Error_Warning 限值 96

1：至少有一个错误计数器已达到 Error_Warning 限值 96

位 5 **EP**：错误被动 (Error Passive)

0：FDCAN 处于 Error_Active 状态。此位通常会参与总线通信，并会在检测到错误后发送主动错误标志

1：FDCAN 处于 Error_Passive 状态

位 4:3 **ACT**：活动 (Activity)

监控模块的 CAN 通信状态。

00：同步中：节点在 CAN 通信时同步

01：空闲：节点既不是接收器，也不是发送器

10：接收器：节点作为接收器工作

11：发送器：节点作为发送器工作

位 2:0 **LEC**：上一错误代码 (Last Error Code)

LEC 指示 CAN 总线上发生的上一错误的类型。消息已无错传输后（接收或发送），此位域将被清零。

000：无错：自消息成功接收或发送而将 LEC 复位后，未发生任何错误。

001：内容错误：在已接收的消息中，有一部分连续出现 5 个以上的相等位，这种情况是不允许发生的。

010：格式错误：已接收帧的固定格式部分的格式不正确。

011：AckError：由 FDCAN 发送的消息未被另一节点确认。

100：Bit1Error：在消息发送过程中（仲裁字段例外），器件希望发送隐性电平（位逻辑值为“1”），但受监控的总线值为显性。

101：Bit0Error：在消息发送过程中（或者确认位，或者主动错误标志，或者过载标志），器件希望发送显性电平（数据或标识符位逻辑值“0”），但受监控的总线值为隐性。在 Bus_Off 恢复期间，每次监控到 11 个隐性位组成的序列时，此状态都会置 1。这样，CPU 便可监控 Bus_Off 恢复序列的进行（指示总线并未保持显性或持续受到干扰）。

110：CRCError：已接收消息的 CRC 校验和不正确。传入消息的 CRC 与通过已接收数据计算出的 CRC 不匹配。

111：NoChange：如果对协议状态寄存器进行读访问，则会将 LEC 重新初始化为“7”。如果 LEC 显示的值为“7”，则自 CPU 上一次对协议状态寄存器进行读访问起，未检测到任何 CAN 总线事件。

访问类型为 RS：进行读访问时置 1。

注： *如果采用 FDCAN 格式的帧已到达 BRS 标志置 1 的数据阶段，下一 CAN 事件（错误或有效帧）将显示在 FLEC 中，而不显示在 LEC 中。FDCAN CRC 序列中的固定内容位中的错误将显示为格式错误，而不是内容错误*

注： *Bus_Off 恢复序列（请参见 CAN 规范第 2.0 版或 ISO11898-1）不能通过将 CCCR[INIT] 置 1 或复位来缩短。如果器件进入 Bus_Off 状态，则会将其自身的 CCCR.INIT 置 1，从而会停止所有总线活动。一旦 CCCR[INIT] 由 CPU 清零，器件随后将等待总线空闲状态出现 129 次（129 × 11 个连续隐性位），然后才会恢复正常操作。Bus_Off 恢复序列结束时，错误管理计数器将复位。在 CCCR[INIT] 复位后的等待时间内，每次监控到由 11 个隐性位组成的序列时，Bit0 错误代码都会写入 PSR[LEC] 中，从而使 CPU 能够准备好检查 CAN 总线保持显性还是持续受到干扰，并能够监控 Bus_Off 恢复序列。ECR[REC] 用于对这些序列进行计数。*

56.4.14 FDCAN 发送器延迟补偿寄存器 (FDCAN_TDCR)

FDCAN Transmitter Delay Compensation Register

地址：0x0048

复位值：0x0000 0000

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

位 31:15 保留

位 14:8 **TDCO**：发送器延迟补偿偏移 (Transmitter Delay Compensation Offset)
定义从 FDCAN_TX 到 FDCAN_RX 测得的延迟与第二采样点之间距离的偏移值。有效值为 0 到 127 个 mtq。

位 7 保留

位 6:0 **TDCF**：发送器延迟补偿过滤器窗口长度 (Transmitter Delay Compensation Filter Window Length)
定义 SSP 位置的最小值，要测量发送器延迟，会忽略 FDCAN_RX 上会导致 SSP 位置提前的显性边沿。

56.4.15 FDCAN 中断寄存器 (FDCAN_IR)

FDCAN Interrupt Register

如果检测到其中一个列出的条件（边沿有效），标志会置 1。在主机将标志清零之前，标志保持置 1 状态。通过向对应位位置写入“1”将标志清零。

写入“0”则不会带来任何影响。硬复位会将寄存器清零。IE 的配置控制着是否生成中断。ILS 的配置控制着是哪条中断线发出中断。

地址：0x0050

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	ARA	PED	PEA	WDI	BO	EW	EP	ELO	Res.	Res.	DRX	TOO	MRAF	TSW
		rw	rw	rw	rw	rw	rw	rw	rw			rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TEFL	TEFF	TEFW	TEFN	TFE	TCF	TC	HPM	RF1L	RF1F	RF1W	RF1N	RF0L	RF0F	RF0W	RF0N
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:30 保留

位 29 **ARA**：访问保留地址 (Access to Reserved Address)
0：未对保留地址进行访问
1：已对保留地址进行访问

位 28 **PED**：数据阶段的协议错误（使用数据位时间）(Protocol Error in Data Phase (Data Bit Time is used))
0：数据阶段没有协议错误
1：检测到数据阶段有协议错误（PSR.DLEC 不是 0、7）

位 27 **PEA**：仲裁阶段中的协议错误（使用标称位时间）(Protocol Error in Arbitration Phase (Nominal Bit Time is used))
0：仲裁阶段中没有协议错误
1：检测到仲裁阶段有协议错误（PSR.LEC 不是 0、7）

位 26 **WDI**：看门狗中断 (Watchdog Interrupt)
0：未发生消息 RAM 看门狗事件
1：因 READY 缺失而发生消息 RAM 看门狗事件

位 25 **BO**：Bus_Off 状态 (Bus_Off Status)
0：Bus_Off 状态未更改
1：Bus_Off 状态已更改

位 24 **EW**：警告状态 (Warning Status)
0：Error_Warning 状态未更改
1：Error_Warning 状态已更改

位 23 **EP**：错误被动 (Error Passive)
0：Error_Passive 状态未更改
1：Error_Passive 状态已更改

位 22 **ELO**：错误记录溢出 (Error Logging Overflow)
0：CAN 错误记录计数器未溢出
1：CAN 错误记录计数器已溢出

位 21:20 保留

位 19 **DRX**：消息存储到专用接收缓冲区 (Message stored to Dedicated Rx Buffer)
接收到的消息已存储到专用接收缓冲区后，此标志会置 1。
0：未更新接收缓冲区
1：至少有一条已接收消息存储到接收缓冲区中

位 18 **TOO**：发生超时 (Timeout Occurred)
0：无超时
1：达到超时

位 17 **MRAF**：消息 RAM 访问失败 (Message RAM Access Failure)

如果接收处理单元未在已接收以下消息的仲裁字段之前完成对已接收消息的验收过滤或存储，

- 此标志会置 1。在这种情况下，验收过滤或消息存储会中止，接收处理单元会开始处理以下消息。
- 无法向消息 RAM 写入消息。在这种情况下，会中止消息存储。

在这两种情况下，FIFO 放入索引不会更新，专用接收缓冲区的新数据标志也不会置 1。下一条消息存储到此位置时，部分存储的消息会被覆盖。

当发送处理单元无法及时从消息 RAM 中读取数据时，此标志也会置 1。在这种情况下，消息发送会中止。如果发送处理单元访问失败，M_TTCAN 会切换到受限工作模式（请参见*受限工作模式*）。要退出受限工作模式，主机 CPU 必须复位 CCCR.ASM。

0：未发生消息 RAM 访问失败

1：发生消息 RAM 访问失败

位 16 **TSW**：时间戳回卷 (Timestamp Wraparound)

0：时间戳未回卷

1：时间戳已回卷

位 15 **TEFL**：发送事件 FIFO 元素丢失 (Tx Event FIFO Element Lost)

0：发送事件 FIFO 元素未丢失

1：发送事件 FIFO 元素已丢失，尝试向大小为零的发送事件 FIFO 进行写入时也会置 1

位 14 **TEFF**：发送事件 FIFO 已满 (Tx Event FIFO Full)

0：发送事件 FIFO 未满

1：发送事件 FIFO 已满

位 13 **TEFW**：达到发送事件 FIFO 水印 (Tx Event FIFO Watermark Reached)

0：发送事件 FIFO 填充级别低于水印

1：发送事件 FIFO 填充级别达到水印

位 12 **TEFN**：发送事件 FIFO 新条目 (Tx Event FIFO New Entry)

0：发送事件 FIFO 未更改

1：发送处理单元写入发送事件 FIFO 元素

位 11 **TFE**：发送 FIFO 为空 (Tx FIFO Empty)

0：发送 FIFO 非空

1：发送 FIFO 为空

位 10 **TCF**：发送取消完成 (Transmission Cancellation Finished)

0：发送取消未完成

1：发送取消已完成

位 9 **TC**：发送完成 (Transmission Completed)

0：发送未完成

1：发送已完成

位 8 **HPM**：高优先级消息 (High Priority Message)

0：未接收到任何高优先级消息

1：已接收到高优先级消息

位 7 **RF1L**：接收 FIFO 1 消息丢失 (Rx FIFO 1 Message Lost)

0：接收 FIFO 1 消息未丢失

1：接收 FIFO 1 消息已丢失，尝试向大小为零的接收 FIFO 1 进行写入后也会置 1

位 6 **RF1F**：接收 FIFO 1 已满 (Rx FIFO 1 Full)
0：接收 FIFO 1 未满
1：接收 FIFO 1 已满

位 5 **RF1W**：达到接收 FIFO 1 水印 (Rx FIFO 1 Watermark Reached)
0：接收 FIFO 1 填充级别低于水印
1：接收 FIFO 1 填充级别达到水印

位 4 **RF1N**：接收 FIFO 1 新消息 (Rx FIFO 1 New Message)
0：未向接收 FIFO 1 写入新消息
1：已向接收 FIFO 1 写入新消息

位 3 **RF0L**：接收 FIFO 0 消息丢失 (Rx FIFO 0 Message Lost)
0：接收 FIFO 0 消息未丢失
1：接收 FIFO 0 消息已丢失，尝试向大小为零的接收 FIFO 0 进行写入后也会置 1

位 2 **RF0F**：接收 FIFO 0 已满 (Rx FIFO 0 Full)
0：接收 FIFO 0 未满
1：接收 FIFO 0 已满

位 1 **RF0W**：达到接收 FIFO 0 水印 (Rx FIFO 0 Watermark Reached)
0：接收 FIFO 0 填充级别低于水印
1：接收 FIFO 0 填充级别达到水印

位 0 **RF0N**：接收 FIFO 0 新消息 (Rx FIFO 0 New Message)
0：未向接收 FIFO 0 写入新消息
1：已向接收 FIFO 0 写入新消息

56.4.16 FDCAN 中断使能寄存器 (FDCAN_IE)

FDCAN Interrupt Enable Register

中断使能寄存器中的设置决定了将在中断线上指示中断寄存器中的哪些状态更改。

地址：0x0054

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	ARAE	PEDE	PEAE	WDIE	BOE	EWE	EPE	ELOE	BEUE	BECE	DRXE	TOOE	MRAFE	TSWE
		rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TEFLE	TEFFE	TEFWE	TEFNE	TFEE	TCFE	TCE	HPME	RF1LE	RF1FE	RF1WE	RF1NE	RF0LE	RF0FE	RF0WE	RF0NE
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:30 保留

位 29 **ARAE**：访问保留地址使能 (Access to Reserved Address Enable)

位 28 **PEDE**：数据阶段的协议错误使能 (Protocol Error in Data Phase Enable)

位 27 **PEAE**：仲裁阶段中的协议错误使能 (Protocol Error in Arbitration Phase Enable)

位 26 **WDIE**：看门狗中断使能 (Watchdog Interrupt Enable)
0：禁止中断
1：使能中断

位 25 **BOE**：Bus_Off 状态 (Bus_Off Status)
0：禁止中断
1：使能中断

位 24 **EWE**：警告状态中断使能 (Warning Status Interrupt Enable)
0：禁止中断
1：使能中断

位 23 **EPE**：错误被动中断使能 (Error Passive Interrupt Enable)
0：禁止中断
1：使能中断

位 22 **ELOE**：错误记录溢出中断使能 (Error Logging Overflow Interrupt Enable)
0：禁止中断
1：使能中断

位 21 **BEUE**：位错误未更正中断使能 (Bit Error Uncorrected Interrupt Enable)
0：禁止中断
1：使能中断

位 20 **BECE**：位错误已更正中断使能 (Bit Error Corrected Interrupt Enable)
0：禁止中断
1：使能中断

位 19 **DRXE**：消息存储到专用接收缓冲区中断使能 (Message stored to Dedicated Rx Buffer Interrupt Enable)
0：禁止中断
1：使能中断

位 18 **TOOE**：超时已发生中断使能 (Timeout Occurred Interrupt Enable)
0：禁止中断
1：使能中断

位 17 **MRAFE**：消息 RAM 访问失败中断使能 (Message RAM Access Failure Interrupt Enable)
0：禁止中断
1：使能中断

位 16 **TSWE**：时间戳回卷中断使能 (Timestamp Wraparound Interrupt Enable)
0：禁止中断
1：使能中断

位 15 **TEFLE**：发送事件 FIFO 元素丢失中断使能 (Tx Event FIFO Element Lost Interrupt Enable)
0：禁止中断
1：使能中断

位 14 **TEFFE**：发送事件 FIFO 已满中断使能 (Tx Event FIFO Full Interrupt Enable)
0：禁止中断
1：使能中断

位 13 **TEFWE**：达到发送事件 FIFO 水印中断使能 (Tx Event FIFO Watermark Reached Interrupt Enable)
0：禁止中断
1：使能中断

位 12 **TEFNE**：发送事件 FIFO 新条目中断使能 (Tx Event FIFO New Entry Interrupt Enable)
0：禁止中断
1：使能中断

位 11 **TFEE**：发送 FIFO 为空中断使能 (Tx FIFO Empty Interrupt Enable)
0：禁止中断
1：使能中断

位 10 **TCFE**：发送取消完成中断使能 (Transmission Cancellation Finished Interrupt Enable)
0：禁止中断
1：使能中断

位 9 **TCE**：发送完成中断使能 (Transmission Completed Interrupt Enable)
0：禁止中断
1：使能中断

位 8 **HPME**：高优先级消息中断使能 (High Priority Message Interrupt Enable)
0：禁止中断
1：使能中断

位 7 **RF1LE**：接收 FIFO 1 消息丢失中断使能 (Rx FIFO 1 Message Lost Interrupt Enable)
0：禁止中断
1：使能中断

位 6 **RF1FE**：接收 FIFO 1 已满中断使能 (Rx FIFO 1 Full Interrupt Enable)
0：禁止中断
1：使能中断

位 5 **RF1WE**：达到接收 FIFO 1 水印中断使能 (Rx FIFO 1 Watermark Reached Interrupt Enable)
0：禁止中断
1：使能中断

位 4 **RF1NE**：接收 FIFO 1 新消息中断使能 (Rx FIFO 1 New Message Interrupt Enable)
0：禁止中断
1：使能中断

位 3 **RF0LE**：接收 FIFO 0 消息丢失中断使能 (Rx FIFO 0 Message Lost Interrupt Enable)
0：禁止中断
1：使能中断

位 2 **RF0FE**：接收 FIFO 0 已满中断使能 (Rx FIFO 0 Full Interrupt Enable)
0：禁止中断
1：使能中断

位 1 **RF0WE**：达到接收 FIFO 0 水印中断使能 (Rx FIFO 0 Watermark Reached Interrupt Enable)
0：禁止中断
1：使能中断

位 0 **RF0NE**：接收 FIFO 0 新消息中断使能 (Rx FIFO 0 New Message Interrupt Enable)
0：禁止中断
1：使能中断

56.4.17 FDCAN 中断线选择寄存器 (FDCAN_ILS)

FDCAN Interrupt Line Select Register

中断线选择寄存器将特定中断标志生成的中断从中断寄存器分配到两个模块中断线之一。要生成中断，必须通过 ILE[EINT0] 和 ILE[EINT1] 使能相应的中断线。

地址：0x0058

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	ARAL	PEDL	PEAL	WDIL	BOL	EWL	EPL	ELOL	BEUL	BECL	DRXL	TOOL	MRAFL	TSWL
		rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TEFLL	TEFFL	TEFWL	TEFNL	TFEL	TCFL	TCL	HPML	RF1LL	RF1FL	RF1WL	RF1NL	RF0LL	RF0FL	RF0WL	RF0NL
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:30 保留

位 29 **ARAL**：访问保留地址线 (Access to Reserved Address Line)

位 28 **PEDL**：数据阶段的协议错误线 (Protocol Error in Data Phase Line)

位 27 **PEAL**：仲裁阶段的协议错误线 (Protocol Error in Arbitration Phase Line)

位 26 **WDIL**：看门狗中断线 (Watchdog Interrupt Line)

位 25 **BOL**：Bus_Off 状态 (Bus_Off Status)

位 24 **EWL**：警告状态中断线 (Warning Status Interrupt Line)

位 23 **EPL**：错误被动中断线 (Error Passive Interrupt Line)

位 22 **ELOL**：错误记录溢出中断线 (Error Logging Overflow Interrupt Line)

位 21 **BEUL**：位错误未更正中断线 (Bit Error Uncorrected Interrupt Line)

位 20 **BECL**：位错误已更正中断线 (Bit Error Corrected Interrupt Line)

位 19 **DRXL**：消息存储到专用接收缓冲区中断线 (Message stored to Dedicated Rx Buffer Interrupt Line)

位 18 **TOOL**：超时已发生中断线 (Timeout Occurred Interrupt Line)

位 17 **MRAFL**：消息 RAM 访问失败中断线 (Message RAM Access Failure Interrupt Line)

位 16 **TSWL**：时间戳回卷中断线 (Timestamp Wraparound Interrupt Line)

位 15 **TEFLL**：发送事件 FIFO 元素丢失中断线 (Tx Event FIFO Element Lost Interrupt Line)

位 14 **TEFFL**：发送事件 FIFO 已满中断线 (Tx Event FIFO Full Interrupt Line)

位 13 **TEFWL**：达到发送事件 FIFO 水印中断线 (Tx Event FIFO Watermark Reached Interrupt Line)

位 12 **TEFNL**：发送事件 FIFO 新条目中断线 (Tx Event FIFO New Entry Interrupt Line)

位 11 **TFEL**：发送 FIFO 为空中断线 (Tx FIFO Empty Interrupt Line)

位 10 **TCFL**：发送取消完成中断线 (Transmission Cancellation Finished Interrupt Line)

位 9 **TCL**：发送完成中断线 (Transmission Completed Interrupt Line)

位 8 **HPML**：高优先级消息中断线 (High Priority Message Interrupt Line)

位 7 **RF1LL**：接收 FIFO 1 消息丢失中断线 (Rx FIFO 1 Message Lost Interrupt Line)

位 6 **RF1FL**：接收 FIFO 1 已满中断线 (Rx FIFO 1 Full Interrupt Line)

位 5 **RF1WL**：达到接收 FIFO 1 水印中断线 (Rx FIFO 1 Watermark Reached Interrupt Line)

位 4 **RF1NL**：接收 FIFO 1 新消息中断线 (Rx FIFO 1 New Message Interrupt Line)

位 3 **RF0LL**：接收 FIFO 0 消息丢失中断线 (Rx FIFO 0 Message Lost Interrupt Line)

位 2 **RF0FL**：接收 FIFO 0 已满中断线 (Rx FIFO 0 Full Interrupt Line)

位 1 **RF0WL**：达到接收 FIFO 0 水印中断线 (Rx FIFO 0 Watermark Reached Interrupt Line)

位 0 **RF0NL**：接收 FIFO 0 新消息中断线 (Rx FIFO 0 New Message Interrupt Line)

56.4.18 FDCAN 中断线使能寄存器 (FDCAN_ILE)

FDCAN Interrupt Line Enable Register

通过对位 EINT0 和 EINT1 进行编程，可分别使能/禁止两条连接至 CPU 的中断线。

地址：0x005C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	EINT1	EINT0
														rw	rw

位 31:2 保留

位 1 **EINT1**：使能中断线 1 (Enable Interrupt Line 1)
0：禁止中断线 fdcan_intr0_it
1：使能中断线 fdcan_intr0_it

位 0 **EINT0**：使能中断线 0 (Enable Interrupt Line 0)
0：禁止中断线 fdcan_intr1_it
1：使能中断线 fdcan_intr1_it

56.4.19 FDCAN 全局过滤器配置寄存器 (FDCAN_GFC)

FDCAN Global Filter Configuration Register

用于消息 ID 过滤的全局设置。全局过滤器配置控制着标准和扩展消息的过滤器路径，请参见*图 730：标准消息 ID 过滤器路径*和*图 731：扩展消息 ID 过滤器路径*。

地址：0x0080

复位值：0x0000 0000

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

位 31:6 保留

位 5:4 **ANFS**：接受非匹配标准帧 (Accept Non-matching Frames Standard)

定义了如何处理与过滤器列表的任何元素都不匹配的 ID 为 11 位的已接收消息。

00：在接收 FIFO 0 中接受

01：在接收 FIFO 1 中接受

10：拒绝

11：拒绝

这些位受写保护 (P)，也就是说，仅当 CCCR 寄存器的位 1 [CCE] 和位 0 [INIT] 置“1”时，才能通过这些位进行写访问。

位 3:2 **ANFE**：接受非匹配扩展帧 (Accept Non-matching Frames Extended)

定义了如何处理与过滤器列表的任何元素都不匹配的 ID 为 29 位的已接收消息。

00：在接收 FIFO 0 中接受

01：在接收 FIFO 1 中接受

10：拒绝

11：拒绝

这些位受写保护 (P)，也就是说，仅当 CCCR 寄存器的位 1 [CCE] 和位 0 [INIT] 置“1”时，才能通过这些位进行写访问。

位 1 **RRFS**：拒绝标准远程帧 (Reject Remote Frames Standard)

0：过滤采用 11 位标准 ID 的远程帧

1：拒绝所有采用 11 位标准 ID 的远程帧

这些位受写保护 (P)，也就是说，仅当 CCCR 寄存器的位 1 [CCE] 和位 0 [INIT] 置“1”时，才能通过这些位进行写访问。

位 0 **RRFE**：拒绝扩展远程帧 (Reject Remote Frames Extended)

0：过滤采用 29 位标准 ID 的远程帧

1：拒绝所有采用 29 位标准 ID 的远程帧

这些位受写保护 (P)，也就是说，仅当 CCCR 寄存器的位 1 [CCE] 和位 0 [INIT] 置“1”时，才能通过这些位进行写访问。

56.4.20 FDCAN 标准 ID 过滤器配置寄存器 (FDCAN_SIDFC)

FDCAN Standard ID Filter Configuration Register

用于 11 位标准消息 ID 过滤的设置。标准 ID 过滤器配置控制着标准消息的过滤器路径，请参见*图 730：标准消息 ID 过滤器路径*。

地址：0x0084

复位值：0x0000 0000

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

位 31:24 保留

位 23:16 **LSS**：标准列表大小 (List Size Standard)

0：无标准消息 ID 过滤器

1-128：标准消息 ID 过滤器元素数量

>128：大于 128 的值会被解析为 128。

这些位受写保护 (P)，也就是说，仅当 CCCR 寄存器的位 1 [CCE] 和位 0 [INIT] 置“1”时，才能通过这些位进行写访问。

位 15:2 **FLSSA**：标准过滤器列表起始地址 (Filter List Standard Start Address)

标准消息 ID 过滤器列表的起始地址（32 位字地址，请参见*表 469：标准消息 ID 过滤器元素*）。这些位受写保护 (P)，也就是说，仅当 CCCR 寄存器的位 1 [CCE] 和位 0 [INIT] 置“1”时，才能通过这些位进行写访问。

位 1:0 保留

56.4.21 FDCAN 扩展 ID 过滤器配置寄存器 (FDCAN_XIDFC)

FDCAN Extended ID Filter Configuration Register

用于 29 位扩展消息 ID 过滤的设置。扩展 ID 过滤器配置控制着标准消息的过滤器路径，请参见*图 731：扩展消息 ID 过滤器路径*。

地址：0x0088

复位值：0x0000 0000

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

位 31:24 保留

位 23:16 **LSE**：扩展列表大小 (List Size Extended)

0：无标准消息 ID 过滤器

1-128：标准消息 ID 过滤器元素数量

>128：大于 128 的值会被解析为 128。

这些位受写保护 (P)，也就是说，仅当 CCCR 寄存器的位 1 [CCE] 和位 0 [INIT] 置“1”时，才能通过这些位进行写访问。

位 15:2 **FLESA**：扩展过滤器列表起始地址 (Filter List Extended Start Address)

标准消息 ID 过滤器列表的起始地址（32 位字地址，请参见*表 471：扩展消息 ID 过滤器元素*）。

这些位受写保护 (P)，也就是说，仅当 CCCR 寄存器的位 1 [CCE] 和位 0 [INIT] 置“1”时，才能通过这些位进行写访问。

位 1:0 保留

56.4.22 FDCAN 扩展 ID 和掩码寄存器 (FDCAN_XIDAM)

FDCAN Extended ID and Mask Register

地址：0x0090

复位值：0x1FFF FFFF

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

位 31:29 保留

位 28:0 **EIDM**：扩展 ID 掩码 (Extended ID Mask)

要对扩展帧进行验收过滤，会将扩展 ID 与掩码与已接收帧的消息 ID 进行与运算。用于屏蔽 SAE J1939 中的 29 位 ID。如果所有位的复位值均设为 1，则掩码无效。

这些位受写保护 (P)，也就是说，仅当 CCCR 寄存器的位 1 [CCE] 和位 0 [INIT] 置“1”时，才能通过这些位进行写访问。

56.4.23 FDCAN 高优先级消息状态寄存器 (FDCAN_HPMS)

FDCAN High Priority Message Status Register

每次消息 ID 过滤器元素配置为生成优先级事件匹配时，都会更新此寄存器。此寄存器可用于监控传入高优先级消息的状态，并可使能对这些消息的快速访问。

地址：0x0094

复位值：0x0000 0000

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

位 31:16 保留

位 15 **FLST**：过滤器列表 (Filter List)

指示匹配过滤器元素的过滤器列表。

0：标准过滤器列表

1：扩展过滤器列表

位 14:8 **FIDX**：过滤器索引 (Filter Index)

匹配过滤器元素的索引。范围为 0 到 SIDFC[LSS] - 1 或 XIDFC[LSE] - 1。

位 7:6 **MSI**：消息存储指示符 (Message Storage Indicator)

00：未选择 FIFO

01：FIFO 上溢

10：消息存储在 FIFO 0 中

11：消息存储在 FIFO 1 中

位 5:0 **BIDX**：缓冲区索引 (Buffer Index)

消息存储到的接收 FIFO 元素的索引。仅当 MSI[1] = “1”时有效。

56.4.24 FDCAN 新数据 1 寄存器 (FDCAN_NDAT1)

FDCAN New Data 1 Register

地址：0x0098

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ND31	ND30	ND29	ND28	ND27	ND26	ND25	ND24	ND23	ND22	ND21	ND20	ND19	ND18	ND17	ND16
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ND15	ND14	ND13	ND12	ND11	ND10	ND9	ND8	ND7	ND6	ND5	ND4	ND3	ND2	ND1	ND0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:0 **NDn**：新数据 [31:0] (New Data[31:0])

该寄存器保存接收缓冲区 0 到 31 的新数据标志。当相应的接收缓冲区通过已接收的帧进行更新时，这些标志会置 1。在主机将标志清零之前，标志保持置 1 状态。通过向对应位位置写入“1”将标志清零。写入“0”不会起任何作用。硬复位会将寄存器清零。

0：接收缓冲区未更新

1：接收缓冲区通过新消息更新

56.4.25 FDCAN 新数据 2 寄存器 (FDCAN_NDAT2)

FDCAN New Data 2 Register

地址：0x009C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ND63	ND62	ND61	ND60	ND59	ND58	ND57	ND56	ND55	ND54	ND53	ND52	ND51	ND50	ND49	ND48
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ND47	ND46	ND45	ND44	ND43	ND42	ND41	ND40	ND39	ND38	ND37	ND36	ND35	ND34	ND33	ND32
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:0 **NDn**：新数据 [63:32] (New Data[63:32])

该寄存器保存接收缓冲区 32 到 63 的新数据标志。当相应的接收缓冲区通过已接收的帧进行更新时，这些标志会置 1。在主机将标志清零之前，标志保持置 1 状态。通过向对应位位置写入“1”将标志清零。写入“0”不会起任何作用。硬复位会将寄存器清零。

0：接收缓冲区未更新

1：接收缓冲区通过新消息更新

56.4.26 FDCAN 接收 FIFO 0 配置寄存器 (FDCAN_RXF0C)

FDCAN Rx FIFO 0 Configuration Register

地址：0x00A0

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
F0OM	F0WM[7:0]							Res.	F0S[7:0]						
	rw	rw	rw	rw	rw	rw	rw		rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
F0SA[13:0]														Res.	Res.
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw		

位 31 **F0OM**：FIFO 0 工作模式 (FIFO 0 Operation mode)
FIFO 0 可工作在阻止模式或覆盖模式下。
0：FIFO 0 处于阻止模式
1：FIFO 0 处于覆盖模式

位 30:24 **F0WM**：FIFO 0 水印 (FIFO 0 Watermark)
0：禁止水印中断
1-64：接收 FIFO 0 水印中断 (IR[RF0W]) 的级别
>64：禁止水印中断
这些位受写保护 (P)，也就是说，仅当 CCCR 寄存器的位 1 [CCE] 和位 0 [INIT] 置“1”时，才能通过这些位进行写访问。

位 23 保留

位 22:16 **F0S**：接收 FIFO 0 大小 (Rx FIFO 0 Size)
0：无接收 FIFO 0
1-64：接收 FIFO 0 元素数
>64：大于 64 的值会被解析为 64
接收 FIFO 0 元素的索引为 0 到 F0S-1。

位 15:2 **F0SA**：接收 FIFO 0 起始地址 (Rx FIFO 0 Start Address)
消息 RAM 中接收 FIFO 0 的起始地址（32 位字地址，请参见*图 729：消息 RAM 配置*）。

位 1:0 保留

56.4.27 FDCAN 接收 FIFO 0 状态寄存器 (FDCAN_RXF0S)

FDCAN Rx FIFO 0 Status Register

地址：0x00A4

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	RF0L	F0F	Res.	Res.	F0PI[5:0]					
						rw	rw			rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	F0GI[5:0]						Res.	F0FL[6:0]						
		rw	rw	rw	rw	rw	rw		rw	rw	rw	rw	rw	rw	rw

位 31:26 保留

位 25 **RF0L**：接收 FIFO 0 消息丢失 (Rx FIFO 0 Message Lost)
此位是中断标志 IR[RF0L] 的副本。当 IR[RF0L] 复位时，此位也会复位。
0：接收 FIFO 0 消息未丢失
1：接收 FIFO 0 消息已丢失，尝试向大小为零的接收 FIFO 0 进行写入后也会置 1

位 24 **F0F**：接收 FIFO 0 已满 (Rx FIFO 0 Full)
0：接收 FIFO 0 未满
1：接收 FIFO 0 已满

位 23:22 保留

位 21:16 **F0PI**：接收 FIFO 0 放入索引 (Rx FIFO 0 Put Index)
接收 FIFO 0 写入索引指针，范围为 0 到 63。

位 15:14 保留

位 13:8 **F0GI**：接收 FIFO 0 获取索引 (Rx FIFO 0 Get Index)
接收 FIFO 0 读取索引指针，范围为 0 到 63。

位 7 保留

位 6:0 **F0FL**：接收 FIFO 填充级别 (Rx FIFO 0 Fill Level)
接收 FIFO 0 中存储的元素数，范围为 0 到 64。

56.4.28 FDCAN 接收 FIFO 0 确认寄存器 (FDCAN_RXF0A)

FDCAN Rx FIFO 0 Acknowledge Register

地址：0x00A8

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	F0AI[5:0]					
										rw	rw	rw	rw	rw	rw

位 31:6 保留

位 5:0 **F0AI**：接收 FIFO 0 确认索引 (Rx FIFO 0 Acknowledge Index)
主机从接收 FIFO 0 读取消息或消息序列后，必须将从接收 FIFO 0 读取的最后一个元素的缓冲区索引写入 F0AI 中。此操作会将接收 FIFO 0 获取索引 RXF0S[F0GI] 设为 F0AI + 1，并会更新 FIFO 0 填充级别 RXF0S[F0FL]。

56.4.29 FDCAN 接收缓冲区配置寄存器 (FDCAN_RXBC)

FDCAN Rx Buffer Configuration Register

地址：0x00AC

复位值：0x0000 0000

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

位 31:16 保留

位 15:2 **RBSA**：接收缓冲区起始地址 (Rx Buffer Start Address)

配置消息 RAM 中接收缓冲区部分的起始地址（32 位字地址）。此外，也用于引用调试消息 A、B、C。

这些位受写保护 (P)，也就是说，仅当 CCCR 寄存器的位 1 [CCE] 和位 0 [INIT] 置“1”时，才能通过这些位进行写访问。

位 1:0 保留

56.4.30 FDCAN 接收 FIFO 1 配置寄存器 (FDCAN_RXF1C)

FDCAN Rx FIFO 1 Configuration Register

地址：0x00B0

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
F1OM	F1WM[7:0]							Res.	F1S[7:0]						
	rw	rw	rw	rw	rw	rw	rw		rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
F1SA[13:0]														Res.	Res.
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw		

位 31 **F1OM**：FIFO 1 工作模式 (FIFO 1 Operation mode)

FIFO 1 可工作在阻止模式或覆盖模式下。

0：FIFO 1 处于阻止模式

1：FIFO 1 处于覆盖模式

位 30:24 **F1WM**：FIFO 1 水印 (FIFO 1 Watermark)

0：禁止水印中断

1-64：接收 FIFO 1 水印中断 (IR[RF1W]) 的级别

>64：禁止水印中断

这些位受写保护 (P)，也就是说，仅当 CCCR 寄存器的位 1 [CCE] 和位 0 [INIT] 置“1”时，才能通过这些位进行写访问。

位 23 保留

位 22:16 **F1S**：接收 FIFO 1 大小 (Rx FIFO 1 Size)
0：无接收 FIFO 1
1-64：接收 FIFO 1 元素数
>64：大于 64 的值会被解析为 64
接收 FIFO 1 元素的索引为 0 到 F1S - 1。
这些位受写保护 (P)，也就是说，仅当 CCCR 寄存器的位 1 [CCE] 和位 0 [INIT] 置“1”时，才能通过这些位进行写访问。

位 15:2 **F1SA**：接收 FIFO 1 起始地址 (Rx FIFO 1 Start Address)
消息 RAM 中接收 FIFO 1 的起始地址（32 位字地址，请参见*图 729：消息 RAM 配置*）。
这些位受写保护 (P)，也就是说，仅当 CCCR 寄存器的位 1 [CCE] 和位 0 [INIT] 置“1”时，才能通过这些位进行写访问。

位 1:0 保留

56.4.31 FDCAN 接收 FIFO 1 状态寄存器 (FDCAN_RXF1S)

FDCAN Rx FIFO 1 Status Register

地址：0x00B4

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DMS[1:0]		Res.	Res.	Res.	Res.	RF1L	F1F	Res.	Res.	F1PI[6:0]					
r	r					r	r			r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	F1GI[6:0]						Res.	F1FL[7:0]						
		r	r	r	r	r	r		r	r	r	r	r	r	r

位 31:30 **DMS**：调试消息状态 (Debug Message Status)
00：空闲状态，等待接收调试消息，DMA 请求清零
01：已接收调试消息 A
10：已接收调试消息 A、B
11：已接收调试消息 A、B、C，DMA 请求置 1

位 29:26 保留

位 25 **RF1L**：接收 FIFO 1 消息丢失 (Rx FIFO 1 Message Lost)
此位是中断标志 IR[RF1L] 的副本。当 IR[RF1L] 复位时，此位也会复位。
0：接收 FIFO 1 消息未丢失
1：接收 FIFO 1 消息已丢失，尝试向大小为零的接收 FIFO 1 进行写入后也会置 1。

位 24 **F1F**：接收 FIFO 1 已满 (Rx FIFO 1 Full)
0：接收 FIFO 1 未满
1：接收 FIFO 1 已满

位 23:22 保留

位 21:16 **F1PI**：接收 FIFO 1 放入索引 (Rx FIFO 1 Put Index)
接收 FIFO 1 写入索引指针，范围为 0 到 63。

位 15:14 保留

位 13:8 **F1GI**：接收 FIFO 1 获取索引 (Rx FIFO 1 Get Index)
接收 FIFO 1 读取索引指针，范围为 0 到 63。

位 7 保留

位 6:0 **F1FL**：接收 FIFO 1 填充级别 (Rx FIFO 1 Fill Level)
接收 FIFO 1 中存储的元素数，范围为 0 到 64

56.4.32 FDCAN 接收 FIFO 1 确认寄存器 (FDCAN_RXF1A)

FDCAN Rx FIFO 1 Acknowledge Register

地址：0x00B8

复位值：0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	F1AI[5:0]					
										rw	rw	rw	rw	rw	rw

位 31:6 保留

位 5:0 **F1AI**：接收 FIFO 1 确认索引 (Rx FIFO 1 Acknowledge Index)
主机从接收 FIFO 1 读取消息或消息序列后，必须将从接收 FIFO 1 读取的最后一个元素的缓冲区索引写入 F1AI 中。此操作会将接收 FIFO 1 获取索引 RXF1S[F1GI] 设为 F1AI + 1，并会更新 FIFO 1 填充级别 RXF1S[F1FL]。

56.4.33 FDCAN 接收缓冲区元素大小配置寄存器 (FDCAN_RXESC)

FDCAN Rx Buffer Element Size Configuration Register

配置属于接收缓冲区/接收 FIFO 元素的数据字节数。如果数据字段大小大于 8 个字节，则该数据字段仅可用于 CAN FD 操作。

地址：0x00BC

复位值：0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	RBDS[2:0]			Res.	F1DS[2:0]			Res.	F0DS[2:0]		
					r	r	r		r	r	r		r	r	r

位 31:11 保留

位 10:8 **RBDS**：接收缓冲区数据字段大小 (Rx Buffer Data Field Size)
000：8 字节数据字段
001：12 字节数据字段
010：16 字节数据字段
011：20 字节数据字段
100：24 字节数据字段
101：32 字节数据字段
110：48 字节数据字段
111：64 字节数据字段

位 7 保留

位 6:4 **F1DS**：接收 FIFO 0 数据字段大小 (Rx FIFO 0 Data Field Size)
000：8 字节数据字段
001：12 字节数据字段
010：16 字节数据字段
011：20 字节数据字段
100：24 字节数据字段
101：32 字节数据字段
110：48 字节数据字段
111：64 字节数据字段

位 3 保留

位 2:0 **F0DS**：接收 FIFO 1 数据字段大小 (Rx FIFO 1 Data Field Size)
000：8 字节数据字段
001：12 字节数据字段
010：16 字节数据字段
011：20 字节数据字段
100：24 字节数据字段
101：32 字节数据字段
110：48 字节数据字段
111：64 字节数据字段

56.4.34 FDCAN 发送缓冲区配置寄存器 (FDCAN_TXBC)

FDCAN Tx Buffer Configuration Register

地址：0x00C0

复位值：0x0000 0000

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

位 31 保留

位 30 **TFQM**：发送 FIFO/队列模式 (Tx FIFO/Queue Mode)。

0：发送 FIFO 操作

1：发送队列操作

这些位受写保护 (P)，也就是说，仅当 CCCR 寄存器的位 1 [CCE] 和位 0 [INIT] 置“1”时，才能通过这些位进行写访问。

位 29:24 **TFQS**：发送 FIFO/队列大小 (Transmit FIFO/Queue Size)。

0：无发送 FIFO/队列

1-32：用于发送 FIFO/队列的发送缓冲区数

>32：大于 32 的值会被解析为 32。

这些位受写保护 (P)，也就是说，仅当 CCCR 寄存器的位 1 [CCE] 和位 0 [INIT] 置“1”时，才能通过这些位进行写访问。

位 23:22 保留

位 21:16 **NDTB**：专用发送缓冲区数 (Number of Dedicated Transmit Buffers)。

0：无专用发送缓冲区

1-32：专用发送缓冲区数

>32：大于 32 的值会被解析为 32。

这些位受写保护 (P)，也就是说，仅当 CCCR 寄存器的位 1 [CCE] 和位 0 [INIT] 置“1”时，才能通过这些位进行写访问。

位 15:2 **TBSA**：发送缓冲区起始地址 (Tx Buffer Start Address)。

消息 RAM 中发送缓冲区部分的起始地址（32 位字地址，请参见*图 729*）。

这些位受写保护 (P)，也就是说，仅当 CCCR 寄存器的位 1 [CCE] 和位 0 [INIT] 置“1”时，才能通过这些位进行写访问。

位 1:0 保留

注： *TFQS 与 NDTB 之和不能大于 32。不会检查配置是否存在错误。消息 RAM 中的发送缓冲区部分从专用发送缓冲区开始。*

56.4.35 FDCAN 发送 FIFO/队列状态寄存器 (FDCAN_TXFQS)

FDCAN Tx FIFO/Queue Status Register

发送 FIFO/队列状态与寄存器 TXBRP 中列出的挂起发送请求相关。因此，添加/取消请求的作用可能因正在运行发送扫描而延迟（TXBRP 尚未更新）。

地址：0x00C4

复位值：0x0000 0000

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

位 31:22 保留

位 21 **TFQF**：发送 FIFO/队列已满 (Tx FIFO/Queue Full)
0：发送 FIFO/队列未满
1：发送 FIFO/队列已满

位 20:16 **TFQPI**：发送 FIFO/队列放入索引 (Tx FIFO/Queue Put Index)
发送 FIFO/队列写入索引指针，范围为 0 到 31

位 15:13 保留

位 12:8 **TFGI**：发送 FIFO 获取索引 (Tx FIFO Get Index)。
发送 FIFO 读取索引指针，范围为 0 到 31。如果已配置发送队列操作（TXBC.TFQM = “1”），则读为零

位 7:6 保留

位 5:0 **TFFL**：发送 FIFO 空闲级别 (Tx FIFO Free Level)
从 TFGI 开始的连续空闲发送 FIFO 元素数，范围为 0 到 32。如果已配置发送队列操作（TXBC[TFQM] = “1”），则读为零。

注： *如果是专用发送缓冲区与发送 FIFO 或发送队列相结合的混合配置，放入和获取索引指示从第一个专用发送缓冲区开始的发送缓冲区数。例如：对于 12 个专用发送缓冲区与包含 20 个缓冲区的发送 FIFO 相结合的混合配置，放入索引 15 会指向发送 FIFO 的第四个缓冲区。*

56.4.36 FDCAN 发送缓冲区元素大小配置寄存器 (FDCAN_TXESC)

FDCAN Tx Buffer Element Size Configuration Register

配置属于发送缓冲区元素的数据字节数。如果数据字段大小大于 8 个字节，则该数据字段仅可用于 CAN FD 操作。

地址：0x00C8

复位值：0x0000 0000

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

位 31:3 保留

位 2:0 **TBDS**：发送缓冲区数据字段大小 (Tx Buffer Data Field Size)
000：8 字节数据字段
001：12 字节数据字段
010：16 字节数据字段
011：20 字节数据字段
100：24 字节数据字段
101：32 字节数据字段
110：48 字节数据字段
111：64 字节数据字段

56.4.37 FDCAN 发送缓冲区请求挂起寄存器 (FDCAN_TXBRP)

FDCAN Tx Buffer Request Pending Register

地址：0x00C8

复位值：0x0000 0000

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

位 31:0 **TRP**：发送请求挂起 (Transmission Request Pending)。

每个发送缓冲区都有自己的发送请求挂起位。这些位通过寄存器 TXBAR 置 1。请求的发送已完成或已通过 TXBCR 取消后，这些位会复位。

TXBRP 位仅会为通过 TXBC 配置的发送缓冲区置 1。TXBRP 位置 1 后，会启动发送扫描（请参见*过滤调试消息*）以检查优先级最高（消息 ID 最小的发送缓冲区）的挂起发送请求。

取消请求会使寄存器 TXBRP 的相应发送请求挂起位复位。如果请求取消时发送已开始进行，则无论发送是否成功，都会在发送结束时执行复位操作。相应的 TXBRP 位复位后，取消请求位也会立即复位。

请求取消后，会在下列情况下通过 TXBCF 指示完成取消

– 与相应的 TXBTO 位一起成功发送后

– 取消时发送尚未开始

– 发送因仲裁丢失而中止

– 帧发送过程中出错

在 DAR 模式下，所有发送在发送失败后都会自动取消。对于所有未成功的发送，相应的 TXBCF 位会置 1。

0：无发送请求挂起

1：发送请求挂起

注： *如果 TXBRP 位在正在进行发送扫描时置 1，则进行该特殊发送扫描期间不会考虑此位。如果请求取消此类发送缓冲区，则该添加请求会立即取消，相应的 TXBRP 位会复位。*

56.4.38 FDCAN 发送缓冲区添加请求寄存器 (FDCAN_TXBAR)

FDCAN Tx Buffer Add Request Register

地址：0x00D0

复位值：0x0000 0000

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

位 31:0 **AR**：添加请求 (Add Request)

每个发送缓冲区都有自己的添加请求位。写入“1”会将相应的添加请求位置 1；写入“0”没有影响。这样，主机通过对 TXBAR 进行一次写操作便可为多个发送缓冲区设置发送请求。仅会为通过 TXBC 配置的发送缓冲区设置 TXBAR 位。如果没有运行发送扫描，这些位会立即复位，否则在发送扫描过程完成之前，这些位保持置 1 状态。

0：未添加发送请求

1：已添加发送请求

注： *如果添加请求应用到具有挂起发送请求的发送缓冲区（相应的 TXBRP 位已置 1），则会忽略请求。*

56.4.39 FDCAN 发送缓冲区取消请求寄存器 (FDCAN_TXBCR)

FDCAN Tx Buffer Cancellation Request Register

地址：0x00D4

复位值：0x0000 0000

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

位 31:0 **CR**：取消请求 (Cancellation Request)

每个发送缓冲区都有自己的取消请求位。写入“1”会将相应的取消请求位置 1；写入“0”没有影响。

这样，主机通过对 TXBCR 进行一次写操作便可为多个发送缓冲区设置取消请求。仅会为通过 TXBC 配置的发送缓冲区设置 TXBCR 位。在相应的 TXBRP 位复位之前，这些位保持置 1 状态。

0：无取消挂起

1：取消挂起

56.4.40 FDCAN 发送缓冲区发送已发生寄存器 (FDCAN_TXBTO)

FDCAN Tx Buffer Transmission Occurred Register

地址：0x00D8

复位值：0x0000 0000

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

位 31:0 **TO**：发送已发生 (Transmission Occurred)。

每个发送缓冲区都有自己的发送已发生位。当相应的 TXBRP 位在发送成功后清零时，这些位会置 1。通过向寄存器 TXBAR 的相应位写入“1”请求进行新发送时，这些位会复位。

0：未进行发送

1：已进行发送

56.4.41 FDCAN 发送缓冲区取消完成寄存器 (FDCAN_TXBCF)

FDCAN Tx Buffer Cancellation Finished Register

地址：0x00DC

复位值：0x0000 0000

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

位 31:0 **CF**：取消完成 (Cancellation Finished)

每个发送缓冲区都有自己的取消完成位。当相应的 TXBRP 位在通过 TXBCR 请求取消后清零时，这些位会置 1。如果相应的 TXBRP 位在取消时未置 1，CF 会立即置 1。通过向寄存器 TXBAR 的相应位写入“1”请求进行新发送时，这些位会复位。

0：无发送缓冲区取消

1：发送缓冲区取消完成

56.4.42 FDCAN 发送缓冲区发送中断使能寄存器 (FDCAN_TXBTIE)

FDCAN Tx Buffer Transmission Interrupt Enable Register

地址：0x00E0

复位值：0x0000 0000

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

位 31:0 **TIE**：发送中断使能 (Transmission Interrupt Enable)
每个发送缓冲区都有自己的发送中断使能位。
0：禁止发送中断
1：使能发送中断

56.4.43 FDCAN 发送缓冲区取消完成中断使能寄存器 (FDCAN_TXBCIE)

FDCAN Tx Buffer Cancellation Finished Interrupt Enable Register

地址：0x00E4

复位值：0x0000 0000

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

位 31:0 **CF**：取消完成中断使能 (Cancellation Finished Interrupt Enable)。
每个发送缓冲区都有自己的取消完成中断使能位。
0：禁止取消完成中断
1：使能取消完成中断

56.4.44 FDCAN 发送事件 FIFO 配置寄存器 (FDCAN_TXEFC)

FDCAN Tx Event FIFO Configuration Register

地址：0x00F0

复位值：0x0000 0000

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

位 31:30 保留

位 29:24 **EFWM**：事件 FIFO 水印 (Event FIFO Watermark)

0：禁止水印中断

1-32：发送事件 FIFO 水印中断 (IR[TEFW]) 的级别

>32：禁止水印中断

这些位受写保护 (P)，也就是说，仅当 CCCR 寄存器的位 1 [CCE] 和位 0 [INIT] 置“1”时，才能通过这些位进行写访问。

位 23:22 保留

位 21:16 **EFS**：事件 FIFO 大小 (Event FIFO Size)。

0：禁止发送事件 FIFO

1-32：发送事件 FIFO 元素数

>32：大于 32 的值会被解析为 32

发送事件 FIFO 元素的索引为 0 到 EFS-1。

这些位受写保护 (P)，也就是说，仅当 CCCR 寄存器的位 1 [CCE] 和位 0 [INIT] 置“1”时，才能通过这些位进行写访问。

位 15:2 **EFSA**：事件 FIFO 起始地址 (Event FIFO Start Address)

消息 RAM 中发送事件 FIFO 的起始地址（32 位字地址，请参见*图 729：消息 RAM 配置*）。

这些位受写保护 (P)，也就是说，仅当 CCCR 寄存器的位 1 [CCE] 和位 0 [INIT] 置“1”时，才能通过这些位进行写访问。

位 1:0 保留

56.4.45 FDCAN 发送事件 FIFO 状态寄存器 (FDCAN_TXEFS)

FDCAN Tx Event FIFO Status Register

地址：0x00F4

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	TEFL	EFF	Res.	Res.	Res.	EFPI[4:0]				
						r	r				r	r	r	r	r

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

位 31:26 保留

位 25 **TEFL**：发送事件 FIFO 元素丢失 (Tx Event FIFO Element Lost)。
此位是中断标志 IR[TEFL] 的副本。当 IR[TEFL] 复位时，此位也会复位。
0：没有发送事件 FIFO 元素丢失
1：发送事件 FIFO 元素已丢失，尝试向大小为零的发送事件 FIFO 进行写入时也会置 1。

位 24 **EFF**：事件 FIFO 已满 (Event FIFO Full)。
0：发送事件 FIFO 未满
1：发送事件 FIFO 已满

位 23:21 保留

位 20:16 **EFPI**：事件 FIFO 放入索引 (Event FIFO Put Index)。
发送事件 FIFO 写入索引指针，范围为 0 到 31。

位 15:13 保留

位 12:8 **EFGI**：事件 FIFO 获取索引 (Event FIFO Get Index)。
发送事件 FIFO 读取索引指针，范围为 0 到 31。

位 7:6 保留

位 5:0 **EFFL**：事件 FIFO 填充级别 (Event FIFO Fill Level)。
发送事件 FIFO 中存储的元素数，范围为 0 到 31。

56.4.46 FDCAN 发送事件 FIFO 确认寄存器 (FDCAN_TXEFA)

FDCAN Tx Event FIFO Acknowledge Register

地址：0x00F8

复位值：0x0000 0000

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

位 31:5 保留

位 4:0 **EFAI**：事件 FIFO 确认索引 (Event FIFO Acknowledge Index)。
主机从发送事件 FIFO 读取元素或元素序列后，必须将从发送事件 FIFO 读取的最后一个元素的索引写入 EFAI 中。此操作会将发送事件 FIFO 获取索引 TXEFS[EFGI] 设为 EFAI + 1，并会更新 FIFO 0 填充级别 TXEFS[EFFL]。

56.4.47 FDCAN TT 触发存储器配置寄存器 (FDCAN_TTTMC)

FDCAN TT Trigger Memory Configuration Register

地址：0x100

复位值：0x0000 0000

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

位 31:23 保留

位 22:16 **TME**：触发存储器元素 (Trigger Memory Elements)。
0：无触发存储器
1-64：触发存储器的元素数
>64：大于 64 的值会被解析为 64
这些位受写保护 (P)，也就是说，仅当 CCCR 寄存器的位 1 [CCE] 和位 0 [INIT] 置“1”时，才能通过这些位进行写访问。

位 15:2 **TMSA**：触发存储器起始地址 (Trigger Memory Start Address)。
消息 RAM 中触发存储器的起始地址（32 位字地址，请参见*图 729：消息 RAM 配置*）。
这些位受写保护 (P)，也就是说，仅当 CCCR 寄存器的位 1 [CCE] 和位 0 [INIT] 置“1”时，才能通过这些位进行写访问。

位 1:0 保留

56.4.48 FDCAN TT 参考消息配置寄存器 (FDCAN_TTRMC)

FDCAN TT Reference Message Configuration Register

地址：0x0104

复位值：0x0000 0000

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

位 31 **RMPS**：参考消息有效负载选择 (Reference Message Payload Select)

时间被控节点会忽略此位。

0：参考消息没有额外的有效负载

1：会从发送缓冲区 0 获取以下元素：

消息标记 MM、

事件 FIFO 控制 EFC、

数据长度代码 DLC、

数据字节 DB（级别 1：字节 2-8，级别 0、2：字节 5-8）

这些位受写保护 (P)，也就是说，仅当 CCCR 寄存器的位 1 [CCE] 和位 0 [INIT] 置“1”时，才能通过这些位进行写访问。

位 30 **XTD**：扩展标识符 (Extended Identifier)

0：11 位标准标识符

1：29 位扩展标识符

这些位受写保护 (P)，也就是说，仅当 CCCR 寄存器的位 1 [CCE] 和位 0 [INIT] 置“1”时，才能通过这些位进行写访问。

位 29 保留

位 28:0 **RID**：参考标识符 (Reference Identifier)。

通过参考消息发送并用于参考消息过滤的标识符。标准参考标识符或扩展参考标识符取决于 XTD 位。标准标识符必须写入 ID[28:18]。

这些位受写保护 (P)，也就是说，仅当 CCCR 寄存器的位 1 [CCE] 和位 0 [INIT] 置“1”时，才能通过这些位进行写访问。

56.4.49 FDCAN TT 操作配置寄存器 (FDCAN_TTOCF)

FDCAN TT Operation Configuration Register

地址：0x0108

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	EVTP	ECC	EGTF	AWL[7:0]							
					rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EECS	IRTO[6:0]							LDSDL[2:0]			TM	GEN	Res.	OM[1:0]	
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw		rw	rw

位 31:27 保留

位 26 **EVTP**：事件触发极性 (Event Trigger Polarity)。

0：上升沿触发

1：下降沿触发

这些位受写保护 (P)，也就是说，仅当 CCCR 寄存器的位 1 [CCE] 和位 0 [INIT] 置“1”时，才能通过这些位进行写访问。

位 25 **ECC**：使能时钟校准 (Enable Clock Calibration)。

0：FDCAN 级别 0、2 中禁止自动时钟校准

1：FDCAN 级别 0、2 中使能自动时钟校准

这些位受写保护 (P)，也就是说，仅当 CCCR 寄存器的位 1 [CCE] 和位 0 [INIT] 置“1”时，才能通过这些位进行写访问。

位 24 **EGTF**：使能全局时间过滤 (Enable Global Time Filtering)。

0：FDCAN 级别 0、2 中禁止全局时间过滤

1：FDCAN 级别 0、2 中使能全局时间过滤

这些位受写保护 (P)，也就是说，仅当 CCCR 寄存器的位 1 [CCE] 和位 0 [INIT] 置“1”时，才能通过这些位进行写访问。

位 23:16 **AWL**：应用看门狗限值 (Application Watchdog Limit)。

可通过将 AWL 编程为 0x00 禁止应用看门狗。

0x00-FF：应用可延迟处理应用看门狗的最长时间。应用看门狗每隔 256 个 NTU 递增一次。

这些位受写保护 (P)，也就是说，仅当 CCCR 寄存器的位 1 [CCE] 和位 0 [INIT] 置“1”时，才能通过这些位进行写访问。

位 15 **EECS**：使能外部时钟同步 (Enable External Clock Synchronization)

若使能，TUR 配置（仅限 TURCF[NCL]）可在 FDCAN 操作期间更新。

0：FDCAN 级别 0、2 中禁止外部时钟同步

1：FDCAN 级别 0、2 中使能外部时钟同步

这些位受写保护 (P)，也就是说，仅当 CCCR 寄存器的位 1 [CCE] 和位 0 [INIT] 置“1”时，才能通过这些位进行写访问。

位 14:8 **IRTO**：初始参考触发偏移 (Initial Reference Trigger Offset)。

0x00-7F 正偏移，范围为 0 到 127

这些位受写保护 (P)，也就是说，仅当 CCCR 寄存器的位 1 [CCE] 和位 0 [INIT] 置“1”时，才能通过这些位进行写访问。

位 7:5 **LDSDL**：同步偏差限值的 LD (LD of Synchronization Deviation Limit)。
同步偏差限值 SDL 是通过其双对数 LDSDL 配置的，公式为 SDL= 2*(LDSDL + 5)。SDL 在 32 和 4096 之间。不应超过 CAN 位定时配置提供的时钟容差。
这些位受写保护 (P)，也就是说，仅当 CCCR 寄存器的位 1 [CCE] 和位 0 [INIT] 置“1”时，才能通过这些位进行写访问。

位 4 **TM**：时间主控节点 (Time Master)。
0：禁止时间主控节点功能
1：潜在时间主控节点
这些位受写保护 (P)，也就是说，仅当 CCCR 寄存器的位 1 [CCE] 和位 0 [INIT] 置“1”时，才能通过这些位进行写访问。

位 3 **GEN**：间隙使能 (Gap Enable)。
0：严格的时间触发操作
1：外部事件同步时间触发操作
这些位受写保护 (P)，也就是说，仅当 CCCR 寄存器的位 1 [CCE] 和位 0 [INIT] 置“1”时，才能通过这些位进行写访问。

位 2 保留

位 1:0 **OM**：工作模式 (Operation Mode)。
00：事件驱动 CAN 通信（默认）
01：TTCAN 1 级
10：TTCAN 2 级
11：TTCAN 0 级
这些位受写保护 (P)，也就是说，仅当 CCCR 寄存器的位 1 [CCE] 和位 0 [INIT] 置“1”时，才能通过这些位进行写访问。

56.4.50 FDCAN TT 矩阵限值寄存器 (FDCAN_TTMLM)

FDCAN TT Matrix Limits Register

地址：0x010C

复位值：0x0000 0000

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

位 31:28 保留

位 27:16 **ENTT**：预期发送触发数 (Expected Number of Tx Triggers)
0x000–FFF：一个矩阵周期中的预期发送触发数。
这些位受写保护 (P)，也就是说，仅当 CCCR 寄存器的位 1 [CCE] 和位 0 [INIT] 置“1”时，才能通过这些位进行写访问。

位 15:12 保留

位 11:8 **TXEW**：发送使能窗口 (Tx Enable Window)
0x0-F：发送使能窗口的长度，1-16 个 NTU 周期
这些位受写保护 (P)，也就是说，仅当 CCCR 寄存器的位 1 [CCE] 和位 0 [INIT] 置“1”时，才能通过这些位进行写访问。

位 7:6 **CSS**：周期开始同步 (Cycle Start Synchronization)
使能同步脉冲输出。
00：无同步脉冲
01：基本周期开始时同步脉冲
10：矩阵周期开始时同步脉冲
11：保留
这些位受写保护 (P)，也就是说，仅当 CCCR 寄存器的位 1 [CCE] 和位 0 [INIT] 置“1”时，才能通过这些位进行写访问。

位 5:0 **CCM**：周期计数最大值 (Cycle Count Max)
0x00：每个矩阵周期 1 个基本周期
0x01：每个矩阵周期 2 个基本周期
0x03：每个矩阵周期 4 个基本周期
0x07：每个矩阵周期 8 个基本周期
0x0F：每个矩阵周期 16 个基本周期
0x1F：每个矩阵周期 32 个基本周期
0x3F：每个矩阵周期 64 个基本周期
其他值：保留
这些位受写保护 (P)，也就是说，仅当 CCCR 寄存器的位 1 [CCE] 和位 0 [INIT] 置“1”时，才能通过这些位进行写访问。

注： *ISO 11898-4 第 5.2.1 部分要求仅配置上面列出的周期计数值。配置其他值也可以，但可能导致矩阵周期不一致。*

56.4.51 FDCAN TUR 配置寄存器 (FDCAN_TURCF)

FDCAN TUR Configuration Register

NTU 长度的计算公式为：NTU = CAN 时钟周期 x NC/DC。

NC 为 18 位值。其高位部分 NCH[17:16] 通过硬接线的方式写入 0b01。因此，NC 的范围从 0x10000 扩展为 0x1FFFF。由 NCL 配置的值是 TURNA[NAV[15:0]] 的初始值。DC 通过硬件复位设为 0x1000，不得写入为 0x0000。

- 1 级：NC 4 × DC 和 NTU = CAN 位时间
- 0 级和 2 级：NC 8 × DC

TUR 的实际值可通过 TTCAN 0 级和 2 级的时钟偏移补偿功能进行更改，以便将 NTU 的节点本地视图调整为 NTU 的时间主控节点视图。DC 不会被自动偏移补偿更改，TURNA[NAV] 可在由 TTOCF[LDSDL] 提供的同步偏差限值范围内在 NC 周围调整。NC 和 DC 应编程为最大的适用值，以便使偏移补偿过程达到最佳计算精度。

地址：0x0110

复位值：0x1000 0000

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

位 31 **ELT**：使能本地时间 (Enable Local Time)。

0：停止本地时间（默认）

1：使能本地时间

这些位受写保护 (P)，也就是说，仅当 CCCR 寄存器的位 1 [CCE] 和位 0 [INIT] 置“1”时，才能通过这些位进行写访问。

注： 本地时间通过将 ELT 置 1 来启动，之后会一直保持激活状态，直到 ELT 复位或下一次硬件复位为止。当 TURCF[ELT] = “1”时，会锁定 TURCF[DC]。如果向 ELT 写入“0”，则在新值同步到 CAN 时钟域之前，可读值始终为“1”。在此期间，对寄存器其他位的写访问仍保持锁定。

位 30 保留

位 29:16 **DC**：分母配置 (Denominator Configuration)。

0x0000：非法值

0x0001 到 3FFF：分母配置

这些位受写保护 (P)，也就是说，仅当 CCCR 寄存器的位 1 [CCE] 和位 0 [INIT] 置“1”时，才能通过这些位进行写访问。

位 15:0 **NCL**：分子配置下限 (Numerator Configuration Low)。

仅当配置为 TURCF[ELT] = “0”、或者 TTOCF[EECS]（外部时钟同步使能）置 1 时，才能对 TUR 分子配置下限进行写访问。如果在 TT 配置模式以外写入 NCL 的新值，新值会在 TTOST.WECS 清零后生效。NCL 锁定，TTOST[WECS] 为“1”。

0x0000-FFFF 分子配置下限

这些位受写保护 (P)，也就是说，仅当 CCCR 寄存器的位 1 [CCE] 和位 0 [INIT] 置“1”时，才能通过这些位进行写访问。

注： 如果 TTCAN 1 级中的 NC < 7 × DC，则要求触发存储器中的后续时间标记之间的差值必须至少为两个 NTU。

56.4.52 FDCAN TT 操作控制寄存器 (FDCAN_TTOCN)

FDCAN TT Operation Control Register

地址：0x0114

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LCKC	Res.	ESCN	NIG	TMG	FGP	GCS	TTIE	TMC[1:0]		RTIE	SWS[1:0]		SWP	ECS	SGT
r		rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:16 保留

位 15 **LCKC**：TT 操作控制寄存器锁定 (TT Operation Control Register Locked)。
对寄存器 TTOCN 进行写访问会将此位置 1。当更新的配置已同步到 CAN 时钟域时，此位会复位。
0：使能对 TTOCN 的写访问
1：锁定对 TTOCN 的写访问

位 14 保留

位 13 **ESCN**：外部同步控制 (External Synchronization Control)
若使能此位，FDCAN 会将其周期时间相位同步到由事件触发引脚的上升沿所指示的外部事件（第 0.1.17 节，同步到外部时间调度）。
0：禁止外部同步
1：使能外部同步

位 12 **NIG**：下一条是间隙 (Next is Gap)。
仅当 FDCAN 为实际时间主控节点，并且配置为进行外部事件同步时间触发操作时（TTOCF[GEN] =“1”），此位才能置 1。
0：无操作，接收任何参考消息时复位
1：Next_is_Gap =“1”时发送下一条参考消息

位 11 **TMG**：时间标记间隙 (Time Mark Gap)。
0：通过每条参考消息复位
1：寄存器时间标记中断 TTIR[RTMI] 激活时，开始发送下一条参考消息

位 10 **FGP**：结束间隙 (Finish Gap)。
由 CPU 置 1，通过每条参考消息复位
0：未请求任何参考消息
1：应用请求开始发送参考消息

位 9 **GCS**：间隙控制选择 (Gap Control Select)
0：间隙控制与事件触发无关
1：通过输入事件触发引脚进行间隙控制

位 8 **TTIE**：触发时间标记中断脉冲使能 (Trigger Time Mark Interrupt Pulse Enable)
外部时间标记事件是通过触发存储器元素 TMEX 配置的。当触发存储器元素生效且 FDCAN 处于同步状态 In_Schedule 或 In_Gap 时，会生成触发时间标记中断脉冲。
0：禁止触发时间标记中断输出 m_ttcan_tmp
1：使能触发时间标记中断输出 m_ttcan_tmp

位 7:6 **TMC**：寄存器时间标记比较 (Register Time Mark Compare)。
00：未生成寄存器时间标记中断
01：如果时间标记 = 周期时间，则会生成寄存器时间标记中断
10：如果时间标记 = 本地时间，则会生成寄存器时间标记中断
11：如果时间标记 = 全局时间，则会生成寄存器时间标记中断
注：　更改时间标记参考（周期时间、本地时间、全局时间）时，建议先写入 TMC =“00”，然后重新配置 TTTMK，最后将 TMC 设为所需时间参考。

位 5 **RTIE**：寄存器时间标记中断脉冲使能 (Register Time Mark Interrupt Pulse Enable)。
寄存器时间标记中断是通过寄存器 TTTMK 配置的。当 TTOCN[TMC] 引用的时间（周期时间、本地时间或全局时间）等于 TTTMK[TM] 时，会生成长度为一个 m_ttcan_clk 周期的寄存器时间标记中断脉冲，与同步状态无关。
0：禁止寄存器时间标记中断输出
1：使能寄存器时间标记中断输出

位 4:3 **SWS**：停止监视源 (Stop Watch Source)。
00：禁止停止监视
01：周期时间的实际值会复制到 TTCPT[SWV]
10：本地时间的实际值会复制到 TTCPT[SWV]
11：全局时间的实际值会复制到 TTCPT[SWV]

位 2 **SWP**：停止监视极性 (Stop Watch Polarity)。
0：上升沿触发
1：下降沿触发

位 1 **ECS**：外部时钟同步 (External Clock Synchronization)。
如果节点为实际时间主控节点，则向 ECS 写入“1”会将 TTOST[WECS] 置 1。一个 APB 时钟周期后，ECS 会复位。外部时钟同步会在下一基本周期开始时生效。

位 0 **SGT**：设置全局时间 (Set Global time)。
如果节点为实际时间主控节点，则向 SGT 写入“1”会将 TTOST[WGDT] 置 1。一个 APB 时钟周期后，SGT 会复位。当节点发送的下一条参考消息的 Master_Ref_Mark 被写入 TTGTP 的预设值修改时，全局时间预设会生效。

56.4.53 FDCAN TT 全局时间预设寄存器 (CAN_TTGTP)

FDCAN TT Global Time Preset Register

如果 TTOST.WGDT 置 1，则发送的下一条参考消息的 Master_Ref_Mark 会被预设值修改，并且 Disc_Bit = “1”，从而会同时预设所有节点中的全局时间。

每次 Disc_Bit = “1”的参考消息生效、或者节点不是当前时间主控节点时，TP 会复位为 0x0000。TTOCN[SGT] 置 1 后，如果 TTOST[WGTD] = “1”，则 TP 会锁定，直至 Disc_Bit = “1”的参考消息生效或节点不再是当前时间主控节点。

地址：0x0118

复位值：0x0000 0000

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

位 31:16 **CTP**：周期时间目标相位 (Cycle Time Target Phase)。
CTP 受写保护，而 TTOCN[ESCN] 或 TTOST[SPL] 置 1（请参见*第 56.3.15 节：与外部时间调度同步*）。
0x0000-FFFF：定义预期事件触发上升沿时周期时间的目标值。

位 15:0 **NCL**：时间预设值 (Time Preset)。
TP 受写保护，而 TTOST[WGTD] 置 1。
0x0000-7FFF：下一主控节点参考标记 = 主控节点参考标记 + TP
0x8000 保留
0x8001-FFFF：下一主控节点参考标记 = 主控节点参考标记 - (0x10000 - TP)。

56.4.54 FDCAN TT 时间标记寄存器 (FDCAN_TTTMK)

FDCAN TT Time Mark Register

当 TTOCN[TMC] 指示的时基（周期时间、本地时间或全局时间）的值与 TM 相同时，会生成时间标记中断（TTIR[TMI] =“1”）。

地址：0x011C

复位值：0x0000 0000

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

位 31 **LCKM**：TT 时间标记寄存器锁定 (TT Time Mark Register Locked)。
始终通过对寄存器 TTOCN 进行写访问来置 1。当 TTOCN[TMC] 为“00”时，向寄存器 TTTMK 进行写访问会将此位置 1。寄存器已同步到 CAN 时钟域后，此位会复位。
0：使能对 TTTMK 的写访问
1：锁定对 TTTMK 的写访问

位 30:23 保留

位 22:16 **TICC**：时间标记周期代码 (Time Mark Cycle Code)。
时间标记有效的周期计数。
0b000000x 对所有周期有效
周期计数 mod2 = c 时，每 2 个周期，0b000001c 有效
周期计数 mod4 = cc 时，每 4 个周期，0b00001cc 有效
周期计数 mod8 = ccc 时，每 8 个周期，0b0001ccc 有效
周期计数 mod16 = cccc 时，每 16 个周期，0b001cccc 有效
周期计数 mod32 = ccccc 时，每 32 个周期，0b01ccccc 有效
周期计数 mod64 = cccccc 时，每 64 个周期，0b1cccccc 有效

位 15:0 **TM**：时间标记 (Time Mark)。
0x0000-FFFF：时间标记

注： *对寄存器 TTTMK 进行字节访问时，建议先禁止时间标记比较功能 （TTOCN[TMC] = “00”），以免对不一致的寄存器值进行比较。*

56.4.55 FDCAN TT 中断寄存器 (FDCAN_TTIR)

FDCAN TT Interrupt Register

如果检测到其中一个列出的条件（边沿有效），标志会置 1。在主机将标志清零之前，标志保持置 1 状态。通过向对应位位置写入“1”将标志清零。写入“0”则不会带来任何影响。硬复位会将寄存器清零。

地址：0x0120

复位值：0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IWT	ELC	SE2	SE1	TXO	TXU	GTE	GTD	GTW	SWE	TTMI	RTMI	SOG	CSM	SMC	SBC
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:19 保留

位 18 **CER**：配置错误 (Configuration Error)。
触发次序颠倒。
0：触发列表中未发现错误
1：触发列表中发现错误

位 17 **AW**：应用看门狗 (Application Watchdog)。
0：及时处理应用看门狗
1：未及时处理应用看门狗

位 16 **WT**：监视触发 (Watch Trigger)。
0：未丢失参考消息
1：丢失参考消息（0 级：周期时间 0xFF00）

位 15 **IWTG**：初始化监视触发 (Initialization Watch Trigger)。
通过复位 IWT 重新开始初始化。
0：系统启动期间未丢失参考消息
1：由于参考消息丢失，系统未启动

位 14 **ELC**：错误级别更改 (Error Level Changed)。
如果初始化过程中错误级别发生了变化，此位不会置 1。
0：错误级别未更改
1：错误级别已更改

位 13 **SE2**：调度错误 2 (Scheduling Error 2)。
0：无调度错误 2
1：发生调度错误 2

位 12 **SE1**：调度错误 1 (Scheduling Error 1)。
0：无调度错误 1
1：发生调度错误 1

位 11 **TXO**：发送计数上溢 (Tx Count Overflow)。
0：发送触发数与预期相同
1：一个周期中的发送触发数比预期多

位 10 **TXU**：发送计数下溢 (Tx Count Underflow)。
0：发送触发数与预期相同
1：一个周期中的发送触发数比预期少

位 9 **GTE**：全局时间错误 (Global Time Error)。
同步偏差 SD 超过 TTOCF[LDSDL] 指定的限值，仅限 TTCAN 0 级、2 级。
0：同步偏差在限值范围内
1：同步偏差超出限值范围

位 8 **GTD**：全局时间不连续 (Global Time Discontinuity)。
0：全局时间没有不连续的情况
1：全局时间不连续

位 7 **GTW**：全局时间回卷 (Global Time Wrap)
0：未发生全局时间回卷
1：发生了从 0xFFFF 到 0x0000 的全局时间回卷

位 6 **SWE**：停止监视事件 (Stop Watch Event)
0：停止监视触发引脚上未检测到上升沿/下降沿
1：停止监视触发引脚上检测到上升沿/下降沿

位 5 **TTMI**：内部触发时间标记事件 (Trigger Time Mark Event Internal)
内部时间标记事件是通过触发存储器元素 TMIN 配置的（请参见*第 56.3.21 节：FDCAN 触发存储器元素*）。当触发存储器元素生效、且 FDCAN 处于同步状态 In_Gap 或 In_Schedule 时，此位会置 1。
0：未达到时间标记
1：已达到时间标记（0 级：时间标记 TTOCF[RTO] x 0x200）

位 4 **RTMI**：寄存器时间标记中断 (Register Time Mark Interrupt)。
当 TTOCN[TMC] 引用的时间（周期时间、本地时间或全局时间）等于 TTTMK[TM] 时，此位会置 1，与同步状态无关。
0：未达到时间标记
1：已达到时间标记

位 3 **SOG**：间隙开始 (Start Of Gap)
0：未发现 Next_is_Gap 位置 1 的参考消息
1：Next_is_Gap 位置 1 的参考消息生效

位 2 **CSM**：同步模式更改 (Change of Synchronization Mode)。
0：主控节点与被控节点的关系或调度同步没有变化
1：主控节点与被控节点的关系或调度同步发生了变化，
TTOST[SPL] 复位时也会置 1

位 1 **SMC**：矩阵周期开始 (Start of Matrix Cycle)。
0：自位复位后未开始任何矩阵周期
1：矩阵周期已开始

位 0 **SBC**：基本周期开始 (Start of Basic Cycle)。
0：自位复位后未开始任何基本周期
1：基本周期已开始

56.4.56 FDCAN TT 中断使能寄存器 (FDCAN_TTIE)

FDCAN TT Interrupt Enable Register

TT 中断使能寄存器中的设置决定了 TT 中断寄存器中的哪些状态更改会产生中断。

地址：0x0124

复位值：0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IWTE	ELCE	SE2E	SE1E	TXOE	TXUE	GTEE	GTDE	GTWE	SWEE	TTMIE	RTMIE	SOGE	CSME	SMCE	SBCE
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:19 保留

位 18 **CERE**：配置错误中断使能 (Configuration Error Interrupt Enable)
0：禁止 TT 中断
1：使能 TT 中断

位 17 **AWE**：应用看门狗中断使能 (Application Watchdog Interrupt Enable)
0：禁止 TT 中断
1：使能 TT 中断

位 16 **WTE**：监视触发中断使能 (Watch Trigger Interrupt Enable)
0：禁止 TT 中断
1：使能 TT 中断

位 15 **IWTGE**：初始化监视触发中断使能 (Initialization Watch Trigger Interrupt Enable)
0：禁止 TT 中断
1：使能 TT 中断

位 14 **ELCE**：更改错误级别中断使能 (Change Error Level Interrupt Enable)
0：禁止 TT 中断
1：使能 TT 中断

位 13 **SE2E**：调度错误 2 中断使能 (Scheduling Error 2 Interrupt Enable)
0：禁止 TT 中断
1：使能 TT 中断

位 12 **SE1E**：调度错误 1 中断使能 (Scheduling Error 1 Interrupt Enable)
0：禁止 TT 中断
1：使能 TT 中断

位 11 **TXOE**：发送计数上溢中断使能 (Tx Count Overflow Interrupt Enable)
0：禁止 TT 中断
1：使能 TT 中断

位 10 **TXUE**：发送计数下溢中断使能 (Tx Count Underflow Interrupt Enable)
0：禁止 TT 中断
1：使能 TT 中断

位 9 **GTEE**：全局时间错误中断使能 (Global Time Error Interrupt Enable)
0：禁止 TT 中断
1：使能 TT 中断

位 8 **GTDE**：全局时间不连续中断使能 (Global Time Discontinuity Interrupt Enable)
0：禁止 TT 中断
1：使能 TT 中断

位 7 **GTWE**：全局时间回卷中断使能 (Global Time Wrap Interrupt Enable)
0：禁止 TT 中断
1：使能 TT 中断

位 6 **SWEE**：停止监视事件中断使能 (Stop Watch Event Interrupt Enable)
0：禁止 TT 中断
1：使能 TT 中断

位 5 **TTMIE**：内部触发时间标记事件中断使能 (Trigger Time Mark Event Internal Interrupt Enable)
0：禁止 TT 中断
1：使能 TT 中断

位 4 **RTMIE**：寄存器时间标记中断使能 (Register Time Mark Interrupt Enable)
0：禁止 TT 中断
1：使能 TT 中断

位 3 **SOGE**：间隙开始中断使能 (Start of Gap Interrupt Enable)
0：禁止 TT 中断
1：使能 TT 中断

位 2 **CSME**：同步模式更改中断使能 (Change of Synchronization Mode Interrupt Enable)
0：禁止 TT 中断
1：使能 TT 中断

位 1 **SMCE**：矩阵周期开始中断使能 (Start of Matrix Cycle Interrupt Enable)
0：禁止 TT 中断
1：使能 TT 中断

位 0 **SBCE**：基本周期开始中断使能 (Start of Basic Cycle Interrupt Enable)
0：禁止 TT 中断
1：使能 TT 中断

56.4.57 FDCAN TT 中断线选择寄存器 (FDCAN_TTILS)

FDCAN TT Interrupt Line Select Register

TT 中断线选择寄存器将特定中断标志生成的中断从 TT 中断寄存器分配到两个模块中断线之一。要生成中断，必须通过 ILE[EINT0] 和 ILE[EINT1] 使能相应的中断线。

地址：0x0128

复位值：0x0000 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IWTL	ELCL	SE2L	SE1L	TXOL	TXUL	GTEL	GTDL	GTWL	SWEL	TTMIL	RTMIL	SOGL	CSML	SMCL	SBCL
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:19 保留

位 18 **CERL**：配置错误中断线 (Configuration Error Interrupt Line)
0：分配给中断线 0 的 TT 中断
1：分配给中断线 1 的 TT 中断

位 17 **AWL**：应用看门狗中断线 (Application Watchdog Interrupt Line)
0：分配给中断线 0 的 TT 中断
1：分配给中断线 1 的 TT 中断

位 16 **WTL**：监视触发中断线 (Watch Trigger Interrupt Line)
0：分配给中断线 0 的 TT 中断
1：分配给中断线 1 的 TT 中断

位 15 **IWTGL**：初始化监视触发中断线 (Initialization Watch Trigger Interrupt Line)
0：分配给中断线 0 的 TT 中断
1：分配给中断线 1 的 TT 中断

位 14 **ELCL**：更改错误级别中断线 (Change Error Level Interrupt Line)
0：分配给中断线 0 的 TT 中断
1：分配给中断线 1 的 TT 中断

位 13 **SE2L**：调度错误 2 中断线 (Scheduling Error 2 Interrupt Line)
0：分配给中断线 0 的 TT 中断
1：分配给中断线 1 的 TT 中断

位 12 **SE1L**：调度错误 1 中断线 (Scheduling Error 1 Interrupt Line)
0：分配给中断线 0 的 TT 中断
1：分配给中断线 1 的 TT 中断

位 11 **TXOL**：发送计数上溢中断线 (Tx Count Overflow Interrupt Line)
0：分配给中断线 0 的 TT 中断
1：分配给中断线 1 的 TT 中断

位 10 **TXUL**：发送计数下溢中断线 (Tx Count Underflow Interrupt Line)
0：分配给中断线 0 的 TT 中断
1：分配给中断线 1 的 TT 中断

位 9 **GTEL**：全局时间错误中断线 (Global Time Error Interrupt Line)
0：分配给中断线 0 的 TT 中断
1：分配给中断线 1 的 TT 中断

位 8 **GTDL**：全局时间不连续中断线 (Global Time Discontinuity Interrupt Line)
0：分配给中断线 0 的 TT 中断
1：分配给中断线 1 的 TT 中断

位 7 **GTWL**：全局时间回卷中断线 (Global Time Wrap Interrupt Line)
0：分配给中断线 0 的 TT 中断
1：分配给中断线 1 的 TT 中断

位 6 **SWEL**：停止监视事件中断线 (Stop Watch Event Interrupt Line)
0：分配给中断线 0 的 TT 中断
1：分配给中断线 1 的 TT 中断

位 5 **TTMIL**：内部触发时间标记事件中断线 (Trigger Time Mark Event Internal Interrupt Line)
0：分配给中断线 0 的 TT 中断
1：分配给中断线 1 的 TT 中断

位 4 **RTMIL**：寄存器时间标记中断线 (Register Time Mark Interrupt Line)
0：分配给中断线 0 的 TT 中断
1：分配给中断线 1 的 TT 中断

位 3 **SOGL**：间隙开始中断线 (Start of Gap Interrupt Line)
0：分配给中断线 0 的 TT 中断
1：分配给中断线 1 的 TT 中断

位 2 **CSML**：同步模式更改中断线 (Change of Synchronization Mode Interrupt Line)
0：分配给中断线 0 的 TT 中断
1：分配给中断线 1 的 TT 中断

位 1 **SMCL**：矩阵周期开始中断线 (Start of Matrix Cycle Interrupt Line)
0：分配给中断线 0 的 TT 中断
1：分配给中断线 1 的 TT 中断

位 0 **SBCL**：基本周期开始中断线 (Start of Basic Cycle Interrupt Line)
0：分配给中断线 0 的 TT 中断
1：分配给中断线 1 的 TT 中断

56.4.58 FDCAN TT 工作状态寄存器 (FDCAN_TTOST)

FDCAN TT Operation Status Register

地址：0x012C

复位值：0x0000 0080

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SPL	WECS	AWE	WFE	GSI	TMP[2:0]			GFI	WGTD	Res.	Res.	Res.	Res.	Res.	Res.
r	r	r	r	r	r	r	r	r	r						

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RTO[7:0]								QCS	QGTP	SYS[1:0]		MS[1:0]		EL[1:0]	
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31 **SPL**：调度相位锁定 (Schedule Phase Lock)。
仅当外部同步使能时（TTOCN[ESCN] = “1”），此位才有效。在这种情况下，此位会指示由 TTGTP[CTP] 配置的周期时间与事件触发引脚上升沿的周期时间之差小于还是等于 9 个 NTU（请参见*第 56.3.15 节：与外部时间调度同步*）。
0：相位超出范围
1：相位在范围内

位 30 **WECS**：等待外部时钟同步 (Wait for External Clock Synchronization)。
0：无外部时钟同步挂起
1：节点等待外部时钟同步生效。此位会在下一基本周期开始时复位。

位 29 **AWE**：应用看门狗事件 (Application Watchdog Event)。
应用看门狗通过读取寄存器 TTOST 进行处理。如果未及时处理看门狗，位 AWE 会置 1，所有 FDCAN 通信都会停止，FDCAN 会设为进入总线监控模式。
0：及时处理应用看门狗
1：未及时处理应用看门狗

位 28 **WFE**：等待事件 (Wait For Event)。
0：未发布间隙，由 Next_is_Gap = “0” 的参考消息复位
1：接收到 Next_is_Gap = “1” 的参考消息

位 27 **GSI**：间隙开始指示符 (Gap Started Indicator)。
0：调度中没有间隙，由每条参考消息复位，用于所有时间被控节点
1：基本周期开始后的间隙时间

位 26:24 **TMP**：时间主控节点优先级 (Time Master Priority)。
0x0-7：实际时间主控节点的优先级

位 23 **GFI**：间隙结束指示符 (Gap Finished Indicator)。
CPU 向 TTOCN[FGP] 进行写操作时置 1，或者在 TMG = “1” 时通过时间标记中断置 1，或者在 TTOCN[GCS] = “1” 时通过输入引脚置 1（事件触发）。不会由 Ref_Trigger_Gap 置 1，间隙由另一发送参考消息的节点结束时也不会置 1。
0：每条参考消息结束时复位
1：间隙由 FDCAN 结束

位 22 **WGTD**：等待全局时间不连续 (Wait for Global Time Discontinuity)。
0：没有全局时间预设挂起
1：节点等待全局时间预设生效 节点已发送 Disc_Bit = “1” 的参考消息或接收到参考消息后，此位会复位。

位 21:16 保留

位 15:8 参考触发偏移 (Reference Trigger Offset)。
参考触发偏移是有符号整型值，范围为 -127 (0x81) 到 127 (0x7F)。达到下限 -127 时，不会发出通知。如果 FDCAN 成为时间主控节点 （MS[1:0] = “11”），由于主机和 CAN 时钟域之间要进行同步，因此 RTO 的复位会延迟。对于时间被控节点，会读取由 TTOCF[IRTO] 配置的值。
0x00-FF：实际参考触发偏移值

位 7 **QCS**：时钟速度质量 (Quality of Clock Speed)。
仅在 TTCAN 0 级和 2 级中有意义，否则会固定为 “1”。
0：本地时钟速度未同步到时间主控节点时钟速度
1：同步偏差 ≤ SDL

位 6 **GTP**：全局时间相位质量 (Quality of Global Time Phase)。
仅在 TTCAN 0 级和 2 级中有意义，否则会固定为“0”。
0：全局时间无效
1：全局时间与时间主控节点同相

位 5:4 **SYS**：同步状态 (Synchronization State)。
00：不同步
01：同步到 FDCAN 通信
10：调度被间隙挂起 (In_Gap)
11：同步到调度 (In_Schedule)

位 3:2 **MS**：主控节点状态 (Master State)。
00：Master_Off，无任何相关主控节点属性
01：作为时间被控节点工作
10：作为备份时间主控节点工作
11：作为当前时间主控节点工作

位 1:0 **EL**：错误级别 (Error Level)。
00：严重程度 0 - 无错误
01：严重程度 1 - 警告
10：严重程度 2 - 错误
11：严重程度 3 - 严重错误

56.4.59 FDCAN TUR 分子实际寄存器 (FDCAN_TURNA)

FDCAN TUR Numerator Actual Register

TTCAN 1 级中没有偏移补偿 (NAV = NC)。在 TTCAN 0 级和 2 级中，会计算节点本地时钟与时间主控节点本地时钟之间的偏移。当同步偏差（NC 与计算出的 NAV 之差）大于 2*(TTOCF[LDSDL] + 5) 时，会对偏移进行补偿。如果 TTOCF[LDSDL] 为 7，会得出 NAV 的最大范围 (NC - 0x1000) ≤ NAV ≤ (NC + 0x1000)。

地址：0x0130

复位值：0x0001 0000

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NAV[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:18 保留

位 17:0 **NAV**：分子实际值 (Numerator Actual Value)。
0x0EFFF：非法值
0x0F000-20FFF：实际分子值
0x21000：非法值

56.4.60 FDCAN TT 本地和全局时间寄存器 (FDCAN_TTLGT)

FDCAN TT Local and Global Time Register

地址：0x0134

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
GT[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LT[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:16 **GT**：全局时间 (Global Time)。
节点本地时间与其本地偏移之和的非小数部分（请参见*第 56.3.10 节：本地时间、周期时间、全局时间和外部时钟同步*）。
0x0000-FFFF：FDCAN 网络的全局时间值

位 15:0 **LT**：本地时间 (Local Time)。
本地时间的非小数部分，每隔一个本地 NTU 递增一次（请参见*第 56.3.10 节：本地时间、周期时间、全局时间和外部时钟同步*）。
0x0000-FFFF：FDCAN 节点的本地时间值

56.4.61 FDCAN TT 周期时间和计数寄存器 (FDCAN_TTCTC)

FDCAN TT Cycle Time and Count Register

地址：0x0138

复位值：0x003F 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	GT[5:0]					
										r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CT[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:22 保留

位 21:16 **CC**：周期计数 (Cycle Count)。
0x00-3F：系统矩阵中的实际基本周期计数

位 15:0 **CT**：周期时间 (Cycle Time)
节点本地时间与 Ref_Mark 之差的非小数部分（请参见*第 56.3.10 节：本地时间、周期时间、全局时间和外部时钟同步*）。
0x0000–FFFF：FDCAN 基本周期的周期时间值

56.4.62 FDCAN TT 捕获时间寄存器 (FDCAN_TTCPT)

FDCAN TT Capture Time Register

地址：0x013C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SWV[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CCV[5:0]					
										r	r	r	r	r	r

位 31:16 **SWV**：停止监视值 (Stop Watch Value)。
当停止监控触发引脚上出现上升/下降沿时（通过 TTOCN[SWP] 配置），如果 TTOCN[SWS] 不为“00”且 TTIR[SWE] 为“0”，则由 TTOCN[SWS] 选择的实际时间值（周期时间、本地时间、全局时间）会复制到 SWV，并且 TTIR[SWE] 将置“1”。通过复位 TTIR[SWE] 使能捕获下一停止监控值。
0x0000-FFFF：捕获的停止监视值

位 15:6 保留

位 5:0 **CT**：周期计数值 (Cycle Count Value)
与 SWV 一起捕获的周期计数值。
0x00-3F：捕获的周期计数值

56.4.63 FDCAN TT 周期同步标记寄存器 (FDCAN_TTCSM)

FDCAN TT Cycle Sync Mark Register

地址：0x0140

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CSM[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:16 保留

位 15:0 **CSM**：周期同步标记 (Cycle Sync Mark)。
周期同步标记以周期时间为单位进行测量。当参考消息生效时，此位会更新，并会在下一参考消息生效之前保持其值。
0x0000-FFFF：捕获的周期时间

56.4.64 FDCAN TT 触发选择寄存器 (FDCAN_TTTS)

FDCAN TT Trigger Select Register

FDCAN_TTTS 寄存器中的设置会选择将用作事件触发和停止监控触发的输入。

地址：0x0300

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	EVTSEL[1:0]		Res.	Res.	SWTDEL[1:0]	
										rw	rw			rw	rw

位 31:6 保留

位 5:4 **EVTSEL**：事件触发输入选择 (Event trigger input selection)
这些位用于选择要用作事件触发的输入
00：fdcan1_swt0
01：fdcan1_swt1
10：fdcan1_swt2
11：fdcan1_swt3

位 3:2 保留

位 1:0 **SWTDEL**：停止监视触发输入选择 (Stop watch trigger input selection)
这些位用于选择要用作停止监视触发的输入
00：fdcan1_evt0
01：fdcan1_evt1
10：fdcan1_evt2
11：fdcan1_evt3

56.4.65 FDCAN 寄存器映射和复位值表

表 475. FDCAN 寄存器映射和复位值

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x0000	FDCAN_CREL	REL [3:0]				STEP [3:0]				SUBSTEP [3:0]				YEAR [3:0]				MON [7:0]								DAY [7:0]							
	Reset value																																
0x0004	FDCAN_ENDN	ETV[31:0]																															
	Reset value	1	0	0	0	0	1	1	1	0	1	1	0	0	1	1	1	0	1	0	0	0	0	1	1	0	0	1	0	0	0	0	1
0x0008	Reserved	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x000C	FDCAN_DBTP	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TDC	Res.	Res.	DBRP[4:0]					Res.	Res.	Res.	DTSEG1[4:0]					DTSEG2[3:0]				DSJW[3:0]			
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	1	1	0	0	1	1
0x0010	FDCAN_TEST	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RX	TX[1:0]		LBCK	CAT	CAM	TAT	TAM
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1	0	1	0
0x0014	FDCAN_RWD	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WDV[7:0]								WDC[7:0]							
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0018	FDCAN _CCCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	NISO	TXP	EFBI	PXHD	Res.	Res.	BRSE	FDOE	TEST	DAR	MON	CSR	CSA	ASM	CCE	INT
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
0x001C	FDCAN_NBTP	NSJW[6:0]								NBRP[8:0]								NTSEG1[7:0]								Res.	NTSEG2[6:0]						
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	1	1	0	0	1	1
0x0020	FDCAN_TSCC	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	NBRP[3:0]				Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TSS[1:0]	
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0024	FDCAN_TSCV	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TSC[15:0]															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0028	FDCAN_TOCC	TOP[15:0]																Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TOS[1:0]		ETOC
	Reset value	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x002C	FDCAN_TOCV	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TOC[15:0]															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0x0030 to 0x003F	Reserved																																
	Reset value																																
0x0040	FDCAN_ECR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TDCV[7:0]								RP	REC[6:0]							TEC[7:0]							
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

表 475. FDCAN 寄存器映射和复位值（续）

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x0044	FDCAN_PSR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TDCV[6:0]							Res.	PXE	REDL	RBRS	RESI	DLEC[2:0]			BO	EW	EP	ACT[1:0]		LEC[2:0]		
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1
0x0048	FDCAN_TDCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TDCO[6:0]							Res.	TDCF[6:0]						
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		0	0	0	0	0	0	0	0
0x0050	FDCAN_IR	Res.	Res.	ARA	PED	PEA	WDI	BO	EW	EP	ELO	BEU	BEC	DRX	TOO	MRAF	TSW	TEFL	TEFF	TEFW	TEFN	TFE	TCF	TC	HPM	RF1L	RF1F	RF1W	RF1N	RF0L	RF0F	RF0W	RF0N
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0054	FDCAN_IE	Res.	Res.	ARAE	PEDE	PEAE	WDIE	BOE	EWE	EPE	ELOE	BEUE	BECE	DRXE	TOOE	MRAFE	TSWE	TEFLE	TEFFE	TEFWE	TEFNE	TFEE	TCFE	TCE	HPME	RF1LE	RF1FE	RF1WE	RF1NE	RF0LE	RF0FE	RF0WE	RF0NE
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0058	FDCAN_ILS	Res.	Res.	ARAL	PEDL	PEAL	WDIL	BOL	EWL	EPL	ELOL	BEUL	BECL	DRXL	TOOL	MRAFL	TSWL	TEFLL	TEFFL	TEFWL	TEFNL	TFEL	TCFL	TCL	HPML	RF1LL	RF1FL	RF1WL	RF1NL	RF0LL	RF0FL	RF0WL	RF0NL
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x005C	FDCAN_ILE	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	EINT1	EINT0
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0060 to 0x007F	Reserved																																
	Reset value																																
0x0080	FDCAN_GFC	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ANFS[1:0]		ANFE[1:0]		RRFS	RRFE
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0084	FDCAN _SIDFC	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LSS[7:0]								FLSSA[13:0]														Res.	Res.
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0088	FDCAN_ILS	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LSE[6:0]							FLESA[13:0]														Res.	Res.
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0090	FDCAN _XIDAM	Res.	Res.	Res.	EIDM[28:0]																												
	Reset value	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0x0094	FDCAN _HPMS	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FLST	FIDX[6:0]							MSI[1:0]	BIDX[5:0]						
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0098	FDCAN _NDAT1	ND31	ND30	ND29	ND28	ND27	ND26	ND25	ND24	ND23	ND22	ND21	ND20	ND19	ND18	ND17	ND16	ND15	ND14	ND13	ND12	ND11	ND10	ND9	ND8	ND7	ND6	ND5	ND4	ND3	ND2	ND1	ND0
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

表 475. FDCAN 寄存器映射和复位值（续）

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x009C	FDCAN _NDAT2	ND63	ND62	ND61	ND60	ND59	ND58	ND57	ND56	ND55	ND54	ND53	ND52	ND51	ND50	ND49	ND48	ND47	ND46	ND45	ND44	ND43	ND42	ND41	ND40	ND39	ND38	ND37	ND36	ND35	ND34	ND33	ND32
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x00A0	FDCAN _RXF0C	F0OM	F0WM[6:0]							Res.	F0S[6:0]							F0SA[13:0]														Res.	Res.
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x00A4	FDCAN _RXF0S	Res.	Res.	Res.	Res.	Res.	Res.	RF0L	F0F	Res.	Res.	F0PI[5:0]						Res.	Res.	F0GI[5:0]						Res.	F0FL[6:0]						
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x00A8	FDCAN _RXF0A	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	F0FL[5:0]					
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x00AC	FDCAN_RXBC	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RBSA[13:0]														Res.	Res.
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x00B0	FDCAN _RXF1C	F1OM	F1WM[6:0]							Res.	F1S[6:0]							F1SA[13:0]														Res.	Res.
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x00B4	FDCAN _RXF1S	DMS[1:0]		Res.	Res.	Res.	Res.	RF1L	F1F	Res.	Res.	Res.F1PI[5:0]						Res.	Res.	Res.F1GI[5:0]						Res.	Res.F1FL[6:0]						
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x00B8	FDCAN _RXF1A	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	F1FL[5:0]					
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x00BC	FDCAN _RXESC	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RBDS[2:0]			Res.	F1DS[2:0]			Res.	F0DS[2:0]		
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x00C0	FDCAN_TXBC	Res.	TFQM	TFQS[5:0]						Res.	Res.	NDTB[5:0]						TBSA[13:0]														Res.	Res.
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x00C4	FDCAN _TXFQS	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TFQF	TFQPI[4:0]					Res.	Res.	Res.	TFGI[4:0]					Res.	Res.	TFFL[5:0]					
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x00C8	FDCAN _TXESC	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	F0DS[2:0]		
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x00CC	FDCAN _TXBRP	TRP31	TRP30	TRP29	TRP28	TRP27	TRP26	TRP25	TRP24	TRP23	TRP22	TRP21	TRP20	TRP19	TRP18	TRP17	TRP16	TRP15	TRP14	TRP13	TRP12	TRP11	TRP10	TRP9	TRP8	TRP7	TRP6	TRP5	TRP4	TRP3	TRP2	TRP1	TRP0
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

表 475. FDCAN 寄存器映射和复位值（续）

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00D0	FDCAN _TXBAR	AR31	AR30	AR29	AR28	AR27	AR26	AR25	AR24	AR23	AR22	AR21	AR20	AR19	AR18	AR17	AR16	AR15	AR14	AR13	AR12	AR11	AR10	AR9	AR8	AR7	AR6	AR5	AR4	AR3	AR2	AR1	AR0
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x00D4	FDCAN _TXBCR	CR31	CR30	CR29	CR28	CR27	CR26	CR25	CR24	CR23	CR22	CR21	CR20	CR19	CR18	CR17	CR16	CR15	CR14	CR13	CR12	CR11	CR10	CR9	CR8	CR7	CR6	CR5	CR4	CR3	CR2	CR1	CR0
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x00D8	FDCAN _TXBTO	TO31	TO30	TO29	TO28	TO27	TO26	TO25	TO24	TO23	TO22	TO21	TO20	TO19	TO18	TO17	TO16	TO15	TO14	TO13	TO12	TO11	TO10	TO9	TO8	TO7	TO6	TO5	TO4	TO3	TO2	TO1	TO0
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x00DC	FDCAN _TXBCF	CF31	CF30	CF29	CF28	CF27	CF26	CF25	CF24	CF23	CF22	CF21	CF20	CF19	CF18	CF17	CF16	CF15	CF14	CF13	CF12	CF11	CF10	CF9	CF8	CF7	CF6	CF5	CF4	CF3	CF2	CF1	CF0
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x00E0	FDCAN _TXBTIE	TIE31	TIE30	TIE29	TIE28	TIE27	TIE26	TIE25	TIE24	TIE23	TIE22	TIE21	TIE20	TIE19	TIE18	TIE17	TIE16	TIE15	TIE14	TIE13	TIE12	TIE11	TIE10	TIE9	TIE8	TIE7	TIE6	TIE5	TIE4	TIE3	TIE2	TIE1	TIE0
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x00E4	FDCAN _TXBCIE	CFIE31	CFIE30	CFIE29	CFIE28	CFIE27	CFIE26	CFIE25	CFIE24	CFIE23	CFIE22	CFIE21	CFIE20	CFIE19	CFIE18	CFIE17	CFIE16	CFIE15	CFIE14	CFIE13	CFIE12	CFIE11	CFIE10	CFIE9	CFIE8	CFIE7	CFIE6	CFIE5	CFIE4	CFIE3	CFIE2	CFIE1	CFIE0
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x00F0	FDCAN _TXEFC	Res.	Res.	EFWM[5:0]						Res.	Res.	EFS[5:0]						EFSA[15:2]														Res.	Res.
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x00F4	FDCAN _TXEFS	Res.	Res.	Res.	Res.	Res.	Res.	TEFL	EFF	Res.	Res.	Res.	EFPI[4:0]					Res.	Res.	Res.	EFG[4:0]					Res.	Res.	EFFL[5:0]					
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x00F8	FDCAN _TXEFA	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	EFAI[4:0]				
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0100	FDCAN _TTTMC	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TME[6:0]							TMSA[15:2]														Res.	Res.
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0104	FDCAN _TTRMC	RMPS	XTD	Res.	RID[28:0]																												
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0108	FDCAN _TTOCF	Res.	Res.	Res.	Res.	Res.	Res.	EVTP	ECC	EGTF	AWL[7:0]							EECS	IRTO[6:0]							LDSDL[2:0]			TM	GEN	Res.	OM[1:0]	
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x010C	FDCAN _TTMLM	Res.	Res.	Res.	Res.	ENTT[11:0]												Res.	Res.	Res.	Res.	TXEW[3:0]				CSS[1:0]		CCM[5:0]					
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

表 475. FDCAN 寄存器映射和复位值（续）

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x0110	FDCAN _TURCF	ELT	Res.	DC[13:0]														NCL[15:0]															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0114	FDCAN _TXBCIE	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LCKC	Res.	ESCN	NIG	TMG	FGP	GCS	TTIE	TMC[1:0]		RTIE	SWS[1:0]		SWP	ECS	SBC
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0118	FDCAN _TTGTP	CTP[15:0]																TP[15:0]															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x011C	FDCAN _TTTMK	LCKM	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TICC[6:0]							TM[15:0]															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0120	FDCAN_TTIR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CER	AW	WT	IWT	ELC	SE2	SE1	TXO	TXU	GTE	GTD	GTW	SWE	TTMI	RTMI	SOG	CSM	SMC	SBC
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0124	FDCAN_TTIE	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CERE	AWE	WTE	IWTE	ELCE	SE2E	SE1E	TXOE	TXUE	GTEE	GTDE	GTWE	SWEE	TTMIE	RTMIE	SOGE	CSME	SMCE	SBCE
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0128	FDCAN_TTILS	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CERL	AWL	WTL	IWTL	ELCL	SE2L	SE1L	TXOL	TXUL	GTEL	GTDL	GTWL	SWEL	TTMIL	RTMIL	SOGL	CSML	SMCL	SBCL
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x012C	FDCAN _TTOST	SPL	WECS	AWE	WFE	GSI	TMP[2:0]			GFI	WGTD	Res.	Res.	Res.	Res.	Res.	Res.	RTO[7:0]								QCS	QGTP	SYS[1:0]		MS[1:0]		EL[1:0]	
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0130	FDCAN _TURNA	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	NAV[17:0]																	
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0134	FDCAN _TTLGT	GT[15:0]																LT[15:0]															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0138	FDCAN _TTCTC	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CC[5:0]						CT[15:0]															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x013C	FDCAN _TTCTP	SWV[15:0]																Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CCV[5:0]					
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0140	FDCAN _TTCSM	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CSM[15:0]															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0144 to 0x01FC	Reserved																																
	Reset value																																

表 475. FDCAN 寄存器映射和复位值（续）

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x0300	FDCAN _TTTS	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	EVTSEL[1:0]		Res.	Res.	SWTSEL[1:0]	
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

56.5 CCU寄存器

56.5.1 时钟校准单元内核释放寄存器 (CCU_CREL)

Clock Calibration Unit Core Release Register

偏移地址：0x0000

复位值：0xrrrd dddd

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
REL[3:0]				STEP[3:0]				SUBSTEP[3:0]				YEAR[3:0]			
r	r	r	r	r	r	r	r	r	r	r	r	d	d	d	d
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MON[7:0]								DAY[7:0]							
d	d	d	d	d	d	d	d	d	d	d	d	d	d	d	d

位 31:28 **REL**：内核释放 (Core Release)
一位，BCD。

位 27:24 **STEP**：内核释放步 (Step of Core Release)
一位，BCD。

位 23:20 **SUBSTEP**：内核释放子步 (Sub-step of Core Release)
一位，BCD。

位 19:16 **YEAR**：时间戳年 (Time Stamp Year)
一位，BCD。

位 15:8 **MON**：时间戳月 (Time Stamp Month)
两位，BCD。

位 7:0 **DAY**：时间戳日 (Time Stamp Day)
两位，BCD。

56.5.2 校准配置寄存器 (CCU_CCFG)

Calibration Configuration Register

偏移地址：0x0004

复位值：0x0000 0004

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SWR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CDIV[3:0]			
rw												rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OCPM[7:0]								CFL	BCC	Res.	TQBT[5:0]				
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw		rw	rw	rw	rw	rw

位 31 **SWR**：软件复位 (Software reset)

向此位写入“1”会将校准 FSM 复位为 Not_Calibrated 状态（CSTAT.CALS = “00”）。校准看门狗值 CWD.WDV 也会复位。寄存器 CCFG、CSTAT 和校准看门狗配置 CWD.WDC 不会更改。在复位完成之前，此位保持置 1 状态。

仅当 M_CAN 控制位 CCCR.CCE = “1”且 CCCR.INIT = “1”时，才能通过主机 CPU 对标有“P = 受写保护”的寄存器/位进行写访问。

位 30:20 保留

位 19:16 **CDIV**：时钟分频器 (Clock Divider)

如果绕过时钟校准（BCC = “1”），则必须配置时钟分频，以确保满足 M_CAN 要求。

0000：1 分频
0001：2 分频
0010：4 分频
0011：6 分频
0100：8 分频
0101：10 分频
0110：12 分频
0111：14 分频
1000：16 分频
1001：18 分频
1010：20 分频
1011：22 分频
1100：24 分频
1101：26 分频
1110：28 分频
1111：30 分频

仅当 M_CAN 控制位 CCCR.CCE = “1”且 CCCR.INIT = “1”时，才能通过主机 CPU 对标有“P = 受写保护”的寄存器/位进行写访问。

位 15:8 **OCPM**：振荡器时钟周期最小值 (Oscillator Clock Periods Minimum)

以两个 CAN 位时间为单位配置最小周期数。在基本校准中会使用 OCPM，以免总线上存在毛刺时出现测量错误。配置的周期数为 OCPM × 32。配置取决于频率（80 MHz 到 500 Mhz）以及在 FDCAN1 和 FDCAN2 中配置的比特率 （125 kb/s 到 1 Mb/s）。建议将该值配置为略小于两个 CAN 位时间。当 fdcan_ker_ck 频率为 80 MHz 且 CAN 比特率为 1 Mb/s 时，复位值为 1.6 倍的位时间。

仅当 M_CAN 控制位 CCCR.CCE = “1”且 CCCR.INIT = “1”时，才能通过主机 CPU 对标有“P = 受写保护”的寄存器/位进行写访问。

位 7 **CFL**：校准字段长度 (Calibration Field Length)

0：校准字段长度为 32 位
1：校准字段长度为 64 位

仅当 M_CAN 控制位 CCCR.CCE = “1”且 CCCR.INIT = “1”时，才能通过主机 CPU 对标有“P = 受写保护”的寄存器/位进行写访问。

位 6 **BCC**：绕过时钟校准 (Bypass Clock Calibration)

如果此位置 1，时钟输入 fdcan_ker_ck 会通过可由 CDIV 配置的时钟分频器发送到时间片时钟，cu_cok 始终为“1”。在这种情况下，必须配置已连接 M_CAN 的波特率预分频器，以生成 M_CAN 内部时间片时钟。

0：时钟校准单元生成时间片时钟
1：绕过时钟校准单元（默认配置）

注： 只要 fdcan_ker_ck 等于或大于 80 MHz，即使 BCC = “1”，CAN 时钟校准单元也会正常工作。校准状态可从 CSTAT 中读取。

位 5 保留

位 4:0 **TQBT**：每个位时间的时间片 (Time Quanta per Bit Time)

配置每个位时间的时间片数，与在 FDCAN1 和 FDCAN2 中配置的值相同。得出的时间片时钟 fdcan_tq_ck 的范围为 0.5 MHz（比特率为 125 kb/s，每个位时间有 4 个 tq）到 25 MHz（比特率为 1 Mb/s，每个位时间有 25 个 tq）.有效值为 4 到 25。如果配置的值小于 4，则会被解析为 4，如果配置的值大于 25，则会被解析为 25。

仅当 M_CAN 控制位 CCCR.CCE = “1”且 CCCR.INIT = “1”时，才能通过主机 CPU 对标有“P = 受写保护”的寄存器/位进行写访问。

56.5.3 校准状态寄存器 (CCU_CSTAT)

Calibration Status Register

偏移地址：0x0008

复位值：0x0203 FFFF

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CALS[1:0]			TQC[11:0]											OCPC[17:16]	
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OCPC[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:30 **CALS**：校准状态 (Calibration State)

00：Not_Calibrated

01：Basic_Calibrated

10：Precision_Calibrated

11：保留

位 29 保留

位 28:18 **TQC**：时间片计数器 (Time Quanta Counter)

在校准字段捕获的时间片数（32 位或 64 位）。仅当时钟校准单元处于 Precision_Calibrated 状态时，这些位才有效。

位 17:0 **OCPC**：振荡器时钟周期计数器 (Oscillator Clock Periods Counter)

在校准字段捕获的振荡器时钟周期数（32 或 64 位）。仅当时钟校准单元处于 Precision_Calibrated 状态时，这些位才有效。

56.5.4 校准看门狗寄存器 (CCU_CWD)

Calibration Watchdog Register

偏移地址：0x000C

复位值：0x0000 0000

当校准 FSM 处于 Not_Calibrated 状态（CSTAT.CALS = “00”）时，校准看门狗会在第一个下降沿后启动。在该状态下，校准看门狗会监控接收到的消息。如果在校准看门狗递减计数到 0 之前未接收到任何消息，校准 FSM 会停留在 Not_Calibrated 状态（CSTAT.CALS = “00”），计数器会重新载入 RWD.WDC，下一下降沿之后，基本校准会重新开始。

当校准看门狗处于 Basic_Calibrated 状态时（CSTAT.CALS = “01”），每次接收到消息，校准看门狗都会重新启动。如果在校准看门狗递减计数到 0 之前未接收到任何消息，校准 FSM 会恢复 Not_Calibrated 状态（CSTAT.CALS = “00”），计数器会重新载入 RWD.WDC，下一下降沿之后，基本校准会重新开始。

接收到石英消息时，校准 FSM 会进入 Precision_Calibrated 状态（CSTAT.CALS = “10”），校准看门狗会重新启动。在该状态下，校准看门狗会监控接收到的石英消息。如果连接的 TTCAN 在校准看门狗递减计数到 0 之前未接收到任何由石英控制的节点发送的消息，校准 FSM 会切换回 Basic_Calibrated 状态（CSTAT.CALS = “01”）。当连接的 TTCAN 上的 CAN 协议引擎启动，也就是 INIT 位复位时，信号有效。

校准看门狗事件还会将中断标志 CUIR.CWE 置 1。如果中断线通过 CUIE.CWEE 使能，则会被激活（设为高电平）。中断标志 CUIR.CWE 复位之前，中断线会保持有效状态。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
WDV[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WDC[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:16 **WDV**：看门狗值 (Watchdog Value)
实际校准看门狗计数器值。

位 15:0 **WDC**：看门狗配置 (Watchdog Configuration)
校准看门狗计数器的起始值。如果使用复位值 “00”，计数器会禁止。
仅当 M_CAN 控制位 CCCR.CCE = “1” 且 CCCR.INIT = “1” 时，才能通过主机 CPU 对标有 “P = 受写保护” 的寄存器/位进行写访问。

56.5.5 时钟校准单元中断寄存器 (CCU_IR)

Clock Calibration Unit Interrupt Register

如果检测到其中一个列出的条件（边沿有效），标志会置 1。在主机将标志清零之前，标志保持置 1 状态。通过向对应位位置写入 “1” 将标志清零。写入 “0” 不会起任何作用。硬复位会将寄存器清零。CUIE 的配置控制着是否生成中断。

偏移地址：0x0010

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CSC	CWE
														rw	rw

位 31:2 保留

位 1 **CSC**：校准状态更改 (Calibration State Changed)
0：校准状态未更改
1：校准状态已更改

位 0 **CWE**：校准看门狗事件 (Calibration Watchdog Event)
0：无校准看门狗事件
1：发生校准看门狗事件

56.5.6 时钟校准单元中断使能寄存器 (CCU_IE)

Clock Calibration Unit Interrupt Enable Register

偏移地址：0x0014

复位值：0x0000 0000

CU 中断使能寄存器中的设置决定了是否将在中断线上指示 CU 中断寄存器中的状态更改。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CSCE	CWEE
														rw	rw

位 31:2 保留

位 1 **CSCE**：校准状态更改使能 (Calibration State Changed Enable)
0：禁止中断
1：使能中断

位 0 **CWEE**：校准看门狗事件使能 (Calibration Watchdog Event Enable)
0：禁止中断
1：使能中断

56.5.7 CCU 寄存器映射和复位值表

表 476. CCU 寄存器映射和复位值

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x0000	CCU_CREL	REL[3:0]				STEP[3:0]				SUBSTEP[3:0]				YEAR[3:0]				MON[7:0]								DAY[7:0]							
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0004	CCU _CCFG	SWR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CDIV[3:0]				CSM[7:0]								CFL	BCC	Res.	TQBT[4:0]				
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0008	CCU _CSTAT	CALS[1:0]		Res.	TQC[10:0]											OCPC[17:0]																	
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x000C	CCU_CWD	WDV[15:0]																WDC[15:0]															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0010	CCU_IR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CSC	CWE
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0014	CCU_IR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CSCE	CWEE
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

57 USB on-the-go 高速(OTG_HS)

57.1 前言

Portions Copyright (c) 2004, 2005 Synopsys, Inc. 保留所有权利。使用须经许可。

本节介绍了 OTG_HS 控制器的架构和编程模型。

本节中使用以下缩写：

FS	全速
LS	低速
HS	高速
MAC	介质访问控制器
OTG	On-the-go
PFC	数据包 FIFO 控制器
PHY	物理层
USB	通用串行总线
UTMI	USB 2.0 收发器宏单元接口 (UTMI)
UTMI	USB 收发器宏单元接口
ULPI	UTMI+ 引脚数目少的接口
LPM	链路层电源管理
BCD	电池充电检测
HNP	主机协商协议
SRP	会话请求协议

参考文档如下：

- USB On-The-Go 补充标准，第 2.0 版
- 通用串行总线规范第 2.0 版
- 基于 USB 2.0 规范的 USB 2.0 链路层电源管理补充工程变更通知，2007 年 7 月 16 日
- USB 2.0 ECN 的勘误表：链路层电源管理 (LPM) - 2007 年 7 月
- 电池充电规范第 1.2 版

USB OTG 是一款双角色设备 (DRD) 控制器，同时支持从机功能和主机功能，完全符合 USB 2.0 规范的 On-The-Go 补充标准。此外，该控制器也可配置为“仅主机”模式或“仅从机”模式，完全符合 *USB 2.0 规范*。OTG_HS 支持下面*表 477：OTG_HS 支持的速度*中定义的速度。USB OTG 既支持 HNP 也支持 SRP。OTG 模式下需要的唯一外部器件是提供 V_{BUS} 电压的电荷泵。

表 477. OTG_HS 支持的速度

	HS (480 Mb/s)	FS (12 Mb/s)	LS (1.5 Mb/s)
主机模式	X	X	X
设备模式	X	X	-

57.2 OTG 主要特性

主要特性可分为三类：通用特性、主机模式特性和从机模式特性。

57.2.1 通用特性

OTG_HS 接口的通用特性如下：

- 经 USB-IF 认证，符合通用串行总线规范第 2.0 版
- OTG HS 支持以下 PHY 接口：
  - 片上全速 PHY
  - 连接外部全速 PHY 的 I2C 接口
  - 连接外部高速 PHY 的 ULPI 接口
- 模块内嵌的 PHY 还完全支持定义在标准规范 OTG 补充第 2.0 版中的 OTG 协议
  - 支持 A-B 器件识别（ID 线）
  - 支持主机协商协议 (HNP) 和会话请求协议 (SRP)
  - 允许主机关闭 V_{BUS} 以在 OTG 应用中节省电池电量
  - 支持通过内部比较器对 V_{BUS} 电平采取 OTG 监控
  - 支持主机到从机的角色动态切换
- 可通过软件配置为以下角色：
  - 具有 SRP 功能的 USB HS 从机（B 器件）
  - 具有 SRP 功能的 USB HS/LS 主机（A 器件）
  - USB On-The-Go 全速双角色设备
- 支持 HS SOF 和 LS Keep-alive 令牌
  - SOF 脉冲可通过 PAD 输出
  - SOF 脉冲通过内部连接到定时器 (TIMx)
  - 可配置的帧周期
  - 可配置的帧结束中断
- OTG_HS 内嵌 DMA，并可软件配置 AHB 的批量传输类型。
- 具有省电功能，例如在 USB 挂起期间停止系统、关闭数字模块时钟、对 PHY 和 DFIFO 电源加以管理。
- 具有采用高级 FIFO 控制的 4 KB 专用 RAM：
  - 可将 RAM 空间划分为不同 FIFO，以便灵活有效地使用 RAM
  - 每个 FIFO 可存储多个数据包
  - 动态分配存储区
  - FIFO 大小可配置为非 2 的幂次方值，以便连续使用存储单元
- 一帧之内可以无需要应用程序干预，以达到最大 USB 带宽。
- 它支持电池充电规范第 1.2 版中介绍的充电端口检测（仅限 FS PHY 收发器）

57.2.2 主机模式特性

OTG_HS 接口在主机模式下具有以下主要特性和要求：

- 通过外部电荷泵生成 V_{BUS} 电压。
- 多达 16 个主机通道（又称之为管道）：每个通道都可以动态实现重新配置，可支持任何类型的 USB 传输。
- 内置硬件调度器可：
  - 在周期性硬件队列中存储多达 16 个中断加同步传输请求
  - 在非周期性硬件队列中存储多达 16 个控制加批量传输请求
- 管理一个共享 Rx FIFO、一个周期性传输 Tx FIFO 和一个非周期性传输 Tx FIFO，以有效使用 USB 数据 RAM。

57.2.3 从机模式特性

OTG_HS 接口在从机模式下具有以下特性：

- 1 个双向控制端点 0
- 8个 IN 端点 (EP)，可配置为支持批量传输、中断传输或同步传输
- 8个 OUT 端点，可配置为支持批量传输、中断传输或同步传输
- 管理一个共享 Rx FIFO 和一个 Tx-OUT FIFO，以高效使用 USB 数据 RAM
- 管理多达 9 个专用 Tx-IN FIFO（分别用于每个使能的 IN EP），以降低应用程序负荷
- 支持软断开功能。

57.3 OTG 实现

表 478. STM32H7 的 OTG 实现 (1)

USB 功能	OTG_HS1(2)	OTG_HS2(3)
设备双向端点（包括 EP0）	9	
主机模式通道	16	
专用 SRAM 的大小	4 KB	
USB 2.0 链路层电源管理 (LPM)	X	
支持 OTG 版本	2.0	
连接检测协议 (ADP) 支持	-	
电池充电检测 (BCD) 支持	X	
ULPI 可通过多路复用连接到主 IO	X	-

1. “X” = 支持，“-” = 不支持
2. 兼容高速操作。
3. 不兼容高速操作。

57.4 OTG 功能说明

57.4.1 OTG 框图

STM32H7 中存在两个 OTG_HS 模块（OTG_HS1 和 OTG_HS2）。

尽管他们都可以编程为 HS 操作，但只有 OTG_HS1 具有可访问的 ULPI 接口，因此允许使用外部 HS 收发器进行高速操作。

图 740. 高速 OTG 模块框图 (OTG_HS1)

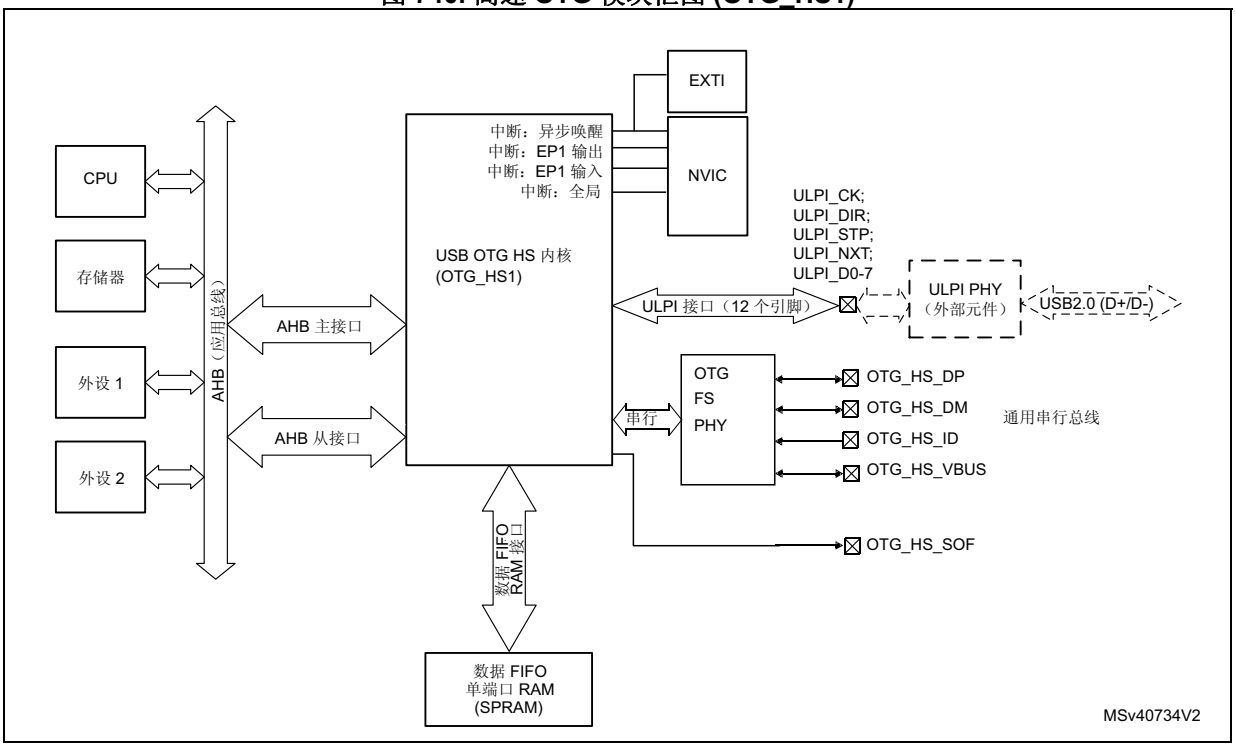
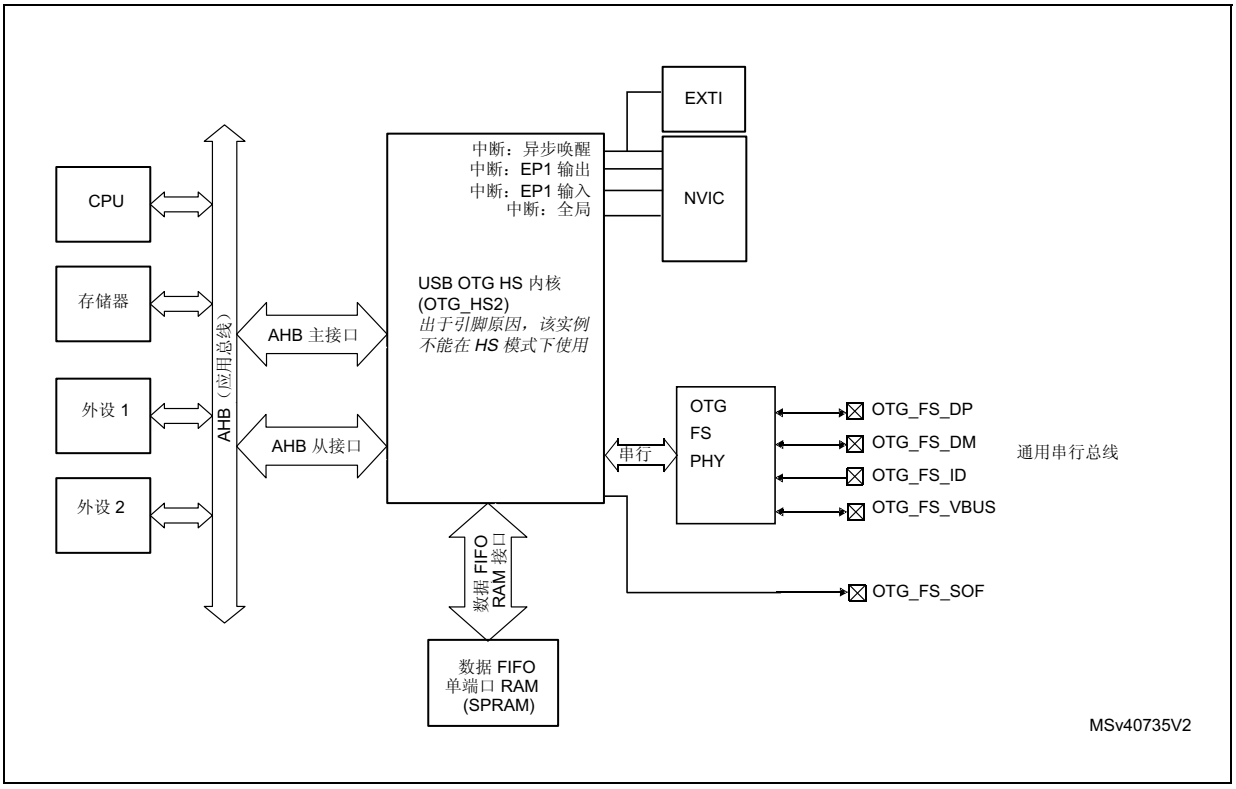


图 741. 高速 OTG 模块框图 (OTG_HS2)



57.4.2 USB OTG 引脚和内核信号

表 479. OTG_FS/OTG_HS 输入/输出信号

信号名称	信号类型	说明
usb_hclk	数字输入	USB OTG 接口时钟
usb_sof_evt	数字输出	USB OTG 帧起始事件（用于片上外设）
usb_wkup	数字输出	USB OTG 唤醒事件输出
usb_gbl_it	数字输出	USB OTG 全局中断
usb_ep1_in_it	数字输出	USB OTG 端点 1 输入中断
usb_ep1_out_it	数字输出	USB OTG 端点 1 输出中断

表 480. OTG_FS/OTG_HS 输入/输出信号

信号名称	信号类型	说明
OTG_[HS/FS]_DP	数字输入/输出	USB OTG 正向数据线
OTG_[HS/FS]_DM	数字输入/输出	USB OTG 负向数据线
OTG_[HS/FS]_ID	数字输入	USB OTG ID
OTG_[HS/FS]_VBUS	数字双向	USB OTG 总线电源
OTG_[HS/FS]_SOF	数字输入	USB OTG 帧起始事件输出（在 GPIO 上）
OTG_HS_ULPI_CK	数字输入	USB OTG ULPI 时钟
OTG_HS_ULPI_DIR	数字输入	USB OTG ULPI 数据总线方向控制
OTG_HS_ULPI_STP	数字输出	USB OTG ULPI 数据流停止
OTG_HS_ULPI_NXT	数字输入	USB OTG ULPI 下一数据流请求
OTG_HS_ULPI_D[0..7]	数字输入/输出	USB OTG ULPI 8 位双向数据总线

57.4.3 OTG 模块

USB OTG 通过复位和时钟控制模块接收 48 MHz 的时钟。USB 时钟用于在全速通信时驱动 48 MHz 时钟域，必须在配置 OTG 模块前使能。

CPU 通过 AHB 外设总线对 OTG 模块寄存器进行读写操作，通过*第 57.12 节：OTG_HS 中断*中所述的 USB OTG 中断线接收 USB 事件通知。

CPU 通过向特定的 OTG 单元（压栈寄存器）写入 32 位字来向 USB 提交数据。数据随即自动存储到 USB 数据 RAM 中配置的数据发送 FIFO 中。每个 IN 端点（从机模式）或 OUT 通道（主机模式）都有一个 Tx FIFO 压栈寄存器。

CPU 从特定的 OTG 地址（出栈寄存器）读取 32 位字，以读取来自 USB 的数据。数据随即从在 4 KB USB 数据 RAM 内配置的共享 Rx FIFO 中弹出。每个 OUT 端点或 IN 通道都有一个 Rx FIFO 出栈寄存器。

USB 协议层通过串行接口引擎 (SIE) 驱动，并通过片上物理层 (PHY) 中的收发器模块经由 USB 进行数据的串行通信，也可以通过外部 OTG_HS PHY 或外部 OTG_FS PHY 经由 I^2C 接口进行数据的串行通信。

57.4.4 嵌入式全速 OTG PHY

全速 OTG PHY 包括以下组成部分：

- 供主机和设备使用的 FS/LS 收发器模块。直接在单端 USB 线上驱动发送和接收操作。
- 集成 ID 上拉电阻，用于对 ID 线进行采样，以便识别 A/B 器件。
- 由 OTG_HS 模块控制的 DP/DM 集成上拉电阻和下拉电阻，具体使能哪种电阻取决于设备的当前角色。作为外设使用时，只要检测到 V_{BUS} 为有效电平（B 会话有效），立即使能 DP 上拉电阻，以告知全速设备的连接。主机模式下则使能 DP/DM 上的下拉电阻。通过主机协商协议 (HNP) 更改从机角色时，将在上拉电阻和下拉电阻之间动态切换。
- 上拉/下拉电阻 ECN 电路。根据适用于 USB 2.0 版的电阻 ECN 规定，DP 上拉电路包括 2 个由 OTG_HS 单独进行控制的电阻。对 DP 上拉阻值的动态调整可以提高噪声抑制能力和 Tx/Rx 信号质量。
- 带滞回功能的 V_{BUS} 感应比较器，用于检测 V_{BUS} 有效、A-B 会话有效和会话端电压阈值。执行 USB 操作期间，这些比较器用于驱动会话请求协议 (SRP)、检测会话的有效启动和结束条件，以及持续监视 V_{BUS} 供电情况。

为确保 USB OTG_HS 模块正常工作，AHB 频率应大于 30 MHz。

57.4.5 高速 OTG PHY

USB OTG_HS 内核包含一个 ULPI 接口以连接外部 HS PHY。

ULPI 接口仅适用于 OTG HS1（请参见*图 57.4.1：OTG 框图*）。

57.4.6 使用 I2C 接口的外部全速 OTG PHY

USB OTG_HS 模块嵌入了一个 I2C 接口，用于连接到外部 FS PHY。

57.5 OTG 双角色设备 (DRD)

57.5.1 ID 线检测

采取主机还是从机（默认设置）角色取决于 ID 输入引脚的电平。插入 USB 电缆时可根据是 MicroA 还是 MicroB 插头连接到 micro-AB 插座来确定 ID 线状态。

- 如果 USB 电缆的 B 端连入，其 ID 线悬空，则由于设备在 ID 线上的集成上拉电阻设备将检测到 ID 高电平并确认采取默认的从机角色。在此配置中，OTG_HS 符合“USB2.0 On-The-Go 规范第 2.0 版补充标准中第 4.2.4 节 ID 引脚”中所述的 FSM 标准。
- 如果 USB 电缆的 A 端连入，其 ID 线接地，则 OTG_FS/OTG_HS 将发出 ID 线状态更改中断（OTG_GINTSTS 中的 CIDSCHG 位）以告知应用程序初始化主机，并自动切换为主机角色。在此配置中，OTG_HS 符合“USB2.0 On-The-Go 规范第 2.0 版补充标准中第 4.2.4 节 ID 引脚”中所述的 FSM 标准。

57.5.2 HNP 双角色设备

全局 USB 配置寄存器中的 HNP 使能位（OTG_ GUSBCFG 中的 HNPCAP 位）可使 OTG_FS/OTG_HS 模块根据主机协商协议 (HNP) 动态切换角色，例如从 A 主机切换为 A 从机（反之亦然），或者从 B 从机切换为 B 主机（反之亦然）。通过全局 OTG 控制和状态寄存器中的连接器 ID 状态位（OTG_GOTGCTL 中的 CIDSTS 位）及全局中断和状态寄存器中的当前工作模式位（OTG_GINTSTS 中的 CMOD 位）二者的组合值可读取设备当前状态。

*第 57.15 节：OTG_HS 编程模型*详细介绍了 HNP 编程模型。

57.5.3 SRP 双角色设备

全局 USB 配置寄存器中的 SRP 使能位（OTG_GUSBCFG 中的 SRPCAP 位）可使 OTG_FS/OTG_HS 模块关闭 VBUS 供电，为 A 器件节省电能。注意，无论 OTG_HS 采取主机角色还是从机角色，A 器件将始终负责 VBUS 的提供。

*第 57.15 节：OTG_HS 编程模型*详细介绍了 SRP A/B 器件编程模型。

57.6 USB 设备

本节介绍了 OTG_HS 在 USB 设备模式下所具有的功能。在以下情形下，OTG_HS 用作 USB 设备：

- OTG B 设备
  - OTG B 器件插入 USB 电缆 B 端时的默认状态
- OTG A 设备
  - HNP 将 OTG_HS 切换到其设备角色后的 OTG A 器件状态
- B 设备
  - 如果 ID 线有连接，器件与 USB 电缆的 B 端相连，并且全局 USB 配置寄存器中的 HNP 功能位（OTG_GUSBCFG 中的 HNPCAP 位）清零。
- 纯设备
  - 将*第 57.14.4 节：OTG USB 配置寄存器 (OTG_GUSBCFG)* 中的强制设备模式位 (FDMOD) 置 1，从而将 OTG_HS 内核强制用作纯 USB 设备。这种情况下，即使 USB 连接器上存在 ID 线，也会将该 ID 线忽略。

注： *要在 B 设备或仅作设备配置情形下构建总线供电的设备方案，需要添加一个外部调压器，用于从 V_{BUS} 生成所需电源。*

57.6.1 支持 SRP 功能的 USB 设备

全局 USB 配置寄存器中的 SRP 功能位（OTG_GUSBCFG 中的 SRPCAP 位）可使 OTG_HS 支持会话请求协议 (SRP)。这样一来，远程 A 器件便可以在 USB 会话挂起时，通过关闭 V_{BUS} 来节省电能。

*B 器件会话请求协议*一节详细介绍了 SRP 设备的编程模型。

57.6.2 USB 设备状态

供电状态

V_{BUS} 输入检测到 B 会话有效电压，就会使 USB 设备进入供电状态（请参见 USB2.0 第 9.1 部分）。然后，OTG_FS/OTG_HS 自动连接 DP 上拉电阻，发出全速设备与主机相连的信号并生成会话请求中断（OTG_GINTSTS 中的 SRQINT 位），指示进入供电状态。

此外，V_{BUS} 输入还可确保主机在 USB 操作期间提供有效的 V_{BUS} 电平。如果检测到 V_{BUS} 降至 B 会话有效电压以下（例如，因电源干扰或主机端口关闭引发），OTG_HS 将自动断开连接并生成检测到会话结束中断（OTG_GOTGINT 中的 SEDET 位），指示 OTG_HS 已退出供电状态。

供电状态下，OTG_HS 期望收到来自主机的复位信号。其它 USB 操作则无法执行。收到复位信号后，立即生成检测到复位中断（OTG_GINTSTS 中的 USBRST）。复位信号结束后，将生成枚举完成中断（OTG_GINTSTS 中的 ENUMDNE 位），OTG_HS 随即进入默认状态。

软断开

供电状态可借助软断开功能通过软件退出。将设备控制寄存器中的软断开位（OTG_DCTL 中的 SDIS 位）置 1 即可移除 DP 上拉电阻，此时尽管没有从主机端口实际拔出 USB 电缆，但主机端仍会发生设备断开检测中断。

默认状态

默认状态下，OTG_HS 期望从主机收到 SET_ADDRESS 命令。其它 USB 操作则无法执行。当 USB 上解码出有效 SET_ADDRESS 命令时，应用程序会将相应的数值写入设备配置寄存器中的设备地址字段（OTG_DCFG 中的 DAD 位）。OTG_HS 随即进入地址状态，并准备好以所配置的 USB 地址对主机事务进行应答。

挂起状态

OTG_HS 设备持续监视 USB 活动。在 USB 空闲时间达到 3 ms 后，将发出早期挂起中断（OTG_GINTSTS 中的 ESUSP 位），并在 3 ms 后由挂起中断（OTG_GINTSTS 中的 USBSUSP 位）确认设备进入挂起状态。然后，设备状态寄存器中的设备挂起位（OTG_DSTS 中的 SUSPSTS 位）自动置 1，OTG_HS 随即进入挂起状态。

可通过设备本身退出挂起状态。这种情况下，应用程序会将设备控制寄存器中的远程唤醒信号位（OTG_DCTL 中的 RWUSIG 位）置 1，并在 1 ms 到 15 ms 后将其清零。

但若设备检测到主机发出的恢复信号时，将生成恢复中断（OTG_GINTSTS 中的 WKUPINT 位），设备挂起位自动清零。

57.6.3 USB 设备端点

OTG_HS 模块实现了以下 USB 端点：

- 控制端点 0：
  - 双向且仅处理控制消息
  - 使用一组单独的寄存器来处理 IN 和 OUT 事务
  - 专用控制 (OTG_DIEPCTL0/OTG_DOEPCTL0) 寄存器、传输配置 (OTG_DIEPTSIZ0/OTG_DOEPTSIZ0) 寄存器和状态中断 (OTG_DIEPINT0/OTG_DOEPINT0) 寄存器。控制和传输大小寄存器中可用的位组与其它端点中稍有不同
- 8 个 IN 端点
  - 每个端点都可配置为支持同步传输、批量传输或中断传输类型
  - 每个端点都有专用控制 (OTG_DIEPCTL*x*) 寄存器、传输配置 (OTG_DIEPTSIZ*x*) 寄存器和状态中断 (OTG_DIEPINT*x*) 寄存器
  - 设备 IN 端点通用中断屏蔽寄存器 (OTG_DIEPMSK) 可用于使能/禁止所有 IN 端点（包括 EP0）上的同一类端点中断源
  - 支持未完成的同步 IN 传输中断（OTG_GINTSTS 中的 IISOIXFR 位），该中断将在当前帧中至少有一个同步 IN 端点上的传输未完成时触发。该中断和周期帧结束中断 (OTG_GINTSTS/EOPF) 一起触发。
- 8 个 OUT 端点
  - 每个端点都可配置为支持同步传输、批量传输或中断传输类型
  - 每个端点都有专用控制 (OTG_DOEPCTL*x*) 寄存器、传输配置 (OTG_DOEPTSIZ*x*) 寄存器和状态中断 (OTG_DOEPINT*x*) 寄存器
  - 设备 OUT 端点通用中断屏蔽寄存器 (OTG_DOEPMSK) 可用于使能/禁止所有 OUT 端点（包括 EP0）上的同一类端点中断源
  - 支持未完成的同步 OUT 传输中断（OTG_GINTSTS 中的 INCOMPISOOUT 位），该中断将在当前帧中至少有一个同步 OUT 端点上的传输未完成时触发。该中断和周期帧结束中断 (OTG_GINTSTS/EOPF) 一起触发。

端点控制

- 应用程序可通过设备端点 *x* IN/OUT 控制寄存器 (OTG_DIEPCTL*x*/OTG_DOEPCTL*x*) 对端点采取以下控制：
  - 端点使能/禁止
  - 在当前配置下激活端点
  - 设置 USB 传输类型（同步、批量和中断）
  - 设置支持的数据包大小
  - 设置与 IN 端点相关的 Tx FIFO 编号
  - 设置希望收到的或发送时要使用到的 data0/data1 PID（仅限批量/中断传输）
  - 设置接收或发送事务时所对应的奇/偶帧（仅限同步传输）
  - 可以设置 NAK 位，从而不论此时 FIFO 的状态如何，都对主机的请求回复 NAK
  - 可以设置 STALL 位，使得主机对该端点的令牌都被硬件回复 STALL
  - 可以将 OUT 端点设置为侦听模式，即对接收到的数据不进行 CRC 检查

端点传输

设备端点 *x* 传输大小寄存器 (OTG_DIEPTSIZ*x*/OTG_DOEPTSIZ*x*) 允许应用程序对传输大小参数进行编程并读取传输状态。必须在端点控制寄存器中的端点使能位置 1 之前完成对此寄存器的设置。使能端点后，这些字段立即变为只读状态，同时 OTG_HS 模块根据当前传输状态对这些字段进行更新。

可对以下传输参数进行编程：

- 以字节为单位的传输大小
- 构成整个传输的数据包个数

端点状态/中断

设备端点 *x* 中断寄存器 (OTG_DIEPINT*x*/OTG_DOPEPINT*x*) 指示端点在出现 USB 和 AHB 相关事件时的状态。当模块中断寄存器中的 OUT 端点中断位或 IN 端点中断位（分别为 OTG_GINTSTS 中的 OEPINT 位或 OTG_GINTSTS 中的 IEPINT 位）置 1 时，应用程序必须读取这些寄存器以获得详细信息。在应用程序读取这些寄存器之前，必须先读取设备全体端点中断 (OTG_DAINT) 寄存器，以获取设备端点 x 中断寄存器的端点编号。应用程序必须将此寄存器中的相应位清零，才能将 OTG_DAINT 和 OTG_GINTSTS 寄存器中的相应位清零。

模块提供以下状态检查和中断产生功能：

- 传输完成中断，指示应用程序 (AHB) 和 USB 端均已完成数据传输
- Setup 阶段已完成（仅针对控制传输类型的 OUT 端点）
- 相关的发送 FIFO 为半空或全空状态（IN 端点）
- NAK 应答已发送到主机（仅针对同步传输的 IN 端点）
- Tx FIFO 为空时接收到 IN 令牌（仅针对批量和中断传输类型的 IN 端点）
- 尚未使能端点时接收到 OUT 令牌
- 检测到串扰错误
- 应用程序关闭端点生效
- 应用程序对端点设置 NAK 生效（仅针对同步传输类型的 IN 端点）
- 接收到 3 个以上连续 setup 数据包（仅针对控制类型的 OUT 端点）
- 检测到超时状况（仅针对控制传输类型的 IN 端点）
- 同步传输类型的数据包未产生中断而丢失

57.7 USB 主机

本节介绍了 OTG_HS 在 USB 主机模式下所具有的功能。在以下情形下，OTG_HS 用作 USB 主机：

- OTG A 主机
  - OTG A 器件在插入 USB 电缆 A 端时的默认状态
- OTG B 主机
  - OTG B 器件被 HNP 切换为主机角色后的状态
- A 主机
  - 如果 ID 线有连接，器件与 USB 电缆的 A 端相连，全局 USB 配置寄存器中的 HNP 功能位（OTG_GUSBCFG 中的 HNPCAP 位）清零。DP/DM 线上的集成下拉电阻自动使能。
- 仅作主机
  - *57.14.4* 全局 USB 配置寄存器中的强制模式位（OTG_GUSBCFG 中的 FHMOD 位）将强制 OTG_HS 模块作为 USB 仅作主机运行。这种情况下，即使 USB 连接器上存在 ID 线，模块也会忽略 ID 线上的电平。DP/DM 线上的集成下拉电阻自动使能。

注： *微控制器不能输出 5 V 以提供 V_{BUS}。为此，必须在微控制器以外添加电荷泵或电源开关（如果应用电路板提供 5 V 电源）来驱动 5 V V_{BUS} 线。外部电荷泵可通过任何 GPIO 输出驱动。OTG A 主机、A 器件和仅作主机配置都需要使用电荷泵。*

57.7.1 支持 SRP 功能的 USB 主机

全局 USB 配置寄存器中的 SRP 使能位（OTG_GUSBCFG 中的 SRPCAP 位）可提供 SRP 支持。使能 SRP 功能后，主机可在 USB 会话挂起时通过关闭 V_{BUS} 电源来节省电能。

*A 器件会话请求协议*一节详细介绍了 SRP 主机的编程模型。

57.7.2 USB 主机状态

给主机端口供电

微控制器不能输出 5 V 以提供 V_{BUS}。为此，必须在微控制器以外添加电荷泵或电源开关（如果应用电路板提供 5 V 电源）来驱动 5 V V_{BUS} 线。外部电荷泵可通过任何 GPIO 输出驱动，或者通过连接至外部 PMIC（电源管理 IC）的 I^2C 接口驱动。当应用程序确定控制外部器件输出 V_{BUS} 时，还必须将主机端口控制和状态寄存器中的端口电源位（OTG_HPRT 中的 PPWR 位）置 1。

V_{BUS} 有效

使能 HNP 或 SRP 后，应将 VBUS 感应引脚连接到 V_{BUS}。V_{BUS} 输入可确保电荷泵在 USB 操作期间提供有效的 V_{BUS} 电平。如果 V_{BUS} 电压意外降至 V_{BUS} 有效阈值 (4.4 V) 以下，将通过会话结束检测位（OTG_GOTGINT 中的 SEDET 位）触发 OTG 中断。之后应用必须断开 V_{BUS} 电源并使端口电源位清零。

当 HNP 和 SRP 均禁止时，无需将 VBUS 感应引脚连接到 V_{BUS}。

电荷泵过流标志也可用来防止电气损坏。将电荷泵的过流标志输出连接到任意 GPIO 输入，然后将其配置为出现有效电平时生成端口中断。过流 ISR 必须立即关闭 V_{BUS} 并清零端口电源位。

主机检测设备连接

如果使能 SRP 或 HNP，即使可以随时连接 USB 外设或 B 器件，但是 OTG_HS 也只有在 VBUS 有效后（5V）才能检测到设备的连接。当 VBUS 处于有效电平且已连接远程 B 器件时，OTG_HS 模块将发出主机端口中断信号，该中断由主机端口控制和状态寄存器中的设备连接位（OTG_HS_HPRT 中的 PCDET 位）触发。

在 HNP 和 SRP 同时关闭的情况下，USB 设备或 B 器件将在连接后立即被检测到。OTG_HS 模块将发出主机端口中断信号，该中断由主机端口控制和状态寄存器中的设备连接位（OTG_HPRT 中的 PCDET 位）触发。

主机检测设备断开

设备断开事件将触发断开连接检测中断（OTG_GINTSTS 中的 DISCINT 位）。

主机枚举

检测到设备连接后，若又有新的设备连接进来，主机必须通过向新的设备发送 USB 复位和配置命令来启动枚举过程。

开始驱动 USB 复位前，应用程序必须等待去抖动完成位（OTG_GOTGINT 中的 DBCDNE 位）触发 OTG 中断，这表示由于在 DP (FS) 或 DM (LS) 上连接上拉电阻而发生电气抖动之后，总线恢复稳定状态。

应用程序通过将主机端口控制和状态寄存器中的端口复位位（OTG_HPRT 中的 PRST 位）置 1，并保持最少 10 ms，最多 20 ms，来在 USB 总线上发出 USB 复位信号（单端零）。应用程序计算这个过程的持续时间，然后将端口复位位清零。

USB 复位序列完成后，端口使能/禁止更改位（OTG_HPRT 中的 PENCHNG 位）立即触发主机端口中断，进而向应用程序发出通知，指示可从主机端口控制和状态寄存器中的端口速度字段（OTG_HPRT 中的 PSPD）读取枚举的设备速度，以及主机已经开始驱动 SOF (FS) 或 Keep-alive 令牌 (LS)。此时主机已就绪，可通过对设备发送命令来完成对设备的枚举。

主机挂起

应用程序通过将主机端口控制和状态寄存器中的端口挂起位（OTG_HPRT 中的 PSUSP）置 1 来挂起 USB 活动。OTG_HS 模块停止发送 SOF 并进入挂起状态。

可由远程设备的自主活动（远程唤醒）使总线退出挂起状态。这种情况下，远程唤醒信号将触发远程唤醒中断（OTG_GINTSTS 中的 WKUPINT 位），硬件把主机端口控制和状态寄存器中的端口恢复位（OTG_HPRT 中的 PRES 位）自动置位，并通过 USB 自动驱动恢复信号。应用程序必须为恢复窗口定时，然后将端口恢复位清零以退出挂起状态并重新发送 SOF。

如果由主机发起退出挂起状态，则应用程序必须将端口恢复位置 1 以启动主机端口上的恢复信号，为恢复窗口定时并最终将端口恢复位清零。

57.7.3 主机通道

OTG_HS 内核实现了 16 主机通道。每个主机通道均可用于 USB 主机传输（USB 管道）。主机最多能同时处理 16 个传输请求。如果应用程序有 8 个以上的传输请求挂起，则在通道从之前任务释放后（即，接收到传输完成和通道停止中断后），主机控制器驱动器 (HCD) 必须为未处理的传输请求重新对通道进行分配。

每个主机通道都可配置为支持输入/输出以及周期性/非周期性事务。每个主机通道都使用专用控制 (OTG_HCCHAR*x*) 寄存器、传输配置 (OTG_HCTSIZ*x*) 寄存器状态/中断 (OTG_HCINT*x*) 寄存器以及和其相关的中断屏蔽寄存器 (OTG_HCINTMSK*x*)。

主机通道控制

- 应用程序可通过主机通道 *x* 特性寄存器 (OTG_HCCHAR*x*) 对主机通道作以下控制：
  - 通道使能/禁止
  - 设置目标 USB 设备的 HS/FS/LS 速度
  - 设置目标 USB 设备的地址
  - 设置与该通道通信的目标 USB 设备上的端点的编号
  - 设置该通道上的传输方向：IN/OUT
  - 设置该通道上的 USB 传输的类型：控制/批量/中断/同步
  - 设置与该通道通信的设备端点的最大包长
  - 设置要进行周期传输的帧：奇帧/偶帧

主机通道传输

主机通道传输大小寄存器 (OTG_HCTSIZ*x*) 允许应用程序对传输大小参数进行编程并读取传输状态。必须在主机通道特性寄存器中的通道使能位置 1 之前完成对此寄存器的设置。使能端点后，数据包计数字段立即变为只读状态，同时 OTG_HS 模块根据当前传输状态对该字段进行更新。

- 可对以下传输参数进行编程：
  - 以字节为单位的传输大小
  - 构成整个传输大小的数据包个数
  - 初始数据 PID

主机通道状态/中断

主机通道 *x* 中断寄存器 (OTG_HCINT*x*) 指示通道在出现 USB 和 AHB 相关事件时的状态。当模块中断寄存器中的主机通道中断位（OTG_GINTSTS 中的 HCINT 位）置 1 时，应用程序必须读取这些寄存器以获得详细信息。在读取这些寄存器之前，应用程序必须先读取主机全体通道中断 (OTG_HAINT) 寄存器，以获取主机通道 x 中断寄存器的通道编号。应用程序必须将该寄存器中的相应位清零，才能将 OTG_HAINT 和 OTG_GINTSTS 寄存器中的对应位清零。OTG_HCINTMSKx 寄存器还提供每个通道各中断源的屏蔽位。

- 主机模块提供以下状态检查和中断产生功能：
  - 传输完成中断，指示应用程序 (AHB) 和 USB 端均已完成数据传输
  - 通道因传输完成、USB 事务错误或应用程序发出禁止命令而停止
  - 相关的发送 FIFO 为半空或全空状态（IN 端点）
  - 接收到 ACK 响应
  - 接收到 NAK 响应
  - 接收到 STALL 响应
  - 由于 CRC 校验失败、超时、位填充错误和错误的 EOP 导致 USB 事务错误
  - 串扰错误
  - 帧上溢
  - 数据同步错误

57.7.4 主机调度器

主机模块内置硬件调度器，可自主对应用程序发出的 USB 事务请求重新排序和管理。每一帧开始时，主机都先执行周期性（同步和中断）事务，然后执行非周期性（控制和批量）事务，以符合 USB 规范对同步和中断传输高优先级的保证。

主机通过请求队列（一个周期性请求队列和一个非周期请求队列）处理 USB 事务。每个请求队列最多可存储 8 个条目。每个条目代表一个应用程序发起但还未得到响应的 USB 事务请求，并存储了执行该 USB 事务所用到的 IN 或 OUT 通道的编号，以及其它相关信息。USB 事务请求在队列中的写入顺序决定了事务在 USB 接口上的执行顺序。

每一帧开始时，主机都先处理周期性请求队列，然后处理非周期性请求队列。如果当前帧结束时，计划在当前帧执行的同步或中断类型的 USB 传输事务请求仍处于挂起状态，则主机将发出未完成周期性传输中断（OTG_GINTSTS 中的 IPXFR 位）。OTG_FS/OTG_HS 模块全面负责对周期性和非周期性请求队列的管理。周期性发送 FIFO 和队列状态寄存器 (OTG_HPTXSTS) 与非周期性发送 FIFO 和队列状态寄存器 (OTG_HNPTXSTS) 都为只读寄存器，应用程序可使用它们来读取各请求队列的状态。其中包括：

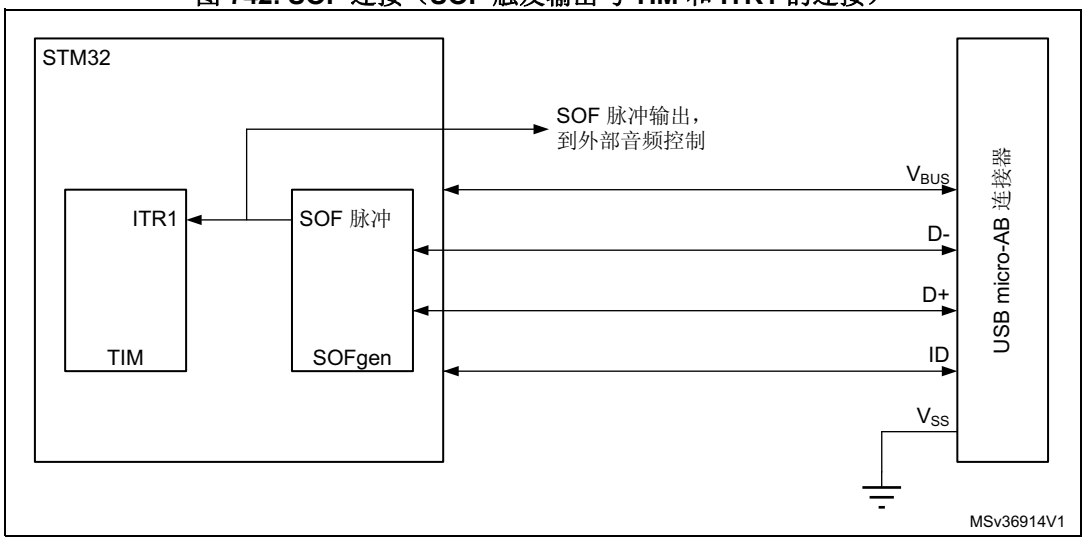
- 周期性（非周期性）请求队列中当前可用的空闲条目数（最多 8 个）
- 周期性（非周期性）Tx FIFO（OUT 事务）中当前可用的空闲空间
- IN/OUT 令牌、主机通道编号和其它状态信息

由于每个请求队列最多可存储 8 个 USB 事务请求，因此应用程序可以把主机 USB 事务请求提前发送给调度器；实际的通信最晚会在调度器处理完已挂起的 8 个周期事务和 8 个非周期事务完成之后出现在 USB 总线上。

要向主机调度器（队列）发出事务请求，应用程序必须读取 OTG_HNPTXSTS 寄存器中的 PTXQSAV 位或 OTG_HNPTXSTS 寄存器中的 NPTQXSAV 位，确保周期性（非周期性）请求队列中至少有一个可用空间来存储当前请求。

57.8 SOF 触发

图 742. SOF 连接（SOF 触发输出与 TIM 和 ITR1 的连接）



OTG_HS 模块在主机和设备模式下都可以监视、跟踪和配置 SOF 帧并且还具备 SOF 脉冲输出连接功能。

此功能尤其适用于自适应音频时钟生成，其中音频设备需要与 PC 提供的同步音频数据流实现同步，或者主机需要根据音频设备的要求调整数据帧率。

57.8.1 主机 SOF

主机模式下，可以在主机帧间隔寄存器 (HFIR) 中对所产生的两个连续 SOF (HS/FS) 或 Keep-alive (LS) 令牌期间所出现的 PHY 时钟数进行编程，进而应用程序可对 SOF 帧周期进行控制。帧开始（OTH_GINTSTS 中的 SOF 位）时都将生成中断。当前帧编号和出现下一个 SOF 前剩余的时间应用程序在主机帧编号寄存器 (HFNUM) 中能够进行跟踪。

SOF 令牌发出的同时会产生 SOF 脉冲信号，并且宽度为 12 个系统时钟周期。此外，SOF 脉冲信号还在内部与定时器的输入触发相连，因此可通过 SOF 脉冲触发输入捕获功能、输出比较功能和定时器。

57.8.2 设备 SOF

在设备模式下，USB 每次接收到 SOF 令牌时，都将触发帧开始中断（OTH_GINTSTS 中的 SOF 位）。相应的帧编号可从设备状态寄存器（OTG_DSTS 中的 FNSOF 位）读取。还可以生成宽度为 12 个系统时钟周期的 SOF 脉冲信号。此外，SOF 脉冲信号还在内部与 TIM 的输入触发相连，因此可通过 SOF 脉冲触发输入捕获功能、输出比较功能和定时器。

周期性帧结束中断 (OTG_GINTSTS/EOPF) 用于在经过了 80%、85%、90% 或 95% 的帧间隔时间时通知应用程序，具体取决于设备配置寄存器中的周期性帧间隔字段（OTG_DCFG 中的 PFIVL 位）。此功能可用于确定该帧的所有同步通信是否完成。

57.9 电源选项

OTG PHY 的功耗由通用模块配置寄存器中的两个或三个位控制，具体取决于 OTG 支持的版本。

- PHY 掉电 (OTG_GCCFG/PWRDWN)
  用于开启/关闭 PHY 的全速收发器模块。先置位后才允许后续的 USB 操作。
- V_{BUS} 检测使能 (OTG_GCCFG/VBDEN)
  用于开启/关闭与 OTG 操作关联的 V_{BUS} 感应比较器。

USB 会话没有开始或设备未连接时，可以在 USB 挂起状态下使用功率降低技术。

- 停止 PHY 时钟（OTG_PCGCCTL 中的 STPPCLK 位）
  将时钟门控控制寄存器中的停止 PHY 时钟位置 1 时，OTG 全速模块的大多数 48 MHz 内部时钟域均由时钟门控关闭。即使应用程序仍提供时钟输入，也会节省掉模块由于时钟信号翻转带来的动态功耗
  还会关掉收发器的大部分单元，只有负责检测异步恢复事件或远程唤醒事件的部分还保持工作状态。
- HCLK 门控（OTG_PCGCCTL 中的 GATEHCLK 位）
  将时钟门控控制寄存器中的 Gate HCLK 位置 1 时，OTG_HS 模块内部的大多数系统时钟域均由时钟门控关闭。只有寄存器读取和写入接口保持活动状态。即使应用程序仍提供时钟输入，也会节省掉模块由于时钟信号翻转带来的动态功耗。

- USB 系统停止

  当 OTG_HS 处于 USB 挂起状态时，应用程序可通过将 USB 系统中的所有时钟源全部关闭来显著降低总功耗。USB 系统停止可通过以下方式激活：首先将停止 PHY 时钟位置 1，然后在电源控制系统模块 (PWR) 中将系统配置为深度睡眠模式。

  OTG_HS 模块通过对 USB 上的远程唤醒（作为主机）或恢复（作为设备）信号进行异步检测，自动重新激活系统时钟和 USB 时钟。

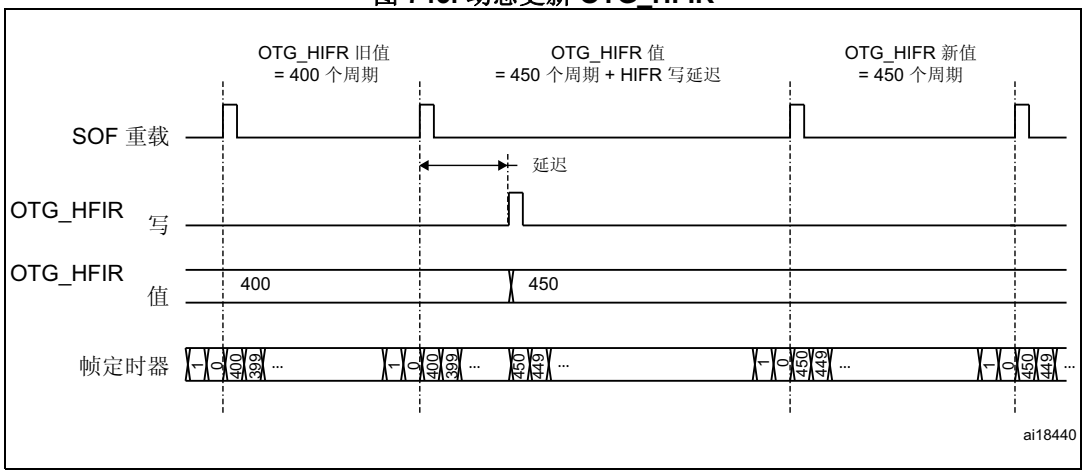
为了节省动态功耗，只在 USB 数据 FIFO 被 OTG_HS 模块访问时为其提供时钟。

57.10 动态更新 OTG_HFIR 寄存器

主机模式下，USB 模块具有对微帧周期进行动态微调的功能，能够将外部设备与 micro-SOF 帧进行同步。

如果 OTG_HS_HFIR 寄存器在当前 micro-SOF 帧内发生更改，则将在下一个帧中对 SOF 周期进行相应修正，具体说明请参见*图 743*。

图 743. 动态更新 OTG_HFIR

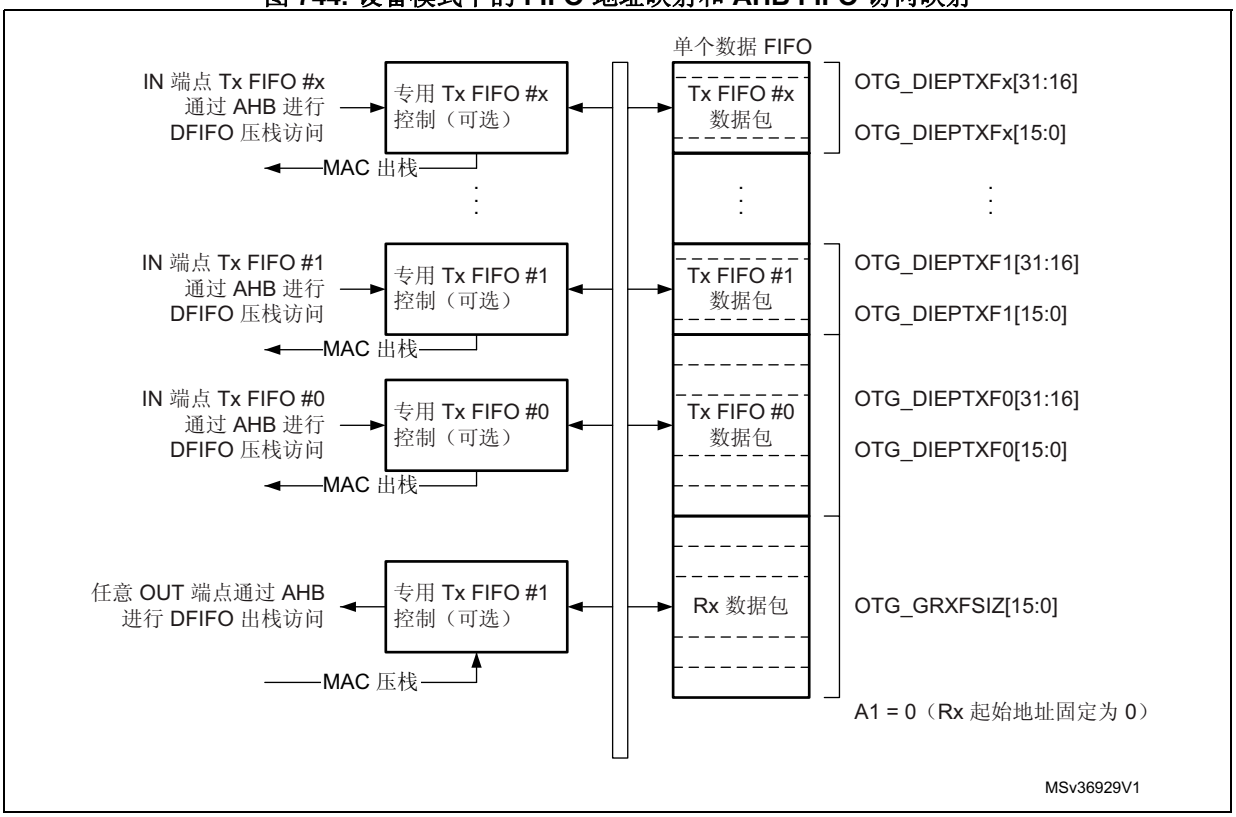


57.11 USB 数据 FIFO

USB 系统具有 4 KB 专用 RAM，采用复杂的 FIFO 控制机制。OTG_HS 模块中的数据包 FIFO 控制器模块将 RAM 空间划分为多个 Tx-FIFO（USB 传输前，应用程序将数据压入其中进行短暂存储）和单个 Rx FIFO（从 USB 接收到的数据被应用程序读取之前，在其中进行短暂存储）。RAM 中所构建的 FIFO 的数量与组织方式取决于设备的角色。设备模式下，为每个激活的 IN 端点配置一个 Tx FIFO。FIFO 的大小均由软件配置，以更好地满足应用要求。

57.11.1 设备 FIFO 架构

图 744. 设备模式下的 FIFO 地址映射和 AHB FIFO 访问映射



设备 Rx FIFO

OTG 设备使用单个接收 FIFO 接收发送到所有 OUT 端点的数据。只要 Rx FIFO 中有空余空间，收到的数据包就挨个填入 Rx FIFO。除了有效数据外，接收到的数据包状态（包含 OUT 端点目标编号、字节数、数据 PID 和对所接收数据的验证）也由模块进行存储。没有可用空间时，设备会回复主机事务 NACK 应答并在被寻址的端点上触发中断。接收 FIFO 的大小在接收 FIFO 大小寄存器 (OTG_GRXFSIZ) 中配置。

单个接收 FIFO 架构使得 USB 设备更高效地填充接收 RAM 缓冲区：

- 所有 OUT 端点共享同一个 RAM 缓冲区（共享 FIFO）
- OTG_HS 模块可将主机发出的任何 OUT 通信序列填充到接收 FIFO，直到没有多余空闲空间

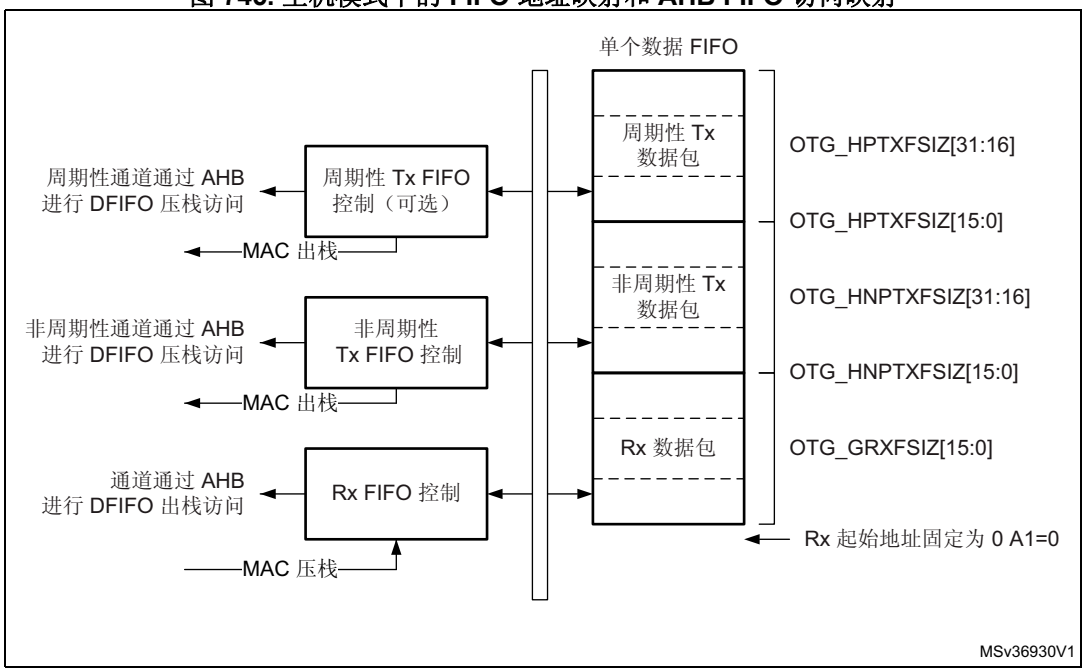
只要至少有一个数据包在 Rx FIFO 中可供读取，应用程序就会一直接收 Rx FIFO 非空中断（OTG_GINTSTS 中的 RXFLVL 位）。应用程序从接收状态读取和出栈寄存器 (OTG_GRXSTSP) 中读取数据包信息，最后通过读取与端点相关的出栈地址从接收 FIFO 读出相应数据。

设备 Tx FIFO

模块为各个 IN 端点提供了专用的 FIFO。应用程序通过端点 0 发送 FIFO 大小寄存器 (OTG_DIEPTXF0) 为 IN 端点 0 配置 FIFO 大小；通过设备 IN 端点发送 FIFOx 寄存器 (OTG_DIEPTXFx) 为 IN 端点 x 配置 FIFO 大小。

57.11.2 主机 FIFO 架构

图 745. 主机模式下的 FIFO 地址映射和 AHB FIFO 访问映射



主机 Rx FIFO

主机使用一个接收 FIFO 处理所有周期和非周期事务。FIFO 用作接收缓冲区以保存从 USB 接收到的数据（接收到的数据包的数据部分），直至这些数据传输到系统存储器。只要 FIFO 中有空间，来自设备 IN 端点的数据包就接收进来并挨个存储。接收到的每个数据包的状态（包含主机目标通道、字节数、数据 PID 和对所接收数据的校验）也存储在 FIFO 中。接收 FIFO 的大小在接收 FIFO 大小寄存器 (OTG_GRXFSIZ) 中配置。

单个接收 FIFO 架构使得 USB 主机高效地填充接收数据缓冲区：

- 所有 IN 配置主机通道共享同一个 RAM 缓冲区（共享 FIFO）
- OTG_HS 模块可将主机发出的任何 IN 通信序列带来的接收数据填充到接收 FIFO，直到没有多余空闲空间

只要至少有一个数据包在 Rx FIFO 中可供读取，应用程序就会接收 Rx FIFO 非空中断。应用程序从接收状态读取和出栈寄存器中读取数据包信息，最后从接收 FIFO 中读出数据。

主机 Tx FIFO

主机使用一个发送 FIFO 处理所有非周期（控制和批量）OUT 事务，使用另一个发送 FIFO 处理所有周期（同步和中断）OUT 事务。FIFO 用作发送缓冲区以保存要通过 USB 发送的数据（发送数据包）。周期（非周期）Tx FIFO 的大小在主机周期（非周期）发送 FIFO 大小 (OTG_HPTXFSIZ/OTG_HNPTXFSIZ) 寄存器中配置。

两个 Tx FIFO 按优先级实施操作，周期性通信的优先级较高，因此在 USB 一帧的时间内首先进行周期性通信。帧起始时，内置的主机调度器先处理周期请求队列，再处理非周期请求队列。

两个发送 FIFO 的架构使得 USB 主机能够对周期和非周期发送数据缓冲区分别进行优化管理：

- 配置为支持周期（非周期）OUT 事务的所有主机通道共享同一个 RAM 缓冲区（共享 FIFO）
- OTG_HS 模块可将主机发出的任何 OUT 通信填充到周期性（非周期性）发送 FIFO，直到没有多余空闲空间

只要周期性 Tx FIFO 为半空或全空，OTG_FS/OTG_HS 模块就会发出周期性 Tx FIFO 空中断（OTG_GINTSTS 中的 PTXFE 位），具体取决于 AHB 配置寄存器中的周期性 Tx FIFO 空门限位（OTG_GAHBCFG 中的 PTXFELVL 位）的值。只要周期性 Tx FIFO 和周期性请求队列中均存在空闲空间，应用程序便可提前写入发送数据。可通过读取主机周期性发送 FIFO 和队列状态寄存器 (OTG_HPTXSTS) 来了解二者的可用空间。

只要非周期性 Tx FIFO 为半空或全空，OTG_FS/OTG_HS 模块就会发出非周期性 Tx FIFO 空中断（OTG_GINTSTS 中的 NPTXFE 位），具体取决于 AHB 配置寄存器中的非周期性 Tx FIFO 空门限位（OTG_GAHBCFG 中的 TXFELVL 位）。只要非周期性 Tx FIFO 和非周期性请求队列中均存在空闲空间，应用程序便可写入发送数据。可通过读取主机非周期性发送 FIFO 和队列状态寄存器 (OTG_HNPTXSTS) 来了解二者的可用空间。

57.11.3 FIFO RAM 分配

设备模式

接收 FIFO RAM 分配：应用程序应为 SETUP 数据包分配 RAM：

- 接收 FIFO 中必须保留 10 个位置以在控制端点上接收 SETUP 数据包。OTG 模块不会向这些为 SETUP 数据包保留的位置写入任何其它数据。
- 将会为全局 OUT NAK 分配一个位置。
- 状态信息随各个接收数据包写入 FIFO。因此，必须至少为接收数据包分配（最大数据包大小 / 4）+ 1 的空间。如果使能了多个同步端点，则为接收连续数据包分配的空间必须至少为（最大数据包大小 / 4）的两倍 + 1。通常，推荐的空间为（最大数据包/4 + 1）的两倍，这样当上一个数据包向 CPU 传送时，USB 可同时接收后续的数据包。
- 传输完成状态信息和该端点收到的最后一个数据包会一起被推入 FIFO。推荐为每个 OUT 端点分配一个位置存储该端点上的传输状态信息。

器件 RxFIFO =

（4 * 控制端点数量 + 6）+（（所使用的最大 USB 数据包/4）+ 用于状态信息的 1）+（2 * OUT 端点数量）+ 用于全局 NAK 的 1

例如：周期性 USB 数据包的 MPS 是 1024 个字节，非周期性 USB 数据包的 MPS 是 512 个字节。有三个 OUT 端点、三个 IN 端点、一个控制端点和三个主机通道。

则器件 RxFIFO =（4 * 1 + 6）+（（1024/4）+1）+（2 * 4）+ 1 = 276

发送 FIFO RAM 分配：各个 IN 端点发送 FIFO 所需的最小 RAM 空间为该特定 IN 端点的最大数据包大小。

注： *为发送 IN 端点 FIFO 分配的空间越多，USB 的性能就越高。*

主机模式

接收 FIFO RAM 分配：

状态信息随各个接收数据包写入 FIFO。因此，必须至少为接收数据包分配（最大数据包大小 / 4）+ 1 的空间。如果使能了多个同步通道，则为接收连续数据包分配的空间必须至少为（最大数据包大小 / 4）的两倍 + 1。通常，推荐的空间为（最大数据包/4 + 1）的两倍，这样当上一个数据包向 CPU 传送时，USB 可同时接收后续的数据包。

传输完成状态信息和主机通道中的最后一个数据包会一起被推入 FIFO。因此，必须为此分配一个位置。

主机 RxFIFO =（（所用的最大 USB 数据包/4）+ 用于状态信息的 1）+ 用于传输完成的 1

例如：主机 RxFIFO =（（1024/4）+ 1）+ 1 = 258

发送 FIFO RAM 分配：

主机非周期性发送 FIFO 所需的最小 RAM 为所支持的所有非周期性 OUT 通道上传输的最大数据包的大小。

通常，推荐的空间为最大数据包大小的两倍，这样当 USB 正在发送当前数据包的同时，AHB 可以往发送 FIFO 填入下一个数据包。

非周期性 TxFIFO = 所用的最大非周期性 USB 数据包/4

例如：非周期性 TxFIFO =（512/4）= 128

主机周期性发送 FIFO 所需的最小 RAM 为所支持的所有周期性 OUT 通道上传输的最大数据包的大小。如果至少有一个同步 OUT 端点，则空间必须至少为该通道中最大数据包大小的两倍。

主机周期性 TxFIFO = 所用的最大周期性 USB 数据包/4

例如：主机周期性 TxFIFO =（1024/4）= 256

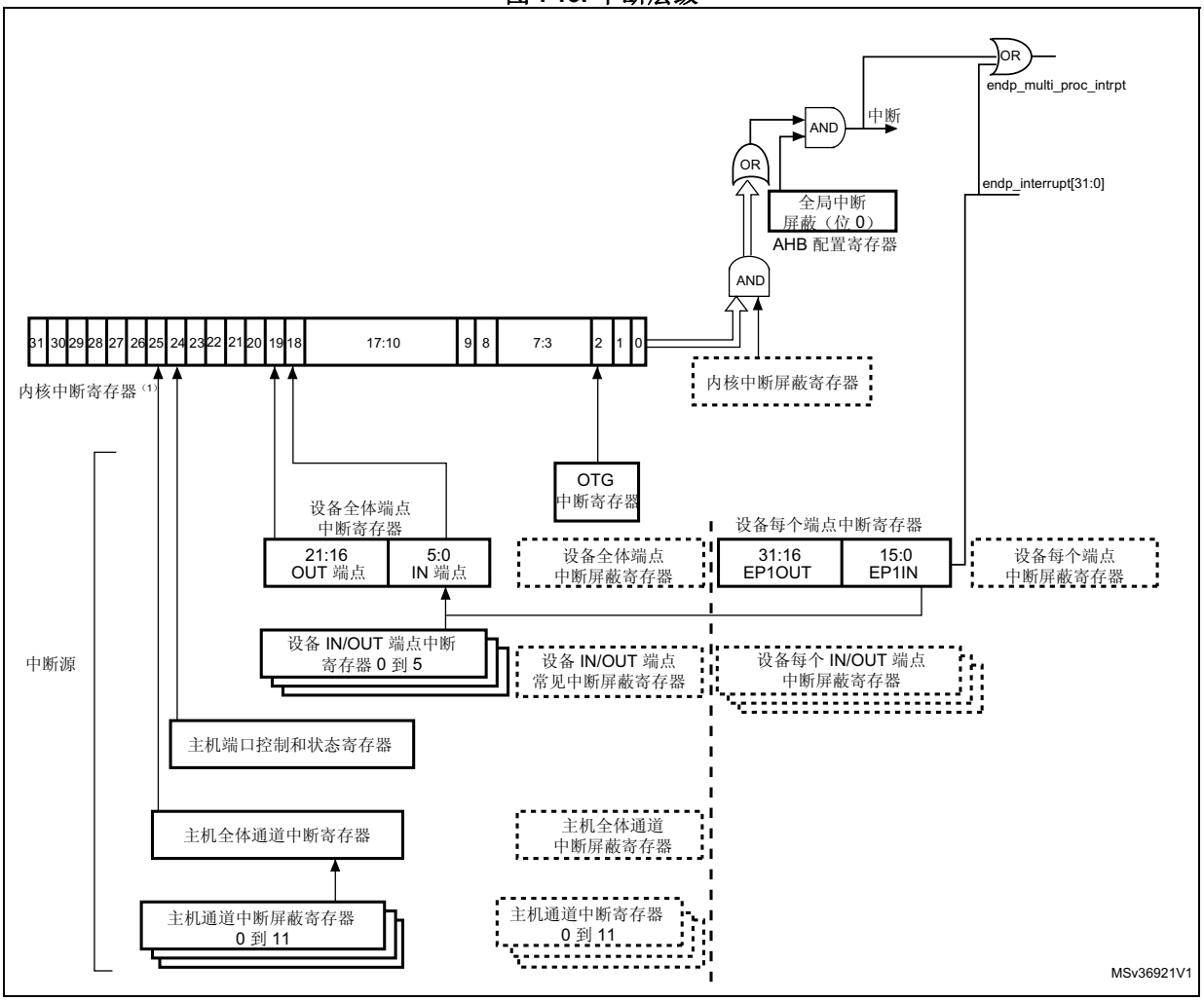
注： *为非周期性发送 FIFO 分配的空间越多，USB 的性能就越高。*

57.12 OTG_HS 中断

当 OTG_HS 控制器在一种模式下（设备模式或主机模式）工作时，应用程序不得以另一种角色模式访问寄存器。如果发生了非法访问，将会产生模式不匹配中断并在模块中断寄存器（OTG_GINTSTS 寄存器中的 MMIS 位）中反映。当模块从一种角色模式切换到另一种角色模式时，新工作模式下的寄存器必须重新编程为上电复位后的状态。

图 746 显示了中断层级。

图 746. 中断层级



1. 有关模块中断寄存器位，请参见*第 2404 页的OTG 模块中断寄存器 (OTG_GINTSTS)*。

57.13 OTG_HS 控制和状态寄存器

应用程序通过 AHB 从接口对控制和状态寄存器 (CSR) 进行读写操作，以此来控制 OTG_HS 模块。这些都是 32 位寄存器，其地址按 32 位对齐，因此只能以 32 位的方式访问。

CSR 分为以下几类：

- 模块全局寄存器
- 主机模式寄存器
- 主机全局寄存器
- 主机端口 CSR
- 主机通道相关寄存器
- 设备模式寄存器
- 设备全局寄存器
- 设备端点相关寄存器
- 电源和时钟门控寄存器
- 数据 FIFO (DFIFO) 访问寄存器

只有模块全局寄存器、电源和时钟门控寄存器、数据 FIFO 访问寄存器以及主机端口控制和状态寄存器，可在主机模式和设备模式下进行访问。当 OTG_HS 控制器在一种模式下（设备模式或主机模式）工作时，应用程序不得以另一种角色模式访问寄存器。如果发生了非法访问，将会产生模式不匹配中断并在模块中断寄存器（OTG_GINTSTS 寄存器中的 MMIS 位）中反映。当模块从一种角色模式切换到另一种角色模式时，新工作模式下的寄存器必须重新编程为上电复位后的状态。

57.13.1 CSR 存储器映射

主机模式寄存器和设备模式寄存器占据不同的地址。所有寄存器均在 AHB 时钟域内实现。

全局 CSR 地址映射

这些寄存器在主机模式和设备模式下均可用。

表 481. 模块全局控制和状态寄存器 (CSR)

缩略语	偏移地址	寄存器名称
OTG_GOTGCTL	0x000	*第 57.14.1 节：OTG 控制和状态寄存器 (OTG_GOTGCTL)*
OTG_GOTGINT	0x004	*第 57.14.2 节：OTG 中断寄存器 (OTG_GOTGINT)*
OTG_GAHBCFG	0x008	*第 57.14.3 节：OTG AHB 配置寄存器 (OTG_GAHBCFG)*
OTG_GUSBCFG	0x00C	*第 57.14.4 节：OTG USB 配置寄存器 (OTG_GUSBCFG)*
OTG_GRSTCTL	0x010	*第 57.14.5 节：OTG 复位寄存器 (OTG_GRSTCTL)*
OTG_GINTSTS	0x014	*第 57.14.6 节：OTG 模块中断寄存器 (OTG_GINTSTS)*
OTG_GINTMSK	0x018	*第 57.14.7 节：OTG 中断屏蔽寄存器 (OTG_GINTMSK)*
OTG_GRXSTSR	0x01C	*第 57.14.8 节：OTG_FS 接收状态调试读取/OTG 状态读取和出栈寄存器 (OTG_GRXSTSR/OTG_GRXSTSP)*
OTG_GRXSTSP	0x020	
OTG_GRXFSIZ	0x024	*第 57.14.9 节：OTG 接收 FIFO 大小寄存器 (OTG_GRXFSIZ)*

表 481. 模块全局控制和状态寄存器 (CSR)（续）

缩略语	偏移地址	寄存器名称
OTG_HNPTXFSIZ/ OTG_DIEPTXF0(1)	0x028	*第 57.14.10 节：OTG 主机非周期性发送 FIFO 大小寄存器 (OTG_HNPTXFSIZ)/ 端点 0 发送 FIFO 大小 (OTG_DIEPTXF0)*
OTG_HNPTXSTS	0x02C	*第 57.14.11 节：OTG 非周期性发送 FIFO/ 队列状态寄存器 (OTG_HNPTXSTS)*
OTG_GI2CCTL	0x030	*第 57.14.12 节：OTG I^2C 访问寄存器 (OTG_GI2CCTL)*
OTG_GCCFG	0x038	*第 57.14.13 节：OTG 通用模块配置寄存器 (OTG_GCCFG)*
OTG_CID	0x03C	*第 57.14.14 节：OTG 模块 ID 寄存器 (OTG_CID)*
OTG_GLPMCFG	0x54	*第 57.14.15 节：OTG 模块 LPM 配置寄存器 (OTG_GLPMCFG)*
OTG_HPTXFSIZ	0x100	*第 57.14.16 节：OTG 主机周期性发送 FIFO 大小寄存器 (OTG_HPTXFSIZ)*
OTG_DIEPTXFx	0x104 0x124 ... 0x1B4	*第 57.14.17 节：OTG 设备 IN 端点发送 FIFO 大小寄存器 (OTG_DIEPTXFx)（x = 1..8，其中 x 为 FIFO 编号）*

1. 通用规则是：主机模式下使用 OTG_HNPTXFSIZ；设备模式下使用 OTG_DIEPTXF0。

主机模式 CSR 地址映射

模块每次切换到主机模式时都必须对这些寄存器进行设置。

表 482. 主机模式控制和状态寄存器 (CSR)

缩略语	偏移地址	寄存器名称
OTG_HCFG	0x400	*第 57.14.19 节：OTG 主机配置寄存器 (OTG_HCFG)*
OTG_HFIR	0x404	*第 57.14.20 节：OTG 主机帧间隔寄存器 (OTG_HFIR)*
OTG_HFNUM	0x408	*第 57.14.21 节：OTG 主机帧编号/ 帧剩余时间寄存器 (OTG_HFNUM)*
OTG_HPTXSTS	0x410	*第 57.14.22 节：OTG_Host 周期性发送 FIFO/ 队列状态寄存器 (OTG_HPTXSTS)*
OTG_HAINT	0x414	*第 57.14.23 节：OTG 主机全体通道中断寄存器 (OTG_HAINT)*
OTG_HAINTMSK	0x418	*第 57.14.24 节：OTG 主机全体通道中断屏蔽寄存器 (OTG_HAINTMSK)*
OTG_HPRT	0x440	*第 57.14.25 节：OTG 主机端口控制和状态寄存器 (OTG_HPRT)*
OTG_HCCHARx	0x500 0x520 ... 0x6E0	*第 57.14.26 节：OTG 主机通道 x 特性寄存器 (OTG_HCCHARx)（x = 0..15，其中 x = 通道编号）*
OTG_HCSPLTx	0x504 0x524 .... 0x6E4	*第 57.14.27 节：OTG 主机通道 x 分离控制寄存器 (OTG_HCSPLTx)（x = 0..15，其中 x = 通道编号）*

表 482. 主机模式控制和状态寄存器 (CSR)（续）

缩略语	偏移地址	寄存器名称
OTG_HCDMAx	0x514 0x534 .... 0x6F4	*第 57.14.31 节：OTG 主机通道 x DMA 地址寄存器 (OTG_HCDMAx)（x = 0..15，其中 x = 通道编号）*
OTG_HCINTx	0x508 0x528 .... 0x6E8	*第 57.14.28 节：OTG 主机通道 x 中断寄存器 (OTG_HCINTx)（x = 0..15，其中 x = 通道编号）*
OTG_HCINTMSKx	0x50C 0x52C 0x6EC	*第 57.14.29 节：OTG 主机通道 x 中断屏蔽寄存器 (OTG_HCINTMSKx)（x = 0..15，其中 x = 通道编号）*
OTG_HCTSIZx	0x510 0x530 .... 0x6F0	*第 57.14.30 节：OTG 主机通道 x 传输大小寄存器 (OTG_HCTSIZx)（x = 0..15，其中 x = 通道编号）*

设备模式 CSR 地址映射

模块每次切换到设备模式时都必须对这些寄存器进行编程。

表 483. 设备模式控制和状态寄存器

缩略语	偏移地址	寄存器名称
OTG_DCFG	0x800	*第 57.14.33 节：OTG 设备配置寄存器 (OTG_DCFG)*
OTG_DCTL	0x804	*第 57.14.34 节：OTG 设备控制寄存器 (OTG_DCTL)*
OTG_DSTS	0x808	*第 57.14.35 节：OTG 设备状态寄存器 (OTG_DSTS)*
OTG_DIEPMSK	0x810	*第 57.14.36 节：OTG 设备 IN 端点通用中断屏蔽寄存器 (OTG_DIEPMSK)*
OTG_DOEPMSK	0x814	*第 57.14.37 节：OTG 设备 OUT 端点通用中断屏蔽寄存器 (OTG_DOEPMSK)*
OTG_DAINT	0x818	*第 57.14.38 节：OTG 设备全体端点中断寄存器 (OTG_HAINT)*
OTG_DAINTMSK	0x81C	*第 57.14.39 节：OTG 全体端点中断屏蔽寄存器 (OTG_DAINTMSK)*
OTG_DVBUSDIS	0x828	*第 57.14.40 节：OTG 设备 V_{BUS} 放电时间寄存器 (OTG_DVBUSDIS)*
OTG_DVBUSPULSE	0x82C	*第 57.14.41 节：OTG 设备 V_{BUS} 脉冲时间寄存器 (OTG_DVBUSPULSE)*
OTG_DTHRCTL	0x0830	*第 57.14.42 节：OTG 设备阈值控制寄存器 (OTG_DTHRCTL)*
OTG_DIEPEMPMSK	0x834	*第 57.14.43 节：OTG 设备 IN 端点 FIFO 空中断屏蔽寄存器 (OTG_DIEPEMPMSK)*

表 483. 设备模式控制和状态寄存器（续）

缩略语	偏移地址	寄存器名称
OTG_DEACHINT	0x838	*第 57.14.44 节：OTG 设备单个端点中断寄存器 (OTG_DEACHINT)*
OTG_DEACHINTMSK	0x83C	*第 57.14.45 节：OTG 设备单个端点中断屏蔽寄存器 (OTG_DEACHINTMSK)*
OTG_DIEPCTLx	0x900 0x920 ... 0x9E0	*第 57.14.46 节：OTG 设备端点 x 控制寄存器 (OTG_DIEPCTLx)（x = 0..8，其中 x = 端点编号）*
OTG_DIEPINTx	0x908 0x928 ... 0x9E8	*第 57.14.49 节：OTG 设备端点 x 中断寄存器 (OTG_DIEPINTx)（x = 0..8，其中 x = 端点编号）*
OTG_DIEPTSIZ0	0x910	*第 57.14.51 节：OTG 设备 IN 端点 0 传输大小寄存器 (OTG_DIEPTSIZ0)*
OTG_DIEPDMAx	0x914	*第 57.14.52 节：OTG 设备通道 x DMA 地址寄存器 (OTG_DIEPDMAx)（x = 0..15，其中 x= 通道编号）*
OTG_DTXFSTSx	0x918 0x938 ..... 0x9F8	*第 57.14.56 节：OTG 设备 IN 端点发送 FIFO 状态寄存器 (OTG_DTXFSTSx)（x = 0..8，其中 x = 端点编号）*
OTG_DIEPTSIZx	0x930 0x950 ... 0x9F0	*第 57.14.55 节：OTG 设备 IN 端点 x 传输大小寄存器 (OTG_DIEPTSIZx)（x = 1..8，其中 x= 端点编号）*
OTG_DOEPCTL0	0xB00	*第 57.14.47 节：OTG 设备控制 OUT 端点 0 控制寄存器 (OTG_DOEPCTL0)*
OTG_DOEPDMAx	0xB14	*第 57.14.54 节：OTG 设备通道 x DMA 地址寄存器 (OTG_DOEPDMAx)（x = 0..15，其中 x= 通道编号）*
OTG_DOEPCTLx	0xB20 0xB40 ... 0xBE0	*第 57.14.48 节：OTG 设备端点 x 控制寄存器 (OTG_DOEPCTLx)（x = 1..8，其中 x = 端点编号）*
OTG_DOEPINTx	0xB08 0XB28 ... 0xBE8	*第 57.14.50 节：OTG 设备端点 x 中断寄存器 (OTG_DIEPINTx)（x = 0..8，其中 x = 端点编号）*

表 483. 设备模式控制和状态寄存器（续）

缩略语	偏移地址	寄存器名称
OTG_DOEPTSIZ0	0xB10	*第 57.14.53 节：OTG 设备 OUT 端点 0 传输大小寄存器 (OTG_DOEPTSIZ0)*
OTG_DOEPTSIZx	0xB30 0xB50 .. 0xBF0	*第 57.14.57 节：OTG 设备 OUT 端点 x 传输大小寄存器 (OTG_DOEPTSIZx)（x = 1..8，其中 x = 端点编号）*

数据 FIFO (DFIFO) 访问寄存器地址映射

这些寄存器在主机模式和设备模式下均可用，用于对给定方向的特定端点或通道的 FIFO 空间进行读写操作。如果主机通道类型为 IN，则只能对该通道上的 FIFO 进行读操作。同样地，如果主机通道类型为 OUT，则只能对该通道上的 FIFO 进行写操作。

表 484. 数据 FIFO (DFIFO) 访问寄存器地址映射

FIFO 访问寄存器段	地址范围	访问
设备 IN 端点 0/主机 OUT 通道 0：DFIFO 写访问 设备 OUT 端点 0/主机 IN 通道 0：DFIFO 读访问	0x1000–0x1FFC	w r
设备 IN 端点 1/主机 OUT 通道 1：DFIFO 写访问 设备 OUT 端点 1/主机 IN 通道 1：DFIFO 读访问	0x2000–0x2FFC	w r
...	...	...
设备 IN 端点 x(1)/主机 OUT 通道 x(1)：DFIFO 写访问 设备 OUT 端点 x(1)/主机 IN 通道 x(1)：DFIFO 读访问	0xX000–0xXFFC	w r

1. 其中，x 为 8（器件模式）或 15（主机模式）。

电源和时钟门控 CSR 地址映射

由一个单独寄存器对电源和时钟门控进行管理。在主机模式和设备模式下均可使用。

表 485. 电源和时钟门控控制和状态寄存器

寄存器名称	缩略语	偏移地址：0xE00–0xFFF
电源和时钟门控控制寄存器	PCGCCTL	0xE00-0xE04
保留	-	0xE05–0xFFF

57.14 OTG_HS 寄存器

这些寄存器在主机模式和设备模式下都可用，且在这两个模式间切换时无需对其进行重新编程。

除非特别说明，否则寄存器描述中的位值以二进制表示。

57.14.1 OTG 控制和状态寄存器 (OTG_GOTGCTL)

OTG control and status register

偏移地址：0x000

复位值：0x0X01 0000

OTG_GOTGCTL 寄存器控制模块的 OTG 功能并反映其状态。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OTG VER	BSVLD	ASVLD	DBCT	CID STS
											rw	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	EHEN	DHNP EN	HSHNP EN	HNP RQ	HNG SCS	BVALO VAL	BVALO EN	AVALO VAL	AVALO EN	VBVAL OVAL	VBVAL OEN	SRQ	SRQ SCS
			rw	rw	rw	rw	r	rw	rw	rw	rw	rw	rw	rw	r

位 31:21 保留，必须保持复位值。

位 20 **OTGVER**：OTG 版本 (OTG version)

选择 OTG 版本。

0：OTG 版本 1.3。OTG1.3 在新产品开发中遭到淘汰。

1：OTG 版本 2.0。在此版本中，模块仅支持通过数据线脉冲实现 SRP。

位 19 **BSVLD**：B 会话有效 (B-session valid)

指示设备模式下收发器的状态。

0：B 会话无效。

1：B 会话有效。

在 OTG 模式下，该位可用于确定设备处于连接状态还是断开状态。

注： 仅可在设备模式下访问。

位 18 **ASVLD**：A 会话有效 (A-session valid)

指示主机模式下收发器的状态。

0：A 会话无效

1：A 会话有效

注： 仅可在主机模式下访问。

位 17 **DBCT**：长/短去抖动时间 (Long/short debounce time)

指示已检测到连接的去抖动时间。

0：长去抖动时间，用于物理连接 (100 ms + 2.5 µs)

1：短去抖动时间，用于软连接 (2.5 µs)

注： 仅可在主机模式下访问。

位 16 **CIDSTS**：连接器 ID 状态 (Connector ID status)

指示发生连接事件时的连接器 ID 状态。

0：OTG_HS 控制器处于 A 器件模式

1：OTG_HS 控制器处于 B 器件模式

注： 在设备模式和主机模式均可访问。

位 15:13 保留，必须保持复位值。

位 12 **EHEN**：嵌入式主机使能 (Embedded host enable)

用于在 OTG A 器件状态机和嵌入式主机状态机之间选择。

0：选择 OTG A 器件状态机

1：选择嵌入式主机状态机

位 11 **DHNPEN**：使能设备 HNP 特性 (Device HNP enabled)

从所连 USB 主机处成功接收到 SetFeature.SetHNPEnable 命令时，应用程序将该位置 1。

0：应用不使能 HNP

1：应用使能 HNP

注： 仅可在设备模式下访问。

位 10 **HSHNPEN**：主机设置 HNP 使能 (Host set HNP enable)

（使用 SetFeature.SetHNPEnable 命令）在所连接设备上成功使能 HNP 后，应用程序将该位置 1。

0：主机未对设备设置了 HNP 使能

1：主机已对设备设置了 HNP 使能

注： 仅可在主机模式下访问。

位 9 **HNPRQ**：HNP 请求 (HNP request)

应用程序将该位置 1 时，将对所连接 USB 主机发起 HNP 请求。当 OTG_GOTGINT 寄存器中的主机协商成功状态更改位（OTG_GOTGINT 中的 HNSSCHG 位）置 1 时，应用程序可通过写 0 将该位清零。HNSSCHG 位清零时，模块会将该位清零。

0：无 HNP 请求

1：HNP 请求

注： 仅可在设备模式下访问。

位 8 **HNGSCS**：主机协商成功 (Host negotiation success)

当主机协商成功时，模块会将该位置 1。当该寄存器中的 HNP 请求位 (HNPRQ) 置 1 时，模块会将该位清零。

0：主机协商失败

1：主机协商成功

注： 仅可在设备模式下访问。

位 7 **BVALOVAL**：B 器件会话有效覆盖值 (B-peripheral session valid override value)

此位用于在 BVALOEN 位置 1 时设置 Bvalid 信号的覆盖值。

0：BVALOEN = 1 时，Bvalid 值为“0”

1：BVALOEN = 1 时，Bvalid 值为“1”

注： 仅可在设备模式下访问。

位 6 **BVALOEN**：B 器件会话有效覆盖使能 (B-peripheral session valid override enable)

该位用于使能/禁止软件通过 BVALOVAL 位来覆盖 Bvalid 信号。

0：禁止覆盖，模块使用来自所选 PHY 的 Bvalid 信号

1：从 PHY 内部接收的 Bvalid 由 BVALOVAL 位的值覆盖

注： 仅可在设备模式下访问。

位 5 **AVALOVAL**：A 器件会话有效覆盖值 (A-peripheral session valid override value)

此位用于在 AVALOEN 位置 1 时设置 Avalid 信号的覆盖值。

0：AVALOEN = 1 时，Avalid 值为“0”

1：AVALOEN = 1 时，Avalid 值为“1”

注： 仅可在主机模式下访问。

位 4 **AVALOEN**：A 器件会话有效覆盖使能 (A-peripheral session valid override enable)

该位用于使能/禁止软件通过 AVALOVAL 位来覆盖 Avalid 信号。

0：禁止覆盖，模块使用来自所选 PHY 的 Avalid 信号

1：从 PHY 内部接收的 Avalid 由 AVALOVAL 位的值覆盖

注： 仅可在主机模式下访问。

位 3 **VBVALOVAL**：V_{BUS} 有效覆盖值 (V_{BUS} valid override value)

此位用于在 VBVALOEN 位置 1 时设置 vbusvalid 信号的覆盖值。

0：VBVALOEN = 1 时，vbusvalid 值为“0”

1：VBVALOEN = 1 时，vbusvalid 值为“1”

注： 仅可在主机模式下访问。

位 2 **VBVALOEN**：V_{BUS} 有效覆盖使能 (V_{BUS} valid override enable)

该位用于使能/禁止软件通过 VBVALOVAL 位来覆盖 vbusvalid 信号。

0：禁止覆盖，模块使用来自所选 PHY 的 vbusvalid 信号

1：从 PHY 内部接收的 vbusvalid 由 VBVALOVAL 位的值覆盖

注： 仅可在主机模式下访问。

位 1 **SRQ**：会话请求 (Session request)

应用程序将该位置 1 时，将在 USB 上发起会话请求。当 OTG_GOTGINT 寄存器中的主机协商成功状态更改位（OTG_GOTGINT 中的 HNSSCHG 位）置 1 时，应用程序可通过写 0 将该位清零。HNSSCHG 位清零时，模块会将该位清零。

如要使用 USB 1.1 全速串行收发器接口来启动会话请求，则应用程序必须在该寄存器中的 B 会话有效位（OTG_GOTGCTL 中的 BSVLD 位）清零后，等待 V_{BUS} 放电至 0.2 V。该放电时间因 PHY 不同而异，可从 PHY 供应商处了解相关信息。

0：无会话请求

1：会话请求

注： 仅可在设备模式下访问。

位 0 **SRQSCS**：会话请求成功 (Session request success)

当会话请求成功时，模块会将该位置 1。

0：会话请求失败

1：会话请求成功

注： 仅可在设备模式下访问。

57.14.2 OTG 中断寄存器 (OTG_GOTGINT)

OTG interrupt register

偏移地址：0x04

复位值：0x0000 0000

应用程序会在出现 OTG 中断时读取该寄存器，并通过将该寄存器中的位清零来清除 OTG 中断。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	IDCHNG	DBCDNE	ADTOCHG	HNGDET	Res.
											rc_w1	rc_w1	rc_w1	rc_w1	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	HNSSCHG	SRSSCHG	Res.	Res.	Res.	Res.	Res.	SEDET	Res.	Res.
						rc_w1	rc_w1						rc_w1		

位 31:21 保留，必须保持复位值。

位 20 IDCHNG：
该位置 1 时，表示 ID 输入引脚的值发生了变化。

位 19 **DBCDNE**：去抖动完成 (Debounce done)
模块会在设备连接后且去抖动结束时将该位置 1。应用程序会在看见该中断后，开始驱动 USB 复位。该位仅在 OTG_GUSBCFG 寄存器中的 HNP 功能位或 SRP 功能位（分别为 OTG_GUSBCFG 中的 HNPCAP 位或 SRPCAP 位）置 1 时有效。
注： 仅可在主机模式下访问。

位 18 **ADTOCHG**：A 器件超时更改 (A-device timeout change)
模块将该位置 1 时，指示 A 器件在等待 B 器件连接时超时。
注： 在设备模式和主机模式均可访问。

位 17 **HNGDET**：检测到主机协商 (Host negotiation detected)
当检测到 USB 上的主机协商请求时，模块会将该位置 1。
注： 在设备模式和主机模式均可访问。

位 16:10 保留，必须保持复位值。

位 9 **HNSSCHG**：主机协商成功状态更改 (Host negotiation success status change)
模块将在 USB 主机协商请求成功或失败时将此位置 1。应用程序必须读取 OTG_GOTGCTL 寄存器中的主机协商成功位（OTG_GOTGCTL 中的 HNGSCS 位）来检查请求成功还是失败。
注： 在设备模式和主机模式均可访问。

位 8 **SRSSCHG**：会话请求成功状态更改 (Session request success status change)
模块将在会话请求成功或失败时将此位置 1。应用程序必须读取 OTG_GOTGCTL 寄存器中的会话请求成功位（OTG_GOTGCTL 中的 SRQSCS 位）来检查请求成功还是失败。
注： 在设备模式和主机模式均可访问。

位 7:3 保留，必须保持复位值。

位 2 **SEDET**：检测到会话结束 (Session end detected)
模块将该位置 1 时，指示 V_{BUS} < 0.8 V 时，V_{BUS} 上的电压不再适用于 B 器件会话。
注： 在设备模式和主机模式均可访问。

位 1:0 保留，必须保持复位值。

57.14.3 OTG AHB 配置寄存器 (OTG_GAHBCFG)

OTG AHB configuration register

偏移地址：0x008

复位值：0x0000 0000

该寄存器可用于在上电后或更改角色模式时对模块进行配置。该寄存器主要包含 AHB 系统相关的配置参数。请勿在初始编程后更改该寄存器。应用程序必须在开始任何 AHB 或 USB 事务前对该寄存器进行编程。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	PTXFE LVL	TXFE LVL	Res.	DMAEN	HBSTLEN				GINT MSK
							rw	rw		rw	rw	rw	rw	rw	rw

位 31:9 保留，必须保持复位值。

位 8 **PTXFELVL**：周期性 Tx FIFO 空门限 (Periodic Tx FIFO empty level)

指示 OTG_GINTSTS 寄存器中的周期性 Tx FIFO 空中断位（OTG_GINTSTS 中的 PTXFE 位）何时触发。

0：PTXFE（位于 OTG_GINTSTS）中断表示周期性 Tx FIFO 为半空状态

1：PTXFE（位于 OTG_GINTSTS）中断表示周期性 Tx FIFO 为全空状态

注： 仅可在主机模式下访问。

位 7 **TXFELVL**：Tx FIFO 空门限 (Tx FIFO empty level)

在设备模式下，该位指示 IN 端点发送 FIFO 空中断（OTG_DIEPINTx 中的 TXFE）何时触发：

0：TXFE（位于 OTG_DIEPINTx）中断表示 IN 端点 Tx FIFO 为半空状态

1：TXFE（位于 OTG_DIEPINTx）中断表示 IN 端点 Tx FIFO 为全空状态

在主机模式下，该位指示非周期性 Tx FIFO 空中断（OTG_GINTSTS 中的 NPTXFE 位）何时触发。

0：NPTXFE（位于 OTG_GINTSTS）中断表示非周期性 Tx FIFO 为半空状态

1：NPTXFE（位于 OTG_GINTSTS）中断表示非周期性 Tx FIFO 为全空状态

位 6 保留，必须保持复位值

位 5 **DMAEN**：DMA 使能 (DMA enabled)

0：模块不使用 DMA 进行数据收发

1：模块使用 DMA 进行数据收发

位 4:1 **HBSTLEN**：突发长度/类型 (Burst length/type)

0000 单次：总线事务使用单次 32 位访问（不建议）

0001 INCR：总线事务使用未指定的长度访问（不建议使用 INCR AHB 总线命令）

0011 INCR4：总线事务指定 4x 32 位访问

0101 INCR8：总线事务指定 8x 32 位访问

0111 INCR16：总线事务基于 16x 32 位访问

其它值：保留

位 0 **GINTMSK**：全局中断屏蔽 (Global interrupt mask)

该位用于屏蔽全局中断或使能全局中断。中断状态寄存器由模块进行更新，与此位的设置无关。

0：对应用程序屏蔽中断。

1：不对应用程序屏蔽中断。

注： 在设备模式和主机模式均可访问。

57.14.4 OTG USB 配置寄存器 (OTG_GUSBCFG)

OTG USB configuration register

偏移地址：0x00C

复位值：0x0000 1400

该寄存器可用于在上电或更改角色模式后对模块进行配置。其中包含与 USB 和 USB-PHY 相关的配置参数。应用程序必须在开始任何 AHB 或 USB 事务前对该寄存器进行编程。请勿在初始编程后更改该寄存器。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	FD MOD	FH MOD	Res.	Res.	Res.	ULPI IPD	PTCI	PCCI	TSDPS	ULPIE VBUSI	ULPIE VBUSD	ULPI CSM	ULPI AR	ULPI FSL	Res.
	rw	rw				rw	rw	rw	rw	rw	rw	rw	rw	rw	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PHYL PC	Res.	TRDT				HNP CAP	SRP CAP	Res.	PHY SEL	Res.	Res.	Res.	TOCAL		
rw		rw				rw	rw		rw				rw		

位 31 保留，必须保持复位值。

位 30 **FDMOD**：强制设备模式 (Force device mode)

向该位写入 1 时，可将模块强制为设备模式，而无需考虑 OTG_ID 输入引脚。

0：正常模式

1：强制设备模式

将强制位置 1 后，应用程序必须等待至少 25 ms 后更改方可生效。

注： 在设备模式和主机模式均可访问。

位 29 **FHMOD**：强制主机模式 (Force host mode)

向该位写入 1 时，可将模块强制为主机模式，而无需考虑 OTG_ID 输入引脚。

0：正常模式

1：强制主机模式

将强制位置 1 后，应用程序必须等待至少 25 ms 后更改方可生效。

注： 在设备模式和主机模式均可访问。

位 28:26 保留，对于 USB OTG HS 和 FS，必须保持复位值。

位 25 **ULPIIPD**：ULPI 接口保护禁止 (ULPI interface protect disable)

该位控制 PHY 中内置的电路，以保护 ULPI 接口。该功能所采用的任何上拉或下拉电阻均可禁止。有关详细信息，请参见 ULPI 规范。

0：使能接口保护电路

1：禁止接口保护电路

位 24 **PTCI**：通过指示符 (Indicator pass through)

该位将控制互补输出是否需要通过内部 V_{BUS} 有效电平比较器的验证，之后方可用于 RX CMD 中的 V_{BUS} 状态。有关详细信息，请参见 ULPI 规范。

0：互补输出信号通过内部 V_{BUS} 有效电平比较器的验证

1：互补输出信号不通过内部 V_{BUS} 有效电平比较器的验证

位 23 **PCCI**：互补指示符 (Indicator complement)

该位控制 PHY 将 ExternalVbusIndicator 输入信号反转并生成互补输出。有关详细信息，请参见 ULPI 规范。

0：PHY 不将 ExternalVbusIndicator 信号反转

1：PHY 将 ExternalVbusIndicator 信号反转

位 22 **TSDPS**：TermSel DLine 脉冲选择 (TermSel DLine pulsing selection)

该位选择 utmi_termselect 来驱动 SRP（会话请求协议）期间的数据线脉冲。

0：使用 utmi_txvalid 驱动数据线脉冲（默认）

1：使用 utmi_termsel 驱动数据线脉冲

位 21 **ULPIEVBUSI**：ULPI 外部 V_{BUS} 指示符 (ULPI external VBUS indicator)

该位指示 ULPI PHY 使用外部 V_{BUS} 过流指示。

0：PHY 使用内部 V_{BUS} 有效比较器

1：PHY 使用外部 V_{BUS} 有效比较器

位 20 **ULPIEVBUSD**：ULPI 外部 V_{BUS} 驱动 (ULPI External V_{BUS} Drive)

该位选择 ULPI PHY 使用内部或外部电源来驱动 V_{BUS} 上的 5 V 电压。

0：PHY 使用内部电荷泵驱动 V_{BUS}（默认）

1：PHY 使用外部电源驱动 V_{BUS}。

位 19 **ULPICSM**：ULPI 时钟 SuspendM (ULPI Clock SuspendM)

该位置位 ULPI PHY 接口控制寄存器中的 ClockSuspendM 位。该位仅适用于串行和 carkit 模式。

0：PHY 在挂起期间断开内部时钟的电源

1：PHY 不断开内部时钟的电源

位 18 **ULPIAR**：ULPI 自动恢复 (ULPI Auto-resume)

该位置位 ULPI PHY 接口控制寄存器中的 AutoResume 位。

0：PHY 不使用自动恢复功能

1：PHY 使用自动恢复功能

位 17 **ULPIFSLS**：ULPI FS/LS 选择 (ULPI FS/LS select)

应用程序使用该位为 ULPI PHY 选择 FS/LS 串行接口。该位仅在 ULPI PHY 上选择了 FS 串行收发器的情况下有效。

0：ULPI 接口

1：ULPI FS/LS 串行接口

位 16 保留，必须保持复位值 。

位 15 **PHYLPCS**：PHY 低功耗时钟选择 (PHY Low-power clock select)

该位用于选择 480 MHz 或 48 MHz（低功耗）PHY 模式。在 FS 和 LS 模式下，PHY 通常可使用 48 MHz 时钟运行，从而节约功耗。

0：480 MHz 内部 PLL 时钟

1：48 MHz 外部时钟

在 480 MHz 模式下，UTMI 接口以 60 MHz 或 30MHz 运行，具体取决于选择的是 8 位还是 16 位数据宽度。在 48 MHz 模式下，UTMI 接口在 FS 和 LS 模式下以 48 MHz 运行。

位 14 保留，必须保持复位值。

位 13:10 **TRDT**：USB 周转时间 (USB turnaround time)

这些位允许以 PHY 时钟数为单位设置周转时间。必须根据*表 487：TRDT 值 (HS)* 来配置这些位，具体取决于应用程序 AHB 频率。TRDT 值越高，USB对 IN 令牌的响应时间就越长，从而可以弥补 AHB 对数据 FIFO 的较长读访问延迟。

注： 仅可在设备模式下访问。

位 9 **HNPCAP**：HNP 功能 (HNP-capable)

应用程序使用该位控制 OTG_HS 控制器的 HNP 功能。

0：不使能 HNP 功能。

1：使能 HNP 功能。

注： 在设备模式和主机模式均可访问。

位 8 **SRPCAP**：SRP 功能 (SRP-capable)

应用程序使用该位控制 OTG_HS 控制器的 SRP 功能。如果模块作为无 SRP 功能的 B 器件，则无法请求连接的 A 器件（主机）激活 V_{BUS} 并开始会话。

0：不使能 SRP 功能。

1：使能 SRP 功能。

注： 在设备模式和主机模式均可访问。

位 7 保留，必须保持复位值。

位 6 **PHYSEL**：全速串行收发器选择 (Full Speed serial transceiver select)

0：USB 2.0 外部 ULPI 高速 PHY。

1：USB 1.1 全速串行收发器。

位 5 保留，必须保持复位值。

位 4 保留，必须保持复位值

位 3 保留，如果存在 UTMI 接口，必须保持复位值。

位 2:0 **TOCAL**：FS 超时校准 (FS timeout calibration)

PHY 引入的额外延迟包括应用程序在该字段中设置的 PHY 时钟数，以及模块的全速数据包间超时间隔。不同 PHY 引入的延迟对数据线状态的影响是不同的。

全速操作的 USB 标准超时值为 16 到 18（含）个位时间。应用程序必须根据枚举速度编程该字段。每个 PHY 时钟增加的位时间数为 0.25 个位时间。

表 486. TRDT 值

AHB 频率范围 (MHz)		TRDT 最小值
最小值	最大值	
14.2	15	0xF
15	16	0xE
16	17.2	0xD
17.2	18.5	0xC
18.5	20	0xB
20	21.8	0xA
21.8	24	0x9
24	27.5	0x8
27.5	32	0x7
32	-	0x6

表 487. TRDT 值 (HS)

AHB 频率范围 (MHz)		TRDT 最小值
最小值	最大值	
30	-	0x9

57.14.5 OTG 复位寄存器 (OTG_GRSTCTL)

OTG reset register

偏移地址：0x10

复位值为：0x8000 0000

应用程序通过此寄存器复位模块中的各项硬件特性。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
AHB IDL	DMAREQ	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
r	r														

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	TXFNUM					TXFFLSH	RXFFLSH	Res.	Res.	PSRST	CSRST
					rw					rs	rs			rs	rs

位 31 **AHBIDL**：AHB 主器件空闲 (AHB master idle)

指示 AHB 主器件状态机处于空闲情况。

注： 在设备模式和主机模式均可访问。

位 30 **DMAREQ**：DMA 请求信号使能 (DMA request signal enabled)

该位指示 DMA 请求正在进行中。用于调试。

位 29:11 保留，必须保持复位值

位 10:6 **TXFNUM**：Tx FIFO 编号 (Tx FIFO number)

使用 Tx FIFO 刷新位进行刷新的 FIFO 编号。只有在模块将 Tx FIFO 刷新位清零后，方可更改此字段。

00000：

– 主机模式下刷新非周期性 Tx FIFO

– 设备模式下刷新 Tx FIFO 0

00001：

– 主机模式下刷新周期性 Tx FIFO

– 设备模式下刷新 Tx FIFO 1

00010：设备模式下刷新 Tx FIFO 2

...

01111：设备模式下刷新 Tx FIFO 15

10000：在设备模式或主机模式下刷新所有的发送 FIFO。

注： 在设备模式和主机模式均可访问。

位 5 **TXFFLSH**：Tx FIFO 刷新 (Tx FIFO flush)

此位选择性地刷新一个或所有的发送 FIFO，但当模块处理通信事务时无法执行该操作。

只有在确认模块当前未对 Tx FIFO 执行读写操作后，应用程序方可对此位执行写操作。使用以下寄存器进行确认：

读 — NAK 有效中断可确保模块当前未对 FIFO 执行读操作。

写 — OTG_GRSTCTL 中的 AHBIDL 位可确保模块当前未对 FIFO 执行任何写操作。

通常建议在重新配置 FIFO 时进行刷新。还建议在设备端点禁止期间进行 FIFO 刷新。应用程序必须等待模块将此位清零，才能执行任意操作。该位需要八个时钟来清零（使用较慢的 phy_clk 或 hclk 时钟）。

注： 在设备模式和主机模式均可访问。

位 4 **RXFFLSH**：Rx FIFO 刷新 (Rx FIFO flush)

应用程序可使用此位刷新整个 Rx FIFO，但必须首先确保模块当前未在处理事务。

只有在确认模块当前未对 Rx FIFO 执行读写操作后，应用程序方可对此位执行写操作。

应用程序必须等到此位清零后，方可执行其它操作。通常需要等待 8 个时钟周期（以 PHY 或 AHB 时钟中最慢的为准）。

注： 在设备模式和主机模式均可访问。

位 3 保留，必须保持复位值。

位 2 保留，必须保持复位值

位 1 **PSRST**：部分软复位 (Partial soft reset)

复位内部状态机，但保留枚举信息。可用于恢复一些特定的 PHY 错误。

注： 在设备模式和主机模式均可访问。

位 0 **CSRST**：模块软复位 (Core soft reset)

按如下所述将 HCLK 和 PHY 时钟域复位：

除以下各位外，将各个中断和所有 CSR 寄存器位清零：

– OTG_PCGCCTL 中的 GAYEHCLK 位
– OTG_PCGCCTL 中的 STPPCLK 位
– OTG_HCFG 中的 FSLSPCS 位
– OTG_DCFG 中的 DSPD 位
– OTG_DCTL 中的 SDIS 位
– OTG_GCCFG 寄存器

将所有模块状态机（AHB 从器件除外）复位至空闲状态，并清空所有发送 FIFO 和接收 FIFO。

在 AHB 传输的最后数据阶段结束后，尽快终止 AHB 主器件上的所有事务。立即终止 USB 上的所有事务。

应用程序可在需要复位模块时随时对该位执行写操作。该位为自清零位，模块将在其中所有必要逻辑复位后将该位清零，该过程需要若干个时钟的时间，具体取决于模块的当前状态。该位一旦清零，软件必须等待至少 3 个 PHY 时钟后才可以访问 PHY 域（同步延迟）。此外，软件还必须在确定该寄存器中的位 31 置 1（AHB 主器件空闲）后方可开始运行。

软件复位通常在两种情况下使用，一是软件开发期间，二是用户动态更改以上所列 USB 配置寄存器中的 PHY 选择位后。用户更改 PHY 时，将为 PHY 选择相应的时钟并用于 PHY 域中。一旦选择了新的时钟，则必须复位 PHY 域，才能保证正常运行。

注： 在设备模式和主机模式均可访问。

57.14.6 OTG 模块中断寄存器 (OTG_GINTSTS)

OTG core interrupt register

偏移地址：0x014

复位值：0x1400 0020

该寄存器用于在当前模式（设备模式或主机模式）下借助系统级别的事件来中断应用程序。

该寄存器中的某些位仅在主机模式下有效，而其它位则仅在设备模式下有效。此外，该寄存器还可指示当前模式。要将 rc_w1 类型中断状态位清零，应用程序必须向该位写入 1。

FIFO 状态中断为只读；如果软件在处理这些中断期间对 FIFO 执行读写操作，则 FIFO 中断标志将自动清零。

使能中断位前，应用程序必须在初始化时将 OTG_GINTSTS 寄存器清零，才可以避免在初始化前产生任何中断。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
WKUP INT	SRQ INT	DISC INT	CIDS CHG	Res.	PTXFE	HCINT	HPRT INT	Res.	DATAF SUSP	IPXFR/ IN COMP ISO OUT	IISOI XFR	OEP INT	IEPINT	Res.	Res.
rc_w1	rc_w1	rc_w1	rc_w1		r	r	r		rc_w1	rc_w1	rc_w1	r	r		
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EOPF	ISOO DRP	ENUM DNE	USB RST	USB SUSP	ESUSP	Res.	Res.	GO NAK EFF	GI NAK EFF	NPTXF E	RXF LVL	SOF	OTG INT	MMIS	CMOD
rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1			r	r	r	r	rc_w1	r	rc_w1	r

位 31 **WKUPINT**：检测到恢复/远程唤醒中断 (Resume/remote wakeup detected interrupt)

挂起 (L2) 或 LPM(L1) 状态期间的唤醒中断。

– 挂起 (L2) 期间：

在设备模式下，当 USB 总线上检测到恢复信号时，将触发该中断。在主机模式下，当 USB 上检测到远程唤醒时，将触发该中断。

– LPM(L1) 期间：

USB 上检测到主机发起的恢复信号或设备发起的远程唤醒时，将触发该中断。

注： 在设备模式和主机模式均可访问。

位 30 **SRQINT**：检测到会话请求/新会话中断 (Session request/new session detected interrupt)

在主机模式下，当检测到来自设备的会话请求时，将触发该中断。在设备模式下，当 V_{BUS} 在 B 外设设备的有效范围内时，将触发该中断。在设备模式和主机模式均可访问。

位 29 **DISCINT**：检测到断开连接中断 (Disconnect detected interrupt)

当检测到设备断开连接时触发该中断。

注： 仅可在主机模式下访问。

位 28 **CIDSCHG**：连接器 ID 线状态更改 (Connector ID status change)

当连接器 ID 线状态发生更改时，模块将该位置 1。

注： 在设备模式和主机模式均可访问。

位 27 保留，必须保持复位值

位 26 **PTXFE**：周期性 Tx FIFO 为空 (Periodic Tx FIFO empty)

当周期性发送 FIFO 为半空或全空状态，且周期性请求队列中存在可写入至少一个条目的空间时，将触发该中断。该 FIFO 为半空状态还是全空状态由 OTG_GAHBCFG 寄存器中的周期性 Tx FIFO 空门限位（OTG_GAHBCFG 中的 PTXFELVL 位）决定。

注： 仅可在主机模式下访问。

位 25 **HCINT**：主机通道中断 (Host channels interrupt)

模块将该位置 1 时，指示模块中一个通道上存在挂起的中断（在主机模式下）。应用程序必须读取 OTG_HAINT 寄存器，以确定发生中断的通道的准确编号，然后读取相应的 OTG_HCINTx 寄存器，以确定引发中断的确切原因。应用程序必须先将 OTG_HCINTx 寄存器的相应状态位清零，之后才能将该位清零。

注： 仅可在主机模式下访问。

位 24 **HPRTINT**：主机端口中断 (Host port interrupt)

模块将该位置 1 时，指示主机模式下 OTG_HS 控制器端口的状态发生变化。应用程序必须读取 OTG_HPRT 寄存器，以确定引发此中断的确切事件。应用程序必须先将 OTG_HPRT 寄存器的相应状态位清零，之后才能将该位清零。

注： 仅可在主机模式下访问。

位 23 保留，必须保持复位值

位 22 **DATAFSUSP**：数据获取挂起 (Data fetch suspended)

该中断仅在 DMA 模式下有效。该中断指示，模块因 TxFIFO 空间或请求队列空间不可用而停止为 IN 端点获取数据。应用程序进行端点不匹配处理时会用到该中断。例如，在检测到端点不匹配后，应用程序将执行以下操作：

- 将全局非周期性 IN NAK 握手信号置 1
- 禁止 IN 端点
- 清空 FIFO
- 根据 IN 令牌序列学习队列确定令牌序列
- 重新使能端点

如果全局非周期性 IN NAK 已清零但模块尚未为 IN 端点获取数据，同时又已接收到 IN 令牌，则清零全局非周期性 IN NAK 握手信号：模块将产生“FIFO 为空时接收到 IN 令牌”中断。然后，OTG 将 NAK 响应发送到主机。为避免这种情况的发生，应用程序可以检查 OTG_GINTSTS 中的 FetSusp 中断，该中断可确保在 FIFO 存满后再将全局 NAK 握手信号清零。或者，应用程序可以在将全局 IN NAK 握手信号清零时屏蔽“当 FIFO 为空时接收到 IN 令牌”中断。

位 21 **IPXFR**：未完成周期性传输 (Incomplete periodic transfer)

在主机模式下，如果存在仍处于挂起状态的未完成周期性事务，而这些事务计划在当前帧期间完成，则模块会将该中断位置 1。

INCOMPISOOUT：未完成 OUT 同步传输 (Incomplete isochronous OUT transfer)

在设备模式下，模块将该中断置 1 时，指示当前帧中至少有一个同步 OUT 端点上的传输未完成。该中断随该寄存器中的周期性帧结束中断 (EOPF) 位一同触发。

位 20 **IISOIXFR**：未完成 IN 同步传输 (Incomplete isochronous IN transfer)

模块将该中断置 1 时，指示当前帧中至少有一个同步 IN 端点上的传输未完成。该中断随该寄存器中的周期性帧结束中断 (EOPF) 位一同触发。

注： 仅可在设备模式下访问。

位 19 **OEPINT**：OUT 端点中断 (OUT endpoint interrupt)

模块将该位置 1 时，指示模块中一个 OUT 端点上存在挂起的中断（在设备模式下）。应用程序必须读取 OTG_DAINT 寄存器，以确定发生中断的 OUT 端点的准确编号，然后读取相应的 OTG_DOEPINTx 寄存器，以确定引发中断的确切原因。应用程序必须先将相应 OTG_DOEPINTx 寄存器的相应状态位清零，之后才能将该位清零。

注： 仅可在设备模式下访问。

位 18 **IEPINT**：IN 端点中断 (IN endpoint interrupt)

模块将该位置 1 时，指示模块中一个 IN 端点上存在挂起的中断（在设备模式下）。应用程序必须读取 OTG_DAINT 寄存器，以确定发生中断的 IN 端点的准确编号，然后读取相应的 OTG_DIEPINTx 寄存器，以确定引发中断的确切原因。应用程序必须先将相应 OTG_DIEPINTx 寄存器的相应状态位清零，之后才能将该位清零。

注： 仅可在设备模式下访问。

位 17:16 保留，必须保持复位值。

位 15 **EOPF**：周期性帧结束中断 (End of periodic frame interrupt)

指示当前帧已达到 OTG_DCFG 寄存器中周期性帧间隔字段（OTG_DCFG 中的 PFIVL 位）所指定的周期。

注： 仅可在设备模式下访问。

位 14 **ISOODRP**：丢弃同步 OUT 数据包中断 (Isochronous OUT packet dropped interrupt)

如果由于 Rx FIFO 空间不足，无法容纳同步 OUT 端点的最大数据包，从而导致模块无法向 Rx FIFO 写入同步 OUT 数据包，模块会将该位置 1。

注： 仅可在设备模式下访问。

位 13 **ENUMDNE**：枚举完成 (Enumeration done)

模块将该位置 1 时，指示速率枚举已完成。应用程序必须读取 OTG_DSTS 寄存器来获取枚举速率。

注： 仅可在设备模式下访问。

位 12 **USBRST**：USB 复位 (USB reset)

模块将该位置 1 时，指示在 USB 上检测到复位信号。

注： 仅可在设备模式下访问。

位 11 **USBSUSP**：USB 挂起 (USB suspend)

模块将该位置 1 时，指示在 USB 上检测到挂起状态。当数据线上的空闲状态保持一段额外的时间后，模块进入挂起状态。

注： 仅可在设备模式下访问。

位 10 **ESUSP**：早期挂起 (Early suspend)

模块将该位置 1 时，指示已检测到 USB 处于空闲状态的时间达到 3 ms。

注： 仅可在设备模式下访问。

位 9:8 保留，必须保持复位值。

位 7 **GONAKEFF**：全局 OUT NAK 有效 (Global OUT NAK effective)

指示 OTG_DCTL 寄存器中由应用程序设置的“将全局 OUT NAK 置 1”位（OTG_DCTL 中的 SGONAK 位）已在模块中生效。通过写入 OTG_DCTL 寄存器中的“将全局 OUT NAK 清零”位（OTG_DCTL 中的 CGONAK 位），可将该位清零。

注： 仅可在设备模式下访问。

位 6 **GINAKEFF**：全局非周期性 IN NAK 有效 (Global IN nonperiodic NAK effective)

指示 OTG_DCTL 寄存器中由应用程序设置的“将全局非周期性 IN NAK 置 1”位（OTG_DCTL 中的 SGINAK 位）已在模块中生效。也就是说，模块已对应用程序设置的全局 IN NAK 位进行采样，结果已生效。通过清零 OTG_DCTL 寄存器中的“将全局非周期性 IN NAK 清零”位（OTG_DCTL 中的 CGINAK 位），可将该位清零。

此中断不一定表示 USB 上已发送了一个 NAK 握手信号。STALL 位优先级高于 NAK 位。

注： 仅可在设备模式下访问。

位 5 **NPTXFE**：非周期性 Tx FIFO 为空 (Non-periodic Tx FIFO empty)

当非周期性 Tx FIFO 为半空或全空状态，且非周期性发送请求队列中至少存在可写入一个条目的空间时，将触发该中断。该 FIFO 为半空状态还是全空状态由 OTG_GAHBCFG 寄存器中的非周期性 Tx FIFO 空门限位（OTG_GAHBCFG 中的 TXFELVL 位）决定。

注： 仅可在主机模式下访问。

位 4 **RXFLVL**：Rx FIFO 非空 (Rx FIFO non-empty)

指示 Rx FIFO 中至少有一个数据包等待读取。

注： 在主机模式和设备模式均可访问。

位 3 **SOF**：帧起始 (Start of frame)

在主机模式下，模块将该位置 1 时，指示 USB 上已发送一个 SOF (FS) 或 Keep-Alive (LS) 信号。应用程序必须将此位置 1 才可清除该中断。

在设备模式下，模块将该位置 1 时，指示 USB 上已接收到一个 SOF 令牌。应用程序可通过读取 OTG_DSTS 寄存器来获得当前的帧编号。只有在模块以 FS 模式运行时，才会出现此中断。

注： 如果上电复位后立即读取，该寄存器可能返回“1”。如果寄存器位在上电复位后立即读取“1”，则不表示已发送 SOF（主机模式下）或已接收 SOF（设备模式下）。只有在主机和设备之间建立有效连接后，此中断的读取值才有效。如果此位在上电复位后置 1，应用程序可把该位清零。

注： 在主机模式和设备模式均可访问。

位 2 **OTGINT**：OTG 中断 (OTG interrupt)

模块将该位置 1 时，指示出现 OTG 协议事件。应用程序必须读取 OTG 中断状态 (OTG_GOTGINT) 寄存器，以确定引发此中断的确切事件。应用程序必须先将 OTG_GOTGINT 寄存器的相应状态位清零，之后才能将该位清零。

注： 在主机模式和设备模式均可访问。

位 1 **MMIS**：模式不匹配中断 (Mode mismatch interrupt)

当应用程序尝试访问以下寄存器时，模块将该位置 1：

– 模块运行在设备模式下访问主机模式寄存器

– 模块运行在主机模式下访问设备模式寄存器

寄存器访问在 AHB 上以 OKAY 响应结束，但该访问在内部被模块忽略并且不会影响模块运行。

注： 在主机模式和设备模式均可访问。

位 0 **CMOD**：当前工作模式 (Current mode of operation)

指示当前模式。

0：设备模式

1：主机模式

注： 在主机模式和设备模式均可访问。

57.14.7 OTG 中断屏蔽寄存器 (OTG_GINTMSK)

OTG interrupt mask register

偏移地址：0x018

复位值：0x0000 0000

该寄存器与模块中断寄存器结合使用，以中断应用程序。如果将某个中断位屏蔽，则不会产生与该位相关的中断。但是，与该中断相对应的模块中断 (OTG_GINTSTS) 寄存器位仍会置 1。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
WUIM	SRQIM	DISCINT	CIDSCHGM	LPMINTM	PTXFEM	HCIM	PRTIM	RSTDETM	FSUSPM	IPXFRM/IISOOXFRM	IISOIXFRM	OEPINT	IEPINT	Res.	Res.
rw	rw	rw	rw	rw	rw	rw	r	rw	rw	rw	rw	rw	rw		
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EOPFM	ISOODRPM	ENUMDNEM	USBRST	USBSUSPM	ESUSPM	Res.	Res.	GONAKEFFM	GINAKEFFM	NPTXFEM	RXFLVLM	SOFM	OTGINT	MMISM	Res.
rw	rw	rw	rw	rw	rw			rw	rw	rw	rw	rw	rw	rw	

位 31 **WUIM**：检测到恢复/远程唤醒中断屏蔽 (Resume/remote wakeup detected interrupt mask)
0：屏蔽中断
1：使能中断
注： 在主机模式和设备模式均可访问。

位 30 **SRQIM**：检测到会话请求/新会话中断屏蔽 (Session request/new session detected interrupt mask)
0：屏蔽中断
1：使能中断
注： 在主机模式和设备模式均可访问。

位 29 **DISCINT**：检测到断开连接中断屏蔽 (Disconnect detected interrupt mask)
0：屏蔽中断
1：使能中断
注： 仅可在设备模式下访问。

位 28 **CIDSCHGM**：连接器 ID 状态更改屏蔽 (Connector ID status change mask)
0：屏蔽中断
1：使能中断
注： 在主机模式和设备模式均可访问。

位 27 **LPMINTM**：LPM 中断屏蔽 (LPM interrupt mask)
0：屏蔽中断
1：使能中断
注： 在主机模式和设备模式均可访问。

位 26 **PTXFEM**：周期性 Tx FIFO 空屏蔽 (Periodic Tx FIFO empty mask)
0：屏蔽中断
1：使能中断
注： 仅可在主机模式下访问。

位 25 **HCIM**：主机通道中断屏蔽 (Host channels interrupt mask)
0：屏蔽中断
1：使能中断
注： 仅可在主机模式下访问。

位 24 **PRTIM**：主机端口中断屏蔽 (Host port interrupt mask)
0：屏蔽中断
1：使能中断
注： 仅可在主机模式下访问。

位 23 **RSTDETM**：检测到复位中断屏蔽 (Reset detected interrupt mask)
0：屏蔽中断
1：使能中断
注： 仅可在设备模式下访问。

位 22 **FSUSPM**：数据获取挂起中断屏蔽 (Data fetch suspended mask)
0：屏蔽中断
1：使能中断
仅可在设备模式下访问。

位 21 **IPXFRM**：未完成周期性传输中断屏蔽 (Incomplete periodic transfer mask)
0：屏蔽中断
1：使能中断
注： 仅可在主机模式下访问。
IISOOXFRM：未完成 OUT 同步传输中断屏蔽 (Incomplete isochronous OUT transfer mask)
0：屏蔽中断
1：使能中断
注： 仅可在设备模式下访问。

位 20 **IISOIXFRM**：未完成 IN 同步传输中断屏蔽 (Incomplete isochronous IN transfer mask)
0：屏蔽中断
1：使能中断
注： 仅可在设备模式下访问。

位 19 **OEPINT**：OUT 端点中断屏蔽 (OUT endpoints interrupt mask)
0：屏蔽中断
1：使能中断
注： 仅可在设备模式下访问。

位 18 **IEPINT**：IN 端点中断屏蔽 (IN endpoints interrupt mask)
0：屏蔽中断
1：使能中断
注： 仅可在设备模式下访问。

位 17:16 保留，必须保持复位值。

位 15 **EOPFM**：周期性帧结束中断屏蔽 (End of periodic frame interrupt mask)
0：屏蔽中断
1：使能中断
注： 仅可在设备模式下访问。

位 14 **ISOODRPM**：丢弃同步 OUT 数据包中断屏蔽 (Isochronous OUT packet dropped interrupt mask)
0：屏蔽中断
1：使能中断
注： 仅可在设备模式下访问。

位 13 **ENUMDNEM**：枚举完成中断屏蔽 (Enumeration done mask)
0：屏蔽中断
1：使能中断
注： 仅可在设备模式下访问。

位 12 **USBRST**：USB 复位中断屏蔽 (USB reset mask)
0：屏蔽中断
1：使能中断
注： 仅可在设备模式下访问。

位 11 **USBSUSPM**：USB 挂起中断屏蔽 (USB suspend mask)
0：屏蔽中断
1：使能中断
注： 仅可在设备模式下访问。

位 10 **ESUSPM**：早期挂起中断屏蔽 (Early suspend mask)
0：屏蔽中断
1：使能中断
注： 仅可在设备模式下访问。

位 9:8 保留，必须保持复位值。

位 7 **GONAKEFFM**：全局 OUT NAK 生效中断屏蔽 (Global OUT NAK effective mask)
0：屏蔽中断
1：使能中断
注： 仅可在设备模式下访问。

位 6 **GINAKEFFM**：全局非周期性 IN NAK 生效中断屏蔽 (Global non-periodic IN NAK effective mask)
0：屏蔽中断
1：使能中断
注： 仅可在设备模式下访问。

位 5 **NPTXFEM**：非周期性 Tx FIFO 空中断屏蔽 (Non-periodic Tx FIFO empty mask)
0：屏蔽中断
1：使能中断
注： 仅可在主机模式下访问。

位 4 **RXFLVLM**：接收 FIFO 非空中断屏蔽 (Receive FIFO non-empty mask)
0：屏蔽中断
1：使能中断
注： 在设备模式和主机模式均可访问。

位 3 **SOFM**：帧起始中断屏蔽 (Start of frame mask)
0：屏蔽中断
1：使能中断
注： 在设备模式和主机模式均可访问。

位 2 **OTGINT**：OTG 中断屏蔽 (OTG interrupt mask)
0：屏蔽中断
1：使能中断
注： 在设备模式和主机模式均可访问。

位 1 **MMISM**：模式不匹配中断屏蔽 (Mode mismatch interrupt mask)
0：屏蔽中断
1：使能中断
注： 在设备模式和主机模式均可访问。

位 0 保留，必须保持复位值。

57.14.8 OTG_FS 接收状态调试读取/OTG 状态读取和出栈寄存器 (OTG_GRXSTSR/OTG_GRXSTSP)

OTG_FS Receive status debug read/OTG status read and pop registers

读取的偏移地址：0x01C

出栈的偏移地址：0x020

复位值：0x0000 0000

读取接收状态调试读取寄存器将返回接收 FIFO 顶部的内容。读取接收状态读取和出栈寄存器将额外弹出 Rx FIFO 顶部的数据条目。

接收状态内容在主机模式和设备模式下的解释不同。当接收 FIFO 为空时，模块会忽略对该寄存器的读取或出栈操作，并返回值 0x0000 0000。当模块中断寄存器的接收 FIFO 非空位（OTG_GINTSTS 中的 RXFLVL 位）置位时，应用程序必须仅弹出接收状态 FIFO。

主机模式：

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PKTSTS				DPID
											r	r	r	r	r

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DPID	BCNT											CHNUM			
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:21 保留，必须保持复位值。

位 20:17 **PKTSTS**：数据包状态 (Packet status)
指示接收的数据包的状态
0010：接收到 IN 数据包
0011：IN 传输完成（触发中断）
0101：数据翻转错误（触发中断）
0111：通道停止（触发中断）
其它值：保留

位 16:15 **DPID**：数据 PID (Data PID)
指示接收的数据包的数据 PID
00：DATA0
10：DATA1
01：DATA2
11：MDATA

位 14:4 **BCNT**：字节数 (Byte count)
指示接收的 IN 数据包的字节数。

位 3:0 **CHNUM**：通道编号 (Channel number)
指示当前接收的数据包所属的通道编号。

设备模式：

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	FRMNUM				PKTSTS				DPID
							r	r	r	r	r	r	r	r	r

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DPID	BCNT											EPNUM			
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:25 保留，必须保持复位值。

位 24:21 **FRMNUM**：帧编号 (Frame number)
这是在 USB 上接收的数据包帧编号的 4 个最低有效位。只有当支持同步 OUT 端点时才支持此字段。

位 20:17 **PKTSTS**：数据包状态 (Packet status)
指示接收的数据包的状态
0001：全局 OUT NAK（触发中断）
0010：接收到 OUT 数据包
0011：OUT 传输完成（触发中断）
0100：SETUP 事务完成（触发中断）
0110：接收到 SETUP 数据包
其它值：保留

位 16:15 **DPID**：数据 PID (Data PID)
指示接收的 OUT 数据包的数据 PID
00：DATA0
10：DATA1
01：DATA2
11：MDATA

位 14:4 **BCNT**：字节数 (Byte count)
指示接收的数据包的字节数。

位 3:0 **EPNUM**：端点编号 (Endpoint number)
指示当前接收的数据包所属的端点编号。

57.14.9 OTG 接收 FIFO 大小寄存器 (OTG_GRXFSIZ)

OTG Receive FIFO size register

偏移地址：0x024

复位值：0x0000 0400

此应用程序可以对分配给 Rx FIFO 的 RAM 大小进行编程。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RXFD															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:16 保留，必须保持复位值。

位 15:0 **RXFD**：Rx FIFO 深度 (Rx FIFO depth)
以 32 位字为单位。
最大值为 1024
编程值必须遵循可分配的 FIFO 存储区，不得超过上电值。

57.14.10 OTG 主机非周期性发送 FIFO 大小寄存器 (OTG_HNPTXFSIZ)/端点 0 发送 FIFO 大小 (OTG_DIEPTXF0)

OTG Host non-periodic transmit FIFO size register

偏移地址：0x028

复位值：0x0200 0200

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NPTXFD/TX0FD															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NPTXFSA/TX0FSA															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

主机模式

位 31:16 **NPTXFD**：非周期性 Tx FIFO 深度 (Non-periodic Tx FIFO depth)
以 32 位字为单位。
最小值为 16
编程值必须遵循可分配的 FIFO 存储区，不得超过上电值。

位 15:0 **NPTXFSA**：非周期性发送 RAM 起始地址 (Non-periodic transmit RAM start address)
此字段配置非周期性发送 FIFO RAM 的存储器起始地址。

设备模式

位 31:16 **TX0FD**：端点 0 Tx FIFO 深度 (Endpoint 0 Tx FIFO depth)
以 32 位字为单位。
最小值为 16
编程值必须遵循可分配的 FIFO 存储区，不得超过上电值。

位 15:0 **TX0FSA**：端点 0 发送 RAM 起始地址 (Endpoint 0 transmit RAM start address)
此字段配置端点 0 发送 FIFO RAM 的存储器起始地址。

57.14.11 OTG 非周期性发送 FIFO/队列状态寄存器 (OTG_HNPTXSTS)

OTG non-periodic transmit FIFO/queue status register

偏移地址：0x02C

复位值：0x0008 0400

注： *在设备模式下，此寄存器无效。*

此只读寄存器包含非周期性 Tx FIFO 和非周期性发送请求队列的空闲空间信息。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	NPTXQTOP							NPTQXSAV							
	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NPTXFSAV															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31 保留，必须保持复位值。

位 30:24 **NPTXQTOP**：非周期性发送请求队列顶部 (Top of the non-periodic transmit request queue)

非周期性发送请求队列中 MAC 目前正在处理的条目。

位 30:27：通道/端点编号 (Channel/endpoint number)

位 26:25：

00：IN/OUT 令牌

01：长度为零的发送数据包（设备 IN/主机 OUT）

11：通道停止命令

位 24：结束（所选通道/端点的最后一个条目）

(Terminate (last entry for selected channel/endpoint))

位 23:16 **NPTQXSAV**：非周期性发送请求队列可用空间 (Non-periodic transmit request queue space available)

指示非周期性发送请求队列中的可用空闲空间大小。该队列既包含 IN 请求，又包含 OUT 请求。

0：非周期性发送请求队列已满

1：1 个位置可用

2：2 个位置可用

n：n 个位置可用 (0 ≤ n ≤ 8)

其它值：保留

位 15:0 **NPTXFSAV**：非周期性 Tx FIFO 可用空间 (Non-periodic Tx FIFO space available)

指示非周期性 Tx FIFO 中的可用空闲空间大小。

以 32 位字为单位。

0：非周期性 TxFIFO 已满

1：1 个字可用

2：2 个字可用

n：n 个位置可用（其中，0 ≤ n ≤ 512）

其它值：保留

57.14.12 OTG I²C 访问寄存器 (OTG_GI2CCTL)

OTG I²C access register

偏移地址：0x030

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
BSYDNE	RW.	Res.	I2CDATSE	I2CDEVADR		Res.	ACK	I2CEN	ADDR						
rw	rw		rw	rw	rw		rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REGADDR								RWDATA							
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31 **BSYDNE**：I2C 忙碌/完成 (I2C Busy/Done)

应用程序将该位置 1 可启动 I²C 接口上的请求。传输完成后，模块会将此位清零。只要该位置 1，即表示 I²C 接口忙碌，应用程序便无法启动该接口的另一请求。

位 30 **RW**：读/写指示符 (Read/Write Indicator)

该位指示接口上必须执行读寄存器传输还是写寄存器传输。

0：写

1：读

注： 寄存器不支持突发读/写操作。

位 29 保留，必须保持复位值。

位 28 **I2CDATSE0**：I²C DatSe0 USB 模式 (I²C DatSe0 USB mode)

该位用于选择全速接口 USB 模式。

0：VP_VM USB 模式

1：DAT_SE0 USB 模式

位 27:26 **I2CDEVADR**：I²C 器件地址 (I²C Device Address)

利用此位可选择模块用于进行 OTG 信号传输所使用的 USB 1.1 全速串行收发器的 I²C 从器件的地址。

位 25 保留，必须保持复位值。

位 24 **ACK**：I²C ACK

该位指示是否从 I²C 从器件中接收到 ACK 应答。当应用程序启动 I²C 访问，模块将 BSYDNE 位清零后，该位才有效。

0：NAK

1：ACK

位 23 **I2CEN**：I²C 使能 (I²C Enable)

该位可使能 I²C 主器件在 I²C 接口上启动通信事务。

位 22:16 **ADDR**：I²C 地址 (I²C Address)

此位是 7 位 I²C 器件地址，应用程序利用此地址来访问任何外部 I²C 从器件，包括 USB 1.1 OTG 全速串行收发器上的 I²C 从器件。

位 15:8 **REGADDR**：I²C 寄存器地址 (I²C Register Address)

这些位用于编程要读取/写入的寄存器的地址。

位 7:0 **RWDATA**：I^2C 读/写数据 (I^2C Read/Write Data)
执行寄存器读操作之后，这些位可存储应用程序的读取数据。
执行写操作期间，应用程序可使用此寄存器编程要写入寄存器的数据。

注： *配置寄存器只适用于 USB OTG HS*

57.14.13 OTG 通用模块配置寄存器 (OTG_GCCFG)

OTG general core configuration register

偏移地址：0x038

复位值：0x0000 XXX0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	VBDEN	SDEN	PDEN	DCD EN	BCDEN	PWR DWN
										rw	rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PS2 DET	SDET	PDET	DCDET
												rw	rw	rw	rw

位 31:22 保留，必须保持复位值。

位 21 **VBDEN**：USB V_{BUS} 检测使能 (USB V_{BUS} detection enable)
使能 V_{BUS} 感应比较器检测 USB 主机和设备工作模式下的 V_{BUS} 引脚上的 V_{BUS} 有效电平。如果使能 HNP 和/或 SRP 支持，将自动使能 V_{BUS} 比较器，而与 VBDEN 值无关。
0 = 禁止 V_{BUS} 检测
1 = 使能 V_{BUS} 检测

位 20 **SDEN**：二次检测 (SD) 模式使能 (Secondary detection (SD) mode enable)
该位由软件置 1，用于将 BCD 置于 SD 模式。要正确工作，应只选择一个检测模式（DCD、PD、SD 或 OFF）

位 19 **PDEN**：初次检测 (PD) 模式使能 (Primary detection (PD) mode enable)
该位由软件置 1，用于将 BCD 置于 PD 模式。要正确工作，应只选择一个检测模式（DCD、PD、SD 或 OFF）。

位 18 **DCDEN**：数据接触点检测 (DCD) 模式使能 (Data contact detection (DCD) mode enable)
该位由软件置 1，用于将 BCD 置于 DCD 模式。要正确工作，应只选择一个检测模式（DCD、PD、SD 或 OFF）。（有待澄清）

位 17 **BCDEN**：电池充电器检测 (BCD) 使能 (Battery charging detector (BCD) enable)
该位由软件置 1，用于在 USB 设备内使能 BCD 支持。使能后，USB PHY 完全由 BCD 控制，无法用于正常通信。BCD 检测完成后，应通过将该位清零使 BCD 置于 OFF 模式，以实现正常的 USB 操作。

位 16 **PWRDWN**：掉电控制 (Power down control)
用于在发送/接收时激活收发器 复位时，收发器保持掉电状态。置 1 时，必须关闭 BCD 功能 (BCDEN=0)。
0 = 禁止 USB FS 收发器
1 = 使能 USB FS 收发器

位 15:4 保留，必须保持复位值。

位 3 **PS2DET**：DM 上拉检测状态 (DM pull-up detection status)

该位只在 PD 期间有效，用于指示 DM 电压电平与 VLGC 阈值之间的比较结果。正常情况下，DM 电平应低于此阈值。如果 DM 电平高于此阈值，则意味着 DM 已外部拉为高电平。原因可能是连接到 PS2 端口（同时上拉 DP 和 DM 线）或某些不遵循 BCD 规范的专有充电器。

0：检测到正常端口（连接到 SDP、CDP 或 DCP）

1：检测到 PS2 端口或专有充电器

位 2 **SDEN**：二次检测 (SD) 状态 (Secondary detection (SD) status)

此位指示 SD 的结果。

0：检测到 CDP

1：检测到 DCP

位 1 **PDEN**：初次检测 (PD) 状态 (Primary detection (PD) status)

此位指示 PD 的结果。

0：未检测到 BCD 支持（连接到 SDP 或专有设备）。

1：检测到 BCD 支持（连接到 CDP 或 DCP）。

位 0 **DCDEN**：数据接触点检测 (DCD) 状态 (Data contact detection (DCD) status)

此位指示 DCD 的结果。

0：未检测到数据点接触

1：检测到数据点接触

57.14.14 OTG 模块 ID 寄存器 (OTG_CID)

OTG core ID register

偏移地址：0x03C

复值：0x0000 3100 该寄存器为只读寄存器，包含产品 ID。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PRODUCT_ID															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PRODUCT_ID															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:0 **PRODUCT_ID**：产品 ID 字段 (Product ID field)

可通过应用程序编程的 ID 字段。

57.14.15 OTG 模块 LPM 配置寄存器 (OTG_GLPMCFG)

OTG core LPM configuration register

偏移地址：0x54

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	EN BESL	LPMRCNTSTS			SND LPM	LPMRCNT			LPMCHIDX				L1RSM OK
			rw	r	r	r	rs	rw	rw	rw	rw	rw	rw	rw	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SLP STS	LPMRSP		L1DS EN	BESLTHRS				L1SS EN	REM WAKE	BESL				LPM ACK	LPM EN
r	r	r	rw	rw	rw	rw	rw	rw	rw/r	rw/r	rw/r	rw/r	rw/r	rw	rw

位 31:29 保留，必须保持复位值。

位 28 **ENBESL**：使能尽力服务延迟 (Enable best effort service latency)

该位用于使能 LPM 勘误表中定义的 BESL 功能：

0：模块按照以下文档中的说明工作：

基于 USB 2.0 规范的 USB 2.0 链路层电源管理补充工程变更通知，2007 年 7 月 16 日

1：模块按照 LPM 勘误表中的说明工作：

USB 2.0 ECN 的勘误表：链路层电源管理 (LPM) - 2007 年 7 月

注： 本文档只考虑（LPM 勘误表中说明的）更新行为，因此 ENBESL 位应由应用程序软件置 1。

位 27:25 **LPMRCNTSTS**：LPM 重试计数状态 (LPM retry count status)

当前 LPM 序列仍要发送的 LPM 主机重试次数

注： 仅可在主机模式下访问。

位 24 **SNDLPM**：发送 LPM 事务 (Send LPM transaction)

当应用程序软件将该位置 1 时，将发送包含两个令牌 EXT 和 LPM 的 LPM 事务。从设备接收到有效响应（STALL、NYET 或 ACK）或模块发送完编程的 LPM 重试次数后，硬件即将该位清零。

注： 只有当主机连接到本地端口时，该位才可置 1。

注： 仅可在主机模式下访问。

位 23:21 **LPMRCNT**：LPM 重试计数 (LPM retry count)

当设备发出 ERROR 响应后，该位表示接收到有效设备响应（STALL、NYET 或 ACK）之前，主机需额外执行的 LPM 重试次数。

注： 仅可在主机模式下访问。

位 20:17 **LPMCHIDX**：LPM 通道索引 (LPM Channel Index)

向本地设备发送 LPM 事务时，必须应用到 LPM 事务的通道编号。基于 LPM 通道编号，模块可自动向 LPM 事务插入相应通道中编程的设备地址和端点号。

注： 仅可在主机模式下访问。

位 16 **L1RSMOK**：睡眠状态恢复正常 (Sleep State Resume OK)

表示设备或主机可以从睡眠状态启动恢复。该位在 LPM 睡眠 (L1) 状态下有效。它在睡眠模式下经 50 µs ($T_{L1Residency}$) 延时后置 1。

该位在 SLPSTS 为 0 时复位。

1：应用程序或主机可以从睡眠状态启动恢复

0：应用程序或主机无法从睡眠状态启动恢复

位 15 **SLPSTS**：端口睡眠状态 (Port sleep status)

设备模式：

只要 USB 总线上存在睡眠条件，该位就会置 1。当 ACK 响应发送给 LPM 事务且 $T_{L1TokenRetry}$ 定时器过期时，模块将进入睡眠模式。要停止 PHY 时钟，应用程序必须将 OTG_PCGCCTL 中的 STPPCLK 位置 1，这将触发 PHY 挂起输入信号。

应用程序必须依靠 LPMRSP 中的 SLPSTS 而不是 ACK 来确认切换到睡眠状态。

出现以下情况时，模块会退出睡眠状态：

– USB 线上有活动时

– 当应用程序对 OTG_DCTL 中的 RWUSIG 位执行写操作或者复位或软断开设备时。

主机模式：

模块通过来自设备的 ACK 响应成功将 LPM 事务发送给本地端口后，主机将切换到睡眠 (L1) 状态。此位的读取值反映该端口的当前睡眠状态。

在以下事件后，模块会将该位清零：

– 模块检测到远程 L1 唤醒信号，

– 应用程序将 OTG_HPRT 寄存器中的 PRST 位或 PRES 位置 1，或者

– 应用程序将模块中断寄存器中的“检测到 L1 恢复/远程唤醒中断”位或“检测到断开连接中断”位（分别为 OTG_GINTSTS 中的 WKUPINT 或 DISCINT 位）置 1。

0：模块未处于 L1

1：模块处于 L1

位 14:13 **LPMRST**：LPM 响应 (LPM response)

设备模式：

这两位反映模块对所接收的 LPM 事务的响应。

主机模式：

从本地设备接收、用于 LPM 事务的握手响应

11：ACK

10：NYET

01：STALL

00：ERROR（无握手响应）

位 12 **L1DSEN**：L1 深度睡眠使能 (L1 deep sleep enable)

在 L1 睡眠模式下使能挂起 PHY。为了在 L1 睡眠模式下最大程度节能，所有情况下该位均应由应用程序软件置 1。

位 11:8 **BESLTHRS**：BESL 阈值 (BESL threshold)

设备模式：

当 BESL 值大于或等于 BESL_Thres[3:0] 字段中定义的值时，模块将 PHY 置于 L1 下的深度低功耗模式。

主机模式：

模块将 PHY 置于 L1 下的深度低功耗模式。BESLTHRS[3:0] 用于指定要由主机在 USB 总线上反映的恢复信号的持续时间 ($T_{L1HubDrvResume2}$)。

在主机模式下，BESLTHRS 不得编程为大于 1100b 的值，原因是这将超出最大 $T_{L1HubDrvResume2}$。

Thres[3:0]主模式恢复信号的持续时间 (µs)

0000:75
0001:100
0010:150
0011:250
0100:350
0101:450
0110:950
所有其它值：保留

位 7 **L1SSEN**：L1 浅度睡眠使能 (L1 Shallow Sleep enable)

在 L1 睡眠模式下使能挂起 PHY。为了在 L1 睡眠模式下最大程度节能，所有情况下该位均应由应用程序软件置 1。

位 6 **REMWAKE**：bRemoteWake 值 (bRemoteWake value)

主机模式：

要在 LPM 事务的 wIndex 字段中发送的远程唤醒值。

设备模式（只读）：

当 ACK、NYET 或 STALL 响应发送给 LPM 事务时，该字段用接收的 LPM 令牌 bRemoteWake bmAttribute 进行更新。

位 5:2 **BESL**：尽力服务延迟 (Best effort service latency)

主机模式：

要在 LPM 事务中发送的 BESL 值 该值还用于发起持续时间为 $T_{L1HubDrvResume1}$ 的恢复信号，以实现主机发起的恢复操作。

器件模式（只读）：

当 ACK、NYET 或 STALL 响应发送给 LPM 事务时，该字段用接收的 LPM 令牌 BESL bmAttribute 进行更新。

BESL[3:0]T_{BESL} (µs)
0000:125
0001:150
0010:200
0011:300
0100:400
0101:500
0110:1000
0111:2000
1000:3000
1001:4000
1010:5000
1011:6000
1100:7000
1101:8000
1110:9000
1111:10000

位 1 **LPMACK**：LPM 令牌应答使能 (LPM token acknowledge enable)

设备应用程序软件预编程的 LPM 令牌握手响应。

1：ACK

即使已预编程 ACK，也只有在成功完成 LPM 事务后，模块才能以 ACK 进行响应。LPM 事务成功的前提是：

– EXT 令牌或 LPM 令牌中没有 PID/CRC5 错误（否则会出现 ERROR）
– LPM 事务中接收到 Valid bLinkState = 0001B (L1)（否则会出现 STALL）
– 发送队列中没有待处理的数据（否则会出现 NYET）

0：NYET

在以下情况下，为响应 LPM 令牌，会重写预编程的软件位：

– 接收到的 bLinkState 不是 L1（STALL 响应），或者
– 由于任一 LPM 令牌数据包损坏而在其中检测到错误（ERROR 响应）。

注： 仅可在设备模式下访问。

位 0 **LPMEN**：LPM 支持使能 (LPM support enable)

应用程序使用该位控制 OTG_HS 模块的 LPM 功能。

如果模块以不支持 LPM 功能的主机工作，则无法请求所连接的设备或集线器激活 LPM 模式。

如果模块以不支持 LPM 功能的设备工作，则无法响应任何 LPM 事务。

0：不使能 LPM 功能
1：使能 LPM 功能

57.14.16 OTG 主机周期性发送 FIFO 大小寄存器 (OTG_HPTXFSIZ)

OTG Host periodic transmit FIFO size register

偏移地址：0x100

复位值：0x0200 0400

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PTXFSIZ															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PTXSA															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:16 **PTXFD**：主机周期性传输 Tx FIFO 深度 (Host periodic Tx FIFO depth)
以 32 位字为单位。
最小值为 16

位 15:0 **PTXSA**：主机周期性传输 Tx FIFO 起始地址 (Host periodic Tx FIFO start address)
此字段用于配置周期性发送 FIFO RAM 的存储器起始地址。

57.14.17 OTG 设备 IN 端点发送 FIFO 大小寄存器 (OTG_DIEPTXFx)（x = 1..8，其中 x 为 FIFO 编号）

OTG device IN endpoint transmit FIFO size register

偏移地址：0x104 + (FIFO 编号 – 1) × 0x04

复位值：

FIFO 编号 = 8: 0x0200 0200 + (8 * 0x200)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
INEPTXFD															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INEPTXSA															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:16 **INEPTXFD**：IN 端点 Tx FIFO 深度 (IN endpoint Tx FIFO depth)
以 32 位字为单位。
最小值为 16

位 15:0 **INEPTXSA**：IN 端点发送 FIFOx RAM 起始地址 (IN endpoint FIFOx transmit RAM start address)
此字段包含 IN 端点发送 FIFOx 的存储器起始地址。该地址必须与 32 位存储器位置对齐。

57.14.18 主机模式寄存器

Host-mode registers

除非特别说明，否则寄存器描述中的位值以二进制表示。

主机模式寄存器会影响主机模式下的模块操作。在设备模式下不得访问主机模式寄存器，因为产生的结果不明确。主机模式寄存器可进行如下分类：

57.14.19 OTG 主机配置寄存器 (OTG_HCFG)

OTG Host configuration register

偏移地址：0x400

复位值：0x0000 0000

此寄存器将在上电后对模块进行配置。请勿在初始化主机后更改此寄存器。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.		Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FSLSS	FSLSPCS	
													r	rw	rw

位 31:24 保留，必须保持复位值。

位 23 保留，必须保持复位值。

位 22:3 保留，必须保持复位值。

位 2 **FSLSS**：仅支持 FS 和 LS (FS- and LS-only support)

应用程序使用此位控制模块的枚举速度。使用此位，应用程序可使模块工作为 FS 主机，即使所连接的设备支持 HS 通信也是如此。请勿在初始编程后更改此字段。

位 1:0 **FSLSPCS**：FS/LS PHY 时钟选择 (FS/LS PHY clock select)

当模块处于 FS 主机模式时

01：PHY 时钟以 48 MHz 运行

其它值：保留

当模块处于 LS 主机模式时

00：保留

01：选择 48 MHz PHY 时钟频率

10：选择 6 MHz PHY 时钟频率

11：保留

注： *当设备连上主机时，必须依照所连接设备的速度设置 FSLSPCS（更改此位后，必须进行软件复位）。*

57.14.20 OTG 主机帧间隔寄存器 (OTG_HFIR)

OTG Host frame interval register

偏移地址：0x404

复位值：0x0000 EA60

此寄存器用于存储 OTG_HS 控制器对已连接设备当前速度所设定的帧间隔信息。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RLD CTRL
															rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FRIVL															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:17 保留，必须保持复位值。

位 16 **RLDCTRL**：重载控制 (Reload control)

该位允许在运行期间动态重载 HFIR 寄存器。

0：无法动态重载 HFIR

1：运行期间可动态重载 HFIR。

该位需要在初始配置期间编程，并且不得在运行期间更改其值。

位 15:0 **FRIVL**：帧间隔 (Frame interval)

应用程序在此字段编程的值用于指定两个连续 micro-SOF (HS) 或 Keep-Alive 令牌 (LS) 之间的时间间隔。此字段包含构成所需帧间隔的 PHY 时钟数。只有将主机端口控制和状态寄存器的端口使能位（OTG_HPRT 的 PENA 位）置 1 后，应用程序才能向此寄存器中写入值。如果未对值进行编程，模块将根据在主机配置寄存器的 FS/LS PHY 时钟选择字段（OTG_HCFG 中的 FSLSPCS）中指定的 PHY 时钟来计算。除非 RLDCTRL 位置 1，否则请勿在初始配置后更改此字段的值。在这种情况下，FRIVL 将在发生每个 SOF 事件时进行重载。

57.14.21 OTG 主机帧编号/帧剩余时间寄存器 (OTG_HFNUM)

OTG Host frame number/frame time remaining register

偏移地址：0x408

复位值：0x0000 3FFF

此寄存器用于指示当前帧编号。它还指示当前帧的剩余时间（以 PHY 时钟数为单位）。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FTREM															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FRNUM															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:16 **FTREM**：帧剩余时间 (Frame time remaining)

指示当前帧的剩余时间（以 PHY 时钟数为单位）。每过去 1 个 PHY 时钟，此字段递减 1。当值达到零时，此字段将重新装载帧间隔寄存器中的值，并由模块在 USB 上发送一个新 SOF。

位 15:0 **FRNUM**：帧编号 (Frame number)

当在 USB 上发送 1 个新 SOF 时此字段的值将递增 1，当达到 0x3FFF 时会清零。

57.14.22 OTG_Host 周期性发送 FIFO/队列状态寄存器 (OTG_HPTXSTS)

OTG_Host periodic transmit FIFO/queue status register

偏移地址：0x410

复位值：0x0008 0100

此只读寄存器包含周期性 Tx FIFO 和周期性发送请求队列的空闲空间信息。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PTXQTOP								PTXQSAV							
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PTXFSAVL															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:24 **PTXQTOP**：周期性传输发送请求队列顶部 (Top of the periodic transmit request queue)

指示周期性传输 Tx 请求队列中 MAC 当前正在处理的项。

该寄存器用于调试。

位 31：奇数/偶数帧 (Odd/Even frame)

0：以偶数帧发送

1：以奇数帧发送

位 30:27：通道/端点编号 (Channel/endpoint number)

位 26:25：类型 (Type)

00：IN/OUT

01：零长度数据包

11：禁止通道命令

位 24：结束（所选通道/端点的最后一个条目）
(Terminate (last entry for the selected channel/endpoint))

位 23:16 **PTXQSAV**：周期性传输发送请求队列可用空间 (Periodic transmit request queue space available)

指示可供写入的周期性传输发送请求队列的空闲位置的数量。该队列既包含 IN 请求，又包含 OUT 请求。

00：周期性发送请求队列已满

01：1 个位置可用

10：2 个位置可用

bxn：n 个位置可用 (0 ≤ n ≤ 8)

其它值：保留

位 15:0 **PTXFSAVL**：周期性传输发送数据 FIFO 可用空间 (Periodic transmit data FIFO space available)
指示可供写入的周期性传输 Tx FIFO 的空闲位置的数量。
以 32 位字为单位
0000：周期性 Tx FIFO 已满
0001：1 个字可用
0010：2 个字可用
bxn：n 个字可用（其中 0 ≤ n ≤ PTXFD）
其它值：保留

57.14.23 OTG 主机全体通道中断寄存器 (OTG_HAINT)

OTG Host all channels interrupt register

偏移地址：0x414

复位值：0x0000 000

当通道上有事件发生时，主机全体通道中断寄存器会使用模块中断寄存器中的主机通道中断位（OTG_GINTSTS 中的 HCINT 位）中断应用程序。相关内容如*图 746* 所示。每个通道对应 1 个中断位，最多有 16 个位。当应用程序通过相应主机通道 x 中断寄存器清零中断时，该寄存器中的位也会清零。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HAINT															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:16 保留，必须保持复位值。

位 15:0 **HAINT**：通道中断 (Channel interrupts)
每个通道一位：通道 0 对应位 0，通道 15 对应位 15

57.14.24 OTG 主机全体通道中断屏蔽寄存器 (OTG_HAINTMSK)

OTG Host all channels interrupt mask register

偏移地址：0x418

复位值：0x0000 0000

主机全体通道中断屏蔽寄存器与主机全体通道中断寄存器结合使用，进而在通道上发生事件时中断应用程序。每个通道对应 1 个中断屏蔽位，最多有 16 个位。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HAINTM															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:16 保留，必须保持复位值。

位 15:0 **HAINTM**：通道中断屏蔽 (Channel interrupt mask)
0：屏蔽中断
1：解除屏蔽中断
每个通道一位：通道 0 对应位 0，通道 15 对应位 15

57.14.25 OTG 主机端口控制和状态寄存器 (OTG_HPRT)

OTG Host port control and status register

偏移地址：0x440

复位值：0x0000 0000

该寄存器仅在主机模式下可用。当前，OTG 主机仅支持一个端口。

该寄存器包含与 USB 端口相关的信息，例如 USB 复位、使能、挂起、恢复、连接状态以及测试模式。相关内容在*图 746* 中进行了说明。该寄存器中的 rc_w1 位可通过模块中断寄存器中的主机端口中断位（OTG_GINTSTS 中的 HPRTINT 位）触发应用程序中断。发生端口中断时，应用程序必须读取该寄存器，并将引起中断的位清零。对于 rc_w1 位，应用程序必须向该位写入 1 以清除该中断。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PSPD		PTCTL
													r	r	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PTCTL			PPWR	PLSTS		Res.	PRST	PSUSP	PRES	POC CHNG	POCA	PEN CHNG	PENA	PCDET	PCSTS
rw	rw	rw	rw	r	r		rw	rs	rw	rc_w1	r	rc_w1	rc_w1	rc_w1	r

位 31:19 保留，必须保持复位值。

位 18:17 **PSPD**：端口速度 (Port speed)
指示连接到该端口的设备的速度。
01：全速
10：低速
11：保留
00：高速

位 16:13 **PTCTL**：端口测试控制 (Port test control)
应用程序向该字段写入一个非零值，以将端口置于测试模式，同时端口上会产生对应模式的信号。
0000：禁止测试模式
0001：Test_J 模式
0010：Test_K 模式
0011：Test_SE0_NAK 模式
0100：Test_Packet 模式
0101：Test_Force_Enable
其它值：保留

位 12 **PPWR**：端口电源 (Port power)

应用程序使用该字段控制该端口的电源，且发生过流情况时，模块会将该位清零。

0：掉电

1：上电

位 11:10 **PLSTS**：端口线状态 (Port line status)

指示 USB 数据线的当前逻辑电平

位 10：OTG_DP 的逻辑电平

位 11：OTG_DM 的逻辑电平

位 9 保留，必须保持复位值。

位 8 **PRST**：端口复位 (Port reset)

应用程序将该位置 1 时，会在该端口上启动复位序列。应用程序必须为复位周期定时，并在复位序列完成后将该位清零。

0：端口未处于复位状态

1：端口处于复位状态

应用程序必须将该位置 1 并最少保持 10 ms，以在端口上启动复位。除所需的最少持续时间之外，在将该位清零前，应用程序可将该位置 1 的状态再保持 10 ms，即使 USB 标准并没有设置最大限制。

高速：50 ms

全速/低速：10 ms

位 7 **PSUSP**：端口挂起 (Port suspend)

应用程序将此位置 1 以将此端口置于挂起模式。只有此位置 1 时，模块才会停止发送 SOF。要停止 PHY 时钟，应用程序必须将端口时钟停止位置 1，这会使能 PHY 的挂起输入引脚。

此位的读取值反映该端口的当前挂起状态。检测到远程唤醒信号，或者应用程序将此寄存器中的端口复位位或端口恢复位置 1 后，模块可将此位清零；或应用程序将模块中断寄存器中的检测到恢复/远程唤醒中断位或检测到断开连接中断位（分别为 OTG_GINTSTS 中的 WKUINT 或 DISCINT）置 1，模块也可将此位清零。

0：端口未处于挂起模式

1：端口处于挂起模式

位 6 **PRES**：端口恢复 (Port resume)

应用程序将此位置 1 以在该端口上驱动恢复信号。模块会持续驱动恢复信号直到应用程序将此位清零。

如模块中断寄存器中的检测到端口恢复/远程唤醒中断位（OTG_GINTSTS 中的 WKUINT 位）指示，如果模块检测到 USB 远程唤醒序列，则开始驱动恢复信号，而无需应用程序进行干预；如果模块检测到断开连接的情况，则将此位清零。此位的读取值指示当前模块是否正在驱动恢复信号。

0：不驱动恢复信号

1：驱动恢复信号

当 LPM 已使能且模块处于 L1 状态时，此位的影响如下：

1. 应用程序将此位置 1 以在该端口上驱动恢复信号。

2. 模块继续驱动恢复信号，直至达到 OTG_GLPMCFG 寄存器的 BESLTHRS[3:0] 字段预定的时间。

3. 如模块中断寄存器中的“检测到端口 L1 恢复/远程 L1 唤醒中断”位（OTG_GINTSTS 中的 WKUPINT 位）指示，如果模块检测到 USB 远程唤醒序列，则开始驱动恢复信号，而无需应用程序进行干预，并在恢复结束时，将此位清零。此位可以由模块和应用程序置 1 和清零。即使主机未连接任何设备，此位也可由模块清零。

位 5 **POCCHNG**：端口过流变化 (Port overcurrent change)

该寄存器中端口过流激活位（位 4）状态发生变化时，模块将此位置 1。

位 4 **POCA**：端口过流激活 (Port overcurrent active)

此位指示端口的过流状况。

0：无过流状况

1：有过流状况

位 3 **PENCHNG**：端口使能/禁止变化 (Port enable/disable change)

该寄存器中的端口使能位 2 的状态发生变化时，模块将此位置 1。

位 2 **PENA**：端口使能 (Port enable)

端口执行复位序列后，只能由模块使能，并且可以由过流状况、断开连接状况或应用程序将此位清零来禁止。应用程序无法通过对寄存器执行写操作将此位置 1。只能将此位清零来禁止端口。对此位的操作不会触发应用程序的任何中断。

0：禁止端口

1：使能端口

位 1 **PCDET**：检测到端口连接 (Port connect detected)

当检测到设备连接时，模块将此位置 1，以使用模块中断寄存器中的主机端口中断位（OTG_GINTSTS 中的 HPRTINT 位）触发应用程序的中断。应用程序必须将此位置 1 才可清除该中断。

位 0 **PCSTS**：端口连接状态 (Port connect status)

0：端口未连接设备

1：端口已连接设备

57.14.26 OTG 主机通道 x 特性寄存器 (OTG_HCCHARx)（x = 0..15，其中 x = 通道编号）

OTG Host channel-x characteristics register

偏移地址：0x500 + (通道编号 × 0x20)

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CHENA	CHDIS	ODD FRM	DAD							MCNT		EPTYP		LSDEV	Res.
rs	rs	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EPDIR	EPNUM				MPSIZ										
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31 **CHENA**：通道使能 (Channel enable)

此字段由应用程序软件置 1，并由 OTG 主机硬件清零。

0：禁止通道

1：使能通道

位 30 **CHDIS**：禁止通道 (Channel disable)

应用程序将此位置 1 以停止通过通道发送/接收数据，即使通过该通道的传输还未完成，停止操作仍然生效。应用程序必须等待禁止通道的中断以确认通道已经被禁止。

位 29 **ODDFRM**：奇数帧 (Odd frame)
此字段由应用程序置位或复位，以分别指示 OTG 主机必须传输奇数帧或偶数帧。此字段只适用于周期性（同步和中断）事务。
0：偶数帧
1：奇数帧

位 28:22 **DAD**：设备地址 (Device Address)
此字段用于指定要与该主机通信的特定设备。

位 21:20 **MCNT**：多重计数 (Multicount)
此字段向主机指示该周期性端点每帧必须执行的事务数。该字段不用于非周期性传输。
00：保留。对该字段的操作会产生不明确的结果
01：1 个事务
10：该端点每帧需要发出 2 个事务
11：该端点每帧需要发出 3 个事务
注： 此字段至少须置为 01。

位 19:18 **EPTYP**：端点类型 (Endpoint type)
指示选择的传输类型。
00：控制
01：同步
10：批量
11：中断

位 17 **LSDEV**：低速设备 (Low-speed device)
此字段由应用程序置 1，表示此通道正在与一个低速设备进行通信。

位 16 保留，必须保持复位值。

位 15 **EPDIR**：端点方向 (Endpoint direction)
指示通信事务的方向是输入还是输出。
0：OUT
1：IN

位 14:11 **EPNUM**：端点编号 (Endpoint number)
指示要与该主机通道通信的 USB 设备的端点号。

位 10:0 **MPSIZ**：最大数据包大小 (Maximum packet size)
指示与该主机通道通信的设备端点的最大数据包大小。

57.14.27 OTG 主机通道 x 分离控制寄存器 (OTG_HCSPLTx)（x = 0..15，其中 x = 通道编号）

OTG Host channel-x split control register

偏移地址：0x504 + (通道编号 × 0x20)

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SPLIT EN	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	COMP LSPLT
rw															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
XACTPOS		HUBADDR							PRTADDR						
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31 **SPLITEN**：分离使能 (Split enable)
应用程序将此位置 1 时，指示允许该通道执行分离通信事务。

位 30:17 保留，必须保持复位值。

位 16 **COMPLSPLT**：执行完成分离 (Do complete split)
应用程序将此位置 1 时，可请求 OTG 主机执行完成分离通信事务。

位 15:14 **XACTPOS**：事务位置 (Transaction position)
此字段用于决定随各 OUT 事务发送全部、第一个、中间还是最后一个有效负载。
11：全部。指此事务的全部数据有效负载（小于或等于 188 字节）
10：起始。指此事务的第一个数据有效负载（大于 188 字节）
00：中间。指此事务的中间有效负载（大于 188 字节）
01：结尾。指此事务的最后一个有效负载（大于 188 字节）

位 13:7 **HUBADDR**：集线器地址 (Hub address)
此字段存储事务转发器的集线器设备地址。

位 6:0 **PRTADDR**：端口地址 (Port address)
此字段是接收方事务转发器的端口编号。

57.14.28 OTG 主机通道 x 中断寄存器 (OTG_HCINTx)（x = 0..15，其中 x = 通道编号）

OTG Host channel-x interrupt register

偏移地址：0x508 + (通道编号 × 0x20)

复位值：0x0000 0000

该寄存器指示在出现 USB 和 AHB 相关事件时通道的状态。相关内容在*图 746* 中进行了说明。当模块中断寄存器中的主机通道中断位（OTG_GINTSTS 中的 HCINT 位）置 1 时，应用程序必须读取该寄存器。在对寄存器执行读操作之前，应用程序必须先读取主机全体通道中断 (OTG_HAINT) 寄存器，以获取主机通道 x 中断寄存器的准确通道编号。应用程序必须将该寄存器中的相应位清零，才能将 OTG_HAINT 和 OTG_GINTSTS 寄存器中的对应位清零。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	DTERR	FRMOR	BBERR	TXERR	NYET	ACK	NAK	STALL	AHBERR	CHH	XFRC
					rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1

位 31:11 保留，必须保持复位值。

位 10 **DTERR**：数据翻转错误 (Data toggle error)

位 9 **FRMOR**：帧溢出 (Frame overrun)

位 8 **BBERR**：串扰错误 (Babble error)

位 7 **TXERR**：通信事务错误 (Transaction error)
指示 USB 上发生下列错误之一：
CRC 校验失败
超时
位填充错误
错误的 EOP

位 6 **NYET**：尚未就绪响应已接收中断 (Not yet ready response received interrupt)

位 5 **ACK**：收到/发出 ACK 响应中断 (ACK response received/transmitted interrupt)

位 4 **NAK**：收到 NAK 响应中断 (NAK response received interrupt)

位 3 **STALL**：收到 STALL 响应中断 (STALL response received interrupt)

位 2 **AHBERR**：AHB 错误 (AHB error)
仅当处于内部 DMA 模式下且 AHB 读/写操作期间发生 AHB 错误时才生成此错误。应用程序可读取相应的 DMA 通道地址寄存器来获取错误地址。

位 1 **CHH**：通道停止 (Channel halted)
因任意 USB 事务错误或为响应应用程序的禁止请求而导致传输非正常结束。

位 0 **XFRC**：传输完成 (Transfer completed)
未出现任何错误，正常完成传输。

57.14.29 OTG 主机通道 x 中断屏蔽寄存器 (OTG_HCINTMSKx)（x = 0..15，其中 x = 通道编号）

OTG Host channel-x interrupt mask register

偏移地址：0x50C + (通道编号 × 0x20)

复位值：0x0000 0000

此寄存器反映了先前部分中介绍的各通道状态的屏蔽情况。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	DTERRM	FRMORM	BBERRM	TXERRM	NYET	ACKM	NAKM	STALLM	AHBERRM	CHHM	XFRCM
					rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:11 保留，必须保持复位值。

位 10 **DTERRM**：数据翻转错误屏蔽 (Data toggle error mask)
0：屏蔽中断
1：解除屏蔽中断

位 9 **FRMORM**：帧溢出屏蔽 (Frame overrun mask)
0：屏蔽中断
1：解除屏蔽中断

位 8 **BBERRM**：串扰错误屏蔽 (Babble error mask)
0：屏蔽中断
1：解除屏蔽中断

位 7 **TXERRM**：通信事务错误屏蔽 (Transaction error mask)
0：屏蔽中断
1：解除屏蔽中断

位 6 NYET：响应接收中断屏蔽 (response received interrupt mask)
0：屏蔽中断
1：解除屏蔽中断

位 5 **ACKM**：接收/发送 ACK 响应中断屏蔽 (ACK response received/transmitted interrupt mask)
0：屏蔽中断
1：解除屏蔽中断

位 4 **NAKM**：NAK 响应接收中断屏蔽 (NAK response received interrupt mask)
0：屏蔽中断
1：解除屏蔽中断

位 3 **STALLM**：STALL 响应接收中断屏蔽 (STALL response received interrupt mask)
0：屏蔽中断
1：解除屏蔽中断

位 2 **AHBERR**：AHB 错误 (AHB error)
0：屏蔽中断
1：解除屏蔽中断

位 1 **CHHM**：通道停止中断屏蔽 (Channel halted mask)
0：屏蔽中断
1：解除屏蔽中断

位 0 **XFRCM**：传输完成中断屏蔽 (Transfer completed mask)
0：屏蔽中断
1：解除屏蔽中断

57.14.30 OTG 主机通道 x 传输大小寄存器 (OTG_HCTSIZx)（x = 0..15，其中 x = 通道编号）

OTG Host channel-x transfer size register

偏移地址：0x510 + (通道编号 × 0x20)

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	DPID		PKTCNT										XFRSIZ		
	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
XFRSIZ															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31 保留，必须保持复位值。

位 30:29 **DPID**：数据 PID (Data PID)

应用程序在此字段设置数据通信的初始同步 PID。后续传输的数据 PID 由主机硬件维护。

00：DATA0

01：DATA2

10：DATA1

11：SETUP（控制传输）/MDATA（非控制传输）

位 28:19 **PKTCNT**：数据包计数 (Packet count)

应用程序在此字段中设置将要发送 (OUT) 或接收 (IN) 的数据包数。

主机每成功发送或接收一个 OUT/IN 数据包便递减一次计数值。此值达到 0 后，将中断应用程序来指示操作正常完成。

位 18:0 **XFRSIZ**：传输大小 (Transfer size)

对于 OUT 操作，此字段为传输期间主机发送的数据字节数。

对于 IN 操作，此字段为应用程序保留给传输的缓冲区大小。对于 IN 事务（周期性和非周期性），应用程序会将此字段编程为最大数据包大小的整数倍。

57.14.31 OTG 主机通道 x DMA 地址寄存器 (OTG_HCDMAx)（x = 0..15，其中 x = 通道编号）

OTG Host channel-x DMA address register

偏移地址：0x514 + (通道编号 × 0x20)

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DMAADDR															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DMAADDR															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:0 **DMAADDR**：DMA 地址 (DMA address)

此字段存储主机从设备端点获取数据或向设备端点发送数据所用的内存的起始地址。每次进行 AHB 传输，该寄存器都会递增。

57.14.32 设备模式寄存器

Device-mode registers

模块每次切换到设备模式时都必须对这些寄存器进行编程。

57.14.33 OTG 设备配置寄存器 (OTG_DCFG)

OTG device configuration register

偏移地址：0x800

复位值：0x0220 0000

此寄存器在上电、执行某些控制命令或枚举后，会将模块配置为设备模式。请勿在初始编程后更改该寄存器。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	PERSCHIVL		Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
						rw	rw								
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ERRATIM	XCVRDLY	Res.	PFIVL		DAD							Res.	NZLSOHSK	DSPD	
rw	rw		rw	rw	rw	rw	rw	rw	rw	rw	rw		rw	rw	rw

位 31:26 保留，必须保持复位值

位 25:24 **PERSCHIVL**：周期性调度间隔 (Periodic schedule interval)

此字段指定内部 DMA 引擎必须为获取周期性 IN 端点数据分配的时间。根据周期性端点数量的不同，必须将此值指定为 25%、50% 或 75% 的（微）帧。

- 只要存在活动的周期性端点，内部 DMA 引擎便会为获取周期性 IN 端点数据分配指定的时间
- 没有任何活动周期性端点时，内部 DMA 引擎将仅用于非周期性端点并忽略此字段
- 经过一（微）帧中指定的时间后，DMA 切换为获取非周期性端点上的传输数据

00：25%（微）帧
01：50%（微）帧
10：75%（微）帧
11：保留

位 23:16 保留，必须保持复位值

位 15 **ERRATIM**：不定错误中断屏蔽 (Erratic error interrupt mask)

1：发送不定错误时不触发早期挂起中断
0：发生不定错误时生成早期挂起中断

位 14 **XCVRDLY**：收发器延时 (Transceiver delay)

使能或禁止器件啁啾期间 ULPI 时序的延迟。

0：禁止延时（使用默认时序）
1：使能默认时序的延时（对于一些 ULPI PHY，必须要使能延时）

位 13 保留，必须保持复位值。

位 12:11 **PFIVL**：周期性帧间隔 (Periodic frame interval)

指示一帧内必须使用周期性帧中断通知应用程序的时间点。此功能可用于确定该帧的所有同步通信是否完成。

00：80% 帧间隔

01：85% 帧间隔

10：90% 帧间隔

11：95% 帧间隔

位 10:4 **DAD**：设备地址 (Device Address)

应用程序必须在执行每个 SetAddress 控制命令后根据命令参数对该字段进行设置。

位 3 保留，必须保持复位值。

位 2 **NZLSOHSK**：非零长度状态 OUT 握手信号 (Non-zero-length status OUT handshake)

在控制传输状态阶段的 OUT 事务期间，当模块收到非零长度数据包后，应用程序可以使用此字段选择要发送的握手信号。

1：收到非零长度状态 OUT 事务时，回复 STALL 握手信号，收到的 OUT 数据包不发送给应用程序。

0：将收到的 OUT 数据包（零长度或非零长度）发送给应用程序，并基于设备端点控制寄存器中端点的 NAK 和 STALL 位回复握手信号。

位 1:0 **DSPD**：设备速度 (Device speed)

指示应用程序要求模块进行枚举所采用的速度，或应用程序支持的最大速度。但是，实际总线速度只有在完成 chirp 序列后才能确定，同时此速度基于与模块连接的 USB 主机的速度。

00：高速

01：使用 HS 进行全速通信

10：保留

11：使用内置 FS PHY 进行全速通信

57.14.34 OTG 设备控制寄存器 (OTG_DCTL)

OTG device control register

偏移地址：0x804

复位值：0x0000 0002

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DS BESL RJCT	Res.	Res.
													rw		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	PO PRG DNE	CGO NAK	SGO NAK	CGI NAK	SGI NAK	TCTL			GON STS	GIN STS	SDIS	RWU SIG
				rw	w	w	w	w	rw	rw	rw	r	r	rw	rw

位 31:19 保留，必须保持复位值。

位 18 **DSBESLRJCT**：深度睡眠 BESL 拒绝 (Deep sleep BESL reject)

模块拒绝 BESL 值大于编程的 BESL 阈值的 LPM 请求。将为 BESL 值大于 BESL 阈值的 LPM 令牌发送 NYET 响应。默认情况下，禁用深度睡眠 BESL 拒绝功能。

位 17:12 保留，必须保持复位值。

位 11 **POPRGDNE**：上电编程完成 (Power-on programming done)
应用程序使用此位指示寄存器从掉电模式唤醒后已完成编程。

位 10 **CGONAK**：将全局 OUT NAK 清零 (Clear global OUT NAK)
对此位执行写操作会将全局 OUT NAK 清零。

位 9 **SGONAK**：将全局 OUT NAK 置 1 (Set global OUT NAK)
对此位执行写操作会将全局 OUT NAK 置 1。
应用程序使用此位在所有 OUT 端点发送 NAK 握手信号。
应用程序只有确定模块中断寄存器中的全局 OUT NAK 有效位（OTG_GINTSTS 中 GONAKEFF 位）已清零时，才可以将此位置 1。

位 8 **CGINAK**：将全局 IN NAK 清零 (Clear global IN NAK)
对此位执行写操作会将全局 IN NAK 清零。

位 7 **SGINAK**：将全局 IN NAK 置 1 (Set global IN NAK)
对此字段执行写操作会将全局非周期性 IN NAK 置 1。应用程序使用此位使所有非周期性 IN 端点发送 NAK 握手信号。
应用程序只有确定模块中断寄存器中的全局 IN NAK 有效位（OTG_GINTSTS 中的 GINAKEFF 位）已清零时，才可以将此位置 1。

位 6:4 **TCTL**：测试控制 (Test control)
000：禁止测试模式
001：Test_J 模式
010：Test_K 模式
011：Test_SE0_NAK 模式
100：Test_Packet 模式
101：Test_Force_Enable
其它值：保留

位 3 **GONSTS**：全局 OUT NAK 状态 (Global OUT NAK status)
0：将根据 FIFO 状态和 NAK 和 STALL 位设置发送握手信号。
1：无论 Rx FIFO 中是否还有空闲空间都不接收数据。除 SETUP 事务之外，对所有收到的数据包回复 NAK 握手信号。所有同步类型的 OUT 数据包都将被丢弃。

位 2 **GINSTS**：全局 IN NAK 状态 (Global IN NAK status)
0：根据发送 FIFO 中是否有待发送的数据回复握手信号。
1：使所有非周期性 IN 端点回复 NAK 握手信号，无需考虑发送 FIFO 中是否有待发送的数据。

位 1 **SDIS**：软断开 (Soft disconnect)
应用程序使用该位向 USB OTG 模块发出执行软断开的信号。该位置 1 时，主机不会看到设备已连接，且该设备也不会接收 USB 上的信号。在应用程序将此位清零之前，模块会保持断开状态。
0：正常工作。此位在软断开之后清零，会使主机收到设备已连接的事件。重新连接设备之后，USB 主机会重新启动设备枚举。
1：模块使 USB 主机收到设备断开连接的事件。

位 0 **RWUSIG**：发送远程唤醒信号 (Remote wakeup signaling)
应用程序将此位置 1 时，模块会启动远程发送信号，以唤醒 USB 主机。应用程序必须将此位置 1 以使模块退出挂起状态。根据 USB 2.0 规范，应用程序必须在将此位置 1 之后的 1 ms 到 15 ms 内将其清零。
如果 LPM 已使能并且模块处于 L1（睡眠）状态，则当应用程序将此位置 1 时，模块会启动 L1 远程信号来唤醒 USB 主机。应用程序必须将此位置 1 以使模块退出睡眠状态。按照 LPM 规范中的规定，在应用程序将此位置 1 后经过 50 µs ($T_{L1DevDrvResume}$)，硬件会自动将此位清零。当前一个 LPM 事务中的 bRemoteWake 为零时（请参见 GLPMCFG 寄存器中的 REMWAKE 位），应用程序不得将此位置 1。

表 488 显示了为使 USB 主机检测到设备断开连接所需的软断开 (SDIS) 位置 1 的最短时间（取决于设备状态）。为了协调时钟抖动，建议应用程序在指定的最小时间基础上再加入一段延迟。

表 488. 软断开的最小时间

运行速度	设备状态	最小时间
全速	挂起	1 ms + 2.5 µs
全速	空闲	2.5 µs
全速	非空闲或挂起（正在通信时）	2.5 µs
高速	非空闲或挂起（正在通信时）	125 µs

57.14.35 OTG 设备状态寄存器 (OTG_DSTS)

OTG device status register

偏移地址：0x808

复位值：0x0000 0010

此寄存器指示模块在出现 USB 相关事件时的状态。必须在发生设备全体中断寄存器 (OTG_DAINT) 中断时读取它。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DEVLNSTS		FNSOF					
								r		r	r	r	r	r	r

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FNSOF								Res.	Res.	Res.	Res.	EERR	ENUMSPD		SUSP STS
r	r	r	r	r	r	r	r					r	r	r	r

位 31:24 保留，必须保持复位值。

位 23:22 **DEVLNSTS**：设备线状态 (Device line status)

指示 USB 数据线的当前逻辑电平。

位 [23]：D+ 的逻辑电平

位 [22]：D- 的逻辑电平

位 21:8 **FNSOF**：接收 SOF 的帧编号 (Frame number of the received SOF)

位 7:4 保留，必须保持复位值。

位 3 **EERR**：不定错误 (Erratic error)

模块将该位置 1 以报告任何不定错误。

由于不定错误，OTG_FS/OTG_HS 控制器会进入挂起状态，并且会以 OTG_GINTSTS 寄存器的早期挂起位（OTG_GINTSTS 中的 ESUSP 位）为应用程序生成一个中断。如果早期挂起中断是由不定错误触发，则应用程序只能执行软断开以恢复通信。

位 2:1 **ENUMSPD**：枚举速度 (Enumerated speed)

指示 OTG_HS 控制器通过 chirp 序列检测速度后被枚举成的速度。

01：保留

10：保留

11：全速（PHY 时钟运行频率为 48 MHz）

其它值：保留

位 0 **SUSPSTS**：挂起状态 (Suspend status)

在设备模式下，只要在 USB 上检测到挂起状态，该位就会置 1。当 USB 数据线上的空闲状态保持 3 ms，模块便会进入挂起状态。出现以下情况时，模块会退出挂起状态：

– USB 数据线上有活动

– 应用程序对 OTG_DCTL 寄存器的远程唤醒信号位（OTG_DCTL 中的 RWUSIG 位）执行写操作。

57.14.36 OTG 设备 IN 端点通用中断屏蔽寄存器 (OTG_DIEPMSK)

OTG device IN endpoint common interrupt mask register

偏移地址：0x810

复位值：0x0000 0000

此寄存器与全体端点的各个 OTG_DIEPINTx 寄存器配合使用，以便在每个 IN 端点上生成中断。通过对此寄存器的相应位执行写操作，可屏蔽 OTG_DIEPINTx 寄存器中特定状态的 IN 端点中断。默认情况下，状态中断都被屏蔽。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
								/							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	NAKM	Res.	Res.	Res.	BMA	TXFURM	Res.	INEPNEM	INEPNMM	ITTXFEMSK	TOM	Res.	EPDM	XFRCM
		rw				rw	rw		rw	rw	rw	rw		rw	rw

位 31:14 保留，必须保持复位值。

位 13 **NAKM**：NAK 中断屏蔽 (NAK interrupt mask)

0：屏蔽中断

1：解除屏蔽中断

位 12:10 保留，必须保持复位值

位 9 **BIM**：BNA 中断屏蔽 (BNA interrupt mask)

0：屏蔽中断

1：解除屏蔽中断

位 8 **TXFURM**：FIFO 下溢中断屏蔽 (FIFO underrun mask)
0：屏蔽中断
1：解除屏蔽中断

位 7 保留，必须保持复位值

位 6 **INEPNEM**：IN 端点 NAK 有效中断屏蔽 (IN endpoint NAK effective mask)
0：屏蔽中断
1：解除屏蔽中断

位 5 **INEPNMM**：EP 不匹配时接收到 IN 令牌中断屏蔽 (IN token received with EP mismatch mask)
0：屏蔽中断
1：解除屏蔽中断

位 4 **ITTXFEMSK**：Tx FIFO 为空时接收到 IN 令牌中断屏蔽 (IN token received when Tx FIFO empty mask)
0：屏蔽中断
1：解除屏蔽中断

位 3 **TOM**：超时中断屏蔽（非同步端点）(Timeout condition mask (Non-isochronous endpoints))
0：屏蔽中断
1：解除屏蔽中断

位 2 保留，必须保持复位值。

位 1 **EPDM**：端点禁止中断屏蔽 (Endpoint disabled interrupt mask)
0：屏蔽中断
1：解除屏蔽中断

位 0 **XFRCM**：传输完成中断屏蔽 (Transfer completed interrupt mask)
0：屏蔽中断
1：解除屏蔽中断

57.14.37 OTG 设备 OUT 端点通用中断屏蔽寄存器 (OTG_DOEPMSK)

OTG device OUT endpoint common interrupt mask register

偏移地址：0x814

复位值：0x0000 0000

此寄存器与所有端点的各个 OTG_DOEPINTx 寄存器配合使用，以便可以在每个 OUT 端点上生成中断。通过对此寄存器的相应位执行写操作，可屏蔽 OTG_DOEPINTx 寄存器中特定状态的 OUT 端点中断。默认情况下，状态中断都被屏蔽。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	NYETMSK	Res.	Res.	Res.	Res.	BOIM	TXFURM	Res.	B2BSTUP	Res.	OTEPDM	STUPM	Res.	EPDM	XFRCM
	rw					rw	rw		rw		rw	rw		rw	rw

位 31:15 保留，必须保持复位值

位 14 **NYET**：NYET 中断屏蔽 (NYET interrupt mask)
0：屏蔽中断
1：解除屏蔽中断

位 13:10 保留，必须保持复位值。

位 9 **BOIM**：BNA 中断屏蔽 (BNA interrupt mask)
0：屏蔽中断
1：解除屏蔽中断

位 8 **TXFURM**：FIFO 下溢中断屏蔽 (FIFO underrun mask)
0：屏蔽中断
1：解除屏蔽中断

位 7 保留，必须保持复位值

位 6 **B2BSTUP**：接收到连续的 SETUP 数据包中断屏蔽(Back-to-back SETUP packets received mask)。仅适用于控制 OUT 端点。
0：屏蔽中断
1：解除屏蔽中断

位 5 保留，必须保持复位值。

位 4 **OTEPDM**：端点禁止时接收到 OUT 令牌中断屏蔽(OUT token received when endpoint disabled mask)。仅适用于控制 OUT 端点。
0：屏蔽中断
1：解除屏蔽中断

位 3 **STUPM**：SETUP 阶段完成中断屏蔽 (SETUP phase done mask)。仅适用于控制端点。
0：屏蔽中断
1：解除屏蔽中断

位 2 保留，必须保持复位值。

位 1 **EPDM**：端点禁止中断屏蔽 (Endpoint disabled interrupt mask)
0：屏蔽中断
1：解除屏蔽中断

位 0 **XFRCM**：传输完成中断屏蔽 (Transfer completed interrupt mask)
0：屏蔽中断
1：解除屏蔽中断

57.14.38 OTG 设备全体端点中断寄存器 (OTG_HAINT)

OTG device all endpoints interrupt register

偏移地址：0x818

复位值：0x0000 0000

当端点上发生有效事件时，OTG_DAINT 寄存器将通过 OTG_GINTSTS 寄存器中的设备 OUT 端点中断位或设备 IN 端点中断位（分别为 OTG_GINTSTS 中的 OEPINT 或 IEPINT 位）来中断应用程序。每个端点对应一个中断位，OUT 端点和 IN 端点均最多有 16 个中断位。双向端点将使用相应的 IN 和 OUT 中断位。当应用程序将相应设备端点 x 中断寄存器 (OTG_DIEPINTx/OTG_DOEPINTx) 中的位置 1 和清零时，此寄存器中的相应位也将置 1 和清零。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
OEPINT															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IEPINT															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:16 **OEPINT**：OUT 端点中断位 (OUT endpoint interrupt bits)
每个 OUT 端点对应一位：
OUT 端点 0 对应位 16，而 OUT 端点 3 对应位 19。

位 15:0 **IEPINT**：IN 端点中断位 (IN endpoint interrupt bits)
每个 IN 端点对应一位：
IN 端点 0 对应位 0，而 IN 端点 3 对应位 3。

57.14.39 OTG 全体端点中断屏蔽寄存器 (OTG_DAINTMSK)

OTG all endpoints interrupt mask register

偏移地址：0x81C

复位值：0x0000 0000

OTG_DAINTMSK 寄存器与设备端点中断寄存器结合使用，在设备端点上发生事件时中断应用程序。但是，与该中断相对应的 OTG_DAINT 寄存器位仍会置 1。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
OEPM															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IEPM															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:16 **OEPM**：OUT EP 中断屏蔽位 (OUT EP interrupt mask bits)
每个 OUT 端点对应一位：
OUT EP 0 对应位 16，而 OUT EP 3 对应位 19
0：屏蔽中断
1：解除屏蔽中断

位 15:0 **IEPM**：IN EP 中断屏蔽位 (IN EP interrupt mask bits)
每个 IN 端点对应一位：
IN EP 0 对应位 0，而 IN EP 3 对应位 3
0：屏蔽中断
1：解除屏蔽中断

57.14.40 OTG 设备 V_{BUS} 放电时间寄存器 (OTG_DVBUSDIS)

OTG device V_{BUS} discharge time register

偏移地址：0x0828

复位值：0x0000 17D7

该寄存器指定 SRP 期间 V_{BUS} 发出脉冲之后的 V_{BUS} 放电时间。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VBUSDT															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:16 保留，必须保持复位值。

位 15:0 **VBUSDT**：设备 V_{BUS} 放电时间 (Device V_{BUS} discharge time)
指定 SRP 期间 V_{BUS} 发出脉冲之后的 V_{BUS} 放电时间。该时间值等于：
V_{BUS} 放电时间（PHY 时钟数）/1024
该值可基于不同的 V_{BUS} 负载根据需要进行调整。

57.14.41 OTG 设备 V_{BUS} 脉冲时间寄存器 (OTG_DVBUSPULSE)

OTG device V_{BUS} pulsing time register

偏移地址：0x082C

复位值：0x0000 05B8

该寄存器指定 SRP 期间的 V_{BUS} 脉冲时间。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DVBUSP															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:16 保留，必须保持复位值。

位 15:0 **DVBUSP**：设备 V_{BUS} 脉冲时间 (Device V_{BUS} pulsing time)。该功能仅与 OTG1.3 相关。
指定 SRP 期间的 V_{BUS} 脉冲时间。
V_{BUS} 脉冲时间（PHY 时钟数）/1024

57.14.42 OTG 设备阈值控制寄存器 (OTG_DTHRCTL)

OTG Device threshold control register

偏移地址：0x0830

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	ARPEN	Res.	RXTHRLEN									RXTHREN
				rw		rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	TXTHRLEN									ISOTHREN	NONISOTHREN
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:28 保留，必须保持复位值。

位 27 **ARPEN**：仲裁器驻留使能 (Arbiter parking enable)

该位用于控制内部 DMA 仲裁对 IN 端点的驻留。当阈值使能且该位置 1 后，DMA 会对在 USB 上收到令牌的 IN 端点驻留其仲裁。采用这种方式可以避免出现下溢情况。默认情况下使能驻留。

位 26 保留，必须保持复位值。

位 25:17 **RXTHRLEN**：接收阈值长度 (Receive threshold length)

该字段以双字为单位指定接收阈值的大小。该字段还指定模块在 AHB 上启动传输之前在 USB 上接收的数据量。阈值长度最小值为八个双字。推荐 RXTHRLEN 的值和设定的 AHB 批量传输长度（OTG_GAHBCFG 中的 HBSTLEN 位）相同。

位 16 **RXTHREN**：接收阈值使能 (Receive threshold enable)

该位置 1 时，模块会使能接收方向的阈值。

位 15:11 保留，必须保持复位值。

位 10:2 **TXTHRLEN**：发送阈值长度 (Transmit threshold length)

该字段以双字为单位指定发送阈值的大小。该字段指定模块在 USB 上启动传输之前相应端点发送 FIFO 中的数据量（单位为字节）。阈值长度最小值为八个双字。该字段控制同步和非同步 IN 端点阈值。推荐 TXTHRLEN 的值和设定的 AHB 批量传输长度（OTG_GAHBCFG 中的 HBSTLEN 位）相同。

位 1 **ISOTHREN**：ISO IN 端点阈值使能 (ISO IN endpoint threshold enable)

该位置 1 时，模块会使能同步 IN 端点的阈值。

位 0 **NONISOTHREN**：非同步 IN 端点阈值使能 (Nonisochronous IN endpoints enable)

该位置 1 时，模块会使能非同步 IN 端点的阈值。

57.14.43 OTG 设备 IN 端点 FIFO 空中断屏蔽寄存器 (OTG_DIEPEMPMSK)

OTG device IN endpoint FIFO empty interrupt mask register

偏移地址：0x834

复位值：0x0000 0000

此寄存器用于控制 IN 端点 FIFO 空中断的生成 (TXFE_OTG_DIEPINTx)。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INEPTXFEM															
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

位 31:16 保留，必须保持复位值。

位 15:0 **INEPTXFEM**：IN EP Tx FIFO 空中断屏蔽位 (IN EP Tx FIFO empty interrupt mask bits)
这些位用作 OTG_DIEPINTx 的屏蔽位。
每个位对应一个 IN 端点的 TXFE 中断：
IN 端点 0 对应位 0，而 IN 端点 3 对应位 3。
0：屏蔽中断
1：解除屏蔽中断

57.14.44 OTG 设备单个端点中断寄存器 (OTG_DEACHINT)

OTG device each endpoint interrupt register

偏移地址：0x0838

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OEP1 INT	Res.
														r	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	IEP1 INT	Res.
														r	

位 31:18 保留，必须保持复位值。

位 17 **OEP1INT**：OUT 端点 1 中断位 (OUT endpoint 1 interrupt bit)

位 16:2 保留，必须保持复位值。

位 1 **IEP1INT**：IN 端点 1 中断位 (IN endpoint 1interrupt bit)

位 0 保留，必须保持复位值。

57.14.45 OTG 设备单个端点中断屏蔽寄存器 (OTG_DEACHINTMSK)

OTG device each endpoint interrupt register mask

偏移地址：0x083C

复位值：0x0000 0000

IN 端点 1 对应一个中断位，OUT 端点 1 对应一个中断位。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OEP1INTM	Res.
														rw	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	IEP1INTM	Res.
														rw	

位 31:18 保留，必须保持复位值。

位 17 **OEP1INTM**：OUT 端点 1 中断屏蔽位 (OUT Endpoint 1 interrupt mask bit)

位 16:2 保留，必须保持复位值。

位 1 **IEP1INTM**：IN 端点 1 中断屏蔽位 (IN Endpoint 1 interrupt mask bit)

位 0 保留，必须保持复位值。

57.14.46 OTG 设备端点 x 控制寄存器 (OTG_DIEPCTLx) （x = 0..8，其中 x = 端点编号）

OTG device endpoint-x control register

偏移地址：0x900 + (端点编号 × 0x20)

复位值：0x0000 0000

应用程序使用此寄存器控制各个逻辑端点（端点 0 除外）的行为。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
EPENA	EPDIS	SODDFRM	SD0PID/SEVNFRM	SNAK	CNAK	TXFNUM				STALL	Res.	EPTYP		NAKSTS	EONUM/DPID
rs	rs	w	w	w	w	rw	rw	rw	rw	rw/rs		rw	rw	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
USBAEP	Res.	Res.	Res.	Res.	MPSIZ										
rw					rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31 **EPENA**：端点使能 (Endpoint enable)

应用程序将此位置 1 以在端点上启动数据发送。

在此端点上触发以下任一中断之前，模块会将此位清零：

– SETUP 阶段完成

– 端点禁止

– 传输完成

位 30 **EPDIS**：端点禁止 (Endpoint disable)

即使在该端点上的传送完成之前，应用程序也可将此位置 1，以停止端点上的数据发送/接收。应用程序必须等到发生端点禁止中断后，才能将端点视为禁止端点。在端点禁止中断位置 1 前，模块会将此位清零。只有在该端点的端点使能位置 1 后，应用程序才可将该位置 1。

位 29 **SODDFRM**：设置奇数帧 (Set odd frame)

仅适用于同步 IN 和 OUT 端点。

对此字段进行写操作会将偶数/奇数帧 (EONUM) 字段设置为奇数帧。

位 28 **SD0PID**：设置 DATA0 PID (Set DATA0 PID)

仅适用于中断/批量 IN 端点。

对此字段进行写操作会将此寄存器中的端点数据 PID (DPID) 字段设置为 DATA0。

SEVNFRM：设置偶数帧 (Set even frame)

仅适用于同步 IN 端点。

对此字段进行写操作会将偶数/奇数帧 (EONUM) 字段设置为偶数帧。

位 27 **SNAK**：将 NAK 置 1 (Set NAK)

对此位进行写操作会将端点的 NAK 位置 1。

通过此位，应用程序可以控制端点上 NAK 握手信号的发送。发生传输完成中断时或端点上接收到 SETUP 后，模块也可以将 OUT 端点的这个位置 1。

位 26 **CNAK**：将 NAK 清零 (Clear NAK)

对此位进行写操作会将端点的 NAK 位清零。

位 25:22 **TXFNUM**：Tx FIFO 编号 (Tx FIFO number)

这些位用于指定与此端点相关联的 FIFO 编号。必须为每个有效的 IN 端点设置单独的 FIFO 编号。

此字段仅针对 IN 端点有效。

位 21 **STALL**：STALL 握手 (STALL handshake)

仅适用于非控制、非同步 IN 端点（访问类型为 rw）。

应用程序将此位置 1 使得设备对来自 USB 主机的所有令牌都回复 STALL。如果 NAK 位、全局 IN NAK 或全局 OUT NAK 与此位同时置 1，则 STALL 位优先。只有应用程序能够将此位清零，而模块则不能。

仅适用于控制端点（访问类型为 rs）

此端点接收到 SETUP 令牌时，应用程序只能将此位置 1，而模块会将其清零。如果 NAK 位、全局 IN NAK 或全局 OUT NAK 与此位同时置 1，则 STALL 位优先。无论此位如何设置，模块总是通过 ACK 握手响应 SETUP 数据包。

位 20 保留，必须保持复位值。

位 19:18 **EPTYP**：端点类型 (Endpoint type)

以下是这个逻辑端点支持的传输类型。

00：控制

01：同步

10：批量

11：中断

位 17 **NAKSTS**：NAK 状态 (NAK status)

它指示以下结果：

0：模块根据 FIFO 状态回复非 NAK 握手。

1：模块在此端点上回复 NAK 握手。

当应用程序或模块将此位置 1 时：

对于非同步 IN 端点：即使 Tx FIFO 中存在可用数据，模块也会停止通过 IN 端点发送任何数据。

对于同步 IN 端点：即使 Tx FIFO 中存在可用数据，模块也会发送长度为零的数据包。

无论此位如何设置，模块总是通过 ACK 握手响应 SETUP 数据包。

位 16 **EONUM**：偶数/奇数帧 (Even/odd frame)

仅适用于同步 IN 端点。

指示模块为此端点发送/接收同步的数据所在的帧的编号。应用程序必须通过此寄存器中的 SEVNFRM 和 SODDFRM 字段对偶数/奇数帧编号进行编程，以便此端点发送/接收同步数据。

0：偶数帧

1：奇数帧

DPID：端点数据 PID (Endpoint data PID)

仅适用于中断/批量 IN 端点。

包含此端点上将要接收或发送的数据包的 PID。端点激活后，应用程序必须对要在此端点上接收或发送的首个数据包的 PID 进行编程。应用程序使用 SD0PID 寄存器字段对 DATA0 或 DATA1 PID 进行编程。

0：DATA0

1：DATA1

位 15 **USBAEP**：USB 活动端点 (USB active endpoint)

指示此端点在当前配置和接口中是否激活。检测到 USB 复位后，模块会为所有端点（端点 0 除外）将此位清零。接收到 SetConfiguration 和 SetInterface 命令后，应用程序必须相应地对端点寄存器进行编程并将此位置 1。

位 14:11 保留，必须保持复位值。

位 10:0 **MPSIZ**：最大数据包大小 (Maximum packet size)

应用程序必须将此字段编程为当前逻辑端点的最大数据包大小。此值以字节为单位。

57.14.47 OTG 设备控制 OUT 端点 0 控制寄存器 (OTG_DOEPCTL0)

OTG device control OUT endpoint 0 control register

偏移地址：0xB00

复位值：0x0000 8000

本节介绍 OTG_DOEPCTL0 寄存器。非零控制端点使用编号 1-3 的寄存器。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
EPENA	EPDIS	Res.	Res.	SNAK	CNAK	Res.	Res.	Res.	Res.	STALL	SNPM	EPTYP		NAK STS	Res.
w	r			w	w					rs	rw	r	r	r	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
USBA EP	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MPSIZ	
r														r	r

位 31 **EPENA**：端点使能 (Endpoint enable)

应用程序将此位置 1 以在端点 0 上启动数据发送。

在此端点上触发以下任一中断之前，模块会将此位清零：

– SETUP 阶段完成

– 端点禁止

– 传输完成

位 30 **EPDIS**：端点禁止 (Endpoint disable)

应用程序无法禁止控制 OUT 端点 0。

位 29:28 保留，必须保持复位值。

位 27 **SNAK**：将 NAK 置 1 (Set NAK)

对此位进行写操作会将端点的 NAK 位置 1。

通过此位，应用程序可以控制端点上 NAK 握手信号的发送。发生传输完成中断时或端点上接收到 SETUP 后，模块也可以将此位置 1。

位 26 **CNAK**：将 NAK 清零 (Clear NAK)

对此位进行写操作会将端点的 NAK 位清零。

位 25:22 保留，必须保持复位值。

位 21 **STALL**：STALL 握手 (STALL handshake)

此端点接收到 SETUP 令牌时，应用程序只能将此位置 1，而模块会将其清零。如果 NAK 位、全局 OUT NAK 与此位同时置 1，则 STALL 位优先。无论此位如何设置，模块总是通过 ACK 握手响应 SETUP 数据包。

位 20 **SNPM**：监听模式 (Snoop mode)

此位用于将端点配置为监听模式。在监听模式下，模块不会在将 OUT 数据包传输到应用存储区前检查其是否正确。

位 19:18 **EPTYP**：端点类型 (Endpoint type)

硬件固定为二进制 00，表示端点为控制传输类型。

位 17 **NAKSTS**：NAK 状态 (NAK status)

指示以下结果：

0：模块根据 FIFO 状态回复非 NAK 握手。

1：模块在此端点上回复 NAK 握手。

当应用程序或模块将此位置 1 时，即使 Rx FIFO 中存在空间可继续容纳收到的数据包，模块也会停止接收数据。无论此位如何设置，模块总是通过 ACK 握手响应 SETUP 数据包。

位 16 保留，必须保持复位值。

位 15 **USBAEP**：USB 活动端点 (USB active endpoint)
此位总是置 1，指示在所有配置和接口中控制端点 0 始终处于激活状态。

位 14:2 保留，必须保持复位值。

位 1:0 **MPSIZ**：最大数据包大小 (Maximum packet size)
控制 OUT 端点 0 的最大数据包大小与在控制 IN 端点 0 中进行编程的值相同。
00：64 字节
01：32 字节
10：16 字节
11：8 字节

57.14.48 OTG 设备端点 x 控制寄存器 (OTG_DOEPCTLx)（x = 1..8，其中 x = 端点编号）

OTG device endpoint-x control register

OUT 端点的偏移地址：0xB00 + (端点编号 × 0x20)

复位值：0x0000 0000

应用程序使用此寄存器控制各个逻辑端点（端点 0 除外）的行为。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
EPENA	EPDIS	SD1 PID/ SODD FRM	SD0 PID/ SEVN FRM	SNAK	CNAK	Res.	Res.	Res.	Res.	STALL	SNPM	EPTYP		NAK STS	EO NUM/ DPID
rs	rs	w	w	w	w					rw/rs	rw	rw	rw	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
USBA EP	Res.	Res.	Res.	Res.	MPSIZ										
rw					rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31 **EPENA**：端点使能 (Endpoint enable)
适用于 IN 和 OUT 端点。
应用程序将此位置 1 以在端点上启动数据发送。
在此端点上触发以下任一中断之前，模块会将此位清零：
– SETUP 阶段完成
– 端点禁止
– 传输完成

位 30 **EPDIS**：端点禁止 (Endpoint disable)
即使在该端点上的传送完成之前，应用程序也可将此位置 1，以停止端点上的数据发送/接收。应用程序必须等到发生端点禁止中断后，才能将端点视为禁止端点。在端点禁止中断位置 1 前，模块会将此位清零。只有在该端点的端点使能位置 1 后，应用程序才可将该位置 1。

位 29 **SD1PID**：设置 DATA1 PID (Set DATA1 PID)
仅适用于中断/批量 IN 和 OUT 端点。对此字段进行写操作会将此寄存器中的端点数据 PID (DPID) 字段设置为 DATA1。

SODDFRM：设置奇数帧 (Set odd frame)
仅适用于同步 IN 和 OUT 端点。对此字段进行写操作会将偶数/奇数帧 (EONUM) 字段设置为奇数帧。

位 28 **SD0PID**：设置 DATA0 PID (Set DATA0 PID)

仅适用于中断/批量 OUT 端点。

对此字段进行写操作会将此寄存器中的端点数据 PID (DPID) 字段设置为 DATA0。

SEVNFRM：设置偶数帧 (Set even frame)

仅适用于同步 OUT 端点。

对此字段进行写操作会将偶数/奇数帧 (EONUM) 字段设置为偶数帧。

位 27 **SNAK**：将 NAK 置 1 (Set NAK)

对此位进行写操作会将端点的 NAK 位置 1。

通过此位，应用程序可以控制端点上 NAK 握手信号的发送。发生传输完成中断时或端点上接收到 SETUP 后，模块也可以将 OUT 端点的这个位置 1。

位 26 **CNAK**：将 NAK 清零 (Clear NAK)

对此位进行写操作会将端点的 NAK 位清零。

位 25:22 保留，必须保持复位值。

位 21 **STALL**：STALL 握手 (STALL handshake)

仅适用于非控制、非同步 OUT 端点（访问类型为 rw）。

应用程序将此位置 1 使得设备对来自 USB 主机的所有令牌都回复 STALL。如果 NAK 位、全局 IN NAK 或全局 OUT NAK 与此位同时置 1，则 STALL 位优先。只有应用程序能够将此位清零，而模块则不能。

仅适用于控制端点（访问类型为 rs）。

此端点接收到 SETUP 令牌时，应用程序只能将此位置 1，而模块会将其清零。如果 NAK 位、全局 IN NAK 或全局 OUT NAK 与此位同时置 1，则 STALL 位优先。无论此位如何设置，模块总是通过 ACK 握手响应 SETUP 数据包。

位 20 **SNPM**：监听模式 (Snoop mode)

此位用于将端点配置为监听模式。在监听模式下，模块不会在将 OUT 数据包传输到应用存储区前检查其是否正确。

位 19:18 **EPTYP**：端点类型 (Endpoint type)

以下是这个逻辑端点支持的传输类型。

00：控制

01：同步

10：批量

11：中断

位 17 **NAKSTS**：NAK 状态 (NAK status)

指示以下结果：

0：模块根据 FIFO 状态回复非 NAK 握手。

1：模块在此端点上回复 NAK 握手。

当应用程序或模块将此位置 1 时：

即使 Rx FIFO 存在空间可容纳传入数据包，模块也会停止在 OUT 端点上接收任何数据。

无论此位如何设置，模块总是通过 ACK 握手响应 SETUP 数据包。

位 16 **EONUM**：偶数/奇数帧 (Even/odd frame)

仅适用于同步 IN 和 OUT 端点。

指示模块为此端点发送/接收同步的数据所在的帧的编号。应用程序必须通过此寄存器中的 SEVNFRM 和 SODDFRM 字段对偶数/奇数帧编号进行编程，以便此端点发送/接收同步数据。

0：偶数帧

1：奇数帧

DPID：端点数据 PID (Endpoint data PID)

仅适用于中断/批量 OUT 端点。

包含此端点上将要接收或发送的数据包的 PID。端点激活后，应用程序必须对要在此端点上接收或发送的首个数据包的 PID 进行编程。应用程序使用 SD0PID 寄存器字段对 DATA0 或 DATA1 PID 进行编程。

0：DATA0

1：DATA1

位 15 **USBAEP**：USB 活动端点 (USB active endpoint)

指示此端点在当前配置和接口中是否激活。检测到 USB 复位后，模块会为所有端点（端点 0 除外）将此位清零。接收到 SetConfiguration 和 SetInterface 命令后，应用程序必须相应地对端点寄存器进行编程并将此位置 1。

位 14:11 保留，必须保持复位值。

位 10:0 **MPSIZ**：最大数据包大小 (Maximum packet size)

应用程序必须将此字段编程为当前逻辑端点的最大数据包大小。此值以字节为单位。

57.14.49 OTG 设备端点 x 中断寄存器 (OTG_DIEPINTx)（x = 0..8，其中 x = 端点编号）

OTG device endpoint-x interrupt register

偏移地址：0x908 + (端点编号 × 0x20)

复位值：0x0000 0080

此寄存器指示端点在出现 USB 和 AHB 相关事件时的状态。相关内容在*图 746* 中进行了说明。当模块中断寄存器中的 IN 端点中断位（OTG_GINTSTS 中的 IEPINT 位）置 1 时，应用程序必须读取此寄存器。在应用程序能够读取此寄存器之前，必须先读取设备全体端点中断 (OTG_DAINT) 寄存器，以获取设备端点 x 中断寄存器的准确端点编号。应用程序必须将该寄存器中的相应位清零，才能将 OTG_DAINT 和 OTG_GINTSTS 寄存器中的对应位清零。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	NAK	BERR	PKTD RPSTS	Res.	BNA	TXFIF OUD RN	TXFE	INEP NE	Res.	ITTXFE	TOC	Res.	EP DISD	XFRC
		rc_w1	rc_w1	rc_w1		rc_w1	rc_w1	r	r		rc_w1	rc_w1		rc_w1	rc_w1

位 31:14 保留，必须保持复位值

位 13 **NAK**：USB OTG HS 的 NAK 输入 (NAK input for USB OTG HS)
当设备发出或收到 NAK 时，模块将生成该中断。
如果是同步 IN 端点，由于 Tx FIFO 中无数据可发而发送长度为零的数据包时也会生成该中断。

位 12 **BERR**：Babble 错误中断 (Babble error interrupt)

位 11 **PKTDRPSTS**：数据包丢弃状态 (Packet dropped status)
该位用于向应用程序指示有 ISOC OUT 数据包被丢弃。该位没有相应的中断屏蔽位，也不会生成中断。

位 10 保留，必须保持复位值

位 9 **BNA**：缓冲区不可用中断 (Buffer not available interrupt)
当所访问的描述符没有准备好供模块处理时（例如主机繁忙或 DMA 已完成），模块将生成该中断。

位 8 **TXFIFOUDRN**：发送FIFO 下溢 (TxfifoUndrn) (Transmit Fifo Underrun (TxfifoUndrn))
当模块检测到该端点的发送 FIFO 下溢时，将生成该中断。相关性：该中断仅在使能了阈值时有效

位 7 **TXFE**：发送 FIFO 为空 (Transmit FIFO empty)
当此端点的 Tx FIFO 为半空或全空时，此中断被置位。Tx FIFO 为半空还是全空状态由 OTG_GAHBCFG 寄存器中的 Tx FIFO 空门限位（OTG_GAHBCFG 中的 TXFELVL 位）决定。

位 6 **INEPNE**：IN 端点 NAK 有效 (IN endpoint NAK effective)
当应用程序通过向 OTG_DIEPCTLx 中的 CNAK 位写入数据来将 IN 端点 NAK 清零时，此位可被清零。
该中断指示模块已对（由应用程序或模块）置 1 的 NAK 采样，结果已生效。该中断指示由应用程序置 1 的 IN 端点 NAK 位已在模块中起作用。
此中断不一定表示 USB 上已发送了一个 NAK 握手信号。STALL 位优先级高于 NAK 位。

位 5 保留，必须保持复位值。

位 4 **ITTXFE**：Tx FIFO 为空时接收到 IN 令牌 (IN token received when Tx FIFO is empty)
仅适用于非周期性 IN 端点。
当和该端点相对应的 Tx FIFO（周期性/非周期性）为空时，接收到 IN 令牌，从而产生中断。

位 3 **TOC**：超时条件 (Timeout condition)
仅适用于控制 IN 端点。
指示该端点对最近收到的 IN 令牌响应超时。

位 2 保留，必须保持复位值。

位 1 **EPDISD**：端点禁止中断 (Endpoint disabled interrupt)
此位指示该端点已经由应用程序禁止掉。

位 0 **XFRC**：传输完成中断 (Transfer completed interrupt)
此字段指示在此端点上设置的传输已经在 USB 和 AHB 上传输完成。

57.14.50 OTG 设备端点 x 中断寄存器 (OTG_DIEPINTx)（x = 0..8，其中 x = 端点编号）

OTG device endpoint-x interrupt register

偏移地址：0x908 + (端点编号 × 0x20)

复位值：0x0000 0080

此寄存器指示端点在出现 USB 和 AHB 相关事件时的状态。相关内容在*图 746* 中进行了说明。当 OTG_GINTSTS 寄存器中的 OUT 端点中断位（OTG_GINTSTS 中的 OEPINT 位）置 1 时，应用程序必须读取此寄存器。在应用程序能够读取此寄存器之前，必须先读取 OTG_DAINT 寄存器，以获取 OTG_DOEPINTx 寄存器的准确端点编号。应用程序必须将该寄存器中的相应位清零，才能将 OTG_DAINT 和 OTG_GINTSTS 寄存器中的对应位清零。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STPK TRx	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	B2B STUP	STSPH SRX	OTEP DIS	STUP	Res.	EP DISD	XFRC
rc_w1									rc_w1	rc_w1	rc_w1	rc_w1		rc_w1	rc_w1

位 31:16 保留，必须保持复位值。

位 15 STPKTRX：接收到设置数据包 (Setup packet received)。

仅适用于缓冲区 DMA 模式下的控制 OUT 端点。该位由 OTG_HS 设置，用于指示该缓冲区中有 8 个字节的设置数据。每个缓冲区中只有一个设置数据包。接收到设置数据包时，OTG_HS 会在 Rx FIFO 呈现 SETUP_COMPLETE 状态后关闭缓冲区并禁用相应的端点。当 OTG_HS 在特定端点的 SETUP 数据包后发现第一个 IN 或 OUT 令牌时，会将 SETUP_COMPLETE 状态置于 Rx FIFO 中。应用程序随后必须重新使能端点以接收用于控制传输的 OUT 数据，并重新编程缓冲区起始地址。鉴于上述行为，OTG_HS 可接收任意数量的背靠背设置数据包，每个设置数据包使用一个缓冲区。

位 14:7 保留，必须保持复位值。

位 6 **B2BSTUP**：接收到连续的 SETUP 数据包 (Back-to-back SETUP packets received)

仅适用于控制 OUT 端点。

此位指示该端点已接收到三个以上的连续 SETUP 数据包。

位 5 **STSPHSRX**：接收到控制写的状态阶段 (StsPhseRcvd) (Status Phase Received For Control Write (StsPhseRcvd))。

该中断仅对控制 OUT 端点有效。只有当 OTG_HS/ 将主机在控制写传输的数据阶段发送的所有数据全部传输到系统存储器缓冲区后，才会生成该中断。该中断向应用程序指示主机已从控制写传输的数据阶段切换到状态阶段。完成数据阶段的解码后，应用程序可使用该中断向状态阶段作出 ACK 或 STALL 响应。

位 4 **OTEPDIS**：端点禁止时接收到 OUT 令牌 (OUT token received when endpoint disabled)

仅适用于控制 OUT 端点。

指示在尚未使能端点时接收到 OUT 令牌，从而产生中断。

位 3 **STUP**：SETUP 阶段完成 (SETUP phase done)

仅适用于控制 OUT 端点。

指示控制端点的 SETUP 阶段已完成，当前控制传输中不再接收到连续的 SETUP 数据包。在此中断上，应用程序可以对接收到的 SETUP 数据包进行解码。

位 2 保留，必须保持复位值。

位 1 **EPDISD**：端点禁止中断 (Endpoint disabled interrupt)
此位指示该端点已经由应用程序禁止掉。

位 0 **XFRC**：传输完成中断 (Transfer completed interrupt)
此字段指示在此端点上设置的传输已经在 USB 和 AHB 上传输完成。

57.14.51 OTG 设备 IN 端点 0 传输大小寄存器 (OTG_DIEPTSIZ0)

OTG device IN endpoint 0 transfer size register

偏移地址：0x910

复位值：0x0000 0000

在使能端点 0 之前，应用程序必须修改此寄存器。通过设备控制端点 0 控制寄存器中的端点使能位（OTG_DIEPCTL0 中的 EPENA）使能端点 0 后，模块对此寄存器进行修改。仅当模块将端点使能位清零后，应用程序才能读取此寄存器。

非零端点使用端点 1-3 的寄存器。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PKTCNT		Res.	Res.	Res.
											rw	rw			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	XFRSIZ						
									rw	rw	rw	rw	rw	rw	rw

位 31:21 保留，必须保持复位值。

位 20:19 **PKTCNT**：数据包计数 (Packet count)
指示端点 0 的一次数据传输包含的数据包个数。
每次从 Tx FIFO 读取数据包（最大大小数据包或短数据包）时，此字段将递减。

位 18:7 保留，必须保持复位值。

位 6:0 **XFRSIZ**：传输大小 (Transfer size)
指示端点 0 的一次数据传输包含的数据量，以字节为单位。仅当应用程序传输完这些数据后，模块才会中断该应用程序。传输大小可以设置为端点的最大数据包大小，以在每个数据包结束时中断。
每次向 Tx FIFO 写入来自外部存储器的数据包时，模块会使此字段递减。

57.14.52 OTG 设备通道 x DMA 地址寄存器 (OTG_DIEPDMAx)（x = 0..15，其中 x= 通道编号）

OTG Device channel-x DMA address register

偏移地址：0x914 + (通道编号 × 0x20)

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DMAADDR															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DMAADDR															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:0 **DMAADDR**：DMA 地址 (DMA address)

该字段保存外部存储器中的起始地址，必须从该地址获取端点的数据。每次进行 AHB 传输，该寄存器都会递增。

57.14.53 OTG 设备 OUT 端点 0 传输大小寄存器 (OTG_DOEPTSIZ0)

OTG device OUT endpoint 0 transfer size register

偏移地址：0xB10

复位值：0x0000 0000

在使能端点 0 之前，应用程序必须修改此寄存器。通过 OTG_DOEPCTL0 寄存器中的端点使能位（OTG_DOEPCTL0 中的 EPENA 位）使能端点 0 后，模块对此寄存器进行修改。仅当模块将端点使能位清零后，应用程序才能读取此寄存器。

非零端点使用端点 1–8 的寄存器。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	STUPCNT		Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PKTCNT	Res.	Res.	Res.
	rw	rw										rw			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	XFRSIZ						
									rw	rw	rw	rw	rw	rw	rw

位 31 保留，必须保持复位值。

位 30:29 **STUPCNT**：SETUP 数据包计数 (SETUP packet count)

此字段指定端点能连续接收的 SETUP 数据包数量。

01：1 个数据包

10：2 个数据包

11：3 个数据包

位 28:20 保留，必须保持复位值。

位 19 **PKTCNT**：数据包计数 (Packet count)

每向 Rx FIFO 写入一个数据包，此字段递减。

位 18:7 保留，必须保持复位值。

位 6:0 **XFRSIZ**：传输大小 (Transfer size)

指示端点 0 的一次数据传输包含的数据量，以字节为单位。仅当应用程序传输完这些数据后，模块才会中断该应用程序。传输大小可以设置为端点的最大数据包大小，以在每个数据包结束时中断。

每次从 Rx FIFO 读取数据包并将其写入外部存储器时，模块会使此字段递减。

57.14.54 OTG 设备通道 x DMA 地址寄存器 (OTG_DOEPDMAx)（x = 0..15，其中 x= 通道编号）

OTG Device channel-x DMA address register

偏移地址：0xB14 +（通道编号 × 0x20）

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DMAADDR															
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DMAADDR															
RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW

位 31:0 **DMAADDR**：DMA 地址 (DMA address)

该字段保存外部存储器中的起始地址，必须从该地址获取端点的数据。每次进行 AHB 传输，该寄存器都会递增。

57.14.55 OTG 设备 IN 端点 x 传输大小寄存器 (OTG_DIEPTSIZx)（x = 1..8，其中 x= 端点编号）

OTG device IN endpoint-x transfer size register

偏移地址：0x910 + (Endpoint_number × 0x20)

复位值：0x0000 0000

在使能该端点之前，应用程序必须修改此寄存器。通过 OTG_DIEPCTLx 寄存器中的端点使能位（OTG_DIEPCTLx 中的 EPENA 位）使能该端点后，模块对此寄存器进行修改。仅当模块将端点使能位清零后，应用程序才能读取此寄存器。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	MCNT		PKTCNT										XFRSIZ		
	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
XFRSIZ															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31 保留，必须保持复位值。

位 30:29 **MCNT**：多重计数 (Multi count)

对于周期性 IN 端点，此字段指示在 USB 上每帧必须发送的数据包数。模块使用此字段计算同步 IN 端点的数据 PID。

01：1 个数据包

10：2 个数据包

11：3 个数据包

位 28:19 **PKTCNT**：数据包计数 (Packet count)

指示该端点上的一次数据传输包含的数据包个数。

每次从 Tx FIFO 读取数据包（最大大小数据包或短数据包）时，此字段将递减。

位 18:0 **XFRSIZ**：传输大小 (Transfer size)

此字段包含当前端点的一次数据传输包含的数据量，以字节为单位。仅当应用程序传输完这些数据后，模块才会中断该应用程序。传输大小可以设置为端点的最大数据包大小，以在每个数据包结束时中断。

每次向 Tx FIFO 写入来自外部存储器的数据包时，模块会使此字段递减。

57.14.56 OTG 设备 IN 端点发送 FIFO 状态寄存器 (OTG_DTXFSTSx)（x = 0..8，其中 x = 端点编号）

OTG device IN endpoint transmit FIFO status register

IN 端点的偏移地址：0x918 + (端点编号 × 0x20) 此只读寄存器包含设备 IN 端点 Tx FIFO 的空闲空间信息。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INEPTFSAV															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

31:16 保留，必须保持复位值。

15:0 **INEPTFSAV**：IN 端点 Tx FIFO 可用空间 (IN endpoint Tx FIFO space available)

指示端点 Tx FIFO 中的可用空闲空间大小。

以 32 位字为单位：

0x0：端点 Tx FIFO 已满

0x1：1 个字可用

0x2：2 个字可用

0xn：n 个字可用

其它值：保留

57.14.57 OTG 设备 OUT 端点 x 传输大小寄存器 (OTG_DOEPTSIZx)（x = 1..8，其中 x = 端点编号）

OTG device OUT endpoint-x transfer size register

偏移地址：0xB10 + (端点编号 × 0x20)

复位值：0x0000 0000

在使能该端点之前，应用程序必须修改此寄存器。通过 OTG_DOEPCTLx 寄存器中的端点使能位（OTG_DOEPCTLx 中的 EPENA 位）使能该端点后，模块对此寄存器进行修改。仅当模块将端点使能位清零后，应用程序才能读取此寄存器。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	RXDPID/ STUPCNT		PKTCNT										XFRSIZ		
	r/rw	r/rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
XFRSIZ															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31 保留，必须保持复位值。

位 30:29 **RXDPID**：接收到的数据 PID (Received data PID)

仅适用于同步 OUT 端点。

这是此端点收到的上一个数据包的 PID。

00：DATA0

01：DATA2

10：DATA1

11：MDATA

STUPCNT：SETUP 数据包计数 (SETUP packet count)

仅适用于控制 OUT 端点。

此字段指定端点能连续接收的 SETUP 数据包数量。

01：1 个数据包

10：2 个数据包

11：3 个数据包

位 28:19 **PKTCNT**：数据包计数 (Packet count)

指示该端点上的一次数据传输包含的数据包个数。

每次向 Rx FIFO 写入数据包（最大大小数据包或短数据包）后，此字段将递减。

位 18:0 **XFRSIZ**：传输大小 (Transfer size)

此字段包含当前端点的一次数据传输包含的数据量，以字节为单位。仅当应用程序传输完这些数据后，模块才会中断该应用程序。传输大小可以设置为端点的最大数据包大小，以在每个数据包结束时中断。

每次从 Rx FIFO 读取数据包并将其写入外部存储器时，模块会使此字段递减。

57.14.58 OTG 电源和时钟门控控制寄存器 (OTG_PCGCCTL)

OTG power and clock gating control register

偏移地址：0xE00

复位值：0x0x200B 8000

此寄存器在主机模式和设备模式下均可用。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SUSP	PHY SLEEP	ENL1 GTG	PHY SUSP	Res.	Res.	GATE HCLK	STPP CLK
								r	r	RW	r			rw	rw

位 31:8 保留，必须保持复位值。

位 7 **SUSP**：深度睡眠 (Deep Sleep)
在 L1 状态下时，此位表示 PHY 处于深度睡眠状态。

位 6 **PHYSLEEP**：PHY 处于睡眠状态 (PHY in Sleep)
此位指示 PHY 处于睡眠状态。

位 5 **ENL1GTG**：使能睡眠时钟门控 (Enable Sleep clock gating)
此位置 1 时，如果模块无法触发 utmi_l1_suspend_n，则会在睡眠状态下使能模块内部时钟门控。当此位不置 1 时，不会在睡眠状态下使能 PHY 时钟门控。

位 4 **PHYSUSP**：PHY 挂起 (PHY Suspended)
指示 PHY 已挂起。应用程序将 STPPCLK 位置 1 后，一旦 PHY 挂起，此位就会更新。

位 3:2 保留，必须保持复位值。

位 1 **GATEHCLK**：门控 HCLK (Gate HCLK)
当 USB 通信挂起或会话无效时，应用程序会将此位置 1，以停止对除 AHB 总线从接口、主接口和唤醒逻辑之外的模块提供时钟。当 USB 恢复通信或新会话启动时，应用程序将此位清零。

位 0 **STPPCLK**：停止 PHY 时钟 (Stop PHY clock)
当 USB 通信挂起、会话无效或设备断开连接时，应用程序将此位置 1 以停止 PHY 时钟。当 USB 恢复通信或新会话启动时，应用程序将此位清零。

57.14.59 OTG_HS 寄存器映射

下表提供了 USB OTG 寄存器映射和复位值。

表 489. 寄存器映射和复位值

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x000	OTG_GOTGCTL	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OTGVER	BSVLD	ASVLD	DBCT	CIDSTS	Res.	Res.	Res.	EHEN	DHNPEN	HSHNPEN	HNPRQ	HNGSCS	BVALOVAL	BVALOEN	AVALOVAL	AVALOEN	VBVALOVAL	VBVALOEN	SRQ	SRQSCS
	Reset value												0	0	0	0	1				0	0	0	0	0	0	0	0	0	0	0	0	0
0x004	OTG_GOTGINT	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	IDCHNG	DBCDNE	ADTOCHG	HNGDET	Res.	Res.	Res.	Res.	Res.	Res.	Res.	HNSSCHG	SRSSCHG	Res.	Res.	Res.	Res.	Res.	SEDET	Res.	Res.
	Reset value												0	0	0	0								0	0						0		
0x008	OTG_GAHBCFG	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PTXFELVL	TXFELVL	Res.	DMAEN	HBSTLEN				GINTMSK
	Reset value																								0	0		0	0	0	0	0	0
0x00C	OTG_GUSBCFG	Res.	FDMOD	FHMOD	Res.	Res.	Res.	ULPIIPD	PTCI	PCCI	TSDPS	ULPIEVBUSI	ULPIEVBUSD	ULPICSM	ULPIAR	ULPIFSL	Res.	PHYLPC	Res.	TRDT				HNPCAP	SRPCAP	Res.	PHYSEL	Res.	Res.	Res.	TOCAL		
	Reset value		0	0				0	0	0	0	0	0	0	0	0		0		0	1	0	1	0	0		1				0	0	0

表 489. 寄存器映射和复位值（续）

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x010	**OTG_ GRSTCTL**	AHBIDL	DMAREQ	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TXFNUM					TXFFLSH	RXFFLSH	Res.	Res.	PSRST	CSRST
	Reset value	1	0																				0	0	0	0	0	0	0			0	0
0x014	**OTG_ GINTSTS**	WKUINT	SRQINT	DISCINT	CIDSCHG	Res.	PTXFE	HCINT	HPRTINT	Res.	DATAFSUSP	IPXFR/INCOMPISOOUT	IISOIXFR	OEPINT	IEPINT	Res.	Res.	EOPF	ISOODRP	ENUMDNE	USBRST	USBSUSP	ESUSP	Res.	Res.	GONAKEFF	GINAKEFF	NPTXFE	RXFLVL	SOF	OTGINT	MMIS	CMOD
	Reset value	0	0	0	1		1	0	0	0	0	0	0	0	0			0	0	0	0	0	0			0	0	1	0	0	0	0	0
0x018	**OTG_ GINTMSK**	WUIM	SRQIM	DISCINT	CIDSCHGM	LPMINTM	PTXFEM	HCIM	PRTIM	RSTDETM	FSUSPM	IPXFRM/IISOOXFRM	IISOIXFRM	OEPINT	IEPINT	Res.	Res.	EOPFM	ISOODRPM	ENUMDNEM	USBRST	USBSUSPM	ESUSPM	Res.	Res.	GONAKEFFM	GINAKEFFM	NPTXFEM	RXFLVLM	SOFM	OTGINT	MMISM	Res.
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0			0	0	0	0	0	0			0	0	0	0	0	0	0	
0x01C	**OTG_ GRXSTSR (host mode)**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PKTSTS				DPID		BCNT												CHNUM		
	Reset value												0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	OTG_ GRXSTSR (Device mode)	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FRMNUM				PKTSTS				DPID		BCNT												EPNUM		
	Reset value								0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x020	**OTG_ GRXSTSR (host mode)**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PKTSTS				DPID		BCNT												CHNUM		
	Reset value												0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	OTG_ GRXSTSPR (Device mode)	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FRMNUM				PKTSTS				DPID		BCNT												EPNUM		
	Reset value								0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x024	**OTG_ GRXFSIZ**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RXFD															
	Reset value																	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
0x028	**OTG_ HNPTXFSIZ/ OTG_ DIEPTXF0**	NPTXFD/TX0FD																NPTXFSA/TX0FSA															
	Reset value	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
0x02C	**OTG_ HNPTXSTS**	Res.	NPTXQTOP							NPTQXSAV								NPTXFSAV															
	Reset value		0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0

表 489. 寄存器映射和复位值（续）

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x030	**OTG_ GI2CCTL**	BSYDNE	RW	Res.	I2CDATSE	I2CD EVAD R		Res.	ACK	I2CEN	ADDR							REGADDR								RWDATA							
	Reset value	0	0		0	0	0		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x038	**OTG_ GCCFG**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	VBDEN	SDEN	PDEN	DCDEN	BCDEN	PWRDWN	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PS2DET	SDET	PDET	DCDET
	Reset value											0	0	0	0	0	0													0	0	0	0
0x03C	**OTG_CID**	PRODUCT_ID																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
0x054	**OTG_ GLPMCFG**	Res.	Res.	Res.	ENBESL	LPMR CNTSTS			SNDLPM	LPM RCNT			LPMCHIDX				L1RSMOK	SLPSTS	LPM RSP		L1DSEN	BESLTHRS			L1SSEN	REMWAKE	BESL				LPMACK	LPMEN	
	Reset value				0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x100	**OTG_ HPTXFSIZ**	PTXFSIZ																PTXSA															
	Reset value	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
0x104	**OTG_ DIEPTXF1**	INEPTXFD																INEPTXSA															
	Reset value	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
0x108	**OTG_ DIEPTXF2**	INEPTXFD																INEPTXSA															
	Reset value	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0
. . . .	. . . .	.																															
. . . .	. . . .	.																															
0x244	**OTG_ DIEPTXF7**	INEPTXFD																INEPTXSA															
	Reset value	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0
0x400	**OTG_ HCFG**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FSLSS	FSLS PCS	
	Reset value																														0	0	0
0x404	**OTG_ HFIR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RLDCTRL	FRIVL															
	Reset value																0	1	1	1	0	1	0	1	0	0	1	1	0	0	0	0	0
0x408	**OTG_ HFNUM**	FTREM																FRNUM															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1

表 489. 寄存器映射和复位值（续）

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x410	**OTG_HPTXSTS**	PTXQTOP								PTXQSAV								PTXFSAVL															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
0x414	**OTG_HAINT**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	HAINT															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x418	**OTG_HAINTMSK**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	HAINTM															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x440	**OTG_HPRT**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PSPD		PTCTL				PPWR	PLSTS		Res.	PRST	PSUSP	PRES	POCCHNG	POCA	PENCHNG	PENA	PCDET	PCSTS
	Reset value														0	0	0	0	0	0	0	0	0		0	0	0	0	0	0	0	0	0
0x500	**OTG_HCCHAR0**	CHENA	CHDIS	ODDFRM	DAD							MCNT		EPTYP	LSDEV	Res.	EPDIR	EPNUM					MPSIZ										
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x504	**OTG_HCSPLT0**	SPLITEN	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	COMPLSPLT	XACTPOS		HUBADDR							PRTADDR						
	Reset value	0															0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x508	**OTG_HCINT0**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DTERR	FRMOR	BBERR	TXERR	Res.	ACK	NAK	STALL	Res.	CHH	XFRC
	Reset value																						0	0	0	0		0	0	0		0	0
0x510	**OTG_HCTSIZ0**	Res.	DPID		PKTCNT										XFRSIZ																		
	Reset value		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x50C	**OTG_HCINTMSK0**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DTERRM	FRMORM	BBERRM	TXERRM	NYET	ACKM	NAKM	STALLM	Res.	CHHM	XFRCM
	Reset value																						0	0	0	0	0	0	0	0		0	0
0x514	**OTG_HCDMA0**	DMAADDR																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x520	**OTG_HCCHAR1**	CHENA	CHDIS	ODDFRM	DAD							MCNT		EPTYP	LSDEV	Res.	EPDIR	EPNUM					MPSIZ										
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x528	**OTG_HCINT1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DTERR	FRMOR	BBERR	TXERR	Res.	ACK	NAK	STALL	Res.	CHH	XFRC
	Reset value																						0	0	0	0		0	0	0		0	0

表 489. 寄存器映射和复位值（续）

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x52C	OTG_ HCINTMSK1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DTERRM	FRMORM	BBERRM	TXERRM	NYET	ACKM	NAKM	STALLM	Res.	CHHM	XFRCM
	Reset value																						0	0	0	0	0	0	0	0		0	0
0x530	OTG_ HCTSIZ1	Res.	DPID		PKTCNT										XFRSIZ																		
	Reset value		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
. . . .	. . . .																	. . . .															
. . . .	. . . .																	. . . .															
. . . .	. . . .																	. . . .															
. . . .	. . . .																	. . . .															
0x6E0	OTG_ HCCHAR15	CHENA	CHDIS	ODDFRM	DAD							MCNT		EPTYP	LSDEV	Res.	EPDIR	EPNUM					MPSIZ											
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
. . . .	. . . .																	. . . .															
0x6E4	OTG_ HCSPLT15	SPLITEN	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	COMPLSPLT	XACT POS		HUBADDR							PRTADDR							
	Reset value	0															0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
. . . .	. . . .																	. . . .															
0x6EC	OTG_ HCINTMSK15	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DTERRM	FRMORM	BBERRM	TXERRM	NYET	ACKM	NAKM	STALLM	Res.	CHHM	XFRCM
	Reset value																						0	0	0	0	0	0	0	0		0	0
. . . .	. . . .																	. . . .															
0x6F0	OTG_ HCTSIZ15	Res.	DPID		PKTCNT										XFRSIZ																		
	Reset value		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

表 489. 寄存器映射和复位值（续）

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
. . . .	. . . .	. . . .																															
0x6F4	**OTG_ HCDMA15**	DMAADDR																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
. . . .	. . . .	. . . .																															
. . . .	. . . .	. . . .																															
0x7A8	**OTG_ HCINT15**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DTERR	FRMOR	BBERR	TXERR	Res.	ACK	NAK	STALL	Res.	CHH	XFRC
	Reset value																						0	0	0	0		0	0	0		0	0
0x800	**OTG_ DCFG**	Res.	Res.	Res.	Res.	Res.	Res.	PERS CHI VL		Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ERRATIM	XCVRDLY	Res.	PFIVL		DAD							Res.	NZLSOHSK	DSPD	
	Reset value																	0	0		0	0	0	0	0	0	0	0	0		0	0	0
0x804	**OTG_ DCTL**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DSBESLRJCT	Res.	Res.	Res.	Res.	Res.	Res.	POPRGDNE	CGONAK	SGONAK	CGINAK	SGINAK	TCTL			GONSTS	GINSTS	SDIS	RWUSIG
	Reset value														0							0	0	0	0	0	0	0	0	0	1	0
0x808	**OTG_ DSTS**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DEV LN STS		FNSOF															Res.	Res.	Res.	Res.	EERR	ENUMSPD	SUSPSTS
	Reset value									0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0					0	0	0
0x810	**OTG_ DIEPMSK**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	NAKM	Res.	Res.	Res.	Res.	Res.	INEPNEM	INEPNMM	ITTXFEMSK	TOM	Res.	EPDM	XFRCM
	Reset value																				0						0	0	0	0		0	0
0x814	**OTG_ DOEPMSK**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	NYETMSK	Res.	Res.	Res.	Res.	BOIM	TXFURM	Res.	B2BSTUP	Res.	OTEPDM	STUPM	Res.	EPDM	XFRCM
	Reset value																		0					0			0		0	0		0	0
0x818	**OTG_ DAINT**	OEPINT																IEPINT															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x81C	**OTG_ DAINTMSK**	OEPM																IEPM															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

表 489. 寄存器映射和复位值（续）

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x828	**OTG_ DVBUSDIS**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	VBUSDT															
	Reset value																	0	0	0	1	0	1	1	1	1	1	0	1	0	1	1	1
0x82C	**OTG_DVB USPULSE**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DVBUSP															
	Reset value																	0	0	0	0	0	1	0	1	1	0	1	1	1	0	0	0
0x830	**OTG_ DTHRCTL**	Res.	Res.	Res.	Res.	ARPEN	Res.	RXTHRLEN									RXTHREN	Res.	Res.	Res.	Res.	TXTHRLEN									ISOTHREN	NONISOTHREN
	Reset value					0											0					0	0	0	0	0	0	0	0	0	0	0	0
0x834	**OTG_DIE PEMPMSK**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	INEPTXFEM															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x838	**OTG_ DEACHINT**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OEP1INT	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	IEP1INT	Res.
	Reset value															0																0	
0x83C	**OTG_DEACHI NTMSK**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OEP1INTM	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	IEP1INTM	Res.
	Reset value															0																0	
0x900	**OTG_ DIEPCTL0**	EPENA	EPDIS	SODDFRM/SD1PID	SD0PID/SEVNFRM	SNAK	CNAK	TXFNUM				STALL	Res.	EPTYP		NAKSTS	EONUM/DPID	USBAEP	Res.	Res.	Res.	Res.	MPSIZ										
	Reset value	0	0	0	0	0	0	0	0	0	0	0		0	0	0	0	0					0	0	0	0	0	0	0	0	0	0	0
0x908	**OTG_ DIEPINT0**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TXFE	INEPNE	Res.	ITTXFE	TOC	Res.	EPDISD	XFRC
	Reset value																									1	0		0	0		0	0
0x910	**OTG_ DIEPTSIZ0**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PKT CNT		Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	XFRSIZ						
	Reset value												0	0													0	0	0	0	0	0	0
0x914	**OTG_ DIEPDMA**	DMAADDR																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x918	**OTG_ DTXFSTS0**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	INEPTFSAV															
	Reset value																	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0

表 489. 寄存器映射和复位值（续）

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x920	**OTG_ DIEPCTL1**	EPENA	EPDIS	SODDFRM/SD1PID	SD0PID/SEVNFRM	SNAK	CNAK	TXFNUM				STALL	Res.	EPTYP		NAKSTS	EONUM/DPID	USBAEP	Res.	Res.	Res.	Res.	MPSIZ										
	Reset value	0	0	0	0	0	0	0	0	0	0	0		0	0	0	0	0					0	0	0	0	0	0	0	0	0	0	0
0x928	**OTG_ DIEPINT1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TXFE	INEPNE	Res.	ITTXFE	TOC	Res.	EPDISD	XFRC
	Reset value																									1	0		0	0		0	0
0x930	**OTG_ DIEPTSIZ1**	Res.	MCNT		PKTCNT									XFRSIZ																			
	Reset value		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x938	**OTG_ DTXFSTS1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	INEPTFSAV															
	Reset value																	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
0x940	**OTG_ DIEPCTL2**	EPENA	EPDIS	SODDFRM	SD0PID/SEVNFRM	SNAK	CNAK	TXFNUM				STALL	Res	EPTYP		NAKSTS	EONUM/DPID	USBAEP	Res.	Res.	Res.	Res	MPSIZ										
	Reset value	0	0	0	0	0	0	0	0	0	0	0		0	0	0	0	0					0	0	0	0	0	0	0	0	0	0	0
. . . .	. . . .	. . . . .																															
. . . .	. . . .	. . . . .																															
. . . .	. . . .	. . . . .																															
. . . .	. . . .	. . . . .																															
0x9E0	**OTG_ DIEPCTL7**	EPENA	EPDIS	SODDFRM	SD0PID/SEVNFRM	SNAK	CNAK	TXFNUM				STALL	Res.	EPTY P		NAKSTS	EONUM/DPID	USBAEP	Res.	Res.	Res.	Res.	MPSIZ										
	Reset value	0	0	0	0	0	0	0	0	0	0	0		0	0	0	0	0					0	0	0	0	0	0	0	0	0	0	0
. . . .	. . . .	. . . . .																															

表 489. 寄存器映射和复位值（续）

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x9E8	**OTG_ DIEPINT7**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TXFE	INEPNE	Res.	ITTXFE	TOC	Res.	EPDISD	XFRC
	Reset value																									1	0		0	0		0	0
. . . .	. . . .	.																															
0x9F0	**OTG_ DIEPTSIZ7**	Res.	MCNT		PKTCNT										XFRSIZ																		
	Reset value		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
. . . .	. . . .	.																															
0x9F8	**OTG_ DTXFSTS7**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	INEPTFSAV															
	Reset value																	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
0xB00	**OTG_ DOEPCTL0**	EPENA	EPDIS	Res.	Res.	SNAK	CNAK	Res.	Res.	Res.	Res.	STALL	SNPM	EPTYP		NAKSTS	Res.	USBAEP	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MPSIZ	
	Reset value	0	0			0	0					0	0	0	0	0		1														0	0
0xB08	**OTG_ DOEPINT0**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	B2BSTUP	STSPHSRX	OTEPDIS	STUP	Res.	EPDISD	XFRC
	Reset value																										0	0	0	0		0	0
0xB10	**OTG_ DOEPTSIZ0**	Res.	STUP CNT		Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PKTCNT	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	XFRSIZ						
	Reset value		0	0										0													0	0	0	0	0	0	0
0xB14	**OTG_ DOEPDMA**	DMAADDR																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0xB20	**OTG_ DOEPCTL1**	EPENA	EPDIS	SODDFRM	SD0PID/SEVNFRM	SNAK	CNAK	Res.	Res.	Res.	Res.	STALL	SNPM	EP 典型值		NAKSTS	EONUM/DPID	USBAEP	Res.	Res.	Res.	Res.	MPSIZ										
	Reset value	0	0	0	0	0	0					0	0	0	0	0	0	0					0	0	0	0	0	0	0	0	0	0	0
0xB28	**OTG_ DOEPINT1**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	B2BSTUP	STSPHSRX	OTEPDIS	STUP	Res.	EPDISD	XFRC
	Reset value																										0	0	0	0		0	0

表 489. 寄存器映射和复位值（续）

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0xB30	**OTG_ DOEPTSIZ1**	Res.	RXDPID/ STUPCNT		PKTCNT										XFRSIZ																		
	Reset value		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0xB50	**OTG_ DOEPTSIZ2**	Res.	RXDPID/ STUPCNT		PKTCNT										XFRSIZ																		
	Reset value		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
. . . .	. . . .																																
. . . .	. . . .																																
0xBE0	**OTG_ DOEPCTL7**	EPENA	EPDIS	SODDFRM	SD0PID/SEVNFRM	SNAK	CNAK	Res.	Res.	Res.	Res.	STALL	SNPM	EPTYP	NAKSTS	EONUM/DPID	USBAEP	Res.	Res.	Res.	Res.	MPSIZ											
	Reset value	0	0	0	0	0	0					0	0	0	0	0	0	0					0	0	0	0	0	0	0	0	0	0	0
. . . .	. . . .																																
. . . .	. . . .																																
. . . .	. . . .																																
0xBE8	**OTG_ DOEPINT7**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	STPKTRX	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	B2BSTUP	STSPHSRX	OTEPDIS	STUP	Res.	EPDISD	XFRC
	Reset value																	0									0	0	0	0		0	0
. . . .	. . . .																																
0xBF0	**OTG_ DOEPTSIZ7**	Res.	RXDPID/ STUPCNT		PKTCNT										XFRSIZ																		
	Reset value		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

表 489. 寄存器映射和复位值（续）

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0xE00	OTG_ PCGCCTL	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SUSP	PHYSLEEP	ENL1GTG	PHYSUSP	Res.	Res.	GATEHCLK	STPPCLK
	Reset value																									0	0	0	0			0	0

有关寄存器边界地址的信息，请参见*第 2.2.2 节：存储器映射和寄存器边界地址*。

57.15 OTG_HS 编程模型

57.15.1 模块初始化

应用程序必须执行模块初始化序列。如果上电期间连接电缆，则 OTG_GINTSTS 中的当前工作模式位（OTG_GINTSTS 中的 CMOD 位）将指示模式。连接“A 型”插头后，OTG_HS 控制器进入主机模式；连接“B 型”插头后，OTG_FS 控制器进入设备模式。

本节介绍了 OTG_HS 控制器在上电后的初始化过程。无论是以主机模式还是设备模式工作，应用程序都必须遵循初始化序列。根据模块配置对所有模块全局寄存器进行初始化：

1. 在 OTG_GAHBCFG 寄存器中编程以下字段：
   - 全局中断屏蔽位 GINTMSK = 1
   - Rx FIFO 非空（OTG_GINTSTS 中的 RXFLVL 位）
   - 周期性 Tx FIFO 空门限
2. 在 OTG_GUSBCFG 寄存器中编程以下字段：
   - HNP 功能位
   - SRP 功能位
   - OTG_HS 超时校准字段
   - USB 周转时间字段
3. 软件必须使能 OTG_GINTMSK 寄存器中的以下位：

   OTG 中断屏蔽

   模式不匹配中断屏蔽
4. 通过读取 OTG_GINTSTS 中的 CMOD 位，软件可确定 OTG_HS 控制器是在主机模式还是设备模式下工作。

57.15.2 主机初始化

要将模块作为主机进行初始化，应用程序必须执行以下步骤：

1. 编程 OTG_GINTMSK 寄存器中的 HPRTINT 以打开中断。
2. 编程 OTG_HCFG 寄存器以选择全速主机。
3. 将 OTG_HPRT 中的 PPWR 位编程为 1，给 USB 总线提供 V_{BUS}。
4. 等待 OTG_HPRT0 中的 PCDET 中断。这表示某设备已连接到主机端口。
5. 将 OTG_HPRT 中的 PRST 位编程为 1，在 USB 总线上发出复位信号。
6. 至少等待 10 ms，以便完成复位过程。
7. 将 OTG_HPRT 中的 PRST 位编程为 0，
8. 等待 OTG_HPRT 中的 PENCHNG 中断。
9. 读取 OTG_HPRT 中的 PSPD 位以获取枚举速度。
10. 使用所选 PHY 时钟 ，相应地设置 HFIR 寄存器。
11. 根据步骤 9 中检测到的设备速度编程 OTG_HCFG 寄存器中的 FSLSPCS 字段。如果 FSLSPCS 发生更改，则必须执行端口复位。
12. 编程 OTG_GRXFSIZ 寄存器以选择接收 FIFO 的大小。
13. 编程 OTG_HNPTXFSIZ 寄存器，以选择用于非周期性通信事务的非周期性发送 FIFO 的大小和起始地址。
14. 编程 OTG_HPTXFSIZ 寄存器，以选择用于周期性通信事务的周期性发送 FIFO 的大小和起始地址。

要与设备通信，系统软件必须初始化并使能至少一个通道。

57.15.3 设备初始化

上电期间或者从主机模式切换为设备模式后，应用程序必须执行下列步骤来将模块作为设备进行初始化。

1. 在 OTG_DCFG 寄存器中编程以下字段：
   - 设备速度
   - 非零长度状态 OUT 握手信号
2. 编程 OTG_GINTMSK 寄存器以使能以下中断：
   - USB 复位
   - 枚举完成
   - 早期挂起
   - USB 挂起
   - SOF
3. 等待 OTG_GINTSTS 中的 USBRST 中断。这表示已在 USB 上检测到复位信号，复位过程自接收到此中断后约持续 10 ms。

等待 OTG_GINTSTS 中的 ENUMDNE 中断。此中断指示 USB 上复位过程结束。接收到此中断时，应用程序必须读取 OTG_DSTS 寄存器以确定枚举速度并执行*第 2502 页的枚举完成时的端点初始化*中所列的步骤。

此时，设备已准备好接受 SOF 数据包并在控制端点 0 上执行控制传输。

57.15.4 DMA 模式

OTG 主机使用 AHB 主接口来获取发送数据包数据（AHB 到 USB）和接收数据更新（USB 到 AHB）。AHB 主接口使用经过编程的 DMA 地址（主机模式下的 OTG_HCDMAx 寄存器和外设模式下的 OTG_DIEPDMAx/OTG_DOEPDMAx 寄存器）来访问数据缓冲区。

分散/收集 DMA 模式

TBC.

57.15.5 主机编程模型

通道初始化

应用程序必须初始化一个或多个通道，之后才能与所连接的设备通信。要初始化和使能通道，应用程序必须执行以下步骤：

1. 编程 OTG_GINTMSK 寄存器以取消对以下位的中断屏蔽：
2. 通道中断
   - 用于 OUT 事务的非周期性发送 FIFO 为空（在流水线事务级别工作且数据包计数字段编程值大于 1 时适用）。
   - 用于 OUT 事务的非周期性发送 FIFO 为半空（在流水线事务级别工作且数据包计数字段编程值大于 1 时适用）。
3. 编程 OTG_HAINTMSK 寄存器以使能所选通道中断。
4. 编程 OTG_HCINTMSK 寄存器，以使能主机通道中断寄存器中反映的和通信事务有关的中断。
5. 编程所选通道的 OTG_HCTSIZx 寄存器，指定以字节为单位的总传输大小和包括短数据包在内的预期数据包个数。应用程序必须使用初始数据 PID（用于第一个 OUT 事务或预期从第一个 IN 事务获取）编程 PID 字段。
6. 编程所选通道的 OTG_HCCHARx 寄存器，指定设备的端点特性，例如类型、速度、方向等。（仅当应用程序准备好发送或接收数据包时，才能通过将通道使能位置 1 来使能通道）。
7. 使用集线器地址和端口地址在 OTG_HCSPLTx 寄存器中对所选通道进行编程（仅分离事务）。
8. 使用缓冲区起始地址在 OTG_HCDMAx 寄存器中对所选通道进行编程（仅 DMA 事务）。

通道的停止

应用程序可以通过编程 OTG_HCCHARx 寄存器将 CHDIS 和 CHENA 位置 1 来禁止任何通道。这会使 OTG_HS 主机清空之前在该通道上发出的请求（如果有）并生成通道停止中断。应用程序在将通道重新分配给其它通信事务之前，必须等待 OTG_HCINTx 中的 CHH 中断。OTG_HS 主机不会中断已在 USB 上启动的通信事务。

要禁止某个在 DMA 模式下工作的通道，应用程序无需检查请求队列中是否有可用空间。OTG_HS 主机检查是否有空间，在仲裁给要禁止的通道时，写入通道禁止的请求。同时，当 OTG_HCCHARx 中的 CHDIS 位置 1 时，将从请求队列中丢弃所有已发出的请求。

禁止通道前，应用程序必须确保非周期性请求队列（禁止非周期性通道时）或周期性请求队列（禁止周期性通道时）中至少有一个空闲空间。应用程序可以在请求队列已满时（禁止通道之前），通过编程 OTG_HCCHARx 寄存器将 CHDIS 位置 1 和将 CHENA 位清零，清空已发出的请求。

出现以下任一情况时，应用程序将禁止通道：

1. IN 或 OUT 通道的 OTG_HCINTx 中接收到 STALL、TXERR、BBERR 或 DTERR 中断。应用程序在接收到通道停止信号之前，必须能够接收相同通道的其它中断（DTERR、Nak、Data、TXERR）。
2. 在非周期性 IN 传输或高带宽中断 IN 传输期间 OTG_HCINTx 中接收到 XFRC 中断
3. 接收到 OTG_GINTSTS 中的 DISCINT（断开设备连接）中断。（应用程序将禁止所有已使能的通道）。
4. 应用程序在传输正常完成之前将其中止。

Ping 协议

当 OTG_HS 主机工作在高速模式下时，如果应用程序要与高速批量或控制（数据和状态阶段）OUT 端点进行通信，则必须使用 ping 协议。当应用程序接收到 NAK/NYET/TXERR 中断时，也必须使用 ping 协议。当 OTG_HS 主机接收到上述其中一个响应时，它不会在该端点上继续执行任何通信事务，而是丢弃所有已发出或已获取的 OUT 请求（从请求队列中），然后清空发送 FIFO 中的相应数据。仅对从模式有效。在从模式下，应用程序可通过以下两种方式发送 ping 令牌：在使能通道之前，将 OTG_HCTSIZx 中的 DOPING 位置 1，或者在通道已使能之后，对 OTG_HCTSIZx 寄存器执行写操作时将 DOPING 位置 1。这将使能 OTG_HS 主机将 ping 请求写入到请求队列中。应用程序必须等待设备对 ping 令牌的响应（NAK、ACK 或 TXERR 中断），然后才能继续执行通信事务或发送另一个 ping 令牌。仅当应用程序从设备的 OUT 端点接收到对 ping 令牌的 ACK 响应后，才能继续执行数据通信事务。在 DMA 模式下工作时，对于批量/控制 OUT 事务，应用程序不需要在收到 NAK/NYET 回复时将 OTG_HCTSIZx 中的 DOPING 位置 1。OTG_HS 主机会自动将 OTG_HCTSIZx 中的 DOPING 位置 1，然后在批量/控制 OUT 传输时发出 ping 令牌。OTG_HS 主机会持续发送 ping 令牌，直至收到 ACK 为止，随后自动切换到数据通信事务。

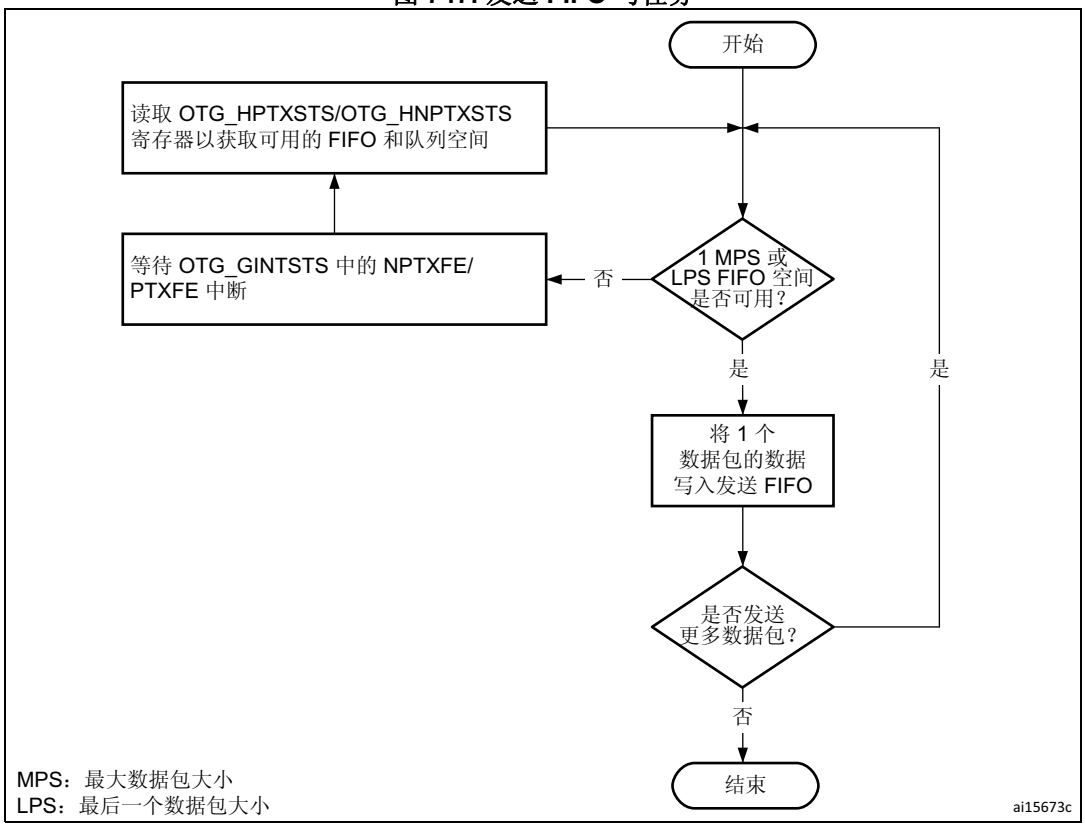
操作模型

应用程序必须初始化一个通道，之后才能与所连接的设备通信。本节介绍了针对不同 USB 事务类型要执行的操作序列。

- **写入发送 FIFO**

  应用程序对数据包执行最后一个 DWORD 写操作的同时，OTG_HS 主机自动向周期性/非周期性请求队列写入一个条目（OUT 请求）。因此开始向发送 FIFO 写入数据之前，应用程序必须确保周期性/非周期性请求队列中至少有一个空闲空间。应用程序必须始终以 DWORD 形式向发送 FIFO 写入数据。如果数据包大小不是 DWORD 的整数倍，则应用程序必须将数据包填充到 DWORD 的整数倍。OTG_HS 主机根据设定的最大数据包大小和传输大小确定实际数据包大小。

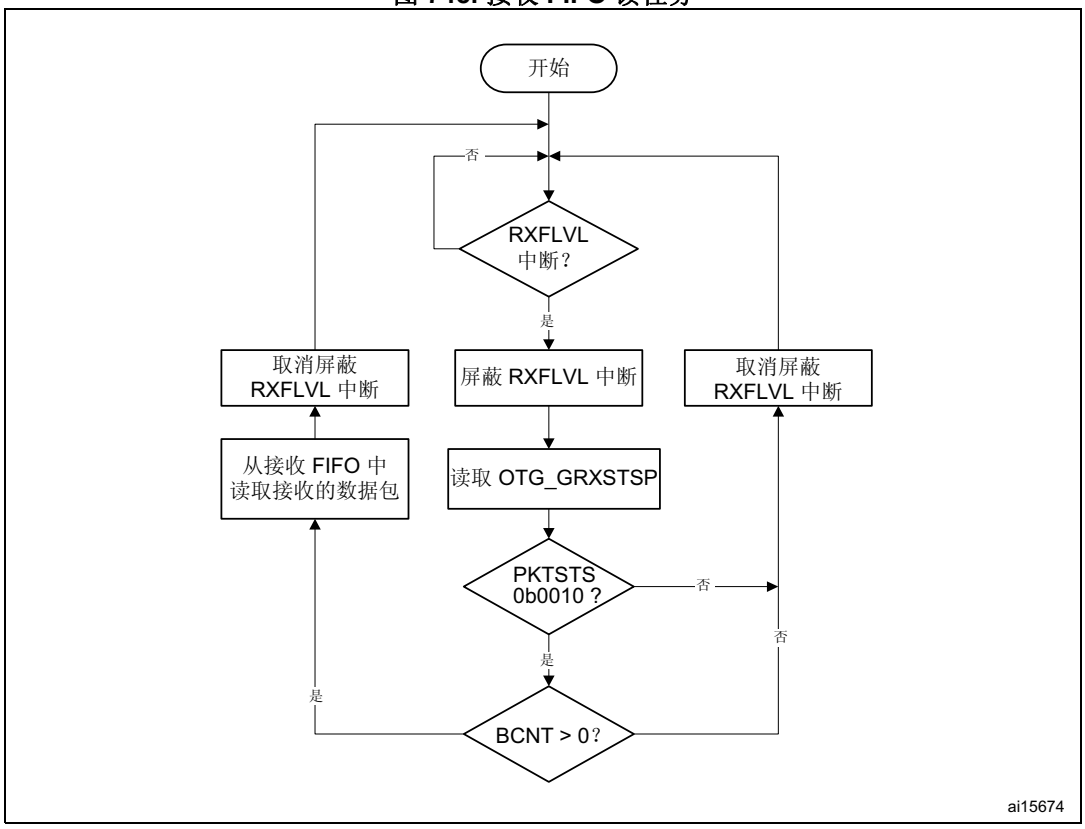
图 747. 发送 FIFO 写任务



- **读取接收 FIFO**

  应用程序必须忽略除 IN 数据包 (bx0010) 以外的所有数据包状态。

图 748. 接收 FIFO 读任务



- **批量和控制传输类型的 OUT/SETUP 通信事务**

  图 749 显示了典型的批量或控制 OUT/SETUP 流水线事务级操作。请参见通道 1 (ch_1)。该通道发送了两个批量 OUT 数据包。控制 SETUP 事务的工作方式相同，只不过只包含一个数据包。假设：

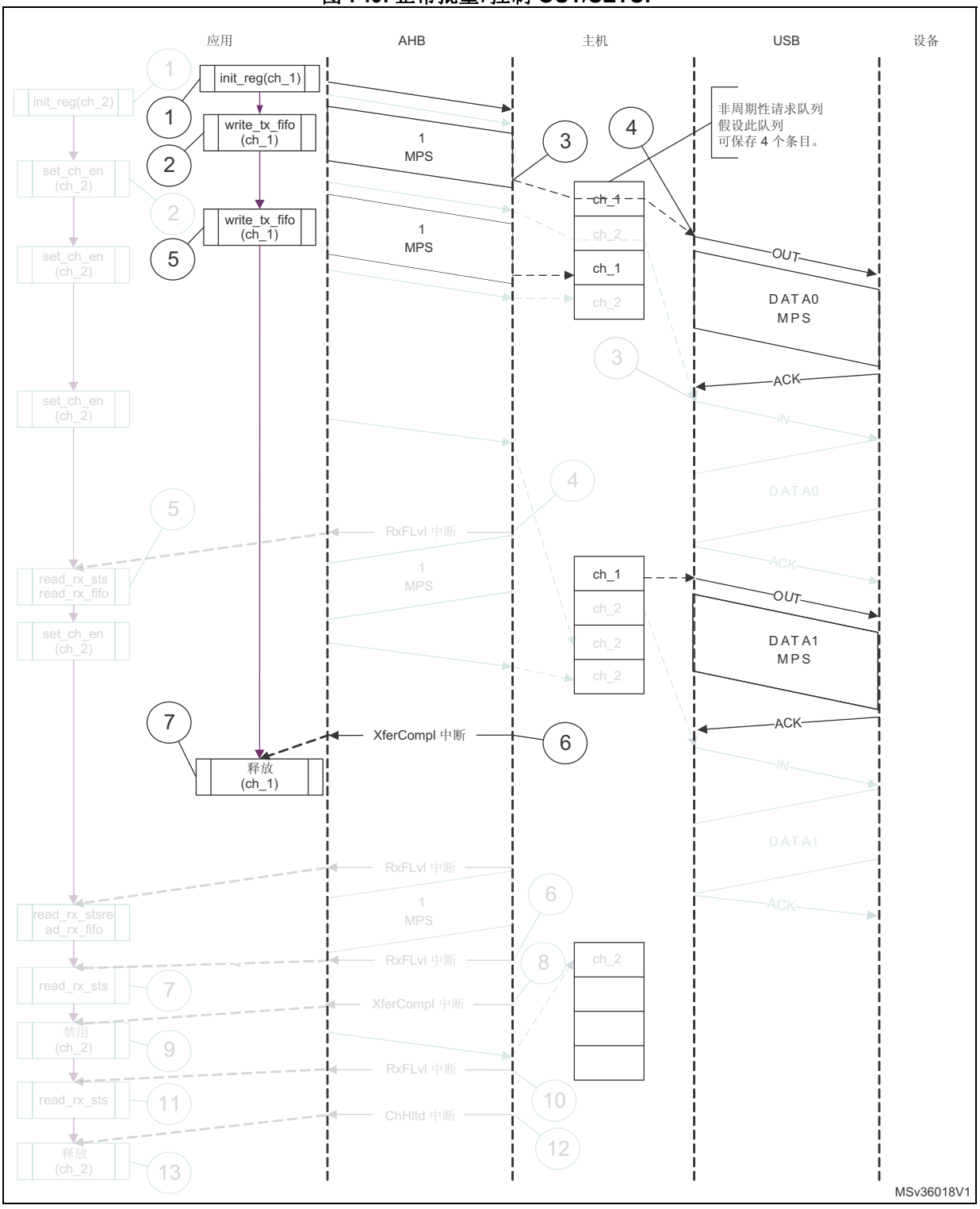
  – 应用程序尝试发送两个最大数据包大小的数据包（传输大小 = 1024 字节）。
  – 非周期性发送 FIFO 可存储两个数据包（HS 模式下为 1 KB）。
  – 非周期性请求队列深度 = 4。

- **正常批量和控制传输类型的 OUT/SETUP 操作**

  （通道 1）中的操作顺序如下：

1. 初始化通道 1
2. 写入通道 1 的第一个数据包
3. 在应用执行最后一次 DWORD 写操作时，模块将向非周期性请求队列写入一个请求条目
4. 只要非周期性队列非空，模块即会尝试在当前帧内发送一个 OUT 令牌
5. 写入通道 1 的第二个（最后一个）数据包
6. 成功完成最后一个事务后，模块立即生成 XFRC 中断
7. 为了响应 XFRC 中断，将释放通道以供其它传输操作使用
8. 处理非 ACK 响应

图 749. 正常批量/控制 OUT/SETUP



1. 灰显元素与此图的上下文不相关。

以下代码示例说明了批量和控制 OUT/SETUP 传输类型的通信事务的通道相关中断服务程序。

- **批量/控制 OUT/SETUP 和批量/控制 IN 事务的中断服务程序**

  a) 批量/控制 OUT/SETUP

```
Unmask (NAK/TXERR/STALL/XFRC)
if (XFRC)
   {
   Reset Error Count
   Mask ACK
   De-allocate Channel
   }
else if (STALL)
   {
   Transfer Done = 1
   Unmask CHH
   Disable Channel
   }
else if (NAK or TXERR )
   {
   Rewind Buffer Pointers
   Unmask CHH
   Disable Channel
   if (TXERR)
      {
      Increment Error Count
      Unmask ACK
      }
   else
      {
      Reset Error Count
      }
      }
else if (CHH)
   {
   Mask CHH
   if (Transfer Done or (Error_count == 3))
      {
      De-allocate Channel
      }
   else
      {
      Re-initialize Channel
      }
   }
   else if (ACK)
      {
      Reset Error Count
      Mask ACK
```

```
    }
```

当发送 FIFO 和请求队列中有可用空间时，应用程序会将数据包写入发送 FIFO。应用程序可利用 OTG_GINTSTS 中的 NPTXFE 中断确定发送 FIFO 空间。

b) 批量/控制 IN

```
Unmask (TXERR/XFRC/BBERR/STALL/DTERR)
if (XFRC)
   {
   Reset Error Count
   Unmask CHH
   Disable Channel
   Reset Error Count
   Mask ACK
   }
else if (TXERR or BBERR or STALL)
   {
   Unmask CHH
   Disable Channel
   if (TXERR)
      {
      Increment Error Count
      Unmask ACK
      }
   }
else if (CHH)
   {
   Mask CHH
   if (Transfer Done or (Error_count == 3))
      {
      De-allocate Channel
      }
   else
      {
      Re-initialize Channel
      }
   }
else if (ACK)
   {
   Reset Error Count
   Mask ACK
   }
else if (DTERR)
   {
   Reset Error Count
   }
```

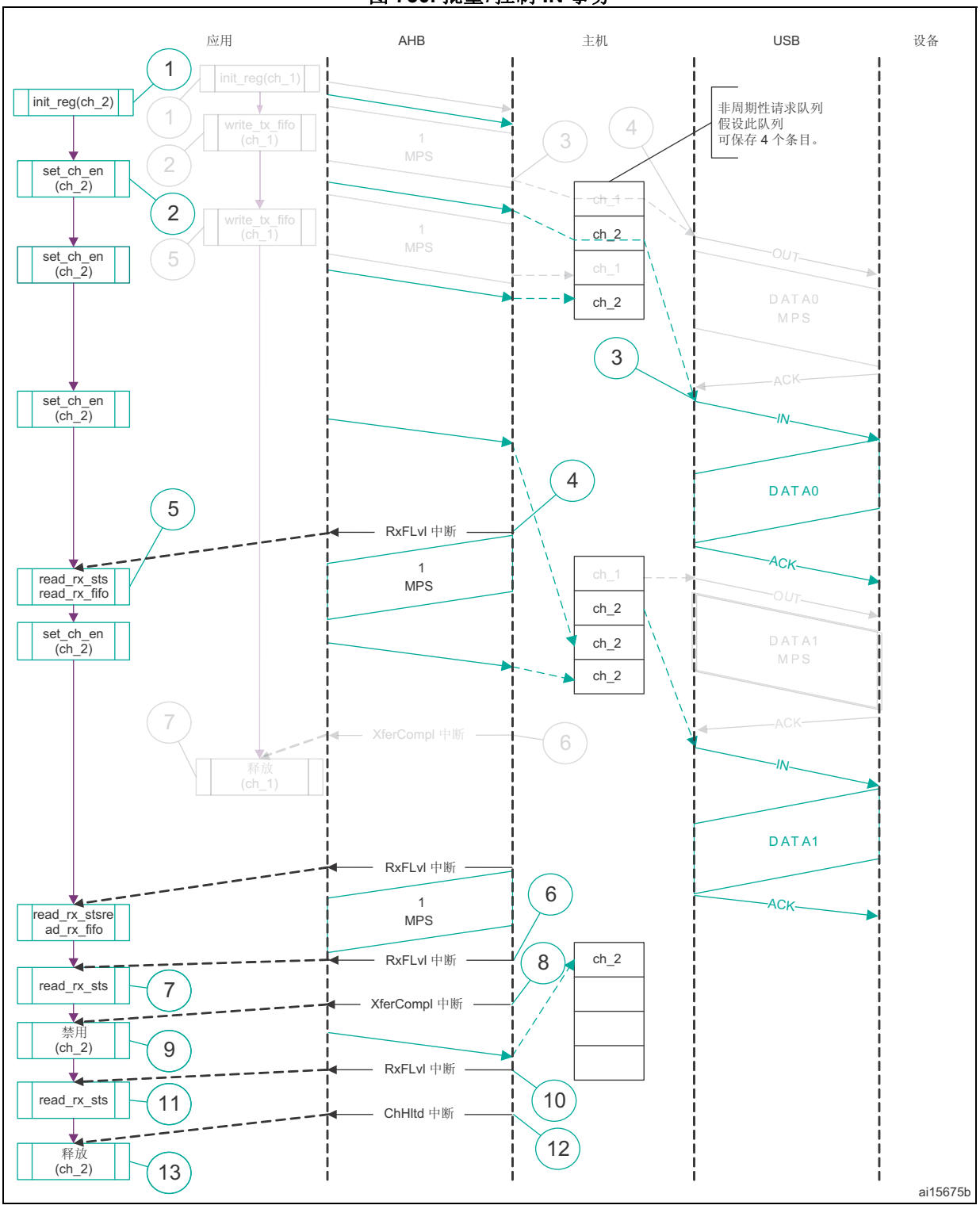
当请求队列空间可用时，应用程序会写入请求，直到接收到 XFRC 中断。

- **批量和控制 IN 事务**

  图 750 显示了典型的批量或控制传输类型的 IN 流水线事务级操作。请参见通道 2 (ch_2)。假设：

  – 应用程序要接收两个最大数据包大小的数据包（传输大小 = 1024 字节）。
  – 接收 FIFO 可以包含至少一个最大数据包大小的数据包和每个数据包的两个状态字（HS 对应 520 字节）。
  – 非周期性请求队列深度 = 4。

图 750. 批量/控制 IN 事务



1. 灰显元素与此图的上下文不相关。

操作顺序如下：

1. 初始化通道 2。
2. 将 OTG_HCCHAR2 中的 CHENA 位置 1，以向非周期性请求队列写入 IN 请求。
3. 模块尝试在完成当前 OUT 事务后发送 IN 令牌。
4. 接收到的数据包写入接收 FIFO 后，模块立即生成 RXFLVL 中断。
5. 为了响应 RXFLVL 中断，将屏蔽 RXFLVL 中断并读取接收到的数据包状态，以此确定接收的字节数，然后相应地读取接收 FIFO。之后使能 RXFLVL 中断。
6. 模块针对接收 FIFO 中的传输完成状态条目生成 RXFLVL 中断。
7. 应用程序必须读取接收数据包状态，并在不是 IN 数据包（OTG_GRXSTSR 中的 PKTSTS ≠ 0b0010）时忽略它。
8. 读取接收数据包状态后，模块立即生成 XFRC 中断。
9. 为了响应 XFRC 中断，将禁止通道并停止针对其它请求向 OTG_HCCHAR2 写入数据。向 OTG_HCCHAR2 寄存器写入数据时，模块立即向非周期性请求队列写入通道禁止请求。
10. 停止状态写入接收 FIFO 后，模块立即生成 RXFLVL 中断。
11. 读取并忽略接收数据包状态。
12. 只要停止状态从接收 FIFO 弹出，模块立即生成 CHH 中断。
13. 为了响应 CHH 中断，将释放通道以供其它传输操作使用。
14. 处理非 ACK 响应

- **控制事务**

  控制传输的建立、数据和状态阶段必须作为三个独立的传输过程来执行。建立、数据和状态阶段的 OUT 事务与上文所述的批量 OUT 事务的执行方式类似。数据或状态阶段的 IN 事务与上文所述的批量 IN 事务的执行方式类似。在所有这三个阶段，应用程序都会将 OTG_HCCHAR1 中的 EPTYP 字段设置为控制传输类型。在建立阶段期间，应用程序会将 OTG_HCTSIZ1 中的 PID 字段设置为 SETUP。

- **中断 OUT 事务**

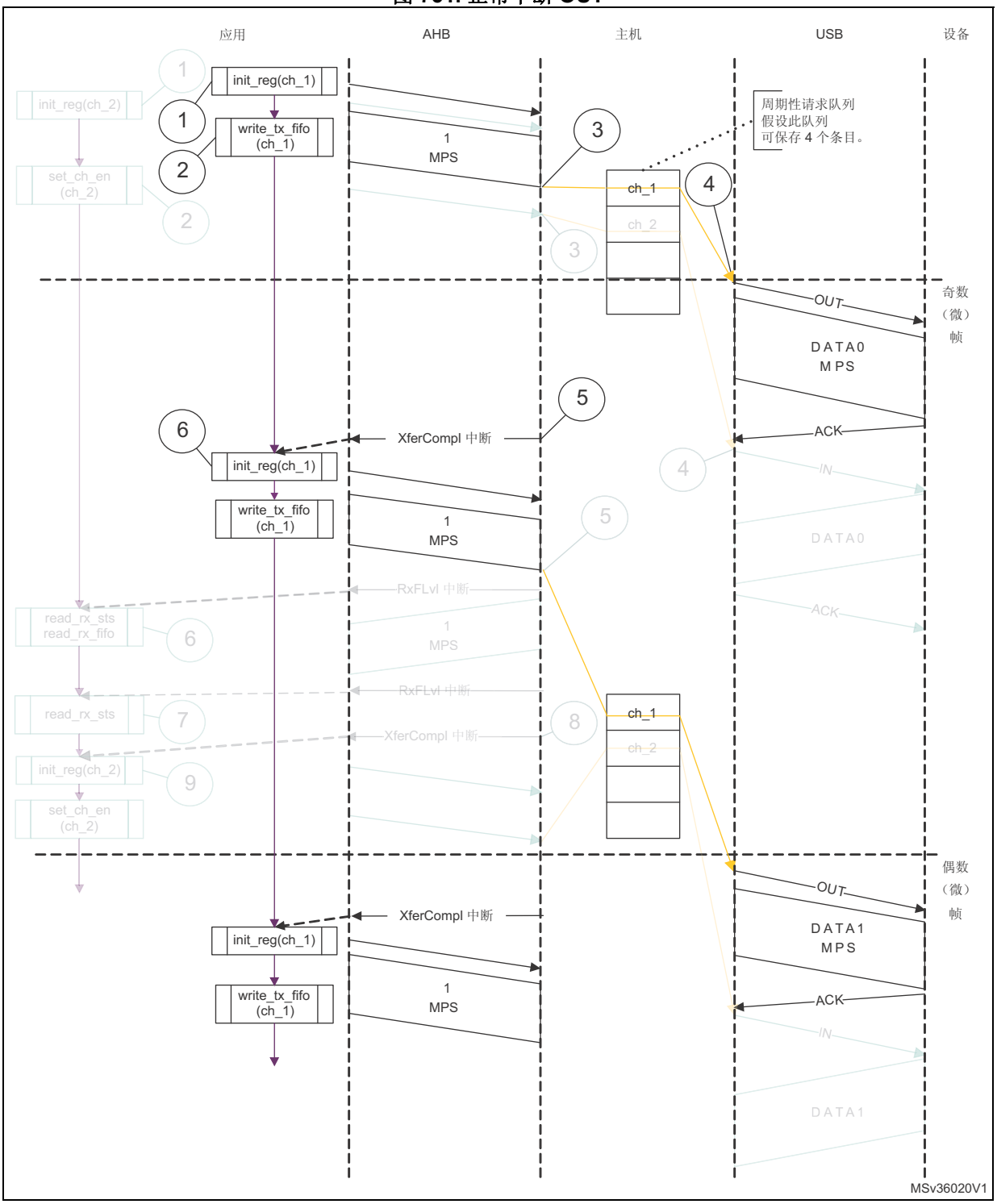
  图 751 显示了典型的中断 OUT 操作。假设：

  – 应用程序尝试从奇数帧开始，每帧发送一个数据包（高达 1 个最大数据包大小）（传输大小 = 1024 字节）
  – 周期性发送 FIFO 可存储一个数据包 (1 KB)
  – 周期性请求队列深度 = 4

  操作顺序如下：

1. 初始化和使能通道 1。应用程序必须将 OTG_HCCHAR1 中的 ODDFRM 位置 1。
2. 写入通道 1 的第一个数据包。
3. 应用程序在执行每个数据包的最后一次 DWORD 写操作时，OTG_HS 主机向周期性请求队列写入一个请求条目。
4. OTG_HS 主机尝试在下一（奇数）帧发送 OUT 令牌。
5. 成功发送最后一个数据包后，OTG_HS 主机立即生成 XFRC 中断。
6. 为了响应 XFRC 中断，将重新初始化通道以供下一传输操作使用。

图 751. 正常中断 OUT



1. 灰显元素与此图的上下文不相关。

- **中断 OUT/IN 事务的中断服务程序**

  a) 中断 OUT

```
Unmask (NAK/TXERR/STALL/XFRC/FRMOR)
if (XFRC)
   {
   Reset Error Count
   Mask ACK
   De-allocate Channel
   }
else
   if (STALL or FRMOR)
      {
      Mask ACK
      Unmask CHH
      Disable Channel
      if (STALL)
         {
         Transfer Done = 1
         }
      }
   else
      if (NAK or TXERR)
         {
         Rewind Buffer Pointers
         Reset Error Count
         Mask ACK
         Unmask CHH
         Disable Channel
         }
      else
         if (CHH)
         {
         Mask CHH
         if (Transfer Done or (Error_count == 3))
            {
            De-allocate Channel
            }
         else
            {
            Re-initialize Channel (in next b_interval - 1 Frame)
            }
         }
      else
         if (ACK)
         {
         Reset Error Count
         Mask ACK
         }
```

  应用程序利用 OTG_GINTSTS 中的 NPTXFE 中断确定发送 FIFO 空间。

  b) 中断 IN

```
Unmask (NAK/TXERR/XFRC/BBERR/STALL/FRMOR/DTERR)
if (XFRC)
   {
   Reset Error Count
   Mask ACK
   if (OTG_HCTSIZx.PKTCNT == 0)
      {
      De-allocate Channel
      }
   else
      {
      Transfer Done = 1
      Unmask CHH
      Disable Channel
      }
   }
else
   if (STALL or FRMOR or NAK or DTERR or BBERR)
      {
      Mask ACK
      Unmask CHH
      Disable Channel
      if (STALL or BBERR)
         {
         Reset Error Count
         Transfer Done = 1
         }
      else
         if (!FRMOR)
         {
         Reset Error Count
         }
      }
else
   if (TXERR)
      {
      Increment Error Count
      Unmask ACK
      Unmask CHH
      Disable Channel
      }
else
   if (CHH)
      {
      Mask CHH
      if (Transfer Done or (Error_count == 3))
         {
         De-allocate Channel
         }
      else
         Re-initialize Channel (in next b_interval - 1 /Frame)
      }
```

```
   }
else
   if (ACK)
     {
     Reset Error Count
     Mask ACK
     }
```

- **中断 IN 事务**

  假设：

  – 应用程序要从奇数帧开始，每帧接收一个数据包（最大 1 个最大数据包大小）（传输大小 = 1024 字节）。

  – 接收 FIFO 可以保存至少一个最大数据包大小的数据包和每个数据包的两个状态字（1031 字节）。

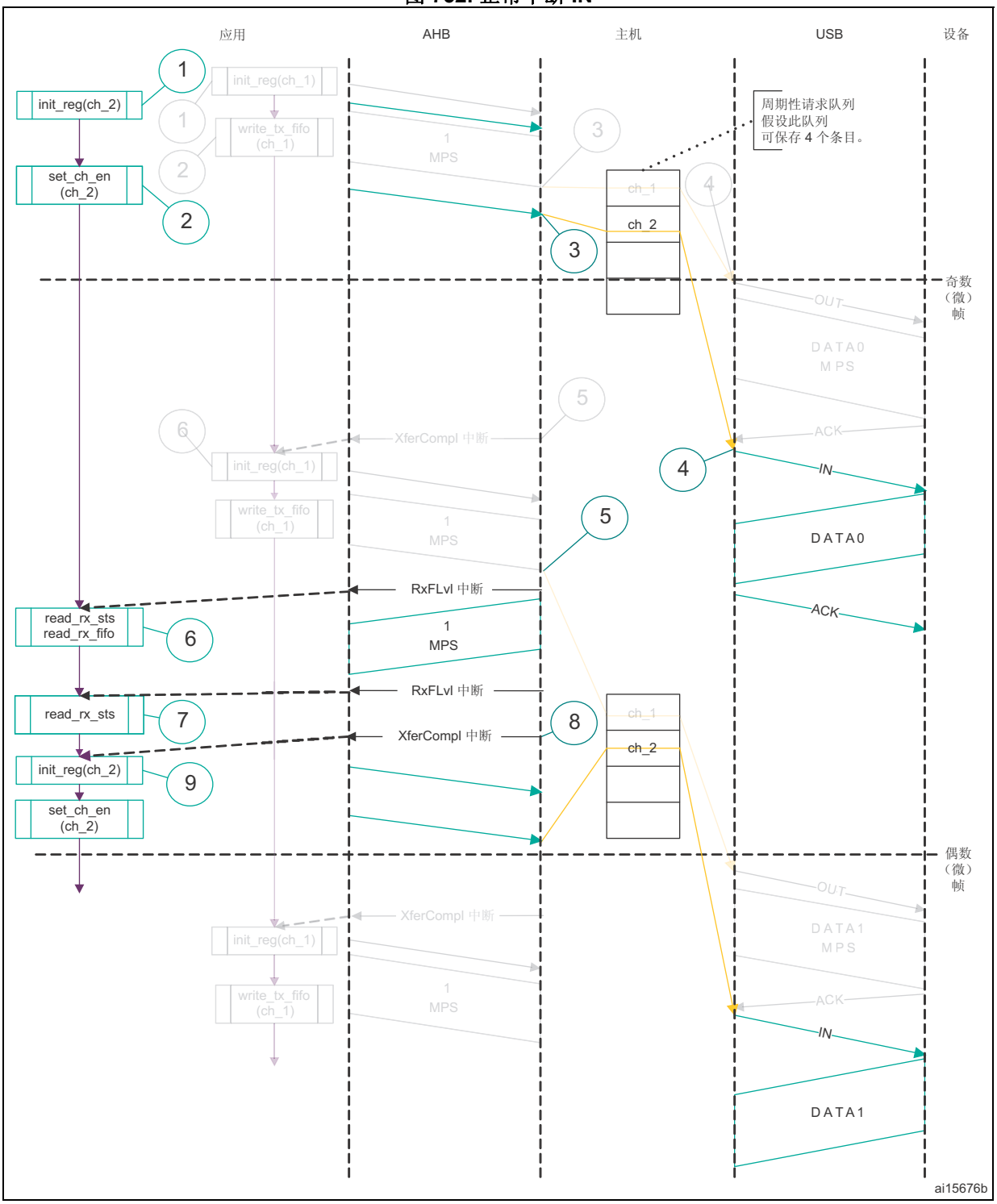
  – 周期性请求队列深度 = 4。

- **正常中断 IN 操作**

  操作顺序如下：

1. 初始化通道 2。应用程序必须将 OTG_HCCHAR2 中的 ODDFRM 位置 1。
2. 将 OTG_HCCHAR2 中的 CHENA 位置 1，以将 IN 请求写入周期性请求队列。
3. 只要 CHENA 置位，对于每次 OTG_HCCHAR2 寄存器写操作,OTG_HS 主机都会将一个 IN 请求写入周期性请求队列。
4. OTG_HS 主机尝试在下一奇数帧发送 IN 令牌。
5. 接收到 IN 数据包并写入接收 FIFO 后，OTG_HS 主机便会立即生成 RXFLVL 中断。
6. 为响应 RXFLVL 中断，读取接收到的数据包状态以确定接收的字节数，然后相应地读取接收 FIFO。应用程序必须在读取接收 FIFO 前屏蔽 RXFLVL 中断，并且在读取整个数据包后使能。
7. 模块针对接收 FIFO 中的传输完成状态条目生成 RXFLVL 中断。应用程序必须读取接收数据包状态，并在不是 IN 数据包（GRXSTSR 中的 PKTSTS ≠ 0b0010）时忽略它。
8. 读取接收数据包状态后，模块立即生成 XFRC 中断。
9. 为响应 XFRC 中断，将读取 OTG_HCTSIZ2 中的 PKTCNT 字段。如果 OTG_HCTSIZ2 中的 PKTCNT 位不等于 0，则在重新初始化通道以进行下次传输前（如果存在），禁止该通道。如果 OTG_HCTSIZ2 中的 PKTCNT 位等于 0，则重新初始化通道以进行下次传输。此时，应用程序必须复位 OTG_HCCHAR2 中的 ODDFRM 位。

图 752. 正常中断 IN



1. 灰显元素与此图的上下文不相关。

- **同步 OUT 事务**

  图 752 中显示了典型的同步 OUT 操作。假设：

  – 应用程序尝试从奇数帧开始，每帧发送一个数据包（最大 1 个最大数据包大小）。（传输大小 = 1024 字节）。

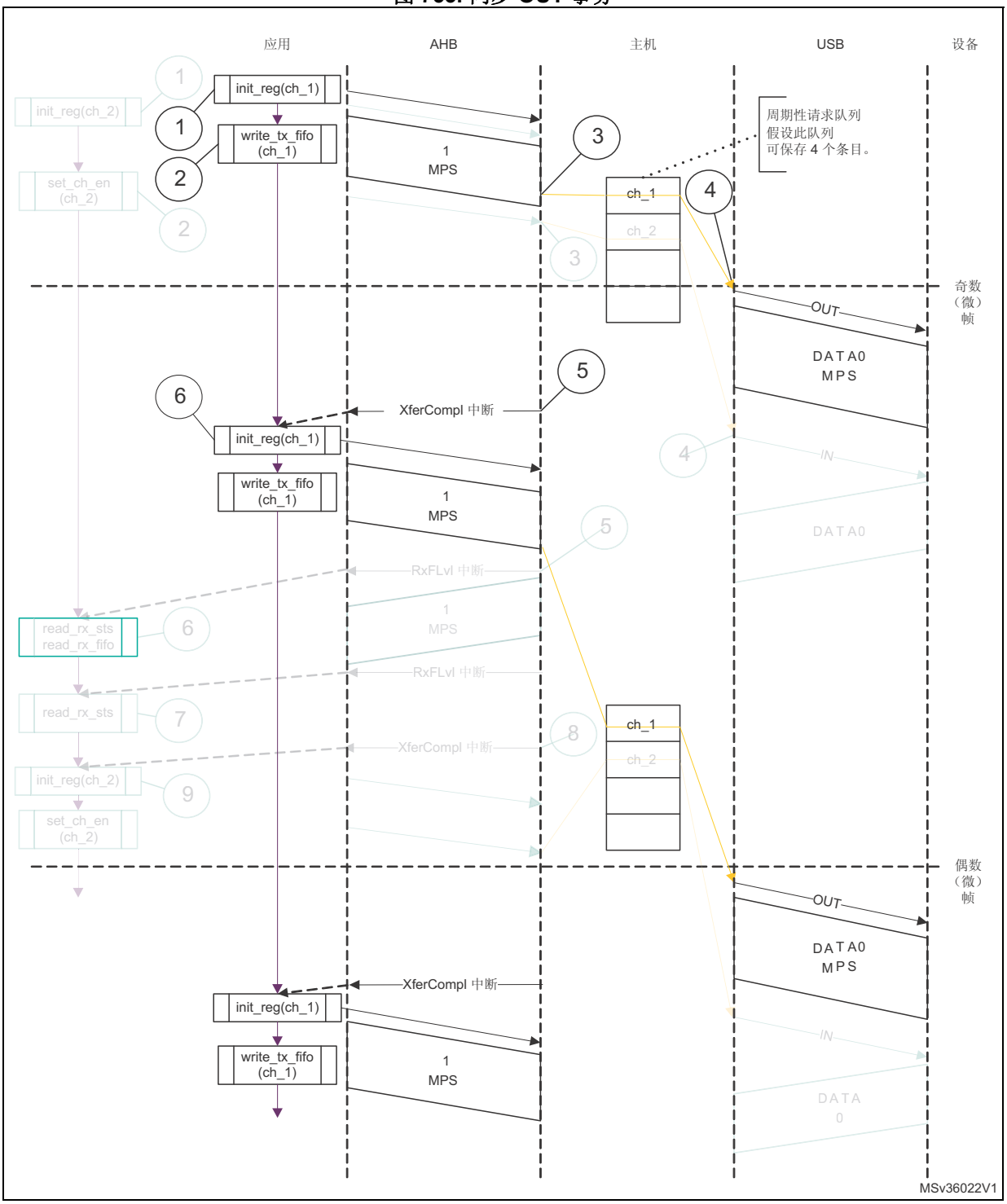
  – 周期性发送 FIFO 可存储一个数据包 (1 KB)。

  – 周期性请求队列深度 = 4。

  操作顺序如下：

1. 初始化和使能通道 1。应用程序必须将 OTG_HCCHAR1 中的 ODDFRM 位置 1。
2. 写入通道 1 的第一个数据包。
3. 应用程序在执行每个数据包的最后一次 DWORD 写操作时，OTG_HS 主机向周期性请求队列写入一个请求条目。
4. OTG_HS 主机尝试在下一（奇数）帧发送 OUT 令牌。
5. 成功发送最后一个数据包后，OTG_HS 主机立即生成 XFRC 中断。
6. 为了响应 XFRC 中断，将重新初始化通道以供下一传输操作使用。
7. 处理非 ACK 响应。

图 753. 同步 OUT 事务



1. 灰显元素与此图的上下文不相关。

- **同步 OUT/IN 事务的中断服务程序**

  代码示例：同步 OUT

```
Unmask (FRMOR/XFRC)
if (XFRC)
   {
   De-allocate Channel
   }
else
   if (FRMOR)
      {
      Unmask CHH
      Disable Channel
      }
   else
   if (CHH)
      {
      Mask CHH
      De-allocate Channel
      }
```

  代码示例：同步 IN

```
Unmask (TXERR/XFRC/FRMOR/BBERR)
if (XFRC or FRMOR)
   {
   if (XFRC and (OTG_HCTSIZx.PKTCNT == 0))
      {
      Reset Error Count
      De-allocate Channel
      }
   else
      {
      Unmask CHH
      Disable Channel
      }
   }
   else
      if (TXERR or BBERR)
         {
         Increment Error Count
         Unmask CHH
         Disable Channel
         }
      else
         if (CHH)
            {
            Mask CHH
            if (Transfer Done or (Error_count == 3))
               {
               De-allocate Channel
               }
            else
               {
```

```
        Re-initialize Channel
        }
    }
```

- **同步 IN 事务**

  假设：

  – 应用程序尝试从下一个奇数帧开始，每帧接收一个数据包（最大 1 个最大数据包大小）（传输大小 = 1024 字节）。

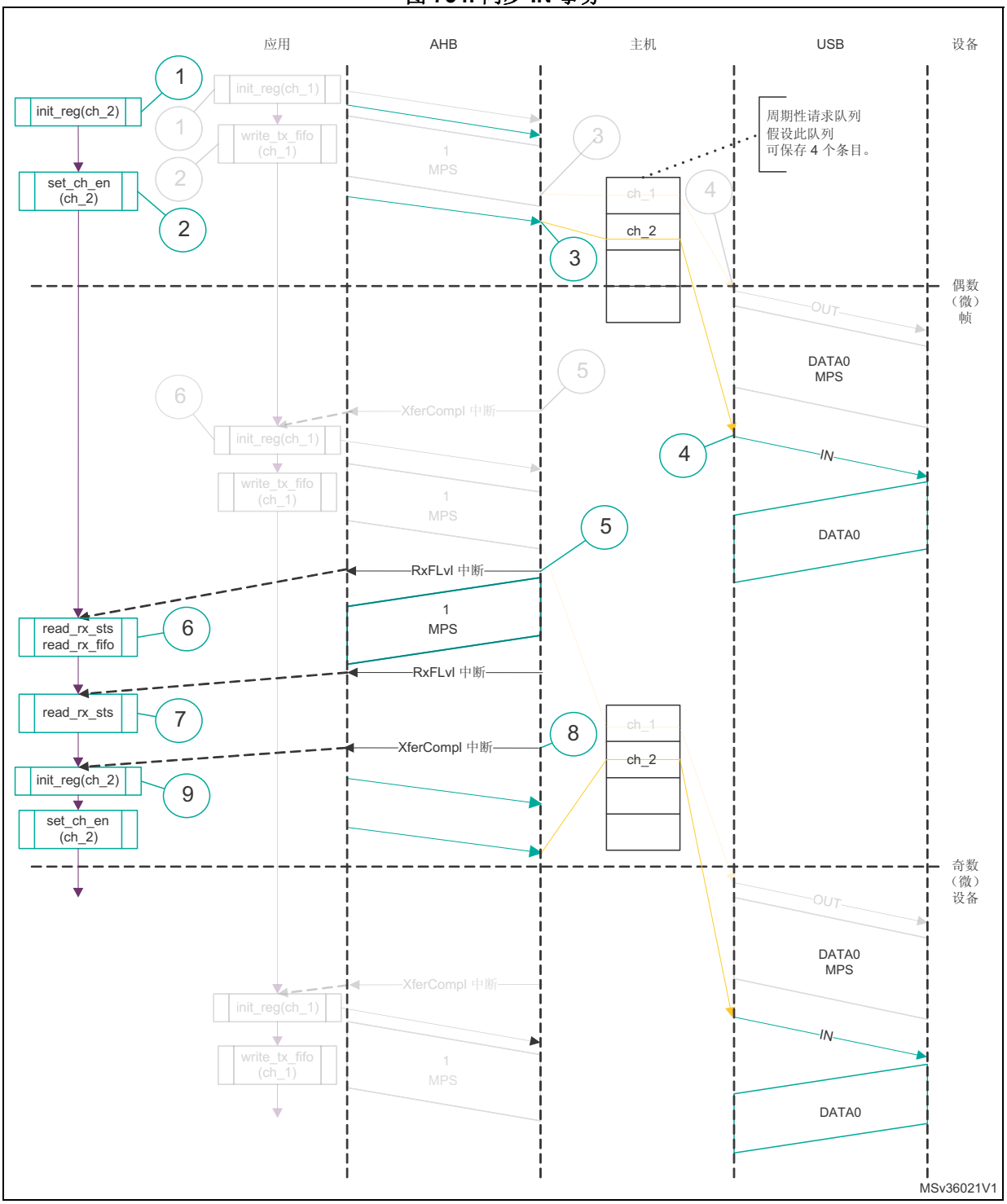
  – 接收 FIFO 可以保存至少一个最大数据包大小的数据包和每个数据包的两个状态字（1031 字节）。

  – 周期性请求队列深度 = 4。

  操作顺序如下：

1. 初始化通道 2。应用程序必须将 OTG_HCCHAR2 中的 ODDFRM 位置 1。
2. 将 OTG_HCCHAR2 中的 CHENA 位置 1，以将 IN 请求写入周期性请求队列。
3. 只要 CHENA 置位，对于每次 OTG_HCCHAR2 寄存器写操作,OTG_HS 主机都会将一个 IN 请求写入周期性请求队列。
4. OTG_HS 主机尝试在下一（奇数）帧发送 IN 令牌。
5. 接收到 IN 数据包并写入接收 FIFO 后，OTG_HS 主机便会立即生成 RXFLVL 中断。
6. 为响应 RXFLVL 中断，读取接收到的数据包状态以确定接收的字节数，然后相应地读取接收 FIFO。应用程序必须在读取接收 FIFO 前屏蔽 RXFLVL 中断，并且在读取整个数据包后使能。
7. 模块针对接收 FIFO 中的传输完成状态条目生成 RXFLVL 中断。应用程序必须读取接收数据包状态，并在不是 IN 数据包（OTG_GRXSTSR 中的 PKTSTS 位 ≠ 0b0010）时忽略它。
8. 读取接收数据包状态后，模块立即生成 XFRC 中断。
9. 为响应 XFRC 中断，将读取 OTG_HCTSIZ2 中的 PKTCNT 字段。如果在 OTG_HCTSIZ2 中的 PKTCNT ≠ 0，则在重新初始化通道以进行下次传输前（如果存在），禁止该通道。如果 OTG_HCTSIZ2 中的 PKTCNT = 0，则重新初始化通道以进行下次传输。此时，应用程序必须复位 OTG_HCCHAR2 中的 ODDFRM 位。

图 754. 同步 IN 事务



1. 灰显元素与此图的上下文不相关。

- **选择队列深度**

  请谨慎选择周期性和非周期性请求队列深度，以与要访问的周期性/非周期性端点数量相匹配。

  非周期性请求队列深度会影响非周期性传输的性能。队列越深（加上足够的 FIFO 大小），模块就更能对非周期性传输进行流水线处理。如果队列大小太小，则仅在队列空间释放时模块才能放入新请求。

  要按调度计划执行周期性传输，模块的周期性请求队列深度至关重要。根据一个微帧内要安排的周期性传输次数，选择周期性队列深度。如果周期性请求队列深度小于一个微帧内要安排的周期性传输数，则会发生帧溢出情况。

- **处理串扰情况**

  OTG_HS 控制器处理两种情况的 babble：数据包 babble 和端口 babble。如果设备发送的数据量超过通道的最大数据包大小，则会发生数据包串扰。如果模块从 EOF2（非常接近下一帧的 SOF）时刻还在从设备接收数据，则发生端口串扰。

  当 OTG_HS 控制器检测到数据包 babble 时，它停止向 Rx 缓冲区中写入数据并等待数据包结束信号 (EOP)。当该控制器检测到 EOP 时，它会清空 Rx 缓冲区中的已写入数据并对应用程序生成串扰中断。

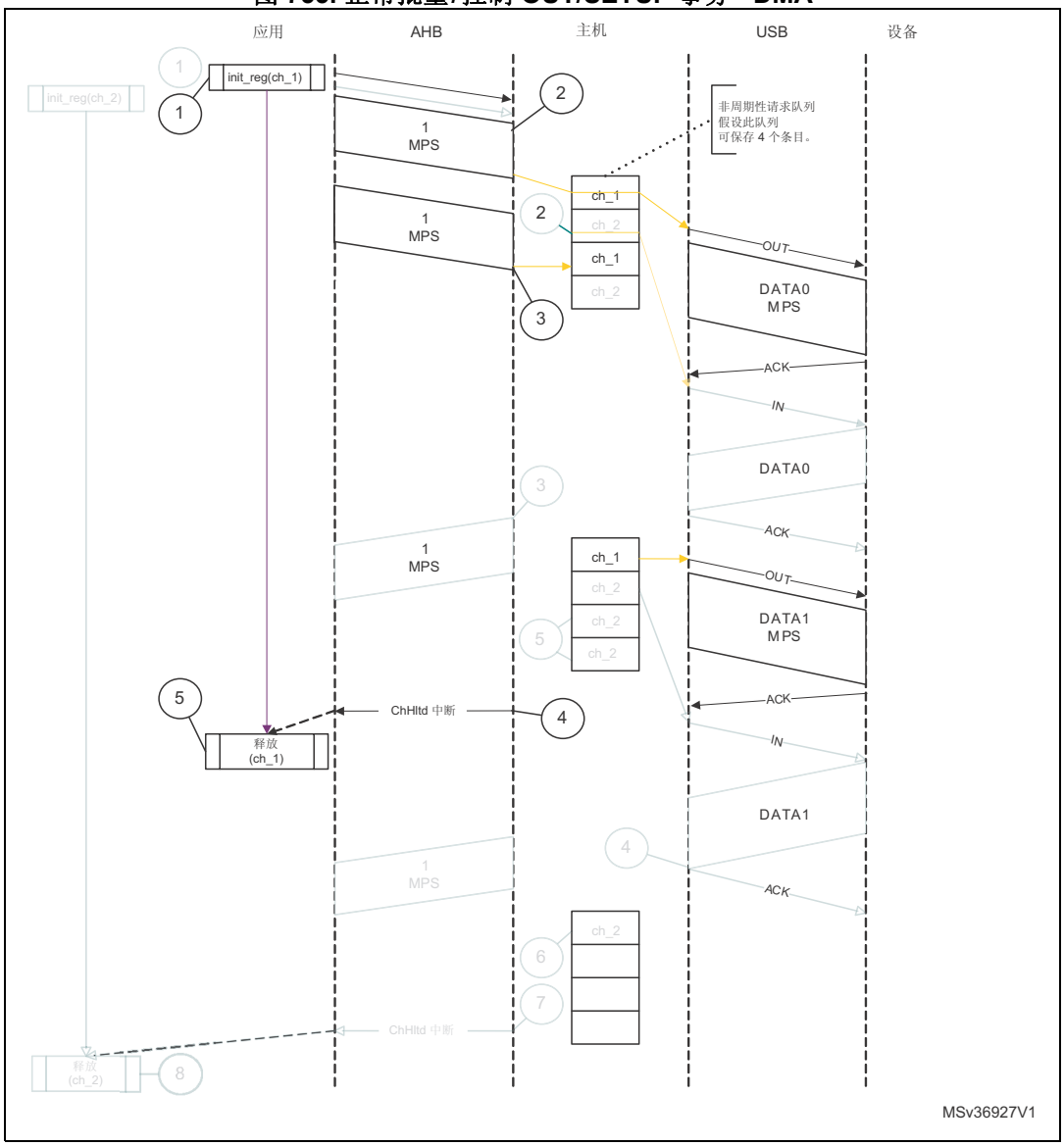
  当 OTG_HS 控制器检测到端口 babble 时，它会清空 RxFIFO 并禁止端口。模块随后将生成“端口已禁止”中断（OTG_GINTSTS 中的 HPRTINT 位和 OTG_HPRT 中的 PENCHNG 位）。在收到该中断时，应用程序必须通过检查 OTG_HPRT 中的 POCA 位，来确定该中断并非由过流（“端口已禁止”中断的另一个原因）所致，然后执行软复位。检测到端口串扰情况后，模块不再发送任何其它令牌。

- **DMA 模式下的批量和控制 OUT/SETUP 事务**

  操作顺序如下：

1. 按照*通道初始化*一节中的说明初始化并使能通道 1。
2. 使能通道后，OTG_HS 主机即开始获取第一个数据包以发送。对于内部 DMA 模式，OTG_HS 主机使用经过编程的 DMA 地址来获取数据包。
3. 获取第二个（最后一个）数据包的最后一个 DWORD 后，OTG_HS 主机会在内部屏蔽通道 1，不参与仲裁。
4. 最后一个数据包发送出去后，OTG_HS 主机立即生成一个 CHH 中断。
5. 为了响应 CHH 中断，将释放通道以供其它传输操作使用。

图 755. 正常批量/控制 OUT/SETUP 事务 - DMA



- **使用内部 DMA 处理 NAK 和 NYET：**

1. OTG_HS 主机发送批量传输类型的 OUT 事务。
2. 设备回复 NAK 或 NYET。
3. 如果应用程序没有屏蔽 NAK 或 NYET 中断，模块将为应用程序生成相应的中断。应用程序并不需要处理这些中断，因为模块会负责调整缓冲区指针和重新初始化通道，而无需应用程序干预。
4. 模块自动发出一个 ping 令牌。
5. 当设备返回 ACK 后，模块会继续传输。应用程序也可以选择使用这些中断（在这种情况下，NAK 或 NYET 中断将被应用程序屏蔽）。

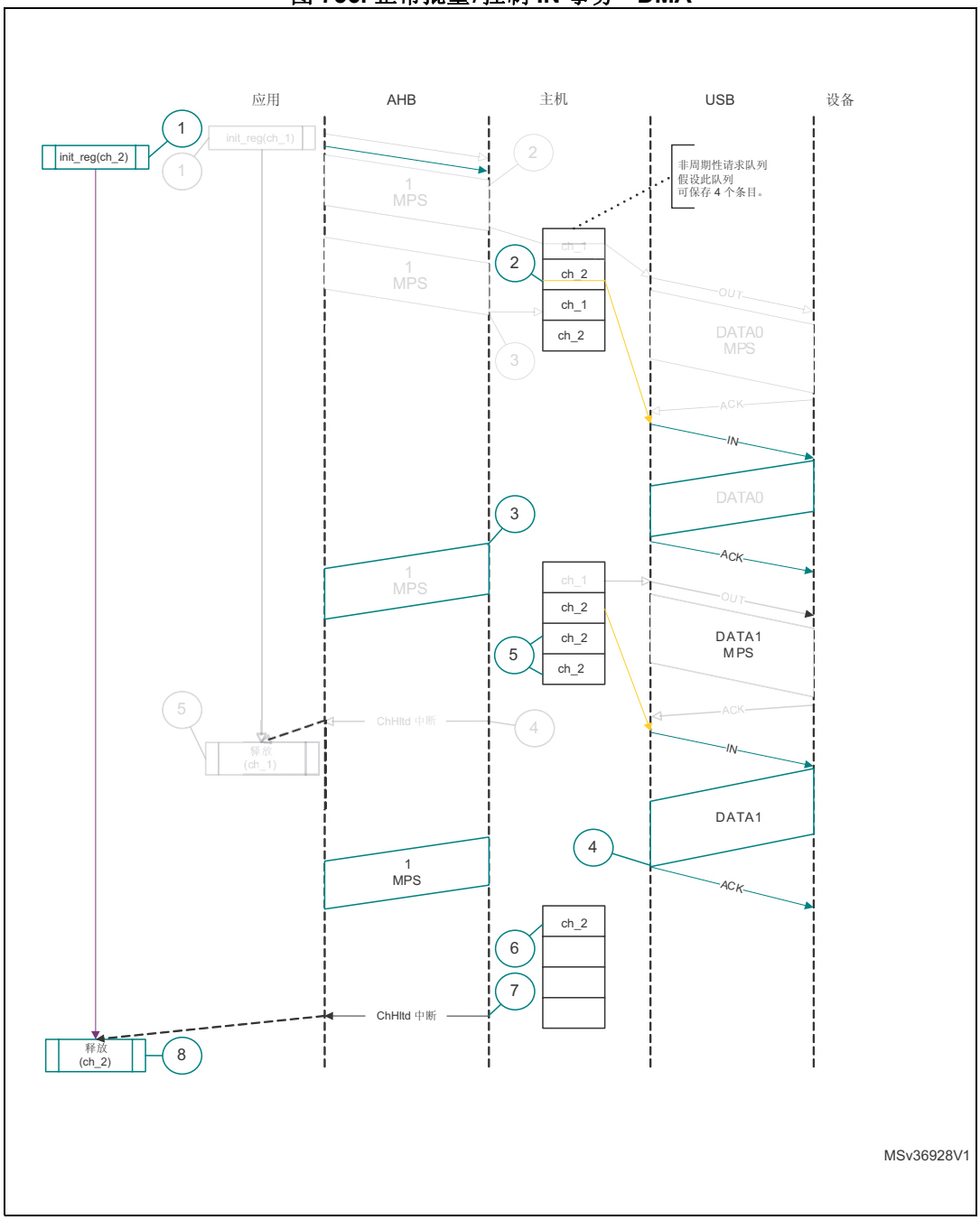
   当主机接收到 NAK 或 NYET 时，模块不会生成单独的中断。

- **DMA 模式下的批量和控制 IN 事务**

  操作顺序如下：

1. 按照*通道初始化*一节中的说明初始化并使能所使用的通道（通道 x）。
2. 当通道接收到来自仲裁器的授权后（以循环方式执行仲裁），OTG_HS 会立即向请求队列写入一个 IN 请求。
3. 当正确接收到最后一个字节后，OTG_HS 主机便开始将接收到的数据写入到系统存储器中。
4. 当接收到最后一个数据包后，OTG_HS 会将一个内部标志置 1，以便从请求队列中删除任何额外的 IN 请求。
5. OTG_HS 主机会清空额外请求。
6. 向请求队列写入最后一个禁止通道的请求。此时，会在内部屏蔽通道 2，不参与仲裁。
7. 当禁止请求到达队列顶端时，OTG_HS 主机即会生成 CHH 中断。
8. 为了响应 CHH 中断，将释放通道以供其它传输操作使用。

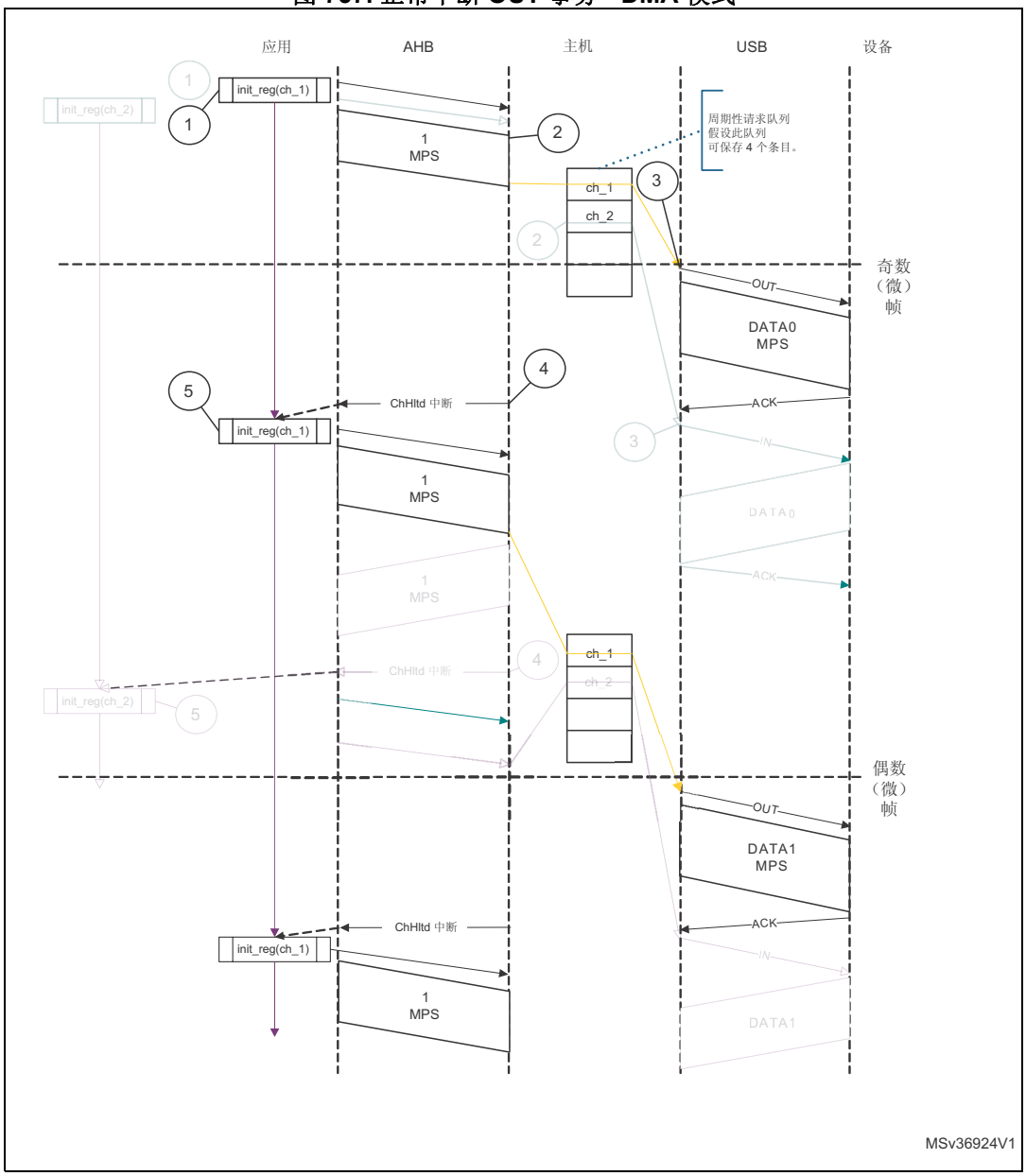
图 756. 正常批量/控制 IN 事务 - DMA



- **DMA 模式下的中断 OUT 事务**

1. 按照*通道初始化*一节中的说明初始化并使能通道 x。
2. 使能通道后，OTG_HS 主机开始获取第一个数据包以发送，并在获取数据包的最后一个 DWORD 后写入 OUT 请求。在高带宽传输模式下，OTG_HS 主机会继续获取下一个数据包（直至数据包数量达到 MC 字段中指定的数值），然后才能切换到下一个通道。
3. OTG_HS 主机尝试在下一个奇数帧/微帧的起始位置发送 OUT 令牌。
4. 成功发送数据包后，OTG_HS 主机将生成一个 CHH 中断。
5. 为了响应 CHH 中断，将重新初始化通道以供下一传输操作使用。

图 757. 正常中断 OUT 事务 - DMA 模式

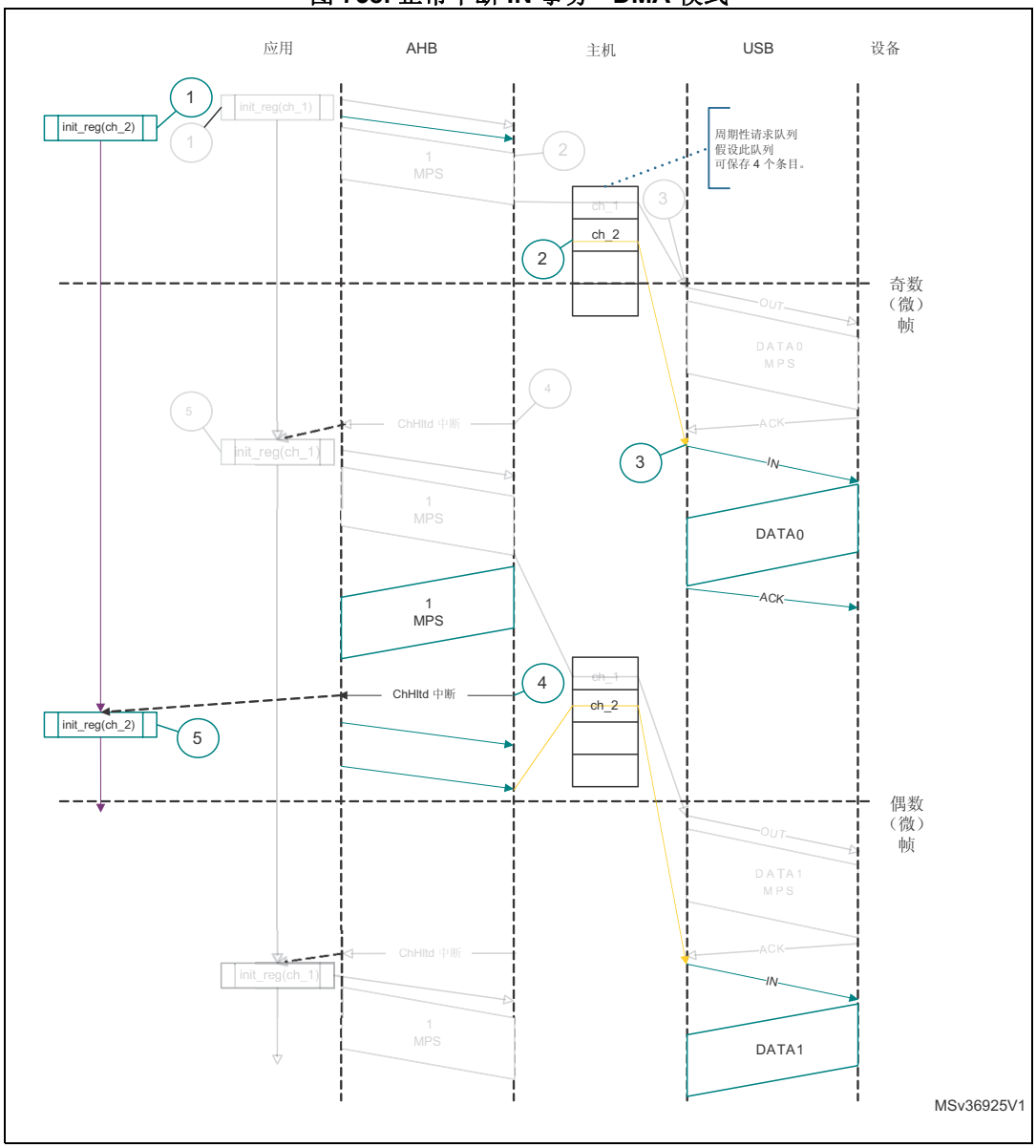


- **DMA 模式下的中断 IN 事务**

  通道 x 的操作顺序如下：

1. 按照*通道初始化*一节中的说明初始化并使能通道 x。
2. 当通道获取了来自仲裁器的授权后（以循环方式执行仲裁），OTG_HS 主机会立即向请求队列写入一个 IN 请求。在高带宽传输模式下，OTG_HS 主机将执行连续写入操作，直到写入的数据包个数达到 MC。
3. OTG_HS 主机尝试在下一个（奇数）帧/微帧的起始位置发送 IN 令牌。
4. 接收到 IN 数据包并写入接收 FIFO 后，OTG_HS 主机便会立即生成 CHH 中断。
5. 为了响应 CHH 中断，将重新初始化通道以供下一传输操作使用。

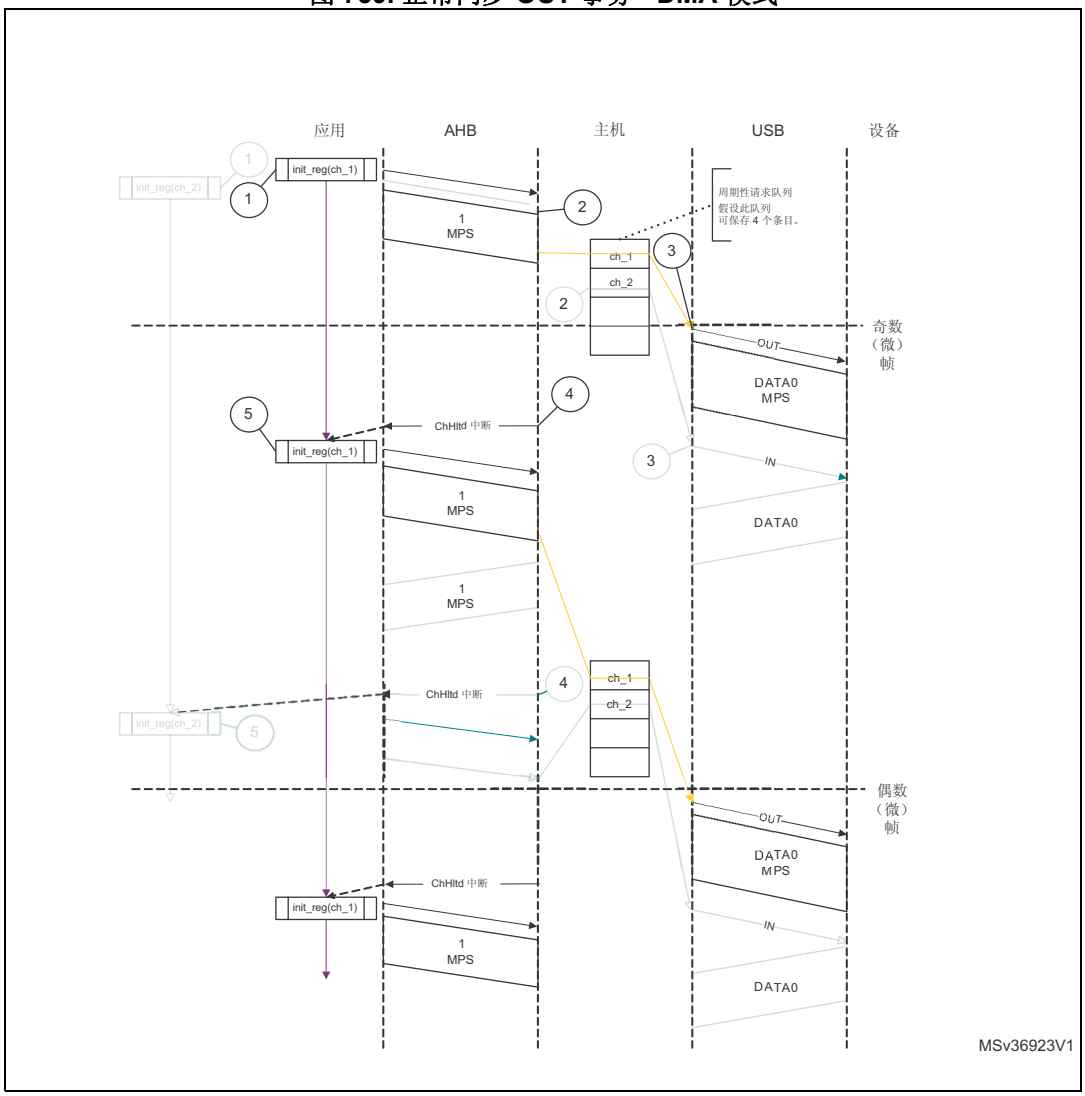
图 758. 正常中断 IN 事务 - DMA 模式



- **DMA 模式下的同步 OUT 事务**

1. 按照*通道初始化*一节中的说明初始化并使能通道 x。
2. 使能通道后，OTG_HS 主机开始获取第一个数据包，并在获取数据包的最后一个 DWORD 后写入 OUT 请求。在高带宽传输模式下，OTG_HS 主机会继续获取下一个数据包（直到写入的数据包个数达到 MC），然后才能切换到下一个通道。
3. OTG_HS 主机尝试在下一个（奇数）帧/微帧的起始位置发送 OUT 令牌。
4. 成功发送数据包后，OTG_HS 主机将生成一个 CHH 中断。
5. 为了响应 CHH 中断，将重新初始化通道以供下一传输操作使用。

图 759. 正常同步 OUT 事务 - DMA 模式

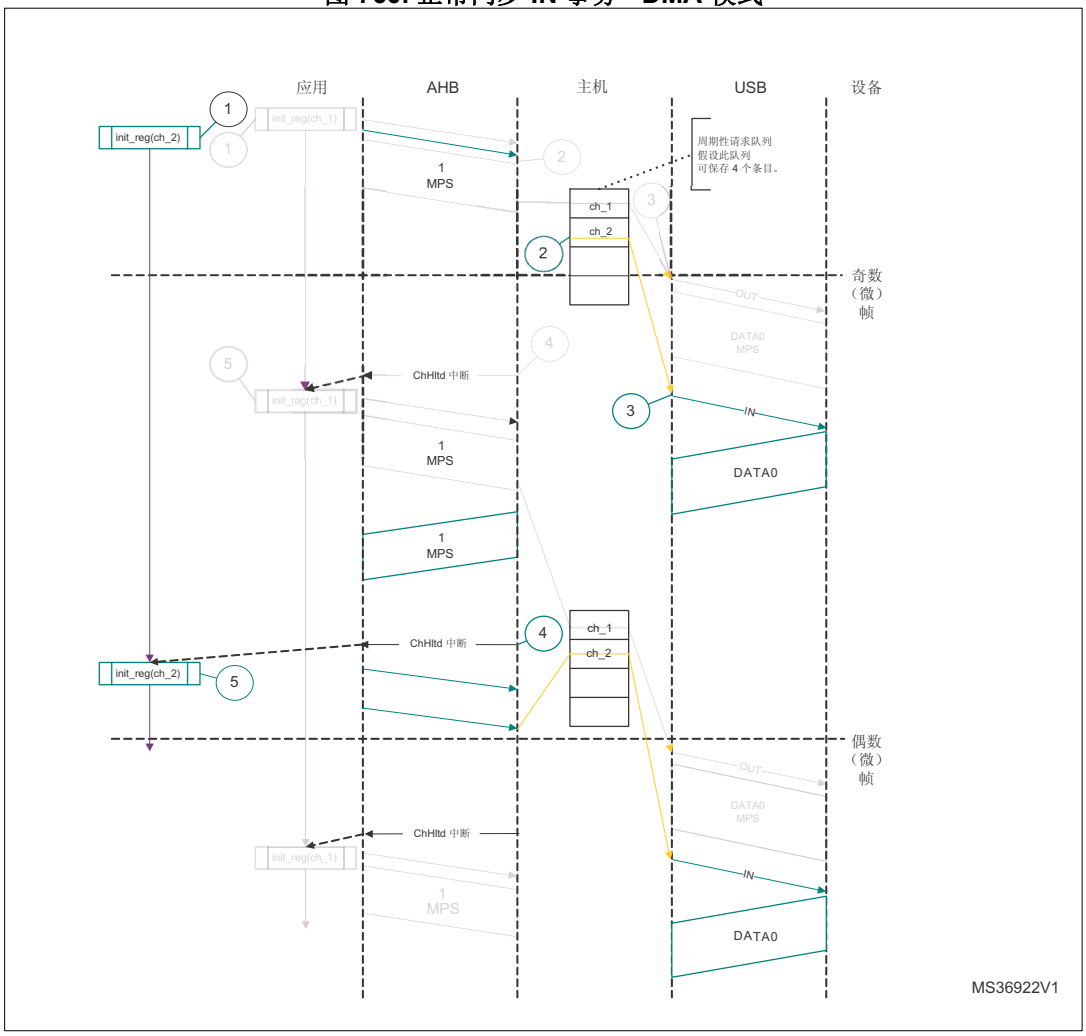


- **DMA 模式下的同步 IN 事务**

  通道 x 的操作顺序如下：

1. 按照*通道初始化*一节中的说明初始化并使能通道 x。
2. 当通道获取了来自仲裁器的授权后（以循环方式执行仲裁），OTG_HS 主机会立即向请求队列写入一个 IN 请求。在高带宽传输模式下，OTG_HS 主机将执行连续写入操作，直到写入的数据包个数达到 MC。
3. OTG_HS 主机尝试在下一个（奇数）帧/微帧的起始位置发送 IN 令牌。
4. 接收到 IN 数据包并写入接收 FIFO 后，OTG_HS 主机便会立即生成 CHH 中断。
5. 为了响应 CHH 中断，将重新初始化通道以供下一传输操作使用。

图 760. 正常同步 IN 事务 - DMA 模式



- **DMA 模式下的批量和控制 OUT/SETUP 分离事务**

（通道 x）中的操作顺序如下：

1. 按照*通道初始化*一节中的说明初始化并使能通道 x 以用于启动分离。
2. 使能通道后，OTG_HS 主机开始获取第一个数据包以发送，并在获取数据包的最后一个 DWORD 后写入 OUT 请求。
3. 成功发送启动分离后，OTG_HS 主机将生成 CHH 中断。
4. 作为对 CHH 中断的响应，将 OTG_HCSPLT1 中的 COMPLSPLT 位置 1，以发送完成分离。
5. 成功发送完成分离后，OTG_HS 主机将生成 CHH 中断。
6. 为了响应 CHH 中断，将会释放通道。

- **DMA 模式下的批量/控制 IN 分离事务**

通道 x 的操作顺序如下：

1. 按照*通道初始化*一节中的说明初始化并使能通道 x。
2. 获取了来自仲裁器的授权后，OTG_HS 主机将会向非周期性请求队列写入启动分离请求。写入该请求后，OTG_HS 主机将在内部屏蔽通道 x，不参与仲裁。
3. 发送 IN 令牌后，OTG_HS 主机立即生成 CHH 中断。
4. 作为对 CHH 中断的响应，将 OTG_HCSPLT2 中的 COMPLSPLT 位置 1 并重新使能通道，以发送完成分离令牌。这将使能通道 x，并让它参与后续仲裁。
5. 接收到来自仲裁器的授权后，OTG_HS 主机将会向非周期性请求队列写入完成分离请求。
6. 成功接收到数据包后，OTG_HS 主机将开始向系统存储器写入数据包。
7. 只要接收到的数据包写入系统存储器，OTG_HS 主机即会生成一个 CHH 中断。
8. 为了响应 CHH 中断，将会释放通道。

- **DMA 模式下的中断 OUT 分离事务**

（通道 x）中的操作顺序如下：

1. 按照*通道初始化*一节中的说明初始化并使能通道 1 以用于启动分离。应用程序必须将 OTG_HCCHAR1 中的 ODDFRM 位置 1。
2. OTG_HS 主机开始读取数据包。
3. OTG_HS 主机尝试发送启动分离事务。
4. 成功发送启动分离后，OTG_HS 主机将生成 CHH 中断。
5. 作为对 CHH 中断的响应，将 OTG_HCSPLT1 中的 COMPLSPLT 位置 1，以发送完成分离。
6. 成功结束完成分离事务后，OTG_HS 主机将生成 CHH 中断。
7. 为了响应 CHH 中断，将会释放通道。

- **DMA 模式下的中断 IN 分离事务**

（通道 x）中的操作顺序如下：

1. 按照*通道初始化*一节中的说明初始化并使能通道 x 以用于启动分离。
2. 当通道 x 接收到来自仲裁器的授权后，OTG_HS 主机会立即向请求队列写入一个 IN 请求。
3. OTG_HS 主机尝试在下一个奇数微帧的起始位置发送启动分离 IN 令牌。
4. 成功发送启动分离 IN 令牌后，OTG_HS 主机将生成 CHH 中断。
5. 作为对 CHH 中断的响应，将 OTG_HCSPLT2 中的 COMPLSPLT 位置 1，以发送完成分离。
6. 成功接收到数据包后，OTG_HS 主机即会开始向系统存储器写入数据。
7. 将接收到的数据传输到系统存储器后，OTG_HS 主机会生成 CHH 中断。
8. 为了响应 CHH 中断，将释放或重新初始化通道以供下一启动分离使用。

- **DMA 模式下的同步 OUT 分离事务**

  通道 x 的操作顺序如下：

1. 按照*通道初始化*一节中的说明初始化并使能通道 x 以用于启动分离（开始）。应用程序必须将 OTG_HCCHAR1 中的 ODDFRM 位置 1。编程 MPS 字段。
2. OTG_HS 主机开始读取数据包。
3. 成功发送启动分离后（开始），OTG_HS 主机将生成 CHH 中断。
4. 为了响应 CHH 中断，将重新初始化寄存器以发送启动分离（结束）。
5. 成功发送启动分离后（结束），OTG_HS 主机将生成 CHH 中断。
6. 为了响应 CHH 中断，将会释放通道。

- **DMA 模式下的同步 IN 分离事务**

  通道 x 的操作顺序如下：

1. 按照*通道初始化*一节中的说明初始化并使能通道 x 以用于启动分离。
2. 当通道 x 接收到来自仲裁器的授权后，OTG_HS 主机会立即向请求队列写入一个 IN 请求。
3. OTG_HS 主机尝试在下一个奇数微帧的起始位置发送启动分离 IN 令牌。
4. 成功发送启动分离 IN 令牌后，OTG_HS 主机将生成 CHH 中断。
5. 作为对 CHH 中断的响应，将 OTG_HCSPLT2 中的 COMPLSPLT 位置 1，以发送完成分离。
6. 成功接收到数据包后，OTG_HS 主机即会开始向系统存储器写入数据。

   将接收到的数据传输到系统存储器后，OTG_HS 主机会生成 CHH 中断。为了响应 CHH 中断，将释放或重新初始化通道以供下一启动分离使用。

注： *本节内容仅适用于 USB OTG HS。*

57.15.6 设备编程模型

USB 复位时的端点初始化

1. 为所有 OUT 端点将 NAK 位置 1
   - OTG_DOEPCTLx 中，SNAK = 1（对于所有 OUT 端点）
2. 使能以下中断位
   - OTG_DAINTMSK 中，INEP0 = 1（控制 0 IN 端点）
   - OTG_DAINTMSK 中，OUTEP0 = 1（控制 0 OUT 端点）
   - OTG_DOEPMSK 中的 STUPM = 1
   - OTG_DOEPMSK 中的 XFRCM = 1
   - OTG_DIEPMSK 中的 XFRCM = 1
   - OTG_DIEPMSK 中的 TOM = 1
3. 为每个 FIFO 设置数据 FIFO RAM
   - 对 OTG_GRXFSIZ 寄存器进行编程，以能够接收控制传输的 OUT 数据和设置数据。如果未使能阈值，则该寄存器必须至少等于控制端点 0 的 1 个最大数据包大小 + 2 个字（用于控制 OUT 数据包的状态）+ 10 个字（用于 SETUP 数据包）。
   - 对 OTG_DIEPTXF0 寄存器进行编程（取决于所选的 FIFO 编号），以能够发送控制 IN 数据。该寄存器至少必须等于控制端点 0 的 1 个最大数据包大小。
4. 对端点相关寄存器中的以下字段进行编程，以使控制 OUT 端点 0 接收 SETUP 数据包
   - OTG_DOEPTSIZ0 中的 STUPCNT = 3（接收最多 3 个连续的 SETUP 数据包）
5. 在 DMA 模式下的 USB OTG_HS，OTG_DOEPDMA0 寄存器应具有一个有效的存储器地址，以存储接收到的任何 SETUP 数据包。

此时，接收 SETUP 数据包所需的所有初始化工作便已完成。

枚举完成时的端点初始化

1. 在枚举完成中断（OTG_GINTSTS 中的 ENUMDNE）中，读取 OTG_DSTS 寄存器以确定设备的枚举速度。
2. 对 OTG_DIEPCTL0 中的 MPSIZ 字段进行编程以设置最大数据包大小。该步骤配置控制端点 0。控制端点的最大数据包大小取决于枚举速度。
3. 对于 DMA 模式下的 USB OTG_HS，编程 OTG_DOEPCTL0 寄存器来使能控制 OUT 端点 0，以接收 SETUP 数据包。

此时，设备已准备好接收 SOF 数据包并配置为在控制端点 0 上执行控制传输。

收到 SetAddress 命令时的端点初始化

本节介绍了应用程序在 SETUP 数据包中接收到 SetAddress 命令时必须执行的操作。

1. 使用在 SetAddress 命令中接收到的设备地址来对 OTG_DCFG 寄存器进行编程
2. 对模块进行编程以发出状态阶段的 IN 数据包

收到 SetConfiguration/SetInterface 命令时的端点初始化

本节介绍了应用程序在 SETUP 包中接收 SetConfiguration 或 SetInterface 命令时必须执行的操作。

1. 接收到 SetConfiguration 命令时，应用程序必须对端点寄存器进行编程，以使用新配置中有效端点的特性来配置这些端点寄存器。
2. 接收到 SetInterface 命令时，应用程序必须对命令指定的端点的端点寄存器进行编程。
3. 在先前配置或其它设置中有效的端点在新的配置或其它设置中无效。必须停用这些无效端点。
4. 使用 OTG_DAINTMSK 寄存器使能有效端点的中断，屏蔽无效端点的中断。
5. 为每个 FIFO 设置数据 FIFO RAM。
6. 配置完所有必需的端点后，应用程序必须对模块进行编程以发送状态阶段的 IN 数据包。

此时，设备模块已可以接收和发送任何类型的数据包。

端点激活

本节介绍激活设备端点或者将现有设备端点配置为新类型所需的步骤。

1. 在 OTG_DIEPCTLx 寄存器（对于 IN 或双向端点）或 OTG_DOEPCTLx 寄存器（对于 OUT 或双向端点）的以下字段中，对所需端点的特性进行编程。
   - 最大数据包大小
   - USB 活动端点位置 1
   - 端点初始数据同步位（对于中断和批量端点）
   - 端点类型
   - Tx FIFO 编号
2. 激活端点后，模块便开始解码发送到该端点的令牌，并在收到的令牌有效的情况下回复有效握手信号。

端点停用

本节介绍停用现有端点所需的步骤。

1. 在要停用的端点中，将 OTG_DIEPCTLx 寄存器（对于 IN 或双向端点）或 OTG_DOEPCTLx 寄存器（对于 OUT 或双向端点）中的 USB 活动端点位清零。
2. 停用端点后，模块便会忽略发送到该端点的令牌，从而导致 USB 超时。

注： *应用程序必须满足以下条件才能设置设备模块以处理通信：*
必须将 OTG_GINTMSK 寄存器中的 NPTXFEM 和 RXFLVLM 清零。

操作模型

SETUP 和 OUT 数据传输：

本节介绍了数据 OUT 传输和 SETUP 事务期间的内部数据流和应用程序操作步骤。

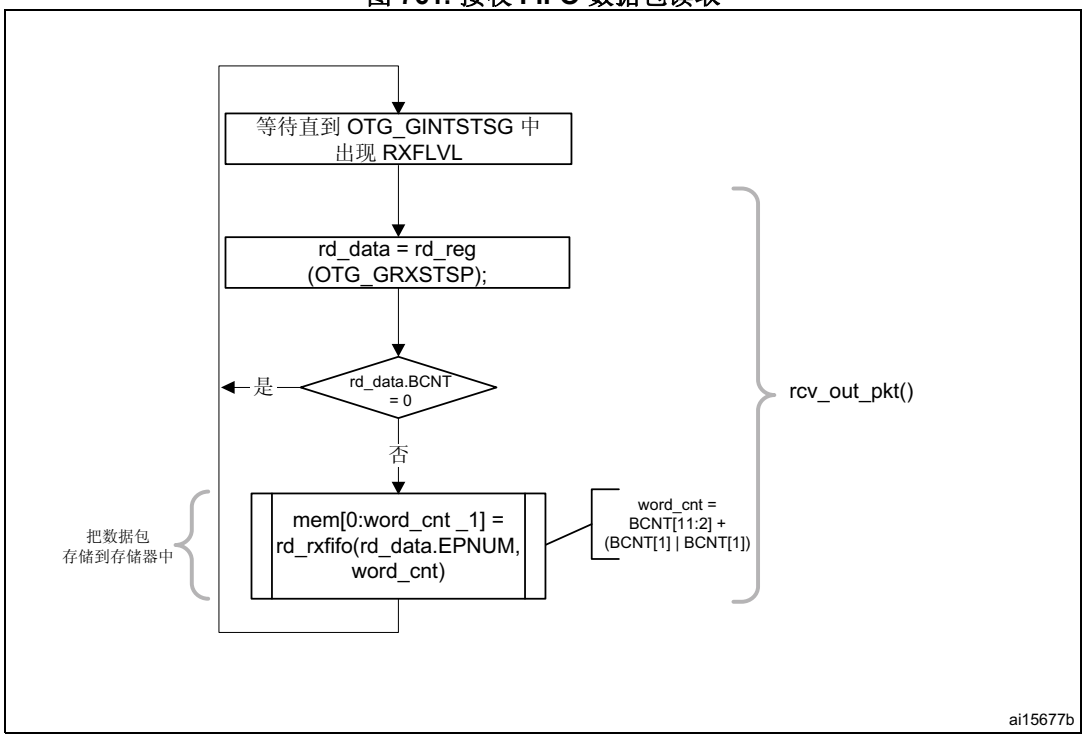
- **数据包读取**

本节介绍如何从接收 FIFO 读取数据包（OUT 数据和 SETUP 数据包）。

1. 捕获到 RXFLVL 中断（OTG_GINTSTS 寄存器）时，应用程序必须读取接收状态弹出寄存器 (OTG_GRXSTSP)。
2. 应用程序可以通过写入 RXFLVLM = 0（在 OTG_GINTMSK 中）来屏蔽 RXFLVL 中断（在 OTG_GINTSTS 中），直到它把数据包从接收 FIFO 中读取出来。
3. 如果已接收数据包的字节计数不是 0，则从接收数据 FIFO 中弹出这些数据并存储在存储器中。如果接收到的数据包字节计数为 0，则不会从接收数据 FIFO 中弹出任何数据。
4. 从接收 FIFO 读出的数据包状态有以下几种状态：
   a) 全局 OUT NAK：
   PKTSTS = 全局 OUT NAK，BCNT = 0x000，EPNUM = (0x0)，DPID = (0b00)。
   这些数据表示全局 OUT NAK 位已生效。
   b) SETUP 数据包：
   PKTSTS = SETUP，BCNT = 0x008，EPNUM = 控制端点编号，DPID = DATA0。
   这些数据表示指定端点上收到的 SETUP 数据包现在可从接收 FIFO 中读取。
   c) 建立阶段完成：
   PKTSTS = 建立阶段完成，BCNT = 0x0，EPNUM = 控制端点编号，
   DPID = (0b00)。
   这些数据表示指定端点的建立阶段完成并且数据阶段已启动。在此状态条目从接收 FIFO 中弹出后，模块将在该控制 OUT 端点上产生建立中断。
   d) OUT 数据包：
   PKTSTS = DataOUT，BCNT = 接收的 OUT 数据包的大小 (0 ≤ BCNT ≤ 1 024)，
   EPNUM = 收到数据包的端点编号，DPID = 实际数据 PID。
   e) 数据传输完成：
   PKTSTS = OUT 数据传输完成，BCNT = 0x0，EPNUM = 完成数据传输的 OUT 端点编号，DPID = (0b00)。
   这些数据表示指定 OUT 端点的 OUT 数据传输完成。在此状态条目从接收 FIFO 中弹出后，模块将在指定的 OUT 端点上引发“传输完成”中断。
5. 从接收 FIFO 中弹出数据后，必须使能 RXFLVL 中断 (OTG_GINTSTS)。
6. 每次应用程序检测到 OTG_GINTSTS 中的 RXFLVL 中断时，都将重复步骤 1 到 5。读取空的接收 FIFO 可能导致未定义的模块行为。

图 761 提供了上述过程的流程图。

图 761. 接收 FIFO 数据包读取



SETUP 事务

本节介绍了模块处理 SETUP 数据包的方式以及应用程序处理 SETUP 事务的顺序。

- **应用程序要求**

1. 要接收 SETUP 数据包，必须将控制 OUT 端点中的 STUPCNT 字段 (OTG_DOEPTSIZx) 编程为非零值。如果应用程序将 STUPCNT 字段编程为非零值，模块会接收 SETUP 数据包并将其写入接收 FIFO，而不考虑 NAK 状态和 OTG_DOEPCTLx 中的 EPENA 位设置。控制端点每收到一个 SETUP 数据包后，STUPCNT 字段都会递减。如果在接收 SETUP 数据包之前，未将 STUPCNT 字段编程为适当值，模块仍能接收 SETUP 数据包并使 STUPCNT 字段递减，但应用程序可能无法确定在控制传输的建立阶段中接收的 SETUP 数据包正确数量。
   – 在 OTG_DOEPTSIZx 中，STUPCNT = 3
2. 应用程序必须始终在接收数据 FIFO 中分配一些额外空间，以便能够在控制端点上接收连续的最多三个 SETUP 数据包。
   – 预留空间 10 个字。第一个 SETUP 数据包需要 3 个字，“建立阶段完成”状态双字需要 1 个字，还需要 6 个字以存储两个额外的 SETUP 数据包。
   – 每个 SETUP 数据包需要 3 个字以存储 8 个字节的 SETUP 数据和 4 个字节的 SETUP 状态。模块将在接收 FIFO 中保留这些空间。
   – 这段 FIFO 仅用于存储 SETUP 包，绝对不会将该空间用于数据包。
3. 应用程序必须从接收 FIFO 中读取 SETUP 数据包的 2 个字。
4. 应用程序必须从接收 FIFO 中读取并丢弃“建立阶段完成”状态字。

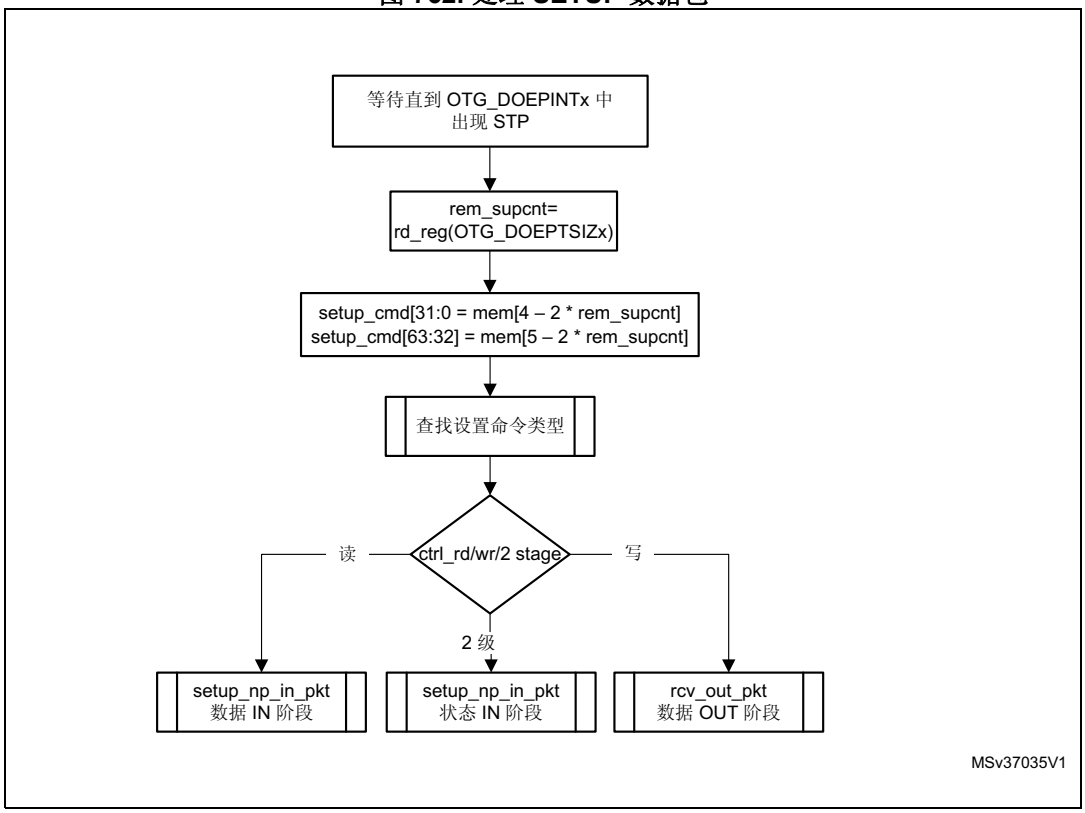
- **内部数据流**

1. 接收到 SETUP 数据包时，模块会将接收到的数据写入接收 FIFO，而不会检查接收 FIFO 中的可用空间，且不考虑端点的 NAK 和 STALL 位设置。
   – 模块会在内部将接收到 SETUP 数据包的控制 IN/OUT 端点的 IN NAK 和 OUT NAK 位置 1。
2. USB 上接收到的每个 SETUP 数据包，模块会将 3 个字的数据写入接收 FIFO，并且将 STUPCNT 字段递减 1。
   – 第一个字包含由模块所使用的内部控制信息
   – 第二个字包含 SETUP 命令的前 4 个字节
   – 第三个字包含 SETUP 命令的最后 4 个字节
3. 当建立阶段结束，数据 IN/OUT 阶段开始时，模块会将一个状态条目（“建立阶段完成”字）写入接收 FIFO，指示建立阶段完成。
4. 在 AHB 端，SETUP 数据包被应用程序读取。
5. 当应用程序从接收 FIFO 中弹出“建立阶段完成”字时，模块将使用 STUP 中断 (OTG_DOEPINTx) 来中断应用程序，指示其可以处理接收到的 SETUP 数据包。
6. 模块会将控制 OUT 端点的端点使能位清零。

- **应用程序编程顺序**

1. 对 OTG_DOEPTSIZx 寄存器进行编程。
   – STUPCNT = 3
2. 等待 RXFLVL 中断 (OTG_GINTSTS) 并且从接收 FIFO 中读取数据包。
3. STUP 中断的触发 (OTG_DOEPINTx) 表示 SETUP 数据传输成功完成。
   – 发生该中断时，应用程序必须读取 OTG_DOEPTSIZx 寄存器以确定接收的 SETUP 数据包数量并处理最后接收的 SETUP 数据包。

图 762. 处理 SETUP 数据包



- **处理三个以上连续的 SETUP 数据包**

根据 USB 2.0 规范，在 SETUP 数据包错误中，主机通常不会向同一个端点发送 3 个以上连续的 SETUP 数据包。但是，USB 2.0 规范并未限制主机可以向同一个端点发送的连续 SETUP 数据包数量。出现这种情况时，OTG_HS 控制器将生成中断（OTG_DOEPINTx 中的 B2BSTUP）。

- **将全局 OUT NAK 置 1**

内部数据流：

1. 如果应用程序将全局 OUT NAK（OTG_DCTL 中的 SGONAK 位）置 1，模块将停止向接收 FIFO 中写入 SETUP 数据包以外的数据。无论接收 FIFO 中可用空间大小如何，设备都会对主机发送的非同步 OUT 令牌回复 NAK 握手，而对同步 OUT 数据包直接予以忽略。
2. 模块将全局 OUT NAK 写入接收 FIFO。应用程序必须为此留出足够空间。
3. 当应用程序从接收 FIFO 中弹出全局 OUT NAK 字时，模块会将 GONAKEFF 中断 (OTG_GINTSTS) 置 1。
4. 应用程序检测到该中断后，会认为模块处于全局 OUT NAK 模式。应用程序可以通过将 OTG_DCTL 中的 SGONAK 位清零来清除该中断。

应用程序编程顺序：

1. 要停止接收任何类型的数据到接收 FIFO 中，应用程序必须通过编程以下字段以将全局 OUT NAK 位置 1。
   – 在 OTG_DCTL 中，SGONAK = 1
2. 等待 OTG_GINTSTS 中的 GONAKEFF 中断。一旦被触发，该中断表示模块已停止接收 SETUP 数据包以外的任何类型数据。
3. 如果应用程序已将 OTG_DCTL 中的 SGONAK 位置 1，则在模块引发 GONAKEFF 中断 (OTG_GINTSTS) 之前，应用程序可以接收有效 OUT 数据包。
4. 应用程序可通过对 OTG_GINTMSK 寄存器中的 GONAKEFFM 位执行写操作来暂时屏蔽此中断。
   – 在 OTG_GINTMSK 寄存器中，GONAKEFFM = 0
5. 当应用程序准备退出全局 OUT NAK 模式时，必须将 OTG_DCTL 中的 SGONAK 位清零。此操作还会清除 GONAKEFF 中断 (OTG_GINTSTS)。
   – 在 OTG_DCTL 中，CGONAK = 1
6. 如果应用程序在之前已屏蔽此中断，则必须按以下方式使能该中断：
   – 在 OTG_GINTMSK 中，GONAKEFFM = 1

- **禁止 OUT 端点**

应用程序必须使用以下顺序禁止已使能的 OUT 端点。

应用程序编程顺序：

1. 禁止任何 OUT 端点前，应用程序必须在模块中使能全局 OUT NAK 模式。
   – 在 OTG_DCTL 中，SGONAK = 1
2. 等待 GONAKEFF 中断 (OTG_GINTSTS)
3. 通过编程以下字段来禁止 OUT 端点：
   – 在 OTG_DOEPCTLx 中，EPDIS = 1
   – 在 OTG_DOEPCTLx 中，SNAK = 1
4. 等待 EPDISD 中断 (OTG_DOEPINTx)，该中断表示已完全禁止 OUT 端点。引发 EPDISD 中断时，模块还会将以下位清零：
   – 在 OTG_DOEPCTLx 中，EPDIS = 0
   – 在 OTG_DOEPCTLx 中，EPENA = 0
5. 应用程序必须将全局 OUT NAK 位清零，以开始从其它未禁止的 OUT 端点接收数据。
   – 在 OTG_DCTL 中，SGONAK = 0

- **通用非同步 OUT 数据传输**

本节介绍一种常规非同步 OUT 数据传输（控制、批量或中断）。

应用程序要求：

1. 建立 OUT 传输前，应用程序必须在存储器中分配一个缓冲区，以容纳要作为 OUT 传输的一部分而接收的所有数据。
2. 对于 OUT 传输，端点的传输大小寄存器中的传输大小字段必须是端点的最大数据包大小的倍数（且以字对齐）。
   – 传输大小[EPNUM] = *n* × (MPSIZ[EPNUM] + 4 – (MPSIZ[EPNUM] mod 4))
   – 数据包计数[EPNUM] = *n*
   – *n* > 0
3. 发生 OUT 端点中断时，应用程序必须读取端点的传输大小寄存器以计算存储器中有效数据量。接收的有效数据量可能小于编程的传输大小。
   – 存储器中的有效数据量 = 应用程序设置的初始传输量 – 模块更新后的剩余传输量
   – 接收到 USB 数据包数 = 应用程序设置的初始数据包数 – 模块更新后的剩余数据包数

内部数据流：

1. 应用程序必须在端点相关寄存器中设置传输大小和数据包计数字段，将 NAK 位清零，并使能端点来接收数据。
2. NAK 位清零后，模块便开始接收数据并将数据写入接收 FIFO（只要接收 FIFO 中有空间）。对于 USB 上接收的每个数据包，数据包及其状态都会写入接收 FIFO。写入接收 FIFO 的每个数据包（数据量达到最大数据包大小的数据包或短数据包）都会使该端点的数据包计数字段递减 1。
   – 收到的数据包若 CRC 无效，则自动被从接收 FIFO 中清除。
   – 在 USB 上为数据包回复 ACK 后，模块将丢弃主机因无法检测到 ACK 而重新发送的非同步 OUT 数据包。应用程序不会在具有相同数据 PID 的相同端点上检测到多个连续的 OUT 数据包。在这种情况下，数据包计数不会递减。
   – 如果接收 FIFO 中没有空间，则会忽略同步或非同步数据包并且不会将它们写入接收 FIFO。此外，非同步 OUT 令牌将会收到 NAK 握手应答。
   – 在上述所有三种情况中，数据包计数都不会递减，因为没有任何数据写入接收 FIFO。
3. 当数据包计数变为 0 或者在端点上接收到短数据包时，该端点的 NAK 位将置 1。NAK 位置 1 后，将忽略同步或非同步数据包并且不会将它们写入接收 FIFO，同时非同步 OUT 令牌会收到 NAK 握手应答。
4. 在数据写入接收 FIFO 后，应用程序将从接收 FIFO 中读取数据并将数据写入外部存储器，一次一个数据包，逐个端点过来。
5. 在 AHB 上向外部存储器写入完每个数据包后，端点的传输大小都会自动减去该数据包的大小。
6. 在以下情况时，OUT 端点的 OUT 数据传输完成状态将写入接收 FIFO：
   – 传输大小为 0 并且数据包计数为 0
   – 写入接收 FIFO 的最后一个 OUT 数据包是短数据包（0 ≤ 数据包大小 < 最大数据包大小）
7. 当应用程序弹出此状态条目（OUT 数据传输完成），并生成该端点的传输完成中断，同时清零端点使能位。

应用程序编程顺序：

1. 使用传输大小和相应数据包个数对 OTG_DOEPTSIZx 寄存器进行编程。
2. 使用端点特性对 OTG_DOEPCTLx 寄存器进行编程，并将 EPENA 和 CNAK 位置 1。
   – 在 OTG_DOEPCTLx 中，EPENA = 1
   – 在 OTG_DOEPCTLx 中，CNAK = 1
3. 等待 RXFLVL 中断（在 OTG_GINTSTS 中）并且从接收 FIFO 中读走数据包。
   – 此步骤可重复多次，具体取决于传输大小。
4. 触发 XFRC 中断 (OTG_DOEPINTx)，以表示非同步 OUT 数据传输成功完成。
5. 读取 OTG_DOEPTSIZx 寄存器，以确定有效数据量。

- **通用同步 OUT 数据传输**

本节介绍常规的同步 OUT 数据传输。

应用程序要求：

1. 非同步 OUT 数据传输的所有应用程序要求均适用于同步 OUT 数据传输。
2. 对于同步 OUT 数据传输中的传输大小和数据包计数字段，必须始终将其设置为单个帧中可接收的最大数据包大小的数据包数目。同步类型的 OUT 数据传输事务必须在一个帧内完成。
3. 在周期性帧结束（OTG_GINTSTS 中的 EOPF 中断）之前，应用程序必须从接收 FIFO 中读取所有同步 OUT 数据包（数据条目和状态条目）。
4. 要接收下一帧中的数据，必须在 EOPF (OTG_GINTSTS) 之后 SOF (OTG_GINTSTS) 之前使能一个同步 OUT 端点。

内部数据流：

1. 同步 OUT 端点的内部数据流与非同步 OTU 端点的基本相同，但稍有差异。
2. 同步 OUT 端点通过将端点使能位置 1 并将 NAK 位清零来使能时，必须相应地将偶数/奇数帧位置 1。仅当符合以下条件时，模块才会在同步 OUT 端点上接收特定帧中的数据：
   – EONUM （在 OTG_DOEPCTLx 中）= FNSOF[0]（在 OTG_DSTS 中）
3. 当应用程序从接收 FIFO 中完整地读取一个同步 OUT 数据包（数据和状态）后，模块会根据从接收 FIFO 中读取的最后一个同步 OUT 数据包的数据 PID 更新 OTG_DOEPTSIZx 中的 RXDPID 字段。

应用程序编程顺序：

1. 使用传输大小和相应数据包个数对 OTG_DOEPTSIZx 寄存器进行编程
2. 使用端点特性对 OTG_DOEPCTLx 寄存器进行编程，并将端点使能位、清除 NAK 位和奇数/偶数帧位置 1。
   – EPENA = 1
   – CNAK = 1
   – EONUM =（0：偶数/1：奇数）
3. 等待 RXFLVL 中断（在 OTG_GINTSTS 中）并且从接收 FIFO 中读走数据包
   – 此步骤可重复多次，具体取决于传输大小。
4. XFRC 中断（在 OTG_DOEPINTx 中）表示同步 OUT 数据传输完成。该中断不一定意味着存储器中的数据是有效的。
5. 对于同步 OUT 传输，应用程序可能并不总会检测到该中断。而是可以检测到 OTG_GINTSTS 中的 INCOMPISOOUT 中断。
6. 读取 OTG_DOEPTSIZx 寄存器以确定接收的传输大小以及帧中所接收数据的有效性。仅当符合以下条件之一时，应用程序才必须将存储器中接收的数据视为有效数据：
   – RXDPID = DATA0（在 OTG_DOEPTSIZx 中）并且接收该有效数据的 USB 数据包数量 = 1
   – RXDPID = DATA1（在 OTG_DOEPTSIZx 中）并且接收该有效数据的 USB 数据包数量 = 2
   – RXDPID = D2（在 OTG_DOEPTSIZx 中）并且接收该有效数据的 USB 数据包数量 = 3

   接收该有效数据的 USB 数据包数量 =
   应用程序编程的初始数据包个数 - 模块更新后的剩余数据包个数

   应用程序可将无效数据包丢弃。

- **不完整的同步 OUT 数据传输**

本节介绍了同步 OUT 数据包出现丢包时应用程序编程顺序。

内部数据流：

1. 对于同步 OUT 端点，可能并不总是触发 XFRC 中断（在 OTG_DOEPINTx 中）。如果模块丢弃同步 OUT 数据包，则在以下情况下，应用程序可能无法检测到 XFRC 中断 (OTG_DOEPINTx)：
   – 在接收 FIFO 无法容纳完整的 ISO OUT 数据包时，模块将丢弃接收到的 ISO OUT 数据
   – 接收到的同步 OUT 数据包存在 CRC 错误
   – 模块接收到的同步 OUT 令牌损坏
   – 应用程序从接收 FIFO 中读取数据的速度非常缓慢
2. 如果模块在所有同步 OUT 端点的传输完成前检测到周期性帧结束，将触发未完成同步 OUT 数据中断（OTG_GINTSTS 中的 INCOMPISOOUT），指示至少有一个同步 OUT 端点上未触发 XFRC 中断（在 OTG_DOEPINTx 中）。此时，未完成传输的端点仍保持使能，但在 USB 的该端点上，没有进行中的有效传输。

应用程序编程顺序：

1. 硬件触发 INCOMPISOOUT 中断 (OTG_GINTSTS) 表示当前帧中至少有一个同步 OUT 端点具有未完成的传输。
2. 如果因未从端点完全读取同步 OUT 数据而发生这种情况，应用程序必须确保首先从接收 FIFO 读取走所有同步 OUT 数据（包括数据条目和状态条目），然后再继续处理。
   – 从接收 FIFO 读取所有数据后，应用程序即可检测到 XFRC 中断 (OTG_DOEPINTx)。在此情况下，应用程序必须重新使能端点以接收下一个帧中的同步 OUT 数据。
3. 当应用程序接收到 INCOMPISOOUT 中断（在 OTG_GINTSTS 中）时，应用程序必须读取所有同步 OUT 端点的控制寄存器 (OTG_DOEPCTLx)，以确定哪些端点在当前微帧中具有不完整的传输。同时满足以下两个条件时，表示端点传输未完成：
   – EONUM 位（在 OTG_DOEPCTLx 中）= FNSOF[0]（在 OTG_DSTS 中）
   – EPENA = 1（在 OTG_DOEPCTLx 中）
4. 在检测到 SOF 中断（在 OTG_GINTSTS 中）前，必须执行完成上一步操作，以确保当前帧编号未发生更改。
5. 对于具有未完成传输的同步 OUT 端点，应用程序必须丢弃存储器中的数据，并通过将 OTG_DOEPCTLx 中的 EPDIS 位置 1 来禁止该端点。
6. 等待 EPDISD 中断（在 OTG_DOEPINTx 中），并使能该端点，以便接收下一个帧中的新数据。
   – 由于模块可能需要一些时间才能禁止端点，因此应用程序在接收到无效同步数据后，可能无法接收下一个帧中的数据。

- **停止非同步 OUT 端点**

本节介绍应用程序如何才能停止非同步端点。

1. 将模块置于全局 OUT NAK 模式。
2. 禁止所需的端点
   – 禁止端点时，请设置 STALL = 1（在 OTG_DOEPCTL 中），而不是将 OTG_DOEPCTL 中的 SNAK 位置 1。

     STALL 位的优先级始终高于 NAK 位。
3. 当应用程序不再需要端点回复 STALL 握手信号时，必须将 STALL 位（在 OTG_DOEPCTLx 中）清零。
4. 如果应用程序由于收到主机的 SetFeature.Endpoint Halt 或 ClearFeature.Endpoint Halt 命令来设置或清除端点的 STALL 状态，则必须在该控制端点上的状态阶段传输前，将 STALL 位置 1 或清零。

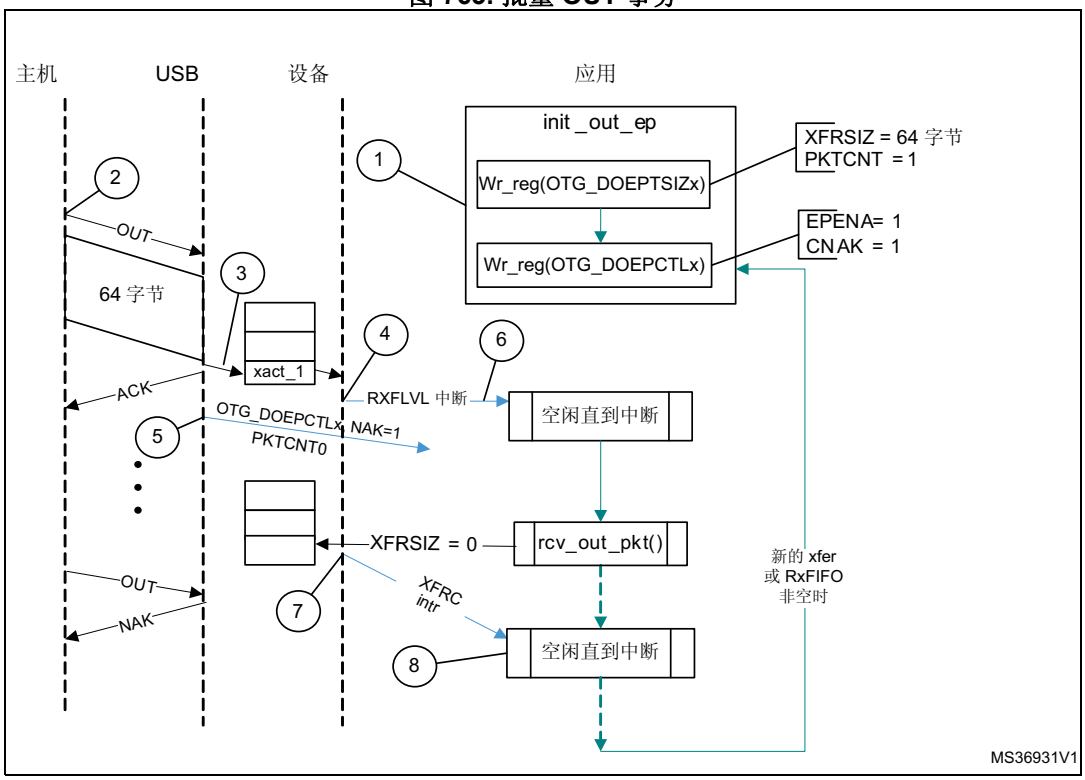
示例

本节介绍并描述了一些基本的传输类型和情景。

- 批量 OUT 事务

图 763 描述了将单个批量 OUT 数据包从 USB 接收到 AHB 中的过程，并介绍了这一过程中涉及到的事件。

图 763. 批量 OUT 事务



接收到 SetConfiguration/SetInterface 命令后，应用程序将初始化所有 OUT 端点：将 CNAK 和 EPENA 置 1（在 OTG_DOEPCTLx 中），并将 OTG_DOEPTSIZx 寄存器中的 XFRSIZ 和 PKTCNT 设置成合适的值。

1. 主机尝试将数据（OUT 令牌）发送到端点。
2. 当模块在 USB 上接收到 OUT 令牌时，模块会将数据包存储在 Rx FIFO 中，因为其中有可用空间。
3. 在 Rx FIFO 中写入完整数据包后，模块随后会触发 RXFLVL 中断（在 OTG_GINTSTS 中）。
4. 接收到 PKTCNT 所指定数量的 USB 数据包后，模块在内部将该端点的 NAK 位置 1，以避免其再接收任何数据包。
5. 应用程序处理中断并从 Rx FIFO 中读取数据。
6. 应用程序读取完所有数据后（相当于 XFRSIZ），模块将生成 XFRC 中断（在 OTG_DOEPINTx 中）。
7. 应用程序处理中断并通过 XFRC 中断位（在 OTG_DOEPINTx 中）的触发得知本次传输完成。

IN 数据传输

- **数据包写入**

本节介绍在已使能专用发送 FIFO 的情况下应用程序如何将数据包写入端点 FIFO。

1. 应用程序可以选择轮询模式或中断模式。
   - 在轮询模式下，应用程序通过读取 OTG_DTXFSTSx 寄存器来监视端点发送数据 FIFO 的状态，从而确定该数据 FIFO 中是否有足够空间。
   - 在中断模式中，应用程序等待 TXFE 中断（在 OTG_DIEPINTx 中），然后读取 OTG_DTXFSTSx 寄存器，以确定数据 FIFO 中是否有足够空间。
   - 要写入单个非零长度的数据包，数据 FIFO 中必须有足够的空间来容纳整个数据包。
   - 要写入零长度的数据包，应用程序不能查看 FIFO 空间。
2. 如果使用上述方法之一，当应用程序确定有足够空间来写入发送数据包时，应用程序必须首先对端点控制寄存器进行相应写操作，然后再将数据写入数据 FIFO。通常，应用程序必须对 OTG_DIEPCTLx 寄存器执行读-修改-写操作，以避免在将端点使能位置 1 的同时，修改寄存器中的其它内容。

如果有足够空间，应用程序可将同一端点的多个数据包写入发送 FIFO。对于周期性 IN 端点，应用程序只能一次写入一个微帧内的多个数据包。只有先前一个微帧的通信事务传输完成之后，应用程序才会写入下一个微帧内要发送的所有数据包。

- **将 IN 端点 NAK 置 1**

内部数据流：

1. 当应用程序将特定端点的 IN NAK 置 1 时，模块将停止端点上的数据发送，而不考虑端点发送 FIFO 中的数据是否可用。
2. 非同步端点收到 IN 令牌，回复 NAK 握手应答
   - 同步端点收到 IN 令牌，返回零长度数据包
3. 模块在 OTG_DIEPINTx 中触发 INEPNE（IN 端点 NAK 有效）中断以响应 OTG_DIEPCTLx 中的 SNAK 位。
4. 应用程序检测到该中断后，便会认为端点处于 IN NAK 模式。应用程序可以通过将 OTG_DIEPCTLx 中的 CNAK 位置 1 来清除该中断。

应用程序编程顺序：

1. 要在特定 IN 端点上停止发送任何数据，应用程序必须将 IN NAK 位置 1。要将该位置 1，必须编程以下字段。
   - 在 OTG_DIEPCTLx 中，SNAK = 1
2. 等待 OTG_DIEPINTx 中的 INEPNE 中断触发。该中断表示模块已在端点上停止发送数据。
3. 在应用程序将 NAK 位置 1 但“NAK 有效”中断尚未触发时，模块可以在端点上发送有效 IN 数据。
4. 应用程序可通过写入 OTG_DIEPMSK 中的 INEPNEM 位来临时屏蔽该中断。
   - 在 OTG_DIEPMSK 中，INEPNEM = 0
5. 要退出端点 NAK 模式，应用程序必须将 OTG_DIEPCTLx 中的 NAK 状态位 (NAKSTS) 清零。此操作还会将 INEPNE 中断位（在 OTG_DIEPINTx 中）清零。
   - 在 OTG_DIEPCTLx 中，CNAK = 1
6. 如果应用程序已将该中断屏蔽，则必须按以下方式使能：
   - 在 OTG_DIEPMSK 中，INEPNEM = 1

- **禁止 IN 端点**

使用以下顺序来禁止先前已使能的特定 IN 端点。

应用程序编程顺序：

1. 应用程序必须先停止在 AHB 上写入数据，之后才能禁止 IN 端点。
2. 应用程序必须将端点设置为 NAK 模式。
   – 在 OTG_DIEPCTLx 中，SNAK = 1
3. 等待 OTG_DIEPINTx 中的 INEPNE 中断。
4. 将必须禁止的端点的 OTG_DIEPCTLx 寄存器中的以下位置 1。
   – 在 OTG_DIEPCTLx 中，EPDIS = 1
   – 在 OTG_DIEPCTLx 中，SNAK = 1
5. OTG_DIEPINTx 中的 EPDISD 中断的触发表示模块已完全禁止指定的端点。在触发中断的同时，模块还会将以下位清零：
   – 在 OTG_DIEPCTLx 中，EPENA = 0
   – 在 OTG_DIEPCTLx 中，EPDIS = 0
6. 应用程序必须读取周期性 IN EP 的 OTG_DIEPTSIZx 寄存器，以计算该端点在 USB 上发送的数据量。
7. 应用程序必须通过将 OTG_GRSTCTL 寄存器中的以下字段置 1，来清空端点发送 FIFO 中的数据：
   – TXFNUM（在 OTG_GRSTCTL 中）= 端点发送 FIFO 编号
   – TXFFLSH（在 OTG_GRSTCTL 中）= 1

应用程序必须轮询 OTG_GRSTCTL 寄存器，直至模块将 TXFFLSH 位清零，这表示 FIFO 清空操作结束。要在该端点上发送新数据，应用程序可以稍后重新使能该端点。

- **通用非周期性 IN 数据传输**

应用程序要求：

1. 建立 IN 传输前，应用程序必须确保组成一次 IN 传输的每个数据包都可以容纳在单个缓冲区中。
2. 对于 IN 传输，端点传输大小寄存器中的传输大小字段表示本次传输的有效数据量，它由多个最大数据包大小和单个短数据包组成。该短数据包在传输结束时发送。
   – 要发送多个最大数据包大小的数据包并在传输结束时外加一个短数据包：
     传输大小[EPNUM] = *x* × MPSIZ[EPNUM] + sp
     如果 (sp > 0)，数据包计数[EPNUM] = x + 1。
     否则，数据包计数[EPNUM] = *x*
   – 要发送单个零长度数据包：
     传输大小[EPNUM] = 0
     数据包计数[EPNUM] = 1
   – 要发送多个最大数据包大小的数据包并在传输结束时外加一个零长度数据包，应用程序必须将传输拆分为两个部分。第一部分发送最大数据包大小的数据包，第二部分仅发送零长度数据包。
     第一次传输：传输大小[EPNUM] = *x* × MPSIZ[epnum]；数据包计数 = *n*；
     第二次传输：传输大小[EPNUM] = 0；数据包计数 = 1；

3. 使能某个端点进行数据传输后，模块会更新传输大小寄存器。在 IN 传输结束时，应用程序必须读取传输大小寄存器，以确定送入发送 FIFO 中的数据已有多少通过 USB 发送出去。
4. 送入发送 FIFO 中的数据量 = 应用程序编程的初始传输大小 - 模块更新后的最终传输大小。
   – 通过 USB 已经发送的数据量 = (应用程序编程的初始数据包计数 - 模块更新后的最终数据包计数) × MPSIZ[EPNUM]
   – 要通过 USB 发送的剩余数据量 = (应用程序编程的初始传输大小 - 已通过 USB 发送的数据量)

内部数据流：

1. 应用程序必须在特定端点的寄存器中设置传输大小和数据包计数字段，并使能该端点来发送数据。
2. 应用程序还必须向该端点的发送 FIFO 写入必需的数据。
3. 应用程序每向发送 FIFO 写入一个数据包，该端点的传输大小便会自动减去该数据包大小。应用程序持续从存储器获取数据来写入发送 FIFO，直到该端点的传输大小变为 0。向 FIFO 写入数据后，“FIFO 中的数据包数”计数会递增（这是一个 3 位计数，由模块在内部进行维护，每个 IN 端点发送 FIFO 对应一个。在 IN 端点 FIFO 中，模块所维护的最大数据包数始终为八个）。对于零长度数据包，每个 FIFO 均另有一个单独的标志，FIFO 中没有任何数据。
4. 当数据写入发送 FIFO 后，模块会在接收到 IN 令牌时将这些数据送出。每个非同步 IN 数据包发送出去并收到回复的 ACK 握手信号后，该端点的数据包计数都会递减 1，直到数据包计数变 0 为止。发生超时时，数据包计数不会递减。
5. 对于零长度数据包（由内部零长度标志指示），模块会针对 IN 令牌发出一个零长度数据包，并递减数据包计数字段的值。
6. 如果接收到 IN 令牌的端点对应的 FIFO 中无数据，且该端点的数据包计数字段为零，则模块会针对该端点生成一个“Tx FIFO 为空时接收到 IN 令牌”(ITTXFE) 中断（前提是该端点的 NAK 位未置 1）。模块在该非同步端点上回复 NAK 握手信号。
7. 模块会在内部使 FIFO 指针重新返回到开头，并且不会生成超时中断。
8. 当传输大小为 0 且数据包计数为 0 时，将生成该端点的传输完成 (XFRC) 中断，同时将端点使能清零。

应用程序编程顺序：

1. 使用传输大小和相应数据包计数对 OTG_DIEPTSIZx 寄存器进行编程。
2. 使用端点特性对 OTG_DIEPCTLx 寄存器进行编程，并将 CNAK 和 EPENA（端点使能）位置 1。
3. 发送非零长度数据包时，应用程序必须轮询 OTG_DTXFSTSx 寄存器（其中 x 为与该端点相关联的 FIFO 编号），以确定数据 FIFO 中是否有足够的空间。写入数据前，应用程序可以选择使用 TXFE（在 OTG_DIEPINTx 中）。

- **通用周期性 IN 数据传输**

本节介绍典型的周期性 IN 数据传输。

应用程序要求：

1. *第 2515 页的通用非周期性 IN 数据传输*的应用程序要求 1、2、3 和 4 对周期性 IN 数据传输同样适用（只是对要求 2 稍加修改）。
   - 应用程序只能发送若干个最大数据包大小的数据包或若干个最大数据包大小的包，外加传输结束时的一个短数据包。要发送多个最大数据包大小的数据包并在传输结束时外加一个短数据包，必须满足以下条件：

     传输大小[EPNUM] = x × MPSIZ[EPNUM] + sp
     （其中 *x* 是 ≥ 0 的整数，且 0 ≤ sp < MPSIZ[EPNUM]）

     如果 (sp > 0)，数据包计数[EPNUM] = *x* + 1
     否则，数据包计数[EPNUM] = *x*；

     MCNT[EPNUM] = 数据包计数[EPNUM]
   - 应用程序无法在传输结束时发送零长度数据包。应用程序可以单独发送一个零长度数据包。要发送单个零长度数据包：
   - 传输大小[EPNUM] = 0

     数据包计数[EPNUM] = 1

     MCNT[EPNUM] = 数据包计数[EPNUM]
2. 应用程序一次只能安排一帧的数据传输。
   - (MCNT – 1) × MPSIZ ≤ XFERSIZ ≤ MCNT × MPSIZ
   - PKTCNT = MCNT（在 OTG_DIEPTSIZx 中）
   - 如果 XFERSIZ < MCNT × MPSIZ，则传输的最后一个数据包为短数据包。
   - 注意：MCNT 在 OTG_DIEPTSIZx 中，MPSIZ 在 OTG_DIEPCTLx 中，PKTCNT 在 OTG_DIEPTSIZx 中，而 XFERSIZ 在 OTG_DIEPTSIZx 中
3. 接收到 IN 令牌前，应用程序必须将要在帧中发送的完整数据写入到发送 FIFO 中。在接收到 IN 令牌时，即使发送 FIFO 中该帧要发送的数据只差 1 个双字未写进来，模块也会执行 FIFO 为空时的操作。当发送 FIFO 为空时：
   - 同步端点上将回复零长度数据包
   - 中断端点上将回复 NAK 握手信号

内部数据流：

1. 应用程序必须在特定端点的寄存器中设置传输大小和数据包计数字段，并使能该端点来发送数据。
2. 应用程序还必须向与该端点相关联的发送 FIFO 写入必需的数据。
3. 应用程序每向发送 FIFO 写入一个数据包，该端点的传输大小便会自动减去该数据包大小。应用程序持续从存储器获取数据来写入发送 FIFO，直到该端点的传输大小变为 0。

4. 当周期性端点接收到 IN 令牌时，模块将开始发送 FIFO 中的数据（如果 FIFO 中有数据）。如果 FIFO 中没有该帧要发送的数据的完整数据包，则模块将为该端点生成一个“TxFIFO 为空时接收到 IN 令牌”中断。
   – 同步端点上将回复零长度数据包
   – 中断端点上将回复 NAK 握手信号
5. 端点的数据包计数会在下列情况下递减 1：
   – 对于同步端点，发送一个零长度或非零长度的数据包时
   – 对于中断端点，在发送 ACK 握手信号时递减
   – 当传输大小和数据包计数均为 0 时，将生成该端点的传输完成中断，同时将端点使能位清零。
6. 在“周期性帧间隔”（由 OTG_DCFG 中的 PFIVL 位控制）内，当模块发现任何在当前帧内应为空的同步 IN 端点 FIFO 中的数据还未发送完成时，都会在 OTG_GINTSTS 中生成一个 IISOIXFR 中断。

应用程序编程顺序：

1. 使用端点特性对 OTG_DIEPCTLx 寄存器进行编程，并将 CNAK 和 EPENA 位置 1。
2. 将需要在下一帧中发送的数据写入发送 FIFO。
3. 硬件触发 ITTXFE 中断（在 OTG_DIEPINTx 中）表示应用程序尚未将需要发送的全部数据写入发送 FIFO。
4. 如果在检测到中断前已使能中断端点，则将忽略该中断。如果中断端点未使能，则使能此端点，以便数据能够在收到下一次 IN 令牌时发送出去。
5. 硬件触发 XFRC 中断（在 OTG_DIEPINTx 中）时如果 OTG_DIEPINTx 中未产生 ITTXFE 中断，则表示成功完成同步 IN 传输。读取 OTG_DIEPTSIZx 寄存器时始终得到传输大小 = 0 且数据包计数 = 0，表示所有数据都已通过 USB 发送完毕。
6. 无论是否产生 ITTXFE 中断（在 OTG_DIEPINTx 中），只要触发 XFRC 中断（在 OTG_DIEPINTx 中），即表示中断 IN 传输成功完成。读取 OTG_DIEPTSIZx 寄存器时始终得到传输大小 = 0 且数据包计数 = 0，表示所有数据都已通过 USB 发送完毕。
7. 如果在未产生任何前述中断的情况下在 OTG_GINTSTS 中触发未完成同步 IN 传输 (IISOIXFR) 中断，则表示模块在当前帧中至少未收到 1 个周期性 IN 令牌。

- **未完成同步 IN 数据传输**

本节介绍应用程序针对未完成同步 IN 数据传输必须执行的操作。

内部数据流：

1. 符合下列条件之一时，即认为同步 IN 传输未完成：
   a) 模块在至少一个同步 IN 端点上接收到损坏的同步 IN 令牌。此时，应用程序检测到未完成同步 IN 传输中断（OTG_GINTSTS 中的 IISOIXFR 位）。
   b) 应用程序向发送 FIFO 写入数据的速度过慢，在将完整数据写入 FIFO 之前便接收到 IN 令牌。此时，应用程序在 OTG_DIEPINTx 中检测到“Tx FIFO 为空时接收到 IN 令牌”中断。应用程序可忽略此中断，因为最终将在周期性帧结束时产生一个未完成同步 IN 传输中断（OTG_GINTSTS 中的 IISOIXFR 位）。
   模块会通过 USB 发送一个零长度数据包来响应接收到的 IN 令牌。

2. 应用程序必须尽快停止向发送 FIFO 写入数据。
3. 应用程序必须将端点的 NAK 位和禁止位置 1。
4. 模块会禁止该端点，将禁止位清零并触发端点的“端点禁止”中断。

应用程序编程顺序：

1. 应用程序可以在任何同步 IN 端点上忽略 OTG_DIEPINTx 中的“Tx FIFO 为空时接收到 IN 令牌”中断，因为最终这将产生一个未完成同步 IN 传输中断（在 OTG_GINTSTS 中）。
2. 硬件触发未完成同步 IN 传输中断（在 OTG_GINTSTS 中）表示在至少一个同步 IN 端点上存在未完成的同步 IN 传输。
3. 应用程序必须读取所有同步 IN 端点的“端点控制”寄存器来检测存在未完成 IN 数据传输的端点。
4. 应用程序必须停止向与这些端点相关联的“周期性发送 FIFO”写入数据。
5. 对 OTG_DIEPCTLx 寄存器中的下列字段进行编程以禁止端点：
   – 在 OTG_DIEPCTLx 中，SNAK = 1
   – 在 OTG_DIEPCTLx 中，EPDIS = 1
6. 硬件触发 OTG_DIEPINTx 中的“端点禁止”中断表示模块已禁止该端点。
   – 此时，应用程序必须清空相关联的发送 FIFO 中的数据，或者通过在下一微帧中使能新传输的端点来覆盖 FIFO 中的现有数据。要刷新数据，应用程序必须使用 OTG_GRSTCTL 寄存器。

- **停止非同步 IN 端点**

本节介绍应用程序如何才能停止非同步端点。

应用程序编程顺序：

1. 禁止要停止的 IN 端点。同时将 STALL 位置 1。
2. OTG_DIEPCTLx 中的 EPDIS = 1（当端点已使能时）
   – OTG_DIEPCTLx 中的 STALL = 1
   – STALL 位的优先级始终高于 NAK 位
3. 硬件触发“端点禁止”中断（在 OTG_DIEPINTx 中），应用程序获知模块已禁止指定的端点。
4. 应用程序必须根据端点类型清空非周期性或周期性发送 FIFO。对于非周期性端点，应用程序必须重新使能另一个无需停止的非周期性端点来发送数据。
5. 当应用程序准备好结束该端点的 STALL 握手信号时，必须将 OTG_DIEPCTLx 的 STALL 位清零。
6. 如果应用程序因收到来自主机的 SetFeature.Endpoint Halt 命令或 ClearFeature.Endpoint Halt 命令来设置或清除端点的 STALL 状态，则必须在该控制端点的状态阶段传输之前将 STALL 位置 1 或清零。

特例：停止控制 OUT 端点

如果在控制传输的数据阶段，主机发送的 IN/OUT 令牌数超过 SETUP 数据包指定的值，则模块必须对这些多余的 IN/OUT 令牌回复 STALL。在这种情况下，在控制传输的数据阶段，当模块传输完 SETUP 数据包指定的数据量后，应用程序必须使能 OTG_DIEPINTx 中的 ITTXFE 中断和 OTG_DOEPINTx 中的 OTEPDIS 中断。随后，当应用程序收到此中断时，必须将相应端点控制寄存器中的 STALL 位置 1 并清除此中断。

57.15.7 最坏情况下的响应时间

当 OTG_FS/OTG_HS 控制器作为设备使用时，对于任何跟随在同步 OUT 之后的令牌，都存在一个最坏响应时间。这个最坏情况响应时间随 AHB 时钟频率的不同而异。

模块寄存器位于 AHB 域内，在更新这些寄存器值之前，模块不会接受新令牌。对于任何跟随在同步 OUT 之后的令牌，最坏的情况是：由于同步事务不需要握手信号，所以下一个令牌可能很快就到达。当 AHB 与 PHY 时钟频率相同时，这个最坏情况值为 7 个 PHY 时钟。AHB 时钟越快，此值越小。

如果发生这种最坏情况，模块将以 NAK 响应批量/中断令牌，并丢弃同步和 SETUP 令牌。对于 SETUP，主机会将这种情况视为超时，并尝试重新发送 SETUP 数据包。对于同步传输，会产生未完成同步 IN 传输中断 (IISOIXFR) 和未完成同步 OUT 传输中断 (IISOOXFR)，来通知应用程序同步 IN/OUT 数据包被丢弃。

选择 OTG_GUSBCFG 中的 TRDT 的值

TRDT 的值 (OTG_GUSBCFG) 是指在接收到 IN 令牌后以 PHY 时钟计算的时间长度，MAC 将在这段时间内获取 FIFO 状态并从 PFC 模块获取第一个数据。这段时间包含 PHY 和 AHB 时钟之间的同步延迟。最坏情况延迟是当 AHB 时钟与 PHY 时钟相等时的延迟。这种情况下的延迟为 5 个时钟周期。

MAC 接收到 IN 令牌后，此信息（接收到的令牌）将由 PFC（PFC 以 AHB 时钟运行）同步到 AHB 时钟。随后，PFC 从 SPRAM 中读取数据并将它们写入双时钟源缓冲区。MAC 再从源缓冲区读出数据（4 个深度）。

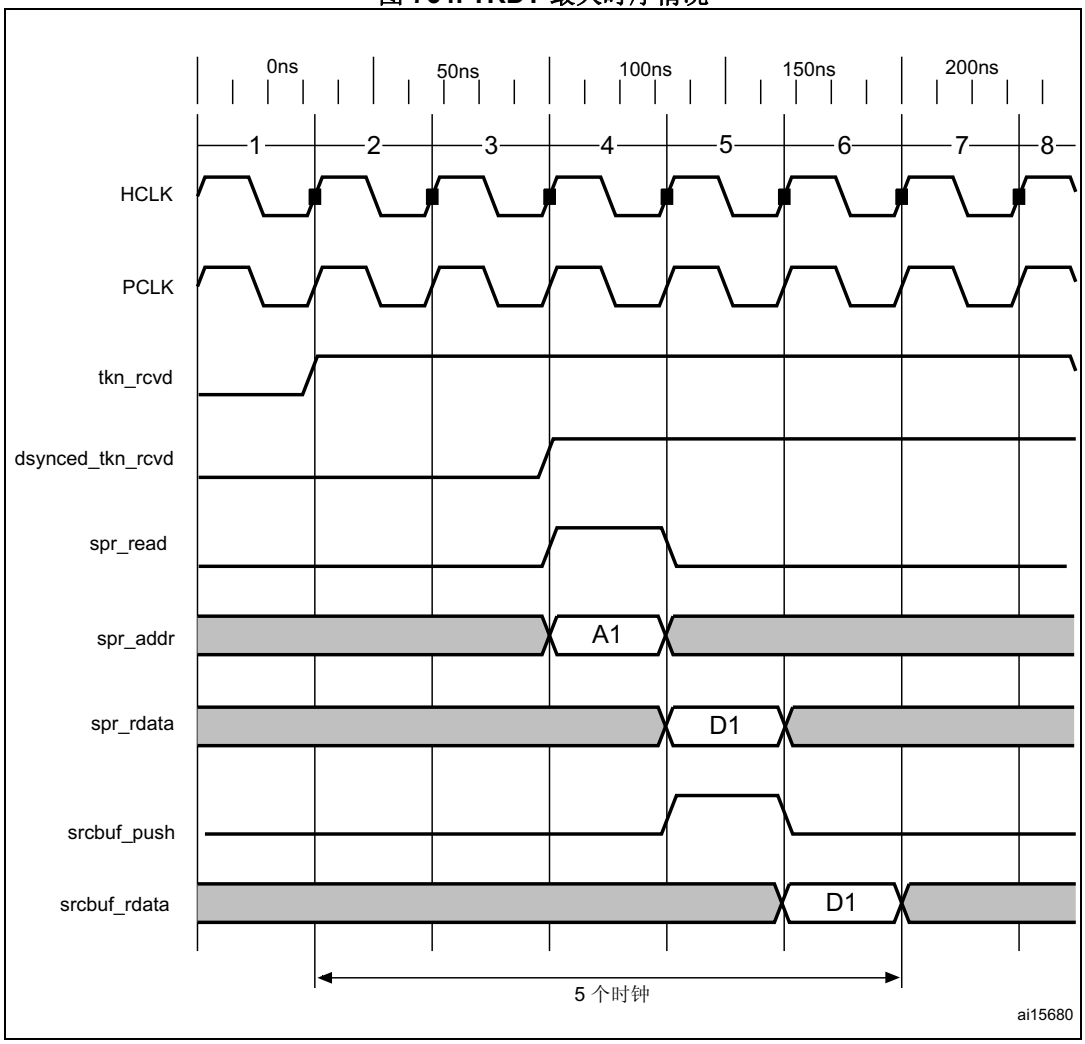
如果 AHB 的运行频率高于 PHY，应用程序可以使用较小的 TRDT 值（在 OTG_GUSBCFG 中）。

图 764 具有以下信号：

- tkn_rcvd：接收到令牌信息（从 MAC 到 PFC）
- dynced_tkn_rcvd：双倍同步 tkn_rcvd（从 PCLK 到 HCLK 域）
- spr_read：读取到 SPRAM
- spr_addr：寻址到 SPRAM
- spr_rdata：从 SPRAM 中读取数据
- srcbuf_push：压入源缓冲区
- srcbuf_rdata：从源缓冲区读取数据。MAC 看到的数据

要计算 TRDT 的值，请参见*表 487：TRDT 值 (HS)*。

图 764. TRDT 最大时序情况



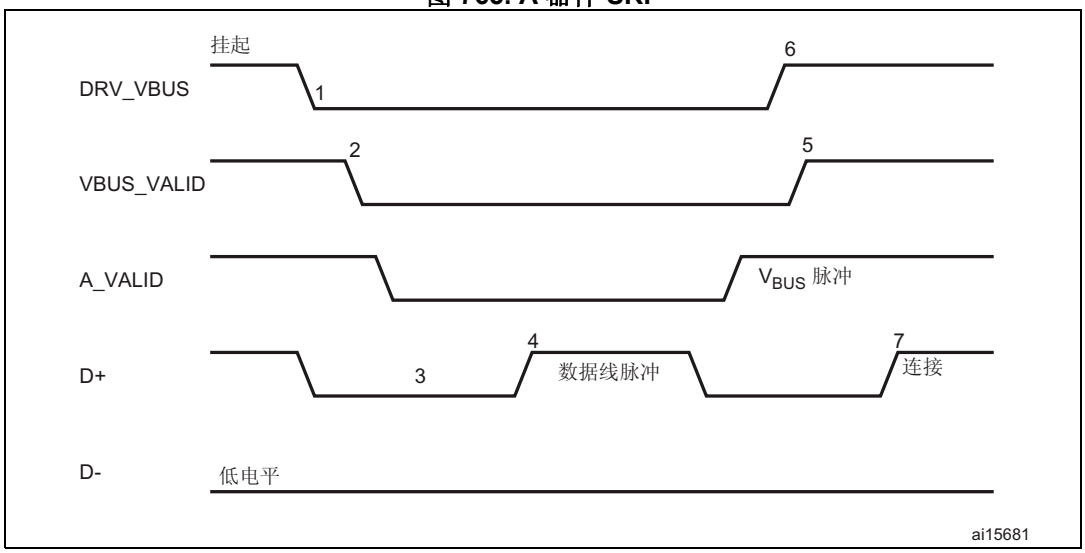
57.15.8 OTG 编程模型

OTG_HS 控制器是一种支持 HNP 和 SRP 的 OTG 设备。该模块接口与“A 型”插头连接时，该模块称为 A 器件。该模块接口与“B 型”插头连接时，该模块称为 B 器件。在主机模式下，OTG_HS 控制器将关闭 VBUS 以节省电能。B 器件可以借助 SRP 请求 A 器件打开 V_{BUS} 电源。设备必须同时执行数据线脉冲和 V_{BUS} 脉冲，但主机可以检测数据线脉冲或者 V_{BUS} 脉冲中的一个用于 SRP。B 器件可以借助 HNP 协商并切换到主机角色。在协商模式下执行 HNP 后，B 器件会挂起总线并恢复到设备角色。

A 器件会话请求协议

应用程序必须将模块 USB 配置寄存器中的 SRP 使能位置 1。这将使能 OTG_HS 控制器在 A 器件模式下检测到 SRP。

图 765. A 器件 SRP



1. DRV_VBUS = 发送到 PHY 的 V_{BUS} 驱动信号
   VBUS_VALID = 来自 PHY 的 V_{BUS} 有效信号
   A_VALID = 发送到 PHY 的 A 器件 V_{BUS} 电平信号
   D+ = 正向数据线
   D- = 负向数据线

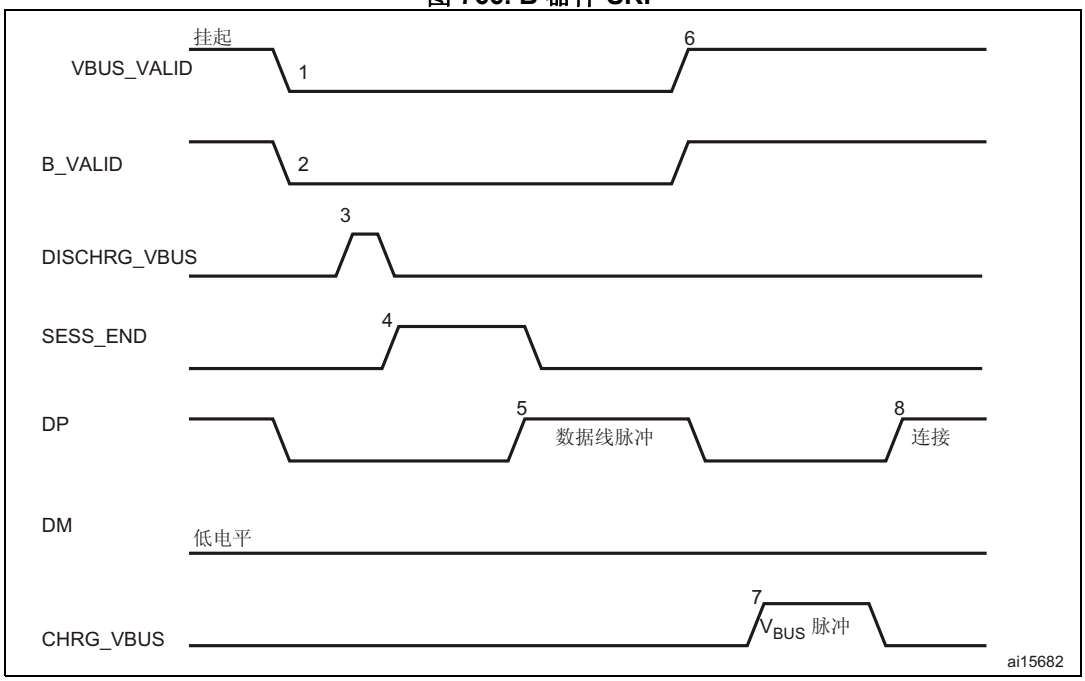
以下几点引用并描述了*图 765* 所示的信号过程：

1. 为节省电能，在总线空闲时，应用程序会通过在主机端口控制和状态寄存器中写入端口挂起位和端口电源位来挂起设备并关闭端口电源。
2. PHY 通过禁止 VBUS_VALID 信号来指示端口电源已关闭。
3. 当 V_{BUS} 电源关闭时，设备必须检测到至少 2 ms 的 SE0 信号才可以启动 SRP。
4. 要启动 SRP，设备需要打开其数据线的上拉电阻并维持 5 到 10 ms。OTG_HS 控制器会检测到数据线脉冲。
5. 设备会驱动 V_{BUS} 到 A 器件会话有效电平（最小 2.0 V）以上，以产生 V_{BUS} 脉冲。
   OTG_HS 控制器检测到 SRP 时将中断应用程序。在全局中断状态寄存器中，检测到会话请求位（OTG_GINTSTS 中的 SRQINT 位）将置 1。
6. 应用程序必须响应“检测到会话请求”中断，并通过在主机端口控制和状态寄存器中写入端口电源位来打开端口电源位。PHY 通过输出 VBUS_VALID 信号来指示端口已通电。
7. 当 USB 通电后，设备将连接，从而完成 SRP 过程。

B 器件会话请求协议

应用程序必须将模块 USB 配置寄存器中的 SRP 使能位置 1。这将使能 OTG_HS 控制器在 B 器件模式下启动 SRP。OTG_HS 控制器可以借助 SRP 向主机请求新会话。

图 766. B 器件 SRP



1. VBUS_VALID = 来自 PHY 的 V_{BUS} 有效信号
   B_VALID = 发送到 PHY 的 B 器件有效会话
   DISCHRG_VBUS = 发送到 PHY 的放电信号
   SESS_END = 发送到 PHY 的会话结束信号
   CHRG_VBUS = 发送到 PHY 的 V_{BUS} 充电信号
   DP = 正向数据线
   DM = 负向数据线

以下几点引用并描述了*图 766* 所示的信号过程：

1. 为节省电能，在总线空闲时主机会挂起并关闭端口电源。

   OTG_HS 控制器会在总线空闲 3 ms 后，将模块中断寄存器中的早期挂起位置 1。随后，OTG_HS 控制器会将模块中断寄存器中的 USB 挂起位置 1。

   OTG_HS 控制器会通知 PHY 对 VBUS 进行放电。
2. PHY 指示会话结束。这是 SRP 的初始条件。启动 SRP 之前，OTG_HS 控制器需要 2 ms 的 SE0 信号。

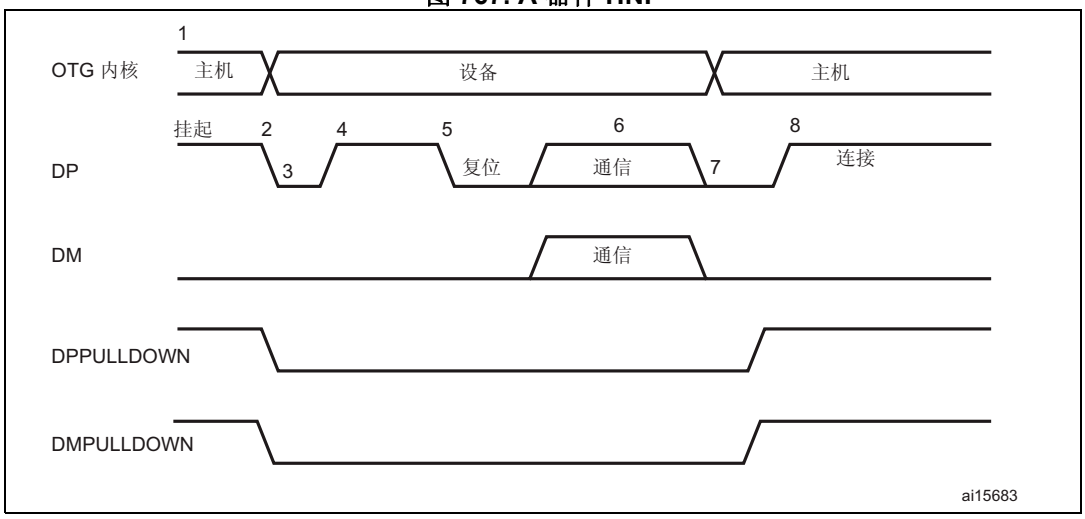
   对于 USB 1.1 全速串行收发器，应用程序必须在 BSVLD 位（位于 OTG_GOTGCTL）被禁止后，等待 V_{BUS} 放电至 0.2 V。此放电时间可从收发器供应商处获取，该值可能因收发器而异。
3. OTG_HS 模块通知 PHY 对 V_{BUS} 加速放电。
4. 应用程序通过将 OTG 控制和状态寄存器中的会话请求位置 1 来启动 SRP。OTG_HS 控制器先执行数据线脉冲，随后执行 VBUS 脉冲。
5. 主机会从数据线脉冲或 V_{BUS} 脉冲检测到 SRP，然后打开 V_{BUS}。PHY 指示 V_{BUS} 对设备上电。
6. OTG_HS 控制器执行 VBUS 脉冲。

   主机通过打开 V_{BUS} 启动一个新会话，指示 SRP 已成功。OTG_HS 控制器通过将 OTG 中断状态寄存器中的会话请求成功状态更改位置 1 来中断应用程序。应用程序读取 OTG 控制和状态寄存器中的会话请求成功位。
7. 当 USB 通电时，OTG_HS 控制器将进行连接，并完成 SRP 过程。

A 器件主机协商协议

通过 HNP 可以将 USB 主机角色从 A 器件切换到 B 器件。应用程序必须将模块 USB 配置寄存器中的 HNP 使能位置 1，以使能 OTG_HS 控制器在 A 器件模式下执行 HNP 功能。

图 767. A 器件 HNP



1. DPPULLDOWN = 从模块发送到 PHY 的信号，用于使能/禁止 PHY 内部 DP 线的下拉电阻。
   DMPULLDOWN = 从模块发送到 PHY 的信号，用于使能/禁止 PHY 内部 DM 线的下拉电阻。

以下几点引用并描述了*图 767* 所示的信号过程：

1. OTG_HS 控制器向 B 器件发送 SetFeature b_hnp_enable 描述符以使能 HNP 支持。B 器件的 ACK 响应表示 B 器件支持 HNP。应用程序必须将 OTG 控制和状态寄存器中的主机设置 HNP 使能位置 1，以向 OTG_FS/OTG_HS 控制器指示 B 器件支持 HNP。
2. 应用程序不再使用总线时，会通过在主机端口控制和状态寄存器中写入端口挂起位来挂起总线。
3. B 器件检测到 USB 挂起时，将断开连接，指示 HNP 的初始条件。B 器件只有在切换到主机角色时才会启动 HNP，否则总线会保持挂起状态。

   OTG_HS 控制器将 OTG 中断状态寄存器中检测到的主机协商中断位置 1，指示 HNP 的启动。

   OTG_HS 控制器将禁止 PHY 中 DP 线和 DM 线的下拉，以指示设备角色。PHY 使能 OTG_DP 上拉电阻，以告知 B 器件 A 器件的连接。

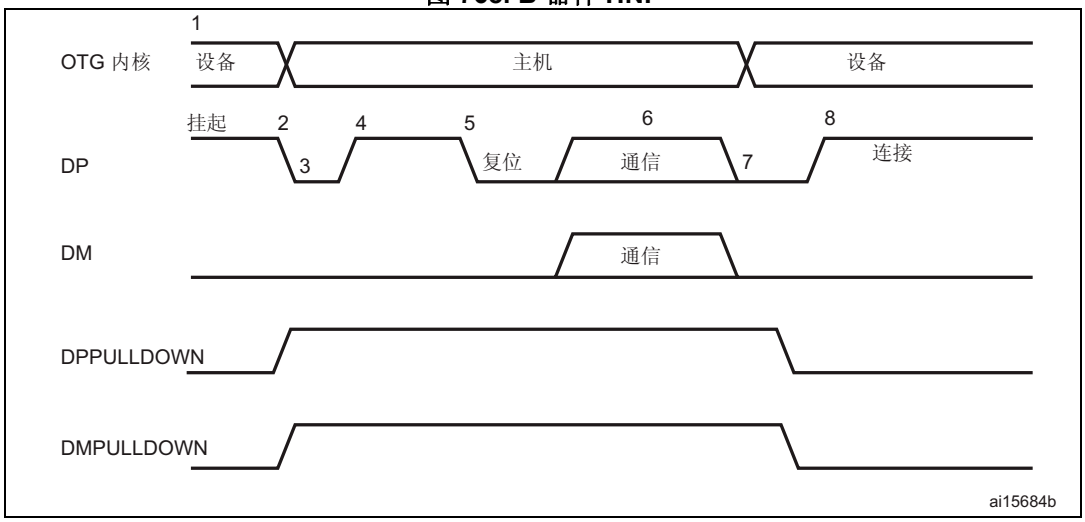
   应用程序必须读取 OTG 控制和状态寄存器中的当前模式位，以确定设备工作模式。
4. B 器件检测到连接，发出 USB 复位信号，并枚举 OTG_HS 进行数据通信。
5. B 器件继续担当主机角色，发起数据通信，并在结束时挂起总线。

   OTG_HS 控制器会在总线空闲 3 ms 后，将模块中断寄存器中的早期挂起位置 1。随后，OTG_HS 控制器会将模块中断寄存器中的 USB 挂起位置 1。
6. 在协商模式下，OTG_HS 控制器会检测到总线挂起，连接断开，然后切换回主机角色。OTG_HS 控制器将使能 PHY 中 DM 和 DM 下拉电阻，以指示其承担主机角色。
7. OTG_HS 控制器将 OTG 中断状态寄存器中的连接器 ID 状态更改中断置 1。应用程序必须读取 OTG 控制和状态寄存器中的连接器 ID 状态位，以确定 OTG_HS 控制器在 A 器件模式下工作。这向应用程序表示 HNP 已经完成。应用程序必须读取 OTG 控制和状态寄存器中的当前模式位，以确定主机工作模式。
8. 连接 B 器件，完成 HNP 过程。

B 器件主机协商协议

通过 HNP 可以将 USB 主机角色从 B 器件切换到 A 器件。应用程序必须将模块 USB 配置寄存器中的 HNP 功能位置 1，以使能 OTG_HS 控制器作为 B 器件的 HNP 功能。

图 768. B 器件 HNP



1. DPPULLDOWN = 从模块发送到 PHY 的信号，用于使能/禁止 PHY 内部 DP 线的下拉电阻。
   DMPULLDOWN = 从模块发送到 PHY 的信号，用于使能/禁止 PHY 内部 DM 线的下拉电阻。

以下几点引用并描述了*图 768* 所示的信号过程：

1. A 器件发出一个 SetFeature b_hnp_enable 描述符以使能 HNP 支持。OTG_HS 控制器的 ACK 响应表示它支持 HNP。应用程序必须将 OTG 控制和状态寄存器中的设备 HNP 使能位置 1，以指示支持 HNP。

   应用程序将 OTG 控制和状态寄存器中的 HNP 请求位置 1，以向 OTG_HS 控制器指示启动 HNP。

2. A 器件不再使用总线时，会通过在主机端口控制和状态寄存器中写入端口挂起位来挂起总线。

   OTG_HS 控制器会在总线空闲 3 ms 后，将模块中断寄存器中的早期挂起位置 1。随后，OTG_HS 控制器会将模块中断寄存器中的 USB 挂起位置 1。

   OTG_HS 控制器会断开连接，A 器件在总线上检测到 SE0，指示 HNP 即将开始。OTG_HS 控制器将使能 PHY 中的 DP 和 DM 下拉电阻，以指示其承担主机角色。

   在检测到 SE0 的 3 ms 内，A 器件会通过激活 OTG_DP 上拉电阻进行响应。OTG_HS 控制器检测到此信号时认为有设备接入。

   OTG_HS 控制器将 OTG 中断状态寄存器中的主机协商成功状态更改中断位置 1，指示 HNP 的状态。应用程序必须读取 OTG 控制和状态寄存器中的主机协商成功位，以确定主机协商成功。应用程序必须读取模块中断寄存器 (OTG_GINTSTS) 中的当前模式位，以确定主机工作模式。

3. 应用程序将复位位（OTG_HS_HPRT 中的 PRST 位）置 1，同时 OTG_HS 控制器发出 USB 复位信号，并枚举 A 器件进行数据通信。
4. OTG_HS 控制器继续保持发起通信的主机角色，在通信结束后，会通过将主机端口控制和状态寄存器中的端口挂起位置 1 来挂起总线。
5. 在协商模式下，当 A 器件检测到挂起时，会断开连接，然后切换回主机角色。OTG_HS 控制器将禁止 PHY 中 DP 线和 DM 线的下拉，以指示其假设的设备角色。
6. 应用程序必须读取模块中断寄存器 (OTG_GINTSTS) 中的当前模式位，以确定主机工作模式。
7. OTG_HS 控制器进行连接，完成 HNP 过程。

58 以太网 (ETH)：通过 DMA 控制器进行介质访问控制 (MAC)

58.1 以太网简介

Portions Copyright (c) 2004, 2005 Synopsys, Inc. 保留所有权利。使用须经许可。

借助以太网外设，器件可以通过以太网按照 IEEE 802.3-2002 标准发送和接收数据。

以太网提供了可配置、灵活的外设，用以满足客户的各种应用需求。它支持与外部物理层 (PHY) 相连的两个工业标准接口：默认情况下使用的介质独立接口 (MII)（在 IEEE 802.3 规范中定义）和简化介质独立接口 (RMII)。它有多种应用领域，例如交换机和网络接口卡。

除了 IEEE 802.3 规范中定义的默认接口外，以太网外设还支持多个与 PHY 相连的工业标准接口。它符合以下标准：

- IEEE 802.3-2008，用于以太网 MAC 和介质独立接口 (MII)
- IEEE 1588-2008，用于精密网络时钟同步协议 (PTP)
- IEEE 802.1AS-2011 和 802.1-Qav-2009，用于音频视频 (AV) 通信
- IEEE 802.3az-2010，用于节能型以太网 (EEE)
- AMBA 2.0，用于 AHB 主端口和 AHB 从端口
- RMII 联盟的 RMII 规范第 1.2 版

58.2 以太网主要特性

以太网外设嵌入了针对直接存储器接口的专用 DMA、介质访问控制器 (MAC) 和支持多种格式的 PHY 接口模块。

58.2.1 MAC 内核特性

接口

- 为应用提供单独的发送、接收和控制接口
- 应用侧有 32 位数据传输接口
- 支持以下 PHY 接口实现 10/100 Mb/s 数据传输速率：
  - 符合 IEEE 802.3 的 MII（默认）接口，与外部快速以太网 PHY 进行通信
  - RMII 接口，与外部快速以太网 PHY 进行通信
- 用于配置和管理 PHY 器件的 MDIO（符合条款 22 和条款 45）主接口
- 支持通过 RMON 计数器 (RFC2819/RFC2665) 进行强制网络统计

主要操作

- 支持全双工和半双工操作：
  - 适用于半双工操作的 CSMA/CD 协议
  - 适用于全双工操作的 IEEE 802.3x 流控制
  - 全双工操作时可以将接收的暂停控制帧转发到用户应用程序
  - 半双工操作时提供背压功能
  - 全双工操作中如果流控制输入信号消失，将自动发送零时间片暂停帧
- 全双工流控制操作（IEEE 802.3x 暂停数据包和优先级流控制）
- 报头和帧起始数据 (SFD) 在发送模式下插入、在接收路径中删除
- 可逐帧控制 CRC 和 pad 自动生成
- 可编程数据包长度，支持标准以太网数据包或高达 16 KB 的巨型以太网数据包
- 可编程数据包间隙
- 已接收数据包的第 3 层/第 4 层校验和减荷
- 计算 IPv4 头校验和与 TCP、UDP 或 ICMP 校验和并将其插入在存储转发模式下发送的帧中
- 两组 FIFO：一个具有可编程阈值功能的 2048 字节发送 FIFO 和一个具有可配置阈值功能的 2048 字节接收 FIFO
- 存储转发机制或阈值模式（直通），用于向 MAC 发送数据
- 可编程 Rx 队列阈值（或直通）模式
- 在 MII 上进行从 Tx 到 Rx 的内部回送以用于调试。

VLAN 管理

- 源地址字段插入或替换，以及发送数据包中的 VLAN 插入、替换和删除（按数据包控制或采用静态-全局控制）
- 插入、替换或删除最多两个 VLAN 标记
- IEEE 802.1Q VLAN 标记检测，并且可删除接收数据包中的 VLAN 标记
- 去除最多两个 VLAN 标记，以及在状态中提供标记。

数据包过滤

- 灵活的地址过滤模式：
  - 3 个附加的 48 位完美目标地址 (DA) 过滤器，对每个字节进行屏蔽操作
  - 3 次附加的 48 位源地址 (SA) 比较检查，对每个字节进行屏蔽操作
  - 64 位散列过滤器，适用于多播和单播 (DA) 地址
  - 可传送所有多播地址数据包
  - 支持混合模式，不进行过滤，直接传送所有数据包，用于网络监控
  - 传送所有传入数据包时（每次过滤时）均附有一份状态报告
- 附加数据包过滤：
  - 基于 VLAN 标记：完美匹配和散列过滤（基于外部或内部 VLAN 标记进行过滤）
  - 基于第 3 层和第 4 层：TCP/UDP 基于 IPv4/IPv6

IEEE 1588-2008/PTPv2

- 以太网数据包时间戳如 IEEE 1588-2002 和 IEEE 1588-2008 规范所述（PTP 数据包的 Tx 或 Rx 状态中给出了 64 位时间戳）。Tx 方向支持一步和两步时间戳。
- 可灵活控制每秒脉冲数 (PPS) 输出信号 (ptp_pps_o)。

低功耗模式

- 标准 IEEE 802.3az-2010，用于 MII PHY 中的节能型以太网。
- 用于检测远程唤醒数据包和 AMD 魔术数据包的模块

58.2.2 DMA 特性

DMA 模块在外设和系统存储器之间交换数据。DMA 传输由软件描述符结构驱动。应用程序可使用一组寄存器（请参见*第 58.11.2 节：以太网 DMA 寄存器*）来控制 DMA 操作。DMA 模块支持以下特性：

- 32 位数据传输
- 发送路径和接收路径中分别具有单独的 DMA
- 通过数据包定界符优化以数据包为导向的 DMA 传输
- 支持对数据缓冲区进行字节对齐寻址
- 支持双缓冲区（环）描述符
- 采用描述符架构，可以在 CPU 几乎不干预的情况下传输大型数据块（每个描述符最多可传输 32 KB 的数据）
- 报告正常工作和传输错误的综合状态
- 可为 Tx DMA 和 Rx DMA 引擎单独编程突发长度，以充分利用主总线
- 可编程中断选项，适用于不同的工作条件
- 按数据包控制发送或接收完成中断
- 接收引擎和发送引擎间采用循环调度仲裁或固定优先级仲裁
- 启动模式和停止模式
- 主机控制 (AHB) 访问和主机数据接口分别采用单独的端口
- 使能了 TCP 分段减荷 (TSO) 功能的 Tx DMA 通道

58.2.3 总线接口功能

AHB 主接口

AHB 主接口的特性如下：

- 通过 AHB 与应用进行接口
- 小端模式
- AHB 主端口上的 32 位数据
- 拆分、重试和错误 AHB 响应
- AHB 1K 边界突发拆分
- 可通过软件选择 AHB 突发类型（固定突发、不确定突发或二者混合突发）

AHB 主接口不会生成以下响应：

- 回卷突发
- 锁定或受保护的传输

AHB 从接口

AHB 从接口支持以下特性：

- 通过 AHB 与应用进行接口
- 小端模式
- AHB 从接口（32 位），用于 CSR 访问
- 所有 AHB 突发类型

AHB 从接口不会生成以下响应：

- 拆分
- 重试
- 错误

58.3 以太网引脚和内部信号

列出了连接到封装引脚或焊球的以太网输入与输出信号，*表 491* 则列出了内部以太网信号。

表 490. 以太网外设引脚

引脚名称	复用功能名称（映射到 AF11）
PA0	ETH_MII_CRS
PA1	ETH_MII_RX_CLK/ETH_RMII_REF_CLK
PA2	ETH_MDIO
PA3	ETH_MII_COL
PA7	ETH_MII_RX_DV/ETH_RMII_CRS_DV
PA9	ETH_TX_ER
PB0	ETH_MII_RXD2
PB1	ETH_MII_RXD3
PB2	ETH_TX_ER
PB5	ETH_PPS_OUT
PB8	ETH_MII_TXD3
PB10	ETH_MII_RX_ER
PB11	ETH_MII_TX_EN/ETH_RMII_TX_EN
PB12	ETH_MII_TXD0/ETH_RMII_TXD0
PB13	ETH_MII_TXD1/ETH_RMII_TXD1
PC1	ETH_MDC
PC2	ETH_MII_TXD2
PC3	ETH_MII_TX_CLK
PC4	ETH_MII_RXD0/ETH_RMII_RXD0

表 490. 以太网外设引脚（续）

引脚名称	复用功能名称（映射到 AF11）
PC5	ETH_MII_RXD1/ETH_RMII_RXD1
PE2	ETH_MII_TXD3
PG8	ETH_PPS_OUT
PG11	ETH_MII_TX_EN/ETH_RMII_TX_EN
PG12	ETH_MII_TXD1/ETH_RMII_TXD1
PG13	ETH_MII_TXD0/ETH_RMII_TXD0
PG14	ETH_MII_TXD1/ETH_RMII_TXD1
PH2	ETH_MII_CRS
PH3	ETH_MII_COL
PH6	ETH_MII_RXD2
PH7	ETH_MII_RXD3
PI10	ETH_MII_RX_ER
PI12	ETH_TX_ER

表 491. 以太网内部输入/输出信号

信号名称	信号类型	说明
eth_hclk	数字输入	AHB 时钟
eth_sbd_intr_it	数字输出	主要以太网中断
lpi_intr_o	数字输出	发送器或接收器进入或退出 LPI 状态时生成的边带信号。
pmt_intr_o	数字输出	接收到有效远程唤醒数据包时生成的边带信号
eth_mii_tx_clk	数字输入	MII Tx 内核时钟
eth_mii_rx_clk	数字输入	MII Rx 内核时钟
eth_rmii_ref_clk	数字输入	RMII 参考内核时钟

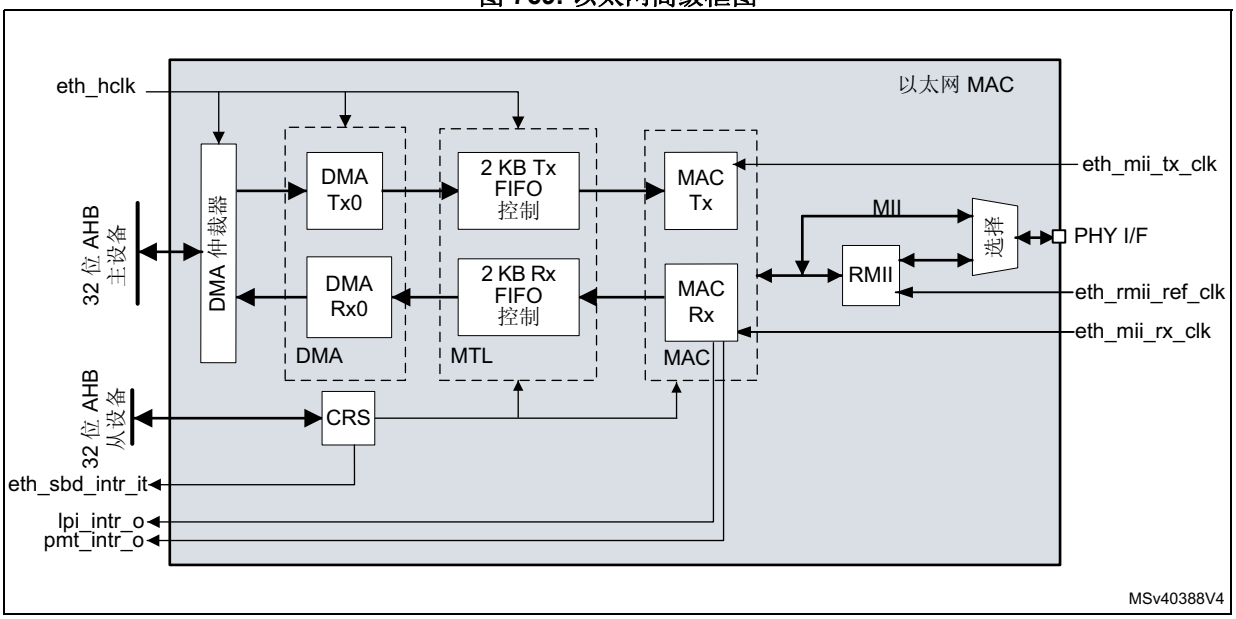
58.4 以太网架构

以太网外设由 4 个主要功能模块组成：

- **控制和状态寄存器模块 (CSR)**，用于控制通过 AHB 32 位从接口进行的寄存器访问
- **直接存储器访问接口 (DMA)**
  此为逻辑 DMA 模块，具有 1 条用于接收的物理通道和 1 条用于发送的物理通道。该模块用于控制通过 AMBA AHB 32 位主接口在 MAC 和系统存储器之间进行的数据传输。
- **介质访问控制模块 (MAC)**，负责实现以太网协议
- **MAC 事务层 (MTL)**，负责控制应用和 MAC 之间的数据流

此外，还增添了协议适配模块以支持 RMII PHY 介质独立接口。

图 769. 以太网高级框图



1. 有关内部信号的定义，请参见*表 491*。
2. 有关以太网时钟架构的详细说明，请参见 RCC 一章的“以太网的时钟分配”。

58.4.1 DMA 控制器

DMA 具有独立的发送 (Tx) 和接收 (Rx) 引擎。Tx 引擎将数据从系统存储器传输到 MAC 事务层 (MTL)，而 Rx 引擎将数据从设备端口 (PHY) 传输到系统存储器。

此控制器可以在应用 CPU 完全不干预的情况下，通过描述符有效地将数据从源传送到目标。DMA 专为面向数据包的数据传送（如以太网中的数据包）而设计。该控制器经过编程后，可在完成数据包发送和接收传送操作时以及其他正常或错误条件下向应用 CPU 提供中断。

DMA 数据结构

DMA 和应用通过以下两种数据结构实现通信：

- 控制和状态寄存器 (CSR)
- 描述符列表和数据缓冲区

DMA 既可将 MAC 接收到的数据包传送到系统存储器中的 Rx 缓冲区，也可传送来自系统存储器的 Tx 缓冲区的 Tx 数据包。位于系统存储器中的描述符包含指向这些缓冲区的指针。

每个列表的基址均写入到相应的 Tx 和 Rx 寄存器：*通道 Tx 描述符列表地址寄存器 (ETH_DMACTxDLAR)* 和*通道 Rx 描述符列表地址寄存器 (ETH_DMACRxDLAR)*。

描述符列表是一种前向链表，相邻两个描述符之间始终相差一个固定的偏移值。列表中的描述符数量在相应的 Tx/Rx（*通道 Tx 描述符环长度寄存器 (ETH_DMACTxRLR)* 和*通道 Rx 描述符环长度寄存器 (ETH_DMACRxRLR)*）中进行编程。

DMA 处理完列表中的最后一个描述符后，会自动跳转回列表地址寄存器中的相应描述符，以形成一个描述符环。描述符列表位于应用的物理存储地址空间内。每个描述符最多可指向两个缓冲区。因此，可使用并以物理方式寻址两个缓冲区，而非存储器中的连续缓冲区。

数据缓冲区位于应用的物理存储空间，通常由整个数据包或部分数据包组成，但不会超过单个数据包。缓冲区中仅包含数据。其状态保存在描述符中。数据链接是指跨越多个数据缓冲区的数据包。但是，单个描述符不能跨越多个数据包。检测到 EOP 时，DMA 会跳到下一个数据包的数据缓冲区。

*第 58.10 节：描述符*中指定了描述符。

DMA 仲裁

DMA 模块配有一个仲裁器，可在 Tx 通道和 Rx 通道对 AHB 主接口进行的访问之间进行仲裁。支持以下两种类型的仲裁（可通过 *DMA 模式寄存器 (ETH_DMAMR)* 进行选择）：

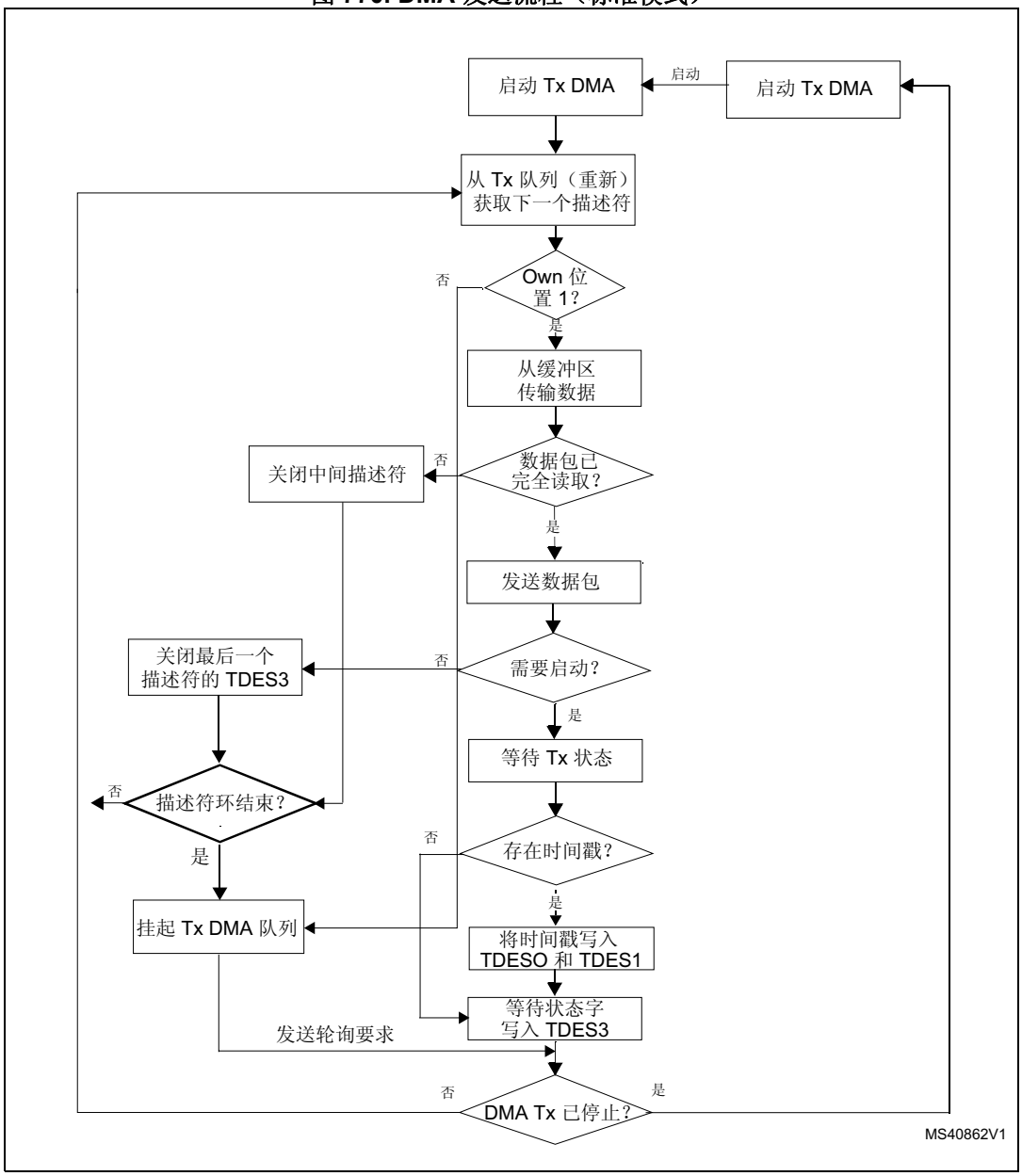
- 循环调度仲裁：仲裁器按照 ETH_DMAMR 的位 [14:12] 设置的比率，在 Rx 和 Tx 之间分配数据总线。
- 固定优先级仲裁：默认情况下，Rx DMA 的数据访问优先级始终高于 Tx DMA。将 ETH_DMAMR 寄存器的位 11 置 1 可为 Tx DMA 分配高优先级。

默认模式下的 DMA 发送

Tx DMA 引擎在默认模式下的操作顺序如下：

1. 在为数据缓冲区设置好相应的以太网数据包数据后，应用配置发送描述符 (TDES0–TDES3) 并将 Own 位 (TDES0[31]) 置 1。
2. 应用对发送通道的描述符尾指针偏移值进行移位。
3. DMA 从应用的存储器中获取描述符。
4. 如果 DMA 检测到以下条件之一，则会挂起该通道的发送操作，相应 DMA 通道状态寄存器的位 2 和位 16 会置 1，Tx 引擎将执行步骤 11：
   – 描述符被标记为应用所有（TDES3 [31] = 0）。
   – 在环描述符列表模式下，描述符尾指针与当前描述符指针相等。
   – 出现错误条件。
5. 如果获得的描述符标记为 DMA 所有（TDES3[31] = 1），则 DMA 将根据取得的描述符对发送数据缓冲区地址进行解码。
6. DMA 从系统存储器中获取发送数据并将数据传送到 MTL 进行发送。
7. 如果以太网数据包保存在多个描述符的数据缓冲区中，则 DMA 将关闭中间描述符并获取下一个描述符。重复执行步骤 3 到步骤 7，直到以太网数据包的结束数据传送到 MTL。
8. 完成数据包发送后，如果已为数据包使能 IEEE 1588 时间戳功能 （按 Tx 状态中的指示），则从 MTL 获得的时间戳值将写入包含 EOP 缓冲区的 Tx 描述符（TDES0 和 TDES1）。随后，状态信息将写入该 Tx 描述符 (TDES3)。该描述符现在为应用所有，因为 Own 位在此步骤中被清零。如果为该数据包禁止时间戳功能，则 DMA 不会更改 TDES0 和 TDES1 的内容。
9. 发送完在其最后一个描述符中将完成时中断 (TDES2[31]) 置 1 的数据包后，*通道状态寄存器 (ETH_DMACSR)* 的位 0 将置 1。随后，DMA 引擎返回步骤 3。
10. 在挂起状态下， DMA 尝试再次获取描述符 （进而返回到步骤 3）。当接收到发送轮询要求并且下溢中断状态位清零时，通过对*通道 Tx 描述符尾指针寄存器 (ETH_DMACTxDTPR)* 写入任一值来触发轮询要求命令。如果应用通过将相应 DMA 通道的发送控制寄存器的位 0 清零来停止 DMA， DMA 将进入停止状态。

图 770. DMA 发送流程（标准模式）



OSP（处理第二个数据包）模式下的 DMA 发送

在运行状态下，如果将*通道发送控制寄存器 (ETH_DMACTxCR)* 的位 4 置 1，则发送过程可同时获取两个数据包，而无需关闭第一个数据包的状态描述符。发送过程完成第一个数据包的传送后，会立即轮询第二个数据包的发送描述符列表。如果第二个数据包有效，则发送过程会在写入第一个数据包的状态信息前发送此数据包。

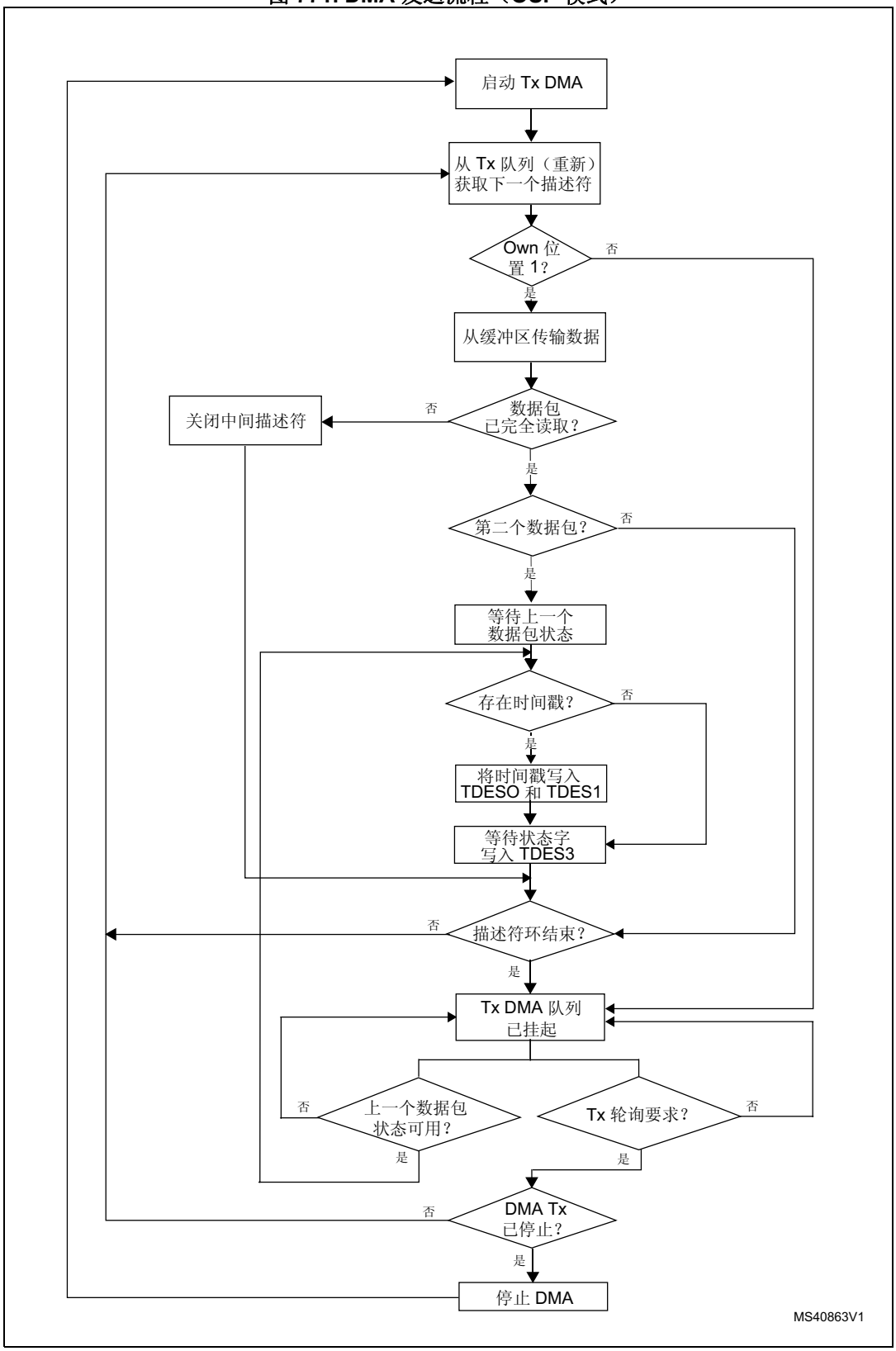
在 OSP 模式下，运行状态下的 DMA 按以下序列进行发送操作：

1. 在默认模式下，DMA 执行 DMA 发送序列的步骤 1 到 7（请参见*默认模式下的 DMA 发送*一节）。
2. DMA 无需关闭前一数据包的最后一个描述符，即可获取下一个描述符。
3. 如果 DMA 拥有所需描述符，便会对该描述符中的发送缓冲区地址进行解码。如果 DMA 未拥有描述符，则会进入挂起模式并跳到步骤 7。
4. DMA 从系统存储器中获取发送数据包并将该数据包发送至 MTL，直到发送完 EOP 数据，如果该数据包拆分到多个描述符中，则会同时关闭中间描述符。
5. DMA 等待前一个数据包的数据包发送状态和时间戳。当状态可用时，如果捕获到时间戳（由状态位指示），DMA 会将这些时间戳写入 TDES0 和 TDES1。Own 位清零，DMA 将状态写入相应的 TDES3，进而关闭描述符。如果没有为前一个数据包使能时间戳功能，则 DMA 不会更改 TDES2 和 TDES3 的内容。
6. 发送中断将置 1（如果已使能）。DMA 获取下一个描述符，然后继续执行步骤 3（状态正常时）。如果上一个发送状态显示下溢错误，则 DMA 会进入挂起模式（步骤 7）。
7. 在挂起模式下，如果从 MTL 接收到挂起状态和时间戳，则 DMA 会执行以下操作：
   a) DMA 将时间戳（如果已为当前数据包使能）写入 TDES2 和 TDES3。
   b) DMA 将状态写入相应的 TDES3。
   c) DMA 将相关中断置 1 并返回挂起模式。

   如果没有挂起状态，且应用通过将相应 DMA 通道的发送控制寄存器的位 0 清零来停止 DMA，DMA 将进入停止状态。
8. 只有在接收到发送轮询要求（在相应通道的发送描述符尾指针寄存器中）后，DMA 才会退出挂起模式并进入运行状态（转到步骤 1 或步骤 2，具体取决于挂起状态）。

*图 771：DMA 发送流程（OSP 模式）*给出了 OSP 模式下基本 DMA 发送流程的说明。

图 771. DMA 发送流程（OSP 模式）



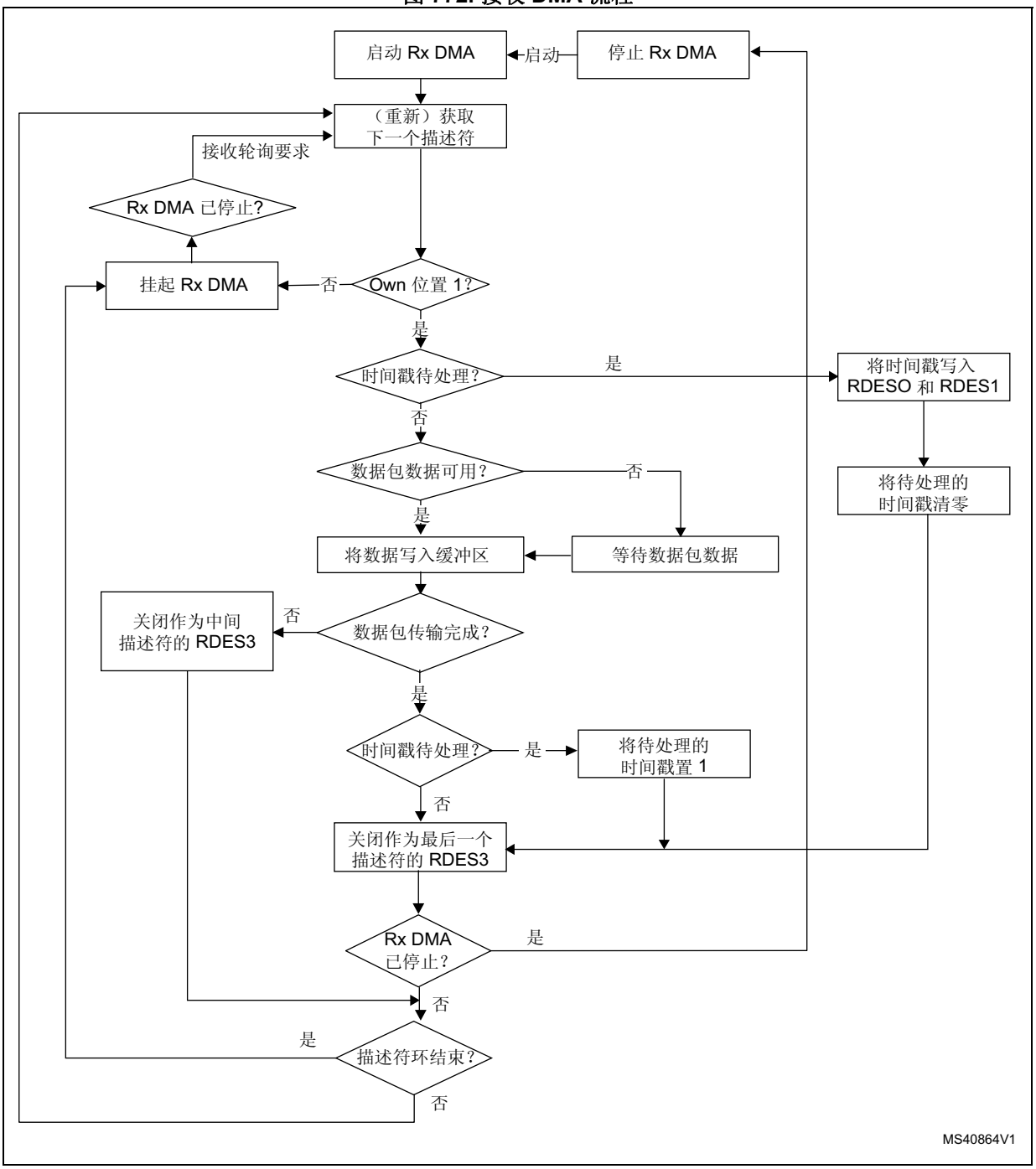
DMA 接收

在接收路径中，DMA 从 MTL 接收队列读取数据包，并将其写入相应 DMA 通道的数据包数据缓冲区。

Rx DMA 引擎的接收序列如下（另请参见*图 772：接收 DMA 流程*）：

1. 应用设置 Rx 描述符 (RDES0-RDES3) 并将 Own 位 (RDES3[31]) 置 1。应用必须在相应 DMA 通道的接收描述符尾指针寄存器中设置正确的值。
2. *通道接收控制寄存器 (ETH_DMACRxCR)* 的位 0 置 1 时，DMA 进入运行状态。DMA 根据 Rx 当前描述符和描述符尾指针寄存器值查找空闲描述符。如果没有空闲描述符，则 DMA 通道会进入挂起状态并转到步骤 11。
3. DMA 在环中获取下一个可用描述符，并对来自所获取描述符的接收数据缓冲区地址进行解码。
4. 如果使能了 IEEE 1588 时间戳，并且该时间戳可用于前一数据包，则 DMA 会将该时间戳（如果可用）写入当前描述符的 RDES0 和 RDES1，并将 CTXT 字段 (RDES3[30]) 置 1。
5. DMA 处理传入数据包并将其存储在所获取描述符的数据缓冲区中。
6. 如果当前数据包传输尚未完成，则 DMA 会将当前描述符作为中间描述符关闭并转至步骤 10。
7. DMA 从 MTL 获取接收帧的状态并将该状态字写入当前描述符，同时 Own 位清零且最后一个描述符位置 1。
8. DMA 将帧长度写入 RDES3，将 VLAN 标记写入 RDES0。DMA 还会将 MAC 控制帧操作码、OAM 控制帧代码和扩展状态信息（如果可用）写入最后一个描述符的 RDES1。
9. 如果使能了 IEEE 1588 时间戳功能，则 DMA 会存储此时间戳（如果可用）。DMA 在当前数据包的最后一个描述符后将上下文描述符写入下一个可用描述符中。
10. 如果 Rx DMA 描述符环中有多个描述符可用，请转到步骤 3，否则转到挂起状态（步骤 11）。
11. 当收到接收轮询要求命令并且应用使通道接收尾指针寄存器的值递增时，接收 DMA 将退出挂起状态。引擎继续执行步骤 2 并获取下一个描述符。

图 772. 接收 DMA 流程



58.4.2 MTL

MAC 事务层 (MTL) 提供 FIFO 存储器接口，来缓冲和调整应用系统存储器和 MAC 之间的数据包。它还可以在应用时钟和 MAC 时钟域之间实现数据传输。MTL 层具有两个 32 位宽的数据路径：发送路径和接收路径。

- 发送路径

  应用或内部 DMA 将从应用或系统存储器中读取的以太网数据包推入到 Tx FIFO。当达到队列阈值（阈值模式）或完整数据包处于队列中（存储转发模式）时，相应数据包将弹出并传输到 MAC。发送 EOP 后，将从 MAC 获取发送状态并传送回应用或内部 DMA。Tx 队列大小为 2048 个字节。

- 接收路径

  MTL Rx 模块从 MAC 接收数据包，并将其推入 Rx 队列。当队列超出配置的接收阈值（*Rx 队列工作模式寄存器 (ETH_MTLRxQOMR)* 中定义的 RTC 位 [1:0]）时，或者在接收到完整数据包后，会将队列的状态（填充级别）报告给应用或 DMA。MTL 还指示队列填充级别，以便 DMA 可向主接口发起预配置的突发传输。Rx 队列大小为 2048 个字节。

58.4.3 MAC

MAC 负责以太网协议处理。在发送模式下，它从 MTL 接收数据，然后再将其传送到 PHY 接口。在接收模式下，MAC 从 PHY 接口接收数据，然后再将其传送到 MTL 模块的 Rx FIFO。

本节简要介绍发送和接收序列。

MAC 发送

发送序列如下：

1. 当 MTL 应用在 SOP（数据包起始）信号置为有效的情况下推入数据时，会启动发送。
2. 检测到 SOP 信号时，MAC 接受数据并开始向 MII 发送。
3. 当 EOP（数据包结束）传送到 MAC 时，MAC 执行以下操作之一：
   - MAC 完成常规发送，并向 MTL 提供发送状态。
   - 如果在发送期间发生常规冲突（半双工模式下），则 MAC 会将发送状态提供给 MTL，同时重试位置 1。 MAC 会提供重试请求，直到下列其中一个条件为真：

     数据包已成功发送

     达到最大重试请求次数。在这种情况下，MAC 中止数据包发送，并处于过度冲突发送状态。MAC 接受并丢弃所有后续数据，直到接收到下一个 SOP。检测到来自 MAC 的重试请求（在状态中）时，MTL 模块应从 SOP 重新发送同一数据包。
   - 如果发生以下任一情况，则 MAC 会中止数据包发送：

     无载波（半双工模式）

     载波丢失（半双工模式）

     过度延迟（半双工模式）

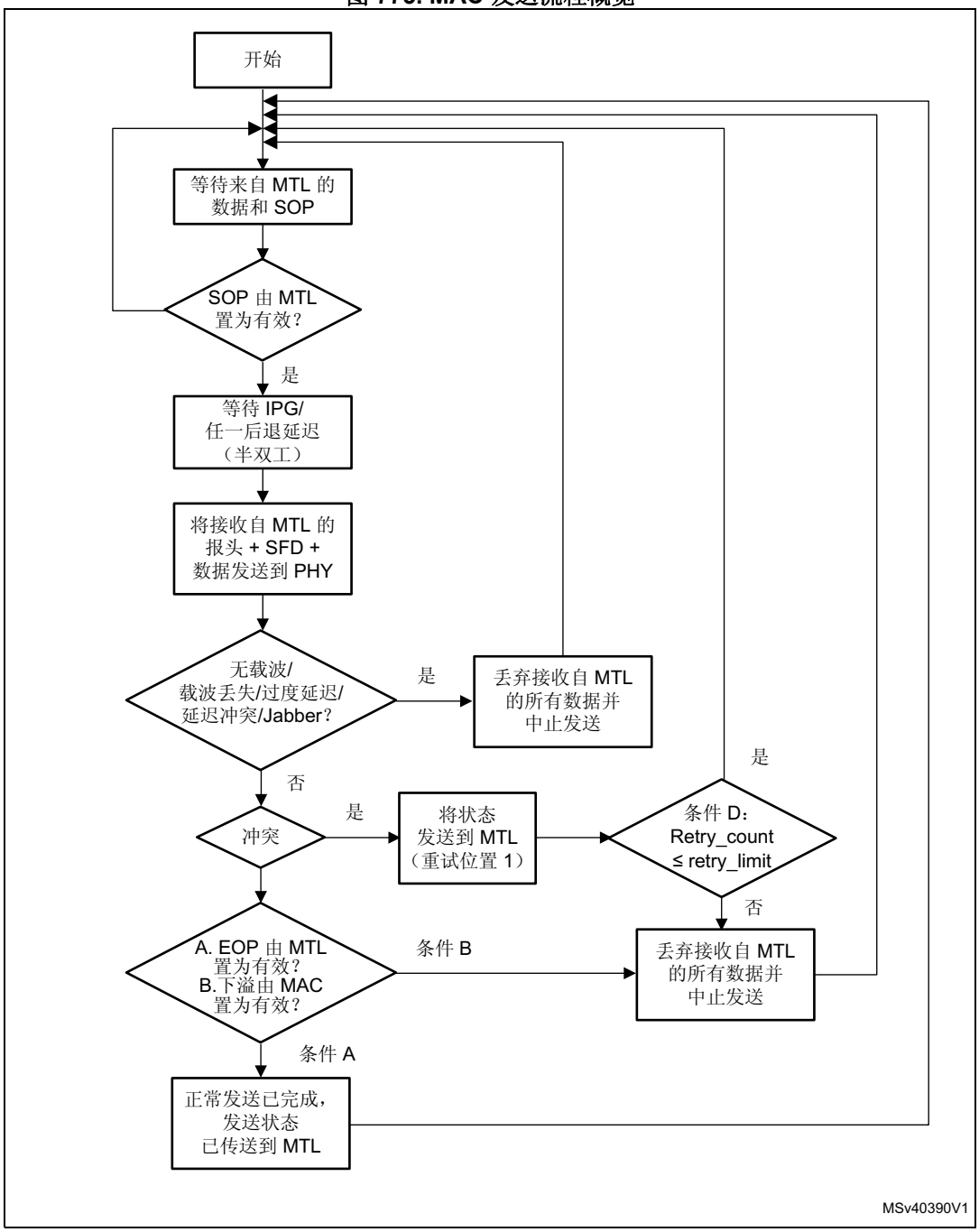
     延迟冲突（半双工模式）

     Jabber

     MAC 接受并丢弃所有后续数据，直到接收到下一个 SOP。
4. 如果发送期间 MTL 无法连续提供数据，MAC 将发出下溢状态。MAC 接受并丢弃所有后续数据，直到接收到下一个 SOP。
5. 从 MTL 正常传输数据包期间，如果 MAC 在未获得前一数据包的 EOP 的情况下接收到 SOP，则会忽略该 SOP 并将新的数据包视为前一数据包的延续。

*图 773：MAC 发送流程概览*所示为 MAC 发送过程流程。

图 773. MAC 发送流程概览



MAC 接收

当 MAC 在 MII 上检测到 SFD 时，将启动接收操作。MAC 将去除报头和 SFD，然后再继续处理数据包。检查报头字段以进行过滤，FCS 字段用于验证数据包的 CRC。接收的数据包存储在浅缓冲区中，直到执行地址过滤。如果数据包未通过地址过滤器，则会在 MAC 中将其丢弃。

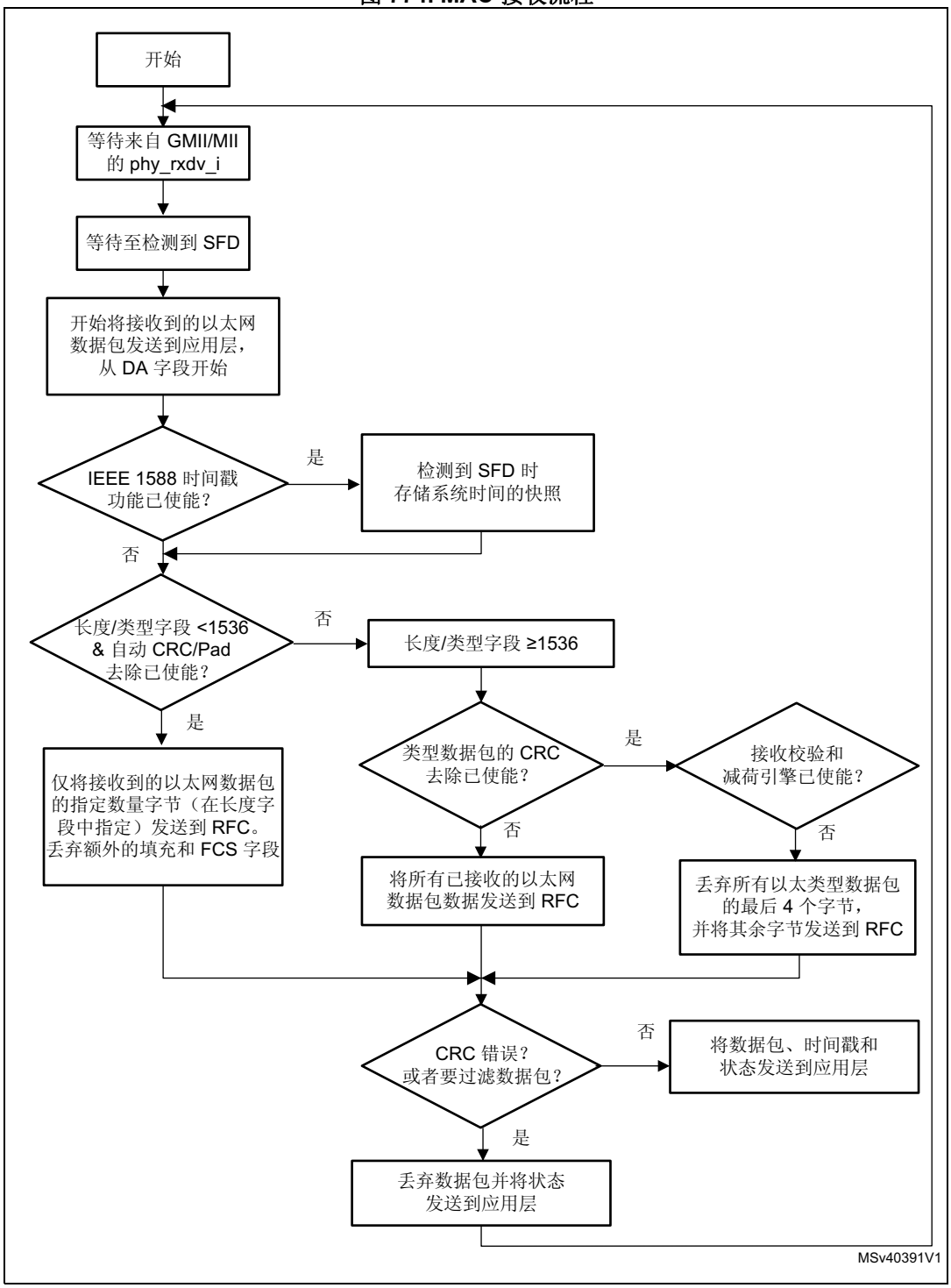
接收序列如下：

1. MII 的接收数据有效信号 (RxDV) 激活时，接收状态机 (RSM) 开始寻找 SFD 字段（MII 模式下的 0xD 半字节）。

   状态机会丢弃接收到的数据包，直至检测到 SFD。
2. 检测到 SFD 时，状态机开始向 RPC 模块发送以太网数据包的数据，从 SFD 之后的第一个字节（目标地址）开始发送。
3. 如果使能了 IEEE 1588 时间戳功能，则在 MII 上检测到任何数据包的 SFD 时，MAC 都将获取系统时间的快照。如果在 MAC 过滤期间该数据包未被丢弃，则时间戳将传递给应用程序。MAC 将接收到的半字节数据转换成字节，并将有效数据包数据转发到 RFC 模块。
4. 接收状态机对正在接收的以太网数据包的长度/类型字段进行解码。

   如果长度/类型字段小于 1,536 并且为 MAC 编程了自动去除 CRC/Pad（*工作模式配置寄存器 (ETH_MACCR)* 的位 20），则状态机将发送数据包数据（数据量不超过长度/类型字段中指定的数量），然后开始丢弃字节（包括 FCS 字段）。状态机对长度/类型字段进行解码并检查长度解析。
5. 当长度/类型字段大于或等于 1,536 时，如果尚未使能 *工作模式配置寄存器 (ETH_MACCR)* 位 21 中的类型数据包的 CRC 去除，则 RPE 模块会将所有接收到的以太网数据包数据发送到 RFC 模块。不过，如果已使能类型数据包的 CRC 去除，但未使能接收校验和减荷引擎，则 MAC 将去除并丢弃所有以太类型数据包的最后 4 个字节，然后将数据包转发到应用程序。
6. 默认情况下，将 MAC 编程为使能看门狗定时器，即在 RPE 模块中会切断大于 2,048（如果使能巨型数据包，则为 10,240）字节的数据包（DA + SA + LT + 数据 + PAD + FCS）。此外，还可以使用可编程看门狗定时器（*看门狗超时寄存器 (ETH_MACWTR)* 的位 16）覆盖 2,048 或 10,240 字节的固定超时。可通过编程 *工作模式配置寄存器 (ETH_MACCR)* 的位 19 来禁止看门狗定时器。但是，即使禁止看门狗定时器，也仍将切断大于 32 KB 的数据包并给出看门狗超时状态。

图 774. MAC 接收流程



58.5 以太网功能说明：MAC

58.5.1 双 VLAN 处理

以太网外设支持双 VLAN（虚拟 LAN）标记功能，通过该功能 MAC 可处理多达两个 VLAN 标记（内部和外部）。

MAC 支持以下操作：

- 在发送路径中插入、替换或删除最多两个 VLAN 标记
- 基于接收路径中两个 VLAN 标记的任意一个进行数据包过滤和去除。在接收路径中去除和提供最多两个 VLAN 标记（作为接收状态的一部分）

发送路径

*表 492：Tx 路径中的双 VLAN 处理功能*介绍了发送端 MAC 所支持的功能。

表 492. Tx 路径中的双 VLAN 处理功能

特性	说明
支持 C-VLAN 和 S-VLAN 标记类型	内部或外部 VLAN 标记可以是 C-VLAN 和 S-VLAN 类型。VLAN 类型分别通过 *VLAN 包含寄存器 (ETH_MACVIR)* 和*内部 VLAN 包含寄存器 (ETH_MACIVIR)* 的 CSVL 位指定。 以太网外设支持处理外部和内部 VLAN 标记的任何序列。但是，它不支持 C-VLAN S-VLAN 序列。 MAC 既不检查应用程序提供的数据包是否具有有效的 VLAN 标记类型序列，也不检查插入或替换操作是否会导致 VLAN 标记类型序列失效。因此，应用程序必须提供正确的 VLAN 标记类型序列，并对 MAC 进行编程以使发送的数据包中具有正确的 VLAN 标记类型序列。应用程序必须确保以下几点： – 使能外部 C-VLAN 标记插入时，内部标记不应为 S-VLAN。 – 使能内部 S-VLAN 标记插入时，外部标记不应为 C-VLAN。 – 用 C-VLAN 替换外部标记时，内部标记不应为 S-VLAN。 – 用 S-VLAN 替换内部标记时，外部标记不应为 C-VLAN。
VLAN 标记删除	可分别通过 *VLAN 包含寄存器 (ETH_MACVIR)* 或*内部 VLAN 包含寄存器 (ETH_MACIVIR)* 中的 VLC 字段为外部或内部标记使能 VLAN 标记删除。使能 VLAN 删除时，MAC 会删除相应位置上的标记。数据包只有一个标记时，会将该标记视为外部标记。如果使能了内部标记删除，并且数据包只有一个标记，则 MAC 不会删除该标记。
VLAN 标记插入或替换	可分别通过 *VLAN 包含寄存器 (ETH_MACVIR)* 或*内部 VLAN 包含寄存器 (ETH_MACIVIR)* 中的 VLC 字段为外部或内部标记使能 VLAN 标记插入或替换。使能 VLAN 标记插入或替换时，前一寄存器中的 VLTI 位用于确定是否应从寄存器或控制字中获取 VLAN 标记。

接收路径

*表 493：Rx 路径中的双 VLAN 处理功能*介绍了接收端 MAC 所支持的功能以及 *VLAN 标记寄存器 (ETH_MACVTR)* 中的相应位。

表 493. Rx 路径中的双 VLAN 处理功能

特性	说明
基于外部或内部 VLAN 标记的过滤	MAC 可基于外部或内部 VLAN 标记（通过 ERIVLT 位选择）对数据包进行过滤。
基于 C-VLAN 或 S-VLAN 标记的过滤	MAC 可基于 C-VLAN 或 S-VLAN 类型（通过 ERSVLM 位选择）对数据包进行过滤。
外部和内部 VLAN 标记去除	MAC 可基于 EVLS 和 EIVLS 位从接收到的帧中去除外部和内部 VLAN 标记。
Rx 状态中的 16 位外部和内部 VLAN 标记和类型	MAC 可分别基于 EVLRXS 和 EIVLRXS 位在 Rx 状态中提供 16 位外部和内部 VLAN 标记和类型。
禁止或跳过外部 VLAN 标记类型的检查	MAC 可基于 DOVLTC 位禁止或跳过外部 VLAN 标记类型的检查以匹配 C-VLAN 或 S-VLAN。

58.5.2 源地址和 VLAN 插入、替换或删除

源地址插入或替换

软件可使用 SA（源地址）插入或替换功能来指示 MAC 对 Tx 数据包执行以下操作：

- 在 SA 字段中插入 MAC 地址寄存器的内容
- 使用 MAC 地址寄存器的内容替换 SA 字段的内容

使能 SA 插入时，应用程序必须确保发送到 MAC 的数据包不包含 SA 字段。MAC 不检查发送数据包中是否存在 SA 字段，并且会在 SA 字段中插入 MAC 地址寄存器的内容。类似地，使能 SA 替换时，应用程序必须确保发送到 MAC 的数据包中存在 SA 字段。MAC 使用 MAC 地址寄存器的内容替换发送数据包中目标地址字段后的六个字节。

可以针对所有发送数据包或选择性数据包使能 SA 插入或替换功能：

- 针对所有数据包使能 SA 插入或替换。

  要针对所有数据包使能此功能，请编程*工作模式配置寄存器 (ETH_MACCR)* 的 SARC 字段。
- 针对选择性数据包使能 SA 插入或替换。

  要针对选择性数据包使能此功能，请按照如下所述对数据包第一个发送描述符的 SA 插入控制字段（发送描述符字 3/TDES3 的位 [25:23]，请参见*第 58.10.3 节：发送描述符*）进行编程。TDES3 的位 25 置 1 时，SA 插入控制字段指示基于 MAC 地址 1 寄存器进行的插入或替换。TDES3 的位 25 复位时，则指示基于 MAC 地址 0 寄存器进行的插入或替换。

如果未使能 MAC 地址 1 寄存器，则使用 MAC 地址 0 寄存器插入或替换 SA 插入控制字段最高有效位的值。

VLAN 插入、替换或删除

软件可使用 VLAN 插入、替换或删除功能来指示 MAC 对 Tx 数据包执行以下操作：

- 删除 VLAN 类型和 VLAN 标记字段
- 插入或替换 VLAN 类型和 VLAN 标记字段

  基于 *VLAN 包含寄存器 (ETH_MACVIR)* 中 VLTI 位的设置执行插入或替换，如*表 494：基于 VLTI 位进行 VLAN 插入或替换*所述。

表 494. 基于 VLTI 位进行 VLAN 插入或替换

条件	说明
VLTI 位置 1	MAC 插入或替换以下字段： VLAN 类型字段（C-VLAN 或 S-VLAN，由 *VLAN 包含寄存器 (ETH_MACVIR)* 的 CSVL 位指示） VLAN 标记字段（使用数据包发送上下文描述符的 VT 字段）
VLTI 位复位	MAC 插入或替换以下字段： VLAN 类型字段（C-VLAN 或 S-VLAN，由 *VLAN 包含寄存器 (ETH_MACVIR)* 的 CSVL 位指示） VLAN 标记字段（使用 *VLAN 包含寄存器 (ETH_MACVIR)* 的 VLT 字段）

使能 VLAN 替换或删除时，MAC 会检查发送数据包中的 DA 和 SA 字段之后是否存在 VLAN 类型字段（0x8100 或 0x88A8）。如果在 DA 和 SA 字段之后的两个字节中未检测到 VLAN 类型字段，则不会进行替换或删除操作。不过，使能 VLAN 插入时，MAC 不会检查发送数据包中是否存在 VLAN 类型字段，只会插入 VLAN 类型和 VLAN 标记字段。

可针对所有 Tx 数据包或选择性数据包使能 VLAN 插入、替换或删除功能：

- 要针对所有数据包使能此功能，请编程 *VLAN 包含寄存器 (ETH_MACVIR)* 的 VLC 和 VLP 字段。
- 要针对选择性数据包使能此功能，请对 TDES2 正常描述符的 VTIR 字段进行编程（请参见*表 514：TDES2 正常描述符（读取格式）*）。

  此外，必须根据配置，将 *VLAN 包含寄存器 (ETH_MACVIR)*（针对外部 VLAN）和*内部 VLAN 包含寄存器 (ETH_MACIVIR)*（针对内部 VLAN）中的 VLP（VLAN 优先级控制）位复位，以使 MAC 从主机获取控制输入。

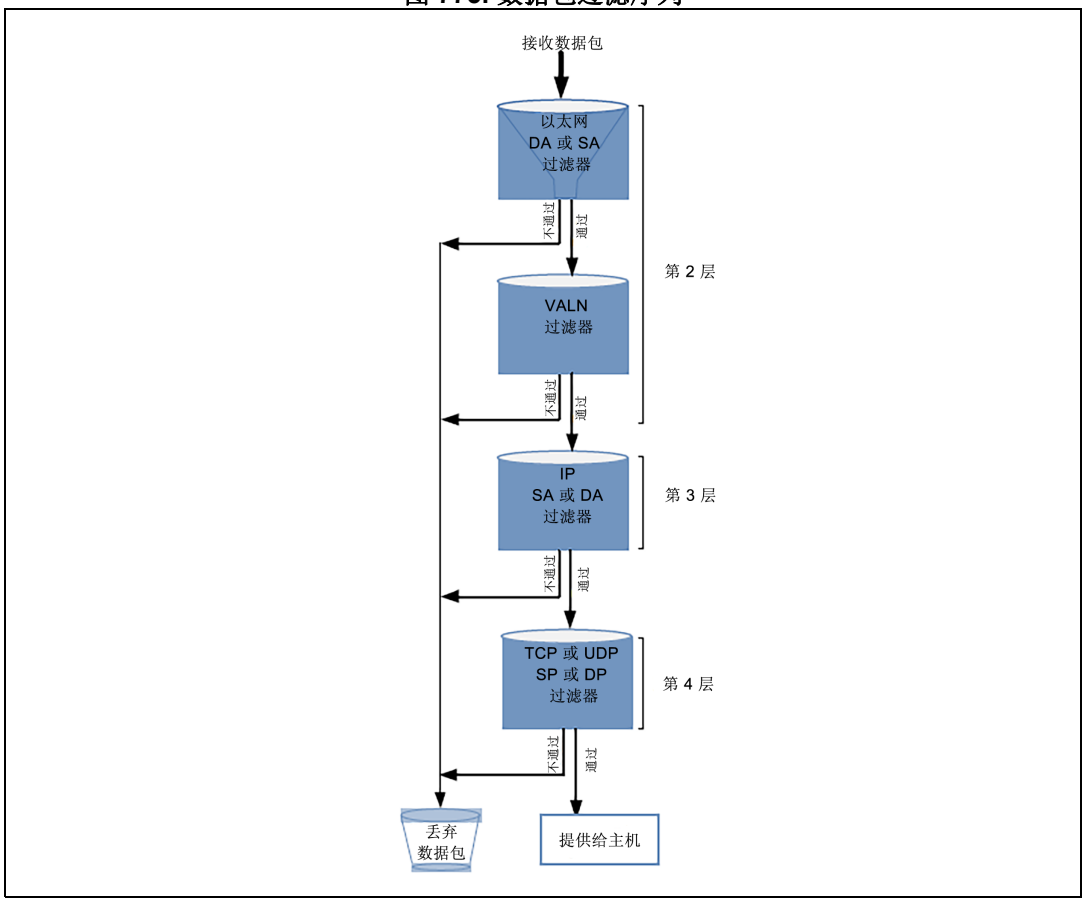
58.5.3 数据包过滤

MAC 支持对 Rx 数据包进行以下类型的过滤：

- *MAC 源地址或目标地址过滤*：地址过滤模块 (AFM) 检查每个传入数据包的源地址和目标地址字段。
- *VLAN 过滤*：MAC 支持基于 VLAN 标记的过滤和 VLAN 散列过滤。
- *第 3 层和第 4 层过滤*：第 3 层过滤是指 IP 源地址和目标地址过滤。第 4 层过滤是指源端口和目标端口过滤。

这三种过滤类型可级联。*图 775* 所示为 Rx 数据包的过滤序列。

图 775. 数据包过滤序列



MAC 源地址或目标地址过滤

MAC 地址过滤模块检查每个传入数据包的源地址 (SA) 和目标地址 (DA) 字段。

单播目标地址过滤

MAC 支持 4 个用于单播完美过滤的 MAC 地址。如果选择完美过滤（复位*数据包过滤控制寄存器 (ETH_MACPFR)* 的 HUC 位），MAC 会将接收的单播地址的所有 48 位与编程的 MAC 地址进行比较来确定是否匹配。默认情况下，始终使能 MacAddr0。

MacAddr1 到 MacAddr3 地址则通过单独的使能位进行选择。与接收的相应 DA 字节进行比较时，可以将寄存器中相应的屏蔽字节控制位置 1 来屏蔽各个字节。这样可以实现对 DA 的组地址过滤。

在散列过滤模式（HUC 位置 1）下，MAC 将使用 64 位散列表执行对单播地址的不完美过滤。对于散列过滤，MAC 将使用接收的目标地址的 6 个高 CRC 位来索引散列表的内容。值为 00000 时，选择所选寄存器的位 0；值为 11111 时，选择散列表寄存器的位 63。如果相应位（由 6 位 CRC 指示）已置 1，将认为单播数据包已通过散列过滤，否则认为数据包未能通过散列过滤。

多播目标地址过滤

要将 MAC 编程为让所有多播数据包通过，请将*数据包过滤控制寄存器 (ETH_MACPFR)* 的 PM 位置 1。如果 PM 位复位，MAC 将根据*数据包过滤控制寄存器 (ETH_MACPFR)* 的 HMC 位对多播地址执行过滤。

在完美过滤模式下，将多播地址与编程的 MAC 目标地址寄存器 (1–31) 进行比较。组地址过滤也受到支持。

在散列过滤模式下，MAC 将使用 64 位散列表执行不完美过滤。MAC 使用接收的多播地址的 6 个高 CRC 位来索引散列表的内容。值为 000000 时，选择所选寄存器的位 0；值为 111111 时，选择散列表寄存器的位 63。如果相应的位置 1，则认为多播数据包已通过散列过滤。否则，认为数据包未通过散列过滤。

散列或完美地址过滤

要将 DA 过滤器配置为让 DA 与散列过滤器或完美过滤器匹配的数据包通过，请将*数据包过滤控制寄存器 (ETH_MACPFR)* 中的 HPF 位和相应的 HUC 或 HMC 位置 1。这对于单播数据包和多播数据包均适用。如果 HPF 位复位，则只有一种过滤器（散列或完美）应用于接收数据包。

广播地址过滤

默认情况下，MAC 不过滤任何广播数据包。要将 MAC 编程为拒绝所有广播数据包，请将*数据包过滤控制寄存器 (ETH_MACPFR)* 的 DBF 位置 1。

单播源地址过滤

MAC 可以根据所接收数据包的源地址字段来执行完美过滤。默认情况下，MAC 将 SA 字段与 SA 寄存器中编程的值进行比较。可通过将相应寄存器的位 30 置 1，来将 MAC 地址寄存器配置为使用 SA（而非 DA）进行比较。

MAC 还支持带 SA 的组地址过滤。您可以通过屏蔽地址的一个或多个字节来过滤一组地址。如果*数据包过滤控制寄存器 (ETH_MACPFR)* 中的 SAF 位置 1，则 MAC 会丢弃未通过 SA 过滤的数据包。否则，SA 过滤的结果将通过接收状态字中的状态位给出（参见*表 496*）。SAF 位置 1 时，对 SA 过滤和 DA 过滤结果进行与运算，以决定是否需要转发数据包。这意味着只要其中一种过滤失败，就会丢弃数据包。只有在数据包依次通过两种过滤时，才能将数据包转发到应用程序。

反向过滤

对于 DA 和 SA 过滤，可通过将*数据包过滤控制寄存器 (ETH_MACPFR)* 的 DAIF 和 SAIF 位置 1 来反转最终输出端的过滤匹配结果。DAIF 位同时适用于单播和多播 DA 数据包。在此模式下，将反转单播或多播目标地址过滤的结果。类似地，当 SAIF 位置 1 时，将反转单播 SA 过滤的结果。

表 495 和*表 496* 按照所接收数据包的类型总结了 DA 和 SA 过滤。

表 495. 目标地址过滤

数据包类型	PR	HPF	HUC	DAIF	HMC	PM	DBF	DA 过滤操作
广播	1	X	X	X	X	X	X	通过
	0	X	X	X	X	X	0	通过
	0	X	X	X	X	X	1	不通过
单播	1	X	X	X	X	X	X	通过所有数据包
	0	X	0	0	X	X	X	完美/组过滤匹配时通过
	0	X	0	1	X	X	X	完美/组过滤匹配时不通过
	0	0	1	0	X	X	X	散列过滤匹配时通过
	0	0	1	1	X	X	X	散列过滤匹配时不通过
	0	1	1	0	X	X	X	散列或完美/ 组过滤匹配时通过
	0	1	1	1	X	X	X	散列或完美/ 组过滤匹配时不通过
多播	1	X	X	X	X	X	X	通过所有数据包
	X	X	X	X	X	1	X	通过所有数据包
	0	X	X	0	0	0	X	完美/组过滤匹配时通过， 并在 PCF = 0x 时丢弃 暂停数据包
	0	0	X	0	1	0	X	散列过滤匹配时通过，并在 PCF = 0x 时丢弃暂停数据包
	0	1	X	0	1	0	X	散列或完美/组过滤匹配时 通过，并在 PCF = 0x 时丢弃 暂停数据包
	0	X	X	1	0	0	X	完美/组过滤匹配时不通过， 并在 PCF = 0x 时丢弃 暂停数据包
	0	0	X	1	1	0	X	散列过滤匹配时不通过，并在 PCF = 0x 时丢弃暂停数据包
	0	1	X	1	1	0	X	散列或完美/组过滤匹配时 不通过，并在 PCF = 0x 时 丢弃暂停数据包

表 496. 源地址过滤

数据包类型	PR	SAIF	SAF	SA 过滤操作
单播	1	X	X	通过所有数据包
	0	0	0	完美或组过滤匹配时为通过状态，但不丢弃未通过的数据包
	0	1	0	完美或组过滤匹配时为不通过状态，但不丢弃数据包
	0	0	1	完美或组过滤匹配时通过，并丢弃不通过的数据包
	0	1	1	完美或组过滤匹配时不通过，并丢弃不通过的数据包

VLAN 过滤

MAC 支持完美和散列 VLAN 过滤。有关详细的编程步骤，请参见*第 58.9.11 节：关于在接收端执行 VLAN 过滤的编程指南*。

VLAN 标记完美过滤

在 VLAN 标记完美过滤中，MAC 比较所接收数据包的 VLAN 标记，并向应用程序提供 VLAN 数据包状态。根据编程的模式，MAC 会选择比较所接收 VLAN 标记的低 12 位或全部 16 位，以确定完美匹配。

如果使能 VLAN 标记完美过滤，则 MAC 会转发 VLAN 标记的数据包以及 VLAN 标记匹配状态，并丢弃不匹配的 VLAN 数据包。还可以通过将 *VLAN 标记寄存器 (ETH_MACVTR)* 的 VTIM 位置 1 来使能 VLAN 数据包的反向匹配。此外，可通过将 ETH_MACVTR 寄存器的 ESVL 位置 1，来使能 S-VLAN 标记数据包与默认 C-VLAN 标记数据包的匹配操作。VLAN 数据包状态位（RDES0 的位 10）指示匹配数据包的 VLAN 标记匹配状态。

注： *源地址或目标地址（如果已使能）的优先级高于 VLAN 标记过滤器。这意味着无论 VLAN 标记过滤结果如何，未通过源地址或目标地址过滤的数据包都会被丢弃。默认情况下，所有配置均支持基于 VLAN 标记的完美过滤。*

VLAN 标记散列过滤

MAC 使用 16 位散列表执行 VLAN 标记散列过滤。MAC 根据 *VLAN 标记寄存器 (ETH_MACVTR)* 的 VTHM 位执行 VLAN 散列匹配。如果 VTHM 位置 1，则使用 VLAN 标记的 CRC-32 的四个最高有效位来索引 VLAN 散列表寄存器的内容。根据索引找到 VLAN 散列表寄存器中的值为 1，表示数据包的 VLAN 标记匹配，应转发数据包。值 0 则表示应丢弃 VLAN 标记的数据包。

注： *根据 ETH_MACVTR 寄存器中的 ETV 位，考虑使用 VLAN 标记的 16 位或 12 位进行 CRC-32 计算。*

ETV 位复位时，VLAN 标记的 CRC-32 的四个最高有效位被反转，并用于索引 VLAN 散列表寄存器 (ETH_MACVHTR) 的内容。

ETV 位置 1 时，VLAN 标记的 CRC-32 的四个最高有效位直接用于索引 VLAN 标记寄存器 (ETH_MACVTR) 的内容。

MAC 还支持 VLAN 数据包的反向匹配。在反向匹配模式下，当数据包的 VLAN 标记与完美或散列过滤匹配时，应丢弃该数据包。如果使能了 VLAN 完美匹配和 VLAN 散列匹配，则在 VLAN 散列过滤或 VLAN 完美过滤匹配时，会将数据包视为匹配。如果反向匹配置 1，则只有在完美和散列过滤指示不匹配时才转发数据包。

表 497 列出了 VLAN 匹配结果和最终 VLAN 匹配状态的各种可能情况。*数据包过滤控制寄存器 (ETH_MACPFR)* 的 RA 位置 1 时，即表示接收到所有数据包，并在 *RDES2 正常描述符（回写格式）*的 VF 位中指示 VLAN 匹配状态。RA 位未置 1 且 *数据包过滤控制寄存器 (ETH_MACPFR)* 的 VTFE 位置 1 时，如果最终 VLAN 匹配状态为未通过，则会丢弃数据包。在 *表 497* 中，值 *X* 表示该列可为任一值。

在 *VLAN 标记寄存器 (ETH_MACVTR)* 的 VL 字段中将 VLAN VID 编程为 0 时，所有 VLAN 标记的数据包均被视为完美匹配，但 VLAN 散列匹配的状态取决于 ETH_MACVTR 寄存器中的 VTHM 和 VTIM 位。

表 497. VLAN 匹配状态(1)

VID	VLAN 完美过滤匹配结果	VTHM 位	VLAN 散列过滤匹配结果	VTIM 位	最终 VLAN 匹配状态
VID = 0	通过	0	X	X	通过
	通过	1	X	0	通过
	通过	1	不通过	1	通过
	通过	1	通过	1	不通过
VID! = 0	通过	X	X	0	通过
	不通过	0	X	0	不通过
	不通过	1	不通过	0	不通过
	不通过	1	通过	0	通过
	不通过	0	X	1	通过
	通过	X	X	1	不通过
	不通过	1	通过	1	不通过
	不通过	1	不通过	1	通过

1. 在此表中，X 表示任一值。

第 3 层和第 4 层过滤

MAC 支持基于第 3 层和第 4 层的数据包过滤。第 3 层过滤是指 IPv4 或 IPv6 数据包中的 IP 源地址或目标地址过滤，而第 4 层过滤则指 TCP 或 UDP 中的源端口号或目标端口号过滤。

第 3 层和第 4 层数据包过滤功能可自动使能接收端的 IPC 全校验和减荷引擎。对于第 3 层或第 4 层过滤操作，必须将*工作模式配置寄存器 (ETH_MACCR)* 的 IPC 位置 1，以使能 Rx 校验和减荷引擎。

使能第 3 层和第 4 层过滤后，将按以下方式过滤数据包：

- **匹配的数据包**

  MAC 将与所有已使能字段匹配的数据包连同状态一起转发给应用程序。只有在*工作模式配置寄存器 (ETH_MACCR)* 的 IPC 位置 1 且以下条件之一为真时，MAC 才会提供匹配的字段状态：

  – 所有已使能的第 3 层和第 4 层字段均匹配。

  – 至少有一个已使能的字段匹配且其他字段被绕过或禁止

    使能多个第 3 层和第 4 层过滤器时，任一过滤器匹配都将被视为匹配。如果多个过滤器匹配，则 MAC 会提供最低级过滤器的状态，其中过滤器 0 为最低级过滤器，过滤器 3 为最高级过滤器。例如，如果过滤器 0 和过滤器 1 均匹配，则 MAC 会提供过滤器 0 对应的状态。

    源地址或目标地址以及 VLAN 标记过滤器（如果已使能）的优先级高于第 3 层和第 4 层过滤器。这意味着无论第 3 层和第 4 层过滤结果如何，未通过源地址或目标地址或者 VLAN 标记过滤的数据包都会被丢弃。

- **不匹配的数据包**

  MAC 会丢弃与所有已使能字段均不匹配的数据包。可使用反向匹配功能阻止或丢弃特定的 TCP/UDP 数据包，并转发所有其他数据包。

- **非 TCP 或 UDP IP 数据包**

  默认情况下，所有非 TCP 或 UDP IP 数据包均从第 3 层和第 4 层过滤器绕过。可以将 MAC 编程为丢弃所有非 TCP over IP 或 UDP over IP 数据包。

第 3 层和第 4 层过滤器寄存器组

MAC 为基于第 3 层和第 4 层的数据包过滤实现了两组寄存器。在其中一个寄存器组中，存在一个控制寄存器（例如，ETH_MACL3L4C0R），用于控制数据包过滤。此外，还存在五个地址寄存器，用于编程要匹配的第 3 层和第 4 层字段，例如：

- *第 4 层地址过滤器 0 寄存器 (ETH_MACL4A0R)*
- *第 3 层地址 0 过滤器 0 寄存器 (ETH_MACL3A00R)*
- *第 3 层地址 1 过滤器 0 寄存器 (ETH_MACL3A10R)*
- *第 3 层地址 2 过滤器 0 寄存器 (ETH_MACL3A20)*
- *第 3 层地址 3 过滤器 0 寄存器 (ETH_MACL3A30)*

另一个独立的寄存器组包括：ETH_MACL3L4C1R、ETH_MACL4A01R、ETH_MACL4A11R、ETH_MACL4A21R 和 ETH_MACL4A31R。

第 3 层过滤

MAC 支持 IP 源地址和目标地址的完美匹配或反向匹配。此外，可以匹配完整的 IP 地址或屏蔽低位匹配，即比较地址中除指定低屏蔽位之外的所有位。

对于 IPv6 数据包过滤，可以使能寄存器组的后四个数据寄存器，以包含 128 位 IP 源地址或 IP 目标地址。IP 源地址或目标地址应按照 IPv6 规范中定义的顺序进行编程，即，所接收数据包中的 IP 源地址或目标地址的第一个字节位于寄存器的高字节中，后续寄存器中的顺序也同样如此。

对于 IPv4 数据包过滤，可以使能寄存器组的第二个和第三个数据寄存器，以包含 32 位 IP 源地址和 IP 目标地址。其余两个数据寄存器保留。IP 源地址或目标地址应按照 IPv4 规范中定义的顺序进行编程，即，所接收数据包中的 IP 源地址或目标地址的第一个字节位于相应寄存器的高字节中。

第 4 层过滤

MAC 支持 TCP 或 UDP 源端口号和目标端口号的完美匹配或反向匹配。不过，一次只能编程一种类型（TCP 或 UDP）。第一个数据寄存器包含 TCP 或 UDP 的 16 位源端口号和目标端口号，即，低 16 位用于源端口号，高 16 位用于目标端口号。

TCP 或 UDP 源端口号和目标端口号应按照 TCP 或 UDP 规范中定义的顺序进行编程，即，所接收数据包中的 TCP 或 UDP 源端口号和目标端口号的第一个字节位于寄存器的高字节中。

58.5.4 IEEE 1588 时间戳

IEEE 1588 标准定义了精密时间协议 (PTP)，此协议支持使用网络通信、局域计算和分布式对象等技术实现的测量和控制系统中的精密时钟同步。PTP 适用于利用支持多播消息传送的局域网（包括但不限于以太网）进行通信的系统。该协议可对非均匀系统进行同步，这类系统包含固有精度、分辨率和稳定性都不断变化的时钟。该协议支持亚微秒范围的系统级同步精度，并且需要极少的网络和本地时钟计算资源。

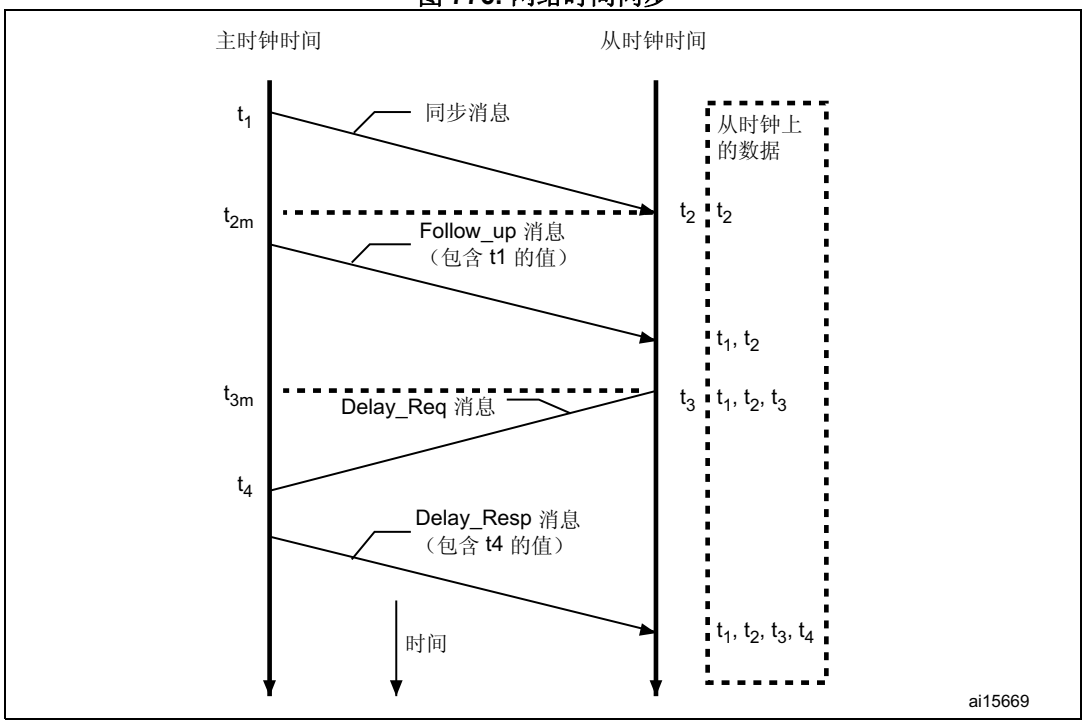
以太网外设支持 IEEE 1588-2002（版本 1）和 IEEE 1588-2008（版本 2）。IEEE 1588-2002 支持通过 UDP/IP 传送的 PTP。IEEE 1588 2008 支持通过以太网传送的 PTP。外设为这两种标准提供了可编程支持。它支持下列功能：

- 支持两种时间戳格式
- 可获取所有数据包或仅 PTP 型数据包的快照
- 可获取仅事件消息的快照
- 可基于时钟类型获取快照：普通、边界、端对端透明和点对点透明
- 可将节点选作普通和边界时钟的主节点或从节点
- 识别数据包中直接通过以太网发送的 PTP 消息类型、版本和 PTP 有效负载，并发送状态
- 支持数字或二进制格式的测量亚秒级时间

延迟请求响应机制

系统或网络归类为主节点和从节点，用于分配时序和时钟信息。*图 776*显示了 PTP 通过交换 PTP 消息将从节点同步到主节点的过程。

图 776. 网络时间同步



如*图 776* 所示，PTP 使用以下过程：

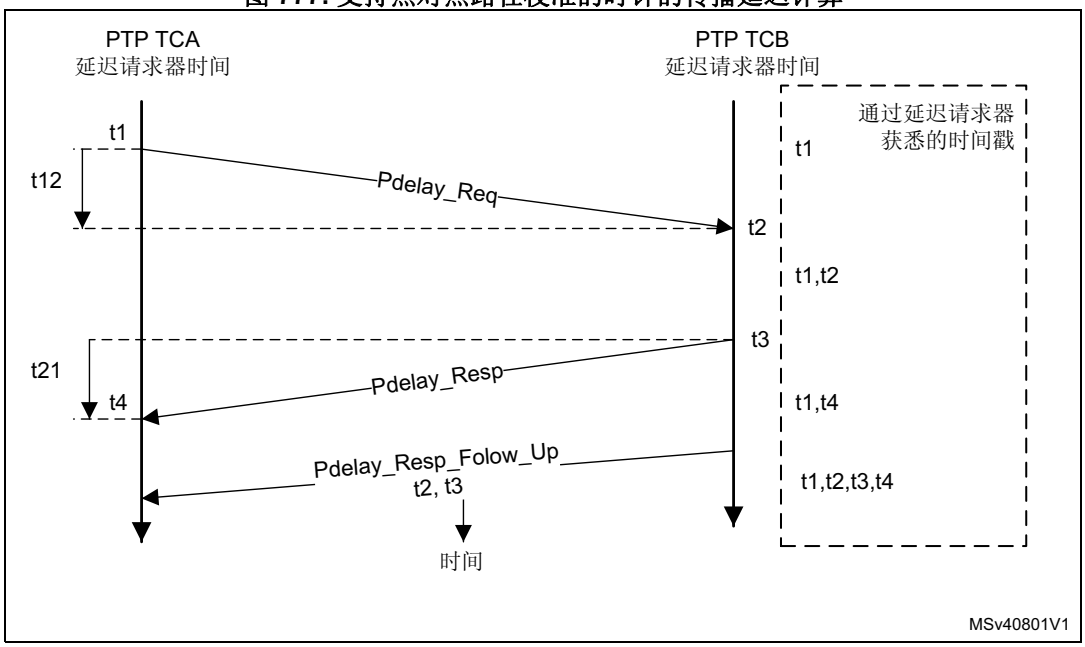
1. 主节点向其所有节点广播 PTP 同步消息。同步消息包含主节点的参考时间信息。该消息在时间 t_1 处离开主节点的系统。对于以太网端口，必须通过 MII 来捕获该时间。
2. 从节点接收同步消息并利用其参考时序捕获准确时间 t_2。
3. 主节点向从节点发送一个包含 t_1 信息的 Follow_up 消息以备后用。
4. 从节点向主节点发送一个 Delay_Req 消息，并标出该数据包离开 MII 接口的准确时间 t_3。
5. 主节点接收该消息，捕获消息进入其系统的准确时间 t_4。
6. 主节点将 Delay_Resp 消息中的 t_4 信息发送给从节点。
7. 从节点使用 t_1、t_2、t_3 和 t_4 这四个值来同步其本地参考时序与主节点的参考时序。

大多数 PTP 都在 UDP 层之上通过软件实现。不过，要通过 MII 接口来捕获特定 PTP 包进入或离开以太网端口时的准确时间，需要硬件支持。必须捕获该时序信息并将其返回给软件，以实现高精度的 PTP。

点对点 PTP 透明时钟 (P2P TC) 消息支持

除同步、延迟请求、后续消息和延迟响应消息外，IEEE 1588-2008 标准还支持点对点 PTP (Pdelay) 消息。*图 777*所示为支持点对点路径校准的时钟的传播延迟计算方法。

图 777. 支持点对点路径校准的时钟的传播延迟计算



如*图 777* 所示，传播延迟的计算过程如下：

1. 端口 1 发出 Pdelay_Req 消息，并生成 Pdelay_Req 消息的时间戳 (t_1)。
2. 端口 2 接收 Pdelay_Req 消息，并生成该消息的时间戳 (t_2)。
3. 端口 2 返回 Pdelay_Resp 消息，并生成该消息的时间戳 (t_3)。

   为了最大限度减少因两个端口之间的频偏而引起的误差，端口 2 会在接收到 Pdelay_Req 消息之后尽快返回 Pdelay_Resp 消息。端口 2 返回以下任意一项：

   – Pdelay_Resp 消息中的时间戳 t_3 和 t_2 之差
   – Pdelay_Resp_Follow_Up 消息中的时间戳 t_3 和 t_2 之差
   – Pdelay_Resp 和 Pdelay_Resp_Follow_Up 消息中分别对应的时间戳 t_3 和 t_2

   端口 1 在接收到 Pdelay_Resp 消息时生成时间戳 (t_4)。

   端口 1 使用全部四个时间戳来计算平均链路延迟。

时钟类型

MAC 支持 IEEE 1588-2008 规范中定义的以下时钟类型：

- **普通时钟**

  域的普通时钟支持协议的单一副本。它具有一个 PTP 状态和一个物理端口。在典型工业自动化应用中，普通时钟与传感器或致动器等应用设备相关联。在电信应用中，普通时钟可与时序划分设备相关联。

  普通时钟可以是主时钟或从时钟。它支持下列功能：

  – 发送和接收 PTP 消息。时间戳快照可以按*时间戳控制寄存器 (ETH_MACTSCR)* 中所述进行控制。

  – 保持数据集，例如时间戳值。

  下表显示了可针对主节点和从节点在接收端拍摄时间戳快照时的消息。

表 498. 普通时钟：拍摄快照的 PTP 消息

主	从
Delay_Req	SYNC

  对于普通时钟，可以拍摄以下 PTP 消息类型之一的快照：版本 1 或版本 2。不能同时拍摄两种 PTP 消息类型的快照。可通过在*时间戳控制寄存器 (ETH_MACTSCR)* 中将 TSVER2ENA 位置 1 并选择快照模式来拍摄快照。

- **边界时钟**

  通常，边界时钟具有多个可与网络进行通信的物理端口。在边界时钟的协议引擎中，与同步、主从层级和信号传输结束相关的消息 不会被转发。MAC 提供的 PTP 消息类型状态有助于识别消息的类型并采取适当的操作。

  除以下功能外，边界时钟与普通时钟类似：

  – 时钟数据集对边界时钟的所有端口通用。

  – 本地时钟对边界时钟的所有端口通用。

- **端对端透明时钟**

  端对端透明时钟支持从时钟和主时钟之间的端对端延迟测量机制。端对端透明时钟转发所有消息，例如普通网桥、路由器或中继器。PTP 数据包的停留时间是指 PTP 数据包从入站端口传输到出站端口所用的时间。

  端对端透明时钟内的 SYNC 数据包的停留时间于发送前在相关 Follow_Up PTP 数据包的校准字段中进行更新。类似地，端对端透明时钟内的 Delay_Req 数据包的停留时间于发送前在相关 Delay_Resp PTP 数据包的校准字段中进行更新。因此，仅需在入站端口和出站端口为相关寄存器中所述的消息拍摄快照即可。端对端透明时钟内的 SYNC 数据包的停留时间于发送前在相关 Follow_Up PTP 数据包的校准字段中进行更新。类似地，端对端透明时钟内的 Delay_Req 数据包的停留时间于发送前在相关 Delay_Resp PTP 数据包的校准字段中进行更新。因此，仅需在入站端口和出站端口为*时间戳控制寄存器 (ETH_MACTSCR)* 中所述的消息拍摄快照即可。可通过将*时间戳控制寄存器 (ETH_MACTSCR)* 中的 SNAPTYPSEL 位设为 10 来拍摄快照。可通过将*时间戳控制寄存器 (ETH_MACTSCR)* 中的 SNAPTYPSEL 位设为 10 来拍摄快照。

表 499. 端对端透明时钟：拍摄快照的 PTP 消息

PTP 消息
SYNC
Delay_Req

- **点对点透明时钟**

  点对点透明时钟对 PTP 时序消息的更正和处理方式与端对端透明时钟不同。除此之外，两类时钟基本相同。

  在点对点透明时钟中，链路延迟的计算基于与链路点的 Pdelay_Req、Pdelay_Resp 和 Pdelay_Resp_Follow_Up 消息的交换。Pdelay_Req 和相关 Pdelay_Resp 数据包的停留时间相加并被插入到相关 Pdely_Resp_Followup 数据包的校准字段中。因此，添加了针对 Pdelay 相关事件消息拍摄快照的支持，如*时间戳控制寄存器 (ETH_MACTSCR)* 所示。

表 500. 点对点透明时钟：拍摄快照的 PTP 消息

PTP 消息
SYNC
Pdelay_Req
Pdelay_Resp

可通过将*时间戳控制寄存器 (ETH_MACTSCR)* 中的 SNAPTYPESEL 位设为 11 来拍摄快照。

参考时序源

为获取时间快照，MAC 提供了采用 80 位格式的内部参考时间，如 IEEE 1588-2008 规范中定义。

MAC 时钟输入用于生成参考时间（亦称作系统时间）以及捕获时间戳。时间戳具有以下字段：

- UInteger48 secondsField
- secondsField 是以秒为单位的时间戳的整数部分。其宽度为 48 位。例如，2.000000001 秒表示为 secondsField = 0x0000_0000_0002。
- UInteger32 nanosecondsField
- nanosecondsField 是以纳秒为单位的时间戳的小数部分。例如，2.000000001 纳秒表示为 nanoSeconds = 0x0000_0001。

  nanosecondsField 支持以下两种模式：

  – **数字翻转模式**：在该模式下，纳秒字段中的最大值为 0x3B9A_C9FF，即 (10e9-1) 纳秒。

  – **二进制翻转模式**：在该模式下，纳秒字段会翻转，并在值 0x7FFF_FFFF 之后使秒字段递增。精度约为 0.466 ns/位。

  可通过*时间戳控制寄存器 (ETH_MACTSCR)* 中的 TSCTRLSSR 位来设置这些模式。

参考时序可通过每秒脉冲数输出信号（请参见*每秒脉冲数输出*）进行访问，或者通过在专用 FIFO 寄存器（*基于外部事件的辅助快照*）中存储多达 4 个快照来进行访问。

每秒脉冲数输出

MAC 支持固定的或灵活的每秒脉冲数输出信号 (ptp_pps_o)

- 每秒脉冲数固定的输出

  在该模式下，只能通过将 *PPS 控制寄存器 (ETH_MACPPSCR)* 中的 PPSCTRL0 字段置 1 来更改 PPS 输出的频率。

- 每秒脉冲数固定的输出

  在该模式下，可以灵活地编程在 ptp_pps_o 输出端生成的脉冲的起始时间或停止时间、宽度和间隔。

  – 起始时间和停止时间通过 *PPS 目标时间秒寄存器 (ETH_MACPPSTTSR)* 和 *PPS 目标时间纳秒寄存器 (ETH_MACPPSTTNR)* 寄存器编程。

  – PPS 宽度和间隔通过相应寄存器 *PPS 宽度寄存器 (ETH_MACPPSWR)* 和 *PPS 间隔寄存器 (ETH_MACPPSIR)* 基于系统时间的粒度（亚秒增量值的单位数）进行编程。

有关如何配置灵活脉冲输出的更多详细信息，请参见 *第 58.9.9 节：关于每秒脉冲数 (PPS) 灵活的输出的编程指南*。

注： *为确保正确 PPS 信号输出，建议为起始或停止时间编程高级系统时间。如果应用程序编程的起始或停止时间已过时，则 MAC 会将指示编程错误的错误状态位置 1。MAC 还会产生“达到目标时间”中断事件（如果已使能）。只有在相应起始或停止时间尚未过时的情况下，应用程序才能取消起始或停止请求。如果时间已过时，则取消命令失效。*

基于外部事件的辅助快照

用户可使用辅助快照功能来基于外部事件存储系统时间的快照。该事件被视为 ptp_aux_ts_trig_i 边带信号的上升沿。

最多可配置四个辅助快照输入，因此最多可存储 4 个快照。FIFO 可通过以下寄存器进行访问：*辅助时间戳秒寄存器 (ETH_MACATSSR)* 和 *辅助时间戳纳秒寄存器 (ETH_MACATSNR)*。

针对任意输入拍摄的快照均存储在一个公共 FIFO 中；只保留 64 位。应用程序可读取 *时间戳状态寄存器 (ETH_MACTSSR)*，以了解可在此 FIFO 顶部读取哪个输入的时间戳。

存储快照时，MAC 会向应用程序发送中断进行指示。快照的值通过 FIFO 寄存器访问进行读取。如果 FIFO 已满，且用于拍摄快照的外部触发信号置为有效，则 *时间戳状态寄存器 (ETH_MACTSSR)* 中的快照触发未命中状态 (ATSSTM) 会置 1。这表示，时间戳的最新辅助快照未存储在 FIFO 中。在 FIFO 已满时未向其中写入最新快照。

应用程序从 FIFO 中读取 64 位时间戳时，将存在可用于存储下一个快照的空间。可通过将 *辅助控制寄存器 (ETH_MACACR)* 中的 ATSFC 位置 1 来清空 FIFO。FIFO 中存在多个快照时，ETH_MACTSSR 寄存器的位 [27:25] 中会指示相应的计数。

系统时间寄存器模块

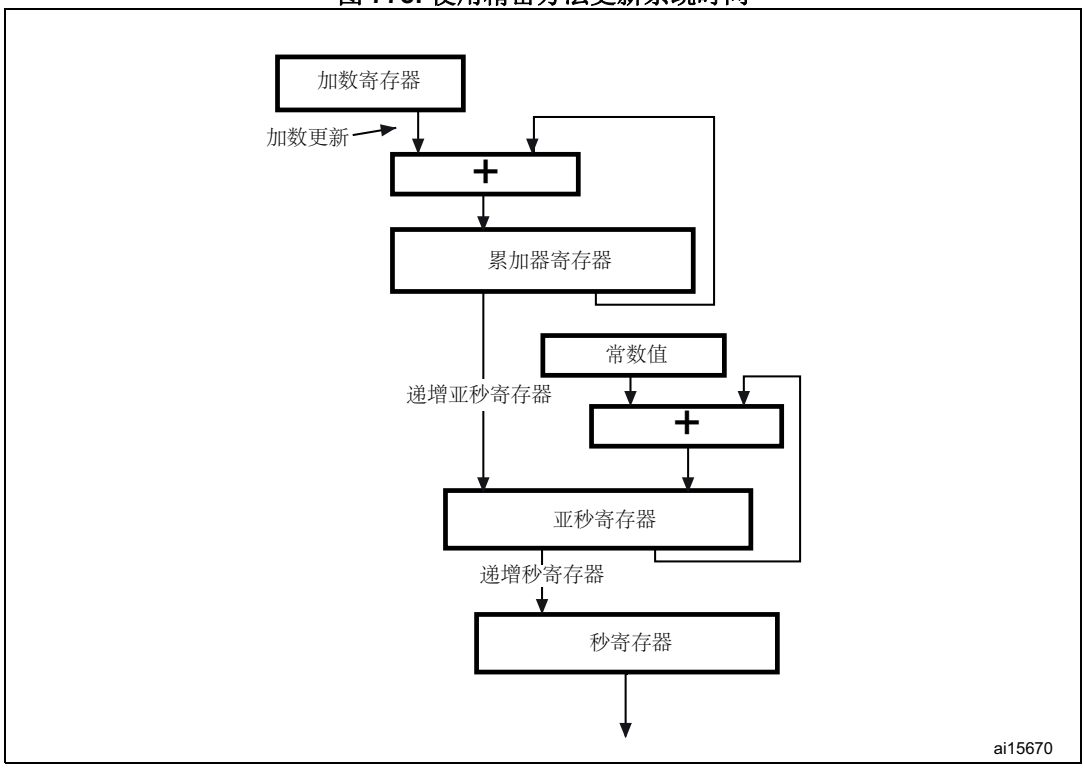
使用 PTP 输入参考时钟 HCLK 更新 64 位 PTP 时间。该 PTP 时间可用作时钟源，以获取 MII 上发送或接收的以太网帧的快照（时间戳）。可使用粗略校准方法或精密校准方法对系统时间计数器进行初始化或校准。

使用粗略校准方法时，初始值或偏移值会写入时间戳更新寄存器。对于初始化，会将时间戳更新寄存器中的值写入系统时间计数器；对于系统时间校准，会将偏移值（时间戳更新寄存器）加到系统时间中或从系统时间中减去。

使用精密校准方法时，从时钟（参考时钟）频率相对于主时钟（如 IEEE 1588 规范中定义）的偏移会在一段时间内进行校准，而不像粗略校准方法中那样，在单个时钟周期内进行校准。校准时间越长，越有助于保持线性时间，并且不会导致各 PTP 同步消息间隔之间的参考时间发生剧烈变化（或者大型抖动）。在此方法中，会使用一个累加器对加数寄存器中的内容求和，如*图 778* 所示。累加器生成的算数进位将用作使系统时间计数器递增的脉冲。累加器和加数寄存器均为 32 位寄存器。累加器用作高精度频率乘法器或除法器。

该系统时间更新算法如*图 778* 所示。

图 778. 使用精密方法更新系统时间



系统时间更新逻辑需要使用 50 MHz 的时钟频率，以达到 20 ns 的精度。分频是指参考时钟频率与所需时钟频率的比率。例如，如果参考时钟 (HCLK) 的频率为 66 MHz，则通过计算可知分频比为 66 MHz/50 MHz = 1.32。因此，寄存器中设置的默认加数值为 2^{32} / 1.32，相当于 0xC1F07C1F。

如果参考时钟向下偏移，例如降至 65 MHz，则分频比为 65 / 50（即，1.3），加数寄存器中要设置的值为 2^{32} / 1.30，相当于 0xC4EC4EC4。

如果时钟向上偏移，例如升至 67 MHz，则必须将加数寄存器设置为 0xBF0B7672。无时钟偏移时，必须编程的默认加法值为 0xC1F07C1F (2^{32} / 1.32)。

在*图 778* 中，用于累加亚秒寄存器的常数值为十进制 43，这可使系统时间精度达到 20 ns（换言之，增量步长时间为 20 ns）。使能外部时间更新时，可选系统时间模块不可用。根据配置的不同，有两种不同的方法可用于更新系统时间寄存器。

软件必须根据 SYNC 消息对频率偏移进行计算，并相应更新加数寄存器。

首先，使用加数寄存器中的 FreqCompensationValue0 设置从时钟。该值的计算公式如下：

$$FreqCompensationValue_0 = 2^{32} / FreqDivisionRatio$$

如果起初假定 MasterToSlaveDelay 对于连续的同步消息是相同的，则必须应用本节介绍的算法。数个同步周期之后，频率会锁定。从时钟随后可确定 MasterToSlaveDelay 的精确值，并使用新值重新与主时钟同步。

该算法如下：

1. 在 $MasterSyncTime_n$ 时刻，主时钟向从时钟发送 SYNC 消息。从时钟在其本地时钟为 $SlaveClockTime_n$ 时接收到该消息，并用如下公式计算 $MasterClockTime_n$：

$$MasterClockTime_n = MasterSyncTime_n + MasterToSlaveDelay_n$$

2. 当前同步周期的主时钟计数 $MasterClockCount_n$ 的计算公式如下：

$$MasterClockCount_n = MasterClockTime_n - MasterClockTime_{n-1}$$

（假定 MasterToSlaveDelay 对于同步周期 n 和 n – 1 是相同的）

3. 当前同步周期的从时钟计数 $SlaveClockCount_n$ 的计算公式如下：

$$SlaveClockCount_n = SlaveClockTime_n - SlaveClockTime_{n-1}$$

4. 当前同步周期的主从时钟计数差值 $ClockDiffCount_n$ 的计算公式如下：

$$ClockDiffCount_n = MasterClockTime_n - SlaveClockTime_n$$

5. 从时钟的分频系数 $FreqScaleFactor_n$ 的计算公式如下：

$$FreqScaleFactor_n = (MasterClockCount_n + ClockDiffCount_n)/SlaveClockCount_n$$

6. 加数寄存器的频率补偿值 $FreqCompensationValue_n$ 的计算公式如下：

$$FreqCompensationValue_n = FreqScaleFactor_n \times FreqCompensationValue_{n-1}$$

理论上，该算法可在一个同步周期内实现锁定。但是，由于网络传播延迟和工作条件会不断变化，因此可能需要多个周期。该算法可进行自校准。如果最初通过主时钟设置的从时钟不正确，则该算法会花费更多同步周期将其校准。

有关详细的编程步骤，请参见*第 58.9.7 节：关于 IEEE 1588 时间戳的编程指南*。

发送路径功能

通过 MII 接口发送数据包的起始数据包分隔符 (SFD) 时，MAC 会捕获时间戳。对于要捕获其时间戳的数据包，可以按数据包进行控制。可对每个发送数据包进行标记，以指示是否应为其捕获时间戳。

MAC 不会处理发送的数据包来识别 PTP 数据包。需要指定要捕获其时间戳的数据包。可以使用发送描述符中的控制位指定数据包（请参见*第 58.10.3 节：发送描述符*）。MAC 将时间戳返回到相应发送描述符内的软件，进而将时间戳自动连接到特定 PTP 数据包。

64 位时间戳信息会被写入到 TDES0 和 TDES1 字段。TDES0 字段会保持时间戳的 32 个最低有效位。

接收路径功能

可以将 MAC 编程为捕获通过 MII 接口接收到的所有数据包的时间戳，或者编程为处理数据包来识别有效 PTP 消息。可使用*时间戳控制寄存器 (ETH_MACTSCR)* 的以下选项来控制要发送到应用程序的时间快照：

- 使能所有数据包的快照
- 使能 IEEE 1588 版本 1 或版本 2 时间戳的快照
- 使能直接通过以太网或 UDP-IP-Ethernet 发送的 PTP 数据包的快照
- 使能所接收到的 IPv4 或 IPv6 数据包的时间戳快照
- 仅使能 EVENT 消息（SYNC、DELAY_REQ、PDELAY_REQ 或 PDELAY_RESP）的时间戳快照
- 将节点使能为主节点或从节点，并选择快照类型

  此功能控制要拍摄快照的消息类型。

DMA 将时间戳返回到相应接收描述符内的软件应用程序。包含时间戳消息状态和 IPC 状态的扩展状态写入 RDES1 正常描述符中，时间戳的快照写入上下文描述符的 RDES0 和 RDES1 字段中。RDES0 字段会保持时间戳的 32 个最低有效位。

时间戳校准

根据 IEEE 1588 规范，必须在消息时间戳点（紧随起始帧分隔符八位字节之后的八位字节的第一个位的前沿）跨越节点和网络之间的边界时捕获时间戳。参考时序源（PTP 时钟 ***HCLK***）不同于 MAC Tx 或 Rx 时钟，因此必须对捕获的时间戳进行校准来解决因同步而产生的延迟问题。此外，还必须修正内部快照点和建议捕获点（节点和网络之间的边界）之间的延迟问题。

入站校准

在接收端，通过加上在入站校准寄存器中编程的校准值（入站校准值）来针对延迟和同步问题校准在内部快照点捕获的时间戳。需要在入站校准寄存器中编程的值的计算如下：

1. 对于因同步而进行的时间戳校准，通过将 INGRESS_SYNC_CORR 与同步的时间戳值相加来进行补偿，公式如下：

   `INGRESS_SYNC_CORR = -(2*PTP_CLK_PER)`

2. 对于消息时间戳点和内部时间戳快照点之间的延迟校准，通过用捕获的时间戳减去延迟值 (INGRESS_LATENCY) 来实现，公式如下：

   `入站校准 = INGRESS_SYNC_CORR - INGRESS_LATENCY`

3. 通过编程 MAC 时间戳入站校准寄存器中的 TSIC 字段来执行入站校准。入站校准始终为负值，并且以纳秒为单位进行表示。该值以补码形式表示，具体如下：
   - *时间戳控制寄存器 (ETH_MACTSCR)* 中的 TSCTRLSSR 位置 1 时，精度为 1 ns：将位 31 设为 1 并使位 30:0 包含 10^9（即 <ingress_correction_value>）的二进制值。例如，如果所需校准值为 -5 ns，则编程的值必须为 0xBB9A_C9FB。
   - ETH_MACTSCR 寄存器中的 TSCTRLSSR 位置复位时，精度约为 0.466 ns：将位 31 设为 1 并使位 30:0 包含 2^31（即 <ingress_correction_value>）的二进制值。

出站校准

在发送端，通过加上在出站校准寄存器中编程的校准值（出站校准值）来针对延迟和同步问题校准在内部快照点捕获的时间戳。需要在出站校准寄存器中编程的值的计算如下：

1. 对于因同步而进行的时间戳校准，通过将 EGRESS_SYNC_CORR 与同步的时间戳值相加来进行补偿，公式如下：

   选择 Enable one step timestamp feature（使能一步时间戳功能）时，

   ```
   EGRESS_SYNC_CORR = (1*PTP_CLK_PER + 4*TX_CLK_PER)
   ```

   否则

   ```
   EGRESS_SYNC_CORR = -(2*PTP_CLK_PER)
   ```

2. 对于建议捕获点和内部时间戳快照点之间的出站延迟校准，通过将延迟值 (EGRESS_LATENCY) 与捕获的时间戳相加来实现，公式如下：

   ```
   出站校准 = EGRESS_SYNC_CORR + EGRESS_LATENCY
   ```

3. 通过编程 MAC 时间戳出站校准寄存器中的 TSEC 字段来执行出站校准。出站校准值可正可负，以纳秒为单位进行表示。负值以补码形式表示，具体如下：

   – *时间戳控制寄存器 (ETH_MACTSCR)* 中的 TSCTRLSSR 位置 1 时，精度为 1 ns。

     如果校准值为正值，则将位 31 设为 0 并使位 30:0 包含 <egress_correction_value> 的二进制值。该值不能超过 0x3B9A_C9FF。

     如果校准值为负值，则将位 31 设为 1 并使位 30:0 包含 10^9（即 <egress_correction_value>）的二进制值。

     例如，如果所需校准值为 -5 ns，则编程的值应为 0xBB9A_C9FB。

   – *时间戳控制寄存器 (ETH_MACTSCR)* 中的 TSCTRLSSR 位复位时，精度约为 0.466 ns。

     如果校准值为正值，则将位 31 设为 0 并使位 30:0 包含 <egress_correction_value> 的二进制值。最大值为 0x7FFF_FFFF。

     如果校准值为负值，则将位 31 设为 1 并使位 30:0 包含 2^31（即 <egress_correction_value>）的二进制值。

一步时间戳

MAC 支持一步时间戳功能：它用于识别数据包中的偏移，并在该偏移处插入从应用程序接收到的时间戳。

可通过将 ATI 控制字中的位 20 (OSTC) 置 1 来为数据包使能一步时间戳功能。插入的时间戳包括从应用程序接收到的 64 位 TSSL 和 TSSH。

一步时间戳功能仅适用于 PTP over Ethernet 数据包，而并不适用于 PTP over IPv4/IPv6 数据包。

PTP 减荷功能

当 MAC 用作 PTP 网络中的特定节点时，使用该功能可自动生成特定 PTP 数据包。这类数据包可定期生成，或者由主机软件触发生成。在其他模式下，该功能可解析接收器端传入的 PTP 数据包，并且自动生成和响应所需 PTP 数据包。它有助于以更高的精度和更低的响应延迟为某些 PTP 节点功能减荷。

根据所编程模式的不同，MAC 会在不同的条件下生成 PTP 以太网消息：定期生成、由应用程序触发生成、或在接收到特定 PTP 消息时生成。*表 501* 列出了 PTP 消息的生成条件。

表 501. PTP 消息生成条件

编程			模式	生成 PTP 消息的条件	生成的 PTP 消息类型
SNAPTYPSEL	TSMSTRENA	TSEVNTENA			
00	0	1	普通或边界从模式	接收到 SYNC 消息	Delay_Req
00	1	1	普通或边界主模式	定期生成或由应用程序触发	SYNC
				接收到 Delay_Req 消息	Delay_Resp
01	0	1	透明从模式	定期生成或由应用程序触发	Pdelay_Req
				接收到 Pdelay_Req 消息	Pdelay_Resp
				接收到 SYNC 消息	Delay_Req
01	1	1	透明主模式	定期生成或由应用程序触发	Pdelay_Req
				接收到 Pdelay_Req 消息	Pdelay_Resp
				定期生成或由应用程序触发	SYNC
				接收到 Delay_Req 消息	Delay_Resp
11	X	X	点对点透明模式	定期生成或由应用程序触发	Pdelay_Req
				接收到 Pdelay_Req 消息	Pdelay_Resp
所有其他编程组合对于 PTP 减荷功能均无效。					

PTP 减荷功能通过 *PTP 减荷控制寄存器 (ETH_MACPOCR)* 寄存器进行配置。80 位 PTP 节点标识在以下三个寄存器中给出：*PTP 源端口标识 0 寄存器 (ETH_MACSPI0R)*、*PTP 源端口标识 1 寄存器 (ETH_MACSPI1R)* 和 *PTP 源端口标识 2 寄存器 (ETH_MACSPI2R)*。

58.5.5 IPv4 ARP 减荷

MAC 支持针对 IPv4 数据包的地址识别协议 (ARP) 减荷。使用该功能可在接收路径中处理 IPv4 ARP 请求数据包，并在发送路径中生成相应 ARP 响应数据包。

MAC 会为相应 ARP 请求数据包生成 ARP 响应数据包。IPv4 的 ARP 数据包为 L2 层数据包，其长度/类型为 0x0806。

ARP 减荷序列如下：

1. 如果请求目标协议地址与 MAC L3 寄存器中编程的 IPv4 地址匹配，则 MAC 接收器会获取 ARP 请求。
2. MAC 生成 ARP 响应数据包。
3. MAC 将 ARP 请求中的发送方硬件地址字段复制到以下字段：
   – 以太网数据包报头的 DA 字段
   – ARP 响应数据包的目标硬件地址字段
4. MAC 将 ARP 请求中的发送方协议地址字段复制到 ARP 响应数据包中的目标协议地址字段。
5. MAC 将其 MAC 地址置于以下字段中：
   – 以太网数据包报头的 SA 字段
   – ARP 响应数据包的发送方硬件地址字段
6. MAC 将 ARP 请求中的目标协议地址字段复制到 ARP 响应数据包中的发送方协议地址字段。
7. MAC 将 ARP 响应数据包中的操作码字段设为 2，以表示 ARP 响应。
8. MAC 重新计算 CRC，并对生成的 ARP 响应数据包进行填充。
9. MAC 发送器发送 ARP 响应

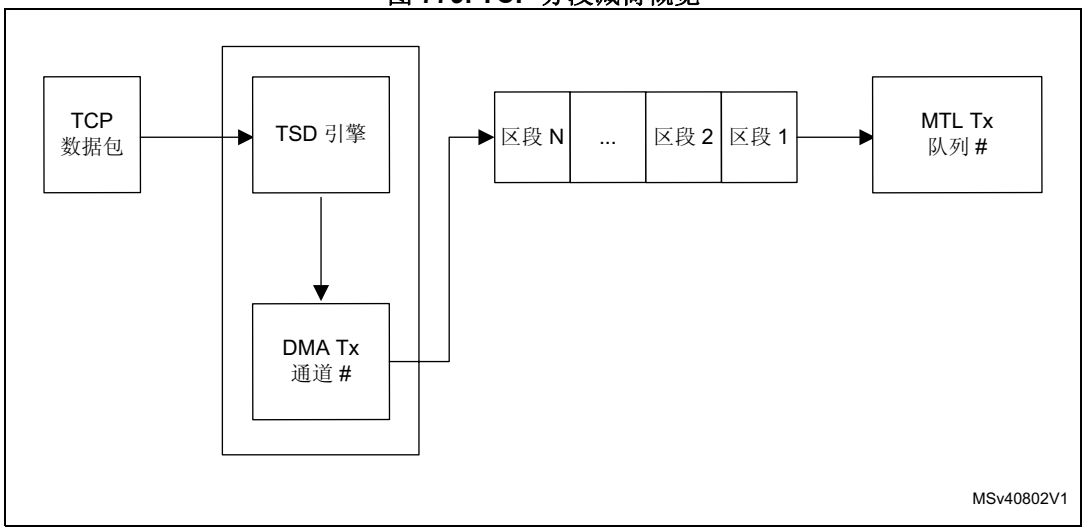
58.5.6 TCP 分段减荷

MAC 支持 TCP 分段减荷 (TSO) 功能，DMA 可使用该功能将大型 TCP 数据包拆分成多个小型数据包，并将这些数据包传递到 MTL，如*图 779* 所示。

通过对相应 ETH_DMACCR 寄存器的 TSE 位进行编程来使能该功能（请参见*通道控制寄存器 (ETH_DMACCR)*）。只有 MAC 工作在全双工模式下时，才支持该功能。

有关详细的编程步骤，请参见*第 58.9.10 节：关于 TSO 的编程指南*。

图 779. TCP 分段减荷概览



使能 TSO 功能

要使能数据包分段功能，应用程序必须将第一个正常描述符（请参见*第 58.10.3 节：发送描述符*）的 TDES3 的 TSE 位置 1。

应用程序必须在 TDES3[17:0] 中编程 TCP 数据包有效负载的长度，并在 TDES3[22:19] 中编程 TCP 报头的长度。可分段的 TCP 数据包有效负载的最大长度为 256 KB。

应在 TSO 数据包第一个正常描述符的缓冲区 1 中提供数据包报头，其中包括以太网报头、L3 报头和 L4 报头。只有为 TSO 使能的数据包的第一个正常描述符缓冲区 1 才被视为包含报头的缓冲区。

TCP 有效负载可从第一个正常描述符的缓冲区 2 开始，并持续到第二个正常描述符和后续描述符的缓冲区 1 和缓冲区 2。

TCP 有效负载可跨越多个缓冲区和多个描述符。包含 TCP 有效负载的缓冲区的总体大小应等于第一个正常描述符的 TDES3[17:0] 中提供的 TCP 有效负载长度。

MAC 始终会计算并附加 CRC，并为由 DMA 分段的所有数据包插入填充（如果需要）。如果使能 TDES3 的 TSE 位，则会保留 TDES3 的 CRC 填充控制 (CPC) 字段。为确定分段后 TCP 数据包的大小，DMA 使用应用程序通过上下文描述符提供的最大段大小 (MSS)。DMA 会对有效负载大小大于 MSS 的数据包进行分段。应用程序必须通过编程 ETH_DMACCR（请参见*通道控制寄存器 (ETH_DMACCR)*）中的 MSS 值或通过提供上下文描述符来提供 MSS。DMA 使用 MSS 的最后一个编程值或通过上下文提供的最后一个 MSS 值（以较后者为准）。

报头长度加上 MSS 大小（等于每个 TCP 段的大小）不应超过 16383 字节，否则 MAC 发送器会将数据包中 16383 字节之后的部分截断，以免导致出现 CRC 错误。

报头长度加上 MSS 大小再加上编程的 PBL 值（位于 ETH_DMACTxCR 寄存器中，请参见*通道发送控制寄存器 (ETH_DMACTxCR)*）应小于在 ETH_MTLTxQOMR 寄存器（请参见*Tx 队列工作模式寄存器 (ETH_MTLTxQOMR)*）的 TQS 字段中编程的 Tx 队列大小。建议使 MSS 加上报头的大小等于所编程 Tx 队列大小的一半。

如果 TCP 数据包具有 VLAN 标记，则无论提供的是哪种 VLAN 标记类型（C-VLAN 或 S-VLAN），都会为所有段使用同一标记。VLAN 标记插入/替换控制位适用于所有段。

如果选择了双 VLAN 功能，则无论提供的是哪种 VLAN 标记类型（C-VLAN 或 S-VLAN），DMA 都会为所有段传递两个标记。两个标记的 VLAN 标记插入/替换控制位适用于所有段。如果未选择双 VLAN 功能，则应用程序不得针对带有两个标记的 TCP/IP 数据包将 TDES3 中的 TSE 位置 1，否则 DMA 将出现不可预测的行为。

TCP/IP 报头字段

对 TCP 数据包进行分段时，DMA 会自动更新 TCP/IP 报头字段。*表 502* 介绍了 TCP 和 IP 报头的更新过程。

表 502. TSO：TCP 和 IP 报头字段

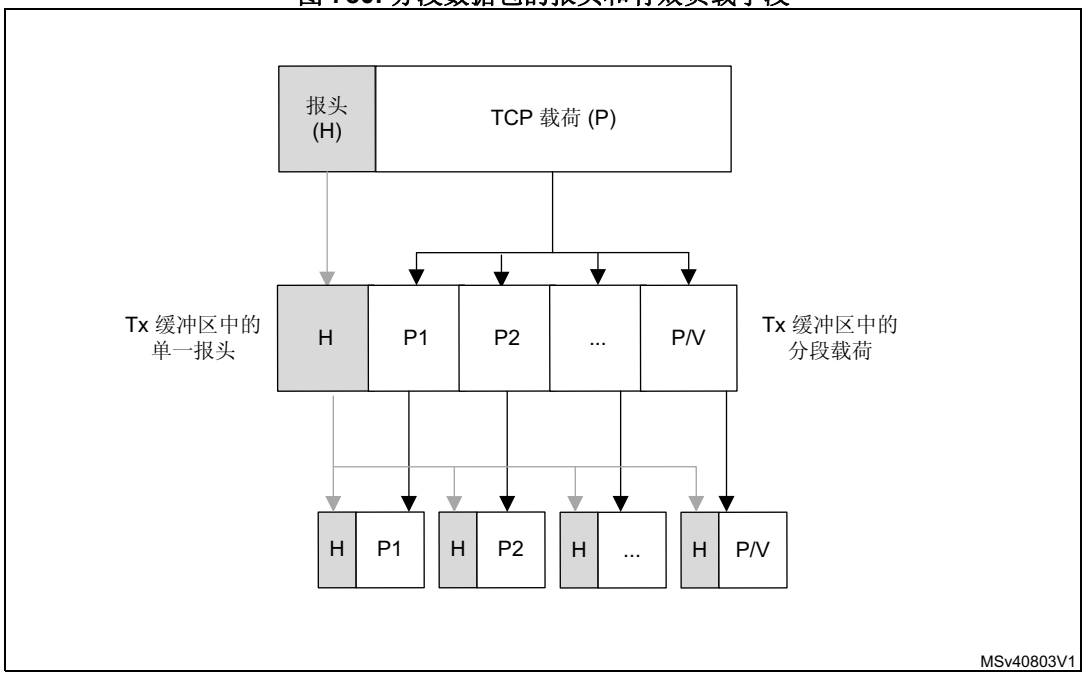
数据包序列	TCP 报头	IP 报头
第一个数据包	1. 不更新序列号。使用报头中提供的值。 2. 如果 FIN 和 PSH 标志置 1，则将其清零。 3. 重新计算 TCP 校验和。	IPv4 报头 – 总长度 = MSS + TCP 报头长度 + IP 报头长度 – 不修改标识字段。按照软件提供的报头进行发送。 – 重新计算 IPv4 报头校验和。 IPv6 报头 – 有效负载长度 = MSS + TCP 报头长度 + IP 扩展报头长度
后续数据包	1. 更新序列号。将 MSS 值与前一段的序列号值相加。 2. 如果 FIN 和 PSH 标志置 1，则将其清零。 3. 重新计算 TCP 校验和。	IPv4 报头 – 总长度 = MSS + TCP 报头长度 + IP 报头长度 – 标识字段 = 前一标识字段 + 1 – 重新计算 IPv4 报头校验和 IPv6 报头 – 有效负载长度 = MSS + TCP 报头长度 + IP 扩展报头长度
最后一个数据包	1. 更新序列号。将 MSS 值与前一段的序列号值相加。 2. 如果 FIN 和 PSH 标志在原始报头中被置 1，则将这些标志置 1。 3. 重新计算 TCP 校验和。	IPv4 报头 – 总长度 = 剩余有效负载 + TCP 报头长度 + IP 报头长度 – 标识字段 = 前一标识字段 + 1 – 重新计算 IPv4 报头校验和 IPv6 报头 – 有效负载长度 = 剩余有效负载长度 + TCP 报头长度 + IP 扩展报头长度

分段数据包的报头和有效负载字段

分段后，除了*表 502：TSO：TCP 和 IP 报头字段*中所介绍的报头字段以外，拆分数据包均与父 TCP 数据包使用相同的报头。*图 780：分段数据包的报头和有效负载字段*显示了如何将同一报头用于各个分段数据包的报头字段。

应用程序必须在要分段的数据包的第一个描述符的缓冲区 1 中创建报头，并在第一个描述符（FD = 1）的 TDES2 中提供报头长度。FD 位置 1 时，DMA 从 TDES0 指向的报头缓冲区读取报头。第一个描述符的缓冲区 2 可用于有效负载以及后续描述符的 TDES0 和 TDES1。对于后续描述符（FD = 0），TDES0 和 TDES1 所指向的地址被视为同一数据包的有效负载缓冲区地址。

图 780. 分段数据包的报头和有效负载字段



上下文描述符序列

上下文描述符可提供用于分段的最大段大小 (MSS) 值。应用程序必须在要用于相应 TCP 数据包的正常描述符之前提供上下文描述符。如果应用程序需要提供新的 MSS，则必须于描述符列表中在要使用新 MSS 值分段的数据包的第一个正常描述符之前，创建上下文描述符。只有在上下文描述符中 TDES3 的 TCMSSV 位置 1 且 OSTC 位复位（请参见*第 58.10.3 节：发送描述符*）时，上下文描述符中的 MSS 值才有效。

应用程序提供包含有效 MSS 值的上下文描述符时，DMA 会在内部存储 MSS 值，并通过 TDES3 正常描述符的 TSE 位将该值用于所有使能了 TSO 的后续数据包。

如果应用程序将一个上下文描述符置于数据包的中间位置（数据包的第一个描述符和最后一个描述符之间），则 DMA 会执行以下操作：

1. DMA 忽略上下文并关闭描述符。
2. DMA 在描述符状态中指示错误。
3. 如果在 DMA 通道对应的中断使能寄存器中将 CDEE 位置 1，则 DMA 会生成中断（请参见*通道中断使能寄存器 (ETH_DMACIER)*）。

应用程序可通过 DMA 通道对应的状态寄存器的 CDE 位来读取中断状态（请参见*通道状态寄存器 (ETH_DMACSR)*）。

58.5.7 回送

MAC 支持对发送到其接收器的数据包进行回送。默认情况下，禁止 MAC 回送功能，可通过编程*工作模式配置寄存器 (ETH_MACCR)* 寄存器中的 LM 位使能该功能。

回送功能适用于所有 PHY 接口。无论选择哪个 PHY 接口，数据都始终在 MII 接口上进行回送。

58.5.8 流控制

本节介绍发送和接收路径的流控制。

发送流控制

Tx 队列流控制寄存器 (ETH_MACQTxFCR) 中的 TFE 位置 1 时，会使能发送流控制。

流控制触发

发送流控制包括在全双工模式下发送暂停数据包以及在半双工模式下提供背压功能，以便控制来自远程端的数据包流。应用程序可通过将相应 *Tx 队列流控制寄存器 (ETH_MACQTxFCR)* 中的 FCB 位置 1 来请求 MAC 发送暂停数据包或激活背压功能。

全双工模式下的流控制：暂停数据包控制

在全双工模式下，MAC 使用 IEEE 802.3x 暂停数据包进行流控制。*表 503* 介绍了暂停数据包的各个字段。

表 503. 暂停数据包字段

字段	说明
DA	包含特殊多播地址
SA	包含 MAC 地址 0
类型	包含 8808
MAC 控制操作码	对于 IEEE 802.3x 暂停控制数据包，包含 0001； 对于 PFC 数据包，包含 0101
PT	包含在 的 PT 字段中指定的暂停时间 *Tx 队列流控制寄存器 (ETH_MACQTxFCR)*

FCB 位置 1 时，MAC 会生成并发送一个暂停数据包。如果在完成暂停数据包发送后再次将 FCB 位置 1，则无论暂停时间是否结束，MAC 都会发送另一个暂停数据包。要在先前发送的暂停数据包中指定的时间之前延长暂停时间或终止暂停，应用程序应使用合适的值对暂停时间寄存器进行编程，然后再次将 FCB 位置 1。

半双工模式下的流控制

在半双工模式下，MAC 使用延迟机制进行流控制（背压）。当应用程序请求停止接收数据包时，如果已使能发送流控制，则只要 MAC 检测到接收数据包，就会发送一个 32 字节的 JAM 信号。这会导致冲突并使远程站回退。如果应用程序请求发送数据包，则即使激活背压功能，也将按调度计划发送。如果背压功能长时间保持激活（发生的连续冲突事件超过 16 个），则远程站将因冲突过多而中止发送。

表 504 介绍了 Tx 路径中基于以下位设置的流控制：

- *Rx 队列工作模式寄存器 (ETH_MTLRxQOMR)* 的 EHFC 位
- *Tx 队列流控制寄存器 (ETH_MACQTxFCR)* 的 TFE 位
- *工作模式配置寄存器 (ETH_MACCR)* 的 DM 位

流控制对于所有队列均类似。

表 504. Tx MAC 流控制

EFC	TFE	DM	说明
x	0	x	MAC 发送器不执行流控制或背压操作。
0	1	0	将 *Tx 队列流控制寄存器 (ETH_MACQTxFCR)* 的位 0 置 1 后，MAC 发送器会执行背压操作。
1	1	0	将 *Tx 队列流控制寄存器 (ETH_MACQTxFCR)* 的位 0 置 1 后，MAC 发送器会执行背压操作。此外，当 Rx 队列级别超出由 *Rx 队列工作模式寄存器 (ETH_MTLRxQOMR)* 的位 [10:8] 设置的阈值时，MAC Tx 也会执行背压操作。
0	1	1	将 *Tx 队列流控制寄存器 (ETH_MACQTxFCR)* 的位 0 置 1 后，MAC 发送器会发送暂停数据包。
1	1	1	将 *Tx 队列流控制寄存器 (ETH_MACQTxFCR)* 的位 0 置 1 后，MAC 发送器会发送暂停数据包。此外，当 Rx 队列级别超出由 *Rx 队列工作模式寄存器 (ETH_MTLRxQOMR)* 的位 [10:8] 设置的阈值时，MAC Tx 也会发送暂停数据包。

接收流控制

在接收路径中，流控制仅在全双工模式下可用。如果在半双工模式下接收到暂停数据包，则该数据包会被视为正常控制数据包。可通过将 *Rx 流控制寄存器 (ETH_MACRxFCR)* 中的 RFE 位置 1 来使能暂停流控制。*表 505* 介绍了 Rx 路径中基于以下位设置的流控制：

- *Rx 流控制寄存器 (ETH_MACRxFCR)* 的 RFE 位
- *工作模式配置寄存器 (ETH_MACCR)* 的 DM 位

表 505. Rx MAC 流控制

RFE	DM	说明
0	x	MAC 接收器未检测出接收的暂停数据包。
1	0	MAC 接收器未检测出接收的暂停数据包， 而将这些数据包识别为控制数据包。
1	1	MAC 接收器检测出或已处理暂停数据包， 并通过停止 MAC 发送器来响应这些数据包。

Rx 流控制的序列如下：

1. MAC 会基于多播或单播目标地址检查已接收暂停数据包的目标地址。
2. MAC 对已接收数据包的类型 (0x8808) 和操作码（0x0001：暂停数据包）字段进行解码，并且捕获暂停数据包中的暂停时间以确定发送器需要被中断的时间。
3. 如果状态的字节计数指示 64 个字节，并且不存在任何 CRC 错误，则 MAC 将暂停任何数据包的发送，暂停时间为解码的暂停时间值乘以时隙（64 字节时间）。

58.5.9 校验和减荷引擎

通信协议（例如 TCP 和 UDP）将实施校验和字段，这有助于确定通过网络发送的数据的完整性。以太网最广泛的用途是通过 IP 数据报封装 TCP 和 UDP。MAC 具有校验和减荷引擎 (COE)，支持校验和计算、发送路径中的校验与插入以及接收路径中的错误检测。

发送校验和减荷引擎

COE 模块支持两种类型的校验和计算与插入。可以通过将 CIC 位（TDES3 位 [17:16]）置 1 来为各个数据包控制校验和引擎。

注： *通过完整的数据包来计算 TCP、UDP 或 ICMP 的校验和，然后将其插入相应的报头字段。鉴于此要求，即使将 MAC 配置为阈值（直通）模式，Tx FIFO 也会自动进入存储转发模式工作。*

IP 报头校验和引擎

在 IPv4 数据报中，报头字段的完整性由 16 位头校验和字段（IPv4 数据报的第十一个字节和第十二个字节）指示。当以太网数据包的类型字段的值为 0x0800 且 IP 数据报的版本字段的值为 0x4 时，COE 会检测到 IPv4 数据报。计算期间，将忽略输入数据包的校验和字段并将其替换为计算出的值。

IPv6 报头无校验和字段。因此，COE 不会修改 IPv6 报头字段。

此 IP 报头校验和计算的结果由发送状态中的 IP 报头错误状态位（*表 519：TDES3 正常描述符（回写格式）*中的位 0）指示。

TCP/UDP/ICMP 校验和引擎

TCP/UDP/ICMP 校验和对 IPv4 或 IPv6 报头（包括扩展报头）进行处理，并确定封装的有效负载是 TCP、UDP 还是 ICMP。计算 TCP、UDP 或 ICMP 有效负载的校验和，然后将其插入报头中的相应字段。Tx COE 可工作在以下两种模式下：

- TCP、UDP 或 ICMPv6 伪报头并未包含在校验和计算中，并假定其存在于输入数据包的校验和字段中。该引擎将校验和字段包含在校验和计算中，然后将校验和字段替换为最终计算得出的校验和。
- 引擎将忽略校验和字段，并将 TCP、UDP 或 ICMPv6 伪报头数据包含在校验和计算中，然后使用最终计算出的值覆盖校验和字段。

注： *对于 ICMP-over-IPv4 数据包，由于没有为其定义伪报头，因此在这两种模式下 ICMP 数据包中的校验和字段都必须始终为 0x0000。如果不等于 0x0000，可能向数据包插入不正确的校验和。*

此操作的结果由发送状态向量中的有效负载校验和错误状态位（*表 519：TDES3 正常描述符（回写格式）*中的位 12）指示。如果该引擎检测到数据包已在存储转发模式下被转发到 MAC 发送器引擎但未向 FIFO 写入数据包结束 (EOP)，或者在接收到 IP 报头的有效负载长度字段所指示的字节数之前数据包已结束，则会将有效负载校验和错误状态位置 1。当数据包的长度大于指示的有效负载长度时，COE 会将其作为填充字节忽略，并且不报告错误。该引擎检测到第一种类型的错误时，不会修改 TCP、UDP 或 ICMP 报头。对于第二种错误类型，仍会将计算得出的校验和插入相应的报头字段。

表 506 介绍了发送校验和减荷引擎针对不同数据包类型所支持的功能。当 MAC 未插入校验和时，表中将以“否”进行表示。

表 506. 针对不同数据包类型的发送校验和减荷引擎功能

数据包类型	硬件 IP 报头校验和插入	硬件 TCP/UDP 校验和插入
非 IPv4 或 IPv6 数据包	否	否
带 TCP、UDP 或 ICMP 的 IPv4	是	是
IPv4 数据包具有 IP 选项（IP 报头长度大于 20 字节）	是	是
数据包为 IPv4 段	是	否
主报头或扩展报头中包含以下后续报头字段的 IPv6 数据包： – 逐跳选项（在 IPv6 主报头中） – 逐跳选项（在 IPv6 扩展报头中） – 目标选项 – 路由（剩余段为 0） – 路由（剩余段 > 0） – TCP、UDP 或 ICMP – 认证 – 主报头或扩展报头中的任何其他后续报头字段	 – 不适用 – 不适用 – 不适用 – 不适用 – 不适用 – 不适用 – 不适用 – 不适用	 – 是 – 否 – 是 – 否 – 否 – 是 – 是 – 否
IPv4 数据包具有带选项字段的 TCP 报头	是	是
IPv4 隧道： – IPv4 隧道中的 IPv4 数据包 – IPv4 隧道中的 IPv6 数据包	 – 是（IPv4 隧道报头） – 是（IPv4 隧道报头）	 – 否 – 否
IPv6 隧道： – IPv6 隧道中的 IPv4 数据包 – IPv6 隧道中的 IPv6 数据包	 – 不适用 – 不适用	 – 否 – 否
IPv4 数据包具有 802.3ac 标记（使能时带有 C-VLAN 标记或 S-VLAN 标记）。	是	是
IPv6 数据包具有 802.3ac 标记（使能时带有 C-VLAN 标记或 S-VLAN 标记）。	不适用	是
带安全功能（例如封装的安全有效负载）的 IPv4 帧	是	否
带安全功能（例如封装的安全有效负载）的 IPv6 帧	不适用	否

接收校验和减荷引擎

可通过选择 *Enable Receive TCP/IP Checksum Check（使能接收 TCP/IP 校验和检查）*选项并将*工作模式配置寄存器 (ETH_MACCR)* 的 IPC 位置 1 来使能接收校验和减荷引擎 (Rx COE)。选择该选项后，将对接收的以太网数据包中的 IPv4 和 IPv6 数据包进行检测和处理，以确保数据完整性。MAC 接收器通过检查所接收的以太网数据包的类型字段中是否存在值 0x0800 或 0x86DD，来识别 IPv4 或 IPv6 数据包。此识别方法也适用于带单一 VLAN 标记的数据包。选择了 *Enable Double VLAN Processing（使能双 VLAN 处理）*选项并将 *VLAN 标记寄存器 (ETH_MACVTR)* 的 EDVLP 位置 1 时，此方法还适用于带有双 VLAN 标记的数据包。

Rx COE 将计算 IPv4 报头校验和，并检查其是否与接收的 IPv4 报头校验和匹配。此操作的结果（通过或失败）将提供给 RFC 模块以插入到接收状态字中。如果指示的有效负载类型（以太网类型字段）与 IP 报头版本之间有任何不匹配，或接收的数据包的字节数小于 IPv4 报头的长度字段中指示的数量（IPv4 或 IPv6 报头中的可用字节数少于 20），IP 报头错误位将置 1。

此引擎还可以识别接收的 IP 数据报（IPv4 或 IPv6）中的 TCP、UDP 或 ICMP 有效负载，并正确计算此类有效负载的校验和，如 TCP、UDP 或 ICMP 规范中所定义。该引擎包括用于校验和计算的 TCP、UDP 或 ICMPv6 伪报头字节，并检查接收的校验和字段是否与计算的值匹配。此操作的结果由接收状态字中的有效负载校验和错误位给出。如果 TCP、UDP 或 ICMP 有效负载的长度与 IP 报头中给出的预期有效负载长度不匹配，此状态位也将置 1。

*表 507：针对不同数据包类型的接收校验和减荷引擎功能*介绍了 Rx COE 针对不同数据包类型所支持的功能。未处理 IP 数据包的有效负载（在表中以“否”进行表示）时，会在接收状态中给出是否绕过校验和引擎的信息。

注： *MAC 不会向接收的以太网数据包附加任何有效负载校验和字节。*

表 507. 针对不同数据包类型的接收校验和减荷引擎功能

数据包类型	硬件 IP 报头校验和检查	硬件 TCP/UDP/ICMP 校验和检查
非 IPv4 或 IPv6	否	否
带 TCP、UDP 或 ICMP 的 IPv4	是	是
IPv4 报头的协议字段包含除 TCP、UDP 或 ICMP 之外的协议	是	否
IPv4 数据包具有 IP 选项（IP 报头长度大于 20 字节）	是	是
数据包为 IPv4 段	是	否
主报头或扩展报头中包含以下后续报头字段的 IPv6 数据包： – 逐跳选项（在 IPv6 主报头中） – 逐跳选项（在 IPv6 扩展报头中） – 目标选项 – 路由（剩余段为 0） – 路由（剩余段 > 0） – TCP、UDP 或 ICMP – 主报头或扩展报头中的任何其他后续报头字段	 – 不适用 – 不适用 – 不适用 – 不适用 – 不适用 – 不适用 – 不适用	 – 是 – 否 – 是 – 是 – 否 – 是 – 否

表 507. 针对不同数据包类型的接收校验和减荷引擎功能（续）

数据包类型	硬件 IP 报头校验和检查	硬件 TCP/UDP/ICMP 校验和检查
IPv4 数据包具有带选项字段的 TCP 报头	是	是
IPv4 隧道： – IPv4 隧道中的 IPv4 数据包 – IPv4 隧道中的 IPv6 数据包	 – 是（IPv4 隧道报头） – 是（IPv4 隧道报头）	 – 否 – 否
IPv6 隧道： – IPv6 隧道中的 IPv4 数据包 – IPv6 隧道中的 IPv6 数据包	 – 不适用 – 不适用	 – 否 – 否
IPv4 数据包具有 802.3ac 标记（使能时带有 C-VLAN 标记或 S-VLAN 标记）。	是	是
IPv6 数据包具有 802.3ac 标记（使能时带有 C-VLAN 标记或 S-VLAN 标记）。	不适用	是
带安全功能（例如封装的安全有效负载）的 IPv4 帧	是	否
带安全功能（例如封装的安全有效负载）的 IPv6 帧	不适用	否

58.5.10 MAC 管理计数器

可将 MAC 管理计数器 (MMC) 模块中的计数器视为 CSR 模块的寄存器地址空间的扩展。MMC 模块保持一组寄存器来收集有关已接收数据包和已发送数据包的统计信息。这组寄存器包括一个用于控制各寄存器行为的控制寄存器、两个包含所生成中断的 32 位寄存器（接收和发送），以及两个包含中断寄存器屏蔽的 32 位寄存器（接收和发送）。这些寄存器可以通过 AHB 从接口从应用程序中进行访问，访问方式与 CSR 寄存器相同。这些寄存器的构成如*第 58.11.4 节：以太网 MAC 和 MMC 寄存器*中所示。

MMC 计数器自由运行。无需单独对计数器进行使能即可开始计数。当接收或发送相应数据包时，特定的 MMC 计数器就会开始计数。

除了控制寄存器外，还实现了以下两组寄存器：

- 6 个用于冲突、错误和良好数据包计数器的寄存器
- 4 个用于记录 LPI 模式转变的寄存器

58.5.11 MAC 生成的中断

MAC 可由于各种事件而生成中断。这些中断事件与 DMA 中的事件结合用于 eth_sbd_intr_it 信号。MAC 中断为电平型中断，即在被应用程序或软件清除之前始终保持有效（高电平）。

中断状态寄存器 (ETH_MACISR) 介绍了可能导致 MAC 生成中断的事件。默认情况下 MAC 中断处于使能状态。可通过将 *中断使能寄存器 (ETH_MACIER)* 中的相应屏蔽位置 1，以阻止各个事件将 eth_sbd_intr_it 信号中断置为有效。

中断寄存器位仅指示报告事件的模块。必须读取相应的状态寄存器和其他寄存器才能清除中断。

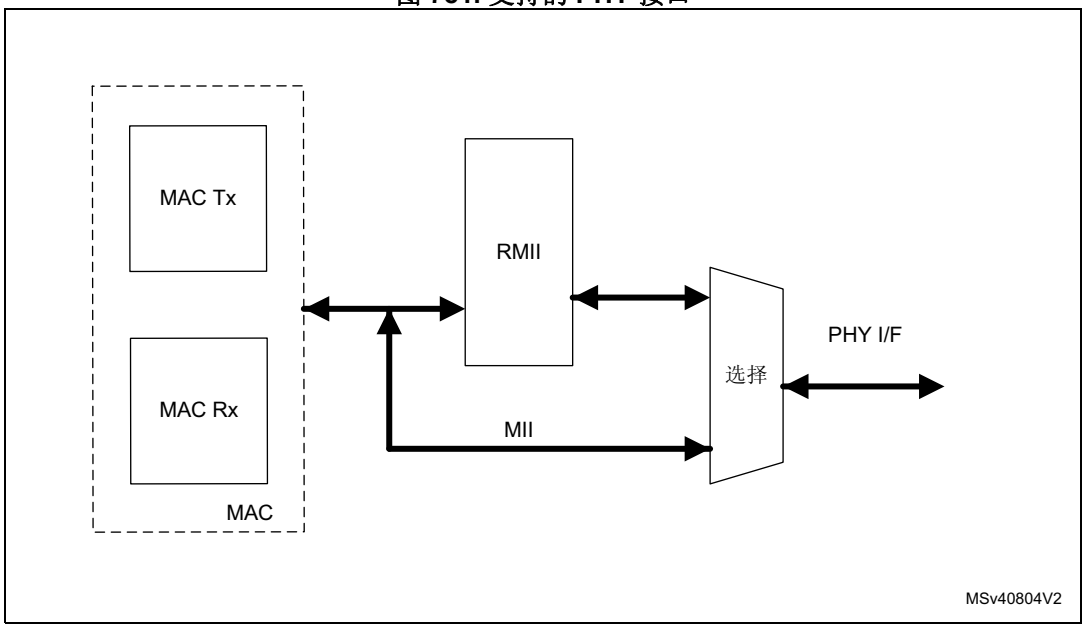
58.5.12 MAC 和 MMC 寄存器说明

请参见*第 58.11.4 节：以太网 MAC 和 MMC 寄存器*。

58.6 以太网功能说明：PHY 接口

以太网外设支持多个 PHY 接口。根接口为 MII 接口。所有其他接口均源自该接口，如*图 781* 所示。

图 781. 支持的 PHY 接口



本节介绍了用于 PHY 控制的 SMA 模块以及不同的 PHY 接口。其中涵盖以下部分：

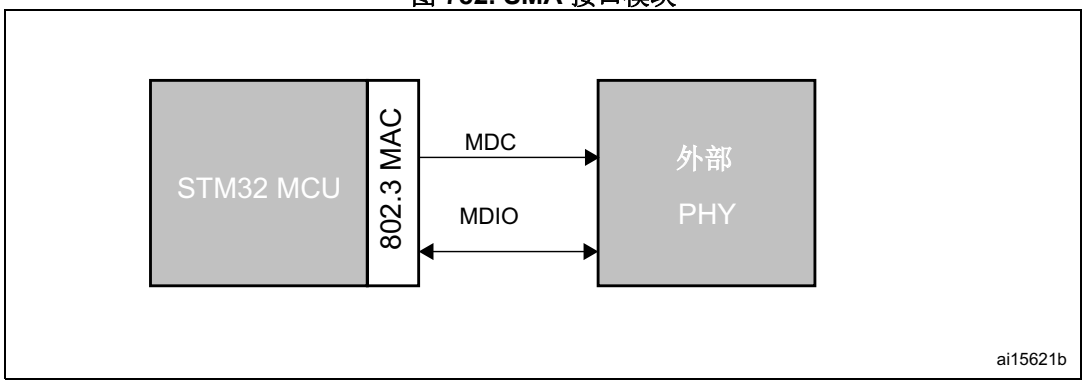
- *站管理代理 (SMA)*
- *介质独立接口 (MII)*
- *精简介质独立接口 (RMII)*

58.6.1 站管理代理 (SMA)

应用程序可通过站管理代理 (SMA) 模块访问 PHY 寄存器。SMA 包含一个双线站管理接口 (MIM)。

SMA 模块支持访问多达 32 个 PHY。应用程序可寻址 32 个 PHY 中的任意一个的 32 个寄存器之一。一次只能对一个 PHY 中的一个寄存器进行寻址。应用程序通过 SMA 模块向 PHY 发送控制数据以及从 PHY 接收状态信息，如*图 782* 所示。

图 782. SMA 接口模块



SMA 功能概述

MAC 基于 MDC 时钟触发管理写操作或读操作。MDC 时钟通过 CSR 时钟导出。分频系数取决于 *MDIO 地址寄存器 (ETH_MACMDIOAR)* 寄存器中设置的时钟范围。MDC 时钟的选择如下：

表 508. MCD 时钟选择

选择	CSR 时钟	MDC 时钟
0000	60-100 MHz	CSR 时钟/42
0001	100-150 MHz	CSR 时钟/62
0010	20-35 MHz	CSR 时钟/16
0011	35-60 MHz	CSR 时钟/26
0100	150-250 MHz	CSR 时钟/102
0101	250-300 MHz	CSR 时钟/124
0110, 0111	保留	-

通过 mdi_i、mdo_o 和 mdo_oe 信号在 MAC 和 PHY 之间进行数据交换。该信号组通过三态缓冲器并作为连接到 PHY 的 MDIO 线进行输出。

下图显示了 MDIO 数据包的结构，而*表 509* 则给出了数据包字段的详细说明。

图 783. MDIO 数据包结构

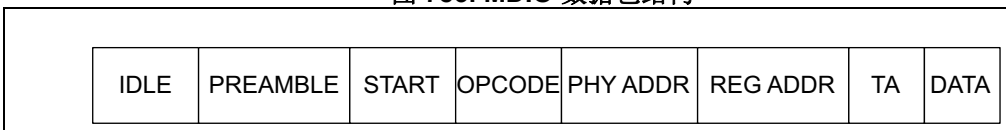


表 509. MDIO 数据包字段说明

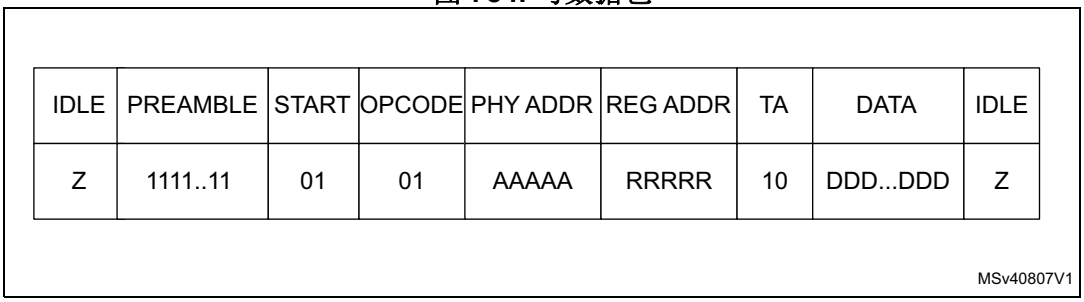
字段	说明
IDLE	MDIO 线为三态形式；ETH_MDC 无时钟。
PREAMBLE	32 个值为 1 的连续位
START	数据包起始为 2'01
OPCODE	2'b10 表示读操作，2'b01 表示写操作
PHY ADDR	32 个 PHY 之一的 5 位地址选择
REG ADDR	所选 PHY 中的寄存器地址
TA	对于读操作，周转值为 2'bZ0；对于写操作，周转值为 2'b10
DATA	任意 16 位值。对于写操作，由 MAC 驱动 MDIO。对于读操作，则由 PHY 驱动。

MII 管理写操作

将 *MDIO 地址寄存器 (ETH_MACMDIOAR)* 中的位 [3:2] 设为 01、位 0 设为 1 时，MAC CSR 模块会将 PHY 地址、PHY 中的寄存器地址和写数据（*MDIO 数据寄存器 (ETH_MACMDIODR)*）发送到 SMA，以触发对 PHY 寄存器的写操作。此时，SMA 模块开始使用 MII 规范（根据 IEEE 802.3-2002 规范的第 *22.2.4.5* 节）中指定的管理数据包格式通过 MII 管理接口进行写操作。

SMA 模块开始写操作时，会通过 MDIO 线发送写数据包。 MAC 会在数据包的整个持续时间内驱动 MDIO 线。写操作完成之前，忙位始终处于高电平。在此期间（忙位为高电平时）， CSR 会忽略对 *MDIO 地址寄存器 (ETH_MACMDIOAR)* 或 *MDIO 数据寄存器 (ETH_MACMDIODR)* 执行的写操作。完成写操作后，SMA 模块会向 CSR 提供相关指示，CSR 随后会将忙位复位。写操作的数据包格式如下：

图 784. 写数据包



MII 管理读操作

将 *MDIO 地址寄存器 (ETH_MACMDIOAR)* 中的位 [3:2] 设为 11、位 0 设为 1 时，MAC CSR 模块会将 PHY 地址和 PHY 中的寄存器地址发送到 SMA，以触发对 PHY 寄存器的读操作。此时，SMA 模块开始使用 MII 规范（根据 IEEE 802.3-2002 规范的第 *22.2.4.5* 节）中指定的管理数据包格式通过 MII 管理接口进行读操作。

当 SMA 模块开始在 MDIO 上进行读操作时，CSR 会忽略在此期间对 *MDIO 地址寄存器 (ETH_MACMDIOAR)* 或 *MDIO 数据寄存器 (ETH_MACMDIODR)* 寄存器进行的写操作（忙位为高电平），并且通过 MCI 接口无错完成事务。完成读操作后，SMA 会向 CSR 提供相关指示。CSR 会将忙位复位，并会使用从 PHY 读取的数据更新 *MDIO 数据寄存器 (ETH_MACMDIODR)*。除了 PHY 驱动 MDIO 线时的相应数据字段期间，MAC 会在帧的整个持续时间内驱动 MDIO 线。有关应用程序与 PHY 通信的更多信息，请参见 IEEE 802.3z 1000BASE 以太网的“协调子层和介质独立接口规范”部分。

读操作的数据包格式如下：

图 785. 读数据包

IDLE	PREAMBLE	START	OPCODE	PHY ADDR	REG ADDR	TA	DATA	IDLE
Z	1111..11	01	10	AAAAA	RRRRR	Z0	DDD...DDD	Z

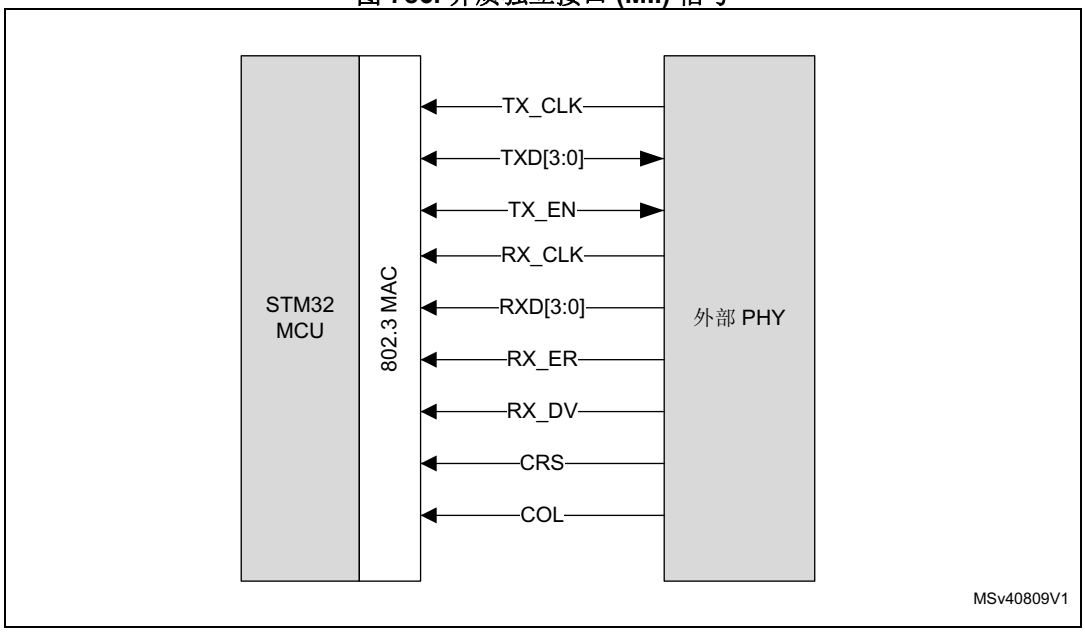
MSv40808V1

58.6.2 介质独立接口 (MII)

介质独立接口 (MII) 定义了 10 Mb/s 和 100 Mb/s 的数据传输速率下 MAC 子层与 PHY 之间的互连。

MII 信号如*图 786：介质独立接口 (MII) 信号*所示。

图 786. 介质独立接口 (MII) 信号



- TX_CLK：连续时钟信号，该信号提供进行 Tx 数据传输时的参考时序。速率为 10 Mb/s 时标称频率为 2.5 MHz；速率为 100 Mb/s 时标称频率为 25 MHz。
- RX_CLK：连续时钟信号，该信号提供进行 Rx 数据传输时的参考时序。速率为 10 Mb/s 时标称频率为 2.5 MHz；速率为 100 Mb/s 时标称频率为 25 MHz。
- TX_EN：发送使能信号，该信号表示 MAC 当前正针对 MII 发送半字节。该信号必须与报头的前半字节进行同步 (TX_CLK)，并在所有待发送的半字节均发送到 MII 时必须保持同步。
- TXD[3:0]：数据发送信号。

  TXD 是 4 个一组的数据信号，由 MAC 子层同步驱动，在 TX_EN 信号有效时才为有效信号（有效数据）。TXD[0] 为最低有效位，TXD[3] 为最高有效位。禁止 TX_EN 时，发送数据不会对 PHY 产生任何影响。
- CRS：载波侦听信号。

  当发送或接收介质处于非空闲状态时，由 PHY 使能该信号。发送和接收介质均处于空闲状态时，由 PHY 禁止该信号。PHY 必须确保 CS 信号在冲突条件下保持有效状态。该信号无需与 Tx 和 Rx 时钟保持同步。在全双工模式下，该信号的状态对 MAC 子层无任何意义。
- COL：冲突检测信号

  检测到介质上存在冲突后，PHY 必须立即使能该信号，并且只要存在冲突条件，该信号就必须保持有效状态。该信号无需与 Tx 和 Rx 时钟保持同步。在全双工模式下，该信号的状态对 MAC 子层无任何意义。

- RXD[3:0]：数据接收信号

  RXD 是 4 个一组的数据信号，由 PHY 同步驱动，在 RX_DV 信号有效时才为有效信号（有效数据）。RXD[0] 为最低有效位，RXD[3] 为最高有效位。当 RX_EN 禁止、RX_ER 使能时，特定的 RXD[3:0] 值用于传输来自 PHY 的特定信息（请参见*表 510*）。

- RX_DV：接收数据有效信号

  该信号表示 PHY 当前正针对 MII 接收已恢复并解码的半字节。该信号必须与恢复帧的头半字节进行同步 (RX_CLK)，并且一直保持同步到恢复帧的最后半字节。该信号必须在最后半字节随后的第一个时钟周期之前禁止。为了正确接收帧，RX_DV 信号必须在时间范围上涵盖要接收的帧，其开始时间不得迟于 SFD 字段出现的时间。

- RX_ER：接收错误信号

  该信号必须保持一个或多个周期 (RX_CLK)，从而向 MAC 子层指示在帧的某处检测到错误。该错误条件必须通过 RX_DV 的有效情况进行验证，如*表 510* 所示。

表 510. RX 接口信号编码

RX_DV	RX_ERR	RXD[3:0]	说明
0	0	0000 到 1111	正常帧间
0	1	0000	正常帧间
0	1	0001 到 1101	保留
0	1	1110	错误载波检测
0	1	1111	保留
1	0	0000 到 1111	正常数据接收
1	1	0000 到 1111	数据接收出现错误

58.6.3 精简介质独立接口 (RMII)

精简介质独立接口 (RMII) 规范降低了以太网 PHY 与 STM32 MCU 之间的引脚数。根据 IEEE 802.3u，MII 包括 16 个数据和控制信号的引脚。RMII 规范将引脚数减少为 7 个。

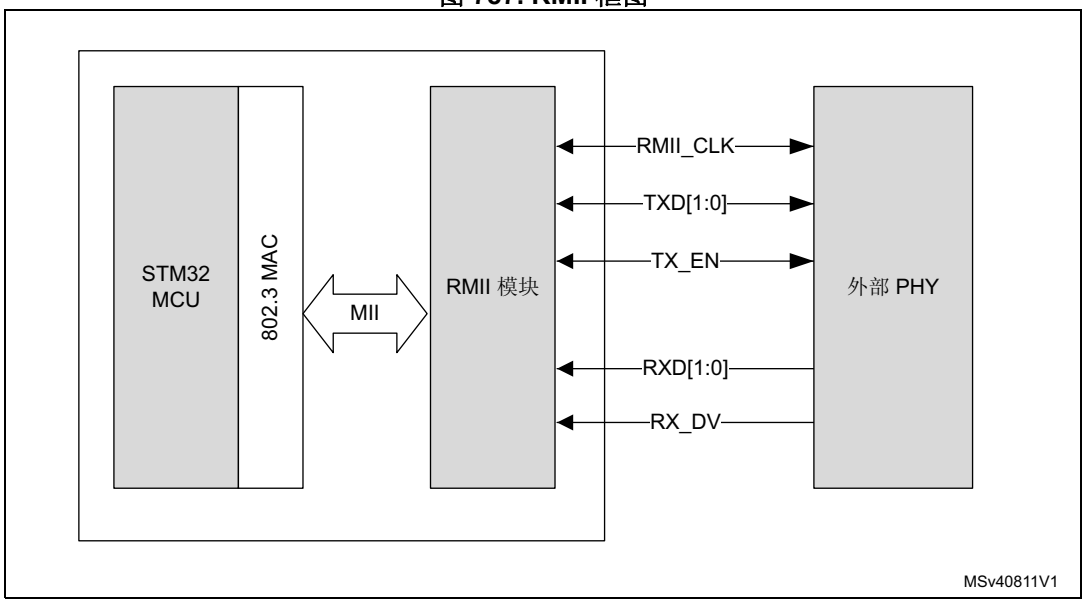
作为以太网外设的一部分，RMII 模块是 MAC 输出端的实例化对象。这有助于将 MAC 的 MII 转换为 RMII。RMII 模块具有以下特性：

- 支持 10 Mbps 和 100 Mbps 运行速率。不支持 1000 Mbps 运行速率。
- 通过在外部提供两个参考时钟来实现独立的 2 位宽发送和接收路径。

RMII 框图

*图 787：RMII 框图*显示了 RMII 模块相对于 MAC 和 RMII PHY 的位置。将 RMII 模块置于 MAC 之前，以将 MII 信号转换为 RMII 信号。

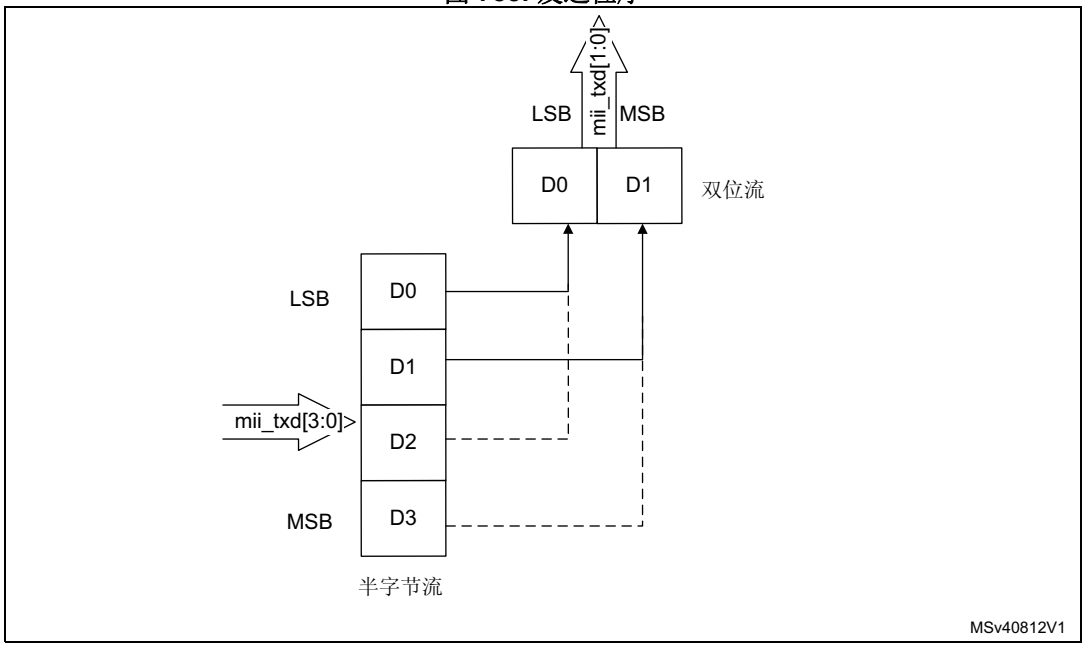
图 787. RMII 框图



发送位序

来自 MII 接口的每个半字节都在 RMII 接口上发送，一次发送双位，双位的发送顺序如*图 788：发送位序*所示。首先发送位序较低的位（D1 和 D0），再发送位序较高的位（D2 和 D3）。

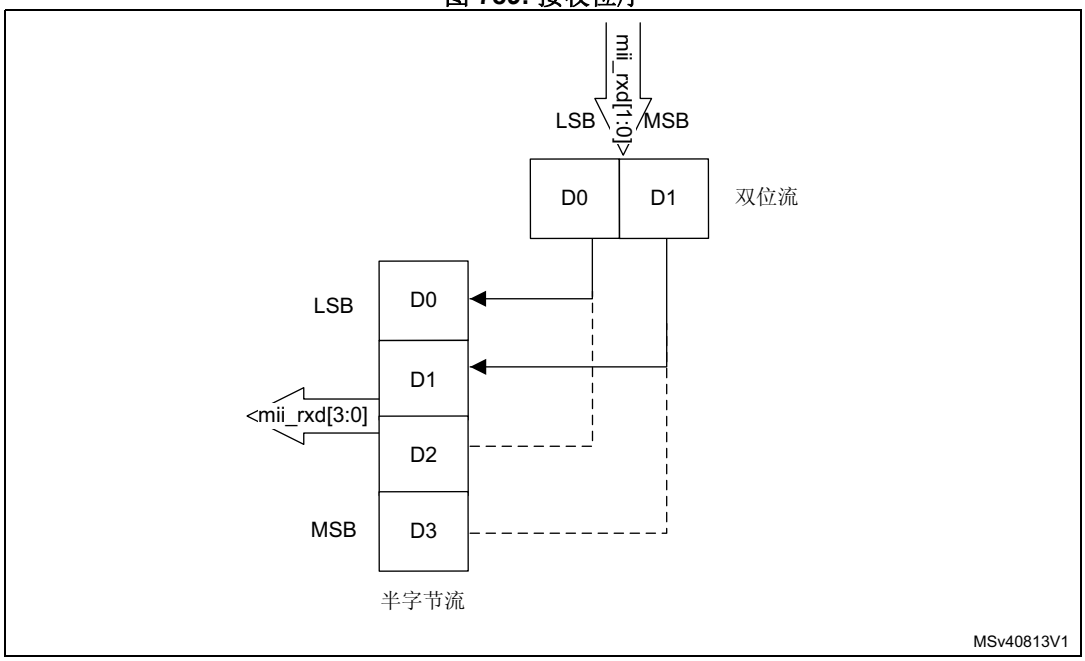
图 788. 发送位序



接收位序

每个半字节都从接收自 RMII 接口的双位发送到 MII 接口，半字节发送顺序如*图 789：接收位序*所示。先接收位序较低的位（D0 和 D1），再接收位序较高的位（D2 和 D3）。

图 789. 接收位序



58.7 以太网低功耗模式

58.7.1 节能型以太网

节能型以太网 (EEE) 是一种工作模式，可使 MAC 子层和几个物理层一起工作在低功耗空闲 (LPI) 模式下。EEE 模式支持 100 Mbps 运行速率。

当不存在要发送和接收的数据时，LPI 模式允许通过关闭一些通信器件功能来实现节能。MAC 会控制系统进入或退出 LPI 模式，并通知 PHY 接口。

EEE 指定协商方法，链路伙伴可使用指定的方法来决定是否支持 EEE，然后可选择一组对于两个器件通用的参数。

注： *将 MAC 配置为使用单一 PHY 接口 RMII 时，不支持 EEE 功能。即使 MAC 支持多个 PHY 接口，也只应在 MAC 使用 MII 接口时才激活 EEE 模式。*

仅全双工模式支持 LPI 模式。

发送路径功能

在发送路径中，软件必须将 *LPI 控制状态寄存器 (ETH_MACLCSR)* 的 LPIEN 位置 1，以请求 MAC 停止发送并启用 LPI 协议。

要使 PHY 退出 LPI 状态，MAC 应执行以下任务：

1. 停止发送 LPI 模式并开始发送 IDLE 模式。
2. 启动 LPI TW 定时器：

   MAC 需在为 PHY 指定的唤醒时间结束之后才能开始发送。自动协商的唤醒间隔在 *LPI 控制状态寄存器 (ETH_MACLCSR)* 的 TWT 字段中编程。
3. 更新 LPI 退出状态（*LPI 控制状态寄存器 (ETH_MACLCSR)* 的 TLPIEX 位），然后生成中断。

有关 LPI 模式的编程指南，请参见*进入和退出 Tx LPI 模式*一节。

在 Tx 路径中自动进入/退出 LPI 模式

MAC 发送器经过编程后，可在长时间处于 IDLE 状态时自动进入 LPI 模式，并在有数据包要传送时自动退出 LPI 模式。这些模式通过 *LPI 控制状态寄存器 (ETH_MACLCSR)* 进行使能和控制。

如果 *LPI 控制状态寄存器 (ETH_MACLCSR)* 的 LPITXA（位 19）和 LPITXEN（位 16）置 1，MAC 发送器会在 MAC 发送路径（包括 MTL 层和 DMA 层）处于空闲状态时进入 LPI IDLE 状态。Tx 路径中的任一功能（DMA、MTL 或 MAC）因触发数据包传送而变为非空闲状态后，MAC 发送器会立即退出 LPI IDLE 状态并将 LPITXEN 位清零。

此外，如果位 [20] (LPIATE) 也置 1，则只有在发送路径保持空闲状态（无活动）的时间达到 *LPI 进入定时器寄存器 (ETH_MACLETR)* 中的值所指示的时间时，MAC 发送器才会进入 LPI IDLE 状态。在这种模式下，只要有任一功能变为非空闲状态，MAC 发送器就会立即退出 LPI IDLE 状态。但是，LPITXEN 位不会被清零，而是继续保持有效状态，以便在 MAC 再次空闲时无需任何软件干预即可重新进入 LPI IDLE 状态。

LPIATE 和 LPITXA 位均清零时，可通过编程 LPITXEN 位直接控制进入和退出 LPI IDLE 状态。

接收路径功能

在接收路径中，当 PHY 从链路伙伴接收到进入 LPI 状态的信号时，PHY 会进入 LPI 模式，MAC 会更新 *LPI 控制状态寄存器 (ETH_MACLCSR)* 的 RLPIEN 位并立即生成中断。

当 PHY 从链路伙伴接收到退出 LPI 状态的信号时，PHY 会退出 LPI 模式，MAC 会更新 *LPI 控制状态寄存器 (ETH_MACLCSR)* 的 RLPIEX 位 并立即生成中断。

LPI 中断

当 Tx 或 Rx 端进入或退出 LPI 状态时，MAC 会生成 LPI 中断。通过对 *LPI 控制状态寄存器 (ETH_MACLCSR)* 进行读操作，可将 LPI 中断清除。

边带信号 lpi_intr_o 与该中断一起生成。该信号供唤醒机制使用。它与 pmt_intr_o 信号（请参见 *PMT 中断*一节）进行或运算，并连至 EXTI 外设（线 86）。

58.7.2 电源管理

电源管理 (PMT) 模块支持接收网络（远程）唤醒数据包和魔术数据包。PMT 模块可为 MAC 接收到的远程唤醒数据包和魔术数据包生成中断。

在 PMT 模块中使能掉电模式时，MAC 将丢弃所有接收到的数据包并且不会将任何数据包转发给 RxFIFO 或应用。仅当接收到魔术数据包或远程唤醒数据包且使能相应检测时，MAC 才会退出掉电模式。

PMT 模块可用于 MAC 的接收路径中。两种电源管理数据包（远程唤醒数据包和魔术数据包）均可供选择。可将 *PMT 控制状态寄存器 (ETH_MACPCSR)* 的 RWKPKTEN 和 MGKPKTEN 位置 1 以生成电源管理事件。

PMT 模块寄存器如下：

- *PMT 控制状态寄存器 (ETH_MACPCSR)*
- *删除唤醒数据包过滤寄存器 (ETH_MACRWKPFR)*

远程唤醒数据包检测

当 MAC 处于睡眠模式并已使能 *PMT 控制状态寄存器 (ETH_MACPCSR)* 中的远程唤醒位时，MAC 可在接收到远程唤醒数据包后恢复正常工作。

远程唤醒帧过滤寄存器

PMT 模块支持 16 种可编程过滤器，从而支持不同的接收数据包模式。如果传入数据包通过过滤命令的地址过滤，并且过滤器 CRC-16 与传入模式的 CRC 相匹配，则 MAC 会将相应数据包识别为唤醒数据包。

删除唤醒数据包过滤寄存器 (ETH_MACRWKPFR) 定义了过滤管理：

- 过滤器字节屏蔽：确定必须检测数据包的哪些字节
- 过滤器偏移：确定要检测的数据包的偏移
- 过滤器 CRC-16

远程唤醒 CRC 模块确定与过滤器 CRC-16 进行比较的 CRC 值。仅检查远程唤醒数据包是否存在长度错误、FCS 错误、dribble 位错误、接收错误和冲突。此外，还会对远程唤醒数据包进行检查，以确保其不是过短数据包。即使远程唤醒数据包长度超过 512 字节，但只要数据包的 CRC 值有效，数据包便有效。PMT 控制和状态寄存器中的远程唤醒数据包检测会针对每个接收到的远程唤醒数据包进行更新。向应用程序发送 PMT 中断会触发对 PMT 控制和状态寄存器的读操作，从而确定是否已接收到远程唤醒数据包。

魔术数据包检测

魔术数据包基于一种特殊的方法，这种方法使用 AMD 公司的魔术数据包技术对网络上处于睡眠模式下的器件上电。MAC 接收称为魔术数据包的特定信息包，此信息包的目标地址是网络上的节点。

MAC 仅检查被寻址到 MAC 或多播地址（包括广播地址）的魔术数据包，以确定这些数据包是否满足唤醒要求。对通过地址过滤（单播或多播（包括广播）地址）的魔术数据包进行检查，确定其是否符合远程唤醒数据包的数据格式，即 6 个字节的数据所有位为全 1 的数据包加上重复出现 16 次的单播 MAC 地址（与 MAC 地址 0 中的值匹配）。

应用程序通过向 *PMT 控制状态寄存器 (ETH_MACPCSR)* 的 MGKPKTEN 位写入 1 来使能魔术数据包唤醒。PMT 模块持续监视目标地址为对应于指定魔术数据包模式的节点的每个数据包。检查每个接收数据包的目标地址和源地址字段之后是否为 48'hFF_FF_FF_FF_FF_FF 形式。之后，PMT 模块检查数据包是否存在重复 16 次且没有任何断开或中断的 MAC 地址。如果重复 16 次的地址中存在中断，则 PMT 模块会再次扫描传入数据包中是否存在 48'hFF_FF_FF_FF_FF_FF 形式。这 16 次重复可位于数据包中的任何位置，但必须在同步数据流后面 (48'hFF_FF_FF_FF_FF_FF)。此外，只要检测到重复 16 次的 MAC 地址，器件就可以接受多播数据包。如果 8'hFF 的重复次数大于 6 次，PMT 模块将在 8'hFF 的最后 6 次重复之后，检查是否存在重复 16 次且没有任何断开或中断的 MAC 地址。

如果节点的 MAC 地址为 48'h00_11_22_33_44_55，则 MAC 扫描的数据序列如下：

目标地址源地址 .....FF FF FF FF FF FF
00 11 22 33 44 55 00 11 22 33 44 55 00 11 22 33 44 55
00 11 22 33 44 55 00 11 22 33 44 55 00 11 22 33 44 55
00 11 22 33 44 55 00 11 22 33 44 55 00 11 22 33 44 55
00 11 22 33 44 55 00 11 22 33 44 55 00 11 22 33 44 55
…CRC

MAC 仅检查远程唤醒数据包是否存在长度错误、FCS 错误、dribble 位错误、接收或冲突错误。此外，还会对远程唤醒数据包进行检查，以确保其不是过短数据包。即使远程唤醒数据包长度超过 512 字节，但只要数据包的 CRC 值有效，MAC 就会将其视为有效数据包。

PMT 控制状态寄存器 (ETH_MACPCSR) 中的魔术数据包检测会针对接收到的魔术数据包进行更新。向应用程序发送 PMT 中断会触发对 *PMT 控制状态寄存器 (ETH_MACPCSR)* 的读操作，从而确定是否已接收到魔术数据包。

PMT 中断

接收到有效远程唤醒数据包时，PMT 中断信号会置为有效。

软件复位*删除唤醒数据包过滤寄存器 (ETH_MACRWKPFR)* 中的 PWRDWN 位时，MAC 会退出掉电模式。该事件不会生成 PMT 中断。

对于 EEE 模式，边带信号 pmt_intr_o 与唤醒中断一起生成。它与 lpi_intr_o 信号（请参见 *LPI 中断*一节）进行或运算，并连至 EXTI 外设（线 86）。

58.7.3 掉电和唤醒序列

推荐的掉电和唤醒序列如下：

1. 禁止发送 DMA，并等待所有之前的数据包发送完成。接收到发送中断（ETH_DMACSR 寄存器的位 0）时，便可检测到这些发送过程。
2. 通过将 MAC 配置寄存器 (ETH_MACCR) 中的相应位（0 和 1）清零来禁止 MAC 发送器和 MAC 接收器。
3. 等待接收 DMA 清空从 Rx FIFO 到系统存储器的所有数据包。可通过读取 DMA (ETH_DMADSR) 和 MTL（ETH_MTLTxQDR 和 ETH_MTLRxQDR）中的相应调试寄存器位来实现上述过程。
4. 通过相应地配置 PMT 寄存器 ETH_MACPCSR 和 ETH_MACRWKPFR 来使能掉电模式。
5. 使能 MAC 接收器并进入掉电模式。
6. 接收到有效远程唤醒数据包时，以太网外设会将 pmt_intr_o 信号置为有效并退出掉电模式。
7. 读取 PMT 状态寄存器以清除中断，然后使能系统中的其他模块并恢复正常工作。

58.8 以太网中断

以太网外设可生成一个中断信号 (eth_sbd_intr_it)。该信号可作为各种事件的结果而生成。这些事件被捕获到状态寄存器中，并且每个中断源都配有一个中断使能位，因此事件的中断信号只有在相应的中断使能位置 1 时才会置为有效。

中断状态和相应的使能寄存器以层级结构进行组织，以便软件更加轻松地快速遍历和识别中断事件源。触发中断时，*中断状态寄存器 (ETH_DMAISR)* 寄存器将作为第一级指示，指示中断事件源对应的主要模块。该寄存器为只读寄存器，其中包含与每个 DMA 通道（TX & RX 对）、MTL 和 MAC 相对应的位。软件应用程序随后必须读取置 1 的位所对应的以下寄存器中的一个（或多个）：

- ETH_DMACSR：通道状态（请参见*通道状态寄存器 (ETH_DMACSR)*）
- ETH_MTLISR：中断状态（请参见*中断状态寄存器 (ETH_MTLISR)*）
- ETH_MACISR：中断状态（请参见*中断状态寄存器 (ETH_MACISR)*）

58.8.1 DMA 中断

中断寄存器说明

ETH_DMACSR：通道状态寄存器（请参见*通道状态寄存器 (ETH_DMACSR)*）捕获 TxDMA 和 RxDMA 通道的所有中断事件。ETH_DMACIER：通道中断使能寄存器（请参见*通道中断使能寄存器 (ETH_DMACIER)*）包含对应于每个中断事件的使能位。

DMA 通道中有两组中断，即正常中断和异常中断。它们分别由 ETH_DMACSR 寄存器的位 [15:14] 进行指示。正常组对应于正常数据包传送期间发生的事件（TI：发送中断，RI：接收中断，TBU：发送缓冲区不可用），而异常中断事件则对应于错误事件。

中断不会形成队列。如果在驱动程序对上一个事件做出响应之前再次发生相同的中断事件，则不会再生成中断。对于同时发生的多个事件，只生成一次中断。驱动程序必须扫描*中断状态寄存器 (ETH_DMAISR)* 以明确中断原因，并清除相应状态寄存器中的中断源。只有在*中断状态寄存器 (ETH_DMAISR)* 的所有位均清零时才会清除中断。

定期调度发送和接收中断

出于系统吞吐量性能的原因，最好不要针对 DMA 传送的每个数据包都生成中断（RI 和 TI）。通过以太网外设可使用以下两种方法灵活地定期调度中断：

1. 每当要发送的数据包达到“所需”数量时，便将发送描述符中的完成时中断位（*表 514：TDES2 正常描述符（读取格式）*中的 TDES2[31]）置 1。
2. 类似地，仅以特定的间隔将接收描述符中的 IOC（*表 527：RDES3 正常描述符（读取格式）*中的 RDES3[30]）置 1。这样，只有在完成将接收到的数据包传送到系统存储器，并且用于该数据包传送的所有描述符均将 IOC 位置 1 时，才会生成 RI 事件。

此外，还提供了中断定时器（ETH_DMACRxIWTR：通道 Rx 中断看门狗定时器），以供灵活控制和定期调度接收中断。当此中断定时器使用非零值编程时，只要 Rx DMA 完成将接收的数据包传送到系统存储器并且未触发接收中断（因为相应的完成中断 IOC 位，即*表 527：RDES3 正常描述符（读取格式）*中的 RDES3[30] 未置 1），此定时器即会被激活。当此定时器按照编程值结束计时时，如果使能了 ETH_DMACIER 寄存器（请参见*通道中断使能寄存器 (ETH_DMACIER)*）中的相应 RIE，则 RI 位会置 1 并会引发中断。如果针对描述符的 IOC 已置 1 的数据包传送将 RI 置 1，则定时器会停止并清零。如果下一个数据包传送完成后未生成 RI 事件，则会自动重新激活定时器。

通道传送完成中断

发送传送完成中断 (TI) 和接收传送完成中断 (RI) 通过通道状态寄存器（*通道状态寄存器 (ETH_DMACSR)*）进行指示。只要 TX DMA 通道关闭了 IOC 位置 1（完成时中断 - TDES2[31]）的描述符，TI 位便置 1。类似地，只要 Rx DMA 通道关闭了 LD 位置 1 的描述符，RI 位便置 1，并且用于传送该数据包的任何描述符中的 IOC 位均置 1（完成时中断使能 - RDES3[30]）。

只有在通道中断使能寄存器（*通道中断使能寄存器 (ETH_DMACIER)*）中使能了相应的中断时，才会针对传送完成中断将中断信号置为有效。

RI/TI 中断的行为随 ETH_DMAMR 寄存器（*DMA 模式寄存器 (ETH_DMAMR)*）中的 INTM 字段（位 [17:16]）设置而变化。*表 511* 对传送完成中断的行为进行了说明。

表 511. 传送完成中断行为

中断模式	TI/RI 和中断信号的行为
INTM=0	检测到传送完成事件时，TI/RI 状态信号会被置 1。 软件驱动程序向这些位写入 1 时，这些位会被清零。 如果还在 ETH_DMACIER 寄存器中使能了相应的中断，则中断信号会被置为有效。
INTM=1	如上所述，TI/RI 置 1。不过，对于任一 RI/TI 事件，都不会触发中断。
INTM=2	RI/TI 状态位会在检测到传送完成事件时置 1，并且会在软件驱动程序通过写入 1 来将其清零时复位。不过，如果在软件将其清除（处理）之前检测到另一个传送完成事件，则会自动再次将这些状态位置 1。但是，并非基于 TI/RI 生成中断。

58.8.2 MTL 中断

MTL 中断事件与 DMA 中的事件结合用于生成中断信号。

中断状态寄存器 (ETH_MTLISR) 报告负责事件的队列号。ETH_MTLQICSR：应读取队列中断控制状态以获取事件说明。

默认情况下使能 MTL 中断。可通过将 *中断状态寄存器 (ETH_MTLISR)* 中的相应屏蔽位置 1，以阻止各个事件将中断置为有效。

MTL 中断信号由以下事件之一驱动：

- 接收队列上溢中断
- 发送队列下溢

58.8.3 MAC 中断

MAC 中断事件与 DMA 中的事件结合用于生成中断信号。

MAC 中断为电平型中断，即在被应用程序或软件清除之前始终保持有效（高电平）。

中断状态寄存器 (ETH_MACISR) 介绍了可能导致 MAC 生成中断的事件。默认情况下 MAC 中断处于使能状态。可通过将 *中断状态寄存器 (ETH_MACISR)* 中的相应屏蔽位置 1，以阻止各个事件将中断置为有效。

中断寄存器位仅指示报告事件的模块。必须读取相应的状态寄存器和其他寄存器才能清除中断。

MAC 中断信号由以下事件之一驱动：

- 接收状态中断
- 发送状态中断
- 时间戳中断状态
- MMC 中断状态
  - MMC 接收校验和减荷中断状态
  - MMC 发送中断状态
  - MMC 接收中断状态
- LPI 中断状态
- PMT 中断状态
- PHY 中断

注： *以下两个边带信号与 LPI 和 PMT 中断一起生成：lpi_intr_o 和 pmt_intr_o。这两个信号用于在 EXTI 级进行唤醒事件检测。*

58.9 以太网编程模型

本章提供了按正确顺序初始化 DMA 或 MAC 寄存器的相关说明。其中涵盖以下部分：

- DMA 初始化（请参见*第 58.9.1 节*）
- MTL 初始化（请参见*第 58.9.2 节*）
- MAC 初始化（请参见*第 58.9.3 节*）
- 执行正常接收和发送操作（请参见*第 58.9.4 节*）
- 停止和开始发送（请参见*第 58.9.5 节*）
- 关于 MII 链路状态转换的编程指南（请参见*第 58.9.6 节*）
- 关于 IEEE 1588 时间戳的编程指南（请参见*第 58.9.7 节*）
- 关于节能型以太网的编程指南（请参见*第 58.9.8 节*）
- 关于每秒脉冲数 (PPS) 灵活的输出的编程指南（请参见*第 58.9.9 节*）
- 关于 TSO 的编程指南（请参见*第 58.9.10 节*）
- 关于接收端 VLAN 过滤的编程指南（请参见*第 58.9.11 节*）

58.9.1 DMA 初始化

要初始化 DMA，请完成以下步骤：

1. 提供软件复位，以复位所有 MAC 内部寄存器和逻辑（*DMA 模式寄存器 (ETH_DMAMR)* 的位 0）。
2. 等待完成复位过程（轮询 *DMA 模式寄存器 (ETH_DMAMR)* 的位 0，当复位操作完成时该位会清零）。
3. 编程以下字段以初始化：
   a) AAL
   b) 固定突发或未定义突发
   c) 突发模式值（如果是 AHB 总线接口）。
   d) 如果使能固定长度值，请选择 AXI 总线上可能的最大突发长度（位 [7:1]）
4. 创建发送和接收描述符列表。此外，还应确保接收描述符为 DMA 所有（将 TDES3/RDES3 描述符的位 31 置 1）。有关描述符的更多信息，请参见*第 58.10 节：描述符*。

注： *从环起始到环结束的描述符地址不应超过 4GB 边界。*

5. 编程 ETH_DMACTxRLR 和 ETH_DMACRxRLR 寄存器（请参见*通道 Tx 描述符环长度寄存器 (ETH_DMACTxRLR)* 和*通道 Rx 描述符环长度寄存器 (ETH_DMACRxRLR)*）。编程的环长度必须至少为 4。
6. 使用发送和接收描述符的基址初始化接收和发送描述符列表地址（*通道 Tx 描述符列表地址寄存器 (ETH_DMACTxDLAR)*，*通道 Rx 描述符列表地址寄存器 (ETH_DMACRxDLAR)*）。此外，还应编程发送和接收尾指针寄存器，这类寄存器会将可用描述符通知给 DMA（请参见*通道 Tx 描述符尾指针寄存器 (ETH_DMACTxDTPR)* 和*通道 Rx 描述符尾指针寄存器 (ETH_DMACRxDTPR)*）。
7. 编程 ETH_DMACCR、ETH_DMACTxCR 和 ETH_DMACRxCR 寄存器（请参见*通道控制寄存器 (ETH_DMACCR)* 和*通道发送控制寄存器 (ETH_DMACTxCR)*）以配置以下参数：DMA 触发的最大突发长度 (PBL)、描述符跳过长度、用于 TxDMA 的 OSP 以及用于 RxDMA 的 RBSZ。

8. 通过编程 ETH_DMACIER 寄存器（请参见*通道中断使能寄存器 (ETH_DMACIER)*）使能中断。
9. 通过将*通道接收控制寄存器 (ETH_DMACRxCR)* 的 SR （位 0）置 1 以及将 ETH_DMACTxCR （请参见*通道发送控制寄存器 (ETH_DMACTxCR)*）的 ST （位 0）置 1 来启动接收和发送 DMA。

58.9.2 MTL 初始化

要初始化 MTL 寄存器，请完成以下步骤：

1. 在 ETH_MTLTxQOMR（请参见 *Tx 队列工作模式寄存器 (ETH_MTLTxQOMR)*）中编程以下字段，以初始化发送工作模式。
   a) 发送存储转发 (TSF) 或发送阈值控制 (TTC)（使用阈值模式时）。
   b) 将发送队列使能 (TXQEN) 编程为值 2‘b10，以使能发送队列 0。
   c) 发送队列大小 (TQS)。
2. 在 ETH_MTLRxQOMR 寄存器（请参见 *Rx 队列工作模式寄存器 (ETH_MTLRxQOMR)*）中编程以下字段，以初始化接收工作模式：
   a) 接收存储转发 (RSF) 或 RTC（使用阈值模式时）。
   b) 流控制激活和取消激活阈值，用于 MTL 接收 FIFO（RFA 和 RFD）。
   c) 错误数据包和过小的良好数据包转发使能（FEP 和 FUP）。
   d) 接收队列大小 (RQS)。

58.9.3 MAC 初始化

在完成 DMA 初始化之后，可执行以下 MAC 初始化操作。如果在配置 DMA 之前已完成 MAC 初始化，则只有在 DMA 激活后才可使能 MAC 接收器（以下序列中的最后一步）。否则，接收到的帧将填充 Rx FIFO 并引发上溢。

1. 提供 MAC 地址寄存器：*地址 x 低位寄存器 (ETH_MACAxLR)* 和*地址 0 高位寄存器 (ETH_MACA0HR)*。如果在配置中使能了多个 MAC 地址（最多 3 个附加地址），请相应编程 MAC 地址。
2. 在*数据包过滤控制寄存器 (ETH_MACPFR)* 中编程以下字段，从而为传入的帧设置相应过滤器：
   a) 接收所有。
   b) 混合模式。
   c) 散列或完美过滤器。
   d) 单播、多播、广播和控制帧过滤设置。
3. 在 *Tx 队列流控制寄存器 (ETH_MACQTxFCR)* 中编程以下字段以实现适当的流控制：
   a) 暂停时间和其他暂停帧控制位。
   b) 发送流控制位。
   c) 流控制忙碌。
4. 根据需要编程*中断使能寄存器 (ETH_MACIER)*，前提是适用于您的配置。
5. 在*工作模式配置寄存器 (ETH_MACCR)* 寄存器中编程相应字段。
   例如：发送和 jabber 禁止时的数据包间隙。
6. 将*工作模式配置寄存器 (ETH_MACCR)* 寄存器中的位 0 和位 1 置 1，以启动 MAC 发送器和接收器。

58.9.4 执行正常接收和发送操作

为实现正常操作，请完成以下步骤：

1. 对于正常发送和接收中断，请读取中断状态。然后，通过读取主机所拥有的描述符的状态轮询描述符（发送或接收）。
2. 将描述符设为相应值。确保发送和接收描述符为 DMA 所有，以恢复数据的发送和接收。
3. 如果描述符并非为 DMA 所有（或无可用描述符），则 DMA 会进入挂起状态。可通过释放描述符并对 ETH_DMACTxDTPR（请参见*通道 Tx 描述符尾指针寄存器 (ETH_DMACTxDTPR)*）和 ETH_DMACRxDTPR（请参见*通道 Rx 描述符尾指针寄存器 (ETH_DMACRxDTPR)*）执行写操作，以恢复发送或接收。
4. 在调试模式下，当前主机发送器或接收器描述符地址指针的值可在 ETH_DMACCATxDR 和 ETH_DMACCARxDR 寄存器（请参见*通道当前应用程序发送描述符寄存器 (ETH_DMACCATxDR)* 和*通道当前应用程序接收描述符寄存器 (ETH_DMACCARxDR)*）中进行读取。
5. 在调试模式下，当前主机发送缓冲区地址指针和接收缓冲区地址指针的值可在 ETH_DMACCATxDR 和 ETH_DMACCARxDR 寄存器（请参见*通道当前应用程序发送描述符寄存器 (ETH_DMACCATxDR)* 和*通道当前应用程序接收描述符寄存器 (ETH_DMACCARxDR)*）中进行读取。

58.9.5 停止和开始发送

要将发送暂停一段时间，请完成以下步骤：

1. 通过将 ETH_DMACTxCR 寄存器（请参见）的位 0 (ST) 清零来禁止发送 DMA（如适用）。
2. 等待完成之前的所有帧发送过程。可通过读取 ETH_MTLTxQDR 寄存器的相应位（TRCSTS 不为 01，TXQSTS = 0）对此进行检查。
3. 通过将*工作模式配置寄存器 (ETH_MACCR)* 寄存器的位 0 (RE) 和位 1 (TE) 清零来禁止 MAC 发送器和 MAC 接收器。
4. 确保 Rx FIFO 中的数据被传送到系统存储器（通过读取 *Tx 队列调试寄存器 (ETH_MTLTxQDR)* 的相应位，PRXQ= 0 且 RXQSTS= 00）后，禁止接收 DMA（如适用）。
5. 确保 Tx 队列和 Rx 队列均为空（*Tx 队列调试寄存器 (ETH_MTLTxQDR)* 中的 TXQSTS 为 0 且 RXQSTS 被设为 0）。
6. 如要重新启动操作，请首先启动 DMA，然后再使能 MAC 发送器和接收器。

58.9.6 关于 MII 链路状态转换的编程指南

链路断开时发送时钟和接收时钟处于运行状态

如果在发送和接收时钟处于运行状态时链路断开，请完成以下步骤：

1. 通过将*通道控制寄存器 (ETH_DMACCR)* 的位 0 (ST) 清零来禁止发送 DMA（如适用）。
2. 通过将*工作模式配置寄存器 (ETH_MACCR)* 的位 2 (RE) 清零来禁止 MAC 接收器。
3. 等待完成之前的所有帧发送过程。可通过读取 *Tx 队列调试寄存器 (ETH_MTLTxQDR)* 的相应位（TRCSTS 不为 01）对此进行检查。

   或

   刷新 Tx FIFO 以加快清空操作。
4. 通过将*工作模式配置寄存器 (ETH_MACCR)* 寄存器的位 1 (TE) 清零来禁止 MAC 发送器。
5. 确保 Tx 和 Rx 队列均为空（*Tx 队列调试寄存器 (ETH_MTLTxQDR)* 中的 TXQSTS 和 *Rx 队列调试寄存器 (ETH_MTLRxQDR)* 中的 RXQSTS 均被设为 0）。
6. 链路接通后，读取 PHY 寄存器以识别最新配置，并相应地对 MAC 寄存器进行编程。
7. 如要重新启动相应操作，请首先启动 Tx DMA，然后再使能 MAC 发送器和接收器。

   Rx DMA 无需使能：因为接收器已被禁止，Rx FIFO 中没有数据。

链路断开时发送时钟和接收时钟处于停止状态

如果链路断开且发送时钟和接收时钟处于停止状态，请完成以下步骤：

1. 通过将*工作模式配置寄存器 (ETH_MACCR)* 中的 RE 位和 TE 位清零来禁止 MAC 发送器和接收器。由于不存在时钟，因此上述操作不会立即生效。
2. 等待链路接通以及时钟恢复。
3. 如果在发送 / 接收时钟停止时正在进行帧传送，请等待至所有部分帧全部完成传送。可通过读取 *调试寄存器 (ETH_MACDR)* 进行检查（所有位均应设为 0）。MAC 发送器停止时，一些旧数据包可能仍保留在 TXFIFO 中。
4. 读取 PHY 寄存器以识别最新工作模式，并相应地对 MAC 寄存器进行编程。
5. 通过将 RE 位和 TE 位置 1 来重启 MAC 发送器和接收器。

58.9.7 关于 IEEE 1588 时间戳的编程指南

初始化系统时间生成

可通过将*时间戳控制寄存器 (ETH_MACTSCR)* 的位 0 置 1 来使能时间戳功能。不过，必须在该位置 1 后初始化时间戳计数器。要执行外设初始化，请完成以下步骤：

1. 通过将 *中断使能寄存器 (ETH_MACIER)* 的位 16 清零来屏蔽时间戳触发中断。
2. 将*时间戳控制寄存器 (ETH_MACTSCR)* 的位 0 置 1 以使能时间戳。
3. 根据 PTP 时钟频率编程*亚秒增量寄存器 (ETH_MACSSIR)*。
4. 如果使用精密校准方法，请编程*时间戳加数寄存器 (ETH_MACTSAR)*，并将*时间戳控制寄存器 (ETH_MACTSCR)* 的位 5 置 1。
5. 轮询*时间戳控制寄存器 (ETH_MACTSCR)*，直到位 5 清零。
6. 编程*时间戳控制寄存器 (ETH_MACTSCR)* 的位 1 以选择精密更新方法（如果需要）。
7. 使用适当的时间值编程*系统时间秒更新寄存器 (ETH_MACSTSUR)* 和*系统时间纳秒更新寄存器 (ETH_MACSTNUR)*。
8. 将*时间戳控制寄存器 (ETH_MACTSCR)* 中的位 2 置 1。

   用时间戳更新寄存器中写入的值初始化时间戳计数器后，时间戳计数器便开始运行。如果需要一步时间戳：

   a) 通过编程 TDES3 上下文描述符的位 27 来使能一步时间戳。

   b) 编程*时间戳入站不对称校准寄存器 (ETH_MACTSIACR)* 以更新 PDelay_Req PTP 消息中的校准字段。
9. 使能 MAC 接收器和发送器以使时间戳功能正常运行。

注： *如果通过将*时间戳控制寄存器 (ETH_MACTSCR)* 的位 0 清零来禁止时间戳操作，请重复所有上述步骤以重新启动时间戳操作。*

系统时间校准

要一次性（粗略校准方法）同步或更新系统时间，请完成以下步骤：

1. 在时间戳更新寄存器（*系统时间秒更新寄存器 (ETH_MACSTSUR)* 和*系统时间纳秒更新寄存器 (ETH_MACSTNUR)*）中设置偏移（正偏移或负偏移）。
2. 将*时间戳控制寄存器 (ETH_MACTSCR)* 的位 3 (TSUPDT) 置 1。

   当 TSUPDT 位清零时，时间戳更新寄存器中的值将加到系统时间中或者从系统时间中减去。

要同步或更新系统时间以减少系统时间抖动（精密校准方法），请完成以下步骤：

1. 借助*系统时间寄存器模块*一节中介绍的算法，计算预期的系统时间递增或递减速率。
2. 使用新值更新*时间戳加数寄存器 (ETH_MACTSAR)*，并将*时间戳控制寄存器 (ETH_MACTSCR)* 寄存器的位 5 置 1。
3. 等待加数寄存器中的新值生效。这可以通过在系统时间达到目标值后，使能时间戳触发中断来实现。
4. 在 *PPS 目标时间秒寄存器 (ETH_MACPPSTTSR)* 和 *PPS 目标时间纳秒寄存器 (ETH_MACPPSTTNR)* 中编程所需目标时间。
5. 在*中断使能寄存器 (ETH_MACIER)* 的位 12 中使能时间戳中断。
6. 将寄存器*时间戳控制寄存器 (ETH_MACTSCR)* 中的位 4 置 1。
7. 当此触发生成中断时，请对 *中断状态寄存器 (ETH_MACISR)* 进行读操作。
8. 使用旧值重新编程*时间戳加数寄存器 (ETH_MACTSAR)*，并再次将位 5 置 1。

58.9.8 关于节能型以太网 (EEE) 的编程指南

进入和退出 Tx LPI 模式

初始化 MAC 时，请完成以下步骤：

1. 通过 MDIO 接口对 PHY 寄存器进行读操作，并检查远程端是否具有 EEE 功能。然后协商定时器值。
2. 通过 MDIO 接口编程 PHY 寄存器（包括 RX_CLK_stoppable 位，该位用于向 PHY 指示是否在 LPI 模式下停止 Rx 时钟）。
3. 对 *LPI 定时器控制寄存器 (ETH_MACLTCR)* 中的位 25 到 16 以及位 1 到 0 进行编程。
4. 使用 MDIO 接口读取 PHY 链路状态，并更新 *LPI 控制状态寄存器 (ETH_MACLCSR)* 的位 17。
5. 相应地更新 *LPI 控制状态寄存器 (ETH_MACLCSR)*。PHY 芯片的链路状态发生变化时，应进行此更新。
6. 根据用于访问 CSR 从端口的时钟频率，编程 *1 微秒节拍计数器寄存器 (ETH_MAC1USTCR)*。
7. 使用 MAC 在自行进入 LPI 状态之前应等待的 IDLE 时间编程 *LPI 进入定时器寄存器 (ETH_MACLETR)* 中的 LPIET 位。
8. 将 *LPI 控制状态寄存器 (ETH_MACLCSR)* 的 LPITE 和 LPITXA（位 20 到 19）置 1。
9. 更新 *LPI 控制状态寄存器 (ETH_MACLCSR)* 以使能 LPI 自动进入和 MAC 自动退出 LPI 状态。
10. 根据用于访问 CSR 从端口的时钟频率，编程 *1 微秒节拍计数器寄存器 (ETH_MAC1USTCR)*。
11. 使用 MAC 在自行进入 LPI 状态之前应等待的 IDLE 时间编程 *LPI 进入定时器寄存器 (ETH_MACLETR)* 寄存器中的 LPIET 位。
12. 将 *LPI 控制状态寄存器 (ETH_MACLCSR)* 的 LPITE 和 LPITXA（位 20 和 19）置 1 以使能 LPI 自动进入和 MAC 自动退出 LPI 状态。
13. 将 *LPI 控制状态寄存器 (ETH_MACLCSR)* 的位 16 置 1 以将 MAC 发送器置于 LPI 状态。

    所有计划的数据包均完成后，MAC 会进入 LPI 状态。它将在 LPIET 位所指示的时间内始终保持空闲。进入 LPI 状态后，它会将 TLPIEN（位 0）置 1。
14. 计划好数据包发送后（TxDMA 退出空闲状态，或者 ATI 或 MTI 接口上出现数据包时），MAC 发送器自动退出 LPI 状态。MAC 发送器在 TLPIEX 中断状态位置 1 之前会等待 TWT 时间，之后才会恢复数据包发送。
15. 如果 MAC 发送器在 LPIET 时间内保持空闲，则会再次进入 LPI 状态。然后，它会将 TLPIEN 位置 1，继续执行进入-退出循环。
16. 如果应用程序需要覆盖自动进入/退出模式，并使 MAC 发送器直接退出 LPI 状态，请复位 LPITXEN 位。

58.9.9 关于每秒脉冲数 (PPS) 灵活的输出的编程指南

基于 PPS 生成单脉冲

要基于 PPS 生成单脉冲：

1. 在 *PPS 控制寄存器 (ETH_MACPPSCR)* 中将 TRGTMODSEL 位编程为 11 或 10（用于中断）。此操作会命令 MAC 将目标时间寄存器（寄存器 736 和 737）用作 PPS 信号输出的开始时间。
2. 在目标时间寄存器（寄存器 736 和 737）中编程开始时间值。
3. 在 *PPS 宽度寄存器 (ETH_MACPPSWR)* 寄存器中编程 PPS 信号输出的宽度。
4. 将 *PPS 控制寄存器 (ETH_MACPPSCR)* 的 PPSCMD 编程为 0001。此操作会命令 MAC 在目标时间寄存器中编程的时间内在 PPS 信号输出上生成单脉冲。

执行 PPSCMD（PPSCMD 位 = 0）时，可通过在经过编程的开始时间之前给出取消开始命令（PPSCMD= 0011）来取消脉冲生成。此外，还可以提前对下一个脉冲的行为进行编程。要编程下一个脉冲：

1. 在目标时间寄存器中编程下一个脉冲的开始时间。此时间应晚于上一个脉冲出现下降沿的时间。
2. 在 *PPS 宽度寄存器 (ETH_MACPPSWR)* 中编程下一个 PPS 信号输出的宽度。
3. 编程 *PPS 控制寄存器 (ETH_MACPPSCR)* 的 PPSCMD 位，以在上一个脉冲无效后生成单个脉冲。此操作会命令 MAC 在目标时间寄存器中编程的时间内在 PPS 信号输出上生成单脉冲。

   如果在上一个脉冲变为低电平之前给出此命令，则新命令将覆盖上一个命令，并且 QOS 可能只生成 1 个扩展脉冲。

基于 PPS 生成脉冲串

要基于 PPS 生成脉冲串：

1. 在 *PPS 控制寄存器 (ETH_MACPPSCR)* 中将 TRGTMODSEL 位编程为 11 或 10（用于中断）。此操作会命令 MAC 将目标时间寄存器（寄存器 736 和 737）用于 PPS 信号输出的开始时间。
2. 在目标时间寄存器（寄存器 736 和 737）中编程开始时间值。
3. 在 *PPS 间隔寄存器 (ETH_MACPPSIR)* 中编程 PPS 信号输出上的脉冲串之间的间隔值。
4. 在 *PPS 宽度寄存器 (ETH_MACPPSWR)* 中编程 PPS 信号输出的宽度。
5. 将 *PPS 控制寄存器 (ETH_MACPPSCR)* 中的 PPSCMD 位编程为 0010。此操作会命令 MAC 在目标时间寄存器中编程的开始时间内在 PPS 信号输出上生成脉冲串。

   除非通过发出“经过一段时间后停止脉冲串”或“立即停止脉冲串”命令停止 PPS 脉冲串，否则默认情况下，PPS 脉冲串自由运行。
6. 在目标时间寄存器中编程停止值。确保在再次编程目标时间寄存器之前，*PPS 目标时间纳秒寄存器 (ETH_MACPPSTTNR)* 中的 TSTRBUSY 位复位。
7. 将 *PPS 控制寄存器 (ETH_MACPPSCR)* 中的 PPSCMD 位编程为 0100，以在第 6 步编程指定的停止时间过后，停止 PPS 信号输出上的脉冲串。

可通过在 PPSCMD 字段中编程 0101 随时停止脉冲串。

类似地，可通过在第 6 步中编程的时间结束之前，将 PPSCMD 位编程为 0110 来取消停止脉冲串命令（在第 5 步中给出）。

可通过在第 2 步中编程的开始时间之前，将 PPSCMD 编程为 0011 来停止生成脉冲串。

在不影响 PPS 的情况下生成中断

可通过 *PPS 控制寄存器 (ETH_MACPPSCR)* 中的 TRGTMODSEL 位将目标时间寄存器（寄存器 736 和 737）编程为执行以下任一操作：

- 仅生成中断。
- 生成中断以及 PPS 开始和停止时间。
- 仅生成 PPS 开始和停止时间。

要将目标时间寄存器编程为仅生成中断事件：

1. 将 *PPS 控制寄存器 (ETH_MACPPSCR)* 的 TRGTMODSEL 位编程为 00（用于中断）。此操作会命令 MAC 将目标时间寄存器用于目标时间中断。
2. 在目标时间寄存器中编程目标时间值。此操作会命令 MAC 在目标时间结束后生成中断。

   如果 TRGTMODSEL 位发生更改（例如，用于控制 PPS），则中断生成操作将采用新模式和新编程的目标时间寄存器值。

58.9.10 关于 TSO 的编程指南

要编程 TSO，请按以下步骤操作：

1. 对相应 ETH_DMACTxCR 寄存器的 TSE 位进行编程，以使能该 DMA 中的 TCP 数据包分段。
2. 要为当前数据包使能 TSO，除了正常传输描述符设置外，还必须对以下描述符字段进行编程：
   a) 使能 TDES3 的 TSE（位 18）。
   b) 在 TDES3 的位 17 到 0 中对未分段的 TCP/IP 数据包有效负载长度进行编程，并在 TDES3 的位 22 到 19 中对 TCP 报头长度进行编程。
   c) 在以下字段中编程最大段大小：
   – ETH_DMACCR 的 MSS
   – 或上下文描述符中的 MSS

   如果在 ETH_DMACCR 和上下文描述符中均对 MSS 字段进行了编程，则会采用通过软件最新编程的序列。
3. 未分段的 TCP/IP 数据包报头应存储在第一个描述符的缓冲区 1 中。该缓冲区不能保存任何有效负载字节。有效负载被分配给缓冲区 2 和后续描述符的缓冲区。

注意： 如果在 TDES3 中针对非 TCP-IP 数据包使能了 TSE，则结果将不可预测。

58.9.11 关于在接收端执行 VLAN 过滤的编程指南

要在接收器端执行 VLAN 过滤，请按以下步骤操作：

1. 编程 *VLAN 标记寄存器 (ETH_MACVTR)* 中的以下位，以选择过滤方法：
   - ETV：使能 12 位 VLAN 标记比较或 16 位 VLAN 标记比较。
   - VTHM：VLAN 标记散列表匹配使能。
   - ERIVLT：使能内部 VLAN 标记或外部 VLAN 标记（以使能内部或外部 VLAN 标记过滤，双 VLAN 处理应通过将 EDVLP 置 1 来使能）
   - ERSVLM：使能接收 S-VLAN 匹配或 C-VLAN 匹配（对于要使能的 S-VLAN 处理，将 ESVL 置 1）
   - DOVLTC：忽略标记匹配的 VLAN 类型
   - VTIM：用于使能 VLAN 标记反向匹配，而非正常 VLAN 标记匹配
2. 针对 12 位或 16 位 VLAN 标记，编程 *VLAN 标记寄存器 (ETH_MACVTR)* 中的 VL 位。
3. 如果使能 VLAN 标记散列过滤，则编程 *VLAN 散列表寄存器 (ETH_MACVHTR)*。计算得出的 CRC 的高四位用于索引 VLAN 散列表的内容。例如，散列值 1000 用于选择 VLAN 散列表的位 8。

58.10 描述符

58.10.1 描述符概述

在以太网外设中，DMA 基于描述符的链表传输数据。应用程序在系统存储器 (SRAM) 中创建描述符。支持以下两种描述符：

- **正常描述符**

  正常描述符用于数据包数据，以及用于提供适用于待发送数据包的控制信息。
- **上下文描述符**

  上下文描述符用于提供适用于待发送数据包的控制信息。

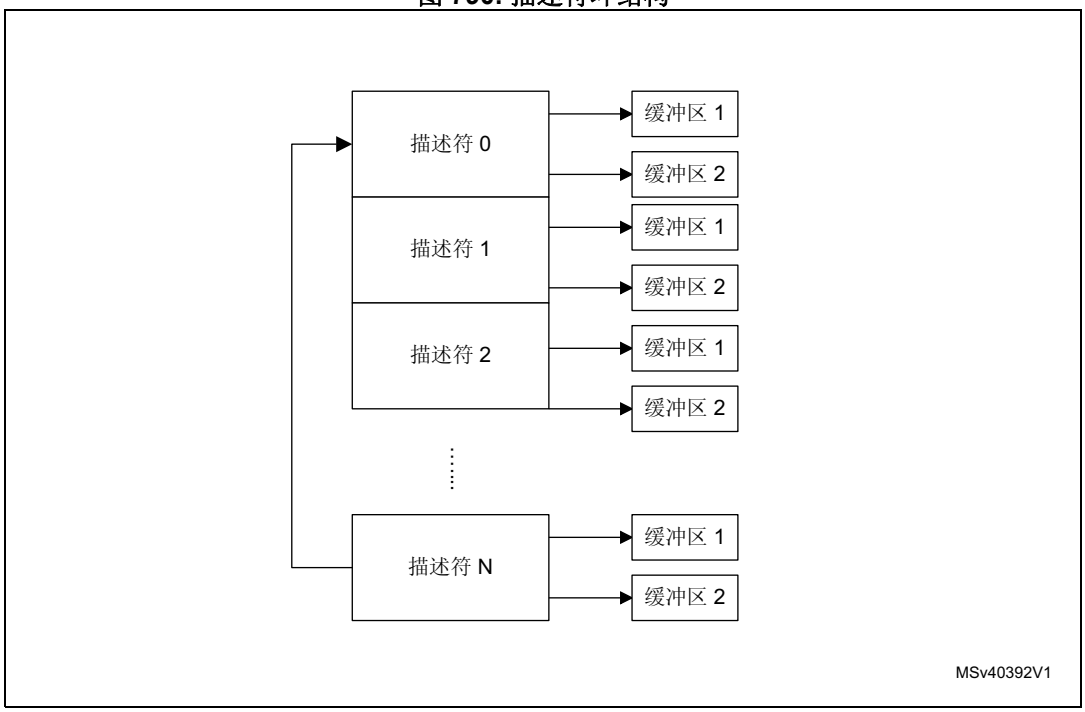
每个正常描述符均包含两个缓冲区和两个地址指针。这些缓冲区可使适配器端口与各种类型的存储器管理方案兼容。

可将任意数量的描述符用于单个数据包。

58.10.2 描述符结构

以太网外设支持 DMA 描述符的环结构。

图 790. 描述符环结构



在环结构中，各描述符之间以*通道控制寄存器 (ETH_DMACCR)* 的 DSL 字段中编程的 32 位字数为间隔彼此分开。应用程序需要在 DMA 通道的以下寄存器中编程总环长度，即，环结构内的描述符总数：

- *通道 Tx 描述符环长度寄存器 (ETH_DMACTxRLR)*
- *通道 Rx 描述符环长度寄存器 (ETH_DMACRxRLR)*

通道 Tx 描述符尾指针寄存器 (ETH_DMACTxDTPR) 或*通道 Rx 描述符尾指针寄存器 (ETH_DMACRxDTPR)* 包含指向描述符地址的指针 (*N*)。基址和当前描述符指针决定 DMA 可处理的当前描述符的地址。对于至少比描述符尾指针 (*N* – 1) 所指示的描述符少一个地址单元的描述符，均为 DMA 所有。DMA 会持续处理这些描述符，直至出现以下条件：

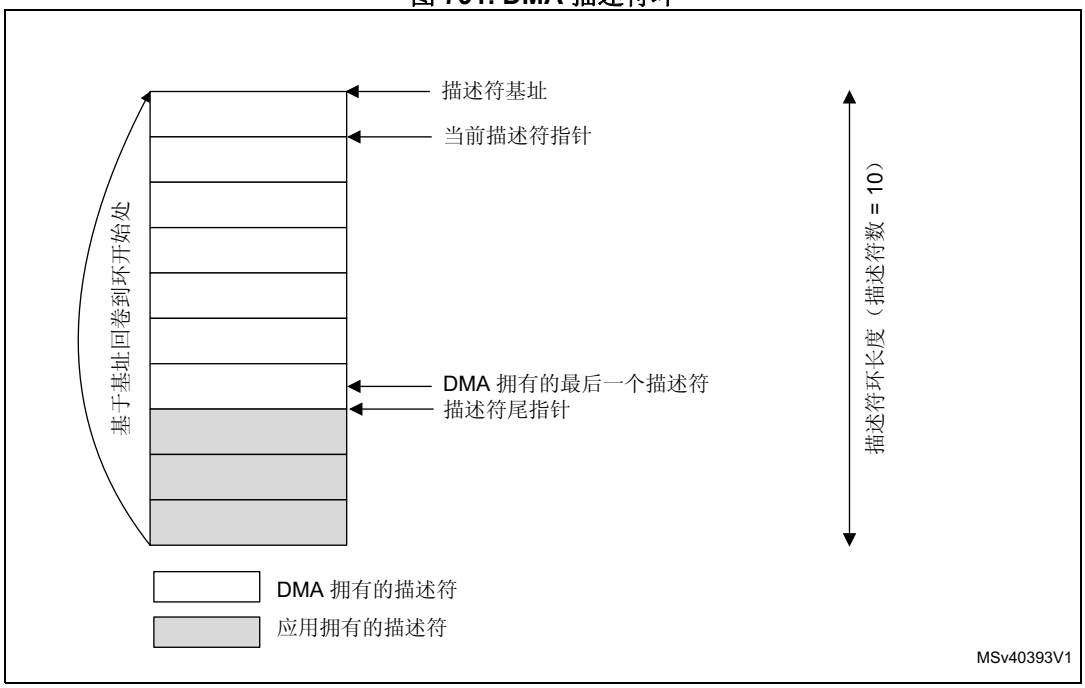
```
Current Descriptor Pointer == Descriptor Tail Pointer;
```

出现如上条件时，DMA 将进入挂起状态。应用程序必须对描述符尾指针寄存器执行写操作，并更新尾指针，以满足以下条件：

```
Current Descriptor Pointer < Descriptor Tail Pointer;
```

到达环结束处时，DMA 会自动基于基址进行回卷，如*图 791：DMA 描述符环*所示。

图 791. DMA 描述符环



对于应用程序拥有的描述符，DES3 的 OWN 位复位为 0。

对于 DMA 拥有的描述符，OWN 位置 1。

在开始时，如果应用程序只有一个描述符，则会将最后一个描述符地址（尾指针）设为描述符基址 + 1。DMA 随即会处理第一个描述符，并等待应用程序对尾指针进行递增。

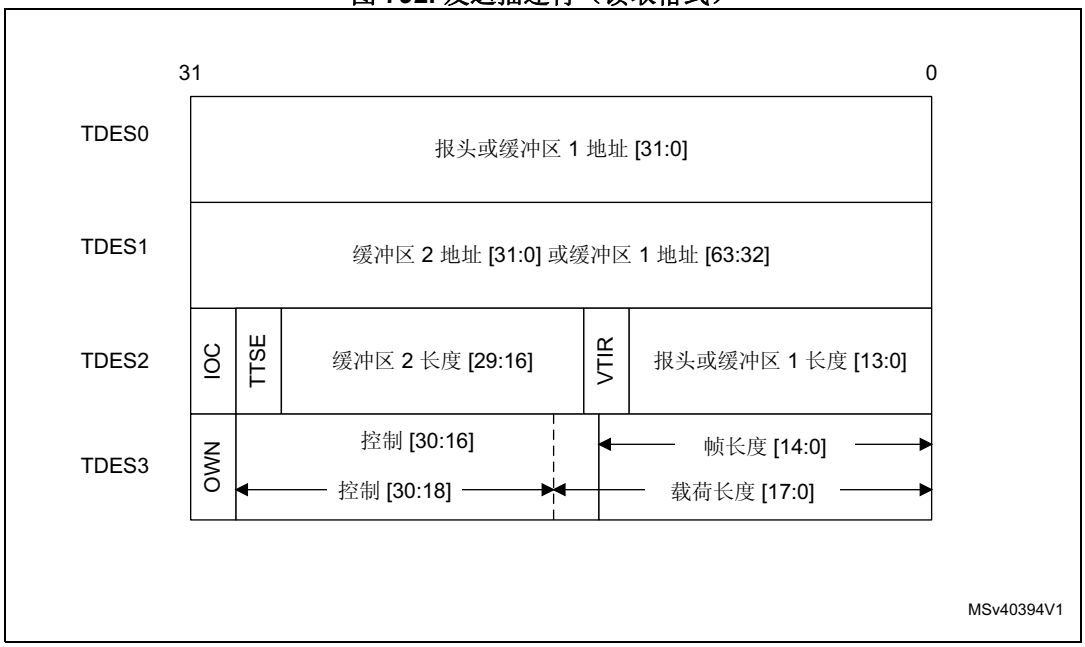
58.10.3 发送描述符

对于一个发送数据包，以太网外设 DMA 至少需要一个描述符。除了两个缓冲区、两个字节计数缓冲区和两个地址指针之外，发送描述符还具有控制字段，此类字段可用于按发送数据包来管理 MAC 操作。发送正常描述符具有以下两种格式：读取格式和回写格式。

发送正常描述符（读取格式）

图 792 显示了发送正常描述符的读取格式。到 *表 512* 到 *表 515* 给出了所有发送正常描述符（读取格式）的详细说明。

图 792. 发送描述符（读取格式）



- **TDES0 正常描述符（读取格式）**

表 512. TDES0 正常描述符（读取格式）

位	名称	说明
31:0	BUF1AP	**缓冲区 1 地址指针或 TSO 报头地址指针 (Buffer 1 Address Pointer or TSO Header Address Pointer)** 将以下位置 1 时，这些位指示缓冲区 1 的物理地址或 TSO 报头地址指针： – TDES3 的 TSE 位 – TDES3 的 FD 位

- **TDES1 正常描述符（读取格式）**

表 513. TDES1 正常描述符（读取格式）

位	名称	说明
31:0	BUF2AP	**缓冲区 2 或缓冲区 1 地址指针 (Buffer 2 or Buffer 1 Address Pointer)：** 使用描述符环结构时，这些位指示缓冲区 2 的物理地址。缓冲区地址对齐方式不限。

- **TDES2 正常描述符（读取格式）**

31	30	29:16	15:14	13:0
IOC	TTSE	B2L	VTIR	HL 或 B1L

表 514. TDES2 正常描述符（读取格式）

位	名称	说明
31	IOC	**完成时中断 (Interrupt on completion)：** 当前数据包发送完成时，该位会将*通道状态寄存器 (ETH_DMACSR)* 中的 TI 位置 1。
30	TTSE	**发送时间戳使能 (Transmit time stamp enable)**
29:16	B2L	**缓冲区 2 长度 (Buffer 2 Length)** 驱动程序会将该字段置 1。驱动程序会设置这个字段，该字段表示缓冲区 2 的长度。
15:14	VTIR	**VLAN 标记插入或替换 (VLAN Tag Insertion or Replacement)：** 这些位请求 MAC 在发送数据包之前执行 VLAN 标记或取消标记操作。为数据包使能 VLAN 标记插入、替换或删除时，应用程序必须相应地设置 CRC 填充控制位。这些位的值如下： – 00：不添加 VLAN 标记。 – 01：在发送前从数据包中删除 VLAN 标记。此选项应只与 VLAN 数据包一起使用。 – 10：使用在 *VLAN 包含寄存器 (ETH_MACVIR)* 或上下文描述符中编程的标记值插入 VLAN 标记。 – 11：使用在 *VLAN 包含寄存器 (ETH_MACVIR)* 或上下文描述符中编程的标记值替换数据包中的 VLAN 标记。此选项应只与 VLAN 数据包一起使用。
13:0	HL 或 B1L	**报头长度或缓冲区 1 长度 (Header Length or Buffer 1 Length)** 对于报头长度，仅考虑位 [9:0]。位 13 到 0 仅应用于缓冲区 1 长度。 如果通过 TDES3 的 TSE 位使能 TCP 分段减荷功能，则该字段等于报头长度。TDES3 中的 TSE 位置 1 时，报头长度包括从以太网源地址到 TCP 报头结束的长度（以字节为单位进行表示）。对于 TSO 功能，支持的最大报头长度为 1023 字节。对于 TSO 功能，支持的最大报头长度为 1023 字节。 如果未使能 TCP 分段减荷功能，则该字段等于缓冲区 1 长度。

- **TDES3 正常描述符（读取格式）**

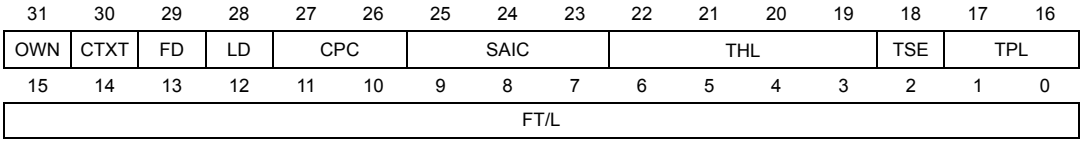


表 515. TDES3 正常描述符（读取格式）

位	名称	说明
31	OWN	**所有关系位 (Own bit)** – 1：描述符为 DMA 所有。 – 0：描述符为应用程序所有。 在传输完相关缓冲区中给出的数据后，DMA 会将此位清零。
30	CTXT	**上下文类型 (Context Type)** 对于正常描述符，该位应设为 0。
29	FD	**第一个描述符 (First descriptor)** 该位置 1 时，指示缓冲区中包含数据包的首段。
28	LD	**最后一个描述符 (Last descriptor)** 该位置 1 时，指示缓冲区中包含数据包的末段。B1L 或 B2L 字段应具有非零值。
27:26	CPC	**CRC 填充控制 (CRC Pad Control)** 该字段控制针对 Tx 数据包的 CRC 和填充插入。只有第一个描述符位 (TDES3[29]) 置 1 时，该字段才有效。位 [27:26] 的值如下： – 00：CRC 和填充插入 MAC 会在长度大于或等于 60 字节的发送数据包末尾附加循环冗余校验 (CRC)。MAC 会自动将填充值和 CRC 附加到长度小于 60 字节的数据包。 – 01：CRC 插入（禁止填充插入） MAC 会在发送数据包的末尾附加 CRC，但不附加填充值。应用程序应确保从发送缓冲区传输的数据包中存在填充字节，即，从发送缓冲区传输的数据包长度大于或等于 60 字节。 – 10：禁止 CRC 插入 MAC 不会在发送数据包的末尾附加 CRC。应用程序应确保从发送缓冲区传输的数据包中存在填充和 CRC 字节。 – 11：CRC 替换 MAC 会用重新计算的 CRC 字节替换发送数据包的最后四个字节。应用程序应确保从发送缓冲区传输的数据包中存在填充和 CRC 字节。 *注：　TSE 位置 1 时，MAC 会忽略此字段，因为将始终为分段进行 CRC 和填充插入。*

表 515. TDES3 正常描述符（读取格式）（续）

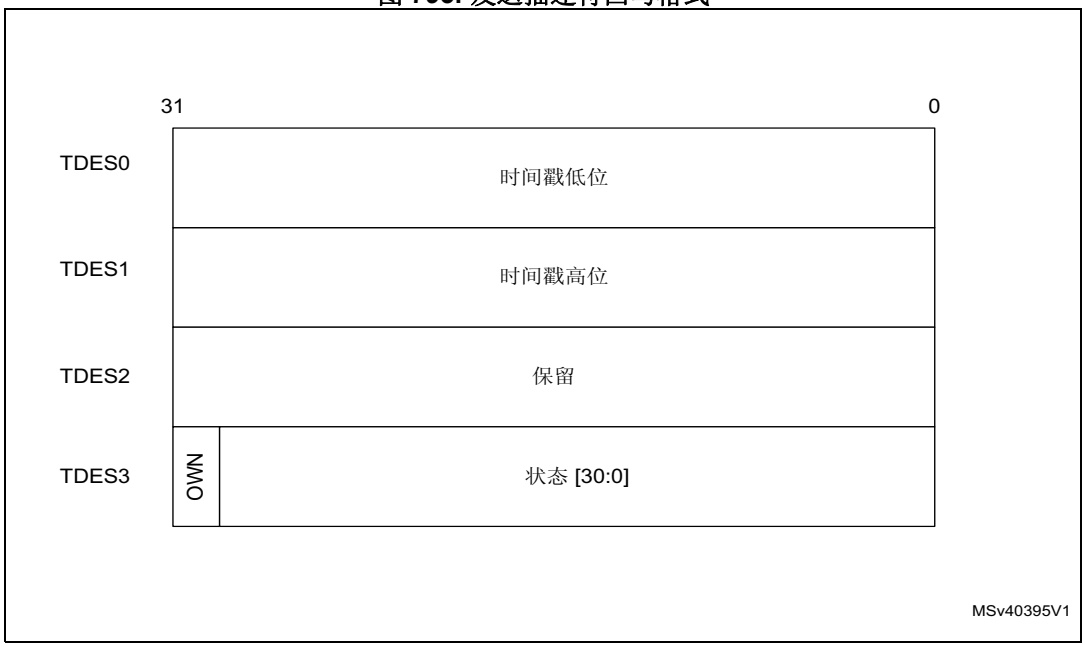
位	名称	说明
25:23	SAIC	**SA 插入控制 (SA Insertion Control)** 这些位请求 MAC 用 MAC 地址 0 寄存器中给出的值添加或替换以太网数据包中的源地址字段。为数据包使能 SA 插入控制时，应用程序必须相应地设置 CRC 填充控制位。 位 25 指定用于源地址插入或替换的 MAC 地址寄存器（1 或 0）值。 下面对位 [24:23] 的值进行了说明： – 00：不包含源地址 – 01：包含或插入源地址。为实现可靠发送，应用程序必须提供不包含源地址的帧。 – 10：替换源地址。为实现可靠发送，应用程序必须提供包含源地址的帧。 – 11：保留 这些位在首段控制位 (TDES3 [29]) 置 1 的情况下有效。
22:19	THL	**THL**：TCP 报头长度 (TCP Header Length) 如果 TSE 位置 1，则该字段包含 TCP 报头的长度。该字段的最小值必须为 5。
18	TSE	**TCP 分段使能 (TCP Segmentation Enable)** 该位置 1 时，DMA 对数据包执行 TCP 分段。只有在 FD 位置 1 时，该位才有效。
17:16	CIC/TPL	**校验和插入控制或 TCP 有效负载长度 (Checksum Insertion Control or TCP Payload Length)** 这些位控制校验和的计算与插入。可采用以下值： – 00：禁止插入校验和。 – 01：仅使能 IP 报头校验和的计算与插入。 – 10：使能 IP 报头校验和以及有效负载校验和的计算与插入，但不会在硬件中计算伪报头校验和。 – 11：使能 IP 报头校验和以及有效负载校验和的计算与插入，并在硬件中计算伪报头校验和。 TSE 位复位时，该字段有效。TSE 位置 1 时，该字段包含 TCP 有效负载长度的高位 [17:16]。这样可使 TCP 数据包长度字段跨越 TDES3[17:0]，从而支持 256 KB 数据包长度。
15	TPL	**保留或 TCP 有效负载长度 (Reserved or TCP Payload Length)** TSE 位复位时，该位保留。TSE 位置 1 时，该位表示 TCP 有效负载长度 [17:0] 的位 15。
14:0	FL/TPL	**数据包长度或 TCP 有效负载长度 (Packet Length or TCP Payload Length)** 该字段等于要发送的数据包的长度（以字节为单位进行表示）。TSE 位未置 1 时，该字段等于要发送的数据包的总长度： 以太网报头长度 + TCP/IP 报头长度 – 报头长度 – SFD 长度 + 以太网有效负载长度 TSE 位置 1 时，该字段等于 TCP 有效负载长度的低 15 位。此长度不包括以太网报头或 TCP/IP 报头长度。

发送正常描述符（回写格式）

回写格式仅适用于相应数据包的最后一个描述符。在 DMA 回写相应发送数据包的状态和时间戳信息的描述符中，LD 位 (TDES3[28]) 置 1。

图 793 显示了发送正常描述符的回写格式。*表 516* 到 *表 519* 给出了所有发送正常描述符（回写格式）的详细说明。

图 793. 发送描述符回写格式



- **TDES0 正常描述符（回写格式）**

表 516. DES0 正常描述符（回写格式）(1)

位	名称	说明
31:0	TTSL	**发送数据包时间戳低位 (Transmit Packet Timestamp Low)** DMA 使用为对应发送数据包捕获的时间戳的 32 个最低有效位更新该字段。只有在数据包第一个描述符中的 TDES2 的 TTSE 位置 1 时，DMA 才会写入时间戳。只有在描述符中的末段位 (LS) 置 1 并且时间戳状态 (TTSS) 位置 1 时，该字段才保持时间戳。

1. 此格式仅适用于数据包的最后一个描述符。

- **TDES1 正常描述符（回写格式）**

表 517. TDES1 正常描述符（回写格式）(1)

位	名称	说明
31:0	TTSH	**发送数据包时间戳高位 (Transmit Packet Timestamp High)** DMA 使用为对应接收数据包捕获的时间戳的 32 个最高有效位更新该字段。只有在数据包第一个描述符中的 TDES2 的 TTSE 位置 1 时，DMA 才会写入时间戳。只有在描述符中的末段位 (LS) 置 1 并且时间戳状态 (TTSS) 位置 1 时，该字段才具有时间戳。

1. 此格式仅适用于数据包的最后一个描述符。

- **TDES2 正常描述符（回写格式）**

表 518. TDES2 正常描述符（回写格式）(1)

位	说明
31:0	保留

1. 此格式仅适用于数据包的最后一个描述符。

- **TDES3 正常描述符（回写格式）**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
OWN	CTXT	FD	LD	保留										TTSS	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ES	JT	FF	PCE	LoC	NC	LC	EC	CC				ED	UF	DB	IHE

表 519. TDES3 正常描述符（回写格式）(1)

位	名称	说明
31	OWN	**所有关系位 (Own bit)** 该位置 1 时，表示 DMA 拥有描述符。DMA 会在完成数据包发送时将该位清零。完成回写后，此位会置 0。
30	CTXT	**上下文类型 (Context Type)** 对于正常描述符，该位应设为 0。
29	FD	**第一个描述符 (First descriptor)** 该位指示缓冲区中包含数据包的首段。
28	LD	**最后一个描述符 (Last descriptor)** 对于数据包的最后一个描述符，该位置 1。DMA 仅在数据包的最后一个描述符中才会对状态字段进行写操作。
27:18	保留	
17	TTSS	**Tx 时间戳状态 (Tx Timestamp Status)** 该状态位指示已经为相应的发送数据包捕获时间戳。该位置 1 时，TDES2 和 TDES3 将包含为发送数据包捕获到的时间戳值。该字段仅在描述符末段控制位 (TDES3 [28]) 置 1 的情况下有效。
16	保留	

表 519. TDES3 正常描述符（回写格式）(1)（续）

位	名称	说明
15	ES	**错误汇总 (Error Summary)：** 该位指示以下位的逻辑或运算结果： – TDES3[0]：IP 报头错误 (IP Header Error) – TDES3[14]：Jabber 超时 (Jabber Timeout) – TDES3[13]：数据包刷新 (Packet Flush) – TDES3[12]：有效负载校验和错误 (Payload Checksum Error) – TDES3[11]：载波丢失 (Loss of Carrier) – TDES3[10]：无载波 (No Carrier) – TDES3[9]：延迟冲突 (Late Collision) – TDES3[8]：过度冲突 (Excessive Collision) – TDES3[3]：过度延迟 (Excessive Deferral) – TDES3[2]：下溢错误 (Underflow Error)
14	JT	**Jabber 超时 (Jabber Timeout)** 该位指示 MAC 发送器经历了 jabber 超时。只有在*工作模式配置寄存器 (ETH_MACCR)* 的 JD 位未置 1 时，该位才会置 1。
13	FF	**数据包已刷新 (Packet Flushed)** 该位指示 DMA 或 MTL 已依照 CPU 发出的软件刷新命令刷新数据包。
12	PCE	**有效负载校验和错误 (Payload Checksum Error)** 该位指示校验和减荷引擎出现故障，并且未将任何校验和插入到封装的 TCP、UDP 或 ICMP 有效负载中。这种故障可能是由于字节数不足（即，未达到 IP 报头的有效负载长度字段所指示的数量）而引起，或者因 MTL 在未计算出校验和的情况下开始在存储转发模式下将数据包转发到 MAC 发送器而引起。第二种错误条件仅发生在发送 FIFO 深度小于发送的以太网数据包的长度的情况下（以避免死锁），MTL 会在 FIFO 已满时开始转发数据包，即使在存储转发模式下也不例外。
11	LoC	**载波丢失 (Loss of Carrier)** 该位指示数据包发送期间丢失载波（即，数据包发送期间有一个或多个发送时钟周期的 ETH_CRS 信号无效）。该位仅对在 MAC 处于半双工模式时实现无冲突发送的数据包有效。
10	NC	**无载波 (No Carrier)** 该位指示在发送期间未由 PHY 触发载波侦听信号。
9	LC	**延迟冲突 (Late Collision)** 该位指示数据包发送过程因冲突窗口（64 个字节时间，含报头）后出现冲突而中止。如果下溢错误位置 1，则该位无效。
8	EC	**过度冲突 (Excessive Collision)** 该位指示尝试发送当前数据包时因出现 16 个连续冲突而中止发送。如果*工作模式配置寄存器 (ETH_MACCR)* 中的 DR 位置 1，则该位会在出现首个冲突后置 1，并且数据包发送过程将中止。

表 519. TDES3 正常描述符（回写格式）(1)（续）

位	名称	说明
7:4	CC	**冲突计数 (Collision Count)** 此 4 位计数器值指示发送数据包之前出现的冲突个数。EC 位置 1 时，该计数无效。
3	ED	**过度延迟 (Excessive Deferral)** *工作模式配置寄存器 (ETH_MACCR)* 中的 DC 位置 1 时，该位指示因延迟超过 24,288 个位时间而结束发送。
2	UF	**下溢错误 (Underflow Error)** 该位指示 MAC 因系统存储器的数据未及时到达而中止了数据包的发送。在以下任意一种情况下，都可能会发生下溢错误： – DMA 在数据包发送期间遇到发送缓冲区为空的情况 – 应用程序填充 MTL Tx FIFO 的速率慢于 MAC 发送速率 发送过程进入挂起状态，并且 ETH_MTLISR 寄存器中对应于队列的下溢位置 1。
1	DB	**延迟位 (Deferred Bit)** 该位指示 MAC 在发送前因存在载波而延迟。该位仅在半双工模式下有效。
0	IHE	**IP 报头错误 (IP Header Error)** IP 报头错误位置 1 时，该位指示校验和减荷引擎检测到 IP 报头错误。该位仅在使能了 Tx 校验和减荷时有效。否则，该位将保留。如果以太网类型字段指示 IPv4 有效负载，即使 COE 检测到 IP 报头错误，也会插入 IPv4 报头校验和。

1. 此格式仅适用于数据包的最后一个描述符。

发送上下文描述符

可在数据包描述符之前的任何时间提供发送上下文描述符。上下文对当前数据包和后续数据包有效。上下文描述符用于提供时间戳（用于进行一步时间戳校准）、VLAN 标记 ID（用于 VLAN 插入功能）和 SA 插入位（用于 SA 插入）。只能针对上下文描述符进行回写来复位 OWN 位。

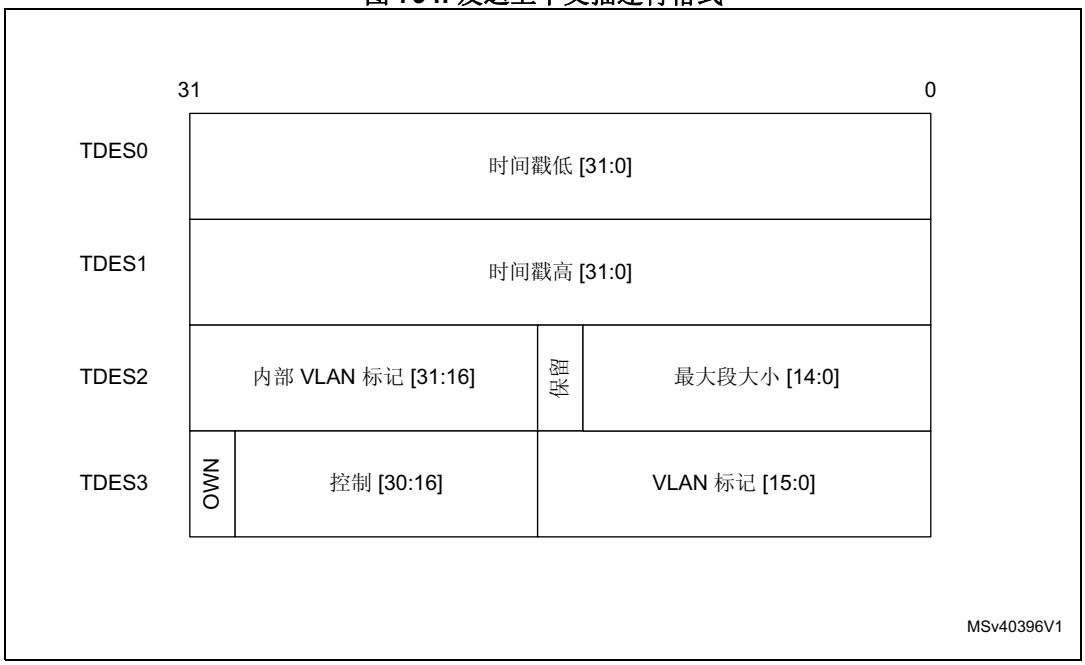
注： *应用程序在上下文描述符中提供的 VLAN 标记 ID 和 MSS 值（其相应的有效位置 1）由 DMA 在内部进行存储。*

所提供的外部或内部 VLAN 标记的有效位置 1 时，DMA 始终会将最后一个有效 VLAN 标记传递给 MTL。应用程序不能使 DMA 存储的有效 VLAN 标记失效。VLAN 标记将基于为数据包提供的控制输入进行插入或替换。

内部 VLAN 标记控制输入仅用于紧跟上下文描述符之后的数据包。如果数据包需要 DMA 为其使用内部 VLAN 标记控制输入，应用程序必须在数据包的正常描述符之前提供上下文描述符。

图 794 显示了发送上下文描述符的格式。*表 520* 到 *表 523* 给出了所有发送上下文描述符的详细说明。

图 794. 发送上下文描述符格式



- **TDES0** 上下文描述符（读取格式）

表 520. TDES0 上下文描述符

位	名称	说明
31:0	TTSL	**发送数据包时间戳低位 (Transmit Packet Timestamp Low)** 对于一步校准，驱动程序可以在该描述符字中提供时间戳的低 32 位。DMA 将该值用作低位字来进行一步时间戳校准。只有在 TDES3 上下文描述符的 OSTC 和 TCMSSV 位置 1 时，该字段才有效。

- **TDES1** 上下文描述符（读取格式）

表 521. TDES1 上下文描述符

位	名称	说明
31:0	TTSH	**发送数据包时间戳高位 (Transmit Packet Timestamp High)** 对于一步校准，驱动程序可以在该描述符中提供时间戳的高 32 位。DMA 将该值用作高位字来进行一步时间戳校准。只有在 TDES3 上下文描述符的 OSTC 和 TCMSSV 位置 1 时，该字段才有效。

- **TDES2 上下文描述符（读取格式）**

表 522. TDES2 上下文描述符

位	名称	说明
31:16	IVT	**内部 VLAN 标记 (Inner VLAN Tag)** TDES3 上下文描述符的 IVLTV 位置 1 且 TDES3 上下文描述符的 TCMSSV 和 OSTC 位复位时，TDES2[31:16] 包含要插入到后续发送数据包中的内部 VLAN 标记。
15:14	保留	
13:0	MSS	**最大段大小 (Maximum Segment Size)** 对 TCP/IP 有效负载进行分段时使用此段大小。只有在 TDES3 上下文描述符的 TCMSSV 位置 1 且 TDES3 上下文描述符的 OSTC 位复位时，该字段才有效。

- **TDES3 上下文描述符（读取格式）**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
OWN	CTXT	保留		OSTC	TCMSSV	保留		CDE	保留					IVLTV	VLTV
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VT															

表 523. TDES3 上下文描述符

位	名称	说明
31	OWN	**所有关系位 (Own bit)** – 1：描述符为 DMA 所有。 – 0：描述符为应用程序所有。 以下任一条件为真时，DMA 会将该位清零： – DMA 完成数据包接收。 – 与描述符相关的缓冲区已满。
30	CTXT	**上下文类型 (Context Type)** 对于上下文描述符，该位应置 1。
29:28	保留	
27	OSTC	**一步时间戳校准使能 (One-Step Timestamp Correction Enable)** 该位置 1 时，DMA 将参考 TDES0 和 TDES1 中提供的时间戳值执行一步时间戳校准。
26	TCMSSV	**一步时间戳校准输入或 MSS 有效 (One-Step Timestamp Correction Input or MSS Valid)** 该位和 OSTC 位置 1 时，表示 TDES0 和 TDES1 中提供的时间戳校准输入有效。 OSTC 位复位，且后续正常描述符中的该位以及 TDES3 的 TSE 位置 1 时，表示 TDES2 中的 MSS 输入有效。
25:24	保留	

表 523. TDES3 上下文描述符（续）

位	名称	说明
23	CDE	**上下文描述符错误 (Context Descriptor Error)** 该位置 1 时，表示上下文描述符以错误的序列提供，DMA 会将其忽略。关闭上下文描述符时，DMA 会在回写期间将此位置 1。
22:20	保留	
19:18	IVTIR	**内部 VLAN 标记插入或替换 (Inner VLAN Tag Insert or Replace)** 这些位置 1 时，会请求 MAC 在发送数据包之前执行内部 VLAN 标记或取消标记操作。如果修改了数据包的 VLAN 标记，则 MAC 会自动重新计算并替换 CRC 字节。 该字段具有以下值： – 00：不添加内部 VLAN 标记。 – 01：在发送前从数据包中删除内部 VLAN 标记。此选项应只与 VLAN 帧一起使用。 – 10：使用在*内部 VLAN 包含寄存器 (ETH_MACIVIR)* 或上下文描述符中编程的标记值插入内部 VLAN 标记。 – 11：使用在*内部 VLAN 包含寄存器 (ETH_MACIVIR)* 或上下文描述符中编程的标记值替换数据包中的内部 VLAN 标记。此选项应只与 VLAN 帧一起使用。
17	IVLTV	**内部 VLAN 标记有效 (Inner VLAN Tag Valid)** 该位置 1 时，表示 TDES2 的 IVT 字段有效。
16	VLTV	**VLAN 标记有效 (VLAN Tag Valid)** 该位置 1 时，表示 TDES3 的 VT 字段有效。
15:0	VT	**VLAN 标记 (VLAN Tag)** 该字段包含要在数据包中插入或替换的 VLAN 标记。只有在 *VLAN 包含寄存器 (ETH_MACVIR)* 的 VLTI 位复位时，该字段才用作 VLAN 标记。

58.10.4 接收描述符

只有在尾指针不同于基址指针或当前指针时，以太网外设中的 DMA 才会尝试读取描述符。建议使用长度至少可容纳 MAC 所接收到的两个完整数据包的描述符环；否则，会因描述符不可用而导致 DMA 的性能受到显著影响。在这种情况下，MTL RxFIFO 已满，将开始丢弃数据包。

存在以下两种接收描述符：

- 支持读取和回写格式的正常描述符
- 上下文描述符

接收到的所有描述符均由软件准备，并被作为“正常”描述符提供给 DMA（有关描述符内容的详细说明，请参见*图 795：接收正常描述符（读取格式）*）。DMA 读取该描述符，并且在将接收到的数据包（或数据包的一部分）传送到由描述符指示的缓冲区后，Rx DMA 会关闭具有相应数据包状态的描述符。状态格式如*图 796：接收正常描述符（回写格式）*所示。

对于某些数据包，正常描述符位不足以写入完整状态。对于这类数据包，Rx DMA 会将扩展状态写入下一个描述符（无需处理或使用嵌入到该描述符中的缓冲区指针）。此回写描述符的格式和内容如*图 797：接收上下文描述符*所示。

接收正常描述符（读取格式）

*图 795*显示了接收正常描述符的读取格式。*表 524* 到*表 527* 给出了所有接收正常描述符（读取格式）的详细说明。

图 795. 接收正常描述符（读取格式）

注： *在接收描述符（读取格式）中，如果缓冲区地址字段只包含 0，则 MAC 不会将数据传输到该缓冲区，并会跳到下一个缓冲区或下一个描述符。*

- **RDES0 正常描述符（读取格式）**

表 524. RDES0 正常描述符（读取格式）

位	名称	说明
31:0	BUF1AP	**报头或缓冲区 1 地址指针 (Header or Buffer 1 Address Pointer)** 应用程序可针对该缓冲区编程字节对齐地址，即，该字段的 LS 位可以为非零值。不过，在传送数据包开始时，DMA 将在 RDES0[1:0]（或者 64/128 位配置时为 RDES0[2:0]/[3:0]）为零的情况下执行写操作。但数据包数据会根据缓冲区地址指针中给出的实际偏移进行移位。 如果地址指针指向用于存储数据包的中间部分或最后一部分的缓冲区，则 DMA 将忽略偏移地址，并会写入完整位置（根据数据宽度的指示）。

- **RDES1 正常描述符（读取格式）**

表 525. RDES1 正常描述符（读取格式）

位	名称	说明
31:0	保留	保留字段。

- **RDES2 正常描述符（读取格式）**

表 526. RDES2 正常描述符（读取格式）

位	名称	说明
31:0	BUF2AP	**缓冲区 2 地址指针 (Buffer 2 Address Pointer)** 这些位指示缓冲区 2 物理地址。 只有在传送数据包的起始字节时，RxDMA 才会使用指针地址的 LS 位。如果 BUF2AP 提供用于存储数据包的中间部分或最后一部分的缓冲区的地址，DMA 将忽略 RDES2[1:0]=0 并会写入完整位置。

- **RDES3 正常描述符（读取格式）**

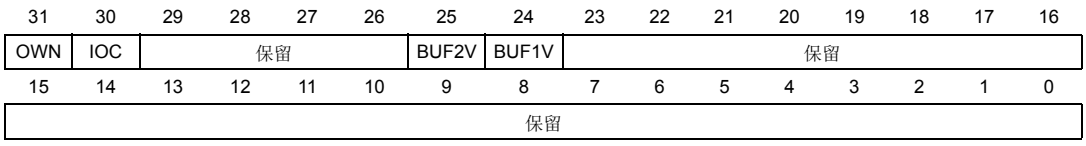


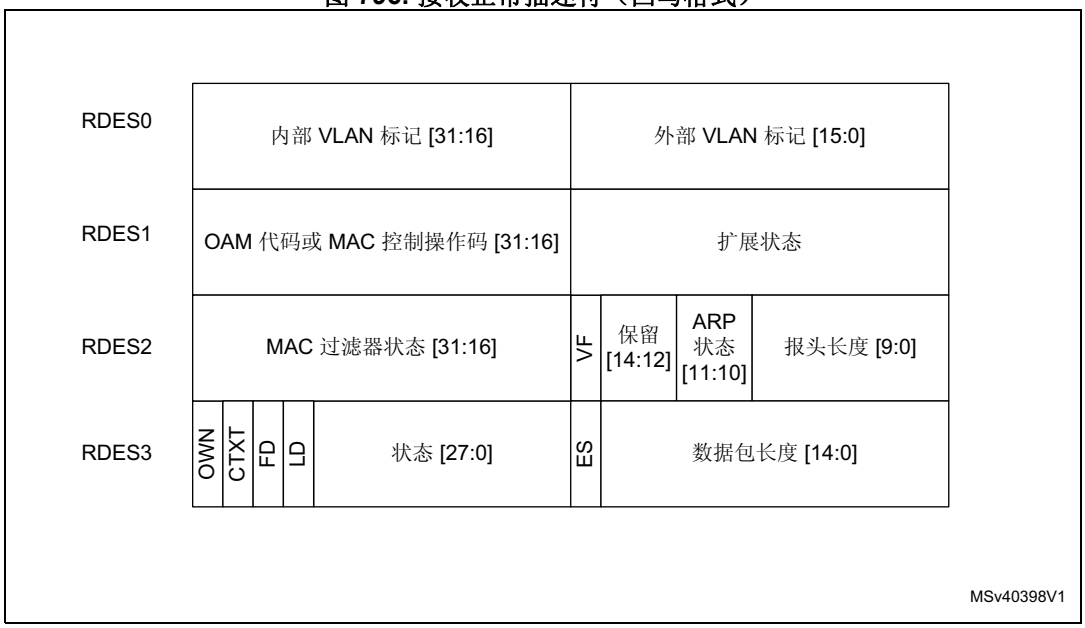
表 527. RDES3 正常描述符（读取格式）

位	名称	说明
31	OWN	**所有关系位 (Own bit)** 该位置 1 时，表示 DMA 拥有描述符。该位复位时，表示应用程序拥有描述符。以下任一条件为真时，DMA 会将该位清零： – DMA 完成数据包接收 – 与描述符相关的缓冲区已满
30	IOC	**完成时使能中断 (Interrupt Enabled on Completion)** 如果该位置 1，则在 DMA 关闭此描述符时会向应用程序发出中断。
29:26	保留	
25	BUF2V	**缓冲区 2 地址有效 (Buffer 2 Address Valid)** 该位置 1 时，向 DMA 指示 RDES0 中指定的缓冲区 1 地址有效。应用程序必须将该位置 1，以便 DMA 可使用 RDES0 中缓冲区 2 地址所指向的地址来写入接收到的数据包数据。
24	BUF1V	**缓冲区 1 地址有效 (Buffer 1 Address Valid)** 置 1 时，向 DMA 指示 RDES1 中指定的缓冲区 1 地址有效。 应用程序必须将该位置 1，以便 DMA 可使用 RDES1 中缓冲区 1 地址所指向的地址来写入接收到的数据包数据。
23:0	保留	

接收正常描述符（回写格式）

图 796 显示了接收正常描述符的回写格式。*表 528* 到 *表 531* 给出了所有接收正常描述符（回写格式）的详细说明。

图 796. 接收正常描述符（回写格式）



- **RDES0 正常描述符（回写格式）**

表 528. RDES0 正常描述符（回写格式）

位	名称	说明
31:16	IVT	**内部 VLAN 标记 (Inner VLAN Tag)** 如果 RDES3 的 RS0V 位置 1，则该字段包含接收到的数据包的内部 VLAN 标记。
15:0	OVT	**外部 VLAN 标记 (Outer VLAN Tag)** 如果 RDES3 的 RS0V 位置 1，则该字段包含接收到的数据包的外部 VLAN 标记。

- **RDES1 正常描述符（回写格式）**

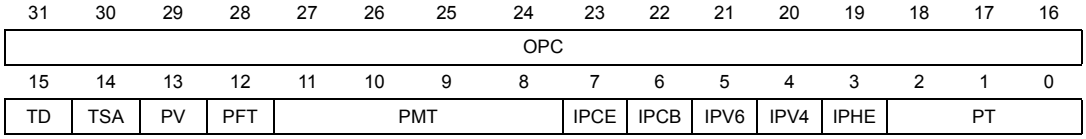


表 529. RDES1 正常描述符（回写格式）(1)

位	名称	说明
31:16	OPC	**OAM 子类型代码 (OAM Sub-Type Code) 或 MAC 控制数据包操作码 (MAC Control Packet opcode)** OAM 子类型代码 如果 RDES3 的位 [18:16] 置为 111，则该字段包含 OAM 子类型和代码字段。 MAC 控制数据包操作码 如果 RDES3 的位 [18:16] 置为 110，则该字段包含 MAC 控制数据包操作码字段。
15	TD	**时间戳被丢弃 (Timestamp Dropped)** 该位表示已为此数据包捕获时间戳，但因发生上溢而在 MTL Rx FIFO 中将时间戳丢弃。
14	TSA	**时间戳可用 (Timestamp Available)** 存在时间戳时，该位表示时间戳值在上下文描述符字 2 (RDES2) 和字 1 (RDES1) 中可用。只有在最后一个描述符位 (RDES3 [28]) 置 1 时，该位才有效。 上下文描述符被写入到紧跟数据包的最后一个正常描述符之后的下一个描述符。
13	PV	**PTP 版本 (PTP Version)** 1：接收到的 PTP 消息采用 IEEE 1588 版本 2 格式 0：接收到的 PTP 消息采用 IEEE 1588 版本 1 格式
12	PFT	**PTP 数据包类型 (PTP Packet Type)** 该位表示 PTP 消息直接通过以太网发送。

表 529. RDES1 正常描述符（回写格式）(1)（续）

位	名称	说明
11:8	PMT	**PTP 消息类型 (PTP Message Type)** 将对这些位进行编码，以给出所接收消息的类型： – 0000：未接收到任何 PTP 消息 – 0001：SYNC（所有时钟类型） – 0010：Follow_Up（所有时钟类型） – 0011：Delay_Req（所有时钟类型） – 0100：Delay_Resp（所有时钟类型） – 0101：Pdelay_Req（针对点对点透明时钟） – 0110：Pdelay_Resp（针对点对点透明时钟） – 0111：Pdelay_Resp_Follow_Up（针对点对点透明时钟） – 1000：发布 – 1001：管理 – 1010：信号传输 – 1011–1110：保留 – 1111：采用保留消息类型的 PTP 数据包 只有在选择时间戳功能时，这些位才可用。
7	IPCE	**IP 有效负载错误 (IP Payload Error)** 该位置 1 时，表示以下任一项： – 由 MAC 计算的 16 位 IP 有效负载校验和（即 TCP、UDP 或 ICMP 校验和）与接收段中对应的校验和字段不匹配。 – TCP、UDP 或 ICMP 段长度与 IP 报头字段中的有效负载长度值不匹配。 – TCP、UDP 或 ICMP 段长度小于 TCP、UDP 或 ICMP 的最小允许段长度。 该位置 1 时，RDES3 的位 15 (ES) 不置 1。
6	IPCB	**绕过 IP 校验和 (IP Checksum Bypassed)** 该位表示绕过校验和减荷引擎。
5	IPV6	**存在 IPv6 报头 (IPv6 header Present)** 该位表示检测到 IPV6 报头。
4	IPV4	**存在 IPV4 报头 (IPV4 Header Present)** 该位表示检测到 IPV4 报头。

表 529. RDES1 正常描述符（回写格式）(1)（续）

位	名称	说明
3	IPHE	**IP 报头错误 (IP Header Error)** 该位置 1 时，表示以下任一项： – 由 MAC 计算的 16 位 IPv4 报头校验和与接收的校验和字节不匹配。 – IP 数据报版本与以太网类型值不一致。 – 以太网数据包不具有预期 IP 报头字节数。 该位在位 5 或位 4 置 1 时有效。
2:0	PT	**有效负载类型 (Payload Type)** 这些位表示由接收校验和减荷引擎 (COE) 处理的 IP 数据报中封装的有效负载类型： – 000：类型未知，或未处理 IP/AV 有效负载 – 001：UDP – 010：TCP – 011：ICMP – 其他值：保留 如果 COE 因存在 IP 报头错误或分段 IP 而未处理 IP 数据报的有效负载，则会将这些位设为 3'b000。

1. 回写格式的状态字段仅对最后一个描述符（RDES3[28] 置 1）有效。

- **RDES2 正常描述符（回写格式）**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
L3L4FM			L4FM	L3FM	MADRM								HF	DAF	SAF
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VF	保留				ARPRN	保留									

表 530. RDES2 正常描述符（回写格式）

位	名称	说明
31:29	L3L4FM	**第 3 层和第 4 层过滤器编号匹配 (Layer 3 and Layer 4 Filter Number Matched)** 这些位表示与接收到的数据包匹配的第 3 层和第 4 层过滤器的编号： – 000：过滤器 0 – 001：过滤器 1 – 010：过滤器 2 – 011：过滤器 3 – 100：过滤器 4 – 101：过滤器 5 – 110：过滤器 6 – 111：过滤器 7 只有在位 28 或位 27 置为高电平时，该字段才有效。如果有多个过滤器匹配，则这些位会给出最低级过滤器的编号。

表 530. RDES2 正常描述符（回写格式）（续）

位	名称	说明
28	L4FM	**第 4 层过滤器匹配 (Layer 4 Filter Match)** 该位置 1 时，表示接收到的数据包与使能的第 4 层端口号字段之一匹配。只有在下列条件之一为真时，才会给出该状态： – 第 3 层字段未使能，所有使能的第 4 层字段均匹配 – 所有已使能的第 3 层和第 4 层过滤器字段均匹配 如果有多个过滤器匹配，该位会给出由位 [31:29] 指示的过滤器的第 4 层过滤器状态。
27	L3FM	**第 3 层过滤器匹配 (Layer 3 Filter Match)** 该位置 1 时，表示接收到的数据包与使能的第 3 层 IP 地址字段之一匹配。只有在下列条件之一为真时，才会给出该状态： – 所有使能的第 3 层字段均匹配，所有使能的第 4 层字段均被绕过 – 所有使能的过滤器字段均匹配 如果有多个过滤器匹配，该位会给出由位 [31:29] 指示的过滤器的第 3 层过滤器状态。
26:19	MADRM	**MAC 地址匹配或散列值 (MAC Address Match or Hash Value)** 当 HF 位复位时，该字段包含与接收到的数据包的目标地址匹配的 MAC 地址寄存器编号。只有在 DAF 位复位时，该字段才有效。 HF 位置 1 时，该字段包含由 MAC 计算的散列值。散列过滤器寄存器中对应于散列值的位置 1 时，数据包会通过散列过滤器。
18	HF	**散列过滤器状态 (Hash Filter Status)** 该位置 1 时，表示数据包通过 MAC 地址散列过滤器。位 [26:19] 表示散列值。
17	DAF	**目标地址过滤失败 (Destination Address Filter Fail)** 该位置 1 时，表示数据包未通过 MAC 中的 DA 过滤。
16	SAF	**SA 地址过滤失败 (SA Address Filter Fail)** 该位置 1 时，表示数据包未通过 MAC 中的 SA 过滤。
15	VF	**VLAN 过滤状态 (VLAN Filter Status)** 该位置 1 时，表示接收到的数据包的 VLAN 标记通过 VLAN 过滤。
14:11	保留	
10	ARPNR	**未生成 ARP 响应 (ARP Reply Not Generated)** 该位置 1 时，表示 MAC 未针对接收到的 ARP 请求数据包生成 ARP 响应。MAC 忙于针对较早 ARP 请求发送 ARP 响应时，该位置 1（一次只能处理一个 ARP 请求）。
9:0	保留	

- **RDES3 正常描述符（回写格式）**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
OWN	CTXT	FD	LD	RS2V	RS1V	RS0V	CE	GP	RWT	OE	RE	DE	LT		
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ES	PL														

表 531. RDES3 正常描述符（回写格式）

位	名称	说明
31	OWN	**所有关系位 (Own bit)** 1：描述符为 DMA 所有。 0：描述符为应用程序所有。 以下任一条件为真时，DMA 会将该位清零： – DMA 完成数据包接收 – 与描述符相关的缓冲区已满
30	CTXT	**接收上下文描述符 (Receive Context Descriptor)** 该位置 1 时，表示当前描述符为上下文类型描述符。对于正常接收描述符，DMA 向该位写入 0。
29	FD	**第一个描述符 (First descriptor)** 该位置 1 时，表示该描述符包含数据包的第一个缓冲区。如果第一个缓冲区的大小为 0，则第二个缓冲区将包含数据包的起始字节。如果第二个缓冲区的大小也为 0，则下一个描述符将包含数据包的起始字节。
28	LD	**最后一个描述符 (Last descriptor)** 该位置 1 时，表示该描述符所指向的缓冲区为数据包的最后几个缓冲区。
27	RS2V	**接收状态 RDES2 有效 (Receive Status RDES2 Valid)** 该位置 1 时，表示 RDES2 中的状态有效，且由 DMA 写入。该位仅在 RDES3 的 LD 位置 1 时有效。
26	RS1V	**接收状态 RDES1 有效 (Receive Status RDES1 Valid)** 该位置 1 时，表示 RDES1 中的状态有效，且由 DMA 写入。该位仅在 RDES3 的 LD 位置 1 时有效。
25	RS0V	**接收状态 RDES0 有效 (Receive Status RDES0 Valid)** 该位置 1 时，表示 RDES0 中的状态有效，且由 DMA 写入。该位仅在 RDES3 的 LD 位置 1 时有效。
24	CE	**CRC 错误 (CRC Error)** 该位置 1 时，表示接收到的数据包发生循环冗余校验 (CRC) 错误。只有在 RDES3 的 LD 位置 1 时，该字段才有效。
23	GP	**大型数据包 (Giant Packet)** 该位置 1 时，表示数据包长度超出指定的最大以太网大小，即，1518、1522 或 2000 字节（如果巨型数据包使能位置 1，则为 9018 或 9022 字节）。 *注：　大型数据包仅表示数据包长度，不会导致数据包截断。*
22	RWT	**接收看门狗超时 (Receive Watchdog Timeout)** 该位置 1 时，表示接收看门狗定时器在接收当前数据包时已到期。看门狗超时后，当前数据包会被截断。
21	OE	**上溢错误 (Overflow Error)** 该位置 1 时，表示接收到的数据包因 Rx FIFO 中发生缓冲区上溢而损坏。 *注：　只有在 DMA 将部分数据包传输到应用程序时，该位才会置 1。仅当 Rx FIFO 工作在阈值模式下时才会出现这种情况。在存储转发模式下，所有部分数据包均会在 Rx FIFO 中被完全丢弃。*

表 531. RDES3 正常描述符（回写格式）（续）

位	名称	说明
20	RE	**接收错误 (Receive Error)** 该位置 1 时，表示在数据包接收期间 ETH_RX_DV 信号置为有效时，ETH_RX_ER 信号置为有效。
19	DE	**Dribble 位错误 (Dribble Bit Error)** 该位置 1 时，表示接收到的数据包具有非整数倍数的字节（奇数半字节）。该位仅在 MII 模式下有效。
18:16	LT	**长度/类型字段 (Length/Type Field)** 该字段指示接收到的数据包为长度数据包或类型数据包。这 3 个位的编码如下： – 000：数据包为长度数据包 – 001：数据包为类型数据包 – 011：数据包为 ARP 请求数据包类型 – 100：数据包为带有 VLAN 标记的类型数据包 – 101：数据包为带有双 VLAN 标记的类型数据包 – 110：数据包为 MAC 控制数据包类型 – 111：数据包为 OAM 数据包类型 – 010：保留
15	ES	**错误汇总 (Error Summary)** – 该位置 1 时，表示以下位的逻辑或运算结果： – RDES3[24]：CRC 错误 (CRC Error) – RDES3[19]：Dribble 错误 (Dribble Error) – RDES3[20]：接收错误 (Receive Error) – RDES3[22]：看门狗超时 (Watchdog Timeout) – RDES3[21]：上溢错误 (Overflow Error) – RDES3[23]：大型数据包 (Giant Packet) 只有在 RDES3 的 LD 位置 1 时，该字段才有效。
14:0	PL	**数据包长度 (Packet Length)** 这些位指示传输到系统存储器中的接收数据包的字节长度（含 CRC）。 RDES3 的 LD 位置 1 且描述符错误 (RDES3[13]) 或上溢错误位复位时，该字段有效。使能 IP 校验和计算并且接收到的数据包不是 MAC 控制数据包时，数据包长度还包括两个附加到以太网数据包的字节。 该字段在 RDES3 的 LD 位置 1 时有效。当最后一个描述符位和错误汇总位均未置 1 时，该字段指示已为当前数据包传输的累计字节数。

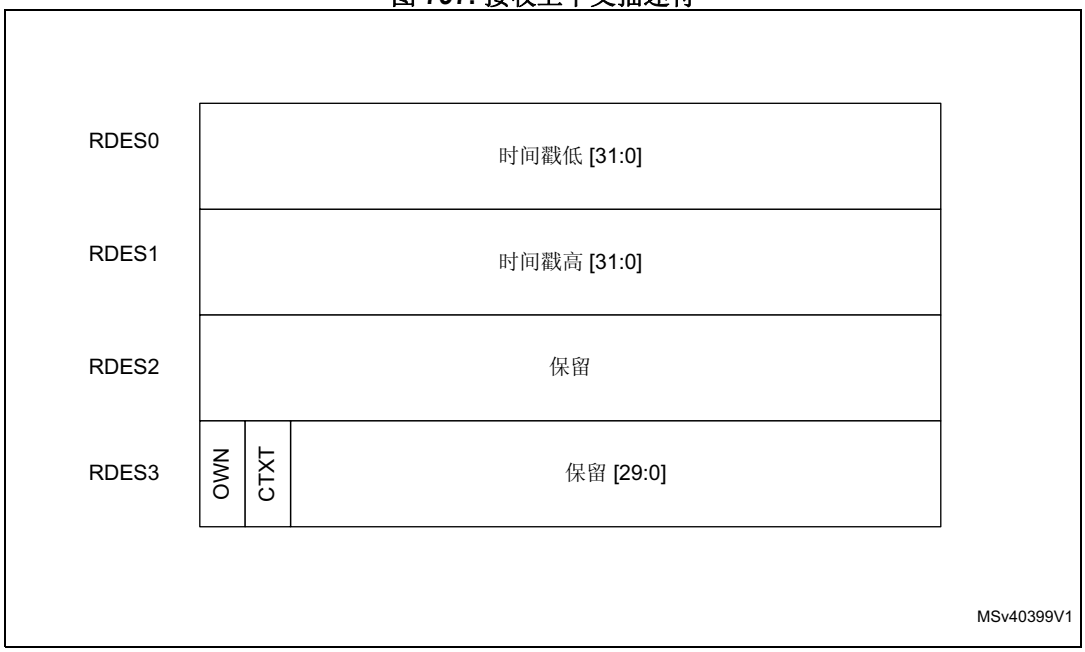
接收上下文描述符

该描述符对于应用程序是只读的。只能由 DMA 对该描述符进行写操作。

上下文描述符提供与最后接收到的数据包有关的扩展状态信息。RDES3 的位 30 表示上下文类型描述符。

图 797 显示了接收上下文描述符的格式。*表 532* 到*表 535* 给出了所有接收上下文描述符的详细说明。

图 797. 接收上下文描述符



- **RDES0** 上下文描述符

表 532. RDES0 上下文描述符

位	名称	说明
31:0	RTSL	**接收数据包时间戳低位 (Receive Packet Timestamp Low)** DMA 使用为对应接收数据包捕获的时间戳的 32 个最低有效位更新该字段。当该字段和 RDES1 的 RTSH 字段为全 1 值时，必须视为时间戳已损坏。

- **RDES1** 上下文描述符

表 533. RDES1 上下文描述符

位	字段	说明
31:0	RTSH	**接收数据包时间戳高位 (Receive Packet Timestamp High)** DMA 使用为对应接收数据包捕获的时间戳的 32 个最高有效位更新该字段。当该字段和 RDES0 的 RTSL 字段为全 1 值时，必须视为时间戳已损坏。

- **RDES2 上下文描述符**

表 534. RDES2 上下文描述符

位	说明
31:0	保留

- **RDES3 上下文描述符**

表 535. RDES3 上下文描述符

位	名称	说明
31	OWN	**所有关系位 (Own Bit)** 1：描述符为 DMA 所有。 0：描述符为应用程序所有。 以下任一条件为真时，DMA 会将该位清零： – DMA 完成数据包接收 – 与描述符相关的缓冲区已满
30	CTXT	**接收上下文描述符 (Receive Context Descriptor)** 该位置 1 时，表示当前描述符为上下文描述符。对于上下文描述符，DMA 向该位写入 1'b1。
29:0	保留	

58.11 以太网 MAC 寄存器

58.11.1 以太网寄存器映射

本节提供以下寄存器映射：

- DMA 寄存器（请参见*第 58.11.2 节：以太网 DMA 寄存器*）
- MTL 寄存器（请参见*第 58.11.3 节：以太网 MTL 寄存器*）
- MAC 寄存器，其中包括 MMC 寄存器（请参见*第 58.11.4 节：以太网 MAC 和 MMC 寄存器*）

58.11.2 以太网 DMA 寄存器

DMA 模式寄存器 (ETH_DMAMR)

DMA mode register

偏移地址：0x1000

复位值：0x0000 0000

DMA 模式寄存器为 DMA 建立总线工作模式。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	INTM[1:0]	
														rw	
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	PR[2:0]			TXPR	Res.	Res.	Res.	Res.	Res.	Res.	Res.			DA	SWR
	r			r										r	rw

位 31:18 保留，必须保持复位值。

位 17:16 **INTM[1:0]**：中断模式 (Interrupt Mode)
该字段定义以太网外设的中断模式。
中断信号和 ETH_DMACSR 寄存器中 RI/TI 位的行为随 INTM 值而变化（请参见 *表 511：传送完成中断行为*）。

位 15 保留，必须保持复位值。

位 14:12 **PR[2:0]**：优先级比 (Priority ratio)
这些位控制 Rx DMA 和 Tx DMA 之间加权循环调度仲裁的优先级比。这些位仅在 DA 位复位时有效。优先级比为 Rx:Tx 或 Tx:Rx，具体取决于 TXPR 位复位还是置 1。
000：优先级比为 1:1
001：优先级比为 2:1
010：优先级比为 3:1
011：优先级比为 4:1
100：优先级比为 5:1
101：优先级比为 6:1
110：优先级比为 7:1
111：优先级比为 8:1

位 11 **TXPR**：发送优先级 (Transmit priority)
该位置 1 时，表示在系统侧总线仲裁期间，Tx DMA 的优先级高于 Rx DMA。

位 10:2 保留，必须保持复位值。

位 1 **DA**：DMA Tx 或 Rx 仲裁方案 (DMA Tx or Rx Arbitration Scheme)
该位指定所有通道的发送和接收路径之间的仲裁方案：
0：采用 Rx:Tx 或 Tx:Rx 的加权循环调度
路径之间的优先级取决于位 [14:12] 中指定的优先级，优先级权重在 TXPR 位中指定。
1：固定优先级
TXPR 位置 1 时，Tx 路径的优先级高于 Rx 路径。否则，Rx 路径的优先级高于 Tx 路径。

位 0 **SWR**：软件复位 (Software Reset)

该位置 1 时，MAC 和 DMA 控制器会复位 DMA、MTL 和 MAC 的逻辑和所有内部寄存器。在所有时钟域中完成复位操作后，该位自动清零。重新编程任何寄存器之前，都应在该位中读取到一个零值。

注： 仅当所有活动时钟域中的所有复位均无效时，才会完成复位操作。因此，必须存在所有 PHY 输入时钟（适用于所选 PHY 接口）才能完成软件复位。完成软件复位操作所用的时间取决于最慢的活动时钟的频率。

系统总线模式寄存器 (ETH_DMASBMR)

System bus mode register

偏移地址：0x1004

复位值：0x0101 0000

系统总线模式寄存器控制 AHB 主设备的行为。它主要用于控制突发拆分和未完成请求的数量。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.				Res.	Res.	Res.	Res.	Res.			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RB	MB	Res.	AAL	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FB
r	r		rw												rw

位 31 保留，必须保持复位值。

位 29:28 保留，必须保持复位值。

位 23:16 保留，必须保持复位值。

位 15 **RB**：重建 INCRx 突发 (Rebuild INCRx Burst)

该位置为高电平且 AHB 主设备获得 SPLIT、RETRY 或提前突发终止 (EBT) 响应时，AHB 主接口将基于 INCRx 和 SINGLE 传输重新构建已发起突发传输的挂起节拍。默认情况下，AHB 主接口基于未指定 (INCR) 突发重建 EBT 的挂起节拍。

位 14 **MB**：混合突发 (Mixed Burst)

该位置为高电平且 FB 位为低电平时，如果突发长度大于等于 16，AHB 主设备会执行未定义的突发传输 (INCR)。如果突发长度小于等于 16，AHB 主设备会执行固定突发传输（INCRx 和 SINGLE）。

位 13 保留，必须保持复位值。

位 12 **AAL**：地址对齐的节拍 (Address-Aligned Beats)

该位置 1 时，主设备会在读通道和写通道上执行地址对齐的突发传输。

位 11:1 保留，必须保持复位值。

位 0 **FB**：固定突发长度 (Fixed Burst Length)

该位置 1 时，AHB 主设备将发起特定长度的突发传输（INCRx 或 SINGLE）。

该位置 0 时，AHB 主设备将发起非特定长度的传输 (INCR) 或 SINGLE 传输。

中断状态寄存器 (ETH_DMAISR)

Interrupt status register

偏移地址：0x1008

复位值：0x0000 0000

应用程序在中断服务程序或轮询期间读取该中断状态寄存器，以确定 DMA 通道、MTL 队列和 MAC 的中断状态。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MACIS	MTLIS
														r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DC0IS
															r

位 31:18 保留

位 17 **MACIS**：MAC 中断状态 (MAC Interrupt Status)

此位指示 MAC 中发生的中断事件。要将该位复位为 1'b0，软件必须读取 MAC 中的相应寄存器，以获取产生中断的具体原因，然后将中断源清除。

位 16 **MTLIS**：MTL 中断状态 (MTL Interrupt Status)

此位指示 MTL 中发生的中断事件。要将该位复位为 1'b0，软件必须读取 MTL 中的相应寄存器，以获取产生中断的具体原因，然后将中断源清除。

位 15:1 保留，必须保持复位值。

位 0 **DC0IS**：DMA 通道中断状态 (DMA Channel Interrupt Status)

此位指示 DMA 通道中发生的中断事件。要将该位复位为 0，软件必须读取 DMA 通道中的相应寄存器，以获取产生中断的具体原因，然后将中断源清除。

调试状态寄存器 (ETH_DMADSR)

Debug status register

偏移地址：0x100C

复位值：0x0000 0000

调试状态寄存器提供 DMA 通道的接收和发送过程状态，以供调试。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.				Res.			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPS0[3:0]				RPS0[3:0]				Res.	Res.	Res.	Res.	Res.	Res.	Res.	AXW HSTS
r				r											r

位 31:16 保留，必须保持复位值。

位 15:12 **TPS0[3:0]**：DMA 通道发送过程状态 (DMA Channel Transmit Process State)
该字段指示通道的 Tx DMA FSM 状态：
000：停止（发出复位或停止发送命令）
001：运行中（正在获取 Tx 传输描述符）
010：运行中（正在等待状态）
011：运行中（正在读取系统存储器缓冲区中的数据并将其加入 Tx 缓冲区 (Tx FIFO) 队列）
100：时间戳写状态
101：保留供将来使用
110：挂起（Tx 描述符不可用或 Tx 缓冲区下溢）
111：运行中（正在关闭 Tx 描述符）
该字段的 MSB 始终返回 0。此字段不会产生中断。

位 11:8 **RPS0[3:0]**：DMA 通道接收过程状态 (DMA Channel Receive Process State)
该字段指示通道的 Rx DMA FSM 状态：
000：停止（发出复位或停止接收命令）
001：运行中（正在获取 Rx 传输描述符）
010：保留供将来使用
011：运行中（正在等待 Rx 数据包）
100：已挂起（Rx 描述符不可用）
101：运行中（正在关闭 Rx 描述符）
110：时间戳写状态
111：运行中（正在将接收的数据包数据从 Rx 缓冲区传输到系统存储器）
该字段的 MSB 始终返回 0。此字段不会产生中断。

位 7:1 保留，必须保持复位值。

位 0 **AXWHSTS**：AHB 主设备写通道 (AHB Master Write Channel)
该位置为高电平时，表示 AHB 主 FMS 的写通道处于非空闲状态。

通道控制寄存器 (ETH_DMACCR)

Channel control register

偏移地址：0x1100

复位值：0x0000 0000

DMA 通道控制寄存器指定用于分段的 MSS 值、两个描述符之间跳过的长度，以及报头拆分和 8xPBL 模式等功能。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DSL[2:0]			Res.	PBL X8
											rw				rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	MSS[13:0]													
		rw													

位 31:24 保留，必须保持复位值。

位 23:21 保留，必须保持复位值。

位 20:18 **DSL[2:0]**：描述符跳过长度 (Descriptor Skip Length)
该位指定两个未链接描述符之间跳过的 32 位字数。地址从当前描述符结束处开始跳到下一个描述符起始处。
DSL 值等于零时，DMA 将描述符表视为连续的。

位 17 保留，必须保持复位值。

位 16 **PBLX8**：8xPBL 模式 (8xPBL mode)
该位置 1 时，在 ETH_DMACTxCR 的位 [21:16] 中编程的 PBL 值被乘以 8 倍。因此，DMA 根据 PBL 值以 8、16、32、64、128 和 256 个节拍传输数据。

位 15:14 保留，必须保持复位值。

位 13:0 **MSS[13:0]**：最大段大小 (Maximum Segment Size)
此字段指定在分段数据包时应使用的最大段大小。只有 ETH_DMACTxCR 寄存器的 TSE 位置 1 时，该字段才有效。
在此字段中编程的值必须大于配置的数据宽度（以字节为单位）。建议使用 64 字节或以上的 MSS 值。
未选择 Enable TCP Segmentation Offloading for TCP/IP Packets（使能 TCP/IP 数据包的 TCP 分段减荷）选项时，该字段保留。

通道发送控制寄存器 (ETH_DMACTxCR)

Channel transmit control register

偏移地址：0x1104

复位值：0x0000 0000

DMA 通道发送控制寄存器控制 Tx 功能，例如 PBL、TCP 分段和 Tx 通道权重。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.				Res.	Res.	TXPBL[5:0]					
										rw					
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	TSE	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OSF	Res.			ST
			rw								rw				rw

位 31:22 保留，必须保持复位值。

位 21:16 **TXPBL[5:0]**：发送可编程突发长度 (Transmit Programmable Burst Length)

这些位指示要在一个 DMA 数据传输过程中传输的最大节拍数。这是在单个块读或块写操作中使用的最大值。DMA 每次在应用总线上开始突发传输时，始终尝试按 PBL 中指定的方式进行突发。可使用以下任一值来编程 PBL：1、2、4、8、16 或 32。任何其他值都会产生未定义的行为。

要传输 32 个以上节拍，请按以下步骤操作：

1. 在 ETH_DMACCR 中设置 PBLx8 模式。
2. 设置 PBL。

位 15:13 保留，必须保持复位值。

位 12 **TSE**：TCP 分段使能 (TCP Segmentation Enabled)

该位置 1 时，DMA 会对通道 i 中的数据包执行 TCP 分段。仅针对在 Tx 正常描述符中将 TSE 位 (TDES0[19]) 置 1 的数据包执行 TCP 分段。该位置 1 时，TxPBL 值必须大于或等于 4。

如果未选择 Enable TCP Segmentation Offloading for TCP/IP Packets（使能 TCP/IP 数据包的 TCP 分段减荷）选项，则该字段保留。

位 11:5 保留，必须保持复位值。

位 4 **OSF**：处理第二个数据包 (Operate on Second Packet)

该位置 1 时会命令 DMA 处理发送数据的第二个数据包，即使尚未获得首个数据包的状态。

位 3:1 保留，必须保持复位值。

位 0 **ST**：启动或停止发送命令 (Start or Stop Transmission Command)

该位置 1 时，发送过程会进入运行状态。DMA 会检查当前位置的发送列表以查找待发送的数据包。

DMA 尝试从以下任一位置获取描述符：

– 列表中的当前位置：这是 ETH_DMACTxDLAR 寄存器设置的发送列表的基址。

– 上次停止发送时的位置

如果当前描述符不为 DMA 所有，则发送过程将进入挂起状态，并且 ETH_DMACSR 的 TBU 位会置 1。启动发送命令只有在已停止发送时才有效。如果该命令在设置 ETH_DMACTxDLAR 寄存器之前发出，则 DMA 行为无法预知。

该位复位时，发送过程会在完成发送当前数据包的任务之后进入停止状态， 并保存发送列表中的下一个描述符位置，该位置在重启发送后会成为当前位置。如要更改列表地址，则需要在该位复位时，使用新值编程 ETH_DMACTxDLAR 寄存器。该位再次置 1 时会采用新值。停止发送命令只有在当前数据包发送过程结束或发送过程处于挂起状态时才有效。

通道接收控制寄存器 (ETH_DMACRxCR)

Channel receive control register

偏移地址：0x1108

复位值：0x0000 0000

DMA 通道接收控制寄存器控制 Rx 功能，例如 PBL、缓冲区大小和扩展状态。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RPF	Res.	Res.	Res.	Res.				Res.	Res.	RXPBL[5:0]					
rw										rw					

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	RBSZ														SR
	rw														rw

位 31 **RPF**：DMA Rx 通道数据包刷新 (DMA Rx Channel Packet Flush)

此位置 1 时，DMA 会在 DMA Rx 通道发生系统总线错误而停止后，自动刷新以 DMA Rx 通道为目标的 Rx 队列中的数据包：

在 Rx 队列的读取端进行刷新。

该位置 0 时，EQOS 不会在 DMA 因系统总线错误而停止后，刷新以 DMA Rx 通道为目标的 Rx 队列中的数据包。

位 30:22 保留，必须保持复位值。

位 21:16 **RXPBL[5:0]**：接收可编程突发长度 (Receive Programmable Burst Length)

这些位指示要在一个 DMA 数据传输过程中传输的最大节拍数。这是在单个块读或块写操作中使用的最大值。DMA 每次在应用总线上开始突发传输时，始终尝试按 PBL 中指定的方式进行突发。可使用以下任一值来编程 PBL：1、2、4、8、16 或 32。任何其他值都会产生未定义的行为。

要传输 32 个以上节拍，请按以下步骤操作：

1. 在 ETH_DMACCR 中设置 PBLx8 模式。
2. 设置 PBL。

位 15 保留，必须保持复位值。

位 14:4 **RBSZ**：接收缓冲区大小 (Receive Buffer size)

该字段指示 Rx 缓冲区的大小（以字节为单位指定）。最大缓冲区大小被限制为 16 KB。使能拆分报头时，缓冲区大小适用于有效负载缓冲区。

注： 缓冲区大小必须是 4、8 或 16 的倍数，具体取决于总线宽度（分别为 32、64 或 128）。即使缓冲区地址指针的值未与总线宽度对齐也不例外。如果缓冲区大小不是 4、8 或 16 的倍数，则可能会产生未定义的行为。

32 位、64 位或 128 位总线宽度的 LSB 位（1:0、2:0 或 3:0）会被忽略，DMA 在内部将 LSB 位视为全零值。因此，这些 LSB 位为只读 (RO)。

位 3:1 保留，必须保持复位值。

位 0 **SR**：启动或停止接收 (Start or Stop Receive)

该位置 1 时，DMA 尝试从接收列表中获取描述符并处理传入的数据包。

DMA 尝试从以下任一位置获取描述符：

– 列表中的当前位置：这是 ETH_DMACRxDLAR 寄存器设置的地址。

– 上次停止 Rx 过程时的位置

如果当前描述符不为 DMA 所有，则接收过程将处于挂起状态，并且 ETH_DMACSR 的 RBU 位会置 1。启动接收命令只有在已停止接收时才有效。如果该命令在设置 ETH_DMACRxDLAR 寄存器之前发出，则 DMA 行为无法预知。

该位复位时，Rx DMA 会在传输当前数据包之后停止操作， 并保存接收列表中的下一个描述符位置，该位置在 Rx 过程重启后会成为当前位置。停止接收命令只有在 Rx 过程处于运行（等待 Rx 数据包）或挂起状态时才有效。

通道 Tx 描述符列表地址寄存器 (ETH_DMACTxDLAR)

Channel Tx descriptor list address register

偏移地址：0x1114

复位值：0x0000 0000

通道 Tx 描述符列表地址寄存器会将 DMA 指向发送描述符列表的起始处。描述符列表位于应用程序的物理存储器空间，且必须为字对齐。DMA 会将描述符列表的对应 LSB 位置为低电平，进而在内部将其转换为总线宽度对齐地址。

只有当 Tx DMA 停止时，即 ETH_DMACiTxCR 寄存器中的 ST 位置为 0 时，才能写入该寄存器。停止时，可以用新的描述符列表地址写入该寄存器。将 ST 位置 1 时，DMA 会使用新编程的描述符基址。如果在将 ST 位置为 0 时，该寄存器未更改，则 DMA 会使用先前停止时对应的描述符地址。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TDESLA															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TDESLA														Res.	Res.
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw		

位 31:2 **TDESLA**：发送列表的起始处 (Start of Transmit List)

此字段包含发送描述符列表中的首个描述符的基址。DMA 会忽略 32 位总线宽度的 LSB 位（1:0、2:0 或 3:0），并在内部将这些位视为全零值。因此，这些 LSB 位为只读 (RO)。

此字段的宽度取决于配置：

31:2 对应于 32 位配置

位 1:0 保留，必须保持复位值。

通道 Rx 描述符列表地址寄存器 (ETH_DMACRxDLAR)

Channel Rx descriptor list address register

偏移地址：0x111C

复位值：0x0000 0000

通道 Rx 描述符列表地址寄存器会将 DMA 指向接收描述符列表的起始处。

该寄存器指向接收描述符列表的起始处。描述符列表位于应用程序的物理存储器空间，且必须为字对齐。DMA 会将描述符列表的对应 LS 位置为低电平，进而在内部将其转换为总线宽度对齐地址。仅当接收停止时，才允许对该寄存器执行写操作。停止后，必须先对该寄存器执行写操作，然后才能发出接收启动命令。只有当 Rx DMA 停止时，即 ETH_DMACRxCR 寄存器中的 SR 位置为 0 时，才能写入该寄存器。停止时，可以用新的描述符列表地址写入该寄存器。

将 SR 位置 1 时，DMA 会使用新编程的描述符基址。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RDESLA															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RDESLA														Res.	Res.
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw		

位 31:2 **RDESLA[28:0]**：接收列表的起始处 (Start of Receive List)

此字段包含 Rx 描述符列表中的首个描述符的基址。DMA 会忽略 32 位总线宽度的 LSB 位（1:0、2:0 或 3:0），并在内部将这些位视为全零值。因此，这些 LSB 位为只读 (RO)。

此字段的宽度取决于配置：

31:2 对应于 32 位配置

位 1:0 保留，必须保持复位值。

通道 Tx 描述符尾指针寄存器 (ETH_DMACTxDTPR)

Channel Tx descriptor tail pointer register

偏移地址：0x1120

复位值：0x0000 0000

通道 Tx 描述符尾指针寄存器指向与基址的偏移，并指示最后一个有效描述符的位置。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TDT															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TDT														Res.	Res.
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw		

位 31:2 **TDT**：发送描述符尾指针 (Transmit Descriptor Tail Pointer)

该字段包含 Tx 描述符环的尾指针。软件对尾指针进行写操作，以向 Tx 通道添加更多描述符。硬件尝试发送由头指针和尾指针寄存器之间描述符引用的所有数据包。

此字段的宽度取决于配置：

31:2 对应于 32 位配置

位 1:0 保留，必须保持复位值。

通道 Rx 描述符尾指针寄存器 (ETH_DMACRxDTPR)

Channel Rx descriptor tail pointer register

偏移地址：0x1128

复位值：0x0000 0000

通道 Rx 描述符尾指针指向与基址的偏移，并指示最后一个有效描述符的位置。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RDT															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RDT														Res.	Res.
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw		

位 31:2 **RDT**：接收描述符尾指针 (Receive Descriptor Tail Pointer)
该字段包含 Rx 描述符环的尾指针。软件对尾指针进行写操作，以向 Rx 通道添加更多描述符。硬件尝试将接收到的数据包写入头指针和尾指针寄存器之间的描述符。
此字段的宽度取决于配置：
31:2 对应于 32 位配置

位 1:0 保留，必须保持复位值。

通道 Tx 描述符环长度寄存器 (ETH_DMACTxRLR)

Channel Tx descriptor ring length register

偏移地址：0x112C

复位值：0x0000 0000

Tx 描述符环长度寄存器包含发送描述符环的长度。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	TDRL[9:0]									
						rw									

位 31:10 保留，必须保持复位值。

位 9:0 **TDRL[9:0]**：发送描述符环长度 (Transmit Descriptor Ring Length)
该字段设置循环描述符环中 Tx 描述符的最大数量。描述符的最大数量限制为 1K 个描述符。Synopsys 建议的最小描述符环长度为 4。

通道 Rx 描述符环长度寄存器 (ETH_DMACRxRLR)

Channel Rx descriptor ring length register

偏移地址：0x1130

复位值：0x0000 0000

通道 Rx 描述符环长度寄存器包含循环接收描述符环的长度。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	RDRL[9:0]									
						rw									

位 31:10 保留，必须保持复位值。

位 9:0 **RDRL[9:0]**：接收描述符环长度 (Receive Descriptor Ring Length)
该寄存器设置循环描述符环中 Rx 描述符的最大数量。描述符的最大数量限制为 1K 个描述符。

通道中断使能寄存器 (ETH_DMACIER)

Channel interrupt enable register

偏移地址：0x1134

复位值：0x0000 0000

通道中断使能寄存器可使能状态寄存器报告的中断。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NIE	AIE	CDEE	FBEE	ERIE	ETIE	RWTE	RSE	RBUE	RIE	Res.	Res.	Res.	TBUE	TXSE	TIE
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw				rw	rw	rw

位 31:16 保留，必须保持复位值。

位 15 **NIE**：正常中断汇总使能 (Normal Interrupt Summary Enable)
该位置 1 时，会使能正常中断汇总。该位可使能 ETH_DMACSR 中的以下中断：
位 0：发送中断
位 2：发送缓冲区不可用
位 6：接收中断
位 11：提前接收中断
当此位复位时，将禁止正常中断汇总。

位 14 **AIE**：异常中断汇总使能 (Abnormal Interrupt Summary Enable)
该位置 1 时，会使能异常中断汇总。该位可使能 ETH_DMACSR 中的以下中断：
位 1：发送过程停止
位 7：Rx 缓冲区不可用
位 8：接收过程停止
位 9：接收看门狗超时
位 10：提前发送中断
位 12：致命总线错误
当此位复位时，将禁止异常中断汇总。

位 13 **CDEE**：上下文描述符错误使能 (Context Descriptor Error Enable)
该位与 AIE 位同时置 1 时，会使能上下文描述符错误中断。该位复位时，会禁止上下文描述符错误中断。

位 12 **FBEE**：致命总线错误使能 (Fatal Bus Error Enable)
该位与 AIE 位同时置 1 时，会使能致命总线错误中断。该位复位时，会禁止致命总线错误中断。

位 11 **ERIE**：提前接收中断使能 (Early Receive Interrupt Enable)
该位与 NIE 位同时置 1 时，会使能提前接收中断。该位复位时，会禁止提前接收中断。

位 10 **ETIE**：提前发送中断使能 (Early Transmit Interrupt Enable)
该位与 AIE 位同时置 1 时，会使能提前发送中断。该位复位时，会禁止提前发送中断。

位 9 **RWTE**：接收看门狗超时使能 (Receive Watchdog Timeout Enable)
该位与 AIE 位同时置 1 时，会使能接收看门狗超时中断。该位复位时，会禁止接收看门狗超时中断。

位 8 **RSE**：接收停止使能 (Receive Stopped Enable)
该位与 AIE 位同时置 1 时，会使能接收停止中断。该位复位时，会禁止接收停止中断。

位 7 **RBUE**：接收缓冲区不可用使能 (Receive Buffer Unavailable Enable)
该位与 AIE 位同时置 1 时，会使能接收缓冲区不可用中断。该位复位时，会禁止接收缓冲区不可用中断。

位 6 **RIE**：接收中断使能 (Receive Interrupt Enable)
该位与 NIE 位同时置 1 时，会使能接收中断。该位复位时，会禁止接收中断。

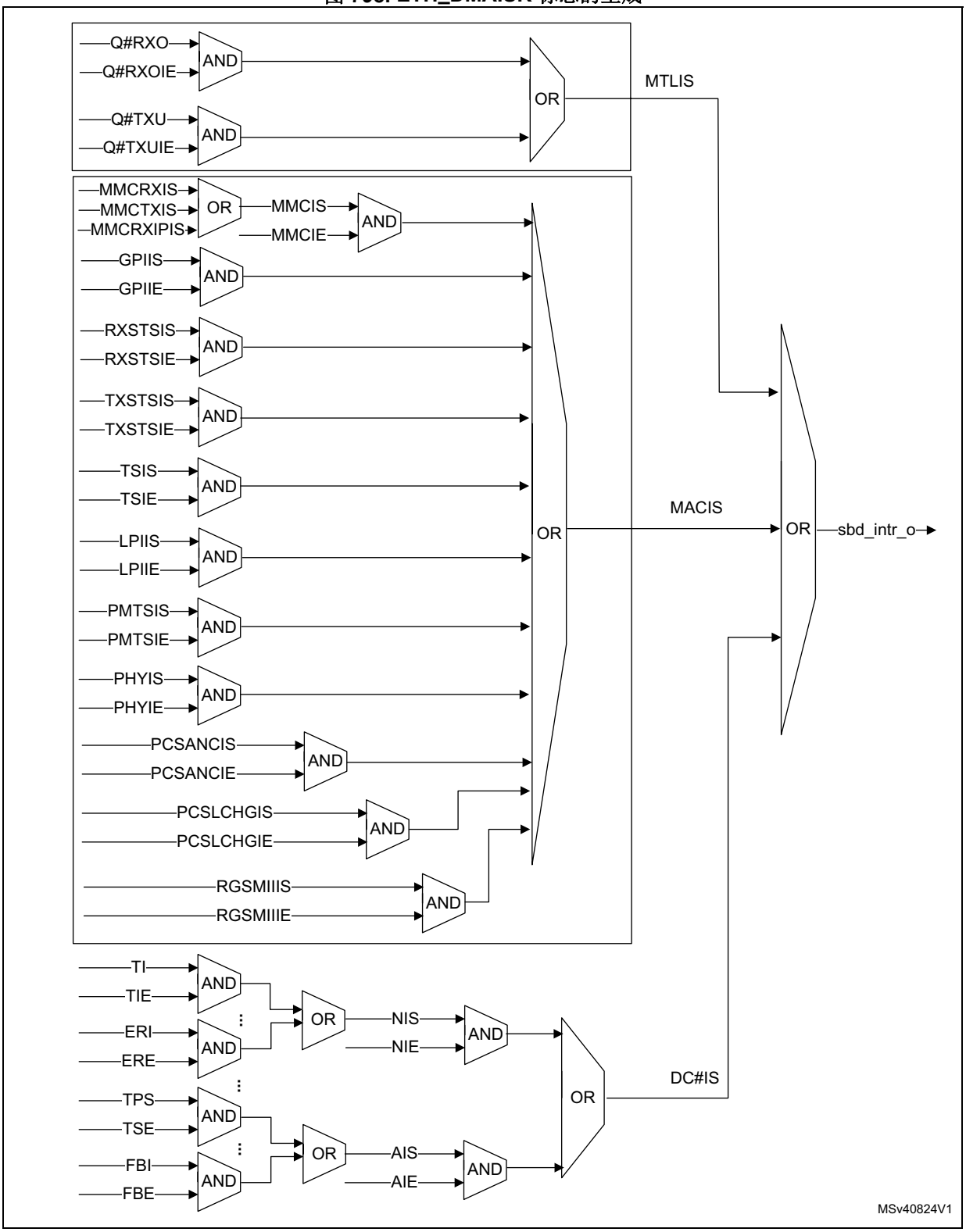
位5:3 保留，必须保持复位值。

位 2 **TBUE**：发送缓冲区不可用使能 (Transmit Buffer Unavailable Enable)
该位与 NIE 位同时置 1 时，会使能发送缓冲区不可用中断。该位复位时，会禁止发送缓冲区不可用中断。

位 1 **TXSE**：发送停止使能 (Transmit Stopped Enable)
该位与 AIE 位同时置 1 时，会使能发送停止中断。该位复位时，会禁止发送停止中断。

位 0 **TIE**：发送中断使能 (Transmit Interrupt Enable)
该位与 NIE 位同时置 1 时，会使能发送中断。该位复位时，会禁止发送中断。

图 798. ETH_DMAISR 标志的生成



通道 Rx 中断看门狗定时器寄存器 (ETH_DMACRxIWTR)

Channel Rx interrupt watchdog timer register

偏移地址：0x1138

复位值：0x0000 0000

接收中断看门狗定时器寄存器指示来自 DMA 的接收中断 (RI) 的看门狗超时。向该寄存器中写入用非零值时，会针对 ETH_DMACSR 寄存器的 RI 位使能看门狗定时器。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RWT[7:0]							
								rw							

位 31:8 保留，必须保持复位值。

位 7:0 **RWT[7:0]**：接收中断看门狗定时器计数 (Receive Interrupt Watchdog Timer Count)

该字段指示系统时钟周期数乘以 RWTU 字段中指示的系数之后的时间值，即看门狗定时器的设定时间值。

看门狗定时器会在 Rx DMA 结束数据包传输之后由编程设定的值触发，但 ETH_DMACSR 中的 RI 位未置 1，因为相应描述符中的中断使能位 RDES3[30] 已置 1。

当看门狗定时器计时结束时，RI 位置 1 且定时器停止。由于 RI 根据每个接收数据包的中断使能位 RDES3[30] 进行自动设置而使 RI 位设为高电平时，看门狗定时器复位。

通道当前应用程序发送描述符寄存器 (ETH_DMACCATxDR)

Channel current application transmit descriptor register

偏移地址：0x1144

复位值：0x0000 0000

通道当前应用程序发送描述符寄存器指向 DMA 所读取的当前发送描述符。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CURTDESAPTR															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CURTDESAPTR															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:0 **CURTDESAPTR[31:0]**：应用程序发送描述符地址指针 (Application Transmit Descriptor Address Pointer)

在 Tx 操作期间，DMA 会更新该指针。该指针在复位时清零。

通道当前应用程序接收描述符寄存器 (ETH_DMACCARxDR)

Channel current application receive descriptor register

偏移地址：0x114C

复位值：0x0000 0000

通道当前应用程序接收描述符寄存器指向 DMA 所读取的当前接收描述符。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CURRDESAPTR															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CURRDESAPTR															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:0 **CURRDESAPTR[31:0]**：应用程序接收描述符地址指针 (Application Receive Descriptor Address Pointer)

在 Rx 操作期间，DMA 会更新该指针。该指针在复位时清零。

通道当前应用程序发送缓冲区寄存器 (ETH_DMACCATxBR)

Channel current application transmit buffer register

偏移地址：0x1154

复位值：0x0000 0000

通道当前应用程序发送缓冲区地址寄存器指向 DMA 所读取的当前 Tx 缓冲区地址。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CURTBUFAPTR															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CURTBUFAPTR															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:0 **CURTBUFAPTR[31:0]**：应用程序发送缓冲区地址指针 (Application Transmit Buffer Address Pointer)

在 Tx 操作期间，DMA 会更新该指针。该指针在复位时清零。

通道当前应用程序接收缓冲区寄存器 (ETH_DMACCARxBR)

Channel current application receive buffer register

偏移地址：0x115C

复位值：0x0000 0000

通道当前应用程序接收缓冲区地址寄存器指向 DMA 所读取的当前 Rx 缓冲区地址。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CURRBUFAPTR															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CURRBUFAPTR															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:0 **CURRBUFAPTR[31:0]**：应用程序接收缓冲区地址指针 (Application Receive Buffer Address Pointer)

在 Rx 操作期间，DMA 会更新该指针。该指针在复位时清零。

通道状态寄存器 (ETH_DMACSR)

Channel status register

偏移地址：0x1160

复位值：0x0000 0000

软件驱动程序（应用程序）在中断服务程序或轮询期间读取状态寄存器，以确定 DMA 的状态。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REB[2:0]			TEB[2:0]		
										r			r		
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NIS	AIS	CDE	FBE	ERI	ETI	RWT	RPS	RBU	RI	Res.	Res.	Res.	TBU	TPS	TI
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw				rw	rw	rw

位 31:22 保留，必须保持复位值。

位 21:19 **REB[2:0]**：Rx DMA 错误位 (Rx DMA Error Bits)
该字段指示导致总线错误的错误类型。例如，AHB 接口上的错误响应。
位 21
1：Rx DMA 传输数据时出错
0：Rx DMA 传输数据时未出错
位 20
1：访问描述符时出错
0：访问数据缓冲区时出错
位 19
1：读取传输时出错
0：写入传输时出错
只有在 FBE 位置 1 时，该字段才有效。此字段不会产生中断。

位 18:16 **TEB[2:0]**：Tx DMA 错误位 (Tx DMA Error Bits)
该字段指示导致总线错误的错误类型。例如，AHB 接口上的错误响应。
位 18
1：Tx DMA 传输数据时出错
0：Tx DMA 传输数据时未出错
位 17
1：访问描述符时出错
0：访问数据缓冲区时出错
位 16
1：读取传输时出错
0：写入传输时出错
只有在 FBE 位置 1 时，该字段才有效。此字段不会产生中断。

位 15 **NIS**：正常中断汇总 (Normal Interrupt Summary)
当使能 ETH_DMACIER 寄存器中对应的中断位时，正常中断汇总位的值是以下位的逻辑或运算结果：
位 0：发送中断
位 2：发送缓冲区不可用
位 6：接收中断
位 11：提前接收中断
只有未屏蔽位（在 ETH_DMACIER 寄存器中将相应中断使能置 1 的中断）会影响正常中断汇总位。
它是黏着位。每当导致 NIS 置 1 的对应位被清零时，必须同时将该位也清零（通过向此位写入 1）。

位 14 **AIS**：异常中断汇总 (Abnormal Interrupt Summary)
当使能 ETH_DMACIER 寄存器中对应的中断位时，异常中断汇总位的值是以下位的逻辑或运算结果：
位 1：发送过程停止
位 7：接收缓冲区不可用
位 8：接收过程停止
位 10：提前发送中断
位 12：致命总线错误
位 13：上下文描述符错误
只有未屏蔽的位会影响异常中断汇总位。
它是黏着位。每当导致 AIS 置 1 的对应位被清零时，必须同时将该位也清零（通过向此位写入 1）。

位 13 **CDE**：上下文描述符错误 (Context Descriptor Error)
此位指示 DMA Tx 引擎接收到数据包中间的上下文描述符（在中间描述符中），并且 DMA Tx 引擎将其忽略。

位 12 **FBE**：致命总线错误 (Fatal Bus Error)
此位指示发生了总线错误（如 EB 字段中所述）。此位置 1 后，对应的 DMA 通道引擎会禁止所有总线访问。

位 11 **ERI**：提前接收中断 (Early Receive Interrupt)
此位指示 DMA 已填满数据包的首个数据缓冲区。该寄存器的 RI 位自动清零此位。

位 10 **ETI**：提前发送中断 (Early Transmit Interrupt)
此位指示要发送的数据包已完全传输到 MTL Tx FIFO。

位 9 **RWT**：接收看门狗超时 (Receive Watchdog Timeout)
当接收到长度大于 2,048 字节（如果使能巨型数据包模式，则为 10,240 字节）的数据包时，该位将置 1。

位 8 **RPS**：接收过程停止 (Receive Process Stopped)
当 Rx 过程进入停止状态时，此位将置 1。

位 7 **RBU**：接收缓冲区不可用 (Receive Buffer Unavailable)
此位指示应用程序拥有接收列表中的下一个描述符，DMA 无法获取。Rx 过程进入挂起状态。要恢复处理 Rx 描述符，应用程序应更改描述符的拥有关系，然后发出接收轮询要求命令。如果未发出该命令，则当接收到下一个识别的传入数据包时，Rx 过程会恢复。在环形模式下，应用程序应将通道的接收描述符尾指针寄存器前移。只有当 DMA 拥有前一 Rx 描述符时，该位才会置 1。

位 6 **RI**：接收中断 (Receive Interrupt)
此位指示数据包接收过程已完成。数据包接收过程完成时，最后一个描述符中的 RDES1 的位 31 复位，并且会在该描述符中更新特定数据包状态信息。
接收保持运行状态。

位 5:3 保留，必须保持复位值。

位 2 **TBU**：发送缓冲区不可用 (Transmit Buffer Unavailable)
此位指示应用程序拥有发送列表中的下一个描述符，DMA 无法获取。发送会进入挂起状态。DMA_Debug_Status0 寄存器的 TPS0 字段对发送过程状态转换进行说明。
要恢复处理发送描述符，应用程序应进行以下操作：
1. 通过将 TDES3 的位 31 置 1 来更改描述符的拥有关系。
2. 发出发送轮询要求命令。
对于环形模式，应用程序应将通道的发送描述符尾指针寄存器前移。

位 1 **TPS**：发送过程停止 (Transmit Process Stopped)
当发送停止时，此位置 1。

位 0 **TI**：发送中断 (Transmit Interrupt)
此位指示数据包发送过程已完成。发送过程完成时，最后一个描述符中的 TDES3 的位 31 复位，并且会在该描述符中更新特定数据包状态信息。

通道丢失帧计数寄存器 (ETH_DMACMFCR)

Channel missed frame count register

偏移地址：0x116C

复位值：0x0000 0000

该寄存器包含 DMA 因总线错误或因编程 ETH_DMACRxCR 寄存器中的 RPF 字段而丢弃的数据包计数器计数。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MFCO	Res.	Res.	Res.	Res.	MFC[10:0]										
r					r	r	r	r	r	r	r	r	r	r	r

位 31:16 保留，必须保持复位值。

位 15 **MFCO**：MFC 计数器的上溢状态 (Overflow status of the MFC Counter)
该位置 1 时，MFC 计数器不会进一步
递增。对此寄存器执行读操作时此位清零。

位 14:11 保留，必须保持复位值。

位 10:0 **MFC[10:0]**：丢弃数据包计数器 (Dropped Packet Counters)
该计数器指示 DMA 因总线错误或因
编程 ETH_DMACRxCR 寄存器中的 RPF 字段而丢弃的数据包计数器计数。对此寄存器执行读操作时计数器清零。

以太网 DMA 寄存器映射和复位值

表 536. ETH_DMA 通用寄存器映射和复位值

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x1000	ETH_DMAMR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	INTM		Res.	PR[2:0]			TXPR	Res.	Res.	Res.	Res.	Res.	Res.	Res.			DA	SWR
	Reset value															0	0		0	0	0	0							0	0	0	0	0
0x1004	ETH_DMASBMR	Res.	Res.	Res.	Res.	Res.				Res.	Res.	Res.	Res.	RD_OSR_LMT[1:0]				RB	MB	Res.	AAL	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FB
	Reset value													0	0	0	1	0	0		0												0
0x1008	ETH_DMAISR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MACIS	MTLIS	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DC0IS
	Reset value															0	0																0
0x100C	ETH_DMADSR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.				Res.				TPS0				RPS0				Res.	Res.	Res.	Res.	Res.	Res.	Res.	AXWHSTS
	Reset value																	0	0	0	0	0	0	0	0								0
0x1010 - 0x101C	Reserved																																

表 537. ETH_DMA_CH 寄存器映射和复位值

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x1100	ETH_DMACCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DSL			Res.	PBLX8	Res.	Res.	MSS													
	Reset value												0	0	0		0			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x1104	ETH_ DMACTxCR	Res.	Res.	Res.	Res.	Res.				Res.	Res.	TXPBL[5:0]					IPBL	Res.	Res.	TSE	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	OSF	Res.				ST
	Reset value											0	0	0	0	0	0	0			0								0				0
0x1108	ETH_ DMACRxCR	RPF	Res.	Res.	Res.	Res.				Res.	Res.	RXPBL[5:0]					Res.	RBSZ[10:0]												Res.	Res.	Res.	SR
	Reset value	0										0	0	0	0	0	0		0	0	0	0	0	0	0	0	0	0	0				0
0x110C - 01110	Reserved																																

表 537. ETH_DMA_CH 寄存器映射和复位值（续）

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x1114	ETH_ DMACTxDLAR	TDESLA																														Res.	Res.
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
0x1118	Reserved																																
0x111C	ETH_ DMACRxDLAR	RDESLA																														Res.	Res.
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
0x1120	ETH_ DMACTxDTPR	TDT																														Res.	Res.
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
0x1124	Reserved																																
0x1128	ETH_ DMACRxDTPR	RDT																														Res.	Res.
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		
0x112C	ETH_ DMACTxDRLR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TDRL[9:0]									
	Reset value																							0	0	0	0	0	0	0	0	0	0
0x1130	ETH_ DMACRxDRLR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RDRL[9:0]									
	Reset value																							0	0	0	0	0	0	0	0	0	0
0x1134	ETH_DMACIER	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	NIE	AIE	CDEE	FBEE	ERIE	ETIE	RWTE	RSE	RBUE	RIE	Res.	Res.	Res.	TBUE	TXSE	TIE
	Reset value																0	0	0	0	0	0	0	0	0	0				0	0	0
0x1138	ETH_ DMACRxIWTR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RWTU[1:0]		Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RWT[7:0]							
	Reset value															0	0									0	0	0	0	0	0	0	0
0x1140	Reserved																																
0x1144	ETH_ DMACCATxDR	CURTDESAPTR[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x1148	Reserved																																
0x0114C	ETH_ DMACCARxDR	CURRDESAPTR[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x1150	Reserved																																
0x1154	ETH_ DMACCATxBR	CURTBUFAPTR[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x1158	Reserved																																
0x115C	ETH_ DMACCARxBR	CURRBUFAPTR[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x1160	ETH_DMACSR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REB[2:0]			TEB[2:0]			NIS	AIS	CDE	FBE	ERI	ETI	RWT	RPS	RBU	RI	Res.	Res.	Res.	TBU	TPS	TI
	Reset value											0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0				0	0	0

表 537. ETH_DMA_CH 寄存器映射和复位值（续）

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x1164 - 0x1168	Reserved																																
0x116C	ETH_ DMACMFCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MFCO	Res.	Res.	Res.	Res.	MFC[10:0]										
	Reset value																	0					0	0	0	0	0	0	0	0	0	0	0

有关寄存器边界地址的信息，请参见*第 100 页的第 2.2.2 节*。

58.11.3 以太网 MTL 寄存器

工作模式寄存器 (ETH_MTLOMR)

Operating mode Register

偏移地址：0x0C00

复位值：0x0000 0000

工作模式寄存器将建立发送和接收工作模式及命令。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	CNTCLR	CNTPRST	Res.	Res.		Res.	Res.	Res.	DTXSTS	Res.
						rw	rw							rw	

位 31:10 保留，必须保持复位值。

位 9 CNTCLR：计数器复位 (Counters Reset)
该位置 1 时，所有计数器都将复位。该位在 1 个时钟周期后自动清零。
如果该位和 CNT_PRESET 位同时置 1，则 CNT_PRESET 优先级较高。

位 8 CNTPRST：计数器预设 (Counters Preset)
该位置 1 时：
- ETH_MTLTxQUR 寄存器被初始化/预设为 0x7F0。
- ETH_MTLRxQMPOCR 寄存器中丢失的数据包和上溢数据包计数器被初始化/预设为 0x7F0。

位 7 保留，必须保持复位值。

位 6:2 保留，必须保持复位值。

位 1 DTXSTS：丢弃发送状态 (Drop Transmit Status)
该位置 1 时，在 MTL 中会丢弃从 MAC 接收到的 Tx 数据包状态。该位复位时，从 MAC 接收到的 Tx 数据包状态会被转发到应用程序。
如果在配置内核时选择 MTL 功能中的禁用发送状态，则该位将被保留并且为只读位。

位 0 保留，必须保持复位值。

中断状态寄存器 (ETH_MTLISR)

Interrupt status Register

偏移地址：0x0C20

复位值：0x0000 0000

软件驱动程序（应用程序）在中断服务程序或轮询期间读取该寄存器，以确定 MTL 队列和 MAC 的中断状态。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Q0IS
															r

位 31:1 保留，必须保持复位值。

位 0 Q0IS：队列中断状态 (Queue interrupt status)

此位指示队列已生成中断。要复位该位，请读取 ETH_MTLQICSR 寄存器，以识别中断原因并清除中断源。

Tx 队列工作模式寄存器 (ETH_MTLTxQOMR)

Tx Queue operating mode Register

偏移地址：0x0D00

复位值：0x0007 0008

队列发送工作模式寄存器将建立发送队列工作模式及命令。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	TQS								
							rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TTC[2:0]			TXQEN[1:0]		TSF	FTQ
									rw			r		rw	rw

位 31:25 保留，必须保持复位值。

位 24:16 **TQS**：发送队列大小 (Transmit Queue Size)

该字段指示分配的发送队列大小（以 256 字节块计）。第十六位是此字段的起始位。此字段的宽度取决于配置中选择的 Tx 存储器大小。例如，如果存储器大小为 2048，则该字段的宽度为 3 位：

LOG2(2048/256) = LOG2(8) = 3 位

TQS = 0 对应于 256 字节。

位 15:7 保留，必须保持复位值。

位 6:4 **TTC[2:0]**：发送阈值控制 (Transmit Threshold Control)

这些位控制 MTL Tx 队列的阈值级别。当 MTL Tx 队列中的数据包大小大于阈值时启动发送。此外，还会发送长度小于阈值的完整数据包。这些位只有在 TSF 位复位后才能使用。

000: 32
001: 64
010: 96
011: 128
100: 192
101: 256
110: 384
111: 512

位 3:2 **TXQEN[1:0]**：发送队列使能 (Transmit Queue Enable)

此字段用于使能/禁止发送队列。

00：未使能
01：保留
10：使能
11：保留

注： 在多个 Tx 队列配置中，默认情况下所有队列均处于禁用状态。通过编程此字段使能 Tx 队列。

位 1 **TSF**：发送存储并转发 (Transmit Store and Forward)

当此位置 1 时，如果 MTL Tx 队列中有一个完整数据包，则会启动发送。当此位置 1 时，会忽略在该寄存器的位 [6:4] 中指定的 TTC 值。该位只有在已停止发送时才能更改。

位 0 **FTQ**：刷新发送队列 (Flush Transmit Queue)

该位置 1 时，Tx 队列控制器逻辑被复位为其默认值。因此，Tx 队列中的所有数据都将丢失或被刷新。刷新操作结束时该位在内部复位。在该位复位之前，不应对 ETH_MTLTxQOMR 寄存器进行写操作。MAC 发送器已接受的数据不会被刷新。其会被安排进行发送，并且会导致下溢且发送的数据包过短。

注： 仅当 Tx 队列为空并且应用程序已接受所有已发送数据包的挂起 Tx 状态时，才会完成刷新操作。要完成此刷新操作，PHY Tx 时钟 (eth_mii_tx_clk) 应处于活动状态。

Tx 队列下溢寄存器 (ETH_MTLTxQUR)

Tx Queue underflow Register

偏移地址：0x0D04

复位值：0x0000 0000

队列下溢计数器寄存器包含针对以下数据包的计数器：因发送队列下溢而被中止的数据包，以及因接收队列数据包刷新而丢失的数据包。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	UFCNTOVF	UFFRMCNT[10:0]										
				r	r										

位 31:12 保留，必须保持复位值。

位 11 **UFCNTOVF**：下溢数据包计数器的上溢位 (Overflow Bit for Underflow Packet Counter)
每次 Tx 队列下溢数据包计数器字段上溢（即，已超出最大计数）时，该位都会置 1。在这种情况下，上溢数据包计数器将复位为全零值，该位指示发生翻转。

位 10:0 **UFFRMCNT[10:0]**：下溢数据包计数器 (Underflow Packet Counter)
此字段指示控制器因 Tx 队列下溢而中止的数据包数。每次 MAC 因下溢而中止传出的数据包时，该计数器都会递增。对此寄存器执行读操作（mci_be_i[0] 为 1'b1）时此计数器清零。

Tx 队列调试寄存器 (ETH_MTLTxQDR)

Tx queue debug Register

偏移地址：0x0D08

复位值：0x0000 0000

队列发送调试寄存器提供与发送队列相关的各个块的调试状态。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	STXSTSF[2:0]			Res.	PTXQ[2:0]		
									r				r		
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TXSTS FSTS	TXQ STS	TWC STS	TRCSTS[1:0]		TXQPA USED
										r	r	r	r		r

位 31:23 保留，必须保持复位值。

位 22:20 **STXSTSF[2:0]**：队列的 Tx 状态 FIFO 中的状态字数 (Number of Status Words in Tx Status FIFO of Queue)

此字段指示该队列的 Tx 状态 FIFO 中的当前状态数。如果在配置内核时选择 MTL 功能中的禁用发送状态，则该字段将被保留。

当 ETH_MTLOMR 寄存器的 DTXSTS 位置 1 时，该字段不反映 Tx 状态 FIFO 中的状态字数。

位 19 保留，必须保持复位值。

位 18:16 **PTXQ[2:0]**：发送队列中的数据包数 (Number of Packets in the Transmit Queue)

此字段指示 Tx 队列中的当前数据包数。如果在配置内核时选择 MTL 功能中的禁用发送状态，则该字段将被保留。

当 ETH_MTLOMR 寄存器的 DTXSTS 位置 1 时，该字段不反映发送队列中的数据包数。

位 15:6 保留，必须保持复位值。

位 5 **TXSTSFSTS**：MTL Tx 状态 FIFO 已满状态 (MTL Tx Status FIFO Full Status)

该位为高电平时，表示 MTL Tx 状态 FIFO 已满。因此，MTL 不能接受发送更多数据包。

如果在配置内核时选择 MTL 功能中的禁用发送状态，则该字段将被保留。

位 4 **TXQSTS**：MTL Tx 队列非空状态 (MTL Tx Queue Not Empty Status)

该位为高电平时，表示 MTL Tx 队列不为空，还留有一些数据要发送。

位 3 **TWCSTS**：MTL Tx 队列写控制器状态 (MTL Tx Queue Write Controller Status)

该位为高电平时，表示 MTL Tx 队列写控制器有效且正在向 Tx 队列传输数据。

位 2:1 **TRCSTS[1:0]**：MTL Tx 队列读控制器状态 (MTL Tx Queue Read Controller Status)

此字段指示 Tx 队列读控制器的状态：

00：空闲状态

01：读状态（将数据传输到 MAC 发送器）

10：等待来自 MAC 发送器的挂起 Tx 状态

11：因来自 MAC 的数据包中止请求而刷新 Tx 队列

位 0 **TXQPAUSED**：发送队列处于暂停状态 (Transmit Queue in Pause)

该位为高电平且使能 Rx 流控制时，表示 Tx 队列出于以下原因而处于暂停状态（仅在全双工模式下）：

– 当使能 PFC 时，接收到优先级分配给 Tx 队列的 PFC 数据包

– 当禁止 PFC 时，接收到 802.3x 暂停数据包

队列中断控制状态寄存器 (ETH_MTLQICSR)

Queue interrupt control status Register

偏移地址：0x0D2C

复位值：0x0000 0000

该寄存器包含队列中断的中断使能和状态位。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	RXOIE	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RXOVFIS
							rw								rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	TXUIE	Res.	Res.	Res.	Res.	Res.	Res.	Res	TXUNFIS
						rw	rw							rw	rw

位 31:25 保留，必须保持复位值。

位 24 **RXOIE**：接收队列上溢中断使能 (Receive Queue Overflow Interrupt Enable)
该位置 1 时，会使能接收队列上溢中断。该位复位时，会禁止接收队列上溢中断。

位 23:17 保留，必须保持复位值。

位 16 **RXOVFIS**：接收队列上溢中断状态 (Receive Queue Overflow Interrupt Status)
此位指示在接收数据包期间，接收队列发生上溢。如果部分数据包已传输到应用程序，则 RDES3[21] 中的上溢状态位置 1。应用程序向该位写入 1 时会将其清零。

位 15:9 保留，必须保持复位值。

位 8 **TXUIE**：发送队列下溢中断使能 (Transmit Queue Underflow Interrupt Enable)
该位置 1 时，会使能发送队列下溢中断。该位复位时，会禁止发送队列下溢中断。

位 7:1 保留，必须保持复位值。

位 0 **TXUNFIS**：发送队列下溢中断状态 (Transmit Queue Underflow Interrupt Status)
此位指示在发送数据包期间，发送队列发生下溢。发送会进入挂起状态，且下溢错误 TDES3[2] 置 1。应用程序向该位写入 1 时会将其清零。

Rx 队列工作模式寄存器 (ETH_MTLRxQOMR)

Rx queue operating mode register

偏移地址：0x0D30

复位值：0x0070 0000

队列接收工作模式寄存器将建立接收队列工作模式及命令。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RQS			Res.	Res.	Res.	RFD2
									r	r	r				rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RFD1	RF0	Res.	Res.	Res.	RFA			EHFC	DIS_TCP_EF	RSF	FEP	FUP	Res.	RTC[1:0]	
rw	rw				rw			rw	rw	rw	rw	rw		rw	

位 31:23	保留，必须保持复位值。
位 22:20	**RQS[2:0]**：接收队列大小 (Receive Queue Size) 该字段指示分配的接收队列大小（以 256 字节块计）。只有在 Rx 队列数大于 1 且复位值为 0x0 时，RQS 字段才为读写字段。
位 19:17	保留，必须保持复位值。
位 16:14	**RFD[2:0]**：用于禁用流控制的阈值（半双工和全双工模式下）(Threshold for Deactivating Flow Control (in half-duplex and full-duplex modes)) 这些位控制流控制在激活后的取消激活阈值（Rx 队列的填充级别）： 0：满值减去 1 KB，即，FULL 1 KB 1：满值减去 1.5 KB，即，FULL 1.5 KB 2：满值减去 2 KB，即，FULL 2 KB 3：满值减去 2.5 KB，即，FULL 2.5 KB 4：满值减去 2.5 KB，即，FULL 3 KB 5：满值减去 2.5 KB，即，FULL 3.5 KB 只有在流控制激活后，取消激活才有效。 *注：　编程时必须确保阈值为正数。* *EHFC 置为高电平时，只有当该寄存器的 RQS 字段确定的 Rx 队列大小等于或大于 4 KB 时，这些值才适用。* *对于给定的队列大小，取值范围介于 0 和 FULL 减去 (QSIZE - 0.5 KB) 的编码之间，所有其他值都是非法的。这里的术语 FULL 和 QSIZE 指的是由该寄存器的 RQS 字段确定的队列大小。*
位 13:11	保留，必须保持复位值。

位 10:8	**RFA[2:0]**：用于激活流控制的阈值（半双工和全双工）(Threshold for Activating Flow Control (in half-duplex and full-duplex)) 这些位控制流控制的激活阈值（Rx 队列的填充级别）： 有关此字段编码的更多信息，请参见 RFD。
位 7	**EHFC**：使能硬件流控制 (Enable Hardware Flow Control) 该位置 1 时，将基于 Rx 队列的填充级别使能流控制信号操作。复位时，将禁止流控制操作。当 Rx 队列大小小于 4 KB 时，该位不适用（将被保留并始终复位）。
位 6	**DIS_TCP_EF**：禁止丢弃 TCP/IP 校验和错误数据包 (Disable Dropping of TCP/IP Checksum Error Packets) 当此位置 1 时，如果数据包中仅存在由接收校验和减荷引擎检测出来的错误，则 MAC 不会丢弃它。对于这类数据包，仅封装的有效负载中有错误。MAC 接收到的以太网数据包无错误（包括 FCS 错误）。 当此位复位时，如果 FEP 位进行了复位，则会丢弃所有错误数据包。
位 5	**RSF**：接收队列存储并转发 (Receive Queue Store and Forward) 当此位置 1 时，只有在向 Rx 队列写入完整数据包后，以太网外设才会从中读取数据包，同时会忽略该寄存器的 RTC 字段。当此位复位时，Rx 队列工作在阈值（直通）模式下，具体取决于该寄存器的 RTC 字段指定的阈值。
位 4	**FEP**：转发错误数据包 (Forward Error Packets) 该位复位时，Rx 队列会丢弃带有错误状态（CRC 错误、接收错误、看门狗超时或上溢）的数据包。不过，如果某个数据包的起始字节（写）指针已传输到读控制器端（阈值模式下），则不会丢弃该数据包。 该位置 1 时，除过短错误数据包之外的所有数据包都将被转发到应用程序或 DMA。如果 RSF 位置 1，并且在写入部分数据包时 Rx 队列上溢，则无论该位的设置如何，该数据包都会被丢弃。不过，如果 RSF 位复位，并且在写入部分数据包时 Rx 队列上溢，则可能会将部分数据包转发到应用程序或 DMA。
位 3	**FUP**：转发过小的良好数据包 (Forward Undersized Good Packets) 当该位置 1 时，Rx 队列会转发过小的良好数据包（即，没有错误但长度小于 64 字节的数据包），其中包括填充字节和 CRC。当该位复位时，Rx 队列会丢弃所有小于 64 字节的数据包，除非因 Rx 阈值下限更低（例如 RTC = 01）而导致数据包已传输。
位 2	保留，必须保持复位值。
位 1:0	**RTC[1:0]**：接收队列阈值控制 (Receive Queue Threshold Control) 这些位控制 MTL Rx 队列的阈值级别（以字节为单位）： 00: 64 01: 32 10: 96 11: 128 当 MTL Rx 队列中的数据包大小大于阈值时，接收到的数据包将传输到应用程序或 DMA。此外，还会自动传输长度小于阈值的完整数据包。 只有在 RSF 位为零时，该字段才有效。当 RSF 位置 1 时，该字段将被忽略。

Rx 队列丢失数据包和上溢计数器寄存器 (ETH_MTLRxQMPOCR)

Rx queue missed packet and overflow counter register

偏移地址：0x=0D34

复位值：0x0000 0000

队列丢失数据包和上溢计数器寄存器包含针对以下数据包的计数器：因接收队列数据包刷新而丢失的数据包，以及因接收队列上溢而丢弃的数据包。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	MISCNTOVF	MISPKTCNT[10:0]										
				r	r										
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	OVFCNTOVF	OVFPKTCNT[10:0]										
				r	r										

位 31:28 保留，必须保持复位值。

位 27 **MISCNTOVF**：丢失数据包计数器上溢位 (Missed Packet Counter Overflow Bit)
该位置 1 时，表示 Rx 队列丢失数据包计数器超出最大限制。

位 26:16 **MISPKTCNT[10:0]**：丢失数据包计数器 (Missed Packet Counter)
该字段指示以太网外设因应用程序针对该队列将 ari_pkt_flush_i[] 置为有效而丢失的数据包数。对此寄存器执行读操作（mci_be_i[0] 为 1b1）时此计数器复位。
当 DMA 因缓冲区不可用而丢弃数据包时，该计数器会以 1 递增。

位 15:12 保留，必须保持复位值。

位 11 **OVFCNTOVF**：上溢计数器上溢位 (Overflow Counter Overflow Bit)
该位置 1 时，表示 Rx 队列上溢数据包计数器字段超出最大限制。

位 10:0 **OVFPKTCNT[10:0]**：上溢数据包计数器 (Overflow Packet Counter)
此字段指示以太网外设因接收队列上溢而丢弃的数据包数。每次以太网外设因上溢而丢弃一个传入数据包时，此计数器值都会递增。对此寄存器执行读操作（mci_be_i[0] 为 1）时此计数器复位。

Rx 队列调试寄存器 (ETH_MTLRxQDR)

Rx queue debug register

偏移地址：0x0D38

复位值：0x0000 0000

队列接收调试寄存器提供与接收队列相关的各个块的调试状态。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	PRXQ[13:0]													
		r													
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RXQSTS[1:0]		Res.	RRCSTS[1:0]		RWCSTS
										r			r		r

位 31:30 保留，必须保持复位值。

位 29:16 **PRXQ[13:0]**：接收队列中的数据包数 (Number of Packets in Receive Queue)
此字段指示 Rx 队列中的当前数据包数。该字段的理论最大值为 256KB/16B = 16K 个数据包，即，Max_Queue_Size/Min_Packet_Size。

位 15:6 保留，必须保持复位值。

位 5:4 **RXQSTS[1:0]**：MTL Rx 队列填充级别状态 (MTL Rx Queue Fill-Level Status)
该字段给出了 Rx 队列填充级别的状态：
00：Rx 队列为空
01：Rx 队列填充级别低于流控制取消激活阈值
10：Rx 队列填充级别高于流控制激活阈值
11：Rx 队列已满

位 3 保留，必须保持复位值。

位 2:1 **RRCSTS[1:0]**：MTL Rx 队列读控制器状态 (MTL Rx Queue Read Controller State)
该字段给出了 Rx 队列读控制器的状态：
00：空闲状态
01：正在读取数据包数据
10：正在读取数据包状态（或时间戳）
11：正在刷新数据包数据和状态

位 0 **RWCSTS**：MTL Rx 队列写控制器有效状态 (MTL Rx Queue Write Controller Active Status)
该位为高电平时，表示 MTL Rx 队列写控制器有效且正在将接收到的数据包传输到 Rx 队列。

以太网 MTL 寄存器映射和复位值

表 538. ETH_MTL 寄存器映射和复位值

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x0C00	ETH_MTLOMR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CNTCLR	CNTPRST	Res.	Res.	Res.	Res.	Res.	Res.	DTXSTS	Res.
	Reset value																							0	0							0	
0x0C04 - 0x0C1C	Reserved																																
0x0C20	ETH_MTLISR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res	Q0TXOIE
	Reset value																																0
0x0C24 - 0x0CFC	Reserved																																
0x0D00	ETH_ MTLTxQOMR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TQS[3:0]				Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TTC [2:0]			TXQEN[1:0]		TSF	FTQ
	Reset value													0	1	1	1										0	0	0	1	0	0	0
0x0D04	ETH_MTLTxQUR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	UFFRMCNT[10:0]										
	Reset value																					0	0	0	0	0	0	0	0	0	0	0	0
0x0D08	ETH_MTLTxQDR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	STXSTSF[2:0]			Res.	PTXQ			Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TXSTSFSTS	TXQSTS	TWCSTS	TRCSTS[1:0]		TXQPAUSED
	Reset value										0	0	0		0	0	0											0	0	0	0	0	0
0x0D0C - 0x0D14	Reserved																																
0x0D18 - 0x0D28	Reserved																																
0x0D2C	ETH_MTLQICSR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RXOIE	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RXOVFIS	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TXUIE	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TXUNFIS
	Reset value								0								0							0	0							0	0

表 538. ETH_MTL 寄存器映射和复位值（续）

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x0D30	ETH_ MTLRxQOMR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RQS2	RQS1	RQS0	Res.	Res.	Res.	RFD			Res.	Res.	Res.	RFA			EHFC	DIS_TCP_EF	RSF	FEP	FUP	Res.	RTC	
	Reset value										1	1	1				0	0	0				0	0	0	0	0	0	0	0		0	0
0x0D34	ETH_ MTLRxQMPOCR	Res.	Res.	Res.	Res.	MISCNTOVF	MISPKTCNT[10:0]										Res.	Res.	Res.	Res.	OVFCNTOVF	OVFPKTCNT[10:0]											
	Reset value					0	0	0	0	0	0	0	0	0	0	0	0					0	0	0	0	0	0	0	0	0	0	0	0
0x0D38	ETH_ MTLRxQDR	Res.	Res.	PRXQ[13:0]													Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RXQSTS[1:0]		Res.	RRCSTS[1:0]		RWCSTS
	Reset value			0	0	0	0	0	0	0	0	0	0	0	0	0	0										0	0		0	0	0
0xD3C 到 0xD7C	Reserved																																
0x0D68	Reserved																																

有关寄存器边界地址的信息，请参见*第 100 页的第 2.2.2 节*。

58.11.4 以太网 MAC 和 MMC 寄存器

工作模式配置寄存器 (ETH_MACCR)

Operating mode configuration register

偏移地址：0x0000

复位值：0x0000 0000

MAC 配置寄存器建立 MAC 的工作模式。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ARPEN	SARC[2:0]			IPC	IPG[2:0]			GPSLCE	S2KP	CST	ACS	WD	Res.	JD	JE
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw		rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	FES	DM	LM	ECRSFD	DO	DCRS	DR	Res.	BL[1:0]		DC	PRELEN[1:0]		TE	RE
	rw	rw	rw	rw	rw	rw	rw		rw	rw	rw	rw	rw	rw	rw

位 31 **ARPEN**：ARP 减荷使能 (ARP Offload Enable)

该位置 1 时，MAC 可以识别传入的 ARP 请求数据包，并会安排发送 ARP 数据包。它会将 ARP 数据包转发到应用程序，还会在 RxStatus 中指示事件。

该位复位时，MAC 接收器不会识别任何 ARP 数据包，并会在 RxStatus 中将其指示为类型帧。

仅当选择 Enable IPv4 ARP Offload（使能 IPv4 ARP 减荷）时，该位才可用。

位 30:28 **SARC[2:0]**：源地址插入或替换控制 (Source Address Insertion or Replacement Control)

该字段控制对所有已发送数据包的源地址插入或替换。位 30 指定用于源地址插入或替换的 MAC 地址寄存器（0 或 1）（基于位 [29:28] 的值）：

0x：mti_sa_ctrl_i 和 ati_sa_ctrl_i 输入信号控制 SA 字段生成。

10：

- 如果位 30 置 0，则 MAC 会将 MAC 地址 0 寄存器（MAC 寄存器 192 和 193）的内容插入到所有已发送数据包的 SA 字段中。
- 如果位 30 置 1，且选择了 Enable MAC Address Register 1（使能 MAC 地址寄存器 1）选项，则 MAC 会将 MAC 地址 1 寄存器（MAC 寄存器 194 和 195）的内容插入到所有已发送数据包的 SA 字段中。

11：

- 如果位 30 置 0，则 MAC 会替换所有已发送数据包 SA 字段中的 MAC 地址 0 寄存器（MAC 寄存器 192 和 193）的内容。
- 如果位 30 置 1，且使能了 MAC 地址寄存器 1，则 MAC 会替换所有已发送数据包 SA 字段中的 MAC 地址 1 寄存器（MAC 寄存器 194 和 195）的内容。

注：对此字段进行的更改仅在数据包起始时生效。如果在发送数据包时对该寄存器字段进行写操作，则只有后续数据包可使用更新值，即，当前数据包不使用更新值。

如果在配置内核时未选择 Enable SA and VLAN Insertion on Tx（在 Tx 端使能 SA 和 VLAN 插入）功能，则这些位将被保留并处于 RO 状态。

位 27 **IPC**：校验和减荷 (Checksum Offload)

该位置 1 时，会使能 IPv4 报头校验和检查以及 IPv4 或 IPv6 TCP、UDP 或 ICMP 有效负载校验和检查。该位复位时，将禁止接收器中的 COE 功能。

如果在配置内核时未使能 IP 校验和减荷功能，则该位保留且处于 RO 状态（具有默认值）。

第 3 层和第 4 层数据包过滤和使能拆分报头功能可自动选择接收端的 IPC 全校验和减荷引擎。使能其中的任一功能时，都必须将 IPC 位置 1。

位 26:24 **IPG[2:0]**：数据包间隙 (Inter-Packet Gap)

这些位控制发送期间数据包间的最小 IPG。

000：96 位时间

001：88 位时间

010：80 位时间

...

111：40 位时间

此最小 IPG 范围在全双工模式下有效。

在半双工模式下，最小 IPG 仅能配置为 64 位时间 (IPG = 100)。不考虑更低的值。

因激活背压而发送 JAM 模式时，MAC 不考虑最小 IPG。

上述功能（IPG 小于 96 位时间）仅在 ETH_MACECR 寄存器中的 EIPGEN 位复位时有效。EIPGEN 置 1 时，将按照 ETH_MACECR 寄存器的 EIPG 字段中给出的说明控制最小 IPG（大于 96 位时间）。

位 23 **GPSLCE**：大型数据包大小限制控制使能 (Giant Packet Size Limit Control Enable)

当该位置 1 时，MAC 会考虑 ETH_MACECR 寄存器的 GPSL 字段中的值，以将接收到的数据包声明为大型数据包。必须将该字段编程为大于 1,518 字节。否则，MAC 会将 1,518 字节视为大型数据包限值。

该位复位时，如果接收到的数据包大小大于 1,518 字节（对于带标记的数据包，为 1522 字节），则 MAC 会将其视为大型数据包。

看门狗超时限制、巨型数据包使能和 2K 数据包使能的优先级高于该位，即，对于接收到的数据包，如果在使能巨型数据包的情况下其大小大于 9,018 字节（对于带标记的数据包，为 9,022 字节）以及在使能 2K 数据包的情况下其大小大于 2,000 字节，则 MAC 会将其视为大型数据包。看门狗超时（如果使能）会在达到看门狗限值时终止接收到的数据包。因此，要获取大型数据包状态，编程的大型数据包限值应小于看门狗限值。

位 22 **S2KP**：2K 数据包的 IEEE 802.3as 支持 (IEEE 802.3as Support for 2K Packets)

该位置 1 时，MAC 将所有长度未超过 2,000 字节的数据包视为正常数据包。JE 位未置 1 时，MAC 将所有接收到的大小超过 2K 字节的数据包视为大型数据包。

该位复位且 JE 位未置 1 时，MAC 将所有接收到的大小超过 1,518 字节（对于带标记的数据包，为 1,522 字节）的数据包视为大型数据包。有关此位和 JE 位的设置会如何影响大型数据包状态的详细信息，请参见基于 S2KP 和 JE 位的大型数据包状态。

注： JE 位置 1 时，将该位置 1 对大型数据包状态无影响。

位 21 **CST**：类型数据包的 CRC 去除 (CRC stripping for Type packets)

该位置 1 时，在将数据包转发到应用程序之前，所有以太类型的数据包（类型字段大于 1,536）的最后四个字节 (FCS) 都会被去除和丢弃。在 MAC 接收器中使能 IP 校验和引擎（类型 1）时，此功能无效。只有在使能类型 2 校验和减荷引擎时，该功能才有效。

注： 有关 ACS 位和该位的设置会如何影响数据包长度的信息，请参见基于 CST 和 ACS 位的数据包长度。

位 20 **ACS**：自动 Pad 或 CRC 去除 (Automatic Pad or CRC Stripping)

该位置 1 时，MAC 仅在长度字段值小于 1,536 字节时去除传入数据包上的 Pad 或 FCS 字段。将所有已接收数据包传送给应用程序时，如果长度字段大于或等于 1,536 字节，将都不去除 Pad 或 FCS 字段。

该位复位时，MAC 将所有传入的数据包传送给应用程序，而不进行任何修改。

注： 有关 CST 位和该位的设置会如何影响数据包长度的信息，请参见表格“基于 CST 和 ACS 位的数据包长度”。

位 19 **WD**：禁止看门狗 (Watchdog Disable)

当该位置 1 时，MAC 禁止接收器上的看门狗定时器。MAC 可以接收多达 16,383 字节的数据包。

该位复位时，MAC 不允许接收超过 2,048 字节（JE 置为高电平时，为 10,240）的数据包。MAC 会截断在 2,048 字节之后接收到的所有字节。

位 18 保留，必须保持复位值

位 17 **JD**：禁止 Jabber (Jabber Disable)

该位置 1 时，MAC 禁止发送器上的 jabber 定时器。MAC 可以发送多达 16,383 字节的数据包。

该位复位时，如果应用程序在发送期间发送超过 2,048 字节的数据（如果 JE 置为高电平，则为 10,240），则 MAC 不会发送该数据包中的剩余字节。

位 16 **JE**：巨型数据包使能 (Jumbo Packet Enable)

该位置 1 时，MAC 允许 9,018 字节的巨型数据包（对于带 VLAN 标记的数据包，为 9,022 字节），且不会在 Rx 数据包状态中报告大型数据包错误。

位 15 保留，必须保持复位值

位 14 **FES**：MAC 速度 (MAC Speed)

该位选择 10/100 Mbps 速度模式：

0：10 Mbps

1：100 Mbps

在仅 1000 Mbps 配置中，该位为只读位，具有复位值。在仅 10 或 100 Mbps 或者默认 10/100 Mbps 配置中，该位为读写位。mac_speed_o[0] 信号反映了该位的值。

位 13 **DM**：双工模式 (Duplex Mode)

当该位置 1 时，MAC 在全双工模式下工作，此时它可以同时发送和接收。该位在仅全双工配置中处于 RO 状态，且默认值为 1'b1。

位 12 **LM**：回送模式 (Loopback Mode)

该位置 1 时，MAC 在 MII 回送模式下工作。回送正确工作需要 MII Rx 时钟输入 (eth_mii_rx_clk)，因为 Tx 时钟不是内部回送的。

位 11 **ECRSFD**：在全双工模式下进行发送之前使能载波侦听 (Enable Carrier Sense Before Transmission in Full-Duplex Mode)

该位置 1 时，MAC 发送器会在全双工模式下发送数据包之前检查 CRS 信号。仅当 CRS 信号为低电平时，MAC 才开始发送。

当该位复位时，MAC 发送器将忽略 CRS 信号的状态。

位 10 **DO**：禁止接收自身 (Disable Receive Own)

如果该位置 1，则在半双工模式下将 ETH_TX_EN 置为有效时，MAC 禁止接收数据包。如果该位复位，MAC 将接收 PHY 提供的所有数据包。

该位不适用于全双工模式。在仅全双工配置中，该位保留，并处于只读 (RO) 状态且具有默认值。

位 9 **DCRS**：在发送期间禁止载波侦听 (Disable Carrier Sense During Transmission)

该位置 1 时，MAC 发送器会在半双工模式下发送数据包期间忽略 (G)MII CRS 信号。因此，不会因在发送期间载波丢失或无载波而生成错误。

该位复位时，MAC 发送器会因载波侦听而生成错误。MAC 甚至会中止发送。

在仅全双工配置中，该位被保留且处于只读状态 (RO)。

位 8 **DR**：禁止重试 (Disable Retry)

该位置 1 时，MAC 仅尝试一次发送。当在 MII 接口发生冲突时，MAC 会忽略当前数据包发送，并以 Tx 数据包状态下过度冲突错误报告数据包中止。

该位复位时，MAC 会根据 BL 字段的设置进行重试。该位仅在半双工模式下适用。在仅全双工配置中，该位被保留且处于只读状态 (RO)。

位 7 保留，必须保持复位值

位 6:5 **BL[1:0]**：后退限制 (Back-Off Limit)

后退限制决定发生冲突后重试期间 MAC 在重新安排一次发送之前等待的随机整数 (r) 个时隙延迟（对于 1000 Mbps 为 4,096 位时间；对于 10/100 Mbps 为 512 位时间）。

00：k = min (n, 10)

01：k = min (n, 8)

10：k = min (n, 4)

11：k = min (n, 1)

其中 n = 重新发送尝试的次数

随机整数 r 的取值范围为 0 <= r < 2^k

该位仅在半双工模式下适用。在仅全双工配置中，该位被保留且处于只读状态 (RO)。

位 4 **DC**：检查延迟 (Deferral Check)

当该位置 1 时，便在 MAC 中使能了检查延迟功能。当 Tx 状态机在 10 或 100 Mbps 模式下延迟超过 24,288 位时间时，MAC 将发出数据包中止状态，同时在 Tx 数据包状态中将过度延迟错误位置 1。

如果将 MAC 配置为 1000 Mbps 操作模式，则延迟阈值为 155,680 位时间。当发送器准备好发送但因 MII 上存在有效载波侦听信号 (CRS) 而被阻止时延迟开始。

延迟时间是非累积性的。例如，如果发送器因为 CRS 信号有效而延迟 10,000 位时间，之后 CRS 信号变为无效，则会导致发送器进行发送时出现冲突。由于发生冲突，因此发送器需要后退，然后在完成后退之后再次延迟。在这种情况下，延迟定时器将复位为 0，并会重新启动。

当该位复位时，禁止延迟检查功能，MAC 发生延迟，直到 CRS 信号变为无效信号。

该位仅在半双工模式下适用。在仅全双工配置中，该位被保留且处于只读状态 (RO)。

位 3:2 **PRELEN[1:0]**：发送数据包的报头长度 (Preamble Length for Transmit packets)

这些位控制被添加到每个 Tx 数据包的开头的报头字节数。仅当 MAC 工作在全双工模式下时，才可缩短报头。

00：7 字节报头

01：5 字节报头

10：3 字节报头

11：保留

位 1 **TE**：发送器使能 (Transmitter Enable)

当该位置 1 时，使能 MAC 的 Tx 状态机，以便在 MII 接口上进行发送。该位复位时，MAC Tx 状态机会在完成当前数据包的发送后被禁止。Tx 状态机不会再发送任何数据包。

位 0 **RE**：接收器使能 (Receiver Enable)

当该位置 1 时，使能 MAC 的 Rx 状态机，以便从 MII 接口接收数据包。该位复位时，MAC Rx 状态机会在完成当前数据包的接收后被禁止。Rx 状态机不会再从 MII 接口接收任何数据包。

表 539 显示了 ETH_MACCR 寄存器的 S2KP 和 JE 位的设置对大型数据包状态的影响。

表 539. 基于 S2KP 和 JE 位的大型数据包状态

长度/类型字段	接收到的数据包长度	S2KP	JE	大型数据包状态
未标记的数据包	> 1,518	0	0	1
	> 2,000	1	0	1
	> 9,018	x	1	1
带 VLAN 标记的数据包	> 1,522	0	0	1
	> 2,000	1	0	1
	> 9,022	x	1	1
注： 对于所有其他组合，大型数据包状态均为 0。				

表 540 显示了 ETH_MACCR 寄存器的 CST 和 ACS 位的设置如何影响数据包长度是否包含 CRC 长度。

表 540. 基于 CST 和 ACS 位的数据包长度

接收校验和减荷引擎	接收到的数据包长度	CST	ACS	已完成 FCS 去除
IPCHKSUM_EN = 0 且 IPC_FULL_OFFLOAD = 0 或 IPCHKSUM_EN = 1 且 IPC_FULL_OFFLOAD = 1	< 1,536	x	0	否
		x	1	是（针对以太网数据包）
	>= 1,536	0	x	否
		1	x	是（针对类型数据包）
IPCHKSUM_EN = 1 且 IPC_FULL_OFFLOAD = 0	< 1,536	x	0	否
		x	1	是（针对以太网数据包）
	>= 1,536	x	x	否

扩展工作模式配置寄存器 (ETH_MACECR)

Extended operating mode configuration register

偏移地址：0x0004

复位值：0x0000 0000

MAC 扩展配置寄存器建立 MAC 的工作模式。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	EIPG[4:0]					EIPGEN	Res.	RESERVED_HDSMS[2:0]			Res.	USP	SPEN	DCRCC
		rw					rw		r				rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	GPSL[13:0]													
		rw													

位 31:30 保留，必须保持复位值

位 29:25 **EIPG[4:0]**：扩展数据包间隙 (Extended Inter-Packet Gap)

EIPGEN 位置 1 时，此字段中的值适用。该字段（作为最高有效位）以及 ETH_MACCR 中的 IPG 字段一起给出最小 IPG（大于 96 位时间，以 8 位时间为步长）：

{EIPG, IPG}

0x00：104 位时间

0x01：112 位时间

0x02：120 位时间

..

0xFF：2144 位时间

位 24 **EIPGEN**：扩展数据包间隙使能 (Extended Inter-Packet Gap Enable)

该位置 1 时，MAC 将 EIPG 字段和 ETH_MACCR 中的 IPG 字段一起解析为最小 IPG（大于 96 位时间，以 8 位时间为步长）。

该位复位时，MAC 将忽略 EIPG 字段，并将 ETH_MACCR 中的 IPG 字段解析为最小 IPG（小于等于 96 位时间，以 8 位时间为步长）。

注： 必须仅在全双工模式下工作时使能扩展数据包间隙功能。如果在半双工模式下使能，则可能会对背压功能和帧发送产生不良影响。

位 23 保留，必须保持复位值

位 22:20 **RESERVED_HDSMS[2:0]**：保留。

位 19 保留，必须保持复位值

位 18 **USP**：单播慢速协议数据包检测 (Unicast Slow Protocol Packet Detect)

该位置 1 时，MAC 会检测具有 ETH_MACA0HR 和 ETH_MACA0LR 寄存器所指定的站单播地址的慢速协议数据包。MAC 还会检测具有慢速协议多播地址 (01-80-C2-00-00-02) 的慢速协议数据包。

该位复位时，MAC 仅检测具有在 IEEE 802.3-2008 的第 5 节中所指定的慢速协议多播地址的慢速协议数据包。

位 17 **SPEN**：慢速协议检测使能 (Slow Protocol Detection Enable)

该位置 1 时，MAC 会处理慢速协议数据包（以太类型 0x8809）并提供 Rx 状态。MAC 会丢弃具有无效子类型的慢速协议数据包。

该位复位时，MAC 将所有无错误的慢速协议数据包转发到应用程序。MAC 将这类数据包视为正常类型数据包。

位 16 **DCRCC**：禁止对已接收数据包的 CRC 检查 (Disable CRC Checking for Received Packets)

该位置 1 时，MAC 接收器不会检查已接收数据包中的 CRC 字段。该位复位时，MAC 接收器将始终检查已接收数据包中的 CRC 字段。

位 15:14 保留，必须保持复位值

位 13:0 **GPSL[13:0]**：大型数据包大小限制 (Giant Packet Size Limit)

如果接收到的数据包大小大于在此字段中编程的值（以字节为单位），则 MAC 会将接收到的数据包声明为大型数据包。在此字段中编程的值必须大于或等于 1,518 字节。任何其他编程值均被视为 1,518 字节。

对于带 VLAN 标记的数据包，MAC 会将编程的值加上 4 个字节。选择 Enable Double VLAN Processing（使能双 VLAN 处理）选项时，MAC 会将带有双 VLAN 标记的数据包的编程值加上 8 个字节。将 ETH_MACCR 寄存器中的 GPSLCE 位置 1 时，该字段中的值适用。

数据包过滤控制寄存器 (ETH_MACPFR)

Packet filtering control register

偏移地址：0x0008

复位值：0x0000 0000

MAC 数据包过滤寄存器包含用于接收数据包的过滤控件。该寄存器的一些控件基于 MAC 的地址检查模块执行第一级地址过滤。第二级过滤基于其他控件（例如，传送不良数据包和传送控制数据包）对传入数据包过滤。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RA	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DNTU	IPFE	Res.	Res.	Res.	VTFE
rw										rw	rw				rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	HPF	SAF	SAIF	PCF[1:0]		DBF	PM	DAIF	HMC	HUC	PR
					rw	rw	rw	rw		rw	rw	rw	rw	rw	rw

位 31 **RA**：接收所有 (Receive All)

该位置 1 时，无论接收到的数据包是否通过地址过滤，MAC 接收器模块都会将它们全部传送至应用程序。SA 或 DA 过滤的结果在 Rx 状态字的相应位进行更新（通过或失败）。

该位复位时，接收器模块仅将通过 SA 或 DA 地址过滤的数据包传送至应用程序。

位 30:22 保留，必须保持复位值

位 21 **DNTU**：丢弃非 TCP/UDP over IP 数据包 (Drop Non-TCP/UDP over IP Packets)

该位置 1 时，MAC 会丢弃非 TCP/UDP over IP 数据包。MAC 仅转发由第 4 层过滤器处理的数据包。该位复位时，MAC 会转发所有非 TCP/UDP over IP 数据包。

如果未选择 Enable Layer 3 and Layer 4 Packet Filter（使能第 3 层和第 4 层数据包过滤器）选项，该位将保留（处于 RO 状态，且具有默认值）。

位 20 **IPFE**：第 3 层和第 4 层过滤器使能 (Layer 3 and Layer 4 Filter Enable)

该位置 1 时，MAC 会丢弃与使能的第 3 层和第 4 层过滤器不匹配的数据包。如果未使能第 3 层或第 4 层过滤器以进行匹配，则该位无任何影响。

该位复位时，无论第 3 层和第 4 层字段的匹配状态如何，MAC 都会转发所有数据包。

如果未选择 Enable Layer 3 and Layer 4 Packet Filter（使能第 3 层和第 4 层数据包过滤器）选项，该位将保留（处于 RO 状态，且具有默认值）。

位 19:17 保留，必须保持复位值

位 16 **VTFE**：VLAN 标记过滤器使能 (VLAN Tag Filter Enable)

该位置 1 时，MAC 将丢弃与 VLAN 标记不匹配的带 VLAN 标记的数据包。该位复位时，无论 VLAN 标记的匹配状态如何，MAC 都会转发所有数据包。

位 15:11 保留，必须保持复位值

位 10 **HPF**：散列或完美过滤器 (Hash or Perfect Filter)

此位置 1 时，如果数据包与完美过滤或散列过滤（通过 HMC 或 HUC 位设置）匹配，则地址过滤器会通过此数据包。

如果该位复位并且 HUC 或 HMC 位置 1，则只有在数据包与散列过滤器匹配时才会通过。如果未选择 Enable Address Filter Hash Table（使能地址过滤器散列表）选项，该位将保留（并处于 RO 状态）。

位 9 **SAF**：源地址过滤器使能 (Source Address Filter Enable)

该位置 1 时，MAC 会将所接收数据包的 SA 字段与使能的 SA 寄存器中编程的值进行比较。如果比较结果为失败，MAC 会丢弃数据包。

该位复位时，MAC 会根据 SA 地址比较结果，通过更新 Rx 状态的 SAF 位来将接收到的数据包转发到应用程序。

注： 根据 IEEE 规范，将保留 SA 的位 47。不过，MAC 会比较所有 48 位。在针对 SA 编程 MAC 地址寄存器时，软件驱动程序应考虑到这一点。

位 8 **SAIF**：SA 反向过滤 (SA Inverse Filtering)

当该位置 1 时，地址检查块在反向过滤模式下进行 SA 地址比较。如果数据包的 SA 与 SA 寄存器中编程的值匹配，则会将其标记为 SA 地址过滤失败。

当该位复位时，如果数据包的 SA 与 SA 寄存器中编程的值不匹配，则会将其标记为 SA 地址过滤失败。

位 7:6 **PCF[1:0]**：通过控制数据包 (Pass Control Packets)

这些位控制所有控制数据包（包括单播和多播暂停数据包）的转发。

00：MAC 会过滤掉所有控制数据包，阻止它们到达应用程序。

01：即使所有控制数据包均未通过地址过滤，MAC 也仍将它们转发给应用程序（暂停数据包除外）。

10：即使所有控制数据包均未通过地址过滤，MAC 也仍将它们转发给应用程序。

11：MAC 转发通过地址过滤的控制数据包。

位 5 **DBF**：禁止广播数据包 (Disable Broadcast Packets)

该位置 1 时，AFM 模块会阻止所有传入的广播数据包。此外，它还会覆盖所有其他过滤设置。

该位复位时，AFM 模块会通过所有接收到的广播数据包。

位 4 **PM**：通过所有多播 (Pass All Multicast)

该位置 1 时，指示通过接收到的带多播目标地址（目标地址字段的第一位是“1”）的所有数据包。该位复位时，多播数据包的过滤取决于 HMC 位。

位 3 **DAIF**：DA 反向过滤 (DA Inverse Filtering)

当该位置 1 时，地址检查块在反向过滤模式下对单播和多播数据包均进行 DA 地址比较。该位复位时，执行数据包的正常过滤。

位 2 **HMC**：散列多播 (Hash Multicast)

该位置 1 时，MAC 根据散列表对接收到的多播数据包执行目标地址过滤。

该位复位时，MAC 对多播数据包执行完美目标地址过滤，即，将 DA 字段与 DA 寄存器中编程的值进行比较。

如果未选择 Enable Address Filter Hash Table（使能地址过滤散列表）选项，该位将保留（且处于 RO 状态）。

位 1 **HUC**：散列单播 (Hash Unicast)

该位置 1 时，MAC 根据散列表对单播数据包执行目标地址过滤。

该位复位时，MAC 对单播数据包执行完美目标地址过滤，即，将 DA 字段与 DA 寄存器中编程的值进行比较。

如果未选择 Enable Address Filter Hash Table（使能地址过滤散列表）选项，该位将保留（且处于 RO 状态）。

位 0 **PR**：混合模式 (Promiscuous Mode)

当该位置 1 时，不论目标或源地址为何，地址过滤模块都会通过所有传入的数据包。PR 置 1 时，始终将 Rx 状态字的 SA 或 DA 过滤失败状态位清零。

看门狗超时寄存器 (ETH_MACWTR)

Watchdog timeout register

偏移地址：0x000C

复位值：0x0000 0000

看门狗超时寄存器用于控制已接收数据包的看门狗超时。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res	Res	Res	Res	Res	Res	Res	PWE	Res	Res	Res	Res	WTO[3:0]			
							rw					rw			

位 31:9 保留，必须保持复位值

位 8 **PWE**：可编程看门狗使能 (Programmable Watchdog Enable)

当该位置 1 且 ETH_MACCR 寄存器的 WD 位复位时，WTO 字段用作已接收数据包的看门狗超时。当该位清零时，通过设置 ETH_MACCR 寄存器的 WD 和 JE 位来控制已接收数据包的看门狗超时。

位 7:4 保留，必须保持复位值

位 3:0 **WTO[3:0]**：看门狗超时 (Watchdog Timeout)

当 PWE 位置 1 且 ETH_MACCR 寄存器的 WD 位复位时，该字段用作已接收数据包的看门狗超时。如果接收到的数据包的长度超过该字段的值，则此数据包会被终止并被声明为错误数据包。

编码如下：

0x0：2 KB

0x1：3 KB

0x2：4 KB

0x3：5 KB

..

0xC：14 KB

0xD：15 KB

0xE：16383 字节

0xF：保留

注： 当 PWE 位置 1 时，该字段的值应大于 1,522 (0x05F2)。否则，IEEE 802.3 指定的带标记的有效数据包会被声明为错误数据包，然后被丢弃。

散列表 0 寄存器 (ETH_MACHT0R)

Hash Table 0 register

偏移地址：0x0010

复位值：0x0000 0000

散列表寄存器 0 包含散列表（64 位）的前 32 位。

对于散列过滤，传入数据包中目标地址的内容通过 CRC 逻辑传送，CRC 寄存器的高六位用于对散列表内容进行索引。最高有效位决定要使用的寄存器（散列表寄存器 0 或 1）。

目标地址的散列值的计算方式如下：

1. 计算 DA 的 32 位 CRC（有关计算 CRC32 的步骤，请参见 IEEE 802.3 的第 3.2.8 节）。
2. 对第 1 步中获得的值执行按位反转。
3. 取第 2 步获得的值的高 7 或 8 位。

如果寄存器的相应位值为 1，则接受该数据包。否则拒绝该帧。如果 ETH_MACPFR 中的 PM 位置 1，则不论多播散列值为何，都会接受所有多播数据包。

如果将散列表寄存器配置为与 MII 时钟域双同步，则只有在写入散列表寄存器 X 寄存器的位 [31:24]（小端模式）或位 [7:0]（大端模式）时，才会触发同步。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
HT31T0[31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HT31T0[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:0 **HT31T0[31:0]**：MAC 散列表的前 32 位 (MAC Hash Table First 32 Bits)

该字段包含散列表的前 32 位，即位 [31:0]。

散列表 1 寄存器 (ETH_MACHT1R)

Hash Table 1 register

偏移地址：0x0014

复位值：0x0000 0000

散列表寄存器 1 包含散列表（64 位）的后 32 位。

对于散列过滤，传入数据包中目标地址的内容通过 CRC 逻辑传送，CRC 寄存器的高六位用于对散列表内容进行索引。最高有效位决定要使用的寄存器（散列表寄存器 0 或 1）。

目标地址的散列值的计算方式如下：

1. 计算 DA 的 32 位 CRC（有关计算 CRC32 的步骤，请参见 IEEE 802.3 的第 3.2.8 节）。
2. 对第 1 步中获得的值执行按位反转。
3. 取第 2 步获得的值的高 7 或 8 位。

如果寄存器的相应位值为 1，则接受该数据包。否则拒绝该帧。如果 ETH_MACPFR 中的 PM 位置 1，则不论多播散列值为何，都会接受所有多播数据包。

如果将散列表寄存器配置为与 MII 时钟域双同步，则只有在写入散列表寄存器 X 寄存器的位 [31:24]（小端模式）或位 [7:0]（大端模式）时，才会触发同步。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
HT63T32[31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HT63T32[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:0 **HT63T32[31:0]**：MAC 散列表的后 32 位 (MAC Hash Table Second 32 Bits)

该字段包含散列表的后 32 位，即位 [63:32]。

VLAN 标记寄存器 (ETH_MACVTR)

VLAN tag register

偏移地址：0x0050

复位值：0x0000 0000

VLAN 标记寄存器定义 IEEE 802.1Q VLAN 类型数据包。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
EIVLRXS	Res.	EIVLS[1:0]		ERIVLT	EDVLP	VTHM	EVLRXS	Res.	EVLS [1:0]		DOVLTC	ERSVLM	ESVL	VTIM	ETV
rw		rw		rw	rw	rw	rw		rw		rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VL[15:0]															
rw															

位 31 **EIVLRXS**：在 Rx 状态中使能内部 VLAN 标记 (Enable Inner VLAN Tag in Rx Status)
该位置 1 时，MAC 会在 Rx 状态中提供内部 VLAN 标记。该位复位时，MAC 不会在 Rx 状态中提供内部 VLAN 标记。

位 30 保留

位 29:28 **EIVLS[1:0]**：在接收端使能内部 VLAN 标记去除 (Enable Inner VLAN Tag Stripping on Receive)
此字段指示在接收到的数据包中对内部 VLAN 标记进行的去除操作：
00：不去除
01：VLAN 过滤通过时进行去除操作
10：VLAN 过滤失败时进行去除操作
11：始终进行去除操作

位 27 **ERIVLT**：使能内部 VLAN 标记 (Enable Inner VLAN Tag)
该位和 EDVLP 字段均置 1 时，MAC 接收器会使能针对内部 VLAN 标记（如果存在）的操作。该位复位时，MAC 接收器会使能针对外部 VLAN 标记（如果存在）的操作。ERSVLM 位确定使能哪种 VLAN 类型进行过滤或匹配。

位 26 **EDVLP**：使能双 VLAN 处理 (Enable Double VLAN Processing)
该位置 1 时，MAC 将支持在 Tx 和 Rx 端处理最多两个 VLAN 标记（如果存在）。该位复位时，MAC 将支持在 Tx 和 Rx 端处理最多一个 VLAN 标记（如果存在）。

位 25 **VTHM**：VLAN 标记散列表匹配使能 (VLAN Tag Hash Table Match Enable)
如果该位置 1，则使用 VLAN 标记的 CRC 的四个最高有效位来索引 ETH_MACVLANHTR 寄存器的内容。VLAN 散列表寄存器的值 1 对应于索引，表示数据包与 VLAN 散列表匹配。
ETV 位置 1 时，使用 12 位 VLAN 标识符 (VID) 的 CRC 进行比较。ETV 位复位时，使用 16 位 VLAN 标记的 CRC 进行比较。
该位复位时，不执行 VLAN 散列匹配操作。如果未使能 VLAN 散列功能，则该位保留（处于 RO 状态，具有默认值）。

位 24 **EVLRXS**：在 Rx 状态中使能 VLAN 标记 (Enable VLAN Tag in Rx status)
该位置 1 时，MAC 会在 Rx 状态中提供外部 VLAN 标记。该位复位时，MAC 不会在 Rx 状态中提供外部 VLAN 标记。

位 23 保留，必须保持复位值

位 22:21 **EVLS[1:0]**：在接收端使能 VLAN 标记去除 (Enable VLAN Tag Stripping on Receive)

此字段指示在接收到的数据包中对外部 VLAN 标记进行的去除操作：

00：不去除

01：VLAN 过滤通过时进行去除操作

10：VLAN 过滤失败时进行去除操作

11：始终进行去除操作

位 20 **DOVLTC**：禁止 VLAN 类型检查 (Disable VLAN Type Check)

该位置 1 时，MAC 不检查由 ERIVLT 位指定的 VLAN 标记是 S-VLAN 类型还是 C-VLAN 类型。该位复位时，只有当 VLAN 标记类型与 ERSVLM 位指定的类型相似时，MAC 才会对 ERIVLT 位指定的 VLAN 标记进行过滤或匹配。

位 19 **ERSVLM**：使能接收 S-VLAN 匹配 (Enable Receive S-VLAN Match)

该位置 1 时，MAC 接收器会使能针对 S-VLAN（类型 = 0x88A8）数据包的过滤或匹配。该位复位时，MAC 接收器会使能针对 C-VLAN（类型 = 0x8100）数据包的过滤或匹配。

ERIVLT 位确定用于过滤或匹配的 VLAN 标记位置。

位 18 **ESVL**：使能 S-VLAN (Enable S-VLAN)

该位置 1 时，MAC 发送器和接收器会将 S-VLAN 数据包（类型 = 0x88A8）视为带 VLAN 标记的有效数据包。

位 17 **VTIM**：VLAN 标记反向匹配使能 (VLAN Tag Inverse Match Enable)

该位置 1 时，会使能 VLAN 标记反向匹配。不含匹配 VLAN 标记的数据包被标记为匹配。该位复位时，会使能 VLAN 标记完美匹配。含有匹配 VLAN 标记的数据包被标记为匹配。

位 16 **ETV**：使能 12 位 VLAN 标记比较 (Enable 12-Bit VLAN Tag Comparison)

该位置 1 时，使用 12 位 VLAN 标识符进行比较和过滤，而不使用完整的 16 位 VLAN 标记。VLAN 标记的位 [11:0] 与所接收带 VLAN 标记的数据包中的相应字段进行比较。类似地，使能时，只有所接收数据包中的 12 位 VLAN 标记用于基于散列的 VLAN 过滤。

该位复位时，所接收 VLAN 数据包的第 15 个和第 16 个字节的所有 16 位都用于比较和 VLAN 散列过滤。

位 15:0 **VL[15:0]**：用于接收数据包的 VLAN 标记标识符 (VLAN Tag Identifier for Receive Packets)

该字段包含用于识别 VLAN 数据包的 802.1Q VLAN 标记。该 VLAN 标记标识符将与接收到的数据包的第 15 个和第 16 个字节进行比较，以识别 VLAN 数据包。下面对该字段的各个位进行了说明：

位 [15:13]：用户优先级

位 12：标准格式指示符 (CFI) 或丢弃合法指示符 (DEI)

位 [11:0]：VLAN 标记的 VLAN 标识符 (VID) 字段

ETV 位置 1 时，仅使用 VID 进行比较。

如果该字段（ETV 置 1 时为 [11:0]）全部为零，则 MAC 不会检查用于 VLAN 标记比较的第 15 个和第 16 个字节，而是将类型字段值为 0x8100 或 0x88a8 的所有数据包均声明为 VLAN 数据包。

VLAN 散列表寄存器 (ETH_MACVHTR)

VLAN Hash table register

偏移地址：0x0058

复位值：0x0000 0000

ETH_MACHT1R 寄存器的 ERSVLM 位置 1 时，16 位 VLAN 散列表寄存器用于基于 VLAN 标记进行组地址过滤。对于散列过滤，传入数据包中的 16 位 VLAN 标记或 12 位 VLAN ID（取决于 ETH_MACVTR 寄存器的 ETV 位）的内容通过 CRC 逻辑传送。计算得出的 CRC 的高四位用于索引 VLAN 散列表的内容。例如，散列值 1000 用于选择 VLAN 散列表的位 8。

目标地址的散列值的计算方式如下：

1. 计算 VLAN 标记或 ID 的 32 位 CRC（有关计算 CRC32 的步骤，请参见 IEEE 802.3 的第 3.2.8 节）。
2. 对第 1 步中获得的值执行按位反转。
3. 取第 2 步获得的值的高四位。

如果将 VLAN 散列表寄存器配置为与 MII 时钟域双同步，则只有在写入该寄存器的位 [15:8]（小端模式）或位 [7:0]（大端模式）时，才会触发同步。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VLHT[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:16 保留，必须保持复位值

位 15:0 **VLHT[15:0]**：VLAN 散列表 (VLAN Hash Table)
该字段包含 16 位 VLAN 散列表。

VLAN 包含寄存器 (ETH_MACVIR)

VLAN inclusion register

偏移地址：0x0060

复位值：0x0000 0000

VLAN 标记包含或替换寄存器包含用于在发送数据包中进行插入或替换的 VLAN 标记。它还包含 VLAN 标记插入控件。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	VLTI	CSVL	VLP	VLC [1:0]	
											rw	rw	rw	rw	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VLT[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:21 保留，必须保持复位值

位 20 **VLTI**：VLAN 标记输入 (VLAN Tag Input)

该位置 1 时，表示应从 Tx 描述符中获取要在 Tx 数据包中插入或替换的 VLAN 标记。

位 19 **CSVL**：C-VLAN 或 S-VLAN (C-VLAN or S-VLAN)

该位置 1 时，会在所发送数据包的第 13 个和第 14 个字节中插入或替换 S-VLAN 类型 (0x88A8)。该位复位时，会在所发送数据包的第 13 个和第 14 个字节中插入或替换 C-VLAN 类型 (0x8100)。

位 18 **VLP**：VLAN 优先级控制 (VLAN Priority Control)

该位置 1 时，使用控制位 [17:16] 进行 VLAN 删除、插入或替换。该位复位时，将使用 mti_vlan_ctrl_i 控制输入，并会忽略位 [17:16]。

位 17:16 **VLC[1:0]**：发送数据包中的 VLAN 标记控制 (VLAN Tag Control in Transmit Packets)

00：不进行 VLAN 标记删除、插入或替换

01：VLAN 标记删除。MAC 将删除所有带 VLAN 标记的已发送数据包的 VLAN 类型（字节 13 和 14）以及 VLAN 标记（字节 15 和 16）。

10：VLAN 标记插入。在字节 13 和 14 中插入类型值（0x8100 或 0x88a8）后，MAC 会在数据包的字节 15 和 16 中插入 VLT。无论已发送数据包是否已具有 VLAN 标记，都会对它们全部进行该操作。

11：VLAN 标记替换。MAC 替换所有 VLAN 类型已发送数据包的字节 15 和 16 中的 VLT（字节 13 和 14 的值为 0x8100 或 0x88a8）。

注： 对此字段进行的更改仅在数据包起始时生效。如果在发送数据包时对该寄存器字段进行写操作，则只有后续数据包可使用更新值，即，当前数据包不使用更新值。

位 15:0 **VLT[15:0]**：发送数据包的 VLAN 标记 (VLAN Tag for Transmit Packets)

该字段包含要插入或替换的 VLAN 标记的值。只有在发送线无效或处于初始化阶段时，才能更改该值。

在 VLAN 标记中，位 [15:13] 为用户优先级字段，位 12 为 CFI/DEI 字段，位 [11:0] 为 VID 字段。

下面对该字段的各个位进行了说明：

位 [15:13]：用户优先级

位 12：标准格式指示符 (CFI) 或丢弃合法指示符 (DEI)

位 [11:0]：VLAN 标记的 VLAN 标识符 (VID) 字段

内部 VLAN 包含寄存器 (ETH_MACIVIR)

Inner VLAN inclusion register

偏移地址：0x0064

复位值：0x0000 0000

内部 VLAN 标记包含或替换寄存器包含要在发送数据包中插入或替换的内部 VLAN 标记。它还包含内部 VLAN 标记插入控件。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	VLTI	CSVL	VLP	VLC [1:0]	
											rw	rw	rw	rw	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VLT[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:21 保留，必须保持复位值

位 20 **VLTI**：VLAN 标记输入 (VLAN Tag Input)

该位置 1 时，表示应从 Tx 描述符中获取要在 Tx 数据包中插入或替换的 VLAN 标记。

位 19 **CSVL**：C-VLAN 或 S-VLAN (C-VLAN or S-VLAN)

该位置 1 时，会在所发送数据包的第 13 个和第 14 个字节中插入或替换 S-VLAN 类型 (0x88A8)。该位复位时，会在所发送数据包的第 13 个和第 14 个字节中插入或替换 C-VLAN 类型 (0x8100)。

位 18 **VLP**：VLAN 优先级控制 (VLAN Priority Control)

该位置 1 时，使用 VLC 字段进行 VLAN 删除、插入或替换。该位复位时，将使用 mti_vlan_ctrl_i 控制输入，并会忽略 VLC 字段。

位 17:16 **VLC[1:0]**：发送数据包中的 VLAN 标记控制 (VLAN Tag Control in Transmit Packets)

00：不进行 VLAN 标记删除、插入或替换

01：VLAN 标记删除

MAC 将删除所有带 VLAN 标记的已发送数据包的 VLAN 类型（字节 17 和 18）以及 VLAN 标记（字节 19 和 20）。

10：VLAN 标记插入

在字节 17 和 18 中插入类型值（0x8100 或 0x88a8）后，MAC 会在数据包的字节 19 和 20 中插入 VLT。无论已发送数据包是否已具有 VLAN 标记，都会对它们全部进行该操作。

11：VLAN 标记替换

MAC 替换所有 VLAN 类型已发送数据包的字节 19 和 20 中的 VLT（字节 17 和 18 的值为 0x8100 或 0x88a8）。

注： 对此字段进行的更改仅在数据包起始时生效。如果在发送数据包时对该寄存器字段进行写操作，则只有后续数据包可使用更新值，即，当前数据包不使用更新值。

位 15:0 **VLT[15:0]**：发送数据包的 VLAN 标记 (VLAN Tag for Transmit Packets)

该字段包含要插入或替换的 VLAN 标记的值。只有在发送线无效或处于初始化阶段时，才能更改该值。

在 VLAN 标记中，位 [15:13] 为用户优先级字段，位 12 为 CFI/DEI 字段，位 [11:0] 为 VID 字段。下面对该字段的各个位进行了说明：

位 [15:13]：用户优先级

位 12：标准格式指示符 (CFI) 或丢弃合法指示符 (DEI)

位 [11:0]：VLAN 标记的 VLAN 标识符 (VID) 字段

Tx 队列流控制寄存器 (ETH_MACQTxFCR)

Tx Queue flow control register

偏移地址：0x0070

复位值：0x0000 0000

流控制寄存器通过 MAC 的流控制模块来对控制（暂停命令）数据包的生成和接收进行控制。忙碌位置 1 时对寄存器进行写操作将触发流控制模块生成暂停数据包。控制数据包的字段根据 802.3x 规范中的指定进行选择，此寄存器中的暂停时间值被用在控制数据包的暂停时间字段中。忙碌位将保持置 1，直到将控制数据包发送到电缆。应用程序必须确保向该寄存器写入前已将忙碌位清零。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PT[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DZPQ	PLT[2:0]			Res.	Res.	TFE	FCB_BPA
								rw	rw	rw	rw			rw	rw

位 31:16 **PT[15:0]**：暂停时间 (Pause Time)

此字段保留 Tx 控制数据包中暂停时间字段要使用的值。如果将暂停时间位配置为与 (G)MII 时钟域双同步，则只能在目标时钟域的至少四个时钟周期后执行对此寄存器的连续写操作。

位 15:8 保留，必须保持复位值

位 7 **DZPQ**：禁止零时间片暂停 (Disable Zero-Quanta Pause)

如果该位置 1，则会在来自 FIFO 层的流控制信号（MTL 或者外部边带流控制信号 sbd_flowctrl_i 或 mti_flowctrl_i）置为有效时，禁止自动生成零时间片暂停数据包。

该位复位时，会使能自动零时间片暂停数据包生成的正常操作。

位 6:4 **PLT[2:0]**：暂停阈值下限 (Pause Low Threshold)

该字段配置暂停定时器的阈值，在该阈值处将检查输入流控制信号 mti_flowctrl_i（或 sbd_flowctrl_i），以便自动重新发送暂停数据包。

该阈值应始终小于在位 [31:16] 中配置的暂停时间。例如，如果 PT = 100H（256 个时隙），而 PLT = 001，则在第一个暂停数据包发送完成后，第二个暂停数据包会在 228 (256 - 28) 个时隙将 mti_flowctrl_i 信号置为有效时自动发送。

下面给出了针对不同值的阈值：

000：暂停时间减去 4 个时隙（PT - 4 个时隙）

001：暂停时间减去 28 个时隙（PT - 28 个时隙）

010：暂停时间减去 36 个时隙（PT - 36 个时隙）

011：暂停时间减去 144 个时隙（PT - 144 个时隙）

100：暂停时间减去 256 个时隙（PT - 256 个时隙）

101：暂停时间减去 512 个时隙（PT - 512 个时隙）

110-111：保留

时隙定义为 MII 接口每发送 512 位（64 字节）所需的时间。

此（近似）计算基于数据包大小（64、1518、2000、9018、16384 或 32768）+ 2 个暂停数据包大小 + IPG（以时隙计）。

位 3:2 保留，必须保持复位值

位 1 **TFE**：发送流控制使能 (Transmit Flow Control Enable)

全双工模式：当此位置 1 时，MAC 将使能流控制操作来发送暂停数据包。当此位复位时，将禁止 MAC 中的流控制操作，MAC 不会发送任何暂停数据包。

半双工模式：当此位置 1 时，MAC 将使能背压操作。当此位复位时，将禁止背压功能。

位 0 **FCB_BPA**：流控制忙碌或背压激活 (Flow Control Busy or Backpressure Activate)

此位在全双工模式下启动暂停数据包，在半双工模式下，TFE 位置 1 时，会激活背压功能。

全双工模式：向该寄存器写入数据前此位应读为 0。要启动暂停数据包，应用程序必须将此位置 1。在控制数据包传输过程中，此位保持置 1 以指示数据包发送正在进行中。当暂停数据包发送完成时，MAC 会将该位复位为 0。在该位清零之前，不应对该寄存器进行写操作。

半双工模式：如果该位在半双工模式下置 1（且 TFE 位置 1），则 MAC 会激活背压。在背压操作期间，当 MAC 接收到新数据包时，发送器会开始发送一个导致冲突的 JAM 模式。该控制寄存器位将与 mti_flowctrl_i 输入信号进行逻辑或运算，以用于背压功能。当 MAC 配置为全双工模式时，将自动禁止 BPA。

Rx 流控制寄存器 (ETH_MACRxFCR)

Rx flow control register

偏移地址：0x0090

复位值：0x0000 0000

接收流控制寄存器根据接收到的暂停数据包来控制 MAC 发送的暂停。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	RESERVED_PFCE	Res.	Res.	Res.	Res.	Res.	Res.	UP	RFE
							r							rw	rw

位 31:2 保留，必须保持复位值

位 1 **UP**：单播暂停数据包检测 (Unicast Pause Packet Detect)

如果暂停数据包具有在 IEEE 802.3 中指定的唯一多播地址，则会对其进行处理。该位置 1 时，MAC 还可以检测具有站的单播地址的暂停数据包。该单播地址应为 ETH_MACA0HR 和 ETH_MACA0LR 所指定。

当该位复位时，MAC 仅检测具有唯一多播地址的暂停数据包。

注： 如果多播地址与该唯一多播地址不同，则 MAC 不会处理暂停数据包。这同样适用于使能优先级流控制 (PFC) 时接收到的 PFC 数据包。唯一多播地址 (0x01_80_C2_00_00_01) 如 IEEE 802.1 Qbb-2011 之规定。

位 0 **RFE**：接收流控制使能 (Receive Flow Control Enable)

当此位置 1 且 MAC 工作在全双工模式下时，MAC 对接收到的暂停数据包进行解码，并禁止其在指定（暂停）时间内发送。当该位复位或 MAC 工作在半双工模式下时，会禁止暂停数据包的解码功能。

使能 PFC 时，将为 PFC 数据包使能流控制。MAC 会对接收到的 PFC 数据包进行解码，并在接收到的暂停时间内禁止优先级匹配的发送队列。

中断状态寄存器 (ETH_MACISR)

Interrupt status register

偏移地址：0x00B0

复位值：0x0000 0000

中断状态寄存器包含中断状态。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED_GPIIS	RXSTSIS	TXSTSIS	TSIS	RESERVED_MMCRXIPIS	MMCTXIS	MMCRXIS	MMCIS	Res.	Res.	LPIIS	PMTIS	PHYIS	RESERVED_PCSANCIS	RESERVED_PCSLCHGIS	Res.
r	r	r	r	r	r	r	r			r	r	r	r	r	

位 31:16 保留，必须保持复位值

位 15 **RESERVED_GPIIS**：保留。

位 14 **RXSTSIS**：接收状态中断 (Receive Status Interrupt)

此位指示所接收数据包的状态。将 ETH_MACISR 寄存器中的 RWT 位置 1 时，该位会置 1。在 ETH_MACISR 寄存器中读取相应的中断源位时，该位会清零。

位 13 **TXSTSIS**：发送状态中断 (Transmit Status Interrupt)

此位指示所发送数据包的状态。将 ETH_MACISR 寄存器中的以下任一位置 1 时，该位会置 1：

- 过度冲突 (EXCOL)
- 延迟冲突 (LCOL)
- 过度延迟 (EXDEF)
- 载波丢失 (LCARR)
- 无载波 (NCARR)
- Jabber 超时 (TJT)

在 ETH_MACISR 寄存器中读取相应的中断源位时，该位会清零。

位 12 **TSIS**：时间戳中断状态 (Timestamp Interrupt Status)

如果使能了时间戳功能，则当以下任一条件为真时，该位会置 1：

- 系统时间值等于或超出目标时间高位和低位寄存器中指定的值。
- 秒寄存器发生上溢。
- 发生目标时间错误，即，编程的目标时间已结束。

如果使能辅助快照功能，则在将辅助快照触发信号置为有效时，该位会置 1。

如果在 MTL 中使能丢弃发送状态，则在 ETH_MACTxTSSNR 和 ETH_MACTxTSSSR 寄存器中更新捕获的发送时间戳时，该位会置 1。

如果使能 PTP 减荷功能，则在 ETH_MACTxTSSNR 和 ETH_MACTxTSSSR 寄存器中更新捕获的发送时间戳时，该位会置 1，以用于 PTO 生成的延迟请求和 Pdelay 请求数据包。

在 ETH_MACTSSR 寄存器中读取相应的中断源位时，该位会清零。

位 11 **RESERVED_MMCRXIPIS**：保留。

位 10 **MMCTXIS**：MMC 发送中断状态 (MMC Transmit Interrupt Status)

当 MMC 发送中断寄存器中生成中断时，此位将置为高电平。当此中断寄存器中的所有位都清零时，此位将清零。

只有在选择 Enable MAC Management Counters (MMC)（使能 MAC 管理计数器 (MMC)）选项时，该位才有效。

位 9 **MMCRXIS**：MMC 接收中断状态 (MMC Receive Interrupt Status)

当 MMC 接收中断寄存器中生成中断时，此位将置为高电平。当此中断寄存器中的所有位都清零时，此位将清零。

只有在选择 Enable MAC Management Counters (MMC)（使能 MAC 管理计数器 (MMC)）选项时，该位才有效。

位 8 **MMCIS**：MMC 中断状态 (MMC Interrupt Status)

位 11、位 10 或位 9 置为高电平时，该位将置为高电平。仅当所有这些位均为低电平时，此位才会清零。只有在选择 Enable MAC Management Counters (MMC)（使能 MAC 管理计数器 (MMC)）选项时，该位才有效。

位 7:6 保留，必须保持复位值

位 5 **LPIIS**：LPI 中断状态 (LPI Interrupt Status)

使能节能型以太网功能时，会针对 MAC 发送器或接收器是进入还是退出 LPI 状态来设置该位。对 ETH_MACLCSR 寄存器的 TLPIEN 位进行读操作时，该位清零。在所有其他模式下，该位将被保留。

位 4 **PMTIS**：PMT 中断状态 (PMT Interrupt Status)

在掉电模式下接收到魔术数据包或局域网唤醒数据包（ETH_MACPCSR 寄存器中的 RWKPRCVD 和 MGKPRCVD 位）时，该位置 1。当位 [6:5] 因对 ETH_MACPCSR 寄存器执行读操作而被清零时，此位也会清零。

只有在选择 Enable Power Management（使能电源管理）选项时，该位才有效。

位 3 **PHYIS**：PHY 中断 (PHY Interrupt)

在 phy_intr_i 输入端检测到上升沿时，该位置 1。对此寄存器执行读操作时此位清零。

位 2:0 保留，必须保持复位值

中断使能寄存器 (ETH_MACIER)

Interrupt enable register

偏移地址：0x00B4

复位值：0x0000 0000

中断使能寄存器包含用于生成中断的屏蔽位。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	RXSTSIE	TXSTSIE	TSIE	Res.	Res.	Res.	Res.	Res.	Res.	LPIIE	PMTIE	PHYIE	RESERVED_PCSANCIE	RESERVED_PCSLCHGIE	Res.
	rw	rw	rw							rw	rw	rw	r	r	

位 31:15 保留，必须保持复位值

位 14 **RXSTSIE**：接收状态中断使能 (Receive Status Interrupt Enable)
该位置 1 时，会因为 ETH_MACISR 寄存器中的 RXSTSIS 位置 1 而将中断信号置为有效。

位 13 **TXSTSIE**：发送状态中断使能 (Transmit Status Interrupt Enable)
该位置 1 时，会因为 ETH_MACISR 寄存器中的 TXSTSIS 位置 1 而将中断信号置为有效。

位 12 **TSIE**：时间戳中断使能 (Timestamp Interrupt Enable)
该位置 1 时，会因为 ETH_MACISR 寄存器中的 TSIS 位置 1 而将中断信号置为有效。
仅当选择了 Enable IEEE 1588 Timestamp Support（使能 IEEE 1588 时间戳支持）选项时，该位才有效。否则，该位将被保留。

位 11:6 保留，必须保持复位值

位 5 **LPIIE**：LPI 中断使能 (LPI Interrupt Enable)
该位置 1 时，会因为 ETH_MACISR 寄存器中的 LPIIS 位置 1 而将中断信号置为有效。
仅当选择了 Enable Energy Efficient Ethernet (EEE)（使能节能型以太网 (EEE)）选项时，该位才有效。否则，该位将被保留。

位 4 **PMTIE**：PMT 中断使能 (PMT Interrupt Enable)
该位置 1 时，会因为 ETH_MACISR 寄存器中的 PMTIS 位置 1 而将中断信号置为有效。

位 3 **PHYIE**：PHY 中断使能 (PHY Interrupt Enable)
该位置 1 时，会因为 ETH_MACISR 寄存器中的 PHYIS 位置 1 而将中断信号置为有效。

位 2:0 保留，必须保持复位值

Rx Tx 状态寄存器 (ETH_MACRxTxSR)

Rx Tx status register

偏移地址：0x00B8

复位值：0x0000 0000

接收发送状态寄存器包含接收和发送错误状态。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	RWT	Res.	Res.	EXCOL	LCOL	EXDEF	LCARR	NCARR	TJT
							r			r	r	r	r	r	r

位 31:9 保留，必须保持复位值

位 8 **RWT**：接收看门狗超时 (Receive Watchdog Timeout)

当接收到长度大于 2,048 字节（如果使能巨型数据包模式，则为 10, 240 字节）的数据包，并且 ETH_MACCR 寄存器中的 WD 位复位时，该位置 1。当接收到长度大于 16,383 字节的数据包，并且 ETH_MACCR 寄存器中的 WD 位置 1 时，该位置 1。

位 7:6 保留，必须保持复位值

位 5 **EXCOL**：过度冲突 (Excessive Collisions)

当 MTL_Operation_Mode 寄存器中的 DTXSTS 位置 1 时，该位指示尝试发送当前数据包时因出现 16 个连续冲突而中止发送。如果 ETH_MACCR 寄存器中的 DR 位置 1，则该位在第一次冲突后置 1，并且数据包发送过程将中止。

位 4 **LCOL**：延迟冲突 (Late Collision)

当 MTL_Operation_Mode 寄存器中的 DTXSTS 位置 1 时，该位指示因冲突窗口（MII 模式下为 64 个字节，含报头）后出现冲突而中止数据包发送。

如果发生下溢错误，则该位无效。

位 3 **EXDEF**：过度延迟 (Excessive Deferral)

当 MTL_Operation_Mode 寄存器中的 DTXSTS 位置 1，且 ETH_MACCR 寄存器中的 DC 位置 1 时，该位指示因延迟超过 24,288 个位时间（在 1000 Mbps 模式下或在使能巨型数据包时，为 155,680）而结束发送。

位 2 **LCARR**：载波丢失 (Loss of Carrier)

当 MTL_Operation_Mode 寄存器中的 DTXSTS 位置 1 时，该位指示数据包发送期间丢失载波，即，数据包发送期间有一个或多个发送时钟周期的 phy_crs_i 信号无效。该位仅对未发生冲突的数据包有效。

位 1 **NCARR**：无载波 (No Carrier)

当 MTL_Operation_Mode 寄存器中的 DTXSTS 位置 1 时，该位指示在报头发送结束时不存在来自 PHY 的载波信号。

位 0 **TJT**：发送 Jabber 超时 (Transmit Jabber Timeout)

该位指示发送 Jabber 定时器已过期，在数据包大小超过 2,048 字节（使能巨型数据包时为 10,240 字节），并且 ETH_MACCR 寄存器中的 JD 位复位时，会出现这种情况。当数据包大小超过 16,383 字节并且 ETH_MACCR 寄存器中的 JD 位置 1 时，该位置 1。

PMT 控制状态寄存器 (ETH_MACPCSR)

PMT control status register

偏移地址：0x00C0

复位值：0x0000 0000

只有在 coreConsultant 中选择了 PMT 模块时，才会存在 PMT 控制和状态寄存器。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RWKFILTRST	Res.	Res.	RWKPTR[4:0]					Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
rw			r	r	r	r	r								

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	RWKPFE	GLBLUCAST	Res.	Res.	RWKPRCVD	MGKPRCVD	Res.	Res.	RWKPKTEN	MGKPKTEN	PWRDWN
					rw	rw			r	r			rw	rw	rw

位 31 **RWKFILTRST**：远程唤醒数据包过滤寄存器指针复位 (Remote wakeup Packet Filter Register Pointer Reset)

该位置 1 时，远程唤醒数据包过滤寄存器指针将复位为 3'b000。它会在 1 个时钟周期后自动清零。

位 30:29 保留，必须保持复位值

位 28:24 **RWKPTR[4:0]**：远程唤醒 FIFO 指针 (Remote wakeup FIFO Pointer)

该字段给出了远程唤醒数据包过滤寄存器指针的当前值（0 到 7）。如果该指针的值等于 7，则对该寄存器进行写操作时，远程唤醒数据包过滤寄存器的内容将被传输到 eth_mii_rx_clk 域。

位 23:11 保留，必须保持复位值

位 10 **RWKPFE**：远程唤醒数据包转发使能 (Remote wakeup Packet Forwarding Enable)

该位与 RWKPKTEN 均置 1 时，MAC 接收器将丢弃所有接收到的帧，直到接收到预期唤醒帧。该事件之后的所有帧（包括接收到的唤醒帧）均被转发给应用程序。接收到唤醒数据包后，该位自动清零。

在接收到预期唤醒帧之前，应用程序也会将该位清零。在这种情况下，MAC 将恢复为默认行为，即，将接收到的数据包转发给应用程序。该位必须仅在 RWKPKTEN 置为高电平且 PWRDWN 置为低电平时置 1。当 PWRDWN 置为高电平时，该位的设置不起任何作用。

注： 如果魔术数据包使能和唤醒帧使能以及该位均置 1，并且在接收到唤醒帧之前接收到魔术数据包，则该位会在接收到魔术数据包时自动清零，接收到的魔术数据包被丢弃，接收到魔术数据包之后的所有帧均被转发给应用程序。

位 9 **GLBLUCAST**：全局单播 (Global Unicast)

当此位置 1 时，MAC (DAF) 地址识别所过滤的所有单播数据包均会被检测为远程唤醒数据包。

位 8:7 保留，必须保持复位值

位 6 **RWKPRCVD**：接收到的远程唤醒数据包 (Remote wakeup Packet Received)

该位置 1 时，表示因接收到远程唤醒数据包而生成了电源管理事件。对此寄存器执行读操作时此位清零。

位 5 **MGKPRCVD**：接收到魔术数据包 (Magic Packet Received)

该位置 1 时，表示因接收到魔术数据包而生成了电源管理事件。对此寄存器执行读操作时此位清零。

位 4:3 保留，必须保持复位值

位 2 **RWKPKTEN**：远程唤醒数据包使能 (Remote wakeup Packet Enable)

如果该位置 1，则会在 MAC 接收到远程唤醒数据包时生成电源管理事件。

位 1 **MGKPKTEN**：魔术数据包使能 (Magic Packet Enable)

如果该位置 1，则会在 MAC 接收到魔术数据包时生成电源管理事件。

位 0 **PWRDWN**：掉电 (Power Down)

该位置 1 时，MAC 接收器将丢弃所有接收到的数据包，直到接收到预期魔术数据包或远程唤醒数据包。该位随后自动清零，并且将禁止掉电模式。在接收到预期魔术数据包或远程唤醒数据包之前，软件可将该位清零。此位清零后 MAC 接收到的数据包会被转发给应用程序。该位必须仅在魔术数据包使能、全局单播或远程唤醒数据包使能位置为高电平时置 1。

注： 在掉电模式下，可以门控关闭 CSR 时钟。但是，CSR 时钟门控关闭时，不能对该寄存器执行任何读操作或写操作。因此，软件不能将此位清零。

删除唤醒数据包过滤寄存器 (ETH_MACRWKPFR)

Remove wakeup packet filter register

偏移地址：0x00C4

复位值：0x0000 0000

地址为 0C4H 的 wkuppktfilter_reg 寄存器加载唤醒数据包过滤寄存器的内容。

要在唤醒数据包过滤寄存器中加载值，必须写入整个寄存器 (wkuppktfilter_reg)。通过分别针对 wkuppktfilter_reg0、wkuppktfilter_reg1 和 wkuppktfilter_reg31 在地址 (0C4H) 中连续加载八个、十六个或三十二个寄存器值来加载 wkuppktfilter_reg 寄存器。读取 wkuppktfilter_reg 寄存器时也采用类似方式。以太网外设在 ETH_MACPCSR 寄存器的位 [26:24] 中更新 wkuppktfilter_reg 寄存器当前指针值。

表 541. 远程唤醒数据包过滤寄存器

ETH_MACRWKPFR + #	字段							
0	过滤器 0 字节屏蔽							
1	过滤器 1 字节屏蔽							
2	过滤器 2 字节屏蔽							
3	过滤器 3 字节屏蔽							
4	保留	过滤器 3 命令	保留	过滤器 2 命令	保留	过滤器 1 命令	保留	过滤器 0 命令
5	过滤器 3 偏移		过滤器 2 偏移		过滤器 1 偏移		过滤器 0 偏移	
6	过滤器 1 CRC-16				过滤器 0 CRC-16			
7	过滤器 3 CRC-16				过滤器 2 CRC-16			
8	过滤器 4 字节屏蔽							
9	过滤器 5 字节屏蔽							
10	过滤器 6 字节屏蔽							

表 541. 远程唤醒数据包过滤寄存器（续）

ETH_MACRWKPFR + #	字段							
11	过滤器 7 字节屏蔽							
12	保留	过滤器 7 命令	保留	过滤器 6 命令	保留	过滤器 5 命令	保留	过滤器 4 命令
13	过滤器 7 偏移		过滤器 6 偏移		过滤器 5 偏移		过滤器 4 偏移	
14	过滤器 5 CRC-16				过滤器 4 CRC-16			
15	过滤器 7 CRC-16				过滤器 6 CRC-16			
16	过滤器 8 字节屏蔽							
17	过滤器 9 字节屏蔽							
18	过滤器 10 字节屏蔽							
19	过滤器 11 字节屏蔽							
20	保留	过滤器 11 命令	保留	过滤器 10 命令	保留	过滤器 9 命令	保留	过滤器 8 命令
21	过滤器 11 偏移		过滤器 10 偏移		过滤器 9 偏移		过滤器 8 偏移	
22	过滤器 9 CRC-16				过滤器 8 CRC-16			
23	过滤器 11 CRC-16				过滤器 10 CRC-16			
24	过滤器 12 字节屏蔽							
25	过滤器 13 字节屏蔽							
26	过滤器 14 字节屏蔽							
27	过滤器 15 字节屏蔽							
28	保留	过滤器 15 命令	保留	过滤器 14 命令	保留	过滤器 13 命令	保留	过滤器 12 命令
29	过滤器 15 偏移		过滤器 14 偏移		过滤器 13 偏移		过滤器 12 偏移	
30	过滤器 13 CRC-16				过滤器 12 CRC-16			
31	过滤器 15 CRC-16				过滤器 14 CRC-16			

表 542. ETH_MACRWKPFR

寄存器	说明
过滤器 i 字节屏蔽	过滤器 *i* 字节屏蔽寄存器定义过滤器 *i*（0、1、2、3、...、15）检测哪些数据包字节来确定数据包是否为唤醒数据包。MSB（第 31 位）必须为零。 位 j [30:0] 为字节屏蔽。如果字节屏蔽的位 j（字节号）置 1，则 CRC 模块会处理传入数据包的过滤器 *i* 偏移 + j；否则会忽略过滤器 *i* 偏移 + j。
过滤器 *i* 命令	4 位过滤器 *i* 命令控制过滤器 *i* 操作。 – 位 3 指定地址类型，用于定义模式的目标地址类型。 该位置 1 时，模式仅应用于多播数据包；该位复位时，模式仅应用于单播数据包。 – 位 2（反向模式）置 1 时，会反转 CRC16 散列函数信号的逻辑，以拒绝具有匹配 CRC_16 值的数据包。 位 2 与位 1 搭配使用，允许 MAC 通过创建过滤器逻辑（例如“模式 1 与模式 2 进行与非运算”）来拒绝远程唤醒数据包的子集。 – 位 1 (And_Previous) 实现布尔逻辑。 该位置 1 时，当前条目的结果将与前一过滤器的结果进行逻辑与运算。使用此与逻辑运算，可以将屏蔽拆分到两个、三个或四个过滤器中，从而支持超过 32 个字节的过滤器模式。这具体取决于将 And_Previous 位置 1 的过滤器数目。 And_Previous 位适用于多过滤器操作，其中的过滤结果与前一过滤器的结果进行与运算。例如，如果将过滤器 1 的 And_Previous 位置 1，则只有在过滤器 0 和过滤器 1 都满足 *表 543* 中提到的远程唤醒数据包检测和中断生成标准时，才会检测到远程唤醒数据包以及生成 PMT 中断。 *注：　And_Previous 位设置适用于一组 4 个过滤器。* *注：　对于未使能的过滤器，将 And_Previous 位置 1 无效。换言之，在一组 4 个过滤器中，编号最小的过滤器的 And_Previous 位置 1 无效。例如，过滤器 0 的 And_Previous 位置 1 无效。* *注：　如果将过滤器的 And_Previous 位置 1 以形成与链形式的过滤器，则与链会在未使能的过滤器处断开。例如：* *如果过滤器 2 的 And_Previous 位置 1（过滤器 2 命令的位 1 置 1），但过滤器 1 未使能（过滤器 1 命令的位 0 复位），则只会考虑过滤器 2 的结果。* *如果过滤器 2 的 And_Previous 位置 1（过滤器 2 命令的位 1 置 1），过滤器 3 的 And_Previous 位置 1（过滤器 3 命令的位 1 置 1），但过滤器 1 未使能（过滤器 1 命令的位 0 复位），则只会考虑过滤器 2 结果与过滤器 3 结果的与运算结果。* *如果过滤器 2 的 And_Previous 位置 1（过滤器 2 命令的位 1 置 1），过滤器 3 的 And_Previous 位置 1（过滤器 3 命令的位 1 置 1），但过滤器 2 未使能（过滤器 2 命令的位 0 复位），则由于过滤器 2 的 And_Previous 位置 1 无效，因此只会考虑过滤器 1 结果与过滤器 3 结果的或运算结果。* *注：　如果通过将 And_Previous 位置 1 形成的过滤器链中存在互补性设定，则帧绝不会通过与链过滤器。例如，如果过滤器 2 的 And_Previous 位置 1（过滤器 2 命令的位 1 置 1），并且过滤器 1 的 Address_Type 位置 1（过滤器 1 命令的位 3 置 1，指示多播检测）、过滤器 2 的 Address_Type 位复位（过滤器 2 命令的位 3 复位，指示单播检测），或反之，则远程唤醒帧不会通过与链过滤器，因为远程唤醒帧不能同时属于单播和多播地址类型。* – 位 0 用于使能过滤器 *i*。如果位 0 未置 1，则会禁止过滤器 *i*。

表 542. ETH_MACRWKPFR（续）

寄存器	说明
过滤器 *i* 偏移	此过滤器 *i* 偏移寄存器定义过滤器 *i* 要检测的数据包的偏移（在数据包内）。 该 8 位模式偏移是要检测的过滤器 *i* 的第一个字节的偏移。允许的最小偏移为 12，指代数据包的第 13 个字节。偏移值 0 指代数据包的第一个字节。
过滤器 *i* CRC-16	该过滤器 *i* CRC-16 寄存器包含根据模式计算的 CRC_16 值，以及对唤醒过滤寄存器模块编程的字节屏蔽。 使用以下多项式进行 16 位 CRC 计算： G(x) = x^16 + x^15 + x^2 + 1 散列函数计算中使用的每个屏蔽均与该屏蔽相关的 16 位值进行比较。每个过滤器均包含以下内容： – 32 位屏蔽：该屏蔽中的每个位均对应于所检测数据包中的一个字节。如果该位为 1’，则会将相应的字节用于 CRC16 计算。 – 8 位偏移指针：指定开始 CRC16 计算的字节。 指针和屏蔽一起用来定位用于 CRC16 计算的字节。

下表列出了会生成 PMT 中断的远程唤醒情况。

表 543. 远程唤醒数据包和 PMT 中断生成(1)

过滤器 i 命令			帧类型和 CRC 状态		中断生成
CAST	INV	EN	接收到的帧播型	CRC 状态	RWK 中断
0	0	1	单播	匹配	检测到远程唤醒数据包并生成 PMT 中断
0	1	1	单播	不匹配	检测到远程唤醒数据包并生成 PMT 中断
1	0	1	多播	匹配	检测到远程唤醒数据包并生成 PMT 中断
1	1	1	多播	不匹配	检测到远程唤醒数据包并生成 PMT 中断

1. 在所有其他组合中，均未检测到远程唤醒数据包，并且不会生成 PMT 中断。

LPI 控制状态寄存器 (ETH_MACLCSR)

LPI control status register

偏移地址：0x00D0

复位值：0x0000 0000

LPI 控制和状态寄存器控制 LPI 功能并提供 LPI 中断状态。对此寄存器执行读操作时状态位清零。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LPITCSE	LPITE	LPITXA	PLSEN	PLS	LPIEN
										rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	RLPIST	TLPIST	Res.	Res.	Res.	Res.	RLPIEX	RLPIEN	TLPIEX	TLPIEN
						r	r					r	r	r	r

位 31:22 保留，必须保持复位值

位 21 **LPITCSE**：LPI Tx 时钟停止使能 (LPI Tx Clock Stop Enable)

该位置 1 时，MAC 在进入 Tx LPI 模式后会将 sbd_tx_clk_gating_ctrl_o 信号置为高电平，以指示可以停止向 MAC 提供 Tx 时钟。

该位复位时，MAC 在进入 Tx LPI 模式后不会将 sbd_tx_clk_gating_ctrl_o 信号置为高电平。

位 20 **LPITE**：LPI 定时器使能 (LPI Timer Enable)

该位控制 MAC 发送器自动进入和退出 LPI 状态。如果 LPITE、LPITXA 和 LPITXEN 位均置 1，则只有在整个 MAC TX 数据路径保持空闲的时长达到 ETH_MACLETR 寄存器所指示的时间后，MAC 发送器才会进入 LPI 状态。

进入 LPI 状态后，如果数据路径变为非空闲（由于接受了新的需要发送的数据包），则发送器会退出 LPI 状态，但不会将 LPITXEN 位清零。这样一来，再次变为空闲时便可以重新返回 LPI 状态。

LPITE 为 0 时，LPI 自动定时器被禁止，MAC 发送器进入 LPI 状态（根据 LPITXA 和 LPITXEN 位设置的说明）。

位 19 **LPITXA**：LPI Tx 自动化 (LPI Tx Automate)

该位控制 MAC 在发送端进入或退出 LPI 模式时的行为。

如果 LPITXA 和 LPIEN 位均置 1，则只有在已发送所有未完成的数据包（在内核中）和挂起数据包（在应用程序接口中）之后，MAC 才会进入 LPI 模式。应用程序发送任何数据包以供传送，或应用程序发出 Tx FIFO 刷新命令时，MAC 将退出 LPI 模式。此外，MAC 退出 LPI 状态时会自动将 LPIEN 位清零。当 MAC 处于 LPI 模式时，如果在 ETH_MTLTxQOMR 的 FTQ 位中将 Tx FIFO 刷新置为有效，则 MAC 会退出 LPI 模式。

该位为 0 时，LPIEN 位直接控制 MAC 进入或退出 LPI 模式时的行为。

位 18 **PLSEN**：PHY 链路状态使能 (PHY Link Status Enable)

该位使能在接收路径上接收到的用于激活 LPI LS 定时器的链路状态。

该位置 1 时，MAC 使用 PLS 位来触发 LPI LS 定时器。该位复位时，MAC 将忽略 ETH_MACPHYCSR 寄存器的链路状态位，并且仅使用 PLS 位。

该位为只读位并且被保留（如果尚未选择 PHY 接口）。

位 17 **PLS**：PHY 链路状态 (PHY Link Status)

该位指示 PHY 的链路状态。只有在链路接通状态 (OKAY) 至少保持 LPI LS 定时器所指示的时间时，MAC 发送器才会将 LPI 模式置为有效。

该位置 1 时，将链路视为连通状态 (UP)；该位复位时，将链路视为断开状态。

位 16 **LPIEN**：LPI 使能 (LPI Enable)

该位置 1 时，会命令 MAC 发送器进入 LPI 状态。该位复位时，会命令 MAC 退出 LPI 状态并恢复正常发送。

当 LPITXA 位置 1 且由于新数据包到达等待发送而使 MAC 退出 LPI 状态时，该位清零。

位 15:10 保留，必须保持复位值

位 9 **RLPIST**：接收 LPI 状态 (Receive LPI State)

该位置 1 时，表示 MAC 正在通过 MII 接口接收 LPI 模式。

位 8 **TLPIST**：发送 LPI 状态 (Transmit LPI State)

该位置 1 时，表示 MAC 正在通过 MII 接口发送 LPI 模式。

位 7:4 保留，必须保持复位值

位 3 **RLPIEX**：接收 LPI 退出 (Receive LPI Exit)

该位置 1 时，表示 MAC 接收器已停止通过 MII 接口接收 LPI 模式，并且退出 LPI 状态和恢复正常接收。通过对此寄存器执行读操作可将此位清零。

注： 如果 MAC 在非常短的时间内（例如不到三个 CSR 时钟周期）停止接收 LPI 模式，则该位可能不会置 1。

位 2 **RLPIEN**：接收 LPI 进入 (Receive LPI Entry)

该位置 1 时，表示 MAC 接收器已接收到 LPI 模式并进入 LPI 状态。通过对此寄存器执行读操作可将此位清零。

注： 如果 MAC 在非常短的时间内（例如不到三个 CSR 时钟周期）停止接收 LPI 模式，则该位可能不会置 1。

位 1 **TLPIEX**：发送 LPI 退出 (Transmit LPI Exit)

该位置 1 时，表示应用程序将 LPIEN 位清零，且 LPI TW 定时器到期后，MAC 发送器退出了 LPI 状态。通过对此寄存器执行读操作可将此位清零。

位 0 **TLPIEN**：发送 LPI 进入 (Transmit LPI Entry)

该位置 1 时，表示由于 LPIEN 位置 1 而使 MAC 发送器进入了 LPI 状态。通过对此寄存器执行读操作可将此位清零。

LPI 定时器控制寄存器 (ETH_MACLTCR)

LPI timers control register

偏移地址：0x00D4

复位值：0x03E8 0000

LPI 定时器控制寄存器控制 LPI 状态下的超时值。它指定 MAC 发送 LPI 模式的时间以及 MAC 在恢复正常发送之前等待的时间。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	LST[9:0]									
						rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TWT[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:26 保留，必须保持复位值

位 25:16 **LST[9:0]**：LPI LS 定时器 (LPI LS Timer)

该字段指定在可以将 LPI 模式发送到 PHY 之前，来自 PHY 的链路状态 (OKAY) 应持续的最短时间（以毫秒为单位）。即使 LPIEN 位置 1，MAC 也不会发送 LPI 模式，除非 LPI LS 定时器达到编程的端子计数。LPI LS 定时器的默认值是 1000（1 秒），如 IEEE 标准中所定义。

位 15:0 **TWT[15:0]**：LPI TW 定时器 (LPI TW Timer)

该字段指定 MAC 停止向 PHY 发送 LPI 模式之后至少需等待多长时间才能恢复正常发送（以微秒为单位）。TLPIEX 状态位在该定时器到期后置 1。

LPI 进入定时器寄存器 (ETH_MACLETR)

LPI entry timer register

偏移地址：0x00D8

复位值：0x0000 0000

LPI 进入定时器寄存器用于存储 LPI 空闲定时器值（以微秒为单位）。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LPIET[16:13]			
												rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LPIET[12:0]													Res.	Res.	Res.
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw			

位 31:20 保留，必须保持复位值

位 19:3 **LPIET[16:0]**：LPI 进入定时器 (LPI Entry Timer)

此字段指定 MAC 发送完所有帧后需等待多长时间才能进入 LPI 模式（以微秒为单位）。只有在 LPITE 和 LPITXA 均置 1 时，该字段才有效且可用。

位 [2:0] 为只读位，因此该定时器的粒度以 8 微秒为步长。

位 2:0 保留，必须保持复位值

1 微秒节拍计数器寄存器 (ETH_MAC1USTCR)

1-microsecond-tick counter register

偏移地址：0x00DC

复位值：0x0000 0000

该寄存器控制所有 LPI 定时器的参考时间（1 微秒节拍）的生成。该定时器最初必须由软件编程。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	TIC_1US_CNTR[11:0]											
				rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:12 保留，必须保持复位值

位 11:0 **TIC_1US_CNTR[11:0]**：1 µs 节拍计数器 (1 µs tick Counter)
应用程序必须对该计数器进行编程，以使 CSR 时钟的时钟周期数为 1 µs。
（从编程前的值中减去 1）。
例如，如果 CSR 时钟为 100 MHz，则需要将该字段编程为 100 - 1 = 99（即 0x63）。
要生成用于更新某些 EEE 相关计数器的 1 µs 事件，此项必不可少。

版本寄存器 (ETH_MACVR)

Version register

偏移地址：0x0110

复位值：0x0000 3041

版本寄存器标识以太网外设的版本。该寄存器包含两个字节：一个字节供 Synopsys 用于标识内核版本号，另一个字节供应用程序在配置内核时进行设置。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
USERVER[7:0]								SNPSVER[7:0]							
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:16 保留，必须保持复位值

位 15:8 **USERVER[7:0]**：用户定义的版本 (User-defined Version)（使用 coreConsultant 配置）

位 7:0 **SNPSVER[7:0]**：Synopsys 定义的版本 (Synopsys-defined Version) (3.7)

调试寄存器 (ETH_MACDR)

Debug register

偏移地址：0x0114

复位值：0x0000 0000

调试寄存器提供各个 MAC 模块的调试状态。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TFCSTS[1:0]		TPESTS
													r		r

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RFCFCSTS[1:0]		RPESTS
													r		r

位 31:19 保留，必须保持复位值

位 18:17 **TFCSTS[1:0]**：MAC 发送数据包控制器状态 (MAC Transmit Packet Controller Status)

此字段指示 MAC 发送数据包控制器模块的状态：

00：空闲状态

01：等待下列其中一项：

– 前一个数据包的状态

– IPG 或回退阶段结束

10：生成并发送暂停控制数据包（在全双工模式下）

11：传输要发送的输入数据包

位 16 **TPESTS**：MAC MII 发送协议引擎状态 (MAC MII Transmit Protocol Engine Status)

该位置 1 时，表示 MAC MII 发送协议引擎正在主动发送数据，而未处于空闲状态。

位 15:3 保留，必须保持复位值

位 2:1 **RFCFCSTS[1:0]**：MAC 接收数据包控制器 FIFO 状态 (MAC Receive Packet Controller FIFO Status)

该字段置 1 时，表示 MAC 接收数据包控制器模块的各个小 FIFO 读和写控制器的有效状态。

位 0 **RPESTS**：MAC MII 接收协议引擎状态 (MII Receive Protocol Engine Status)

该位置 1 时，表示 MAC MII 接收协议引擎正在主动接收数据，而未处于空闲状态。

硬件功能 1 寄存器 (ETH_MACHWF1R)

HW feature 1 register

偏移地址：0x0120

复位值：0x1184 1904

该寄存器指示存在第二组可选的以太网外设功能。软件驱动程序可使用该寄存器来动态使能或禁止与可选模块相关的程序。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	L3L4FNUM[3:0]				Res.	HASHTBLSZ[1:0]		Res.	Res.	Res.	AVSEL	DBGMEMA	TSOEN	SPHEN	DCBEN
	r	r	r	r		r	r				r	r	r	r	r

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDR64[1:0]		ADVTHWORD	PTOEN	OSTEN	TXFIFOSIZE[4:0]					Res.	RXFIFOSIZE[4:0]				
r	r	r	r	r	r	r	r	r	r		r	r	r	r	r

位 31 保留，必须保持复位值

位 30:27 **L3L4FNUM[3:0]**：L3 或 L4 过滤器总数 (Total number of L3 or L4 Filters)

该字段指示 L3 或 L4 过滤器的总数：
0000：无 L3 或 L4 过滤器
0001：1 个 L3 或 L4 过滤器
0010：2 个 L3 或 L4 过滤器
..
1000：8 个 L3 或 L4 过滤器

位 26 保留，必须保持复位值

位 25:24 **HASHTBLSZ[1:0]**：散列表大小 (Hash Table Size)

该字段指示散列表的大小：
00：无散列表
01: 64
10: 128
11: 256

位 23:21 保留，必须保持复位值

位 20 **AVSEL**：AV 功能使能 (AV Feature Enable)

选择 Enable Audio Video Bridging（使能音视频桥接）选项时，该位置 1。

位 19 **DBGMEMA**：DMA 调试寄存器使能 (DMA Debug Registers Enable)

选择 Debug Mode Enable（调试模式使能）选项时，该位置 1。

位 18 **TSOEN**：TCP 分段减荷使能 (TCP Segmentation Offload Enable)

选择 Enable TCP Segmentation Offloading for TCP/IP Packets（使能 TCP/IP 数据包的 TCP 分段减荷）选项时，该位置 1。

位 17 **SPHEN**：拆分报头功能使能 (Split Header Feature Enable)
选择 Enable Split Header Structure（使能拆分报头结构）选项时，该位置 1。

位 16 **DCBEN**：DCB 功能使能 (DCB Feature Enable)
选择 Enable Data Center Bridging（使能数据中心桥接）选项时，该位置 1。

位 15:14 保留，必须保持复位值

位 13 **ADVTHWORD**：IEEE 1588 高位字寄存器使能 (IEEE 1588 High Word Register Enable)
选择 Add IEEE 1588 Higher Word Register（添加 IEEE 1588 高位字寄存器）选项时，该位置 1。

位 12 **PTOEN**：PTP 减荷使能 (PTP Offload Enable)
选择 Enable PTP Timestamp Offload Feature（使能 PTP 时间戳减荷功能）时，该位置 1。

位 11 **OSTEN**：一步时间戳使能 (One-Step Timestamping Enable)
选择 Enable One-Step Timestamp Feature（使能一步时间戳功能）时，该位置 1。

位 10:6 **TXFIFOSIZE[4:0]**：MTL 发送 FIFO 大小 (MTL Transmit FIFO Size)
该字段包含 MTL Tx FIFO 的配置值（以字节表示），表示为以 2 为底的对数减去 7，即 Log2(TXFIFO_SIZE) -7：
00000：128 字节
00001：256 字节
00010：512 字节
00011：1,024 字节
00100：2,048 字节
00101：4,096 字节
00110：8,192 字节
00111：16,384 字节
01000：32 KB
01001：64 KB
01010：128 KB
01011 11111：保留

位 5 保留，必须保持复位值

位 4:0 **RXFIFOSIZE[4:0]**：MTL 接收 FIFO 大小 (MTL Receive FIFO Size)
该字段包含 MTL Rx FIFO 的配置值（以字节表示），表示为以 2 为底的对数减去 7，即 Log2(RXFIFO_SIZE) -7：
00000：128 字节
00001：256 字节
00010：512 字节
00011：1,024 字节
00100：2,048 字节
00101：4,096 字节
00110：8,192 字节
00111：16,384 字节
01000：32,767 字节
01000：32 KB
01001：64 KB
01010：128 KB
01011：256 KB
01100 11111：保留

硬件功能 2 寄存器 (ETH_MACHWF2R)

HW feature 2 register

偏移地址：0x0124

复位值：0x4100 0000

该寄存器指示存在第三组可选的以太网外设功能。软件驱动程序可使用该寄存器来动态使能或禁止与可选模块相关的程序。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	AUXSNAPNUM[2:0]			Res.	PPSOUTNUM[2:0]			Res.	Res.	TXCHCNT[3:0]				Res.	Res.
	r	r	r		r	r	r			r	r	r	r		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RXCHCNT[3:0]				Res.	Res.	TXQCNT[3:0]				Res.	Res.	RXQCNT[3:0]			
r	r	r	r			r	r	r	r			r	r	r	r

位 31 保留，必须保持复位值

位 30:28 **AUXSNAPNUM[2:0]**：辅助快照输入数 (Number of Auxiliary Snapshot Inputs)

此字段指示辅助快照输入数：
000：无辅助输入
001：1 个辅助输入
010：2 个辅助输入
011：3 个辅助输入
100：4 个辅助输入
101-111：保留

位 27 保留，必须保持复位值

位 26:24 **PPSOUTNUM[2:0]**：PPS 输出数 (Number of PPS Outputs)

此字段指示 PPS 输出数：
000：无 PPS 输出
001：1 个 PPS 输出
010：2 个 PPS 输出
011：3 个 PPS 输出
100：4 个 PPS 输出
101-111：保留

位 23:22 保留，必须保持复位值

位 21:18 **TXCHCNT[3:0]**：DMA 发送通道数 (Number of DMA Transmit Channels)

此字段指示 DMA 发送通道数：
0000：1 个 DMA Tx 通道
0001：2 个 DMA Tx 通道
..
0111：8 个 DMA Tx 通道

位 17:16 保留，必须保持复位值

位 15:12 **RXCHCNT[3:0]**：DMA 接收通道数 (Number of DMA Receive Channels)
该字段指示 DMA 接收通道数：
0000：1 个 DMA Rx 通道
0001：2 个 DMA Rx 通道
..
0111：8 个 DMA Rx 通道

位 11:10 保留，必须保持复位值

位 9:6 **TXQCNT[3:0]**：MTL 发送队列数 (Number of MTL Transmit Queues)
该字段指示 MTL 发送队列数：
0000：1 个 MTL Tx 队列
0001：2 个 MTL Tx 队列
..
0111：8 个 MTL Tx 队列

位 5:4 保留，必须保持复位值

位 3:0 **RXQCNT[3:0]**：MTL 接收队列数 (Number of MTL Receive Queues)
此字段指示 MTL 接收队列数：
0000：1 个 MTL Rx 队列
0001：2 个 MTL Rx 队列
..
0111：8 个 MTL Rx 队列

MDIO 地址寄存器 (ETH_MACMDIOAR)

MDIO address register

偏移地址：0x0200

复位值：0x0000 0000

MDIO 地址寄存器通过管理接口控制外部 PHY 的管理周期。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	PSE	BTB	PA[4:0]					RDA[4:0]				
				rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	NTC[2:0]			CR[3:0]				Res.	Res.	Res.	SKAP	GOC_1	GOC_0	C45E	MB
	rw	rw	rw	rw	rw	rw	rw				rw	rw	rw	rw	rw

位 31:28 保留，必须保持复位值

位 27 **PSE**：报头抑制使能 (Preamble Suppression Enable)
该位置 1 时，SMA 将抑制 32 位报头，并发送仅包含 1 个报头位的 MDIO 帧。
该位为 0 时，MDIO 帧始终具有如 IEEE 规范中定义的 32 位报头。

位 26 **BTB**：背靠背事务 (Back to Back transactions)
该位置 1 且 NTC 的值大于 0 时，MAC 将在帧传输结束时（在发送尾部时钟之前）发出已完成读命令或写命令的通知。这样，软件便可启动下一条命令，而且无论为前一帧生成的尾部时钟数为多少，都会立即执行此命令。
该位复位时，只有在生成尾部时钟之后，才会完成读/写命令（MII 忙碌位清零）。在该模式下，确保始终在每个帧之后生成 NTC。
NTC=0 时，该位不能置 1。

位 25:21 **PA[4:0]**：物理层地址 (Physical Layer Address)
该字段指示 MAC 正在访问哪些符合条款 22 的 PHY 器件（32 个 PHY 器件中）。该字段指示 MAC 正在访问哪些符合条款 45 的 PHY 器件（32 个 PHY 器件中）。

位 20:16 **RDA[4:0]**：寄存器/器件地址 (Register/Device Address)
这些位用于选择所选符合条款 22 的 PHY 器件中的 PHY 寄存器。这些位用于选择所选符合条款 45 的 PHY 的器件 (MMD)。

位 15 保留，必须保持复位值

位 14:12 **NTC[2:0]**：尾部时钟数 (Number of Training Clocks)
该字段控制 MDIO 帧发送结束后在 ETH_MDC 上生成的尾部时钟周期数。有效值范围为 0 到 7。将值编程为 3'h3 表示 MDIO 帧完成传输后 MDC 线上存在三个额外时钟周期。

位 11:8 **CR[3:0]**：CSR 时钟范围 (CSR Clock Range)
CSR 时钟范围选择根据设计中使用的 CSR 时钟频率确定 MDC 时钟的频率：
0000：CSR 时钟 = 60-100 MHz；MDC 时钟 = CSR 时钟/42
0001：CSR 时钟 = 100-150 MHz；MDC 时钟 = CSR 时钟/62
0010：CSR 时钟 = 20-35 MHz；MDC 时钟 = CSR 时钟/16
0011：CSR 时钟 = 35-60 MHz；MDC 时钟 = CSR 时钟/26
0100：CSR 时钟 = 150-250 MHz；MDC 时钟 = CSR 时钟/102
0101：CSR 时钟 = 250-300 MHz；MDC 时钟 = CSR 时钟/124
0110，0111：保留
适用于各个值的建议 CSR 时钟频率范围（位 11 = 0 时）可确保 MDC 时钟大致介于 1.0 MHz 到 2.5 MHz 频率范围之间。
位 11 置 1 时，可以实现高于 2.5 MHz 频率限制（在 IEEE 802.3 中规定）的 MDC 时钟频率，并编程较低值的时钟分频器。例如，当 CSR 时钟频率为 100 MHz，并且将这些位编程为 1010 时，所得到的 MDC 时钟为 12.5 MHz，此值高于 IEEE 802.3 中规定的范围。仅当接口芯片支持更快的 MDC 时钟时才编程以下值：
1000：CSR 时钟/4
1001：CSR 时钟/6
1010：CSR 时钟/8
1011：CSR 时钟/10
1100：CSR 时钟/12
1101：CSR 时钟/14
1110：CSR 时钟/16
1111：CSR 时钟/18

位 7:5 保留，必须保持复位值

位 4 **SKAP**：跳过地址数据包 (Skip Address Packet)
该位置 1 时，SMA 在对增量地址数据包进行读取、写入或后读取操作之前不会发送地址数据包。只有在 C45E 置 1 时，该位才有效。

位 3:2 **GOC**：MII 操作命令 (MII Operation Command)

该位指示 PHY 的操作命令。

00：保留

01：写

10：对符合条款 45 的 PHY 的增量地址进行后读取操作

11：读

使能符合条款 22 的 PHY 时，只有读命令和写命令有效。

位 1 **C45E**：符合条款 45 的 PHY 使能 (Clause 45 PHY Enable)

该位置 1 时，符合条款 45 的 PHY 连接到 MDIO。该位复位时，符合条款 22 的 PHY 连接到 MDIO。

位 0 **MB**：MII 忙碌 (MII Busy)

应用程序将该位置 1 可命令 SMA 启动对 MDIOS 的读访问或写访问。 MAC 会在 MDIO 帧传输完成后将该位清零。因此，只要该位置 1，软件就不能对 ETH_MACMDIOAR 和 ETH_MACMDIODR 寄存器中的任何字段进行写入或更改操作。

对于写传输，应用程序必须先在 ETH_MACMDIODR 寄存器的 MD 字段（C45E 置 1 时，还包括 RA 字段）中写入 16 位数据，然后再将该位置 1。C45E 置 1 时，还应在启动读传输之前，写入 ETH_MACMDIODR 寄存器的 RA 字段。完成读传输（MII 忙碌=0）时，从 PHY 寄存器中读取的数据在 ETH_MACMDIODR 寄存器的 MD 字段中有效。

注： 即使不存在寻址的 PHY，该位的功能也不会改变。

MDIO 数据寄存器 (ETH_MACMDIODR)

MDIO data register

偏移地址：0x0204

复位值：0x0000 0000

MDIO 数据寄存器既存储要写入 PHY 寄存器（该寄存器的地址在 ETH_MACMDIOAR 中指定）的数据，也存储从 PHY 寄存器（该寄存器的地址由 MDIO 地址寄存器指定）读取的数据。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RA[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MD[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:16 **RA[15:0]**：寄存器地址 (Register Address)

只有在 C45E 置 1 时，该字段才有效。它包含要使用 MDIO 帧的 PHY 中的寄存器地址。

位 15:0 **MD[15:0]**：MII 数据 (MII Data)

该字段包含在某次管理读操作之后从 PHY 中读取的 16 位数据值，或在某次管理写操作之前要写入 PHY 的 16 位数据值。

ARP 地址寄存器 (ETH_MACARPAR)

ARP address register

偏移地址：0x0AE0

复位值：0x0000 0000

ARP 地址寄存器包含 MAC 的 IPv4 目标地址。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ARPPA[31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ARPPA[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:0 **ARPPA[31:0]**：ARP 协议地址 (ARP Protocol Address)

该字段包含 MAC 的 IPv4 目标地址。该地址用于与接收到的 ARP 数据包中的目标协议地址字段进行完美匹配。

仅当选择 Enable IPv4 ARP Offload（使能 IPv4 ARP 减荷）选项时，该字段才可用。

地址 0 高位寄存器 (ETH_MACA0HR)

Address 0 high register

偏移地址：0x0300

复位值：0x8000 FFFF

MAC 地址 0 高位寄存器可保存站的第一个 6 字节 MAC 地址的高 16 位。在 MII 接口上接收到的第一个 DA 字节与 MAC 地址低位寄存器的 LS 字节（位 [7:0]）相对应。例如，如果在 MII 上接收到 0x112233445566（第一列通道 0 中的 0x11）作为目标地址，则 MAC 地址 0 寄存器 [47:0] 会与 0x665544332211 进行比较。

如果将 MAC 地址寄存器配置为与 MII 时钟域双同步，则只有在写入 MAC 地址 0 低位寄存器的位 [31:24]（小端模式）或位 [7:0]（大端模式）时，才会触发同步。为了正确进行同步更新，应在目标时钟域中的至少四个时钟周期之后，对该地址低位寄存器执行连续写操作。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
AE	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res
r															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDRHI[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31 **AE**：地址使能 (Address Enable)

该位始终置 1。

位 30:16 保留，必须保持复位值

位 15:0 **ADDRHI[15:0]**：MAC 地址 0 [47:32] (MAC Address0[47:32])

该字段包含第一个 6 字节 MAC 地址的高 16 位 [47:32]。MAC 使用此字段过滤所接收到的数据包，以及将 MAC 地址插入到发送流控制（暂停）数据包中。

地址 x 低位寄存器 (ETH_MACAxLR)

Address x low register

偏移地址：0x0304 + x*0x8（其中 x = 0 到 3）

复位值：0xFFFF FFFF

MAC 地址 x 低位寄存器可保存站的第一个 6 字节 MAC 地址的低 32 位。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ADDRLO[31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDRLO[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:0 **ADDRLO[31:0]**：MAC 地址 x [31:0] (MAC Address x [31:0])（x = 0 到 3）

此字段包含第一个 6 字节 MAC 地址的低 32 位。MAC 使用此字段过滤所接收到的数据包，以及将 MAC 地址插入到发送流控制（暂停）数据包中。

地址 x 高位寄存器 (ETH_MACAxHR)

Address x high register

偏移地址：0x0308 + (x-1)*0x8（其中 x = 1 到 3）

复位值：0x0000 FFFF

MAC 地址 x 高位寄存器可保存站的第二个 6 字节 MAC 地址的高 16 位。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
AE	SA	MBC[5:0]						Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
rw	rw	rw	rw	rw	rw	rw	rw								
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDRHI[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31 **AE**：地址使能 (Address Enable)

此位置 1 时，地址过滤器模块使用第二个 MAC 地址实现完美过滤。此位复位时，地址过滤器模块会忽略用于过滤的地址。

位 30 **SA**：源地址 (Source Address)

此位置 1 时，将使用 MAC 地址 1 [47:0] 与所接收数据包的 SA 字段进行比较。此位复位时，将使用 MAC 地址 x [47:0] 与所接收数据包的 DA 字段进行比较。

位 29:24 **MBC[5:0]**：屏蔽字节控制 (Mask Byte Control)

这些位是用于比较每个 MAC 地址字节的屏蔽控制位。当将其设为高电平时，MAC 不会将所接收到的 DA 或 SA 的相应字节与 MAC 地址 1 寄存器的内容进行比较。每个位都用于控制字节的屏蔽，如下所示：

位 29：寄存器 194[15:8]

位 28：寄存器 194[7:0]

位 27：寄存器 195[31:24]

..

位 24：寄存器 195[7:0]

可通过屏蔽地址的一个或多个字节来过滤一组地址（被称为组地址过滤）。

位 23:16 保留，必须保持复位值

位 15:0 **ADDRHI[15:0]**：MAC 地址 1 [47:32] (MAC Address1 [47:32])

该字段包含第二个 6 字节 MAC 地址的高 16 位 [47:32]。

MMC 控制寄存器 (MMC_CONTROL)

MMC control register

偏移地址：0x0700

复位值：0x0000 0000

该寄存器配置 MMC 工作模式。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	UCDBC	Res.	Res.	CNTPRSTLVL	CNTPRST	CNTFREEZ	RSTONRD	CNTSTOPRO	CNTRST
							rw			rw	rw	rw	rw	rw	rw

位 31:9 保留，必须保持复位值

位 8 **UCDBC**：针对所丢弃广播数据包更新 MMC 计数器 (Update MMC Counters for Dropped Broadcast Packets)

CNTRST 位的优先级高于 CNTPRST 位。因此，软件尝试在同一写周期内将这两个位置 1 时，所有计数器都将清零，并且 CNTPRST 位不会置 1。

置 1 时，MAC 会针对因 ETH_MACPFR 寄存器的 DBF 位置 1 而被丢弃的广播数据包更新所有相关 MMC 计数器。

复位时，不会针对丢弃的广播数据包更新 MMC 计数器。

位 7:6 保留，必须保持复位值

位 5 **CNTPRSTLVL**：全-半预设置 (Full-Half Preset)

该位为低电平且 CNTPRST 位置 1 时，所有 MMC 计数器均预设为几乎一半值。所有八位字节计数器均预设为 0x7FFF_F800（半个 2 KB），所有数据包计数器均预设为 0x7FFF_FFF0（半个 16）。

该位为高电平且 CNTPRST 位置 1 时，所有 MMC 计数器均预设为几乎全值。所有八位字节计数器均预设为 0xFFFF_F800（全 2 KB），所有数据包计数器均预设为 0xFFFF_FFF0（全 16）。

对于 16 位计数器，相应八位字节和数据包计数器的几乎一半预设值为 0x7800 和 0x7FF0。

类似地，16 位计数器的几乎全预设值为 0xF800 和 0xFFF0。

位 4 **CNTPRST**：计数器预设 (Counters Preset)

该位置 1 时，所有计数器均初始化或预设为几乎全值或几乎一半值，具体取决于 CNTPRSTLVL 位。该位在 1 个时钟周期后自动清零。

该位与 CNTPRSTLVL 位一起用于调试和测试中断的有效情况，因为 MMC 计数器变为半-全或全值。

位 3 **CNTFREEZ**：MMC 计数器冻结 (MMC Counter Freeze)

该位置 1 时，会将所有 MMC 计数器冻结为当前值。

该位复位为 0 之前，收到或发送任何数据包 MMC 计数器都不会更新。在此模式下，“读取时复位”位置 1 时，如果读取任何 MMC 计数器，则该计数器也会清零。

位 2 **RSTONRD**：读取时复位 (Reset on Read)

该位置 1 时，读取 MMC 计数器后，该计数器会复位为零（复位后自清零）。读取最低有效字节通道（位 [7:0]）后，计数器会清零。

位 1 **CNTSTOPRO**：计数器停止翻转 (Counter Stop Rollover)

该位置 1 时，计数器达到其最大值后，不会返回到零。

位 0 **CNTRST**：计数器复位 (Counters Reset)

该位置 1 时，所有计数器都将复位。该位在 1 个时钟周期后自动清零。

MMC Rx 中断寄存器 (MMC_RX_INTERRUPT)

MMC Rx interrupt register

偏移地址：0x0704

复位值：0x0000 0000

该寄存器保存所有接收统计计数器生成的中断。

MMC 接收中断寄存器保存在发生以下情况时生成的中断：

- 接收统计计数器达到其最大值的一半（对于 32 位计数器为 0x8000_0000，对于 16 位计数器为 0x8000）。
- 接收统计计数器超过其最大值（对于 32 位计数器为 0xFFFF_FFFF，对于 16 位计数器为 0xFFFF）。

计数器停止翻转置 1 时，中断将置为有效，但计数器保持为全 1。MMC 接收中断寄存器为 32 位寄存器。当读取引发中断的各个 MMC 计数器时，会将中断位清零。必须读取相应计数器的最低有效字节通道（位 [7:0]），才能将中断位清零。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	RXLPITRCIS	RXLPIUSCIS	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RXUCGPIS	Res.
				r	r									r	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RXALGNERPIS	RXCRCERPIS	Res.	Res.	Res.	Res.	Res.
									r	r					

位 31:28 保留，必须保持复位值

位 27 **RXLPITRCIS**：MMC 接收 LPI 转换计数器中断状态 (MMC Receive LPI transition counter interrupt status)

当 Rx_LPI_Tran_Cntr 计数器达到最大值的一半或最大值时，该位置 1。

位 26 **RXLPIUSCIS**：MMC 接收 LPI 微秒计数器中断状态 (MMC Receive LPI microsecond counter interrupt status)

当 Rx_LPI_USEC_Cntr 计数器达到最大值的一半或最大值时，该位置 1。

位 25:18 保留，必须保持复位值

位 17 **RXUCGPIS**：MMC 接收单播良好数据包计数器中断状态 (Receive Unicast Good Packet Counter Interrupt Status)

当 rxunicastpackets_g 计数器达到最大值的一半或最大值时，该位置 1。

位 16:7 保留，必须保持复位值

位 6 **RXALGNERPIS**：MMC 接收对齐错误数据包计数器中断状态 (MMC Receive Alignment Error Packet Counter Interrupt Status)

当 rxalignmenterror 计数器达到最大值的一半或最大值时，该位置 1。

位 5 **RXCRCERPIS**：MMC 接收 CRC 错误数据包计数器中断状态 (MMC Receive CRC Error Packet Counter Interrupt Status)

当 rxcrcerror 计数器达到最大值的一半或最大值时，该位置 1。

位 4:0 保留，必须保持复位值

MMC Tx 中断寄存器 (MMC_TX_INTERRUPT)

MMC Tx interrupt register

偏移地址：0x0708

复位值：0x0000 0000

该寄存器保存所有发送统计计数器生成的中断。

MMC 发送中断寄存器保存发送统计计数器达到其最大值的一半（对于 32 位计数器为 0x8000_0000，对于 16 位计数器为 0x8000）时生成的中断，以及发送统计计数器超过最大值（对于 32 位计数器为 0xFFFF_FFFF，对于 16 位计数器为 0xFFFF）时生成的中断。

计数器停止翻转置 1 时，中断将置为有效，但计数器保持为全 1。

MMC 发送中断寄存器为 32 位寄存器。当读取引发中断的各个 MMC 计数器时，会将中断位清零。

必须读取相应计数器的最低有效字节通道（位 [7:0]），才能将中断位清零。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	TXLPITRCIS	TXLPIUSCIS	Res.	Res.	Res.	Res.	TXGPKTIS	Res.	Res.	Res.	Res.	Res.
				r	r					r					
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TXMCOLGPIS	TXSCOLGPIS	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
r	r														

位 31:28 保留，必须保持复位值

位 27 **TXLPITRCIS**：MMC 发送 LPI 转换计数器中断状态 (MMC Transmit LPI transition counter interrupt status)

当 Tx_LPI_Tran_Cntr 计数器达到最大值的一半或最大值时，该位置 1。

位 26 **TXLPIUSCIS**：MMC 发送 LPI 微秒计数器中断状态 (MMC Transmit LPI microsecond counter interrupt status)

当 Tx_LPI_USEC_Cntr 计数器达到最大值的一半或最大值时，该位置 1。

位 25:22 保留，必须保持复位值

位 21 **TXGPKTIS**：MMC 发送良好数据包计数器中断状态 (MMC Transmit Good Packet Counter Interrupt Status)

当 txpacketcount_g 计数器达到最大值的一半或最大值时，该位置 1。

位 20:16 保留，必须保持复位值

位 15 **TXMCOLGPIS**：MMC 发送多次冲突良好数据包计数器中断状态 (MMC Transmit Multiple Collision Good Packet Counter Interrupt Status)

当 txmulticol_g 计数器达到最大值的一半或最大值时，该位置 1。

位 14 **TXSCOLGPIS**：MMC 发送单次冲突良好数据包计数器中断状态 (MMC Transmit Single Collision Good Packet Counter Interrupt Status)

当 txsinglecol_g 计数器达到最大值的一半或最大值时，该位置 1。

位 13:0 保留，必须保持复位值

MMC Rx 中断屏蔽寄存器 (MMC_RX_INTERRUPT_MASK)

MMC Rx interrupt mask register

偏移地址：0x070C

复位值：0x0000 0000

MMC 接收中断屏蔽寄存器用于保存接收统计计数器达到其最大值的一半或最大值时所生成中断的屏蔽。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	RXLPITRCIM	RXLPIUSCIM	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RXUCGPIM	Res.
				r	rw									rw	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RXALGNERPIM	RXCRCERPIM	Res.	Res.	Res.	Res.	Res.
									rw	rw					

位 31:28 保留，必须保持复位值

位 27 **RXLPITRCIM**：MMC 接收 LPI 转换计数器中断屏蔽 (MMC Receive LPI transition counter interrupt Mask)

当 Rx_LPI_Tran_Cntr 计数器达到最大值的一半或最大值时，将该位置 1 会屏蔽中断。

位 26 **RXLPIUSCIM**：MMC 接收 LPI 微秒计数器中断屏蔽 (MMC Receive LPI microsecond coun)

当 Rx_LPI_USEC_Cntr 计数器达到最大值的一半或最大值时，将该位置 1 会屏蔽中断。

位 25:18 保留，必须保持复位值

位 17 **RXUCGPIM**：MMC 接收单播良好数据包计数器中断屏蔽 (MMC Receive Unicast Good Packet Counter Interrupt Mask)

当 rxunicastpackets_g 计数器达到最大值的一半或最大值时，将该位置 1 会屏蔽中断。

位 16:7 保留，必须保持复位值

位 6 **RXALGNERPIM**：MMC 接收对齐错误数据包计数器中断屏蔽 (MMC Receive Alignment Error Packet Counter Interrupt Mask)

当 rxalignmenterror 计数器达到最大值的一半或最大值时，将该位置 1 会屏蔽中断。

位 5 **RXCRCERPIM**：MMC 接收 CRC 错误数据包计数器中断屏蔽 (MMC Receive CRC Error Packet Counter Interrupt Mask)

当 rxcrcerror 计数器达到最大值的一半或最大值时，将该位置 1 会屏蔽中断。

位 4:0 保留，必须保持复位值

MMC Tx 中断屏蔽寄存器 (MMC_TX_INTERRUPT_MASK)

MMC Tx interrupt mask register

偏移地址：0x0710

复位值：0x0000 0000

该寄存器保存所有发送统计计数器生成的中断的屏蔽。

MMC 发送中断屏蔽寄存器用于保存发送统计计数器达到其最大值的一半或最大值时所生成中断的屏蔽。该寄存器为 32 位宽。只有在内核配置期间选择任何一个 MMC 发送计数器时，该寄存器才存在。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	TXLPITRCIM	TXLPIUSCIM	Res.	Res.	Res.	Res.	TXGPKTIM	Res.	Res.	Res.	Res.	Res.
				r	rw					rw					

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TXMCOLGPIM	TXSCOLGPIM	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
rw	rw														

位 31:28 保留，必须保持复位值

位 27 **TXLPITRCIM**：MMC 发送 LPI 转换计数器中断屏蔽 (MMC Transmit LPI transition counter interrupt Mask)

当 Tx_LPI_Tran_Cntr 计数器达到最大值的一半或最大值时，将该位置 1 会屏蔽中断。

位 26 **TXLPIUSCIM**：MMC 发送 LPI 微秒计数器中断屏蔽 (MMC Transmit LPI microsecond counter interrupt Mask)

当 Tx_LPI_USEC_Cntr 计数器达到最大值的一半或最大值时，将该位置 1 会屏蔽中断。

位 25:22 保留，必须保持复位值

位 21 **TXGPKTIM**：MMC 发送良好数据包计数器中断屏蔽 (MMC Transmit Good Packet Counter Interrupt Mask)

当 txpacketcount_g 计数器达到最大值的一半或最大值时，将该位置 1 会屏蔽中断。

位 20:16 **RESERVED_TXGOCTIM**：保留。

位 15 **TXMCOLGPIM**：MMC 发送多次冲突良好数据包计数器中断屏蔽 (MMC Transmit Multiple Collision Good Packet Counter Interrupt Mask)

当 txmulticol_g 计数器达到最大值的一半或最大值时，将该位置 1 会屏蔽中断。

位 14 **TXSCOLGPIM**：MMC 发送单次冲突良好数据包计数器中断屏蔽 (MMC Transmit Single Collision Good Packet Counter Interrupt Mask)

当 txsinglecol_g 计数器达到最大值的一半或最大值时，将该位置 1 会屏蔽中断。

位 13:0 保留，必须保持复位值

Tx 单次冲突良好数据包寄存器 (TX_SINGLE_COLLISION_GOOD_PACKETS)

Tx single collision good packets register

偏移地址：0x074C

复位值：0x0000 0000

该寄存器提供在半双工模式下于单次冲突后由以太网外设成功发送的数据包的数量。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TXSNGLCOLG[31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TXSNGLCOLG[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:0 **TXSNGLCOLG[31:0]**：Tx 单次冲突良好数据包 (Tx Single Collision Good Packets)

该字段指示在半双工模式下于单次冲突后成功发送的数据包的数量。

Tx 多次冲突良好数据包寄存器 (TX_MULTIPLE_COLLISION_GOOD_PACKETS)

Tx multiple collision good packets register

偏移地址：0x0750

复位值：0x0000 0000

该寄存器提供在半双工模式下于多次冲突后由以太网外设成功发送的数据包的数量。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TXMULTCOLG[31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TXMULTCOLG[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:0 **TXMULTCOLG[31:0]**：Tx 多次冲突良好数据包 (Tx Multiple Collision Good Packets)

该字段指示在半双工模式下于多次冲突后成功发送的数据包的数量。

Tx 良好数据包计数寄存器 (TX_PACKET_COUNT_GOOD)

Tx packet count good register

偏移地址：0x0768

复位值：0x0000 0000

该寄存器提供以太网外设发送的良好数据包的数量。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TXPKTG[31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TXPKTG[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:0 **TXPKTG[31:0]**：Tx 良好数据包计数 (Tx Packet Count Good)

该字段指示发送的良好数据包的数量。

Rx CRC 错误数据包寄存器 (RX_CRC_ERROR_PACKETS)

Rx CRC error packets register

偏移地址：0x0794

复位值：0x0000 0000

该寄存器提供以太网外设接收到的含 CRC 错误的数据包数量。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RXCRCERR[31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RXCRCERR[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:0 **RXCRCERR[31:0]**：Rx CRC 错误数据包 (Rx CRC Error Packets)

该字段指示接收到的含 CRC 错误的数据包数量。

Rx 对齐错误数据包寄存器 (RX_ALIGNMENT_ERROR_PACKETS)

Rx alignment error packets register

偏移地址：0x0798

复位值：0x0000 0000

该寄存器提供以太网外设接收到的含对齐 (dribble) 错误的数据包数量。它仅在 10/100 模式下有效。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RXALGNERR[31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RXALGNERR[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:0 **RXALGNERR[31:0]**：Rx 对齐错误数据包 (Rx Alignment Error Packets)

该字段指示接收到的含对齐 (dribble) 错误的数据包数量。它仅在 10/100 模式下有效。

Rx 良好单播数据包寄存器 (RX_UNICAST_PACKETS_GOOD)

Rx unicast packets good register

偏移地址：0x07C4

复位值：0x0000 0000

该寄存器提供以太网外设接收的良好单播数据包数量。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RXUCASTG[31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RXUCASTG[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:0 **RXUCASTG[31:0]**：Rx 良好单播数据包 (Rx Unicast Packets Good)

该字段指示接收的良好单播数据包数量。

Tx LPI 微秒定时器寄存器 (TX_LPI_USEC_CNTR)

Tx LPI microsecond timer register

偏移地址：0x07EC

复位值：0x0000 0000

该寄存器提供以太网外设将 Tx LPI 置为有效的微秒数。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TXLPIUSC[31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TXLPIUSC[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:0 **TXLPIUSC[31:0]**：Tx LPI 微秒计数器 (Tx LPI Microseconds Counter)

该字段指示将 Tx LPI 置为有效的微秒数。对于每次 Tx LPI 进入和退出，定时器值均会有 +/- 1 微秒的误差。

Tx LPI 转换计数器寄存器 (TX_LPI_TRAN_CNTR)

Tx LPI transition counter register

偏移地址：0x07F0

复位值：0x0000 0000

该寄存器提供以太网外设已进入 Tx LPI 的次数。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TXLPITRC[31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TXLPITRC[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:0 **TXLPITRC[31:0]**：Tx LPI 转换计数器 (Tx LPI Transition counter)

该字段指示已进入 Tx LPI 的次数。即使在自动模式下进入 Tx LPI（因为 LPI 控制和状态寄存器中的 LPITXA 位置 1），计数器也会递增。

Rx LPI 微秒计数器寄存器 (RX_LPI_USEC_CNTR)

Rx LPI microsecond counter register

偏移地址：0x07F4

复位值：0x0000 0000

该寄存器提供以太网外设对 Rx LPI 进行采样的微秒数。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RXLPIUSC[31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RXLPIUSC[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:0 **RXLPIUSC[31:0]**：Rx LPI 微秒计数器 (Rx LPI Microseconds Counter)

该字段指示将 Rx LPI 置为有效的微秒数。对于每次 Rx LPI 进入和退出，定时器值均会有 +/- 1 微秒的误差。

Rx LPI 转换计数器寄存器 (RX_LPI_TRAN_CNTR)

Rx LPI transition counter register

偏移地址：0x07F8

复位值：0x0000 0000

该寄存器提供以太网外设已进入 Rx LPI 的次数。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RXLPITRC[31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RXLPITRC[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:0 **RXLPITRC[31:0]**：Rx LPI 转换计数器 (Rx LPI Transition counter)

该字段指示已进入 Rx LPI 的次数。

L3 和 L4 控制 0 寄存器 (ETH_MACL3L4C0R)

L3 and L4 control 0 register

偏移地址：0x0900

复位值：0x0000 0000

第 3 层和第 4 层控制寄存器控制第 3 层和第 4 层的过滤器 0 的操作。如果在内核配置期间未选择第 3 层和第 4 层过滤功能，则该寄存器被保留。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	L4DPIM0	L4DPM0	L4SPIM0	L4SPM0	Res.	L4PEN0
										rw	rw	rw	rw		rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
L3HDBM0[4:0]					L3HSBM0[4:0]					L3DAIM0	L3DAM0	L3SAIM0	L3SAM0	Res.	L3PEN0
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw		rw

位 31:22 保留，必须保持复位值

位 21 **L4DPIM0**：第 4 层目标端口反向匹配使能 (Layer 4 Destination Port Inverse Match Enable)

该位置 1 时，将使能第 4 层目标端口号字段以进行反向匹配。该位复位时，将使能第 4 层目标端口号字段以进行完美匹配。

只有在 L4DPM0 位置为高电平时，该位才有效且适用。

位 20 **L4DPM0**：第 4 层目标端口匹配使能 (Layer 4 Destination Port Match Enable)

该位置 1 时，将使能第 4 层目标端口号字段以进行匹配。该位复位时，MAC 将忽略用于匹配的第 4 层目标端口号字段。

位 19 **L4SPIM0**：第 4 层源端口反向匹配使能 (Layer 4 Source Port Inverse Match Enable)

该位置 1 时，将使能第 4 层源端口号字段以进行反向匹配。该位复位时，将使能第 4 层源端口号字段以进行完美匹配。

只有在 L4SPM0 位置为高电平时，该位才有效且适用。

位 18 **L4SPM0**：第 4 层源端口匹配使能 (Layer 4 Source Port Match Enable)

该位置 1 时，将使能第 4 层源端口号字段以进行匹配。该位复位时，MAC 将忽略用于匹配的第 4 层源端口号字段。

位 17 保留，必须保持复位值

位 16 **L4PEN0**：第 4 层协议使能 (Layer 4 Protocol Enable)

该位置 1 时，将使用 UDP 数据包的源端口号和目标端口号字段进行匹配。该位复位时，将使用 TCP 数据包的源端口号和目标端口号字段进行匹配。

只有当 L4SPM0 或 L4DPM0 位置 1 时，才能完成第 4 层匹配。

位 15:11 **L3HDBM0[4:0]**：第 3 层 IP DA 高位匹配 (Layer 3 IP DA Higher Bits Match)

IPv4 数据包：

该字段包含在 IPv4 数据包中匹配的 IP 目标地址的高位数目。下面对该字段的各个值进行了说明：

0：未屏蔽任何位。

1：屏蔽 LSb[0]。

2：屏蔽两位 LSb [1:0]。

..

31：屏蔽除了 MSb 之外的所有位。

IPv6 数据包：

该字段的位 [12:11] 对应于 L3HSBM0 的位 [6:5]，表示在 IPv6 数据包中被屏蔽的 IP 源地址或目标地址的低位数目。下面对 L3HDBM0[1:0] 和 L3HSBM0 位的各个值进行了说明：

0：未屏蔽任何位。

1：屏蔽 LSb[0]。

2：屏蔽两位 LSb [1:0]。

..

127：屏蔽除了 MSb 之外的所有位。

只有在 L3DAM0 或 L3SAM0 位置 1 时，该字段才有效且适用。

位 10:6 **L3HSBM0[4:0]**：第 3 层 IP SA 高位匹配 (Layer 3 IP SA Higher Bits Match)

IPv4 数据包：

该字段包含为在 IPv4 数据包中匹配而被屏蔽的 IP 源地址的低位数目。下面对该字段的各个值进行了说明：

0：未屏蔽任何位。

1：屏蔽 LSb[0]。

2：屏蔽两位 LSb [1:0]。

..

31：屏蔽除了 MSb 之外的所有位。

IPv6 数据包：

该字段包含 L3HSBM0 的位 [4:0]。这些位表示 IPv6 数据包中匹配的 IP 源地址或目标地址的高位数目。只有在 L3DAM0 或 L3SAM0 位置为高电平时，该字段才有效且适用。

位 5 **L3DAIM0**：第 3 层 IP DA 反向匹配使能 (Layer 3 IP DA Inverse Match Enable)

该位置 1 时，将使能第 3 层 IP 目标地址字段以进行反向匹配。该位复位时，将使能第 3 层 IP 目标地址字段以进行完美匹配。

只有在 L3DAM0 位置为高电平时，该位才有效且适用。

位 4 **L3DAM0**：第 3 层 IP DA 匹配使能 (Layer 3 IP DA Match Enable)

该位置 1 时，将使能第 3 层 IP 目标地址字段以进行匹配。该位复位时，MAC 将忽略用于匹配的第 3 层 IP 目标地址字段。

注： L3PEN0 位置 1 时，应将该位或 L3SAM0 位置 1，因为这样可检查 IPv6 DA 或 SA 以进行过滤。

位 3 **L3SAIM0**：第 3 层 IP SA 反向匹配使能 (Layer 3 IP SA Inverse Match Enable)

该位置 1 时，将使能第 3 层 IP 源地址字段以进行反向匹配。该位复位时，将使能第 3 层 IP 源地址字段以进行完美匹配。

只有在 L3SAM0 位置 1 时，该位才有效且适用。

位 2 **L3SAM0**：第 3 层 IP SA 匹配使能 (Layer 3 IP SA Match Enable)

该位置 1 时，将使能第 3 层 IP 源地址字段以进行匹配。该位复位时，MAC 将忽略用于匹配的第 3 层 IP 源地址字段。

注： L3PEN0 位置 1 时，应将该位或 L3DAM0 位置 1，因为这样可检查 IPv6 SA 或 DA 以进行过滤。

位 1 保留，必须保持复位值

位 0 **L3PEN0**：第 3 层协议使能 (Layer 3 Protocol Enable)

该位置 1 时，将为 IPv6 数据包使能第 3 层 IP 源地址或目标地址匹配。该位复位时，将为 IPv4 数据包使能第 3 层 IP 源地址或目标地址匹配。

只有当 L3SAM0 或 L3DAM0 位置 1 时，才能完成第 3 层匹配。

第 4 层地址过滤器 0 寄存器 (ETH_MACL4A0R)

Layer4 address filter 0 register

偏移地址：0x0904

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
L4DP0[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
L4SP0[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:16 **L4DP0[15:0]**：第 4 层目标端口号字段 (Layer 4 Destination Port Number Field)

ETH_MACL3L4C0R 寄存器中的 L4PEN0 位复位且 L4DPM0 位置 1 时，该字段包含要与 IPv4 或 IPv6 数据包中的 TCP 目标端口号字段进行匹配的值。

ETH_MACL3L4C0R 寄存器中的 L4PEN0 和 L4DPM0 位置 1 时，该字段包含要与 IPv4 或 IPv6 数据包中的 UDP 目标端口号字段进行匹配的值。

位 15:0 **L4SP0[15:0]**：第 4 层源端口号字段 (Layer 4 Source Port Number Field)

ETH_MACL3L4C0R 寄存器中的 L4PEN0 位复位且 L4DPM0 位置 1 时，该字段包含要与 IPv4 或 IPv6 数据包中的 TCP 源端口号字段进行匹配的值。

ETH_MACL3L4C0R 寄存器中的 L4PEN0 和 L4DPM0 位置 1 时，该字段包含要与 IPv4 或 IPv6 数据包中的 UDP 源端口号字段进行匹配的值。

第 3 层地址 0 过滤器 0 寄存器 (ETH_MACL3A00R)

Layer 3 Address 0 filter 0 register

偏移地址：0x0910

复位值：0x0000 0000

对于 IPv4 数据包，第 3 层地址 0 过滤器 0 寄存器包含 32 位 IP 源地址字段。对于 IPv6 数据包，它包含 128 位 IP 源地址或目标地址字段的位 [31:0]。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
L3A00[31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
L3A00[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:0 **L3A00[31:0]**：第 3 层地址 0 字段 (Layer 3 Address 0 Field)

ETH_MACL3L4C0R 寄存器中的 L3PEN0 和 L3SAM0 位置 1 时，该字段包含要与 IPv6 数据包中 IP 源地址字段的位 [31:0] 进行匹配的值。

ETH_MACL3L4C0R 寄存器中的 L3PEN0 和 L3DAM0 位置 1 时，该字段包含要与 IPv6 数据包中 IP 目标地址字段的位 [31:0] 进行匹配的值。

ETH_MACL3L4C0R 寄存器中的 L3PEN0 位复位且 L3SAM0 位置 1 时，该字段包含要与 IPv4 数据包中的 IP 源地址字段进行匹配的值。

第 3 层地址 1 过滤器 0 寄存器 (ETH_MACL3A10R)

Layer3 address 1 filter 0 register

偏移地址：0x0914

复位值：0x0000 0000

对于 IPv4 数据包，第 3 层地址 1 过滤器 0 寄存器包含 32 位 IP 目标地址字段。对于 IPv6 数据包，它包含 128 位 IP 源地址或目标地址字段的位 [63:32]。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
L3A10[31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
L3A10[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:0 **L3A10[31:0]**：第 3 层地址 1 字段 (Layer 3 Address 1 Field)

ETH_MACL3L4C0R 寄存器中的 L3PEN0 和 L3SAM0 位置 1 时，该字段包含要与 IPv6 数据包中 IP 源地址字段的位 [63:32] 进行匹配的值。

ETH_MACL3L4C0R 寄存器中的 L3PEN0 和 L3DAM0 位置 1 时，该字段包含要与 IPv6 数据包中 IP 目标地址字段的位 [63:32] 进行匹配的值。

ETH_MACL3L4C0R 寄存器中的 L3PEN0 位复位且 L3SAM0 位置 1 时，该字段包含要与 IPv4 数据包中的 IP 目标地址字段进行匹配的值。

第 3 层地址 2 过滤器 0 寄存器 (ETH_MACL3A20)

Layer3 Address 2 filter 0 register

偏移地址：0x0918

复位值：0x0000 0000

对于 IPv4 数据包，第 3 层地址 2 过滤器 0 寄存器保留。对于 IPv6 数据包，它包含 128 位 IP 源地址或目标地址字段的位 [95:64]。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
L3A20[31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
L3A20[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:0 **L3A20[31:0]**：第 3 层地址 2 字段 (Layer 3 Address 2 Field)

ETH_MACL3L4C0R 寄存器中的 L3PEN0 和 L3SAM0 位置 1 时，该字段包含要与 IPv6 数据包中 IP 源地址字段的位 [95:64] 进行匹配的值。

ETH_MACL3L4C0R 寄存器中的 L3PEN0 和 L3DAM0 位置 1 时，该字段包含要与 IPv6 数据包中 IP 目标地址字段的位 [95:64] 进行匹配的值。

ETH_MACL3L4C0R 寄存器中的 L3PEN0 位复位时，不使用该字段。

第 3 层地址 3 过滤器 0 寄存器 (ETH_MACL3A30)

Layer3 Address 3 filter 0 register

偏移地址：0x091C

复位值：0x0000 0000

对于 IPv4 数据包，第 3 层地址 3 过滤器 0 寄存器保留。对于 IPv6 数据包，它包含 128 位 IP 源地址或目标地址字段的位 [127:96]。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
L3A30[31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
L3A30[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:0 **L3A30[31:0]**：第 3 层地址 3 字段 (Layer 3 Address 3 Field)

ETH_MACL3L4C0R 寄存器中的 L3PEN0 和 L3SAM0 位置 1 时，该字段包含要与 IPv6 数据包中 IP 源地址字段的位 [127:96] 进行匹配的值。

ETH_MACL3L4C0R 寄存器中的 L3PEN0 和 L3DAM0 位置 1 时，该字段包含要与 IPv6 数据包中 IP 目标地址字段的位 [127:96] 进行匹配的值。

ETH_MACL3L4C0R 寄存器中的 L3PEN0 位复位时，不使用该字段。

L3 和 L4 控制 1 寄存器 (ETH_MACL3L4C1R)

L3 and L4 control 1 register

偏移地址：0x0930

复位值：0x0000 0000

第 3 层和第 4 层控制寄存器控制第 3 层和第 4 层的过滤器 0 的操作。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	RESERVED_DMCHEN1	Res.	Res.	Res.	Res.	Res.	Res.	L4DPIM1	L4DPM1	L4SPIM1	L4SPM1	Res.	L4PEN1
			r							rw	rw	rw	rw		rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
L3HDBM1[4:0]					L3HSBM1[4:0]					L3DAIM1	L3DAM1	L3SAIM1	L3SAM1	Res.	L3PEN1
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw		rw

位 31:22 保留，必须保持复位值

位 21 **L4DPIM1**：第 4 层目标端口反向匹配使能 (Layer 4 Destination Port Inverse Match Enable)
该位置 1 时，将使能第 4 层目标端口号字段以进行反向匹配。该位复位时，将使能第 4 层目标端口号字段以进行完美匹配。
只有在 L4DPM0 位置为高电平时，该位才有效且适用。

位 20 **L4DPM1**：第 4 层目标端口匹配使能 (Layer 4 Destination Port Match Enable)
该位置 1 时，将使能第 4 层目标端口号字段以进行匹配。该位复位时，MAC 将忽略用于匹配的第 4 层目标端口号字段。

位 19 **L4SPIM1**：第 4 层源端口反向匹配使能 (Layer 4 Source Port Inverse Match Enable)
该位置 1 时，将使能第 4 层源端口号字段以进行反向匹配。该位复位时，将使能第 4 层源端口号字段以进行完美匹配。
只有在 L4SPM0 位置为高电平时，该位才有效且适用。

位 18 **L4SPM1**：第 4 层源端口匹配使能 (Layer 4 Source Port Match Enable)
该位置 1 时，将使能第 4 层源端口号字段以进行匹配。该位复位时，MAC 将忽略用于匹配的第 4 层源端口号字段。

位 17 保留，必须保持复位值

位 16 **L4PEN1**：第 4 层协议使能 (Layer 4 Protocol Enable)
该位置 1 时，将使用 UDP 数据包的源端口号和目标端口号字段进行匹配。该位复位时，将使用 TCP 数据包的源端口号和目标端口号字段进行匹配。
只有当 L4SPM0 或 L4DPM0 位置 1 时，才能完成第 4 层匹配。

位 15:11 **L3HDBM1[4:0]**：第 3 层 IP DA 高位匹配 (Layer 3 IP DA Higher Bits Match)

IPv4 数据包：

该字段包含在 IPv4 数据包中匹配的 IP 目标地址的高位数目。下面对该字段的各个值进行了说明：

0：未屏蔽任何位。

1：屏蔽 LSb[0]。

2：屏蔽两位 LSb [1:0]。

..

31：屏蔽除了 MSb 之外的所有位。

IPv6 数据包：

该字段的位 [12:11] 对应于 L3HSBM0 的位 [6:5]，表示在 IPv6 数据包中被屏蔽的 IP 源地址或目标地址的低位数目。下面对 L3HDBM0[1:0] 和 L3HSBM0 位的各个值进行了说明：

0：未屏蔽任何位。

1：屏蔽 LSb[0]。

2：屏蔽两位 LSb [1:0]。

..

127：屏蔽除了 MSb 之外的所有位。

只有在 L3DAM0 或 L3SAM0 位置 1 时，该字段才有效且适用。

位 10:6 **L3HSBM1[4:0]**：第 3 层 IP SA 高位匹配 (Layer 3 IP SA Higher Bits Match)

IPv4 数据包：

该字段包含为在 IPv4 数据包中匹配而被屏蔽的 IP 源地址的低位数目。下面对该字段的各个值进行了说明：

0：未屏蔽任何位。

1：屏蔽 LSb[0]。

2：屏蔽两位 LSb [1:0]。

..

31：屏蔽除了 MSb 之外的所有位。

IPv6 数据包：

该字段包含 L3HSBM0 的位 [4:0]。这些位表示 IPv6 数据包中匹配的 IP 源地址或目标地址的高位数目。只有在 L3DAM0 或 L3SAM0 位置为高电平时，该字段才有效且适用。

位 5 **L3DAIM1**：第 3 层 IP DA 反向匹配使能 (Layer 3 IP DA Inverse Match Enable)

该位置 1 时，将使能第 3 层 IP 目标地址字段以进行反向匹配。该位复位时，将使能第 3 层 IP 目标地址字段以进行完美匹配。

只有在 L3DAM0 位置为高电平时，该位才有效且适用。

位 4 **L3DAM1**：第 3 层 IP DA 匹配使能 (Layer 3 IP DA Match Enable)

该位置 1 时，将使能第 3 层 IP 目标地址字段以进行匹配。该位复位时，MAC 将忽略用于匹配的第 3 层 IP 目标地址字段。

注： L3PEN0 位置 1 时，应将该位或 L3SAM0 位置 1，因为这样可检查 IPv6 DA 或 SA 以进行过滤。

位 3 **L3SAIM1**：第 3 层 IP SA 反向匹配使能 (Layer 3 IP SA Inverse Match Enable)

该位置 1 时，将使能第 3 层 IP 源地址字段以进行反向匹配。该位复位时，将使能第 3 层 IP 源地址字段以进行完美匹配。

只有在 L3SAM0 位置 1 时，该位才有效且适用。

位 2 **L3SAM1**：第 3 层 IP SA 匹配使能 (Layer 3 IP SA Match Enable)

该位置 1 时，将使能第 3 层 IP 源地址字段以进行匹配。该位复位时，MAC 将忽略用于匹配的第 3 层 IP 源地址字段。

注： L3PEN0 位置 1 时，应将该位或 L3DAM0 位置 1，因为这样可检查 IPv6 SA 或 DA 以进行过滤。

位 1 保留，必须保持复位值

位 0 **L3PEN1**：第 3 层协议使能 (Layer 3 Protocol Enable)

该位置 1 时，将为 IPv6 数据包使能第 3 层 IP 源地址或目标地址匹配。该位复位时，将为 IPv4 数据包使能第 3 层 IP 源地址或目标地址匹配。

只有当 L3SAM0 或 L3DAM0 位置 1 时，才能完成第 3 层匹配。

第 4 层地址过滤器 1 寄存器 (ETH_MACL4A1R)

Layer 4 address filter 1 register

偏移地址：0x0934

复位值：0x0000 0000

如果在配置内核时未选择 Enable Layer 3 and Layer 4 Packet Filter（使能第 3 层和第 4 层数据包过滤器）选项，则第 4 层地址 0 寄存器和寄存器 580 至 667 保留（处于 RO 状态，具有默认值）。

可通过在配置内核时选择 Synchronize Layer 3 and Layer 4 Address Registers to Rx Clock Domain（将第 3 层和第 4 层地址寄存器与 Rx 时钟域同步）选项，来将第 3 层和第 4 层地址寄存器配置为双同步。如果选择此选项，则仅在写入第 3 层和第 4 层地址寄存器的位 [31:24]（小端模式）或位 [7:0]（大端模式）时，才会触发同步。为了正确进行同步更新，应在目标时钟的至少四个时钟周期延迟之后，对相同第 3 层和第 4 层地址寄存器执行连续写操作。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
L4DP1[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
L4SP1[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:16 **L4DP1[15:0]**：第 4 层目标端口号字段 (Layer 4 Destination Port Number Field)

ETH_MACL3L4C0R 寄存器中的 L4PEN0 位复位且 L4DPM0 位置 1 时，该字段包含要与 IPv4 或 IPv6 数据包中的 TCP 目标端口号字段进行匹配的值。

ETH_MACL3L4C0R 寄存器中的 L4PEN0 和 L4DPM0 位置 1 时，该字段包含要与 IPv4 或 IPv6 数据包中的 UDP 目标端口号字段进行匹配的值。

位 15:0 **L4SP1[15:0]**：第 4 层源端口号字段 (Layer 4 Source Port Number Field)

ETH_MACL3L4C0R 寄存器中的 L4PEN0 位复位且 L4DPM0 位置 1 时，该字段包含要与 IPv4 或 IPv6 数据包中的 TCP 源端口号字段进行匹配的值。

ETH_MACL3L4C0R 寄存器中的 L4PEN0 和 L4DPM0 位置 1 时，该字段包含要与 IPv4 或 IPv6 数据包中的 UDP 源端口号字段进行匹配的值。

第 3 层地址 0 过滤器 1 寄存器 (ETH_MACL3A01R)

Layer3 address 0 filter 1 Register

偏移地址：0x0940

复位值：0x0000 0000

对于 IPv4 数据包，第 3 层地址 0 过滤器 0 寄存器包含 32 位 IP 源地址字段。对于 IPv6 数据包，它包含 128 位 IP 源地址或目标地址字段的位 [31:0]。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
L3A01[31:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
L3A01[31:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:0 **L3A01[31:0]**：第 3 层 地址 0 字段 (Layer 3 Address 0 Field)

ETH_MACL3L4C0R 寄存器中的 L3PEN0 和 L3SAM0 位置 1 时，该字段包含要与 IPv6 数据包中 IP 源地址字段的位 [31:0] 进行匹配的值。

ETH_MACL3L4C0R 寄存器中的 L3PEN0 和 L3DAM0 位置 1 时，该字段包含要与 IPv6 数据包中 IP 目标地址字段的位 [31:0] 进行匹配的值。

ETH_MACL3L4C0R 寄存器中的 L3PEN0 位复位且 L3SAM0 位置 1 时，该字段包含要与 IPv4 数据包中的 IP 源地址字段进行匹配的值。

第 3 层地址 1 过滤器 1 寄存器 (ETH_MACL3A11R)

Layer3 address 1 filter 1 register

偏移地址：0x0944

复位值：0x0000 0000

对于 IPv4 数据包，第 3 层地址 1 过滤器 0 寄存器包含 32 位 IP 目标地址字段。对于 IPv6 数据包，它包含 128 位 IP 源地址或目标地址字段的位 [63:32]。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
L3A11[31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
L3A11[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:0 **L3A11[31:0]**：第 3 层地址 1 字段 (Layer 3 Address 1 Field)

ETH_MACL3L4C0R 寄存器中的 L3PEN0 和 L3SAM0 位置 1 时，该字段包含要与 IPv6 数据包中 IP 源地址字段的位 [63:32] 进行匹配的值。

ETH_MACL3L4C0R 寄存器中的 L3PEN0 和 L3DAM0 位置 1 时，该字段包含要与 IPv6 数据包中 IP 目标地址字段的位 [63:32] 进行匹配的值。

ETH_MACL3L4C0R 寄存器中的 L3PEN0 位复位且 L3SAM0 位置 1 时，该字段包含要与 IPv4 数据包中的 IP 目标地址字段进行匹配的值。

第 3 层地址 2 过滤器 1 寄存器 (ETH_MACL3A21R)

Layer3 address 2 filter 1 Register

偏移地址：0x0948

复位值：0x0000 0000

对于 IPv4 数据包，第 3 层地址 2 过滤器 0 寄存器保留。对于 IPv6 数据包，它包含 128 位 IP 源地址或目标地址字段的位 [95:64]。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
L3A21[31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
L3A21[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:0 **L3A21[31:0]**：第 3 层地址 2 字段 (Layer 3 Address 2 Field)

ETH_MACL3L4C0R 寄存器中的 L3PEN0 和 L3SAM0 位置 1 时，该字段包含要与 IPv6 数据包中 IP 源地址字段的位 [95:64] 进行匹配的值。

ETH_MACL3L4C0R 寄存器中的 L3PEN0 和 L3DAM0 位置 1 时，该字段包含要与 IPv6 数据包中 IP 目标地址字段的位 [95:64] 进行匹配的值。

ETH_MACL3L4C0R 寄存器中的 L3PEN0 位复位时，不使用该字段。

第 3 层地址 3 过滤器 1 寄存器 (ETH_MACL3A31R)

Layer3 address 3 filter 1 register

偏移地址：0x94C

复位值：0x0000 0000

对于 IPv4 数据包，第 3 层地址 3 过滤器 0 寄存器保留。对于 IPv6 数据包，它包含 128 位 IP 源地址或目标地址字段的位 [127:96]。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
L3A31[31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
L3A31[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:0 **L3A31[31:0]**：第 3 层地址 3 字段 (Layer 3 Address 3 Field)

ETH_MACL3L4C0R 寄存器中的 L3PEN0 和 L3SAM0 位置 1 时，该字段包含要与 IPv6 数据包中 IP 源地址字段的位 [127:96] 进行匹配的值。

ETH_MACL3L4C0R 寄存器中的 L3PEN0 和 L3DAM0 位置 1 时，该字段包含要与 IPv6 数据包中 IP 目标地址字段的位 [127:96] 进行匹配的值。

ETH_MACL3L4C0R 寄存器中的 L3PEN0 位复位时，不使用该字段。

时间戳控制寄存器 (ETH_MACTSCR)

Timestamp control Register

偏移地址：0x0B00

复位值：0x0000 2000

该寄存器控制系统时间发生器的工作以及针对接收器中时间戳的 PTP 数据包处理。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	TXTSSTSM	Res.	Res.	Res.	Res.	CSC	TSENMACADDR	SNAPTYPSEL[1:0]	
							rw					r	rw	rw	

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TSMSTRENA	TSEVNTENA	TSIPV4ENA	TSIPV6ENA	TSIPENA	TSVER2ENA	TSCTRLSSR	TSENALL	Res.	Res.	TSADDREG	Res.	TSUPDT	TSINIT	TSCFUPDT	TSENA
rw	rw	rw	rw	rw	rw	rw	rw			rw		rw	rw	rw	rw

位 31:25 保留，必须保持复位值

位 24 **TXTSSTSM**：发送时间戳状态模式 (Transmit Timestamp Status Mode)

该位置 1 时，MAC 会覆盖较早发送时间戳状态，即使软件未对其进行读取操作也如此。MAC 通过将 ETH_MACTxTSSNR 寄存器的 TXTSSTSMIS 位置 1 对此进行指示。

该位复位时，如果软件未读取先前数据包的时间戳状态，则 MAC 将忽略当前数据包的时间戳状态。MAC 通过将 ETH_MACTxTSSSR 寄存器的 TXTSSTSHI 位置 1 对此进行指示。

位 23:20 保留，必须保持复位值

位 19 **CSC**：针对 PTP over UDP/IPv4 数据包使能在 OST 期间校准校验和 (Enable checksum correction during OST for PTP over UDP/IPv4 packets)

该位置 1 时，对于一步时间戳操作中对原始时间戳和/或校准字段进行的更改，通过 UDP/IPv4 发送的 PTP 消息的最后两个字节会被更新，以保持 UDP 校验和的正确性。应用程序将构建含这两个空字节的数据包。

该位复位时，不会进行任何更新来保持 UDP 校验和的正确性。应用程序将构建 UDP 校验和被设为 0 的数据包。

位 18 **TSENMACADDR**：使能 MAC 地址以用于 PTP 数据包过滤 (Enable MAC Address for PTP Packet Filtering)

如果该位置 1，则在通过以太网直接发送 PTP 时，会使用 DA MAC 地址（与任一 MAC 地址寄存器匹配）过滤 PTP 数据包。

如果该位置 1，则在通过以太网直接发送 PTP 时，对于所接收到的具有 DA（包含特殊多播或单播地址，与 MAC 地址寄存器中编程的地址匹配）的 PTP 数据包，将被视为要进行处理的数据包，如下所述。

对于正常时间戳操作，会将 MAC 地址寄存器 0 到 31 用于单播目标地址匹配。

对于 PTP 减荷，仅将 MAC 地址寄存器 0 用于单播目标地址匹配。

位 17:16 **SNAPTYPSEL[1:0]**：选择 PTP 数据包以拍摄快照 (Select PTP packets for Taking Snapshots)

这些位与位 15 和位 14 共同决定需要拍摄快照的 PTP 数据包类型组。时间戳快照对寄存器位的相依性表格中给出了相应编码。

位 15 **TSMSTRENA**：使能主节点相关消息的快照 (Enable Snapshot for Messages Relevant to Master)

此位置 1 时，仅拍摄主节点相关消息的快照。否则，将拍摄从节点相关消息的快照。

位 14 **TSEVNTENA**：使能事件消息的时间戳快照 (Enable Timestamp Snapshot for Event Messages)

此位置 1 时，仅拍摄事件消息的时间戳快照（SYNC、Delay_Req、Pdelay_Req 或 Pdelay_Resp）。此位复位时，将拍摄除 Announce、Management 和 Signaling 以外所有消息的快照。有关时间戳快照的更多信息，请参见时间戳快照对寄存器位的相依性表格。

位 13 **TSIPV4ENA**：使能对通过 IPv4-UDP 发送的 PTP 数据包的处理 (Enable Processing of PTP Packets Sent over IPv4-UDP)

该位置 1 时，MAC 接收器处理封装在 IPv4-UDP 数据包中的 PTP 数据包。该位复位时，MAC 将忽略通过 IPv4-UDP 数据包传送的 PTP。默认情况下，此位置 1。

位 12 **TSIPV6ENA**：使能对通过 IPv6-UDP 发送的 PTP 数据包的处理 (Enable Processing of PTP Packets Sent over IPv6-UDP)

该位置 1 时，MAC 接收器处理封装在 IPv6-UDP 数据包中的 PTP 数据包。该位清零时，MAC 将忽略通过 IPv6-UDP 数据包传送的 PTP。

位 11 **TSIPENA**：使能对 PTP over Ethernet 数据包的处理 (Enable Processing of PTP over Ethernet Packets)

该位置 1 时，MAC 接收器处理直接封装在以太网数据包中的 PTP 数据包。该位复位时，MAC 将忽略 PTP over Ethernet 数据包。

位 10 **TSVER2ENA**：针对版本 2 格式使能 PTP 数据包处理 (Enable PTP Packet Processing for Version 2 Format)

该位置 1 时，会使用 IEEE 1588 版本 2 格式来处理 PTP 数据包。该位复位时，会使用 IEEE 1588 版本 1 格式来处理 PTP 数据包。“PTP 处理和控制”中对 IEEE 1588 格式进行了介绍。

位 9 **TSCTRLSSR**：时间戳数字或二进制翻转控制 (Timestamp Digital or Binary Rollover Control)

该位置 1 时，时间戳低位寄存器会在 0x3B9A_C9FF 值之后翻转（即，1 纳秒精度），并递增时间戳（高位）秒数。该位复位时，亚秒寄存器的翻转值为 0x7FFF_FFFF。必须根据 PTP 参考时钟频率和该位的值，正确对亚秒增量进行编程。

位 8 **TSENALL**：针对所有数据包使能时间戳 (Enable Timestamp for All Packets)

此位置 1 时，会针对 MAC 所接收的全部数据包使能时间戳快照。

位 7:6 保留，必须保持复位值

位 5 **TSADDREG**：更新加数寄存器 (Update Addend Register)

此位置 1 时，时间戳加数寄存器的内容会被更新到 PTP 块中以进行精密校准。更新结束时此位会清零。该位在置 1 之前应为零。

位 4 保留，必须保持复位值

位 3 **TSUPDT**：更新时间戳 (Update Timestamp)

该位置 1 时，将使用 ETH_MACSTSUR 和 ETH_MACSTNUR 中指定的值更新（加或减）系统时间。

该位在更新之前应为零。通过硬件完成更新时，此位会复位。不会更新时间戳高位字寄存器（如果在内核配置期间使能）。

位 2 **TSINIT**：初始化时间戳 (Initialize Timestamp)

该位置 1 时，系统时间将以 MAC 寄存器 80（系统时间秒更新寄存器）和 MAC 寄存器 81（系统时间纳秒更新寄存器）中指定的值进行初始化（覆盖）。

该位在更新之前应为零。该位在完成初始化时会复位。只能初始化时间戳高位字寄存器（如果在内核配置期间使能）。

位 1 **TSCFUPDT**：时间戳精密或粗略更新 (Fine or Coarse Timestamp Update)

该位置 1 时，使用精密方法更新系统时间戳。该位复位时，使用粗略方法更新系统时间戳。

位 0 **TSENA**：使能时间戳 (Enable Timestamp)

该位置 1 时，为发送和接收数据包添加时间戳。该位被禁止时，不会为发送和接收数据包添加时间戳，并且时间戳发生器也会被挂起。使能此模式后，需要初始化时间戳（系统时间）。

在接收端，MAC 只在该位置 1 时才处理 1588 个数据包。

表 544 指示根据 ETH_MACTSCR 寄存器中的 SNAPTYPSEL 字段拍摄快照的 PTP 消息。

表 544. 时间戳快照对寄存器位的相依性

SNAPTYPSEL	TSMSTRENA	TSEVNTENA	PTP 消息
00	X	0	SYNC、Follow_Up、Delay_Req、Delay_Resp
00	0	1	SYNC
00	1	1	Delay_Req
01	X	0	SYNC、Follow_Up、Delay_Req、Delay_Resp、Pdelay_Req、Pdelay_Resp、Pdelay_Resp_Follow_Up
01	0	1	SYNC、Pdelay_Req、Pdelay_Resp
01	1	1	Delay_Req、Pdelay_Req、Pdelay_Resp
10	X	X	SYNC、Delay_Req
11	X	X	Pdelay_Req、Pdelay_Resp

亚秒增量寄存器 (ETH_MACSSIR)

Sub-second increment register

偏移地址：0x0B04

复位值：0x0000 0000

只有在没有外部时间戳输入的情况下选择 IEEE 1588 时间戳功能时，才会存在亚秒增量寄存器。在粗略更新模式下 [ETH_MACTSCR 寄存器中的位 1]，此寄存器中的值在每个 HCLK 时钟周期后都会被添加到系统时间。在精密更新模式下，此寄存器中的值将在累加器溢出时被添加到系统时间。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SSINC[7:0]							
								rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SNSINC[7:0]								Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
rw	rw	rw	rw	rw	rw	rw	rw								

位 31:24 保留，必须保持复位值

位 23:16 **SSINC[7:0]**：亚秒增量值 (Sub-second Increment Value)

在该字段中编程的值在（clk_ptp_i 的）每个时钟周期都会与亚秒寄存器的内容进行累加。例如，当 PTP 时钟为 50 MHz（周期为 20 ns）时，如果系统时间纳秒寄存器的精度为 1 ns [ETH_MACTSCR 中的位 9 (TSCTRLSSR) 置 1]，则应编程 20 (0x14)。当 TSCTRLSSR 清零时，纳秒寄存器的分辨率约为 0.465 ns。在这种情况下，应编程值 43 (0x2B)（由 20 ns/0.465 得出）。

位 15:8 **SNSINC[7:0]**：亚纳秒增量值 (Sub-nanosecond Increment Value)

该字段包含亚纳秒增量值，以纳秒数乘以 2^8 进行表示。

该值与亚秒寄存器的亚纳秒字段进行累加。

例如，在 ETH_MACTSCR 寄存器中的 TSCTRLSSR 字段置 1 时，如果所需增量为 5.3 ns，则 SSINC 应为 0x05，SNSINC 应为 0x4C。

位 7:0 保留，必须保持复位值

系统时间秒寄存器 (ETH_MACSTSR)

System time seconds register

偏移地址：0x0B08

复位值：0x0000 0000

系统时间秒寄存器与系统时间纳秒寄存器一起指示由 MAC 保持的系统时间的当前值。虽然该值是连续更新的，但由于存在时钟域传输延迟（从 HCLK 到 CSR 时钟），因此实际时间有一些延迟。

只有在没有外部时间戳输入的情况下选择 IEEE 1588 时间戳功能时，才会存在该寄存器。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TSS[31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TSS[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:0 **TSS[31:0]**：时间戳秒 (Timestamp Second)

此字段中的值表示由 MAC 保持的以秒为单位的系统时间当前值。

系统时间纳秒寄存器 (ETH_MACSTNR)

System time nanoseconds register

偏移地址：0x0B0C

复位值：0x0000 0000

系统时间纳秒寄存器与系统时间秒寄存器一起指示由 MAC 保持的系统时间的当前值。

只有在没有外部时间戳输入的情况下选择 IEEE 1588 时间戳功能时，才会存在该寄存器。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	TSSS[30:16]														
	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TSSS[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31 保留，必须保持复位值

位 30:0 **TSSS[30:0]**：时间戳亚秒 (Timestamp Sub-seconds)

该字段中的值包含亚秒时间表示，精度为 0.46 ns。ETH_MACTSCR 中的位 9 置 1 时，每个位表示 1 ns。最大值为 0x3B9A_C9FF，之后会翻转为零。

系统时间秒更新寄存器 (ETH_MACSTSUR)

System time seconds update register

偏移地址：0x0B10

复位值：0x0000 0000

系统时间秒更新寄存器与系统时间纳秒更新寄存器一起用于初始化或更新 MAC 保持的系统时间。在将 ETH_MACTSCR 寄存器中的 TSINIT 或 TSUPDT 位置 1 前，必须写入这两个寄存器。

只有在没有外部时间戳输入的情况下选择 IEEE 1588 时间戳功能时，才会存在该寄存器。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TSS[31:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TSS[31:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:0 **TSS[31:0]**：时间戳秒 (Timestamp Seconds)

此字段中的值是更新的亚秒部分。ADDSUB 复位时，该字段必须用更新值的亚秒部分进行编程，其精度取决于 ETH_MACTSCR 寄存器的 TSCTRLSSR 位。ADDSUB 置 1 时，该字段必须用更新值亚秒部分的补码进行编程，如下所述。

TSCTRLSSR 置 1 时，编程值必须为 10^9 - <sub-second value>。TSCTRLSSR 复位时，编程值必须为 2^31 - <sub-second_value>。

例如，当 TSCTRLSSR 位置 1 时，如果需要从系统时间中减去 2.000000001 秒，则 MAC_Timestamp 秒更新寄存器中的 TSS 字段必须为 0xFFFF_FFFE（即 2^32 - 2），该寄存器中的 ADDSUB 位应置 1，TSSS 字段必须为 0x3B9A_C9FF（即，10^9 - 1）。

系统时间纳秒更新寄存器 (ETH_MACSTNUR)

System time nanoseconds update register

偏移地址：0x0B14

复位值：0x0000 0000

只有在没有外部时间戳输入的情况下选择 IEEE 1588 时间戳功能时，才会存在该寄存器。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ADDSUB	TSSS[30:16]														
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TSSS[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31 **ADDSUB**：加上或减去时间 (Add or Subtract Time)

该位置 1 时，用更新寄存器的内容减去时间值。该位复位时，时间值与更新寄存器的内容相加。

位 30:0 **TSSS[30:0]**：时间戳亚秒 (Timestamp Sub-seconds)

该字段中的值包含亚秒时间表示，精度为 0.46 ns。ETH_MACTSCR 寄存器中的 TSCTRLSSR 位置 1 时，每个位表示 1 ns，编程的值不应超过 0x3B9A_C9FF。

时间戳加数寄存器 (ETH_MACTSAR)

Timestamp addend register

偏移地址：0x0B18

复位值：0x0000 0000

只有在没有外部时间戳输入的情况下选择 IEEE 1588 时间戳功能时，才会存在时间戳加数寄存器。仅当系统时间配置为精密更新模式（ETH_MACTSCR 中的 TSCFUPDT 位）时，才使用该寄存器值。该寄存器的内容在（HCLK 的）每个时钟周期添加到一个 32 位累加器中，系统时间在该累加器发生上溢时更新。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TSAR[31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TSAR[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:0 **TSAR[31:0]**：时间戳加数寄存器 (Timestamp Addend Register)

该字段指示要添加到累加器寄存器以实现时间同步的 32 位时间值。

时间戳状态寄存器 (ETH_MACTSSR)

Timestamp status register

偏移地址：0x0B20

复位值：0x0000 0000

只有在选择 IEEE 1588 时间戳功能时，才会存在时间戳状态寄存器。当应用程序读取该寄存器时，除位 [27:25] 之外的所有位都将被清零。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	ATSNS[4:0]					ATSSTM	Res.	Res.	Res.	Res.	ATSSTN[3:0]			
		r	r	r	r	r	r					r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TXTSSIS	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TSTRGTERR0	AUXTSTRIG	TSTARGT0	TSSOVF
r												r	r	r	r

位 31:30 保留，必须保持复位值

位 29:25 **ATSNS[4:0]**：辅助时间戳快照数 (Number of Auxiliary Timestamp Snapshots)
该字段指示 FIFO 中可用的快照数。值等于所选 FIFO 的深度（4、8 或 16）即表示辅助快照 FIFO 已满。当辅助快照 FIFO 清除位置 1 时，这些位清零 (00000)。仅当选择了 Add IEEE 1588 Auxiliary Snapshot（添加 IEEE 1588 辅助快照）选项时，该位才有效。

位 24 **ATSSTM**：辅助时间戳快照触发未命中 (Auxiliary Timestamp Snapshot Trigger Missed)
辅助时间戳快照 FIFO 已满并且设置了外部触发时，该位置 1。这表示最新快照未存储在 FIFO 中。仅当选择了 Add IEEE 1588 Auxiliary Snapshot（添加 IEEE 1588 辅助快照）选项时，该位才有效。

位 23:20 保留，必须保持复位值

位 19:16 **ATSSTN[3:0]**：辅助时间戳快照触发标识符 (Auxiliary Timestamp Snapshot Trigger Identifier)
这些位标识辅助快照寄存器中的时间戳所适用的辅助触发输入。多个位同时置 1 时，即表示相应的辅助触发采用相同时钟进行采样。只有辅助快照数超过一个时，这些位才适用。为每个触发分配一个位，如下所述：
位 16：辅助触发 0
位 17：辅助触发 1
位 18：辅助触发 2
位 19：辅助触发 3
软件可以读取该寄存器以找到在获取时间戳时设置的触发。

位 15 **TXTSSIS**：Tx 时间戳状态中断状态 (Tx Timestamp Status Interrupt Status)
如果在 MTL 中使能丢弃发送状态，则在 ETH_MACTxTSSNR 和 ETH_MACTxTSSSR 寄存器中更新捕获的发送时间戳时，该位会置 1。
如果使能 PTP 减荷功能，则在 ETH_MACTxTSSNR 和 ETH_MACTxTSSSR 寄存器中更新捕获的发送时间戳时，该位会置 1，以用于 PTO 生成的延迟请求和 Pdelay 请求数据包。
对 ETH_MACTxTSSSR 寄存器执行读操作时此位清零。
在所有其他配置中，该位保留。

位 14:4 保留，必须保持复位值

位 3 **TSTRGTERR0**：时间戳目标时间错误 (Timestamp Target Time Error)
在 ETH_MACPPS_Target_Time_seconds 和 ETH_MACPPS_Target_Time_Nanoseconds 寄存器中编程的最新目标时间结束后，
该位置 1。应用程序读取该位时会将其清零。

位 2 **AUXTSTRIG**：辅助时间戳触发快照 (Auxiliary Timestamp Trigger Snapshot)
将辅助快照写入 FIFO 时，该位置为高电平。仅当选择了 Add IEEE 1588 Auxiliary Snapshot（添加 IEEE 1588 辅助快照）选项时，该位才有效。

位 1 **TSTARGT0**：达到时间戳目标时间 (Timestamp Target Time Reached)
该位置 1 时，表示系统时间值大于或等于在 ETH_MACPPS_Target_Time_seconds 和 ETH_MACPPS_Target_Time_Nanoseconds 寄存器中指定的值。

位 0 **TSSOVF**：时间戳秒上溢 (Timestamp Seconds Overflow)
该位置 1 时，表示时间戳的秒值（支持版本 2 格式时）已超过 32'hFFFF_FFFF。

Tx 时间戳状态纳秒寄存器 (ETH_MACTxTSSNR)

Tx timestamp status nanoseconds register

偏移地址：0x0B30

复位值：0x0000 0000

该寄存器包含在 Tx 状态被禁止时为发送数据包捕获的时间戳的纳秒部分。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TXTSSMIS	TXTSSLO[30:16]														
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TXTSSLO[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31 **TXTSSMIS**：发送时间戳状态丢失 (Transmit Timestamp Status Missed)

该位置 1 时，表示以下任一项：

– 如果 ETH_MACTSCR 寄存器的 TXTSSTSM 位复位，则会忽略当前数据包的时间戳

– 如果 ETH_MACTSCR 寄存器的 TXTSSTSM 位置 1，则前一数据包的时间戳将被当前数据包的时间戳覆盖。

位 30:0 **TXTSSLO[30:0]**：发送时间戳状态低位 (Transmit Timestamp Status Low)

该字段包含发送数据包捕获的时间戳纳秒字段的 31 位。

Tx 时间戳状态秒寄存器 (ETH_MACTxTSSSR)

Tx timestamp status seconds register

偏移地址：0x0B34

复位值：0x0000 0000

该寄存器包含发送 PTP 数据包时捕获的时间戳（以秒为单位）的高 32 位。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TXTSSHI[31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TXTSSHI[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:0 **TXTSSHI[31:0]**：发送时间戳状态高位 (Transmit Timestamp Status High)

该字段包含发送数据包捕获的时间戳秒字段的低 32 位。

辅助控制寄存器 (ETH_MACACR)

Auxiliary control register

偏移地址：0x0B40

复位值：0x0000 0000

辅助时间戳控制寄存器控制辅助时间戳快照。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ATSEN3	ATSEN2	ATSEN1	ATSEN0	Res.	Res.	Res.	ATSFC
								rw	rw	rw	rw				rw

位 31:8 保留，必须保持复位值

位 7 **ATSEN3**：辅助快照 3 使能 (Auxiliary Snapshot 3 Enable)

该位控制辅助快照触发 3 的捕获。该位置 1 时，会使能 ptp_aux_trig_i[3] 输入上的事件辅助快照。该位复位时，此输入上的事件将被忽略。

在下列条件之一为真时，该位将被保留：

– 配置内核时未选择 Add IEEE 1588 Auxiliary Snapshot（添加 IEEE 1588 辅助快照）选项。

– Number of IEEE 1588 Auxiliary Snapshot Inputs（IEEE 1588 辅助快照输入数）选项中的所选数字小于 4。

位 6 **ATSEN2**：辅助快照 2 使能 (Auxiliary Snapshot 2 Enable)

该位控制辅助快照触发 2 的捕获。该位置 1 时，会使能 ptp_aux_trig_i[2] 输入上的事件辅助快照。该位复位时，此输入上的事件将被忽略。

在下列条件之一为真时，该位将被保留：

– 配置内核时未选择 Add IEEE 1588 Auxiliary Snapshot（添加 IEEE 1588 辅助快照）选项。

– Number of IEEE 1588 Auxiliary Snapshot Inputs（IEEE 1588 辅助快照输入数）选项中的所选数字小于 3。

位 5 **ATSEN1**：辅助快照 1 使能 (Auxiliary Snapshot 1 Enable)

该位控制辅助快照触发 1 的捕获。该位置 1 时，会使能 ptp_aux_trig_i[1] 输入上的事件辅助快照。该位复位时，此输入上的事件将被忽略。

在下列条件之一为真时，该位将被保留：

– 配置内核时未选择 Add IEEE 1588 Auxiliary Snapshot（添加 IEEE 1588 辅助快照）选项。

– Number of IEEE 1588 Auxiliary Snapshot Inputs（IEEE 1588 辅助快照输入数）选项中的所选数字小于 2。

位 4 **ATSEN0**：辅助快照 0 使能 (Auxiliary Snapshot 0 Enable)

该位控制辅助快照触发 0 的捕获。该位置 1 时，会使能 ptp_aux_trig_i[0] 输入上的事件辅助快照。该位复位时，此输入上的事件将被忽略。

如果在配置内核时未选择 Add IEEE 1588 Auxiliary Snapshot（添加 IEEE 1588 辅助快照）选项，则该位保留。

位 3:1 保留，必须保持复位值

位 0 **ATSFC**：辅助快照 FIFO 清除 (Auxiliary Snapshot FIFO Clear)

该位置 1 时，会复位辅助快照 FIFO 的指针。当指针复位并且 FIFO 为空时，该位清零。该位为高电平时，辅助快照存储在 FIFO 中。

如果在配置内核时未选择 Add IEEE 1588 Auxiliary Snapshot（添加 IEEE 1588 辅助快照）选项，则该位保留。

辅助时间戳纳秒寄存器 (ETH_MACATSNR)

Auxiliary timestamp nanoseconds register

偏移地址：0x0B48

复位值：0x0000 0000

辅助时间戳纳秒寄存器与 ETH_MACATSSR 一起提供以辅助快照形式存储的 64 位时间戳。这两个寄存器构成深度为 4 个字的 64 位宽 FIFO 的读端口。

可将多个快照存储在此 FIFO 中。ETH_MACTSSR 中的位 [29:25] 表示 FIFO 的填充级别。仅当读取 MAC 寄存器 91（辅助时间戳 - 秒寄存器）的最后一个字节时，才会将 FIFO 的顶部删除。在小端模式下，这表示在读取位 [31:24] 时；在大端模式下，则表示在读取位 [7:0] 时。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	AUXTSLO[30:16]														
	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AUXTSLO[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31 保留，必须保持复位值

位 30:0 **AUXTSLO[30:0]**：辅助时间戳 (Auxiliary Timestamp)

包含辅助时间戳的低 31 位（纳秒字段）。

辅助时间戳秒寄存器 (ETH_MACATSSR)

Auxiliary timestamp seconds register

偏移地址：0x0B4C

复位值：0x0000 0000

辅助时间戳秒寄存器包含辅助时间戳寄存器秒字段的低 32 位。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
AUXTSHI[31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AUXTSHI[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:0 **AUXTSHI[31:0]**：辅助时间戳 (Auxiliary Timestamp)
包含辅助时间戳秒字段的低 32 位。

时间戳入站不对称校准寄存器 (ETH_MACTSIACR)

Timestamp Ingress asymmetric correction register

偏移地址：0x0B50

复位值：0x0000 0000

MAC 时间戳入站不对称校准寄存器包含要在更新 PDelay_Resp PTP 消息中的校准字段时使用的入站不对称校准值。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
OSTIAC[31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OSTIAC[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:0 **OSTIAC[31:0]**：一步时间戳入站不对称校准 (One-Step Timestamp Ingress Asymmetry Correction)
该字段包含要添加到 Pdelay_Resp PTP 数据包的校准字段的入站路径不对称值。编程值应以纳秒为单位，并乘以 2^16。例如，2.5 ns 表示为 0x00028000。
该值也可以是负数，它以 2 的补码形式表示，位 31 表示符号位。

时间戳出站不对称校准寄存器 (ETH_MACTSEACR)

Timestamp Egress asymmetric correction register

偏移地址：0x0B54

复位值：0x0000 0000

MAC 时间戳出站不对称校准寄存器包含要在更新 PDelay_Req PTP 消息中的校准字段时使用的出站不对称校准值。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
OSTEAC[31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OSTEAC[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:0 **OSTEAC[31:0]**：一步时间戳出站不对称校准 (One-Step Timestamp Egress Asymmetry Correction)

该字段包含要从 Pdelay_Resp PTP 数据包的校准字段中减去的出站路径不对称值。编程值必须为负值（以纳秒为单位），并乘以 2^16。

例如，如果所需校准值为 +2.5 ns，则编程值必须为 0xFFFD_8000，它是 0x0002_8000 (2.5 * 216) 的 2 的补码。类似地，如果所需校准值为 -3.3 ns，则编程值为 0x0003_4CCC (3.3 * 216)。

时间戳入站校准纳秒寄存器 (ETH_MACTSICNR)

Timestamp Ingress correction nanosecond register

偏移地址：0x0B58

复位值：0x0000 0000

该寄存器包含要与入站路径中捕获的时间戳值一起使用的校准值（以纳秒为单位）。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TSIC[31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TSIC[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:0 **TSIC[31:0]**：时间戳入站校准 (Timestamp Ingress Correction)

该字段包含由入站校准表达式定义的入站路径校准值。

时间戳出站校准纳秒寄存器 (ETH_MACTSECNR)

Timestamp Egress correction nanosecond register

偏移地址：0x0B5C

复位值：0x0000 0000

该寄存器包含要与出站路径中捕获的时间戳值一起使用的校准值（以纳秒为单位）。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TSEC[31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TSEC[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:0 **TSEC[31:0]**：时间戳出站校准 (Timestamp Egress Correction)

该字段包含由出站校准表达式定义的出站路径校准值的纳秒部分。

PPS 控制寄存器 (ETH_MACPPSCR)

PPS control register

偏移地址：0x0B70

复位值：0x0000 0000

仅在选择了时间戳功能并且未使能外部时间戳时，才存在 PPS 控制寄存器。

该寄存器的位 [30:24] 仅在选择了四个灵活 PPS 输出时才有效。位 [22:16] 仅在选择了三个或以上灵活 PPS 输出时才有效。位 [14:8] 仅在选择了两个或以上灵活 PPS 输出时才有效。位 [6:4] 仅在选择了灵活 PPS 功能时才有效。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TRGTMODSEL0[1:0]		PPSEN0	PPSCTRL_PPSCMD[3:0]			
									rw	rw	rw	rw	rw	rw	rw

位 31:7 保留，必须保持复位值

位 6:5 **TRGTMODSEL0[1:0]**：PPS 输出的目标时间寄存器模式 (Target Time Register Mode for PPS Output)

该字段指示 PPS 输出信号的目标时间寄存器（MAC 寄存器 96 和 97）模式：

00：仅为生成中断事件而编程目标时间寄存器。

01：保留

10：为生成中断事件以及开始或停止 PPS 输出信号生成而编程目标时间寄存器。

11：仅为开始或停止 PPS 输出信号生成而编程目标时间寄存器。未将任何中断置为有效。

位 4 **PPSEN0**：灵活 PPS 输出模式使能 (Flexible PPS Output Mode Enable)

该位置 1 时，位 [3:0] 用作 PPSCMD。该位复位时，位 [3:0] 用作 PPSCTRL（固定 PPS 模式）。

位 3:0 **PPSCTRL[3:0]**：PPS 输出频率控制 (PPS Output Frequency Control)

该字段控制 PPS 输出 (ptp_pps_o) 信号的频率。PPSCTRL 的默认值为 0000，PPS 输出为每秒 1 个脉冲（宽度为 clk_ptp_i）。当 PPSCTRL 为其他值时，PPS 输出则为所生成的以下频率的时钟：

0001：二进制翻转为 2 Hz，数字翻转为 1 Hz。

0010：二进制翻转为 4 Hz，数字翻转为 2 Hz。

0011：二进制翻转为 8 Hz，数字翻转为 4 Hz。

0100：二进制翻转为 16 Hz，数字翻转为 8 Hz。

..

1111：二进制翻转为 32.768 KHz，数字翻转为 16.384 KHz。

注：　对于这些频率，在二进制翻转模式下，PPS 输出 (ptp_pps_o) 的占空比为 50%。在数字翻转模式下，PPS 输出频率为平均数。实际时钟的频率不同，每秒同步一次。例如：

- *PPSCTRL = 0001 时，PPS (1 Hz) 具有 537 ms 的低电平周期和 463 ms 的高电平周期*
- *PPSCTRL = 0010 时，PPS (2 Hz) 为如下序列：*
  第一个时钟的占空比为 50%，周期为 537 ms
  第二个时钟的周期为 463 ms（268 ms 低电平，195 ms 高电平）
- *PPSCTRL = 0011 时，PPS (4 Hz) 为如下序列：*
  前三个时钟的占空比为 50%，周期为 268 ms
  第四个时钟的周期为 195 ms（134 ms 低电平，61 ms 高电平）

此行为是因为在 ETH_MACSTNR 寄存器中于数字翻转模式下对位进行非线性翻转而引起。

位 3:0 或

PPSCMD[3:0]：灵活 PPS 输出 (ptp_pps_o[0]) 控制 (Flexible PPS Output (ptp_pps_o[0]) Control)
为这些位编程非零值将命令 MAC 发起事件。当命令被传送或同步到 PTP 时钟域时，这些位将自动清零。软件应确保仅在这些位为“全零”时才进行编程。下面对 PPSCMD0 的值进行了说明：
0000：无命令
0001：启动单脉冲
该命令会生成单脉冲，该脉冲在目标时间寄存器（寄存器 455 和 456）中定义的起始点上升，持续时间为 PPS 宽度寄存器中所定义。
0010：启动脉冲串
该命令会生成脉冲串，该脉冲串在目标时间寄存器中定义的起始点上升，持续时间为 PPS 宽度寄存器中所定义，并以 PPS 间隔寄存器中定义的间隔进行重复。除非通过“经过一段时间后停止脉冲串”或“立即停止脉冲串”命令停止 PPS 脉冲串，否则默认情况下，PPS 脉冲串自由运行。
0011：取消启动
如果系统时间未超过编程的启动时间，该命令将取消启动单脉冲和启动脉冲串命令。
0100：经过一段时间后停止脉冲串
目标时间寄存器中编程的时间结束后，该命令将停止由启动脉冲串命令 (PPSCMD = 0010) 启动的脉冲串。
0101：立即停止脉冲串
该命令会立即停止由启动脉冲串命令 (PPSCMD = 0010) 启动的脉冲串。
0110：取消停止脉冲串
如果编程的停止时间尚未结束，该命令将取消经过一段时间后停止脉冲串命令。PPS 脉冲串在该命令成功执行后将自由运行。
0111-1111：保留

PPS 目标时间秒寄存器 (ETH_MACPPSTTSR)

PPS target time seconds register

偏移地址：0x0B80

复位值：0x0000 0000

PPS 目标时间秒寄存器与 PPS 目标时间纳秒寄存器一起，用于在系统时间超过这些寄存器中编程的值时规划中断事件 [ETH_MACTSSR 的位 1]。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TSTRH0[31:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TSTRH0[31:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:0 **TSTRH0[31:0]**：PPS 目标时间秒寄存器 (PPS Target Time Seconds Register)
该字段存储以秒为单位的时间。当时间戳值匹配或超过两个目标时间戳寄存器时，MAC 会启动或停止 PPS 信号输出，并根据在 ETH_MACPPSCR 寄存器中为相应 PPS 输出选择的目标时间模式生成中断（如果使能）。

PPS 目标时间纳秒寄存器 (ETH_MACPPSTTNR)

PPS target time nanoseconds register

偏移地址：0x0B84

复位值：0x0000 0000

仅在选择多个灵活 PPS 输出时，才存在 PPS 目标时间纳秒寄存器。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TRGTBUSY0	TTSL0[30:16]														
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TTSL0[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31 **TRGTBUSY0**：PPS 目标时间寄存器忙 (PPS Target Time Register Busy)

将 ETH_MACPPSCR 寄存器中的 PPSCMD0 字段编程为 010 或 011 时，MAC 会将该位置 1。将 PPSCMD0 字段编程为 010 或 011 时，会命令 MAC 将目标时间寄存器同步到 PTP 时钟域。在将目标时间寄存器同步到 PTP 时钟域之后，MAC 会将该位清零。当该位读为 1 时，应用程序不得更新目标时间寄存器。否则，先前编程的时间同步会损坏。

位 30:0 **TTSL0[30:0]**：PPS 的目标时间低位寄存器 (Target Time Low for PPS Register)

该寄存器存储以纳秒为单位的时间（带符号）。当时间戳的值与两个目标时间戳寄存器中的值匹配时，MAC 将启动或停止 PPS 信号输出，并根据 ETH_MACPPSCR 中的 TRGTMODSEL0 字段（位 [6:5]）生成中断（如果使能）。

ETH_MACTSCR 寄存器中的 TSCTRLSSR 位置 1 时，该值不应超过 0x3B9A_C9FF。PPS 信号输出的实际启动或停止时间最高可能具有一个单位的亚秒增量值的误差。

PPS 间隔寄存器 (ETH_MACPPSIR)

PPS interval register

偏移地址：0x0B88

复位值：0x0000 0000

PPS 间隔寄存器包含 PPS 信号输出 (ptp_pps_o[0]) 的上升沿之间的亚秒增量值的单位数。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PPSINT0[31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PPSINT0[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:0 **PPSINT0[31:0]**：PPS 输出信号间隔 (PPS Output Signal Interval)

这些位存储 PPS 信号输出的上升沿之间的间隔。间隔以亚秒增量值的单位数进行存储。

需要编程一个小于所需间隔的值。例如，如果 PTP 参考时钟为 50 MHz（周期为 20 ns），并且 PPS 信号输出的上升沿之间的预期间隔为 100 ns（即，5 个单位的亚秒增量值），则应在该寄存器中编程值 4 (5-1)。

PPS 宽度寄存器 (ETH_MACPPSWR)

PPS width register

偏移地址：0x0B8C

复位值：0x0000 0000

PPS 宽度寄存器包含 PPS 信号输出 (ptp_pps_o) 的上升沿和相应下降沿之间的亚秒增量值的单位数。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PPSWIDTH0[31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PPSWIDTH0[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:0 **PPSWIDTH0[31:0]**：PPS 输出信号宽度 (PPS Output Signal Width)

这些位存储 PPS 信号输出的上升沿和相应下降沿之间的宽度。宽度以亚秒增量值的单位数进行存储。

需要编程一个小于所需间隔的值。例如，如果 PTP 参考时钟为 50 MHz（周期为 20 ns），并且 PPS 信号输出的上升沿和相应下降沿之间的宽度为 80 ns（即，4 个单位的亚秒增量值），则应在该寄存器中编程值 3 (4-1)。

注： 该寄存器中编程的值必须小于 ETH_MACPP0IR 寄存器中编程的值。

PTP 减荷控制寄存器 (ETH_MACPOCR)

PTP Offload control register

偏移地址：0x0BC0

复位值：0x0000 0000

该寄存器控制 PTP 减荷引擎操作。仅在选择 Enable PTP Timestamp Offload（使能 PTP 时间戳减荷）功能时，该寄存器才可用。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DN[7:0]								PDRDIS	DRRDIS	APDREQTRIG	ASYNCTRIG	Res.	APDREQEN	ASYNCEN	PTOEN
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw		rw	rw	rw

位 31:16 保留，必须保持复位值

位 15:8 **DN[7:0]**：域号 (Domain Number)
该字段指示 PTP 节点正在使用的域号。

位 7 保留，必须保持复位值

位 6 **DRRDIS**：禁止 PTO 延迟请求/响应生成 (Disable PTO Delay Request/Response response generation)
如果该位置 1，则会按照所编程模式的要求，分别在接收到 SYNC 和延迟请求数据包时不生成延迟请求和延迟响应。

位 5 **APDREQTRIG**：自动 PTP Pdelay_Req 消息触发 (Automatic PTP Pdelay_Req message Trigger)
该位置 1 时，会发送一条 PTP Pdelay_Req 消息。发送 PTP Pdelay_Req 消息后，该位自动清零。为实现此操作，应用程序应将 APDREQEN 位置 1。

位 4 **ASYNCTRIG**：自动 PTP SYNC 消息触发 (Automatic PTP SYNC message Trigger)
该位置 1 时，会发送一条 PTP SYNC 消息。发送 PTP SYNC 消息后，该位自动清零。为实现此操作，应用程序应将 ASYNCEN 位置 1。

位 3 保留，必须保持复位值

位 2 **APDREQEN**：自动 PTP Pdelay_Req 消息使能 (Automatic PTP Pdelay_Req message Enable)
该位置 1 时，如果将 MAC 编程为点对点透明模式，则会基于编程设定的间隔或在应用程序的触发下周期性地生成 PTP Pdelay_Req 消息。

位 1 **ASYNCEN**：自动 PTP SYNC 消息使能 (Automatic PTP SYNC message Enable)
该位置 1 时，如果将 MAC 编程为时钟主模式，则会基于编程设定的间隔或在应用程序的触发下周期性地生成 PTP SYNC 消息。

位 0 **PTOEN**：PTP 减荷使能 (PTP Offload Enable)
此位置 1 时，会使能 PTP 减荷功能。

PTP 源端口标识 0 寄存器 (ETH_MACSPI0R)

PTP Source Port Identity 0 Register

偏移地址：0x0BC4

复位值：0x0000 0000

该寄存器包含 PTP 节点的 80 位源端口标识的位 [31:0]。仅在选择 Enable PTP Timestamp Offload（使能 PTP 时间戳减荷）功能时，该寄存器才可用。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SPI0[31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SPI0[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:0 **SPI0[31:0]**：源端口标识 0 (Source Port Identity 0)
该字段指示 PTP 节点的源端口标识的位 [31:0]。

PTP 源端口标识 1 寄存器 (ETH_MACSPI1R)

PTP Source port identity 1 register

偏移地址：0x0BC8

复位值：0x0000 0000

该寄存器包含 PTP 节点的 80 位源端口标识的位 [63:32]。仅在选择 Enable PTP Timestamp Offload（使能 PTP 时间戳减荷）功能时，该寄存器才可用。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SPI1[31:16]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SPI1[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:0 **SPI1[31:0]**：源端口标识 1 (Source Port Identity 1)
该字段指示 PTP 节点的源端口标识的位 [63:32]。

PTP 源端口标识 2 寄存器 (ETH_MACSPI2R)

PTP Source port identity 2 register

偏移地址：0x0BCC

复位值：0x0000 0000

该寄存器包含 PTP 节点的 80 位源端口标识的位 [79:64]。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SPI2[15:0]															
rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:16 保留，必须保持复位值

位 15:0 **SPI2[15:0]**：源端口标识 2 (Source Port Identity 2)
该字段指示 PTP 节点的源端口标识的位 [79:64]。

日志消息间隔寄存器 (ETH_MACLMIR)

Log message interval register

偏移地址：0x0BD0

复位值：0x0000 0000

该寄存器包含用于自动生成 PTP 数据包的周期性间隔。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LMPDRI[7:0]								Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
rw	rw	rw	rw	rw	rw	rw	rw								
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	DRSYNCR[2:0]			LSI[7:0]							
					rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:24 **LMPDRI[7:0]**：日志最小 Pdelay_Req 间隔 (Log Min Pdelay_Req Interval)

该字段指示 PTP 节点的 logMinPdelayReqInterval。它用于调度周期性的 Pdelay 请求数据包发送。允许值为 -15 到 15。负值必须以 2 的补码形式表示。例如，如果所需值为 -1，则编程的值必须为 0xFF。

位 23:11 保留，必须保持复位值

位 10:8 **DRSYNCR[2:0]**：

Delay_Req 与 SYNC 之比 (Delay_Req to SYNC Ratio)

在从模式下，它用于控制发送的 Delay_Req 消息的频率。

0：每次接收到 SYNC 都会生成一条 DelayReq 消息

1：每接收到 2 个 SYNC 会生成一条 DelayReq 消息

2：每接收到 4 个 SYNC 会生成一条 DelayReq 消息

3：每接收到 8 个 SYNC 会生成一条 DelayReq 消息

4：每接收到 16 个 SYNC 会生成一条 DelayReq 消息

5：每接收到 32 个 SYNC 会生成一条 DelayReq 消息

6-7：保留

主节点将 DelayResp PTP 消息中的此信息 (logMinDelayReqInterval) 发送到从节点。接收端会对接收到的 DelayResp 消息中的此值进行处理，并相应地更新此字段。在从模式下，主机不得对该寄存器进行写/更新操作，除非其必须覆盖接收到的值。在主模式下，该字段与 logSyncInterval (LSI) 字段的总和在生成的多播 Delay_Resp PTP 消息的 logMinDelayReqInterval 字段中提供。

位 7:0 **LSI[7:0]**：

日志同步间隔 (Log Sync Interval)

该字段指示 PTP 节点为主节点时自动生成 SYNC 消息的周期。允许值为 -15 到 15。负值必须以 2 的补码形式表示。例如，如果所需值为 -1，则编程的值必须为 0xFF。

以太网 MAC 寄存器映射和复位值

表 545. 以太网 MAC 寄存器映射和复位值

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x0000	ETH_MACCR	ARPEN	SARC [2:0]			IPC	IPG[2:0]			GPSLCE	S2KP	CST	ACS	WD	Res.	JD	JE	Res.	FES	DM	LM	ECRSFD	DO	DCRS	DR	Res.	BL [1:0]		DC	PRELEN[1:0]		TE	RE
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0		0	0	0	0	0	0	0	0	0	0		0	0	0	0	0	0	0
0x0004	ETH_MACECR	Res.	Res.	EIPG[4:0]					EIPGEN	Res.	Res.	Res.	Res.	Res.	USP	SPEN	DCRCC	Res.	Res.	GPSL[13:0]														
	Reset value			0	0	0	0	0	0						0	0	0			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0008	ETH_MACPFR	RA	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DNTU	IPFE	Res.	Res.	Res.	VTFE	Res.	Res.	Res.	Res.	Res.	HPF	SAF	SAIF	PCF[1:0]		DBF	PM	DAIF	HMC	HUC	PR
	Reset value	0										0	0				0						0	0	0	0	0	0	0	0	0	0	0
0x000C	ETH_MACWTR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PWE	Res.	Res.	Res.	Res.	WTO[3:0]			
	Reset value																								0					0	0	0	0
0x0010	ETH_MACHT0R	HT31T0[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0014	ETH_MACHT1R	HT63T32[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0018 - 0x004C	Reserved																																
0x0050	ETH_MACVTR	EIVLRXS	Res.	EIVLS[1:0]		ERIVLT	EDVLP	VTHM	EVLRXS	Res.	EVLS[1:0]		DOVLTC	ERSVLM	ESVL	VTIM	ETV	VL[15:0]																
	Reset value	0		0	0	0	0	0	0		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0054	Reserved																																
0x0058	ETH_MACVHTR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	VLHT[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x005C	Reserved																																
0x0060	ETH_MACVIR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	VLTI	CSVL	VLP	VLC [1:0]		VLT[15:0]															
	Reset value												0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0064	ETH_MACIVIR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	VLTI	CSVL	VLP	VLC [1:0]		VLT[15:0]															
	Reset value												0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0068 - 0x006C	Reserved																																
0x0070	ETH_ MACQTxFCR	PT[15:0]																Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DZPQ	PLT[2:0]			Res.	Res.	TFE	FCB_BPA
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0								0	0	0	0			0	0
0x0074 - 0x008C	Reserved																																

表 545. 以太网 MAC 寄存器映射和复位值（续）

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x0090	ETH_MACRxFCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	UP	RFE
	Reset value																															0	0
0x0094	Reserved																																
0x00A0 - 0x00A8	Reserved																																
0x00AC	Reserved																																
0x00B0	ETH_MACISR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RXSTSIS	TXSTSIS	TSIS	Res.	MMCTXIS	MMCRXIS	MMCIS	Res.	Res.	LPIIS	PMTIS	PHYIS	Res.	Res.	Res. S
	Reset value																		0	0	0		0	0	0			0	0	0			
0x00B4	ETH_MACIER	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RXSTSIE	TXSTSIE	TSIE	Res.	Res.	Res.	Res.	Res.	Res.	LPIIE	PMTIE	PHYIE	Res.	Res.	Res.
	Reset value																		0	0	0							0	0	0			0
0x00B8	ETH_ MACRXTXSR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RWT	Res.	Res.	EXCOL	LCOL	EXDEF	LCARR	NCARR	TJT
	Reset value																								0			0	0	0	0	0	
0x00BC	Reserved																																
0x00C0	ETH_MACPCSR	RWKFILTRST	Res.	Res.	RWKPTR[4:0]					Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RWKPFE	GLBLUCAST	Res.	Res.	RWKPRCVD	MGKPRCVD	Res.	Res.	RWKPKTEN	MGKPKTEN	PWRDWN
	Reset value	0			0	0	0	0	0														0	0			0	0			0	0	0
0x00C4	ETH_ MACRWKPFR	WKUPFRMFTR[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x00C8 - 0x00CC	Reserved																																
0x00D0	ETH_MACLCSR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LPITCSE	LPIATE	LPITXA	PLSEN	PLS	LPIEN	Res.	Res.	Res.	Res.	Res.	Res.	RLPIST	TLPIST	Res.	Res.	Res.	Res.	RLPIEX	RLPIEN	TLPIEX	TLPIEN
	Reset value											0	0	0	0	0	0							0	0					0	0	0	0
0x00D4	ETH_MACLTCR	Res.	Res.	Res.	Res.	Res.	Res.	LST[9:0]											TWT[15:0]														
	Reset value							1	1	1	1	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
								0x03E8											0														
0x00D8	ETH_MACLETR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LPIET[16:0]																	Res.	Res.	Res.
	Reset value													0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
0x00DC	ETH_ MAC1USTCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TIC_1US_CNTR[11:0]											
	Reset value																					0	0	0	0	0	0	0	0	0	0	0	0
0x00E0 - 0x00F8	Reserved																																

表 545. 以太网 MAC 寄存器映射和复位值（续）

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00FC - 0x010C	Reserved																																
0x0110	ETH_MACVR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	USERVER[7:0]								SNPSVER[7:0]							
	Reset value																	0	0	1	1	0	0	0	0	0	1	0	0	0	0	0	1
0x0114	ETH_MACDR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TFCSTS[1:0]		TPESTS	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RFCFCSTS[1:0]		RPESTS
	Reset value														0	0	0														0	0	0
0x0118	Reserved																																
0x0120	ETH_MACHWF1R	Res.	L3L4FNUM[3:0]				Res.	HASHTBLSZ[1:0]		POUOST	Res.	RAVSEL	AVSEL	DBGMEMA	TSOEN	SPHEN	DCBEN	ADDR64[1:0]		ADVTHWORD	PTOEN	OSTEN	TXFIFOSIZE[4:0]					SPRAM		RXFIFOSIZE[4:0]			
	Reset value		0	0	1	0		0	1	1		0	0	0	1	0	0	0	0	0	1	1	0	0	1	0	0	0	0	0	1	0	0
0x0124	ETH_MACHWF2R	Res.	AUXSNAPNUM[2:0]			Res.	PPSOUTNUM[2:0]			Res.	Res.	TXCHCNT[3:0]				Res.	Res.	RXCHCNT[3:0]				Res.	Res.	TXQCNT[3:0]				Res.	Res.	RXQCNT[3:0]			
	Reset value		1	0	0		0	0	1			0	0	0	0			0	0	0	0			0	0	0	0			0	0	0	0
0x012C - 0x01FC	Reserved																																
0x0200	ETH_MACMDIOAR	Res.	Res.	Res.	Res.	PSE	BTB	PA[4:0]					RDA[4:0]					Res.	NTC[2:0]			CR[3:0]				Res.	Res.	Res.	SKAP	GOC_1	GOC_0	C45E	MB
	Reset value					0	0	0	0	0	0	0	0	0	0	0	0		0	0	0	0	0	0	0				0	0	0	0	0
0x0204	ETH_ MACMDIODR	RA[15:0]																MD[15:0]															
	Reset value		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0208 - 0x020C	Reserved																																
0x0AE0	ETH_MACARPAR	ARPPA[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0214 - 0x02FC	Reserved																																
0x0300	ETH_MACA0HR	AE	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ADDRHI[15:0]															
	Reset value	1																1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0x0304	ETH_MACA0LR	ADDRLO[31:0]																															
	Reset value	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

表 545. 以太网 MAC 寄存器映射和复位值（续）

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x0308	ETH_MACA1HR	AE	SA	MBC[5:0]						Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ADDRHI[15:0]															
	Reset value	0	0	0	0	0	0	0	0									1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0x30C	ETH_MACA1LR	ADDRLO[31:0]																															
	Reset value	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0x0310	ETH_MACA2HR	AE	SA	MBC[5:0]						Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ADDRHI[15:0]															
	Reset value	0	0	0	0	0	0	0	0									1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0x0314	ETH_MACA2LR	ADDRLO[31:0]																															
	Reset value	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0x0318	ETH_MACA3HR	AE	SA	MBC[5:0]						Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ADDRHI[15:0]															
	Reset value	0	0	0	0	0	0	0	0									1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0x031C	ETH_MACA3LR	ADDRLO[31:0]																															
	Reset value	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0x0320 - 0x06FC	Reserved																																
0x0700	MMC_CONTROL	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	UCDBC	Res.	Res.	CNTPRSTLVL	CNTPRST	CNTFREEZ	RSTONRD	CNTSTOPRO	CNTRST
	Reset value																								0			0	0	0	0	0	0
0x0704	MMC_RX_ INTERRUPT	Res.	Res.	Res.	Res.	RXLPITRCIS	RXLPIUSCIS	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RXUCGPIS	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RXALGNERPIS	RXCRCERPIS	Res.	Res.	Res.	Res.	Res.
	Reset value					0	0									0											0	0					
0x0708	MMC_TX_ INTERRUPT	Res.	Res.	Res.	Res.	TXLPITRCIS	TXLPIUSCIS	Res.	Res.	Res.	Res.	TXGPKTIS	Res.	Res.	Res.	Res.	Res.	TXMCOLGPIS	TXSCOLGPIS	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	Reset value					0	0					0						0	0														
0x070C	MMC_RX_ INTERRUPT_ MASK	Res.	Res.	Res.	Res.	RXLPITRCIM	RXLPIUSCIM	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RXUCGPIM	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RXALGNERPIM	RXCRCERPIM	Res.	Res.	Res.	Res.	Res.
	Reset value					0	0									0											0	0					
0x0710	MMC_TX_ INTERRUPT_ MASK	Res.	Res.	Res.	Res.	TXLPITRCIM	TXLPIUSCIM	Res.	Res.	Res.	Res.	TXGPKTIM	Res.	Res.	Res.	Res.	Res.	TXMCOLGPIM	TXSCOLGPIM	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	Reset value					0	0					0						0	0														

表 545. 以太网 MAC 寄存器映射和复位值（续）

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x0714 - 0x0748	Reserved																																
0x074C	TX_SINGLE_COLLISION_GOOD_PACKETS	TXSNGLCOLG[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0750	TX_MULTIPLE_COLLISION_GOOD_PACKETS	TXMULTCOLG[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0754 - 0x0764	Reserved																																
0x0768	TX_PACKET_COUNT_GOOD	TXPKTG[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x076C - 0x0790	Reserved																																
0x0794	RX_CRC_ERROR_PACKETS	RXCRCERR[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0798	RX_ALIGNMENT_ERROR_PACKETS	RXALGNERR[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x079C - 0x07C0	Reserved																																
0x07C4	RX_UNICAST_PACKETS_GOOD	RXUCASTG[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x07C8 - 0x07E8	Reserved																																
0x07EC	TX_LPI_USEC_CNTR	TXLPIUSC[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x07F0	TX_LPI_TRAN_CNTR	TXLPITRC[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x07F4	RX_LPI_USEC_CNTR	RXLPIUSC[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x07F8	RX_LPI_TRAN_CNTR	RXLPITRC[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x07FC - 0x08FC	Reserved																																

表 545. 以太网 MAC 寄存器映射和复位值（续）

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x0900	ETH_ MACL3L4C0R	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	L4DPIM0	L4DPM0	L4SPIM0	L4SPM0	Res.	L4PEN0	L3HDBM0 [4:0]					L3HSBM0 [4:0]					L3DAIM0	L3DAM0	L3SAIM0	L3SAM0	Res.	L3PEN0
	Reset value											0	0	0	0		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		0
0x0904	ETH_MACL4A0R	L4DP0[15:0]																L4SP0[15:0]															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0908 - 0x090C	Reserved																																
0x0910	ETH_MACL3A00R	L3A00[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0914	ETH_MACL3A10R	L3A10[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0918	ETH_MACL3A20R	L3A20[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x091C	ETH_MACL3A30R	L3A30[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0920 - 0x092C	Reserved																																
0x0930	ETH_MACL3L4C1R	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	L4DPIM1	L4DPM1	L4SPIM1	L4SPM1	Res.	L4PEN1	L3HDBM1 [4:0]					L3HSBM 1[4:0]					L3DAIM1	L3DAM1	L3SAIM1	L3SAM1	Res.	L3PEN1
	Reset value											0	0	0	0		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0		0
0x0934	ETH_MACL4A1R	L4DP1[15:0]																L4SP1[15:0]															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0938 - 0x093C	Reserved																																
0x0940	ETH_MACL3A01R	L3A01[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0944	ETH_MACL3A11R	L3A11[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0948	ETH_MACL3A21R	L3A21[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x094C	ETH_MACL3A31R	L3A31[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0950 - 0x0AFC	Reserved																																
0x0B00	ETH_MACTSCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TXTSSTSM	Res.	Res.	Res.	Res.	CSC	TSENMACADDR	SNAPTYPSEL[1:0]		TSMSTRENA	TSEVNTENA	TSIPV4ENA	TSIPV6ENA	TSIPENA	TSVER2ENA	TSCTRLSSR	TSENALL	Res.	Res.	TSADDREG	Res.	TSUPDT	TSINIT	TSCFUPDT	TSENA
	Reset value				0				0					0	0	0	0	0	0	1	0	0	0	0	0			0		0	0	0	0

表 545. 以太网 MAC 寄存器映射和复位值（续）

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x0B04	ETH_MACSSIR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SSINC[7:0]								RESERVED_SNSINC[7:0]								Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	Reset value									0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0								
0x0B08	ETH_MACSTSR	TSS[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0B0C	ETH_MACSTNR	Res.	TSSS[30:0]																														
	Reset value		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0B10	ETH_MACSTSUR	TSS[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0B14	ETH_MACSTNUR	ADDSUB	TSSS[30:0]																														
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0B18	ETH_MACTSAR	TSAR[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0B1C	Reserved																																
0x0B20	ETH_MACTSSR	Res.	Res.	ATSNS[4:0]					ATSSTM	Res.	Res.	Res.	Res.	ATSSTN[3:0]				TXTSSIS	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TSTRGTERR0	AUXTSTRIG	TSTARGT0	TSSOVF
	Reset value			0	0	0	0	0	0					0	0	0	0	0												0	0	0	0
0x0B24 - 0x0B2C	Reserved																																
0x0B30	ETH_MACTxTSSNR	TXTSSMIS	TXTSSLO[30:0]																														
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0B34	ETH_MACTxTSSSR	TXTSSHI[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0B38 - 0x0B3C	Reserved																																
0x0B40	ETH_MACACR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ATSEN3	ATSEN2	ATSEN1	ATSEN0	Res.	Res.	Res.	ATSFC
	Reset value																									0	0	0	0				0
0x0B44	Reserved																																

表 545. 以太网 MAC 寄存器映射和复位值（续）

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x0B48	ETH_MACATSNR	Res.	AUXTSLO[30:0]																														
	Reset value		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0B4C	ETH_MACATSSR	AUXTSHI[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0B50	ETH_MACTSIACR	OSTIAC[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0B54	ETH_MACTSEACR	OSTEAC[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0B58	ETH_ MACTSICNR	TSIC[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0B5C	ETH_ MACTSECNR	TSEC[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0B60 - 0x0B6C	Reserved																																
0x0B70	ETH_MACPPSCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TRGTMODSEL0[1:0]		PPSEN0	PPSCTRL_PPSCMD[3:0]			
	Reset value																										0	0	0	0	0	0	0
0x0B74 - 0x0B7C	Reserved																																
0x0B80	ETH_ MACPPSTTSR	TSTRH0[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0B84	ETH_ MACPPSTTNR	TRGTBUSY0	TTSL0[30:0]																														
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0B88	ETH_MACPPSIR	PPSINT0[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0B8C	ETH_MACPPSWR	PPSWIDTH0[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0B90 - 0x0BBC	Reserved																																

表 545. 以太网 MAC 寄存器映射和复位值（续）

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x0BC0	ETH_MACPOCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DN[7:0]								PDRDIS	DRRDIS	APDREQTRIG	ASYNCTRIG	Res.	APDREQEN	ASYNCEN	PTOEN
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0		0	0	0
0x0BC4	ETH_MACSPI0R	SPI0[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0BC8	ETH_MACSPI1R	SPI1[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0BCC	ETH_MACSPI2R	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SPI2[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x0BD0	ETH_MACLMIR	LMPDRI[7:0]								Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DRSYNCR[2:0]			LSI[7:0]							
	Reset value	0	0	0	0	0	0	0	0														0	0	0	0	0	0	0	0	0	0	0

有关寄存器边界地址的信息，请参见*第 100 页的第 2.2.2 节*。

59 HDMI-CEC 控制器 (HDMI-CEC)

59.1 简介

消费性电子产品控制 (CEC) 是 HDMI（高清多媒体接口）标准的一部分，作为该标准的 *附录 1*。它包含为各种视听产品提供高级控制功能的协议。CEC 可在低速下工作，具有极低的处理和存储开销。

HDMI-CEC 控制器为此协议提供硬件支持。

59.2 HDMI-CEC 控制器主要特性

- 符合 HDMI-CEC v1.4 规范
- 独立 32 kHz CEC 内核（请参见 *RCC 内核时钟分配部分*）
- 针对超低功耗应用可在停止模式下工作
- 开始传输前的空闲信号时间可配置
  - 通过硬件配置时，根据 CEC 状态和发送历史自动完成
  - 通过软件配置时，设置为固定值（7 个时序选项）
- 可配置外设地址 (OAR)
- 支持监听模式
  - 在不干涉 CEC 线路的情况下，能接收发送到 OAR 以外的目标地址的 CEC 消息
- 可配置的接收容差裕量
  - 标准容差
  - 扩展容差
- 接收错误检测
  - 位上升错误 (BRE)，可选择停止接收 (BRESTP)
  - 短位周期错误 (SBPE)
  - 长位周期错误 (LBPE)
- 可配置错误位生成
  - BRE 检测时 (BREGEN)
  - LBPE 检测时 (LBPEGEN)
  - 始终在 SBPE 检测时生成
- 发送错误检测 (TXERR)
- 仲裁丢失检测 (ARBLST)
  - 自动发送重试
- 发送下溢检测 (TXUDR)
- 接收上溢检测 (RXOVR)

59.3 HDMI-CEC 功能说明

59.3.1 HDMI-CEC 引脚和内核信号

CEC 总线是一根双向线，通过它实现数据的输入输出。通过 27 kΩ 上拉电阻连接到 +3.3 V 电源电压。器件的输出级必须为漏极开路或集电极开路，以便进行线“与”连接。

HDMI-CEC 控制器将 CEC 双向线作为标准 GPIO 的复用功能来管理，假设其配置为复用功能开漏输出。必须从外部为 STM32 添加 27 kΩ 上拉电阻。

要想在移除应用电源时不影响 CEC 总线，必须在此类情况下将 CEC 引脚与总线隔离。可通过 MOS 晶体管来实现此操作，如*图 799* 所示。

表 547 列出了 HDMI-CEC 和其他功能块（例如 RCC 和 EXTI）之间交换的内部信号。

表 546. HDMI 引脚

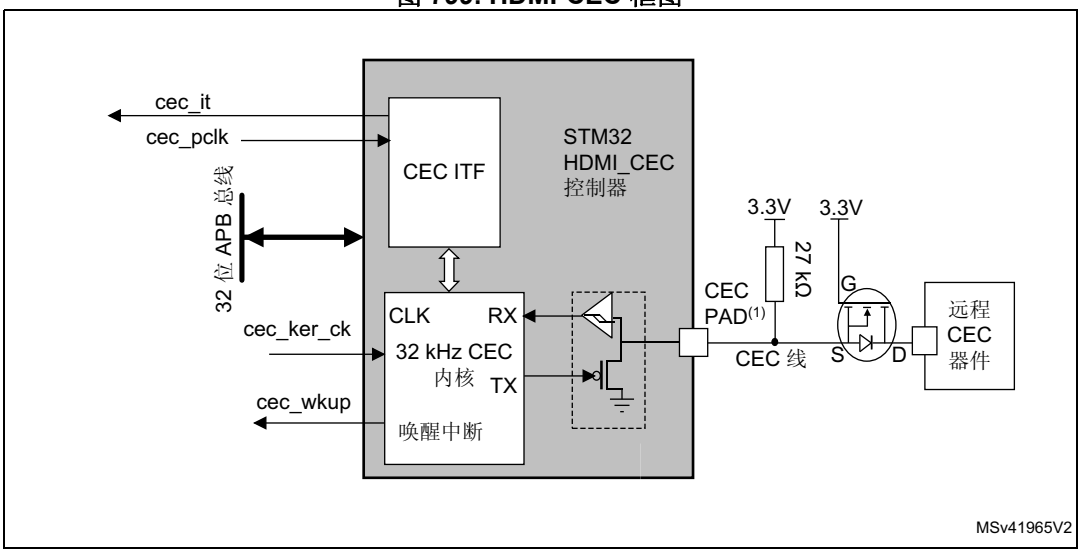
名称	信号类型	备注
CEC	双向	两种状态： 1 = 高阻抗 0 = 低阻抗 必须从外部添加 27 kΩ 电阻。

表 547. HDMI-CEC 内部输入/输出信号

信号名称	信号类型	说明
cec_wkup	数字输出	HDMI-CEC 唤醒信号
cec_it	数字输出	HDMI-CEC 中断信号
cec_pclk	数字输入	APB 时钟
cec_ker_ck	数字输入	HDMI-CEC 内核时钟

59.3.2 HDMI-CEC 框图

图 799. HDMI-CEC 框图



59.3.3 消息说明

CEC 线上的所有活动均包含一个发起方和一个或多个接收方。发起方负责发送消息和数据。跟随方为任意数据的接收方，负责设置任意应答位。

消息以单个帧的形式传送，帧中包含一个起始位、后跟一个报头块、一个可选的操作码和多个操作数。

这些块均由 8 位有效负载组成，首先发送最高有效位，接着发送消息结束 (EOM) 位和应答 (ACK) 位。

EOM 位在消息的最后一个块中置 1，在其它所有块中保持复位。如果指示 EOM 后消息中仍包含额外的块，则应忽略这些额外的块。可在报头块中将 EOM 位置 1 以向其它器件发送“ping”，确保这些器件已激活。

应答位始终由发起方设置为高阻态，这样一来，应答位便可由地址被包含在报头段的接收方或需要拒绝广播消息的接收方驱动为低电平。

报头包含源逻辑地址字段和目标逻辑地址字段。请注意，特定地址 0xF 用于广播消息。

图 800. 消息结构

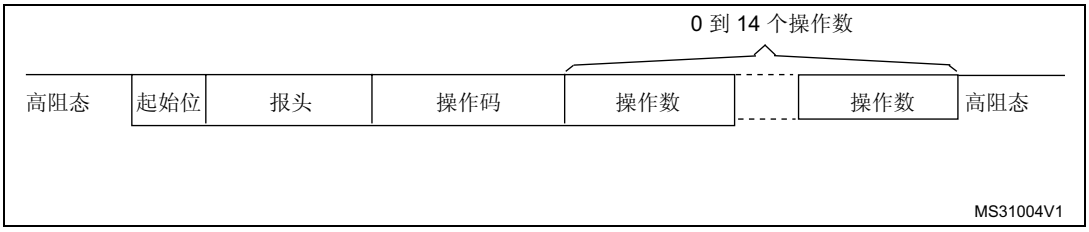
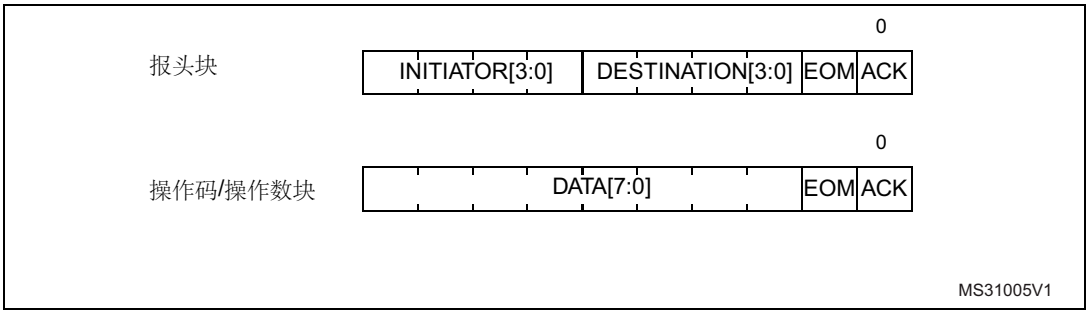


图 801. 块

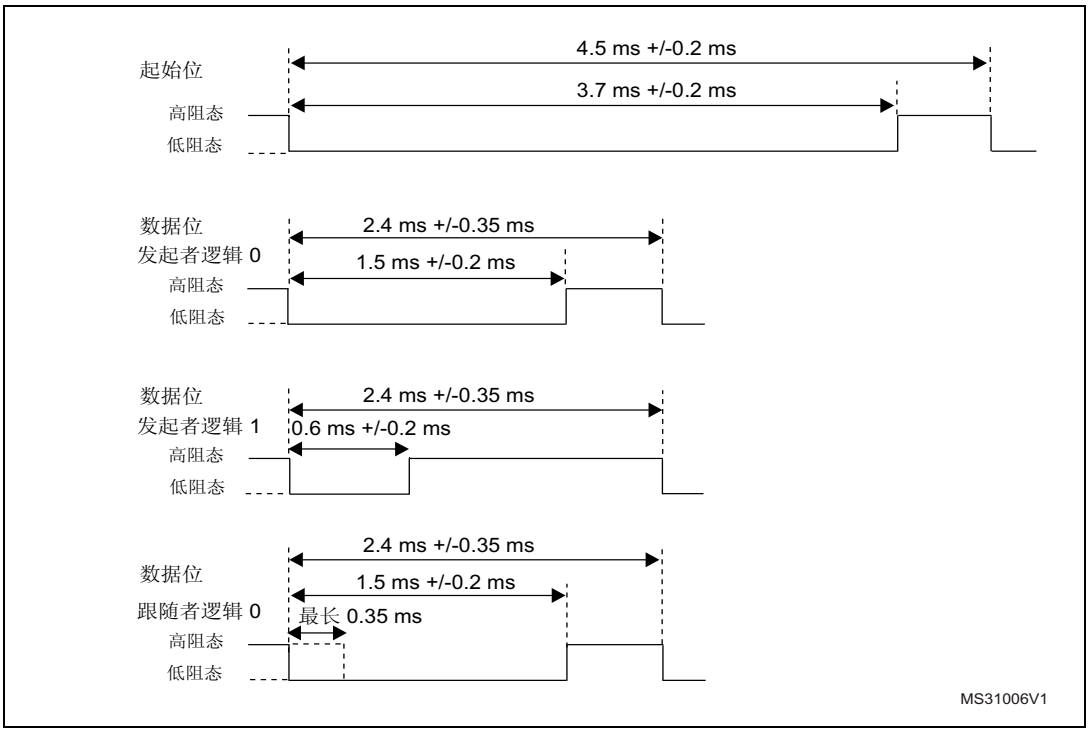


59.3.4 位时序

起始位的格式是唯一的，用于识别消息开始。应通过其低电平持续时间和总的持续时间识别起始位。

消息中起始位后的所有剩余数据位均有一致的时序。数据位结束时从高电平跳变为低电平即表示下一个数据位开始，但最后一位除外（此时 CEC 线保持高电平）。

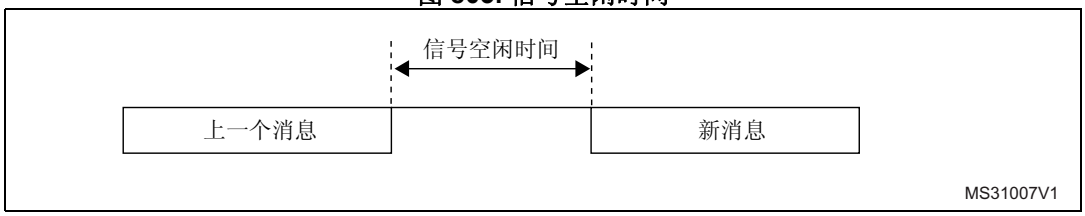
图 802. 位时序



59.4 仲裁

必须向 CEC 线发送或重新发送消息的所有器件必须确保 CEC 线已在多个位周期内处于空闲状态。该信号空闲时间定义为从前一个帧的最后一位到下一条新消息开始的间隔，如下表所示。

图 803. 信号空闲时间



由于一次只允许一个发起方，因此提供了一种仲裁机制来避免多个发起方同时开始发送时发生冲突。

CEC 线的仲裁从起始位的上升沿开始，并持续到报头块内的发起方地址位结束时为止。在此周期内，发起方将监视 CEC 线，发起方驱动 CEC 线为高阻态时，同时会读取 CEC 线的状态，如果读回的状态为 0，随后它会认为丢失仲裁、停止发送并变为发送方。

图 804. 仲裁阶段

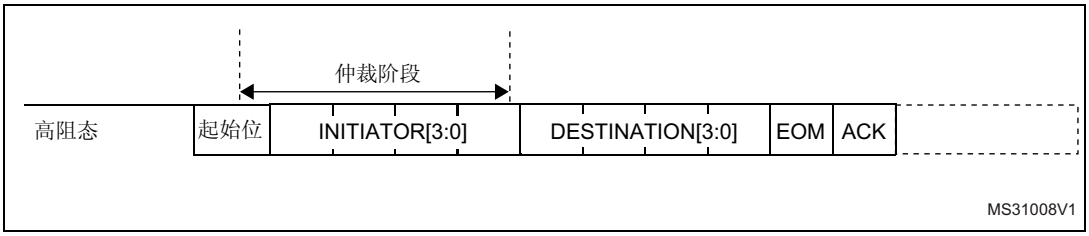
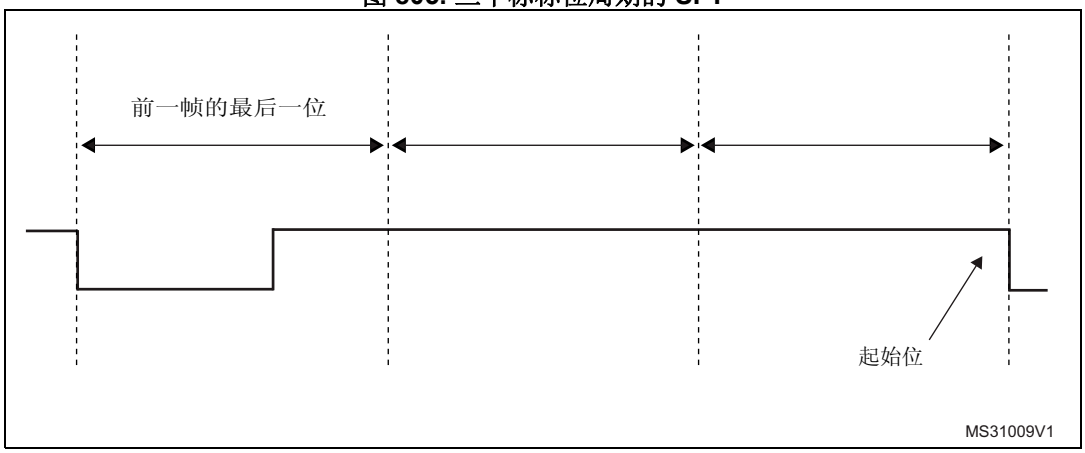


图 805 给出了三个标称位周期的 SFT 的示例

图 805. 三个标称位周期的 SFT



可配置时间窗口在开始发送前计数。

在 SFT=0x0 配置下，HDMI-CEC 器件执行自动 SFT 计算，以确保符合 HDMI-CEC 标准：

- 如果 CEC 为最后一个发送失败的总线发起方，则为 2.5 个数据位周期
- 如果 CEC 是新的总线发起方，则为 4 个数据位周期
- 如果 CEC 为最后一个发送成功的总线发起方，则为 6 个数据位周期

这样的目的是保证将最高优先级赋予失败的发送，将最低优先级赋予最后一个成功完成发送的发起方。

另外，还可以将 SFT 位配置为计数固定的时序值。可能的值为 0.5、1.5、2.5、3.5、4.5、5.5 和 6.5 个数据位周期。

59.4.1 SFT 选项位

在 SFTOPT=0 的配置下，当由软件设置发送开始命令时 (TXSOM=1)，开始对 SFT 计数

在 SFTOPT=1 时，若检测到总线空闲或线路错误条件，HDMI-CEC 器件将自动开始对 SFT 计数。如果在 TXSOM 命令时，SFT 定时器已经结束计数，则将立即开始发送，没有任何延迟。如果 SFT 定时器仍在运行，系统将等待定时器超时后再开始发送。

当 SFTOPT=1 时，下列检测到的总线事件将启动 SFT 定时器：

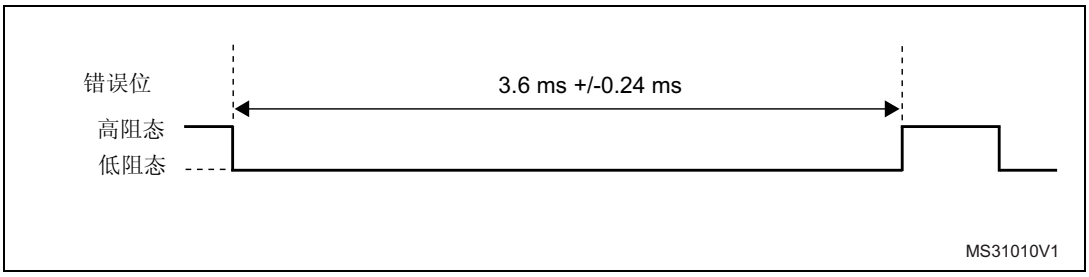
- 在常规结束发送/接收的情况下，当 TXEND/RXEND 位在消息中最后一位（ACK 位）的最短标称数据位持续时间内置 1 时。
- 在检测到发送错误的情况下，当检测到 TXERR 发送错误时 (TXERR=1) 启动 SFT 定时器。
- 在丢失 CEC 接收方应答的情况下，当在 ACK 位的标称采样时间内将 TXACKE 位置 1，将启动 SFT 定时器。
- 在出现发送下溢错误的情况下，当 TXUDR 位在 ACK 位结束时置 1 时，将启动 SFT 定时器。
- 在检测到表示接收中止的接收错误的情况下，将在检测到错误的同时启动 SFT 定时器。如果生成错误位，则在错误位结束时开始对 SFT 计数。
- 在出现错误起始位或任何未编码低阻抗空闲总线状态的情况下，当总线恢复为高阻抗空闲状态时，将立即重启 SFT 定时器。

59.5 错误处理

59.5.1 位错误

如果一个数据位（不包括起始位）被视为无效，则接收方应在 CEC 线上产生一个 1.4 倍到 1.6 倍标称数据位周期的低电平 (3.6 ms+/-0.24 ms) 来通知此错误。

图 806. 错误位时序



59.5.2 消息错误

在以下情况下，认为消息丢失，可重新发送：

- 在直接寻址的消息中未应答消息
- 在广播消息中否定应答消息
- 在 CEC 线上检测到意外的低阻抗（线路错误）

当 CEC 接口接收数据位时，可检测到三种错误标志：

59.5.3 位上升错误 (BRE)

BRE（位上升错误）：当在预期窗口外检测到位上升沿时置 1（请参见*图 807*）。BRE 标志还会在 BREIE=1 时生成 CEC 中断。

检测到 BRE 时，可根据 BRESTP 位的值停止接收消息，并会在 BREGEN 位置 1 时生成错误位。

若 BRESTP=1 时在广播消息中检测到 BRE，即使 BREGEN=0，也会生成错误位，以强制发起方在发送失败后重试。可通过配置 BREGEN=0 和 BRDNOGEN=1 禁止生成错误位。

59.5.4 短位周期错误 (SBPE)

提前检测到位下降沿时，SBPE 位将置 1（请参见*图 807*）。SBPE 标志还会在 SBPEIE=1 时生成 CEC 中断。

检测到 SBPE 错误时，始终会在线路上生成一个错误位。仅当设置监听模式 (LSTN=1) 并且满足以下条件时，检测到 SBPE 后才不生成错误位：

- 接收到包含 SBPE 的直接寻址消息
- 接收到包含 SBPE 和 BRDNOGEN 都为 1 的广播消息

59.5.5 长位周期错误 (LBPE)

未在有效窗口中检测到位下降沿时，LBPE 位置 1（请参见*图 807*）。LBPE 标志还会在 LBPEIE=1 时生成 CEC 中断。

LBPE 始终会停止接收过程，当 LBPEGEN 位置 1 时会在线路上生成错误位。

若在广播消息中检测到 LBPE，则即使 LBPEGEN=0 也会生成错误位，以强制发起方在发送失败后重试。可通过配置 LBPEGEN=0 和 BRDNOGEN=1 禁止生成错误位。

注： *必须避免 BREGEN=1 和 BRESTP=0 的配置*

图 807. 错误处理

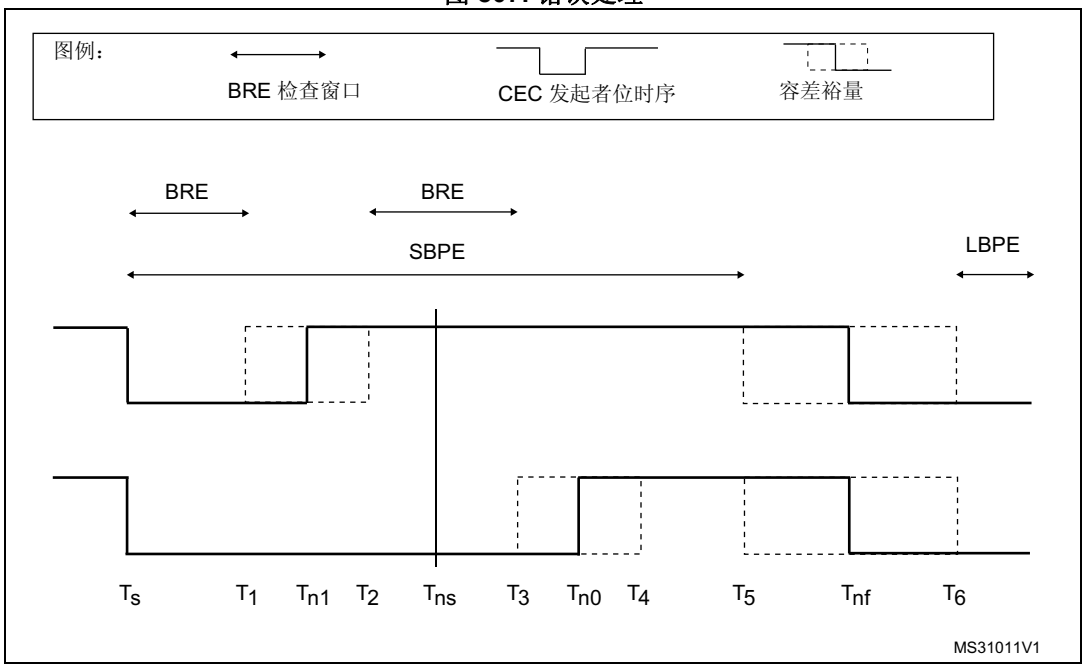


表 548. 错误处理时序参数

时间	RXTOL	ms	说明
T_s	x	0	位起始事件。
T_1	1	0.3	指示逻辑 1 时，低电平到高电平跳变的最早时间。
	0	0.4	
T_{n1}	x	0.6	指示逻辑 1 时，低电平到高电平跳变的标称时间。
T_2	0	0.8	指示逻辑 1 时，低电平到高电平跳变的最晚时间。
	1	0.9	
T_{ns}	x	1.05	标称采样时间。
T_3	1	1.2	允许器件恢复为高阻态（逻辑 0）的最早时间。
	0	1.3	
T_{n0}	x	1.5	允许器件恢复为高阻态（逻辑 0）的标称时间。
T_4	0	1.7	允许器件恢复为高阻态（逻辑 0）的最晚时间。
	1	1.8	
T_5	1	1.85	后续位开始的最早时间。
	0	2.05	
T_{nf}	x	2.4	标称数据位周期。
T_6	0	2.75	后续位开始的最晚时间。
	1	2.95	

59.5.6 发送错误检测 (TXERR)

如果 CEC 发起方在发送高阻抗且不需要接收方触发位时检测到 CEC 线处于低阻抗，则会将 TXERR 标志置 1。TXERR 标志还会在 TXERRIE=1 时生成 CEC 中断。

TXERR 触发后会停止发送消息。应用负责在发送失败后重试，最多可重试 5 次。

可按不同方式执行 TXERR 校验，具体取决于 CEC 线的不同状态和 RX 容差配置。

图 808. TXERR 检测

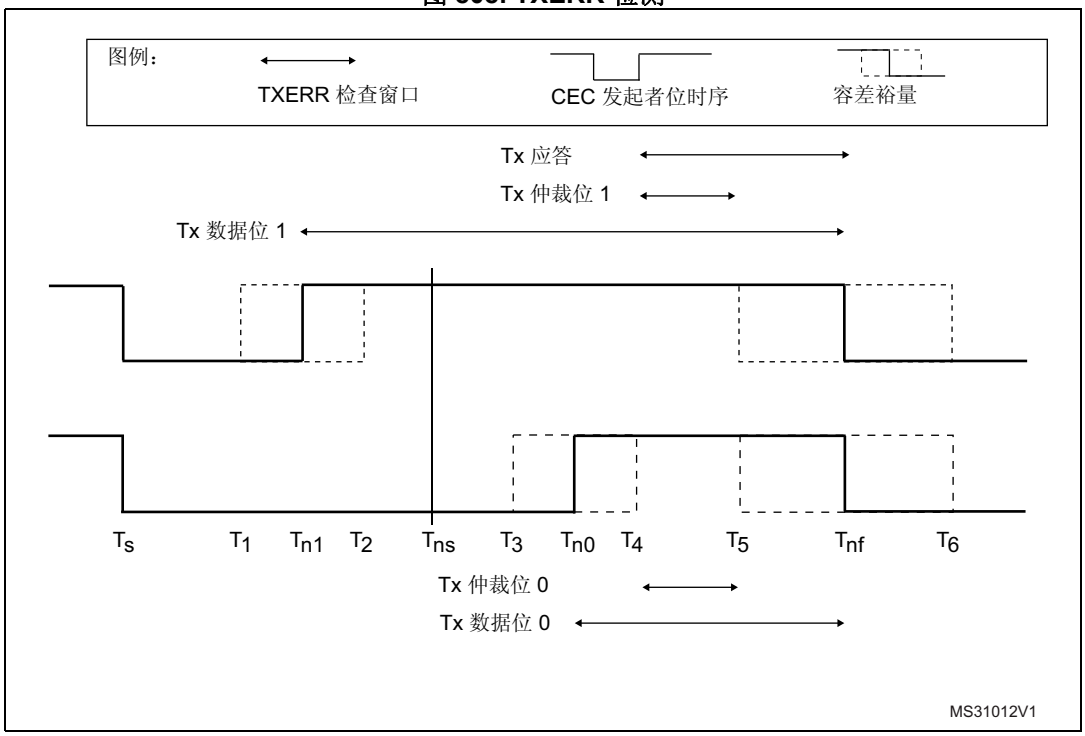


表 549. TXERR 时序参数

时间	RXTOL	ms	说明
T_s	x	0	位起始事件。
T_1	1	0.3	指示逻辑 1 时，低电平到高电平跳变的最早时间。
	0	0.4	
T_{n1}	x	0.6	指示逻辑 1 时，低电平到高电平跳变的标称时间。
T_2	0	0.8	指示逻辑 1 时，低电平到高电平跳变的最晚时间。
	1	0.9	
T_{ns}	x	1.05	标称采样时间。
T_3	1	1.2	允许器件恢复为高阻态（逻辑 0）的最早时间。
	0	1.3	
T_{n0}	x	1.5	允许器件恢复为高阻态（逻辑 0）的标称时间。

表 549. TXERR 时序参数（续）

时间	RXTOL	ms	说明
T_4	0	1.7	允许器件恢复为高阻态（逻辑 0）的最晚时间。
	1	1.8	
T_5	1	1.85	后续位开始的最早时间。
	0	2.05	
T_{nf}	x	2.4	标称数据位周期。
T_6	0	2.75	后续位开始的最晚时间。
	1	2.95	

59.6 HDMI-CEC 中断

以下情况下可以产生中断：

- 在接收期间，如果完成接收块传输或出现接收错误。
- 在发送期间，如果完成发送块传输或出现发送错误。

表 550. HDMI-CEC 中断

中断事件	事件标志	使能控制位
接收到 Rx 字节	RXBR	RXBRIE
接收结束	RXEND	RXENDIE
Rx 上溢	RXOVR	RXOVRIE
Rx 位上升错误	BRE	BREIE
Rx 短位周期错误	SBPE	SBPEIE
Rx 长位周期错误	LBPE	LBPEIE
Rx 丢失应答错误	RXACKE	RXACKEIE
仲裁丢失	ARBLST	ARBLSTIE
Tx 字节请求	TXBR	TXBRIE
发送结束	TXEND	TXENDIE
Tx 缓冲区下溢	TXUDR	TXUDRIE
Tx 错误	TXERR	TXERRIE
Tx 丢失应答错误	TXACKE	TXACKEIE

59.7 HDMI-CEC 寄存器

有关寄存器说明中使用的缩写，请参见*第 94 页的第 1.1 节*。

59.7.1 CEC 控制寄存器 (CEC_CR)

CEC control register

偏移地址：0x00

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TX EOM	TX SOM	CEC EN
													rs	rs	rw

位 31:3 保留，必须保持复位值。

位 2 **TXEOM**：Tx 消息结束 (Tx End Of Message)

TXEOM 位由软件置 1，来控制 CEC 消息的最后一个字节的发送。

TXEOM 由硬件清零，清零的时刻和条件与 TXSOM 相同。

0：EOM=0 时发送 TXDR 数据字节

1：EOM=1 时发送 TXDR 数据字节

注： CECEN=1 时，TXEOM 必须置 1

TXEOM 必须在向 TXDR 写入发送数据前置 1

如果 TXEOM 在 TXSOM=0 时置 1，发送的消息将仅包含 1 个字节 (HEADER)（PING 消息）

位 1 **TXSOM**：Tx 消息开始 (Tx Start Of Message)

TXSOM 位由软件置 1 来控制发送 CEC 消息的第一个字节。如果 CEC 消息只包含一个字节，TXEOM 必须在 TXSOM 之前置 1。

SFT 结束后，立刻在 CEC 线上发送起始位。如果 TXSOM 在接收消息的过程中置 1，则将在接收结束时开始发送。

在出现发送下溢 (TXUDR=1)、否定应答 (TXACKE=1) 和发送错误 (TXERR=1) 的情况下，在发送消息的最后一个字节和肯定应答 (TXEND=1) 后，TXSOM 将由硬件清零。此位还可通过 CECEN=0 清零。在仲裁丢失的情况下 (ARBLST=1)，此位不会清零，同时会自动重试发送。

TXSOM 还可用作状态位，用于通知应用程序是否存在正在挂起或执行的任何发送请求。应用程序可通过将 CECEN 位清零随时中止发送请求。

0：没有正在进行的 CEC 发送

1：CEC 发送命令

注： CECEN=1 时，TXSOM 必须置 1

发送数据可用于 TXDR 时，TXSOM 必须置 1

从 TXDR[7:4] 而非从仅用于接收的 CEC_CFGR.OAR 获取 HEADER 的前四位（包含自有外设地址）

位 0 **CECEN**：CEC 使能 (CEC Enable)

CECEN 位由软件置 1 和清零。CECEN=1 启动消息接收过程并使能 TXSOM 控制。CECEN=0 禁止 CEC 外设、将 CEC_CR 寄存器的所有位清零并中止任何正在进行的接收或发送过程。

0：CEC 外设关闭

1：CEC 外设开启

59.7.2 CEC 配置寄存器 (CEC_CFGR)

CEC configuration register

此寄存器用于配置 HDMI-CEC 控制器。

偏移地址：0x04

复位值：0x0000 0000

注意： 必须只在 CECEN=0 时对 CEC_CFGR 执行写操作。

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
LSTN	OAR[14:0]														
rw	rw														
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	SFT OPT	BRDN OGEN	LBPE GEN	BRE GEN	BRE STP	RX TOL	SFT[2:0]		
							rw	rw	rw	rw	rw	rw	rw		

位 31 **LSTN**：监听模式 (Listen mode)

LSTN 位由软件置 1 和清零。

0：CEC 外设只接收寻址到其自有地址 (OAR) 的消息。寻址到不同目标地址的消息将被忽略。始终接收广播消息。

1：CEC 外设接收寻址到其自有地址 (OAR) 的消息，此时会发送肯定应答。也接收寻址到不同目标地址的消息，但此时不发送应答。

位 30:16 **OAR**：自有地址配置 (Own addresses configuration)

OAR 位由软件置 1 来选择接收模式下必须考虑的目标逻辑地址。每个 bit 置 1 后，它所对应的位置的值，就是被使能的 CEC 逻辑地址的值。OAR 字段的最低位，对应 CEC 逻辑地址 0。

在 HEADER 接收结束时，所接收的目标地址将与使能的地址进行比较。在地址匹配的情况下，将应答并接收传入的消息。在地址不匹配的情况下，仅在监听模式下 (LSTN=1) 接收传入的消息，但不发送应答。始终接收广播消息。

示例：

OAR = 0b000 0000 0010 0001 意味着 CEC 应答地址 0x0 和 0x5。因此，将接收传入这两个地址中之一的每个消息。

位 15:9 保留，必须保持复位值。

位 8 **SFTOP**：*SFT 选项位 (SFT Option Bit)*

SFTOPT 位由软件置 1 和清零。

0：当 TXSOM 由软件置 1 时，将启动 SFT 定时器

1：在消息发送/接收结束时自动启动 SFT 定时器

位 7 **BRDNOGEN**：避免*在广播中生成*错误位 (Avoid Error-Bit *Generation in Broadcast)*

BRDNOGEN 位由软件置 1 和清零。

0：在广播消息上检测到 BRE 且满足 BRESTP=1 和 BREGEN=0 时会在 CEC 线上生成错误位。在广播消息上检测到 LBPE 且满足 LBPEGEN=0 时会在 CEC 线上生成错误位。

1：在上述相同条件下不会生成错误位。即使在监听模式下，在广播消息中检测到 SBPE，也不会生成错误位。

位 6 **LBPEGEN**：在长*位周期错误*时生成错误位 (Generate Error-Bit on Long Bit Period Error)

LBPEGEN 位由软件置 1 和清零。

0：检测到 LBPE 时不在 CEC 线上生成错误位

1：检测到 LBPE 时在 CEC 线上生成错误位

注： *如果 BRDNOGEN=0，即使 LBPEGEN=0，也会在广播中检测到 LBPE 后生成错误位。*

位 5 **BREGEN**：*在位上升错误*时生成错误位 (Generate Error-Bit on Bit Rising Error)

BREGEN 位由软件置 1 和清零。

0：检测到 BRE 时不在 CEC 线上生成错误位

1：检测到 BRE 时在 CEC 线上生成错误位（如果 BRESTP 置 1）

注： 如果 BRDNOGEN=0，即使 BREGEN=0，也会在广播中检测到 BRE 和 BRESTP=1 后生成错误位。

位 4 **BRESTP**：*在位上升错误时 Rx 停止 (Rx-Stop on Bit Rising Error)*

BRESTP 位由软件置 1 和清零。

0：检测到 BRE 时不停止接收 CEC 消息。在 1.05 ms 内对数据位进行采样。

1：检测到 BRE 时停止接收消息

位 3 **RXTOL**：Rx 容差 (Rx-Tolerance)

RXTOL 位由软件置 1 和清零。

0：标准容差裕量：

– 起始位，+/- 200 µs 上升，+/- 200 µs 下降。

– 数据位：+/- 200 µs 上升。+/- 350 µs 下降。

1：扩展容差

– 起始位，+/- 400 µs 上升，+/- 400 µs 下降

– 数据位，+/- -300 µs 上升，+/- 500 µs 下降

位 2:0 **SFT**：*信号空闲时间 (Signal Free Time)*

SFT 位由软件置 1。在 SFT=0x0 的配置下，发送前等待的标称数据位周期数由硬件根据发送历史管理。在所有其它配置下，SFT 数由软件决定。

″ 0x0

– 如果 CEC 为最后一个发送失败的总线发起方，则为 2.5 个数据位周期（ARBLST=1、TXERR=1、TXUDR=1 或 TXACKE= 1）

– 如果 CEC 是新的总线发起方，则为 4 个数据位周期

– 如果 CEC 为最后一个发送成功的总线发起方，则为 6 个数据位周期 (TXEOM=1)

″ 0x1：0.5 个标称数据位周期

″ 0x2：1.5 个标称数据位周期

″ 0x3：2.5 个标称数据位周期

″ 0x4：3.5 个标称数据位周期

″ 0x5：4.5 个标称数据位周期

″ 0x6：5.5 个标称数据位周期

″ 0x7：6.5 个标称数据位周期

59.7.3 CEC 发送数据寄存器 (CEC_TXDR)

CEC Tx data register

偏移地址：0x8

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TXD[7:0]							
								w	w	w	w	w	w	w	w

位 31:8 保留，必须保持复位值。

位 7:0 **TXD[7:0]**：Tx 数据寄存器 (Tx Data register)。
TXD 是一个只写寄存器，其中包含要发送的数据字节。

59.7.4 CEC 接收数据寄存器 (CEC_RXDR)

CEC Rx Data Register

偏移地址：0xC

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RXD[7:0]							
								r	r	r	r	r	r	r	r

位 31:8 保留，必须保持复位值。

位 7:0 **RXD[7:0]**：Rx 数据寄存器 (Rx Data register)。
RXD 是一个只读寄存器，包含从 CEC 线接收的最后一个数据字节。

59.7.5 CEC 中断和状态寄存器 (CEC_ISR)

CEC Interrupt and Status Register

偏移地址：0x10

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	TX ACKE	TX ERR	TX UDR	TX END	TXBR	ARB LST	RX ACKE	LBPE	SBPE	BRE	RX OVR	RX END	RXBR
			rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1	rc_w1

位 31:13 保留，必须保持复位值。

位 12 **TXACKE**：Tx 丢失应答错误 (Tx-Missing Acknowledge Error)

在发送模式下，TXACKE 由硬件置 1 以通知应用程序未接收到任何应答。在广播发送时，TXACKE 通知应用程序已接收到否定应答。TXACKE 中止消息发送并清除 TXSOM 和 TXEOM 控制位。

TXACKE 由软件写 1 清零。

位 11 **TXERR**：Tx 错误 (Tx-Error)

在发送模式下，如果 CEC 发起方在 CEC 线释放时检测到 CEC 处于低阻抗，则 TXERR 将由硬件置 1。TXERR 中止消息发送并清除 TXSOM 和 TXEOM 控制位。

TXERR 由软件写 1 清零。

位 10 **TXUDR**：Tx 缓冲区下溢 (Tx-Buffer Underrun)

在发送模式下，如果应用程序未在下一个字节发送前及时加载 TXDR，则 TXUDR 将由硬件置 1。TXUDR 中止消息发送并清除 TXSOM 和 TXEOM 控制位。

TXUDR 由软件写 1 清零。

位 9 **TXEND**：发送结束 (End of Transmission)

TXEND 由硬件置 1 后，可向应用程序通知已成功发送 CEC 消息的最后一个字节。TXEND 将 TXSOM 和 TXEOM 控制位清零。

TXEND 由软件写 1 清零。

位 8 **TXBR**：Tx 字节请求 (Tx-Byte Request)

TXBR 由硬件置 1 后可通知应用程序必须向 TXDR 写入下一个发送数据。当发送完当前发送字节的第 4 位时，TXBR 将置 1。应用程序必须在出现发送下溢错误 (TXUDR) 前，在 6 个标称数据位周期内向 TXDR 发送下一个字节。

TXBR 由软件写 1 清零。

位 7 **ARBLST**：仲裁丢失 (Arbitration lost)

ARBLST 由硬件置 1 以通知应用程序：由于 TXSOM 命令后发生仲裁丢失事件，CEC 器件正切换为接收模式。引起 ARBLST 的原因可能是提前开始争用 CEC 器件，也可能是有别的 CEC 器件提前开始争用总线，或者是同时开始但是拥有较高的优先级。ARBLST 触发后，TXSOM 位将一直等待下一次发送尝试。

ARBLST 由软件写 1 清零。

位 6 **RXACKE**：Rx 丢失应答 (Rx-Missing Acknowledge)

在发送模式下，RXACKE 由硬件置 1 后以通知应用程序未在 CEC 线上检测到任何应答。RXACKE 仅适用于广播消息，在监听模式下仅适用于非直接寻址的消息（OAR 中不使能目标地址）。RXACKE 用于中止接收消息。

RXACKE 由软件写 1 清零。

位 5 **LBPE**：Rx 长位周期错误 (Rx-Long Bit Period Error)

当数据位波形上检测到长位周期错误时，LBPE 由硬件置 1。在仍需要下降沿的情况下，当 RXTOL 允许的最大位扩展容差结束时，LBPE 将置 1。LBPE 总是会停止接收 CEC 消息。如果 LBPEGEN=1，LBPE 会在 CEC 线上生成错误位。广播时，即使 LBPEGEN=0 也会生成错误位。

LBPE 由软件写 1 清零。

位 4 **SBPE**：Rx 短位周期错误 (Rx-Short Bit Period Error)

当数据位波形上检测到短位周期错误时，SBPE 由硬件置 1。出现所预期的下降沿时，SBPE 将置 1。SBPE 在 CEC 线上生成错误位。

SBPE 由软件写 1 清零。

位 3 **BRE**：Rx 位上升错误 (Rx-Bit Rising Error)

当数据位波形上检测到位上升错误时，BRE 由硬件置 1。在仍需要上升沿的情况下，在上升沿出现的时机不对时或 RXTOL 允许的最大 BRE 容差结束时，BRE 将置 1。如果 BRESTP=1，BRE 将停止接收消息。如果 BREGEN=1，BRE 会在 CEC 线上生成错误位。

BRE 由软件写 1 清零。

位 2 **RXOVR**：Rx 上溢 (Rx-Overrun)
如果在 CEC 线上接收到新字节并将新字节存储到 RXD 中时 RXBR 位已经是置位状态，则 RXOVR 将由硬件置 1。RXOVR 触发后会停止接收消息，因此不会发送应答。广播时，将发送否定应答。
RXOVR 由软件写 1 清零。

位 1 **RXEND**：接收结束 (End Of Reception)
RXEND 由硬件置 1 以通知应用程序已从 CEC 线接收到 CEC 消息的最后一个字节并将其存储到 RXD 缓冲区。RXEND 与 RXBR 同时置 1。
RXEND 由软件写 1 清零。

位 0 **RXBR**：接收到 Rx 字节 (Rx-Byte Received)
RXBR 位由硬件置 1 以通知应用程序已从 CEC 线接收到新字节并将其存储到 RXD 缓冲区。
RXBR 由软件写 1 清零。

59.7.6 CEC 中断使能寄存器 (CEC_IER)

CEC interrupt enable register

偏移地址：0x14

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	TXACKIE	TXERRIE	TX UDRIE	TXENDIE	TXBRIE	ARBLSTIE	RXACKIE	LBPEIE	SBPEIE	BREIE	RXOVRIE	RXENDIE	RXBRIE
			rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:13 保留，必须保持复位值。

位 12 **TXACKIE**：Tx 丢失应答错误中断使能 (Tx-Missing Acknowledge Error Interrupt Enable)
TXACKEIE 位由软件置 1 和清零。
0：禁止 TXACKE 中断
1：使能 TXACKE 中断

位 11 **TXERRIE**：Tx 错误中断使能 (Tx-Error Interrupt Enable)
TXERRIE 位由软件置 1 和清零。
0：禁止 TXERR 中断
1：使能 TXERR 中断

位 10 **TXUDRIE**：Tx 下溢中断使能 (Tx-Underrun Interrupt Enable)
TXUDRIE 位由软件置 1 和清零。
0：禁止 TXUDR 中断
1：使能 TXUDR 中断

位 9 **TXENDIE**：Tx 消息结束中断使能 (Tx-End Of Message Interrupt Enable)
TXENDIE 位由软件置 1 和清零。
0：禁止 TXEND 中断
1：使能 TXEND 中断

位 8 **TXBRIE**：Tx 字节请求中断使能 (Tx-Byte Request Interrupt Enable)
TXBRIE 位由软件置 1 和清零。
0：禁止 TXBR 中断
1：使能 TXBR 中断

位 7 **ARBLSTIE**：仲裁丢失中断使能 (Arbitration Lost Interrupt Enable)
ARBLSTIE 位由软件置 1 和清零。
0：禁止 ARBLST 中断
1：使能 ARBLST 中断

位 6 **RXACKIE**：Rx 丢失应答错误中断使能 (Rx-Missing Acknowledge Error Interrupt Enable)
RXACKIE 位由软件置 1 和清零。
0：禁止 RXACKE 中断
1：使能 RXACKE 中断

位 5 **LBPEIE**：长位周期错误中断使能 (Long Bit Period Error Interrupt Enable)
LBPEIE 位由软件置 1 和清零。
0：禁止 LBPE 中断
1：使能 LBPE 中断

位 4 **SBPEIE**：短位周期错误中断使能 (Short Bit Period Error Interrupt Enable)
SBPEIE 位由软件置 1 和清零。
0：禁止 SBPE 中断
1：使能 SBPE 中断

位 3 **BREIE**：位上升错误中断使能 (Bit Rising Error Interrupt Enable)
BREIE 位由软件置 1 和清零。
0：禁止 BRE 中断
1：使能 BRE 中断

位 2 **RXOVRIE**：Rx 缓冲区上溢中断使能 (Rx-Buffer Overrun Interrupt Enable)
RXOVRIE 位由软件置 1 和清零。
0：禁止 RXOVR 中断
1：使能 RXOVR 中断

位 1 **RXENDIE**：接收结束中断使能 (End Of Reception Interrupt Enable)
RXENDIE 位由软件置 1 和清零。
0：禁止 RXEND 中断
1：使能 RXEND 中断

位 0 **RXBRIE**：接收到 Rx 字节中断使能 (Rx-Byte Received Interrupt Enable)
RXBRIE 位由软件置 1 和清零。
0：禁止 RXBR 中断
1：使能 RXBR 中断

注意： (*) 必须只在 CECEN=0 时对 CEC_IER 执行写操作。

59.7.7 HDMI-CEC 寄存器映射

下表对 HDMI-CEC 寄存器进行了汇总。

表 551. HDMI-CEC 寄存器映射和复位值

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x00	**CEC_CR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TXEOM	TXSOM	CECEN
	Reset value																														0	0	0
0x04	**CEC_CFGR**	LSTN	OAR[14:0]															Res.	Res.	Res.	Res.	Res.	Res.	Res.	SFTOPT	BRDNOGEN	LBPEGEN	BREGEN	BRESTP	RXTOL	SFT[2:0]		
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0								0	0	0	0	0	0	0	0	0
0x08	**CEC_TXDR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TXD[7:0]							
	Reset value																									0	0	0	0	0	0	0	0
0x0C	**CEC_RXDR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RXD[7:0]							
	Reset value																									0	0	0	0	0	0	0	0
0x10	**CEC_ISR**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TXACKE	TXERR	TXUDR	TXEND	TXBR	ARBLST	RXACKE	LBPE	SBPE	BRE	RXOVR	RXEND	RXBR
	Reset value																				0	0	0	0	0	0	0	0	0	0	0	0	0
0x14	**CEC_IER**	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TXACKIE	TXERRIE	TXUDRIE	TXENDIE	TXBRIE	ARBLSTIE	RXACKIE	LBPEIE	SBPEIE	BREIE	RXOVRIE	RXENDIE	RXBRIE
	Reset value																				0	0	0	0	0	0	0	0	0	0	0	0	0

有关寄存器边界地址的信息，请参见*第 100 页的第 2.2.2 节*。

60 调试基础结构

60.1 前言

调试基础结构允许软件设计人员调试和跟踪他们的嵌入式软件。

调试功能可以通过 JTAG/串行线调试访问端口，使用行业标准调试工具进行控制。跟踪端口允许捕获数据以进行记录和分析。

跟踪和调试系统被设计为支持各种典型用例：

- **低成本跟踪**

  通过单线调试输出提供有限跟踪功能， 可以支持使用“printf”进行代码指令、跟踪数据和地址观察点、中断检测以及程序计数器采样。即使在关闭处理器或其时钟信号时，也可以保持单线跟踪功能。

- **断点调试**

  处理器可以使用连接到 JTAG/SWD 调试端口的设备进行调试， 可以支持断点和观察点设置、逐步执行代码以及存储器访问等。

- **通过跟踪端口跟踪代码执行**

  跟踪信息合并为单个跟踪流并实时输出到跟踪端口分析器。跟踪中嵌入的 ID 可使分析器识别每个信息包的来源。

- **在循环缓冲区中连续捕获跟踪**

  合并的跟踪信息可以存储到片上循环缓冲区中，而不是输出到片外。跟踪存储可以通过调试器命令、软件命令、外部触发信号或内部事件等来启动和停止。

- **将缓冲区内容清空到跟踪端口**

  存储的跟踪信息可以转存到片外的跟踪端口分析器。清空缓冲区可以通过调试器、软件、外部触发和内部事件等启动。

- **使用调试器读取缓冲区**

  调试器可以通过调试端口读取跟踪缓冲区的内容。这与通过跟踪端口相比，虽然读取速度要慢一些，但是可以在不使用跟踪端口分析器的情况下实现基本跟踪功能。

- **通过软件分析存储的跟踪信息**

  跟踪缓冲区可通过处理器内核读取，或通过 DMA 传输到系统存储器。借助这一强大的功能，可以通过内置的测试软件来实时监视代码执行，分析并识别故障以及自动处理异常等。

- **上传存储的跟踪信息**

  存储的跟踪信息也可以使用 MCU 的各种通信接口（USB、USART、SPI、I2C、以太网以及 CAN 等）上传到主机。如果跟踪端口不可访问，例如对部署产品进行远程监控和故障分析时，这一点尤其有用。

60.2 调试基本接口特性

为支持软件开发和系统集成，提供了以下一整套跟踪和调试功能：

- 断点调试
- 代码执行跟踪
- 软件指令
- 交叉触发
- JTAG 调试端口
- 串行线调试端口
- 触发输入和输出
- 串行线跟踪端口
- 跟踪端口
- ARM® CoreSight™ 调试和跟踪组件

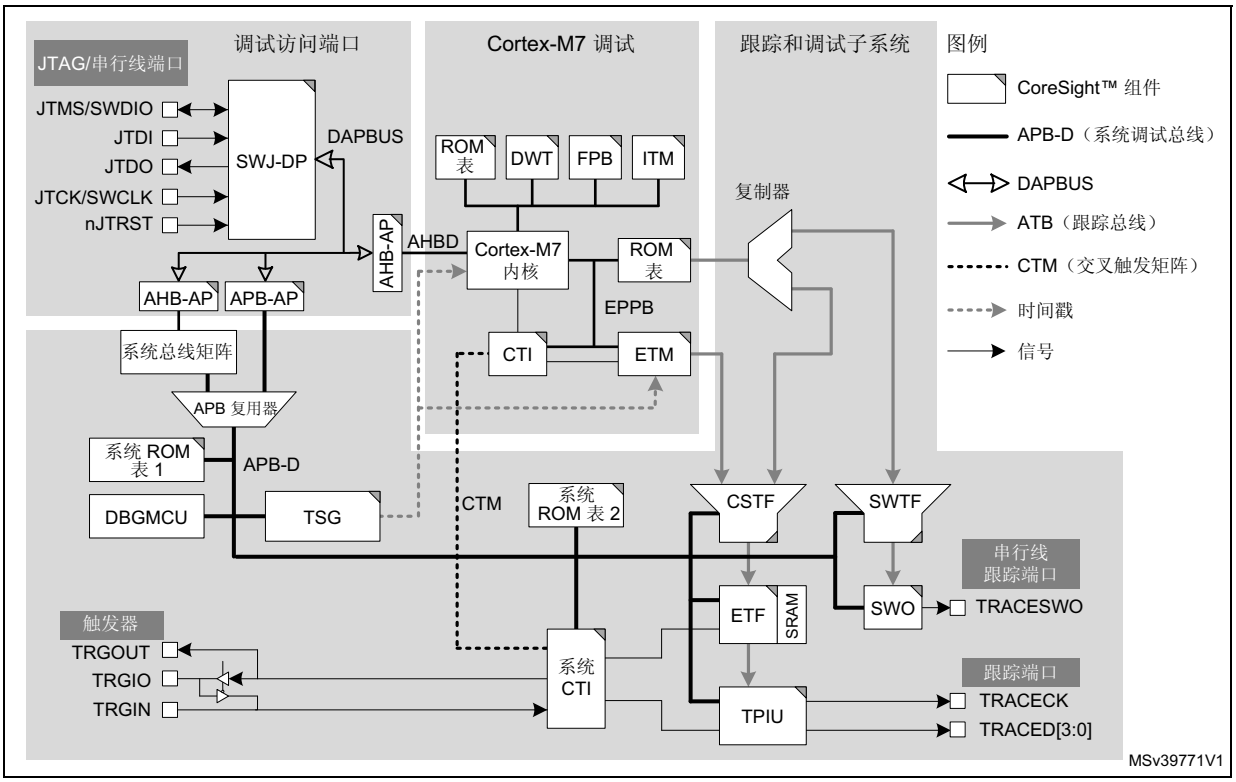
本文档在比较高的高度上对 CoreSight 组件进行了介绍， 有关详细信息，请参见*第 60.7 节*中的 ARM® 文档。

60.3 调试基础结构功能描述

60.3.1 调试基础结构框图

框图显示了调试基础结构的逻辑分区。

图 809. 调试基础结构框图



60.3.2 调试基础结构引脚和内部信号

表 552. JTAG/串行调试端口引脚

引脚名称	JTAG 调试端口		SW 调试端口		引脚分配
	类型	说明	类型	说明	
JTMS/SWDIO	I	JTAG 测试模式选择	IO	串行线数据输入/输出	PA13
JTCK/SWCLK	I	JTAG 测试时钟	I	串行线时钟	PA14
JTDI	I	JTAG 测试数据输入	-	-	PA15
JTDO	O	JTAG 测试数据输出	-	-	PB3
NJTRST	I	JTAG 测试复位	-	-	PB4

表 553. 跟踪端口引脚

引脚名称	类型	说明	引脚分配
TRACED0	O	跟踪同步数据输出 0	请参见数据手册
TRACED1	O	跟踪同步数据输出 1	
TRACED2	O	跟踪同步数据输出 2	
TRACED3	O	跟踪同步数据输出 3	
TRACECK	O	TRACE 时钟	

表 554. 串行线跟踪端口引脚

引脚名称	类型	说明	引脚分配
TRACESWO	O	单线跟踪同步数据输出	PB3(1)

1. TRACESWO 与 JTDO 多路复用。这意味着单线跟踪仅在使用串行线调试接口时可用，而不是在使用 JTAG 时可用。

表 555. 触发引脚

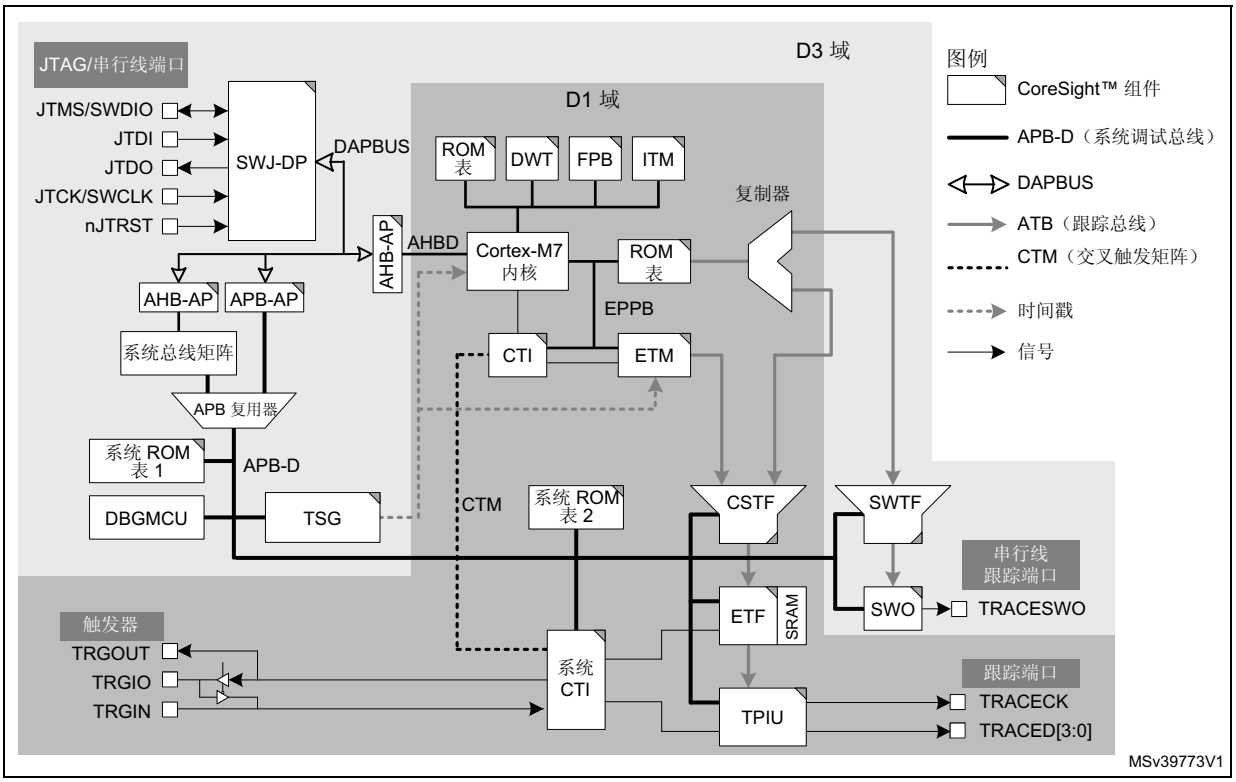
引脚名称	类型	说明	引脚分配
TRGIN	I	外部触发输入	请参见数据手册
TRGOUT	O	外部触发输出	
TRGIO	IO	外部触发输入和输出(1)	

1. TRGIO 可以通过 DBGMCU 中的 TRGOEN 位配置为输入或输出。如果配置为输入，则将其连接到 TRGIN。如果配置为输出，则将其连接到 TRGOUT。这是因为 TRGIN 和 TRGOUT 在某些封装上不可用。

60.3.3 调试基础结构电源、时钟和复位

电源域

图 810. 调试基础结构的电源域

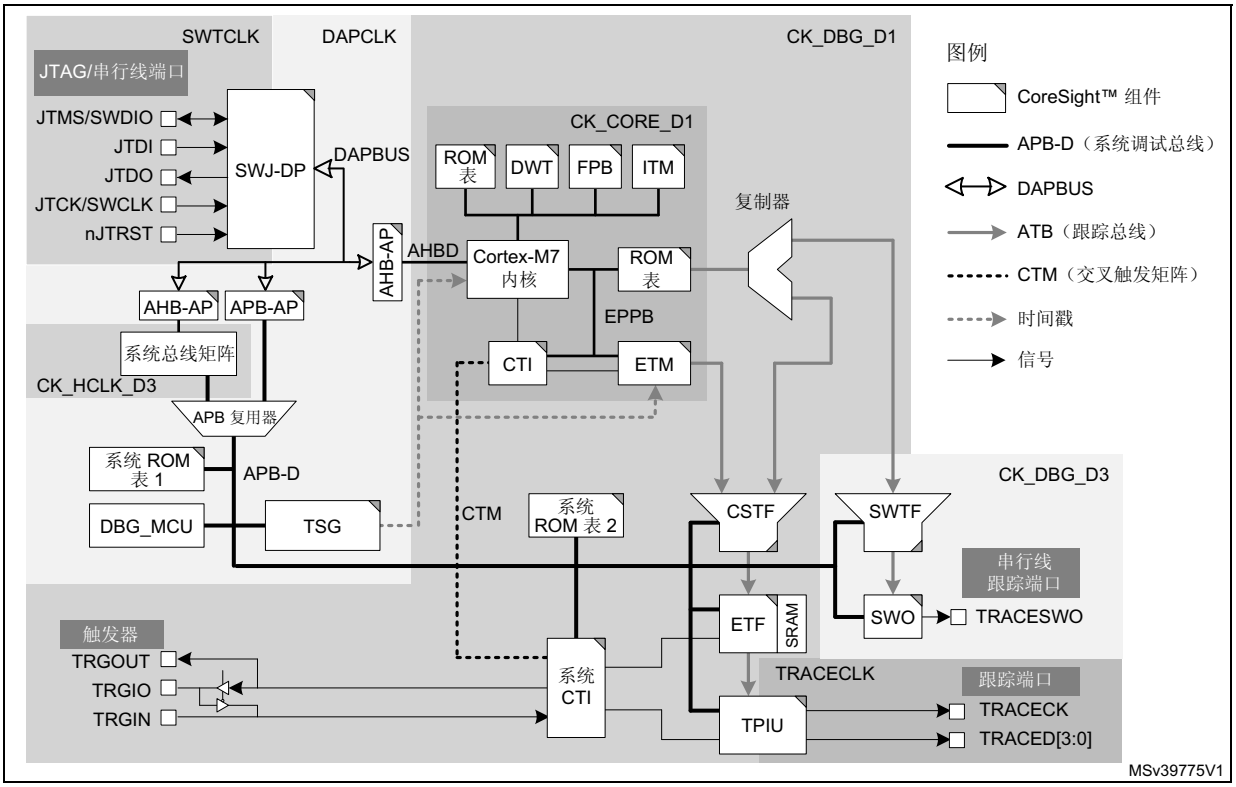


调试组件分布在 D1 和 D3 电源域中。只要调试器接通，就认为 D3 电源域处于开启状态。因此 D3 电源域包含 SWJ-DP，当其他电源域关闭时，调试器仍与 SoC 保持连接状态。此外，D3 电源域还包含时间戳发生器、DBGMCU 和串行线跟踪功能。

D1 电源域包含 Cortex-M7 内核和相关的调试跟踪组件， 还包含位于 APB-D 上的系统跟踪组件。因此，只要需要对 Cortex-M7 进行调试访问，或者当处理器上的跟踪功能处于活动状态时，此电源域就需要启用。

时钟域

图 811. 调试基础结构的时钟域



调试器通过调试接口引脚 JTCK/SWCLK 提供调试端口时钟 SWTCLK。该时钟用于在串行线和 JTAG 模式下寄存串行输入数据，同时操作状态机和调试端口的内部逻辑。因此，为确保调试端口返回到空闲状态，在访问结束之后，该时钟还必须持续几个周期。

SWJ-DP 包含一个异步接口，用来连接到 DAPCLK 域，该域覆盖 SWJ-DP 的余部和访问端口。DBGMCU、时间戳生成器和系统 ROM 表 1 也包含在 DAPCLK 域中。

CK_DBG_D3 为 SWO 和串行线跟踪聚合器提供时钟。

DAPCLK 和 CK_DBG_D3 都是通过对 D3 域系统时钟 (CK_HCLK_D3) 进行门控而得到的。

CK_DBG_D1 为 D1 电源域中的跟踪组件（系统 ROM 表 2、CoreSight 跟踪聚合器、ETF 以及系统 CTI 和 TPIU）提供时钟，该时钟是通过对 D1 域系统时钟 (CK_HCLK_D1) 进行门控而得到的。

TRACECLK 是跟踪端口输出时钟， 是通过对系统时钟 (CK_SYS) 进行门控而得到的，但是当 PLL1 为系统时钟源时， TRACECLK 直接取自 PLL1 VCO 输出，大小为其三分之一， 之所以需要如此操作，其目的是为了支持跟踪端口在处理器以其最大频率运行时的高数据吞吐量。

所有的调试时钟（DAPCLK 除外）都可以通过 DBGMCU 中的寄存器位使能和禁止。DAPCLK 域可以通过调试器使用调试端口 CTRL/STAT 寄存器中的 CDBGPWRUPREQ 位来使能。必须先使能时钟，然后调试器才能访问器件上的所有调试功能。上电时和调试器断开时应禁止时钟，以免浪费能源。

处理器中包含的调试跟踪组件（ETM ITM、DWG 以及 FPB 等）由相应的内核时钟 (CK_CORE_D1) 提供时钟信号。

低功耗模式调试

器件具有节能特性，允许在不需要的时候关闭或停止单个电源域。如果某个电源域被关闭或没有时钟信号，则调试器无法访问该域中的所有调试组件。为了避免这种情况，设计了节能模式仿真这一解决方案。如果对该域使能了仿真功能，则该域仍会进入节能模式，但其时钟和电源保持不变。换句话说，该域的行为就像处于节能模式，但调试器仍然可以对其进行访问。

仿真模式在 MCU 调试 (DBGMCU) 单元中编程。有关详细信息，请参见*第 60.5.8 节*。

调试基础结构的复位

除了调试端口和访问端口之外，调试组件将通过其各自的电源域复位进行复位。调试端口 (SWJ-DP) 只能通过 D3 域的上电复位而复位。

60.4 调试访问端口功能描述

调试访问端口 (DAP) 是一个调试子系统，包含串行线和 JTAG 调试端口 (SWJ-DP) 以及三个访问端口。

60.4.1 串行线和 JTAG 调试端口 (SWJ-DP)

SWJ-DP 是一个 CoreSight 组件，提供用来连接调试设备的外部访问端口。

端口可以配置为：

- 一个 5 针标准 JTAG 调试端口 (JTAG-DP)
- 一个 2 针（时钟 + 数据）“串行线”调试端口 (SW-DP)

这两种模式共用相同的 IO 引脚，因此两者互斥。

默认情况下，系统复位或上电复位后选择 JTAG-DP 模式。五个 IO 由硬件在调试备用功能模式下配置。在 SWJ-DP 模式下，在 JTDI、JTMS/SWDIO 和 nJTRST 引线上使用了上拉电阻，在 JTCK/SWCLK 引线上使用了下拉电阻。

调试器可以通过在 JTMS/SWDIO 上传送以下串行数据序列来选择 SW-DP 模式：

...,(50 或以上个 1),...,0,1,1,1,1,1,0,0,1,1,1,1,1,0,0,1,1,1,...,(50 或以上个 1),...

对于每一数据位都必须循环选通 JTCK/SWCLK。

在 SW-DP 模式下，未使用的 JTAG 引线 JTDI、JTDO 和 nJTRST 可用于其他功能。

所有 SWJ 端口 IO 都可以通过软件重新配置为其他功能，不过在这种情况下不能再进行调试。

串行线调试端口

串行线调试协议使用两个引脚：

- SWCLK：从主机到目标的时钟
- SWDIO：双向串行数据（需要 100kΩ 上拉电阻）

串行数据传输与时钟同步，首先传输 LSB。传输过程包括三个阶段：

1. 主机发送的数据包请求（8 位）
2. 目标发送的确认响应（3 位）
3. 主机（写操作时）或目标（读操作时）发送的传输数据（33 位）

只有在确认响应为“OK”时才进行数据传输。

在每一个传输阶段之间，如果数据传输方向发生改变，则插入一个单个时钟周期的周转时间。

表 556. 数据包请求

字段位	名称	说明
0	启动	必须为 1
1	APnDP	0：DP 寄存器访问——有关 DP 寄存器的列表，请参见*表 560* 1：AP 寄存器访问——请参见*第 60.4.2 节*
2	RnW	0：写请求 1：读请求
4:3	A(3:2)	DP 或 AP 寄存器的地址字段（请参见*表 560* 和*表 561*）
5	奇偶校验	前面几位的单位奇偶校验
6	停止	0
7	驻留	不接受主机指令控制。必须由目标读为 1

表 557. ACK 响应

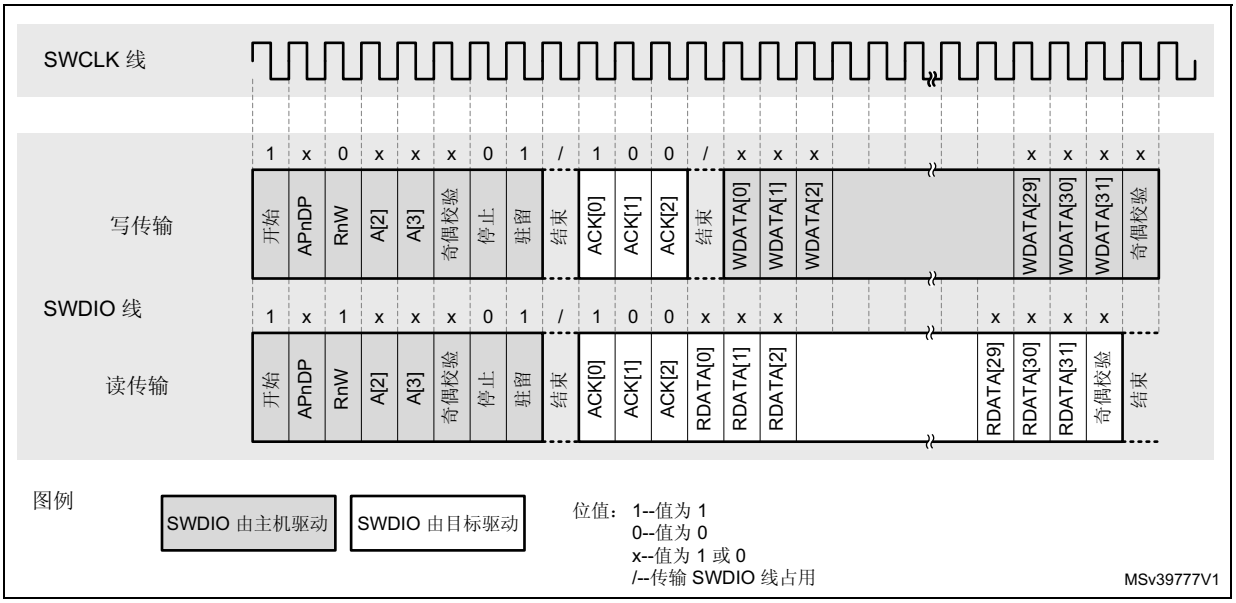
字段位	名称	说明
2:0	ACK	000b：FAULT 010b：WAIT 100b：OK

表 558. 数据传输

位字段	名称	说明
31:0	WDATA 或 RDATA	写入或读取数据
32	奇偶校验	32 个数据位的单位奇偶校验

图 812 显示了成功的写操作和读操作传输过程。

图 812. SWD 成功的数据传输过程



如果目标响应为 FAULT 或 WAIT ACK，则数据传输过程取消，除非使能了上溢检测，在这种情况下，数据会被目标忽略（写操作时）或不接受指令控制（读操作时）。

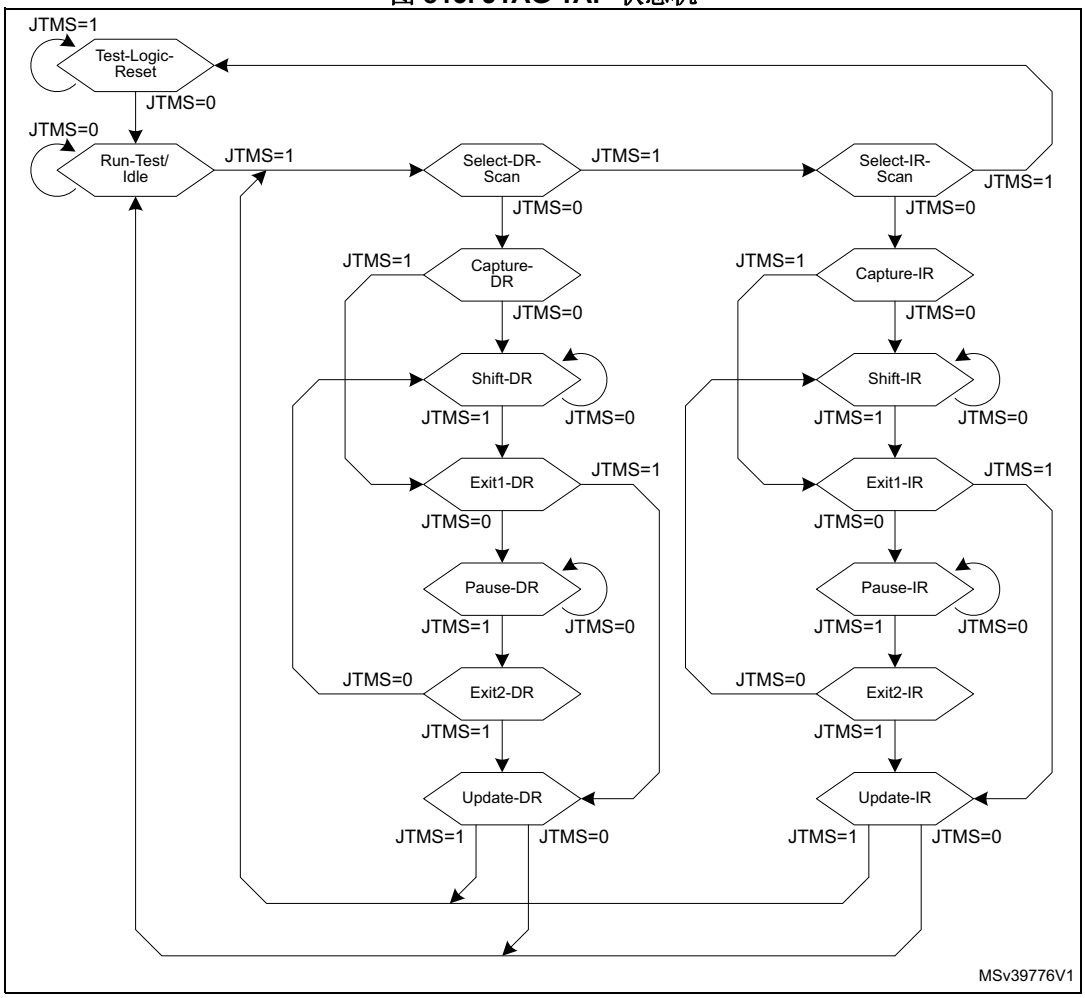
当引线第一次接通时，必须由主机生成引线复位信号，否则会出现协议错误。引线复位信号由 50 或以上个 SWDIO 为高电平的 SWCLK 周期和紧随的两个 SWDIO 为低电平的 SWCLK 周期组成。

有关串行线调试协议的更多详细信息，请参见 ARM® 调试接口架构规范 *[1]*。

注： *SWJ-DP 适用 SWD 协议版本 2。*

JTAG 调试端口

图 813. JTAG TAP 状态机



JTAG-DP 设计了一个基于 IEEE 1149.1-1990 的 TAP 状态机 (TAPSM)。状态机如*图 813* 所示。状态机控制两个扫描通道，一个与指令寄存器 (IR) 相关，一个与多个数据寄存器 (DR) 相关。

当 TAPSM 通过 Capture-IR 状态时，0b0001 被传送到指令寄存器 (IR) 扫描通道。IR 扫描通道连接在 JTDI 和 JTDO 之间。

当 TAPSM 处于 Shift-IR 状态时，IR 扫描通道在 JTCK 的每个上升沿移位一位。这意味，在第一个节拍上：

- IR 扫描通道的 LSB 是 JTDO 上的输出。
- IR 扫描通道的位 [n] 被传送到位 [n-1]。
- JTDI 上的值会传送到 IR 扫描通道的 MSB 中。

当 TAPSM 通过 Update-IR 状态时，扫描进 IR 扫描通道中的值被传送到指令寄存器。

当 TAPSM 通过 Capture-DR 状态时，值从其中一个数据寄存器传送到其中一个 DR 扫描通道，DR 扫描通道连接在 JTDI 和 JTDO 之间。

指令寄存器中保存的值决定了数据寄存器和相关的 DR 扫描通道的选择。

当 TAPSM 处于 Shift-DR 状态时，数据进行移位，与处于 Shift-IR 状态时的 IR 移位情形相同。

当 TAPSM 通过 Update-DR 状态时，扫描进 DR 扫描通道中的值被传送到所选择的数据寄存器中。

当 TAPSM 处于 Run-Test/Idle 状态时，不会发生特殊操作。IDCODE 指令加载在 IR 中。

nJTRST 信号在激活后将状态机异步重置为 Test-Logic-Reset 状态。

表 559 中列出与 4 位 IR 指令对应的数据寄存器。

表 559. JTAG-DP 数据寄存器

指令寄存器	数据寄存器	扫描通道长度	说明
0000 到 0111	(BYPASS)	1	保留：选择 BYPASS
1000	ABORT	35	**中止寄存器** – 位 31:1 = 保留 – 位 0 = APABORT：写入 1 以生成 AP 中止。
1001	(BYPASS)	1	保留：选择 BYPASS
1010	DPACC	35	**调试接口寄存器** 初始化调试端口，并允许访问调试端口寄存器。 – 传输 IN 数据时： 位 34:3 = DATA[31:0] = 为写请求传输的 32 位数据。 位 2:1 = A[3:2] = 调试端口寄存器的 2 位地址。 位 0 = RnW = 读请求 (1) 或写请求 (0)。 – 传输 OUT 数据时： 位 34:3 = DATA[31:0] = 读请求后读取的 32 位数据。 位 2:0 = ACK[2:0] = 3 位确认： 010b = OK/FAULT 001b = WAIT 其它 = 保留
1011	APACC	35	**存取接口寄存器** 初始化存取接口并允许访问存取接口寄存器。 – 传输 IN 数据时： 位 34:3 = DATA[31:0] = 为写请求移入的 32 位数据。 位 2:1 = A[3:2] = 访问端口寄存器的 2 位子地址。 位 0 = RnW = 读请求 (1) 或写请求 (0)。 – 传输 OUT 数据时： 位 34:3 = DATA[31:0] = 读请求后读取的 32 位数据。 位 2:0 = ACK[2:0] = 3 位确认： 010b = OK/FAULT 001b = WAIT 其它 = 保留
1100	(BYPASS)	1	保留：选择 BYPASS
1101	(BYPASS)	1	保留：选择 BYPASS
1110	IDCODE	32	**ID CODE** 0x05BA 0477：ARM® JTAG 调试端口 ID 代码
1111	BYPASS	1	**BYPASS** 在 JTDI 和 JTDO 之间插入单个 JTCK 周期延迟

有关 DR 寄存器的详细说明，请参见《ARM® 调试接口架构规范》*[1]*。

调试端口寄存器

SW-DP 和 JTAG-DP 都可以访问调试端口 (DP) 寄存器。*表 560* 中列出了这些寄存器。

调试器可访问 DP 寄存器，如下所示：

1. 对 DP 中的 SELECT 寄存器 DPBANKSEL 字段编程，选择要访问的寄存器存储区域（请参见*表 560*）
2. 如果使用 JTAG，则使用存储区域中的寄存器地址对 DPACC 寄存器中的 A(3:2) 字段编程。对 R/W 位编程，选择读操作或写操作。如果选择写操作，则使用写入数据对 DATA 字段进行编程。如果使用 SWD，则 A(3:2) 和 R/W 字段包含在发送到 SW-DP 的数据包请求字当中，数据包请求字的 APnDP 位为零（请参见*表 556*）。写入数据在数据阶段发送。

表 560. 调试端口寄存器

地址	A(3:2) 字段值	R/W	说明
0x0	00	R	DP_DPIDR 寄存器。包含调试端口的 IDCODE。
		W	DP_ABORT 寄存器(1)。中止当前的 AP 事务。该寄存器还用于清除 DP_CTRL/STAT 寄存器中的错误标志。
0x4	01	R/W	当 DPBANKSEL[3:0] = 0x0（DP_SELECT 寄存器）时： CTRL/STAT 寄存器。控制 DP 并提供状态信息。
			当 DPBANKSEL[3:0] = 0x1（DP_SELECT 寄存器）时： DP_DLCR 寄存器(2)。控制 SWD 数据链路的工作模式。
			当 DPBANKSEL[3:0] = 0x2（DP_SELECT 寄存器）时： DP_TARGETID 寄存器： 提供目标标识信息。
			当 DPBANKSEL[3:0] = 0x3（DP_SELECT 寄存器）时： DLPIDR 寄存器(2)。提供 SWD 协议版本。
0x8	10	R	RESEND 寄存器(2)。返回由最后一个 AP 读操作或 DP_RDBUFF 读操作返回的值，在读传输损坏情况下使用。
		W	DP_SELECT 寄存器。选择访问端口、访问端口寄存器存储区域和地址为 0x4 的 DP 寄存器。
0xC	11	R	DP_RDBUFF 寄存器 通过 JTAG-DP，用于允许调试器在执行一系列操作后获取最后结果（无需请求新的 JTAG-DP 操作）。 通过 SW-DP，包含以前的 AP 读访问的结果，从而避免了重新进行 AP 访问。
		W	DP_TARGETSEL 寄存器(2)。在引线复位序列后立即对 DP_TARGETSEL 执行写入操作时，如果同时满足以下两个条件，则目标就会被选择： – 位 [31:28] 与 DP_DLPIDR 寄存器中的位 [31:28] 匹配。 – 位 [27:0] 与 DP_TARGETID 寄存器中的位 [27:0] 匹配。 写入其他任意值取消选择目标。调试工具必须写入 0xFFFFFFFF 以取消选择所有目标。这是一个无效的 DP_TARGETID 值。所有其它无效的 DP_TARGETID 值都被保留。

1. 使用 ABORT 指令实现从 JTAG-DP 访问 AP ABORT 寄存器。
2. 只能通过 SW-DP 访问。通过 JTAG-DP “保留” 寄存器。

调试端口标识寄存器 (DP_DPIDR)

Debug port identification register

偏移地址：0x00

复位值：0x6BA0 2477

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
REVISION[3:0]				PARTNO[7:0]								Res.	Res.	Res.	MIN
r	r	r	r	r	r	r	r	r	r	r	r				r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VERSION[3:0]				DESIGNER[10:0]											Res.
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	

位 31:28 **REVISION[3:0]**：版本代码 (Revision code)
0x6

位 27:20 **PARTNO[7:0]**：调试端口的产品编号 (Part number for the debug port)
0xBA

位 19:17 保留，必须保持复位值。

位 16 **MIN**：最少调试端口 (MINDP) 设计 (Minimal debug port (MINDP) implementation)
0：未设计 MINDP（支持事务计数器和推入操作）

位 15:12 **VERSION[3:0]**：DP 架构版本 (DP architecture version)
0x2：DPv2

位 11:1 **DESIGNER[10:0]**：JEDEC 设计人员身份代码 (JEDEC designer identity code)
0x23B：ARM®

位 0 保留，必须保持复位值。

调试端口中止寄存器 (DP_ABORT)

Debug port abort register

偏移地址：0x0

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	ORUNERRCLR	WDERRCLR	STKERRCLR	STKCMPCLR	DAPABORT
											w	w	w	w	w

位 31:5 保留，必须保持复位值。

位 4 **ORUNERRCLR**：上溢错误清零位 (Overrun error clear bit)
0：保留
1：将 CTRL/STAT 寄存器的 STICKYORUN 位清零

位 3 **WDERRCLR**：写入数据错误清零位 (Write data error clear bit)
0：保留
1：将 CTRL/STAT 寄存器的 WDATAERR 位清零

位 2 **STKERRCLR**：粘滞错误清零位 (Sticky error clear bit)
0：保留
1：将 CTRL/STAT 寄存器的 STICKYERR 位清零

位 1 **STKCMPCLR**：粘滞比较清零位 (Sticky compare clear bit)
0：保留
1：将 CTRL/STAT 寄存器的 STICKYCMP 位清零

位 0 **DAPABORT**：中止当前的 AP 事务 (Abort current AP transaction)
如果返回的 WAIT 响应数量过多，则事务中止，表示事务已停止。

0：保留
1：中止事务

调试端口控制/状态寄存器 (DP_CTRL/STAT)

Debug port control/status register

偏移地址：0x4

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CSYSPWRUPACK	CSYSPWRUPREQ	CDBGPWRUPACK	CDBGPWRUPREQ	CDBGRSTACK	CDBGRSTREQ	Res.	Res.	TRNCNT[11:4]							
r	rw	r	rw	r	rw			rw	rw	rw	rw	rw	rw	rw	rw
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TRNCNT[3:0]				MASKLANE[3:0]				WDATAERR	READOK	STICKYERR	ORUNERRCLR	WDERRCLR	STKERRCLR	STKCMPCLR	DAPABORT
rw	rw	rw	rw	rw	rw	rw	rw	r	r	rw	w	w	w	w	w

位 31 **CSYSPWRUPACK**：系统域上电状态位 (System domain power-up status bit)——此器件不使用

位 30 **CSYSPWRUPREQ**：系统域上电控制位 (System domain power-up control bit)——此器件不使用

位 29 **CDBGPWRUPACK**：调试域上电状态位 (Debug domain power-up status bit)
此位为只读。从电源控制器返回调试域上电确认信号的状态。

0：域掉电
1：域上电

位 28 **CDBGPWRUPREQ**：调试域上电/掉电控制位 (Debug domain power-up/down control bit)
该位控制到电源控制器的调试域上电/掉电请求信号。

0：请求掉电
1：掉电上电

位 27 **CDBGRSTACK**：调试域复位状态位 (Debug domain reset status bit)——此器件不使用

位 26 **CDBGRSTREQ**：调试域复位控制位 (Debug domain reset control bit)——此器件不使用

位 25:24 保留，必须保持复位值。

位 23:12 **TRNCNT[11:0]**：事务计数器 (Transaction counter)
若要通过 AP 将事务序列编程为增量地址，需 TRNCNT 位加载要执行的事务数。每个事务成功完成时，事务数会递减。

位 11:8 **MASKLANE[3:0]**：推入比较和推入验证屏蔽位 (Pushed-compare and pushed-verify masking bits)
该字段指出推入比较和推入验证操作（DP_CTRL/STAT 寄存器的字段 TRNMODE = 1 或 2）中要屏蔽的字节。在推入操作中，在 AP 写入事务中提供的字与目标 AP 地址中的当前值进行比较。

0b1XXX：比较中包含字节通道 3
0bX1XX：比较中包含字节通道 2
0bXX1X：比较中包含字节通道 1
0bXXX1：比较中包含字节通道 0

位 7 **WDATAERR**：SW-DP 模式下写入数据错误 (Write data error in SW-DP)
该位表示
– 写入操作的数据阶段存在奇偶校验错误或成帧错误，或
– 已被 DP 接受的写入操作被丢弃，而不是被提交给 AP。
此位为只读。通过向 DP_ABORT 寄存器的 WDERRCLR 位写入 1 可将此位复位。

0：无错误。
1：发生错误。

在 JTAG-DP 模式下，该位保留。

位 6 **READOK**：SW-DP 模式下 AP 读操作响应 (AP read response in SW-DP)
该位表示对最后一个 AP 读访问的响应。该位为只读。

0：读操作不正常
1：读操作正常

在 JTAG-DP 模式下，该位保留。

位 5 **STICKYERR**：事务错误 (Transaction error)（在 SW-DP 模式下，该位只读；在 JTAG-DP 模式下，该位可读/写）
该位表示在 AP 事务期间发生了错误。

0：无错误。
1：发生错误。

在 SW-DP 模式下，通过向 DP_ABORT 寄存器的 STKERRCLR 位写入 1 可将此位复位。在 JTAG-DP 模式下，将此位编程为 1 可将此位复位。

位 4 **STICKYCMP**：比较匹配 (Compare match)（在 SW-DP 模式下，该位只读；在 JTAG-DP 模式下，该位可读/写）

该位表示在推入操作中发生了匹配。

0：匹配（当 TRNMODE = 0x1 时）；不匹配（当 TRNMODE = 0x2 时）
1：不匹配（当 TRNMODE = 0x1 时）；匹配（当 TRNMODE = 0x2 时）

在 SW-DP 模式下，通过向 DP_ABORT 寄存器的 STKCMPCLR 位写入 1 可将此位复位。在 JTAG-DP 模式下，将此位编程为 1 可将此位复位。

位 3:2 **TRNMODE**：AP 写操作传输模式 (Transfer mode for AP write operations)

对于读操作，该字段必须设置为 0x0。

0x0：正常操作——AP 事务直接传递到 AP。
0x1：推入验证操作。DP 存储写入数据并执行目标 AP 地址的读取事务。读操作的结果与存储的数据进行比较。如果两者不匹配，则 STICKYCMP 位置 1。
0x2：推入比较操作。DP 存储写入数据并执行目标 AP 地址的读取事务。读取的结果与存储的数据进行比较。如果两者匹配，则 STICKYCMP 位置 1。
0x3：保留

在推入操作中，只将 MASKLANE 字段表示的数据字节进行比较。

位 1 **STICKYORUN**：上溢 (Overrun)（在 SW-DP 模式下，该位只读；在 JTAG-DP 模式下，该位可读/写）

该位表示发生了上溢（在完成上一个事务之前接收到新的事务）。该位仅在 ORUNDETECT 位置 1 时才置 1。

0：无上溢
1：发生上溢

在 SW-DP 模式下，通过向 ABORT 寄存器的 ORUNERRCLR 位写入 1 可将此位复位。在 JTAG-DP 模式下，通过向该位写入 1 可将此位复位。

位 0 **ORUNDETECT**：上溢检测模式使能 (Overrun detection mode enable)

0：禁止上溢检测。
1：使能上溢检测。如果发生上溢，则 STICKYORUN 位将置 1，同时后续事务将被阻止，直到 STICKYORUN 位清零。

调试端口数据链路控制寄存器 (DP_DLCR)

Debug port data link control register

偏移地址：0x4

复位值：0x0000 0040

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	TURNROUND [1:0]		Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
						rw	rw								

位 31:10 保留，必须保持复位值。

位 9:8 **TURNROUND[1:0]**：SWDIO 的三态周期 (Tristate period for SWDIO)
0X0：1 个数据位周期
0X1：2 个数据位周期
0X2：3 个数据位周期
0X3：4 个数据位周期

位 7:0 保留，必须保持复位值。

调试端口目标标识寄存器 (DP_TARGETID)

Debug port target identification register

偏移地址：0x4

复位值：0x1045 0401

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TREVISION[3:0]				TPARTNO[15:4]											
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPARTNO[3:0]				TDESIGNER[10:0]											Res.
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	

位 31:28 **TREVISION[3:0]**：目标版本 (Target revision)
0x1：版本 1

位 27:12 **TPARTNO[15:0]**：目标产品编号 (Target part number)
0x0450: STM32H7

位 11:1 **TDESIGNER[10:0]**：目标设计人员 JEDEC 代码 (Target designer JEDEC code)
0x020：STMicroelectronics

位 0 保留，必须保持复位值。

调试端口数据链路协议标识寄存器 (DP_DLPIDR)

Debug port data link protocol identification register

偏移地址：0x4

复位值：0x0000 0001

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TINSTANCE[3:0]				Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
r	r	r	r												
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PROTSVN[3:0]			
												r	r	r	r

位 31:28 **TINSTANCE[3:0]**：目标实例编号 (Target instance number)
在多点系统中，这些位定义此器件的实例编号。
0x0

位 27:4 保留，必须保持复位值。

位 3:0 **PROTSVN[3:0]**：串行线调试协议版本 (Serial Wire Debug protocol version)
0x1：版本 2

调试端口重新发送寄存器 (DP_RESEND)

Debug port resend register

偏移地址：0x8

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESEND[31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESEND[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:0 **RESEND**：最后一个 AP 读操作或 DP RDBUFF 读操作值 (Last AP read or DP RDBUFF read value)
这些位包含由最后一个 AP 读操作或 DP RDBUFF 读操作返回的值。在读传输损坏的情况下使用。

调试端口访问端口选择寄存器 (DP_SELECT)

Debug port access port select register

偏移地址：0x8

复位值：N/A

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
APSEL[3:0]				Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
w	w	w	w												
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	APBANKSEL[3:0]				DPBANKSEL[3:0]			
								w	w	w	w	w	w	w	w

位 31:28 **APSEL[3:0]**：访问端口选择位 (Access port select bits)
这些位选择下一个事务的访问端口。

0x0：AP0——Cortex-M7 调试访问端口 (AHB-AP)
0x1：AP1——D3 访问端口 (AHB-AP)
0x2：AP2——系统调试访问端口 (APB-AP)
0x3 到 0xF：保留

位 27:8 保留，必须保持复位值。

位 7:4 **APBANKSEL[3:0]**：AP 寄存器存储区域选择位 (AP register bank select bits)
这些位选择在役 AP 上的 4 字寄存器存储区域，以用于下一个事务。

位 3:0 **DPBANKSEL[3:0]**：DP 寄存器存储区域选择位 (DP register bank select bits)
这些位选择调试端口地址为 0x4 的寄存器。

0x0：CTRL/STAT 寄存器
0x1：DLCR 寄存器
0x2：TARGETID 寄存器
0x3：DLPIDR 寄存器
0x4 到 0xF：保留

调试端口读缓冲区寄存器 (DP_RDBUFF)

Debug port read buffer register

偏移地址：0xC

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RDBUFF[31:16]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RDBUFF[15:0]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:0 **RDBUFF[31:0]**：最后一个 AP 读操作值 (Last AP read value)
该字段包含最后一个 AP 读访问返回的值。有两种方法来检索 AP 读访问返回的值：

– 对同一地址执行二次读访问，这将在相应的总线上启动新的事务，或者
– 从 DP_RDBUFF 寄存器中读取由最后一个 AP 读访问返回的值，这种情况下，不会发生新的 AP 事务。

调试端口目标标识寄存器 (DP_TARGETSEL)

Debug port target identification register

偏移地址：0xC

复位值：N/A

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TINSTANCE[3:0]				TPARTNO[15:4]											
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	w
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TPARTNO[3:0]				TDESIGNER[11:0]											Res.
w	w	w	w	w	w	w	w	w	w	w	w	w	w	w	

位 31:28 **TINSTANCE[3:0]**：目标实例编号 (Target instance number)

在多点系统中，该字段定义目标器件的实例编号。必须使用与 DP_DLPIDR 寄存器中的 TINSTANCE 字段相同的值来编程，才能选择该器件。

位 27:12 **TPARTNO[15:0]**：目标产品编号 (Target part number)

该字段定义了目标器件的产品编号。必须使用与 DP_TARGETID 寄存器中的 TPARTNO 字段相同的值来编程，才能选择该器件。

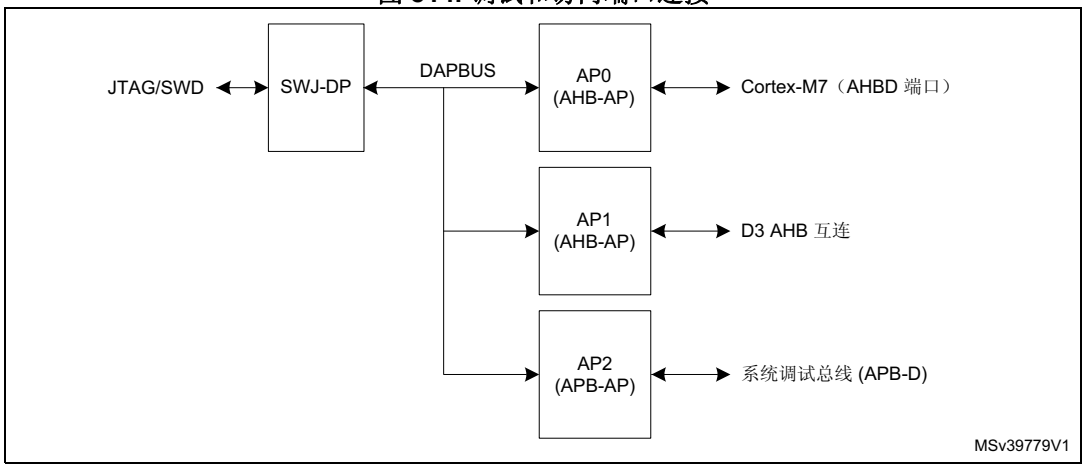
位 11:1 **TDESIGNER[10:0]**：目标设计人员 JEDEC 代码 (Target designer JEDEC code)

该字段定义了目标器件的 JEDEC 代码。必须使用与 DP_TARGETID 寄存器中的 TDESIGNER 字段相同的值来编程，才能选择该器件。

位 0 保留，必须保持复位值。

60.4.2 访问端口

图 814. 调试和访问端口连接



DP 上附有以下访问端口 (AP)：

1. AP0：Cortex-M7 访问端口 (AHB-AP)。允许通过 AHB-Lite 总线连接到处理器的 AHBD 端口来访问集成在 Cortex-M7 处理器内核中的调试和跟踪功能。
2. AP1：D3 访问端口 (AHB-AP)。允许访问 D3 域中的总线矩阵。当 D1 和 D2 域关闭时，可以看到 D3 域存储器和外设。但不能通过该端口访问 CoreSight 组件。
3. AP2：系统访问端口 (APB-AP)。允许访问系统 APB 调试总线（即所有不包含在处理器内核中的组件）上的调试和跟踪功能。

所有的访问端口都是 MEM-AP 类型，即调试和跟踪组件寄存器都映射在相关调试总线的地址空间中。调试器将 AP 视为一组 32 位寄存器，每个存储区域有四个寄存器。其中一些寄存器用于配置或监视 AP 本身，而其他寄存器则用于执行总线传输。在*表 561* 中列出了 AP 寄存器。

AP 寄存器的地址包括：

- 位 [7:4]：DP_SELECT 寄存器的 APBANKSEL 字段的内容
- 位 [3:2]：JTAG-DP 模式下 APACC 数据寄存器（请参见*表 559*）或 SW-DP 数据包请求（请参见*表 556*）中的 A(3:2) 字段，具体取决于使用的调试接口
- 位 [1:0]：始终设置为 0

DP 中 SELECT 寄存器 APSEL 字段的内容定义访问的 MEM-AP。

调试器可访问 AP 寄存器，如下所示：

1. 对 DP_SELECT 寄存器的 APSEL 字段编程，以选择其中一个 AP；对 APBANKSEL 字段编程，以选择要访问的寄存器存储区域。
2. 如果使用 JTAG，则使用存储区域中的寄存器地址对 APACC 寄存器中的 A(3:2) 字段编程。对 RnW 位编程，选择读操作或写操作。如果选择写操作，则使用写入数据对 DATA 字段进行编程。如果使用 SWD，则 A(3:2) 和 RnW 字段包含在发送到 SW-DP 的数据包请求字当中，数据包请求字的 APnDP 位置 1（请参见*表 556*）。写入数据在数据阶段发送。

调试器可通过 MEM-AP 寄存器访问存储器映射的调试组件寄存器（使用上述 AP 寄存器访问过程），如下所示：

1. 对 TAR 寄存器中的事务目标地址编程。
2. 对 CSW 寄存器编程，必要时，请使用传输参数（如 AddrInc）。
3. 针对 TAR 寄存器中保存的地址，写入或读取 DRW 寄存器，以启动总线事务。或者，读取或写入分组数据寄存器 Bdn，以触发访问地址 TAR[31:4] + n（支持无需更改 TAR 寄存器中的地址即可访问多达四个连续的地址）。

有关 MEM-AP 的更多详细信息，请参见《ARM® 调试接口架构规范》*[1]*。有关如何使用 MEM-AP 将调试端口连接到调试组件（例如处理器、ETM 或 ROM 表），请参见第 7.1.2 节。

MEM-AP寄存器

表 561. MEM-AP寄存器

地址	APBANKSEL	A(3:2)	名称	说明
0x00	0x0	0	AP_CSW	控制/状态字寄存器
0x04	0x0	1	AP_TAR	传输地址寄存器 总线事务的目标地址。
0x08	-	-	-	保留
0x0C	0x0	3	AP_DRW	数据读/写寄存器 访问该寄存器可触发调试总线上的相应事务，其目标地址为 TAR[31:0] 中的地址。
0x10	0x1	0	AP_BD0	分组数据 0 寄存器 访问该寄存器可触发调试总线上的相应事务，其目标地址为 TAR[31:4] 中的地址左移 4 位 + 0x0。
0x14	0x1	1	AP_BD1	分组数据 1 寄存器 访问该寄存器可触发调试总线上的相应事务，其目标地址为 TAR[31:4] 中的地址左移 4 位 + 0x4。
0x18	0x1	2	AP_BD2	分组数据 2 寄存器 访问该寄存器可触发调试总线上的相应事务，其目标地址为 TAR[31:4] 中的地址左移 4 位 + 0x8。
0x1C	0x1	3	AP_BD3	分组数据 3 寄存器 访问该寄存器可触发调试总线上的相应事务，其目标地址为 TAR[31:4] 中的地址左移 4 位 + 0xC。
0x20-0xEC	-	-	-	保留
0xF0	-	-	-	保留
0xF4	-	-	-	保留
0xF8	0xF	2	AP_BASE	调试基址寄存器 (RO) ROM 表的基址
0xFC	0xF	3	AP_IDR	标识寄存器 (RO)

访问端口控制/状态字寄存器 (AP_CSW)

Access port control/status word register

偏移地址：0x0

复位值：0x0000 0002 (APB-AP)、0x4000 0002 (AHB-AP)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	SPROT	Res.	PROT[4:0]					SPI STATUS	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	rw	rw	rw	rw	rw	rw	rw	r							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	MODE[3:0]				TRIN PROG	DEVIC EEN	ADDRINC[1:0]		Res.	SIZE[2:0]		
				rw	rw	rw	rw	r	r	rw	rw		rw	rw	rw

位 31 保留，必须保持复位值。

位 30 **SPROT**：加密传输请求位 (Secure transfer request bit)
在 APB-AP 中，该字段保留。在 AHB-AP 中，该字段为总线传输设置保护属性 HPROT[6]。
0：如果 SPIDEN 为高电平，加密传输。如果 SPIDEN 为低电平，非加密传输。
1：非加密传输。

位 29 保留，必须保持复位值。

位 28:24 **PROT[4:0]** 总线传输保护位 (Bus transfer protection bits)
在 APB-AP 中，该字段保留。在 AHB-AP 中，该字段为总线传输设置保护属性 HPROT[4:0]。
0bXXXX0：指令读取
0bXXXX1：数据访问
0bXXX0X：用户模式
0bXXX1X：特权模式
0bXX0XX：不可缓冲
0bXX1XX：可缓冲
0bX0XXX：不可缓存
0bX1XXX：可缓存
0b0XXXX：非独占
0b1XXXX：独占

位 23 **SPISTATUS**：SPIDEN 状态选项位 (Status of SPIDEN option bit)
该位确定了调试器是否可以访问加密存储器。在 APB-AP 中，该字段保留。
0：禁止加密 AHB 传输
1：允许加密 AHB 传输

位 22:12 保留，必须保持复位值。

位 11:8 **MODE[3:0]**：屏障支持使能位 (Barrier support enabled bit)
这些位定义是否支持存储器屏障操作。
0x0：不支持

位 7 **TRINPROG**：传输正在进行 (Transfer in progress)
该位表示正在 AP 上进行总线传输。
0：无正在进行的总线传输。
1：正在进行总线传输。

位 6 **DEVICEEN**：器件使能位 (Device Enable bit)
该位定义了是否可以访问 AP。

1：使能 AP 访问。

位 5:4 **ADDRINC[1:0]**：自动递增模式位 (Auto-increment mode bits)
这些位定义了在事务结束之后 TAR 地址是否会自动递增。

0x0：不自动递增
0x1：地址按事务字节（SIZE 字段）的大小递增。
0x2：使能打包传输（仅限于 AHB-AP——在 APB-AP 中，该值保留）。32 位 AP 访问将产生与编程的事务大小相对应的 1 个 32 位、2 个 16 位或 4 个 8 位总线事务。数据将相应地被打包或解包。
0x3：保留

位 3 保留，必须保持复位值。

位 2:0 **SIZE[2:0]**：下一个存储器访问事务的大小 (Size of next memory access transaction)（仅限于 AHB-AP）
0x0：字节（8 位）
0x1：半字（16 位）
0x2：字（32 位）
0x3-0x7：保留

对于 APB-AP，该字段只读并固定为 0x2（32 位）。

访问端口基址寄存器 (AP_BASE)

Access port base address register

偏移地址：0xF8

复位值：0xE00F E003 (AP0)、0x0000 0002 (AP1)、0xE00E 0003 (AP2)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
BASEADDR[19:4]															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BASEADDR[3:0]				Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FORMAT	ENTRYPRESENT
r	r	r	r											r	r

位 31:12 **BASEADDR[19:0]**：AP ROM 表基址（位 31 到 12）(Base address (bits 31 to 12) of ROM table for the AP)
由于 ROM 表必须与 4 KB 边界对齐，因此 12 个 LSB 都为零。
AP0 (Cortex-M7 AHB-AP)：0xE00FE
AP1 (D3 AHB-AP)：0x00000（无 ROM 表）
AP2（系统 APB-AP）：0xE00E0

位 11:2 保留，必须保持复位值。

位 1 **FORMAT**：基址寄存器格式 (Base address register format)
1：ARM® 调试接口 v5。

位 0 **ENTRYPRESENT**：调试组件有无状态位 (Debug component present status bit)
该位表示访问端口总线上有无调试组件。

0：无调试组件 (AP1)
1：有调试组件（AP0、AP2）

访问端口标识寄存器 (AP_IDR)

Access port identification register

偏移地址：0xFC

复位值：0x6477 0001（AP0 和 AP1）、0x4477 0002 (AP2)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
REVISION[3:0]				JEDECBANK[3:0]				JEDECBANK[6:0]							MEM AP
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	IDENTITY[7:0]							
								r	r	r	r	r	r	r	r

位 31:28 **REVISION[3:0]**：ARM 内核版本 (ARM core revision)
0x2：r0p3
0x4：r0p5
0x6：r0p7

位 27:24 **JEDECBANK[3:0]**：JEDEC 存储区 (JEDEC bank)
0x4：ARM®

位 23:17 **JEDECCODE[6:0]**：JEDEC 代码 (JEDEC code)
0x3B：ARM®

位 16 **MEMAP**：存储器访问端口 (Memory access port)
1：标准寄存器映射

位 15:8 保留，必须保持复位值。

位 7:0 **IDENTITY[7:0]**：AP 类型标识 (AP type identification)
0x01：AHB-AP（AP0 和 AP1）
0x02：APB-AP (AP2)
0x11：预留

60.5 跟踪和调试子系统功能描述

跟踪和调试子系统特有以下 CoreSight 组件：

- 系统 ROM 表
- 全局时间戳发生器 (TSG)
- 系统交叉触发接口 (CTI)
- 交叉触发矩阵 (CTM)
- 跟踪端口接口单元 (TPIU)
- 跟踪总线聚合器 (CSTF)
- 嵌入式跟踪 FIFO (ETF)
- 串行线输出 (SWO)
- 串行线输出跟踪聚合器 (SWTF)

调试器可通过系统 APB-AP 及其相关的 APB-D 调试总线访问这些组件。Cortex-M7 处理器也可访问这些组件。

MCU 调试单元 (DBGMCU) 也可通过 APB-D 访问，该单元不是 CoreSight 组件，其包含的寄存器用于在调试模式下配置器件的行为。

跟踪总线复制器通过跟踪总线聚合器将跟踪总线从 CPU 的 ITM CoreSight 组件分接到 ETF 和 SWO。

60.5.1 系统 ROM 表

APB-D 总线上有两个 ROM 表。ROM 表为 CoreSight 组件，包含 APB-D 总线上所有 CoreSight 组件的基址。通过这些表，调试器可以自动发现 CoreSight 组件的拓扑。

第一个表指向第二个表和位于 D3 电源域的 CoreSight 组件（SWO、SWTF 和 TSG）。DBGMCU 未被该表引用，因为它不是一个标准的 CoreSight 组件。该表占用一个 4 KB、32 位宽的 APB-D 地址空间块——从 0xE00E0000 到 0xE00E0FFC（调试器访问时）或者从 0x59000000 到 0x59000FFC（从系统总线访问时）。

表 562. 系统 ROM 表 1

ROM 表中的偏移地址	元件名称	组件基址（调试器）	组件基址（系统总线）	组件偏移地址	大小	起始
0x000	系统 ROM 表 2	0xE00F0000	0x59010000	0x10000	4 KB	0x00010003
0x004	SWO	0xE00E2000	0x59002000	0x02000	4 KB	0x00002003
0x008	时间戳发生器	0xE00E3000	0x59003000	0x03000	4 KB	0x00003003
0x00C	SWO 聚合器	0xE00E4000	0x59004000	0x04000	4 KB	0x00004003
0x010	表顶部	-	-	-	-	0x00000000
0x014 到 0xFC8	保留	-	-	-	-	0x00000000
0xFCC 到 0xFFC	ROM 表寄存器	-	-	-	-	请参见*系统 ROM 寄存器*

第二个表占用一个 4 KB、32 位宽的 APB-D 地址空间块——从 0xE00F0000 到 0xE00F0FFC（调试器访问时）或者从 0x59010000 到 0x59010FFC（从系统总线访问时）。

表 563. 系统 ROM 表 2

ROM 表中的偏移地址	元件名称	组件基址（调试器）	组件基址（系统总线）	组件偏移地址	大小	起始
0x000	系统 CTI	0xE00F1000	0x59011000	0x1000	4 KB	0x00001003
0x004	跟踪聚合器	0xE00F3000	0x59013000	0x3000	4 KB	0x00003003
0x008	ETF	0xE00F4000	0x59014000	0x4000	4 KB	0x00004003
0x00C	TPIU	0xE00F5000	0x59015000	0x5000	4 KB	0x00005003
0x010	表顶部	-	-	-	-	0x00000000
0x014 到 0xFC8	保留	-	-	-	-	0x00000000
0xFCC 到 0xFFC	ROM 表寄存器	-	-	-	-	请参见*系统 ROM 寄存器*

每个 ROM 表的顶部包含多个只读寄存器，包括标准的 CoreSight 组件和外设标识寄存器，请参见*系统 ROM 寄存器*部分。

每个调试组件占用一个或多个 4 KB 地址空间块。这个地址空间块被称为组件的调试寄存器文件。

ROM 表项的组件偏移地址指向组件地址空间的最后一个 4 KB 块的起始处。该块始终包含组件及组件的外设标识寄存器，其起始地址为块起始地址 + 偏移地址 0xFD0。4 KB 计数字段 PIDR4 [7:4] 指定了组件的 4 KB 块的数量。因此，查找组件地址空间起始地址的过程如下：

1. 首先，读取组件的 ROM 表项并从 ROM 表项的位 [31:12] 提取其偏移地址 Address_Offset[18:0]。
2. 使用该偏移地址与 ROM 表的基址 (ROM_Base_Address) 一起计算组件的基址：

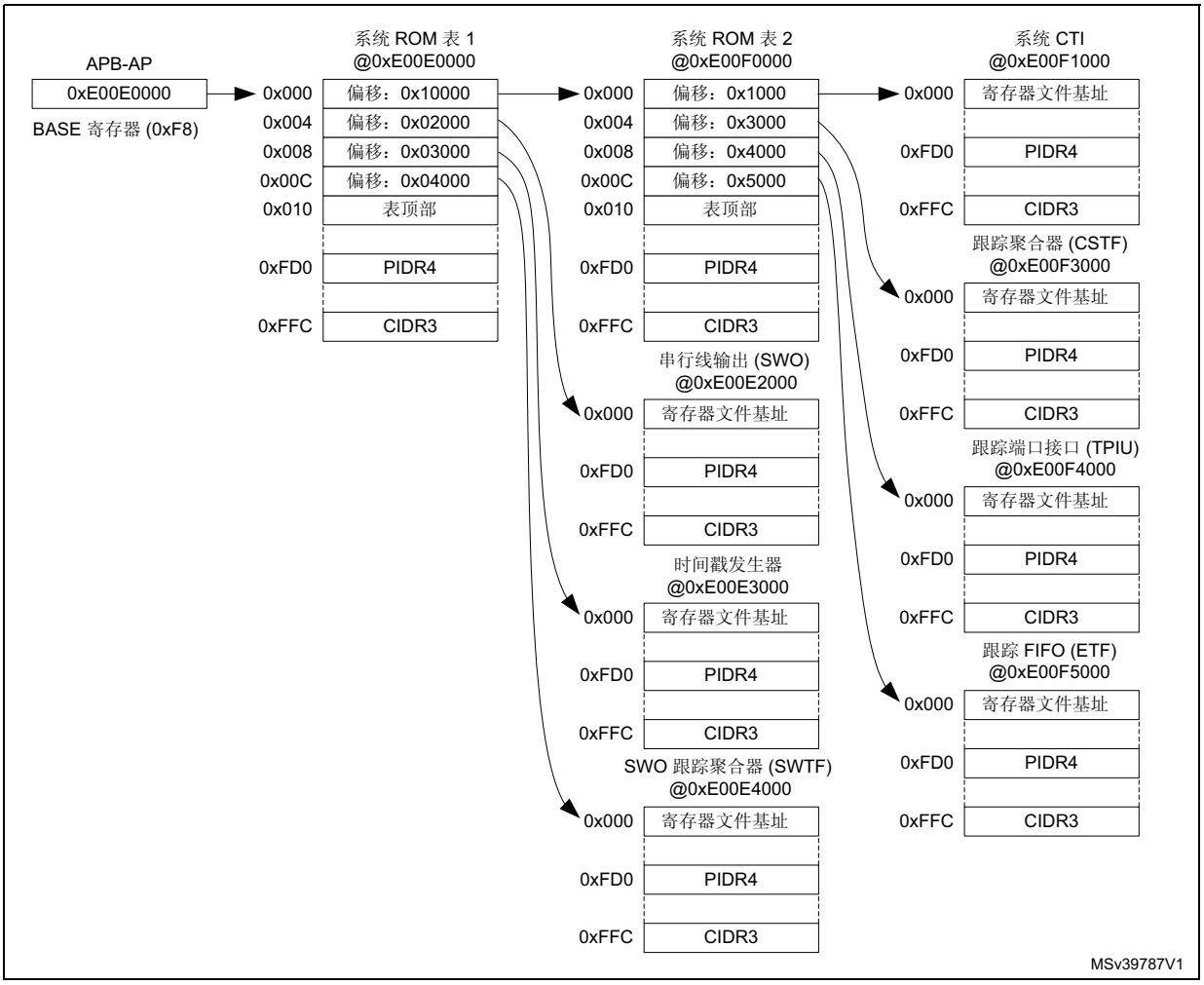
   组件基址 (Component_Base_Address) = ROM 基址 (ROM_Base_Address) + 偏移地址 (Address_Offset)

   组件基址 (Component_Base_Address) 就是组件地址空间的 4 KB 块的起始地址。
3. 读取组件的外设 ID4 寄存器。该寄存器的地址为：

   外设 ID4 寄存器地址 (Peripheral_ID4_address) = 组件基址 (Component_Base_Address) + 0xFD0。
4. 从外设 ID4 寄存器的地址值中提取 4 KB 计数字段 [7:4]。
5. 使用该 4 KB 计数字段值计算组件地址空间的起始地址。如果字段值为 0b0000（对应于计数值 1），则组件的地址空间的起始地址为步骤 2 中得到的组件基址 (Component_Base_Address)。

APB-D CoreSight 组件拓扑如*图 815* 所示。

图 815. APB-D CoreSight 组件拓扑



有关 ROM 表使用的更多信息，请参见《ARM® 调试接口架构规范》*[1]*。

系统 ROM 寄存器

SYSROM 存储器类型寄存器 (SYSROM_MEMTYPE)

SYSROM memory type register

偏移地址：0xFCC

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SYS MEM
															r

位 31:1 保留，必须保持复位值。

位 0 **SYSMEM**：系统存储器 **(System memory)**
0：该总线无系统存储器

SYSROM CoreSight 外设标识寄存器 4 (SYSROM_PIDR4)

SYSROM CoreSight peripheral identity register 4

偏移地址：0xFD0

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	4KCOUNT[3:0]				JEP106CON[3:0]			
								r				r			

位 31:8 保留，必须保持复位值。

位 7:4 **4KCOUNT[3:0]**：寄存器文件大小 (Register file size)
0x0：寄存器文件占用单个 4 KB 区域

位 3:0 **JEP106CON[3:0]**：JEP106 连续代码 (JEP106 continuation code)
0x0：意法半导体 JEDEC 连续代码

SYSROM CoreSight 外设标识寄存器 0 (SYSROM_PIDR0)

SYSROM CoreSight peripheral identity register 0

偏移地址：0xFE0

复位值：0x0000 0050

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PARTNUM[7:0]							
								r							

位 31:8 保留，必须保持复位值。

位 7:0 **PARTNUM[7:0]**：器件产品编号字段 (Device part number field)，位 [7:0]
0x50：STM32H7 器件

SYSROM CoreSight 外设标识寄存器 1 (SYSROM_PIDR1)

SYSROM CoreSight peripheral identity register 1

偏移地址：0xFE4

复位值：0x0000 0004

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	JEP106ID[3:0]				PARTNUM[11:8]			
								r				r			

位 31:8 保留，必须保持复位值。

位 7:4 **JEP106ID[3:0]**：JEP106 标识代码字段 (JEP106 identity code field)，位 [3:0]
0x0：意法半导体 JEDEC 代码

位 3:0 **PARTNUM[11:8]**：器件产品编号字段 (Device part number field)，位 [11:8]
0x4：STM32H7 器件

SYSROM CoreSight 外设标识寄存器 2 (SYSROM_PIDR2)

SYSROM CoreSight peripheral identity register 2

偏移地址：0xFE8

复位值：0x0000 000A

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REVISION[3:0]				JEDEC	JEP106ID[6:4]		
								r				r	r		

位 31:8 保留，必须保持复位值。

位 7:4 **REVISION[3:0]**：器件版本号 (Device revision number)
0x1：版本 1

位 3 **JEDEC**：JEDEC 分配的值 (JEDEC assigned value)
1：JEDEC 指定的设计人员 ID

位 2:0 **JEP106ID[6:4]**：JEP106 标识代码字段 (JEP106 identity code field)，位 [6:4]
0x2：意法半导体 JEDEC 代码

SYSROM CoreSight 外设标识寄存器 3 (SYSROM_PIDR3)

SYSROM CoreSight peripheral identity register 3

偏移地址：0xFEC

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REVAND[3:0]				CMOD[3:0]			
								r				r			

位 31:8 保留，必须保持复位值。

位 7:4 **REVAND[3:0]**：金属修复版本 (Metal fix version)
0x0：无金属修复

位 3:0 **CMOD[3:0]**：客户修改 (Customer modified)
0x0：无客户修改

SYSROM CoreSight 组件标识寄存器 0 (SYSROM_CIDR0)

SYSROM CoreSight component identity register 0

偏移地址：0xFF0

复位值：0x0000 000D

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[7:0]							
								r							

位 31:8 保留，必须保持复位值。

位 7:0 **PREAMBLE[7:0]**：组件 ID 字段 (Component ID field)，位 [7:0]
0x0D：公共 ID 值

SYSROM CoreSight 组件标识寄存器 1 (SYSROM_CIDR1)

SYSROM CoreSight component identity register 1

偏移地址：0xFF4

复位值：0x0000 0010

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CLASS[3:0]				PREAMBLE[11:8]			
								r				r			

位 31:8 保留，必须保持复位值。

位 7:4 **CLASS[3:0]**：组件 ID 字段 (Component ID field)，位 [15:12]——组件类
0x1：ROM 表组件

位 3:0 **PREAMBLE[11:8]**：组件 ID 字段 (Component ID field)，位 [11:8]
0x0：公共 ID 值

SYSROM CoreSight 组件标识寄存器 2 (SYSROM_CIDR2)

SYSROM CoreSight component identity register 2

偏移地址：0xFF8

复位值：0x0000 0005

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[19:12]							
								r							

位 31:8 保留，必须保持复位值。

位 7:0 **PREAMBLE[19:12]**：组件 ID 字段 (Component ID field)，位 [23:16]
0x05：公共 ID 值

SYSROM CoreSight 组件标识寄存器 3 (SYSROM_CIDR3)

SYSROM CoreSight component identity register 3

偏移地址：0xFFC

复位值：0x0000 00B1

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[27:20]							
								r							

位 31:8 保留，必须保持复位值。

位 7:0 **PREAMBLE**：组件 ID 字段 (Component ID field)，位 [31:24]
0xB1：公共 ID 值

系统 ROM 寄存器映射和复位值

表 564. 系统 ROM 表寄存器映射和复位值

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0xFCC	SYSROM_ MEMTYPE	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SYSMEM
	Reset value																																0
0xFD0	SYSROM_PIDR4	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	4KCOUNT[3:0]				JEP106CON[3:0]			
	Reset value																									0	0	0	0	0	0	0	0
0xFD4	SYSROM_PIDR5	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Reserved							
	Reset value																									0	0	0	0	0	0	0	0
0xFD8	SYSROM_PIDR6	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Reserved							
	Reset value																									0	0	0	0	0	0	0	0
0xFDC	SYSROM_PIDR7	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Reserved							
	Reset value																									0	0	0	0	0	0	0	0
0xFE0	SYSROM_PIDR0	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PARTNUM[7:0]							
	Reset value																									0	1	0	1	0	0	0	0
0xFE4	SYSROM_PIDR1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	JEP106ID [3:0]				PARTNUM [11:8]			
	Reset value																									0	0	0	0	0	1	0	0
0xFE8	SYSROM_PIDR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REVISION [3:0]				JEDEC	JEP106ID [6:4]		
	Reset value																									0	0	0	1	1	0	1	0
0xFEC	SYSROM_PIDR3	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REVAND [3:0]				CMOD [3:0]			
	Reset value																									0	0	0	0	0	0	0	0
0xFF0	SYSROM_CIDR0	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[7:0]							
	Reset value																									0	0	0	0	1	1	0	1
0xFF4	SYSROM_CIDR1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CLASS [3:0]				PREAMBLE [11:8]			
	Reset value																									0	0	0	1	0	0	0	0
0xFF8	SYSROM_CIDR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[19:12]							
	Reset value																									0	0	0	0	0	1	0	1
0xFFC	SYSROM_CIDR3	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[27:20]							
	Reset value																									1	0	1	1	0	0	0	1

60.5.2 全局时间戳发生器 (TSG)

全局时间戳发生器包含一个 64 位计数器，该计数器为系统中的所有跟踪源提供公共的参考时序，即处理器内核中的 ETM 和 ITM。当多个跟踪源在聚合器处被多路复用为一个数据流时，跟踪数据包的时序可能会丢失，此时，这些组件就在跟踪数据流中插入时间戳，以允许跟踪分析器恢复跟踪数据包的时序。

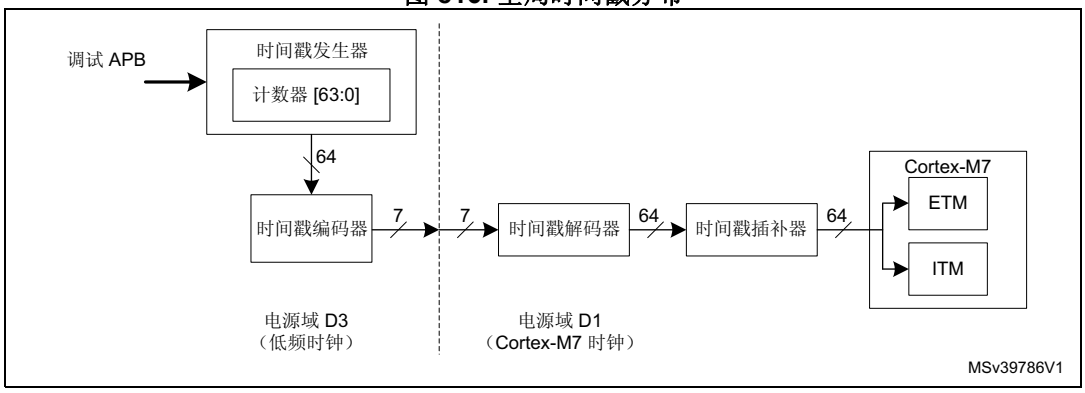
TSG 寄存器可通过 APB-D 访问，因此调试器或调试软件可以：

- 启动和停止时间戳递增
- 读取当前的时间戳值
- 更改当前的时间戳值
  - 在更改时，必须停止时间戳计数器。当时间戳值被更改时，时间戳发生器重新同步所有跟踪源。
- 更改报告的时间戳增量

有关全局时间戳发生器 CoreSight 组件的更多信息，请参见《ARM® CoreSight™ SoC-400 技术参考手册》*[2]*。

时间戳发生器位于 D3 电源域，而时间戳分布到 Cortex-M7。为了简化电源域边界上的分布，64 位时间戳采用七位编码，然后在目标电源域中解码。如果处理器时钟明显快于发生器时钟，则通过内插来提高其分辨率。时间戳分布如*图 816* 所示。

图 816. 全局时间戳分布



TSG寄存器

TSG 计数器控制寄存器 (TSG_CNTCR)

TSG counter control register

偏移地址：0x000

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	HDBG	EN
														rw	rw

位 31:2 保留，必须保持复位值。

位 1 **HDBG**：遇调试停止 (Halt on debug)

0：正常工作

1：检测到系统级调试状态时停止计数器——未设计

位 0 **EN**：使能 (Enable)

0：禁止计数器

1：使能计数器并递增计数

TSG 计数器状态寄存器 (TSG_CNTSR)

TSG counter status register

偏移地址：0x004

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DBGH	Res.
														r	

位 31:2 保留，必须保持复位值。

位 1 **DBGH**：遇调试停止 (Debug halt)

0：正常工作

1：因系统级调试状态而停止计数器

位 0 保留，必须保持复位值。

TSG 当前计数器值低位寄存器 (TSG_CNTCVL)

TSG current counter value lower register

偏移地址：0x008

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CNTCVL[31:16]															
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CNTCVL[15:0]															
rw															

位 31:0 **CNTCVL[31:0]**：TSG 当前计数器值字段 (TSG current counter value field)，位 [31:0]
要更改当前的时间戳值，需先将新值的低 32 位写入该寄存器，再将高 32 位写入 CNTCVU。在写入 CNTVCVU 寄存器前，时间戳值会保持不变。注意：必须先将 TSG_CNTCR 寄存器的 EN 位清零，再执行写操作。

TSG 当前计数器值高位寄存器 (TSG_CNTCVU)

TSG current counter value upper register

偏移地址：0x00C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CNTCVU[31:16]															
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CNTCVU[15:0]															
rw															

位 31:0 **CNTCVU[31:0]**：TSG 当前计数器值字段 (TSG current counter value field)，位 [63:32]
要更改当前的时间戳值，需先将新值的低 32 位写入 CNTCVL，再将高 32 位写入该寄存器。当写入该寄存器时，64 位时间戳值会更新为两次写入的值。注意：必须先将 TSG_CNTCR 寄存器的 EN 位清零，再执行写操作。

TSG 基频 ID 寄存器 (TSG_CNTFID0)

TSG base frequency ID register

偏移地址：0x020

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
FREQ[31:16]															
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FREQ[15:0]															
rw															

位 31:0 **FREQ**：以 Hz 为单位的 TSG 寄存器的增量频率 (Increment frequency of TSG counter in Hz)
只要跟踪发生器时钟频率发生变化，就使用其对该字段编程。

TSG CoreSight 外设标识寄存器 4 (TSG_PIDR4)

TSG CoreSight peripheral identity register 4

偏移地址：0xFD0

复位值：0x0000 0004

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	4KCOUNT[3:0]				JEP106CON[3:0]			
								r				r			

位 31:8 保留，必须保持复位值。

位 7:4 **4KCOUNT[3:0]**：寄存器文件大小 (Register file size)
0x0：寄存器文件占用单个 4 KB 区域

位 3:0 **JEP106CON[3:0]**：JEP106 连续代码 (JEP106 continuation code)
0x4：ARM® JEDEC 代码

TSG CoreSight 外设标识寄存器 0 (TSG_PIDR0)

TSG CoreSight Peripheral identity register 0

偏移地址：0xFE0

复位值：0x0000 0001

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PARTNUM[7:0]							
								r							

位 31:8 保留，必须保持复位值。

位 7:0 **PARTNUM[7:0]**：产品编号字段 (Part number field)，位 [7:0]
0x01：TSG 产品编号

TSG CoreSight 外设标识寄存器 1 (TSG_PIDR1)

TSG CoreSight peripheral identity register 1

偏移地址：0xFE4

复位值：0x0000 00B1

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	JEP106ID[3:0]				PARTNUM[11:8]			
								r				r			

位 31:8 保留，必须保持复位值。

位 7:4 **JEP106ID[3:0]**：JEP106 标识代码字段 (JEP106 identity code field)，位 [3:0]
0xB：ARM® JEDEC 代码

位 3:0 **PARTNUM[11:8]**：产品编号字段 (Part number field)，位 [11:8]
0x1：TSG 产品编号

TSG CoreSight 外设标识寄存器 2 (TSG_PIDR2)

TSG CoreSight peripheral identity register 2

偏移地址：0xFE8

复位值：0x0000 001B

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REVISION[3:0]				JEDEC	JEP106ID[6:4]		
								r				r	r		

位 31:8 保留，必须保持复位值。

位 7:4 **REVISION[3:0]**：组件版本号 (Component revision number)
0x1：r0p1

位 3 **JEDEC**：JEDEC 分配的值 (JEDEC assigned value)
1：JEDEC 指定的设计人员 ID

位 2:0 **JEP106ID[6:4]**：JEP106 标识代码字段 (JEP106 identity code field)，位 [6:4]
0x3：ARM® JEDEC 代码

TSG CoreSight 外设标识寄存器 3 (TSG_PIDR3)

TSG CoreSight peripheral identity register 3

偏移地址：0xFEC

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REVAND[3:0]				CMOD[3:0]			
								r				r			

位 31:8 保留，必须保持复位值。

位 7:4 **REVAND[3:0]**：金属修复版本 (Metal fix version)
0x0：无金属修复

位 3:0 **CMOD[3:0]**：客户修改 (Customer modified)
0x0：无客户修改

TSG CoreSight 组件标识寄存器 0 (TSG_CIDR0)

TSG CoreSight component identity register 0

偏移地址：0xFF0

复位值：0x0000 000D

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[7:0]							
								r							

位 31:8 保留，必须保持复位值。

位 7:0 **PREAMBLE[7:0]**：组件 ID 字段 (Component ID field)，位 [7:0]
0x0D：公共 ID 值

TSG CoreSight 组件标识寄存器 1 (TSG_CIDR1)

TSG CoreSight component identity register 1

偏移地址：0xFF4

复位值：0x0000 00F0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CLASS[3:0]				PREAMBLE[11:8]			
								r				r			

位 31:8 保留，必须保持复位值。

位 7:4 **CLASS[3:0]**：组件 ID 字段 (Component ID field)，位 [15:12]——组件类
0xF：CoreSight Soc-400 组件

位 3:0 **PREAMBLE[11:8]**：组件 ID 字段 (Component ID field)，位 [11:8]
0x0：公共 ID 值

TSG CoreSight 组件标识寄存器 2 (TSG_CIDR2)

TSG CoreSight component identity register 2

偏移地址：0xFF8

复位值：0x0000 0005

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[19:12]							
								r							

位 31:8 保留，必须保持复位值。

位 7:0 **PREAMBLE[19:12]**：组件 ID 字段 (Component ID field)，位 [23:16]
0x05：公共 ID 值

TSG CoreSight 组件标识寄存器 3 (TSG_CIDR3)

TSG CoreSight component identity register 3

偏移地址：0xFFC

复位值：0x0000 00B1

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[27:20]							
								r							

位 31:8 保留，必须保持复位值。

位 7:0 **PREAMBLE[27:20]**：组件 ID 字段 (Component ID field)，位 [31:24]
0xB1：公共 ID 值

TSG 寄存器映射和复位值

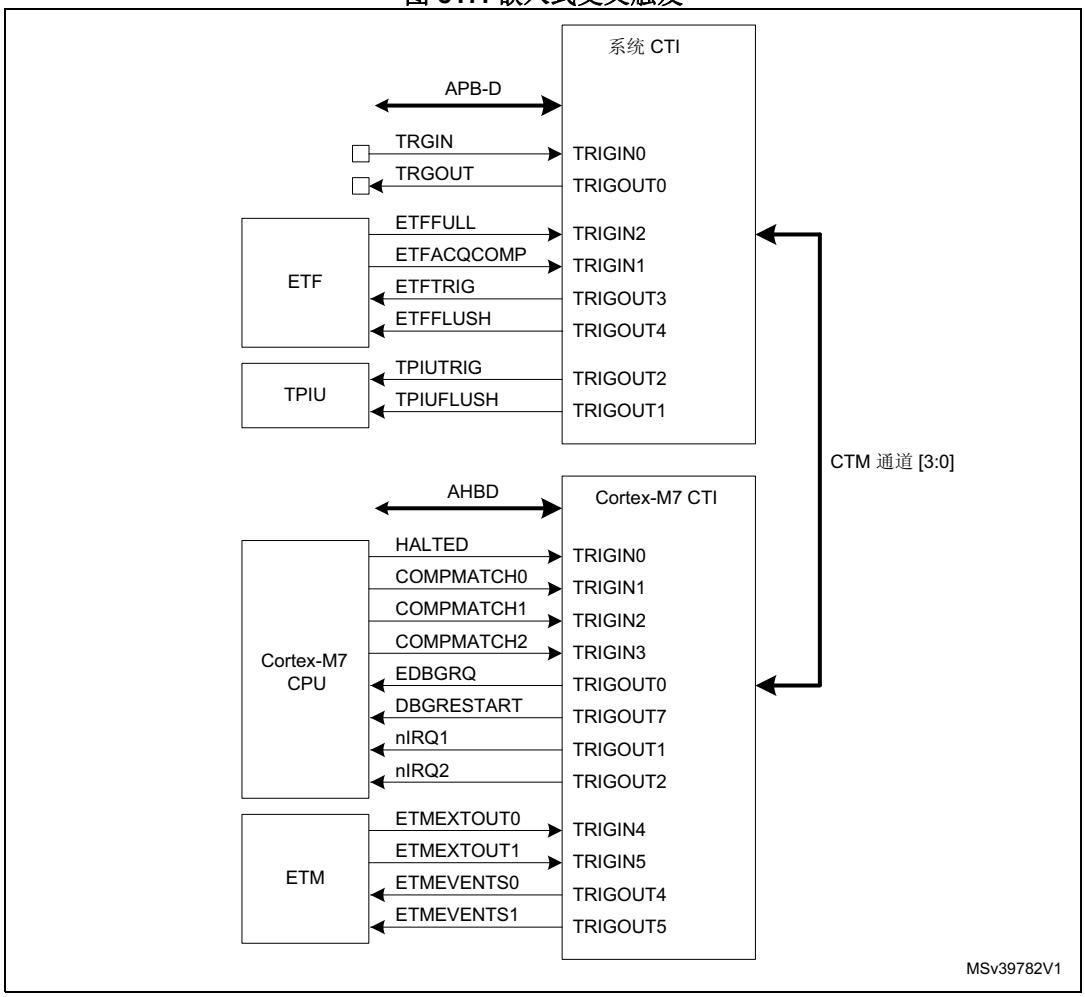
表 565. TSG 寄存器映射和复位值

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x000	TSG_CNTCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	HDBG	EN
	Reset value																															0	0
0x004	TSG_CNTSR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DBGH	Res.
	Reset value																															0	
0x008	TSG_CNTCVL	CNTCVL_L_32																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x00C	TSG_CNTCVU	CNTCVU_U_32																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x020	TSG_CNTFID0	FREQ																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0xFD0	TSG_PIDR4	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	4KCOUNT				JEP106CON			
	Reset value																									0	0	0	0	0	1	0	0
0xFD4	TSG_PIDR5	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	Reset value																																
0xFD8	TSG_PIDR6	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	Reset value																																
0xFDC	TSG_PIDR7	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	Reset value																																
0xFE0	TSG_PIDR0	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PARTNUM[7:0]							
	Reset value																									0	1	1	0	0	0	0	1
0xFE4	TSG_PIDR1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	JEP106ID [3:0]				PARTNUM [11:8]			
	Reset value																									1	0	1	1	1	0	0	1
0xFE8	TSG_PIDR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REVISION [3:0]				JEDEC	JEP106ID [6:4]		
	Reset value																									0	0	0	1	1	0	1	1
0xFEC	TSG_PIDR3	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REVAND [3:0]				CMOD [3:0]			
	Reset value																									0	0	0	0	0	0	0	0
0xFF0	TSG_CIDR0	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[7:0]							
	Reset value																									0	0	0	0	1	1	0	1
0xFF4	TSG_CIDR1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CLASS [3:0]				PREAMBLE [11:8]			
	Reset value																									1	0	0	1	0	0	0	0
0xFF8	TSG_CIDR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[19:12]							
	Reset value																									0	0	0	0	0	1	0	1
0xFFC	TSG_CIDR3	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[27:20]							
	Reset value																									1	0	1	1	0	0	0	1

60.5.3 交叉触发接口 (CTI) 和矩阵 (CTM)

交叉触发接口 (CTI) 和交叉触发矩阵 (CTM) 一起形成了 CoreSight 嵌入式交叉触发功能。有两个 CTI 组件，一个为系统级，另一个专用于 Cortex-M7。这两个CTI 通过 CTM 相互相连。调试器可以通过系统访问端口及相关的 APB-D 访问系统级 CTI。Cortex-M7 CTI 物理集成在 Cortex-M7 内核中，可以通过 Cortex-M7 访问端口及相关的 AHBD 访问。

图 817. 嵌入式交叉触发



CTI 允许来自各种来源的事件触发调试和/或跟踪活动。例如，如果检测到外部触发输入信号发生变换，则可以启动代码跟踪。

每个 CTI 最多拥有 8 个触发输入和 8 个触发输出。任何输入都可以连接到同一个 CTI 的任何输出，也可通过 CTM 连接到另一个 CTI 上的任何输出。

表 566 到*表 569* 中列出了各个 CTI 的触发输入和输出信号。

表 566. 系统 CTI 输入

#	源信号	源组件	注释
0	DBTRIGI	GPIO	外部触发输入——允许外部信号生成调试事件
1	ETFACQCOMP	ETF	ETF 捕获完成——当跟踪 FIFO 为空时，允许生成调试事件
2	ETFFULL	ETF	ETF 已满标志——当跟踪 FIFO 已满时，允许生成调试事件
3	-	-	未使用
4	-	-	未使用
5	-	-	未使用
6	-	-	未使用
7	-	-	未使用

表 567. 系统 CTI 输出

#	输出信号	目标组件	注释
0	DBTRIGO	GPIO	外部 IO 触发输出——允许监视外部 DBTRIGO 引脚上的事件
1	TPIUFLUSH	TPIU	跟踪端口刷新触发——使 TPIU FIFO 被刷新
2	TPIUTRIG	TPIU	跟踪端口使能触发——启动外部跟踪端口上的跟踪输出
3	ETFTRIG	ETF	ETF 使能触发——开始填充跟踪 FIFO
4	ETFFLUSH	ETF	ETF 刷新触发——使跟踪 FIFO 被刷新
5	-	-	未使用
6	-	-	未使用
7	-	-	未使用

表 568. Cortex-M7 CTI 输入

#	源信号	源组件	注释
0	HALTED	Cortex-M7 CPU	CPU 停止——指示 CPU 处于调试模式
1	COMPMATCH0	Cortex-M7 DWT	DWT 比较器 0 匹配
2	COMPMATCH1	Cortex-M7 DWT	DWT 比较器 1 匹配
3	COMPMATCH2	Cortex-M7 DWT	DWT 比较器 2 匹配
4	ETMEXTOUT0	Cortex-M7 ETM	ETM 外部触发输出
5	ETMEXTOUT1	Cortex-M7 ETM	ETM 外部触发输出
6	-	-	未使用
7	-	-	未使用

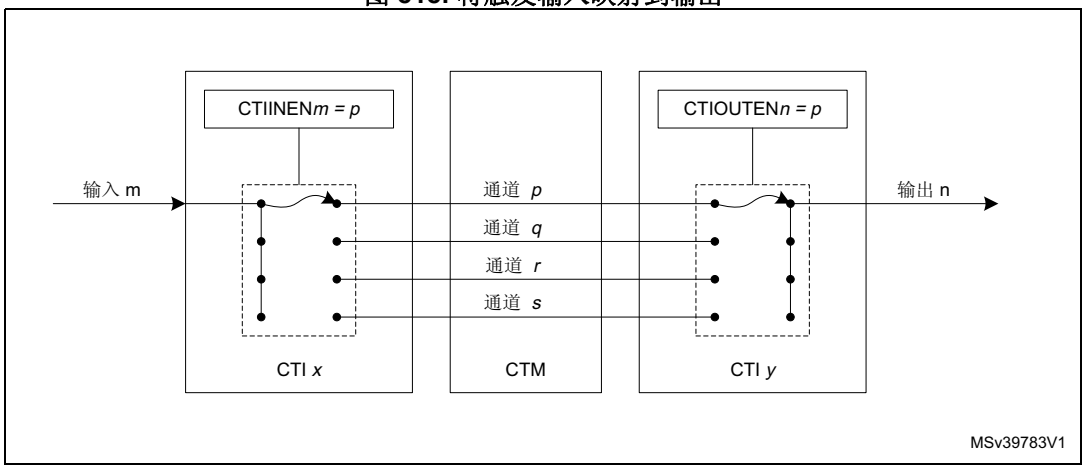
表 569. Cortex-M7 CTI 输出

#	输出信号	目标组件	注释
0	EDBGRQ	Cortex-M7 CPU	CPU 停止请求——将 CPU 置于调试模式
1	nIRQ1	Cortex-M7 NVIC	中断请求
2	nIRQ2	Cortex-M7 NVIC	中断请求
3	-	-	未使用
4	ETMEVENTS0	Cortex-M7 ETM	ETM 触发请求——使能 CPU 执行跟踪
5	ETMEVENTS1	Cortex-M7 ETM	ETM 触发请求——使能 CPU 执行跟踪
6	-	-	未使用
7	DBGRESTART	Cortex-M7 CPU	CPU 重启请求——CPU 退出调试模式

交叉触发矩阵中有四个事件通道，在不同 CTI 的触发输入和输出之间，最多支持四个并行双向连接。要将 CTI *x* 上的输入编号 *m* 连接到 CTI *y* 上的输出编号 *n*，则必须使用 CTI *x* 的 CTIINEN*m* 寄存器将输入连接到事件通道 *p*。同一通道 *p* 必须使用 CTI *y* 的 CTIOUTEN*n* 寄存器连接到输出。注意：这同样适用于同一个 CTI 的输入和输出之间的连接。

一个输入可连接到多个通道（最多四个），因此一个输入可被发送到多个输出。类似地，一个输出可连接到多个输入。还可以将多个输入/输出连接到同一通道。

图 818. 将触发输入映射到输出



有关交叉触发接口 CoreSight 组件的更多信息，请参见《ARM® CoreSight™ SoC-400 技术参考手册》*[2]*。

CTI 寄存器

每个 CTI 的寄存器文件基址通过其所连接总线的 ROM 表定义。每个 CTI 的寄存器是相同的。

CTI 控制寄存器 (CTI_CONTROL)

CTI control register

偏移地址：0x000

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	GLBEN
															rw

位 31:1 保留，必须保持复位值。

位 0 **GLBEN**：全局使能 (Global enable)。

0：禁止交叉触发

1：使能交叉触发

CTI 触发确认寄存器 (CTI_INTACK)

CTI trigger acknowledge register

偏移地址：0x010

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	INTACK[7:0]							
								rw							

位 31:8 保留，必须保持复位值。

位 7:0 **INTACK[7:0]**：触发确认 (Trigger acknowledge)

每个 CTITRIGOUT 输出都使用一个寄存器位。当向该寄存器的某位写入 1 时，则确认相应的 CTITRIGOUT 输出，使其被清零。

CTI 应用程序触发设置寄存器 (CTI_APPSET)

CTI application trigger set register

偏移地址：0x014

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	APPSET[3:0]			
												rw			

位 31:4 保留，必须保持复位值。

位 3:0 **APPSET[3:0]**：设置通道事件 (Set channel event)

读：
0bXXX0：通道 0 事件处于未激活状态
0bXXX0：通道 0 事件处于激活状态
0bXX0X：通道 1 事件处于未激活状态
0bXX1X：通道 1 事件处于激活状态
0bX0XX：通道 2 事件处于未激活状态
0bX1XX：通道 2 事件处于激活状态
0b0XXX：通道 3 事件处于未激活状态
0b1XXX：通道 3 事件处于激活状态

写：
0bXXX0：无影响
0bXXX0：设置通道 0 事件
0bXX0X：无影响
0bXX1X：设置通道 1 事件
0bX0XX：无影响
0bX1XX：设置通道 2 事件
0b0XXX：无影响
0b1XXX：设置通道 3 事件

CTI 应用程序触发清零寄存器 (CTI_APPCLEAR)

CTI application trigger clear register

偏移地址：0x018

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	APPCLEAR[3:0]			
												w			

位 31:4 保留，必须保持复位值。

位 3:0 **APPCLEAR[3:0]**：清除通道事件 (Clear channel event)

0bXXX0：无影响
0bXXX0：清除通道 0 事件
0bXX0X：无影响
0bXX1X：清除通道 1 事件
0bX0XX：无影响
0bX1XX：清除通道 2 事件
0b0XXX：无影响
0b1XXX：清除通道 3 事件

CTI 应用程序脉冲寄存器 (CTI_APPPULSE)

CTI application pulse register

偏移地址：0x01C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	APPPULSE[3:0]			
												w			

位 31:4 保留，必须保持复位值。

位 3:0 **APPULSE**：脉冲通道事件 (Pulse channel event)

该寄存器立即将自己清零。

0bXXX0：无影响
0bXXX0：生成通道 0 脉冲
0bXX0X：无影响
0bXX1X：生成通道 1 脉冲
0bX0XX：无影响
0bX1XX：生成通道 2 脉冲
0b0XXX：无影响
0b1XXX：生成通道 3 脉冲

CTI 触发 IN x 使能寄存器 (CTI_INENx)

CTI trigger IN x enable register

偏移地址：0x020 + 4 * x，其中 x = 0 到 7

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TRIGINEN[3:0]			
												rw			

位 31:4 保留，必须保持复位值。

位 3:0 **TRIGINEN[3:0]**：交叉触发事件使能 (Cross-trigger event enable)

当 CTITRIGINx（x = 0 到 7）处于激活状态时，使能或禁止四个通道中的每个通道上的交叉触发事件。

0bXXX0：触发 n 不生成通道 0 事件
0bXXX0：触发 n 生成通道 0 事件
0bXX0X：触发 n 不生成通道 1 事件
0bXX1X：触发 n 生成通道 1 事件
0bX0XX：触发 n 不生成通道 2 事件
0bX1XX：触发 n 生成通道 2 事件
0b0XXX：触发 n 不生成通道 3 事件
0b1XXX：触发 n 生成通道 3 事件

CTI 触发 OUT x 使能寄存器 (CTI_OUTENx)

CTI trigger OUT x enable register

偏移地址：0x0A0 + 4 * x，其中 x = 0 到 7

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.					TRIGOUTEN[3:0]			
												rw			

位 31:4 保留，必须保持复位值。

位 3:0 **TRIGOUTEN[3:0]**：使能事件触发 (Enable trigger upon event)
对于每个通道，此字段定义了该通道上的事件是否能够生成 CTITRIGOUTx（x = 0 到 7）触发。

0bXXX0：通道 0 事件不生成触发输出 n 触发
0bXXX0：通道 0 事件生成触发输出 n 触发
0bXX0X：通道 1 事件不生成触发输出 n 触发
0bXX1X：通道 1 事件生成触发输出 n 触发
0bX0XX：通道 2 事件不生成触发输出 n 触发
0bX1XX：通道 2 事件生成触发输出 n 触发
0b0XXX：通道 3 事件不生成触发输出 n 触发
0b1XXX：通道 3 事件生成触发输出 n 触发

CTI 触发 IN 状态寄存器 (CTI_TRGISTS)

CTI trigger IN status register

偏移地址：0x130

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TRIGINSTATUS[7:0]							
								r							

位 31:8 保留，必须保持复位值。

位 7:0 **TRIGINSTATUS[7:0]**：触发输入状态 (Trigger input status)
每个 CTITRIGIN 输入都使用一个寄存器位。如果某位设置为 1，则表示相应触发输入处于激活状态。如果某位设置为 0，则表示相应触发输入处于未激活状态。

CTI 触发 OUT 状态寄存器 (CTI_TRGOSTS)

CTI trigger OUT status register

偏移地址：0x134

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TRIGOUTSTATUS[7:0]							
								r							

位 31:8 保留，必须保持复位值。

位 7:0 **TRIGOUTSTATUS[7:0]**：触发输出状态 (Trigger output status)
每个 CTITRIGOUT 输出都使用一个寄存器位。如果某位设置为 1，则表示相应触发输出处于激活状态。如果某位设置为 0，则表示相应触发输出处于未激活状态。

CTI 通道 IN 状态寄存器 (CTI_CHINSTS)

CTI channel IN status register

偏移地址：0x138

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CHINSTATUS[3:0]			
												r			

位 31:4 保留，必须保持复位值。

位 3:0 **CHINSTATUS[3:0]**：通道输入状态 (Channel input status)

每个通道输入都使用一个寄存器位。如果某位设置为 1，则表示相应通道输入处于激活状态。如果某位设置为 0，则表示相应通道输入处于未激活状态。

CTI 通道 OUT 状态寄存器 (CTI_CHOUTSTS)

CTI channel OUT status register

偏移地址：0x13C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CHOUTSTATUS[3:0]			
												r			

位 31:4 保留，必须保持复位值。

位 3:0 **CHOUTSTATUS[3:0]**：通道输出状态 (Channel output status)

每个通道输出都使用一个寄存器位。如果某位设置为 1，则表示相应通道输出处于激活状态。如果某位设置为 0，则表示相应通道输出处于未激活状态。

CTI 通道门控寄存器 (CTI_GATE)

CTI channel gate register

偏移地址：0x140

复位值：0x0000 000F

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	GATEEN[3:0]			
												rw			

位 31:4 保留，必须保持复位值。

位 3:0 **GATEEN[3:0]**：通道输出使能 (Channel output enable)

对于每个通道，定义了该通道事件能否通过 CTM 传播到其他 CTI。

0bXXX0：通道 0 事件不能传播
0bXXX0：通道 0 事件能够传播
0bXX0X：通道 1 事件不能传播
0bXX1X：通道 1 事件能够传播
0bX0XX：通道 2 事件不能传播
0bX1XX：通道 2 事件能够传播
0b0XXX：通道 3 事件不能传播
0b1XXX：通道 3 事件能够传播

将 CTI 声明标记置位寄存器 (CTI_CLAIMSET)

CTI claim tag set register

偏移地址：0xFA0

复位值：0x0000 000F

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CLAIMSET[3:0]			
												rw			

位 31:4 保留，必须保持复位值。

位 3:0 **CLAIMSET[3:0]**：置位声明标记位 (Set claim tag bits)

写：
0000：无影响
xxx1：将位 0 置 1
xx1x：将位 1 置 1
x1xx：将位 2 置 1
1xxx：将位 3 置 1

读：
0xF：表示声明标记使用四个位

CTI 声明标记清零寄存器 (CTI_CLAIMCLR)

CTI claim tag clear register

偏移地址：0xFA4

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CLAIMCLR[3:0]			
												rw			

位 31:4 保留，必须保持复位值。

位 3:0 **CLAIMCLR[3:0]**：复位声明标记位 (Reset claim tag bits)

写：
0000：无影响
xxx1：将位 0 清零
xx1x：将位 1 清零
x1xx：将位 2 清零
1xxx：将位 3 清零

读操作：返回声明标记的当前值

CTI 锁定访问寄存器 (CTI_LAR)

CTI lock access register

偏移地址：0xFB0

复位值：N/A

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ACCESS_W[31:16]															
w															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ACCESS_W[15:0]															
w															

位 31:0 **ACCESS_W[31:0]**：CTI 寄存器写访问使能 (CTI register write access enable)
使能处理器内核写访问某些 CTI 寄存器（调试器无需解锁组件）

0xC5ACCE55：使能写访问
其他值：禁止写访问

CTI 锁定状态寄存器 (CTI_LSR)

CTI lock status register

偏移地址：0xFB4

复位值：0x0000 0003

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LOCK TYPE	LOCK GRANT	LOCK EXIST
													r	r	r

位 31:3 保留，必须保持复位值。

位 2 **LOCKTYPE**：CTI_LAR 寄存器大小 (Size of the CTI_LAR register)
0：32 位

位 1 **LOCKGRANT**：当前锁定状态 (Current status of lock)
当外部调试器读取该位时，该位始终返回零。

0：允许写访问
1：阻止写访问。仅允许读访问。

位 0 **LOCKEXIST**：存在锁定控制机制 (Existence of lock control mechanism)
该位表示是否存在锁定控制机制。当外部调试器读取该位时，该位始终返回零。

0：不存在锁定控制机制
1：存在锁定控制机制

CTI 认证状态寄存器 (CTI_AUTHSTAT)

CTI authentication status register

偏移地址：0xFB8

复位值：0x0000 000A

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SNID[1:0]		SID[1:0]		NSNID[1:0]		NSID[1:0]	
								r		r		r		r	

位 31:8 保留，必须保持复位值。

位 7:6 **SNID[1:0]**：安全非侵入式调试的安全等级 (Security level for secure non-invasive debug)

0x0：未设计

位 5:4 **SID[1:0]**：安全侵入式调试的安全等级 (Security level for secure invasive debug)

0x0：未设计

位 3:2 **NSNID[1:0]**：非安全非侵入式调试的安全等级 (Security level for non-secure non-invasive debug)

0x2：禁止

0x3：使能

位 1:0 **NSID[1:0]**：非安全侵入式调试的安全等级 (Security level for non-secure invasive debug)

0x2：禁止

0x3：使能

CTI 设备配置寄存器 (CTI_DEVID)

CTI device configuration register

偏移地址：0xFC8

复位值：0x0004 0800

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	NUMCH[3:0]			
												r			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NUMTRIG[7:0]								Res.	Res.	Res.	EXTMUXNUM[4:0]				
r											r				

位 31:20 保留，必须保持复位值。

位 19:16 **NUMCH[3:0]**：可用 ECT 通道数量 (Number of ECT channels available)
0x4：4 个通道

位 15:8 **NUMTRIG[7:0]**：可用 ECT 触发数量 (Number of ECT triggers available)
0x8：8 个触发输入和 8 个触发输出

位 7:5 保留，必须保持复位值。

位 4:0 **EXTMUXNUM[4:0]**：触发输入/输出多路复用器数量 (Number of trigger input/output multiplexers)
0x0：无

CTI 器件类型标识寄存器 (CTI_DEVTYPE)

CTI device type identifier register

偏移地址：0xFCC

复位值：0x0000 0014

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SUBTYPE[3:0]				MAJORTYPE[3:0]			
								r				r			

位 31:8 保留，必须保持复位值。

位 7:4 **SUBTYPE[3:0]**：子分类 (Sub-classification)
0x1：表示该组件为交叉触发组件。

位 3:0 **MAJORTYPE[3:0]**：主分类 (Major classification)
0x4：表示该组件允许调试器控制
CoreSight SoC-400 系统中的其他组件。

CTI CoreSight 外设标识寄存器 4 (CTI_PIDR4)

CTI CoreSight peripheral identity register 4

偏移地址：0xFD0

复位值：0x0000 0004

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	4KCOUNT[3:0]				JEP106CON[3:0]			
								r				r			

位 31:8 保留，必须保持复位值。

位 7:4 **4KCOUNT[3:0]**：寄存器文件大小 (Register file size)
0x0：寄存器文件占用单个 4 KB 区域

位 3:0 **JEP106CON[3:0]**：JEP106 连续代码 (JEP106 continuation code)
0x4：ARM® JEDEC 代码

CTI CoreSight 外设标识寄存器 0 (CTI_PIDR0)

CTI CoreSight peripheral identity register 0

偏移地址：0xFE0

复位值：0x0000 0006

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PARTNUM[7:0]							
								r							

位 31:8 保留，必须保持复位值。

位 7:0 **PARTNUM[7:0]**：产品编号字段 (Part number field)，位 [7:0]
0x06：CTI 产品编号

CTI CoreSight 外设标识寄存器 1 (CTI_PIDR1)

CTI CoreSight peripheral identity register 1

偏移地址：0xFE4

复位值：0x0000 00B9

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	JEP106ID[3:0]				PARTNUM[11:8]			
								r				r			

位 31:8 保留，必须保持复位值。

位 7:4 **JEP106ID[3:0]**：JEP106 标识代码字段 (JEP106 identity code field)，位 [3:0]
0xB：ARM® JEDEC 代码

位 3:0 **PARTNUM[11:8]**：产品编号字段 (Part number field)，位 [11:8]
0x9：CTI 产品编号

CTI CoreSight 外设标识寄存器 2 (CTI_PIDR2)

CTI CoreSight peripheral identity register 2

偏移地址：0xFE8

复位值：0x0000 004B

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REVISION[3:0]				JEDEC	JEP106ID[6:4]		
								r				r	r		

位 31:8 保留，必须保持复位值。

位 7:4 **REVISION[3:0]**：组件版本号 (Component revision number)
0x4：r0p5

位 3 **JEDEC**：JEDEC 分配的值 (JEDEC assigned value)
1：JEDEC 指定的设计人员 ID

位 2:0 **JEP106ID[6:4]**：JEP106 标识代码字段 (JEP106 identity code field)，位 [6:4]
0x3：ARM® JEDEC 代码

CTI CoreSight 外设标识寄存器 3 (CTI_PIDR3)

CTI CoreSight peripheral identity register 3

偏移地址：0xFEC

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REVAND[3:0]				CMOD[3:0]			
								r				r			

位 31:8 保留，必须保持复位值。

位 7:4 **REVAND[3:0]**：金属修复版本 (Metal fix version)
0x0：无金属修复

位 3:0 **CMOD[3:0]**：客户修改 (Customer modified)
0x0：无客户修改

CTI CoreSight 组件标识寄存器 0 (CTI_CIDR0)

CTI CoreSight component identity register 0

偏移地址：0xFF0

复位值：0x0000 000D

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[7:0]							
								r							

位 31:8 保留，必须保持复位值。

位 7:0 **PREAMBLE[7:0]**：组件 ID 字段 (Component ID field)，位 [7:0]
0x0D：公共 ID 值

CTI CoreSight 组件标识寄存器 1 (CTI_CIDR1)

CTI CoreSight component identity register 1

偏移地址：0xFF4

复位值：0x0000 0090

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CLASS[3:0]				PREAMBLE[11:8]			
								r				r			

位 31:8 保留，必须保持复位值。

位 7:4 **CLASS[3:0]**：组件 ID 字段 (Component ID field)，位 [15:12]——组件类
0x9：CoreSight 组件

位 3:0 **PREAMBLE[11:8]**：组件 ID 字段 (Component ID field)，位 [11:8]
0x0：公共 ID 值

CTI CoreSight 组件标识寄存器 2 (CTI_CIDR2)

CTI CoreSight component identity register 2

偏移地址：0xFF8

复位值：0x0000 0005

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[19:12]							
								r							

位 31:8 保留，必须保持复位值。

位 7:0 **PREAMBLE[19:12]**：组件 ID 字段 (Component ID field)，位 [23:16]
0x05：公共 ID 值

CTI CoreSight 组件标识寄存器 3 (CTI_CIDR3)

CTI CoreSight component identity register 3

偏移地址：0xFFC

复位值：0x0000 00B1

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[27:20]							
								r							

位 31:8 保留，必须保持复位值。

位 7:0 **PREAMBLE[27:20]**：组件 ID 字段 (Component ID field)，位 [31:24]
0xB1：公共 ID 值

CTI 寄存器映射和复位值

表 570. CTI 寄存器映射和复位值

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x000	CTI_CONTROL	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	GLBEN
	Reset value																																0
0x010	CTI_INTACK	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	INTACK[7:0]							
	Reset value																									0	0	0	0	0	0	0	0
0x014	CTI_APPSET	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	APPSET[3:0]			
	Reset value																													0	0	0	0
0x018	CTI_APPCLEAR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	APPCLEAR [3:0]			
	Reset value																													0	0	0	0
0x01C	CTI_APPPULSE	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	APPPULSE [3:0]			
	Reset value																													0	0	0	0
0x020	CTI_INEN0	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TRIGINEN [3:0]			
	Reset value																													0	0	0	0
0x024	CTI_INEN1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TRIGINEN [3:0]			
	Reset value																													0	0	0	0
0x028	CTI_INEN2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TRIGINEN [3:0]			
	Reset value																													0	0	0	0
0x02C	CTI_INEN3	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TRIGINEN [3:0]			
	Reset value																													0	0	0	0
0x030	CTI_INEN4	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TRIGINEN [3:0]			
	Reset value																													0	0	0	0
0x034	CTI_INEN5	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TRIGINEN [3:0]			
	Reset value																													0	0	0	0
0x038	CTI_INEN6	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TRIGINEN [3:0]			
	Reset value																													0	0	0	0
0x03C	CTI_INEN7	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TRIGINEN [3:0]			
	Reset value																													0	0	0	0
0x0A0	CTI_OUTEN0	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TRIGOUTEN [3:0]			
	Reset value																													0	0	0	0
0x0S4	CTI_OUTEN1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TRIGOUTEN [3:0]			
	Reset value																													0	0	0	0
0x0S8	CTI_OUTEN2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TRIGOUTEN [3:0]			
	Reset value																													0	0	0	0

表 570. CTI 寄存器映射和复位值（续）

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x0SC	CTI_OUTEN3	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TRIGOUTEN [3:0]			
	Reset value																													0	0	0	0
0x0B0	CTI_OUTEN4	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TRIGOUTEN [3:0]			
	Reset value																													0	0	0	0
0x0B4	CTI_OUTEN5	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TRIGOUTEN [3:0]			
	Reset value																													0	0	0	0
0x0B8	CTI_OUTEN6	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TRIGOUTEN [3:0]			
	Reset value																													0	0	0	0
0x0BC	CTI_OUTEN7	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TRIGOUTEN [3:0]			
	Reset value																													0	0	0	0
0x130	CTI_TRIGISTS	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TRIGINSTATUS[7:0]							
	Reset value																									0	0	0	0	0	0	0	0
0x134	CTI_TRIGOSTS	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TRIGOUTSTATUS[7:0]							
	Reset value																									0	0	0	0	0	0	0	0
0x138	CTI_CHINSTS	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CHISTATUS [3:0]			
	Reset value																													0	0	0	0
0x13C	CTI_CHOUTSTS	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CHOSTATUS [3:0]			
	Reset value																													0	0	0	0
0x140	CTI_GATE	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	GATEEN [3:0]			
	Reset value																													1	1	1	1
0xFA0	CTI_CLAIMSET	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CLAIMSET [3:0]			
	Reset value																													1	1	1	1
0xFA4	CTI_CLAIMCLR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CLAIMCLR [3:0]			
	Reset value																													0	0	0	0
0xFB0	CTI_LAR	KEY																															
	Reset value																																
0xFB4	CTI_LSR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	NTT	SLK	SLI
	Reset value																														0	1	1
0xFB8	CTI_AUTHSTAT	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SNID [1:0]		SID [1:0]		NSNID [1:0]		NSID [1:0]	
	Reset value																									0	0	0	0	0	1	0	1
0xFC8	CTI_DEVID	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	NUMCH[3:0]				NUMTRIG[7:0]								Res.	Res.	Res.	EXMUXNUM [4:0]				
	Reset value													0	1	0	0	0	0	0	0	1	0	0	0				0	0	0	0	0
0xFCC	CTI_DEVTYPE	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SUB[3:0]				MAJOR[3:0]			
	Reset value																									0	0	0	1	0	1	0	0

表 570. CTI 寄存器映射和复位值（续）

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0xFD0	CTI_PIDR4	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	4KCOUNT [3:0]				JEP106CON [3:0]			
	Reset value																									0	0	0	0	0	1	0	0
0xFD4	CTI_PIDR5	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	Reset value																																
0xFD8	CTI_PIDR6	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	Reset value																																
0xFDC	CTI_PIDR7	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	Reset value																																
0xFE0	CTI_PIDR0	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PARTNUM[7:0]							
	Reset value																									0	0	0	0	0	1	1	0
0xFE4	CTI_PIDR1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	JEP106ID [3:0]				PARTNUM [11:8]			
	Reset value																									1	0	1	1	1	0	0	1
0xFE8	CTI_PIDR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REVISION [3:0]				JEDEC	JEP106ID [6:4]		
	Reset value																									0	1	0	0	1	0	1	1
0xFEC	CTI_PIDR3	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REVAND[3:0]				CMOD[3:0]			
	Reset value																									0	0	0	0	0	0	0	0
0xFF0	CTI_CIDR0	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[7:0]							
	Reset value																									0	0	0	0	1	1	0	1
0xFF4	CTI_CIDR1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CLASS[3:0]				PREAMBLE [11:8]			
	Reset value																									1	0	0	1	0	0	0	0
0xFF8	CTI_CIDR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[19:12]							
	Reset value																									0	0	0	0	0	1	0	1
0xFFC	CTI_CIDR3	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[27:20]							
	Reset value																									1	0	1	1	0	0	0	1

60.5.4 跟踪聚合器 (CSTF)

跟踪聚合器是一款将两个跟踪源的 ATB 总线合并为一个单个 ATB 总线的 CoreSight 组件，具有两个 ATB 从端口及一个 ATB 主端口。仲裁器根据可编程优先级选择从端口。

从端口连接如下：

- S0：Cortex-M7 ETM
- S1：Cortex-M7 ITM

CSTF 寄存器允许单独使能从端口，并允许配置其优先级设置。只有在禁止了跟踪时才能修改优先级。仲裁工作过程如下所述：

- 仲裁器选择分配优先级最高的从端口，该端口具有有效数据。
- 多达 *min_hold_time* 的传输从所选从端口传递到主端口，其中 *min_hold_time* 可在 CONTROL 寄存器中编程。
- 然后执行新的仲裁

高优先级应分配给连接到源的从端口，并且该从端口具有少量缓冲，或者在该从端口上不容许数据丢失。低优先级应分配给连接到不太关键的源的从端口，或者是缓冲区很大的从端口。

有关 ATB 聚合器 CoreSight 组件的更多信息，请参见《ARM® CoreSight™ SoC-400 技术参考手册》*[2]*。

跟踪聚合器寄存器

CSTF 控制寄存器 (CSTF_CTRL)

CSTF control register

偏移地址：0x000

复位值：0x0000 0300

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	MIN_HOLD_TIME[3:0]				Res.	Res.	Res.	Res.	Res.	Res.	ENS1	ENS0
				rw										rw	rw

位 31:12 保留，必须保持复位值。

位 11:8 **MIN_HOLD_TIME[3:0]**：仲裁间的事务数量 (Number of transactions between arbitrations)。

0x0：1 个事务
:
0xE：15 个事务
0xF：保留

位 7:2 保留，必须保持复位值。

位 1 **ENS1**：从端口 S1 使能 (Slave port S1 enable)

0：禁止端口
1：使能端口

位 0 **ENS0**：从端口 S0 使能 (Slave port S0 enable)

0：禁止端口
1：使能端口

CSTF 优先级寄存器 (CSTF_PRIORITY)

CSTF priority register

偏移地址：0x004

复位值：0x0000 0688

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PRIPORT1[2:0]			PRIPORT0[2:0]		
										rw			rw		

位 31:6 保留，必须保持复位值。

位 5:3 **PRIPORT1[2:0]**：从端口 S1 优先级 (Slave port S1 priority)

0：最高优先级

:

7：最低优先级

位 2:0 **PRIPORT0[2:0]**：从端口 S0 优先级 (Slave port S0 priority)

0：最高优先级

:

7：最低优先级

将 CSTF 声明标记置位寄存器 (CSTF_CLAIMSET)

CSTF claim tag set register

偏移地址：0xFA0

复位值：0x0000 000F

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CLAIMSET[3:0]			
												rw			

位 31:4 保留，必须保持复位值。

位 3:0 **CLAIMSET[3:0]**：置位声明标记位 (Set claim tag bits)

写：

0000：无影响

xxx1：将位 0 置 1

xx1x：将位 1 置 1

x1xx：将位 2 置 1

1xxx：将位 3 置 1

读：

0xF：表示声明标记使用四个位

CSTF 声明标记清零寄存器 (CSTF_CLAIMCLR)

CSTF claim tag clear register

偏移地址：0xFA4

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CLAIMCLR[3:0]			
												rw			

位 31:4 保留，必须保持复位值。

位 3:0 **CLAIMCLR[3:0]**：复位声明标记位 (Reset claim tag bits)

写：
0000：无影响
xxx1：将位 0 清零
xx1x：将位 1 清零
x1xx：将位 2 清零
1xxx：将位 3 清零

读操作：返回声明标记的当前值

CSTF 锁定访问寄存器 (CSTF_LAR)

CSTF lock access register

偏移地址：0xFB0

复位值：N/A

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ACCESS_W[31:16]															
w															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ACCESS_W[15:0]															
w															

位 31:0 **ACCESS_W[31:0]**：CSTF 寄存器写访问使能 (CSTF register write access enable)

该字段使能处理器内核写访问某些 CSTF 寄存器（调试器无需解锁组件）。

0xC5ACCE55：使能写访问
其他值：禁止写访问

CSTF 锁定状态寄存器 (CSTF_LSR)

CSTF lock status register

偏移地址：0xFB4

复位值：0x0000 0003

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LOCK TYPE	LOCK GRANT	LOCK EXIST
													r	r	r

位 31:3 保留，必须保持复位值。

位 2 **LOCKTYPE**：CSTF_LAR 寄存器大小 (Size of the CSTF_LAR register)

0：32 位

位 1 **LOCKGRANT**：当前锁定状态 (Current status of lock)

当外部调试器读取该位时，该位始终返回零。

0：允许写访问

1：阻止写访问。仅允许读访问。

位 0 **LOCKEXIST**：存在锁定控制机制 (Existence of lock control mechanism)

该位表示是否存在锁定控制机制。当外部调试器读取该位时，该位始终返回零。

0：不存在锁定控制机制

1：存在锁定控制机制

CSTF 认证状态寄存器 (CSTF_AUTHSTAT)

CSTF authentication status register

偏移地址：0xFB8

复位值：0x0000 000A

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SNID[1:0]		SID[1:0]		NSNID[1:0]		NSID[1:0]	
								r		r		r		r	

位 31:8 保留，必须保持复位值。

位 7:6 **SNID[1:0]**：安全非侵入式调试的安全等级 (Security level for secure non-invasive debug)
0x0：未设计

位 5:4 **SID[1:0]**：安全侵入式调试的安全等级 (Security level for secure invasive debug)
0x0：未设计

位 3:2 **NSNID[1:0]**：非安全非侵入式调试的安全等级 (Security level for non-secure non-invasive debug)
0x2：禁止
0x3：使能

位 1:0 **NSID[1:0]**：非安全侵入式调试的安全等级 (Security level for non-secure invasive debug)
0x2：禁止
0x3：使能

CSTF CoreSight 器件标识寄存器 (CSTF_DEVID)

CSTF CoreSight device identity register

偏移地址：0xFC8

复位值：0x0000 0024

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SCHEME[3:0]				PORTCNT[3:0]			
								r				r			

位 31:8 保留，必须保持复位值。

位 7:4 **SCHEME[3:0]**：优先级方案 (Priority scheme)
0x2：静态优先级

位 3:0 **PORTCNT[3:0]**：连接的输入端口数量 (Number of input ports connected)
0x4：四个输入端口

CSTF CoreSight 器件类型标识寄存器 (CSTF_TYPEID)

CSTF CoreSight device type identity register

偏移地址：0xFCC

复位值：0x0000 0012

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DEVTYPEID[7:0]							
								r							

位 31:8 保留，必须保持复位值。

位 7:0 **DEVTYPEID[7:0]**：器件类型标识符 (Device type identifier)
0x12：跟踪聚合器

CSTF CoreSight 外设标识寄存器 0 (CSTF_PIDR0)

CSTF CoreSight peripheral identity register 0

偏移地址：0xFE0

复位值：0x0000 0008

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PARTNUM[7:0]							
								r							

位 31:8 保留，必须保持复位值。

位 7:0 **PARTNUM[7:0]**：产品编号字段 (Part number field)，位 [7:0]
0x08：CSTF 产品编号

CSTF CoreSight 外设标识寄存器 1 (CSTF_PIDR1)

CSTF CoreSight peripheral identity register 1

偏移地址：0xFE4

复位值：0x0000 00B9

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	JEP106ID[3:0]				PARTNUM[11:8]			
								r				r			

位 31:8 保留，必须保持复位值。

位 7:4 **JEP106ID[3:0]**：JEP106 标识代码字段 (JEP106 identity code field)，位 [3:0]
0xB：ARM® JEDEC 代码

位 3:0 **PARTNUM[11:8]**：产品编号字段 (Part number field)，位 [11:8]
0x9：CSTF 产品编号

CSTF CoreSight 外设标识寄存器 2 (CSTF_PIDR2)

CSTF CoreSight peripheral identity register 2

偏移地址：0xFE8

复位值：0x0000 001B

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REVISION[3:0]				JEDEC	JEP106ID[6:4]		
								r				r	r		

位 31:8 保留，必须保持复位值。

位 7:4 **REVISION[3:0]**：组件版本号 (Component revision number)
0x1：r0p1

位 3 **JEDEC**：JEDEC 分配的值 (JEDEC assigned value)
1：JEDEC 指定的设计人员 ID

位 2:0 **JEP106ID[6:4]**：JEP106 标识代码字段 (JEP106 identity code field)，位 [6:4]
0x3：ARM® JEDEC 代码

CSTF CoreSight 外设标识寄存器 3 (CSTF_PIDR3)

CSTF CoreSight peripheral identity register 3

偏移地址：0xFEC

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REVAND[3:0]				CMOD[3:0]			
								r				r			

位 31:8 保留，必须保持复位值。

位 7:4 **REVAND[3:0]**：金属修复版本 (Metal fix version)
0x0：无金属修复

位 3:0 **CMOD[3:0]**：客户修改 (Customer modified)
0x0：无客户修改

CSTF CoreSight 外设标识寄存器 4 (CSTF_PIDR4)

CSTF CoreSight peripheral identity register 4

偏移地址：0xFD0

复位值：0x0000 0004

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	4KCOUNT[3:0]				JEP106CON[3:0]			
								r				r			

位 31:8 保留，必须保持复位值。

位 7:4 **4KCOUNT[3:0]**：寄存器文件大小 (Register file size)

0x0：寄存器文件占用单个 4 KB 区域

位 3:0 **JEP106CON[3:0]**：JEP106 连续代码 (JEP106 continuation code)

0x4：ARM® JEDEC 代码

CSTF CoreSight 组件标识寄存器 0 (CSTF_CIDR0)

CSTF CoreSight component identity register 0

偏移地址：0xFF0

复位值：0x0000 000D

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[7:0]							
								r							

位 31:8 保留，必须保持复位值。

位 7:0 **PREAMBLE[7:0]**：组件 ID 字段 (Component ID field)，位 [7:0]

0x0D：公共 ID 值

CSTF CoreSight 组件标识寄存器 1 (CSTF_CIDR1)

CSTF CoreSight component identity register 1

偏移地址：0xFF4

复位值：0x0000 0090

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CLASS[3:0]				PREAMBLE[11:8]			
								r				r			

位 31:8 保留，必须保持复位值。

位 7:4 **CLASS[3:0]**：组件 ID 字段 (Component ID field)，位 [15:12]——组件类
0x9：CoreSight 组件

位 3:0 **PREAMBLE[11:8]**：组件 ID 字段 (Component ID field)，位 [11:8]
0x0：公共 ID 值

CSTF CoreSight 组件标识寄存器 2 (CSTF_CIDR2)

CSTF CoreSight component identity register 2

偏移地址：0xFF8

复位值：0x0000 0005

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[19:12]							
								r							

位 31:8 保留，必须保持复位值。

位 7:0 **PREAMBLE[19:12]**：组件 ID 字段 (Component ID field)，位 [23:16]
0x05：公共 ID 值

CSTF CoreSight 组件标识寄存器 3 (CSTF_CIDR3)

CSTF CoreSight component identity register 3

偏移地址：0xFFC

复位值：0x0000 00B1

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[27:20]							
								r							

位 31:8 保留，必须保持复位值。

位 7:0 **PREAMBLE[27:20]**：组件 ID 字段 (Component ID field)，位 [31:24]
0xB1：公共 ID 值

跟踪聚合器寄存器映射和复位值

表 571. CSTF 寄存器映射和复位值

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x000	CSTF_CTRL	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MIN_HOLD_TIME [3:0]				Res.	Res.	Res.	Res.	ENS3	ENS2	ENS1	ENS0
	Reset value																					0	0	1	1					0	0	0	0
0x004	CSTF_PRIORITY	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PRIPORT3 [2:0]			PRIPORT2 [2:0]			PRIPORT1 [2:0]			PRIPORT0 [2:0]		
	Reset value																					0	1	1	0	1	0	0	0	1	0	0	0
0xFA0	CSTF_CLAIMSET	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CLAIMSET [3:0]			
	Reset value																													1	1	1	1
0xFA4	CSTF_CLAIMCLR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CLAIMCLR [3:0]			
	Reset value																													0	0	0	0
0xFB0	CSTF_LAR	KEY[31:0]																															
	Reset value																																
0xFB4	CSTF_LSR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	NTT	SLK	SLI
	Reset value																														0	1	1

表 571. CSTF 寄存器映射和复位值（续）

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0xFB8	CSTF_AUTHSTAT	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SNID [1:0]		SID [1:0]		NSNID [1:0]		NSID [1:0]	
	Reset value																									0	0	0	0	0	0	0	0
0xFC8	CSTF_DEVID	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SCHEME [3:0]				PORTCNT [3:0]			
	Reset value																									0	0	1	0	0	1	0	0
0xFCC	CSTF_TYPEID	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DEVTYPEID[7:0]							
	Reset value																									0	0	0	1	0	0	1	0
0xFD0	CSTF_PIDR4	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	4KCOUNT [3:0]				JEP106CON [3:0]			
	Reset value																									0	0	0	0	0	1	0	0
0xFD4	CSTF_PIDR5	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	Reset value																																
0xFD8	CSTF_PIDR6	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	Reset value																																
0xFDC	CSTF_PIDR7	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	Reset value																																
0xFE0	CSTF_PIDR0	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PARTNUM[7:0]							
	Reset value																									0	0	0	0	1	0	0	0
0xFE4	CSTF_PIDR1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	JEP106ID [3:0]				PARTNUM [11:8]			
	Reset value																									1	0	1	1	1	0	0	1
0xFE8	CSTF_PIDR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REVISION [3:0]				JEDEC	JEP106ID [6:4]		
	Reset value																									0	0	0	1	1	0	1	1
0xFEC	CSTF_PIDR3	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REVAND[3:0]				CMOD[3:0]			
	Reset value																									0	0	0	0	0	0	0	0
0xFF0	CSTF_CIDR0	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[7:0]							
	Reset value																									0	0	0	0	1	1	0	1
0xFF4	CSTF_CIDR1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CLASS[3:0]				PREAMBLE [11:8]			
	Reset value																									1	0	0	1	0	0	0	0
0xFF8	CSTF_CIDR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[19:12]							
	Reset value																									0	0	0	0	0	1	0	1
0xFFC	CSTF_CIDR3	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[27:20]							
	Reset value																									1	0	1	1	0	0	0	1

60.5.5 嵌入式跟踪 FIFO (ETF)

ETF 是一款 8 KB 存储器，用来从两个跟踪源（即CPU 内核的 ETM 和 ITM）捕获跟踪数据。ETF 是 CoreSight™ 跟踪存储器控制器组件的设计组态。

ETF 可用于三种模式（在模式寄存器中选择）：

1. 硬件 FIFO 模式

   跟踪存储器作为 FIFO 使用，通过 ATB 主接口排空。跟踪数据被捕获到跟踪 RAM 中，当跟踪 RAM 已满时，传入的跟踪流将停止。当跟踪缓冲区非空时，跟踪数据流从 ATB 主接口送到 TPIU。

   在该模式下，FIFO 的作用是平滑到达跟踪端口的跟踪信息流。由于跟踪数据本身具有突发性，峰值数据速率可以轻松超出端口容量，从而导致上溢。ETF 可在跟踪端口实现稳定的数据速率，其大小可以根据平均速率而非峰值速率来调整。跟踪信息由跟踪端口分析器工具实时片外存储，因此跟踪日志能够非常大。

2. 软件 FIFO 模式

   跟踪存储器作为 FIFO 使用，当跟踪被捕获时，可通过 RRD 寄存器读取。跟踪数据被捕获到跟踪 RAM 中，当跟踪 RAM 已满时，传入的跟踪流将停止。

   该模式允许通过软件由 DMA 将跟踪信息传输到系统存储器、或传输到一个高速接口（SPI、USB 等等），甚至对跟踪信息进行监视。注意，与硬件 FIFO 模式不同，该模式是侵入式的，因为其使用由处理器共享的系统资源。

3. 循环缓冲区模式

   跟踪存储器作为循环缓冲区使用。跟踪数据被捕获到跟踪存储器，该跟踪存储器起始地址为写指针寄存器指向的位置。即使跟踪存储器已满，传入的跟踪数据仍继续覆盖到跟踪存储器，直到出现停止条件。

   在该模式下，ETF 在片上存储跟踪数据，所以跟踪日志的大小限制为 ETF SRAM 的大小，本案例中为 8 KB。作为一个循环缓冲区，当 FIFO 变满时，传入的跟踪数据会覆盖最早存储的数据，最早存储的数据将丢失。因此，在缓冲区停止之前，跟踪缓冲区中的内容都表示最新的处理器活动，而不是跟踪开始后的所有活动。

   有三种方法可以在跟踪停止后读取缓冲区内容：

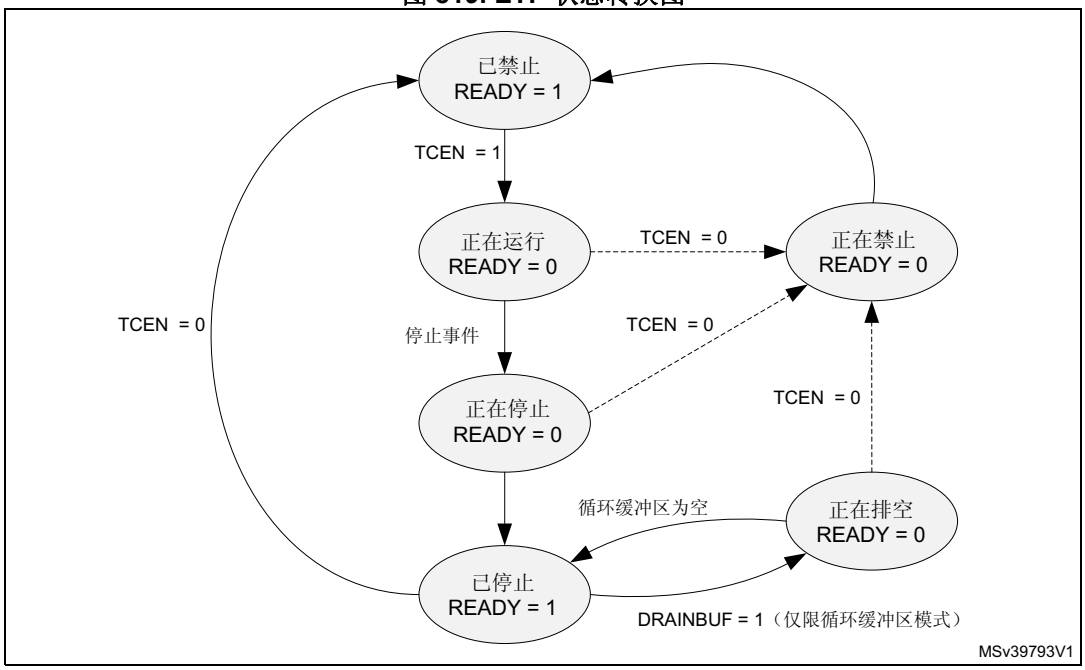
   – 通过跟踪端口——在使能了 TPIU 的情况下，缓冲区的内容通过跟踪端口输出。可通过将 ETF_FFCR 寄存器中的 DRAINBUF 位置 1 实现此操作。
   – 通过调试端口——调试器可以通过 RRD 寄存器读取缓冲区，该寄存器可通过系统 APB-D 访问。
   – 通过软件——由于 APB-D 可以通过系统总线访问，所以处理器可以通过 RRD 寄存器读取缓冲区。

ETF 可在以下状态间转换：

- **输出关闭**
  当禁止跟踪捕获时，或在复位之后可以进入该状态。只能在该状态下，才能对 ETF 编程。
- **运行中**
  在该状态下，执行跟踪捕获。在禁止跟踪捕获状态下，通过使能跟踪捕获可以进入该状态。
- **停止**
  在该状态下，跟踪捕获停止，但可读取或排空缓冲区内容。在停止事件（触发或刷新）后可进入该状态。
- **禁止中**
  这是一个禁止跟踪捕获时的过渡状态。
- **停止中**
  这是一个停止跟踪捕获时的过渡状态。
- **排空中**
  在停止状态下排空缓冲区时进入该状态。

状态转换图如*图 819* 所示。

图 819. ETF 状态转换图



有关跟踪存储器控制器 CoreSight™ 组件的更多信息，请参见《ARM® CoreSight™ 跟踪存储器控制器技术参考手册》*[3]*。

ETF 寄存器

ETF RAM 大小寄存器 (ETF_RSZ)

ETF RAM size register

偏移地址：0x004

复位值：0x0000 0800

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	RSZ[30:16]														
	r														

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSZ[15:0]															
r															

位 31 保留，必须保持复位值。

位 30:0 **RSZ[30:0]**：RAM 大小 (RAM size)
该字段的值表示 32 位字的数量

0X800：2048 个字 = 8 KB

ETF 状态寄存器 (ETF_STS)

ETF status register

偏移地址：0x00C

复位值：0x0000 001C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	EMPTY	FTEMPTY	READY	TRIGD	FULL
											r	r	r	r	r

位 31:5 保留，必须保持复位值。

位 4 **EMPTY**：跟踪 FIFO 为空 (Trace FIFO empty)
该位仅当 ETF_CTL 寄存器的 TCEN 位为高电平时有效。当 TCEN 为低电平时，该位读为零。

0：跟踪 FIFO 含有数据
1：跟踪 FIFO 为空。
注： 跟踪 FIFO 为空并不表示 ETF 管道也为空。ETF 管道是否为空由 FTEMPTY 位表示。

位 3 **FTEMPTY**：格式化器为空 (Formatter empty)
当跟踪捕获停止时，该位置 1，所有内部管道和缓冲区都已排空。与 READY 不同，该位不受缓冲区排空影响。ACQCOMP 输出反映了该位的值。

位 2 **READY**：ETF 就绪 (ETF ready)

当跟踪捕获停止时，该位置 1，所有内部管道和缓冲区都已排空（停止或禁止状态）。

位 1 **TRIGD**：触发 (Triggered)

当跟踪捕获正在进行并且 TMC 已检测到触发事件时，该位置 1。当退出禁止状态时，该位清零。

该位仅在循环缓冲区模式下运行。在其他模式下，该位始终为低电平。

该位并不表示触发已嵌入在 TMC 的格式化输出跟踪数据中。输出跟踪流上的触发指示由格式化器和刷新控制寄存器 (ETF_FFCR) 的编程确定。

位 0 **FULL**：跟踪缓冲区已满 (Trace buffer full)

在循环缓冲区模式下，当 RAM 写指针回卷在缓冲区的顶部时，该标志置 1，并保持置 1，直到 ETF_CTL 寄存器的 TCEN 位清零并置 1。

在软件和硬件 FIFO 模式下，该标志表示跟踪存储器中的当前空间小于或等于在 ETF_BUFWM 寄存器中编程的值，即填充级别 >= MEM_SIZE - BUFWM。

当退出禁止状态时，该位清零。FULL 输出反映了该寄存器位的值。

ETF RAM 读数据寄存器 (ETF_RRD)

ETF RAM read data register

偏移地址：0x010

复位值：未知

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RRD[31:16]															
r															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RRD[15:0]															
r															

位 31:0 **RRD[31:0]**：RAM 读数据 (RAM Read Data)。

循环缓冲区模式：

当处于停止状态且缓冲区非空时，读取该寄存器将从跟踪缓冲区返回下一个数据。读取所有的跟踪缓冲区后，ETF_STS 寄存器的 Empty 位置 1，后续读操作返回 0xFFFFFFFF。在非停止状态下读取该寄存器将返回 0xFFFFFFFF。

软件 FIFO 模式：

读取该寄存器将返回 FIFO 中的数据。如果在 FIFO 为空时读取该寄存器，则会返回数据 0xFFFFFFFF。

硬件 FIFO 模式：

读取该寄存器将返回 0xFFFFFFFF。

ETF RAM 读指针寄存器 (ETF_RRP)

ETF RAM read pointer register

偏移地址：0x014

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	RRP[12:0]												
			rw												

位 31:13 保留，必须保持复位值。

位 12:0 **RRP[12:0]**：RAM 读指针 (RAM Read Pointer)

RAM 读指针寄存器包含读指针的值，用于通过 ETF_RRD 寄存器和 APB 接口读取跟踪存储器条目。指针可以用一个字节地址编程，64 位对齐（即位 0 到 3 应为零）。每写入一个完整的 64 位 FIFO 条目，该指针就会增加 8。当该指针达到其最大值时，就会回卷。

只有在禁止状态下，才能写入该寄存器， 但可在禁止状态、循环缓冲区模式和 SW FIFO 模式的停止状态以及 SW FIFO 模式的运行和停止状态下读取。

ETF RAM 写指针寄存器 (ETF_RWP)

ETF RAM write pointer register

偏移地址：0x018

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	RWP[12:0]												
			rw												

位 31:13 保留，必须保持复位值。

位 12:0 **RWP[12:0]**：RAM 写指针 (RAM write pointer)

RAM 写指针寄存器包含写指针的值，用于通过 ETF_RWD 寄存器和 APB 接口写入跟踪存储器条目。指针可以用一个字节地址编程，64 位对齐（即位 0 到 3 应为零）。每读取一个完整的 64 位 FIFO 条目，该指针就会增加 8。当该指针达到其最大值时，就会回卷。

只有在禁止状态下，才能写入该寄存器， 但可在禁止状态、循环缓冲区模式和 SW FIFO 模式的停止状态以及 SW FIFO 模式的运行和停止状态下读取。

ETF 触发计数器寄存器 (ETF_TRG)

ETF trigger counter register

偏移地址：0x01C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	TRG[10:0]										
					rw										

位 31:11 保留，必须保持复位值。

位 10:0 **TRG[10:0]**：触发计数器 (Trigger counter)

在循环缓冲模式中，在检测到 TRIGIN 输入上的上升沿或输入跟踪流中的触发数据包 (ATID = 7'h7D) 之后，指定要在跟踪 RAM 中捕获的 32 位字的数量。一旦捕获了指定数量的数据字，就发生触发事件。触发事件对 ETF 行为的影响由 FFCR 寄存器控制。

触发后写入跟踪 RAM 的 32 位字的数量为存储在寄存器中的值加 1。当 ETF 处于软件 FIFO 模式或硬件 FIFO 模式时，该寄存器被忽略。当触发计数器开始计数时，在计数器达到零之前，所有其他触发都被忽略，不管是 TRIGIN 上的触发还是传入跟踪流中的触发。当触发计数器达到零时，它将保持为零，直到通过写入该寄存器对其重新编程为止。

当 READY 变为高电平时，该寄存器会清零，因此，跟踪捕获停止时计数器的状态不会影响后续跟踪捕获会话。在非禁止状态下写入该寄存器将引发不可预测的行为。

在禁止状态下或在循环缓冲区模式下，允许随时读取该寄存器。读访问返回触发计数器的当前值。

ETF 控制寄存器 (ETF_CTL)

ETF control register

偏移地址：0x020

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TCEN
															rw

位 31:1 保留，必须保持复位值。

位 0 **TCEN**：跟踪捕获使能 (Trace capture enable)

写入时：

0：禁止跟踪捕获（从运行中、停止中或停止状态转为禁止中或禁止状态）

1：使能跟踪捕获（从禁止状态转为运行中状态）

读取时，该位为低电平（禁止中或禁止状态下）或未高电平（其他状态下）。

ETF RAM 写数据寄存器 (ETF_RWD)

ETF RAM write data register

偏移地址：0x024

复位值：未知

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RWD[31:16]															
w															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RWD[15:0]															
w															

位 31:0 **RWD[31:0]**：RAM 写数据 (RAM write data)

在禁止状态下，写入该寄存器可将数据存储在 RWP 指向的位置。在非禁止状态下，忽略对该寄存器的写操作。当写入完整存储器宽度（64 位）的数据时，数据被写入存储器，同时 RAM 写指针递增到下一个存储器字。

该寄存器用于测试目的。

ETF 模式寄存器 (ETF_MODE)

ETF mode register

偏移地址：0x028

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MODE[1:0]	
														rw	

位 31:2 保留，必须保持复位值。

位 1:0 **MODE[1:0]**：操作模式 (Operation mode)

00b：循环缓冲区模式

在该模式下，跟踪存储器作为循环缓冲区使用。跟踪数据被捕获到跟踪存储器，该跟踪存储器起始地址为写指针寄存器指向的位置。即使跟踪存储器已满，传入的跟踪数据仍继续覆盖到跟踪存储器，直到出现了停止条件。

01b：软件 FIFO 模式

在该模式下，跟踪存储器作为 FIFO 使用，当跟踪被捕获时，可通过 RRD 寄存器读取。跟踪数据被捕获到跟踪 RAM 中，当跟踪 RAM 已满时，传入的跟踪流将停止。

10b：硬件 FIFO 模式

在该模式下，跟踪存储器作为 FIFO 使用，通过 ATB 主接口排空。跟踪数据被捕获到跟踪 RAM 中，当跟踪 RAM 已满时，传入的跟踪流将停止。当跟踪缓冲区非空时，通过 ATB 主接口到 TPIU 来排空跟踪数据。

11b：保留

ETF 锁存缓冲区填充级别寄存器 (ETF_LBUFLVL)

ETF latched buffer fill level register

偏移地址：0x02C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res	LBUFLEVEL[11:0]											
				r											

位 31:12 保留，必须保持复位值。

位 11:0 **LBUFLEVEL[11:0]**：锁存缓冲区填充级别 (Latched buffer fill level)

读取该寄存器将以 32 位字的形式返回上一次读取该寄存器之后追踪存储器的最大填充级别。读取该寄存器还会使其内容更新为当前的填充级别。

当进入禁止状态时，该寄存器将保持其最近一次的值。当处于禁止状态时，读取该寄存器不会影响其值。当退出禁止状态时，LBUFLEVEL 寄存器清零。

该寄存器用于跟踪系统的性能分析。

ETF 当前缓冲区填充级别寄存器 (ETF_CBUFLVL)

ETF current buffer fill level register

偏移地址：0x030

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res	CBUFLEVEL[11:0]											
				r											

位 31:12 保留，必须保持复位值。

位 11:0 **CBUFLEVEL[11:0]**：当前缓冲区填充级别 (Current buffer fill level)

读取该寄存器将以 32 位字的形式返回追踪存储器的当前填充级别。

当 TCEN 为低电平时，该寄存器清零。

ETF 缓冲级水印寄存器 (ETF_BUFWM)

ETF buffer level watermark register

偏移地址：0x034

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res	Res	BUFWM[10:0]										
					rw										

位 31:11 保留，必须保持复位值。

位 10:0 **BUFWM[10:0]**：缓冲级水印 (Buffer level watermark)

编程到该寄存器中的值以 32 位字的形式表示跟踪存储器中所需的阈值空位级别。当 FIFO 的空间小于或等于该值时（即填充级别 >= MEM_SIZE - BUFWM），FULL 输出被拉高，同时 STS 寄存器中的 FULL 位置 1。

该寄存器仅在软件 FIFO 和硬件 FIFO 模式下使用。在循环缓冲区模式下，可将 RWP 编程到所需空位触发级别来实现此功能，因此，当指针回卷时，FULL 位置 1，表示空位级别已经降到所需级别以下。

可写入该寄存器的最大值为 MEM_SIZE - 1。这种情况下，当第一个 32 位字写入跟踪存储器时，FULL 位输出被置为有效。

在非禁止状态下写入该寄存器会引发不可预测的行为。

ETF 格式化器和刷新状态寄存器 (ETF_FFSR)

ETF formatter and flush status register

偏移地址：0x300

复位值：0x0000 0002

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FTSTOPPED	FLINPROG
														r	r

位 31:2 保留，必须保持复位值。

位 1 **FTSTOPPED**：格式化器停止 (Formatter stopped)

该位的工作方式与 ETF_STS 寄存器中的 FTEMPTY 位相同。

位 0 **FLINPROG**：刷新正在进行 (Flush in progress)

指示 ATB 从端口上是否有正在进行的刷新。该位反映 AFVALIDS 输出的状态。刷新可由 ETF_FFCR 寄存器中的刷新控制位启动，或者由 ATB 主端口请求。

0：无正在进行的刷新
1：正在进行刷新

ETF 格式化器和刷新控制寄存器 (ETF_FFCR)

ETF formatter and flush control register

偏移地址：0x304

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	DRAINBUF	STOPONTRGEV	STOPONFL	Res.	TRIGONFL	TRGONTRGEV	TRGONTRGIN	Res.	FLUSHMAN	FONTRGEV	FONFLIN	Res.	Res.	ENTI	ENFT
	rw	rw	rw		rw	rw	rw		rw	rw	rw			rw	rw

位 31:15 保留，必须保持复位值。

位 14 **DRAINBUF**：排空缓冲区 (Drain buffer)

该位用于使能在格式化器停止后通过 ATB 主接口排空跟踪数据。这在循环缓冲区模式下，可用于先将跟踪数据捕获到跟踪存储器，再通过 ATB 主接口排空捕获的跟踪数据。

在停止状态下，向该位写入 1 可启动通过 ATB 主接口排空跟踪缓冲区的内容。该位始终读为零。当排空正在进行时，ETF_STS 寄存器的 READY 位变为低电平。

该位仅当 ETF 处于循环缓冲区模式且使能了格式化（即 ETF_FFCR 寄存器的 ENFT 位置 1）时有效。如果 ETF 处于任何其他模式或处于非停止状态，将该位置 1 会引发不可预测的行为。

当在循环缓冲模式下跟踪捕获完成时，必须通过相同的机制从跟踪存储器检索所有捕获的跟踪数据，要么通过 RRD 读操作读取所有跟踪数据，要么通过将 DRAINBUF 位置 1 来排空所有跟踪数据。在通过 RRD 读取一些捕获的跟踪数据后将 DRAINBUF 位置 1 会引发不可预测的行为。

位 13 **STPONTRGEV**：通过触发事件的停止指示 (Stop on trigger event)

0：无影响

1：当发生触发事件时，停止跟踪捕获

在软件 FIFO 模式或硬件 FIFO 模式且该位置 1 的情况下，使能 ETF 会引发不可预测的行为。

位 12 **STOPONFL**：通过刷新的停止指示 (Stop on flush)

0：无影响

1：当完成刷新时，停止跟踪捕获

如果刷新是由 ATB 主接口启动，则完成刷新时不会导致格式化器停止，无论在该位中编程的值是多少。

位 11 保留，必须保持复位值。

位 10 **TRIGONFL**：通过刷新的触发指示 (Trigger on flush)

0：无影响

1：指示刷新完成时跟踪流存在触发

如果 ENFT 和 ENTI 都清零，则忽略该位且跟踪流中未插入触发。

如果刷新是由 ATB 主接口启动，则完成刷新时不会导致触发指示，无论在该位中编程的值是多少。

位 9 **TRGONTRGEV**：通过触发事件的触发指示 (Trigger on trigger event)

0：无影响

1：指示发生触发事件时跟踪流中存在触发

如果 ENFT 和 ENTI 都清零，则忽略该位且跟踪流中未插入触发。
该位不能在软件 FIFO 模式或硬件 FIFO 模式下使用。

位 8 **TRGONTRGIN**：通过触发输入的触发指示 (Trigger on trigger in)

0：无影响

1：指示在 TRIGIN 输入上检测到上升沿时跟踪流中存在触发。

如果 ENFT 和 ENTI 都清零，则忽略该位且跟踪流中未插入触发。

位 7 保留，必须保持复位值。

位 6 **FLUSHMAN**：手动刷新 (Manual flush)

0：无影响

1：刷新跟踪 FIFO 和管道

当刷新完成时，该位自动清零。如果 ETF_CTL 寄存器的 TCEN 位为 0，则忽略对该位的写操作。

位 5 **FONTRGEV**：通过触发事件的刷新指示 (Flush on trigger event)

0：无影响

1：如果发生触发事件，则刷新跟踪 FIFO 和管道

该位不能在软件 FIFO 模式或硬件 FIFO 模式下使用。如果 STPONTRGEV 置 1，则忽略该位。

位 4 **FONFLIN**：通过刷新输入的刷新指示 (Flush on flush in)

0：无影响

1：如果在 FLUSHIN 输入上检测到上升沿，则刷新跟踪 FIFO 和管道

位 3:2 保留，必须保持复位值。

位 1 **ENTI**：使能触发插入 (Enable trigger insertion)

将该位置 1 使能在格式化的跟踪流中插入触发。如果在跟踪流中插入了一个字节的数据 8'h00，且 ATID = 7'h7D，则指示存在触发。另外，跟踪流上的触发指示还受 FFCR 寄存器中的寄存器位 TRIGONFL、TRGONTRGEV 和 TRGONTRGIN 控制。只有当 READY 为高电平且 TCEN 为低电平时，该位才能更改。该位仅在该寄存器中的 ENFT 寄存器位置 1 时才有效。如果在 ENFT 为低电平时将 ENTI 位设置为高电平，则会导致使能格式化。

位 0 **ENFT**：使能格式化 (Enable formatting)。

0：禁止格式化。假设传入的跟踪数据源于单个跟踪源。

1：使能格式化。

当使能了跟踪捕获且禁止了格式化器时，如果 ETF 接收了多个 ATID，则会引起跟踪数据的交错。仅在循环缓冲模式下才支持禁止格式化。如果在非循环缓冲区模式且 ENFT 为低电平的情况下使能 ETF，则会导致使能格式化。如果在 ENFT 为低电平时将 ENTI 位设置为高电平，则会导致使能格式化。
当处于禁止状态时，忽略该位。

ETF 周期性同步计数器寄存器 (ETF_PSCR)

ETF periodic synchronization counter register

偏移地址：0x308

复位值：0x0000 000A

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PSCOUNT[4:0]				
											rw				

位 31:5 保留，必须保持复位值。

位 4:0 **PSCOUNT[4:0]**：同步计数器重载值 (Synchronization counter reload value)

确定同步计数器的重载值。重载值在下次计数器达到零时生效。读取该寄存器会返回编程到该寄存器中的重载值。一旦复位，该寄存器会被置为 0xA，对应于 1024 个字节的同步周期。

0x0：禁止同步
0x1-0x6：保留
0x7-0x1B：同步周期为 $2^{PSCOUNT}$ 个字节
0x1C-0x1F：保留

将 ETF 声明标记置位寄存器 (ETF_CLAIMSET)

ETF claim tag set register

偏移地址：0xFA0

复位值：0x0000 000F

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CLAIMSET[3:0]			
												rw			

位 31:4 保留，必须保持复位值。

位 3:0 **CLAIMSET[3:0]**：置位声明标记位 (Set claim tag bits)

写：
0000：无影响
xxx1：将位 0 置 1
xx1x：将位 1 置 1
x1xx：将位 2 置 1
1xxx：将位 3 置 1

读：
0xF：表示声明标记使用四个位

ETF 声明标记清零寄存器 (ETF_CLAIMCLR)

ETF claim tag clear register

偏移地址：0xFA4

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CLAIMCLR[3:0]			
												rw			

位 31:4 保留，必须保持复位值。

位 3:0 **CLAIMCLR[3:0]**：复位声明标记位 (Reset claim tag bits)

写：
0000：无影响
xxx1：将位 0 清零
xx1x：将位 1 清零
x1xx：将位 2 清零
1xxx：将位 3 清零

读操作：返回声明标记的当前值

ETF 锁定访问寄存器 (ETF_LAR)

ETF lock access register

偏移地址：0xFB0

复位值：N/A

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ACCESS_W[31:16]															
w															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ACCESS_W[15:0]															
w															

位 31:0 **ACCESS_W[31:0]**：ETF 寄存器访问使能 (ETF register access enable)

使能处理器内核写访问某些 ETF 寄存器（调试器无需解锁组件）

0xC5ACCE55：使能写访问
其他值：禁止写访问

ETF 锁定状态寄存器 (ETF_LSR)

ETF lock status register

偏移地址：0xFB4

复位值：0x0000 0003

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LOCK TYPE	LOCK GRANT	LOCK EXIST
													r	r	r

位 31:3 保留，必须保持复位值。

位 2 **LOCKTYPE**：ETF_LAR 寄存器大小 (Size of the ETF_LAR register)

0：32 位

位 1 **LOCKGRANT**：当前锁定状态 (Current status of lock)

当外部调试器读取该位时，该位始终返回零。

0：允许写访问

1：阻止写访问。仅允许读访问。

位 0 **LOCKEXIST**：存在锁定控制机制 (Existence of lock control mechanism)

该位表示是否存在锁定控制机制。当外部调试器读取该位时，该位始终返回零。

0：不存在锁定控制机制

1：存在锁定控制机制

ETF 认证状态寄存器 (ETF_AUTHSTAT)

ETF authentication status register

偏移地址：0xFB8

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SNID[1:0]		SID[1:0]		NSNID[1:0]		NSID[1:0]	
								r		r		r		r	

位 31:8 保留，必须保持复位值。

位 7:6 **SNID[1:0]**：安全非侵入式调试的安全等级 (Security level for secure non-invasive debug)
0x0：未设计

位 5:4 **SID[1:0]**：安全侵入式调试的安全等级 (Security level for secure invasive debug)
0x0：未设计

位 3:2 **NSNID[1:0]**：非安全非侵入式调试的安全等级 (Security level for non-secure non-invasive debug)
0x0：未设计

位 1:0 **NSID[1:0]**：非安全侵入式调试的安全等级 (Security level for non-secure invasive debug)
0x0：未设计

ETF 设备配置寄存器 (ETF_DEVID)

ETF device configuration register

偏移地址：0xFC8

复位值：0x0000 01C0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	MEMWIDTH[2:0]			CONFIGTYP[1:0]		CLK SCHEM	ATBINPORTCNT[4:0]				
					r			r		r	r				

位 31:11 保留，必须保持复位值。

位 10:8 **MEMWIDTH[2:0]**：存储器接口数据总线宽度 (Memory interface data bus width)
0x3：64 位（对应于 32 位 ATB 数据）

位 7:6 **CONFIGTYP[1:0]**：组件（ETB、ETR 或 ETF）的配置类型 (Configuration type of component (ETB, ETR or ETF))
0x2：ETF

位 5 **CLKSCHEM**：RAM 时钟方案（同步或异步）(RAM clocking scheme (synchronous or asynchronous))
0：同步

位 4:0 **ATBINPORTCNT**[4:0]：ATB 输入端口多路复用的数量/类型 (Number/type of ATB input port multiplexing)
0x0：无

ETF 器件类型标识寄存器 (ETF_DEVTYPE)

ETF device type identifier register

偏移地址：0xFCC

复位值：0x0000 0032

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SUBTYPE[3:0]				MAJORTYPE[3:0]			
								r				r			

位 31:8 保留，必须保持复位值。

位 7:4 **SUBTYPE[3:0]**：子分类 (Sub-classification)

0x3：将跟踪数据从 ATB 从接口捕获到可通过 ATB 主接口排空的 RAM 中

位 3:0 **MAJORTYPE[3:0]**：主分类 (Major classification)

0x2：组件为一个跟踪链路，因为它具有一个 ATB 主接口，在硬件 FIFO 模式下可通过该接口排空跟踪数据。

ETF CoreSight 外设标识寄存器 4 (ETF_PIDR4)

ETF CoreSight peripheral identity register 4

偏移地址：0xFD0

复位值：0x0000 0004

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	4KCOUNT[3:0]				JEP106CON[3:0]			
								r				r			

位 31:8 保留，必须保持复位值。

位 7:4 **4KCOUNT[3:0]**：寄存器文件大小 (Register file size)

0x0：寄存器文件占用单个 4 KB 区域

位 3:0 **JEP106CON[3:0]**：JEP106 连续代码 (JEP106 continuation code)

0x4：ARM® JEDEC 代码

ETF CoreSight 外设标识寄存器 0 (ETF_PIDR0)

ETF CoreSight peripheral identity register 0

偏移地址：0xFE0

复位值：0x0000 0061

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PARTNUM[7:0]							
								r							

位 31:8 保留，必须保持复位值。

位 7:0 **PARTNUM[7:0]**：产品编号字段 (Part number field)，位 [7:0]
0x61：ETF 产品编号

ETF CoreSight 外设标识寄存器 1 (ETF_PIDR1)

ETF CoreSight peripheral identity register 1

偏移地址：0xFE4

复位值：0x0000 00B9

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	JEP106ID[3:0]				PARTNUM[11:8]			
								r				r			

位 31:8 保留，必须保持复位值。

位 7:4 **JEP106ID[3:0]**：JEP106 标识代码字段 (JEP106 identity code field)，位 [3:0]
0xB：ARM® JEDEC 代码

位 3:0 **PARTNUM[11:8]**：产品编号字段 (Part number field)，位 [11:8]
0x9：ETF 产品编号

ETF CoreSight 外设标识寄存器 2 (ETF_PIDR2)

ETF CoreSight peripheral identity register 2

偏移地址：0xFE8

复位值：0x0000 001F

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REVISION[3:0]				JEDEC	JEP106ID[6:4]		
								r				r	r		

位 31:8 保留，必须保持复位值。

位 7:4 **REVISION[3:0]**：组件版本号 (Component revision number)
0x1：r0p1

位 3 **JEDEC**：JEDEC 分配的值 (JEDEC assigned value)
1：JEDEC 指定的设计人员 ID

位 2:0 **JEP106ID[6:4]**：JEP106 标识代码字段 (JEP106 identity code field)，位 [6:4]
0x3：ARM® JEDEC 代码

ETF CoreSight 外设标识寄存器 3 (ETF_PIDR3)

ETF CoreSight peripheral identity register 3

偏移地址：0xFEC

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REVAND[3:0]				CMOD[3:0]			
								r				r			

位 31:8 保留，必须保持复位值。

位 7:4 **REVAND[3:0]**：金属修复版本 (Metal fix version)
0x0：无金属修复

位 3:0 **CMOD[3:0]**：客户修改 (Customer modified)
0x0：无客户修改

ETF CoreSight 组件标识寄存器 0 (ETF_CIDR0)

ETF CoreSight component identity register 0

偏移地址：0xFF0

复位值：0x0000 000D

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[7:0]							
								r							

位 31:8 保留，必须保持复位值。

位 7:0 **PREAMBLE[7:0]**：组件 ID 字段 (Component ID field)，位 [7:0]
0x0D：公共 ID 值

ETF CoreSight 组件标识寄存器 1 (ETF_CIDR1)

ETF CoreSight component identity register 1

偏移地址：0xFF4

复位值：0x0000 0090

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CLASS[3:0]				PREAMBLE[11:8]			
								r				r			

位 31:8 保留，必须保持复位值。

位 7:4 **CLASS[3:0]**：组件 ID 字段 (Component ID field)，位 [15:12]——组件类
0x9：CoreSight 组件

位 3:0 **PREAMBLE[11:8]**：组件 ID 字段 (Component ID field)，位 [11:8]
0x0：公共 ID 值

ETF CoreSight 组件标识寄存器 2 (ETF_CIDR2)

ETF CoreSight component identity register 2

偏移地址：0xFF8

复位值：0x0000 0005

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[19:12]							
								r							

位 31:8 保留，必须保持复位值。

位 7:0 **PREAMBLE[19:12]**：组件 ID 字段 (Component ID field)，位 [23:16]
0x05：公共 ID 值

ETF CoreSight 组件标识寄存器 3 (ETF_CIDR3)

ETF CoreSight component identity register 3

偏移地址：0xFFC

复位值：0x0000 00B1

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[27:20]							
								r							

位 31:8 保留，必须保持复位值。

位 7:0 **PREAMBLE[27:20]**：组件 ID 字段 (Component ID field)，位 [31:24]
0xB1：公共 ID 值

ETF 寄存器映射和复位值

表 572. ETF 寄存器映射和复位值

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x004	ETF_RSZ	Res.	RSZ[30:0]																														
	Reset value		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
0x00C	ETF_STS	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	EMPTY	FTEMPTY	READY	TRIGD	FULL
	Reset value																												1	1	1	0	0
0x010	ETF_RRD	RRD[31:0]																															
	Reset value																																
0x014	ETF_RRP	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RRP[12:0]												
	Reset value																				0	0	0	0	0	0	0	0	0	0	0	0	0
0x018	ETF_RWP	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RWP[12:0]												
	Reset value																				0	0	0	0	0	0	0	0	0	0	0	0	0
0x01C	ETF_TRG	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TRG[10:0]										
	Reset value																						0	0	0	0	0	0	0	0	0	0	0
0x020	ETF_CTL	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TCEN
	Reset value																																0
0x024	ETF_RWD	RWD[31:0]																															
	Reset value																																
0x028	ETF_MODE	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MODE[1:0]	
	Reset value																															0	0
0x02C	ETF_LBUFLVL	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LBUFLEVEL[11:0]											
	Reset value																					0	0	0	0	0	0	0	0	0	0	0	0
0x030	ETF_CBUFLVL	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CBUFLEVEL[11:0]											
	Reset value																					0	0	0	0	0	0	0	0	0	0	0	0
0x034	ETF_BUFWM	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BUFWM[10:0]										
	Reset value																						0	0	0	0	0	0	0	0	0	0	0
0x300	ETF_FFSR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FTSTOPPED	FLINPROG
	Reset value																															1	0
0x304	ETF_FFCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DRAINBUF	STPONTRGEV	STOPONFL	Res.	TRGONFL	TRGONTRGEV	TRGONTRGIN	Res.	FLUSHMAN	FONTRGEV	FONFLIN	Res.	Res.	ENTI	ENFT
	Reset value																		0	0	0		0	0	0		0	0	0			0	0
0x308	ETF_PSCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PSCOUNT[4:0]				
	Reset value																												0	1	0	1	0

表 572. ETF 寄存器映射和复位值（续）

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0xFA0	ETF_CLAIMSET	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CLAIMSET [3:0]			
	Reset value																													1	1	1	1
0xFA4	ETF_CLAIMCLR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CLAIMCLR [3:0]			
	Reset value																													0	0	0	0
0xFB0	ETF_LAR	ACCESS_W[31:0]																															
	Reset value																																
0xFB4	ETF_LSR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LOCKTYPE	LOCKGRANT	LOCKEXIST
	Reset value																														0	1	1
0xFB8	ETF_AUTHSTAT	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SNID [1:0]		SID [1:0]		NSNID [1:0]		NSID [1:0]	
	Reset value																									0	0	0	0	0	0	0	0
0xFC8	ETF_DEVID	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MEMWDTH [2:0]			CONFIGTYP [1:0]		CLKSCHEM	ATBINPORTCNT [4:0]				
	Reset value																						0	1	1	1	0	0	0	0	0	0	0
0xFD0	ETF_DEVTYPE	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SUBTYPE [3:0]				MAJORTYPE [3:0]			
	Reset value																									0	0	1	1	0	0	1	0
0xFD0	ETF_PIDR4	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	4KCOUNT [3:0]				JEP106CON [3:0]			
	Reset value																									0	0	0	0	0	1	0	0
0xFD4	ETF_PIDR5	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	Reset value																																
0xFD8	ETF_PIDR6	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	Reset value																																
0xFDC	ETF_PIDR7	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	Reset value																																
0xFE0	ETF_PIDR0	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PARTNUM[7:0]							
	Reset value																									0	1	1	0	0	0	0	1
0xFE4	ETF_PIDR1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	JEP106ID [3:0]				PARTNUM [11:8]			
	Reset value																									1	0	1	1	1	0	0	1
0xFE8	ETF_PIDR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REVISION [3:0]				JEDEC	JEP106ID [6:4]		
	Reset value																									0	0	0	1	1	0	1	1

表 572. ETF 寄存器映射和复位值（续）

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0xFEC	ETF_PIDR3	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REVAND[3:0]				CMOD[3:0]			
	Reset value																									0	0	0	0	0	0	0	0
0xFF0	ETF_CIDR0	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[7:0]							
	Reset value																									0	0	0	0	1	1	0	1
0xFF4	ETF_CIDR1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CLASS[3:0]				PREAMBLE [11:8]			
	Reset value																									1	0	0	1	0	0	0	0
0xFF8	ETF_CIDR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[19:12]							
	Reset value																									0	0	0	0	0	1	0	1
0xFFC	ETF_CIDR3	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[27:20]							
	Reset value																									1	0	1	1	0	0	0	1

60.5.6 跟踪端口接口单元 (TPIU)

TPIU 是一款 CoreSight™ 组件，用来对跟踪流进行格式化处理并将其输出到外部跟踪端口信号上。TPIU 具有单个 ATB 从端口，用于传入跟踪数据。跟踪端口为同步并行端口，包括一个时钟输出 TRACECK 和四个数据输出 TRACED(7:0)。跟踪端口宽度的可编程范围为 1 到 8。使用较小的端口宽度可减少所需测试点/连接器引脚的数量，可将腾出的 IO 用于其它用途。然而，这限制了跟踪端口的带宽，从而限制了可以实时输出的跟踪信息的数量。必须先通过将 DBGMCU 控制寄存器 TRACECLKEN 位置 1 使能了 TRACECK 输出，才能将跟踪数据发送到 TPIU。此外，可在 RCC 中编程 TRACECK 频率。

有关跟踪端口接口 CoreSight™ 组件的更多信息，请参见《ARM® CoreSight™ SoC-400 技术参考手册》*[2]*。

TPIU 寄存器

TPIU 支持的端口大小寄存器 (TPIU_SUPPSIZE)

TPIU supported port size register

偏移地址：0x000

复位值：0x0000 000F

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PORTSIZE[31:16]															
r															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PORTSIZE[15:0]															
r															

位 31:0 **PORTSIZE[31:0]**：指示支持的跟踪端口大小，从 1 到 32 个引脚 (Indicates supported trace port sizes, from 1 to 32 pins)。位 n-1 置 1 指示支持端口大小 n。

0x0000 000F：支持端口大小 1 到 4

TPIU 当前端口大小寄存器 (TPIU_CURPSIZE)

TPIU current port size register

偏移地址：0x004

复位值：0x0000 0001

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PORTSIZE[31:16]															
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PORTSIZE[15:0]															
rw															

位 31:0 **PORTSIZE[31:0]**：指示当前跟踪端口大小 (Indicates current trace port size)

位 n-1 置 1 指示当前端口大小为 n 个引脚。n 的值必须在支持的端口大小范围 (1-4) 内。只能将一个位置 1，否则可能导致不可预测的行为。只有当格式化器停止时，才应修改该寄存器。

TPIU 支持的触发模式寄存器 (TPIU_SUPTRGM)

TPIU supported trigger modes register

偏移地址：0x100

复位值：0x0000 011F

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TRG RUN	TRGD
														r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	TCOUNT8	Res.	Res.	Res.	MULT 64K	MULT 256	MULT 16	MULT4	MULT2
							r				r	r	r	r	r

位 31:18 保留，必须保持复位值。

位 17 **TRGRUN**：触发运行中 (Trigger running)

0：未发生触发或计数器为 0

1：已发生触发且计数器不为 0

位 16 **TRIGD**：触发 (Triggered)

0：未发生触发

1：已发生触发且计数器已达到 0

位 15:9 保留，必须保持复位值。

位 8 **TCOUNT8**：8 位计数器寄存器 (8-bit counter register)

1：已设计

位 7:5 保留，必须保持复位值。

位 4 **MULT64K**：支持将触发计数器乘以 65536 (Multiplying the trigger counter by 65536 support)

1：支持

位 3 **MULT256**：支持将触发计数器乘以 256 (Multiplying the trigger counter by 256 support)
1：支持

位 2 **MULT16**：支持将触发计数器乘以 16 (Multiplying the trigger counter by 16 support)
1：支持

位 1 **MULT4**：支持将触发计数器乘以 4 (Multiplying the trigger counter by 4 support)
1：支持

位 0 **MULT2**：支持将触发计数器乘以 2 (Multiplying the trigger counter by 2 support)
1：支持

TPIU 触发计数器值寄存器 (TPIU_TRGCNT)

TPIU trigger counter value register

偏移地址：0x104

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TRIGCOUNT[7:0]							
								rw							

位 31:8 保留，必须保持复位值。

位 7:0 **TRIGCOUNT[7:0]**：使能触发器延迟指示 (Enable trigger delay indication)
使能将触发指示延迟到任意外部连接的跟踪捕获或存储器件。该计数器只有八位宽，并且仅与触发乘数寄存器 0x108 中的计数器乘数一起使用。当触发启动时，该值（连同乘数一起）为指示触发之前的字数。当触发计数器达到 0 时，在此处写入的值将被重新加载。写入该寄存器会使触发计数器值复位，但不会复位乘数上的任何值。读取该寄存器会返回预设值，而不是当前的计数。

TPIU 触发乘数寄存器 (TPIU_TRGMULT)

TPIU trigger multiplier register

偏移地址：0x108

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MULT 64K	MULT 256	MULT 16	MULT4	MULT2
											rw	rw	rw	rw	rw

位 31:5 保留，必须保持复位值。

位 4 **MULT64K**：将触发计数器乘以 65536 (Multiply the trigger counter by 65536)
0：禁止
1：使能

位 3 **MULT256**：将触发计数器乘以 256 (Multiply the trigger counter by 256)
0：禁止
1：使能

位 2 **MULT16**：将触发计数器乘以 16 (Multiply the trigger counter by 16)
0：禁止
1：使能

位 1 **MULT4**：将触发计数器乘以 4 (Multiply the trigger counter by 4)
0：禁止
1：使能

位 0 **MULT2**：将触发计数器乘以 2 (Multiply the trigger counter by 2)
0：禁止
1：使能

TPIU 支持的测试模式寄存器 (TPIU_SUPTPM)

TPIU supported test patterns/modes register

偏移地址：0x200

复位值：0x0003 000F

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PCONTEN	PTIMEEN
														r	r

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PATF0	PATA5	PATW0	PATW1
												r	r	r	r

位 31:18 保留，必须保持复位值。

位 17 **PCONTEN**：支持连续模式 (Support of continuous mode)
1：支持

位 16 **PTIMEEN**：支持计时模式 (Support of timed mode)
1：支持

位 15:4 保留，必须保持复位值。

位 3 **PATF0**：支持 FF/00 模式 (Support of FF/00 pattern)
指示是否支持 FF/00 模式的跟踪端口输出。
1：支持

位 2 **PATA5**：支持 AA/55 模式 (Support of AA/55 pattern)
指示是否支持 AA/55 模式的跟踪端口输出。
1：支持

位 1 **PATW0**：支持走 0 模式 (Support of walking 0's pattern)
指示是否支持走 0 模式的跟踪端口输出。
1：支持

位 0 **PATW1**：支持走 1 模式 (Support of walking 1's pattern)
指示是否支持走 1 模式的跟踪端口输出。
1：支持

TPIU 当前测试模式寄存器 (TPIU_CURTPM)

TPIU current test pattern/mode register

偏移地址：0x204

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PCONTEN	PTIMEEN
														rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PATF0	PATA5	PATW0	PATW1
												rw	rw	rw	rw

位 31:18 保留，必须保持复位值。

位 17 **PCONTEN**：连续模式使能 (Continuous mode enable)
0：禁止
1：使能

位 16 **PTIMEEN**：计时模式使能 (Timed mode enable)
0：禁止
1：使能

位 15:4 保留，必须保持复位值。

位 3 **PATF0**：FF/00 模式使能 (FF/00 pattern enable)
指示是否使能 FF/00 模式的跟踪端口输出。
0：禁止
1：使能

位 2 **PATA5**：AA/55 模式使能 (AA/55 pattern is enable)
指示是否使能 AA/55 模式的跟踪端口输出。
0：禁止
1：使能

位 1 **PATW0**：走 0 模式使能 (Walking 0's pattern enable)
指示是否使能走 0 模式的跟踪端口输出。
0：禁止
1：使能

位 0 **PATW1**：走 1 模式使能 (Walking 1's pattern enable)
指示是否使能走 1 模式的跟踪端口输出。
0：禁止
1：使能

TPIU 测试模式重复计数器寄存器 (TPIU_TPRCR)

TPIU test pattern repeat counter register

偏移地址：0x208

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PATTCOUNT[7:0]							
								rw							

位 31:8 保留，必须保持复位值。

位 7:0 **PATTCOUNT[7:0]**：TRACECLKIN 周期数 (Number of TRACECLKIN cycles)

该字段提供一个 8 位计数器值，以指示某个模式在切换到下一个模式之前运行的 TRACECLKIN 周期数。

TPIU 格式化器和刷新状态寄存器 (TPIU_FFSR)

TPIU formatter and flush status register

偏移地址：0x300

复位值：0x0000 0002

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TCPRESENT	FTSTOPPED	FLINPROG
													r	r	r

位 31:3 保留，必须保持复位值。

位 2 **TCPRESENT**：TRACECTL 输出引脚可用性 (TRACECTL output pin availability)

指示可选的 TRACECTL 输出引脚是否可用。

0：该器件中不存在 TRACECTL 引脚。

位 1 **FTSTOPPED**：格式化器停止 (Formatter stopped)

格式化器收到了停止请求信号和所有跟踪数据，并发送了后置码。ATB 接口上的任何其他跟踪数据都被忽略。

0：格式化器未停止
1：格式化器已停止

位 0 **FLINPROG**：刷新正在进行 (Flush in progress)

指示 ATB 从端口上是否有正在进行的刷新。该位反映 AFVALIDS 输出的状态。刷新可由 TPIU_FFCR 寄存器中的刷新控制位启动。

0：无正在进行的刷新
1：正在进行刷新

TPIU 格式化器和刷新控制寄存器 (TPIU_FFCR)

TPIU formatter and flush control register

偏移地址：0x304

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	STOP TRIG	STOPF L	Res.	TRIG FL	TRIG EVT	TRIGIN	Res.	FON MAN	FON TRIG	FON FLIN	Res.	Res.	ENF CONT	EN FTC
		rw	rw		rw	rw	rw		rw	rw	rw			rw	rw

位 31:14 保留，必须保持复位值。

位 13 **STOPTRIG**：通过触发事件的停止指示 (Stop on trigger event)

0：无影响
1：当触发事件发生时，停止格式化器

位 12 **STOPFL**：通过刷新的停止指示 (Stop on flush)

0：无影响
1：当完成刷新时，停止格式化器

位 11 保留，必须保持复位值。

位 10 **TRIGFL**：通过刷新的触发指示 (Trigger on flush)

0：无影响
1：指示刷新完成时跟踪流存在触发

位 9 **TRIGEVT**：通过触发事件的触发指示 (Trigger on trigger event)

0：无影响
1：指示发生触发事件时跟踪流中存在触发

位 8 **TRIGIN**：通过触发输入的触发指示 (Trigger on trigger in)

0：无影响
1：指示当系统 CTI 的 TRIGIN 输入被置为有效时，跟踪流中存在触发。

位 7 保留，必须保持复位值。

位 6 **FONMAN**：生成手动刷新 (Generate a manual flush)

0：无影响
1：刷新跟踪

当刷新完成时，该位自动清零。

位 5 **FONTRIG**：通过触发事件的刷新指示 (Flush on trigger event)

当触发计数器达到 0 时，或者触发计数器为 0，且系统 CTI 的 TRIGIN 输入为高电平时，触发事件会发生。

0：无影响

1：如果触发事件发生，则刷新跟踪

位 4 **FONFLIN**：通过刷新输入的刷新指示 (Flush on flush in)

0：无影响

1：当系统 CTI 的 TRIGIN 输入被置为有效时，刷新跟踪

位 3:2 保留，必须保持复位值。

位 1 **ENFCONT**：使能连续格式化 (Enable continuous formatting)

0：禁止连续格式化

1：使能连续格式化

位 0 **ENFTC**：使能在格式化的跟踪数据中嵌入触发 (Enable the embedding of triggers in formatted trace)

0：禁止格式化

1：使能格式化

TPIU 格式化器同步计数器寄存器 (TPIU_FSCR)

TPIU formatter synchronization counter register

偏移地址：0x308

复位值：0x0000 0040

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CYCCOUNT[4:0]				
											rw				

位 31:5 保留，必须保持复位值。

位 4:0 **CYCCOUNT[4:0]**：使能有效使用 TPA (Enables effective use of TPAs)

使能有效使用不同大小的 TPA，而不会浪费捕获器件的大量存储容量。该计数器包含自最上一个 128 位同步数据包以来的格式化帧数。它是一个 12 位计数器，最大计数器值为 4096。相当于每 65536 个字节（即 4096 个数据包 x 16 字节/数据包）同步一次。默认设置为每 1024 个字节（即每 64 个格式化帧）传输一个同步数据包。如果格式化器配置为连续模式，则在正常运行情况下插入全字和半字同步帧。在这些情况下，计数器值为完全同步数据包间的最大完整帧数。

将 TPIU 声明标记置位寄存器 (TPIU_CLAIMSET)

TPIU claim tag set register

偏移地址：0xFA0

复位值：0x0000 000F

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CLAIMSET[3:0]			
												rw			

位 31:4 保留，必须保持复位值。

位 3:0 **CLAIMSET[3:0]**：置位声明标记位 (Set claim tag bits)

写：
0000：无影响
xxx1：将位 0 置 1
xx1x：将位 1 置 1
x1xx：将位 2 置 1
1xxx：将位 3 置 1

读：
0xF：表示声明标记使用四个位

TPIU 声明标记清零寄存器 (TPIU_CLAIMCLR)

TPIU claim tag clear register

偏移地址：0xFA4

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CLAIMCLR[3:0]			
												rw			

位 31:4 保留，必须保持复位值。

位 3:0 **CLAIMCLR[3:0]**：复位声明标记位 (Reset claim tag bits)

写：
0000：无影响
xxx1：将位 0 清零
xx1x：将位 1 清零
x1xx：将位 2 清零
1xxx：将位 3 清零

读操作：返回声明标记的当前值

TPIU 锁定访问寄存器 (TPIU_LAR)

TPIU lock access register

偏移地址：0xFB0

复位值：N/A

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ACCESS_W[31:15]															
w															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ACCESS_W[15:0]															
w															

位 31:0 **ACCESS_W[31:0]**：TPIU 寄存器访问使能 (TPIU register access enable)
使能处理器内核写访问某些 TPIU 寄存器（调试器无需解锁组件）

0xC5ACCE55：使能写访问
其他值：禁止写访问

TPIU 锁定状态寄存器 (TPIU_LSR)

TPIU lock status register

偏移地址：0xFB4

复位值：0x0000 0003

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LOCK TYPE	LOCK GRANT	LOCK EXIST
													r	r	r

位 31:3 保留，必须保持复位值。

位 2 **LOCKTYPE**：TPIU_LAR 寄存器大小 (Size of the TPIU_LAR register)
0：32 位

位 1 **LOCKGRANT**：当前锁定状态 (Current status of lock)
当外部调试器读取该位时，该位始终返回零。

0：允许写访问
1：阻止写访问。仅允许读访问。

位 0 **LOCKEXIST**：存在锁定控制机制 (Existence of lock control mechanism)
该位表示是否存在锁定控制机制。当外部调试器读取该位时，该位始终返回零。

0：不存在锁定控制机制
1：存在锁定控制机制

TPIU 认证状态寄存器 (TPIU_AUTHSTAT)

TPIU authentication status register

偏移地址：0xFB8

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SNID[1:0]		SID[1:0]		NSNID[1:0]		NSID[1:0]	
								r		r		r		r	

位 31:8 保留，必须保持复位值。

位 7:6 **SNID[1:0]**：安全非侵入式调试的安全等级 (Security level for secure non-invasive debug)
0x0：未设计

位 5:4 **SID[1:0]**：安全侵入式调试的安全等级 (Security level for secure invasive debug)
0x0：未设计

位 3:2 **NSNID[1:0]**：非安全非侵入式调试的安全等级 (Security level for non-secure non-invasive debug)
0x0：未设计

位 1:0 **NSID[1:0]**：非安全侵入式调试的安全等级 (Security level for non-secure invasive debug)
0x0：未设计

TPIU 设备配置寄存器 (TPIU_DEVID)

TPIU device configuration register

偏移地址：0xFC8

复位值：0x0000 00A0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	SWO UART NRZ	SWO MAN	TCLK DATA	FIFO SIZE[2:0]			CLK RELAT	MAXNUM[3:0]				
				r	r	r	r			r	r				

位 31:12 保留，必须保持复位值。

位 11 **SWOUARTNRZ**：支持 SWO UART 或 NRZ (Support of SWO UART or NRZ)
指示是否支持串行线输出 UART 或 NRZ。

0：不支持

位 10 **SWOMAN**：支持 SWO 曼彻斯特格式 (Support of SWO Manchester format)
表示是否支持曼彻斯特编码格式的串行线输出。

0：不支持

位 9 **TCLKDATA**：支持跟踪时钟加数据 (Support of trace clock plus data)
0：不支持

位 8:6 **FIFOSIZE[2:0]**：以 2 的幂表示的 FIFO 大小 (FIFO size in powers of 2)
0x2：FIFO 大小 = 4（16 字节）

位 5 **CLKRELAT**：ATB 时钟和 TRACECLKIN 关系 (ATB clock and TRACECLKIN relation)
指示 ATB 时钟和 TRACECLKIN 之间的关系（同步或异步）

1：异步

位 4:0 **MAXNUM[4:0]**：ATB 输入端口多路复用的数量/类型 (Number/type of ATB input port multiplexing)
0x0：无

TPIU 器件类型标识寄存器 (TPIU_DEVTYPE)

TPIU device type identifier register

偏移地址：0xFCC

复位值：0x0000 0011

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SUBTYPE[3:0]				MAJORTYPE[3:0]			
								r				r			

位 31:8 保留，必须保持复位值。

位 7:4 **SUBTYPE[3:0]**：子分类 (Sub-classification)
0x1：跟踪端口组件

位 3:0 **MAJORTYPE[3:0]**：主分类 (Major classification)
0x1：跟踪 Sink 组件

TPIU CoreSight 外设标识寄存器 4 (TPIU_PIDR4)

TPIU CoreSight peripheral identity register 4

偏移地址：0xFD0

复位值：0x0000 0004

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	4KCOUNT[3:0]				JEP106CON[3:0]			
								r				r			

位 31:8 保留，必须保持复位值。

位 7:4 **4KCOUNT[3:0]**：寄存器文件大小 (Register file size)
0x0：寄存器文件占用单个 4 KB 区域

位 3:0 **JEP106CON[3:0]**：JEP106 连续代码 (JEP106 continuation code)
0x4：ARM® JEDEC 代码

TPIU CoreSight 外设标识寄存器 0 (TPIU_PIDR0)

TPIU CoreSight peripheral identity register 0

偏移地址：0xFE0

复位值：0x0000 0012

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PARTNUM[7:0]							
								r							

位 31:8 保留，必须保持复位值。

位 7:0 **PARTNUM[7:0]**：产品编号字段 (Part number field)，位 [7:0]
0x12：TPIU 产品编号

TPIU CoreSight 外设标识寄存器 1 (TPIU_PIDR1)

TPIU CoreSight peripheral identity register 1

偏移地址：0xFE4

复位值：0x0000 00B9

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	JEP106ID[3:0]				PARTNUM[11:8]			
								r				r			

位 31:8 保留，必须保持复位值。

位 7:4 **JEP106ID[3:0]**：JEP106 标识代码字段 (JEP106 identity code field)，位 [3:0]
0xB：ARM® JEDEC 代码

位 3:0 **PARTNUM[11:8]**：产品编号字段 (Part number field)，位 [11:8]
0x9：TPIU 产品编号

TPIU CoreSight 外设标识寄存器 2 (TPIU_PIDR2)

TPIU CoreSight peripheral identity register 2

偏移地址：0xFE8

复位值：0x0000 004B

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REVISION[3:0]				JEDEC	JEP106ID[6:4]		
								r				r	r		

位 31:8 保留，必须保持复位值。

位 7:4 **REVISION[3:0]**：组件版本号 (Component revision number)
0x4：r0p5

位 3 **JEDEC**：JEDEC 分配的值 (JEDEC assigned value)
1：JEDEC 指定的设计人员 ID

位 2:0 **JEP106ID[6:4]**：JEP106 标识代码字段 (JEP106 identity code field)，位 [6:4]
0x3：ARM® JEDEC 代码

TPIU CoreSight 外设标识寄存器 3 (TPIU_PIDR3)

TPIU CoreSight peripheral identity register 3

偏移地址：0xFEC

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REVAND[3:0]				CMOD[3:0]			
								r				r			

位 31:8 保留，必须保持复位值。

位 7:4 **REVAND[3:0]**：金属修复版本 (Metal fix version)
0x0：无金属修复

位 3:0 **CMOD[3:0]**：客户修改 (Customer modified)
0x0：无客户修改

TPIU CoreSight 组件标识寄存器 0 (TPIU_CIDR0)

TPIU CoreSight component identity register 0

偏移地址：0xFF0

复位值：0x0000 000D

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[7:0]							
								r							

位 31:8 保留，必须保持复位值。

位 7:0 **PREAMBLE[7:0]**：组件 ID 字段 (Component ID field)，位 [7:0]
0x0D：公共 ID 值

TPIU CoreSight 组件标识寄存器 1 (TPIU_CIDR1)

TPIU CoreSight component identity register 1

偏移地址：0xFF4

复位值：0x0000 0090

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CLASS[3:0]				PREAMBLE[11:8]			
								r				r			

位 31:8 保留，必须保持复位值。

位 7:4 **CLASS[3:0]**：组件 ID 字段 (Component ID field)，位 [15:12]——组件类
0x9：CoreSight 组件

位 3:0 **PREAMBLE[11:8]**：组件 ID 字段 (Component ID field)，位 [11:8]
0x0：公共 ID 值

TPIU CoreSight 组件标识寄存器 2 (TPIU_CIDR2)

TPIU CoreSight component identity register 2

偏移地址：0xFF8

复位值：0x0000 0005

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[19:12]							
								r							

位 31:8 保留，必须保持复位值。

位 7:0 **PREAMBLE[19:12]**：组件 ID 字段 (Component ID field)，位 [23:16]
0x05：公共 ID 值

TPIU CoreSight 组件标识寄存器 3 (TPIU_CIDR3)

TPIU CoreSight component identity register 3

偏移地址：0xFFC

复位值：0x0000 00B1

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[27:20]							
								r							

位 31:8 保留，必须保持复位值。

位 7:0 **PREAMBLE[27:20]**：组件 ID 字段 (Component ID field)，位 [31:24]
0xB1：公共 ID 值

TPIU 寄存器映射和复位值

表 573. TPIU 寄存器映射和复位值

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x000	TPIU_SUPPSIZE	PORTSIZE[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
0x004	TPIU_CURPSIZE	PORTSIZE[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
0x100	TPIU_SUPTRGM	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TRGRUN	TRIGD	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TCOUNT8	Res.	Res.	Res.	MULT64K	MULT256	MULT16	MULT4	MULT2
	Reset value															0	0								1				1	1	1	1	1
0x104	TPIU_TRGCNT	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TRIGCOUNT[7:0]							
	Reset value																									0	0	0	0	0	0	0	0
0x108	TPIU_TRGMULT	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MULT64K	MULT256	MULT16	MULT4	MULT2
	Reset value																												0	0	0	0	0
0x200	TPIU_SUPTPM	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PCONTEN	PTIMEEN	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PATF0	PATA5	PATW0	PATW1
	Reset value															1	1													1	1	1	1
0x204	TPIU_CURTPM	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PCONTEN	PTIMEEN	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PATF0	PATA5	PATW0	PATW1
	Reset value															0	0													0	0	0	0
0x104	TPIU_TPRCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PATTCOUNT[7:0]							
	Reset value																									0	0	0	0	0	0	0	0
0x300	TPIU_FFSR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TCPRESENT	FTSTOPPED	FLINPROG
	Reset value																														0	1	0
0x304	TPIU_FFCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res	STOPTRIG	STOPFL	Res.	TRiGFL	TRiGEVt	TRIGIN	Res.	FONMAN	FONTRIG	FONFLIN	Res.	Res.	ENFCONT	ENFTC
	Reset value																			0	0		0	0	0		0	0	0			0	0
0x308	TPIU_FSCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CYCCOUNT[11:0]											
	Reset value																					0	0	0	0	0	1	0	0	0	0	0	0
0xFA0	TPIU_CLAIMSET	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CLAIMSET [3:0]			
	Reset value																													1	1	1	1
0xFA4	TPIU_CLAIMCLR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CLAIMCLR [3:0]			
	Reset value																													0	0	0	0

表 573. TPIU 寄存器映射和复位值（续）

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0xFB0	TPIU_LAR	ACCESS_W[31:0]																															
	Reset value																																
0xFB4	TPIU_LSR	Res.																													LOCKTYPE	LOCKGRANT	LOCKEXIST
	Reset value																														0	1	1
0xFB8	TPIU_AUTHSTAT	Res.																								SNID [1:0]		SID [1:0]		NSNID [1:0]		NSID [1:0]	
	Reset value																									0	0	0	0	0	0	0	0
0xFC8	TPIU_DEVID	Res.																				SWOUARTNRZ	SWOMAN	TCLKDATA	FIFOSIZE[2:0]			CLKRELAT	MUXNUM[4:0]				
	Reset value																					0	0	0	0	1	0	1	0	0	0	0	0
0xFD0	TPIU_DEVTYPE	Res																								SUBTYPE [3:0]				MAJORTYPE [3:0]			
	Reset value																									0	0	0	1	0	0	0	1
0xFD0	TPIU_PIDR4	Res.																								4KCOUNT [3:0]				JEP106CON [3:0]			
	Reset value																									0	0	0	0	0	1	0	0
0xFD4	TPIU_PIDR5	Res																															
	Reset value																																
0xFD8	TPIU_PIDR6	Res																															
	Reset value																																
0xFDC	TPIU_PIDR7	Res.																															
	Reset value																																
0xFE0	TPIU_PIDR0	Res.																								PARTNUM [7:0]							
	Reset value																									0	0	0	1	0	0	1	0
0xFE4	TPIU_PIDR1	Res.																								JEP106ID [3:0]				PARTNUM [11:8]			
	Reset value																									1	0	1	1	1	0	0	1
0xFE8	TPIU_PIDR2	Res.																								REVISION [3:0]				JEDEC	JEP106ID [6:4]		
	Reset value																									0	1	0	0	1	0	1	1
0xFEC	TPIU_PIDR3	Res.																								REVAND[3:0]				CMOD[3:0]			
	Reset value																									0	0	0	0	0	0	0	0
0xFF0	TPIU_CIDR0	Res.																								PREAMBLE[7:0]							
	Reset value																									0	0	0	0	1	1	0	1
0xFF4	TPIU_CIDR1	Res.																								CLASS[3:0]				PREAMBLE [11:8]			
	Reset value																									1	0	0	1	0	0	0	0

表 573. TPIU 寄存器映射和复位值（续）

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0xFF8	TPIU_CIDR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[19:12]							
	Reset value																									0	0	0	0	0	1	0	1
0xFFC	TPIU_CIDR3	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[27:20]							
	Reset value																									1	0	1	1	0	0	0	1

60.5.7 串行线输出 (SWO) 和 SWO 跟踪聚合器 (SWTF)

SWO 是一款 CoreSight 组件，用来对来自处理器 ITM 的跟踪流进行格式化处理并将其输出在单线 TRACESWO 输出上。必须先对 SWO 跟踪聚合器 (SWTF) 编程以使能 Cortex-M7 ITM 的跟踪总线，再使能跟踪。*表 575* 中列出了 SWTF 寄存器。

与 TPIU 相比，SWO 包含：

- 无格式化器
- 无模式发生器
- 一个 8 位 ATB 输入
- 无同步跟踪输出，即无 TRACEDATA 或 TRACECLK 引脚
- 不支持刷新，因为不需要刷新
- 不支持触发

SWO 输出支持曼彻斯特编码和 UART NRZ 格式。

有关串行线输出 CoreSight™ 组件的更多信息，请参见《ARM® CoreSight™ 组件技术参考手册》*[4]*。

SWO 寄存器

SWO 当前输出分频因子寄存器 (SWO_CODR)

SWO current output divisor register

偏移地址：0x010

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	PRESCALER[12:0]												
			rw												

位 31:13 保留，必须保持复位值。

位 12:0 **PRESCALER[12:0]**：SWO 波特率缩放 (SWO baud rate scaling)

波特率为跟踪时钟频率除以 (PRESCALER - 1)。由于波特率会瞬时发生变化，所以建议在写入该寄存器之前，先停止跟踪源并等待至端口处于空闲状态。

SWO 所选引脚协议寄存器 (SWO_SPPR)

SWO selected pin protocol register

偏移地址：0x0F0

复位值：0x0000 0001

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.							PPROT[1:0]	
														rw	

位 31:2 保留，必须保持复位值。

位 1:0 **PPROT[1:0]**：引脚协议 (Pin protocol)

0x0：保留
0x1：曼彻斯特
0x2：NRZ
0x3：保留

SWO 格式化器和刷新状态寄存器 (SWO_FFSR)

SWO formatter and flush status register

偏移地址：0x300

复位值：0x0000 0008

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FTNON STOP	TCPRE SENT	FTSTO PPED	FLIN PROG
												r	r	r	r

位 31:4 保留，必须保持复位值。

位 3 **FTNONSTOP**：更改设置，但不停止格式化器 (Change of settings without stopping formatter)

1：允许在格式化器运行时更改设置

位 2 **TCPRESENT**：SWO 上存在 TRACECTL 引脚 (TRACECTL pin present on SWO)

0：TRACECTL 引脚不存在

位 1 **FTSTOPPED**：格式化器停止 (Formatter stopped)

0：格式化器运行中
由于在该器件中无法停止 SWO 格式化器，所以该位始终返回 0。

位 0 **FLINPROG**：刷新正在进行 (Flush in progress)

0：无正在进行的刷新
由于该器件不支持 SWO 刷新，所以该位始终返回 0。

将 SWO 声明标记置位寄存器 (SWO_CLAIMSET)

SWO claim tag set register

偏移地址：0xFA0

复位值：0x0000 000F

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CLAIMSET[3:0]			
												rw			

位 31:4 保留，必须保持复位值。

位 3:0 **CLAIMSET[3:0]**：置位声明标记位 (Set claim tag bits)

写：
0000：无影响
xxx1：将位 0 置 1
xx1x：将位 1 置 1
x1xx：将位 2 置 1
1xxx：将位 3 置 1

读：
0xF：表示声明标记使用四个位

SWO 声明标记清零寄存器 (SWO_CLAIMCLR)

SWO claim tag clear register

偏移地址：0xFA4

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CLAIMCLR[3:0]			
												rw			

位 31:4 保留，必须保持复位值。

位 3:0 **CLAIMCLR[3:0]**：复位声明标记位 (Reset claim tag bits)

写：
0000：无影响
xxx1：将位 0 清零
xx1x：将位 1 清零
x1xx：将位 2 清零
1xxx：将位 3 清零

读操作：返回声明标记的当前值

SWO 锁定访问寄存器 (SWO_LAR)

SWO lock access register

偏移地址：0xFB0

复位值：N/A

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ACCESS_W[31:15]															
w															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ACCESS_W[15:0]															
w															

位 31:0 **ACCESS_W[31:0]**：SWO 寄存器写访问使能 (SWO register write access enable)
使能处理器内核写访问某些 SWO 寄存器（调试器无需解锁组件）

0xC5ACCE55：使能写访问
其他值：禁止写访问

SWO 锁定状态寄存器 (SWO_LSR)

SWO lock status register

偏移地址：0xFB4

复位值：0x0000 0003

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LOCK TYPE	LOCK GRANT	LOCK EXIST
													r	r	r

位 31:3 保留，必须保持复位值。

位 2 **LOCKTYPE**：SWO_LAR 寄存器大小 (Size of the SWO_LAR register)
0：32 位

位 1 **LOCKGRANT**：当前锁定状态 (Current status of lock)
当外部调试器读取该位时，该位始终返回零。

0：允许写访问
1：阻止写访问——仅允许读访问

位 0 **LOCKEXIST**：存在锁定控制机制 (Existence of lock control mechanism)
该位表示是否存在锁定控制机制。当外部调试器读取该位时，该位始终返回零。

0：不存在锁定控制机制
1：存在锁定控制机制

SWO 认证状态寄存器 (SWO_AUTHSTAT)

SWO authentication status register

偏移地址：0xFB8

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SNID[1:0]		SID[1:0]		NSNID[1:0]		NSID[1:0]	
								r		r		r		r	

位 31:8 保留，必须保持复位值。

位 7:6 **SNID[1:0]**：安全非侵入式调试的安全等级 (Security level for secure non-invasive debug)

0x0：未设计

位 5:4 **SID[1:0]**：安全侵入式调试的安全等级 (Security level for secure invasive debug)

0x0：未设计

位 3:2 **NSNID[1:0]**：非安全非侵入式调试的安全等级 (Security level for non-secure non-invasive debug)

0x0：未设计

位 1:0 **NSID[1:0]**：非安全侵入式调试的安全等级 (Security level for non-secure invasive debug)

0x0：未设计

SWO 设备配置寄存器 (SWO_DEVID)

SWO device configuration register

偏移地址：0xFC8

复位值：0x0000 0EA0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	SWO UART NRZ	SWO MAN	TCLK DATA	FIFO SIZE[2:0]			CLK RELAT	MAXNUM[4:0]				
				r	r	r	r			r	r				

位 31:12 保留，必须保持复位值。

位 11 **SWOUARTNRZ**：支持 SWO UART 或 NRZ (SWO UART or NRZ support)
指示是否支持串行线输出 UART 或 NRZ。

1：支持

位 10 **SWOMAN**：支持 SWO 曼彻斯特格式 (SWO Manchester format support)
表示是否支持曼彻斯特编码格式的串行线输出。
1：支持

位 9 **TCLKDATA**：支持跟踪时钟加数据 (Trace clock plus data support)
指示是否支持跟踪时钟加数据

1：支持

位 8:6 **FIFOSIZE[2:0]**：以 2 的幂表示的 FIFO 大小 (FIFO size in powers of 2)
0x2：FIFO 大小 = 4（16 字节）

位 5 **CLKRELAT**：ATB 时钟和 TRACECLKIN 关系 (ATB clock to TRACECLKIN relation)
指示 ATB 时钟和 TRACECLKIN 之间的关系（同步或异步）

1：异步

位 4:0 **MAXNUM[4:0]**：ATB 输入端口多路复用的数量/类型 (Number/type of ATB input port multiplexing)
0x0：无

SWO 器件类型标识寄存器 (SWO_DEVTYPE)

SWO device type identifier register

偏移地址：0xFCC

复位值：0x0000 0011

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SUBTYPE[3:0]				MAJORTYPE[3:0]			
								r				r			

位 31:8 保留，必须保持复位值。

位 7:4 **SUBTYPE[3:0]**：子分类 (Sub-classification)
0x1：跟踪端口组件

位 3:0 **MAJORTYPE[3:0]**：主分类 (Major classification)
0X1：跟踪 Sink 组件

SWO CoreSight 外设标识寄存器 4 (SWO_PIDR4)

SWO CoreSight peripheral identity register 4

偏移地址：0xFD0

复位值：0x0000 0004

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	4KCOUNT[3:0]				JEP106CON[3:0]			
								r				r			

位 31:8 保留，必须保持复位值。

位 7:4 **4KCOUNT[3:0]**：寄存器文件大小 (Register file size)

0x0：寄存器文件占用单个 4 KB 区域

位 3:0 **JEP106CON[3:0]**：JEP106 连续代码 (JEP106 continuation code)

0x4：ARM® JEDEC 代码

SWO CoreSight 外设标识寄存器 0 (SWO_PIDR0)

SWO CoreSight peripheral identity register 0

偏移地址：0xFE0

复位值：0x0000 0014

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PARTNUM[7:0]							
								r							

位 31:8 保留，必须保持复位值。

位 7:0 **PARTNUM[7:0]**：产品编号字段 (Part number field)，位 [7:0]

0x14：SWO 产品编号

SWO CoreSight 外设标识寄存器 1 (SWO_PIDR1)

SWO CoreSight peripheral identity register 1

偏移地址：0xFE4

复位值：0x0000 00B9

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	JEP106ID[3:0]				PARTNUM[11:8]			
								r				r			

位 31:8 保留，必须保持复位值。

位 7:4 **JEP106ID[3:0]**：JEP106 标识代码字段 (JEP106 identity code field)，位 [3:0]
0xB：ARM® JEDEC 代码

位 3:0 **PARTNUM[11:8]**：产品编号字段 (Part number field)，位 [11:8]
0x9：SWO 产品编号

SWO CoreSight 外设标识寄存器 2 (SWO_PIDR2)

SWO CoreSight peripheral identity register 2

偏移地址：0xFE8

复位值：0x0000 001B

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REVISION[3:0]				JEDEC	JEP106ID[6:4]		
								r				r	r		

位 31:8 保留，必须保持复位值。

位 7:4 **REVISION[3:0]**：组件版本号 (Component revision number)
0x1：r0p0

位 3 **JEDEC**：JEDEC 分配的值 (JEDEC assigned value)
1：JEDEC 指定的设计人员 ID

位 2:0 **JEP106ID[6:4]**：JEP106 标识代码字段 (JEP106 identity code field)，位 [6:4]
0x3：ARM® JEDEC 代码

SWO CoreSight 外设标识寄存器 3 (SWO_PIDR3)

SWO CoreSight peripheral identity register 3

偏移地址：0xFEC

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REVAND[3:0]				CMOD[3:0]			
								r				r			

位 31:8 保留，必须保持复位值。

位 7:4 **REVAND[3:0]**：金属修复版本 (Metal fix version)
0x0：无金属修复

位 3:0 **CMOD[3:0]**：客户修改 (Customer modified)
0x0：无客户修改

SWO CoreSight 组件标识寄存器 0 (SWO_CIDR0)

SWO CoreSight component identity register 0

偏移地址：0xFF0

复位值：0x0000 000D

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[7:0]							
								r							

位 31:8 保留，必须保持复位值。

位 7:0 **PREAMBLE[7:0]**：组件 ID 字段 (Component ID field)，位 [7:0]
0x0D：公共 ID 值

SWO CoreSight 组件标识寄存器 1 (SWO_CIDR1)

SWO CoreSight component identity register 1

偏移地址：0xFF4

复位值：0x0000 0090

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CLASS[3:0]				PREAMBLE[11:8]			
								r				r			

位 31:8 保留，必须保持复位值。

位 7:4 **CLASS[3:0]**：组件 ID 字段 (Component ID field)，位 [15:12]——组件类
0x9：CoreSight 组件

位 3:0 **PREAMBLE[11:8]**：组件 ID 字段 (Component ID field)，位 [11:8]
0x0：公共 ID 值

SWO CoreSight 组件标识寄存器 2 (SWO_CIDR2)

SWO CoreSight component identity register 2

偏移地址：0xFF8

复位值：0x0000 0005

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[19:12]							
								r							

位 31:8 保留，必须保持复位值。

位 7:0 **PREAMBLE[19:12]**：组件 ID 字段 (Component ID field)，位 [23:16]
0x05：公共 ID 值

SWO CoreSight 组件标识寄存器 3 (SWO_CIDR3)

SWO CoreSight component identity register 3

偏移地址：0xFFC

复位值：0x0000 00B1

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[27:20]							
								r							

位 31:8 保留，必须保持复位值。

位 7:0 **PREAMBLE[27:20]**：组件 ID 字段 (Component ID field)，位 [31:24]
0xB1：公共 ID 值

SWO 寄存器映射和复位值

表 574. SWO 寄存器映射和复位值

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x000	SWO_SPR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	Reset value																																
0x004	SWO_CPR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	Reset value																																
0x010	SWO_CODR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PRESCALER[12:0]												
	Reset value																				0	0	0	0	0	0	0	0	0	0	0	0	0
0x0F0	SWO_SPPR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PPROT [1:0]	
	Reset value																															0	1
0x100	SWO_STMR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	Reset value																																
0x200	SWO_STPR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	Reset value																																
0x300	SWO_FFSR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FTNONSTOP	TCPRESENT	FTSTOPPED	FLINPROG
	Reset value																													1	0	0	0
0x304	SWO_FFCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	Reset value																																

表 574. SWO 寄存器映射和复位值（续）

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0xFA0	SWO_CLAIMSET	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CLAIMSET [3:0]			
	Reset value																													1	1	1	1
0xFA4	SWO_CLAIMCLR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CLAIMCLR [3:0]			
	Reset value																													0	0	0	0
0xFB0	SWO_LAR	ACCESS_W[31;0]																															
	Reset value																																
0xFB4	SWO_LSR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LOCKTYPE	LOCKGRANT	LOCKEXIST
	Reset value																														0	1	1
0xFB8	SWO_AUTHSTAT	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SNID [1:0]		SID [1:0]		NSNID [1:0]		NSID [1:0]	
	Reset value																									0	0	0	0	0	0	0	0
0xFC8	SWO_DEVID	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SWOUARTNRZ	SWOMAN	TCLKDATA	FIFOSIZE[2:0]			CLKRELAT	MUXNUM[4:0]				
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	1	0	1	0	0	0	0	0
0xFD0	SWO_DEVTYPE	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SUBTYPE [3:0]				MAJORTYPE [3:0]			
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1
0xFD0	SWO_PIDR4	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	4KCOUNT [3:0]				JEP106CON [3:0]			
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
0xFD4	SWO_PIDR5	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	Reset value																																
0xFD8	SWO_PIDR6	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	Reset value																																
0xFDC	SWO_PIDR7	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	Reset value																																
0xFE0	SWO_PIDR0	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PARTNUM [7:0]							
	Reset value																									0	0	0	1	0	1	0	0
0xFE4	SWO_PIDR1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	JEP106ID [3:0]				PARTNUM [11:8]			
	Reset value																									1	0	1	1	1	0	0	1
0xFE8	SWO_PIDR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REVISION [3:0]				JEDEC	JEP106ID [6:4]		
	Reset value																									0	0	0	1	1	0	1	1

表 574. SWO 寄存器映射和复位值（续）

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0xFEC	SWO_PIDR3	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REVAND[3:0]				CMOD[3:0]			
	Reset value																									0	0	0	0	0	0	0	0
0xFF0	SWO_CIDR0	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[7:0]							
	Reset value																									0	0	0	0	1	1	0	1
0xFF4	SWO_CIDR1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CLASS[3:0]				PREAMBLE [11:8]			
	Reset value																									1	0	0	1	0	0	0	0
0xFF8	SWO_CIDR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[19:12]							
	Reset value																									0	0	0	0	0	1	0	1
0xFFC	SWO_CIDR3	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[27:20]							
	Reset value																									1	0	1	1	0	0	0	1

SWTF 寄存器

SWTF 控制寄存器 (SWTF_CTRL)

SWTF control register

偏移地址：0x000

复位值：0x0000 0300

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	MIN_HOLD_TIME[3:0]				Res.	Res.	Res.	Res.	Res.	Res.	Res.	ENS0
				rw											rw

位 31:12 保留，必须保持复位值。

位 11:8 **MIN_HOLD_TIME[3:0]**：仲裁间的事务数量 (Number of transactions between arbitrations)。

0x0：1 个事务

:

0xE：15 个事务

0xF：保留

位 7:1 保留，必须保持复位值。

位 0 **ENS0**：从端口 S0 使能 (Slave port S0 enable)

0：禁止端口

1：使能端口

SWTF 优先级寄存器 (SWTF_PRIORITY)

SWTF priority register

偏移地址：0x004

复位值：0x0000 0008

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PRIPORT0[2:0]		
													rw		

位 31:3 保留，必须保持复位值。

位 2:0 **PRIPORT0[2:0]**：从端口 S0 优先级 (Slave port S0 priority)

0：最高优先级

:

7：最低优先级

将 SWTF 声明标记置位寄存器 (SWTF_CLAIMSET)

SWTF claim tag set register

偏移地址：0xFA0

复位值：0x0000 000F

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CLAIMSET[3:0]			
												rw			

位 31:4 保留，必须保持复位值。

位 3:0 **CLAIMSET[3:0]**：置位声明标记位 (Set claim tag bits)

写：

0000：无影响

xxx1：将位 0 置 1

xx1x：将位 1 置 1

x1xx：将位 2 置 1

1xxx：将位 3 置 1

读：

0xF：表示声明标记使用四个位

SWTF 声明标记清零寄存器 (SWTF_CLAIMCLR)

SWTF claim tag clear register

偏移地址：0xFA4

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CLAIMCLR[3:0]			
												rw			

位 31:4 保留，必须保持复位值。

位 3:0 **CLAIMCLR[3:0]**：复位声明标记位 (Reset claim tag bits)

写：
0000：无影响
xxx1：将位 0 清零
xx1x：将位 1 清零
x1xx：将位 2 清零
1xxx：将位 3 清零

读操作：返回声明标记的当前值

SWTF 锁定访问寄存器 (SWTF_LAR)

SWTF lock access register

偏移地址：0xFB0

复位值：N/A

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ACCESS_W[31:16]															
w															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ACCESS_W[15:0]															
w															

位 31:0 **ACCESS_W[31:0]**：SWTF 寄存器写访问使能 (SWTF register write access enable)

使能处理器内核写访问某些 SWTF 寄存器（调试器无需解锁组件）

0xC5ACCE55：使能写访问
其他值：禁止写访问

SWTF 锁定状态寄存器 (SWTF_LSR)

SWTF lock status register

偏移地址：0xFB4

复位值：0x0000 0003

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LOCK TYPE	LOCK GRANT	LOCK EXIST
													r	r	r

位 31:3 保留，必须保持复位值。

位 2 **LOCKTYPE**：SWTF_LAR 寄存器大小 (Size of the SWTF_LAR register)

0：32 位

位 1 **LOCKGRANT**：当前锁定状态 (Current status of lock)

当外部调试器读取该位时，该位始终返回零。

0：允许写访问

1：阻止写访问——仅允许读访问

位 0 **LOCKEXIST**：存在锁定控制机制 (Existence of lock control mechanism)

该位表示是否存在锁定控制机制。当外部调试器读取该位时，该位始终返回零。

0：不存在锁定控制机制

1：存在锁定控制机制

SWTF 认证状态寄存器 (SWTF_AUTHSTAT)

SWTF authentication status register

偏移地址：0xFB8

复位值：0x0000 000A

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SNID[1:0]		SID[1:0]		NSNID[1:0]		NSID[1:0]	
								r		r		r		r	

位 31:8 保留，必须保持复位值。

位 7:6 **SNID[1:0]**：安全非侵入式调试的安全等级 (Security level for secure non-invasive debug)
0x0：未设计

位 5:4 **SID[1:0]**：安全侵入式调试的安全等级 (Security level for secure invasive debug)
0x0：未设计

位 3:2 **NSNID[1:0]**：非安全非侵入式调试的安全等级 (Security level for non-secure non-invasive debug)
0x2：禁止
0x3：使能

位 1:0 **NSID[1:0]**：非安全侵入式调试的安全等级 (Security level for non-secure invasive debug)
0x2：禁止
0x3：使能

SWTF CoreSight 器件标识寄存器 (SWTF_DEVID)

SWTF CoreSight device identity register

偏移地址：0xFC8

复位值：0x0000 0022

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SCHEME[3:0]				PORTCNT[3:0]			
								r				r			

位 31:8 保留，必须保持复位值。

位 7:4 **SCHEME[3:0]**：优先级方案 (Priority scheme)
0x2：静态优先级

位 3:0 **PORTCNT[3:0]**：连接的输入端口数量 (Number of input ports connected)
0x2：两个输入端口

SWTF CoreSight 器件类型标识寄存器 (SWTF_TYPEID)

SWTF CoreSight device type identity register

偏移地址：0xFCC

复位值：0x0000 0012

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DEVTYPEID[7:0]							
								r							

位 31:8 保留，必须保持复位值。

位 7:0 **DEVTYPEID[7:0]**：器件类型标识符 (Device type identifier)
0x12：跟踪聚合器

SWTF CoreSight 外设标识寄存器 0 (SWTF_PIDR0)

SWTF CoreSight peripheral identity register 0

偏移地址：0xFE0

复位值：0x0000 0008

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PARTNUM[7:0]							
								r							

位 31:8 保留，必须保持复位值。

位 7:0 **PARTNUM[7:0]**：产品编号字段 (Part number field)，位 [7:0]
0x08：SWTF 产品编号

SWTF CoreSight 外设标识寄存器 1 (SWTF_PIDR1)

SWTF CoreSight peripheral identity register 1

偏移地址：0xFE4

复位值：0x0000 00B9

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	JEP106ID[3:0]				PARTNUM[11:8]			
								r				r			

位 31:8 保留，必须保持复位值。

位 7:4 **JEP106ID[3:0]**：JEP106 标识代码字段 (JEP106 identity code field)，位 [3:0]
0xB：ARM® JEDEC 代码

位 3:0 **PARTNUM[11:8]**：产品编号字段 (Part number field)，位 [11:8]
0x9：SWTF 产品编号

SWTF CoreSight 外设标识寄存器 2 (SWTF_PIDR2)

SWTF CoreSight peripheral identity register 2

偏移地址：0xFE8

复位值：0x0000 001B

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REVISION[3:0]				JEDEC	JEP106ID[6:4]		
								r				r	r		

位 31:8 保留，必须保持复位值。

位 7:4 **REVISION[3:0]**：组件版本号 (Component revision number)
0x1：r0p1

位 3 **JEDEC**：JEDEC 分配的值 (JEDEC assigned value)
1：JEDEC 指定的设计人员 ID

位 2:0 **JEP106ID[6:4]**：JEP106 标识代码字段 (JEP106 identity code field)，位 [6:4]
0x3：ARM® JEDEC 代码

SWTF CoreSight 外设标识寄存器 3 (SWTF_PIDR3)

SWTF CoreSight peripheral identity register 3

偏移地址：0xFEC

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REVAND[3:0]				CMOD[3:0]			
								r				r			

位 31:8 保留，必须保持复位值。

位 7:4 **REVAND[3:0]**：金属修复版本 (Metal fix version)
0x0：无金属修复

位 3:0 **CMOD[3:0]**：客户修改 (Customer modified)
0x0：无客户修改

SWTF CoreSight 外设标识寄存器 4 (SWTF_PIDR4)

SWTF CoreSight peripheral identity register 4

偏移地址：0xFD0

复位值：0x0000 0004

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	4KCOUNT[3:0]				JEP106CON[3:0]			
								r				r			

位 31:8 保留，必须保持复位值。

位 7:4 **4KCOUNT[3:0]**：寄存器文件大小 (Register file size)
0x0：寄存器文件占用单个 4 KB 区域

位 3:0 **JEP106CON[3:0]**：JEP106 连续代码 (JEP106 continuation code)
0x4：ARM® JEDEC 代码

SWTF CoreSight 组件标识寄存器 0 (SWTF_CIDR0)

SWTF CoreSight component identity register 0

偏移地址：0xFF0

复位值：0x0000 000D

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[7:0]							
								r							

位 31:8 保留，必须保持复位值。

位 7:0 **PREAMBLE[7:0]**：组件 ID 字段 (Component ID field)，位 [7:0]
0x0D：公共 ID 值

SWTF CoreSight 组件标识寄存器 1 (SWTF_CIDR1)

SWTF CoreSight component identity register 1

偏移地址：0xFF4

复位值：0x0000 0090

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CLASS[3:0]				PREAMBLE[11:8]			
								r				r			

位 31:8 保留，必须保持复位值。

位 7:4 **CLASS[3:0]**：组件 ID 字段 (Component ID field)，位 [15:12]——组件类
0x9：CoreSight 组件

位 3:0 **PREAMBLE[11:8]**：组件 ID 字段 (Component ID field)，位 [11:8]
0x0：公共 ID 值

SWTF CoreSight 组件标识寄存器 2 (SWTF_CIDR2)

SWTF CoreSight component identity register 2

偏移地址：0xFF8

复位值：0x0000 0005

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[19:12]							
								r							

位 31:8 保留，必须保持复位值。

位 7:0 **PREAMBLE[19:12]**：组件 ID 字段 (Component ID field)，位 [23:16]
0x05：公共 ID 值

SWTF CoreSight 组件标识寄存器 3 (SWTF_CIDR3)

SWTF CoreSight component identity register 3

偏移地址：0xFFC

复位值：0x0000 00B1

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[27:20]							
								r							

位 31:8 保留，必须保持复位值。

位 7:0 **PREAMBLE[27:20]**：组件 ID 字段 (Component ID field)，位 [31:24]
0xB1：公共 ID 值

SWTF 寄存器映射和复位值

表 575. SWTF 寄存器映射和复位值

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x000	SWTF_CTRL	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MIN_HOLD_TIME[3:0]				Res.	Res.	Res.	Res.	Res.	Res.	ENS1	ENS0
	Reset value																					0	0	1	1							0	0
0x004	SWTF_PRIORITY	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PRIPORT1[2:0]			PRIPORT0[2:0]		
	Reset value																											0	0	1	0	0	0
0xFA0	SWTF_CLAIMSET	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CLAIMSET[3:0]			
	Reset value																													1	1	1	1
0xFA4	SWTF_CLAIMCLR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CLAIMCLR[3:0]			
	Reset value																													0	0	0	0
0xFB0	SWTF_LAR	ACCESS_W[31:0]																															
	Reset value																																
0xFB4	SWTF_LSR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LOCKTYPE	LOCKGRANT	LOCKEXIST
	Reset value																														0	1	1
0xFB8	SWTF_AUTHSTAT	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SNID[1:0]		SID[1:0]		NSNID[1:0]		NSID[1:0]	
	Reset value																									0	0	0	0	0	0	0	0

表 575. SWTF 寄存器映射和复位值（续）

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0xFC8	SWTF_DEVID	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SCHEME [3:0]				PORTCNT [3:0]			
	Reset value																									0	0	1	0	0	0	1	0
0xFCC	SWTF_TYPEID	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DEVTYPEID[7:0]							
	Reset value																									0	0	0	1	0	0	1	0
0xFD0	SWTF_PIDR4	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	4KCOUNT [3:0]				JEP106CON [3:0]			
	Reset value																									0	0	0	0	0	1	0	0
0xFD4	SWTF_PIDR5	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	Reset value																																
0xFD8	SWTF_PIDR6	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	Reset value																																
0xFDC	SWTF_PIDR7	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	Reset value																																
0xFE0	SWTF_PIDR0	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PARTNUM [7:0]							
	Reset value																									0	0	0	0	1	0	0	0
0xFE4	SWTF_PIDR1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	JEP106ID [3:0]				PARTNUM [11:8]			
	Reset value																									1	0	1	1	1	0	0	1
0xFE8	SWTF_PIDR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REVISION]				JEDEC	JEP106ID [6:4]		
	Reset value																									0	0	0	1	1	0	1	1
0xFEC	SWTF_PIDR3	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REVAND				CMOD			
	Reset value																									0	0	0	0	0	0	0	0
0xFF0	SWTF_CIDR0	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE							
	Reset value																									0	0	0	0	1	1	0	1
0xFF4	SWTF_CIDR1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CLASS				PREAMBLE			
	Reset value																									1	0	0	1	0	0	0	0
0xFF8	SWTF_CIDR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE							
	Reset value																									0	0	0	0	0	1	0	1
0xFFC	SWTF_CIDR3	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE							
	Reset value																									1	0	1	1	0	0	0	1

60.5.8 微控制器调试单元 (DBGMCU)

DBGMCU 组件包含多个寄存器，用来在调试模式下控制电源和时钟行为。具体地说，它允许调试器或调试软件：

- 在低功耗模式（睡眠、停止或待机）下，保持处理器内核的时钟和电源
- 在低功耗模式下，保持系统调试和跟踪组件的时钟和电源
- 当处理器内核在调试模式下停止时，停止某些外设（CAN、SMBUS 超时、看门狗、定时器、RTC）的时钟。为安全起见，当计数器停止时（DBGMCU_APB2FZ1 中 TIM1/8/15/16/17 = 1），禁止了互补输出计时器的输出（就像 MOE 位复位一样）。

DBGMCU 寄存器不通过系统复位而复位，只能通过上电复位而复位。调试器可以通过基址为 0xE00E1000 的 APB-D 总线对其进行访问。另外，基址为 0x5C001000 的两个处理器内核也可以对其进行访问。

注意：DBGMCU 并非标准的 CoreSight 组件。因此，它不在系统 ROM 表中。

DBGMCU寄存器

DBGMCU 标识代码寄存器 (DBGMCU_IDC)

DBGMCU identity code register

偏移地址：0x000

复位值：0x100X 6450

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
REV_ID[15:0]															
r															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	DEV_ID[11:0]											
				r											

位 31:16 **REV_ID[15:0]**：版本 (Revision)
0x1001 = 版本 Z
0x1003 = 版本 Y

位 15:12 保留，必须保持复位值。

位 11:0 **DEV_ID[11:0]**：器件 ID (Device ID)
0x450：STM32H7

DBGMCU 配置寄存器 (DBGMCU_CR)

DBGMCU configuration register

偏移地址：0x004

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	TRGOEN	Res.	Res.	Res.	Res.	Res.	D3DBGCKEN	D1DBGCKEN	TRACECLKEN	Res.	Res.	Res.	Res.
			rw						rw	rw	rw				

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	DBGSTBY_D3	DBGSTOP_D3	Res.	Res.	Res.	Res.	DBGSTBY_D1	DBGSTOP_D1	DBGSLEEP_D1
							rw	rw					rw	rw	rw

位 31:29 保留，必须保持复位值。

位 28 **TRGOEN**：外部触发输出使能 (External trigger output enable)
该位控制双向触发引脚 TRGIO 的方向。

0：输入——TRGIO 连接到 TRGIN
1：输出——TRGIO 连接到 TRGOUT

位 27:23 保留，必须保持复位值。

位 22 **D3DBGCKEN**：D3 调试时钟使能 (D3 debug clock enable)
该位允许关闭 D3 域中不需要的调试组件（DAPCLK 域中的组件除外）。

0：禁止——禁止 D3 域调试组件并对其时钟进行门控
1：使能——只要相应的域时钟 (CK_HCLK_D3) 处于激活状态，就对 D3 域调试组件提供时钟信号

位 21 **D1DBGCKEN**：D1 调试时钟使能 (D1 debug clock enable)
该位允许关闭 D1 域中不需要的调试组件（处理器内核中的组件除外）。

0：禁止——禁止 D1 域调试组件并对其时钟进行门控
1：使能——只要相应的域时钟 (CK_HCLK_D1) 处于激活状态，就对 D1 域调试组件提供时钟信号

位 20 **TRACECLKEN**：跟踪端口时钟使能 (Trace port clock enable)
该位使能跟踪端口时钟 TRACECLK。

0：禁止——TRACECLK 被禁止
1：使能——TRACECLK 处于激活状态

位 19:9 保留，必须保持复位值。

位 8 **DBGSTBY_D3**：允许在 D3 待机模式下调试 (Allow debug in D3 Standby mode)
0：正常操作——在待机模式下，禁止所有时钟，且该域自动掉电(1)。
1：禁止时钟自动停止/掉电——在待机模式下，所有处于激活状态的时钟和振荡器继续运行，同时域电源保持不变，因此支持全面调试功能。在退出待机模式时，执行系统复位。

位 7 **DBGSTOP_D3**：允许在 D3 停止模式下调试 (Allow debug in D3 Stop mode)

0：正常操作——在停止模式下自动禁止域时钟(2)

1：禁止时钟自动停止/掉电——在停止模式下，所有处于激活状态的时钟和振荡器继续运行。从停止模式退出时，时钟设置被设置为停止模式退出状态。

位 6:3 保留，必须保持复位值。

位 2 **DBGSTBY_D1**：允许待机模式下的 D1 域调试 (Allow D1 domain debug in Standby mode)

0：正常操作——在待机模式下，禁止所有时钟，且该域自动掉电。

1：禁止时钟自动停止/掉电——在待机模式下，所有处于激活状态的时钟和振荡器继续运行，同时域电源保持不变，因此支持全面调试功能。在退出待机模式时，执行域复位。

位 1 **DBGSTOP_D1**：允许停止模式下的 D1 域调试 (Allow D1 domain debug in Stop mode)

0：正常操作——在停止模式下，自动禁止所有时钟

1：禁止时钟自动停止——在停止模式下，所有处于激活状态的时钟和振荡器继续运行，因此支持全面调试功能。从停止模式退出时，时钟设置被设置为停止模式退出状态。

位 0 **DBGSLEEP_D1**：允许睡眠模式下的 D1 域调试 (Allow D1 domain debug in Sleep mode)

0：正常操作——在睡眠模式下，自动停止处理器时钟

1：禁止时钟自动停止——处理器时钟继续运行，因此支持全面调试功能

1. 如果调试端口控制/状态寄存器中的 CDBGPWRUPREQ 置为有效，则在待机模式下，D3 域将保持上电状态，并且 DAPCLK 处于激活状态，即使 DBGSTBY_D3 被复位，亦是如此。但其他 D3 域时钟将被关闭。
2. 如果调试端口控制/状态寄存器中的 CDBGPWRUPREQ 置为有效，则在停止模式下，D3 域将保持上电状态，并且 DAPCLK 处于激活状态，即使 DBGSTOP_D3 被复位，亦是如此。但其他 D3 域时钟将被关闭。

DBGMCU APB3 外设冻结寄存器 CPU (DBGMCU_APB3FZ1)

DBGMCU APB3 peripheral freeze register CPU

偏移地址：0x034

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WWDG1	Res.	Res.	Res.	Res.	Res.	Res.
									rw						

位 31:7 保留，必须保持复位值。

位 6 **WWDG1**：在调试模式下 WWDG1 停止 (WWDG1 stop in debug)

0：正常操作——当内核处于调试模式时，WWDG1 继续运行

1：在调试模式下停止——当内核处于调试模式时，WWDG1 被冻结

位 5:0 保留，必须保持复位值。

DBGMCU APB1L 外设冻结寄存器 CPU (DBGMCU_APB1LFZ1)

DBGMCU APB1L peripheral freeze register CPU

偏移地址：0x03C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	I2C3	I2C2	I2C1	Res.	Res.	Res.	Res.	Res.
								rw	rw	rw					
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	LPTIM1	TIM14	TIM13	TIM12	TIM7	TIM6	TIM5	TIM4	TIM3	TIM2
				rw		rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

位 31:24 保留，必须保持复位值。

位 23 **I2C3**：在调试模式下 I2C3 SMBUS 超时停止 (I2C3 SMBUS timeout stop in debug)

0：正常操作——当内核处于调试模式时，I2C3 SMBUS 超时继续运行
1：在调试模式下停止——当 Cortex-M7 处于调试模式时，I2C3 SMBUS 超时被冻结

位 22 **I2C2**：在调试模式下 I2C2 SMBUS 超时停止 (I2C2 SMBUS timeout stop in debug)

0：正常操作——当内核处于调试模式时，I2C2 SMBUS 超时继续运行
1：在调试模式下停止——当 Cortex-M7 处于调试模式时，I2C2 SMBUS 超时被冻结

位 21 **I2C1**：在调试模式下 I2C1 SMBUS 超时停止 (I2C1 SMBUS timeout stop in debug)

0：正常操作——当内核处于调试模式时，I2C1 SMBUS 超时继续运行
1：在调试模式下停止——当内核处于调试模式时，I2C1 SMBUS 超时被冻结

位 20:11 保留，必须保持复位值。

位 10 保留，必须保持复位值。

位 9 **LPTIM1**：在调试模式下 LPTIM1 停止 (LPTIM1 stop in debug)

0：正常操作——当内核处于调试模式时，LPTIM1 继续运行
1：在调试模式下停止——当 Cortex-M7 处于调试模式时，LPTIM1 被冻结

位 8 **TIM14**：在调试模式下 TIM14 停止 (TIM14 stop in debug)

0：正常操作——当内核处于调试模式时，TIM14 继续运行
1：在调试模式下停止——当 Cortex-M7 处于调试模式时，TIM14 被冻结

位 7 **TIM13**：在调试模式下 TIM13 停止 (TIM13 stop in debug)

0：正常操作——当内核处于调试模式时，TIM13 继续运行
1：在调试模式下停止——当 Cortex-M7 处于调试模式时，TIM13 被冻结

位 6 **TIM12**：在调试模式下 TIM12 停止 (TIM12 stop in debug)

0：正常操作——当内核处于调试模式时，TIM12 继续运行
1：在调试模式下停止——当 Cortex-M7 处于调试模式时，TIM12 被冻结

位 5 **TIM7**：在调试模式下 TIM7 停止 (TIM7 stop in debug)

0：正常操作——当内核处于调试模式时，TIM7 继续运行
1：在调试模式下停止——当 Cortex-M7 处于调试模式时，TIM7 被冻结

位 4 **TIM6**：在调试模式下 TIM6 停止 (TIM6 stop in debug)

0：正常操作——当内核处于调试模式时，TIM6 继续运行
1：在调试模式下停止——当 Cortex-M7 处于调试模式时，TIM6 被冻结

位 3 **TIM5**：在调试模式下 TIM5 停止 (TIM5 stop in debug)

0：正常操作——当内核处于调试模式时，TIM5 继续运行
1：在调试模式下停止——当 Cortex-M7 处于调试模式时，TIM5 被冻结

位 2 **TIM4**：在调试模式下 TIM4 停止 (TIM4 stop in debug)

0：正常操作——当内核处于调试模式时，TIM4 继续运行
1：在调试模式下停止——当 Cortex-M7 处于调试模式时，TIM4 被冻结

位 1 **TIM3**：在调试模式下 TIM3 停止 (TIM3 stop in debug)

0：正常操作——当内核处于调试模式时，TIM3 继续运行
1：在调试模式下停止——当 Cortex-M7 处于调试模式时，TIM3 被冻结

位 0 **TIM2**：在调试模式下 TIM2 停止 (TIM2 stop in debug)

0：正常操作——当内核处于调试模式时，TIM2 继续运行
1：在调试模式下停止——当 Cortex-M7 处于调试模式时，TIM2 被冻结e

DBGMCU APB1H 外设冻结寄存器 CPU (DBGMCU_APB1HFZ1)

DBGMCU APB1H peripheral freeze register CPU

偏移地址：0x044

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	FDCAN	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
							rw								

位 31:9 保留，必须保持复位值。

位 8 **FDCAN**：在调试模式下 FDCAN 停止 (FDCAN stop in debug)

0：正常操作——当内核处于调试模式时，FDCAN 继续运行
1：在调试模式下停止——当内核处于调试模式时，FDCAN 被冻结

位 7:0 保留，必须保持复位值。

DBGMCU APB2 外设冻结寄存器 CPU (DBGMCU_APB2FZ1)

DBGMCU APB2 peripheral freeze register CPU

偏移地址：0x04C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	HRTIM	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TIM17	TIM16	TIM15
		rw											rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TIM8	TIM1
														rw	rw

位 31:30 保留，必须保持复位值。

位 29 **HRTIM**：在调试模式下 HRTIM 停止 (HRTIM stop in debug)

0：正常操作——当内核处于调试模式时，HRTIM 继续运行
1：在调试模式下停止——当 Cortex-M7 处于调试模式时，HRTIM 被冻结

位 28:19 保留，必须保持复位值。

位 18 **TIM17**：在调试模式下 TIM17 停止 (TIM17 stop in debug)

0：正常操作——当内核处于调试模式时，TIM17 继续运行
1：在调试模式下停止——当 Cortex-M7 处于调试模式时，TIM17 被冻结并禁止 TIM17 输出

位 17 **TIM16**：在调试模式下 TIM16 停止 (TIM16 stop in debug)

0：正常操作——当内核处于调试模式时，TIM16 继续运行
1：在调试模式下停止——当 Cortex-M7 处于调试模式时，TIM16 被冻结并禁止 TIM16 输出

位 16 **TIM15**：在调试模式下 TIM15 停止 (TIM15 stop in debug)

0：正常操作——当内核处于调试模式时，TIM15 继续运行
1：在调试模式下停止——当 Cortex-M7 处于调试模式时，TIM15 被冻结并禁止 TIM15 输出

位 15:2 保留，必须保持复位值。

位 1 **TIM8**：在调试模式下 TIM8 停止 (TIM8 stop in debug)

0：正常操作——当内核处于调试模式时，TIM8 继续运行
1：在调试模式下停止——当 Cortex-M7 处于调试模式时，TIM8 被冻结并禁止 TIM8 输出

位 0 **TIM1**：在调试模式下 TIM1 停止 (TIM1 stop in debug)

0：正常操作——当内核处于调试模式时，TIM1 继续运行
1：在调试模式下停止——当 Cortex-M7 处于调试模式时，TIM1 被冻结并禁止 TIM1 输出

DBGMCU APB4 外设冻结寄存器 CPU (DBGMCU_APB4FZ1)

DBGMCU APB4 peripheral freeze register CPU

偏移地址：0x04C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WDGLSD1	Res.	RTC
													rw		rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	LPTIM5	LPTIM4	LPTIM3	LPTIM2	Res.	I2C4	Res.	Res.	Res.	Res.	Res.	Res.	Res.
			rw	rw	rw	rw		rw							

位 31:19 保留，必须保持复位值。

位 18 **WDGLSD1**：在调试模式下停止 D1 LS 看门狗 (LS watchdog for D1 stop in debug)

0：正常操作——当内核处于调试模式时，看门狗继续计数
1：在调试模式下停止——当 Cortex-M7 处于调试模式时，看门狗被冻结

位 17 保留，必须保持复位值。

位 16 **RTC**：在调试模式下停止 RTC (RTC stop in debug)

0：正常操作——当内核处于调试模式时，RTC 继续运行
1：在调试模式下停止——当 Cortex-M7 处于调试模式时，RTC 被冻结

位 15:13 保留，必须保持复位值。

位 12 **LPTIM5**：在调试模式下停止 LPTIM5 (LPTIM5 stop in debug)

0：正常操作——当内核处于调试模式时，LPTIM5 继续运行
1：在调试模式下停止——当 Cortex-M7 处于调试模式时，LPTIM5 被冻结

位 11 **LPTIM4**：在调试模式下停止 LPTIM4 (LPTIM4 stop in debug)

0：正常操作——当内核处于调试模式时，LPTIM4 继续运行
1：在调试模式下停止——当 Cortex-M7 处于调试模式时，LPTIM4 被冻结

位 10 **LPTIM3**：在调试模式下停止 LPTIM3 (LPTIM3 stop in debug)

0：正常操作——当内核处于调试模式时，LPTIM3 继续运行
1：在调试模式下停止——当 Cortex-M7 处于调试模式时，LPTIM3 被冻结

位 9 **LPTIM2**：在调试模式下停止 LPTIM2 (LPTIM2 stop in debug)

0：正常操作——当内核处于调试模式时，LPTIM2 继续运行
1：在调试模式下停止——当 Cortex-M7 处于调试模式时，LPTIM2 被冻结

位 8 保留，必须保持复位值。

位 7 **I2C4**：在调试模式下停止 I2C4 SMBUS 超时 (I2C4 SMBUS timeout stop in debug)

0：正常操作——当内核处于调试模式时，I2C4 SMBUS 超时继续运行
1：在调试模式下停止——当内核处于调试模式时，I2C4 SMBUS 超时被冻结

位 6:0 保留，必须保持复位值。

DBGMCU 寄存器映射和复位值

表 576. DBGMCU 寄存器映射和复位值

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x000	DBGMCU_IDC	REV_ID[15:0]																Res.	Res.	Res.	Res.	DEV_ID[11:0]											
	Reset value	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	1	0	0	0	1	0	1	0	0	0	0
0x004	DBGMCU_CR	Res.	Res.	Res.	TRGOEN	Res.	Res.	Res.	Res.	Res.	D3DBGCKEN	D1DBGCKEN	TRACECLKEN	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DBGSTBD3	DBGSTPD3	Res.	DBGSTBD2	DBGSTPD2	DBGSLPD2	DBGSTBD1	DBGSTPD1	DBGSLPD1
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x01C to 0x030	Reserved																																
0x034	DBGMCU_APB3FZ1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WWDG1	Res.	Res.	Res.	Res.	Res.	Res.
	Reset value																										0						
0x038	Reserved																																
0x03C	DBGMCU_APB1LFZ1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	I2C3	I2C2	I2C1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LPTIM1	TIM14	TIM13	TIM12	TIM7	TIM6	TIM5	TIM4	TIM3	TIM2
	Reset value									0	0	0												0	0	0	0	0	0	0	0	0	0
0x040	Reserved																																
0x044	DBGMCU_APB1HFZ1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FDCAN	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	Reset value																								0								
0x048	Reserved																																
0x04C	DBGMCU_APB2FZ1	Res.	Res.	HRTIM	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TIM17	TIM16	TIM15	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TIM8	TIM1
	Reset value			0											0	0	0														0	0
0x050	Reserved																																
0x054	DBGMCU_APB4FZ1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	WDGLSD1	Res.	Res.	Res.	Res.	Res.	LPTIM5	LPTIM4	LPTIM3	LPTIM2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	Reset value														0						0	0	0	0									
0x058	Reserved																																

60.6 Cortex-M7 调试功能描述

Cortex-M7 子系统特有以下 CoreSight™ 组件：

- ROM 表
- 系统控制空间 (SCS)
- 断点单元 (FPB)
- 数据观察点和跟踪单元 (DWT)
- 指令跟踪宏单元 (ITM)
- 嵌入式跟踪宏单元(ETM)
- 交叉触发接口 (CTI)

调试器可通过 Cortex-M7 AHB-AP 及其相关的 AHBD 总线访问这些组件。

60.6.1 Cortex-M7 ROM 表

ROM 表是一款 CoreSight™ 组件，包含可通过 AHBD 访问的所有 CoreSight 调试组件的基址。这些表允许调试器自动发现 CoreSight 系统的拓扑。

Cortex-M7 子系统有两个 ROM 表：

- Cortex-M7 CPU ROM 表

  该表由 Cortex-M7 AHB-AP 中的 BASE 寄存器指向。包含 ETM 和 CTI 以及 Cortex-M7 CPU ROM 表的基址指针。
- Cortex-M7 PPB（专用外设总线）ROM 表

  该表包含指向 Cortex-M7 系统控制空间寄存器的指针（以允许调试器识别 CPU 内核）以及指向 Cortex-M7 子系统其他 CoreSight 组件（FPB、DWT 和 ITM）的指针。

CPU ROM 表占用一个 4 KB、32 位宽的 AHBD 地址空间块，地址范围从 0xE00FE000 到 0xE00FEFFC。

表 577. Cortex-M7 CPU ROM 表

ROM 表中的地址	元件名称	组件基址	组件偏移地址	大小	起始
0xE00FE000	Cortex-M7 PPB ROM 表	0xE00FF000	0x00001000	4 KB	0x00001003
0xE00FE004	Cortex-M7 ETM	0xE0041000	0xFFF43000	4 KB	0xFFF43003
0xE00FE008	Cortex-M7 CTI	0xE0043000	0xFFF44000	4 KB	0xFFF44003
0xE00FE00C	保留	-	-	-	0x1FF02002
0xE00FE010	表顶部	-	-	-	0x00000000
0xE00FE010 到 0xE00FEFC8	保留	-	-	-	0x00000000
0xE00FEFCC 到 0xE00FEFFC	ROM 表寄存器	-	-	-	请参见*表 579*

Cortex-M7 PPB ROM 表占用一个 4 KB、32 位宽的 AHBD 地址空间块，地址范围从 0xE00FF000 到 0xE00FFFFC。

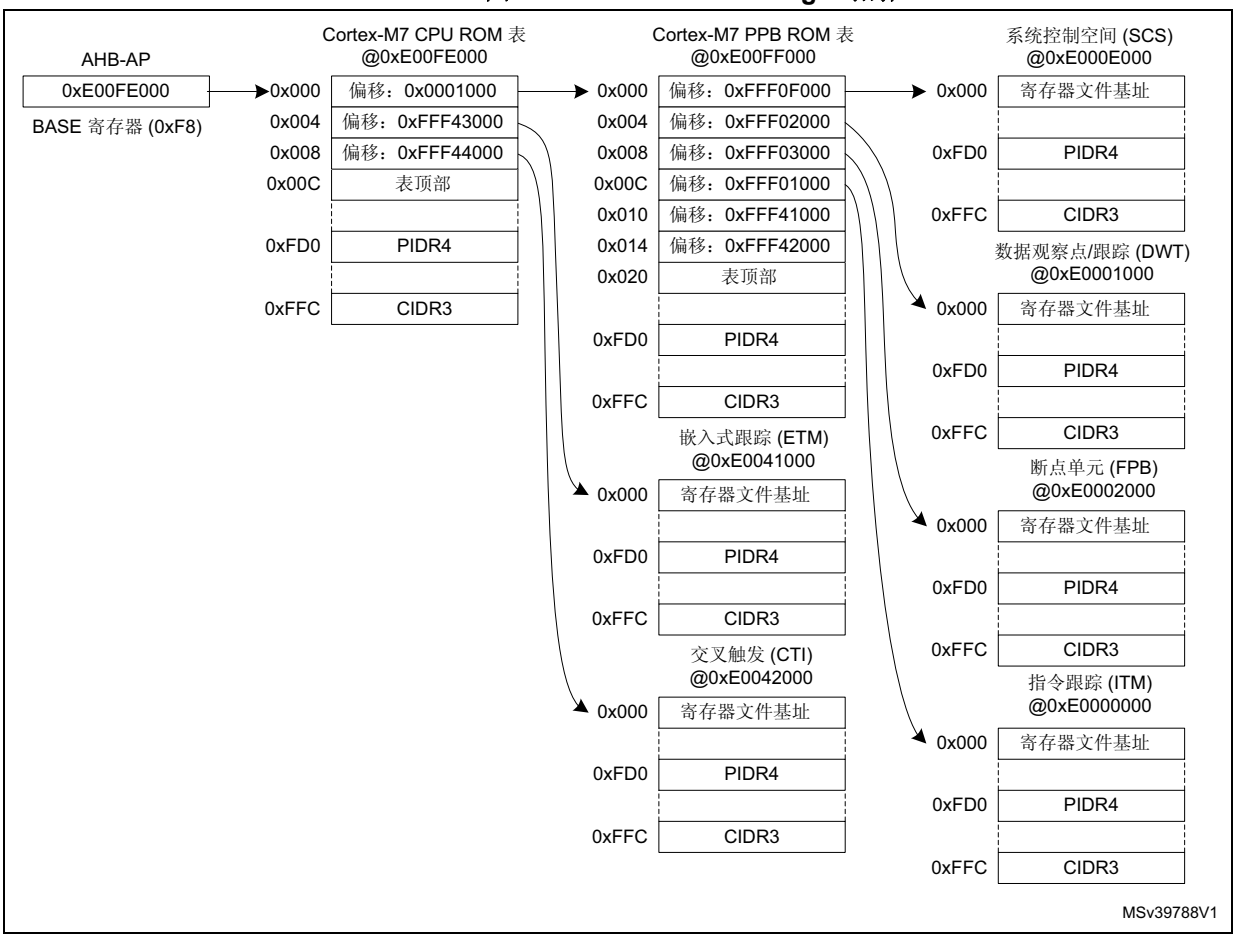
表 578. Cortex-M7 PPB ROM 表

ROM 表中的地址	元件名称	组件基址	组件偏移地址	大小	起始
0xE00FF000	SCS	0xE000E000	0xFFF0F000	4 KB	0xFFF0F003
0xE00FF004	DWT	0xE0001000	0xFFF02000	4 KB	0xFFF02003
0xE00FF008	FPB	0xE0002000	0xFFF03000	4 KB	0xFFF03003
0xE00FF00C	ITM	0xE0000000	0xFFF01000	4 KB	0xFFF01003
0xE00FF010	TPIU(1)	0xE0040000	0xFFF41000	4 KB	0xFFF41002
0xE00FF014	ETM(1)	0xE0041000	0xFFF42000	4 KB	0xFFF42002
0xE00FF018	表顶部	-	-	-	0x00000000
0xE00FF01C 到 0xE00FFFC8	保留	-	-	-	0x00000000
0xE00FFFCC 到 0xE00FFFFC	ROM 表寄存器	-	-	-	请参见*表 580*

1. 默认情况下，该表包含 TPIU 和 ETM，但位 0 被复位以指示它们不存在。

Cortex-M7 子系统的 CoreSight™ 组件拓扑如*图 820* 所示。

图 820. Cortex-M7 CoreSight 拓扑



Cortex-M7 CPU ROM 寄存器

CPU ROM 存储器类型寄存器 (M7_CPUROM_MEMTYPE)

CPU ROM memory type register

偏移地址：0xFCC

复位值：0x0000 0001

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SYSMEM
															r

位 31:1 保留，必须保持复位值。

位 0 **SYSMEM**：系统存储器有无指示 (System memory presence)
1：该总线上有系统存储器

CPU ROM CoreSight 外设标识寄存器 4 (M7_CPUROM_PIDR4)

CPU ROM CoreSight peripheral identity register 4

偏移地址：0xFD0

复位值：0x0000 0004

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	4KCOUNT[3:0]				JEP106CON[3:0]			
								r				r			

位 31:8 保留，必须保持复位值。

位 7:4 **4KCOUNT[3:0]**：寄存器文件大小 (Register file size)
0x0：寄存器文件占用单个 4 KB 区域

位 3:0 **JEP106CON[3:0]**：JEP106 连续代码 (JEP106 continuation code)
0x4：ARM® JEDEC 连续代码

CPU ROM CoreSight 外设标识寄存器 0 (M7_CPUROM_PIDR0)

CPU ROM CoreSight peripheral identity register 0

偏移地址：0xFE0

复位值：0x0000 0c8

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PARTNUM[7:0]							
								r							

位 31:8 保留，必须保持复位值。

位 7:0 **PARTNUM[7:0]**：产品编号字段 (Part number field)，位 [7:0]

0xC8：Cortex-M7 处理器 ROM 表

CPU ROM CoreSight 外设标识寄存器 1 (M7_CPUROM_PIDR1)

CPU ROM CoreSight peripheral identity register 1

偏移地址：0xFE4

复位值：0x0000 00B4

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	JEP106ID[3:0]				PARTNUM[11:8]			
								r				r			

位 31:8 保留，必须保持复位值。

位 7:4 **JEP106ID[3:0]**：JEP106 标识代码字段 (JEP106 identity code field)，位 [3:0]

0xB：ARM® JEDEC 代码

位 3:0 **PARTNUM[11:8]**：产品编号字段 (Part number field)，位 [11:8]

0x4：Cortex-M7 处理器 ROM 表

CPU ROM CoreSight 外设标识寄存器 2 (M7_CPUROM_PIDR2)

CPU ROM CoreSight peripheral identity register 2

偏移地址：0xFE8

复位值：0x0000 000B

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REVISION[3:0]				JEDEC	JEP106ID[6:4]		
								r				r	r		

位 31:8 保留，必须保持复位值。

位 7:4 **REVISION[3:0]**：组件版本号 (Component revision number)
0x0：rev r0p0

位 3 **JEDEC**：JEDEC 分配的值 (JEDEC assigned value)
1：JEDEC 指定的设计人员 ID

位 2:0 **JEP106ID[6:4]**：JEP106 标识代码字段 (JEP106 identity code field)，位 [6:4]
0x3：ARM® JEDEC 代码

CPU ROM CoreSight 外设标识寄存器 3 (M7_CPUROM_PIDR3)

CPU ROM CoreSight peripheral identity register 3

偏移地址：0xFEC

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REVAND[3:0]				CMOD[3:0]			
								r				r			

位 31:8 保留，必须保持复位值。

位 7:4 **REVAND[3:0]**：金属修复版本 (Metal fix version)
0x0：无金属修复

位 3:0 **CMOD[3:0]**：客户修改 (Customer modified)
0x0：无客户修改

CPU ROM CoreSight 组件标识寄存器 0 (M7_CPUROM_CIDR0)

CPU ROM CoreSight component identity register 0

偏移地址：0xFF0

复位值：0x0000 000D

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[7:0]							
								r							

位 31:8 保留，必须保持复位值。

位 7:0 **PREAMBLE[7:0]**：组件 ID 字段 (Component ID field)，位 [7:0]
0x0D：公共 ID 值

CPU ROM CoreSight 组件标识寄存器 1 (M7_CPUROM_CIDR1)

CPU ROM CoreSight component identity register 1

偏移地址：0xFF4

复位值：0x0000 0010

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CLASS[3:0]				PREAMBLE[11:8]			
								r				r			

位 31:8 保留，必须保持复位值。

位 7:4 **CLASS[3:0]**：组件 ID 字段 (Component ID field)，位 [15:12]——组件类
0x1：ROM 表组件

位 3:0 **PREAMBLE[11:8]**：组件 ID 字段 (Component ID field)，位 [11:8]
0x0：公共 ID 值

CPU ROM CoreSight 组件标识寄存器 2 (M7_CPUROM_CIDR2)

CPU ROM CoreSight component identity register 2

偏移地址：0xFF8

复位值：0x0000 0005

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[19:12]							
								r							

位 31:8 保留，必须保持复位值。

位 7:0 **PREAMBLE[19:12]**：组件 ID 字段 (Component ID field)，位 [23:16]
0x05：公共 ID 值

CPU ROM CoreSight 组件标识寄存器 3 (M7_CPUROM_CIDR3)

CPU ROM CoreSight component identity register 3

偏移地址：0xFFC

复位值：0x0000 00B1

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[27:20]							
								r							

位 31:8 保留，必须保持复位值。

位 7:0 **PREAMBLE[27:20]**：组件 ID 字段 (Component ID field)，位 [31:24]
0xB1：公共 ID 值

Cortex-M7 CPU ROM 表寄存器映射和复位值

表 579. Cortex-M7 CPU ROM 表寄存器映射和复位值

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0xFCC	M7_CPUROM_MEMTYPE	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SYSMEM
	Reset value																																1
0xFD0	M7_CPUROM_PIDR4	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	4KCOUNT [3:0]				JEP106CON [3:0]			
	Reset value																									0	0	0	0	0	0	0	0
0xFD4	M7_CPUROM_PIDR5	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	Reset value																																
0xFD8	M7_CPUROM_PIDR6	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	Reset value																																
0xFDC	M7_CPUROM_PIDR7	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	Reset value																																
0xFE0	M7_CPUROM_PIDR0	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PARTNUM [7:0]							
	Reset value																									1	1	0	0	1	0	0	0
0xFE4	M7_CPUROM_PIDR1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	JEP106ID [3:0]				PARTNUM [11:8]			
	Reset value																									1	0	1	1	0	1	0	0
0xFE8	M7_CPUROM_PIDR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REVISION [3:0]				JEDEC	JEP106ID [6:4]		
	Reset value																									0	0	0	0	1	0	1	1

表 579. Cortex-M7 CPU ROM 表寄存器映射和复位值（续）

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0xFEC	M7_CPUROM_PIDR3	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REVAND[3:0]				CMOD[3:0]			
	Reset value																									0	0	0	0	0	0	0	0
0xFF0	M7_CPUROM_CIDR0	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[7:0]							
	Reset value																									0	0	0	0	1	1	0	1
0xFF4	M7_CPUROM_CIDR1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CLASS[3:0]				PREAMBLE[11:8]			
	Reset value																									0	0	0	1	0	0	0	0
0xFF8	M7_CPUROM_CIDR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[19:12]							
	Reset value																									0	0	0	0	0	1	0	1
0xFFC	M7_CPUROM_CIDR3	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[27:20]							
	Reset value																									1	0	1	1	0	0	0	1

Cortex-M7 PPB ROM 寄存器

PPB ROM 存储器类型寄存器 (M7_PPBROM_MEMTYPE)

PPB ROM memory type register

偏移地址：0xFCC

复位值：0x0000 0001

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SYSMEM
															r

位 31:1 保留，必须保持复位值。

位 0 **SYSMEM**：系统存储器有无指示 (System memory presence)

1：该总线上有系统存储器

PPB ROM CoreSight 外设标识寄存器 4 (M7_PPBROM_PIDR4)

PPB ROM CoreSight peripheral identity register 4

偏移地址：0xFD0

复位值：0x0000 0004

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	4KCOUNT[3:0]				JEP106CON[3:0]			
								r				r			

位 31:8 保留，必须保持复位值。

位 7:4 **4KCOUNT[3:0]**：寄存器文件大小 (Register file size)
0x0：寄存器文件占用单个 4 KB 区域

位 3:0 **JEP106CON[3:0]**：JEP106 连续代码 (JEP106 continuation code)
0x4：ARM® JEDEC 连续代码

PPB ROM CoreSight 外设标识寄存器 0 (M7_PPBROM_PIDR0)

PPB ROM CoreSight peripheral identity register 0

偏移地址：0xFE0

复位值：0x0000 0c8

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PARTNUM[7:0]							
								r							

位 31:8 保留，必须保持复位值。

位 7:0 **PARTNUM[7:0]**：产品编号字段 (Part number field)，位 [7:0]
0xC7：Cortex-M7 PPB ROM 表

PPB ROM CoreSight 外设标识寄存器 1 (M7_PPBROM_PIDR1)

PPB ROM CoreSight peripheral identity register 1

偏移地址：0xFE4

复位值：0x0000 00B4

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	JEP106ID[3:0]				PARTNUM[11:8]			
								r				r			

位 31:8 保留，必须保持复位值。

位 7:4 **JEP106ID[3:0]**：JEP106 标识代码字段 (JEP106 identity code field)，位 [3:0]
0xB：ARM® JEDEC 代码

位 3:0 **PARTNUM[11:8]**：产品编号字段 (Part number field)，位 [11:8]
0x4：Cortex-M7 PPB ROM 表

PPB ROM CoreSight 外设标识寄存器 2 (M7_PPBROM_PIDR2)

PPB ROM CoreSight peripheral identity register 2

偏移地址：0xFE8

复位值：0x0000 000B

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REVISION[3:0]				JEDEC	JEP106ID[6:4]		
								r				r	r		

位 31:8 保留，必须保持复位值。

位 7:4 **REVISION[3:0]**：组件版本号 (Component revision number)
0x0：rev r0p0

位 3 **JEDEC**：JEDEC 分配的值 (JEDEC assigned value)
1：JEDEC 指定的设计人员 ID

位 2:0 **JEP106ID[6:4]**：JEP106 标识代码字段 (JEP106 identity code field)，位 [6:4]
0x3：ARM® JEDEC 代码

PPB ROM CoreSight 外设标识寄存器 3 (M7_PPBROM_PIDR3)

PPB ROM CoreSight peripheral identity register 3

偏移地址：0xFEC

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REVAND[3:0]				CMOD[3:0]			
								r				r			

位 31:8 保留，必须保持复位值。

位 7:4 **REVAND[3:0]**：金属修复版本 (Metal fix version)
0x0：无金属修复

位 3:0 **CMOD[3:0]**：客户修改 (Customer modified)
0x0：无客户修改

PPB ROM CoreSight 组件标识寄存器 0 (M7_PPBROM_CIDR0)

PPB ROM CoreSight component identity register 0

偏移地址：0xFF0

复位值：0x0000 000D

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[7:0]							
								r							

位 31:8 保留，必须保持复位值。

位 7:0 **PREAMBLE[7:0]**：组件 ID 字段 (Component ID field)，位 [7:0]
0x0D：公共 ID 值

PPB ROM CoreSight 组件标识寄存器 1 (M7_PPBROM_CIDR1)

PPB ROM CoreSight component identity register 1

偏移地址：0xFF4

复位值：0x0000 0010

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CLASS[3:0]				PREAMBLE[11:8]			
								r				r			

位 31:8 保留，必须保持复位值。

位 7:4 **CLASS[3:0]**：组件 ID 字段 (Component ID field)，位 [15:12]——组件类
0x1：ROM 表组件

位 3:0 **PREAMBLE[11:8]**：组件 ID 字段 (Component ID field)，位 [11:8]
0x0：公共 ID 值

PPB ROM CoreSight 组件标识寄存器 2 (M7_PPBROM_CIDR2)

PPB ROM CoreSight component identity register 2

偏移地址：0xFF8

复位值：0x0000 0005

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[19:12]							
								r							

位 31:8 保留，必须保持复位值。

位 7:0 **PREAMBLE[19:12]**：组件 ID 字段 (Component ID field)，位 [23:16]
0x05：公共 ID 值

PPB ROM CoreSight 组件标识寄存器 3 (M7_PPBROM_CIDR3)

PPB ROM CoreSight component identity register 3

偏移地址：0xFFC

复位值：0x0000 00B1

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[27:20]							
								r							

位 31:8 保留，必须保持复位值。

位 7:0 **PREAMBLE[27:20]**：组件 ID 字段 (Component ID field)，位 [31:24]
0xB1：公共 ID 值

Cortex-M7 PPB ROM 表寄存器映射和复位值

表 580. Cortex-M7 PPB ROM 表寄存器映射和复位值

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0xFCC	M7_PPBROM_MEMTYPE	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SYSMEM
	Reset value																																1
0xFD0	M7_PPBROM_PIDR4	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	4KCOUNT [3:0]				JEP106CON [3:0]			
	Reset value																									0	0	0	0	0	1	0	0
0xFD4	M7_PPBROM_PIDR5	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	Reset value																																
0xFD8	M7_PPBROM_PIDR6	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	Reset value																																
0xFDC	M7_PPBROM_PIDR7	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	Reset value																																
0xFE0	M7_PPBROM_PIDR0	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PARTNUM [7:0]							
	Reset value																									1	1	0	0	0	1	1	1
0xFE4	M7_PPBROM_PIDR1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	JEP106ID [3:0]				PARTNUM [11:8]			
	Reset value																									1	0	1	1	0	1	0	0
0xFE8	M7_PPBROM_PIDR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REVISION [3:0]				JEDEC	JEP106ID [6:4]		
	Reset value																									0	0	0	0	1	0	1	1

表 580. Cortex-M7 PPB ROM 表寄存器映射和复位值（续）

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0xFEC	M7_PPBROM_PIDR3	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REVAND[3:0]				CMOD[3:0]			
	Reset value																									0	0	0	0	0	0	0	0
0xFF0	M7_PPBROM_CIDR0	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[7:0]							
	Reset value																									0	0	0	0	1	1	0	1
0xFF4	M7_PPBROM_CIDR1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CLASS[3:0]				PREAMBLE[11:8]			
	Reset value																									0	0	0	1	0	0	0	0
0xFF8	M7_PPBROM_CIDR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[19:12]							
	Reset value																									0	0	0	0	0	1	0	1
0xFFC	M7_PPBROM_CIDR3	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[27:20]							
	Reset value																									1	0	1	1	0	0	0	1

60.6.2 Cortex-M7 数据观察点和跟踪单元 (DWT)

DWT 提供了四个比较器，可用作：

- 观察点
- ETM 触发
- PC 采样触发器
- 数据地址采样触发器
- 数据比较器（仅适用于比较器 1）
- 时钟周期计数比较器（仅适用于比较器 0）

还包含以下计数器：

- 时钟周期计数器
- 分支指令
- 加载存储单元操作
- 睡眠周期
- 每条指令周期数计数器
- 中断开销

DWT 比较器将以下其中一个值与其 DWT_COMP 寄存器中保存的值进行比较：

- 数据地址
- 指令地址
- 数据值
- 周期计数值（仅适用于比较器 0）

比较器可以使用掩码进行地址匹配，因此可匹配一系列地址。

一旦匹配成功时，比较器会生成以下其中一项：

- 一个或多个 DWT 数据跟踪数据包，包括一个或多个：
  - 引发数据访问的指令的地址
  - 偏移地址，数据访问地址的位 [15:0]
  - 匹配的数据值
- 观察点调试事件（在 PC 值上或者在被访问的数据地址上）
- 指示匹配超出 DWT 单元的 CMPMATCH[N] 事件

观察点调试事件会生成 DebugMonitor 异常，或者导致处理器停止执行并进入调试状态。

有关如何使用 DWT 的更多详细信息，请参见《ARM®v7-M 架构参考手册》*[5]*。

Cortex-M7 DWT 寄存器

DWT 控制寄存器 (M7_DWT_CTRL)

DWT control register

偏移地址：0x000

复位值：0x4000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NUMCOMP[3:0].				NOTRCPKT	NOEXTTRIG	NOCYCCNT	NOPRFCNT	Res.	CYCEVTENA	FOLDEVTENA	LSUEVTENA	SLEEPEVTENA	EXCEVTENA	CPIEVTENA	EXCTRCENA
r				r	r	r	r		rw	rw	rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	PCSAMPLENA	SYNCTAP[1:0]		CYCTAP	POSTINIT[3:0]				POSTRESET[3:0]				CYCCNTENA
			rw	rw		rw	rw				rw				rw

位 31:28 **NUMCOMP[3:0]**：设计的比较器数量 (Number of comparators implemented)，只读值
0x4：四个比较器

位 27 **NOTRCPKT**：支持跟踪采样和异常跟踪 (Trace sampling and exception tracing support)，只读值
0：支持

位 26 **NOEXTTRIG**：外部信号匹配 CMPMATCH 支持 (External match signal, CMPMATCH support)，只读值
0：支持

位 25 **NOCYCCNT**：循环计数器支持 (Cycle counter support)，只读值
0：支持

位 24 **NOPRFCNT**：性能分析计数器支持 (Profiling counter support)，只读值
0：支持

位 23 保留，必须保持复位值。

位 22 **CYCEVTENA**：使能 POSTCNT 下溢事件计数器数据包生成 (Enable for POSTCNT underflow event counter packet generation)
0：禁止
1：使能

位 21 **FOLDEVTENA**：使能折叠指令计数器上溢事件生成 (Enable for folded instruction counter overflow event generation)

0：禁止
1：使能

位 20 **LSUEVTENA**：使能 LSU 计数器上溢事件生成 (Enable for LSU counter overflow event generation)

0：禁止
1：使能

位 19 **SLEEPEVTENA**：使能睡眠计数器上溢事件生成 (Enable for sleep counter overflow event generation)

0：禁止
1：使能

位 18 **EXCEVTENA**：使能异常开销计数器上溢事件生成 (Enable for exception overhead counter overflow event generation)

0：禁止
1：使能

位 17 **CPIEVTENA**：使能 CPI 计数器上溢事件生成 (Enable for CPI counter overflow event generation)

0：禁止
1：使能

位 16 **EXCTRCENA**：使能异常跟踪生成 (Enable for exception trace generation)

0：禁止
1：使能

位 15:13 保留，必须保持复位值。

位 12 **PCSAMPLENA**：POSTCNT 计数器使用使能 (POSTCNT counter use enable)

使能将 POSTCNT 计数器用作定时器，用于定期生成 PC 样本数据包。

0：禁止
1：使能

位 11:10 **SYNCTAP[1:0]**：同步数据包计数器点击 CYCCNT 计数器的位置 (Position of synchronization packet counter tap on CYCCNT counter)

该选择确定同步数据包速率。

0x0：禁止——无同步数据包
0x1：点击位置为 CYCCNT[24]
0x2：点击位置为 CYCCNT[26]
0x3：点击位置为 CYCCNT[28]

位 9 **CYCTAP**：POSTCNT 点击 CYCCNT 计数器的位置 (Position of the POSTCNT tap on the CYCCNT counter)

0：点击位置为 CYCCNT[6]
1：点击位置为 CYCCNT[10]

位 8:5 **POSTINIT[3:0]**：POSTCNT 计数器初始值 (Initial value of the POSTCNT counter)
如果使能了 POSTCNT 计数器，则忽略对该字段的写操作（即在写入 POSTINIT 之前必须将 CYCEVTENA 或 PCSAMPLENA 复位）。

位 4:1 **POSTPRESET[3:0]**：POSTCNT 计数器的重载值 (Reload value of the POSTCNT counter)。

位 0 **CYCCNTENA**：CYCCNT 计数器使能 (CYCCNT counter enable)
0：禁止
1：使能

DWT 周期计数寄存器 (M7_DWT_CYCCNT)

DWT cycle count register

偏移地址：0x004

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CYCCNT[31:15]															
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CYCCNT[15:0]															
rw															

位 31:0 **CYCCNT[31:0]**：处理器时钟周期计数器 (Processor clock cycle counter)

DWT CPI 计数寄存器 (M7_DWT_CPICNT)

DWT CPI count register

偏移地址：0x008

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CPICNT[7:0]							
								rw							

位 31:8 保留，必须保持复位值。

位 7:0 **CPICNT[7:0]**：CPI 计数器 (CPI counter)
对执行多周期指令（由 DWT_LSUCNT 记录的多周期指令除外）进行计数，并对任何指令读取停止 (Stall) 进行计数。

DWT 异常计数寄存器 (M7_DWT_EXCCNT)

DWT exception count register

偏移地址：0x00C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	EXCCNT[7:0]							
								rw							

位 31:8 保留，必须保持复位值。

位 7:0 **EXCCNT[7:0]**：异常开销周期计数器 (Exception overhead cycle counter)
对在异常处理中花费的周期数进行计数。

DWT 睡眠计数寄存器 (M7_DWT_SLPCNT)

DWT sleep count register

偏移地址：0x010

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SLEEPCNT[7:0]							
								rw							

位 31:8 保留，必须保持复位值。

位 7:0 **SLEEPCNT[7:0]**：睡眠周期计数器 (Sleep cycle counter)
对在睡眠模式（WFI、WFE、退出时睡眠）下消耗的周期数进行计数。

DWT LSU 计数寄存器 (M7_DWT_LSUCNT)

DWT LSU count register

偏移地址：0x014

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LSUCNT[7:0]							
								rw							

位 31:8 保留，必须保持复位值。

位 7:0 **LSUCNT[7:0]**：加载存储计数器 (Load store counter)
对执行加载和存储指令所需的其他周期进行计数。

DWT 折叠计数寄存器 (M7_DWT_FOLDCNT)

DWT fold count register

偏移地址：0x018

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FOLDCNT[7:0]							
								rw							

位 31:8 保留，必须保持复位值。

位 7:0 **FOLDCNT[7:0]**：折叠指令计数器 (Folded instruction counter)
每一条只花费 0 周期的指令就递增计数。

DWT 程序计数器采样寄存器 (M7_DWT_PCSR)

DWT program counter sample register

偏移地址：0x01C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
EIASAMPLE[31:15]															
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EIASAMPLE[15:0]															
rw															

位 31:0 **EIASAMPLE[31:0]**：执行指令地址采样值 (Executed instruction address sample value)
采样程序计数器的当前值。

DWT 比较器寄存器 x (M7_DWT_COMPx)

DWT comparator register x

偏移地址：0x020 + x * 0x10（适用于 x = 0 到 3）

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
COMP[31:15]															
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
COMP[15:0]															
rw															

位 31:0 **COMP[31:0]**：用于比较的参考值 (Reference value for comparison)。

DWT 掩码寄存器 x (M7_DWT_MASKx)

DWT mask register x

偏移地址：0x024 + x * 0x10（适用于 x = 0 到 3）

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MASK[4:0]				
											rw				

位 31:5 保留，必须保持复位值。

位 4:0 **MASK[4:0]**：比较器掩码大小 (Comparator mask size)
为比较器 n 匹配的地址范围提供访问地址的忽略掩码的大小。调试器可写入 0b11111 到该字段，然后读回寄存器以确定支持的最大掩码大小。

DWT 功能寄存器 x (M7_DWT_FUNCTx)

DWT function register x

偏移地址：0x028 + x * 0x10（适用于 x = 0 到 3）

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	MATCHED	Res.	Res.	Res.	Res.	DATAVADDR1[3:0]			
							r					rw			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATAVADDR0[3:0]				DATAVSIZE[1:0]		LINK1ENA	DATAVMATCH	CYCMATCH	Res.	EMITRANGE	Res.	FUNCTION[3:0]			
rw				rw		rw	rw	rw		rw		rw			

位 31:25 保留，必须保持复位值。

位 24 **MATCHED**：比较器匹配 (Comparator match)，只读值
指示自上次读取寄存器以来是否发生了比较器匹配。

0：不匹配。
1：发生了匹配

位 23:20 保留，必须保持复位值。

位 19:16 **DATAVADDR1[3:0]**：第二比较器的比较器编号 (Comparator number of a second comparator)
当 DATAVMATCH 和 LNK1ENA 位均为 1 时，该字段可存储第二比较器的比较器编号用于关联地址比较。

位 15:12 **DATAVADDR0[3:0]**：比较器的比较器编号 (Comparator number of a comparator)
当 DATAVMATCH 和 LNK1ENA 位均为 1 时，该字段可存储比较器的比较器编号用于关联地址比较。

位 11:10 **DATAVSIZE[1:0]**：所需数据比较大小 (Size of required data comparison)
对于数据值匹配，指定所需数据比较的大小。

0x0：字节
0x1：半字
0x2：字
0x3：保留

位 9 **LNK1ENA**：支持第二比较器(Support of a second linked comparator)，只读值
指示是否支持使用第二比较器，只读值。

1：支持

位 8 **DATAVMATCH**：周期比较使能 (Cycle comparison enable)
0：执行地址比较
1：执行数据值比较

位 7 **CYCMATCH**：比较器 0 周期计数比较使能 (Cycle count comparison enable on comparator 0)
对于其他比较器，该字段保留。

0：无周期计数比较
1：将 DWT_COMP0 与周期计数器 DWT_CYCCNT 进行比较

位 6 保留，必须保持复位值。

位 5 **EMITRANGE**：数据跟踪地址偏移数据包使能 (Data trace address offset packet enable)
使能生成数据跟踪地址偏移数据包（包括数据地址位 0 到 15）
0：禁止
1：使能

位 4 保留，必须保持复位值。

位 3:0 **FUNCTION[3:0]**：通过比较器匹配的操作指示 (Action on comparator match)
该字段的含义取决于 DATAVMATCH 和 CYCMATCH 字段的设置。请参见*[5]*。

DWT CoreSight 外设标识寄存器 4 (M7_DWT_PIDR4)

DWT CoreSight peripheral identity register 4

偏移地址：0xFD0

复位值：0x0000 0004

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	4KCOUNT[3:0]				JEP106CON[3:0]			
								r				r			

位 31:8 保留，必须保持复位值。

位 7:4 **4KCOUNT[3:0]**：寄存器文件大小 (Register file size)
0x0：寄存器文件占用单个 4 KB 区域

位 3:0 **JEP106CON[3:0]**：JEP106 连续代码 (JEP106 continuation code)
0x4：ARM® JEDEC 代码

DWT CoreSight 外设标识寄存器 0 (M7_DWT_PIDR0)

DWT CoreSight peripheral identity register 0

偏移地址：0xFE0

复位值：0x0000 0002

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PARTNUM[7:0]							
								r							

位 31:8 保留，必须保持复位值。

位 7:0 **PARTNUM[7:0]**：产品编号字段 (Part number field)，位 [7:0]
0x02：DWT 产品编号

DWT CoreSight 外设标识寄存器 1 (M7_DWT_PIDR1)

DWT CoreSight peripheral identity register 1

偏移地址：0xFE4

复位值：0x0000 00B9

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	JEP106ID[3:0]				PARTNUM[11:8]			
								r				r			

位 31:8 保留，必须保持复位值。

位 7:4 **JEP106ID[3:0]**：JEP106 标识代码字段 (JEP106 identity code field)，位 [3:0]
0xB：ARM® JEDEC 代码

位 3:0 **PARTNUM[11:8]**：产品编号字段 (Part number field)，位 [11:8]
0x0：DWT 产品编号

DWT CoreSight 外设标识寄存器 2 (M7_DWT_PIDR2)

DWT CoreSight peripheral identity register 2

偏移地址：0xFE8

复位值：0x0000 003B

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REVISION[3:0]				JEDEC	JEP106ID[6:4]		
								r				r	r		

位 31:8 保留，必须保持复位值。

位 7:4 **REVISION[3:0]**：组件版本号 (Component revision number)
0x3：r0p4

位 3 **JEDEC**：JEDEC 分配的值 (JEDEC assigned value)
1：JEDEC 指定的设计人员 ID

位 2:0 **JEP106ID[6:4]**：JEP106 标识代码字段 (JEP106 identity code field)，位 [6:4]
0x3：ARM® JEDEC 代码

DWT CoreSight 外设标识寄存器 3 (M7_DWT_PIDR3)

DWT CoreSight peripheral identity register 3

偏移地址：0xFEC

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REVAND[3:0]				CMOD[3:0]			
								r				r			

位 31:8 保留，必须保持复位值。

位 7:4 **REVAND[3:0]**：金属修复版本 (Metal fix version)
0x0：无金属修复

位 3:0 **CMOD[3:0]**：客户修改 (Customer modified)
0x0：无客户修改

DWT CoreSight 组件标识寄存器 0 (M7_DWT_CIDR0)

DWT CoreSight component identity register 0

偏移地址：0xFF0

复位值：0x0000 000D

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[7:0]							
								r							

位 31:8 保留，必须保持复位值。

位 7:0 **PREAMBLE[7:0]**：组件 ID 字段 (Component ID field)，位 [7:0]
0x0D：公共 ID 值

DWT CoreSight 组件标识寄存器 1 (M7_DWT_CIDR1)

DWT CoreSight component identity register 1

偏移地址：0xFF4

复位值：0x0000 00E0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CLASS[3:0]				PREAMBLE[11:8]			
								r				r			

位 31:8 保留，必须保持复位值。

位 7:4 **CLASS[3:0]**：组件 ID 字段 (Component ID field)，位 [15:12]——组件类
0xE：跟踪发生器组件

位 3:0 **PREAMBLE[11:8]**：组件 ID 字段 (Component ID field)，位 [11:8]
0x0：公共 ID 值

DWT CoreSight 组件标识寄存器 2 (M7_DWT_CIDR2)

DWT CoreSight component identity register 2

偏移地址：0xFF8

复位值：0x0000 0005

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[19:12]							
								r							

位 31:8 保留，必须保持复位值。

位 7:0 **PREAMBLE[19:12]**：组件 ID 字段 (Component ID field)，位 [23:16]
0x05：公共 ID 值

DWT CoreSight 组件标识寄存器 3 (M7_DWT_CIDR3)

DWT CoreSight component identity register 3

偏移地址：0xFFC

复位值：0x0000 00B1

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[27:20]							
								r							

位 31:8 保留，必须保持复位值。

位 7:0 **PREAMBLE[27:20]**：组件 ID 字段 (Component ID field)，位 [31:24]
0xB1：公共 ID 值

Cortex-M7 DWT 寄存器映射和复位值

Cortex-M7 DWT 寄存器的 AHBD 地址范围为 0xE0001000 到 0xE0001FFC。

表 581. Cortex-M7 DWT 寄存器映射和复位值

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x000	M7_DWT_CTRL	NUMCOMP [3:0]				NOTRCPKT	NOEXTTRIG	NOCYCCNT	NOPRFCNT	Res.	CYCEVTENA	FOLDEVTENA	LSUEVTENA	SLEEPEVTENA	EXCEVTENA	CPIEVTENA	EXCTRCENA	Res.	Res.	Res.	PCSAMPLENA	SYNCTAP [1:0]		CYCTAP	POSINIT [3:0]				POSTPRESET [3:0]				CYCCNTENA
	Reset value	0	1	0	0	0	0	0	0		0	0	0	0	0	0	0				0	0	0	0	0	0	0	0	0	0	0	0	0
0x004	M7_DWT_CYCCNT	CYCCNT[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x008	M7_DWT_CPICNT	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CPICNT[7:0]							
	Reset value																									0	0	0	0	0	0	0	0
0x00C	M7_DWT_EXCCNT	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	EXCCNT[7:0]							
	Reset value																									0	0	0	0	0	0	0	0
0x010	M7_DWT_SLPCNT	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SLEEPCNT[7:0]							
	Reset value																									0	0	0	0	0	0	0	0
0x014	M7_DWT_LSUCNT	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LSUCNT[7:0]							
	Reset value																									0	0	0	0	0	0	0	0
0x018	M7_DWT_FOLDCNT	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	FOLDCNT[7:0]							
	Reset value																									0	0	0	0	0	0	0	0
0x01C	M7_DWT_PCSR	EIASAMPLE[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x020	M7_DWT_COMP0	COMP[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x024	M7_DWT_MASK0	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MASK[4:0]				
	Reset value																												0	0	0	0	0
0x028	M7_DWT_FUNCT0	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MATCHED	Res.	Res.	Res.	Res.	DATAVADDR1 [3:0]				DATAVADDR0 [3:0]				DATAVSIZE [1:0]		LNK1ENA	DATAVMATCH	CYCMATCH	Res.	EMITRANGE	Res.	FUNCTION [3:0]			
	Reset value								0					0	0	0	0	0	0	0	0	0	0	0	0	0		0		0	0	0	0
0x030	M7_DWT_COMP1	COMP[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x034	M7_DWT_MASK1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MASK[4:0]				
	Reset value																												0	0	0	0	0
0x038	M7_DWT_FUNCT1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MATCHED	Res.	Res.	Res.	Res.	DATAVADDR1 [3:0]				DATAVADDR0 [3:0]				DATAVSIZE [1:0]		LNK1ENA	DATAVMATCH	CYCMATCH	Res.	EMITRANGE	Res.	FUNCTION [3:0]			
	Reset value								0					0	0	0	0	0	0	0	0	0	0	0	0	0		0		0	0	0	0
0x040	M7_DWT_COMP2	COMP[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x044	M7_DWT_MASK2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MASK[4:0]				
	Reset value																												0	0	0	0	0

表 581. Cortex-M7 DWT 寄存器映射和复位值（续）

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x048	M7_DWT_FUNCT2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MATCHED	Res.	Res.	Res.	Res.	DATAVADDR1 [3:0]				DATAVADDR0 [3:0]				DATAVSIZE [1:0]		LNK1ENA	DATAVMATCH	CYCMATCH	Res.	EMITRANGE	Res.	FUNCTION [3:0]			
	Reset value								0					0	0	0	0	0	0	0	0	0	0	0	0	0		0		0	0	0	0
0x050	M7_DWT_COMP3	COMP[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x054	M7_DWT_MASK3	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.		MASK[4:0]			
	Reset value																												0	0	0	0	0
0x058	M7_DWT_FUNCT3	Res.	Res.	Res.	Res.	Res.	Res.	Res.	MATCHED	Res.	Res.	Res.	Res.	DATAVADDR1 [3:0]				DATAVADDR0 [3:0]				DATAVSIZE [1:0]		LNK1ENA	DATAVMATCH	CYCMATCH	Res.	EMITRANGE	Res.	FUNCTION [3:0]			
	Reset value								0					0	0	0	0	0	0	0	0	0	0	0	0	0		0		0	0	0	0
0xFD0	M7_DWT_PIDR4	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	4KCOUNT [3:0]				JEP106CON [3:0]			
	Reset value																									0	0	0	0	0	1	0	0
0xFD4	M7_DWT_PIDR5	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	Reset value																																
0xFD8	M7_DWT_PIDR6	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	Reset value																																
0xFDC	M7_DWT_PIDR7	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	Reset value																																
0xFE0	M7_DWT_PIDR0	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PARTNUM [7:0]							
	Reset value																									0	0	0	0	0	0	1	0
0xFE4	M7_DWT_PIDR1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	JEP106ID [3:0]				PARTNUM [11:8]			
	Reset value																									1	0	1	1	0	0	0	0
0xFE8	M7_DWT_PIDR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REVISION [3:0]				JEDEC	JEP106ID [6:4]		
	Reset value																									0	0	1	1	1	0	1	1
0xFEC	M7_DWT_PIDR3	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REVAND[3:0]				CMOD[3:0]			
	Reset value																									0	0	0	0	0	0	0	0
0xFF0	M7_DWT_CIDR0	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[7:0]							
	Reset value																									0	0	0	0	1	1	0	1
0xFF4	M7_DWT_CIDR1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CLASS[3:0]				PREAMBLE [11:8]			
	Reset value																									0	1	1	1	0	0	0	0
0xFF8	M7_DWT_CIDR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[19:12]							
	Reset value																									0	0	0	0	0	1	0	1
0xFFC	M7_DWT_CIDR3	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[27:20]							
	Reset value																									1	0	1	1	0	0	0	1

60.6.3 Cortex-M7 指令跟踪宏单元 (ITM)

ITM 将跟踪信息生成为数据包。有四个源可生成数据包。如果多个源同时生成数据包，则 ITM 仲裁数据包输出的顺序。按照优先级降序排列，四个源依次为：

1. 软件跟踪

   软件可直接写入任意 32 x 32 位 ITM 激励寄存器以生成数据包。可以对每个端口的权限级别进行编程。当软件写入使能的激励端口时，ITM 将端口标识、写访问的大小和被写入的数据合并成其写入 FIFO 的数据包。ITM 将 FIFO 中的数据包输出到跟踪总线上。读取激励端口寄存器会返回激励寄存器的状态（空或挂起），通过位 0 指示。

2. 硬件跟踪

   DWT 生成跟踪数据包以响应数据跟踪事件、PC 采样或性能分析计数器回卷。ITM 将这些数据包输出在跟踪总线上。

3. 局域时间戳

   ITM 包含一个 21 位计数器，由（预分频的）处理器时钟提供时钟信号。计数器值以时间戳数据包的形式输出到跟踪总线上。每生成一个时间戳数据包，计数器就复位为零。因此，时间戳指示自从上一个时间戳数据包以来经过的时间。

4. 全局系统时间戳

   时间戳还可以使用时间戳发生器组件的系统级 64 位计数器值生成。

Cortex-M7 ITM 寄存器

ITM 激励寄存器 x (M7_ITM_STIMx)

ITM stimulus register x

偏移地址：0x000 + x * 0x4（x = 0 到 31）

复位值：未定义

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
STIMULUS[31:15]															
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STIMULUS[15:0]															
rw															

位 31:0 **STIMULUS[31:0]**：软件事件数据包/FIFOREADY (Software event packet / FIFOREADY)
写数据以软件事件数据包的形式输出在跟踪总线上。读取时，位 0 为 FIFOREADY 指示符：

0：激励端口缓冲区已满（或端口被禁止）
1：激励端口可接受新的写数据

ITM 跟踪使能寄存器 (M7_ITM_TER)

ITM trace enable register

偏移地址：0x080

复位值：0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
STIMENA[31:15]															
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STIMENA[15:0]															
rw															

位 31:0 **STIMENA[31:0]**：激励端口使能 (Stimulus port enable)
每一位 n (0:31) 使能与 M7_ITM_STIMn 寄存器关联的激励端口。

0：禁止端口
1：使能端口

ITM 跟踪特权寄存器 (M7_ITM_TPR)

ITM trace privilege registers

偏移地址：0xE00

复位值：0x00000000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PRIVMASK[31:15]															
rw															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PRIVMASK[15:0]															
rw															

位 31:0 **PRIVMASK[31:0]**：使能非特权访问 ITM 激励端口 (Enable unprivileged access to ITM stimulus ports)
每一位控制八个激励端口：

0bXXX0：允许非特权访问端口 0 到 7
0bXXX1：仅允许特权访问端口 0 到 7
0bXX0X：允许非特权访问端口 8 到 15
0bXX1X：仅允许特权访问端口 8 到 15
0bX0XX：允许非特权访问端口 16 到 23
0bX1XX：仅允许特权访问端口 16 到 23
0b0XXX：允许非特权访问端口 24 到 31
0b1XXX：仅允许特权访问端口 24 到 31

ITM 跟踪控制寄存器 (M7_ITM_TCR)

ITM trace control register

偏移地址：0xE80

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BUSY	TRACEBUSID[6:0]						
								rw	rw						

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	GTSFREQ[1:0]		TSPRESCALE [1:0]		Res.	Res.	Res.	SWOE NA	TXENA	SYNC ENA	TSENA	ITM ENA
				rw		rw					r	rw	rw	rw	rw

位 31:24 保留，必须保持复位值。

位 23 **BUSY**：ITM 忙碌 (ITM busy)

指示 ITM 当前是否正在处理事件，只读值：

0：不忙
1：忙

位 22:16 **TRACEBUSID[6:0]**：多源跟踪流格式化标识符 (Identifier for multi-source trace stream formatting)

如果使用了多源跟踪，则调试器必须将一个非零值写入该字段。注意：系统中每个跟踪源必须使用不同的 ID。

位 15:12 保留，必须保持复位值。

位 11:10 **GTSFREQ[1:0]**：全局时间戳频率 (Global timestamp frequency)

定义 ITM 生成全局时间戳的频率（基于全局时间戳时钟频率）或禁止生成全局时间戳。可能的值为：

0x0：禁止生成全局时间戳
0x1：只要 ITM 检测到全局时间戳计数器位 [63:7] 发生变化，就生成时间戳请求；大概每 128 个周期就生成一次。
0x2：只要 ITM 检测到全局时间戳计数器位 [63:13] 发生变化，就生成时间戳请求；大概每 8192 个周期就生成一次。
0x3：如果输出 FIFO 为空，则在每个数据包后生成一个时间戳

位 9:8 TSPRESCALE[1:0]：局部时间戳预分频 (Local timestamp prescale)

预分频与跟踪数据包参考时钟一起使用，可能的值为：

0x0：无预分频
0x1：4 分频
0x2：16 分频
0x3：64 分频

位 7:5 保留，必须保持复位值。

位 4 **SWOENA**：时间戳计数器异步时钟使能 (Asynchronous clocking enable for the timestamp counter)，只读值

0：时间戳计数器使用处理器时钟

位 3 **TXENA**：硬件事件数据包转发使能 (Hardware event packet forwarding enable)
使能将硬件数据包从 DWT 单元转发到跟踪端口。

0：禁止
1：使能

位 2 **SYNCENA**：同步数据包传输使能 (Synchronization packet transmission enable)
如果调试器将该位置 1，则还必须配置 DWT 中 DWT_CTRL 寄存器 SYNCTAP 字段，以获得正确的同步速度。

0：禁止
1：使能

位 1 **TSENA**：局部时间戳生成使能 (Local timestamp generation enable)
0：禁止
1：使能

位 0 **ITMENA**：ITM 使能 (ITM enable)
0：禁止
1：使能

ITM CoreSight 外设标识寄存器 4 (M7_ITM_PIDR4)

ITM CoreSight peripheral identity register 4

偏移地址：0xFD0

复位值：0x0000 0004

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	4KCOUNT[3:0]				JEP106CON[3:0]			
								r				r			

位 31:8 保留，必须保持复位值。

位 7:4 **4KCOUNT[3:0]**：寄存器文件大小 (Register file size)
0x0：寄存器文件占用单个 4 KB 区域

位 3:0 **JEP106CON[3:0]**：JEP106 连续代码 (JEP106 continuation code)
0x4：ARM® JEDEC 代码

ITM CoreSight 外设标识寄存器 0 (M7_ITM_PIDR0)

ITM CoreSight peripheral identity register 0

偏移地址：0xFE0

复位值：0x0000 0001

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PARTNUM[7:0]							
								r							

位 31:8 保留，必须保持复位值。

位 7:0 **PARTNUM[7:0]**：产品编号字段 (Part number field)，位 [7:0]
0x01：ITM 产品编号

ITM CoreSight 外设标识寄存器 1 (M7_ITM_PIDR1)

ITM CoreSight peripheral identity register 1

偏移地址：0xFE4

复位值：0x0000 00B0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	JEP106ID[3:0]				PARTNUM[11:8]			
								r				r			

位 31:8 保留，必须保持复位值。

位 7:4 **JEP106ID[3:0]**：JEP106 标识代码字段 (JEP106 identity code field)，位 [3:0]
0xB：ARM® JEDEC 代码

位 3:0 **PARTNUM[11:8]**：产品编号字段 (Part number field)，位 [11:8]
0x1：ITM 产品编号

ITM CoreSight 外设标识寄存器 2 (M7_ITM_PIDR2)

ITM CoreSight peripheral identity register 2

偏移地址：0xFE8

复位值：0x0000 003B

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REVISION[3:0]				JEDEC	JEP106ID[6:4]		
								r				r	r		

位 31:8 保留，必须保持复位值。

位 7:4 **REVISION[3:0]**：组件版本号 (Component revision number)
0x3：r0p4

位 3 **JEDEC**：JEDEC 分配的值 (JEDEC assigned value)
1：JEDEC 指定的设计人员 ID

位 2:0 **JEP106ID[6:4]**：JEP106 标识代码字段 (JEP106 identity code field)，位 [6:4]
0x3：ARM® JEDEC 代码

ITM CoreSight 外设标识寄存器 3 (M7_ITM_PIDR3)

ITM CoreSight peripheral identity register 3

偏移地址：0xFEC

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REVAND[3:0]				CMOD[3:0]			
								r				r			

位 31:8 保留，必须保持复位值。

位 7:4 **REVAND[3:0]**：金属修复版本 (Metal fix version)
0x0：无金属修复

位 3:0 **CMOD[3:0]**：客户修改 (Customer modified)
0x0：无客户修改

ITM CoreSight 组件标识寄存器 0 (M7_ITM_CIDR0)

ITM CoreSight component identity register 0

偏移地址：0xFF0

复位值：0x0000 000D

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[7:0]							
								r							

位 31:8 保留，必须保持复位值。

位 7:0 **PREAMBLE[7:0]**：组件 ID 字段 (Component ID field)，位 [7:0]
0x0D：公共 ID 值

ITM CoreSight 组件标识寄存器 1 (M7_ITM_CIDR1)

ITM CoreSight component identity register 1

偏移地址：0xFF4

复位值：0x0000 00E0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CLASS[3:0]				PREAMBLE[11:8]			
								r				r			

位 31:8 保留，必须保持复位值。

位 7:4 **CLASS[3:0]**：组件 ID 字段 (Component ID field)，位 [15:12]——组件类
0xE：跟踪发生器组件

位 3:0 **PREAMBLE[11:8]**：组件 ID 字段 (Component ID field)，位 [11:8]
0x0：公共 ID 值

ITM CoreSight 组件标识寄存器 2 (M7_ITM_CIDR2)

ITM CoreSight component identity register 2

偏移地址：0xFF8

复位值：0x0000 0005

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[19:12]							
								r							

位 31:8 保留，必须保持复位值。

位 7:0 **PREAMBLE[19:12]**：组件 ID 字段 (Component ID field)，位 [23:16]
0x05：公共 ID 值

ITM CoreSight 组件标识寄存器 3 (M7_ITM_CIDR3)

ITM CoreSight component identity register 3

偏移地址：0xFFC

复位值：0x0000 00B1

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[27:20]							
								r							

位 31:8 保留，必须保持复位值。

位 7:0 **PREAMBLE[27:20]**：组件 ID 字段 (Component ID field)，位 [31:24]
0xB1：公共 ID 值

Cortex-M7 ITM 寄存器映射和复位值

ITM 寄存器的 AHBD 地址范围为 0xE0000000 到 0xE0000FFC。

表 582. Cortex-M7 ITM 寄存器映射和复位值

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x000 to 0x07C	M7_ITM_STIM0-31	STIMULUS[31:0]																															
	Reset value																																
0x0E00	M7_ITM_TER	STIMENA[31:0]																															
	Reset value																																
0x0E40	M7_ITM_TPR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PRIVMASK[3:0]			
	Reset value																							0	0	0	0	0	0	0	0	0	0
0xE80	M7_ITM_TCR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	BUSY	TRACEBUSID[6:0]							Res.	Res.	Res.	Res.	GTSFREQ[1:0]		TSPRESCALE[1:0]		Res.	Res.	Res.	SWOENA	TXENA	SYNCENA	TSENA	ITMENA
	Reset value									0	0	0	0	0	0	0	0					0	0	0	0				0	0	0	0	0
0xFD0	M7_ITM_PIDR4	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	4KCOUNT[3:0]				JEP106CON[3:0]			
	Reset value																									0	0	0	0	0	1	0	0
0xFD4	M7_ITM_PIDR5	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	Reset value																																
0xFD8	M7_ITM_PIDR6	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	Reset value																																
0xFDC	M7_ITM_PIDR7	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	Reset value																																
0xFE0	M7_ITM_PIDR0	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PARTNUM[7:0]							
	Reset value																									0	0	0	0	0	0	0	1
0xFE4	M7_ITM_PIDR1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	JEP106ID[3:0]				PARTNUM[11:8]			
	Reset value																									1	0	1	1	0	0	0	0
0xFE8	M7_ITM_PIDR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REVISION[3:0]				JEDEC	JEP106ID[6:4]		
	Reset value																									0	0	1	1	1	0	1	1
0xFEC	M7_ITM_PIDR3	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REVAND[3:0]				CMOD[3:0]			
	Reset value																									0	0	0	0	0	0	0	0
0xFF0	M7_ITM_CIDR0	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[7:0]							
	Reset value																									0	0	0	0	1	1	0	1
0xFF4	M7_ITM_CIDR1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CLASS[3:0]				PREAMBLE[11:8]			
	Reset value																									0	1	1	1	0	0	0	0
0xFF8	M7_ITM_CIDR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[19:12]							
	Reset value																									0	0	0	0	0	1	0	1
0xFFC	M7_ITM_CIDR3	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[27:20]							
	Reset value																									1	0	1	1	0	0	0	1

60.6.4 Cortex-M7 断点单元 (FPB)

FPB 允许设置硬件断点， 其含有八个比较器，这些比较器监视指令读取地址，并在检测到匹配时返回断点指令。Cortex-M7 FPB 不支持 Flash 补丁功能。

Cortex-M7 FPB 寄存器

FPB 控制寄存器 (M7_FPB_CTRL)

FPB control register

偏移地址：0x000

复位值：0x0000 0080

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	NUM_CODE[6:4]			NUM_LIT[3:0]				NUM_CODE[3:0]				Res.	Res.	KEY	ENABLE
	r			r				r						rw	rw

位 31:15 保留，必须保持复位值。

位 14:12 **NUM_CODE[6:4]**：指令地址比较器数量字段 (Instruction address comparator number field)，三个最高位

该只读字段在三个最高位保存支持的指令地址比较器数量。

0x0：用来保存比较器数量的最高位都为 0

位 11:8 **NUM_LIT[3:0]**：支持的文字地址比较器数量 (Number of literal address comparators supported)，只读值。

0x0：无支持的文字比较器

位 7:4 **NUM_CODE[3:0]**：指令地址比较器数量字段 (Instruction address comparator number field)，四个最低位

该只读字段在四个最低位保存支持的指令地址比较器数量。

0x8：支持 8 个指令比较器

位 3:2 保留，必须保持复位值

位 1 **KEY**：写保护密钥 (Write protect key)

如果该位未设置为 1，则忽略对 M7_FPB_CTRL 寄存器的写操作。

位 0 **ENABLE**：FPB 使能 (FPB enable)

0：禁止

1：使能

FPB 重映射寄存器 (M7_FPB_REMAP)

FPB remap register

偏移地址：0x004

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	RMPSPT	REMAP[23:11]												
		r	rw												

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REMAP[10:0]											Res.	Res.	Res.	Res.	Res.
rw															

位 31:30 保留，必须保持复位值。

位 29 **RMPSPT**：Flash 补丁重映射支持 (Flash patch remap support)，只读值
0：不支持重映射

位 28:5 **REMAP[23:0]**：保留——不支持

位 4:0 保留，必须保持复位值。

FPB 比较器寄存器 (M7_FPB_COMPx)

FPB comparator registers

偏移地址：0x008 + x * 0x4（适用于 x = 0 到 7）

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
REPLACE[1:0]		Res.	COMP[26:14]												
rw			rw												

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
COMP[13:0]														Res.	ENABLE
rw															rw

位 31:30 **REPLACE[1:0]**：通过 COMP 与指令读取地址匹配的行为指示 (Behavior upon COMP versus instruction fetch address match)
定义 COMP 字段和指令读取地址之间发生匹配时的行为：

0x0：预留
0x1：断点在低位半字上，高位半字不受影响。
0x2：断点在高位半字上，低位半字不受影响。
0x3：断点在高位半字和低位半字上。

位 29 保留，必须保持复位值。

位 28:2 **COMP[26:0]**：与代码存储器访问地址进行比较的值 (Value to compare with code memory access address)
与指令代码存储器（0x00000000 到 0x1FFFFFFF）的访问地址位 28:2 进行比较的值。如果发生匹配，则由 REPLACE 字段定义要采取的操作。

位 1 保留，必须保持复位值。

位 0 **ENABLE**：比较器使能 (Comparator enable)
仅当该位与 M7_FPB_CTRL 寄存器中的 FPB ENABLE 位均置 1 时，比较器才使能。

0：禁止
1：使能

FPB CoreSight 外设标识寄存器 4 (M7_FPB_PIDR4)

FPB CoreSight peripheral identity register 4

偏移地址：0xFD0

复位值：0x0000 0004

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	4KCOUNT[3:0]				JEP106CON[3:0]			
								r				r			

位 31:8 保留，必须保持复位值。

位 7:4 **4KCOUNT[3:0]**：寄存器文件大小 (Register file size)
0x0：寄存器文件占用单个 4 KB 区域

位 3:0 **JEP106CON[3:0]**：JEP106 连续代码 (JEP106 continuation code)
0x4：ARM® JEDEC 代码

FPB CoreSight 外设标识寄存器 0 (M7_FPB_PIDR0)

FPB CoreSight peripheral identity register 0

偏移地址：0xFE0

复位值：0x0000 000C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PARTNUM[7:0]							
								r							

位 31:8 保留，必须保持复位值。

位 7:0 **PARTNUM[7:0]**：产品编号字段 (Part number field)，位 [7:0]
0x0C：FPB 产品编号

FPB CoreSight 外设标识寄存器 1 (M7_FPB_PIDR1)

FPB CoreSight peripheral identity register 1

偏移地址：0xFE4

复位值：0x0000 00B0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	JEP106ID[3:0]				PARTNUM[11:8]			
								r				r			

位 31:8 保留，必须保持复位值。

位 7:4 **JEP106ID[3:0]**：JEP106 标识代码字段 (JEP106 identity code field)，位 [3:0]
0xB：ARM® JEDEC 代码

位 3:0 **PARTNUM[11:8]**：产品编号字段 (Part number field)，位 [11:8]
0x0：FPB 产品编号

FPB CoreSight 外设标识寄存器 2 (M7_FPB_PIDR2)

FPB CoreSight peripheral identity register 2

偏移地址：0xFE8

复位值：0x0000 002B

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REVISION[3:0]				JEDEC	JEP106ID[6:4]		
								r				r	r		

位 31:8 保留，必须保持复位值。

位 7:4 **REVISION[3:0]**：组件版本号 (Component revision number)
0x2：r0p3

位 3 **JEDEC**：JEDEC 分配的值 (JEDEC assigned value)
1：JEDEC 指定的设计人员 ID

位 2:0 **JEP106ID[6:4]**：JEP106 标识代码字段 (JEP106 identity code field)，位 [6:4]
0x3：ARM® JEDEC 代码

FPB CoreSight 外设标识寄存器 3 (M7_FPB_PIDR3)

FPB CoreSight peripheral identity register 3

偏移地址：0xFEC

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REVAND[3:0]				CMOD[3:0]			
								r				r			

位 31:8 保留，必须保持复位值。

位 7:4 **REVAND[3:0]**：金属修复版本 (Metal fix version)
0x0：无金属修复

位 3:0 **CMOD[3:0]**：客户修改 (Customer modified)
0x0：无客户修改

FPB CoreSight 组件标识寄存器 0 (M7_FPB_CIDR0)

FPB CoreSight component identity register 0

偏移地址：0xFF0

复位值：0x0000 000D

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[7:0]							
								r							

位 31:8 保留，必须保持复位值。

位 7:0 **PREAMBLE[7:0]**：组件 ID 字段 (Component ID field)，位 [7:0]
0x0D：公共 ID 值

FPB CoreSight 组件标识寄存器 1 (M7_FPB_CIDR1)

FPB CoreSight component identity register 1

偏移地址：0xFF4

复位值：0x0000 00E0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CLASS[3:0]				PREAMBLE[11:8]			
								r				r			

位 31:8 保留，必须保持复位值。

位 7:4 **CLASS[3:0]**：组件 ID 字段 (Component ID field)，位 [15:12]——组件类
0xE：跟踪发生器组件

位 3:0 **PREAMBLE[11:8]**：组件 ID 字段 (Component ID field)，位 [11:8]
0x0：公共 ID 值

FPB CoreSight 组件标识寄存器 2 (M7_FPB_CIDR2)

FPB CoreSight component identity register 2

偏移地址：0xFF8

复位值：0x0000 0005

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[19:12]							
								r							

位 31:8 保留，必须保持复位值。

位 7:0 **PREAMBLE[19:12]**：组件 ID 字段 (Component ID field)，位 [23:16]
0x05：公共 ID 值

FPB CoreSight 组件标识寄存器 3 (M7_FPB_CIDR3)

FPB CoreSight component identity register 3

偏移地址：0xFFC

复位值：0x0000 00B1

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[27:20]							
								r							

位 31:8 保留，必须保持复位值。

位 7:0 **PREAMBLE[27:20]**：组件 ID 字段 (Component ID field)，位 [31:24]
0xB1：公共 ID 值

Cortex-M7 FPB 寄存器映射和复位值

Cortex-M7 FPB 的地址范围为 0xE0002000 到 0xE0002FFC。

表 583. Cortex-M7 FPB 寄存器映射和复位值

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x000	M7_FPB_CTRL	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	NUM_CODE [6:4]			NUM_LIT [3:0]				NUM_CODE [3:0]				Res.	Res.	KEY	ENABLE
	Reset value																		0	0	0	0	0	0	0	1	0	0	0			0	0
0x004	M7_FPB_REMAP	Res.	Res.	RMPSPT	REMAP[23:0]																								Res.	Res.	Res.	Res.	Res.
	Reset value			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0					
0x008 to 0x024	M7_FPB_COMP0-7	REPLACE [1:0]		Res.	COMP[26:0]																										Res.	ENABLE
	Reset value																																0
0xFD0	M7_FPB_PIDR4	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	4KCOUNT [3:0]				JEP106CON [3:0]			
	Reset value																									0	0	0	0	0	1	0	0
0xFD4	M7_FPB_PIDR5	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	Reset value																																
0xFD8	M7_FPB_PIDR6	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	Reset value																																
0xFDC	M7_FPB_PIDR7	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	Reset value																																

表 583. Cortex-M7 FPB 寄存器映射和复位值（续）

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0xFE0	M7_FPB_PIDR0	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PARTNUM [7:0]							
	Reset value																									0	0	0	0	1	1	0	0
0xFE4	M7_FPB_PIDR1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	JEP106ID [3:0]				PARTNUM [11:8]			
	Reset value																									1	0	1	1	0	0	0	0
0xFE8	M7_FPB_PIDR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REVISION [3:0]				JEDEC	JEP106ID [6:4]		
	Reset value																									0	0	1	0	1	0	1	1
0xFEC	M7_FPB_PIDR3	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REVAND[3:0]				CMOD[3:0]			
	Reset value																									0	0	0	0	0	0	0	0
0xFF0	M7_FPB_CIDR0	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[7:0]							
	Reset value																									0	0	0	0	1	1	0	1
0xFF4	M7_FPB_CIDR1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CLASS[3:0]				PREAMBLE [11:8]			
	Reset value																									1	1	1	0	0	0	0	0
0xFF8	M7_FPB_CIDR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[19:12]							
	Reset value																									0	0	0	0	0	1	0	1
0xFFC	M7_FPB_CIDR3	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[27:20]							
	Reset value																									1	0	1	1	0	0	0	1

60.6.5 Cortex-M7 嵌入式跟踪宏单元 (ETM)

Cortex-M7 ETM 是一款与 CPU 紧密结合的 CoreSight™ 组件。ETM 生成跟踪数据包，允许跟踪 Cortex-M7 内核的执行。在 STM32H7 中，ETM 仅配置为指令跟踪，因此跟踪信息中不包含数据访问。

ETM 通过处理器跟踪接口从 CPU 接收信息，包括：

- 在同一周期内执行的指令数量
- 程序流的变化
- 当前处理器指令状态
- 由加载和存储指令访问的存储器位置的地址
- 传输的类型、方向和大小
- 条件代码信息
- 异常信息
- 等待中断状态信息

有关更多信息，请参见《ARM® CoreSight™ ETM™-M7 技术参考手册》*[6]*。

Cortex-M7 ETM 寄存器

ETM 编程控制寄存器 (M7_ETM_PRGCTL)

ETM programming control register

偏移地址：0x004

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	EN
															rw

位 31:1 保留，必须保持复位值。

位 0 **EN**：跟踪程序使能 (Trace program enable)

0：禁止跟踪单元

1：使能跟踪单元

ETM 处理器选择控制寄存器 (M7_ETM_PROCSEL)

ETM processor select control register

偏移地址：0x008

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PROC SEL
															rw

位 31:1 保留，必须保持复位值。

位 0 **PROCSEL**：处理器选择 (Processor select)

只有 Cortex-M7 使用此 ETM，因此该字段无效。

ETM 状态寄存器 (M7_ETM_STAT)

ETM status register

偏移地址：0x00C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PMSTABLE	IDLE
														r	r

位 31:2 保留，必须保持复位值。

位 1 **PMSTABLE**：编程器模型稳定 (Programmers model stable)
指示 ETM 寄存器是否处于稳定状态并可读取。
0：寄存器处于不稳定状态
1：寄存器处于稳定状态

位 0 **IDLE**：跟踪单元处于未激活状态 (Trace unit inactive)
0：ETM 处于非空闲状态
1：ETM 处于空闲状态

ETM 跟踪配置寄存器 (M7_ETM_CONFIG)

ETM trace configuration register

偏移地址：0x010

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DV	DA
														rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	RS	TS	COND[2:0]			Res.	Res.	Res.	CCI	BB	INSTP0[1:0]		Res.
			rw	rw	rw						rw	rw	r		

位 31:18 保留，必须保持复位值。

位 17 **DV**：数据值跟踪 (Data value tracing)，只读值
0：禁止

位 16 **DA**：数据地址跟踪 (Data address tracing)，只读
0：禁止

位 15:13 保留，必须保持复位值。

位 12 **RS**：返回堆栈使能 (Return stack enable)
0：禁止
1：使能

位 11 **TS**：全局时间戳跟踪 (Global timestamp tracing)
0：禁止
1：使能

位 10:8 **COND[2:0]**：条件指令跟踪 (Conditional instruction tracing)
0x0：禁止条件指令跟踪
0x1：跟踪条件加载指令
0x2：跟踪条件存储指令
0x3：跟踪条件加载和存储指令
0x7：跟踪所有的条件指令
其它：保留

位 7:5 保留，必须保持复位值。

位 4 **CCI**：在指令跟踪中循环计数 (Cycle counting in instruction trace)
0：禁止
1：使能

位 3 **BB**：分支广播模式 (Branch broadcast mode)
0：禁止
1：使能

位 2:1 **INSTP0[1:0]**：确定哪条指令为 P0 指令 (Determines which instructions are P0 instructions)，只读值
0x0：仅分支为 P0 指令

位 0 保留，必须保持复位值。

ETM 事件控制 0 register (M7_ETM_EVENTCTL0)

ETM event control 0 register

当跟踪单元被禁止时，仅接受写操作

偏移地址：0x020

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TYPE1	Res.	Res.	Res.	SEL1[3:0]				TYPE0	Res.	Res.	Res.	SEL0[3:0]			
rw				rw				rw				rw			

位 31:16 保留，必须保持复位值。

位 15 **TYPE1**：事件 1 资源类型 (Resource type for event 1)
0：单选资源
1：布尔组合资源对

位 14:12 保留，必须保持复位值。

位 11:8 **SEL1[3:0]**：事件 1 的资源/布尔组合资源对 (Resource / Boolean combined resource pair, for event 1)

当 TYPE1 为 0 时，从位 [3:0] 定义的 0-15 中选择一个单选资源
当 TYPE1 为 1 时，从位 [2:0] 定义的 0-7 中选择一个布尔组合资源对

位 7 **TYPE0**：事件 0 资源类型 (Resource type for event 0)

0：单选资源
1：布尔组合资源对

位 6:4 保留，必须保持复位值。

位 3:0 **SEL0[3:0]**：事件 0 的资源/布尔组合资源对 (Resource / Boolean combined resource pair for event 0)

当 TYPE0 为 0 时，从位 [3:0] 定义的 0-15 中选择一个单选资源
当 TYPE0 为 1 时，从位 [2:0] 定义的 0-7 中选择一个布尔组合资源对

ETM 事件控制 1 寄存器 (M7_ETM_EVENTCTL1)

ETM event control 1 register

当跟踪单元被禁止时，仅接受写操作

偏移地址：0x024

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	LPOVERRIDE	ATB	Res.	Res.	Res.	Res.	Res.	Res.	Res.	INSTEN[3:0]			
			rw	rw								rw			

位 31:13 保留，必须保持复位值。

位 12 **LPOVERRIDE**：低功耗状态行为覆盖 (Low power state behavior override)

0：低功耗状态下保持正常行为
1：进入低功耗状态不影响资源和事件跟踪生成

位 11 **ATB**：ATB 触发使能 (ATB trigger enable)

0：禁止
1：使能

位 10:4 保留，必须保持复位值。

位 3:0 **INSTEN[3:0]**：指令跟踪事件元素使能 (Instruction trace event element enable)
每一位对应一个事件：

0bXXX0：事件 0 不引起一个事件元素
0bXXX1：事件 0 引起一个事件元素
0bXX0X：事件 1 不引一个起事件元素
0bXX1X：事件 1 引起一个事件元素
0bX0XX：事件 2 不引起一个事件元素
0bX1XX：事件 2 引起一个事件元素
0b0XXX：事件 3 不引起一个事件元素
0b1XXX：事件 3 引起一个事件元素

ETM 停止控制寄存器 (M7_ETM_STALLCTL)

ETM stall control register

当跟踪单元被禁止时，仅接受写操作

偏移地址：0x02C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	DSTALL	ISTALL	Res.	Res.	Res.	Res.	LEVEL[1:0]		Res.	Res.
						rw	rw					rw			

位 31:10 保留，必须保持复位值。

位 9 **DSTALL**：基于数据跟踪缓冲区空间停止处理器 (Stall processor based on data trace buffer space)
0：不停止处理器
1：停止处理器

位 8 **ISTALL**：基于指令跟踪缓冲区空间停止处理器 (Stall processor based on instruction trace buffer space)
0：不停止处理器
1：停止处理器

位 7:4 保留，必须保持复位值。

位 3:2 **LEVEL[1:0]**：停止阈值级别 (Stalling threshold level)
低阈值级别可将停止的处理器数量降到最少，但存在更高的 FIFO 上溢风险。高阈值级别可将 FIFO 上溢风险降到最低，但增加了停止的处理器数量。

位 1:0 保留，必须保持复位值。

ETM 全局时间戳控制寄存器 (M7_ETM_TSCTL)

ETM global timestamp control register

当跟踪单元被禁止时，仅接受写操作

偏移地址：0x030

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TYPE	Res.	Res.	Res.	SEL[1:0]			
								rw				rw			

位 31:8 保留，必须保持复位值。

位 7 **TYPE**：时间戳插入的资源类型 (Resource type for time stamp insertion)
0：单选资源
1：布尔组合资源对

位 6:4 保留，必须保持复位值。

位 3:0 **SEL[3:0]**：资源/布尔组合资源对 (Resource / Boolean combined resource pair)
当 TYPE 为 0 时，从位 [3:0] 定义的 0-15 中选择一个单选资源
当 TYPE 为 1 时，从位 [2:0] 定义的 0-7 中选择一个布尔组合资源对

ETM 同步周期寄存器 (M7_ETM_SYNCP)

ETM synchronization period register

偏移地址：0x034

复位值：0x0000 000A

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PERIOD[4:0]				
											r				

位 31:5 保留，必须保持复位值。

位 4:0 **PERIOD[4:0]**：同步请求之间的跟踪字节 (Trace bytes between synchronization requests)
定义跟踪同步请求之间跟踪信息的字节数。
0xA：1024 个字节

ETM 循环计数控制寄存器 (M7_ETM_CCCTL)

ETM cycle count control register

偏移地址：0x038

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	PERIOD[11:0]											
				rw											

位 31:12 保留，必须保持复位值。

位 11:0 **THRESHOLD[11:0]**：指令跟踪循环计数的阈值 (Threshold value for instruction trace cycle counting)

该阈值表示循环计数跟踪数据包之间的最小间隔。

0x0：预留

其他：阈值

ETM 跟踪 ID 寄存器 (M7_ETM_TRACEID)

ETM trace ID register

偏移地址：0x040

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.						TRACEID[6:0]						
									rw						

位 31:7 保留，必须保持复位值。

位 6:0 **TRACEID[6:0]**：跟踪 ID (Trace ID)

0x00：预留

0x01 到 0x6F：有效 ID

0x70 到 0x7F：保留

ETM ViewInst 主控制寄存器 (M7_ETM_VICTL)

ETM ViewInst main control register

偏移地址：0x080

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	EXLEVEL_S3	Res.	Res.	EXLEVEL_S0
												rw			rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	TRCERR	TRCRESET	SSSTATUS	Res.	TYPE				SEL[3:0]			
				rw	rw	rw		rw				rw			

位 31:20 保留，必须保持复位值。

位 19 **EXLEVEL_S3**：禁止跟踪，异常级别 3 (Trace disable, exception level 3)
禁止安全状态下对指定级别的异常进行跟踪（适用于异常级别 3）。
0：使能该异常级别的 ViewInst
1：禁止该异常级别的 ViewInst

位 18:17 保留，必须保持复位值。

位 16 **EXLEVEL_S0**：禁止跟踪，异常级别 0 (Trace disable, exception level 0)
禁止安全状态下对指定级别的异常进行跟踪（适用于异常级别 0）。
0：使能该异常级别的 ViewInst
1：禁止该异常级别的 ViewInst

位 15:12 保留，必须保持复位值。

位 11 **TRCERR**：跟踪系统错误异常 (Tracing of system error exception)
选择是否总是跟踪系统错误异常。

0：仅当指令或异常恰好发生在系统错误异常被跟踪之前，才跟踪系统错误异常
1：总是跟踪系统错误异常，不受 ViewInst 的值限制

位 10 **TRCRESET**：跟踪复位异常 (Tracing of reset exception)
选择是否总是跟踪复位异常。

0：仅当指令异常恰好发生在复位异常被跟踪之前，才跟踪复位异常
1：总是跟踪复位异常，不受 ViewInst 的值限制

位 9 **SSSTATUS**：启/停逻辑的当前状态 (Current status of the start/stop logic)
0：停止状态
1：已启动状态

位 8 保留，必须保持复位值。

位 7 **TYPE**：资源类型 (Resource type)
0：单选资源
1：布尔组合资源对

位 6:4 保留，必须保持复位值。

位 3:0 **SEL[3:0]**：资源/布尔组合资源对 (Resource / Boolean combined resource pair)
当 TYPE 为 0 时，从位 [3:0] 定义的 0-15 中选择一个单选资源
当 TYPE 为 1 时，从位 [2:0] 定义的 0-7 中选择一个布尔组合资源对

ETM ViewInst 启/停控制寄存器 (M7_ETM_VISSCTL)

ETM ViewInst start/stop control register

偏移地址：0x088

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	STOP[7:0]							
								rw							

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	START[7:0]							
									rw						

位 31:24 保留，必须保持复位值。

位 23:16 **STOP[7:0]**：选择单个地址比较器以停止跟踪 (Selector of single address comparators to stop trace)
定义单个地址比较器，以通过 ViewInst 启/停控制来停止跟踪。设计的每一个单个地址比较器使用一个位。

位 15:8 保留，必须保持复位值。

位 7:0 **START[7:0]**：选择单个地址比较器以启动跟踪 (Selector of single address comparators to start trace)
定义单个地址比较器，以通过 ViewInst 启/停控制来启动跟踪。设计的每一个单个地址比较器使用一个位。

ETM ViewInst 启/停处理器比较器控制寄存器 (M7_ETM_VIPCSSCTL)

ETM ViewInst start/stop processor comparator control register

偏移地址：0x08C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	STOP[3:0]			
												rw			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	START[3:0]			
												rw			

位 31:20 保留，必须保持复位值。

位 19:16 **STOP[3:0]**：选择处理器比较器输入以停止跟踪 (Selector of processor comparator input to stop trace)

选择由哪些处理器比较器输入与 ViewInst 启/停控制一起用于停止跟踪。每一个处理器比较器输入使用一个位。

位 15:4 保留，必须保持复位值。

位 3:0 **START[3:0]**：选择处理器比较器输入以启动跟踪 (Selector of processor comparator input to start trace)

选择由哪些处理器比较器输入与 ViewInst 启/停控制一起用于启动跟踪。每一个处理器比较器输入使用一个位。

ETM 计数器重载值寄存器 (M7_ETM_CNTRLDV)

ETM counter reload value register

偏移地址：0x140

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VALUE[15:0]															
rw															

位 31:16 保留，必须保持复位值。

位 15:0 **VALUE[15:0]**：计数器重载值 (Counter reload value)

每当发生重载事件时，该值就被加载到计数器中。

ETM ID 寄存器 8 (M7_ETM_IDR8)

ETM ID register 8

偏移地址：0x180

复位值：0x0000 0002

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
MAXSPEC[31:16]															
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MAXSPEC[15:0]															
r															

位 31:0 **MAXSPEC[31:0]**：最大推测深度 (Maximum speculation depth)

指示指令跟踪流的最大推测深度。即在任一时间未提交到跟踪流中的 P0 元素的最大数量。

0x2：最大跟踪推测深度为 2

ETM ID 寄存器 9 (M7_ETM_IDR9)

ETM ID register 9

偏移地址：0x184

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NUMP0KEY[31:16]															
r															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NUMP0KEY[15:0]															
r															

位 31:0 **NUMP0KEY[31:0]**：使用的 P0 右手键数量 (Number of P0 right-hand keys used)

0x0：在指令跟踪或配置中没有使用 P0 键

ETM ID 寄存器 10 (M7_ETM_IDR10)

ETM ID register 10

偏移地址：0x188

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NUMP1KEY[31:16]															
r															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NUMP1KEY[15:0]															
r															

位 31:0 **NUMP1KEY[31:0]**：P1 右手键总数 (Total number of P1 right-hand keys)

指示 P1 右手键总数，包含正常键和特殊键。

0x0：在指令跟踪或配置中没有使用 P1 键

ETM ID 寄存器 11 (M7_ETM_IDR11)

ETM ID register 11

偏移地址：0x18C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NUMP1SPC[31:16]															
r															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NUMP1SPC[15:0]															
r															

位 31:0 **NUMP1SPC[31:0]**：使用的特殊 P1 右手键总数 (Total number of special P1 right-hand keys used)

0x0：没有使用特殊 P1 右手键

ETM ID 寄存器 12 (M7_ETM_IDR12)

ETM ID register 12

偏移地址：0x190

复位值：0x0000 0001

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NUMCONDKEY[31:16]															
r															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NUMCONDKEY[15:0]															
r															

位 31:0 **NUMCONDKEY[31:0]**：

指示条件指令右手键总数，包含正常键和特殊键。

0x1：设计了一个条件指令右手键

ETM ID 寄存器 13 (M7_ETM_IDR13)

ETM ID register 13

偏移地址：0x194

复位值：0x0000 0001

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NUMCONDSPC[31:16]															
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NUMCONDSPC[15:0]															
r															

位 31:0 **NUMCONDSPC[31:0]**：特殊条件指令右手键的数量 (Number of special conditional instruction right-hand keys)

0x0：没有设计特殊条件指令右手键

ETM 针对特定设计方案的寄存器 0 (M7_ETM_IMSPEC0)

ETM implementation specific register 0

偏移地址：0x1C0

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SUPPORT[3:0]			
												r			

位 31:4 保留，必须保持复位值。

位 3:0 **SUPPORT[3:0]**：支持针对特定设计方案的扩展 (Support for implementation specific extensions)

0x0：不支持针对特定设计方案的扩展

ETM ID 寄存器 0 (M7_ETM_IDR0)

ETM ID register 0

偏移地址：0x1E0

复位值：0x0C00 1EE1

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	COMMOPT	TSSIZE[4:0]					Res.	Res.	Res.	Res.	Res.	Res.	Res.	QSUPP[1]
		r	r												r

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
QSUPP[0]	Res.	CONDTYPE[1:0]		NUMEVENT[1:0]		RETSTACK	Res.	TRCCCI	TRCCOND	TRCBB	Res.	Res.	Res.	Res.	Res.
r		r		r		r		r	r	r					

位 31:30 保留，必须保持复位值。

位 29 **COMMOPT**：某些数据包提交字段的含义 (Meaning of the commit field in some packets)
0：提交模式 0

位 28:24 **TSSIZE[4:0]**：全局时间戳大小 (Global timestamp size)
0x08：设计了最大 64 位全局时间戳

位 23:17 保留，必须保持复位值。

位 16:15 **QSUPP[1:0]**：Q 元素支持 (Q element support)
0x0：不支持 Q 元素

位 14 保留，必须保持复位值。

位 13:12 **CONDTYPE[1:0]**：条件结果跟踪方式 (Way of conditional result tracing)
0x1：APSR 条件标记值跟踪方式

位 11:10 **NUMEVENT[1:0]**：跟踪中支持的事件数量 (Number of events supported in the trace)
0x1：支持两个事件（适用于纯指令配置）

位 9 **RETSTACK**：返回堆栈支持 (Return stack support)
1：支持两条目返回堆栈

位 8 保留，必须保持复位值。

位 7 **TRCCCI**：支持在指令跟踪中进行循环计数 (Support for cycle counting in the instruction trace)
1：设计了在指令跟踪中进行循环计数

位 6 **TRCCOND**：支持条件指令跟踪 (Support for conditional instruction tracing)
1：设计了条件指令跟踪

位 5 **TRCBB**：支持分支广播跟踪 (Support for branch broadcast tracing)
1：设计了分支广播跟踪

位 4:0 保留，必须保持复位值。

ETM ID 寄存器 1 (M7_ETM_IDR1)

ETM ID register 1

偏移地址：0x1E4

复位值：0x4100 F401

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
DESIGNER[7:0]								Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
r															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	TRCARCHMAJ[3:0]				TRCARCHMIN[3:0]				REVISION[3:0]			
				r				r				r			

位 31:24 **DESIGNER[7:0]**：跟踪单元设计人员标识 (Trace unit designer Identity)
0x41：ARM®

位 23:12 保留，必须保持复位值。

位 11:8 **TRCARCHMAJ[3:0]**：主跟踪单元架构版本号 (Major trace unit architecture version number)
0x4：ETM v4

位 7:4 **TRCARCHMIN[3:0]**：子跟踪单元架构版本号 (Minor trace unit architecture version number)
0x0：Minor 版本 0

位 3:0 **REVISION[3:0]**：设计版本号 (Implementation revision number)
0x1：版本 1

ETM ID 寄存器 2 (M7_ETM_IDR2)

ETM ID register 2

偏移地址：0x1E8

复位值：0x0000 0004

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	CCSIZE[3:0]				DVSIZE[4:0]					DASIZE[4:1]			
			r				r					r			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DASIZE[0]	VMIDSIZE[4:0]					CIDSIZE[4:0]					IASIZE[4:0]				
r	r					r					r				

位 31:29 保留，必须保持复位值。

位 28:25 **CCSIZE[3:0]**：循环计数器大小 (Cycle counter size)
指示循环计数器的大小（以位为单位）减去 12。
0x0：循环计数器为 12 位

位 24:20 **DVSIZE[4:0]**：以字节为单位的数据值大小 (Data value size in bytes)
0x0：在纯指令配置中不支持数据值大小

位 19:15 **DASIZE[4:0]**：以字节为单位的数据地址大小 (Data address size in bytes)
0x0：在纯指令配置中不支持数据地址大小

位 14:10 **VMIDSIZE[4:0]**：虚拟机 ID 大小 (Virtual machine ID size)
0x0：未设计虚拟机 ID 跟踪

位 9:5 **CIDSIZE[4:0]**：上下文 ID 大小 (Context ID size)
0x0：未设计上下文 ID 跟踪

位 4:0 **IASIZE[4:0]**：指令地址大小 (Instruction address size)
0x4：32 位最大地址大小

ETM ID 寄存器 3 (M7_ETM_IDR3)

ETM ID register 3

偏移地址：0x1EC

复位值：0x0509 0004

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NOOVERFLOW	NUMPROC[2:0]			SYSSTALL	STALLCTL	SYNCPR	TRCERR	Res.	Res.	Res.	Res.	EXLEVEL_S[3:0]			
r	r			r	r	r	r					r			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	CCITMIN[11:0]											
				r											

位 31 **NOOVERFLOW**：支持 NOOVERFLOW (Support of NOOVERFLOW)
指示是否设计了跟踪停止控制的 NOOVERFLOW。
0：未设计

位 30:28 **NUMPROC[2:0]**：可用于跟踪的处理器数量 (Number of processors available for tracing)
0x0：仅可跟踪一个处理器

位 27 **SYSSTALL**：系统支持处理器的停止控制 (System support for stall control of the processor)
0：不支持

位 26 **STALLCTL**：停止控制支持 (Stall control support)
1：设计了跟踪停止控制 (TRCSTALLCTLR)。

位 25 **SYNCPR**：跟踪同步周期支持 (Trace synchronization period support)
0：对于纯指令跟踪配置，TRCSYNCPR 为只读值，跟踪同步周期为固定值

位 24 **TRCERR**：支持 TRCVICTLR.TRCERR (Support of TRCVICTLR.TRCERR)
指示是否设计了 TRCVICTLR.TRCERR。
0x4：32 位最大地址大小

位 23:20 保留，必须保持复位值。

位 19:16 **EXLEVEL_S[3:0]**：支持特权级别 (Support of privilege levels)
设计了特权级别，每个级别使用一个位。
0x9：设计了特权级别线程和处理程序

位 15:12 保留，必须保持复位值。

位 11:0 **CCITMIN[11:0]**：指令跟踪循环计数最小阈值 (Instruction trace cycle counting minimum threshold)
0x4：最小阈值为 4 个指令跟踪周期

ETM ID 寄存器 4 (M7_ETM_IDR4)

ETM ID register 4

偏移地址：0x1F0

复位值：0x0001 4000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NUMVMIDC[3:0]				NUMCIDC[3:0]				NUMSSCC[3:0]				NUMRSPAIR[3:0]			
r				r				r				r			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NUMPC[3:0]				Res.	Res.	Res.	SUPPADC	NUMDVC[3:0]				NUMACPAIRS[3:0]			
r							r	r				r			

位 31:28 **NUMVMIDC[3:0]**：设计的虚拟机 ID 比较器数量 (Number of Virtual Machine ID comparators implemented)
0x0：无

位 27:24 **NUMCIDC[3:0]**：设计的上下文 ID 比较器数量 (Number of Context ID comparators implemented)
0x0：无

位 23:20 **NUMSSCC[3:0]**：设计的单发比较器控件数量 (Number of single-shot comparator controls implemented)
0x0：无

位 19:16 **NUMRSPAIR[3:0]**：设计的资源选择对数量 (Number of resource selection pairs implemented)
0x1：无

位 15:12 **NUMPC[3:0]**：设计的处理器比较器输入数量 (Number of processor comparator inputs implemented)
0x4：四个

位 11:9 保留，必须保持复位值。

位 8 **SUPPADC**：支持数据地址比较 (Support of data address comparisons)
0：未设计

位 7:4 **NUMDVC[3:0]**：设计的数据值比较器数量 (Number of data value comparators implemented)
0x0：无

位 3:0 **NUMACPAIRS[3:0]**：设计的地址比较器对数量 (Number of address comparator pairs implemented)。
0x0：无

ETM ID 寄存器 5 (M7_ETM_IDR5)

ETM ID register 5

偏移地址：0x1F4

复位值：0x90C7 0402

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
REDFUNCNTR	NUMCNTR[2:0]			NUMSEQSTATE[2:0]			Res.	LPOVERRIDE	ATBTRIG	TRACEIDSIZE[5:0]					
r	r			r				r	r	r					

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	NUMEXTINSEL[2:0]			NUMEXTIN[8:0]								
				r			r								

位 31 **REDFUNCNTR**：支持简化功能计数器 (Support of reduced function counter)
1：已设计

位 30:28 **NUMCNTR[2:0]**：设计的计数器数量 (Number of counters implemented)
0x1：设计了一个计数器

位 27:25 **NUMSEQSTATE[2:0]**：设计的定序器状态数量 (Number of sequencer states implemented)
0x0：无

位 24 保留，必须保持复位值。

位 23 **LPOVERRIDE**：支持低功耗状态覆盖 (Support of low-power state override)
1：已设计

位 22 **ATBTRIG**：支持 ATB 触发 (Support of ATB trigger)
1：已设计

位 21:16 **TRACEIDSIZE[5:0]**：跟踪 ID 的位数 (Number of bits of trace ID)
0x07：设计了七位跟踪 ID。

位 15:12 保留，必须保持复位值。

位 11:9 **NUMEXTINSEL[2:0]**：设计的外部输入选择器数量 (Number of external input selectors implemented)
0x2：设计了两个外部输入选择器

位 8:0 **NUMEXTIN[8:0]**：设计的外部输入数量 (Number of external inputs implemented)
0x2：设计了两个外部输入

ETM 资源选择寄存器 2 (M7_ETM_RSCTL2)

ETM resource selection register 2

偏移地址：0x208

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PAIRINV	INV	Res.	GROUP[2:0]		
										rw	rw		rw		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SELECT[7:0]							
								rw							

位 31:22 保留，必须保持复位值。

位 21 **PAIRINV**：反转组合资源对结果 (Inversion of result of a combined pair of resources)
0：未反转
1：已反转

位 20 **INV**：反转所选资源 (Inversion of the selected resources)
0：未反转
1：已反转

位 19 保留，必须保持复位值。

位 18:16 **GROUP[2:0]**：选择一组资源 (Selects a group of resources)

位 15:8 保留，必须保持复位值。

位 7:0 **SELECT[7:0]**：从所需组中选择资源 (Selector of resources from desired group)
从所需组中选择一个或多个资源。每一个资源使用一个位。

ETM 资源选择寄存器 3 (M7_ETM_RSCTL3)

ETM resource selection register 3

偏移地址：0x20C

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	INV	Res.	GROUP[2:0]		
											rw		rw		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SELECT[7:0]							
								rw							

位 31:21 保留，必须保持复位值。

位 20 **INV**：反转所选资源 (Inversion of the selected resources)
0：未反转
1：已反转

位 19 保留，必须保持复位值。

位 18:16 **GROUP[2:0]**：选择一组资源 (Selects a group of resources)

位 15:8 保留，必须保持复位值。

位 7:0 **SELECT[7:0]**：从所需组中选择资源 (Selector of resources from desired group)
从所需组中选择一个或多个资源。每一个资源使用一个位。

ETM 单发比较器控制寄存器 0 (M7_ETM_SSCC0)

ETM single-shot comparator control register 0

偏移地址：0x280

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	RST	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
							rw								

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

位 31:25 保留，必须保持复位值。

位 24 **RST**：单发比较器资源复位使能 (Single-shot comparator resource reset enable)
使能在发生比较器匹配时复位单发比较器资源，以使能检测另一个比较器匹配。
0：禁止
1：使能了复位，可发生多个匹配

位 23:0 保留，必须保持复位值。

ETM 单发比较器状态寄存器 0 (M7_ETM_SSCS0)

ETM single-shot comparator status register 0

偏移地址：0x2A0

复位值：0x0000 0001

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
STATUS	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
rw															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DV	DA	INST
													r	r	r

位 31 **STATUS**：单发状态 (Single-shot status)

指示是否有任何所选比较器发生了匹配。如果 SSCC0.RST 设置为 0，则 STATUS 位必须写入 0，以使能单发比较器控制。

0：没有发生匹配

1：至少发生一次匹配。

位 30:3 保留，必须保持复位值。

位 2 **DV**：数据值比较器支持 (Data value comparator support)

0：不支持单发数据值比较

位 1 **DA**：数据地址比较器支持 (Data address comparator support)

0：不支持单发数据地址比较

位 0 **INST**：指令地址比较器支持 (Instruction address comparator support)

1：支持单发指令地址比较

ETM 单发处理器比较器输入控制寄存器 (M7_ETM_SSPCIC0)

ETM single-shot processor comparator input control register

偏移地址：0x2C0

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PC[7:0]							
								rw							

位 31:8 保留，必须保持复位值。

位 7:0 **PC[7:0]**：针对单发控制的比较器输入选择器 (Comparator input selector for single-shot control)

为单发控制选择一个或多个处理器比较器输入。每一个处理器比较器输入使用一个位。

ETM 掉电控制寄存器 (M7_ETM_PDC)

ETM power-down control register

偏移地址：0x2C0

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PU	Res.	Res.	Res.
												r			

位 31:4 保留，必须保持复位值。

位 3 **PU**：上电请求 (Power up request)
请求保持 ETM 电源和对跟踪寄存器的访问。

0：未请求保持电源
1：请求了保持电源

位 2:0 保留，必须保持复位值。

ETM 掉电状态寄存器 (M7_ETM_PDS)

ETM power-down status register

偏移地址：0x2C0

复位值：0x0000 0003

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	STICKYPD	POWER
														r	r

位 31:2 保留，必须保持复位值。

位 1 **STICKYPD**：粘滞掉电状态 (Sticky power-down state)
当移除 ETM 寄存器的电源时，该位置 1，指示编程状态已丢失。读取 TRCPDSR 后，该位清零。

0：自上一次读取 PDS 寄存器以来，跟踪寄存器电源保持不间断
1：自上一次读取 PDS 寄存器以来，跟踪寄存器电源发生间断

位 0 **POWER**：ETM 上电 (ETM powered up)
1：ETM 已上电，可以访问所有寄存器

将 ETM 声明标记置位寄存器 (M7_ETM_CLAIMSET)

ETM claim tag set register

偏移地址：0xFA0

复位值：0x0000 000F

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CLAIMSET[3:0]			
												rw			

位 31:4 保留，必须保持复位值。

位 3:0 **CLAIMSET[3:0]**：置位声明标记位 (Set claim tag bits)

写：
0000：无影响
xxx1：将位 0 置 1
xx1x：将位 1 置 1
x1xx：将位 2 置 1
1xxx：将位 3 置 1

读：
0xF：表示声明标记使用四个位

ETM 声明标记清零寄存器 (M7_ETM_CLAIMCLR)

ETM claim tag clear register

偏移地址：0xFA4

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CLAIMCLR[3:0]			
												rw			

位 31:4 保留，必须保持复位值。

位 3:0 **CLAIMCLR[3:0]**：复位声明标记位 (Reset claim tag bits)

写：
0000：无影响
xxx1：将位 0 清零
xx1x：将位 1 清零
x1xx：将位 2 清零
1xxx：将位 3 清零

读操作：返回声明标记的当前值

ETM 锁定访问寄存器 (M7_ETM_LAR)

ETM lock access register

偏移地址：0xFB0

复位值：N/A

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ACCESS_W[31:16]															
w															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ACCESS_W[15:0]															
w															

位 31:0 **ACCESS_W[31:0]**：ETM 寄存器写访问 (ETM register write access)
使能处理器内核写访问某些 ETM 寄存器（调试器无需解锁组件）

0xC5ACCE55：使能写访问
其他值：禁止写访问

ETM 锁定状态寄存器 (M7_ETM_LSR)

ETM lock access register

偏移地址：0xFB4

复位值：0x0000 0003

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LOCK TYPE	LOCK GRANT	LOCK EXIST
													r	r	r

位 31:3 保留，必须保持复位值。

位 2 **LOCKTYPE**：M7_ETM_LAR 寄存器大小 (Size of the M7_ETM_LAR register)
0：32 位

位 1 **LOCKGRANT**：当前锁定状态 (Current status of lock)
当外部调试器读取该位时，该位始终返回零。

0：允许写访问
1：阻止写访问。仅允许读访问。

位 0 **LOCKEXIST**：存在锁定控制机制 (Existence of lock control mechanism)
该位表示是否存在锁定控制机制。当外部调试器读取该位时，该位始终返回零。

0：不存在锁定控制机制
1：存在锁定控制机制

ETM 认证状态寄存器 (M7_ETM_AUTHSTAT)

ETM authentication status register

偏移地址：0xFB8

复位值：0x0000 000A

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SNID[1:0]		SID[1:0]		NSNID[1:0]		NSID[1:0]	
								r		r		r		r	

位 31:8 保留，必须保持复位值。

位 7:6 **SNID[1:0]**：安全非侵入式调试的安全等级 (Security level for secure non-invasive debug)

0x0：未设计

位 5:4 **SID[1:0]**：安全侵入式调试的安全等级 (Security level for secure invasive debug)

0x0：未设计

位 3:2 **NSNID[1:0]**：非安全非侵入式调试的安全等级 (Security level for non-secure non-invasive debug)

0x2：禁止

0x3：使能

位 1:0 **NSID[1:0]**：非安全侵入式调试的安全等级 (Security level for non-secure invasive debug)

0x2：禁止

0x3：使能

ETM CoreSight 器件架构寄存器 (M7_ETM_DEVARCH)

ETM CoreSight device architecture register

偏移地址：0xFBC

复位值：0x4770 4A13

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ARCHITECT[10:0]											PRESENT	REVISION[3:0]			
r											r	r			

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ARCHID[15:0]															
r															

位 31:21 **ARCHITECT[10:0]**：组件架构 (Component architect)

0x23B：ARM®

位 20 **PRESENT**：指示有无该寄存器 (Indicates the presence of this register)

1：有

位 19:16 **REVISION[3:0]**：架构版本 (Architecture revision)
0x0：版本 0

位 15:0 **ARCHID[15:0]**：架构 ID (Architecture ID)
0x4A13：ETMv4 组件

ETM CoreSight 器件标识寄存器 (M7_ETM_DEVID)

ETM CoreSight device identity register

偏移地址：0xFC8

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

位 31:0 保留，必须保持复位值。

ETM CoreSight 器件类型标识寄存器 (M7_ETM_DEVTYPE)

ETM CoreSight device type identity register

偏移地址：0xFCC

复位值：0x0000 0013

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SUBTYPE[3:0]				MAJORTYPE[3:0]			
								r				r			

位 31:8 保留，必须保持复位值。

位 7:4 **SUBTYPE**：器件子类型标识符 (Device sub-type identifier)
0x1：处理器跟踪

位 3:0 **MAJORTYPE**：器件主类型标识符 (Device main type identifier)
0x3：跟踪源

ETM CoreSight 外设标识寄存器 4 (M7_ETM_PIDR4)

ETM CoreSight peripheral identity register 4

偏移地址：0xFD0

复位值：0x0000 0004

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	4KCOUNT[3:0]				JEP106CON[3:0]			
								r				r			

位 31:8 保留，必须保持复位值。

位 7:4 **4KCOUNT[3:0]**：寄存器文件大小 (Register file size)

0x0：寄存器文件占用单个 4 KB 区域

位 3:0 **JEP106CON[3:0]**：JEP106 连续代码 (JEP106 continuation code)

0x4：ARM® JEDEC 代码

ETM CoreSight 外设标识寄存器 0 (M7_ETM_PIDR0)

ETM CoreSight peripheral identity register 0

偏移地址：0xFE0

复位值：0x0000 0075

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PARTNUM[7:0]							
								r							

位 31:8 保留，必须保持复位值。

位 7:0 **PARTNUM[7:0]**：产品编号字段 (Part number field)，位 [7:0]

0x75：ETM 产品编号

ETM CoreSight 外设标识寄存器 1 (M7_ETM_PIDR1)

ETM CoreSight peripheral identity register 1

偏移地址：0xFE4

复位值：0x0000 00B9

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	JEP106ID[3:0]				PARTNUM[11:8]			
								r				r			

位 31:8 保留，必须保持复位值。

位 7:4 **JEP106ID[3:0]**：JEP106 标识代码字段 (JEP106 identity code field)，位 [3:0]
0xB：ARM® JEDEC 代码

位 3:0 **PARTNUM[11:8]**：产品编号字段 (Part number field)，位 [11:8]
0x9：ETM 产品编号

ETM CoreSight 外设标识寄存器 2 (M7_ETM_PIDR2)

ETM CoreSight peripheral identity register 2

偏移地址：0xFE8

复位值：0x0000 002B

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REVISION[3:0]				JEDEC	JEP106ID[6:4]		
								r				r	r		

位 31:8 保留，必须保持复位值。

位 7:4 **REVISION[3:0]**：组件版本号 (Component revision number)
0x1：r0p2

位 3 **JEDEC**：JEDEC 分配的值 (JEDEC assigned value)
1：JEDEC 指定的设计人员 ID

位 2:0 **JEP106ID[6:4]**：JEP106 标识代码字段 (JEP106 identity code field)，位 [6:4]
0x3：ARM® JEDEC 代码

ETM CoreSight 外设标识寄存器 3 (M7_ETM_PIDR3)

ETM CoreSight peripheral identity register 3

偏移地址：0xFEC

复位值：0x0000 0000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REVAND[3:0]				CMOD[3:0]			
								r				r			

位 31:8 保留，必须保持复位值。

位 7:4 **REVAND[3:0]**：金属修复版本 (Metal fix version)
0x0：无金属修复

位 3:0 **CMOD[3:0]**：客户修改 (Customer modified)
0x0：无客户修改

ETM CoreSight 组件标识寄存器 0 (M7_ETM_CIDR0)

ETM CoreSight component identity register 0

偏移地址：0xFF0

复位值：0x0000 000D

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[7:0]							
								r							

位 31:8 保留，必须保持复位值。

位 7:0 **PREAMBLE[7:0]**：组件 ID 字段 (Component ID field)，位 [7:0]
0x0D：公共 ID 值

ETM CoreSight 组件标识寄存器 1 (M7_ETM_CIDR1)

ETM CoreSight component identity register 1

偏移地址：0xFF4

复位值：0x0000 0090

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CLASS[3:0]				PREAMBLE[11:8]			
								r				r			

位 31:8 保留，必须保持复位值。

位 7:4 **CLASS[3:0]**：组件 ID 字段 (Component ID field)，位 [15:12]——组件类
0x9：寄存器与 CoreSight 兼容的调试组件

位 3:0 **PREAMBLE[11:8]**：组件 ID 字段 (Component ID field)，位 [11:8]
0x0：公共 ID 值

ETM CoreSight 组件标识寄存器 2 (M7_ETM_CIDR2)

ETM CoreSight component identity register 2

偏移地址：0xFF8

复位值：0x0000 0005

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[19:12]							
								r							

位 31:8 保留，必须保持复位值。

位 7:0 **PREAMBLE[19:12]**：组件 ID 字段 (Component ID field)，位 [23:16]
0x05：公共 ID 值

ETM CoreSight 组件标识寄存器 3 (M7_ETM_CIDR3)

ETM CoreSight component identity register 3

偏移地址：0xFFC

复位值：0x0000 00B1

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[27:20]							
								r							

位 31:8 保留，必须保持复位值。

位 7:0 **PREAMBLE[27:20]**：组件 ID 字段 (Component ID field)，位 [31:24]
0xB1：公共 ID 值

Cortex-M7 ETM 寄存器映射和复位值

ETM 寄存器的地址范围为 0xE0041000 到 0xE0041FFC，调试器可以通过 Cortex-M7 PPB 对其进行访问。

表 584. Cortex-M7 ETM 寄存器映射和复位值

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x004	M7_ETM_PRGCTL	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	EN
	Reset value																																0
0x008	M7_ETM_PROCSEL	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PROCSEL[2:0]		
	Reset value																														0	0	0
0x00C	M7_ETM_STAT	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PMSTABLE	IDLE
	Reset value																															0	0
0x010	M7_ETM_CONFIG	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DV	DA	Res.	Res.	Res.	RS	TS	COND[2:0]			VMID	CID	Res.	CCI	BB	INSTP0[1:0]		Res.
	Reset value															0	0				0	0	0	0	0	0	0		0	0	0	0	1
0x020	M7_ETM_EVENTCTL0	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TYPE1	Res.	Res.	Res.	SEL1[3:0]				TYPE0	Res.	Res.	Res.	SEL0[3:0]			
	Reset value																	0				0	0	0	0	0				0	0	0	0

表 584. Cortex-M7 ETM 寄存器映射和复位值（续）

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x024	M7_ETM _EVENTCTL1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	LPOVERRIDE	ATB	Res.	Res.	Res.	Res.	Res.	Res.	Res.	INSTEN[3:0]			
	Reset value																				0	0								0	0	0	0
0x02C	M7_ETM _STALLCTL	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DSTALL	ISATLL	Res.	Res.	Res.	Res.	LEVEL [1:0]		Res.	Res.
	Reset value																							0	0					0	0		
0x030	M7_ETM_TSCTL	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TYPE	Res.	Res.	Res.	SEL[3:0]			
	Reset value																									0				0	0	0	0
0x034	M7_ETM_SYNCP	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PERIOD[4:0]			
	Reset value																												0	0	0	0	0
0x038	M7_ETM_CCCTL	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	THRESHOLD[11:0]											
	Reset value																					0	0	0	0	0	0	0	0	0	0	0	0
0x040	M7_ETM_TRACEID	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TRACEID[6:0]						
	Reset value																										0	0	0	0	0	0	0
0x080	M7_ETM_VICTL	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	EXLEVEL_S3	Res.	Res.	EXLEVEL_S0	Res.	Res.	Res.	Res.	TRCERR	TRCRESET	SSSTATUS	Res.	TYPE	Res.	Res.	Res.	SEL[3:0]			
	Reset value													0			0					0	0	0		0				0	0	0	0
0x088	M7_ETM_VISSCTL	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	STOP[7:0]								Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	START[7:0]							
	Reset value									0	0	0	0	0	0	0	0									0	0	0	0	0	0	0	0
0x08C	M7_ETM _VIPCSSCTL	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	STOP[3:0]				Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	START[3:0]			
	Reset value													0	0	0	0													0	0	0	0
0x140	M7_ETM _CNTRLDV	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	VALUE[15:0]															
	Reset value																	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x180	M7_ETM_IDR8	MAXSPEC[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
0x184	M7_ETM_IDR9	NUMP0KEY[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x188	M7_ETM_IDR10	NUMP1KEY[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x18C	M7_ETM_IDR11	NUMP1SPC[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x190	M7_ETM_IDR12	NUMCONDKEY[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
0x194	M7_ETM_IDR13	NUMCONDSPC[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0x1C0	M7_ETM_IMSPEC0	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SUPPORT [3:0]			
	Reset value																													0	0	0	0

表 584. Cortex-M7 ETM 寄存器映射和复位值（续）

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x1E0	M7_ETM_IDR0	Res.	Res.	COMMOPT	TSSIZE[4:0]					Res.	Res.	Res.	Res.	Res.	Res.	Res.	QSUPP[1:0]		Res.	CONDTYPE [1:0]		NUMEVENT [1:0]		RETSTACK	Res.	TRCCCI	TRCCOND	TRCBB	Res.	Res.	Res.	Res.	Res.
	Reset value			0	0	1	0	0	0								0	0		0	1	1	1	1		1	1	1					1
0x1E4	M7_ETM_IDR1	DESIGNER[7:0]								Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	TRCARCHMAJ [3:0]				TRCARCHMIN [3:0]			REVISION [3:0]				
	Reset value	0	0	1	0	0	0	0	1									1	1	1	1	0	1	0	0	0	0	0	0	0	0	0	1
0x1E8	M7_ETM_IDR2	Res.	Res.	Res.	CCSIZE[3:0]				DVSIZE[4:0]					DASIZE[4:0]				VMIDSIZE[4:0]				CIDSIZE[4:0]							IASIZE[4:0]			
	Reset value				0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
0x1EC	M7_ETM_IDR3	NOOVERFLOW	NUMPROC[2:0]		SYSSTALL	STALLCTL	SYNCPR	TRCERR		Res.	Res.	Res.	Res.	EXLEVEL_S [3:0]				Res.	Res.	Res.	Res.	CCITMIN[11:0]											
	Reset value	0	0	0	0	0	1	0	1					1	0	0	1					0	0	0	0	0	0	0	0	0	1	0	0
0x1F0	M7_ETM_IDR4	NUMVMIDC [3:0]				NUMCIDC [3:0]				NUMSCC [3:0]				NUMRSPAIR [3:0]				NUMPC [3:0]				Res.	Res.	Res.	SUPPDAC	NUMDVC [3:0]				NUMACPAIRS [3:0]			
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0				0	0	0	0	0	0	0	0	0
0x1F4	M7_ETM_IDR5	REDFUNCNTR	NUMCNTR[2:0]			NUMSEQSTATE [2:0]			Res.	LPOVERRIDE	ATBTRIG	TRACEIDSIZE[5:0]					Res.	Res.	Res.	Res.	NUMEXTINSEL [2:0]			NUMEXTIN[8:0]									
	Reset value	1	0	0	1	0	0	0		1	1	0	0	0	1	1	1					0	1	0	0	0	0	0	0	0	0	1	0
0x208	M7_ETM_RSCTL2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PAIRINV	INV	Res.	GROUP [2:0]			Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SELECT[7:0]							
	Reset value											0	0		0	0	0									0	0	0	0	0	0	0	0
0x20C	M7_ETM_RSCTL3	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PAIRINV	INV	Res.	GROUP [2:0]			Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SELECT[7:0]							
	Reset value											0	0		0	0	0									0	0	0	0	0	0	0	0
0x280	M7_ETM_SSCC0	Res.	Res.	Res.	Res.	Res.	Res.	Res.	RST	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	Reset value								0																								
0x2A0	M7_ETM_SSCS0	STATUS	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	DV	DA	INST
	Reset value	0																												0	0	1	
0x2C0	M7_ETM_SSPCIC0	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PC[7:0]							
	Reset value																									0	0	0	0	0	0	0	0
0x310	M7_ETM_PDC	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.
	Reset value																													0			

表 584. Cortex-M7 ETM 寄存器映射和复位值（续）

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x314	M7_ETM_PDS	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	STICKYPD	POWER
	Reset value																															1	1
0xFA0	M7_ETM _CLAIMSET	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SET[3:0]			
	Reset value																													0	1	0	0
0xFA4	M7_ETM _CLAIMCLR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CLR[3:0]			
	Reset value																													0	0	0	0
0xFB0	M7_ETM_LAR	KEY[31:0]																															
	Reset value																																
0xFB4	M7_ETM_LSR	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	NTT	SLK	SLI
	Reset value																														0	1	1
0xFB8	M7_ETM _AUTHSTAT	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	SNID [1:0]		SID [1:0]		NSNID [1:0]		NSID [1:0]	
	Reset value																									0	0	0	0	0	0	0	0
0xFBC	M7_ETM _DEVARCH	ARCHITECT[10:0]											PRESENT	REVISION [3:0]				ARCHID[15:0]															
	Reset value	0	1	0	0	0	1	1	1	0	1	1	1	0	0	0	0	0	0	1	0	1	0	1	0	0	0	0	1	0	0	1	1
0xFC8	M7_ETM_DEVID	DEVICEID[31:0]																															
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0xFD0	M7_ETM _DEVTYPE	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	SUBTYPE [3:0]				MAJORTYPE [3:0]			
	Reset value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	1
0xFD0	M7_ETM_PIDR4	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	4KCOUNT [3:0]				JEP106CON [3:0]			
	Reset value																									0	0	0	0	0	1	0	0
0xFD4	M7_ETM_PIDR5	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res
	Reset value																																
0xFD8	M7_ETM_PIDR6	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res
	Reset value																																
0xFDC	M7_ETM_PIDR7	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res
	Reset value																																
0xFE0	M7_ETM_PIDR0	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	PARTNUM [7:0]							
	Reset value																									0	1	1	1	0	1	0	1
0xFE4	M7_ETM_PIDR1	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	JEP106ID [3:0]				PARTNUM [11:8]			
	Reset value																									1	0	1	1	1	0	0	1
0xFE8	M7_ETM_PIDR2	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	Res	REVISION [3:0]				JEDEC	JEP106ID [6:4]		
	Reset value																									0	0	0	1	1	0	1	1

表 584. Cortex-M7 ETM 寄存器映射和复位值（续）

偏移	寄存器名称	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0xFEC	M7_ETM_PIDR3	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	REVAND[3:0]				CMOD[3:0]			
	Reset value																									0	0	0	0	0	0	0	0
0xFF0	M7_ETM_CIDR0	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[7:0]							
	Reset value																									0	0	0	0	1	1	0	1
0xFF4	M7_ETM_CIDR1	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	CLASS[3:0]				PREAMBLE [11:8]			
	Reset value																									1	0	0	1	0	0	0	0
0xFF8	M7_ETM_CIDR2	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[19:12]							
	Reset value																									0	0	0	0	0	1	0	1
0xFFC	M7_ETM_CIDR3	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	Res.	PREAMBLE[27:20]							
	Reset value																									1	0	1	1	0	0	0	1

60.6.6 Cortex-M7 交叉触发接口 (CTI)

请参见*第 60.5.3 节*。

60.7 针对调试基础结构的参考文献

1. IHI 0031C (ID080813)——《ARM® 调试接口架构规范 ADIv5.0 到 ADIv5.2》，发布版本 C
2. DDI 0480F (ID100313)——《ARM® CoreSight™ SoC-400 r3p2 技术参考手册》，发布版本 G
3. DDI 0461B (ID010111)——《ARM® CoreSight™ 跟踪存储器控制器 r0p1 技术参考手册》，发布版本 B
4. DDI 0314H——《ARM® CoreSight™ 组件技术参考手册》，发布版本 H
5. DDI 0403D (ID100710)——《ARM®v7-M 架构参考手册》，发布版本 E.b
6. DDI 0494-2a (ID062813)——《ARM® CoreSight™ ETM™-M7 r0p1 技术参考手册》，发布版本 D

61 设备电子签名

电子签名存储在 Flash 区。 可以使用 JTAG/SWD 或 CPU 对其进行读取。它包含出厂前编程的标识数据，这些标识数据允许用户固件或其它外部设备将其接口与 STM32H7x3 微控制器的特性自动匹配。

61.1 唯一设备 ID 寄存器（96 位）

Unique device ID register

唯一设备标识符最适合：

- 用作序列号（例如 USB 字符串序列号或其它终端应用程序）
- 在对内部 Flash 进行编程前将唯一 ID 与软件加密原语和协议结合使用时用作安全密钥以提高 Flash 中代码的安全性
- 激活安全自举过程等

96 位的唯一设备标识符提供了一个对于任何设备和任何上下文都唯一的参考号码。 用户永远不能改变这些位。

96 位的唯一设备标识符也可以以单字节/半字/字等不同方式读取，然后使用自定义算法连接起来。

基址：0x1FF1 E800

偏移地址：0x00

只读 = 0xXXXX XXXX，其中 X 是出厂前编程的

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
U_ID(31:0)																															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:0 **U_ID(31:0)**：31:0 唯一 ID 位 (31:0 unique ID bits)

偏移地址：0x04

只读 = 0xXXXX XXXX，其中 X 是出厂前编程的

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
U_ID(63:48)															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
U_ID(47:32)															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:0 **U_ID(63:32)**：63:32 唯一 ID 位 (63:32 unique ID bits)

偏移地址：0x08

只读 = 0xXXXX XXXX，其中 X 是出厂前编程的

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
U_ID(95:80)															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
U_ID(79:64)															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 31:0 **U_ID(95:64)**：95:64 唯一 ID 位 (95:64 Unique ID bits)

61.2 Flash 大小

Flash size

基址：0x1FF1 E880

偏移地址：0x00

只读 = 0xXXXX，其中 X 是出厂前编程的

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
F_SIZE															
r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r

位 15:0 F_ID(15:0)：Flash 大小 (Flash memory size)

此处的位域指示以 KB 表示的设备 Flash 大小。

例如，0x0400 对应于 1024 KB。

61.3 封装数据寄存器

Package data register

有关封装标识，请参见 SYSCFG 封装寄存器 (SYSCFG_PKGR)。应首先在 RCC_APB4ENR 寄存器中使能 SYSCFG 时钟。

版本历史

表 585. 文档版本历史

日期	版本	变更
2016 年 12 月 12 日	1	初始版本。
2017 年 5 月 26 日	2	*第 2 章*重命名为存储器和总线架构。 更新了*内部 Bootloader*一节。 ***第 3 节：嵌入式 Flash (FLASH)*** 更新了*Flash 扇区擦除*一节、*标准 Flash 存储区擦除*一节、*使用自动取消保护来执行 Flash 存储区擦除操作*一节和*Flash 批量擦除*一节。 增加了*自动保护取消时的Flash 批量擦除*一节。 更新了*表 16：FLASH 寄存器映射和复位值*： – FLASH_BOOT7_CURR/PRG（偏移地址 = 0x140/0x144）仅适用于 RM0399。 – 增添了 FLASH_CRCEADD2R（偏移 地址 = 0x158） – 增添了 FLASH_CRCEADD2R（偏移地址 = 0x15C） ***第 6 节：电源控制 (PWR)*** 更新了*第 6.3.1 节：PWR 引脚和内部信号*。 更新了*第 6.4 节：电源*中的 V_{BAT}。 更新了*图 13：电源概述*和*图 15：由稳压器提供 V_{CORE} 时的器件启动情形*。 删除了*第 6.5 节：电源监控*中的 VCORE。 更新了*图 24：V_{CORE} 电压调节与系统电源模式*和*图 25：电源控制模式详细状态图*。 ***第 8 节：复位和时钟控制 (RCC)*** 所有时钟源选择位的名称均从 XXSCR 更改为 XXSEL。 RC48 重命名为 HSI48。 在 RCC_BDCR 中 VSWRST 位重命名为 BDRST 并修改了位说明。 在 RCC_AHB2ENR 中 CAMITFEN 重命名为 DCMIEN； 在 RCC_AHB2LPENR 中 CAMITFLPEN 重命名为 DCMILPEN。 在 RCC_AHB3LPENR 中 FLITFLPEN 位重命名为 FLASHLPEN。 在 RCC_APB1LLPENR 中 HDMICECLPEN 重命名为 CECLPEN。 在整个文档范围内，更新了外设内核时钟名称。 更新了 *图 35：系统复位电路*，其中删除了 V_{DDA}。 更新了*表 50：内核时钟分配概述*中 ADC1、2、3 的最大频率。 删除了*专用于控制和数据传输的外设*一节中用作 USBxOTG 时钟的 lsi_ck。

表 585. 文档版本历史（续）

日期	版本	变更
2017 年 5 月 26 日	2（续）	***第 8 节：复位和时钟控制 (RCC)***（续） 在*第 8.7.31 节：RCC APB1 外设复位寄存器 (RCC_APB1LRSTR)*、*第 8.7.43 节：RCC APB1 时钟寄存器 (RCC_APB1LENR)* 和*第 8.7.52 节：RCC APB1 低位睡眠时钟寄存器 (RCC_APB1LLPENR)* 中，将 USART7RST/EN/LPEN 位重命名为 UART7RST/EN/LPEN，将 USART8RST/EN/LPEN 位重命名为 UART8RST/EN/LPEN。 ***第 10 节：硬件信号量 (HSEM)*** 在整个文档范围内，将 pclk 重命名为 hsem_hclk。 更新了*第 10.4.1 节：HSEM 寄存器 (HSEM_R0 - HSEM_R31)* 中的 COREID 位说明。 ***第 11 节：通用 I/O (GPIO)*** *表 83：GPIO 寄存器映射和复位值*： – 更新了 GPIOx_MODER 复位值 – 更改了 GPIOx_AFRH 的 A 到 K 的索引。 – 增添了 GPIOC..K_PUPDR ***第 12 节：系统配置控制器 (SYSCFG)*** 更新了 SYSCFG_UR3 寄存器。 ***第 13 节：块互连*** 在*表 91：DMAMUX1、DMA1 和 DMA2 连接*中，将 dac1_dma 和 dac2_dma 分别重命名为 dac_ch1_dma 和 dac_ch2_dma。 更新了*表 87：外设互连矩阵详细信息*和*表 88：EXTI 唤醒输入*中的多个源和目标信号。 ***第 17 节：DMA 请求复用器 (DMAMUX)*** 更新了*表 109：DMAMUX1：复用器输入到资源的分配*到*表 111：DMAMUX1：同步输入到资源的分配*中的资源。 ***第 19 节：嵌套向量中断控制器*** 在*表 129：NVIC* 中增添了 LCD-TFT 中断（ltdc_it 和 ltdc_err_it）。 ***扩展中断和事件控制器 (EXTI)*** 在*表 132：EXTI 事件输入映射*中，针对事件 66 到 73，将 DMA1 替换为 BDMA。

表 585. 文档版本历史（续）

日期	版本	变更
2017 年 5 月 26 日	2（续）	***第 22 节：灵活存储控制器（FMC）*** – 更新了*图 87：FMC 框图*中的内部信号 – HCLK 重命名为 fmc_hclk – KCK_FMC 重命名为 fmc_ker_ck 更新了*第 22.5 节：AXI 接口*，其中增添了 32 位访问。 读 FIFO 深度更改为 6x64 位。更正了 AXI 总线宽度（64 位，而非 32 位）。 针对数据总线和 NBL 位，所有波形均是通用的。 在*SRAM/NOR-Flash 片选时序寄存器 1..4 (FMC_BTR1..4)*一节中，修改了 DATAST 示例并更新了 BURSTURN 说明。在*SRAM/NOR-Flash 写入时序寄存器 1..4 (FMC_BWTR1..4)*一节中：更新了 BURSTURN 说明。更新了*SDRAM 控制寄存器 1，2（FMC_SDCR1，FMC_SDCR2）*一节，其中增添了位域宽度。 在*表 180：FMC 寄存器映射*中增添了缺少的 FMC_SDCR2 位。 ***第 23 节：QuadSPI 接口（QUADSPI）*** 更新了*图 116：QUADSPI 功能框图（双闪存模式禁止）*和*图 117：QUADSPI 功能框图（双闪存模式使能）*中的内部信号。增加了*第 23.3.2 节：QUADSPI 引脚和内部信号*。在*第 23.3.7 节：QUADSPI 内存映射模式*中，修改了 Cortex CPU 访问时的错误类型。 增添了*第 23.3.8 节：QUADSPI 自由运行时钟模式*并在*第 23.5.6 节：QUADSPI 通信配置寄存器 (QUADSPI_CCR)* 中增添了 FRCM 位。更新了*第 23.5.1 节：QUADSPI 控制寄存器 (QUADSPI_CR)*。在*表 184：QUADSPI 寄存器映射和复位值*中，将 DMAEN 位更改为为 QUADSPI_CR 寄存器保留。 ***第 24 节：延迟模块 (DLYB)*** 在*图 124：DLYB 框图*中增添了内部信号并增添了*第 24.3.2 节：DLYB 引脚和内部信号*。 ***第 25 节：模数转换器 (ADC)*** 在*第 25.1 节：简介*中，将 ADC 数更改为 3。 在*第 25.3.1 节：ADC 框图*中增添了内部信号。在*图 125：ADC 框图*和*表 189：ADC 输入/输出引脚*中，将 ADCx_IN[19:0] 更改为 ADCx_INP[19:0] 和 ADCx_INN[19:0]。将 ADC_CLK 更改为 adc_ker_ck。 更新了*图 127：ADC1 连接功能*、*图 128：ADC2 连接功能*和*图 129：ADC3 连接功能*，并在图下面增添了注释。更新了*图 176：双重 ADC 框图*(1)。 更新了*双时钟域架构*一节中的注释。 更新了*第 25.3.11 节：通道选择（SQRx、JSQRx）*和*第 25.3.2 节：ADC 引脚和内部信号*。

表 585. 文档版本历史（续）

日期	版本	变更
2017 年 5 月 26 日	2（续）	***第 25 节：模数转换器 (ADC)***（续） 在*第 25.3.13 节：可独立设置各通道采样时间（SMPR1、SMPR2）*和*第 25.3.17 节：时序*中针对 24 MHz 给出了示例，而非 72 MHz。 在*第 25.3.23 节：可编程分辨率 (RES) - 快速转换模式*中删除了 6 位分辨率，并更新了*表 194：TSAR 与分辨率的对应关系*。 更新了*混合型常规/注入同步模式*一节和*常规同步 + 交替触发组合模式*一节中的注释。 增加了*双重 ADC 同步模式下的 DFSDM 模式*一节。 在*双重 ADC 同步模式下的 DFSDM 模式*一节中，将 ADC3_IN18 更改为 ADC3 VINP[18]。 在*第 25.3.34 节：VBAT 电源监测*中，将 ADC3_IN17 更改为 ADC3 VINP[17]。 在*第 25.3.35 节：监测内部参考电压*中，将 ADC3_IN19 更改为 ADC3 VINP[19]。 在*第 25.5.15 节：ADC x 常规数据寄存器 (ADCx_DR)（x=1 到 3）*中，将所有位访问类型更改为“r”。 更新了*表 205：ADC 寄存器映射和复位值（主 ADC 和从 ADC 通用寄存器，偏移=0x300）*。 ***第 26 节：数模转换器 (DAC)*** 更新了*图 195：DAC 通道框图*中的内部信号/向该图中增添了内部信号。增加了*第 26.3.2 节：DAC 引脚和内部信号*。 在整个文档范围内，APB1 时钟和 LSI 时钟被替换为 dac_pclk 和 lsi_ck。 ***第 28 节：比较器 (COMP)*** 在*第 28.7.1 节：比较器状态寄存器 (COMP_SR)*中，将 COMP_IFCR 替换为 COMP_ICFR。 ***第 29 节：运算放大器（OPAMP）*** 将出现的所有 DACx_int 均替换为 dac_outx（x = 1、2）。更新了*第 29.6.1 节：OPAMP1 控制/状态寄存器 (OPAMP1_CSR)* 中的 VP_SEL 位说明。 ***第 31 节：数字摄像头接口 (DCMI)*** *“DCMI 引脚”部分与第 31.4.4 节：DCMI 物理接口*融合在了一起。 在整个文档范围内，外部信号 HSYNC、VSYNC 和 PIXCLK 标准化为 DCMI_HSYNC、DCMI_VSYNC 和 DCIM_PIXCLK。 更新了*图 225：DCMI 框图*和*图 226：顶级框图*。

表 585. 文档版本历史（续）

日期	版本	变更
2017 年 5 月 26 日	2（续）	***第 32 节：LCD-TFT 显示控制器 (LTDC)*** *LCD-TFT 引脚和信号接口*重命名为 *LCD-TFT 引脚和外部信号接口*。 更新了*图 234：LTDC 框图*中的内部信号名称。 在整个文档范围内，ck_axi_d1 重命名为 ltdc_aclk。 更新了*第 32.3.4 节：LTDC 复位和时钟*。 更新了*第 32.7.3 节：LTDC 有效宽度配置寄存器 (LTDC_AWCR)* 中的 CFBP 位域大小。 ***第 33 节：JPEG 编解码器 (JPEG)*** 在*图 239：JPEG 编解码器框图*中增添了内部信号名称，并增添了*第 33.3.2 节：JPEG 内部信号*。 删除了 DMA 功能。 更新了*表 258：JPEG 编解码器寄存器映射和复位值*中的 JPEG_CR、JPEG_SR 和 JPRG_CFR 寄存器。 ***第 37 节：高分辨率定时器 (HRTIM)*** 在整个章节内，将 SCOUT 重命名为 HRTIM_SCOUT。 在所有相关图中，将 Tx 重命名为 HRTIM_CHxy。 更新了*图 275：高分辨率定时器框图*中的内部信号。在*表 282：HRTIM 输入/输出汇总*中添加了 hrtim_in_sync1 和 hrtim_out_sync1。 更新了*术语定义*一节。 在所有相关图和 HRTIM 功能说明中更新了内部信号名称。 在*表 282：HRTIM 输入/输出汇总*中增加了与 hrtim_ker_ck 相关的注释。 修改了*第 37.5.56 节：HRTIM ADC 触发 3 寄存器 (HRTIM_ADC3R)* 中的 ADC3TAPER 位说明。 修改了*第 37.5.57 节：HRTIM ADC 触发 4 寄存器 (HRTIM_ADC4R)* 中的 ADC4TCRST、ADC4TAPER、ADC4TAC2、ADC4EEV6、ADC4MPER 和 DC4MC1 位说明。 ***第 43 节：低功耗定时器 (LPTIM)*** 在*图 516：低功耗定时器框图（LPTIM1 和 LPTIM2）*、*图 517：低功耗定时器框图 (LPTIM3)* 和*图 518：低功耗定时器框图（LPTIM4 和 LPTIM5）*中增添了内部信号。增加了*第 43.4.2 节：LPTIM 引脚和内部信号*。 *表 334：LPTIM1 外部触发连接*到*表 338：LPTIM5 外部触发连接*进行了更新，并移至*第 43.4.3 节：LPTIM 输入和触发映射*下。 *表 339：LPTIM1 输入 1 连接*到*表 343：LPTIM3 输入 1 连接*进行了更新，并移至*第 43.4.3 节：LPTIM 输入和触发映射*下。 更新了*第 43.6.4 节：LPTIM 配置寄存器 (LPTIM_CFGR)* 中的 TRIGSEL 位域说明。

表 585. 文档版本历史（续）

日期	版本	变更
2017 年 5 月 26 日	2（续）	***第 43 节：低功耗定时器 (LPTIM)***（续） 在*第 43.6.5 节：LPTIM 控制寄存器 (LPTIM_CR)* 中增添了 COUNTRST 位的注意事项。 在*第 43.6.9 节：LPTIM 配置寄存器 2 (LPTIM_CFGR2)* 中更新了 IN1SEL 和 IN2SEL 位域说明并增添了注意事项。 在*第 43.6.10 节：LPTIM3 配置寄存器 2 (LPTIM3_CFGR2)* 中增添了注意事项。 ***第 46 节：实时时钟 (RTC)*** 增加了*图 528：RTC 块概览*。 更新了*图 529：详细的 RTC 框图*的标题。 增加了*第 46.3.2 节：RTC 引脚和内部信号*。更新了*表 352：RTC 引脚和内部信号*。 更新了*第 46.3.3 节：RTC 控制的 GPIO*。 更新了*第 46.6.16 节：RTC 入侵配置寄存器 (RTC_TAMPCR)*。 ***第 47 节：内部集成电路 (I2C) 接口*** 更新了*第 47.7.3 节：设备自身地址 1 寄存器 (I2C_OAR1)* 和*第 47.7.4 节：设备自身地址 2 寄存器 (I2C_OAR2)* 中的 OA1[7:1] 和 OA2[7:1] 位说明。 在*第 47.4.14 节：地址匹配时从停止模式唤醒*中，将 HSI16 替换为 HSI 或 CSI 或者内部振荡器。 ***第 48 节：通用同步异步收发器 (USART)*** *第 48.5.14 节：USART 同步模式*：删除了图“*RX 数据建立/保持时间*”，*在图 575：USART 在同步主模式下的数据时钟时序图（M 位 = “00”）*和*图 576：USART 在同步主模式下的数据时钟时序图（M 位 = “01”）*中增添了对同步主模式的参考，并更新了图的内容使其涉及两个 M 位（而非一个）。 更新了*第 48.5.21 节：USART 低功耗管理*中的*图 587：唤醒事件通过验证（唤醒事件 = 地址匹配，禁止 FIFO）*。 ***第 51 节：串行音频接口 (SAI)*** 更新了*第 51.5.1 节：全局配置寄存器 (SAI_GCR)* 中的 SYNCIN 位域说明。

表 585. 文档版本历史（续）

日期	版本	变更
2017 年 5 月 26 日	2（续）	***第 50 节：串行外设接口 (SPI)*** 更新了*单工通信*一节中的注释。 在*PCM 标准*一节中增添了有关 PCM 长帧和短帧定义的注释。增添了*第 50.9.3 节：I2S/PCM 模式下可用的位和位域*说明和*表 390：PCM/I2S 模式下可用的位域*。 更新了*第 50.11.14 节：SPI/I2S 配置寄存器 (SPI_I2SCGFR)* 和*第 50.12 节：SPI 寄存器映射和复位值*中的 WSINV ***第 52 节：SPDIF 接收器接口 (SPDIFRX)*** 在*第 图 659. 节：SPDIFRX 框图*中增添了内部信号并增添了*第 52.3.1 节：SPDIFRX 引脚和内部信号*。 ***第 55 节：安全数字输入/输出多媒体卡接口 (SDMMC)*** – 在整个章节内重命名了内部信号；更新了*图 695：SDMMC 框图*；增添了*第 55.4.2 节：SDMMC 引脚和内部信号* – 增加了*第 55.4.7 节：MDMA 请求生成* – 删除了 SDMMC_VER、SDMMC_ID、SDMMC_SID – 在*表 448：SDMMC 寄存器映射*中更新了 SDMMC_STAR 位 12 和 13。 ***第 56 节：FD 控制器局域网络 （FDCAN）*** 在整个章节中删除了 ASC（异步串行通信）。 分别在 *FDCAN 接收 FIFO 0 配置寄存器 (FDCAN_RXF0C)* 和 *FDCAN 接收 FIFO 1 配置寄存器 (FDCAN_RXF1C)* 中增添了 F0OM 位（位 31）和 F1OM 位（位 31）。 ***第 57 节：USB on-the-go 高速(OTG_HS)*** 增加了*第 57.4.2 节：USB OTG 引脚和内核信号*。 ***第 60 节：调试基础结构*** 更新了*第 60.5.8 节：微控制器调试单元 (DBGMCU)*。 在所有 DBGMCU 寄存器名称中： – 将 D1APB1 替换为 APB3 – 将 D2APB1 替换为 APB1 – 将 D2APB2 替换为 APB2 – 将 D3APB4 替换为 APB4 在*DBGMCU APB4 外设冻结寄存器 CPU (DBGMCU_APB4FZ1)* 一节 DBGMCU_APB4FZ1 中，WDGLSD2 位更改为保留。 删除了*表 576：DBGMCU 寄存器映射和复位值*中的保留寄存器。

表 585. 文档版本历史（续）

日期	版本	变更
2017 年 8 月 11 日	3	***第 2.3 节：内部 SRAM*** 增加了 *误码校正 (ECC)* 一节。 ***第 12 节：系统配置控制器 (SYSCFG)*** 在 *SYSCFG 补偿单元控制/状态寄存器 (SYSCFG_CCCSR)* 的 HSLV 位说明和 *SYSCFG 用户寄存器 17 (SYSCFG_UR17)* 的 IO_HSLV 中将 2.5 更改为 2.7 V。 ***第 3 节：嵌入式 Flash (FLASH)*** 在 *表 16：FLASH 寄存器映射和复位值* 中： – 更新了偏移地址 0x11C 的 FLASH_OPTSR_CUR。 – 更新了偏移地址 0x120 的 FLASH_OPTSR_PRG。 在 *FLASH 选项状态寄存器（当前值）(FLASH_OPTSR_CUR)* 和 *FLASH 选项状态寄存器（要编程的值）(FLASH_OPTSR_PRG)* 的 IO_HSLV 位说明中将 2.5 更改为 2.7 V。 ***第 6 节：电源控制 (PWR)*** 在 *图 27：D1、D2 和系统处于停止模式时的动态电压调节行为*、*图 28：D1、D2 和系统待机模式下的动态电压调节* 和 *图 29：D1 和 D2 处于 DStandby 模式且 D3 处于自动模式时的动态电压调节行为* 中，将 AIEC 替换为 EXTI，将等待 VDD11 替换为等待 VCORE。 更新了 *第 6.4.7 节：USB 稳压器* 更新了 *进入停止模式* 一节。 ***第 7 节：低功耗 D3 域*** 将 VDD11 替换为 VCORE。 ***第 8 节：复位和时钟控制 (RCC)*** 更新了 *表 50：内核时钟分配概述* 中 ADC1、2、3 的最大频率。 ***第 10 节：硬件信号量 (HSEM)*** 更新了整个章节，以便与产品功能对齐。 ***第 14 节：MDMA 控制器 (MDMA)*** 删除了 MDMA_CxISR 和 MDMA_CxIFCR 位名称中的迭代。 ***第 19 节：嵌套向量中断控制器*** 在 *表 129：NVIC* 中，将 usart4_gbl_it 更改为 uart4_gbl_it。 ***第 21 节：循环冗余校验计算单元 (CRC)*** 更新了 *第 21.2 节：CRC 主要特性* 的前两个功能。

表 585. 文档版本历史（续）

日期	版本	变更
2017 年 8 月 11 日	3（续）	***第 25 节：模数转换器 (ADC)*** *第 25.2 节：ADC 主要特性*：删除了 ADC 电源要求。 在整个章节内，将 V_{TS} 重命名为 V_{SENSE}。 更新了*图 125：ADC 框图*、*图 127：ADC1 连接功能*、*图 128：ADC2 连接功能*和*图 129：ADC3 连接功能*。 在*表 189：ADC 输入/输出引脚*中，V_{INN} 连接至 ADC_INN 而非 ADC_INN-1。 *I/O 模拟开关升压器*一节：升压器使能位重命名为 BOOTSE（而非 BOOTSEN），相应的寄存器重命名为 SYSCFG_PMCR（而非 SYSCFG_CFGR1）。 在*第 25.3.16 节：开始转换（ADSTART、JADSTART）*中增加了与软件触发选择相关的注释。 更新了*图 137：触发由主 ADC 和从 ADC 共享*、*图 192：温度传感器通道框图*、*图 193：VBAT 通道框图*和*图 194：VREFINT 通道框图*。 更新了*模拟看门狗*一节。 在*支持独立注入的交替模式*一节中更新了转换结束时采用中断方式的软件通知；更新了*图 179：连续转换模式下 1 通道的交替模式：双重 ADC 模式*和*图 180：单次转换模式下 1 通道的交替模式：双重 ADC 模式*。 在*双重 ADC 同步模式下的 DFSDM 模式*一节中，删除了温度传感器精度并将环境温度替换为结温。 *第 25.6.2 节：ADC x 通用控制寄存器 (ADCx_CCR)（x=12 或 3）*： – VBATEN 位说明通用 – TSEN 重命名为 VSENSEEN 且说明通用 – 更新了*表 202：DELAY 位与 ADC 分辨率的关系*。 删除了 ADCx_OR 寄存器，因为所有位均保留。 ADCx_LHTR1 重命名为 ADCx_HTR1。 将 ADCx_ISR 位访问类型更改为 rc_w1。 更新了所有 ADC 通用寄存器的迭代 ***第 26 节：数模转换器 (DAC)*** 更新了*表 206：DAC 输入/输出引脚*中的 VREF+ 范围。 更新了*第 26.3.5 节：DAC 转换*。 在*第 26.3.12 节：DAC 通道缓冲器校准*中增添了 CENx 置 1 时 ENx 限制的相关注释。 在*第 26.3.13 节：DAC 双通道转换（如果可用）*中，增添了访问 DHRxxxD 以供生成波形的情况。 将 DAC_M_ID 重命名为 DAC_SIDR 并将 M_ID 位重命名为 SID。

表 585. 文档版本历史（续）

日期	版本	变更
2017 年 8 月 11 日	3（续）	***第 26 节：数模转换器 (DAC)***（续） 增加了*第 26.5 节：DAC 中断* *第 26.6.1 节：DAC x 控制寄存器 (DACx_CR)（x=1 到 2）*： – TEN2 位：将 DACx_DHRy 替换为 DACx_DHR2 – TEN1 位：将 DACx_DHRy 替换为 DACx_DHR1 – TSELx 位：更新了触发选择/映射的相关信息 *第 26.6.16 节：DAC x 模式控制寄存器 (DACx_MCR)（x=1 到 2）*：增添了用于 MODEx 位说明的 ENx 位的相关注释。 *第 26.6.19 节：DAC x 采样和保持保持时间寄存器 (DACx_SHHR)（x=1 到 2）*：增添了 ENx=0 时的 THOLDx 修改的相关注释。 *第 26.6.20 节：DAC x 采样和保持刷新时间寄存器 (DACx_SHRR)（x=1 到 2）*：增添了 ENx=0 时的 TREFRESHx 修改的相关注释。 删除了 DAC_OR 寄存器。 ***第 29 节：运算放大器（OPAMP）*** 更新了*表 224：运算放大器连接情形*中的 OPAMP1_VINM 内部连接。 ***第 37 节：高分辨率定时器 (HRTIM)*** 更新了*表 288：外部事件映射和关联特性*中的注释 *1*。 ***第 44 节：系统窗口看门狗 (WWDG)*** 更正了用于计算 WWDG 周期的数学公式。 ***第 47 节：内部集成电路 (I2C) 接口*** 更新了*第 47.7.8 节：中断清零寄存器 (I2C_ICR)* 中的 NACKCF 位定义。 ***第 51 节：串行音频接口 (SAI)*** 在 SAI_xCR 寄存器中 SAIXEN 位重命名为 SAIEN。 更新了*图 644：PDM 典型连接和时序*，使此框图与数据和时钟线数无关。 更新了*图 645：详细的 PDM 接口模块框图*，其中增添了“n”和“p”索引（n 表示数据线数，p 表示麦克风对数）。 ***第 52 节：SPDIF 接收器接口 (SPDIFRX)*** 更新了*图 659：SPDIFRX 框图*。 ***第 59 节：HDMI-CEC 控制器 (HDMI-CEC)*** 已更新 – *第 59.2 节：HDMI-CEC 控制器主要特性* – 更新了*图 799：HDMI-CEC 框图*

表 585. 文档版本历史（续）

日期	版本	变更
2017 年 8 月 11 日	3（续）	***第 61 节：设备电子签名*** 更新了*第 61.1 节：唯一设备 ID 寄存器（96 位）*中的唯一设备 ID 寄存器基址。 更新了*第 61.2 节：Flash 大小*中的 Flash 大小基址。 在*第 61.3 节：封装数据寄存器*中，将封装数据寄存器说明替换为 SYSCFG_PKGR 的参考。

索引

A

ADC_JSQR 899
ADCx_AWD2CR 902
ADCx_AWD3CR 903
ADCx_CALFACT 906
ADCx_CALFACT2 907
ADCx_CCR 910
ADCx_CDR 913
ADCx_CDR2 913
ADCx_CFGR 885
ADCx_CFGR2 889
ADCx_CR 881
ADCx_CSR 908
ADCx_DIFSEL 905
ADCx_DR 898
ADCx_HTR1 894
ADCx_HTR2 904
ADCx_HTR3 905
ADCx_IER 879
ADCx_ISR 877
ADCx_JDRy 902
ADCx_LTR1 893
ADCx_LTR2 903
ADCx_LTR3 904
ADCx_OFRy 901
ADCx_PCSEL 893
ADCx_SMPR1 891
ADCx_SMPR2 892
ADCx_SQR1 895
ADCx_SQR2 896
ADCx_SQR3 897
ADCx_SQR4 898
AXI_COMP_ID_0 191
AXI_COMP_ID_1 192
AXI_COMP_ID_2 192
AXI_COMP_ID_3 193
AXI_INIx_FN_MOD 197
AXI_INIx_FN_MOD_AHB 196
AXI_INIx_FN_MOD2 195
AXI_INIx_READ_QOS 196
AXI_INIx_WRITE_QOS 197
AXI_PERIPH_ID_0 189
AXI_PERIPH_ID_1 190
AXI_PERIPH_ID_2 190
AXI_PERIPH_ID_3 191
AXI_PERIPH_ID_4 189
AXI_TARGx_FN_MOD 195
AXI_TARGx_FN_MOD_ISS_BM 193
AXI_TARGx_FN_MOD_LB 194
AXI_TARGx_FN_MOD2 194

C

CAN_TTGTP 2342
CCU_CCFG 2361
CCU_CREL 2361
CCU_CSTAT 2363
CCU_CWD 2364
CCU_IE 2365
CCU_IR 2364
CEC_CFGR 2763
CEC_CR 2762
CEC_IER 2767
CEC_ISR 2765
CEC_RXDR 2765
CEC_TXDR 2765
COMP_CFGR1 965
COMP_CFGR2 967
COMP_ICFR 964
COMP_OR 964
COMP_SR 963
CRC_CR 687
CRC_DR 686
CRC_IDR 686
CRC_INIT 688
CRC_POL 688
CRS_CFGR 428
CRS_CR 427
CRS_ICR 431
CRS_ISR 429
CRYP_CR 1183
CRYP_DIN 1185
CRYP_DMACR 1187
CRYP_DOUT 1186
CRYP_IMSCR 1187
CRYP_IV0RR 1193
CRYP_IV1LR 1193
CRYP_IV1RR 1194
CRYP_K0LR 1189
CRYP_K1LR 1190
CRYP_K1RR 1190
CRYP_K2LR 1191
CRYP_K2RR 1191
CRYP_K3LR 1192
CRYP_K3RR 1192
CRYP_MISR 1188
CRYP_RISR 1188

CRYP_SR 1185
CSTF_AUTHSTAT 2835
CSTF_CIDR0 2839
CSTF_CIDR1 2840
CSTF_CIDR2 2840
CSTF_CIDR3 2841
CSTF_CLAIMCLR 2834
CSTF_CLAIMSET 2833
CSTF_CTRL 2832
CSTF_DEVID 2836
CSTF_LAR 2834
CSTF_LSR 2835
CSTF_PIDR0 2837
CSTF_PIDR1 2837
CSTF_PIDR2 2838
CSTF_PIDR3 2838
CSTF_PIDR4 2839
CSTF_PRIORITY 2833
CSTF_TYPEID 2836
CTI_APPCLEAR 2815
CTI_APPPULSE 2816
CTI_APPSET 2815
CTI_AUTHSTAT 2823
CTI_CHINSTS 2819
CTI_CHOUTSTS 2819
CTI_CIDR0 2827
CTI_CIDR1 2827
CTI_CIDR2 2828
CTI_CIDR3 2828
CTI_CLAIMCLR 2821
CTI_CLAIMSET 2820
CTI_CONTROL 2814
CTI_DEVID 2823
CTI_DEVTYPE 2824
CTI_GATE 2820
CTI_INENx 2817
CTI_INTACK 2814
CTI_LAR 2822
CTI_LSR 2822
CTI_OUTENx 2817
CTI_PIDR0 2825
CTI_PIDR1 2825
CTI_PIDR2 2826
CTI_PIDR3 2826
CTI_PIDR4 2824
CTI_TRGISTS 2818
CTI_TRGOSTS 2818

D

DACx_CCR 944
DACx_CR 934
DACx_DHR12L1 938
DACx_DHR12L2 939
DACx_DHR12LD 941
DACx_DHR12R1 937
DACx_DHR12R2 939
DACx_DHR12RD 940
DACx_DHR8R1 938
DACx_DHR8R2 940
DACx_DHR8RD 941
DACx_DOR1 942
DACx_DOR2 942
DACx_MCR 944
DACx_SHHR 947
DACx_SHRR 947
DACx_SHSR1 946
DACx_SHSR2 946
DACx_SR 943
DACx_SWTRGR 937
DBGMCU_CR 2907
DBGMCU_D1APB1FZ1 2908
DBGMCU_D2APB1HFZ1 2910
DBGMCU_D2APB1LFZ1 2909
DBGMCU_D2APB2FZ1 2910
DBGMCU_D3APB4FZ1 2911
DBGMCU_IDC 2906
DCMI_CR 1060
DCMI_CWSIZE 1070
DCMI_CWSTRT 1070
DCMI_DR 1071
DCMI_ESCR 1068
DCMI_ESUR 1069
DCMI_ICR 1067
DCMI_IER 1065
DCMI_MIS 1066
DCMI_RIS 1064
DCMI_SR 1063
DFSDM_CHyAWSCDR 1020
DFSDM_CHyCFGR1 1017
DFSDM_CHyCFGR2 1019
DFSDM_CHyDATINR 1022
DFSDM_CHyWDATR 1021
DFSDM_FLTxAWCFR 1036
DFSDM_FLTxAWHTR 1034
DFSDM_FLTxAWLTR 1035
DFSDM_FLTxAWSR 1036
DFSDM_FLTxCNVTIMR 1038
DFSDM_FLTxCR1 1023
DFSDM_FLTxCR2 1025
DFSDM_FLTxEXMAX 1037
DFSDM_FLTxEXMIN 1037
DFSDM_FLTxFCR 1031
DFSDM_FLTxICR 1029

DFSDM_FLTxISR 1027
DFSDM_FLTxJCHGR 1030
DFSDM_FLTxJDATAR 1032
DFSDM_FLTxRDATAR 1033
DLYB_CFGR 799
DLYB_CR 798
DMA_CCRx 583
DMA_CMARx 586
DMA_CNDTRx 585
DMA_CPARx 585
DMA_HIFCR 564
DMA_HISR 563
DMA_IFCR 582
DMA_ISR 581
DMA_LIFCR 527, 564
DMA_LISR 525, 527, 562
DMA_SxCR 529, 531, 565
DMA_SxFCR 570
DMA_SxM0AR 569
DMA_SxM1AR 570
DMA_SxNDTR 534, 568
DMA_SxPAR 535-540, 569
DMA2D_AMTCR 639
DMA2D_BGCMAR 635
DMA2D_BGCOLR 634
DMA2D_BGMAR 628
DMA2D_BGOR 628
DMA2D_BGPFCCR 632
DMA2D_CR 623
DMA2D_FGCMAR 634
DMA2D_FGCOLR 631
DMA2D_FGMAR 627
DMA2D_FGOR 627
DMA2D_FGPFCCR 629
DMA2D_IFCR 626
DMA2D_ISR 625
DMA2D_LWR 639
DMA2D_NLR 638
DMA2D_OCOLR 636
DMA2D_OMAR 637
DMA2D_OOR 638
DMA2D_OPFCCR 635
DMAMUX1_CSR 602
DP_DPIDR 2781

E

ETF_AUTHSTAT 2856
ETF_BUFWM 2851
ETF_CBUFLVL 2850
ETF_CIDR0 2861
ETF_CIDR1 2861
ETF_CIDR2 2862
ETF_CIDR3 2862
ETF_CLAIMCLR 2855
ETF_CLAIMSET 2854
ETF_CTL 2848
ETF_DEVID 2857
ETF_DEVTYPE 2858
ETF_FFCR 2852
ETF_FFSR 2851
ETF_LAR 2855
ETF_LBUFLVL 2850
ETF_LSR 2856
ETF_MODE 2849
ETF_PIDR0 2859
ETF_PIDR1 2859
ETF_PIDR2 2860
ETF_PIDR3 2860
ETF_PIDR4 2858
ETF_PSCR 2854
ETF_RRD 2846
ETF_RRP 2847
ETF_RSZ 2845
ETF_RWD 2849
ETF_RWP 2847
ETF_STS 2845
ETF_TRG 2848
ETH_DMAC0CARxBR 2639
ETH_DMAC0CARxDR 2638
ETH_DMAC0RxCR 2628
ETH_DMAC0RxDLAR 2631
ETH_DMAC0RxDTPR 2632
ETH_DMAC0RxIWTR 2637
ETH_DMAC0RxRLR 2633
ETH_DMACiCATxBR 2638
ETH_DMACiCATxDR 2637
ETH_DMACiCR 2626
ETH_DMACiIER 2634
ETH_DMACiMFCR 2642
ETH_DMACiSR 2639
ETH_DMACiTxCR 2627
ETH_DMACiTxDLAR 2630
ETH_DMACiTxDTPR 2632
ETH_DMACiTxRLR 2633
ETH_DMADSR 2625
ETH_DMAISR 2624
ETH_DMAMR 2620
ETH_DMASBMR 2623
ETH_MAC1USTCR 2690
ETH_MACA0HR 2698
ETH_MACACR 2730
ETH_MACARPAR 2698
ETH_MACATSNR 2731

ETH_MACATSSR 2732
ETH_MACAxHR 2699
ETH_MACAxLR 2699
ETH_MACCR 2658
ETH_MACDR 2691
ETH_MACECR 2663
ETH_MACHT0R 2668
ETH_MACHT1R 2669
ETH_MACHWF1R 2692
ETH_MACHWF2R 2694
ETH_MACIER 2680
ETH_MACISR 2678
ETH_MACIVIR 2674
ETH_MACL3A00R 2714
ETH_MACL3A01R 2719
ETH_MACL3A10R 2714
ETH_MACL3A11R 2719
ETH_MACL3A20 2715
ETH_MACL3A21R 2720
ETH_MACL3A30 2715
ETH_MACL3A31R 2720
ETH_MACL3L4C0R 2711
ETH_MACL3L4C1R 2716
ETH_MACL4A0R 2713
ETH_MACL4A1R 2718
ETH_MACLCSR 2687
ETH_MACLETR 2689
ETH_MACLMIR 2741
ETH_MACLTCR 2689
ETH_MACMDIOAR 2695
ETH_MACMDIODR 2697
ETH_MACPCSR 2682
ETH_MACPFR 2664
ETH_MACPOCR 2738
ETH_MACPPSCR 2734
ETH_MACPPSIR 2737
ETH_MACPPSTTNR 2737
ETH_MACPPSTTSR 2736
ETH_MACPPSWR 2738
ETH_MACQ0TxFCR 2675
ETH_MACRWKPFR 2683
ETH_MACRxFCR 2677
ETH_MACRxTxSR 2681
ETH_MACSPI0R 2739
ETH_MACSPI1R 2740
ETH_MACSPI2R 2740
ETH_MACSSIR 2724
ETH_MACSTNR 2725
ETH_MACSTNUR 2726
ETH_MACSTSR 2725
ETH_MACSTSUR 2726
ETH_MACTSAR 2727
ETH_MACTSCR 2721
ETH_MACTSEACR 2733
ETH_MACTSECNR 2734
ETH_MACTSIACR 2732
ETH_MACTSICNR 2733
ETH_MACTSSR 2727
ETH_MACTxTSSNR 2729
ETH_MACTxTSSSR 2729
ETH_MACVHTR 2672
ETH_MACVIR 2673
ETH_MACVR 2690
ETH_MACVTR 2670
ETH_MACWTR 2667
ETH_MTLISR 2647
ETH_MTLOMR 2646
ETH_MTLQiICSR 2651
ETH_MTLRxQ0OMR 2652
ETH_MTLRxQiDR 2655
ETH_MTLRxQiMPOCR 2654
ETH_MTLTxQiDR 2649
ETH_MTLTxQiOMR 2647
ETH_MTLTxQiUR 2649
EXTI_EMR 674, 676
EXTI_FTSR 663, 667, 671
EXTI_IMR 664-665, 668-670, 672-675, 677
EXTI_PR 675, 677, 679
EXTI_RTSR 662, 666, 670
EXTI_SWIER 663, 668, 672

F

FDCAN_ TXBCIE 2332
FDCAN_CCCR 2299
FDCAN_CREL 2295
FDCAN_DBTP 2296
FDCAN_ECR 2305
FDCAN_ENDN 2296
FDCAN_GFC 2316
FDCAN_HPMS 2319
FDCAN_IE 2311
FDCAN_ILE 2315
FDCAN_ILS 2314
FDCAN_IR 2308
FDCAN_NBTP 2300
FDCAN_NDAT1 2320
FDCAN_NDAT2 2320
FDCAN_PSR 2306
FDCAN_RWD 2298
FDCAN_RXBC 2323
FDCAN_RXESC 2325
FDCAN_RXF0A 2322
FDCAN_RXF0C 2321

FDCAN_RXF0S 2321
FDCAN_RXF1A 2325
FDCAN_RXF1C 2323
FDCAN_RXF1S 2324
FDCAN_SIDFC 2317
FDCAN_TDCR 2308
FDCAN_TEST 2297
FDCAN_TOCC 2303
FDCAN_TOCV 2304
FDCAN_TSCC 2301
FDCAN_TSCV 2303
FDCAN_TTCPT 2353
FDCAN_TTCSM 2353
FDCAN_TTCTC 2352
FDCAN_TTIE 2346
FDCAN_TTILS 2348
FDCAN_TTIR 2344
FDCAN_TTLGT 2352
FDCAN_TTMLM 2338
FDCAN_TTOCF 2337
FDCAN_TTOCN 2340
FDCAN_TTOST 2349
FDCAN_TTRMC 2336
FDCAN_TTTMC 2335
FDCAN_TTTMK 2343
FDCAN_TTTS 2354
FDCAN_TURCF 2339
FDCAN_TURNA 2351
FDCAN_TXBAR 2330
FDCAN_TXBC 2326
FDCAN_TXBCF 2331
FDCAN_TXBCR 2330
FDCAN_TXBRP 2329
FDCAN_TXBTIE 2332
FDCAN_TXBTO 2331
FDCAN_TXEFA 2335
FDCAN_TXEFC 2333
FDCAN_TXEFS 2334
FDCAN_TXESC 2328
FDCAN_TXFQS 2327
FDCAN_XIDAM 2318
FDCAN_XIDFC 2318
FLASH_ACR 131
FLASH_BOOT7_CUR 151
FLASH_BOOT7_PRG 152
FLASH_CCR_A 139
FLASH_CCR_B 163
FLASH_CR_A 133
FLASH_CRCCR_A 152
FLASH_CRCCR_B 168
FLASH_CRCDATA_A 155
FLASH_CRCEADD_A 154
FLASH_CRCEADD_B 170
FLASH_CRCSADD_A 154
FLASH_CRCSADD_B 170
FLASH_ECC_FA_A 155, 163
FLASH_ECC_FA_B 171
FLASH_FKEYR_B 156
FLASH_FLASH_CR_B 156
FLASH_KEYR_A 132
FLASH_OPTCCR 146
FLASH_OPTCR 140
FLASH_OPTKEYR 133
FLASH_OPTSR_CUR 141
FLASH_OPTSR_PRG 144
FLASH_PRAR_CUR_A 147
FLASH_PRAR_CUR_B 164
FLASH_PRAR_PRG_A 148
FLASH_PRAR_PRG_B 165
FLASH_SCAR_CUR_A 149
FLASH_SCAR_CUR_B 166
FLASH_SCAR_PRG_A 150
FLASH_SCAR_PRG_B 167
FLASH_SR_A 136
FLASH_SR_B 160
FLASH_WPSn_CUR_A 150
FLASH_WPSn_CUR_B 167
FLASH_WPSn_PRG_A 151
FLASH_WPSn_PRG_B 168
FMC_BCR1..4 730
FMC_BTR1..4 733
FMC_BWTR1..4 736
FMC_ECCR 748
FMC_PATT 747
FMC_PCR 744
FMC_PMEM 746
FMC_SDCMR 762
FMC_SDCR1,2 759
FMC_SDRTR 763
FMC_SDSR 765
FMC_SDTR1,2 760
FMC_SR 745
FMPI2C_ISR 1843

G

GPIOx_AFRH 460
GPIOx_AFRL 459
GPIOx_BSRR 458
GPIOx_IDR 457
GPIOx_LCKR 458
GPIOx_MODER 455
GPIOx_ODR 457
GPIOx_OSPEEDR 456

GPIOx_OTYPER 455
GPIOx_PUPDR 456

H

HASH_CR 1199
HASH_CSRx 1219
HASH_DIN 1213
HASH_HR0 1215
HASH_HR1 . . . . . .1215, 1217
HASH_HR2 1216-1217
HASH_HR3 1216
HASH_HR4 1216
HASH_IMR 1217
HASH_SR 1218
HASH_STR 1214
HRTIM_ADC1R 1367
HRTIM_ADC2R 1368
HRTIM_ADC3R 1369
HRTIM_ADC4R 1371
HRTIM_BDMADR 1379
HRTIM_BDMUPR 1377
HRTIM_BDTxUPR 1378
HRTIM_BMCMPR 1362
HRTIM_BMCR 1358
HRTIM_BMPER 1362
HRTIM_BMTRGR 1360
HRTIM_CHPxR 1340
HRTIM_CMP1CxR 1324
HRTIM_CMP1xR 1323
HRTIM_CMP2xR 1324
HRTIM_CMP3xR 1325
HRTIM_CMP4xR 1325
HRTIM_CNTxR 1322
HRTIM_CPT1xCR 1341
HRTIM_CPT1xR 1326
HRTIM_CPT2xCR 1342
HRTIM_CPT2xR 1326
HRTIM_CR1 1349
HRTIM_CR2 1351
HRTIM_DTxR 1327
HRTIM_EECR1 1363
HRTIM_EECR2 1365
HRTIM_EECR3 1366
HRTIM_EEFxR1 1333
HRTIM_EEFxR2 1335
HRTIM_FLTINR1 1373
HRTIM_FLTINR2 1375
HRTIM_FLTxR 1348
HRTIM_ICR 1353
HRTIM_IER 1354
HRTIM_ISR 1352
HRTIM_MCMP1R 1309
HRTIM_MCMP2R 1310
HRTIM_MCMP3R 1310
HRTIM_MCMP4R 1311
HRTIM_MCNTR 1308
HRTIM_MCR 1301
HRTIM_MDIER 1306
HRTIM_MICR 1305
HRTIM_MISR 1304
HRTIM_MPER 1308
HRTIM_MREP 1309
HRTIM_ODISR 1356
HRTIM_ODSR 1357
HRTIM_OENR 1355
HRTIM_OUTxR 1345
HRTIM_PERxR 1322
HRTIM_REPxR 1323
HRTIM_RSTAR 1336
HRTIM_RSTBR 1338
HRTIM_RSTCR 1338
HRTIM_RSTDR 1339
HRTIM_RSTER 1339
HRTIM_RSTx1R 1331
HRTIM_RSTx2R 1332
HRTIM_SETx1R 1329
HRTIM_SETx2R 1331
HRTIM_TIMxCR 1312
HRTIM_TIMxDIER 1319
HRTIM_TIMxICR 1318
HRTIM_TIMxISR 1316
HSEM_CnICR 441
HSEM_CnIER 441
HSEM_CnISR 442
HSEM_CnMISR 442
HSEM_CR 443
HSEM_KEYR 443
HSEM_R0 - HSEM_R31 439
HSEM_RLR0 - HSEM_RLR31 440

I

I2C_CR1 1833
I2C_CR2 1836
I2C_ICR 1845
I2C_ISR 1843
I2C_OAR1 1839
I2C_OAR2 1840
I2C_PECR 1846
I2C_RXDR 1847
I2C_TIMEOUTR 1842
I2C_TIMINGR 1841
I2C_TXDR 1847

I2Cx_CR2 1836
IWDG_KR 1739
IWDG_PR 1740
IWDG_RLR 1741
IWDG_SR 1742
IWDG_WINR 1743

J

JPEG_CFR 1118
JPEG_CONFR0 1113
JPEG_CONFR1 1113
JPEG_CONFR2 1114
JPEG_CONFR3 1115
JPEG_CONFR4-7 1115
JPEG_CR 1116
JPEG_DIR 1119
JPEG_DOR 1119
JPEG_SR 1117

L

LPTIM_ARR 1724
LPTIM_CFGR 1720
LPTIM_CMP 1724
LPTIM_CNT 1725
LPTIM_CR 1723
LPTIM_ICR 1718
LPTIM_IER 1719
LPTIM_ISR 1717
LPUART_PRESC 1960
LTDC_AWCR 1086
LTDC_BCCR 1089
LTDC_BPCR 1085
LTDC_CDSR 1093
LTDC_CPSR 1093
LTDC_GCR 1087
LTDC_ICR 1091
LTDC_IER 1090
LTDC_ISR 1091
LTDC_LIPCR 1092
LTDC_LxBFCR 1099
LTDC_LxCACR 1098
LTDC_LxCFBAR 1101
LTDC_LxCFBLNR 1102
LTDC_LxCFBLR 1101
LTDC_LxCKCR 1097
LTDC_LxCLUTWR 1103
LTDC_LxCR 1094
LTDC_LxDCCR 1099
LTDC_LxPFCR 1097
LTDC_LxWHPCR 1095
LTDC_LxWVPCR 1096
LTDC_SRCR 1089
LTDC_SSCR 1085
LTDC_TWCR 1087

M

M7_DWT_CIDR0 2938
M7_DWT_CIDR1 2938
M7_DWT_CIDR2 2939
M7_DWT_CIDR3 2939
M7_DWT_COMPx 2933
M7_DWT_CPICNT 2930
M7_DWT_CTRL 2928
M7_DWT_CYCCNT 2930
M7_DWT_EXCCNT 2931
M7_DWT_FOLDCNT 2932
M7_DWT_FUNCTx 2934
M7_DWT_LSUCNT 2932
M7_DWT_MASKx 2933
M7_DWT_PCSR 2933
M7_DWT_PIDR0 2936
M7_DWT_PIDR1 2936
M7_DWT_PIDR2 2937
M7_DWT_PIDR3 2937
M7_DWT_PIDR4 2935
M7_DWT_SLPCNT 2931
M7_ETM_AUTHSTAT 2983
M7_ETM_CCCTL 2965
M7_ETM_CIDR0 2987
M7_ETM_CIDR1 2988
M7_ETM_CIDR2 2988
M7_ETM_CIDR3 2989
M7_ETM_CLAIMCLR 2981
M7_ETM_CLAIMSET 2980
M7_ETM_CNTRLDV 2968
M7_ETM_CONFIG 2960
M7_ETM_DEVARCH 2983
M7_ETM_DEVID 2984
M7_ETM_DEVTYPE 2984
M7_ETM_EVENTCTL0 2961
M7_ETM_EVENTCTL1 2962
M7_ETM_IDR0 2972
M7_ETM_IDR1 2973
M7_ETM_IDR10 2969
M7_ETM_IDR11 2970
M7_ETM_IDR12 2970
M7_ETM_IDR13 2971
M7_ETM_IDR2 2973
M7_ETM_IDR3 2974
M7_ETM_IDR4 2975
M7_ETM_IDR5 2976
M7_ETM_IDR8 2968

M7_ETM_IDR9 2969
M7_ETM_IMSPEC0 2971
M7_ETM_LAR 2982
M7_ETM_LSR 2982
M7_ETM_PDC 2979
M7_ETM_PDS 2980
M7_ETM_PIDR0 2985
M7_ETM_PIDR1 2986
M7_ETM_PIDR2 2986
M7_ETM_PIDR3 2987
M7_ETM_PIDR4 2985
M7_ETM_PRGCTL 2959
M7_ETM_PROCSEL 2959
M7_ETM_RSCTL2 2977
M7_ETM_RSCTL3 2977
M7_ETM_SSCC0 2978
M7_ETM_SSCS0 2978
M7_ETM_SSPCIC0 2979
M7_ETM_STALLCTL 2963
M7_ETM_STAT 2960
M7_ETM_SYNCP 2964
M7_ETM_TRACEID 2965
M7_ETM_TSCTL 2964
M7_ETM_VICTL 2966
M7_ETM_VIPCSSCTL 2967
M7_ETM_VISSCTL 2967
M7_FPB_CIDR0 2955
M7_FPB_CIDR1 2956
M7_FPB_CIDR2 2956
M7_FPB_CIDR3 2957
M7_FPB_COMPx 2952
M7_FPB_CTRL 2951
M7_FPB_PIDR0 2953
M7_FPB_PIDR1 2954
M7_FPB_PIDR2 2954
M7_FPB_PIDR3 2955
M7_FPB_PIDR4 2953
M7_FPB_REMAP 2952
M7_ITM_CIDR0 2948
M7_ITM_CIDR1 2948
M7_ITM_CIDR2 2949
M7_ITM_CIDR3 2949
M7_ITM_PIDR0 2946
M7_ITM_PIDR1 2946
M7_ITM_PIDR2 2947
M7_ITM_PIDR3 2947
M7_ITM_PIDR4 2945
M7_ITM_STIMx 2942
M7_ITM_TCR 2944
M7_ITM_TER 2943
M7_ITM_TPR 2943
M7_PILROM_CIDR0 2918
M7_PILROM_CIDR1 2919
M7_PILROM_CIDR2 2919
M7_PILROM_CIDR3 2920
M7_PILROM_MEMTYPE 2916
M7_PILROM_PIDR0 2917
M7_PILROM_PIDR1 2917
M7_PILROM_PIDR2 2917
M7_PILROM_PIDR3 2918
M7_PILROM_PIDR4 2916
M7_PPBROM_CIDR0 2924
M7_PPBROM_CIDR1 2925
M7_PPBROM_CIDR2 2925
M7_PPBROM_CIDR3 2926
M7_PPBROM_MEMTYPE 2921
M7_PPBROM_PIDR0 2922
M7_PPBROM_PIDR1 2923
M7_PPBROM_PIDR2 2923
M7_PPBROM_PIDR3 2924
M7_PPBROM_PIDR4 2922
MDIOS_CLRFR 2162
MDIOS_CRDFR 2160
MDIOS_CWRFR 2159
MDIOS_DINR0-MDIOS_DINR31 2163
MDIOS_DOUTR0-MDIOS_DOUTR31 2163
MDIOS_RDFR 2160
MDIOS_SR 2161
MMC_CONTROL 2700
MMC_RX_INTERRUPT 2701
MMC_RX_INTERRUPT_MASK 2704
MMC_TX_INTERRUPT 2702
MMC_TX_INTERRUPT_MASK 2705

N

NBTP 2300

O

OPAMP_OR 984
OPAMP1_CSR 981
OPAMP1_HSOTR 984
OPAMP1_OTR 983
OPAMP2_CSR 985
OPAMP2_HSOTR 987
OPAMP2_OTR 987
OTG_CID 2417
OTG_DAINT 2441
OTG_DAINTMSK 2442
OTG_DCFG 2435
OTG_DCTL 2436
OTG_DEACHINT 2445
OTG_DEACHINTMSK 2446
OTG_DIEPCTLx 2446

OTG_DIEPEMPMSK 2445
OTG_DIEPINTx 2452
OTG_DIEPMSK 2439
OTG_DIEPTSIZ0 2455
OTG_DIEPTSIZx 2455, 2457
OTG_DIEPTXF0 2413
OTG_DIEPTXFx 2422
OTG_DOEPCTL0 2448
OTG_DOEPCTLx 2450
OTG_DOEPINTx 2454
OTG_DOEPMSK 2440
OTG_DOEPTSIZ0 2456
OTG_DOEPTSIZx 2458
OTG_DSTS 2438
OTG_DTHRCTL 2444
OTG_DTXFSTSx 2458
OTG_DVBUSDIS 2443
OTG_DVBUSPULSE 2443
OTG_GAHBCFG 2398
OTG_GCCFG 2416
OTG_GI2CCTL 2415
OTG_GINTMSK 2408
OTG_GINTSTS 2404
OTG_GLPMCFG 2418
OTG_GOTGCTL 2394
OTG_GOTGINT 2397
OTG_GRSTCTL 2402
OTG_GRXFSIZ 2412
OTG_GRXSTSP 2411
OTG_GRXSTSR 2411
OTG_GUSBCFG 2399
OTG_HAINT 2426
OTG_HAINTMSK 2426
OTG_HCCHARx 2429
OTG_HCDMAx 2434
OTG_HCFG 2423
OTG_HCINTMSKx 2432
OTG_HCINTx 2431
OTG_HCSPLTx 2430
OTG_HCTSIZx 2433
OTG_HFIR 2424
OTG_HFNUM 2424
OTG_HNPTXFSIZ 2413
OTG_HNPTXSTS 2414
OTG_HPRT 2427
OTG_HPTXFSIZ 2422
OTG_HPTXSTS 2425
OTG_PCGCCTL 2459

P

purpose 1610
PWR_CSR 250, 254-255

Q

QUADSPI _PIR 794
QUADSPI _PSMAR 793
QUADSPI _PSMKR 793
QUADSPI_ABR 792
QUADSPI_AR 791
QUADSPI_CCR 789
QUADSPI_CR 783
QUADSPI_DCR 786
QUADSPI_DLR 788
QUADSPI_DR 792
QUADSPI_FCR 788
QUADSPI_LPTR 794
QUADSPI_SR 787

R

RCC_BDCR 353
RCC_CFGR 320
RCC_CICR 351
RCC_CIER 347
RCC_CIFR 349
RCC_CR 315
RCC_CRRCR 319
RCC_CSR 355
RCC_D1AHB1ENR 375
RCC_D1AHB1LPENR 392
RCC_D1AHB1RSTR 356
RCC_D1APB1ENR 382
RCC_D1APB1LPENR 400-401
RCC_D1APB1RSTR 361
RCC_D1CCIPR 340
RCC_D1CFGR 322
RCC_D2AHB1ENR 376
RCC_D2AHB1LPENR 394
RCC_D2AHB1RSTR 357
RCC_D2AHB2ENR 378
RCC_D2AHB2LPENR 396
RCC_D2AHB2RSTR 358
RCC_D2APB1HENR 386
RCC_D2APB1HLPENR 405
RCC_D2APB1HRSTR 365
RCC_D2APB1LENR 383
RCC_D2APB1LRSTR 362
RCC_D2APB2ENR 387
RCC_D2APB2LPENR 406
RCC_D2APB2RSTR 366
RCC_D2CCIP1R 341
RCC_D2CCIP2R 343
RCC_D2CFGR 324

RCC_D3AHB1ENR 380
RCC_D3AHB1LPENR 398
RCC_D3AHB1RSTR 359
RCC_D3AMR 370
RCC_D3APB1ENR 390
RCC_D3APB1LPENR 409
RCC_D3APB1RSTR 368
RCC_D3CCIPR 345
RCC_D3CFGR 325
RCC_GCR 369
RCC_ICSCR 318
RCC_PLL1DIVR 331
RCC_PLL1FRACR 333
RCC_PLL2DIVR 334
RCC_PLL2FRACR 336
RCC_PLL3DIVR 337
RCC_PLL3FRACR 339
RCC_PLLCFGR 328
RCC_PLLCKSELR 326
RCC_RSR 373
RNG_CR 1131
RNG_DR 1133
RNG_SR 1132
RTC_ALRMAR 1770
RTC_ALRMBR 1771
RTC_ALRMBSSR 1781
RTC_BKPxR 1782
RTC_CALR 1776
RTC_CR 1763
RTC_DR 1762
RTC_ISR 1766
RTC_OR 1782
RTC_PRER 1768
RTC_SHIFTR 1773
RTC_SSR 1772
RTC_TR 247-249, 252-253, 1761, 2158-2159
RTC_TSDR 1775
RTC_TSSSR 1775
RTC_TSTR 1774
RTC_WPR 1772
RTC_WUTR 1769
RX_ALIGNMENT_ERROR_PACKETS 2708
RX_CRC_ERROR_PACKETS 2707
RX_LPI_TRAN_CNTR 2710
RX_LPI_USEC_CNTR 2710
RX_UNICAST_PACKETS_GOOD 2708

S

SAI_ACLRFR 2078
SAI_ACR1 2068
SAI_ACR2 2071
SAI_ADR 2080
SAI_AFRCR 2073
SAI_AIM 2075
SAI_ASLOTR 2074
SAI_ASR 2076
SAI_BCLRFR 2078
SAI_BCR1 2068
SAI_BCR2 2071
SAI_BDR 2080
SAI_BFRCR 2073
SAI_BIM 2075
SAI_BSLOTR 2074
SAI_BSR 2076
SAI_GCR 2068
SAI_PDMCR 2080
SAI_PDMDLY 2081
SDMMC_ACKTIMER 2232
SDMMC_ARGR 2217
SDMMC_CLKCR 2215
SDMMC_CMDR 2217
SDMMC_DCNTR 2224
SDMMC_DCTRL 2222
SDMMC_DLENR 2221
SDMMC_DTIMER 2220
SDMMC_FIFOR 2233
SDMMC_ICR 2227
SDMMC_IDMABASE0R 2235
SDMMC_IDMABASE1R 2236
SDMMC_IDMABSIZER 2235
SDMMC_IDMACTRLR 2233
SDMMC_MASKR 2230
SDMMC_POWER 2214
SDMMC_RESPCMDR 2219
SDMMC_RESPxR 2219
SDMMC_STAR 2225
SMPMI_IER 2147
SPI_DR 2026-2028
SPI_I2SCFGR 2029
SPIx_SR 2021-2022, 2025
SWO_AUTHSTAT 2887
SWO_CIDR0 2891
SWO_CIDR1 2892
SWO_CIDR2 2892
SWO_CIDR3 2893
SWO_CLAIMCLR 2885
SWO_CLAIMSET 2885
SWO_CODR 2883
SWO_DEVID 2887
SWO_DEVTYPE 2888
SWO_FFSR 2884
SWO_LAR 2886
SWO_LSR 2886

SWO_PIDR0 2889
SWO_PIDR1 2890
SWO_PIDR2 2890
SWO_PIDR3 2891
SWO_PIDR4 2889
SWO_SPPR 2884
SWPMI_BRR 2144
SWPMI_CR 2143
SWPMI_ICR 2146
SWPMI_ISR 2145
SWPMI_OR 2150
SWPMI_RDR 2150
SWPMI_RFL 2149
SWPMI_TDR 2149
SWTF_AUTHSTAT 2898
SWTF_CIDR0 2902
SWTF_CIDR1 2903
SWTF_CIDR2 2903
SWTF_CIDR3 2904
SWTF_CLAIMCLR 2897
SWTF_CLAIMSET 2896
SWTF_CTRL 2895
SWTF_DEVID 2899
SWTF_LAR 2897
SWTF_LSR 2898
SWTF_PIDR0 2900
SWTF_PIDR1 2900
SWTF_PIDR2 2901
SWTF_PIDR3 2901
SWTF_PIDR4 2902
SWTF_PRIORITY 2896
SWTF_TYPEID 2899
SYSCFG_CCCR 470
SYSCFG_CCCSR 469
SYSCFG_CCVR 470
SYSCFG_EXTICR1 465
SYSCFG_EXTICR2 466
SYSCFG_EXTICR3 467
SYSCFG_EXTICR4 468
SYSCFG_PKGR 471
SYSCFG_PMCR 463
SYSCFG_UR0 471
SYSCFG_UR10 477
SYSCFG_UR11 477
SYSCFG_UR12 478
SYSCFG_UR13 478
SYSCFG_UR14 479
SYSCFG_UR15 480
SYSCFG_UR16 480
SYSCFG_UR17 481
SYSCFG_UR2 472
SYSCFG_UR3 473
SYSCFG_UR4 473
SYSCFG_UR5 474
SYSCFG_UR6 474
SYSCFG_UR7 475
SYSCFG_UR8 475
SYSCFG_UR9 476
SYSROM_CIDR0 2799
SYSROM_CIDR1 2799
SYSROM_CIDR2 2800
SYSROM_CIDR3 2800
SYSROM_MEMTYPE 2796
SYSROM_PIDR0 2797
SYSROM_PIDR1 2798
SYSROM_PIDR2 2798
SYSROM_PIDR3 2799
SYSROM_PIDR4 2797

T

TIM1_AF1 1476
TIM1_AF2 1478
TIM1_TISEL 1482
TIM12_TISEL 1596
TIM13_TISEL 1607
TIM14_TISEL 1607
TIM15_AF1 1664
TIM15_ARR 1658
TIM15_BDTR 1660
TIM15_CCER 1655
TIM15_CCMR1 1652
TIM15_CCR1 1659
TIM15_CCR2 1660
TIM15_CNT 1658
TIM15_CR1 1644
TIM15_CR2 1645
TIM15_DCR 1663
TIM15_DIER 1648
TIM15_DMAR 1663
TIM15_EGR 1651
TIM15_PSC 1658
TIM15_RCR 1659
TIM15_SMCR 1647
TIM15_SR 1649
TIM15_TISEL 1665
TIM16_AF1 1683
TIM16_TISEL 1684
TIM17_AF1 1685
TIM17_TISEL 1686
TIM2_AF1 1554
TIM2_TISEL 1555
TIM3_AF1 1554
TIM3_TISEL 1556

TIM5_AF1 1555
TIM5_TISEL 1557
TIM8_AF1 1479
TIM8_AF2 1480
TIM8_TISEL 1483
TIMx_ARR . . . 1466, 1550, 1595, 1606, 1678, 1700
TIMx_BDTR 1469, 1680
TIMx_CCER 1463, 1547, 1593, 1605, 1675
TIMx_CCMR1 . . 1458, 1542, 1590, 1603, 1673
TIMx_CCMR2 . . . . . .1461, 1545
TIMx_CCMR3 1474
TIMx_CCR1 . . . . 1467, 1550, 1595, 1607, 1679
TIMx_CCR2 . . . . . 1468, 1551, 1596
TIMx_CCR3 . . . . . .1468, 1551
TIMx_CCR4 . . . . . .1469, 1552
TIMx_CCR5 1475
TIMx_CCR6 1476
TIMx_CNT . . . 1466, 1549, 1594, 1606, 1678, 1699
TIMx_CR1 . . . 1447, 1533, 1584, 1599, 1668, 1696
TIMx_CR2 . . . . . .1448, 1535, 1669, 1698
TIMx_DCR 1472, 1553, 1682
TIMx_DIER . . . 1453, 1539, 1587, 1601, 1670, 1698
TIMx_DMAR 1473, 1553, 1683
TIMx_EGR . . . 1456, 1541, 1589, 1602, 1672, 1699
TIMx_PSC . . . 1466, 1549, 1594, 1606, 1678, 1700
TIMx_RCR 1467, 1679
TIMx_SMCR 1451, 1536, 1585
TIMx_SR 1455, 1540, 1588, 1601, 1671, 1699
TPIU_AUTHSTAT 2875
TPIU_CIDR0 2879
TPIU_CIDR1 2879
TPIU_CIDR2 2880
TPIU_CIDR3 2880
TPIU_CLAIMCLR 2873
TPIU_CLAIMSET 2873
TPIU_CURPSIZE 2866
TPIU_CURTPM 2869
TPIU_DEVID 2875
TPIU_DEVTYPE 2876
TPIU_FFCR 2871
TPIU_FFSR 2870
TPIU_FSCR 2872
TPIU_LAR 2874
TPIU_LSR 2874
TPIU_PIDR0 2877
TPIU_PIDR1 2877
TPIU_PIDR2 2878
TPIU_PIDR3 2878
TPIU_PIDR4 2876
TPIU_SUPPSIZE 2865
TPIU_SUPTPM 2868
TPIU_SUPTRGM 2866
TPIU_TPRCR 2870
TPIU_TRGCNT 2867
TPIU_TRGMULT 2867
TSG_CIDR0 2808
TSG_CIDR1 2808
TSG_CIDR2 2809
TSG_CIDR3 2809
TSG_CNTCR 2803
TSG_CNTCVL 2804
TSG_CNTCVU 2804
TSG_CNTFID0 2805
TSG_CNTSR 2803
TSG_PIDR1 2806
TSG_PIDR2 2807
TSG_PIDR3 2807
TSG_PIDR4 2805
TX_LPI_TRAN_CNTR 2709
TX_LPI_USEC_CNTR 2709
TX_MULTIPLE_COLLISION_GOOD_PACKETS 2706
TX_PACKET_COUNT_GOOD 2707
TX_SINGLE_COLLISION_GOOD_PACKETS 2706

U

USART_BRR 1905, 1952
USART_CR1 . . . . .356, 1893, 1945, 2109, 2112
USART_CR2 . . . . . .1896, 1948
USART_CR3 . . . . . .1900, 1949
USART_DR 1915, 1959, 2116-2120
USART_GTPR 1905-1906
USART_ISR 1908, 1953
USART_PRESC 1916
USART_SR 1907, 1953, 1957, 2113, 2115

V

VREFBUF_CCR 953
VREFBUF_CSR 952

W

WWDG_CFR 1734
WWDG_CR 1733
WWDG_SR 1735

重要通知 - 请仔细阅读

意法半导体公司及其子公司 （“ST”）保留随时对 ST 产品和 / 或本文档进行变更、更正、增强、修改和改进的权利，恕不另行通知。买方在订货之前应获取关于 ST 产品的最新信息。 ST 产品的销售依照订单确认时的相关 ST 销售条款。

买方自行负责对 ST 产品的选择和使用，ST 概不承担与应用协助或买方产品设计相关的任何责任。

ST 不对任何知识产权进行任何明示或默示的授权或许可。

转售的 ST 产品如有不同于此处提供的信息的规定，将导致 ST 针对该产品授予的任何保证失效。

ST 和 ST 徽标是 ST 的商标。所有其他产品或服务名称均为其各自所有者的财产。

本文档中的信息取代本文档所有早期版本中提供的信息。本文档的中文版本为英文版本的翻译件，仅供参考之用；若中文版本与英文版本有任何冲突或不一致，则以英文版本为准。

© 2017 STMicroelectronics - 保留所有权利